AVAILABILITY MANAGEMENT IN CONFIGURE-TO-ORDER MANUFACTURING SYSTEMS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

MIDDLE EAST TECHNICAL UNIVERSITY

BY

HÜSEYİN ERDEM YÖNTEM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
OPERATIONAL RESEARCH

DECEMBER 2009

Approval of the thesis:

AVAILABILITY MANAGEMENT IN CONFIGURE-TO-ORDER MANUFACTURING SYSTEMS

submitted by **HÜSEYİN ERDEM YÖNTEM** in partial fulfillment of the requirements for the degree of **Master of Science in Operational Research, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____________

Prof. Dr. Nur Evin Özdemirel
Head of Department, **Industrial Engineering Department** _____________

Asst. Prof. Dr. Sedef Meral
Supervisor, **Industrial Engineering Department, METU** _____________

Examining Committee Members

Assoc. Prof. Dr. Canan Sepil _____________
Industrial Engineering Department, METU

Asst. Prof. Dr. Sedef Meral _____________
Industrial Engineering Department, METU

Prof. Dr. Meral Azizoğlu _____________
Industrial Engineering Department, METU

Asst. Prof. Dr. Ferda Can Çetinkaya _____________
Industrial Engineering Department, Çankaya University

M.Sc. Şakir Karakaya _____________
National Productivity Center

Date: 11.12.2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name	**:** Hüseyin Erdem YÖNTEM
Signature	**:**

ABSTRACT

AVAILABILITY MANAGEMENT IN CONFIGURE-TO-ORDER MANUFACTURING SYSTEMS

Yöntem, Hüseyin Erdem
M.Sc., Operational Research
Supervisor: Asst. Prof. Dr. Sedef Meral

December 2009, 98 pages

In resource constrained supply chains, where demand is higher than the supply, the decision whether to accept or reject the customer order is a very critical task from resource planning and customer service level perspectives. Since the customers, in today's e-business environment, expect quick responses to their orders, some in-advance work has to be done before the arrival of actual customer orders, especially in configure-to-order (CTO) and make-to-order (MTO) production systems.

Available-to-Promise (ATP) is a business function that is becoming the central management system for today's dynamic supply chains whose responsibility is to respond customer orders by considering the trade-off between front-end customer satisfaction and back-end capacity allocation. In this study, we propose an availability management approach that introduces push-based allocation planning by using order segmentation before the arrival of actual customer orders in CTO production environments. Moreover, a two-step order promising framework is introduced in order to increase customer service levels through giving certain or tentative delivery dates immediately to customer orders before the batch, rule-based actual resource consumption processes.

The proposed approach is applied to the real-life processes of an enterprise in order to analyze its applicability and evaluate the benefits that accrue. The results of the experiments prove that, the four-phased availability management approach contribute to both overall profit and customer service levels.

Keywords: Availability Management, Configure-to-Order, Available-to-Promise, Order Segmentation, Clustering, Allocation Planning

ÖZ

SİPARİŞE GÖRE YAPILANDIRILAN ÜRETİM SİSTEMLERİNDE UYGUNLUK YÖNETİMİ

Yöntem, Hüseyin Erdem
Yüksek Lisans, Yöneylem Araştırması
Tez Yöneticisi: Y.Doç. Dr. Sedef Meral

Aralık 2009, 98 sayfa

Talebin arzdan daha fazla olduğu kaynak kısıtlı tedarik zincirlerinde, gelen bir müşteri siparişini kabul etmek veya red etmek kaynak planlaması ve müşteri servis seviyeleri perspektifine göre çok kritik bir iştir. Günümüzdeki e-iş ortamında, müşteriler siparişlerine çok kısa bir süre içerisinde cevap almak istemektedirler. Bu nedenle özellikle siparişe göre yapılandırılan ve siparişe istinaden üretim yapılan üretim sistemlerinde, müşteri siparişleri gelmeden once bazı ön çalışmaların yapılması gerekmektedir.

Söz Vermeye Uygun (SVU), ön yüzdeki müşteri memnuniyeti ve arka yüzdeki kapasite tahsisi arasındaki dengeyi kurma sorumluluğu ile; günümüzün dinamik tedarik zincirlerinin merkezi yönetim sistemi olmaya başlamaktadır. Bu çalışmada, siparişe göre yapılandırılan üretim sistemlerinde, müşteri siparişleri henüz gelmeden, sipariş bölümlemesi ile itiş-tabanlı tahsis planlaması uygulayan; bir uygunluk yönetimi yaklaşımı sunulmuştur. Buna ek olarak, kural tabanlı parti kaynak tüketimi sürecinden önce müşterilere kesin veya yaklaşık sonuçların anında verilebilmesine olanak sağlayan iki aşamalı bir sipariş sözverme çatısı geliştirilmiştir.

Gelistirilen çözüm, bir kurumun gerçek hayattaki süreçleri üzerinde uygulanmış ve uygulanabilirliği ve oluşabilecek faydalar incelenmiştir. Deney sonuçlarına göre, sunulan dört aşamalı uygunluk yönetimi çözümü, hem toplam kara hem de müşteri hizmet seviyelerine katkı sağlamaktadır.

Anahtar Kelimeler: Uygunluk Yönetimi, Siparişe İstinaden Yapılandırma, Söz-Vermeye-Uygun, Sipariş Bölümlemesi, Kümeleme, Tahsis Planlama

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor Asst. Prof. Dr. Sedef Meral, not only for her endless supervision, but also for the encouragement she provided me for research. Studying with her was a great experience for me.

I would also want to thank Assoc. Prof. Dr. Canan Sepil, Prof. Dr. Meral Azizoğlu, Asst. Prof. Dr. Ferda Can Çetinkaya and M.Sc. Şakir Karakaya for their constructive comments on my thesis. I am very grateful that they accepted to be the members of my examining committee.

I would also present my special thanks to Dr. Tom Ervolina from IBM Watson Research Center, for his guidance and comments on my study.

I would also want to thank to my brother Şevki Yöntem for his help during my operational research studies.

I would also want to thank my friend Fehmi Çivitci for his help during my literature survey.

I would also want to thank my family for their support in my hard times.

Finally, I would like to thank my managers and colleagues in Meteksan Sistem and IBM Turk for their support and tolerance.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Advanced information technologies and incredible progress in e-business and data processing systems are reshaping today’s global business environment. Customers are demanding more and more customized products and they are able to investigate different suppliers easily in terms of price and service levels. In order to survive in such a competitive environment companies are trying to structure their supply chains in order to minimize costs, maximize profits and service levels. This is the reason why effective supply chain management has gained so much importance in today’s companies.

In resource constrained supply chains, determining which orders to accept and more importantly which orders to reject is one of the most important decisions that companies face today. Available-to-Promise (ATP) is a business function having the capability to respond to customer requests by matching them with enterprise resources while evaluating the trade-off between front-end customer satisfaction and back-end logistics performance.

The difficulty of responding to customer orders changes according to the position of the decoupling point and supply chain strategy. In make-to-stock (MTS) supply chains, promising customer orders needs a simple database lookup for the inventory position. On the other hand in configure-to-order (CTO) and make-to-order (MTO) supply chains, raw material and resource availability, assembly and production capacities and other supply chain attributes should be considered in order to promise customer orders.

Capable to promise (CTP), similar to ATP, is an output of master production schedule (MPS) that is the available production or assembly capacity and resources that can be used to promise customer orders. MPS systems calculate the optimum (or near optimum) amounts of ATP and CTP quantities by considering the resources of the company and forecasted customer demands. On the other hand, ATP and CTP execution results can be used as an input to the MPS in order to feed MPS with the actual business runtime environment. Modern APS (advanced planning and scheduling) systems are paying more and more attention to ATP systems and position ATP with the MPS as the central management systems of the supply chains.

Revenue management ideas, which are originated from the airline and service industries, can be applied to manufacturing environments as well by differentiating demand classes and responding them differently. Especially in resource constrained supply chains, some allocation mechanisms should be developed in order to avoid low profitable demand groups to over-consume some scarce resources that will result in order rejections from high profitable demand groups.

In this study, we propose an availability management approach that optimizes profits and service levels by considering the heterogeneity of orders, products and customers in a CTO production environment. As a long term planning stage; order segmentation is executed based on order profit, customer value and product characteristics. A medium term planning stage; detailed allocation of resources to order segments in terms of product families and components is proposed. Having the order segments and allocated ATPs and CTPs to them, a two-stage order promising model is executed to quickly respond to customer orders at the first stage while optimizing the consumption of resources at the second stage.

The proposed approach is applied to the IBM’s Enterprise Server Hardware Division processes in order to analyze its applicability and evaluate the benefits that accrue. The experiments and sensitivity analyses on IBM’s CTO server production environment prove the insights of the approach from profit, customer satisfaction and service levels perspectives.

One of the main strengths of the four-staged availability management approach we propose is its applicability to different production environments and companies by its modular structure and loosely coupled design. Each stage can be replaced by a custom approach while respecting to its inputs and outputs which are clearly defined.

In the following chapters, the details of our study are presented regarding the below outline.

In Chapter 2, a comprehensive literature review is given. Firstly, a general introduction to availability management and ATP is given. Then ATP systems are classified according to their availability level, operation mode, functional role in order promising systems and some other dimensions. After that, the decoupling point concept, push-based, pull-based and push-pull integrated ATP systems are introduced. Then optimization based ATP systems and their properties are investigated and some of them are compared. After that, customer-segmentation and revenue management concepts are studied and their applicability to manufacturing environments is defined. Then properties of IT systems that should be constructed in order to satisfy ATP needs are defined and some technologies are introduced. Lastly, APS approach to ATP Systems is examined and their modules are introduced.

In Chapter 3, the problem definition is given. After giving some general definitions for CTO environments, the motivation of the study; IBM Enterprise Server Hardware division and its processes are introduced. Then the ways to apply order segmentation and resource allocation are discussed and the applicability is examined. The advantages and disadvantages of real-time and batch ATP execution are discussed and a two-stage order promising approach is introduced to merge the advantages of both approaches. Lastly, the general structure and assumptions of the four-staged solution approach are proposed.

In Chapter 4, the four-staged solution approach is proposed with a general introduction to the methodology and a detailed explanation for each stage. Besides the individual stage explanations, the relations between the stages are defined. For Stage 0, orders are segmented according to customer priority, product complexity and order profit. After this segmentation a clustering method is applied in order to generate order classes that are passed to Stage 1. At Stage 1, ATP allocation for order segments is done according to demands of order segments, global ATP and CTP quantities and order segment's respective values (priorities) coming from stage 0. At Stage 2, online (very short term) ATP execution is done in order to quickly respond to customer with a positive or a negative reply according to resource availability. The customer orders are either rejected or responded with hard promises (with exact delivery date) or soft promises (with delivery time window). Soft promised orders are passed to Stage 3, where a batch mode ATP execution is done to optimize the resource consumption during the batching horizon.

In Chapter 5, the solution approach is illustrated via IBM Enterprise Server case and CTO processes. Several experiments and scenarios are constructed in order to define the strengths and weaknesses of the proposed solution approach.

In Chapter 6, we conclude our study briefly by highlighting the important outcomes of our approach. Moreover, we propose some directions for future research.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we discuss several characteristics of availability management, available-to-promise, order promising, order fulfillment and present a comprehensive literature review about them.

2.1. Introduction to Available-to-Promise (ATP)

Incredible developments in information technology (IT), data processing and logistics infrastructures are reshaping the global business environments. Buyers and sellers are sharing information and decisions in real time, while products can move from one place to another in days or even in hours. Now, buyers have the ability to investigate different products and sellers online, and compare them easily. In order to survive in such a competitive environment, companies have to redefine their business processes not only to enhance back-end logistics infrastructure but also to improve front-end customer service and satisfaction. Available-to-Promise (ATP) is one of the most important business processes of supply chain management (SCM) that plays an important role of directly linking customer orders with enterprise resources and evaluate the tradeoff between front-end customer service and back-end logistics performance (Ball et al., 2004).

According to Ervolina and Dietrich (2001), the first introduction to ATP comes in the late 1980's with MRPII (Manufacturing Resource Planning). The early definition was simple that takes the total availability from the Master Production Schedule (MPS), reduce it by the actual customer orders and leaves the planner with the amount ATP. Enterprise Resource Planning (ERP) systems integrate the order system with the MPS so that the ATP is decremented by every accepted customer order or reset when a new MPS is generated.

According to Chen (2003) and Chen et al. (2009), ATP is an advanced execution and planning mechanism that connects customer orders and enterprise resources by performing two order management functions: order promising and order fulfillment (also known as demand fulfillment) (Figure 2-1). When a customer order is received, the firm has to promise the order with specific delivery time, quantity, product configuration or even price. On the other hand, fulfilling this promised order may require complicated production and distribution operations (Figure 2-2).

According to Ervolina and Dietrich (2001), ATP will continue to grow and will become the central management system of a supply chain. They also divide their ATP system called Availability Management System (AMS) into two core engine components; the promising engine (front-end) and the availability planning engine (back-end).

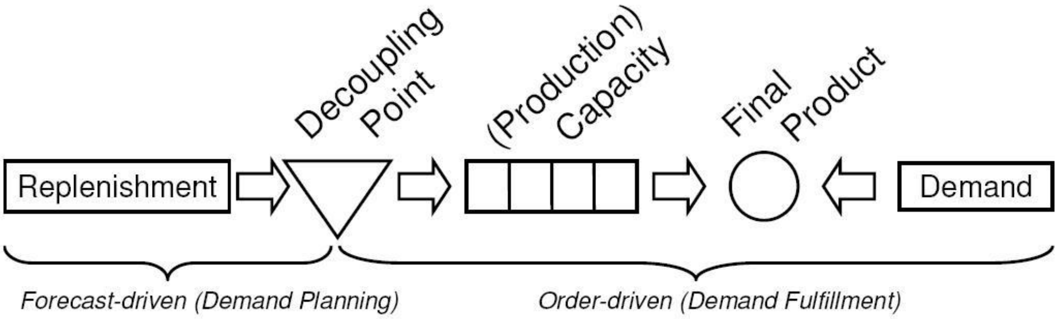


Figure 2-1: Supply chain framework (Quante et al., 2007)

According to Chen et al. (2001), the customer order can have two flexible dimensions: quantity and due date. Hence, they classify ATP execution algorithms into three groups: quantity promising, due date promising, quantity and due date promising. At quantity promising, the customer specifies a range of acceptable quantities with a certain due date. At due date promising, the customer specifies a range of acceptable due dates with a certain quantity. At quantity and due date promising the customer specifies both the ranges of acceptable quantities and due dates. In these situations, the ATP mechanism outputs whether to accept or reject the order and in case of accepting, the quantity, due date and quantity and due date to be promised, respectively. Chen et al. (2000) develop a mixed integer programming model for quantity promising with a pre-defined due date. Chen et al. (2001) they add the due date flexibility dimension to their model and formulate a quantity and due date promising ATP model. Ervolina and Dietrich (2001) also mention that price can also be another dimension and that the order may include an offering price which further complicates the ATP systems decision responsibility.

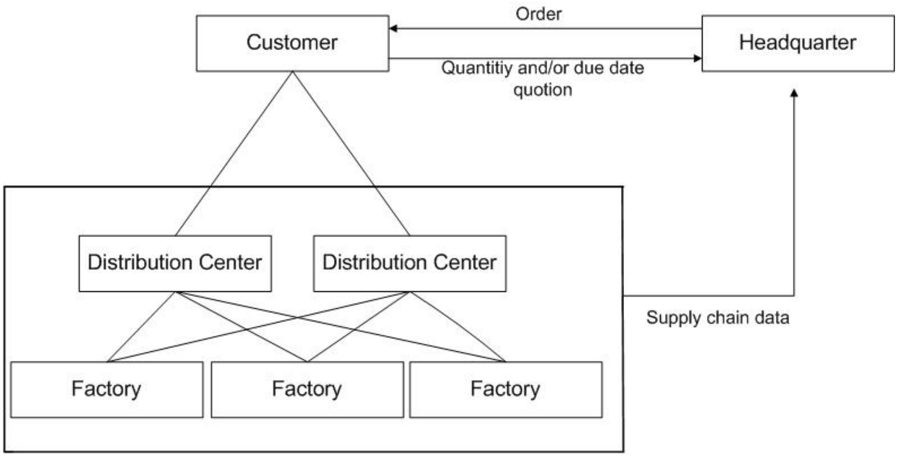


Figure 2-2: Supply chain related to ATP system (Jeong et al., 2002)

Fleischmann and Meyr (2003) define ATP by using its close relationship with advanced planning and scheduling (APS) modules of enterprise software applications. They argue that most ATP publications and research are motivated by APS. APS has the ability to match the available inventory and projected supply/production (that can be also called ATP quantities) which have been triggered by potential forecasted future sales and arriving customer orders.

Zhao et al. (2005) distinguish ATP from traditional planning, scheduling and inventory management processes by its time constraint. ATP system should operate in a short-term operational environment where resource availability is considered as fixed because of relatively longer raw material procurement lead times especially in e-business environment.

2.2. Classification of ATP Strategies

There are potentially many ways to classify and struct ATP systems. The most common classification dimensions are: availability level, operating mode, functional role in order promising systems and interaction with manufacturing resource planning (Pibernik, 2005). In this part, we investigate each of them deeply.

2.2.1. Conventional ATP and Advanced ATP

The first and fundamental classification can be defined according to product and production methodology perspective. Conventional ATP (CATP) and Advanced ATP (AATP) (Pibernik, 2005).

CATP keeps track of uncommitted portion of current and future available finished products. According to Ball et al. (2004), APICS dictionary defines CATP as “The uncommitted portion of a company’s inventory and planned production, maintained in the master schedule to support customer order promising.” A simple database lookup is enough for promising customer orders in this type of ATP mechanism.

In order to investigate AATP, we will give a short description of production strategies below.

In SCM literature, there are four main production strategies: Make-to-stock (MTS), configure-to-order (CTO), assemble-to-order (ATO) and make-to-order (MTO). In MTS environments, production decisions are made based on historical data and sales forecasts. In CTO environments, customers configure their products by selecting the type and quantity of components to include in their order. In ATO environments, the producer prepares the final product by using the pre-allocated raw materials after the arrival of the customer order. In MTO environments, the full production process is started by the arrival of the customer order.

Today, customers are demanding more customizable products. In such an environment, it is almost impossible to foresee the short term demand and setup each customizable product by a pure MTS strategy. This is the reason why, more and more firms are shifting their business strategies from mass production (MTS) to mass customization (CTO and MTO) in order to quickly respond to market dynamics (Chen et al., 2009).

CATP is associated with the traditional MTS environment that has relatively standard products and stable demand. On the other hand, Chen et al. (2001) define AATP as an execution mechanism that allocates and reallocates available resources, including raw materials, work in process, production and distribution capacities and even resource constraints across the supply chain. This is the reason why AATP is commonly associated with CTO, ATO and MTO production environments.

According to Pibernik (2005), AATP mechanisms can also be applicable to MTS environments. AATP provides a decision making mechanism for allocating available finished goods inventory to customer orders and concluding order quantities and due dates. AATP based on supply chain resources implements a systematic resource allocation process. It needs detailed information about supply chain capacity requirements for each product included in the product range. In order to get this detailed information, the bill of materials (BOM), the routing plan and information on manufacturing and distribution capacity requirements must be available to perform the resource allocation. This is the reason why, AATP based on supply chain resources is especially appropriate for MTO environments.

2.2.2. Push-Pull Framework and the Decoupling Point

ATP models can be classified according to their functional role in the order promising business process. Ball et al. (2004) classify ATP as push based ATP models and pull based ATP models in this perspective. Chen (2003) adds a third class as push-pull integrated ATP model which is more applicable and common to real life cases.

Fleischman and Meyr (2003) divide supply chain processes as push based (forecast driven) and pull passed (order driven). Pull based processes are triggered by customer orders and they are only started upon a customer order arrival. Push based processes are derived by demand forecasts that are executed before customer orders arrive. The virtual interface between forecast driven (push based) and order driven (pull based) processes is called the decoupling point, or the order penetration point or the push-pull boundary.

With a push dominated ATP strategy, the company pre-allocates lower level properties (production and supply chain resources) into higher level availabilities (usually finished products) before the arrival of actual customer orders (Chen, 2003). This pre-allocation can be made based on past sales information, customer specific, product specific and market specific properties. Later, these allocated availabilities are used to support future order promising upon actual order arrivals.

According to Ball et al. (2004), the main advantage of push based ATP models is up-to-date pre-calculated ATP quantities over multiple periods available to support order promising even in real time. Another advantage is the ability to incorporate long term profitability into short term order promising decisions by emphasizing more profitable demand categories in the pre-allocation step. On the other hand, Ball et al. (2004) also mention that, as we depend more on pre-allocation step (which means more dependence on forecasts and other pre-works) and dependence on less pull based (on-demand) processes, limitations of inaccurate forecast become significant. The resulting inefficiencies become more significant when the complexity of the supply chain grows as in the bullwhip effect.

Pull based ATP models are executed after the actual arrival of customer orders. According to Ball et al. (2004), models of this type can range from a simple database lookup to advanced optimization. The purpose of pull based ATP execution is to promise customer orders by best using the available finished goods and resources across the supply chain. The decisions that are made by this type of models include whether to accept or reject the order, the quantity to produced and the due date to produced (Chen et al., 2001)

The main advantage of pull based ATP systems is their ability to respond quickly to customer order preferences. They also minimize the inconsistency between forecast-driven resource planning and order-driven resource consumption (Ball et al., 2004). On the other hand, Pibernik (2006) and Chen et al. (2009) emphasize the myopic short-term optimization scope of pull based systems that are unable to satisfy the long term objectives of the company. According to their study, promising the current customer order in the most profitable way does not guarantee the long term profit maximization.

When we consider the advantages and disadvantages of both push based and pull based ATP models mentioned above, it is clearly seen that the power of the final model can be improved by an hybrid approach. Chen (2003) and Robinson (2007) introduce the push-pull integrated ATP mechanism that combines the strengths of both models and minimizes their weaknesses. This means that the order promising model uses push based models before the order arrives and returns to pull based models after the arrival of the order. By this strategy, the aim is not only improving customer response time by pull-based models, but also enhancing long-term profitability with a push-pull integrated planning approach. In order to clarify this approach we define the decoupling point concept in more detail.

Figure 2-3 shows a portion of the supply chain processes (procurement, production, distribution) related to ATP and corresponding push-pull models. Fleischmann and Meyr (2003) divide these processes into order-driven (pull, to order) and forecast-driven (push, to stock) ones. As mentioned before, for the forecast-driven processes, customer orders are not known, and demand forecasts are used to anticipate them. Order-driven processes are triggered by customer orders. The interface between forecast driven and order driven ATP processes is called the decoupling point (DP).

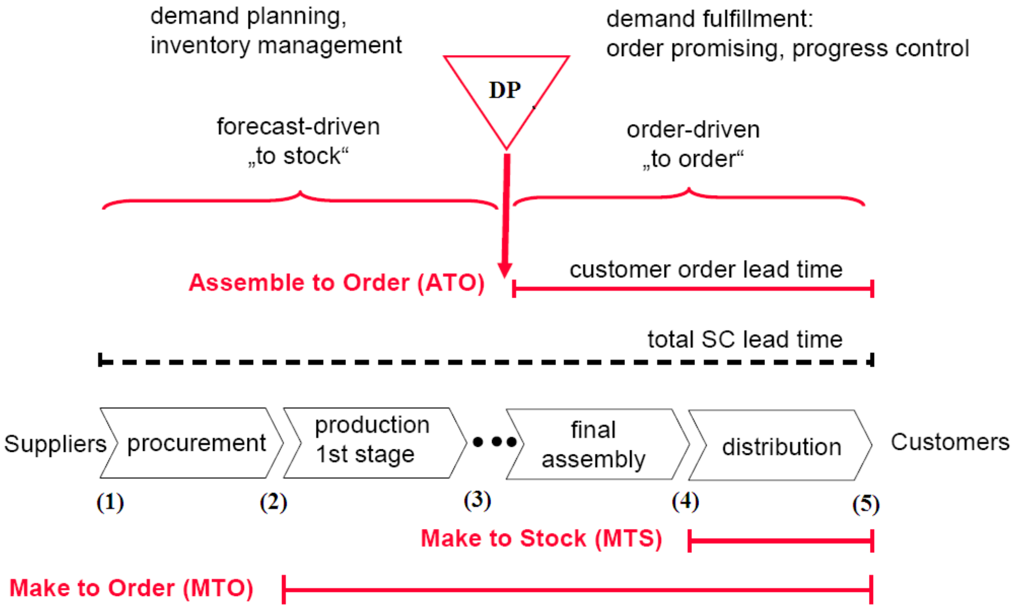


Figure 2-3: MTO, ATO and MTS decoupling points (Fleischmann and Meyr, 2003)

Fleischmann and Meyr (2003) define customer order lead time (service time) as the time between the order entry and the actual delivery of the products to the customer. Customer order lead time is an order driven concept and equals the duration of total processes downstream from the decoupling point which happens after the arrival of the customer order. According to Figure 2-3, it is clearly seen that customer order lead time mainly depends on the position of the decoupling point which depends on the type of supply chain (business strategy).

In MTO production strategy, procurement and material availability are driven by forecasts and all the production processes are executed after the arrival of the customer order. The decoupling point is located just before the production process starts (point 2 in Figure 2-3) and customer order lead time is longer with respect to other supply chains (points 2→5 in Figure 2-3). According to Kilger and Meyr (2008), besides material availability, the required capacity is also an important constraint for promising customer orders. In order to handle this constraint, most APS treated capacity as a component and ATP of this component is calculated in the same manner.

In CTO/ATO production strategy, final products are assembled and configured after the arrival of customer orders. In other words, low level resources (components) are made into finished products after the arrival of the customer orders. In these types of supply chains, the decoupling point is placed just before the final assembly (point 3 in Figure 2-3). At the decoupling point, the firm should have produced or procured its components based on demand forecasts (Fleischmann and Meyr, 2003). Actually these demand forecasts are done on the finished product level and then transformed by the master planning to the supply plan or component type level by using bill of materials (BOM) structure (Kilger and Meyr, 2008). In ATO and CTO environments, customer order lead time is the time consumed downwards the decoupling point for product assembly, configuration and distribution (points 3→5 in Figure 2-3).

In MTS supply chains, all the supply and production processes are driven by forecasts, not customer orders. Decoupling point is located after the production process (point 4 in Figure 2-3). Since all of the products are ready when the customer order comes, the only customer order lead time is due to distribution (points 4→5 in Figure 2-3). Kilger and Meyr (2008) mention an interesting point about multiple facilities at several echelons (e.g. regional warehouses). In other words, the demand close to regional warehouses can be forecasted and products can be shipped before the arrival of the customer orders. This can reduce the distribution lead time but increases the transportation costs.

As a summary for the decoupling point concept, from MTO to MTS, as the decoupling point moves downwards along the supply chain, the customer order lead time decreases. Kilger and Meyr (2008) also touch another interesting point about hybrid supply chains, i.e., at some periods of the time, the supply chain acts as MTS and at some periods as an ATO, CTO or MTO. Moreover, a contribution to this perspective can be defined as follows: the supply chain can also act different to different customer classes/types. Some critical customers can be served from stock (short lead time) and production process can take place for some others (long lead time).

2.2.3. Real Time vs. Batch ATP Execution

ATP execution systems are divided into two main groups according to their execution (operating) mode: real-time mode ATP and batch mode ATP (Ball et al., 2004). For real-time mode ATP (real-time ATP) a quantity and/or due-date are determined with the corresponding resource allocation at the time of customer order receipt. For batch mode ATP (batch ATP) customer orders are collected in discrete time periods (pre-defined batch interval) and processed together to determine ATP commitments and resource allocation for each customer order (Chen et al., 2000).

The choice of the operation mode and the length of the batching interval depend on the characteristics of the business environment, customer expectations, service level agreements and technical considerations. Today, especially at the e-business environment, customers expect ATP responses at web speed (real time). On the other hand, companies prefer to respond to customer orders in batch mode in order to gain advantage of scheduling and optimizing resources over a longer horizon and increase their profit.

According to Chen et al. (2000) and Ball et al. (2004), some companies use a hybrid approach (both real-time and batch) to respond to customer orders. They give an initial (soft) promise to customers in real time and then generate a certain (hard) promise later, after the batching time horizon is over. For example, Dell Corporation uses a two-stage order promising approach as many other e-commerce companies. When a customer places his/her order, Dell sends an e-mail that declares the approximate shipment date (about 14 days) of the customer order. Then, a few days later, the exact delivery time of the order is sent to the customer after the batch execution of all customer orders is completed.

The ability to respond to customer orders in real time or not mainly depends on the production strategy of the company and corresponding ATP strategy (conventional ATP, advanced ATP). If the company is a MTS company, order promising only needs a database lookup for finished goods inventory and planned production schedule. On the other hand, order promising in ATO, CTO and MTO environments needs complicated resource and supply chain controls in addition to the inventory lookup. Thus, real-time ATP is more applicable in MTS environments and it is difficult to implement pure real time approaches in ATO, CTO and MTO environments.

2.2.4. Other ATP Classifications

According to our literature survey, the most comprehensive ATP classification is given by Pibernik (2005). He classifies ATP systems according to different dimensions including availability level, operating mode and interaction with material requirements planning (MRP). These dimensions and corresponding ATP types are illustrated at Figure 2-4. We have already investigated availability level (conventional (finished goods) vs. advanced (supply chain resources)) and operating mode (real time vs. batch) dimensions.

According to interaction with MRP, Pibernik (2005) classifies ATP systems into two: Active ATP and Passive ATP. Passive ATP systems perform conventional order promising and do not have impact on the master schedule. Active ATP systems are integrated with the company’s MRP and while performing order promising active ATP generates and modifies the master schedule especially in MTO environments.

		Availability Level			
		Finished goods(FG)		Supply Chain Resources(SCR)	
Operating Mode	RealTime(RT)	RT/FG/A	RT/FG/P	RT/SCR/A	RT/SCR/P
	Batch (B)	B/FG/A	B/FG/P	B/SCR/A	B/SCR/P
		Active (A)	Passive (P)	Active (A)	Passive (P)
		Interaction with MRP			

Figure 2-4: ATP types (Pibernik, 2005)

Pibernik (2005) also mention additional ATP functionalities such as ATP with substitute products, multi-site ATP and ATP with partial delivery.

In some business environments, substitute products or components can be delivered instead of the products or components that the customer actually orders. This action depends on the availability of the original product, the substitutes and the acceptance of the customer. Substitute products increase the complexity of the ATP models because of the extra constraints related to the material compatibility. Ball et al. (2003) investigate the material compatibility constraints in MTO environments in more detail.

If the customer order cannot be fulfilled completely with the available resources at a certain location, the missing resources can be provided from another location. Multi location ATP systems take transportation lead times and costs into consideration. Tsai and Wang (2008) investigate multi-site ATP in detail and present models that include distribution and transportation network characteristics.

If the desired quantity is not available within the given delivery time window, the customer order can be fulfilled with more than one deliveries such that the first delivery before the due date and others are after the due date. Such a relaxation can only be possible if the customer accepts partial deliveries. These partial deliveries can be processed from different locations and with substitute products or components (Pibernik, 2005).

Pibernik (2006) categorizes allocation (order promising) mechanisms into 4 groups: First Come First Served (FCFS) allocation, rank-based allocation, optimization-based allocation and pre-allocation. In FCFS allocation, orders are promised according to their arrival sequence. In rank-based allocation orders are promised in a sequence which is determined according to the relative priority of the customer placing the order. This priority is usually calculated from the historical sales data. In comparison to rank-based allocation, optimization-based allocation allows more detailed modeling of the short and long term effects of the order promising. By allocating orders to the ATP inventory, it considers the interdependencies between order promising decisions for individual orders and can therefore minimize the internal problems of an FCFS and a rank-based approach. Pre-allocation is the allocation of ATP quantities to different customer classes before the order promising phase.

2.3. Optimization-Based ATP Models

ATP models, in the supply chain and ATP literature can be divided into two fundamental groups: Deterministic ATP models and stochastic ATP models. Detailed information about stochastic ATP models can be found in Littlewood (1972), Meixell and Chen (2004), Pan and Shi (2004), Quante et al. (2009), Talluri and Van Ryzin (2004) and Chen et al. (2009).

The mixed integer programming models, introduced by Chen et al. (2000), Chen et al. (2001), Chen (2003), Ball et al. (2004) and Zhao et al. (2005) have been respected by many researchers and constructed the basis for today's optimization based deterministic ATP research.

Chen et al. (2000) develop a mixed integer programming (MIP) ATP model that quotes quantities to customer orders in a MTO environment. Their model collects customer orders in a pre-defined batching interval B and quotes quantities to them at the end of it. They discretize the entire planning horizon into equal length time periods and run their model in a rolling horizon. At every run, the orders are promised for the forthcoming T time periods. They define two dimensions of customer flexibility: quantity and configuration. For the quantity flexibility, they let the customer define the minimum (acceptable) and maximum (desired) quantity levels while ordering. They use these variables in order to force the promised quantity to be in the desired range. For the configuration flexibility, they let the customer to select multiple preferred suppliers for each raw material and use this opportunity to replace the originally desired raw material with the customer-defined substitute according to the real time availability while promising the orders. While doing these substitutions, they consider material incompatibilities that may occur among raw materials.

Their objective function maximizes the profit in a T-period batch ATP execution which is the revenue from accepted orders minus various costs including inventory holding cost for raw materials, finished products and WIP, the lost-sales costs for the denied orders and penalties for under-utilization. They define various constraints representing inventory balance for raw materials, customer preference restrictions, BOM relationships, capacity utilization, acceptable range of order quantities, due date restrictions and material compatibility.

Chen et al. (2001) add a third customer flexibility to their batch MIP ATP model: order due date. Their new model handles customer orders with pre-specified acceptable range of due dates. Their model maximizes the profit for each batch period by defining the most appropriate quantities, configuration and due dates for the customer orders but without considering future forecasts and opportunities.

Ervolina and Dietrich (2001) develop a push-based model that allocates ATP to different demand classes by using feature sets in a CTO environment. Features are customer selectable parts and that are attached to the product families. Their implosion models use forecasted demands and ATP availabilities that come from the MPS to optimize the allocation of resources to demand classes through maximizing the potential value of satisfied customer orders.

Ball et al. (2004) define both push-based and pull-based deterministic ATP models. Their push-based model pre-allocates the available resources to customer-order classes by considering the potential values of satisfying the corresponding customer class, BOM structures, holding and production costs. They define a fill rate for each customer class that defines the minimum amount that should be allocated to the corresponding class in order to catch pre-defined minimum service levels. They also model dynamic BOMs and material substitution. In their pull-based models, they analyze material compatibility constraints deeply by using bipartite graphs.

Zhao et al. (2005) define MIP models based on previous work of Chen et al. (2000) and Chen et al. (2001). Their objective function is minimizing the due date violation cost, inventory holding cost and variation in day-to-day production smoothness measure. The model also reschedules the previously promised orders by respecting the previously determined due dates as new orders arrive. They also divide their model into master model and sub-models in order to manage several millions of variables in the experimental studies.

Fleischman and Meyr (2004) define three push based ATP allocation models for MTS, ATO and MTO production environments. Their models serve the order promising, demand-supply matching and shortage planning tasks of supply chains. They develop objective functions that penalize backlogging, earliness and tardiness in delivery of the products to the customer. They also introduce capable to promise (CTP) resources that can be added to ATP quantities if it is profitable. They also add several extensions to their models including, no reduction, no backlogging, introduction of lost sales and lost customers, order splitting, alternative locations and product substitution.

Tsai and Wang (2009) define multi-site ATP models for CTO production environments. They introduce a three-stage ATP model that is suitable for multi-national corporations. The first stage is to assign orders to the most appropriate plant, the second stage allocates ATP to the assigned orders in each plant and the third stage reallocates unassigned orders from the first two stages to plants. Their objective functions maximize the production based profits that are obtained from ATP allocation and consumption minus the cost attributes including earliness, tardiness and even opportunity costs.

Robinson and Carlson (2007) introduce a dynamic real-time ATP model that considers BOM balancing, demand, due-date and maximum availability constraints. Their model executes in a mixed MTS/MTO environment, and remote sourcing decisions are made according to in-house production and outside (remote) sourcing decision judgments.

2.4. Customer Segmentation and Revenue Management in ATP

According to Wikipedia (2009), revenue management can be defined as the process of understanding, anticipating and influencing the customer behavior in order to maximize revenue or profits from a fixed and perishable resource. Although revenue management process is more common in airline and hotel industry, it is also practiced in manufacturing and retail industries.

According to Meyr (2009), one of the biggest challenges in airline industry is to prevent that a high margin (more profitable) customer cannot get a seat, because a lower margin (less profitable) customer has booked the last one a few minutes ago. Revenue management has developed methods to overcome these challenges by defining customer classes by profit contribution and other measures, and defining booking limits to them. After this allocation of booking limits to customer classes, actual booking requests are executed by using the limits of the corresponding class by FCFS policy.

Actually, customer segmentation can be matched with the demand aggregation concept of the supply chain literature. Instead of treating each customer individually in planning phases, to decrease complexity, customers that show similar characteristics (buying behavior, location, sensitivity to price etc.) can be grouped and treated as one customer. This approach, as the product aggregation approach, has found much interest in the literature.

According to Chen et al., (2009), Littlewood (1972) suggested to continue accepting lower profit customers in a two class booking problem as long as

$$v2 > v1. Pr\{D_1 > x\} \tag{2.1}$$

where *v1* and *v2*, respectively, are mean revenues from higher profit and lower profit passengers (*v1* > *v2*), and *Pr {D_1 > x}* represents the probability that the higher profit random demand D_1 is greater than the remaining seat inventory x. This acceptance rule maximizes the expected total revenue.

As we discussed previously, in push-based ATP models, product and supply chain resources are pre-allocated into higher level availabilities (finished products). On the other hand in pull-based ATP models, all the planning and execution processes are triggered by the arrival of customer orders.

In ATP literature, reserving certain amount to customer/order classes at the planning stage is called allocated ATP (aATP). In aATP, certain quotas are allocated to customer/order classes according to profit and other measures during the push based planning phase. Thus, during the order promising phase, actual customer orders are promised from these allocations by a pull based approach. Most of this promising action is executed online by a FCFS policy, but it can also be executed offline (batch).

Chen et al. (2009) discuss that, because of the nature of order promising and fulfillment problems, many existing optimization based ATP models are pull based. Moreover, because of the myopic short-term optimization scopes of these pull-based problems, they may not be able to satisfy the long-term goals of the company. Thus, they argue that it is possible to improve the long-term performance of pull-based ATP models by taking potential future customer orders into consideration. To do this, they segment actual demands into four groups by using two dimensions: profit contribution (less-profitable customers, more-profitable customers) and demand stage (current, future) (Figure 2-5) and apply a stochastic programming model. By doing this segmentation, they try to reserve a certain amount of critical resources for future more profitable customer orders. This mechanism prevents current less profitable customer orders from over consuming the scarce resource.

Demand stage	Profit Contribution	
	More Profitable	Less Profitable
Current	Class1	Class3
Future	Class2	Class4

Figure 2-5: A classification of customer (order) classes (Chen et al., 2009)

Chen et al. (2009) assume that a critical non-perishable (durable) component is required to assemble a line of certain products for fulfilling different groups of customer orders in short supply. If all components are committed to the current customer orders, the manufacturer can no longer promise more customer orders even the orders with higher profit margins. Thus, they defined a variable R, which represents the amount that should be passed to stage 2 (future) even though there may exist more customer orders belonging to low profit customers in stage 1. They also define variables denoting the demand, marginal profit, lost sales of each customer group in each demand stage. They also generate different scenarios for their stochastic model that implements different situations that the system may face when the actual customer order comes. Their two-stage stochastic optimization model calculates the optimum reservation level (R) that should be passed to future demand stage. They also prove that their reservation level is optimum by means of different mathematical approaches and simulations.

Chen et al. (2009) also mention the differences between their approach and classical revenue management (RM) approaches which are also mostly applicable to other allocated ATP (aATP) approaches. Firstly, aATP problems specifically consider non-perishable resources which can be stored as inventory over a significant period of time. On the other hand, RM problems deal with perishable resources (plane seats, hotel rooms) that cannot be stored as inventory. Secondly, in aATP problems, every demand segment exists at every demand stage. Due to current RM approaches, only specific (mostly one) demand group responds to the current price offer at the corresponding demand stage. However, in aATP models, there is no such a one-to-one matching between demand stages and customer segments. Thirdly, in RM problems the profit (price) is monotonically increasing or decreasing over time. For example, the willingness to pay generally increases for airline seats towards departure time, but decreases for fashion goods towards the end of fashion season. On the other hand, in aATP models both more profitable and less profitable transactions exist at every stage. Lastly, special to their model, they apply a batch execution approach that is executed at the end of every demand stage while in RM models the customer requests are fulfilled by using online execution approaches.

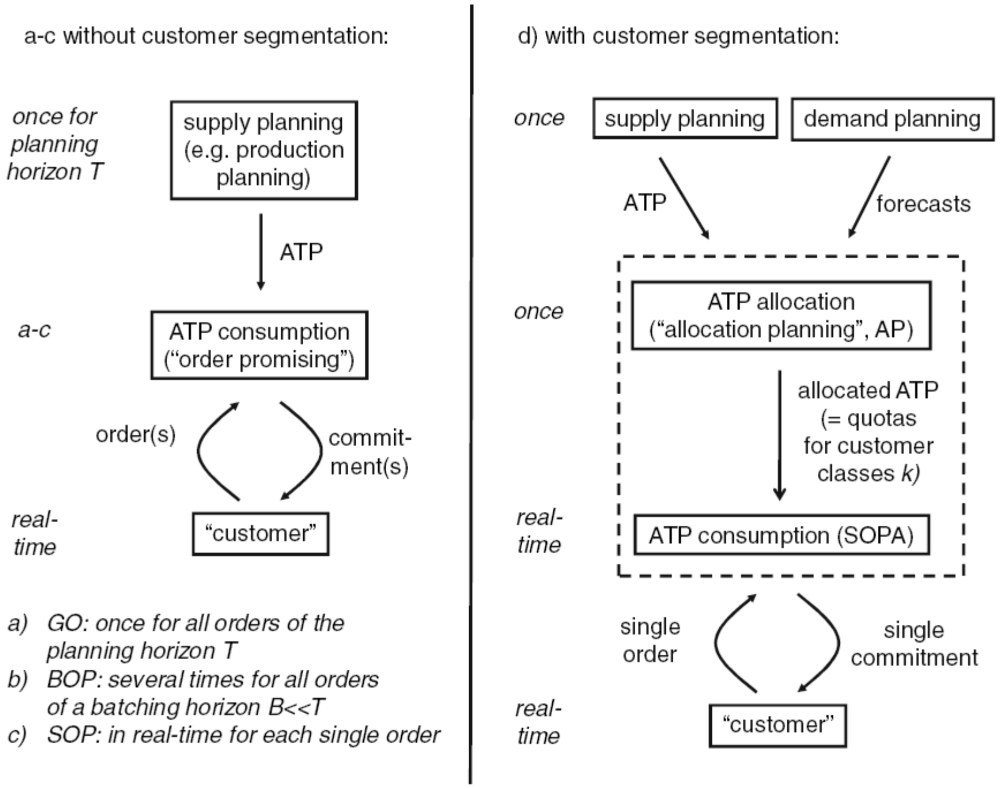
Meyr (2009) define the process of customer segmentation, ATP allocation and ATP consumption as “allocation planning and ATP consumption AP&C” in his comprehensive research paper. He defines the shortage situation as the stage where demand is greater than the capacity. He develops a model to avoid shortage situations by using aATP. The main idea of his research is to improve demand fulfillment in MTS supply chains by making use of heterogeneity of different customers through allocation planning and ATP consumption (AP&C) order promising. His model has three fundamental steps:

- to segment customers with respect to their importance and profitability into several priority classes,
- to allocate ATP to these classes on the basis of a deterministic profit maximization process taking advantage of short–term demand information,
- to promise customer orders, i.e. to consume ATP, in real time with respect to these customer hierarchies.

The main purpose of the AP&C model is to prove whether the same or even better profits as in the batch ATP promising can be achieved, even though a customer gets his answer immediately when he enters his order.

Meyr (2009) defines four order promising models seen in Figure 2-6. At global optimization (GO) all of the orders are collected and promised at the end of the period T. At batch order processing (BOP), as we mentioned previously as batch promising, the orders are collected for batching interval B<<T and promised at the end of each batching period. At single order processing (SOP), as we mentioned as real time promising previously, customer orders are promised in real time. Lastly, at SOP after allocation planning (SOPA), customer orders are promised in real time according to the allocation of the corresponding customer class.

The model a-c does not use customer segmentation. Model d uses customer segmentation and puts revenue management idea into practice by differentiating *k* customer classes, allocating ATP to these classes and satisfying the demand only if enough allocated ATP of the customer's corresponding class is available (Meyr, 2009).



a-c without customer segmentation:
once for planning horizon T
supply planning (e.g. production planning)
ATP
a-c
ATP consumption ("order promising")
order(s)
commit-ment(s)
real-time
"customer"
a) GO: once for all orders of the planning horizon T
b) BOP: several times for all orders of a batching horizon B<<T
c) SOP: in real-time for each single order
d) with customer segmentation:
once
supply planning
demand planning
ATP
forecasts
once
ATP allocation ("allocation planning", AP)
allocated ATP (= quotas for customer classes k)
real-time
ATP consumption (SOPA)
single order
single commitment
real-time
"customer"

Figure 2-6: Order promising models (Meyr, 2009)

Meyr (2009) also compares the models that do not use customer segmentation according to objective function values:

$$GO^* > \sum_{s=1}^{T/B} BOP_s^* \geq \sum_{s=1}^{T} SOP_s^* \qquad (2.2)$$

Naturally, the profit of GO^* will be more than BOP^* and BOP^* will be more than SOP*. Thus, he defines the profit of GO^* as the benchmark, that is possible if the full demand information is known before promising the orders, for evaluating other models' performance.

Meyr (2009) and Meyr (2008) suggest clustering methods to segment customers into different *k* demand groups. He suggests the distance metrics and some clustering algorithms to segment customers. He also mentions the importance of the number of segments and its effect on the success and complexity of the models. Moreover, Han and Kamber (2001) extensively analyze general data mining approaches including clustering and classification. They also clarify different processes such as data processing, data warehousing and online analytic processing (OLAP) that are indispensable for data mining.

Kilger and Meyr (2008) analyze the concept of customer hierarchies in ATP allocation. They argue that customer structures can form a hierarchy similar to the geographic dimensions in demand planning. In this structure the forecasts are aggregated to the root element and then passed to master planning for production.

In Figure 2-7 Kilger and Meyr (2008) demonstrate the ATP allocation in customer hierarchies. The numbers in parenthesis show the forecasted amounts and other numbers show the production quantities approved by master planning. The aggregated production quantities are passed downwards from the root according to different partitioning rules including rank based, per committed and fixed split.

Kilger and Meyr (2008) also demonstrate a simulation study that implements ATP consumption rules between hierarchies. For example, if the requested quantity cannot be promised from East Germany’s ATP quota then it can be promised from the next higher node Germany or even from the higher nodes (Figure 2-7). Also they consider the time dimension of ATP allocation, i.e. if the ATP quota of the current time bucket is not enough, past and future time buckets for the current node and higher nodes can also be searched for ATP promising.

Quante et al., (2007) perform a comprehensive research on revenue management and demand fulfillment. They investigate the relations between traditional revenue management research and demand fulfillment and investigate industries and software applications.

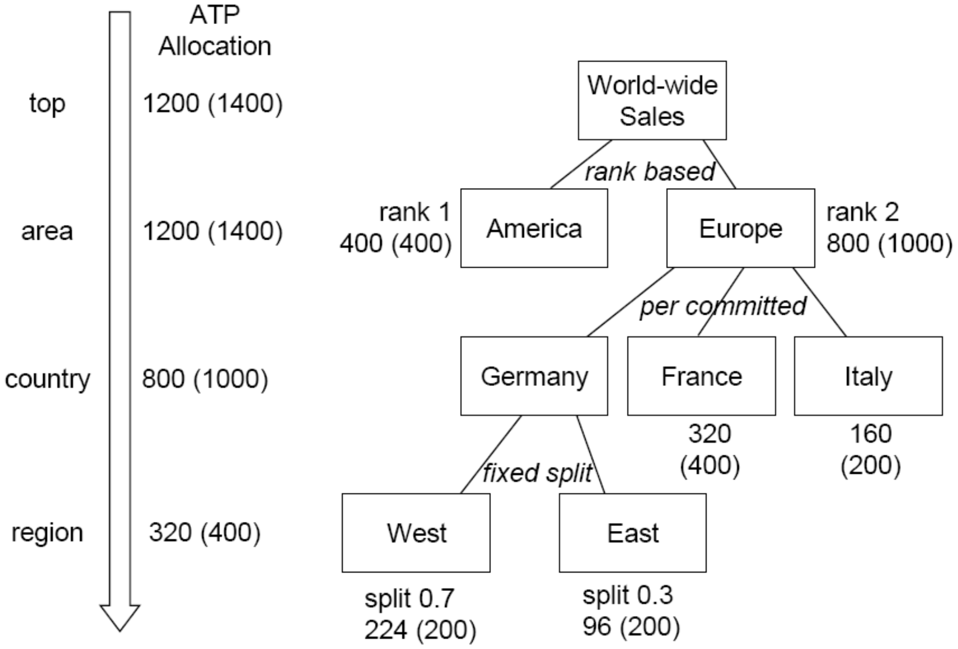
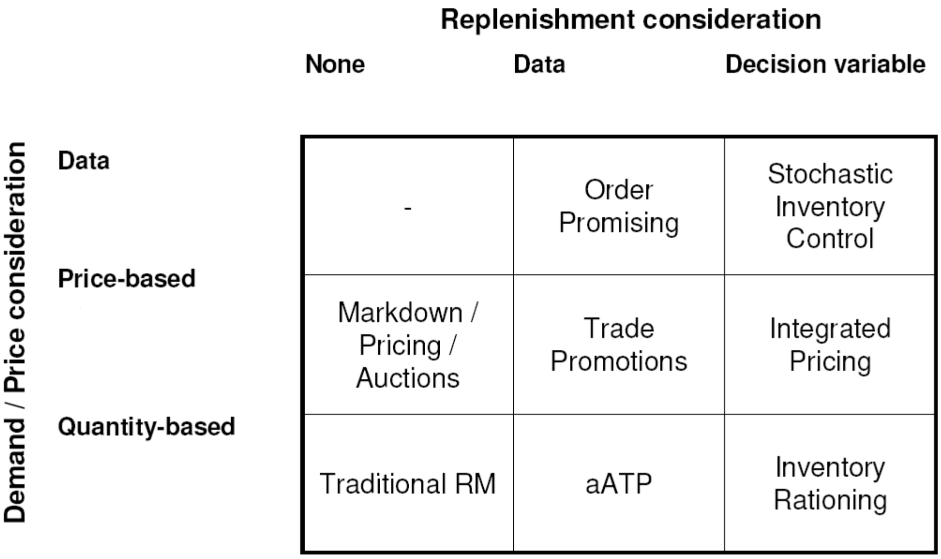


Figure 2-7: Allocation of ATP in the customer hierarchy (Kilger and Meyr, 2008)

Quante et al. (2007), as Talluri and van Ryzin (2004), divide revenue management (RM) into two fundamental dimensions: Quantity based approaches and price based approaches. Quantity based RM approaches uses customer heterogeneity and prioritizes customer classes while allocating scarce resources to them. Price based approaches use pricing decisions as a lever for demand management. This includes adjusting prices dynamically over time, in response to non-stationary demand or a finite selling season and actions as a price decision mechanism.

As seen in Figure 2-8 price based models can be divided into 3 groups according to their replenishment consideration. Pure pricing models aim to determine the optimum selling price that maximizes the total revenues. Markdown models determine the right price path for the inventory that cannot be replenished during the planning horizon. Auctions provide a price determination process within the supply chain that is alternative to fixed prices. Trade promotion models consider replenishment as an exogenous input. In integrated pricing models, the price decisions are integrated into the quantity and due date models that we mentioned previously.



Demand / Price consideration	Replenishment consideration		
	None	Data	Decision variable
Data	-	Order Promising	Stochastic Inventory Control
Price-based	Markdown / Pricing / Auctions	Trade Promotions	Integrated Pricing
Quantity-based	Traditional RM	aATP	Inventory Rationing

Figure 2-8: Revenue management and demand fulfillment model types (Quante et al., 2007)

Quantity based models, are also seen in Figure 2-8 can be divided into three groups according to their replenishment consideration. We have investigated traditional RM and aATP before. The main difference between aATP and inventory rationing (IR) is in terms of exogenous and endogenous replenishment. IR models consider replenishment decisions with stationary deterministic or stochastic lead times. In contrast aATP models consider capacitated, dynamic and deterministic arrivals of replenishment (push based production) quantities. Namely, aATP usually assumes deterministic and dynamic demand forecasts while IR assumes stochastic demand.

All in all, it is clear that pure real-time or batch order promising models without revenue management (RM) approaches are not capable of satisfying the dynamic needs of today’s supply chains and markets. Although RM approaches increase the complexity of the ATO models, their contribution cannot be ignored. This contribution has been perceived by the software vendors and put into practice. We investigate some of them in the next section.

2.5. IT Challenges

Today, IT (information technology) systems become one of the most important strategic and operational differentiators for the companies. Incredible developments in data processing, network speed and internet technology enhance the capabilities of enterprise resource planning (ERP), supply chain management (SCM) and advanced planning and scheduling systems (APS). Companies are investing billions of dollars on IT systems in order to increase their efficiency and minimize operational costs.

ATP systems are generally short-term order promising systems that are executed too often, even for every customer order. Moreover, they require huge amount of data from different resources in a company including sales, purchases, production, CRM, etc. in order to generate the availability outlook database (Lee, 2006). On the reverse side, updating the availability outlook after the promised orders is also an IT challenge. The availability level (conventional-advanced) and operating mode (online-batch) are the two most important properties that determine the complexity of the availability outlook generation. The simplest case is a conventional ATP system that is executed in batch mode. Here only finished goods inventory outlook is generated and used rarely on batch order promising process. On the other hand, the most difficult case is the advanced ATP system operating on real-time. Here, besides the finished goods, the availability outlook for production and supply chain resources should be generated and be attainable in real time for front end customer interacted systems. Thus, on the IT perspective, there is a tradeoff in selecting the most appropriate ATP system for the company. Moreover, the capabilities of the ATP system, i.e., quantity promising, quantity and due-date promising, partial delivery, integrated pricing etc. determine the complexity and the required investment of ATP systems.

Since real time execution is too expensive, even today, most of the ATP systems are fully or partially executed in batch time horizons. Lee (2006) defines the availability at IT systems as system availability and the actual availability as physical availability. It is common that there becomes a slight difference between system availability and physical availability because of batch execution and replenishment.

Moreover, over-confirmation issues can occur when more than one promising engine works for the same resource for different customers. These inconsistencies can cause planning problems and customer dissatisfaction.

When service level agreements, market dynamics and customer expectations force companies to real time order promising and ATP execution instead of pure real time systems, companies can use some hybrid solutions. One of them is to use two-stage order promising in which the customer is first given a soft promise in real time and a hard (certain) promise later after the batch ATP execution. The second approach is using some special data processing structures such as data marts, data warehouses, and materialized views that enable almost real time promising. Figure 2-9 shows sample data warehouse architecture.

Data warehouses integrate the company's electronically stored data that are available at different locations in different formats. Data marts are smaller data warehouse entities that are constructed for business needs and enable fast access to the computed results. For example, in Figure 2-9, four data marts are constructed (ERP, sales, finance, and customer) in order to serve business needs and feed the data warehouse. The data marts are constructed from the online transaction processing (OLTP) data and generally replenished in batch time horizons. By shortening this replenishment time period, they can be used by ATP systems to promise customer orders almost in real time by using summarized and pure data that are constructed specially for ATP execution. They are widely used for forecasting and push based ATP planning.

Materialized views are dynamic data structures that are used in relational databases. They are very similar to data marts, but they are not multi-dimensional. They can be replenished in certain periods and be used for ATP outlook generation. For example a materialized view can be constructed by using tens of tables including sales transactions, customer information, financing transactions, etc. and can be replenished in minutes only to provide almost real time summarized pure ATP outlook.

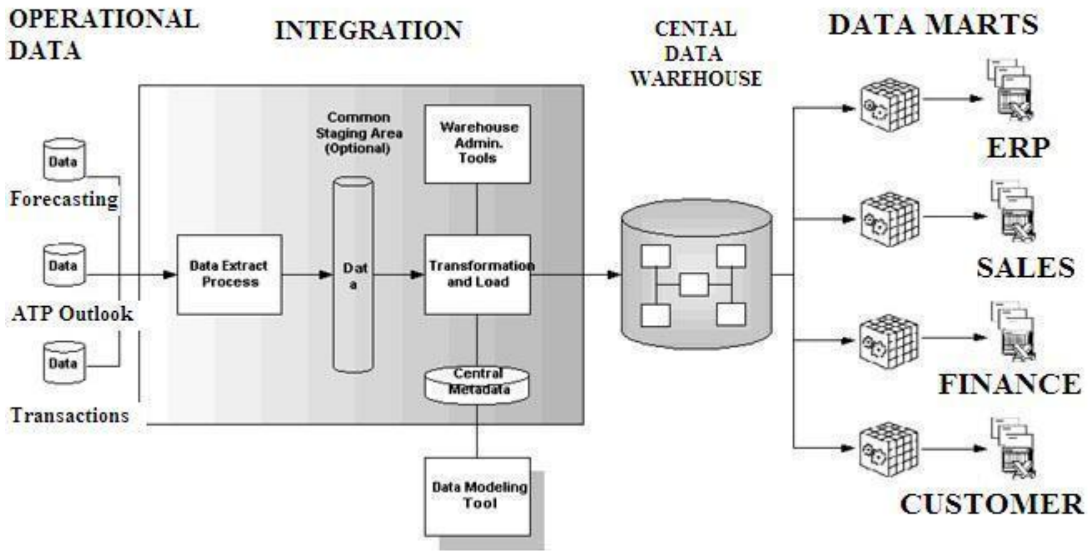


Figure 2-9: Sample data warehouse architecture

Ervolina and Dietrich (2001) mention another important feature of IT Systems -high availability and clustering- and their relation with ATP. In an e-business environment, the ATP system must be up and running 24 hours a day and be able to respond to customer requests at web-speed. Because of these business requirements, generally there are more than one IT systems that work in a synchronized manner and respond to ATP requests. In these environments, the availability information is refreshed in one system, while the orders are still responded to by the other systems (Ervolina and Dietrich, 2001). This may give rise to some inconsistency issues that can be solved by various IT approaches. Another important consideration is the high availability issue, which means keeping the systems 24 hours alive. If there arises a problem with a system that is executing an important job, the job could be passed to other alive systems without loss of information and state. That is why clustering and backup systems are very important for mission critical ATP systems especially implemented in e-business environments.

According to Meyr et al. (2005), memory resident databases (also known as live Cache) can also be used for real time order promising. They allow fast access to data, because they are stored in the server memory, and accessing to them is much faster than the actual physical database at disks.

As a summary, IT systems are one of the fundamental parts of today's ATP and SCM systems. They are defining not only the company's way of doing business but also the market conditions, service level agreements and even customer expectations. In order to win this challenge, companies are investing more in IT and keep IT in the middle of enterprise architecture.

2.6. ATP in the Commercial Software

ATP systems have been implemented at almost all of the commercial APS providers. Especially advanced ATP has become an important component of SCM and APS systems and a tool for competitive advantage.

Meyr et al. (2005) introduce the most important tasks of SCM and classify them based on the two dimensions: planning horizon and supply chain processes. They also form a supply chain planning (SCP) matrix to demonstrate the relations between tasks and dimensions (Figure 2-10). They argue that the name of the modules can change from one APS provider to another, but the planning tasks that are supported are basically the same. They also mention that the third dimension can be added to the matrix in order to demonstrate industry-specific modules and differentiation.

Since demand fulfillment and ATP module is a short term planning task and sales oriented, it is placed at the bottom right of the SCP matrix. Meyr et al. (2005) analyze the general structure and modules of the three most common commercial APS providers: I2 Six.One of I2 Technologies, EnterpriseOne of PeopleSoft and APO of SAP. Despite their different names (Demand Fulfillment in Six.One, Order Promising in EnterpriseOne and Global ATP in APO), the capabilities and logic underlying them are very similar. Moreover, they commonly use ILOG CPLEX for the optimization models of linear and mixed integer programming (Meyr et al., 2005).

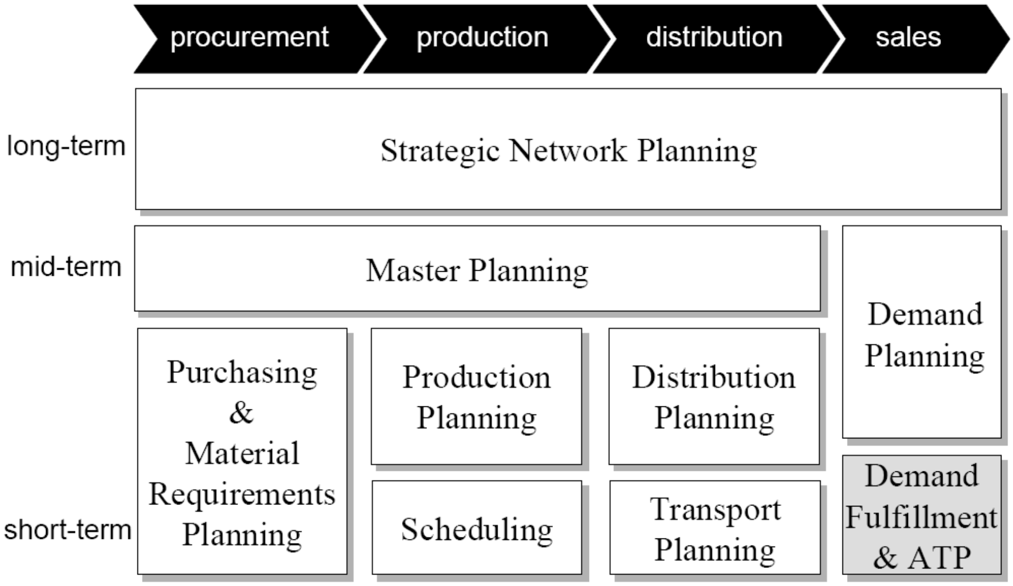


Figure 2-10: General software modules of APSs (Meyr et al., 2005)

According to Ball et al. (2004), APS providers enhance their ATP functionalities by techniques such as heuristic search, rule-based search or optimization based models. Rule-based search mechanisms are commonly used, because it is easy to implement and use them. Moreover, some APS providers provide allocation mechanisms that enable implementing allocated ATP models.

The software modules of APS are dedicated to deterministic planning. On the other hand, there are uncertainties on both inbound (unreliable suppliers, machine and labor problems, etc) and outbound (stochastic customer demand) side. To hedge against these uncertainties, buffers have to be installed either in the form of safety stock or safety times (Meyr et al., 2005). Thus, most of the APS solutions offer buffering mechanisms in order to hedge forecast errors in push based allocations. However, this approach can vary among different industries and mostly needs attention from more than one module, even from the entire supply chain planning.

Quante et al. (2009) classify demand fulfillment and ATP software in three groups: traditional order promising, price-based solutions and quantity based solutions. Traditional order promising contains software modules for short term order promising under known inventory availability. Price-based solutions are relatively newer due to the vast requirements of computing power and availability of past sales data. The rise of data warehouses and computing power has recently made the use of automated pricing systems that also uses vast amount of past sales data possible. Auction based systems at e-commerce, promotion optimization at retail environments and enterprise profit optimization are the new challenging areas in price-based systems. Quantity-based solutions are integrated with master planning and prepare deterministic demand forecasts and prices to it. Then, master planning systems determine the best combinations of sales, production and replenishment quantities and the corresponding inventories under given capacity constraints. These quantities can be allocated to different demand classes and revenue management approaches can be applied.

Lee (2006) analyzes the availability management system (AMS) of IBM. The main element of IBM’s AMS is the availability outlook. He defines four types of events that change the availability outlook: demand event, supply event, roll-forward event and data refresh event. Moreover, he has given an example that simulates all of these four events and their effect to order promising.

The current availability management approaches from commercial APS vendors are mostly applying rule-based solutions and they are mostly focusing on one side of the availability management. There is still a need for more comprehensive solutions; and in order to increase capacity utilizations and customer service levels, more optimization based approaches should be implemented.

CHAPTER 3

PROBLEM DEFINITION

3.1. General Definitions

Today, customers put three main pressures on companies that reshape today's market environment. Firstly, many customers are not satisfied with standard products and they want products that satisfy their specific needs. Secondly, many customers want their orders to be satisfied almost at web-speed despite the products' complex customized configurations. Thirdly, the customers do not accept to pay extra amounts for their customized products compared to the standard products. In order to satisfy these requirements, more and more firms are examining and improving their supply chain processes for providing almost completely customizable products and services to their customers at low costs. It is almost impossible to foresee the short-term demand and get prepared for the high-mix, low-volume products before the actual customer orders and then to serve the customers immediately from stock by MTS strategies. That is why companies are refactoring their production processes, even the structure of their full supply chain to support ATO, CTO and MTO approaches.

In MTS production environments, finished products are prepared according to forecasts and customer orders are met from stock. On the other hand, in MTO production environments, all of the production processes are initiated after the arrival of customer orders, considering the order-specific properties and configurations. In CTO production environments, raw materials are produced or procured according to forecasts, but final assembly and configuration of the finished products are made after the arrival of the customer order.

Order lead time is one of the most important and strategic properties that reflects customer satisfaction. In MTO environments, companies are able to satisfy almost every customer-order specific configuration because of the relatively higher marginal revenue and the relatively longer time that they have after the arrival of the customer order. However, many customers in MTO environments expect their orders to be satisfied almost with the same cost and lead time as in other strategies and also they expect the companies to be ready for their orders in order to respond them rapidly.

These expectations of the customers can be satisfied by CTO approaches where the only lead time is due to the assembly and final configuration which is generally acceptable by the customers. Since the main time and resource consuming process is the production, in CTO supply chains, the company has all the components that the customer configuration requires when the customer order arrives. Only the assembly operation is delayed to the customer order arrival, which starts immediately upon the customer order arrival.

Moreover, in CTO production environments, the companies have the advantage to replace some customer configurations with the proper substitutes according to the real time inventory and resource position if they are allowed to. By the help of this flexibility, order lead times can be further decreased.

When we consider the order promising perspective, conventional available to promise (CATP) systems are more appropriate for MTS supply chains. On the other hand; CTO and MTO supply chains require much more advanced ATP mechanisms that consider resource and raw material availability which we address as advanced ATP (AATP). In CATP, simple database lookup can be enough to calculate the availability outlook, while in AATP, all of the critical resources, materials and their relationships should be considered.

ATP resources are determined by the MPS and send as input to the allocation planning engines. Moreover, companies generally reserve some of their production capacity and resources as CTP (capable-to-promise). CTP quantities can be used when ATP resources become scarce.

3.2. Motivation of the Problem

3.2.1. Introduction

The motivation of this study originates from the real-life problems and research studies of IBM's Enterprise Server Hardware Division. IBM is one of the biggest hardware and software suppliers in the world. IBM offers a wide range of enterprise hardware solutions including Blade Servers, Blue Gene, Cluster Servers, Power Systems (System i, System p), Storage Servers, System x (x86) and System z (mainframe) and more.

In our study, we focus on System x servers; one of the most well-known product families of IBM. Like most of the other server families, System x servers are produced in a CTO environment. The servers can have complex product structures in terms of complex BOMs and configuration rules. Before continuing to our problem definition, there is a need to clarify some terminology about System x production environment.

System x servers are sold in two main ways. A *Fixed Product Model* is a predefined, customer-ready configuration of the product that can be ordered as a single part number. IBM creates these fixed products in its CTO production environment in order to satisfy frequently requested and most common customer configurations. These fixed models are called *Single Entity Offerings (SEOs)* and can be ordered by a single part number. However, as in all of the CTO environments, it is impossible for IBM to predefine and forecast all of the product configurations that the customers may desire. That's why IBM lets customers to configure their products at their respective orders as in classical CTO cases. IBM calls this environment as CTO. All of the possible configurations based on the same main part are called a product family.

There are two main components that form a server: Machine Type Model (MTM) and Features (Figure 3-1). MTM is the main component (part) of a product family that cannot be ordered alone. A Feature is a customer selectable item (part) on the configuration menu of a product family that is added to MTM. The combination of a MTM (main part) and a set of Features (parts) form a customer ready configuration in both fixed product and CTO case.

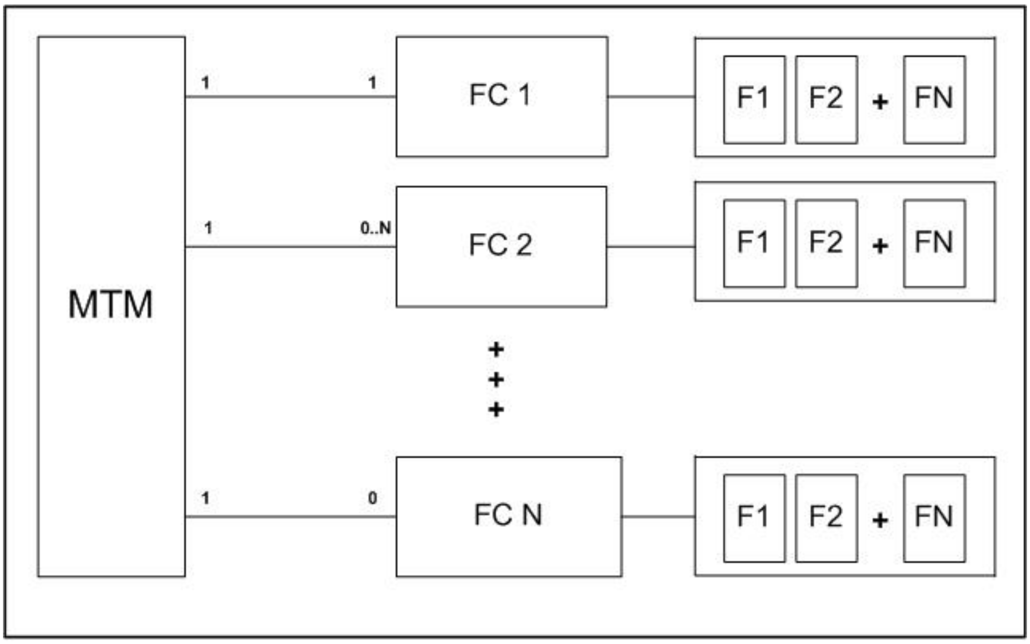


Figure 3-1: IBM enterprise server CTO architecture

Features can also be grouped by feature categories representing logical classification of them (Figure 3-1). For instance all of the hard disks that can be used in a server can be grouped into a feature category (FC) called "hard disks". The mapping relations between a CTO configuration and feature categories can be in many ways and can be structured according to business needs. The relation can be one-to-one that means every configuration should have exactly one of the features in that feature category or it can be one-to-N and one-to-0 relationships, representing that every configuration may have one or more of the features in that feature category.

In CTO and MTO environments, as in IBM case, we can call the parts (features) that the customer selects as a "Sales Configuration" and its representation in MRP as "BOM Configuration". Generally, the complexities in the BOM structure are simplified while presenting them to the customer. For example, there can be incompatibilities between parts or some parts need other parts to work, or there can be some hidden parts that the customer does not need to know. In such cases, there is a need for a translation from sales configuration to BOM configuration. IBM handles this need with a specific solution called “BOM Configurator”. Sometimes BOM Configurator makes easy the one-to-one transformations from features to MRP components. Sometimes, it makes complex transformations that are not one-to-one as exampled above (Ervolina and Dietrich, 2001)

One of the main difficulties that arise in modeling CTO environments is the dynamic BOM structures that can be changed on every customer order. Especially it is difficult to associate features with MTMs; to calculate feature forecasts and define compatibilities between features.

There are two main ways for calculating demand forecasts of features. Firstly, the demands for the features can be directly forecasted by examining the past orders and considering current market dynamics without considering their relations to MTMs and product families. The result of such a forecast is $d_{f,t}$ representing the forecasted amount of feature f in period t. The second approach is using forecasted attach rates in order to calculate the feature demands. Here, the forecast $d_{f,t}$ is calculated by

$$d_{f,t} = d_{p,t} \quad r_{f,p} \tag{3.1}$$

where $d_{p,t}$ represents the forecasted demand of product family p in period t and $r_{f,p}$ represents the forecasted attach ratio of feature f to finished product p. The second approach is widely used in CTO environments, since it is very difficult to forecast individual feature demands.

In System x servers CTO production environment, a server is made up of several components and a component can be used for several servers. This multi-product and multi-component production structure also increases the complexity of the planning models and introduces the need for modeling approaches at the feature level.

3.2.2. Allocated ATP and Order Segmentation

ATP models can be classified into two main groups as push-based ATP models and pull-based ATP models. Push based ATP models are executed according to forecasts and other planning data before the arrival of customer orders, while pull-based ATP models are executed after the arrival of the customer orders. When we consider the dynamic needs of current business environments, hybrid approaches that involve both push and pull based ATP models are needed. This need can be realized by using push-based ATP approaches by using demand forecasts and allocating the raw materials and resources to customer and product classes before the decoupling point and performing pull-based ATP approaches after the arrival of customer orders.

The multi-product and multi-component structure that we introduce in the previous section requires ATP modeling at the feature and MTM level. For each product family there can be some bottleneck resources or scarce components that can be used by more than one server types. Then ATP planning should be done for these bottleneck resources or components in order to increase efficiency.

When we consider the ATP of a feature, we see that there are two main groups that consume this ATP: the product and the customer. In other words, the feature is used by a finished product (server) and it will be sold to a customer. When considering different finished products using the same scarce resources or materials, it is clear that the priority of the products may differ with respect to profit contribution or some other factors. On the other hand when considering the customers that are requesting the same scarce components in their configuration may have different priorities again with respect to profit contribution or some other strategic non-monetary factors. Then it can be worth to search for the contribution of these priority differences and their effect on the overall model performance from the revenue management perspective. Push based ATP allocation models can be applied in order to allocate scarce resources to important customer or demand classes.

Customer segmentation approaches have been applied to manufacturing environments that are similar to the practices in airline sector in the literature (Ball et al., 2004; Meyr, 2008; Chen et al., 2009). Customers are segmented into a pre-defined number of classes according to their priorities for the company by using several clustering and classification approaches. These priorities are mostly calculated according to profit contributions and contractual relationships. Then ATP of scarce resources is allocated to high priority (more profitable) customer classes and they are protected from the consumption of low priority (less profitable) customer classes.

According to the current business practices, it is clear that non-monetary factors also play an important role while classifying customers and assigning priorities to them. Moreover, pre-defining the number of customer classes is also a problematic decision that sometimes can only be determined by simulations. In addition to this, the specific order and product characteristics can also play an important role while determining the priority when the order arrives. In other words, the same customer may contribute different potential profits (values) with different products and other order characteristics.

ATP allocation at the feature level also requires the consideration of feature’s relation with the finished product and other features in terms of BOM structure and material compatibilities. Moreover, determining the priorities of the customer order classes and their effects on the ATP allocation is important. In other words, the method of calculating the value of satisfying one unit of product in the respective order class is an important decision.

Order segmentation, allocated ATP and revenue management approaches in the manufacturing environments have a great potential to add values to supply chains that should not be overlooked. We see a potential value for applying them to IBM’s Enterprise Server production environment.

3.2.3. ATP Execution Mode

There are two types of ATP executions: Real-time mode ATP execution and Batch mode ATP execution. In real-time ATP mode, both the quantity and due date are quoted at the time of customer order arrival. In batch mode ATP, customer orders are collected for a pre-defined batching interval and processed together at the end of the interval.

The choice of the execution strategy depends on the characteristics of the business environment, customer expectations, service level agreements and technical considerations. Today, especially at the e-business environment, customers desire ATP responses at web speed (real time). On the other hand, companies prefer to respond to customer orders in batch mode in order to gain advantage of scheduling and optimizing resource usages for a longer horizon and hence maximizing their profit. That is why the decision makers should consider both parties and look for the tradeoff for the selection of the batching time window.

A hybrid two-stage approach that includes the advantages of both real-time and batch ATP executions can contribute to the success of order promising. A slightly easy model with some of the constraints relaxed and/or shorter planning horizons can be executed online or with a short batching interval, while advanced optimization or rule-based ATP models with harder constraints and longer planning horizons can be executed in relatively longer batching horizons. This two-stage order promising approach can improve the allocation of resources, if not optimal; while delivering higher customer satisfaction in shorter response times. Moreover, this approach contributes to the supply chain transparency in that the customer gets the position of the company at the time of her order; whether the company is able to fulfill her order immediately or not, and if not, the exact due date of her order after the first batch execution.

3.3. General Structure and Assumptions of the Solution Approach

After considering the above problems that companies face in today’s competitive high-mix and low-volume environment, we develop a four-stage approach that includes order segmentation, allocation planning and two-stage order promising.

We investigate the applicability of the proposed framework to the IBM'S Enterprise Server Production processes. Our framework is designed for CTO environments, but it is also applicable to MTS, ATO and MTO environments with only slight changes.

In our solution framework, we make some assumptions about the CTO production environment, orders structure and historical data availability. We assume that historical sales data have the cost and revenue attributes so that we can calculate the profit. Moreover, the business owners are able to segment customers as low, medium and high priority customers. We also assume that there is no substitution among components, i.e., if the exact customer configuration can be satisfied, then the order is accepted, otherwise it is rejected. In addition to this, there is no partial delivery; in other words, the order has to be promised fully, otherwise it is rejected. Moreover, we assume that there is only a single production facility and multi-site ATP is not applicable, hence there is no transportation cost. We only consider ATPs and CTPs of the parts as the only bottleneck resources in our models.

3.4. Contributions of the Study

The main aim of our availability management approach is to investigate the contribution of order segmentation, allocated ATP and two-stage order promising to the CTO production environments. We analyze the results of our approach on IBM Enterprise Server data and compare our results against several approaches' results, including the ones without order segmentation. Moreover, various sensitivity analyses are carried out based on the number of order classes, demand and supply changes and batching intervals.

Another important characteristic of the study is its ability to handle CTO environments where there is no fixed BOM and it is impossible to forecast all of the possible customer configurations. Moreover, the fixed model (SEO) expansion of the model addresses hybrid production environments having MTS, CTO, ATO and/or MTO processes at the same time.

Despite the fact that all four phases of the proposed approach are loosely coupled, Order Promising System (OPS) that is developed for our computational studies is an integrated system that satisfies all of the order promising needs with minimal manual adjustments. Also it is easy to understand and use it. Since a relational database is used for outputs of the system, all results can be pushed to other enterprise software systems easily or other systems can access our results easily.

Since our availability management approach includes calculations and decisions with some non-monetary factors such as; customer priority, effect of product complexity on company strategy and effect of direct responding to customers via the two-stage order promising, some of the results cannot be measured directly. More interestingly, these non-monetary decisions may decrease profit or increase operational costs in order to increase service levels for all or some customers.

CHAPTER 4

THE SOLUTION APPROACH

4.1. Methodology

We develop a four-stage availability management approach that addresses the issues of order segmentation, allocation planning and ATP consumption. All of the four stages are so designed that they are loosely coupled and they can be considered independently. The inputs and outputs from one stage to another are clearly defined in Figure 4-1.

At Stage 0, historical sales data and customer information are evaluated in order to define potential order segments. There are three attributes that determine the order segments and their corresponding values: profit of the order, importance of the customer placing the order, and complexity of the product ordered. The individual values coming from these three attributes are normalized and weighted in order to calculate a single value for the order. Then the orders are segmented according to their values.

At Stage 1, a push-based ATP allocation is executed by using the order segments and their respective values from Stage 0. Moreover, this stage needs demand planning information that is the forecasted demand of each order segment in the planning period and supply planning information from the MPS that includes the ATP and CTP quantities available in the planning period. The mathematical model -MIP- helps in allocating the optimum quotas to the order segments in order to maximize potential order promising profit.

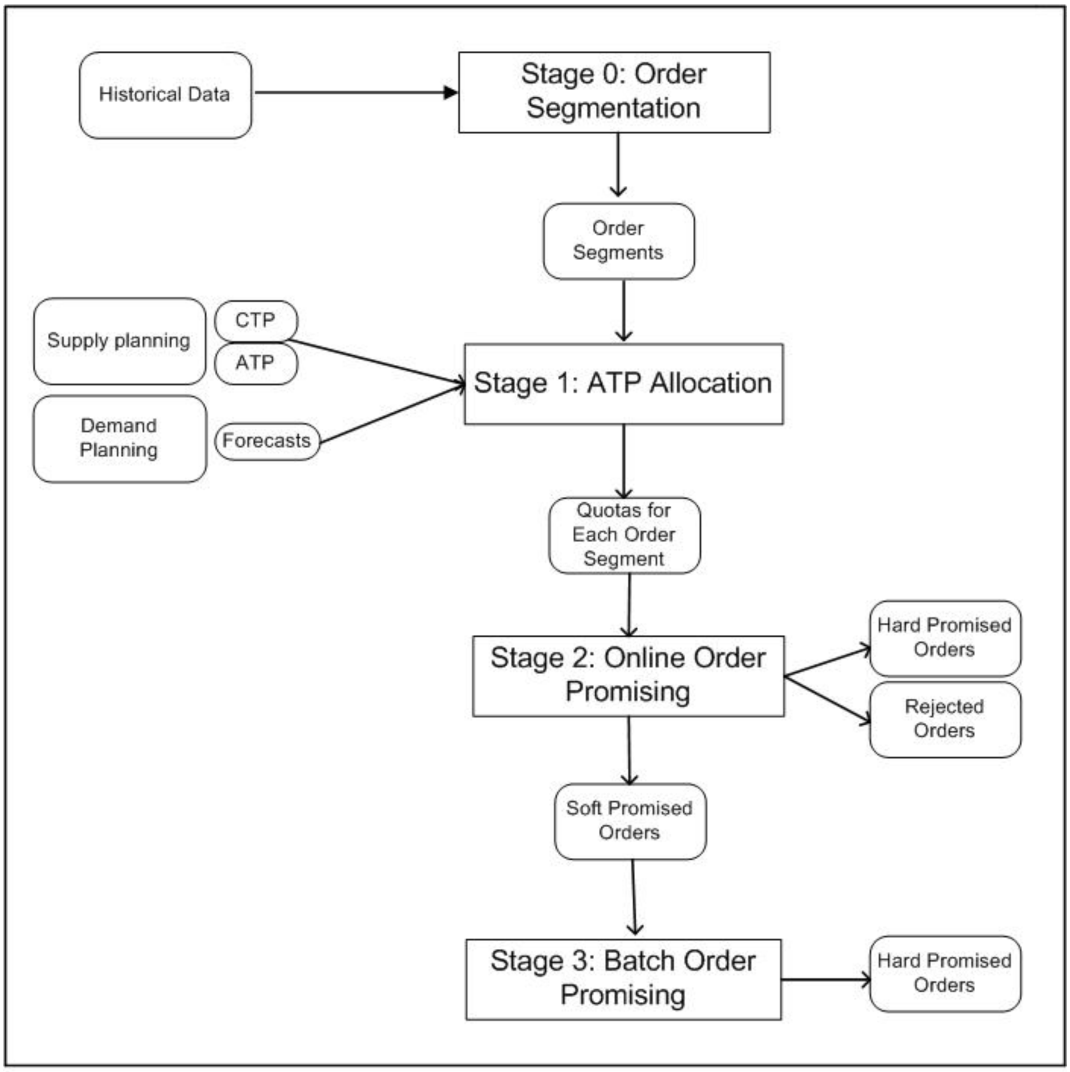


Figure 4-1: The four-stage solution approach

At Stage 2, a pull-based ATP consumption model is executed online or in very short time periods. Every order can get one of the three replies from the model: 1: The order is accepted and due date is given (hard promise), 2: The order is rejected for the planning time horizon, 3: The order is accepted, and given a time window for delivery (soft promise) and the date when the firm due date will be given. The orders that are soft promised are passed to stage 3. This approach contributes to the customer service levels by giving a reply to every order in a very short time interval.

At Stage 3, a batch ATP execution is done for the soft promised orders coming from the previous stage. Every order that comes to Stage 3 is certainly given an exact due-date; in other words, it is certainly hard promised.

Stage 0 can be considered a strategic planning phase that should be conducted for a long period of time; for example, at the beginning of every year for the entire year. Stage 1 can be considered a planning phase where global ATP is allocated to the order segments. Stage 1 can be executed for shorter time periods, for example, at the beginning of every half-year for the entire half-year. Stage 2 is almost an online execution mechanism that is executed more often. On the other hand, Stage 3 is a batch execution mechanism that is executed at the end of each batching horizon.

Now, we introduce our 4-stage solution framework stage by stage and elaborate on our models.

4.2. Stage 0: Order Segmentation

Stage 0 can be defined as a strategic planning phase in which order segments and their values (priorities) are determined. In this stage, we use historical data of sales transactions and demand. In other words we are not only interested in the orders accepted, but also in the orders rejected during the past years. Moreover, we also assume that both the total revenue and total cost of every order i are known whether the order is accepted or rejected. Revenue for the rejected order can be thought as the total price that the customer would be willing to pay at the time of the order.

Stage 0 has two main parts: first calculation of val_i and then clustering on val_i. First calculation of val_i is presented. Then clustering this val_i in order to get order clusters is examined.

Three major attributes contribute to the value indicator val_i of order i:

- Profit of order i: P_i (that is normalized as P_i^{norm} such that $0 \leq P_i^{norm} \leq 1$)

- Priority of customer of order *i*: C_i (that is normalized as C_i^{norm} such that 0 ≤ C_i^{norm} ≤ 1)
- Complexity value of the product configuration of order i: *Comp*$Comp_i$ (that is normalized as $Comp_i^{norm}$ such that 0 ≤ $Comp_i^{norm}$ ≤ 1)

The first attribute, profit of order i, P_i, is the difference of the total cost of order *i* from the total revenue of order *i*. We normalize P_i as:

$$P_i^{norm} = \frac{P_i - P^{min}}{P^{max} - P^{min}} \tag{4.1}$$

where P^{max} and P^{min} are the maximum and minimum profits that is observed in the past, respectively.

The second attribute, priority of customer who places the order *i*, C_i, is used to define the importance of the customer for the company. The sub-attributes that can be used to define the importance of the customer can be her profit contribution, historical relationships, contractual relationships, sector the customer belongs to, site (geography) of the customer, customer's sensitivity to order promising decisions, etc. Here it is important to emphasize that non-monetary factors also contribute to the priority of the customer.

Without any loss of generality we categorize the customers based on priority in three groups:

- Low priority customers (C_i = 1)
- Medium priority customers (C_i = 2)
- High priority customers (C_i = 4)

Here, the priority of the respective customer is a subjective criterion and should be determined by the business owner. Besides the techniques like AHP (Analytic Hierarchy Process) and some technical calculations, subjective comments of the business owners can be used to define them.

We also normalize C_i as:

$$C_i^{norm} = \frac{C_i - C^{min}}{C^{max} - C^{min}} = \frac{C_i - 1}{4 - 1} = \frac{C_i - 1}{3} \qquad (4.2)$$

where C^{max} and C^{min} are the pre-determined maximum and minimum priority values, respectively.

The third attribute, the complexity value of the product configuration of order i can be defined based on the cost of the product configuration. It is the cost of the MTM, plus costs of the add-in features. We use this attribute in order to emphasize that companies wish to produce and sell more complex and more value added products instead of the basic products. The reasons can be the respectability that they can get, the high competency they can acquire or the potential high profits and market share in the future.

Normalized cost of the product configuration of order i, $Cost_i^{norm}$, is used as a surrogate measure for the complexity of the product configuration:

$$Comp_i^{norm} = Cost_i^{norm} = \frac{Cost_i - Cost^{min}}{Cost^{max} - Cost^{min}} \qquad (4.3)$$

where $Cost^{max}$ and $Cost^{min}$ are the maximum and minimum costs that is observed in the past, respectively.

By using the three normalized attributes, the value val_i of order i can be calculated as:

$$val_i = w_1\, P_i^{norm} + w_2\, C_i^{norm} + w_3\, Comp_i^{norm} \qquad (4.4)$$

where w_j is the respective weight of the attribute in calculating val_i and without loss of generality:

$$\sum_{j=1}^{3} w_j \; = \; 30 \text{ and } 0 \leq \; val_i \leq \; 30. \tag{4.5}$$

The selection of w_j s defines the weights given to the three attributes in accordance with the company's business structure. For example, a company may give a higher value to w_j in order to make it more effective.

An interesting observation about this order valuation approach is that two orders having different normalized attribute values can have the same *val*$_i$. This is similar to the grading of students at the end of a term. A student whose homework grades are low but exam grades are high can get the same final grade as another student with higher homework grades and lower exam grades. In our case, an order belonging to a high priority customer but with a low profit margin can have the same value as an order belonging to a low priority customer with a high profit margin.

After calculating the values, *val*$_i$ s, of the orders from the past sales data, we come across with m different *val*$_i$ values associated with n orders, where $m \leq n$. The difference between *m* and *n* is because of the different orders having the same value. Since it is almost impossible to plan our supply chain for each order type, we cluster *n* orders (with *m* different values) into *k* classes in order to make an aggregation (see Figure 4-2).

Most of the clustering algorithms in data mining (like K-Means) require the number of clusters *k* to be given as input. It is very difficult to select the most appropriate number of clusters *k* in advance; hence it is usually done by trying different *k* values. In order to avoid this difficulty, we use the TwoStep Clustering component of SPSS. TwoStep Clustering is introduced in order to deal with large datasets and avoid the problem of determining the number of clusters *k* in advance.

SPSS TwoStep clustering method uses a two-step procedure that determines and then applies the optimum number of clusters to the given dataset. It uses a cluster number range, i.e., asks the user the minimum and maximum number of clusters. At its first stage, it goes through the data and evaluates the initial estimates of the number of clusters. Then in the second stage the initial estimate is refined and thus the optimum number of clusters is set; then each dataset member is assigned to a cluster.

After having the *m* different values of *n* many orders coming from the historical data, we use the SPSS TwoStep Clustering method to segment our orders into *k* number of classes where $1 \leq k \leq 3$. For example, $k = 2$ means that we have two order classes: high-priority orders and low-priority orders classes. For $k = 3$ we have low-priority orders, medium-priority orders and high-priority orders. This important decision variable *k* is determined in a dynamic manner by the algorithm in order to maximize inter-class difference and minimize in-class difference.

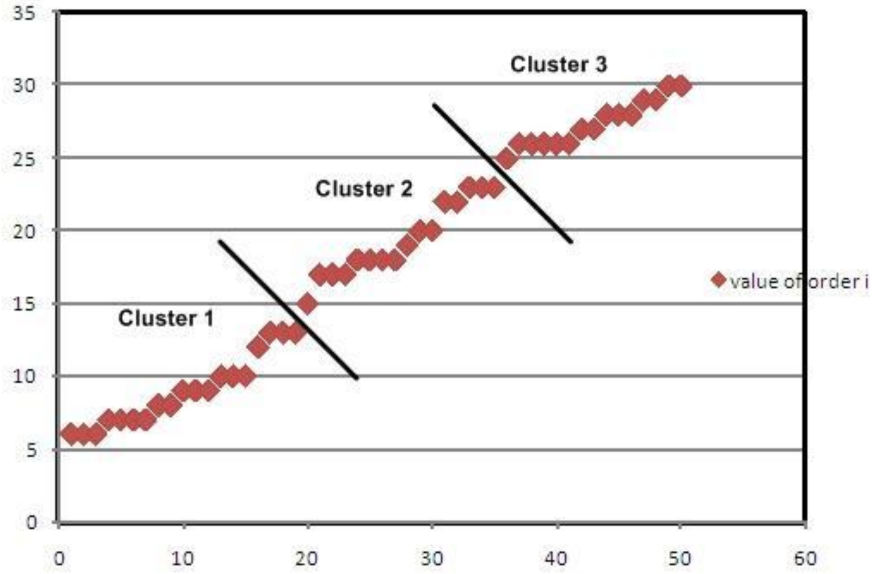


Figure 4-2: Example data set for clustering (x: orders, y: their values)

Here it is important to mention that, since we have only one dimension of data, that is val_i, it is possible to define order segments without using a statistical software. Segments can also be introduced by using average of val_i s or number of orders. However, especially when there is more than one dimension of data, it is worth using statistical software.

4.3. Stage 1: Allocation Planning

In Stage 1 of our availability management approach, we discuss our MIP model that aims at allocating ATP and CTP quantities to the order classes. First we introduce the relevant notation.

Indices:

$k = 1, \ldots, K$	order classes
$m = 1, \ldots, M$	fixed models
$p = 1, \ldots, P$	product families
$i = 1, \ldots, I$	parts
$t = 1, \ldots, T$	time periods

Parameters:

$ATP_{i,t}$	ATP value of part *i* in period *t*
h_i	inventory holding cost of one unit of part *i*
$CTP_{i,t}$	CTP value of part *i* in period *t*
c_i	production cost of one unit of part *i*
$d_{m,k,t}$	forecasted demand for fixed model *m* in order class *k* in period *t*
$d_{p,k,t}$	forecasted demand for product family *p* in order class *k* in period *t*
$v_{m,k}$	value of satisfying one unit of demand of fixed model *m* in order class *k*
$v_{p,k}$	value of satisfying one unit of demand of product family *p* in order class *k*
$r_{i,p}$	attach ratio of part *i* in product family *p*
$\alpha_{i,p}$	per unit penalty for the deficit allocation of part *i* to product family *p* with respect to $r_{i,p}$
$\beta_{i,p}$	per unit penalty for the excess allocation of part *i* to product family *p* with respect to $r_{i,p}$

$BOM_{i,m}$ the amount of part *i* required for a fixed product *m*

SLA_k required service level for order class *k*

Decision Variables:

$y1_{m,k.t}$ supply availability of fixed model *m* allocated to order class *k* in period *t*

$y2_{p,k,t}$ supply availability of product family *p* allocated to order class *k* in period *t*

$y3_{i,p,t}$ supply availability of part *i* allocated to product family *p* in period *t*

$y4_{i,m,k,t}$ supply availability of part *i* allocated to fixed model *m* and order class *k* in period *t*

$I_{i,t}$ ending inventory of part *i* in period *t*

$x_{i,t}$ quantity of part *i* produced in period *t*

$exc_{i,p,t}$ amount of excess allocation of part *i* to product family *p* in period *t* with respect to $r_{i,p}$

$def_{i,p,t}$ amount of deficit allocation of part *i* to product family *p* in period *t* with respect to $r_{i,p}$

MIP Model:

maximize

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{t=1}^{T} v_{m,k}\ y1_{m,k,t} + \sum_{p=1}^{P}\sum_{k=1}^{K}\sum_{t=1}^{T} v_{p,k}\ y2_{p,k,t} - \sum_{i=1}^{I}\sum_{t=1}^{T}(h_i\, I_{i,t} + c_i\, x_{i,t}) - \sum_{i=1}^{I}\sum_{p=1}^{P}\sum_{t=1}^{T}(\alpha_{i,p}\ exc_{i,p,t} + \beta_{i,p}\, def_{i,p,t}) \tag{4.6}$$

subject to

$$\sum_{t'=1}^{t} y1_{m,k,t'} \le \sum_{t'=1}^{t} d_{m,k,t'} \qquad \forall m,k,t \tag{4.7}$$

$$\sum_{t'=1}^{t} y2_{p,k,t'} \le \sum_{t'=1}^{t} d_{p,k,t'} \qquad \forall p,k,t \tag{4.8}$$

$$y4_{i,m,k,t} = y1_{m,k,t}\ BOM_{i,m} \qquad \forall i,m,k,t \tag{4.9}$$

$$x_{i,t} + ATP_{i,t} + I_{i,t-1} = I_{i,t} + \sum_{m=1}^{M}\sum_{k=1}^{K} y4_{i,m,k,t} + \sum_{p=1}^{P} y3_{i,p,t} \qquad \forall i,t \tag{4.10}$$

$$\sum_{t=1}^{T}\sum_{m=1}^{M} y1_{m,k,t} \ \ge\ SLA_k \sum_{t=1}^{T}\sum_{m=1}^{M} d_{m,k,t} \qquad \forall k \tag{4.11}$$

$$\sum_{t=1}^{T}\sum_{p=1}^{P} y2_{p,k,t} \ \ge\ SLA_k \sum_{t=1}^{T}\sum_{p=1}^{P} d_{p,k,t} \qquad \forall k \tag{4.12}$$

$$y3_{i,p,t} = \sum_{k=1}^{K} r_{i,p}\ y2_{p,k,t} + exc_{i,p,t} - def_{i,p,t} \qquad \forall i,p,t \tag{4.13}$$

$$x_{i,t} \leq CTP_{i,t} \qquad \forall i,t \qquad (4.14)$$

$$y1_{m,k,t}\ ,\ y2_{p,k,t}\ ,\ y3_{i,p,t}\ ,\ y4_{i,m,k,t} \geq 0 \qquad \forall m,k,t,p,i \qquad (4.15)$$

$$y1_{m,k,t},\ y2_{p,k,t},\ y3_{i,p,t} \geq 0\ , \text{ and } integer \qquad \forall m,k,t,p,i \qquad (4.16)$$

$$I_{i,t}\ , x_{i,t}, exc_{i,p,t}\ , def_{i,p,t} \geq 0 \qquad \forall t,p,i \qquad (4.17)$$

In (4.6) we have the objective function that maximizes total value (revenue) of allocating available parts to products and order classes minus production, inventory, backlogging costs, penalty for deviation from attach ratios.

In constraints (4.7) and (4.8) we limit the cumulative ATP allocation by the cumulative demand of order classes for fixed products and product families respectively.

In equation (4.9) the amount of a certain part allocated to a fixed model in an order class for a period is equated to the amount of usage of the part by that fixed model in that order class.

Constraint (4.10) is the primary constraint that balances the total inflow and total outflow of a part in a period. Total inflow is the sum of production amount, ATP, and inventory from the previous period; total outflow is the sum of inventory left to the following period, amount used for fixed product allocation, and amount used for product family allocation.

In constraints (4.11) and (4.12) the service level requirements are met for the corresponding order classes in all periods.

In constraint (4.13) the amount of a part allocated to a product family is forced to be equal to the expected usage of the part by making use of the attach ratio. The positive or negative deviation from the expected usage is punished by means of the excess and deficit variables in the objective function.

Constraint (4.14) is the CTP constraint that limits the production amount of a part by the CTP available in the corresponding time period.

Constraints (4.15) and (4.17) are the sign restrictions of the decision variables. Constraint (4.16) requires the main decision variables that are passed to Stage 2 to be integer.

Assembly capacity constraints for fixed model, product family and component assembly can be incorporated into the model in case any of these resources is a bottleneck resource.

One of the most interesting parts of the allocation model of our availability management approach is the meaning and calculation of the parameters $v_{m,k}$ and $v_{p,k}$ in the objective function that represent the values of allocating one unit of fixed model m to order class k and one unit of product family p to order class k, respectively. The calculation of these values is different for the fixed models and the product families as defined below:

i) Value for the fixed model ($v_{m,k}$)

In the order placement of the fixed model product configuration, the customer orders the model with a single part number the BOM structure of which is fixed. Fixed models have certain prices that are determined based on the costs of its constituent parts and a constant profit margin for them (e.g. 15%).

The price is constant for all of the order classes and the order; however, the value of satisfying the fixed model demand may change from one order class to another. We define the value of satisfying one unit of fixed model demand of order class k as:

$$v_{m,k} = f\{price(m), \bar{V}_{m,k}\} \tag{4.18}$$

where

$$\bar{V}_{m,k} = \sum_{i \in C_k^m} val_i \,/\, |C_k^m| \qquad (4.19)$$

In (4.19) we calculate the average value, $\bar{V}_{m,k}$, over all orders that include the fixed model m in the order class k, where $|C_k^m|$ is the number of orders that include the fixed model *m* in the order class *k*. Then we use both the price of the fixed model *m* and $\bar{V}_{m,k}$ to calculate $v_{m,k}$ by using the function *f* as expressed in (4.18).

The implementation of the function *f* can change from one company to another; but without loss of generality, the below implementation can be used where l is a constant coefficient:

$$f(price(m), \bar{V}_{m,k}) = price(m)(1 + l\ \bar{V}_{m,k}) \qquad (4.20)$$

ii) Value for the product family ($v_{p,k}$)

Unlike the fixed models, the BOM structure is not known for the configurations in the product families until the arrival of the customer order. Moreover, because of the unknown BOM, the price of the product family is not known in advance.

We introduce an approach that estimates the price of a product family by summing up the prices of its expected constituent parts according to the attach ratios:

$$est_p = price\ (base\ part\ of\ p) + \sum_{i=1}^{I} r_{i,p}\ price\ (part\ i) \qquad (4.21)$$

Then the calculation of $v_{p,k}$ is similar to the calculation of the values of the fixed models:

$$v_{p,k} = f\{est_p, \bar{V}_{p,k}\} \qquad (4.22)$$

$$\bar{V}_{p,k} = \sum_{i \in C_k^p} val_i \,/\, |C_k^p| \tag{4.23}$$

$$f(est_p, \bar{V}_{p,k}) \;=\; est_p(1 \,+\, l\ \bar{V}_{p,k}) \tag{4.24}$$

where $\bar{V}_{p,k}$ is the average value over all orders that include the product family *p* in the order class *k*, where $|C_k^p|$ is the number of orders that include the family *p* in the order class *k*, and *l* is a constant coefficient.

Based on the *val* 's of the objective function, the supply of the fixed models is allocated to the order classes, and the parts are allocated to the product families and the fixed products so as to maximize the value of satisfying the demand of the order classes (a derivative of revenue) minus costs.

4.4. Stage 2 and Stage 3: Online and Batch Order Promising

In Stage 2 of our 4-stage availability management approach, arriving customer orders are responded immediately or in a very short time period according to their order class and respective allocations. The main assumption is that every customer would prefer to be given a due date for their order as soon as possible (ASAP) which is very common for current business practices. The model is executed on a rolling horizon basis with a horizon length of *T*. The planning horizon *T* is divided into two parts, the first part with *t1* periods and the second part with *t2* periods (*t1<t2*). Every new customer order has the opportunity to be promised for the rest of the periods within *t1* periods and the forthcoming *t2* time periods. The orders which can be promised for the first part of the planning horizon (*t1* periods) are accepted and given a certain due date (hard promise). However, the orders which can be promised for the second part of the planning horizon (*t2* periods) are given a time window for the due date instead of an exact due date, which is *t2Start – t2End* (soft promise). The orders which can be promised neither a due date nor a due time window for the rest of the planning horizon within *t1* + *t2* are rejected (Figure 4-3, Figure 4-4).

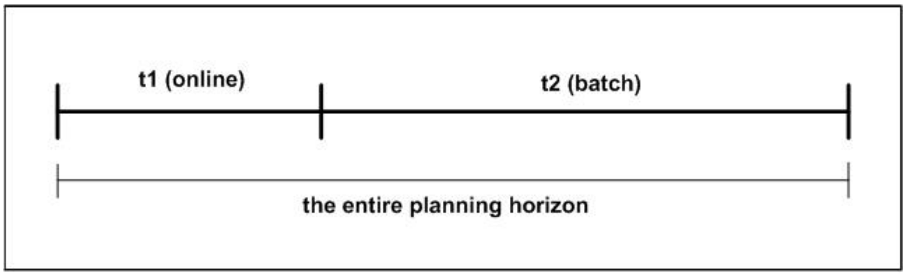


Figure 4-3: Planning time periods

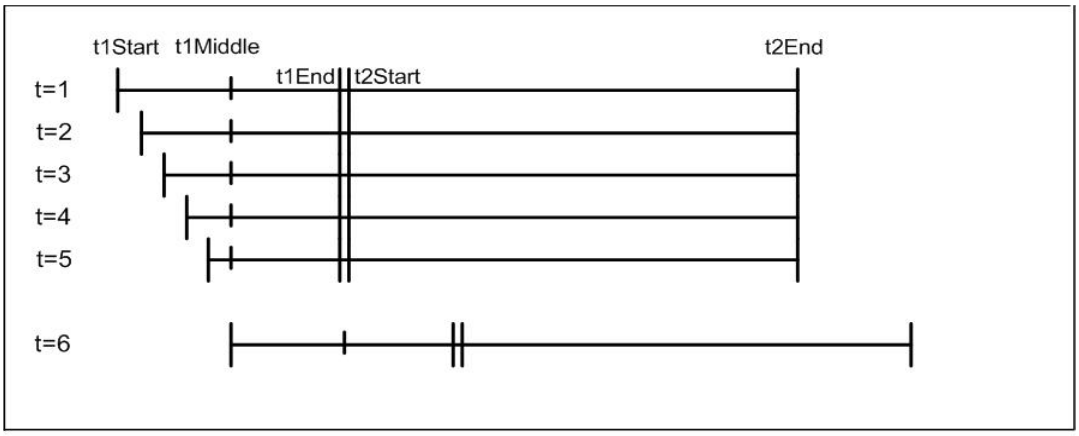


Figure 4-4: Rolling horizon

In order to cover the full planning horizon T (= *t1* + *t2*), we work on a rolling horizon basis. The planning horizon is divided into two parts and the middle point is named as *t1Middle*. Until the time point, *t1Middle*, all of the incoming orders are hard promised, if they can be promised till the end of *t1* periods or soft promised if they can be promised within the following *t2* periods. At the time point *t1Middle*, the batch order promising is executed for the orders which have been received and accumulated up to *t1Middle* and soft promised for the coming *t2* periods. Then the current planning horizon is rolled forward by a period of length which is (*t1Middle*-*t1Start*). This horizon rolling shifts both *t1* and *t2* planning periods forward and bring the execution to the initial state (Figure 4-5).

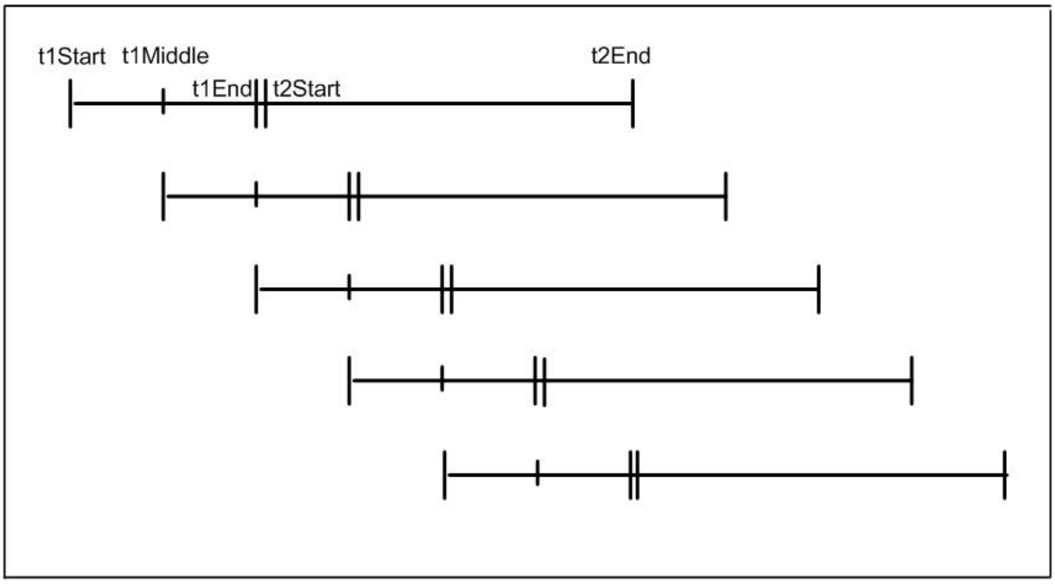


Figure 4-5: Full rolling horizon

The system considers the availability of all items in *t1* periods on a daily basis. On the other hand, the availability of all items in the following *t2* periods is aggregated on a full batch promising period basis. In other words, if the cumulative total availability of an item for the *t2* planning periods horizon is *y*, *t2* periods are aggregated as a single time period which can be considered as temporal aggregation and the aggregated period then is the period *t1End+1* with an availability of *y* for the corresponding item. This simplifying approach lets us solve the batch order promising problem in an integrated manner.

The online promising of the fixed models (FM) and product families (PF) are fairly different:

i) Online promising of the fixed models

Since the availability of a fixed model *m* allocated to an order class *k* in period *t*, $y1_{m,k,t}$, is based on the finished product and does not need consideration of the configuration parts, we only check the corresponding availability of model *m* starting from the time when the order arrives. If availability for model *m* in order class *k* is found until *t1End*, this order is hard promised and the availability of model *m* is decreased by one unit at the corresponding day. If availability for model *m* is found at *t2Start* which represents the aggregated availability of model *m* in the following *t2* periods, then this order is soft promised, given a due time window of [*t2Start*, *t2End*] and the availability of model *m* is decreased by one unit for *t2Start*. If there is no availability even in *t2Start*, the order is rejected according to the current rolling horizon (Figure 4-6).

An important property of fixed model promising is that we allow the higher order classes to consume the allocated availability of the lower order classes, when there is not sufficient availability allocated to their own order class in the planning period. This extension lets the system promise the higher order classes in case of supply shortage for their own allocated availability.

ii) Online promising of the product families

Promising of product families is a more complicated process than that of the fixed models. It requires promising every individual part selected by the customer and there after consolidation of promises that are given to these independent parts to create the main order promise result. The availability of all parts allocated to product family *p* in period *t*, $y3_{i,p,t}$, should be checked and consolidated.

Figure 4-7 depicts the process flow of the product family order promising. The availability of all parts required by the order is controlled and the first-time availability of them is recorded. The maximum of the first-time availabilities of the constituent parts determines either the hard promised due date or the due time window of the order.

However, if the due date falls in the *t2* time period, the order has to be soft promised by giving a due time window ([*t2Start*, *t2End]*). In this case we search for the availability of the parts in *t2* period whose due dates (the first availability) are actually up to *t1End*. If possible we shift the due dates of the corresponding parts to the next *t2* time periods starting after *t1End* in order not to consume these parts unnecessarily earlier than they are needed. If anyone of the parts' availability cannot be found up to time *t2End*, the order is rejected.

Another characteristic of the approach is that it lets the parts to be promised in more than one time periods. In other words, if the availability of the current time period is not enough for the requested quantity for that part, the available amount is promised, and the remaining (unmet) amount is shifted to the next period for availability lookup. It is normally expected that some of the parts will be configured with more than one unit in CTO orders.

In Stage 3 of our 4-stage availability management approach, the soft promised orders coming from Stage 2 are promised exact delivery due dates (hard promises) according to their due date time windows that are defined in stage 2. All of the soft promised orders are ordered in a decreasing sequence with respect to their order values. Then the system promises due dates for all of the orders starting from the one with the biggest value in the time period, *t2Start* in a forward fashion.

Normally, all of the orders that are soft promised in stage 2 will eventually be hard promised in Stage 3 of our approach which will give priority to the higher value orders in order to decrease the order lead time for the higher value orders as much as possible. This is also in accordance with our ASAP assumption in due date quoting.

Due date quoting (or hard promising) for the soft promised orders in a backward fashion, that is, starting in period *t2End* and coming back to period *t2Start* with the higher value orders first can be an alternative way to the forward quoting described above. In this way, orders are given due dates within their soft promised due time windows as late as possible. In spite of the fact that this backward due date quoting increases the order lead times and thus decreases customer service levels somehow, it may increase the fraction of the hard promised orders for the next *t1* periods in the rolled horizon which is the closest time to the then current time, because there will be more unused availabilities in the first *t1* periods of the next rolled horizon.

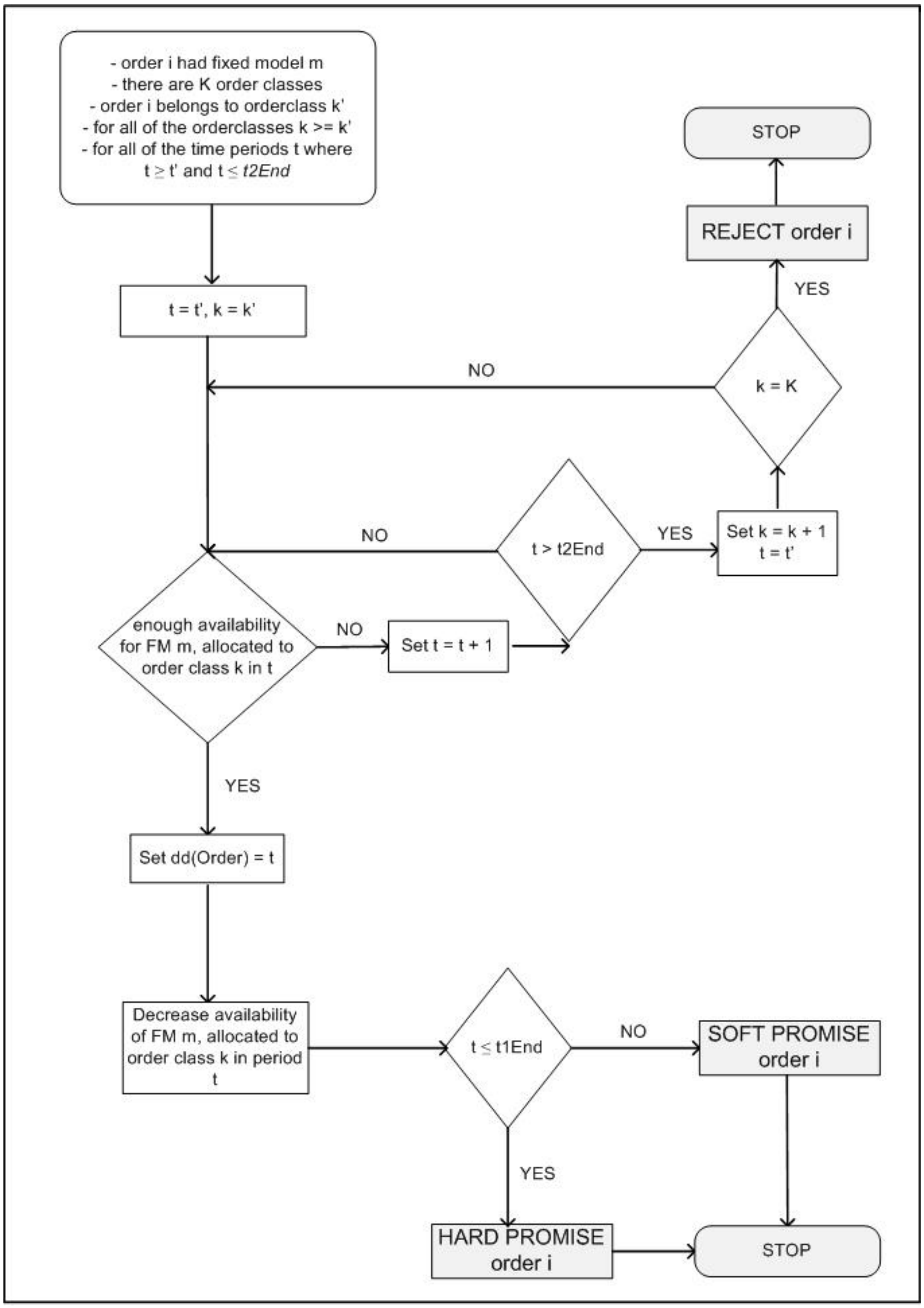


Figure 4-6: Fixed model order promising

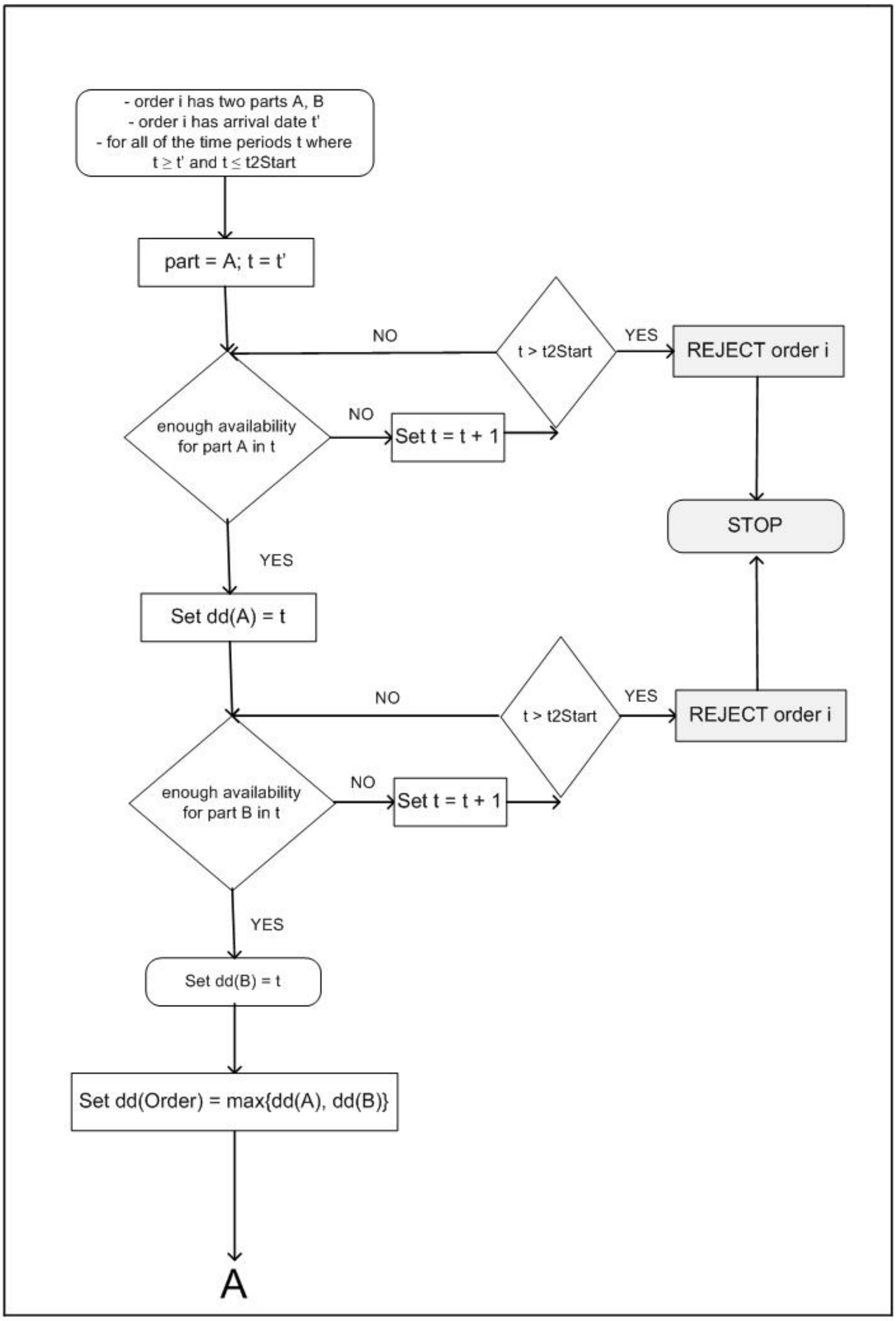


Figure 4-7: Product family order promising

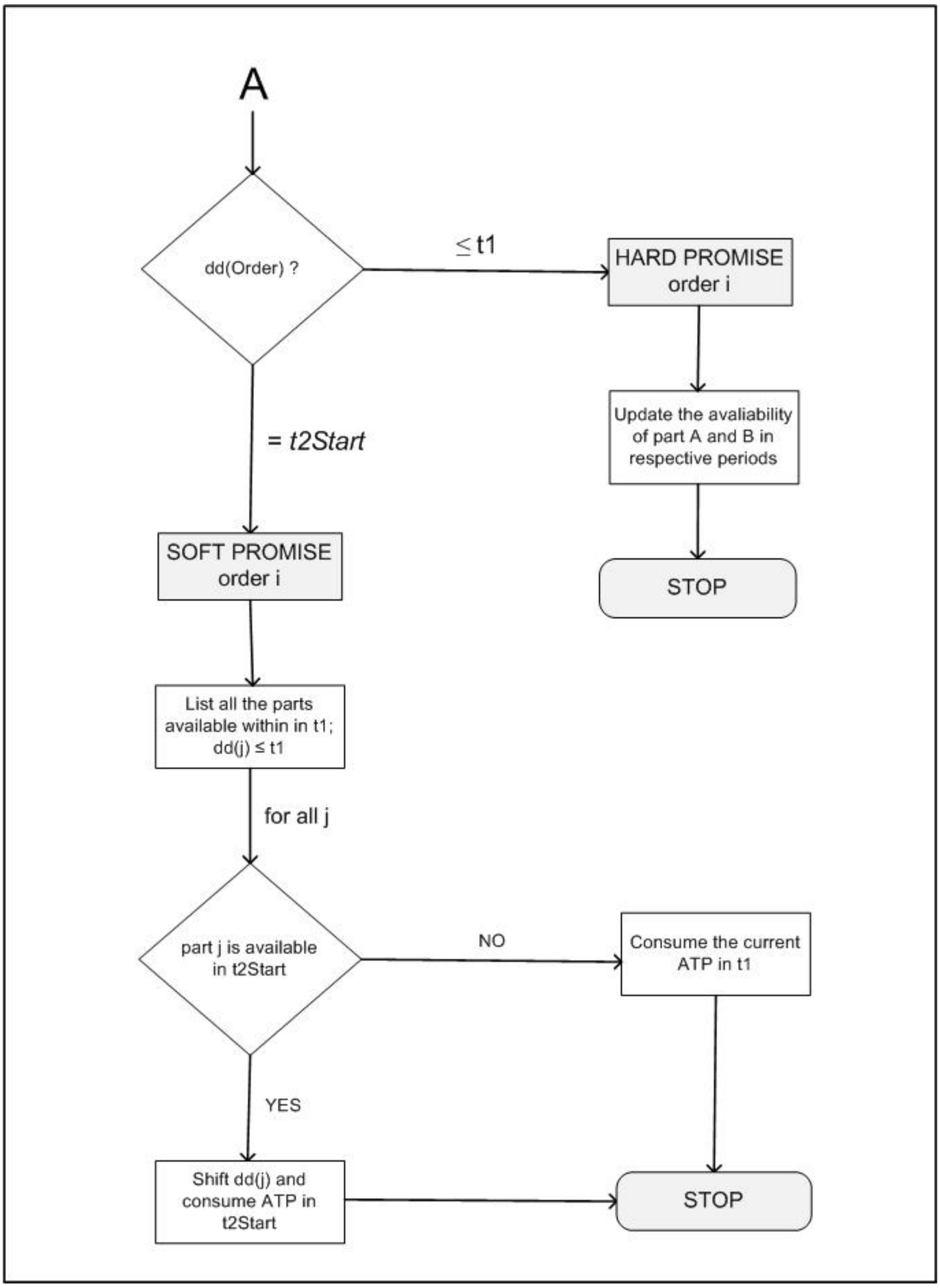

A
dd(Order) ?
≤ t1
HARD PROMISE order i
Update the avaliability of part A and B in respective periods
STOP
= t2Start
SOFT PROMISE order i
List all the parts available within in t1; dd(j) ≤ t1
for all j
part j is available in t2Start
NO
Consume the current ATP in t1
YES
Shift dd(j) and consume ATP in t2Start
STOP

Figure 4-7: Product family order promising (continued)

In some environments, the actual ATP and CTP quantities for the parts can deviate from the planned quantities. There may be an unexpected supply increasing the availabilities, and similarly there may be unexpected resource shortages decreasing the availabilities. In such cases the process of hard promising in stage 3 can be more interesting. There may even be situations in which some of the already soft promised orders should be rejected. Here, the choice of already soft promised orders for rejection can also be determined by this approach; for example, the orders that have the minimum contribution to the total value may be rejected.

CHAPTER 5

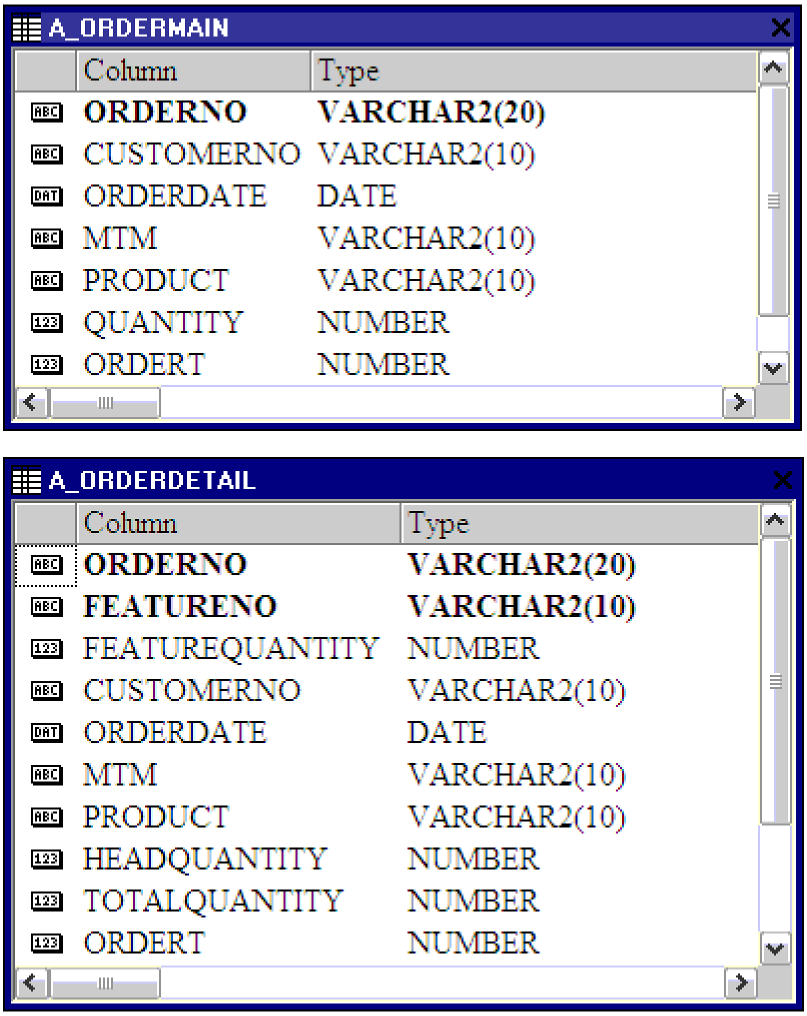
COMPUTATIONAL STUDY

Our 4-stage approach that includes the stages of order segmentation, allocation planning, both online order promising and batch order promising is tested with the real-life data of IBM Enterprise Server Hardware Division. We have all the information about the actual customer orders, product families, fixed models, parts, attach ratios and some other related information of the configuration environment for a defined time horizon of six months.

Firstly, we introduce the experiment data and define its attributes. Then we apply stage 0 to the data at the database level in order to determine the order segments. After the introduction of our MIP model at stage 1 that allocates availabilities to the predefined order segments, we describe our web-based Order Promising System (OPS), developed in Java for stages 2 and 3: online and batch order promising processes. Finally we go over our experimental runs and analyze the results.

5.1. Introduction to the Experiment Data

IBM Enterprise hardware division produces several servers to satisfy various demands from the customers. The system x series division consists of middle segments servers that can not only be used by big companies, but also by small and medium-sized companies. This is the reason why the business growth and transaction amounts are relatively larger and the prices are relatively lower with respect to the other divisions. We have disguised some parts of the data due to confidentiality. IBM has provided us with their order structure and actual transactions within a specific 6-month period. The original data seemed very complicated; for this reason, we have applied several data cleaning and data transformation processes. Then we converted the data to the following format; the entity relationship (ER) diagram that is more readable (Figure 5-1). The only information that was missing in the original data was the customer information, prices and costs. We have defined random customers and meaningful price/cost information, and associated them with the orders.



A_ORDERMAIN

	Column	Type
ABC	**ORDERNO**	**VARCHAR2(20)**
ABC	CUSTOMERNO	VARCHAR2(10)
DAT	ORDERDATE	DATE
ABC	MTM	VARCHAR2(10)
ABC	PRODUCT	VARCHAR2(10)
123	QUANTITY	NUMBER
123	ORDERT	NUMBER

A_ORDERDETAIL

	Column	Type
ABC	**ORDERNO**	**VARCHAR2(20)**
ABC	**FEATURENO**	**VARCHAR2(10)**
123	FEATUREQUANTITY	NUMBER
ABC	CUSTOMERNO	VARCHAR2(10)
DAT	ORDERDATE	DATE
ABC	MTM	VARCHAR2(10)
ABC	PRODUCT	VARCHAR2(10)
123	HEADQUANTITY	NUMBER
123	TOTALQUANTITY	NUMBER
123	ORDERT	NUMBER

Figure 5-1: ER diagram of the main data

Here there are two main structures. One of them is the ORDERMAIN which holds the order information and the other is the ORDERDETAIL which holds the information about the order configuration, i.e., the constituent parts of the configuration. We have analyzed these two main structures extensively to extract all of the necessary information we need in our study. We have extracted the MTMs, features (parts), time periods, attach ratios, single-entity-offerings (SEOs), configure-to-orders (CTOs) and some other necessary information from this initial data.

After doing some statistical analyses in order to define the mostly and uniformly used items we have selected 2 MTMs, 5 SEOs and 10 features for our experimental study which are found to be the most common and widely used components in the order transactions. The selected components are listed in Tables 5-1, 5-2 and 5-3. It is clearly seen that if a product name is ending with a "NEW", it means that this product is a CTO, in other words it has been configured by the customer.

Table 5-1: Selected MTM's

MTM	**LIST COST**	**LIST PRICE**
7978	9000	12000
7979	8000	10000

Table 5-2: Selected products (SEO + CTO)

MTM	**PRODUCT**	**TYPE**
7978	7978BJU	SEO
7978	7978EHU	SEO
7978	7978NEW	CTO
7979	7979B4U	SEO
7979	7979B9U	SEO
7979	7979NEW	CTO
7979	X1RDRUS	SEO

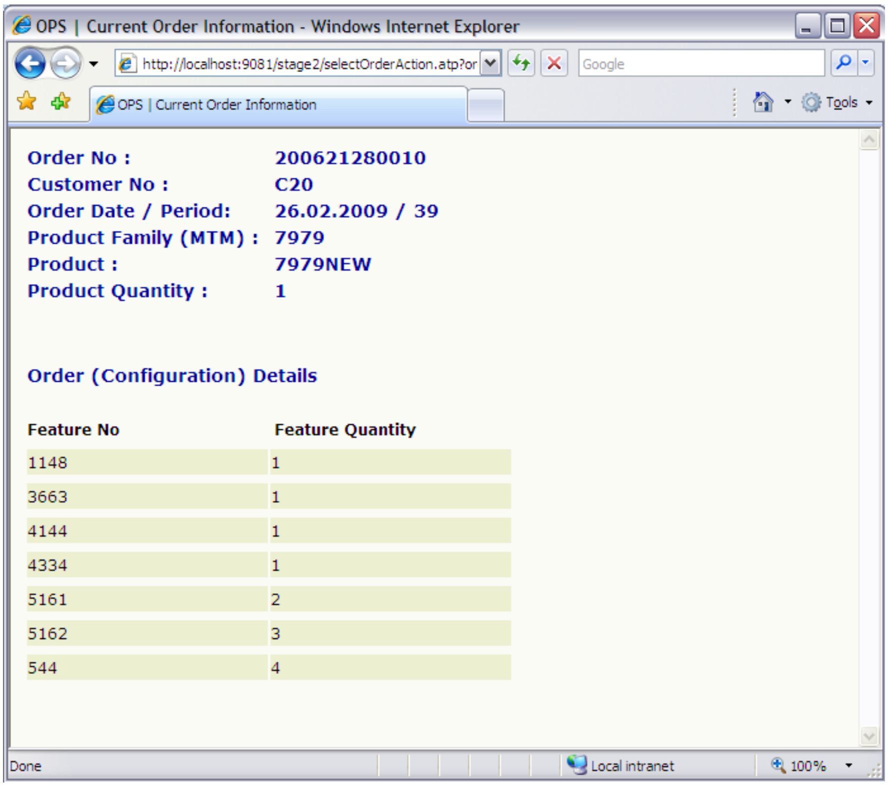
After these selections we come up with 2,897 orders and 11,583 order details (order configurations) within 120 time periods, which corresponds to approximately 24 orders per day. In Figure 5-2, an order structure is illustrated from the OPS. At the main part related to the order, we see the order no, customer, period, MTM, product, quantity information. At the order configuration part, we see seven feature configurations and their respective quantities.

Table 5-3: Selected features

F.NO	FEATURE NAME	LIST COST	LIST PRICE
1148	RF3 system planar	600	750
3663	QC In Pr E5430 2.66GHz/1333MHz	1200	1500
3682	QC IntXProc E5420 2.5GHz 80W	1500	1750
4144	CD-RW/DVD ComboV Ultrabay	300	500
4334	PCI-Express Riser card	500	700
5161	73GB 15K 3.5 Hot-Swap SAS HDD	700	900
5162	146GB 15K 3.5 H-Swap SAS HDD	1000	1300
542	1GB PC2-5300 CL5 ECC DDR2	100	140
544	2GB PC2-5300 CL5 ECC DDR2	200	250
556	4GB PC2-5300 CL5 ECC DDR2	400	550

Similar to the transformations mentioned above, we have made some other transformations to have a relational data structure which will construct the base of the OPS. We have tried to minimize data repetition and increase data ‘read and write’ transaction performance.

We have used the same data for all of the four stages. Normally using two different data sets, i.e., belonging to different time periods, for stages 0 & 1 and stages 2 & 3 is more appropriate. For this reason, we have used some randomization to create random deviations from the data used in stages 0 & 1, in order to have a different data at stages 1 & 2. Some examples are random disruption in costs, demand forecasts and attach ratios.



OPS | Current Order Information - Windows Internet Explorer
http://localhost:9081/stage2/selectOrderAction.atp?or
Google
OPS | Current Order Information
Tools
Done
Local intranet
100%

Order No : 200621280010
Customer No : C20
Order Date / Period: 26.02.2009 / 39
Product Family (MTM) : 7979
Product : 7979NEW
Product Quantity : 1

Order (Configuration) Details

Feature No	Feature Quantity
1148	1
3663	1
4144	1
4334	1
5161	2
5162	3
544	4

Figure 5-2: An example order structure from OPS

5.2. Stage 0: Order Segmentation

Order segmentation is the process of segmenting orders according to their values into meaningful number of order classes. An order value is calculated from tree main attributes: the normalized profit of that order, the normalized importance of the customer and the normalized complexity of the product. The calculation of these attributes is explained in detail at Chapter 4.

We assume all of the w_i's in equation (4.4) as 10. In other words we give the same importance to all of the three attributes generating the order value. After this assumption we come up with 758 distinct order values for 2897 orders having different values between 25.89 and 0.39. After applying SPSS Two-Step clustering to these 758 different order values, we get the order segments structure for k=2 order classes and k=3 order classes. In our experiments k=2 will be our main structure which we will use for much of the analysis and k=3 will be a comparison dataset. We also experiment results of order segmentation without using a clustering software.

5.3. Stage 1: Allocation Planning

We implement the MIP of Stage 1 by using ILOG OPL Development Studio v6.1.1 and ILOG CPLEX v11.2.1 which is given in Appendix A. We get input from an Excel sheet and write output to the database of OPS in our MIP model implementation.

We have forced three of our eight decision variables to be integer-valued in our MIP problem: $y1_{m,k,t}$, $y2_{p,k,t}$ and $y3_{i,p,t}$. As we mentioned in Chapter 4, there are two main outputs of Stage 1 that will be used as input in Stages 2 and 3: $y1_{m,k,t}$ and $y3_{i,p,t}$, hence only these decision variables are forced to be integer-valued. Each of the variables, $y4_{i,m,k,t}$, automatically becomes an integer because of integer BOM values. The other four decision variables, *I, x, exc, def,* take on values as either integer or non-integer.

Another approach to deal with the computational complexity of the MIP model might be relaxing all of the eight decision variables and rounding them later to integers. By this LP relaxation approach we can increase the performance of the model from the computational standpoint especially in larger problems. The rounding errors that might arise during the rounding process are negligible, because it should have very small effect on the optimality of the global 4-stage problem.

The calculation of the *value parameters* in the objective function is very important. We use the formulas (4.18) through (4.24) in order to calculate $v_{m,k}$ and $v_{p,k}$ for the MIP model.

We use a personal PC for all of our development and experiments having an Intel 2.0 GHz Dual Core CPU with 3 GB of memory. The MIP model has 18,484 constraints and 23,281 decision variables. CPLEX cannot find an optimum solution for the MIP problem and does not improve its solution significantly after a couple of minutes. That is why we use a time limit of 120 seconds after which we stop our execution, and use the current near optimal MIP solution (Figure 5-3).

Our MIP implementation writes the results of the decision variables, $y1_{m,k,t}$ and $y3_{i,p,t}$, to the database for the use of OPS at Stages 2 and 3.

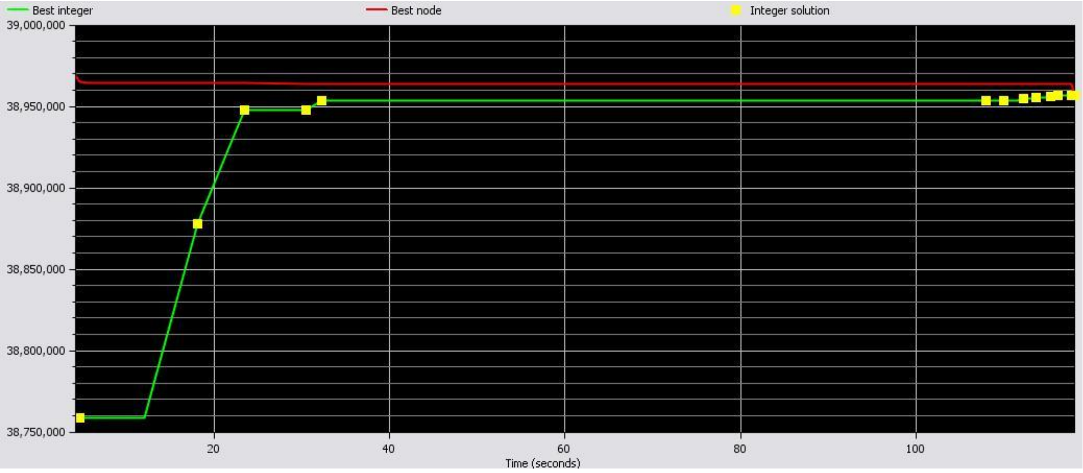


Figure 5-3: CPLEX statistics tab

5.4. Stage 2 and 3: Online and Batch Order Promising

For stages 2 and 3 we develop a web-based Order Promising System (OPS) that is developed by using Java programming language. The system is designed to be able to host online order promising, batch order promising, online/batch integrated order promising and near global optimization in order to be used in our analyses.

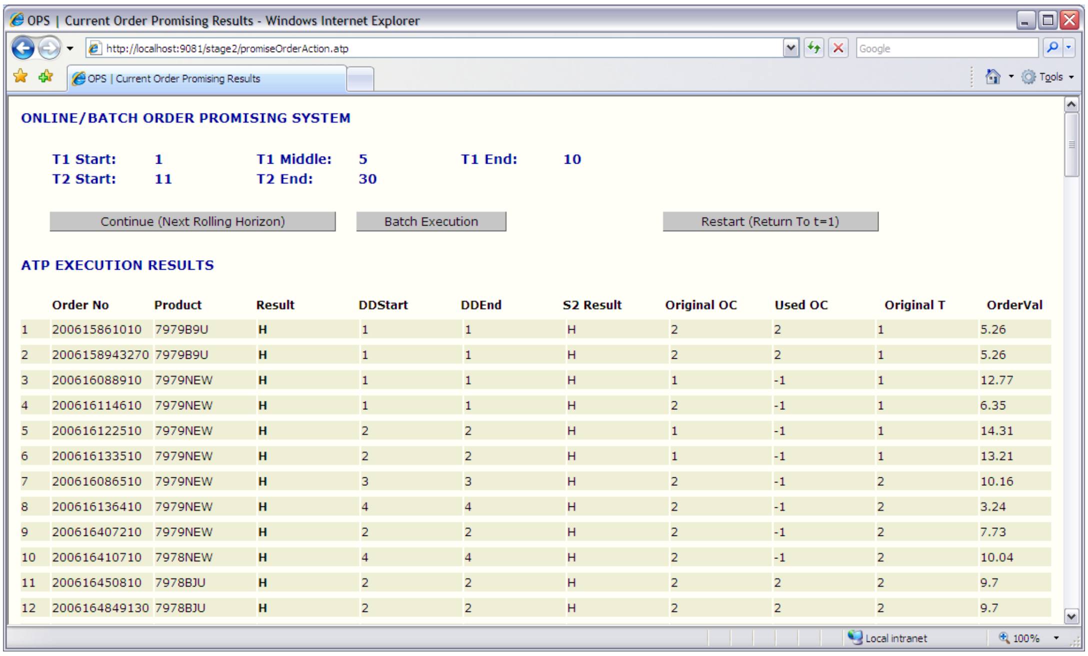
All of the parameters *t1Start, t1Middle, t1End, t2Start, t2End, rollingT*, *numberOfOrderClasses*, and *numberOfPeriods* can be changed easily for other scenarios or for the expectations of other enterprises.

As mentioned in Chapter 4, the system works on a rolling horizon basis with a total of 120 time periods. We assume that one week has 5 days; one month has 20 days and one month has 4 weeks. It is possible to execute the system one by one for every rolling horizon or all of the rolling horizons at a time. In Figure 5-4, we see the main order promising screen of OPS. In this screen it is possible to execute the system one by one for every rolling time horizon. The *Continue* button moves the system to the next rolling time horizon; the *Batch Execution* button executes Stage 3 batch order promising for the currently soft promised orders; and the *Restart* button takes the system to the initial position.

The columns of the *ATP Execution Results* give all of the important information to the user about the order promising results. Besides the basic information such as *OrderNo, Product*, and *Result*, it also reports the results of Stage 2 execution, order arriving date, value of the order, original order class and "used" order class. These results are also open for further analysis by other external systems, since they are input to the database.

In addition to the main order promising screen, it is also possible to get the promising details of features for CTO orders. Since order promising is done at the feature level for CTO orders, the system reports them to the user. For example, when we double click one of the CTO order results, we get the following screen in Figure 5-5.

The main order that has the above configuration (Figure 5-5) arrives at OPS at t=7. At this time the hard promising time interval (*t1*) is between t=7 and t=15, and the batch promising interval (*t2)* is between t=16 and t=35 which is represented by t=16 only with the aggregated availabilities.


OPS | Current Order Promising Results - Windows Internet Explorer
http://localhost:9081/stage2/promiseOrderAction.atp
Google
OPS | Current Order Promising Results
Tools
Local intranet
100%

ONLINE/BATCH ORDER PROMISING SYSTEM

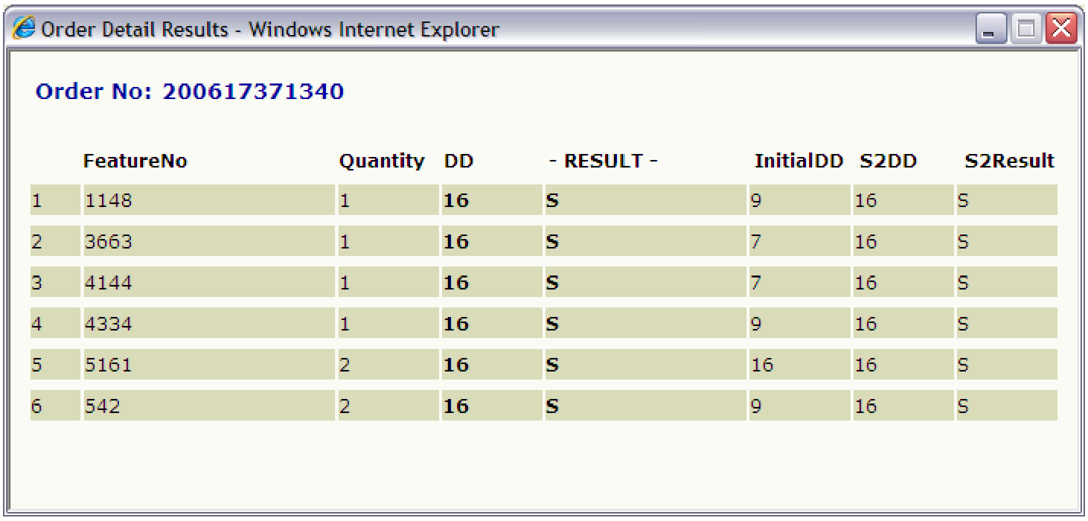
T1 Start:	1	T1 Middle:	5	T1 End:	10
T2 Start:	11	T2 End:	30		

Continue (Next Rolling Horizon)
Batch Execution
Restart (Return To t=1)

ATP EXECUTION RESULTS

	Order No	Product	Result	DDStart	DDEnd	S2 Result	Original OC	Used OC	Original T	OrderVal
1	200615861010	7979B9U	H	1	1	H	2	2	1	5.26
2	2006158943270	7979B9U	H	1	1	H	2	2	1	5.26
3	200616088910	7979NEW	H	1	1	H	1	-1	1	12.77
4	200616114610	7979NEW	H	1	1	H	2	-1	1	6.35
5	200616122510	7979NEW	H	2	2	H	1	-1	1	14.31
6	200616133510	7979NEW	H	2	2	H	1	-1	1	13.21
7	200616086510	7979NEW	H	3	3	H	2	-1	2	10.16
8	200616136410	7979NEW	H	4	4	H	2	-1	2	3.24
9	200616407210	7979NEW	H	2	2	H	2	-1	2	7.73
10	200616410710	7978NEW	H	4	4	H	2	-1	2	10.04
11	200616450810	7978BJU	H	2	2	H	2	2	2	9.7
12	2006164849130	7978BJU	H	2	2	H	2	2	2	9.7

Figure 5-4: OPS main order promising screen



Order No: 200617371340

	FeatureNo	Quantity	DD	- RESULT -	InitialDD	S2DD	S2Result
1	1148	1	16	S	9	16	S
2	3663	1	16	S	7	16	S
3	4144	1	16	S	7	16	S
4	4334	1	16	S	9	16	S
5	5161	2	16	S	16	16	S
6	542	2	16	S	9	16	S

Figure 5-5: Order detail results for a soft promised order

The Features 1148, 4334 and 542 are available at t=9 which is within *t1*. The features 3663 and 4144 are available at t=7 which is also within *t1*. The only feature which is not available within *t1* is 5161, which is available in t=16 within *t2* period. This causes the main order to be soft promised. According to the algorithm that we have proposed in Chapter 4, all of the features that we initially promised within *t1* (InitialDD column) are tried to be shifted to *t2* interval. We see that from S2DD column all of them are successfully shifted to t=16 within *t2*. After that, the main order is soft promised with a time window of t=16 and t=35 which are actually *t2Start* and *t2End* values of the current iteration.

The above case illustrates one of the main properties of our proposed approach for CTO environments especially where there is no fixed BOM.

5.5. Experiments

In order to be able to investigate the behavior of our approach under different situations, we have created different solution methods out of our main approach and tested them via different problem instances.

We have defined two problem instances to diversify the problem and generated several solution methods. As we mention above, we have created our main problem instance from the real life order transactions of IBM after some transformations including data cleaning, transformation and normalization. This problem instance (experiment data) is our main problem instance on which most of the test runs are based (D1). In order to generate a different problem instance, we have increased all of the ATP quantities of the features which we have actually defined according to statistical analyses and diversification by 20% (D2). After this increase, all of the availabilities that will be assigned to order classes and the general results are changed. We keep the order transactions as the same in order not to move away from the real life data.

We generate various solution methods out of our main 4-stage approach in order to investigate the behavior of our approach and the stages individually. These solution methods are differentiated based on the following characteristics:

- number of order classes
- clustering approach

- stages of the main 4-stage approach that are skipped
- length of the batch order promising time interval
- length of the online order promising time interval

From the order class perspective, we create four different cases with three different order class numbers K=1 (K1), K=2 (K2) and K=3 (K3). In K1, we have only one order class and all of the orders belong to the same class. In other words we do not use Stage 0, that is, we skip Stage 0. In K2, we have two order classes with k=1 being the high priority orders and k=2 being the low priority orders. This case is our main selection in this dimension. In K3, we have three order classes with k=1 being the high priority orders, k=2 being the normal priority orders, and k=3 being the low priority orders. Moreover, for the case K=2, we have defined an extra case where no clustering software is used for order segmentation. Here the number of distinct order values is divided by two and then orders are assigned to order classes based on their position. If they are in the half having higher values, they are assigned to order class 1; if they are in the half having lower order values, they are assigned to order class 2 (K2NCS: K2-No-Clustering Software).

Then we generate seven different solution methods out of our main 4-stage approach by including all or some of the four stages:

NGO: Near Global Optimization. All of the 2897 orders within 120 time periods are promised together with the rule-based batch promising feature of OPS.

N1020: Normal (online/batch) execution with *t1*=10 and *t2*=20 days.

B30: Batch order promising within 30 days. Here Stage 2 of our approach (online order promising) is skipped.

O10: Online order promising within 10 days. Here Stage 3 of our approach (batch order promising) is skipped.

N2040: Normal (online/batch) execution with *t1*=20 and *t2*=40 days.

B60: Batch order promising within 60 days. Here Stage 2 of our approach (online order promising) is skipped.

O20: Online order promising within 20 days. Here Stage 3 of our approach (batch order promising) is skipped.

When we consolidate these seven solution methods that are independent of the order clustering method, and the clustering methods called as K1, K2, K3 and K2NCS, we can come up with 49 different solution methods. We have only experimented with 29 of them which we think is sufficient to understand the behavior of the 4-stage approach. Table 5-4 shows the solution methods together with the information which are implemented (as marked by X).

When order class type or problem instance is changed, the Stage 1 MIP is to be solved with the new dataset. However, changing the solution method (independent of the order classes) only requires running OPS again with different parameters in Stages 2 and 3.

Table 5-4: Problem instances and solution methods

	D1				**D2**		
	K1	**K2**	**K3**	**K2NCS**	**K1**	**K2**	**K3**
NGO	X	-	-	-	X	-	-
N1020	X	X	X	X	X	X	X
B30	X	X	X	X	X	X	X
O10	X	X	X	X	X	X	X
N2040	X	X	-	-	-	-	-
B60	X	X	-	-	-	-	-
O20	X	X	-	-	-	-	-

After examining the problem instances and solution methods, the performance measures of the executions should be given. All of the test runs' solutions are evaluated with respect to the three performance measures:

1. Total value of orders that are hard promised within the full planning horizon
2. Number of hard promised orders
3. Average lead time

All of these performance measures are also measured with respect to the number of order classes. The first one can be thought as the primary performance measure and the other two can be thought as secondary performance measures.

5.5.1. Experiments With the Problem Instance D1

The problem instance D1 that corresponds to the original company data has been solved using each of the nineteen solution methods. All of the performance measures obtained with each solution method, i.e., total value, number of hard promised orders, average lead time are listed in Table 5-5.

It is seen that the highest total order value is achieved by NGO with 18,638.8. The highest number of hard-promised accepted orders is achieved by two order classes with N2040 method with a value of 1701. The smallest average lead time is achieved by K2NCS with B30 method with a value of 0.906 days.

When we compare the N (Normal), B (Batch) and O (Online) methods with the same time horizons, we see that the total value is maximized in batch methods and minimized in online methods as expected. It is important to notice that, sometimes, despite the fact that the number of hard promised orders is decreased; the total value is increased because the orders that have higher values are promised. We can see this interesting result when we compare the methods N1020 and B30. Method B30 has 99 less orders that are hard promised, but has a total order value which is 858.22 units higher.

Table 5-5: Performance measures' values - Problem instance D1

		Clustering Method			
Solution Method	**Performance Measure**	**K1**	**K2**	**K3**	**K2NCS**
	Total value	18638.8	-	-	-
NGO	# of hard promised orders	1655	-	-	-
	Avg. lead time	7.961	-	-	-
	Total value	16340.37	16564.68	16752.46	15988.06
N1020	# of hard promised orders	1681	1668	1682	1615
	Avg. lead time	9.052	8.889	8.724	9.015
	Total value	17198.59	17162.95	17131.06	16372.15
B30	# of hard promised orders	1582	1558	1565	1473
	Avg. lead time	1.277	1.394	1.285	0.906
	Total value	15840.59	15989.4	15919.79	15047.56
O10	# of hard promised orders	1626	1612	1606	1524
	Avg. lead time	2.413	2.493	2.250	2.587
	Total value	16458.6	16884.54	-	-
N2040	# of hard promised orders	1686	1701	-	-
	Avg. lead time	10.883	11.171	-	-
	Total value	17716.6	17857.82	-	-
B60	# of hard promised orders	1601	1612	-	-
	Avg. lead time	1.877	2.386	-	-
	Total value	16008.9	16278.54	-	-
O20	# of hard promised orders	1651	1645	-	-
	Avg. lead time	6.114	6.141	-	-

When we compare the method N1020 with respect to the clustering method (number of order classes), we see that the total value increases when the number of order classes increases. This is also true for N2040 method and the other shorter term methods such as O10 and O20, with the exception of the decrease in O10 from K2 to K3. Without loss of generality we can conclude that order classification has increased the total order value.

It is also important to notice that using clustering software instead of manually segmenting the orders through dividing the number of distinct total order values into two increases the total order value in the methods N1020, B30 and O10.

When we examine the average lead times, we notice that pure batch executions are more successful from the lead-time perspective. On the other hand 2-stage order promising methods have the maximum average lead times. However, one should also consider that in 2-stage order promising methods, a time window is given to the customers at the time of the order and the exact due date is also given to them as soon as possible (i.e., every Friday afternoon) which may somehow increase customer satisfaction.

When we examine the solution method N1020 in more detail, we get some more insights. In Table 5-6, 5-7, 5-8 and 5-9 the detailed results of the solution method N1020 for the three order classes are listed. At first glance, it is clearly seen that the percentage of hard promised CTOs is lower than the percentage of hard promised SEOs. This is an interesting result that needs more attention.

Table 5-6: Order promising results based on number of order classes

Number of Order Classes	Result	Number of Orders
1	Hard Promised	1681
1	Rejected	1216
2	Hard Promised	1668
2	Rejected	1229
3	Hard Promised	1682
3	Rejected	1215

Table 5-7: Order promising results based on number of order classes and product types

Number of Order Classes	Product Type	Result	Number of Orders
1	CTO	Hard Promised	532
1	CTO	Rejected	1104
1	SEO	Hard Promised	1149
1	SEO	Rejected	112
2	CTO	Hard Promised	552
2	CTO	Rejected	1084
2	SEO	Hard Promised	1116
2	SEO	Rejected	145
3	CTO	Hard Promised	540
3	CTO	Rejected	1096
3	SEO	Hard Promised	1142
3	SEO	Rejected	119

At Stage-1 MIP model, in CTO configurations, features are attached to MTMs in a loosely coupled manner by means of equation (4.13). In other words, by tolerating the deficiency costs, which are set as 1.2 times the production costs in our experiments, it is possible to have a solution in which certain amount, $y_{p,k,t}$, is allocated to an order class, but without a sufficient amount of features attached to this MTM. This flexibility for CTOs, which does not exist for the fixed models, results in lower CTO promises in the overall. One way to avoid this might be increasing the deficiency costs. However, having increased the deficiency costs, we end up with lower total objective function values and fewer fixed model allocations that also decrease the overall performance of the OPS.

Table 5-8: Order promising results based on number of order classes, order class and product types

Number of Order Classes	Order Class	Product Type	Result	Number of Orders
1	1	CTO	Hard Promised	532
1	1	CTO	Rejected	1104
1	1	SEO	Hard Promised	1149
1	1	SEO	Rejected	112
2	1	CTO	Hard Promised	214
2	1	CTO	Rejected	439
2	1	SEO	Hard Promised	259
2	1	SEO	Rejected	4
2	2	CTO	Hard Promised	338
2	2	CTO	Rejected	645
2	2	SEO	Hard Promised	857
2	2	SEO	Rejected	141
3	1	CTO	Hard Promised	187
3	1	CTO	Rejected	374
3	1	SEO	Hard Promised	233
3	1	SEO	Rejected	1
3	2	CTO	Hard Promised	218
3	2	CTO	Rejected	477
3	2	SEO	Hard Promised	687
3	2	SEO	Rejected	66
3	3	CTO	Hard Promised	135
3	3	CTO	Rejected	245
3	3	SEO	Hard Promised	222
3	3	SEO	Rejected	52

It is also observed in Table 5-9 that more than 50% of the hard promised CTO orders are firstly soft promised at stage 2. The main reason for that is the resource shortage for CTO orders.

Table 5-9: Soft promised orders based on number of order classes

Number of Order Classes	Product Type	Result	Number of Orders
1	CTO	Soft Promised	296
1	SEO	Soft Promised	512
2	CTO	Soft Promised	328
2	SEO	Soft Promised	422
3	CTO	Soft Promised	327
3	SEO	Soft Promised	462

5.5.2. Experiments With the Problem Instance D2

The problem instance D2 that corresponds to the new dataset having more allocations with respect to D1 has been solved using ten solution methods. All of the performance measures obtained with each solution method, i.e., total value, number of hard promised orders, average lead time are listed in Table 5-10.

It is clearly seen that all of the total order values and number of hard promised orders increase and all of the average lead times decrease with respect to the problem instance D1. This is somewhat expected, because we have increased the resources by 20 %, but left the order structure as is. In other words, the same amount of demand is satisfied by 20 % more availability.

In this problem instance D2, we achieve the maximum total order value as 21,164.85 with the method NGO as expected. The maximum number of hard promised orders is achieved as 1975 by the method N1020 with one order class only. The minimum average lead time is achieved as 0.868 days by the method B30 with 3 order classes.

It is also clearly seen that as the number of order classes is increased, the total order value also increases in N1020 method. The only decrease in the total order value is observed in the methods O10 and B30 from K2 to K3.

Table 5-10: Performance measures' values - Problem instance D2

		Clustering Method		
Solution Method	**Performance Measure**	**K1**	**K2**	**K3**
	Total value	21164.85	-	-
NGO	# of hard promised orders	1927	-	-
	Avg. lead time	7.942	-	-
	Total value	19549.78	19635.99	19735.62
N1020	# of hard promised orders	1975	1969	1969
	Avg. lead time	8.195	8.396	7.883
	Total value	19118.05	19265.49	19179.79
B30	# of hard promised orders	1758	1770	1755
	Avg. lead time	1.120	1.045	0.868
	Total value	18218.95	18651.31	18498.77
O10	# of hard promised orders	1863	1887	1850
	Avg. lead time	2.190	2.295	2.294

Similar to the results of the experiments with D1, average lead time is shorter with the pure batch methods (B10); however, it is higher in 2-stage order promising methods (N1020). Again it is worth mentioning that, in 2-stage methods, customers can get immediate answers for their order which means an improvement in customer satisfaction.

One of the most unexpected results of this dataset D2 is the decrease in both the number of hard promised orders and total order value with the B30 method compared to the N1020 method. This result may be due to the order structure. When a high value order with an order value of *a* is promised, it is possible that two lower value orders having order values *b* and *c* respectively may be rejected because of resource shortage that may result after promising the previous high value order *a*. If $a < b + c$, then both the number of hard promised orders and total order value may decrease. This might be a drawback of the batch order promising logic of OPS.

5.6. Conclusion and General Comments about the Experiments

Having obtained the results for the two problem instances D1 and D2, there are some common observations that can be stated for the overall behavior of the approach.

Firstly, increasing the number of order classes increases the total order value with the exception of the pure batch methods as expected. Secondly, in 2-stage order promising methods, it is possible to reach similar total order values as in pure batch order promising methods, especially when there is more supply that can be used to satisfy the demand. Without a 2-stage order promising approach, all of the soft promised orders from stage 2 are to be rejected which may result in degradation in the total performance as in pure online methods.

The relative importance of the performance measures is very decisive while determining the overall performance of the solution methods. In our approach the main performance measure is the total order value, because all of the stages of our approach try to maximize it. The normal way of doing it is also to maximize the number of hard promised orders, but as we illustrate above, sometimes total order value maximization may decrease the number of hard promised orders. Besides promising orders ASAP, nothing has been done in order to minimize average lead times, i.e. backorder costs, increasing inventory holding costs, etc.

Testing the proposed approach with other real life data especially with those belonging to other sectors might add some more value to the approach. Moreover having more accurate forecasts and an execution horizon with more than six months may help to validate the performance of the approach in a better way.

CHAPTER 6

CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

In this study, a 4-stage availability management approach is proposed for CTO production environments. Before the arrival of the customer orders, the orders are segmented based on the historical sales data and then ATP quotas are allocated to these segments by means of an MIP model in order to increase the potential profit that can be obtained while promising the actual orders. Then the actual orders are promised by a rule-based Order Promising System (OPS) that is developed by Java.

The insights of the study are experimented on IBM Enterprise Server Hardware division’s processes and data. The applicability of the proposed approach to the dynamically changing needs is tested and verified through the experimental runs based on several scenarios.

The proposed 4-stage approach can be considered an end-to-end solution approach that includes both push-based planning and pull-based execution processes. Since the stages are loosely coupled with each other, one or more of the stages can be removed as we illustrate in our experiments or replaced by another internal or external system according to the needs of the process and the enterprise.

The results of the OPS are kept in the database. This lets the other external transactional or planning systems such as ERP, MPS or APS access that database and read the order promising results. Moreover, some interfaces can be given to the customers that enable them to check their order status from the central system. Alternatively, informing the customers by external messaging systems such as SMS or e-mails can be added to the solution.

The assumption of eventually hard promising every soft promised order at stage 3 can be relaxed. This might be done in order to incorporate some supply and demand changes into the model in the forthcoming time periods. On the other hand, since our proposed approach will certainly make some of the soft promised customers unhappy, some offerings to those customers might be developed such as discount option for their other future orders or paying them penalty costs.

Dynamic pricing, i.e., negotiating the price of the order with the customer during the order promising process and deciding whether to accept or reject the order by considering these negotiations can be added to the proposed approach. Moreover economies of scale, while determining the price and quantity of an order might be an interesting area to research.

Applicability and value of the proposed approach for the other production environments such as ATO, MTO and MTS might be investigated deeply. Actually, our fixed model option is very similar to ATO and MTS environments. That is why the approach can easily been adapted to them with only slight changes. Moreover, hybrid production environments containing MTS, CTO and MTO processes at the same time might be investigated particularly from the availability management perspective.

Determining the number of order classes and the actual orders classes can be considered to be one of the most challenging parts of the proposed approach. Moreover, the attribute -customer priority- that contributes to the value of the order is a subjective item that should be determined by the company. Increasing the number of order classes may increase the effectiveness of the approach; however, not only the computations and analyses become more complex, but also the approach becomes more vulnerable to forecast errors with respect to the order classes. In our approach, we have used only one dimension -total order value- as the segmentation attribute. On the other hand, more than one attribute can be used for order segmentation which means that the computations should be carried out in more than one dimension. In such cases, the use of a clustering algorithm/software like K-Means or Two-Step Clustering becomes more valuable and unavoidable, since it becomes even impossible to define the relationships among the attribute values manually.

In our approach, for batch order promising, we use a rule-based java execution mechanism that tries to promise the orders having higher values as soon as possible. In other words, the order which has the highest value is hard promised first, then the second one and so on among the incoming orders during the whole batching horizon. This approach finds near-optimal solutions, because there is a possibility to get better results through a more involved optimization-based approach. The batch ATP mechanisms that are proposed in Ball et al. (2004), Chen et al. (2000), Chen et al. (2001) and Chen (2003) can provide better results from the total order value, number of hard promised orders and average order lead time perspectives.

In our 4-stage order promising approach, we do not use given due dates for the orders and backorder costs in both stage 1 and stages 2 & 3 online/batch java execution. We assume that all of the orders are expected to be promised as soon as possible (ASAP) and we give all the responsibility for early order promise to inventory holding costs. Introducing order due dates and backorder costs for the incoming orders or order classes might differentiate the problem and make the problem more appropriate and valid for some other enterprises and sectors.

During our experimental studies, we do not face with performance problems in OPS system since our data are relatively small. In real life executions, especially where the number of transactions (orders) is extremely larger and number of clients using OPS at the same time is high, some technical considerations such as high availability, clustering, caching, etc. should be taken on board. Moreover, in e-business environments, where customers are also the stakeholders of the system, the response times of OPS might be a critical performance measure.

REFERENCES

Ball, M. O., Chen, C. Y. and Zhao, Z. Y. (2003). "Material compatibility constraints for make-to-order production planning." Operations Research Letters 31(6): 420-428.

Ball, M. O., Chen, C. Y. and Zhao, Z. Y. (2004). Available to promise. Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era. Simchi-Levi, D., Wu, S. D. and Shen, Z. M., Kluwer Academic Publishers: 447-483.

Chen, C. Y. (2003). Optimization based available-to-promise. Decision and Information Technologies. University of Maryland. Doctor of Philosophy Thesis.

Chen, J. H. and Chen, C. T. (2007). "Improvement of Order Promise with Material Constraints and Finite Capacity." International Journal of the Computer the Internet and Management 15(2): 63-69.

Chen, C. Y., Zhao, Z. and Ball, M. O. (2000). "A Model for Batch Advanced Available-to-Promise." Production and Operations Management 11(4): 424-440.

Chen, C. Y., Zhao, Z. Y. and Ball, M. O. (2001). "Quantity and Due Date Quoting Available to Promise." Information Systems Frontiers 3(4): 477-488.

Chen, C. H., Zhao, Z. and Ball, M. O. (2009). "An Available-to-Promise Model for Periodical Order Promising of a Non-perishable Resource." Presented at the INFORMS national meeting in Atlanta, GA, October 21, 2003.

Defregger, F. and Kuhn, H. (2007). "Revenue management for a make-to-order company with limited inventory capacity." OR Spectrum 29(1): 137-156.

Ervolina, T. and Dietrich, B. (2001). "Moving toward dynamic available-to-promise." Supply Chain Management Practice and Research: Status and Future Directions: 1-19.

Fleischmann, B. and Meyr, H. (2003). "Customer orientation in advanced planning systems." Supply chain management and reverse logistics: 297.

Gordon, V., Proth, J. M. and Chu, C. (2002). "A survey of the state-of-the-art of common due date assignment and scheduling research." European Journal of Operational Research 139(1): 1-25.

Han, J. and Kamber, M. (2001). Data Mining: Concepts and Techniques, Morgan Kaufmann.

Jeong, B., Sim, S. B., Jeong, H. S. and Kim, S. W. (2002). "An available-to-promise system for TFT LCD manufacturing in supply chains" Computers & Industrial Engineering 43(1-2): 191-212.

Jung, H., Chen, F. F. and Jeong, B. (2007). Trends in supply chain design and management, Springer.

Keskinocak, P. and Tayur, S. (2004). "Due date management policies." D. Simchi-Levi, S.D. Wu, Z.M. Shen, eds. Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era. Kluwer Academic Publishers, Norwell, MA, 485 – 553.

Kilger, C. and Schneeweiss, L. (2008). "Demand fulfilment and ATP." Supply chain management and advanced planning: 181–198.

Lee, M. Y. (2006). Simulating impact of available-to-promise generation on supply chain performance. Proceedings of the 38th conference on Winter simulation. Monterey, California, Winter Simulation Conference.

Lee, Y. M. (2006). "Analyzing the Effectiveness of Availability Management Process." In: Jung H, Chen FF, Jeong B (eds) Trends in supply chain design and management: technologies and methodologies. Springer, Heidelberg

Littlewood, K. (1972). "Forecasting and control of passenger bookings." AGIFORS Annual symposium Proceedings 12: 95-117.

Meixell, M. J. and Chen, C. Y. (2004). A Scenario-Based Bayesian Forecasting Engine for Stochastic Available-to-Promise. 15th Annual Production and Operations Management Conference.

Meyr, H. (2008). Clustering methods for rationing limited resources. Lars Monch, Giselher Pankratz, eds., Intelligente Systeme zur Entscheidungsunterstutzung. Multikonferenz Wirtschaftsinformatik, Munchen, 26.02.2008-28.02.2008, SCS Publishing House eV, San Diego et al., 19-31

Meyr, H. (2008). "Customer segmentation, allocation planning and order promising in make-to-stock production." OR Spectrum (in press).

Meyr, H., Rohde, J., Wagner, M. and Wetterauer, U. (2005). "Architecture of selected APS." Supply Chain Management and Advanced Planning. Concepts, Models, Software and Case Studies: 241-249.

Meyr, H., Wagner, M. and Rohde, J. (2005). "Structure of advanced planning systems." Supply Chain Management and Advanced Planning–Concepts, Models, Software and Case Studies, 2nd ed., Springer, Berlin.

Moses, S., Grant, H., Gruenwald, L. and Pulat, S. (2004). "Real-time due-date promising by build-to-order environments." International Journal of Production Research 42(20): 4353-4375.

Pan, Y. and Shi, L. (2004). A stochastic on-line model for shipment date quoting with on-time delivery guarantees, Proceedings of the 2004 Winter Simulation Conference, (eds.) Ingalls RG, Rossetti MD, Smith JS, Peters BA.

Pibernik, R. (2005). "Advanced available-to-promise: Classification, selected methods and requirements for operations and inventory management." International Journal of Production Economics 93: 239-252.

Pibernik, R. (2006). "Managing stock-outs effectively with order fulfilment systems." Journal of Manufacturing Technology Management 17(6): 721–736

Quante, R., Fleischmann, M. and Meyr, H. (2009). "A Stochastic Dynamic Programming Approach to Revenue Management in a Make-to-Stock Production System." ERIM Report Series Reference No. ERS-2009-015-LIS. Available at SSRN: http://ssrn.com/abstract=1365058

Quante, R., Meyr, H. and Fleischmann, M. (2007). "Revenue Management and Demand Fulfillment: Matching Applications, Models, and Software." OR Spectrum 31(1) 31–62.

Robinson, A. G. and Carlson, R. C. (2007). "Dynamic order promising: real-time ATP." International Journal of Integrated Supply Management, 3:283–301, 2007.

Simchi-Levi, D., Kaminsky, P. and Simchi-Levi, E. (2000). "Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies" Irwin McGraw-Hill, New York, NY (2000).

Spengler, T., Rehkopf, S. and Volling, T. (2007). "Revenue management in make-to-order manufacturing—an application to the iron and steel industry." OR Spectrum 29(1): 157-171.

Stadtler, H. (2005). "Supply chain management and advanced planning—basics, overview and challenges." European Journal of Operational Research 163(3): 575-588.

Talluri, K. T. and Van Ryzin, G. J. (2004). The theory and practice of revenue management, Kluwer Academic Publishers, 2004.

Taylor, S. G. and Plenert, G. J. (1999). "Finite Capacity Promising." Production and Inventory Management Journal 40: 50-56.

Tsai, K. and Wang, S. (2009). "Multi-site available-to-promise modeling for assemble-to-order manufacturing: An illustration on TFT-LCD manufacturing." International Journal of Production Economics 117: 174-184.

Wikipedia. (2009), Last Updated 13.02.2009. Wikipedia. Retrieved 24.05.2009, from http://en.wikipedia.org/wiki/Available-to-promise.

Xiong, M., Tor, S. B., Khoo, L. P. and Chen, C. H. (2003). "A web-enhanced dynamic BOM-based available-to-promise system." International Journal of Production Economics 84(2): 133-147.

Xiong, M. H., Tor, S. B. and Khoo, L. P. (2003). "WebATP: a Web-based flexible available-to-promise computation system." Production Planning & Control 14(7): 662-672.

Zhao, Z., Ball, M. O. and Kotake, M. (2005). "Optimization-Based Available-To-Promise with Multi-Stage Resource Availability." Annals of Operations Research 135(1): 65-85.

APPENDIX

STAGE 1 OPL/CPLEX MODEL

```
// *** DATA ***
{string}        ProductFamilies = ...;          // MTMs
{string}        FixedModels = ...;              // SEOs
{string}        Features = ...;
{int}           Periods = ...;
{int}           OrderClasses = ...;

tuple BOMTuple {
        key string m;
        key string f;
        int BOMAmount;
};
{BOMTuple} BOM with m in FixedModels, f in Features = ...;
int BOMValue[myBOM in BOM] = myBOM.BOMAmount;

tuple ATPCTPTuple{
        key string f;
        key int t;
        int ATPAmount;
        int CTPAmount;
};
{ATPCTPTuple} ATPCTP with f in Features, t in Periods = ...;
int ATPValue[myATPCTP in ATPCTP] = myATPCTP.ATPAmount;
int CTPValue[myATPCTP in ATPCTP] = myATPCTP.CTPAmount;

tuple ForecastFMTuple {
        key string m;
        key int k;
        key int t;
        int forecastAmountFM;
};
{ForecastFMTuple} ForecastFM with m in FixedModels, k in OrderClasses, t in Periods =
...;
int ForecastFMValue[myForecastFM in ForecastFM] = myForecastFM.forecastAmountFM;

tuple ForecastPFTuple {
        key string p;
        key int k;
        key int t;
        int forecastAmountPF;
};
{ForecastPFTuple} ForecastPF with p in ProductFamilies, k in OrderClasses, t in Periods =
...;
int ForecastPFValue[myForecastPF in ForecastPF] = myForecastPF.forecastAmountPF;

tuple ValueTuple{
        key string mp;
        key int k;
        float val;
};
{ValueTuple} ValueFM with mp in FixedModels, k in OrderClasses = ...;
float ValueFMk[myValueFM in ValueFM] = myValueFM.val;

{ValueTuple} ValuePF with mp in ProductFamilies, k in OrderClasses = ...;
float ValuePFk[myValuePF in ValuePF] = myValuePF.val;

tuple SLATuple {
        key int k;
        float SLAk;
};
{SLATuple} SLA with k in OrderClasses = ...;
float SLAkValue[mySLA in SLA] = mySLA.SLAk;

float c[Features] = ...;
float h[Features] = ...;

tuple AttachRateTuple {
        key string f;
        key string p;
        float attachRate;
};
{AttachRateTuple} AttachRate with f in Features, p in ProductFamilies = ...;
float AttachRateValue[myAttachRate in AttachRate] = myAttachRate.attachRate;

tuple ExcDefTuple {
        key string f;
        key string p;
        int exc;
        int def;
};
{ExcDefTuple} ExcDef with f in Features, p in ProductFamilies = ...;
int ExcValue[myExcDef in ExcDef] = myExcDef.exc;
int DefValue[myExcDef in ExcDef] = myExcDef.def;

int NumberOfPeriods = ...;

// *** DECISION VARIABLES ***
dvar int+ yFM[FixedModels][OrderClasses][Periods];
dvar float+ yFMFeature[Features][FixedModels][OrderClasses][Periods];
dvar int+ yPF[ProductFamilies][OrderClasses][Periods];
dvar int+ yPFFeature[Features][ProductFamilies][Periods];
dvar float+ x[Features][Periods];
dvar float+ I[Features][Periods];
dvar float+ exc[Features][ProductFamilies][Periods];
dvar float+ def[Features][ProductFamilies][Periods];

// *** OBJECTIVE FUNCTION ***
maximize
        sum (m in FixedModels, k in OrderClasses, t in Periods) ValueFMk[<m,k>] *
yFM[m][k][t]
  + sum (p in ProductFamilies, k in OrderClasses,t in Periods) ValuePFk[<p,k>] *
yPF[p][k][t]
                - sum (f in Features, t in Periods) h[f] * I[f][t]
                - sum (f in Features, t in Periods) c[f] * x[f][t]
                - sum (f in Features, p in ProductFamilies, t in Periods) ExcValue[<f,p>] *
exc[f][p][t]
                - sum (f in Features, p in ProductFamilies, t in Periods) DefValue[<f,p>] *
def[f][p][t];

// *** CONSTRAINTS ***
subject to {

// Constraint 1 - FixedModel demand
forall ( m in FixedModels, k in OrderClasses, t in Periods )
        sum ( t1 in 1..t ) yFM[m][k][t1] <= sum ( t1 in 1..t ) ForecastFMValue[<m,k,t1>];


// Constraint 2 - ProductFamily demand
forall ( p in ProductFamilies, k in OrderClasses, t in Periods )
        sum ( t1 in 1..t ) yPF[p][k][t1] <= sum ( t1 in 1..t ) ForecastPFValue[<p,k,t1>];

// Constraint 3 - calculate y-i,m,k,t
forall (m in FixedModels, f in Features, k in OrderClasses, t in Periods)
        yFMFeature[f][m][k][t] == yFM[m][k][t] * BOMValue[<m,f>];

// Constraint 3 - is divided into two to avoid t-1=0 problem
forall (f in Features)
        ATPValue[<f,1>] + x[f][1] ==
                I[f][1] + sum (m in FixedModels, k in OrderClasses)
(yFMFeature[f][m][k][1]) +

        sum(p in ProductFamilies) yPFFeature[f][p][1];

forall (f in Features, t in 2..NumberOfPeriods)
        ATPValue[<f,t>] + I[f][t-1] + x[f][t] ==
                I[f][t] + sum (m in FixedModels, k in OrderClasses)
(yFMFeature[f][m][k][t]) +

        sum(p in ProductFamilies) yPFFeature[f][p][t];

// Constraint 4 - FixedModel SLA
forall (k in OrderClasses)
sum ( m in FixedModels, t in Periods ) yFM[m][k][t] >=
                SLAkValue[<k>] * sum ( m in FixedModels, t in Periods )
ForecastFMValue[<m,k,t>];
```

```
// Constraint 5 - ProductFamilies SLA
forall(k in OrderClasses)
sum ( p in ProductFamilies, t in Periods ) yPF[p][k][t] >=
                    SLAkValue[<k>] * sum ( p in ProductFamilies, t in Periods )
ForecastPFValue[<p,k,t>];

// Constraint 6 - AttachRates - Excess - Deficit
forall ( f in Features, p in ProductFamilies, t in Periods)
        yPFFeature[f][p][t] == sum (k in OrderClasses) AttachRateValue[<f,p>] *
yPF[p][k][t] + exc[f][p][t] - def[f][p][t];

// Constraint 7 - CTO Capacity
forall (f in Features, t in Periods)
        x[f][t] <= CTPValue[<f,t>];

}

// POST PROCESSING

tuple sonucSIIF{
        string a;
        int b;
        int c;
        float d;
};
{sonucSIIF} yFMler = {<m,k,t,yFM[m][k][t]> | m in FixedModels, k in OrderClasses, t in
Periods};
{sonucSIIF} yPFler = {<p,k,t,yPF[p][k][t]> | p in ProductFamilies, k in OrderClasses, t in
Periods};

tuple sonucSSIF{
        string a;
        string b;
        int c;
        float d;
};
{sonucSSIF} yPFFeatureler = {<f,p,t,yPFFeature[f][p][t]> | f in Features, p in
ProductFamilies, t in Periods};

tuple sonucSSIIF{
        string a;
        string b;
        int c;
        int d;
        float e;
};
{sonucSSIIF} yFMFeatureler = {<f,m,k, t,yFMFeature[f][m][k][t]> | f in Features, m in
FixedModels, k in OrderClasses, t in Periods};

tuple sonucSIFF{
        string a;
        int b;
        float c;
        float d;
};
{sonucSIFF} XIler = {<f, t, x[f][t], I[f][t] > | f in Features, t in Periods};

tuple sonucSSIFF{
        string a;
        string b;
        int c;
        float d;
        float e;
};

{sonucSSIFF} ExcDefler = {<f, p, t, exc[f][p][t], def[f][p][t] > | f in Features, p in
ProductFamilies, t in Periods};
```