Integrated Decision Stations: A Framework for Dynamic Distributed Decision Making

Huabing Hu

A Thesis

in

The John Molson School of Business

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Administration (DS/MIS) at
Concordia University
Montreal, Quebec, Canada

March 2006

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-14368-1
Our file Notre référence
ISBN: 0-494-14368-1

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

Integrated Decision Stations: A Framework for Dynamic Distributed Decision Making

Huabing Hu

The increasing complexity of the current dynamic business environment makes some of centralized decision making either difficult or inefficient. While some attention has been paid to the dynamic distributed decision making (DDM), the research on decision support systems (DDS) supporting DDM is quite limited. In particular, how to model the structure of dynamic DDM and use information technology to support dynamic DDM remains as a critical question.

In this thesis, from information processing prospective, we propose a conceptual model, i.e. integrated decision stations (IDS), to address how to support dynamic DDM. Our model is based on the decomposition of decision criteria and information flows. To demonstrate the applicability of this framework, we apply it to lead time management and illustrate it with a prototype. It shows that our framework is powerful to describe various scenarios of DDM. In addition, we conduct a simulation study to investigate the impacts of two distinct features of IDS, i.e. real-time information and information coordination, on the decision performance. The results partially support that both information delay and information coordination have significant impacts on decision performance. The contribution of our study is that it provides both practitioners and academic researchers with a comprehensive framework to design effective DDM support systems and study their impacts on decision performance.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

The business world is becoming more complex due to increased market competition, overwhelming information, demanding and unfaithful customers, unstable political policies, and many the other uncertain factors. This complexity makes some of centralized decision making almost either impossible or inefficient. The key reason, as Simon (1960) states, is that a human has very limited information process capacity. Therefore, people must combine individual processing capacity to tackle complicated business problems. Consequently, decision making is distributed over empowered employees within an organization as well as over suppliers and/or customers across organizations, which is evidence by emerging business practice such as decentralization, customerization(Wind & Rangaswamy, 2001) and supply chain management (Davenport & Brooks, 2004).

This phenomenon, we calls it distributed decision making (DDM), is defined in this paper as a process involving one or more interdependent decision units that have compatible or incompatible decision goals. The decision units consist of either human beings or machines or both of them. This definition is very general in that it can potentially include every kind of decision making process except one-time individual decision making. In fact, group decision making, organizational group making, and negotiation are special cases of DDM.

Regarding decision support systems (DSS), though much research has been done on DSS with multiple decision makers, for instance, group decision support systems (GDSS) (Barkhi, Jacob, Pipino, & Pirkul, 1998; Jarvenpaa, Rao, & Huber, 1988), organizational decision support systems (ODSS) (Aggarwal & Mirani, 1995; King, Star, & George, 1990), and negotiation support system (NSS), there is a lack of comprehensive framework that can fully describe and support DDM. In particular, how to describe interrelated goals and coordination remains as a crucial question. In fact ,the classical DSS models, for example, Sprague's (1980) three-component DSS architecture does not articulate how to handle DDM.

On the other hand, our world is becoming more dynamic than ever. With the advent of advanced information technologies, information is changing at unprecedented speed. Decision makers will have to get timely information and response quickly in this dynamic environment. Otherwise they can hardly survive in this increasingly competitive world. To support this dynamic decision making, a decision support system should be able to capture required information in time and implement decision makers' decision immediately. Though demand for dynamic decision making support is urgent, the research on real-time DSS (Tseng & Gmytrasiewicz, 2002) or active, situated DSS (Vahidov & Kersten, 2004) is just emerging.

Therefore, this gap motivates us to develop a new DSS architecture that facilitates to build flexible DSS to support dynamic distributed decision making. On the one hand, this framework should help us to understand how decision is distributed over individual decision unit, in terms of the decomposition of decision criteria and related information flows. On the other hand, it should be acted as a solid foundation for the development of decision support systems, which is in alignment with organizational or inter-organizational distributed decision structure.

To fulfill our research objective, we will start our paper by reviewing two basic interrelated concepts: problem solving and decision making. Section 2 introduces a formal description of problem solving/decision making process. This formal description is further extended to describe distributed decision making. The classification of decision making systems and distributed decision making systems is discussed in section 3. Section 4 briefly reviews the conceptual foundation of decision support systems and their applicability. In section 5, we assess some concepts and development of distributed decision support systems. Finally, in section 6, we propose a framework, i.e. integrated decision stations (IDS) to support dynamic DDM.

Besides the above theoretical work, we build an IDS prototype to support lead time management. Section 7 introduces a business case and elaborates the architectures of the IDS basing upon our proposed framework. Moreover, in section 8, we describe our research model and depict how to evaluate this prototype system. Two distinct features of IDS, i.e. the timely information and coordination, are examined in the simulation. We present and discuss the results of evaluation in section 9. As a final point, in section 10, we conclude this thesis with our contributions, limitations, and future direction of our study.

2 FORMAL DESCRIPTION OF PROBLEM SOLVING/DECISION MAKING

2.1. The concept of problem solving and decision making

Given that our topic is about decision making, it is important to clarity the concept of decision making. Decision making is sometimes confused with problem solving. While some people treat decision making is part of problem solving, some others consider problem solving is in fact decision making. Simon and his associates (1987) stated that problem solving and decision making are two connected work consisting of four activities: fixing agendas, setting goals, designing actions, and evaluating and choosing among alternative actions. They called first three activities problem solving and the last one decision making. In this paper, we adopt this idea but we try to provide more formal and clearer description of problem solving and decision making process. The first step is to clarify these four terms: problem, problem solving, decision, and decision making.

In our daily language, a problem is regards as a difficulty or an unsolved question(Merriam-Webster, 2002a). It implies an undesired situation. Hence, more precisely, we define a problem as a gap between current situation and a desired situation. A situation consists of one or more observable factors, e.g. the sales revenue and market share of a business. People and organizations cannot control these factors directly, for example they cannot change sales revenue directly, though they can affect these factors in some indirect ways, for example they can control product development, pricing, promotion, and so on.

Basing on the above definition, we define problem solving as a process that searches for the courses of action to change the current situation to a desired situation. These actions are associated with the factors that an individual or organization can directly control, for example, product price as we aforementioned. We call each of these courses of action a solution. It is not unusual that there is more than one solution for a given problem. For example, the ways to increase sales could be improving brand awareness by intensive advertising or developing new products. It is also possible that there is no available solution for a problem at all.

Then, we need to define the term decision and the term decision making. A decision is a choice among alternatives. These alternatives are solutions to a problem as we defined above. Furthermore, we define decision making as a process that evaluates some or all of the recognized alternatives and selects a favored one according to a set of criteria (Wilson & Keil, 2001). This definition is consistent with Simon et al.'s (1987) description.

By these definitions, problem solving and decision making are two interrelated parts of Simon's (1960) three-phase model of problem solving/decision making process. Later, Simon (1977) extends it to a four-phase model. He defines these four stages:

> "The first phase of the decision-making process-searching the environment for conditions calling for decision–I shall call intelligence activity (borrowing the military meaning of intelligence). The second phase–inventing, developing, and analyzing possible courses of action–I shall call design activity. The third phase–selecting a particular course of action from those available–I shall call choice activity. The fourth phase, assessing past choices, I shall call review activity. " (Herbert A Simon, 1977, pp. 40-41)

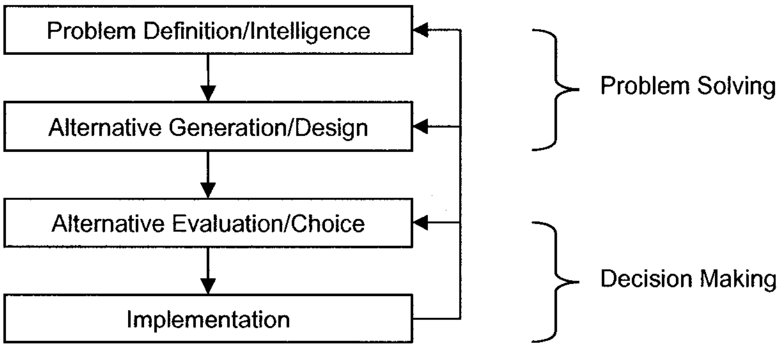


Figure 1. A generic model of problem solving process

A commonly used model of problem solving/decision making process, originated from Simon's (1960) Intelligence-Design-Choice model, is described in Figure 1. The first two activities, i.e. intelligence and design, are problem solving, and the remaining two are decision making activities. Because most of DSS research focus on the first three phases of problem solving/decision making process, the fourth phase, review phase in Simon's terms or implementation phase in our terms, is largely ignored. Many researchers (Vahidov & Kersten,

2004) argue that implementation phase is very important due to the improved the connectivity of information systems. For this reason, we include this phase in the decision making process. Furthermore, we should notice that, as Simon stated, “There are wheels within wheels. Nevertheless, the three major phases are often clearly discernible as the organizational decision process unfolds.” (Herbert A Simon, 1977, p. 43) They are parallel sub-processes and frequently overlap.

Therefore, strictly speaking, DSS, which are built upon Simon’s model, are used to support manager both in problem solving and in decision making. However, in practice, most of DSS today are developed to evaluate and choose alternatives (Eom, 2002), i.e. to support decision making process.

In the next section, we will provide a formal description of this process.

2.2. Problem solving/decision making process

After the above conceptual review, we can start our formal description of problem solving process. First, we formulize a problem as:

$$P = \{s \mid \Delta(s, s^*) \geq v, s \in S, s^* \in S\} \tag{1}$$

where s and s^* are current situation and desired situation of a problem environment respectively, $\Delta(s, s^*)$ is a function measuring the discrepancy between s and s^*, and v is the tolerance, and S represents the all set of various situations of a problem.

Therefore, if the deviation of current situation from the desired situation is beyond the tolerance, the current situation becomes a problem to managers. In the first stage of problem solving, i.e. the problem identification, we should identify and define *P* in terms of s and s^*, and then search for a course of action $a \in A$ to change the current situation into the desired situation.

However, to measure whether a change is desirable we need a preference structure $\succ$ on *C*. Here *C* is a set of criteria used to evaluate the change of the situation, for example product cost and advertising expense are two criteria.

$C(s_1, a_1) \succ C(s, \phi)$ means that taking action a_1 to change situation from s to s_1 is preferable to staying at situation s. $C(s_1, a_1) \succ C(s_1, a_2)$ means that action a_1 is preferable to action a_2 to achieve situation s_1. It is worthwhile to notice that a situation s_1 is preferable to the current situation s does not always create a problem. Only if the discrepancy between the current situation s and desired situation s^* is larger than the tolerance, the current situation becomes a problem and decision makers may implement a course of action a to change the current situation s into another situation s_1, where $\Delta(s_1, s^*) < v$, v is the tolerance.

The next step is to find a course of actions $a \in A$ that can eliminate or reduce the discrepancy between s and s^*, i.e. we need to find a mapping function $f_i : S \times A \to S$, where $f_i(s, a_i) = s_i$ and $\Delta(s_i, s^*) < v$. Hence, we denote the whole set of possible alternatives by:

$$F = \{ f_i : S \times A \to S \mid \Delta(f_i(s, a_i), s^*) < v, a_i \in A, i \in \square \}. \tag{2}$$

The third step is to evaluate these alternatives and choose a satisfactory solution. The most preferable situation and related action are

$$(s^*, a^*) = \arg \underset{a \in A, s \in S}{opt} \; E(C(s, a)) \tag{3}$$

where $\arg f(x) = x$, E = expectation function, and *opt* means to choose an alternative that optimizes decision maker's expectation value.

As a last step of problem solving/decision making process, we will implement our decision by selecting one alternative from the whole available set of alternatives:

$$f^* = f(s, a^*), \; f^* \in F. \tag{4}$$

Now, we summarize our formal description of problem solving/decision making process in Figure 2. In this Figure, the symbol "||" stands for parallel activities.

From now on, we denote this problem solving/decision making process by $PDP = PDP(S, F, C, A)$. Most of DSS in use today are developed to evaluate and choose alternatives(Eom, 2002). Hence, to emphasize decision making process rather than problem solving process, we simply call it a decision model and denoted it by $DM = DM(S, F, C, A)$.

```
PS_DM_Process ( ) //Problem Solving/Decision Making Process
{
 // Problem Identification/Definition/Intelligence
        $P = \{s \mid \Delta(s, s^*) > v, s \in S, s^* \in S\}$
  While $(P != \varnothing)$
    {
        // design alternatives
||      $F = \{f_i : S \times A \rightarrow S \mid \Delta(f_i(s, a_i), s^*) < v, a_i \in A\}$
        // evaluate alternatives
||      $(s^*, a^*) = \arg \underset{\alpha \in A, s \in S}{opt} E(C(s, a))$
        // Implementation
||      $f^* = f(s, a^*)$
        // Problem Identification
||      $P = \{s \mid \Delta(s, s^*) > v, s \in S, s^* \in S\}$
    }
}
```

Figure 2. Formal description of problem solving/decision making process

2.3. Distributed problem solving/decision making process

Simply stated, the process of distributed problem solving/decision making involves decomposition of a complex problem/decision into a set of subproblems/subdecisions. The decomposition itself is a problem that we have to solve to obtain an efficient and effective decision structure. In fact, this decomposition is a meta decision making process that determines “who can decide what”. Usually this is an unstructured decision involving organizational factors, political issues, social norms, and etc. For this reason, we withdraw ourselves from this discussion. We assume that the decision structure, i.e. “who decides what”, is determined already. The purpose of DDM support systems is to improve the effectiveness of each decision unit and facilitate the coordination between or among these decision units. The formal description of a decision unit will be discussed later.

Similar to our formal description of problem solving/decision making process in section 2, we define a distributed problem as a set of sub problems:

$$DP = \bigcup_{i=1}^{n} P_i = \{s_i \mid \Delta(s_i, s_i^*) \geq v_i, s_i \in S_i, s_i^* \in S_i\},\ n >= 2 \tag{5}$$

where s_i and s_i* are current situation and desirable situation of environment respectively, $\Delta(s_i, s_i*)$ is a function measuring the discrepancy between s_i and s_i*, and v_i is the tolerance with respect to the i^{th} problem solving/decision making unit (We will define it later). We denote the all set of various situations by S_i for this decision unit. Therefore, if the deviation of current situation from the desired situation is beyond the tolerance, the current situation becomes a problem to that unit. In the first stage of problem solving, we should identify and define P_i in terms of s_i and s_i*, and then search for a course of action $a_i \in A_i$ to change situations.

The design stage is similar, i.e.

$$F_i = \{f_j : S_i \times A_i \rightarrow S_i \mid \Delta(f_j(s_i, a_{ij}), s_i^*) < v_i, a_{ij} \in A_i\}, \text{ and } F = \bigcup_{i=1}^{n} F_i \tag{6}$$

The evaluation stage is a little bit different. When making its own decision, a decision unit perhaps needs to consider the information $I_i \subset I$ from the other units. This stage can be described as:

$$(s_i^*, a_i^*) = \arg \underset{\alpha_i \in A_i, s_i \in S_i}{opt} \quad E(C_i(s_i, a_i) \mid I_i) \tag{7}$$

where arg $f(x) = x$, E = mathematical expectation, *opt* means to choose an alternative that optimizes decision maker's expectation value, and I_i is information status regarding i^{th} decision unit.

Furthermore, in a distributed problem solving/decision making environment, one decision unit may assess the other decision units' preferences as well as its own. For this reason, we decompose preference structure of a decision unit into two parts: C^L stands for local preference structure, and C^E stands for external preference structures regarding the other decision units. C^E is not necessarily equal to the preference structure of the other decision units. Sometime it is just a guess or approximation. The decomposition is expressed as:

$$C_i = C_i(C_i^L, C_i^E) \tag{8}$$

Hence, the evaluation stage is now described as:

$$(s_i^*, a_i^*) = \arg \underset{\alpha_i \in A_i, s_i \in S_i}{opt} \; E(C_i[C_i^L(s_i, a_i), C_i^E(s_i, a_i)] \mid I_i) \tag{9}$$

The primary difference between distributed problem solving process and single centralized problem solving is that distributed problem solving requires coordination. From information processing perspective, we treat coordination as the exchange of information. Later we will discuss the types of required information. Here we simply denote it by I, which stands for information status. Hence we define coordination structure as

$CS_i : S_i \times A_i \to 2^{I_i}$, 2^{I_i} is the power set of I_i. (10)

After choosing an initial alternative, this unit may send some corresponding information to the other units. We denote it by

$$I_i^* = CS_i(s_i^*, a_i^*), I_i^* \subset I_i. \tag{11}$$

The last step, i.e. the implementation of the decision, is similar. The decision unit simply implements its most preferable alternative:

$$f_j^* = f(s_i, a_i^*) \tag{12}$$

Now we summarize the whole distributed problem solving/decision making process in Figure 3. We denote a distributed problem solving/decision making process by

$$DM = DM(S, F, C, A, CS)$$

```
Distributed_PS_DM_Process ( ) //Distributed Problem Solving/Decision Making Process
{
  // Problem Identification/Definition/Intelligence
  $DP = \bigcup_{i=1}^{n} P_i = \{s_i \mid \Delta(s_i, s_i^*) \ge v_i, s_i \in S_i, s_i^* \in S_i\}$
  While $(P_i != \varnothing)$
    {
        // Design alternatives
||      $F_i = \{f_j : S_i \times A_i \to S_i \mid \Delta(f_j(s_i, a_{ij}), s_i^*) < v_i, a_{ij} \in A_i\}$
        // Evaluate alternatives
||      $(s_i^*, a_i^*) = \arg \underset{a_i \in A_i, s_i \in S_i}{opt} \; E(C_i[C_i^L(s_i, a_i), C_i^E(s_i, a_i)] \mid I_i)$
        //Coordination
||      $I_i^* = CS_i(s_i^*, a_i^*)$     // $CS_i : S_i \times A_i \to 2^{I_i}$
        // Implementation
||      $f_j^* = f(s_i, a_i^*)$
        // Problem Identification
||      $DP = \bigcup_{i=1}^{n} P_i = \{s_i \mid \Delta(s_i, s_i^*) \ge v_i, s_i \in S_i, s_i^* \in S_i\}$
    }
}
```

Figure 3. Formal Description of distributed problem solving/decision making process

3 DECISION MAKING SYSTEMS (DMS) AND DISTRIBUTED DMS

3.1. Decision unit and decision making systems

A decision unit DU is an organization that carries out a decision making process DM. It involves at least one decision maker, who may be a human or not. We can formulate a decision unit as:

$$DU = \{DM, H, T, L\} \tag{13}$$

where DM = decision making process, H is a set of human decision maker(s), T representing time and L representing location, indicate when and where a decision making process takes place. Moreover, we define a decision making system (DMS) as a system including at least one decision unit, i.e.

$$DMS(DM, H, T, L) = \bigcup_{i=1}^{n} DU_i(DM_i, H_i, T_i, L_i)$$

$$DM = \bigcup_{i=1}^{n} DM_i, H = \bigcup_{i=1}^{n} H_i, T = \bigcup_{i=1}^{n} T_i, L = \bigcup_{i=1}^{n} L_i, \tag{14}$$

Using this notation, we can easily define various kinds of DMS according to whether DM(s) is distributed over people, or time, or space, or the combinations of them[1]. Table 1 lists some variants of decision making systems with corresponding decision support technologies.

DMS Type	Notation	Description	Decision Support
1	$DMS = DU(DM, H, T, L), H = \varnothing$	It includes only one decision unit without human involvement. It is an automatic decision making.	Transaction Processing Systems (TPS)

---

[1] Physical resources such as computer hardware and software are ignored in our paper, though they are distributable.

2	$DMS = DU(DM, H, T, L), \|H\| = 1$	It includes only one decision unit with only one human decision maker.	Individual DSS
3	$DMS = DU(DM, H, T, L), \|H\| > 1$	It includes only one decision unit with a group of people who share common goals	GDSS
4	$DMS = \bigcup_{i=1}^{n} DU_i(DM_i, H_i, T_i, L_i)$ $n >= 2, H = \varnothing$	It contains multiple decision making process without human involvement	Multi-Agent Systems
5	$DMS = \bigcup_{i=1}^{n} DU_i(DM_i, H_i, T_i, L_i)$ $n >= 2, \|H\| = 1$	It includes two more decision making processes that carried out by single decision maker.	Individual DSS with multiple model management
6	$DMS = \bigcup_{i=1}^{n} DU_i(DM_i, H_i, T_i, L_i)$ $n >= 2, \exists H_i \neq H_j$	It is group decision making with different goals	Organizational Decision Support Systems, Negotiation Support System

Table 1. Classification of decision making systems

Basing on this view, there are at least 2(One decision unit vs. two and more DUs) X 3(no human, one human, and two and more human decision makers) X 2 (the same or different location, i.e. geographic distribution) X 2 (the same time or different time, i.e. time distribution) = 24 possible scenarios of decision making systems. The basic reason to introduce this concept is to reveal the diversity of decision making system. Therefore, though it might be interesting, the full treatment of various scenarios and required decision support systems is beyond scope of this paper. In fact, no matter whether we distribute DM over people, time, or space, the result is that DM is divided among different decisions units. Hence, we ignore *H, T,* and *L* in our further discussion and focus on the division of DM and required coordination.

3.2. Distributed decision making systems

Distributed decision making systems (DDMS) are special cases of decision making systems (DMS) that consist of at least two decision units. Formally we defined a DDMS is:

$$DDMS = DMS(DM,H,T,L) = \bigcup_{i=1}^{n} DU_i(DM_i,H_i,T_i,L_i)$$

$$DM = \bigcup_{i=1}^{n} DM_i, H = \bigcup_{i=1}^{n} H_i, T = \bigcup_{i=1}^{n} T_i, L = \bigcup_{i=1}^{n} L_i,$$

Where n >=2, and $\exists DM_i \neq DM_j$, when $i \neq j$. (15)

According to this definition, we can find that the only difference between DMS and DDMS is that DDMS do not include DMS that have only one decision unit.

Basing on our definition, DDMS describes how individual decisions are coordinated. Thus an easy way to classify DDMS is to categorize them by the types of decision and coordination. The definition of coordination is numerous. However, because our purpose is to include in our framework as many types of DDMS as possible, coordinate is broadly defined as “to bring into a common action, movement, or condition” (Merriam-Webster, 2002b). Furthermore, we can define three broad types of coordination. The first one is *No Coordination* referring to the systems in which each organization does not interact with each other. They do not share any necessary information and thus their organizational systems are stand-alone. The second type of coordination, Cooperation, represents the systems in which organizations interact with each other to achieve a common goal or conflict-free goals. The last type of coordination, Competitive, enables organizations with competitive goals to achieve compromise or a conflict resolution.

Furthermore, Thompson’s (1967) concept of the relationship between technology and organizational structure and his “interdependence” view of the organization is a useful scheme to classify information systems. Basing on it, Kumar and van Dissel (1996) propose a typology for Inter-organizational systems (IOS), which is very relevant to our study. The simplest one is pooled information resource IOS, which corresponds to pooled interdependency relationship. By the aid of this type of IOS, a group of organizations can share common IS/IT resources and they have minimal potential conflicts. A good example is electronic markets. The second type of IOS is value/supply-chain IOS. It corresponds to sequential interdependence relationship where output from one organization is input to another organization. EDI is a typical application of this type of IOS. Most of current SCMS, such as collaborative demand management, belong to this category. The most complicated type of IOS is networked IOS by which organizations often obtains input from and delivers output to others interactively. Their interdependencies are reciprocal and might lead to a high level of potential conflicts.

As a result, we can get six types of inter-organizational relationship: 1) *Independent*, which represents that each decision unit makes its own decision without considering the other decision units. 2) *Cooperation with Pooled interdependence*, which represents that two and more decision units have non-conflict goals and share common IS/IT resources. 3) *Cooperation with Sequential interdependence,* which means that one decision unit makes a part of a decision and then passes it to the other decision units. 4) *Cooperation with Reciprocal interdependence,* which represents that two and more decision units have non-conflict goals and they involve frequent communication. 5) *Competition with Pooled interdependence*, which refers that two and more decision units have conflict goals but share common IS/IT resources. 6) *Competition with Reciprocal interdependence,* which usually refers to negotiation consisting of conflict goals between or among various decision units.

Table 2 provides some examples of distributed decision making systems. It is a table with one dimension representing the type of inter-organizational coordination and the other dimension representing the structure level of a decision. The meaning of structured, semistructured, and unstructured decision will be discussed in the following sections. This table provides a overview picture where how different kinds of information systems fit each other basing on our distributed decision making concept. For example, regarding structured decision, the typical information systems are those facilitating automatic decision making, for example, batch processing and EDI applications.

Types of Decision	**Distributed Decision Making (DDM) Systems**					
	No Coordination	Cooperation			Competition	
	Independent organizational systems	**IOS** (Inter-organizational systems)			(the other systems)	
		Pooled Interdependence	Sequential Interdependence	Reciprocal Interdependence	Pooled Interdependence	Reciprocal Interdependence
Unstructured Decision	Communication technology	Communication Technology (Telephone, Fax, e-mail, video conference, groupware, etc.)			Communication Technology	
Semi-structured Decision	Individual DSS	Group DSS, Interorganizational DSS (IODSS)	Supply Chain planning	GDSS, IODSS	Auction support systems	Negotiation support systems
Structured Decision	TPS	Shared Databases, Electronic Markets	EDI, XML	Automatic data exchange (CAD)	Automatic auction systems	Automatic negotiation

Table 2. DDM and inter-organizational coordination

4 DECISION SUPPORT SYSTEMS

Before we can discuss distributed decision making and support systems, we should examine the development of traditional decision support systems (DSS). This section is devoted to examine the conceptual foundation of DSS and identify how DSS can support decision making process. Particularly, we concentrate on discussion of the underlying meaning of semistructured problems/decisions.

4.1. The conceptual foundation of DSS

The concept of (DSS) was conceived in the mid 1960s (Power, 2003) and first articulated by Morton (Scott Morton, 1971) under the term "management decision systems". During that period, DSS had developed from two primary research areas: the theoretical research on organizational decision making headed by H.A. Simon at Carnegie Institute of Technology during the late 1950s and the early 1960s and the technical studies of interactive computer systems conducted at the Massachusetts Institute of Technology (Keen & Morton, 1978).

Basing upon Anthony's (1965) classifications of management planning and control activities and Simon's (1960) discussion of decision types, In their seminal paper (1971), Gorry and Morton built a two-dimensional model for management information systems, which is shown in Table 3. One dimension comes from Anthony's analysis of planning and control system. He argued that management activities can be divided into three levels and each level requires a distinct system. The highest level is strategic planning, which is "the process of deciding on objectives of the organization, on changes in these objectives, on the resources used to attain these objectives, and on the polices that are to govern the acquisition, use and disposition of these resources" (p. 24). Because the strategic planning process is often complex and problems in strategic planning are irregular, it is hard to define rules to conduct and evaluate this process. The second level is management control, which is "the process by which managers assure that resources are obtained and used effectively and efficiently in the accomplishment of the organization's objectives" (p. 27). Anthony emphasized that management control follows the objectives and policies established in the strategic planning process. The last level is operational control, which is "the process of assuring that specific tasks are carried out effectively and efficiently" (p. 69). In the operational control, those specific tasks are well defined. However, there are no clear-cut boundaries among these levels.

Decision Type	Management Activity			Support Needed
	Operational Control	Management Control	Strategic Planning	
Structured	1	4	7	EDP
Semistructured	2	5	8	DSS
Unstructured	3	6	9	Human

Table 3. A Framework for information systems (Adapted from Keen and Morton (1978)).

The second dimension of Gorry and Morton's model is derived from Simon's (1960) work on the processes of management decision and his distinction between programmed and nonprogrammed decisions. Programmed or structured decisions are repetitive and routine. These decisions can be made according to predefined procedures. On the contrary, nonprogrammed or unstructured decisions are novel and irregular, and there is no specific procedure to handle them. Gorry and Morton proposed a new type of decisions, i.e. semi-structured decisions, by incorporating Simon's three phases of decision making process: Intelligence, Design, and Choice. They defined that a structure decision is one in which all three phases are structured and an unstructured decision is one in which none of three phases is structured. The remaining are semistructured decisions.

Combining these two dimensions, Gorry and Morton divided decision into nine categories. Further, Keen and Morton (1978) claimed that structured decisions can be supported by clerical or electronic data processing (EDP), semistructured ones can be supported by DSS, and unstructured ones can support only by human intuition.

In correspondence with the above conceptual foundation, in this paper we define a DSS is an interactive human-computer system that support, rather than replace, decision makers to solve semistructured problem effectively in their decision processes.

4.2. What is a semistructured problem/decision?

It is important to clarify what we mean by a semistructured problem/decision. According to our formal description aforementioned, a problem solving/decision making process (PDP) can be stated as $PDP = PDP(S, F, C, A)$. We call a PDP semistructured if at least one of P, F, C, A is not well-defined.

More specifically, $P = \{s \mid \Delta(s, s^*) \geq v, s \in S, s^* \in S\}$ is not well defined means:

1) The factors consisting of S are not completely identified, i.e. some factors are neglected. For example, when we measure the performance of a business, many intangible factors, especially those social factors, are ignored.

2) Some of identified factors are not measurable in a quantitative way. For example, it is hard to measure the reputation of a business.

3) How to measure the discrepancy between two different situations is ambiguous. For instance, moving a business from number two to number one in market share is certainly different from moving a business from number 100 to number 99.

4) The tolerance level v is ambiguous. For example, if sales expense of this year is 9,999 higher than the budget, is it a problem? Or it must be equal to or greater than 10,000, and then it become a problem. Our point is that tolerance level v is sometimes fuzzy in nature.

Second, $F = \{f_i : S \times A \to S \mid \Delta(f_i(s, a_i), s^*) < v, a_i \in A, i \in I\}$ is not well defined means F is empty. We cannot find a feasible and clear course of action, i.e. path, along which we can move the current situation closer to the desired situation till we can tolerate.

In the third stage, i.e. evaluation stage $(s^*, a^*) = \arg \underset{\alpha \in A, s \in S}{opt} E(C(s, a))$, C is not well defined means:

1) The components of C and A are not fully identified. For example, some criteria or actions might be ignored.

Stage	Description
$P = \{s \mid \Delta(s, s^*) \geq v, s \in S,$ $s^* \in S\}$	▪ The composition of s is not fully identified ▪ Identified but not measurable ▪ Δ is ambiguous, i.e. we do not know how to measure the discrepancy ▪ We are not certain the exact level of tolerance v
$F = \{f_i : S \times A \to S \mid$ $\Delta(f_i(s, a_i), s^*) < v, a_i \in A, i \in I\}$	▪ No feasible solution, i.e. F is empty
$(s^*, a^*) = \arg \underset{a \in A, s \in S}{opt} E(C(s, a))$	▪ The composition of C and A are not completely identified ▪ The preference structure $\succ$ on C is ambiguous. For example, it is not transitive. $C(s_1, a_1) \succ C(s_2, a_2)$, $C(s_2, a_2) \succ C(s_3, a_3)$ but $C(s_3, a_3) \succ C(s_1, a_1)$. ▪ Not comparable.

Table 4. The nature of semistructured problem/decision

2) The preference structure $\succ$ on C is ambiguous. Ideally, $\succ$ is transitive. For example, if $C(s_1, a_1) \succ C(s_2, a_2)$ and $C(s_2, a_2) \succ C(s_3, a_3)$, then we should get $C(s_1, a_1) \succ C(s_3, a_3)$. However, if we get $C(s_3, a_3) \succ C(s_1, a_1)$, then we might think $\succ$ is inconsistent or ambiguous. Furthermore, it is ideal that each pairs of elements of C are comparable. Otherwise, we cannot

make full evaluation of all possible alternatives. However, this inconsistency and ambiguity frequently appear in real human problem solving/decision making process.

Table 4 provides a summary of discussion on semistructured decisions.

4.3. How DSS support semistructured problem solving/decision making?

Sprague's (1980) framework, shown in Figure 4, is a classical model for the development of DSS. In his model, DSS consist of three key components: Database Management Software (DBMS), Model Management Software (MBMS), and the Dialog Generation and Management Software (DGMS). The purpose of DGMS is to facilitate the interaction between the user and the systems. DBMS is used to capture and store data from multiple external and internal data sources that are relevant to problem domain. It also enables the user to manipulate data and data sources conveniently. While DBMS is necessary to support the user to identify and represent problems, MBMS is more important because it help the user to develop and evaluate alternatives that ultimately lead to problem solving.

Though DSS are supposed to support the whole problem solving process as we have described above, in practice, most of DSS developed are focused on the evaluation of alternatives (Eom, 2002). That is decision making process by our definition. Similarly, in our paper, we focus on decision making process rather than the whole problem solving process. Therefore, we will not discuss how to obtain a set of alternatives, i.e. F in our formal description of problem solving process. In fact, this stage is largely supported by management sciences (MS)/operations research (OR) (Sprague, 1980). Instead, we assume that a set of models, i.e. F for this specific problem have been developed to generate decision alternatives and thus the related action A is available. Thus a problem solving process is simplified as a decision making process by which we denote $DM = DM(S, F, C, A)$

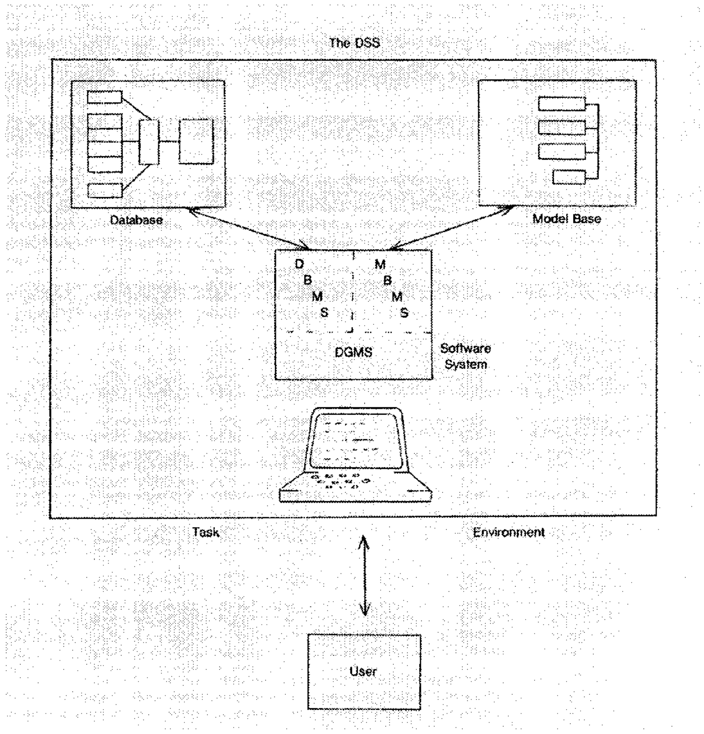


Figure 4. Traditional DSS architecture (Sprague, 1980)

Finally, we have to emphasize that DSS are used to support semistructured decisions. According to our definition and notation, a semistructured decision is a decision that is carried out through a decision making process $DM = DM(S, F, C, A)$, where some of *S, F, C, and A* are not well identified and/or quantified.

Phases of Decision Making Process	Sprague's (1980) DSS Framework
Problem Identification/Intelligence $P = \{s \mid \Delta(s, s^*) \geq v, s \in S, s^* \in S\}$	DBMS: data capture and extraction; data representation and manipulation. It helps to understand the current situation s and the desired situation s^*.
Alternative Generation/Design $F = \{f_i : S \times A \to S \mid \Delta(f_i(s, a_i), s^*) < v,$ $a_i \in A, i \in \square\}$	MBMS: decision models management; communication with DBMS to get required data. It supports to find alternatives/solutions to achieve desired situation.

Alternative Evaluation/Choice $(s^*, a^*) = \arg \underset{\alpha \in A, s \in S}{opt} E(C(s,a))$	DGMS: flexible human-machine interface. It supports the efficient interactions between the users and DBMS and MBMS and thus helps them to evaluate alternatives.
Implementation $f^* = f(s, a^*)$	Not Available.

Table 5. How traditional DSS support decision making

Table 5 presents how Sprague's (1980) three-component DSS architecture support decision making process. Note that Sprague's framework does not explicitly describe how to support the implementation of a decision. In addition, it does not discuss how to handle distributed decision making, which is the focus of this paper.

5 DISTRIBUTED DECISION MAKING AND SUPPORT SYSTEMS

The major part of our framework for dynamic distributed decsion support system comes from operations research (OR), decision support systems (DSS) and distributed artificial intelligence (DAI) research. In this section, first we will investigate the concept of distributed decision support systems (DDSS) by comparing it with the other group support systems. Then we will set our own definition of DDSS by focusing on distributed decision making (DDM) process. Last we will briefly review current development of DSS for single user and examine the necessity and possibility to extend it to DDM area.

5.1. Traditional concepts of distributed decision support systems

Eom's (2002) review on DSS research dated from 1970 to 1999 shows that distributed decision making is largely ignored by DSS researchers, though there are quite a lot research on some similar systems like group decision support systems (GDSS), organizational decision support systems (ODSS) and negotiation support systems (NSS). In the discussion of past, present, and future of decision support technology, Shim et al. (2002) pay little attention to DDSS when they evaluate collaborative support systems. This lack of research on DDSS could be attributed to the ambiguity of DDSS concept itself.

There are indeed some research papers on DDSS. However, they do not provide a clear definition of DDSS and thus are mixed up with the other group support systems. The term DDSS, originally introduced by Scher (1981), is referred to as a conference-based system in an organization. Rathwell & Burns (1985) define a DDM system as a cooperative network facilitating communication and conflict resolution among equal decision makers. Later, a model of a DDM and a prototype is built to demonstrate their non-hierarchical concept of DDSS (Burns, Rathwell, & Thomas, 1987). (Because their discussion is in the context of a single organization, the competition among decision makers was not mentioned.)

Swanson (1990) treats DDSS as one perspective of organizational DSS. He further identifies the domain of DDSS, which is characterized by semi-structured decision model and semidetermined decision criteria, by comparing distributed decision support with distributed computing and distributed communication. Basing on Swanson's definition of ODSS(Swanson, 1990), Chung et al. (1993) states that distributed decision support system (DDSS), conceived as a network of decision making nodes in an organization, is a subset of ODSS. In addition, they divide categorize DDSS into two categories: Rigid DDSS and Flexible DDSS.

Some authors discuss DDSS in a more technical way. Chi and Turban (1995) describe how to use agent technology to distribute multiple resources, such as knowledge base and DBMS, to support executive decision making. Jeusfeld and Bui (1997) propose a script language to allow construction of DSS from DSS components stored on various Internet sites. They do not consider distributed decision making process. Rather they focus on the distribution of physical components of DSS. Ju, Ling, and Norman (2000) construct an agent-based architecture of DDSS and discuss how these agents coordinate to support decision making. However, they are more interested in how these agents coordinate to facilitate automatic problem solving rather than what to coordinate in distributed decision making process. Gachet (2002) discuss a decentralized technical architecture for distributed DSS with little mention of distributed decision making. Gachet and Haettenschwiler (2003) conduct a case study to identify the impact of single-user DSS vs. distributed DSS in the collective decision making process. Similarly, they focus on distributed software architectures.

Some of the above views, labeled as DDSS, are confined in the context of one single organization. Hence, it is not strange that they treat DDSS as a special case of ODSS (Aggarwal & Mirani, 1995). However, with the development of advanced information technologies, such as Internet and World Wide Web, as well as new business practice, such as supply chain management (Davenport & Brooks, 2004) and customer integration(Wind & Rangaswamy, 2001), the scope of DDSS is extended from within one organization to all the stakeholders of an organization. For this reason, the concept of DDSS support systems should also be broadened to accommodate the new reality. Additionally, though some of these papers discuss distributed decision making (DDM) process, they are too general to identify the distinctive nature of DDM. We will examine this issue in details later.

Base on this review, we find that the traditional view of DDSS is either too narrow to cover some important aspects of distributed decision making process, for example, the peer-to-peer relationship between two decision units, or too technical to consider the nature of distributed decision making process, i.e. the interdependence of preference structure of each decision units.

5.2. Emerging concepts of distributed decision making

Distributed decision making can be automated. Multiagent systems (MAS) is emerging as a promising approach to solve such distributed problem(Weiss, 1999). The basic property of an agent is autonomy, defined by Wooldridge and Jennings (1995) as "to operate without the direct intervention of humans or others" (p. 4). MAS aim at solving complicated problems automatically by the aid of each autonomous agent. However, from the point of view of decision making support, MAS, as a new generation of distributed computing, are at their best to handle fully structured decision model and fully determined decision criteria (Swanson, 1990).

In contrast to MAS research, which almost completely remove human from decision making process, cognitive research on DDM is primarily focused on such as factors as forms of coordination, characteristics of tasks, and organizational structure. It is stated that "designing computer-based systems for cooperative work environments such as organizational decision making is like writing on water" (p. 76) (Rasmussen, Brehmer, & Leplat, 1991) due to that DDM support systems and organizational decision structure influence each other and thus will not work as designed. For this reason, cognitive research on DDM provides little hints on DDM support systems.

To synergize diverse research on DDM from different disciplines like computer sciences, economics, organizational theory, psychology, and many others, Schneeweiss (2003b) proposes a unified model of distributed decision making (DDM) by synthesizing different disciplines' research on DDM. He characterizes "DDM as the design and coordination of connected decisions" (P. 580). Decision making can be distributed over different people, different time periods, and different locations, which might lead to hierarchical or asymmetric dependences. In this model, DDM can almost include every decision scenarios except one-time single-person decisions.

Before we go further, we simply discuss why decision making has to be distributed over various decision making units (DUs). There are at least two reasons: One is that the decision itself is so complex that no single DU can handle it completely and timely. It is a matter of division of labor. The other reason is that the distribution of decision authority is determined by established power structure, a political issue. No matter for what reasons, a coordination mechanism is required. Hence, in this paper, instead of attempting to answer why decision is distributed, we examine what decision elements are distributed and to what extent, and how different ways of decision distributions affect decision performance.

Schneeweiss' (2003b) DDM model can provide a more comprehensive view on decision making. In his paper, a decision model consists of a set of decision criteria C, action space A, and status of information I. In its simplest form, decision making is distributed over two DUs. If these two DUs have an identical decision model, i.e. sharing same decision criteria, action space, and information, we may say that they are in perfect cooperation. On the contrary, if their decision models are completely different, they are in fact isolated from each other. Usually the relationship of two DUs is located somewhere between these two extreme situations. In this paper, we adopt his model as a reference point.

5.3. Distributed decision support systems, distribute what over what?

As we see in the above papers, many systems call themselves distributed decision support systems (DDSS). However, this term is quite ambiguous in that we do not know what DDSS distribute and over what. On the one hand, it could mean that the components of DSS are distributed or decentralized. Jeusfeld and Bui (1997) describe a case that DSS components such as models are stored on the Internet and accessible to various users. Gachet (2002) conceive of DDSS as a decentralized network of hardware and software to facilitate common decision support, available at any time and from any where. On the other hand, DDSS could mean the systems supporting DDM, i.e. distribution of decision making authority. The later case is our concern in this paper.

Hence, we define a DDSS is an interactive human-computer system that support, rather than replace, decision makers to solve semistructured problem effectively in the distributed decision making process. We have discussed what we mean by semistructured problem and distributed decision making process in the previous section.

$$DM = DM(S, F, C^L, C^E, A, CS), \tag{16}$$

where S: Situation space, F: Alternatives

C^L: Local preference structure (Decision Criteria)

C^E: External (the others') preference structure

A: Action space, CS: Coordination structure

5.4. Situated decision support systems

Vahidov and Kersten (2004) argued that traditional DSS research ignored implementation and monitoring phases of decision making process. The systems built on the traditional DSS concept, i.e. Simon's (1960) intelligence-design-choice model, are disconnected from their problem domains. Inspired by software agent technologies and research on active DSS, they proposed a new framework of DSS named as decision station architecture, which can implement decisions as well as monitor the change of problem environment by the aid of sensors and effectors.

Sensors and effectors are key components that differentiate situated DSS from traditional "standalone" DSS. They can be equipped with passive capabilities, such as connecting, transforming, querying and alerting users, and/or active capabilities, such as adapting and planning. The architecture of a decision station (DS) is reproduced in Figure 5.

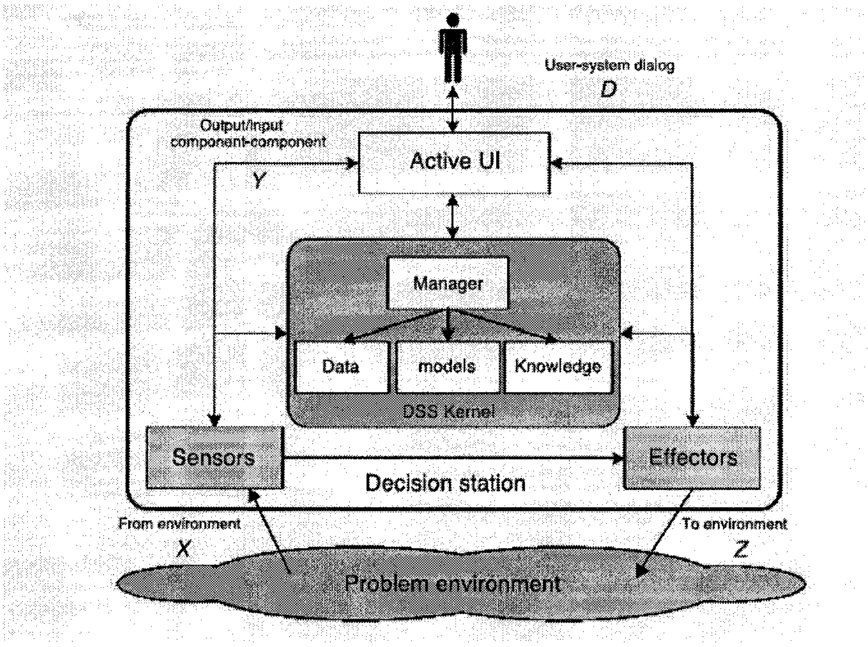


Figure 5. Architecture of decision station (Vahidov & Kersten, 2004)

According to their framework, a decision station can be stated as DS = $\{S, E, N, KN\}$, where S and E stands for sensors and effectors respectively, N represents the user interface, and KN is DSS kernel, i.e. traditional DSS. Furthermore, DSS kernel can be defined as KN = $\{C, G, K\}$, where C is the capabilities of a DSS, G represents the goal of a DSS, and K is the knowledge base. The advantage of this architecture is that it can provide human decision makers with timely and proactive decision support in dynamic and complex environment by connecting the sensors and effectors with problem domain. In their paper, this concept is further illustrated with a prototype supporting individual investment management.

While this framework can improve efficiency and/or effectiveness of decision making for single decision maker, it does not consider the issue of distributed decision making. They do not explicit consider decision making model and treat a decision station as an individual decision unit implicitly (please note that decision making model is not the models of DSS kernel shown in Figure 5). Hence, it is necessary to extend this framework to handle distributed decision making pervaded in the increasingly complicated business environment of today. For this reason, we can extend the structure of a decision station by embedding our model of decision making process.

The goal G in the original DS architecture is replaced by the decision model DM that we have discussed in previous sections.

6 INTEGRATED DECISION STATIONS

In this section, we will provide a formal description of distributed decisions. It is a foundation to build integrated decision stations. First we will examine the simplest case, i.e. how decision making is distributed between two decision units. Next, we bring a general definition of DDM. Last, we combine it with decision station architecture to build a comprehensive model: Integrated decision stations.

6.1. Generic architecture

First, we consider the issue of distributing decision making between two decision units. In the previous discussion of the types of DDSS, we define decision unit (DU) as DU = $\{DM, H, T, L\}$ .However, because the focus in on the distribution of decision authority, we will ignore the other factors and consider decision model (DM) only.

According to Schneeweiss' (2003b) definition, DM = $\{C, A, I\}$, where C is decision maker(s)'s preference structure (decision criteria), A is action space (decision variables), and I is information status. The decision making process is to find out appropriate value of decision variables, which is formulated as:

$\alpha = \arg \underset{\alpha \in A}{opt} \; E\{C \mid I\}$, where $\arg f(x) = x$, E = mathematical expectation, and *opt* means to choose an alternative that optimize decision maker(s)'s expectation value.

You may notice that we have extended the definition of DM by including more components, i.e. $DM = DM(S, F, C^L, C^E, A, CS)$. However, here we focus on C, A and I .

Though Schneeweiss' (2003b) definition and corresponding coupling structures are very useful in understanding distributed decision making, they do not provide direct link to DSS. To accommodate the design of DSS, we need to decompose *I,* i.e. information status. Kim et al (2000) presents four kinds of input and output of a model when designing a model coordination subsystem for organizational decision support systems. Basing on their ideas, we divide *I* into four categories, i.e. $I = \{I^L, I^S, I^C, I^R\}$, where I^L represents the local information of a DU, I^S represents the commonly shared information between DUs, I^C is the control information from other DU, and I^R is the feedback information from the other DU. When a DU receives control information, it can do nothing but follow this information in its decision making process. On the contrary, the feedback information I^R may be either evolved into control information or dropped during the interaction. Compared with I^L and I^S, which are relatively stable, I^C and I^R are dependent upon the interaction between two DUs.

Considering that information asymmetry is more popular than information symmetry, Schneeweiss (2003b) define one DU is at top level and the other one is at base model. We adopt this concept but treat two levels equally, i.e. by default, there is no hierarchy between top level and base level. This structure is shown in Figure 6 with the decomposition of information status *I*.

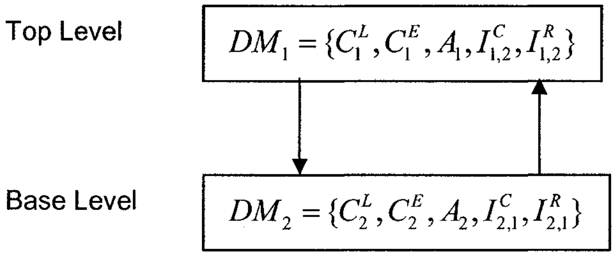


Figure 6. Independence between two decision models

C_1^L is the private criterion of top level, while C_1^E is the criterion considering base-level preferences. Similarly, C_2^L is the private criterion of base level, while C_2^E is the criterion considering top-level preferences. The constitution of criterion C determines whether a DU considers the others. For example, if $C_1^E = \varnothing$, it means that top-level decision maker is egocentric. If $C_1^L = \varnothing$ top-level is altruistic. Normally, these criteria are not null. It means that each DU considers its own criteria as well as that of the other. Though we do not intend to explain every combination of these Cs, we want to emphasize that this structure is so powerful that it can almost describe every kind of decision leadership styles(see (Schneeweiss, 2003a) for details).

Basing on the decomposition of criteria structure, we can arrive at six styles of coordination between two decision units. These styles are listed in Table 6. In this table, 0 stands for a empty set and 1 stands for a nonempty set. For example, regarding coordination style 1, the local criteria of both decision units are empty, but their external criteria are not empty. This composition is very important because it helps us to understand the decision structure in an organization and guide us to develop an efficient DSS to support this structure.

Style of Coordination	C_1^L	C_1^E	C_2^L	C_2^E	Description
1	0	1	0	1	Both DM_1 and DM_2 are altruistic (consider the other's preference only)
2	0	1	1	0	DM_1 considers the other's preference only while DM_2 consider itself only.
3	0	1	1	1	DM_1 considers the other's preference only while DM_2 considers both.
4	1	0	1	0	Both DM_1 and DM_2 are egocentric (consider its own preference only)
5	1	0	1	1	DM_1 considers its own preference only while DM_2 considers both.
6	1	1	1	1	Both DM_1 and DM_2 consider their preference as well as the others' preference.

Table 6. Coordination styles

The above decomposition only solves the issue of division of labor in decision making, i.e. "who decides what". The remaining issue is to how to coordinate these decision making activities. Hence, we need to formally define coordination mechanisms, the control of information flows. There are at least four levels of coordination: data exchange, negotiation, planning, and leadership, which are listed with increasing sophistication(Schneeweiss, 2003a). However, using our approach of information decomposition, we can get eight types of information coordination, which are presented in Table 7. In this table, 0 stands for empty set whereas 1 stands for nonempty set.

Type of Information Coordination	$I_{1,2}^C$	$I_{1,2}^R$	$I_{2,1}^C$	$I_{2,1}^R$	Description
1	0	0	0	0	No direct communication between DM_1 and DM_2. However, they may affect each other indirectly if $S_i \bigcap S_j \neq \varnothing$.
2	0	0	0	1	DM_2 provides some reference information to DM_1, for example query or feedback.
3	0	0	1	0	DM_2 commands DM_1 to follow an instruction.
4	0	0	1	1	DM_2 provides both control and reference information to DM_1
5	0	1	0	1	Both DM_1 and DM_2 provide reference information.
6	0	1	1	0	DM_2 commands DM_1 to follow an instruction, and DM_2 provides reference information
7	0	1	1	1	DM_2 gives DM_1 both control and reference information, and DM_2 provides reference information

8	1	1	1	1	Both DM_1 and DM_2 provide each other with control and reference information.

Table 7. Information coordination between two decision units

If we consider decision styles and information coordination together, we will get 6X8 = 48 scenarios of distributed decision making from information processing perspective. Just for illustration, some of them are listed in Table 8. We do not intent to explain every scenario. What we want to emphasize is that our framework has strong descriptive capability, which enable DSS designer to understand and build an appropriate DSS to support a vast range of distributed decision making processes.

Type of DDSS	$\{C_1^L, C_1^E, C_2^L, C_2^E\}$	$\{I_{1,2}^C, I_{1,2}^R, I_{2,1}^C, I_{2,1}^R\}$	Description
1	$\{0,1,0,1\}$	$\{0,1,0,1\}$	DM_1 and DM_2 are altruistic and provide each other with reference information
2	$\{1,0,1,0\}$	$\{0,1,0,1\}$	Both DM_1 and DM_2 are egocentric, but they provide each other with reference information
3	$\{1,0,1,0\}$	$\{1,0,1,0\}$	Both DM_1 and DM_2 are egocentric, but they can command each other in different area.
...			

Table 8. The classification of distributed decision support systems

Finally, combining the result for single decision station, we arrive at the definition of integrated decision stations (IDS):

$$IDS = \bigcup_{i}^{n} DS_i(SR_i, E_i, N_i, KN_i), \text{ n >=2} \tag{17}$$

where

DS_i stands for the i^{th} decision station,

$SR_i = (C_i, G_i, K_i, R_i, O_i)$ is the sensor

C_i : the capabilities of the sensor, G_i : the goals of the sensor

K_i : the knowledge base of the sensor, R_i : the requests from the other parts

O_i : the operations of the sensor

$E_i = \{C_i, K_i, G_i, O_i\}$ is the effector

C_i : the capabilities of the effector, G_i : the goals of the effector

K_i : the knowledge base of the effector, O_i : the operations of the effector

$N_i = \{C_i, V_i, K_i, G_i, O_i\}$ stands for the interface

C_i : the capabilities of the interface, V_i : the user profiles

G_i : the goals of the interface, K_i : the knowledge base of the interface

O_i : the operations of the interface

$KN_i = \{C_i, DM_i, K_i\}$ is the DSS Kernel

C_i : the capabilities of the kernel

$DM_i = DM_i(S_i, F_i, C_i^L, C_i^E, A_i, CS_i)$ is the decision making model

S_i : Situation space, F_i : Alternatives

C_i^L : Local preference structure (decision criteria)

C_i^E : External (the others') preference structure

A_i : Action space

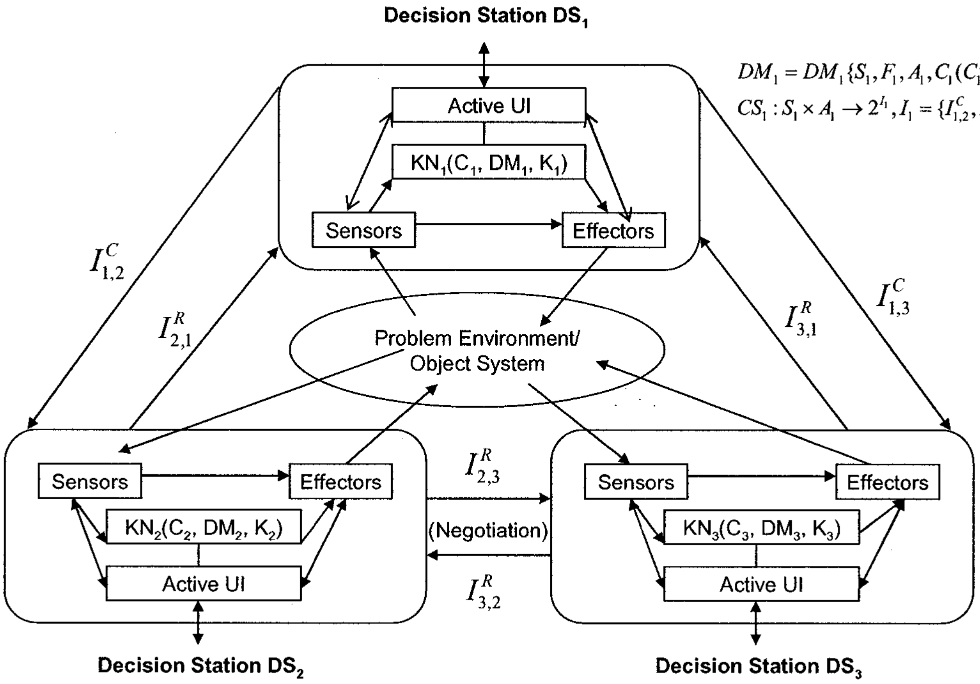
CS_i : Coordination structure

$CS_i : S_i \times A_i \rightarrow 2^{I_i}$, $I_i = I_i^L \cup I_i^S \cup I_i^C \cup I_i^R$

I_i^L : Local information, I_i^S : Common/shared information

I_i^C : Control information, I_i^R : Reference information

To illustrate the above concept, a typical model with three integrated decision stations is presented in Figure 6. The coordination between the top-level DS and base-level DSs is planning process, which is enabled by control information flow and feedback information flow. The coordination between two base-level DSs is negotiation process enabled by the exchange of feedback information (Local and shared information are ignored in this figure).



$$DM_1 = DM_1\{S_1, F_1, A_1, C_1(C_1^L, C_1^2, C_1^3), CS_1\}$$
$$CS_1 : S_1 \times A_1 \to 2^{I_1}, I_1 = \{I^C_{1,2}, I^C_{1,3}\}$$

$$DM_2 = DM_2\{S_2, F_2, A_2, C_2(C_2^L, C_2^1, C_2^3), CS_2\}$$
$$CS_2 : S_2 \times A_2 \to 2^{I_2}, I_2 = \{I^R_{2,1}, I^R_{2,3}\}$$

$$DM_3 = DM_3\{S_3, F_3, A_3, C_3(C_3^L, C_3^1, C_3^2), CS_3\}$$
$$CS_3 : S_3 \times A_3 \to 2^{I_3}, I_3 = \{I^R_{3,1}, I^R_{3,2}\}$$

Figure 7. A typical model of integrated decision stations.

6.2. Implementation guideline

The general steps to design integrated decision stations can be stated as follows:

1. Identify the decision model DM for each decision station. It might be a step in organizational design.
2. Analyze the styles of coordination between these DMs, i.e. the decomposition of decision criteria.

3. Examine control information and feedback information required for each DS, i.e. the decomposition of information.
4. Examine local information and shared information required for each DS
5. Design knowledge and/or model base such as liner programming to help decision makers to create alternatives.
6. Design sensors and effectors to gather data from and implement decision actions to problem environment.

These are general guideline facilitating the development of integrated decision station. We must realize that the development of distributed decision support system is still in its infant stage. Besides our conceptual framework, we need more empirical study on how to design an effective distributed DSS. In the following sections, we try to apply our framework to a concrete business case to test applicability of our conceptual work. We also conduct a simulation study to evaluate the performance of this integrated decision stations.

7 INTEGRATED DECISION STATIONS FOR LEAD TIME MANAGEMENT

Designing integrated decision stations requires a methodology that is somewhat different from that for traditional information systems (IS) development. The very early IS development is function-oriented, being concentrated on individual business function, such as accounting information system and inventory management system. Data redundancy and inconsistency inherited in these systems are gradually overcome by lately developed integrated enterprise systems such as enterprise resource planning (ERP) and supply chain management (SCM) packages. Most of these packages are based on value-chain model and are process-oriented. While they can improve business processes performance by shared information and automated information flows, they provide little decision support capabilities. Both function-oriented and process-oriented development methodologies seem inappropriate for the development of information systems supporting distributed decision making. Hence, we propose a decision-oriented methodology to facilitate the development of integrated decision stations. In this section, we will first describe the problem of lead time management in a web-base make-to order manufacturer, and then provide a business solution. After business analysis, we present how to map business solution into our conceptual model of IDS. At the end, a prototype is demonstrated to show the applicability of our conceptual model.

7.1. Lead time management

During the past decade, the increasing popularity of the Internet has promoted many manufacturing companies to adopt e-commerce business model, which enables them directly interact with end consumers by eliminating costly intermediaries in the traditional supply chain(Warkentin, Bapna, & Sugumaran, 2000). Correspondingly, their business philosophy is shifted from production-centric strategy to customer-centric strategy. The Internet provides them with timely information about their customers' needs, which in turn helps them build specific products for specific customers. Therefore, their production models is changed from mass production, to mass customization, and even further to customerization(Wind & Rangaswamy, 2001), an effort to integrate customer into internal business processes, such as collaborative product design. In mass production model products are usually made to stock (MTS) according to sale forecasts, whereas in mass customization model products are made to actual customer orders (MTO). As a result, MTO companies can enjoy the benefits of product flexibility and lower inventory cost, but at the same time they might suffer from a longer and unstable order-to-delivery (OTD) lead time.

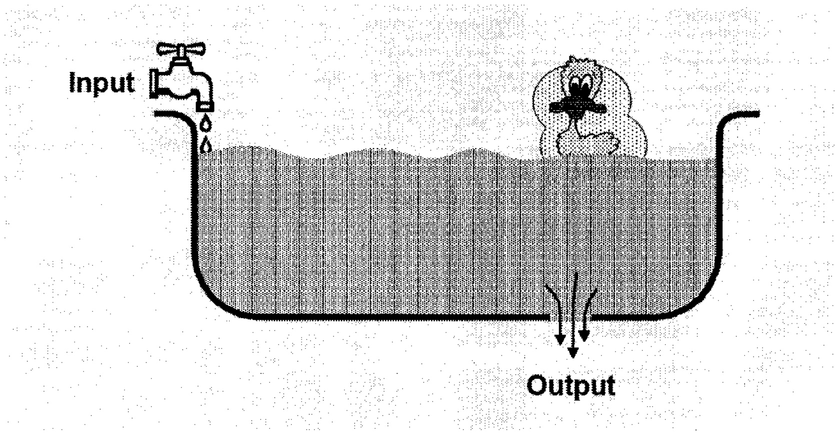


Figure 8. Workload control concept (Land, 2004)

Research in production and operations management shows that the popular production planning and control methods, for example Just in Time (JIT) and Theory of Constraints (TOC) philosophies, are inappropriate for MTO companies due to the great uncertainty of production and market environment (Kingsman, 2000). This challenge faced by MTO firms stimulates much research on the lead time management regarding MTO companies(Haskose, Kingsman, & Worthington, 2004; Henrich, Land, Gaalman, & Zee, 2004; Kingsman, 2000; Land, 2004), and workload control (WLC) concept is emerged as a promising mechanism for production planning and control in MTO environment. Simply stated, the principle of WLC, is to achieve stable OTD lead time by adjusting the input to and output from production system, i.e. the load and capacity of production system, respectively (Wight, 1970). The WLC concept is illustrated in Figure 8. It shows how to maintain a stable water level by adjusting input and output rate. Though WLC is pretty simple in concept and may be very useful in practice, how to effectively implement this concept remains as a question.

7.2. Work load control

According to input-process-output model, production can be treated as a transformation process that employs various resources, such as machine and labor, to convert raw materials into final products (Kingsman, 2000). Workload control (WLC), a concept first introduced by Wight (1970), is aimed to maintain transformation time at a normal level by controlling input/output. Conceptually, workload is modeled as a queue where work is waiting to be processed at a certain resource. As a whole, a production system can be modeled as a queuing network, and hence a computational solution is almost impossible(Haskose, Kingsman, & Worthington, 2002).

The main purpose of workload control is to manage lead time. Though WLC concept is originally devised to handle lead time in manufacturing process, it has been extended to the whole process of order to delivery (OTD) cycle. In his formal analysis of workload control model, Kingsman (2000)identified four phases of order fulfillment process in MTO companies, and pointed out four corresponding levels of workload control, i.e. customer inquiry, order acceptance, job release, and priority dispatching.

7.3. A business case

A fictitious web-based make-to-order manufacturing company, like Dell Computer Inc., is considered to implement lead time management using work load control concept. The sales orders of the company primarily come from the Internet. Therefore order arrivals are random and the interval of order arrivals is highly varied. Moreover, the company allows its customers to personalize products and thus production processing time of each order is also varied with the specific product configuration. These characteristics seem best fit the application of WLC concept(Henrich, Land, & Gaalman, 2004).

Basing on Kingsman's (2000) workload control model, the order flow and lead time structure is depicted in Figure 9.

We assume that firms employ just-in-time production philosophy, i.e. their supplier can delivery required materials without any delay, just like Dell Computer Inc. dose. Hence, the time waiting for materials can be ignored in our model. In addition, we assume that total manufacturing lead time T_m is relatively small due to the high efficiency of manufacturing and that the default scheduling rule is first-in-first-out (FIFO). Under these assumptions, the total order-to-delivery (OTD) lead time is largely dependent on the time waiting in order pool T_w. Thus the management of OTD lead time equals to the management of order pool size.

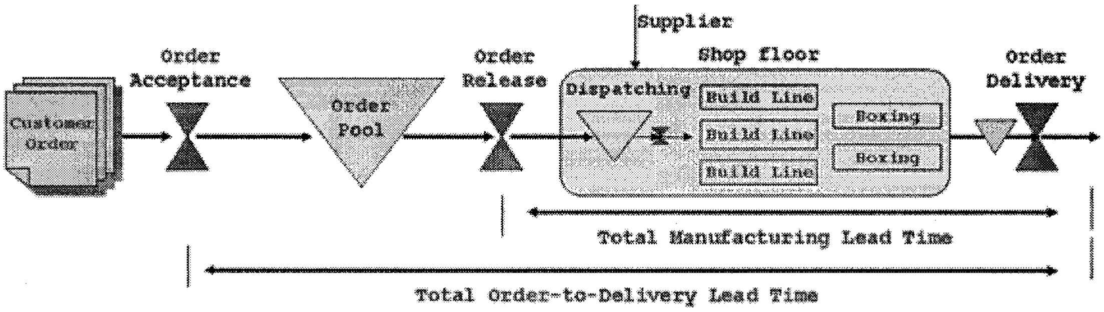


Figure 9. Order flow of web-based MTO firms

The stable size of order pool can be maintained by adjusting demand and/or production capacity. In this company, a demand/capacity planner is assigned to coordinate marketing and production department and has a final say on demand and production management to ensure a good customer service level while keeping down marketing expenses and/or production cost.

To delight its customers further, the company conceives a new idea that customers not only can track order but also can schedule order by themselves if they want. The principle is that customers can switch their positions in the order pool by negotiating a corresponding compensation. As a consequence, a customer can get his/her order processed earlier if this person is willing to pay a certain amount of money to the other customer. By combining work load control and customer self-scheduling mechanism, the company can provide a flexible and stable OTD lead time to its customers.

7.4. Business analysis and system architecture

In analyzing the above case, we can find out four different kinds of decision stations, which are presented in Figure 10. The roles of these decision stations are described as follows:

Decision station DS_C is designed to support demand/capacity planners. First, we analyze its decision model DM_C.

The criterion $C_C = \{C_C^L, C_C^{CM}, C_C^{CP}\}$,

where $C_C^L = LT$, $C_C^{CM} = \{R_{sales}, C_{mk}\}$, $C_C^{CP} = C_{prd}$, LT stands for lead time, R_{sales} is sales revenue, C_{mk} is marketing cost, and C_{prd} is production cost. The action space $A^C = \{\overline{O}_{rate}, \overline{P}_{cap}\}$, where $\overline{O}_{rate}$ is the expected mean of demand and $\overline{P}_{cap}$ is the expected mean of production capacity per shift.

Its sensor SR_C can fetch real-time information about the current size of orders in pool O_{size}^C, compute average demand O_{rate}^C for recent period, for example last day or last hour, the current production capacity P_{Cap}^C, and alert users when actual O_{size}^C and/or production capacity is beyond the user-defined limits.

In summary, the conceptual model for DS_C are:

$$DS_C = \{SR_C, E_C, N_C, KN_C\}$$

$$SR_C = \{conn_C : \{Pool, Shopfloor\} \to \{O_{size}^C, P_{cap}^C\},\ trans_C : \{Pool, Shopfloor\} \to \{O_{rate}^C, O_{LT}^C, P_{utl}^C\}$$
$$alert_C : \{Pool, Shopfloor\} \to Msg^C\}$$

$$E_C = \varnothing, N_C : \ User\ Interface$$

$$KN_C = \{CAP_C, DM_C, K_C\},\ CAP_C = \{query, generate, comm\}$$

$$DM_C = \{S_C, F_C, A_C, C_C, SC_C\}$$

$$S_C = \{O_{size}^C, O_{rate}^C, P_{Cap}^C\},\ A_C = \{\overline{O}_{rate}, \overline{P}_{cap}\}$$

$$F_C = \{f_C : \{O^C_{size}, O^C_{rate}, P^C_{Cap}\} \times \{\overline{O}_{rate}, \overline{P}_{cap}\} \to \{O^C_{size}, O^C_{rate}, O^C_{LT}, P^C_{Cap}, P^C_{utl}\}\}$$

$$C_C = \{C^L_C, C^{CM}_C, C^{CP}_C\},\ C^L_C = LT,\ C^{CM}_C = Rate,\ C^{CP}_C = Utl$$

$$I_C = \{I^C_{C,M}, I^C_{C,P}\}, CS_C : S_C \times A_C \to 2^{I_C}$$

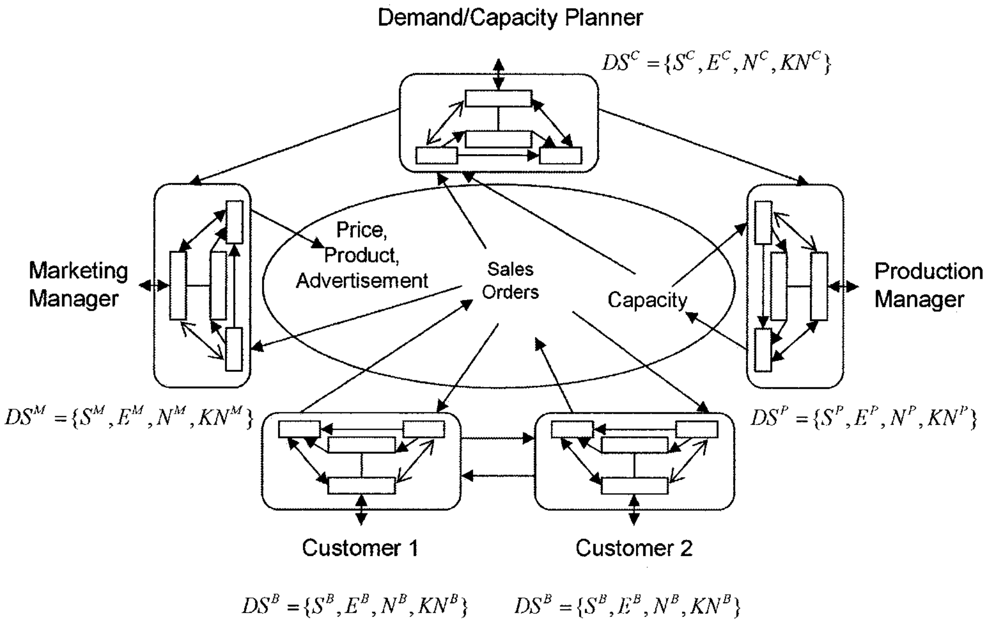


Figure 10. IDS architecture for lead time management.

Decision station DS^M is designed to support marketing managers for short-term marketing control. It captures the real-time information of sales order and provides marketing managers with trend analysis. Marketing managers can control demand by changing product price W_{price}, product mix W_{prdt}, and advertisement W_{advt}. In the same way, we can get its conceptual model as follows:

$$DS_M = \{SR_M, E_M, N_M, KN_M\}$$

$$SR_M = \{conn_M : Pool \to O^M_{size},\ trans_M : Pool \to O^M_{rate},\ alert_M : Pool \to Msg^M\}$$

$$E_M = conn_M : \{O_{prdt}, O_{price,} O_{advt}\} \to \{W_{prdt}, W_{price,} W_{advt}\}$$

N_M : *User Interface*

$KN_M = \{CAP_M, DM_M, K_M\}$, $CAP_M = \{query, generate, comm\}$

$DM_M = \{S_M, F_M, A_M, C_M, SC_M\}$

$S_M = \{O^M_{size}, O^M_{rate}\}$, $A_M = \{O_{price}, O_{prdt}, O_{advt}\}$

$F_M = \{f_M : \{O^M_{size}, O^M_{rate}\} \times \{O_{price}, O_{prdt}, O_{advt}\} \to \{O^M_{size}, O^M_{rate}\}\}$

$C_M = \{C^L_M, C^{MC}_M, C^{MP}_M\}$, $C^L_M = \{Rate, C_{mk}\}$, $C^{MC}_M = C^{MP}_M = \varnothing$

$I_M = \{I^R_{M,C}\}$, $CS_M : S_M \times A_M \to 2^{I_M}$

Decision station DS_P is designed to production managers for short-term capacity control. It captures the real-time information of current production capacity P^P_{cap} and provides marketing managers with cost analysis. Production managers can control capacity by hiring temporary employee P_{emp}, adding shifts P_{shift}, and subcontracting P_{out}. Similarly, we can build its conceptual model as follows:

$DS_P = \{SR_P, E_P, N_P, KN_P\}$

$SR_P = \{conn_P : O_{Shopfloor} \to P^P_{cap}, alert_P : O_{pool} \to Msg^P\}$

$E_P = conn_P : \{O_{emp}, O_{shift}, O_{out}\} \to \{P_{emp}, P_{shift}, P_{out}\}$

N_P : *User Interface*

$KN_P = \{CAP_P, DM_P, K_P\}$, $CAP_P = \{query, generate, comm\}$

$DM_P = \{S_P, F_P, A_P, C_P, SC_P\}$

$S_P = P^P_{cap}$, $A_P = \{O_{emp}, O_{shift}, O_{out}\}$

$F_P = \{f_P : P^P_{cap} \times \{O_{emp}, O_{shift}, O_{out}\} \to P^P_{cap}\}$

$$C_P = \{C_P^L, C_P^{PC}, C_P^{PM}\},\ C_P^L = \{Utl, C_{prd}\},\ C_P^{PC} = C_P^{PM} = \varnothing$$

$$I_P = \{I_{P,C}^R\}, CS_P : S_P \times A_P \to 2^{I_P}$$

Decision station DS^B is designed for buyers, i.e. customers on the Internet. When a customer places an order on the Internet, a default standard order-to-delivery lead time is placed in this order. However, if this customer is eager to get the product, he/she can negotiate with the other customers to advance the order. Decision criterion is customer's utility $C_{utility}$ of due date of the order O_{date} and the agreed compensation O_{comp}. The customers could use multiple criteria utility model to support his/her negotiation. Unlike the coordination among DS_C, DS_M, and DS_P, which is static, the relationship between any two DS_B s is dynamically established. The conceptual model of DS^B are:

$$DS_B = \{SR_B, E_B, N_B, KN_B\}$$

$$SR_B = \{conn_B : Pool \to\ O_{date}, alert_B : Pool \to Msg^B\},$$

$$N_B :\ User\ Interface$$

$$KN_B = \{CAP_B, DM_B, K_B\},\ CAP_B = \{query, generate, comm\}$$

$$DM_B = \{S_B, F_B, A_B, C_B, SC_B\}$$

$$S_B = O_{date}, A_B = O_{comp}$$

$$F_B = \{f_B : O_{date} \times O_{comp} \to O_{date}\}$$

$$C_B = C_B^L(O_{comp}, O_{date})$$

$$I_B = I_B^R(O_{comp}), CS_B : S_B \times A_B \to 2^{I_B}$$

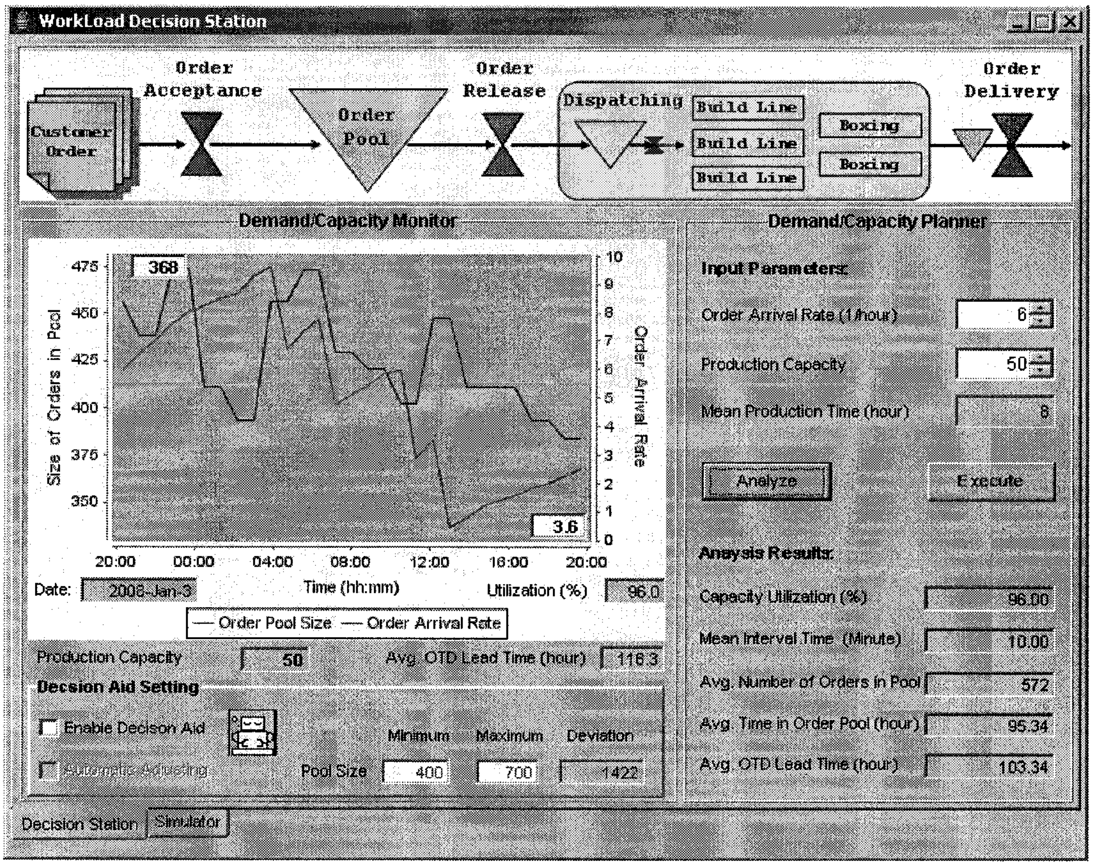
As a whole, this integrated decision station can be expressed as:

$IDS_{LT} = DS_C \cup DS_M \cup DS_P \bigcup_{i=1}^{n} DS_{B_i}$, which include one decision station for demand/capacity planner, one decision station for marketing manager, one decision station for production manager, and at least one decision station for consumers.

7.5. Realization

After we demonstrate how to describe a distribute decision problem using our conceptual framework, we need to know to realize it by the aid of current information technology. It is not our purpose to provide a technical solution that implements our framework. However, to test applicability of our framework, we develop a simple prototype using Java language.

In addition, to support our simulation study, a real-time driven approach is adopted to develop this prototype.



Date: 2008-Jan-3

Utilization (%) 96.0

Production Capacity 50

Avg. OTD Lead Time (hour) 116.3

☐ Enable Decison Aid

☐ Automatic Adjusting

	Minimum	Maximum	Deviation
Pool Size	400	700	1422

Order Arrival Rate (1/hour) 6

Production Capacity 50

Mean Production Time (hour) 8

Capacity Utilization (%) 96.00

Mean Interval Time (Minute) 10.00

Avg. Number of Orders in Pool 572

Avg. Time in Order Pool (hour) 95.34

Avg. OTD Lead Time (hour) 103.34

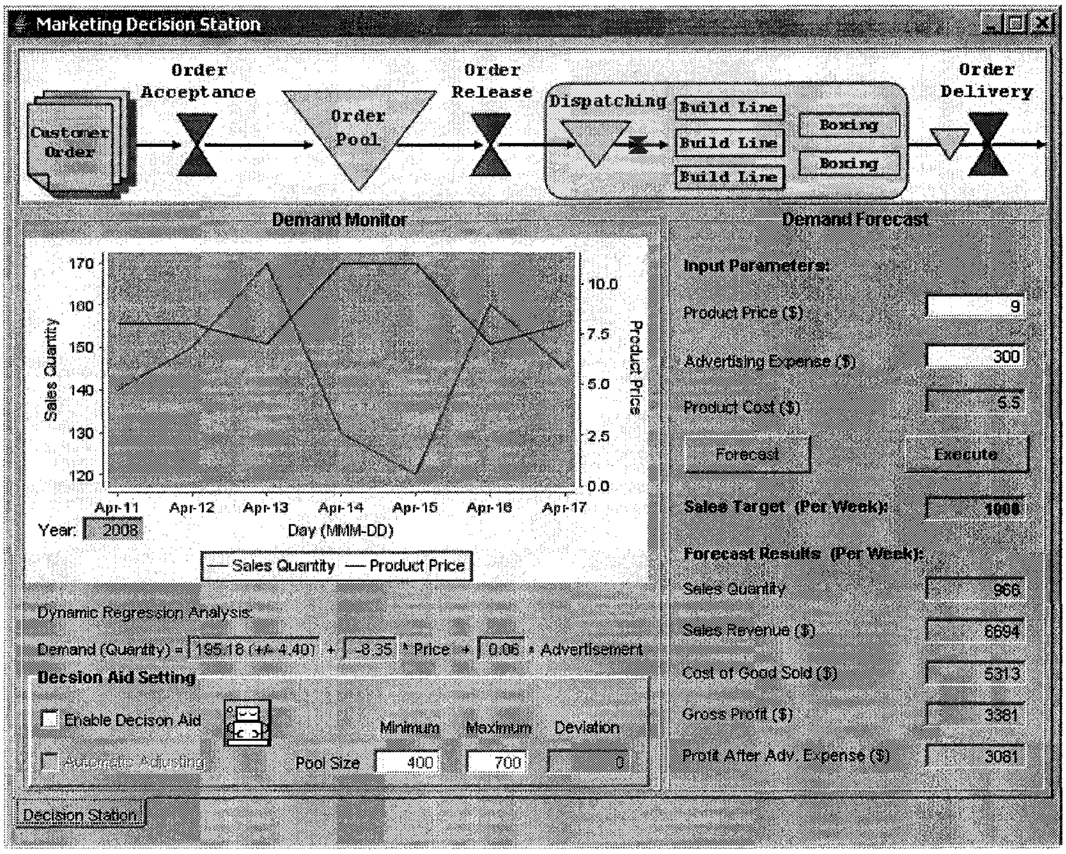
Figure 11. A prototype of decision station for demand/capacity planner

A sample screenshot of the decision station for demand/capacity planner, i.e. DS^C in our conceptual model is shown in Figure 11. The left pane of the screen is demand and capacity monitor, i.e. the sensor S^C. The sensor can capture and calculate required data in real-time or in a period of delay (it is used to support our simulation study). The right panel is a queuing model to support lead time decisions, where two decision variables, the mean demand (shown as order arrival rate) and production capacity are input parameter. The queuing model corresponds to

$$F_C = \{f_C : \{O^C_{size}, O^C_{rate}, P^C_{Cap}\} \times \{\overline{O}_{rate}, \overline{P}_{cap}\} \to \{O^C_{size}, O^C_{rate}, O^C_{LT}, P^C_{Cap}, P^C_{utl}\}\}.$$

The planner can use queuing model to generate various alternatives, i.e. different combination of order arrival rate and production capacity. When planners are satisfied with analysis result, they can pass control information to the decision stations for marketing managers and production managers, triggering a serial of the other decision making processes.

Figure 12 illustrates a prototype of decision station for marketing department. The left panel is demand monitor. The sensor can automatically do regression analysis of demand on product price and advertisement expenses. The right panel is forecasting model, which can be used to generate various alternatives (the combination of product price and advertising expenses) to achieve desired sales revenue. We should notice, in this case, sales target, is control information from demand/capacity planner decision station.



Year: 2008

Dynamic Regression Analysis:

Demand (Quantity) = 195.18 (+/- 4.40) + -8.35 * Price + 0.06 * Advertisement

Decsion Aid Setting

☐ Enable Decison Aid

☐ Automatic Adjusting

	Minimum	Maximum	Deviation
Pool Size	400	700	0

Input Parameters:

Product Price ($): 9

Advertising Expense ($): 300

Product Cost ($): 5.5

Sales Target (Per Week): 1008

Forecast Results (Per Week):

Sales Quantity: 966

Sales Revenue ($): 8694

Cost of Good Sold ($): 5313

Gross Profit ($): 3381

Profit After Adv. Expense ($): 3081

Figure 12. A prototype of decision station for marketing department

8 The EVALUATION OF INTEGRATED DECISION STATIONS

The impact of DSS on the performance of decision makers is inconclusive. Some literature shows significant improvement of decision performance, while others demonstrate that the impact of DSS is very limited. Because DSS is an interactive man-computer system supporting decision makers to solve semistructured problems, the impact of it at least can be attributed to four types of factors: those related to DSS itself, i.e. the technical characteristics of DSS, those related to decision makers, for example, individual personality and commitment, those related to the structure of problems, and those associated with organizational environment. While all of these factors might influence the effectiveness of DSS, we focus ourselves on the technical aspect of DSS. It does not mean that the other factors are not important. In fact, they might be more important than technical factors. However, in this paper, our interest is in the difference between the traditional decision support systems and our dynamic distributed decision support systems and its impact on decision performance. More specifically, IDS can provide timely information and information coordination mechanism, which are not well presented in traditional DSS. We hypothesize that these two factors might affect decision performance.

Under this consideration, we assume that decision makers are rational and will choose an alternative basing our predefined rules. Hence the decision process is automated and we can conduct a simulation study using the aforementioned prototype.

8.1. Research model and questions

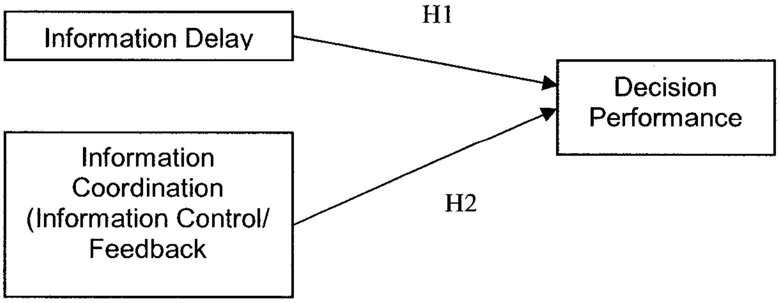


Figure 13. Research model

Our research model is depicted in Figure 13. In the following, we will explain each variable and their relationships.

Independent Variables:

Information Delay: It refers to the currency of the relevant information that decision makers have to get to make decision. While situated decision support systems can provide real-time information, the traditional DSS can only provided delayed information. In the simulation experiment, it is manipulated by setting an interval between the time when relevant data are created and the time when those data are received by decision maker.

In the simulation, we assume that decision makers make their decision immediately. That is there is no decision delay in our model. In fact, from the standpoint of system dynamics, these two types of delay have the same effect. Hence, we focus on information delay only.

Three level of information delay are set in this simulation:

- Real-time: 50 minutes;
- Delay 10: 500 minute;
- Delay 20: 1000 minutes.

Information Coordination: according to our framework, information coordination refers to information control and feedback by the aid of control information I^C and reference information I^R. In our experiment, we setup three types of information control:

- Control marketing department by setting up an expected normal demand;
- Control production department by setting up an expected normal production level;
- Control both of marketing and production department.

When examining the effect of control information, we assume that production and marketing department are able and willing to follow the instruction from demand/capacity planner. Therefore, there is no necessary to consider the effect of feedback information.

Regarding information feedback, we design an experiment in which demand/capacity planner can control both marketing and production department. However, either marketing or production department is unable to fulfill the instruction from demand/capacity planner. If information feedback is enabled, they can send this information, the lack of capacity, to the planner, and then planner can adjust his/her instructions correspondingly. Simply stated, with information feedback, the demand/capacity plan will have a broader control scope. For example, without information feedback, the planner assumes that marketing department is able to adjust demand to any level between 5 orders per hour and 7 orders per hour. But in fact, the marketing department can only achieve a demand level between 5 orders per hour and 6 order per hour. Due to the lack of feedback, the planner might lose a chance to do the other corrective actions, for example, decreasing production capacity. In the experiment, we set up four types of information feedback:

- Information feedback from both marketing and production;
- Information feedback from marketing only;
- Information feedback from production only;
- No information feedback at all.

Dependent Variable:

Decision Performance: It is an objective measure that computes the cumulative deviation of actual output from predetermined goal. In the simulation, it is the summation of absolute deviation of actual order pool size, which equals $\sum_{i=1}^{N}[Max(O_{size}^{i}, O_{low}) - Min(O_{size}^{i}, O_{high})]\Delta t$, where O_{size}^{i} is actual size of order pool at the simulation time t_i , Δt is the time interval between two consecutive check of the order size (in our cases, it is one second), and O_{low} and O_{high} are the predetermined boundary of order pool. This real value of the deviation is shown in Figure 14 as shaded areas. Our calculation is a very close approximation of the value of the deviation because our sampling interval Δt is small.

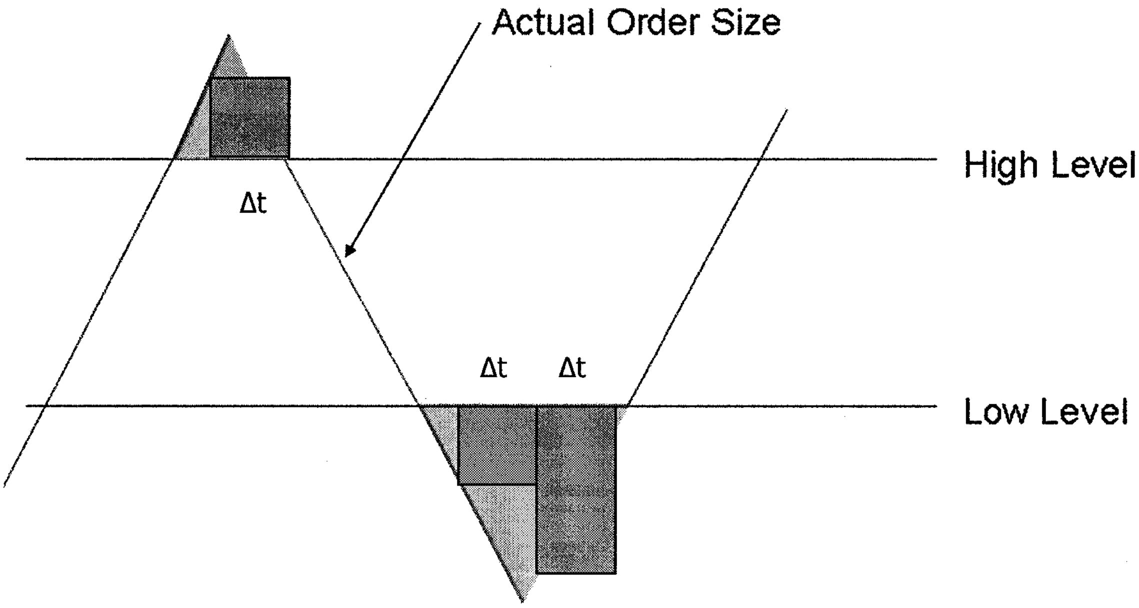


Figure 14. The measure of decision performance

Hypothesis:

One of distinctive features of IDS is that they are connected to problem environment. Their sensors can often get timely information from the environment and thus the information used in IDS is more current than that used in traditional stand-alone DSS. The timely information may help to improve decision performance. Hence, we hypothesize:

H1: The level of information delay is negatively related to decision performance.

This also implies that Integrated Decision Stations have better performance than traditional stand-alone DSS because of the less delay of required information in situated DSS.

Information coordination is another feature of IDS. IDS enable one decision station to communicate with other decision stations. On the one hand, they can exchange reference information to facilitate decision making. On the other hand, one decision station can control other decision stations by issuing control information. This information coordination mechanism could lead to better decision from global perspective. Therefore, we hypothesize:

H2: The level of information coordination is positively related to decision performance.

More specifically, we can state:

H2a: The level of information control is positively related to decision performance.

In our case, it refers to controlling both marketing and production department will achieve better performance than controlling either marketing or production department.

H2b: The level of information feedback is positively related to decision performance.

In our case, it refers to receiving information feedback from both marketing and production department will achieve better performance than receiving information feedback from either marketing or production department, and receiving information feedback either from marketing or production will achieve better performance than receiving no information feedback.

This implies, due to the Information control and feedbacks, Integrated Decision Stations have better performance than traditional stand-alone DSS.

8.2. Experimental design

With three levels of information delay and three types of information coordination, we can get nine treatments. For each treatment, we use the same eight set of random serials to test the decision performance in lead time management. The decision performance measured as the deviation of order pool size from the predetermined scope is calculated and recorded automatically by computer (it is shown on the screenshot Figure 11).

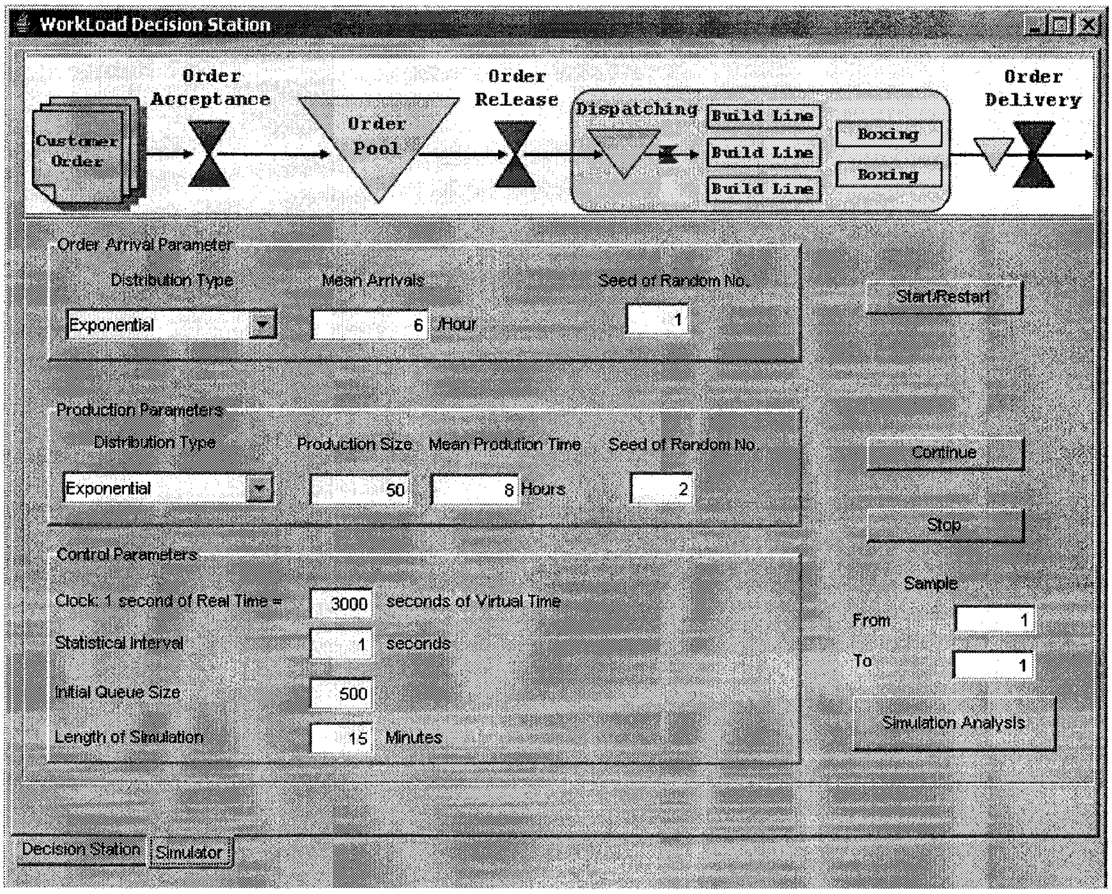
1. Experiment Parameters

We define two set of parameters regarding problem domain. We call them Case 1 and Case 2 separately. In case 1, the arrivals of purchase order from website follows Poisson distribution with mean six orders per hour. The production time of finishing one batch of orders follows exponential distribution with mean eight hours. In case 2, the distribution of order arrivals is the same as in case 1, but the production time follows uniform distribution with lower bound as seven hours and higher bound as nine hours. In both cases, the standard production time is eight hours.

In addition, we setup a simple decision rule. If the current order pool size is bigger than the upper limit, the decision aid will either decrease the mean of order arrivals down to three orders per hours or increase production capacity up to 100 orders per shift. If the current order pool size is smaller than the lower limit, the decision aid will either increase the mean of order arrival up to nine orders per hours or decrease production capacity down to 50 orders per shift.

Furthermore, we assume the above adjustment can be immediately implemented. However, it does not mean that they become effective immediately. For purchase orders, the change of mean of order arrival is immediately effective. However, only after the current production batch is finished can the change of the production capacity become effective.

2. Experiment Procedure



Order Arrival Parameter

Distribution Type: Exponential

Mean Arrivals: 6 /Hour

Seed of Random No.: 1

Production Parameters

Distribution Type: Exponential

Production Size: 50

Mean Prodution Time: 8 Hours

Seed of Random No.: 2

Control Parameters

Clock: 1 second of Real Time = 3000 seconds of Virtual Time

Statistical Interval: 1 seconds

Initial Queue Size: 500

Length of Simulation: 15 Minutes

Sample

From: 1

To: 1

Figure 15. The screenshot of simulation study

We use fast-than-real-time method to conduct this experiment. In the simulation, we set one second real time = 3000 seconds of virtual time. For each random input of every treatment, we collect one month's data (15 minutes in real time). In total, we collect 72 data for case 1 and case 2 respectively. The screenshot of simulation program is illustrated in Figure 15.

One experiment is to test the impact of information delay and information control. The simulation procedure is presented in Figure 16.

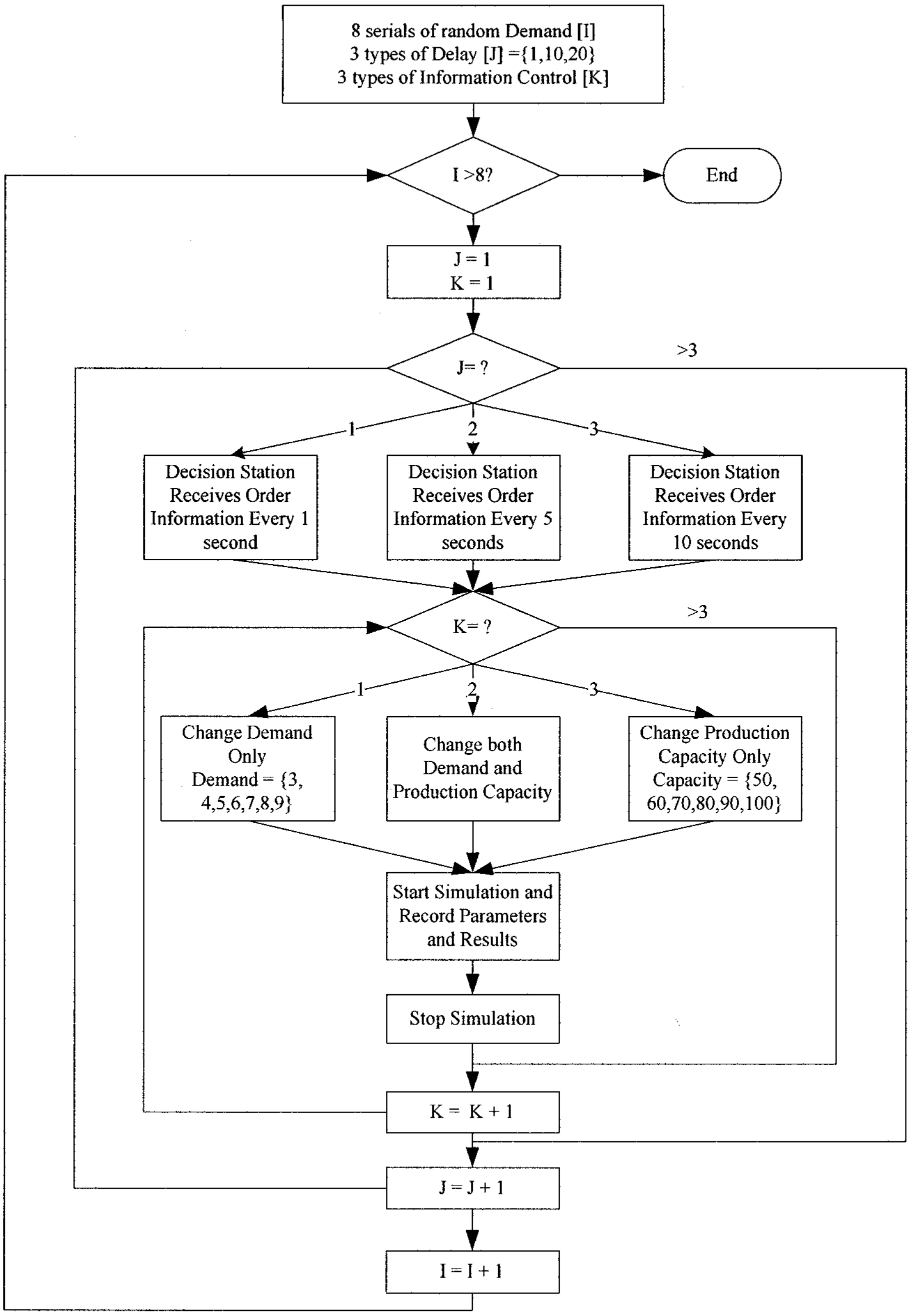


Figure 16. Simulation procedure for information delay and control

Another experiment is to test the impact of information feedback. The simulation procedure is presented in Figure 17.

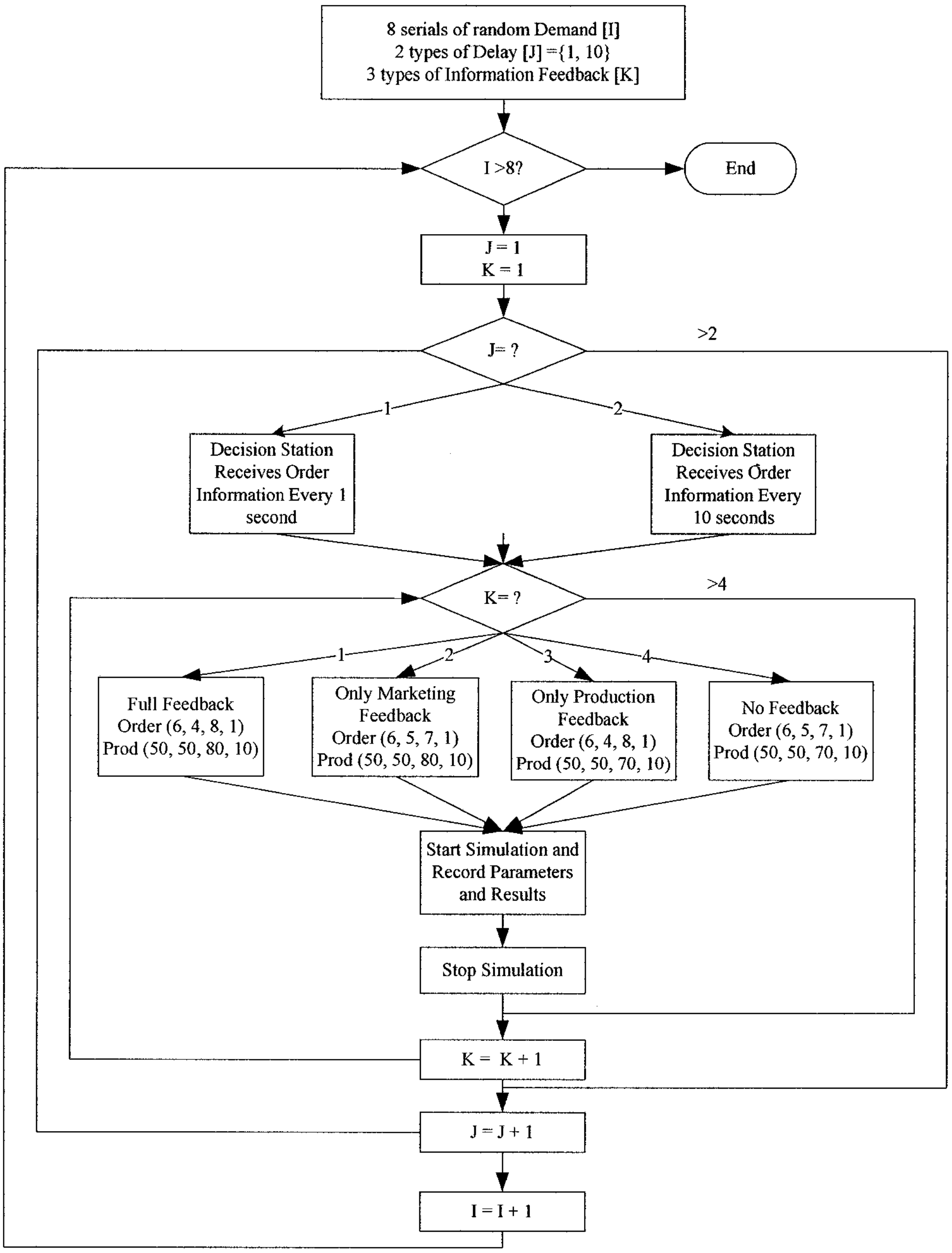


Figure 17. Simulation procedure for information feedback

9 THE RESULTS OF SIMULATION EXPERIMENTS

9.1. Information delay and information control

Case 1: Order arrivals follow Poisson distribution and production time follows exponential distribution.

Table 9 shows the number of samples, the mean, and the standard deviation of the deviation of order pool size for each treatment. From Table 10, we can see that there is no significant interaction between information delay and information control (F = 0.611, p = 0.656). The deviation of order pool size is significantly related to the level of information control, F (2, 63) = 21.63, p< 0.001.

Dependent Variable: Deviation of Order Pool Size

Information Control	Information Delay	Mean	Std. Deviation	N
Marketing	Real Time	16679.13	20536.192	8
	8 Hrs Delay	37408.63	27975.611	8
	16 Hrs Delay	56998.00	28853.751	8
	Total	37028.58	30045.287	24
Marketing & Production	Real Time	23617.88	22040.995	8
	8 Hrs Delay	53292.38	28101.428	8
	16 Hrs Delay	56340.63	25955.254	8
	Total	44416.96	28645.196	24
Production	Real Time	115910.0	72000.860	8
	8 Hrs Delay	114374.3	69003.359	8
	16 Hrs Delay	112647.9	65072.389	8
	Total	114310.7	65707.731	24
Total	Real Time	52069.00	63156.918	24
	8 Hrs Delay	68358.42	55464.512	24
	16 Hrs Delay	75328.83	49737.864	24
	Total	65252.08	56448.378	72

Table 9. Descriptive Statistics (Case 1 of experiment 1)

Dependent Variable: Deviation of Order Pool Size

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	99073193666[a]	8	1.238E+10	6.135	.000	.438
Intercept	3.0656E+11	1	3.066E+11	151.881	.000	.707
CONTROL	87298009750	2	4.365E+10	21.625	.000	.407
DELAY	6839613204	2	3419806602	1.694	.192	.051
CONTROL* DELAY	4935570712	4	1233892678	.611	.656	.037
Error	1.2716E+11	63	2018453670			
Total	5.3280E+11	72				
Corrected Total	2.2624E+11	71				

a. R Squared = .438 (Adjusted R Squared = .367)

Table 10. ANOVA table (Case 1 of experiment 1)

Further, we do contrast tests to compare the mean of three levels of information control. Contrast 1 checks the difference between controlling marketing and controlling both marketing and production. Contrast 2 looks at the difference between controlling marketing and production. Contrast 3 examines the difference between controlling both marketing and production and controlling production. The results are shown in Table 11.

Contrast Tests

		Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
Deviation of Order Pool Size	Assume equal variances	1	-7388.38	12953.734	-.570	69	.570
		2	-77282.13	12953.734	-5.966	69	.000
		3	-69893.75	12953.734	-5.396	69	.000
	Does not assume equal variances	1	-7388.38	8473.652	-.872	45.896	.388
		2	-77282.13	14748.199	-5.240	32.215	.000
		3	-69893.75	14631.663	-4.777	31.438	.000

Table 11. Contrast tests (Case 1 of experiment 1)

Levene's Test is significant ($F = 8.18$, $p < 0.001$). Hence the variances are significantly different and we need to check the second half of the table of contrast tests. After examining the result of contrast tests, we can find that, because $p = 0.39$, there is no significant difference between controlling marketing and controlling both marketing and production. However, controlling marketing only will achieve significantly ($p < 0.001$) less fluctuation in order pool than controlling production only. Similarly, controlling both marketing and production will get significantly ($p < 0.001$) less fluctuation in order pool than controlling production only.

It is surprising that the impact of information delay is insignificant in the overall sample. According to interaction plot shown in Figure 20, If we consider the level of information control separately, we will find some interesting results. In fact, if we drop the data regarding the third type of information control, i.e. controlling production only, it shows that information delay is significantly related to the deviation of order pool size at $p = 0.001$ (Shown in Table 12).

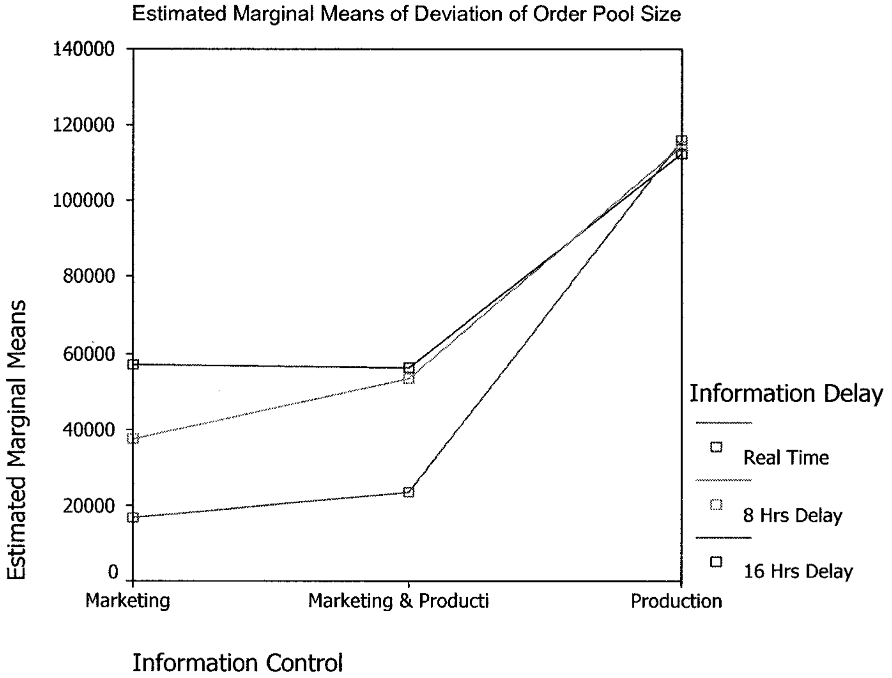


Figure 18. The interaction plot (Case 1 of experiment 1)

More specifically, we make three contrasts tests. Contrast 1 compares real-time information with 8 hours delay of information. Contrast 2 compares real-time information with 18 hours delay of information, and contrast 3 compares 8 hours' delay with 16 hours' delay. Table 13 shows that real-time information can achieve significantly better decision performance than 8 hours' delayed information at $p < 0.01$, and real-time information can achieve significantly better decision performance than 16 hours' delayed information at *p* <0.001. However, Information delay of 8 hours is not significantly different from 16 hours delay (*p* = .252).

Dependent Variable: Deviation of Order Pool Size

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	12387626649[a]	5	2477525330	3.729	.007	.307
Intercept	79600515089	1	7.960E+10	119.818	.000	.740
CONTROL	655057021.7	1	655057021.7	.986	.326	.023
DELAY	11184139019	2	5592069510	8.417	.001	.286
CONTROL * DELAY	548430608.4	2	274215304.2	.413	.664	.019
Error	27902559357	42	664346651.4			
Total	1.1989E+11	48				
Corrected Total	40290186006	47				

a. R Squared = .307 (Adjusted R Squared = .225)

Table 12. ANOVA table (Case 1 sub dataset)

		Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
Deviation of Order Pool Size	Assume equal variances	1	-25202.00	8991.670	-2.803	45	.007
		2	-36520.81	8991.670	-4.062	45	.000
		3	-11318.81	8991.670	-1.259	45	.215
	Does not assume equal variances	1	-25202.00	8794.131	-2.866	27.603	.008
		2	-36520.81	8438.651	-4.328	28.442	.000
		3	-11318.81	9695.505	-1.167	29.873	.252

Table 13. Contrast tests (Case 1 sub dataset)

Basing on the above results, regarding Case 1, we can conclude that:

1) The level of information control could significantly affect decision performance. The decision performance of controlling marketing only is significantly better than that of controlling production. The decision performance of controlling both marketing and production is significantly better than that of controlling production only. However, there is

no significant difference between controlling marketing only and controlling both marketing and production.

2) Under most situations (except controlling production only), information delay is significantly related to decision performance. The less delay the required information, the better decision performance.

Case 2: Order arrivals follow Poisson distribution and production time follows exponential distribution.

Table 14 shows the number of samples, the mean, and the standard deviation of the deviation of order pool size for each treatment. From Table 15, we can see that there is significant interaction between information delay and information control. The interaction plot is shown in Figure 19. We can find that the deviation of order pool size is significantly related to the types of information control, *F* (2, 63) = 39.51, $p < 0.001$.

Dependent Variable: Deviation of Order Pool Size

Information Control	Information Delay	Mean	Std. Deviation	N
Marketing	Real Time	1166.50	105.919	8
	8 Hrs DL	11717.75	4437.498	8
	16 Hrs DL	14353.13	10348.317	8
	Total	9079.13	8512.467	24
Marketing & Production	Real Time	2063.88	760.635	8
	8 Hrs DL	8151.38	2369.325	8
	16 Hrs DL	16106.63	11308.236	8
	Total	8773.96	8677.622	24
Production	Real Time	57527.25	25448.522	8
	8 Hrs DL	31335.13	18993.454	8
	16 Hrs DL	27937.63	19278.348	8
	Total	38933.33	24543.847	24
Total	Real Time	20252.54	30369.884	24
	8 Hrs DL	17068.08	15030.191	24
	16 Hrs DL	19465.79	14919.920	24
	Total	18928.81	21117.179	72

Table 14. Descriptive Statistics (Case 2 of experiment 1)

Dependent Variable: Deviation of Order Pool Size

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	20175158176[a]	8	2521894772	13.832	.000	.637
Intercept	25797576943	1	2.580E+10	141.495	.000	.692
CONTROL	14407638258	2	7203819129	39.512	.000	.556
DELAY	132070045.5	2	66035022.8	.362	.698	.011
CONTROL* DELAY	5635449872	4	1408862468	7.727	.000	.329
Error	11486243733	63	182321329			
Total	57458978852	72				
Corrected Total	31661401909	71				

a. R Squared = .637 (Adjusted R Squared = .591)

Table 15. ANOVA Table (Case 2 of experiment 1)

In the same way, we do contrast tests to compare the mean of three levels of information control. Contrast 1 checks the difference between controlling marketing and controlling both marketing and production. Contrast 2 looks at the difference between controlling marketing and production. Contrast 3 examines the difference between controlling both marketing and production and controlling production. The results are shown in Table 16.

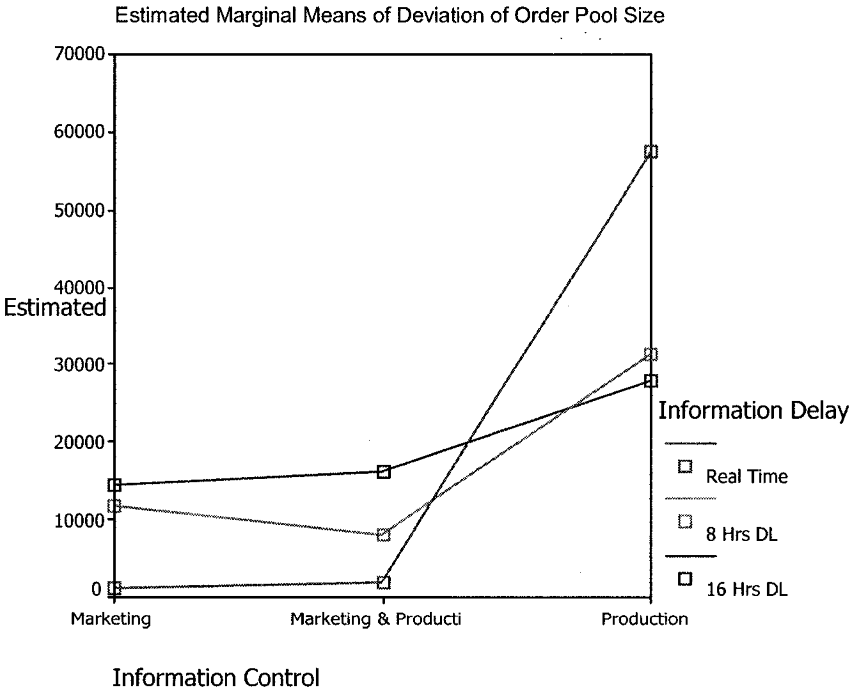


Figure 19. The Interaction plot (Case 2 of experiment 1)

Levene's Test is significant (F= 18.53, p< 0.001). Hence the variances are significantly different and we need to check the second half of table of contrast tests. After examining the result of contrast tests, we can find that, because p = 0.90, there is no significant difference between controlling marketing and controlling both marketing and production. However, controlling marketing only will achieve significantly (p<0.001) less fluctuation in order pool than controlling production only. Similarly, controlling both marketing and production will get significantly (p< 0.001) less fluctuation in order pool than controlling production only.

		Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
Deviation of Order Pool Size	Assume equal variances	1	305.17	4564.853	.067	69	.947
		2	-29854.21	4564.853	-6.540	69	.000
		3	-30159.38	4564.853	-6.607	69	.000
	Does not assume equal variances	1	305.17	2481.290	.123	45.983	.903
		2	-29854.21	5302.761	-5.630	28.454	.000
		3	-30159.38	5313.903	-5.676	28.662	.000

Table 16. Contrast tests (Case 2 of experiment 1)

Similarly, we try to identify the impact of information delay by dropping data regarding the third level of information control. The results, shown in Table 17, are similar to that of Case 1, i.e. information delay is significantly related to decision performance at p < 0.001. More specifically, we make three contrasts tests. Contrast 1 compares real-time information with 8 hours delay of information. Contrast 2 compares real-time information with 18 hours delay of information, and contrast 3 compares 8 hours' delay with 16 hours' delay. Table 18 shows that real-time information can achieve significantly better decision performance than delayed information (both 8 hours and 16 hours delay) at p < 0.001. Information delay of 8 hours can achieve better decision performance than that of 16 hours at p < 0.1.

Dependent Variable: Deviation of Order Pool Size

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	1573660569^a	5	314732113.8	7.239	.000	.463
Intercept	3824791014	1	3824791014	87.974	.000	.677
CONTROL	1117520.333	1	1117520.333	.026	.873	.001
DELAY	1507264270	2	753632134.9	17.334	.000	.452
CONTROL*DELAY	65278778.792	2	32639389.40	.751	.478	.035
Error	1826011063	42	43476453.88			
Total	7224462646	48				
Corrected Total	3399671632	47				

a. R Squared = .463 (Adjusted R Squared = .399)

Table 17. ANOVA table (Case 2 sub dataset)

		Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
Deviation of Order Pool Size	Assume equal variances	1	-8319.38	2292.747	-3.629	45	.001
		2	-13614.69	2292.747	-5.938	45	.000
		3	-5295.31	2292.747	-2.310	45	.026
	Does not assume equal variances	1	-8319.38	990.289	-8.401	15.966	.000
		2	-13614.69	2633.435	-5.170	15.133	.000
		3	-5295.31	2802.572	-1.889	19.051	.074

Table 18. Contrast tests (Case 2 with sub dataset)

Therefore, the results of Case 2 are identical to those of Case 2, i.e.:

1) The level of information control could significantly affect decision performance. The decision performance of controlling marketing only is significantly better than that of controlling production. The decision performance of controlling both marketing and production is significantly better than that of controlling production only. However, there is no significant difference between controlling marketing only and controlling both marketing and production.

2) Under most situations (except controlling production only), information delay is significantly related to decision performance. The less delay the required information, the better decision performance.

9.2. Information delay and information control

Case 1: Order arrivals follow Poisson distribution and production time follows exponential distribution.

Table 19 shows the sample size, mean, and standard deviation of dependent variable, i.e. the deviation of order pool size under two levels of information delay and four levels of information feedback. In the ANOVA Table 20, we can find that there is no significant interaction between information delay and information feedback ($F = 0.012$, $p = 0.998$). Furthermore, it shows that information feedback is significantly related to decision performance ($p = 0.1$). In this case, eta for information feedback is about 0.32 ($\sqrt{0.105} = 0.32$). According to Cohen (1988), because eta > 0.31, it is a large effect. However, information delay is insignificant in this case. This result is also illustrated in the interaction plot shown in Figure 20 .

Dependent Variable: DEV

DELAY	FEEDBACK	Mean	Std. Deviation	N
1	1	71245.87	52562.131	8
	2	111710.25	82555.046	8
	3	66612.25	48256.567	8
	4	114077.38	84035.620	8
	Total	90911.44	69177.904	32
10	1	74371.00	41714.121	8
	2	112044.62	74293.599	8
	3	69971.13	42749.181	8
	4	109666.75	73845.413	8
	Total	91513.38	60602.822	32
Total	1	72808.44	45868.641	16
	2	111877.44	75870.314	16
	3	68291.69	44074.520	16
	4	111872.06	76456.405	16
	Total	91212.41	64514.378	64

Table 19. Descriptive statistics (Case 1 of Experiment 2)

Dependent Variable: DEV

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	27649340695[a]	7	3949905814	.943	.481	.105
Intercept	5.3246E+11	1	5.325E+11	127.121	.000	.694
DELAY	5797260.063	1	5797260.063	.001	.970	.000
FEEDBACK	27486885227	3	9162295076	2.187	.100	.105
DELAY * FEEDBACK	156658208.2	3	52219402.73	.012	.998	.001
Error	2.3456E+11	56	4188629902			
Total	7.9467E+11	64				
Corrected Total	2.6221E+11	63				

[a] R Squared = .105 (Adjusted R Squared = -.006)

Table 20. ANOVA Table (Case 1 of Experiment 2)

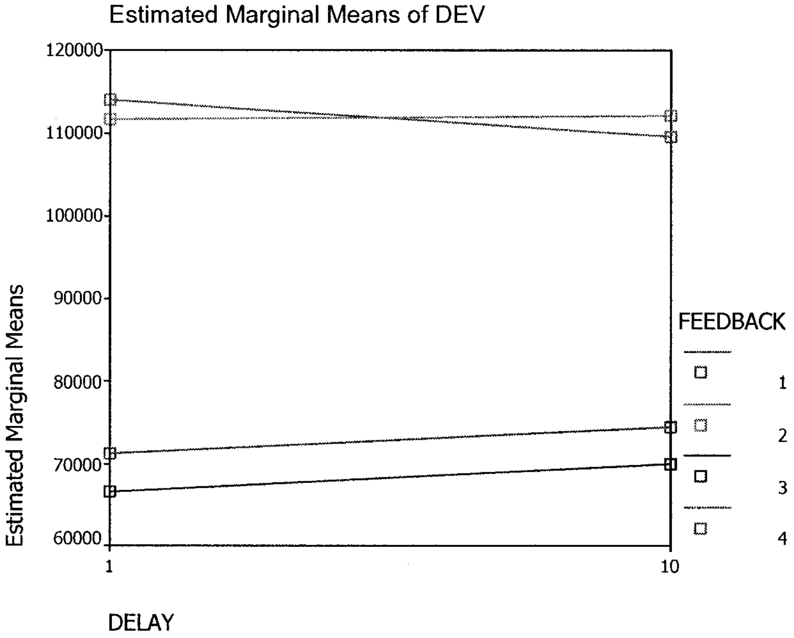


Figure 20. The interaction plot (Case 1 of experiment 2)

To identify how the level of information feedback affects decision performance, we do three contrasts test. Contrast 1 compares full information feedback with no information feedback and contrast 2 compares information feedback from production with no information feedback, while contrast 3 compares information feedback from marketing with no information feedback. The result is shown in Table 21. Both Contrast 1 and 3 are significant at $p < 0.1$, but contrast 2 is insignificant. It means that receiving information feedback from marketing department only does not improve decision performance significantly compared to no information feedback at all. On the contrary, receiving information feedback from production or from both marketing and production can significantly improve decision performance compared to no information feedback.

		Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
DEV	Assume equal variances	1	-39063.63	22113.614	-1.766	60	.082
		2	5.38	22113.614	.000	60	1.000
		3	-43580.38	22113.614	-1.971	60	.053
	Does not assume equal variances	1	-39063.63	22290.012	-1.753	24.559	.092
		2	5.38	26928.013	.000	29.998	1.000
		3	-43580.38	22062.617	-1.975	23.978	.060

Table 21. Contrast tests (Case 1 of experiment 2)

Case 2: Order arrivals follow Poisson distribution and production time follows exponential distribution.

Table 22 shows the sample size, mean, and standard deviation of dependent variable, i.e. the deviation of order pool size under two levels of information delay and four levels of information feedback. In the ANOVA Table 23, we can find that there is no significant interaction between information delay and information feedback ($F = 0.405$, $p = 0.750$). Furthermore, it shows that information feedback is significantly related to decision performance at $p < 0.001$.

In this case, eta for information feedback is about 0.58 ($\sqrt{0.339} = 0.58$). According to Cohen (1988), because eta > 0.31, it is a large effect. However, information feedback is insignificant in this case. It means that real-time information can lead to significantly better decision performance than delayed information. This result is also illustrated in the interaction plot shown in Figure 21 .

Dependent Variable: DEV

DELAY	FEEDBACK	Mean	Std. Deviation	N
1	1	2043.25	951.624	8
	2	3467.88	2132.844	8
	3	1799.00	590.278	8
	4	3744.25	2227.729	8
	Total	2763.59	1783.590	32
10	1	5602.63	2258.378	8
	2	5464.63	1885.141	8
	3	4950.88	1790.741	8
	4	6392.63	3637.740	8
	Total	5602.69	2437.614	32
Total	1	3822.94	2486.197	16
	2	4466.25	2201.023	16
	3	3374.94	2075.629	16
	4	5068.44	3218.976	16
	Total	4183.14	2556.613	64

Table 22. Descriptive Statistics (Case 2 of Experiment 2)

Dependent Variable: DEV

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	160766909.4[a]	7	22966701.34	5.124	.000	.390
Intercept	1119914591	1	1119914591	249.843	.000	.817
DELAY	128967253.1	1	128967253.1	28.771	.000	.339
FEEDBACK	26349440.922	3	8783146.974	1.959	.131	.095
DELAY * FEEDBACK	5450215.297	3	1816738.432	.405	.750	.021
Error	251018250.4	56	4482468.757			
Total	1531699751	64				
Corrected Total	411785159.7	63				

a. R Squared = .390 (Adjusted R Squared = .314)

Table 23. ANOVA table (Case 2 of experiment 2)

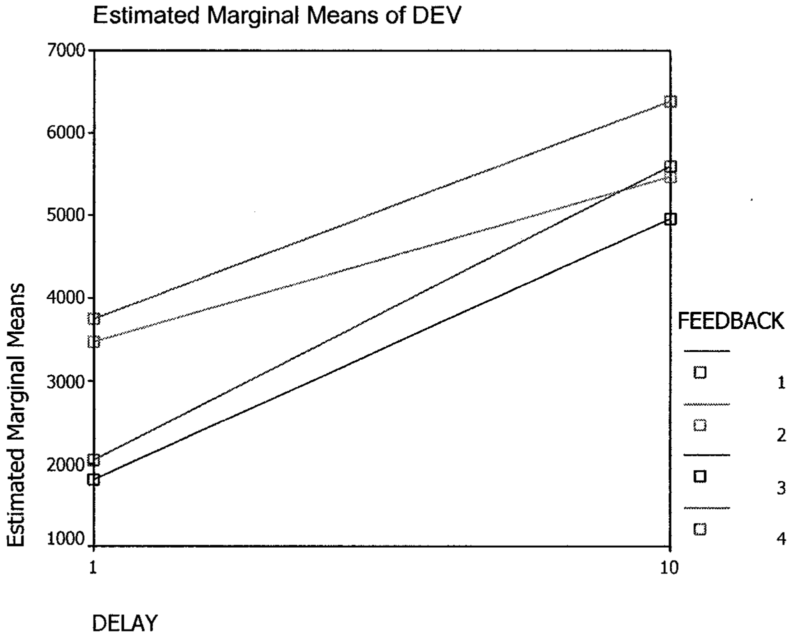


Figure 21. The interaction plot (Case 2 of experiment 2)

Like in case 1, we also do contrast test to further examine the impact of information feedback. Again the results in Table 24 shows that only contrast 3 are significant at $p < 0.1$.

		Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
DEV	Assume equal variances	1	-1245.50	896.098	-1.390	60	.170
		2	-602.19	896.098	-.672	60	.504
		3	-1693.50	896.098	-1.890	60	.064
	Does not assume equal variances	1	-1245.50	1016.827	-1.225	28.199	.231
		2	-602.19	974.882	-.618	26.510	.542
		3	-1693.50	957.537	-1.769	25.635	.089

Table 24. Contrast Tests (Case 2 of Experiment 2)

9.3. Discussion

In the above analysis, our hypotheses are partially supported. It is logical to think that real-time information will improve decision performance. However, it is not always true. When we only control production capacity, the real-time information cannot improve decision performance significantly. One reason is that we cannot change production capacity immediately. It usually takes eight hours to finish one batch, and then we have a chance to modify production capacity. For this reason, even we can make timely decision with timely information, we cannot implement it timely. Hence, the timely information becomes useless.

On the contrary, because we immediately change the product price on the web and thus change order arrival rate. The timely information, the timely decision, and the timely implementation certainly improve decision performance.

Therefore, we can say that real-time information does not mean better decision performance. It depends on the nature of the problem. It implies that DSS designer cannot assume that the faster the better. The effective DSS application should be tailored to the actual requirement. It should provide right person with right information at right time. Real-time information may overwhelm decision makers. Furthermore, real-time information may destabilize the systems, if decision makers intervene too much. This point is also evidenced by our simulations results.

In our case, the level of information control significantly influences decision performance. However, it is not as expected that the higher the level of information, the better the decision performance. Controlling both marketing and production capacity is significantly worse than controlling marketing only but it is significantly better than controlling production only. Therefore, it shows the benefits of information coordination are inconclusive. They also depend on problem structure. But our point is, a well-designed distributed DSS can help improve decision performance by the aid of information coordination.

We also find that information feedback, no matter whether it is full feedback or partial feedback, can achieve significantly better decision performance than no information feedback. However, information feedback from both marketing and production department does not achieve significantly better decision performance than information feedback from either marketing or production only..

10 CONCLUSION

Distributed decision making (DDM) is a complex process and is largely ignored in the research on decision support systems. The current group decision support research, such as GDSS, ODSS, and distributed DSS, does not cover the full spectrum of DDM. Our research bridges this gap by introducing a conceptual framework of integrated decision stations to support dynamic distributed decision making process. Our idea of decomposition of decision criteria is largely drawn from Schneeweiss' (2003a) formal description of DDM. In addition, we propose information decomposition to facilitate information coordination between/among decision stations. We also bring the concept of situated decision support systems(Vahidov & Kersten, 2004) into our framework to handle dynamic environment. Basing upon all of the previous work, one major contribution of our work is that, it provides an integrated view on distributed decision making within and/or across organizations from information processing perspective. We hope our conceptual work can 1) promote academic researchers to build a more solid theoretical foundation for DSS research; 2) support DSS parishioners such as systems analysts and DSS designers to systematically understand organizational decision structure and build an effective DSS in alignment with this structure.

In addition, the applicability of this framework is demonstrated with a prototype for lead management. It implements four types of decision units, both inside and outside of the company. Two coordination process, i.e. top-down planning process and peer-to-peer negotiation process are also demonstrated in this prototype. It shows that our model can be acted as a strong base of developing various distributed decision making systems.

Furthermore, we conduct a simulation study to evaluate the potential benefits of integrated decision stations. We examine two distinct features of IDS compared with traditional DSS. One is the real-time information captured by situated sensors. The other is information coordination. Our results show that real-time information can improve decision performance when the decision derived from this information can immediately be implemented. However, if a decision cannot be implemented immediately, the benefit of real-time information is not significant.

The level of information control and the level of information feedback also have significant influence on decision performance. However, our results show that the higher level of information control does not necessarily have significant improve on decision performance. It shows that one type of information coordination can achieve significantly better decision performance than another one, though they have the same level of information coordination. For example, in our cases, coordinating with marketing department is significantly better than coordinating production department. We may conclude that information control and information feedback are beneficial if the decision derived from those information can be immediately implemented.

Overall, we believe that IDS can improve decision performance if it is well designed and fits with problem structure.

10.1. Limitations

The real DSS should be used by human decision makers. For this reason, the best way of evaluating DSS perhaps is to ask human subjects to solve a real decision problem using a DSS. However, due to the time and budget constraints, and the complexity of our business case, we choose simulation as an approach to conduct our research.

As a result, the biggest limitation is that our DSS is no longer a real DSS but an automatic system. We cannot examine its impact on human decision makers' behavior and their actual performance. We cannot examine whether distributed DSS improve their coordination, and we cannot investigate whether real-time information create burden on them.

The second limitation is that our simulation case might be too specific. Therefore, it is hard for us to generalize our results. The characteristics of our problem, i.e. lead time management, may put constraints on the behavior of our prototype systems.

The last and very important limitation is that our business problem may be too complex and theory behind this problem, i.e. workload control concept, is not yet mature. There is few business applications utilizing this concept.

10.2. Future research

Though we have built a conceptual framework to support distributed decision making, to make it works, we need to develop a technical architecture to support this framework. For this reason, it is worth to study how to use modern information technology, such as agent technologies, to build a solid and flexible structure supporting our conceptual framework. For example, information control and feedback can be realized with Agent Control Language (ACL) or Knowledge Query and Manipulation Language (KQML).

The second potential research is using human decision makers. We should develop a simple but well-studied business case, for example, the "Beer Game". So we have a good empirical foundation to test how the timely information and information coordination will affect game-players' performance. We can also investigate some behavior factors.

Dynamic distributed decision making and support systems are emerging areas. We hope our work will promote research interests in these areas. On the one hand, we can conduct technical-oriented research, for example building new DSS artifact. On the other hand, we can conduct behavioral-oriented research, for example, investigating how this new DSS artifact influences human decision makers. We believe both research have promising future.

REFERENCES:

Aggarwal, A. I., & Mirani, R. (1995). *Macro issues in the development of organizational decision support systems.* Paper presented at the the 28th Hawaii International Conference on System Sciences

Anthony, R. N. (1965). *Planning and Control Systems: A Framework for Analysis.* Boston: Harvard University Graduate School of Business Administration.

Barkhi, R., Jacob, V. S., Pipino, L., & Pirkul, H. (1998). A study of the effect of communication channel and authority on group decision processes and outcomes. *Decision Support Systems, 23*(3), 205.

Burns, A., Rathwell, M. A., & Thomas, R. C. (1987). A Distributed Decision-Making System. *Decision Support Systems, 3*(2), 121.

Chi, R. T., & Turban, E. (1995). Distributed intelligent executive information systems. *Decision Support Systems, 14*(2), 117.

Chung, H. M., Mahapatra, R., & Marin, G. (1993). *Distributed Decision Support Systems: Characterization and Design Choices.* Paper presented at the Twenty-sixth Annual Hawaii International Conference on System Sciences.

Cohen, J. (1988). *Statistical power and analysis for the behavioral sciences* (2nd ed.). Hillsdale,NJ: Lawrence Erlbaum Associates.

Davenport, T. H., & Brooks, J. D. (2004). Enterprise systems and the supply chain. *Journal of Enterprise Information Management, 17*(1), 8.

Eom, S. B. (2002). *Decision support systems research, 1970-1999 : a cumulative tradtion and reference disciplines*: The Edwin Mellen Press.

Gachet, A. (2002). *A New Vision for Distributed Decision Support Systems.* Paper presented at the DSIage 2002.

Gachet, A., & Haettenschwiler, P. (2003). A Decentralized Approach to Distributed Decision Support Systems. *Journal of Decision Systems, 12*(2), 141-158.

Gorry, G. A., & Morton, M. S. S. (1971). A framework for management information systems. *Sloan Management Review (pre-1986), 13*(1), 55.

Haskose, A., Kingsman, B. G., & Worthington, D. (2002). Modelling flow and jobbing shops as a queueing network for workload control. *78*(3), 271.

Haskose, A., Kingsman, B. G., & Worthington, D. (2004). Performance analysis of make-to-order manufacturing systems under different workload control regimes. *International Journal of Production Economics, 90*(2), 169.

Henrich, P., Land, M., & Gaalman, G. (2004). Exploring applicability of the workload control concept. *90*(2), 187.

Henrich, P., Land, M., Gaalman, G., & Zee, D.-J. V. D. (2004). Reducing feedback requirements of workload control. *International Journal of Production Research, 42*(24), 5235.

Jarvenpaa, S. L., Rao, V. S., & Huber, G. P. (1988). Computer Support for Meetings of Groups Working on Unstructured Problems: A Field Experiment. *MIS Quarterly, 13*, 645-665.

Jeusfeld, M. A., & Bui, T. X. (1997). Distributed decision support and organizational connectivity: A case study. *Decision Support Systems, 19*(3), 215.

Ju, C., Ling, Y., & Norman, T. J. (2000). *Agent-Based and Software Composite DDSS.* Paper presented at the Technology Object-orieted language system tools.

Keen, P. G. W., & Morton, M. S. S. (1978). *Decision Support Systems: An organizational Perspective*: Addison-Wesley.

Kim, G.-M. J., Graves, E. G., Burns, J. R., & Myung, R. (2000). An architecture for organizational decision support systems (ODSS) that utilizes a model coordination subsystem. *Journal of End User Computing, 12*(3), 23.

King, J. L., Star, S. L., & George, J. F. (1990). *Conceptual foundations for the development of organizational decision support systems*

The conceptualization and development of organizational decision support systems.

Kingsman, B. G. (2000). Modelling input-output workload control for dynamic capacity planning in production planning systems. *International Journal of Production Economics, 68*(1), 73.

Land, M. (2004). *Workload control in job shops, grasping the tap.*

Merriam-Webster. (2002a). Webster's Online Dictionary. Retrieved 10 Feb. , 2006, from http://www.m-w.com/cgi-bin/dictionary?va=problem

Merriam-Webster. (2002b). Webster's Third New International Dictionary, Unabridged. Retrieved 1 Dec. , 2005, from http://unabridged.merriam-webster.com

Power, D. J. (2003). A Brief History of Decision Support Systems. version 2.8. Retrieved Jan 9, 2006, from http://DSSResources.COM/history/dsshistory.html

Rasmussen, J., Brehmer, B., & Leplat, J. (Eds.). (1991). *Distributed decision making : cognitive models for cooperative work*: John Wiley & Sons Ltd.

Rathwell, M. A., & Burns, A. (1985). Information Systems Support for Group Planning and Decision-Making Activities. *MIS Quarterly, 9*(3), 255.

Scher, J. M. (1981). *Distributed Decision Suppport Systems for Management and Organizations.* Paper presented at the DSS-81 Transactions, Atlanta, GA.

Schneeweiss, C. (2003a). *Distributed Decision Making* (2nd ed.): Springer-Verlag.

Schneeweiss, C. (2003b). Distributed decision making - A unified appraoch. *European Journal of Operational Research, 150*(2), 237.

Scott Morton, M. S. (1971). *Management Decision Systems.* Boston: Harvard University Graduate School of Business Administration.

Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., & Carlsson, C. (2002). Past, present, and future of decision support technology. *Decision Support Systems, 33*(2), 111.

Simon, H. A. (1960). *The new science of management decision,* (1st ed.). New York: Harper.

Simon, H. A. (1977). *The new science of management decision*: Prentice-Hall, Inc.

Simon, H. A., Dantzig, G. B., Hogarth, R., Plott, C. R., Raiffa, H., Schelling, T. C., et al. (1987). Decision Making and Problem Solving. *Interfaces, 17*(5), 11.

Sprague, R. H., Jr. (1980). A Framework for the Development of Decision Support Systems *MIS Quarterly, 4*(4), 1-26.

Swanson, E. B. (1990). *Distributed decision support systems: a perspective.* Paper presented at the Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences.

Tseng, C.-C., & Gmytrasiewicz, P. J. (2002). *Real Time Decision Support System for Portfolio Management.* Paper presented at the the 35th Hawaii International Conference on System Sciences.

Vahidov, R., & Kersten, G. E. (2004). Decision station: situating decision support systems. *Decision Support Systems, 38*(2), 283.

Warkentin, M., Bapna, R., & Sugumaran, V. (2000). THE ROLE OF MASS CUSTOMIZATION IN ENHANCING SUPPLY CHAIN RELATIONSHIPS IN B2C E-COMMERCE MARKETS. *Journal of Electronic Commerce Research,, 1*(2), 45-52.

Weiss, G. (Ed.). (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*: The MIT Press.

Wight, O. (1970). Input/output control: a real handle on lead time. *Production and Inventory Managment, 11*(3), 9-31.

Wilson, R. A., & Keil, F. C. (2001). *The MIT Encyclopedia of the Cognitive Sciences*: MIT Press.

Wind, J., & Rangaswamy, A. (2001). Customerization: The next revolution in mass customization. *Journal of Interactive Marketing, 15*(1), 13.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review, 10*(2), 115-152.

Appendix A: The data of Case 1 (Experiment 1)

DEMAND	CORD	DELAY	DEV_SIZE	CELLCODE
1	1	1	7492	1
2	1	1	9935	1
3	1	1	1756	1
4	1	1	65343	1
5	1	1	4547	1
6	1	1	21771	1
7	1	1	10918	1
8	1	1	11671	1
1	1	10	15784	2
2	1	10	16570	2
3	1	10	18415	2
4	1	10	71992	2
5	1	10	15857	2
6	1	10	36390	2
7	1	10	87937	2
8	1	10	36324	2
1	1	20	43040	3
2	1	20	63193	3
3	1	20	34099	3
4	1	20	78366	3
5	1	20	20551	3
6	1	20	42544	3
7	1	20	112241	3
8	1	20	61950	3
1	2	1	14875	4
2	2	1	11395	4
3	2	1	6876	4
4	2	1	69789	4
5	2	1	7840	4
6	2	1	20241	4
7	2	1	43884	4
8	2	1	14043	4
1	2	10	34096	5
2	2	10	24257	5
3	2	10	27593	5

4	2	10	111018	5
5	2	10	60833	5
6	2	10	66184	5
7	2	10	44412	5
8	2	10	57946	5
1	2	20	41934	6
2	2	20	18980	6
3	2	20	33913	6
4	2	20	104486	6
5	2	20	72294	6
6	2	20	60061	6
7	2	20	58365	6
8	2	20	60692	6
1	3	1	57302	7
2	3	1	184664	7
3	3	1	81858	7
4	3	1	204882	7
5	3	1	36962	7
6	3	1	170582	7
7	3	1	162588	7
8	3	1	28442	7
1	3	10	44471	8
2	3	10	195570	8
3	3	10	117135	8
4	3	10	189504	8
5	3	10	39740	8
6	3	10	151725	8
7	3	10	151221	8
8	3	10	25628	8
1	3	20	51378	9
2	3	20	189951	9
3	3	20	113918	9
4	3	20	188279	9
5	3	20	45953	9
6	3	20	134490	9
7	3	20	152279	9
8	3	20	24935	9

Appendix B: The data of Case 2 (Experiment 1)

DEMAND	CORD	DELAY	DEV_SIZE	CELLCODE
1	1	1	1075	1
2	1	1	1286	1
3	1	1	1208	1
4	1	1	1043	1
5	1	1	1232	1
6	1	1	1308	1
7	1	1	1128	1
8	1	1	1052	1
1	1	10	15553	2
2	1	10	10061	2
3	1	10	10710	2
4	1	10	19118	2
5	1	10	8219	2
6	1	10	5226	2
7	1	10	10281	2
8	1	10	14574	2
1	1	20	8480	3
2	1	20	29038	3
3	1	20	14881	3
4	1	20	4844	3
5	1	20	6565	3
6	1	20	6523	3
7	1	20	13242	3
8	1	20	31252	3
1	2	1	1787	4
2	2	1	2205	4
3	2	1	1689	4
4	2	1	1884	4
5	2	1	3581	4
6	2	1	1509	4
7	2	1	1175	4
8	2	1	2681	4
1	2	10	8015	5
2	2	10	8208	5
3	2	10	11898	5

4	2	10	9810	5
5	2	10	4155	5
6	2	10	6086	5
7	2	10	9514	5
8	2	10	7525	5
1	2	20	13424	6
2	2	20	38028	6
3	2	20	7422	6
4	2	20	7155	6
5	2	20	11004	6
6	2	20	5681	6
7	2	20	26243	6
8	2	20	19896	6
1	3	1	52876	7
2	3	1	102259	7
3	3	1	25392	7
4	3	1	39570	7
5	3	1	82742	7
6	3	1	45355	7
7	3	1	69561	7
8	3	1	42463	7
1	3	10	31968	8
2	3	10	61100	8
3	3	10	14796	8
4	3	10	9937	8
5	3	10	58624	8
6	3	10	22019	8
7	3	10	29404	8
8	3	10	22833	8
1	3	20	13153	9
2	3	20	64559	9
3	3	20	17437	9
4	3	20	8409	9
5	3	20	49875	9
6	3	20	22060	9
7	3	20	25327	9
8	3	20	22681	9

Appendix C: The data of Case 1 (Experiment 2)

Serial	Delay	Feedback	Dev
1	1	1	32923
1	1	2	52202
1	1	3	35459
1	1	4	54886
2	1	1	146822
2	1	2	235621
2	1	3	136205
2	1	4	233266
3	1	1	16148
3	1	2	28948
3	1	3	16712
3	1	4	21573
4	1	1	127874
4	1	2	191867
4	1	3	128466
4	1	4	189769
5	1	1	18104
5	1	2	27080
5	1	3	13718
5	1	4	29271
6	1	1	84004
6	1	2	140207
6	1	3	78069
6	1	4	154443
7	1	1	110311

7	1	2	170308
7	1	3	86651
7	1	4	180340
8	1	1	33781
8	1	2	47449
8	1	3	37618
8	1	4	49071
1	10	1	47447
1	10	2	58770
1	10	3	39439
1	10	4	54176
2	10	1	143040
2	10	2	216578
2	10	3	132028
2	10	4	219154
3	10	1	41456
3	10	2	35237
3	10	3	27681
3	10	4	29310
4	10	1	125409
4	10	2	179281
4	10	3	123530
4	10	4	175835
5	10	1	32457
5	10	2	37366
5	10	3	23428
5	10	4	38829
6	10	1	69695

6	10	2	140536
6	10	3	83578
6	10	4	152168
7	10	1	92074
7	10	2	177747
7	10	3	86862
7	10	4	155795
8	10	1	43390
8	10	2	50842
8	10	3	43223
8	10	4	52067

Appendix D: The data of Case 2 (Experiment 2)

Serial	Delay	Feedback	Dev
1	1	1	3026
1	1	2	3987
1	1	3	2353
1	1	4	4612
2	1	1	3419
2	1	2	7676
2	1	3	2427
2	1	4	7930
3	1	1	1770
3	1	2	1978
3	1	3	1767
3	1	4	2167
4	1	1	3031
4	1	2	5447
4	1	3	2479
4	1	4	5776
5	1	1	1067
5	1	2	2156
5	1	3	1183
5	1	4	2392
6	1	1	1302
6	1	2	1944
6	1	3	1851
6	1	4	2730
7	1	1	1298

7	1	2	2930
7	1	3	1379
7	1	4	3188
8	1	1	1433
8	1	2	1625
8	1	3	953
8	1	4	1159
1	10	1	4807
1	10	2	6760
1	10	3	4249
1	10	4	5754
2	10	1	10391
2	10	2	8932
2	10	3	8811
2	10	4	11599
3	10	1	3332
3	10	2	4014
3	10	3	3728
3	10	4	4109
4	10	1	5943
4	10	2	7131
4	10	3	6243
4	10	4	12713
5	10	1	6384
5	10	2	4216
5	10	3	4703
5	10	4	4054
6	10	1	5802

6	10	2	4434
6	10	3	4586
6	10	4	3964
7	10	1	3190
7	10	2	4238
7	10	3	3272
7	10	4	3691
8	10	1	4972
8	10	2	3992
8	10	3	4015
8	10	4	5257