

AN ONTOLOGY FOR INTEGRATED PLANNING AND SCHEDULING  
SYSTEMS WITH HIERARCHIES

by

Osman Nihat Koyuncu

B.S., Systems Engineering, Yeditepe University, 2006

Submitted to the Institute for Graduate Studies in  
Science and Engineering in partial fulfillment of  
the requirements for the degree of  
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2009

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis supervisor, Assoc. Prof. Ali Tamer Ünal, for his continuous support, guidance, encouragement and motivation throughout this study. Working with him was a privilege. I was lucky to have one of the best faculty members with his strong technical background and kind personality. I am thankful to his invaluable comments and always being there to coach me through difficulties.

I would also wish to express my sincere thanks to Prof. Ümit Bilge and Prof. Tülin Aktin for their suggestions as members of my thesis committee.

I would like to thank to all members of my family for their never-ending support, encouragement and patience.

I would like to thank my wife Gamze Koyuncu for her enlightening ideas helping me during this work.

Finally, I would like to thank TUBITAK very much for the financial support during my education.

## ABSTRACT

### AN ONTOLOGY FOR INTEGRATED PLANNING AND SCHEDULING SYSTEMS WITH HIERARCHIES

In this study, an ontology is proposed for hierarchical detailed scheduling and planning systems. By using this ontology, planning and scheduling problems at different levels can be modeled. A complete system which considers long term planning if it includes detailed information like it is in short term planning takes a lot of time and is hard to solve in many cases. Due to this reason it is beneficial to divide the complete problem into sub problems which leads to hierarchical planning systems. Main elements of the hierarchical planning system are bucket based capacity planning and continuous time advanced planning and scheduling. Bucket based capacity planning is the long term planning element in which tactical, strategic decisions are taken and continuous time advanced planning and scheduling is the short term planning item in which operational decisions are taken.

Existing of hierarchy planning between bucket based capacity planning and continuous time advanced planning and scheduling brings along necessity for aggregation between hierarchical planning system elements. In a hierarchical planning system , there is an aggregation from base level to top level element of the hierarchy. Different types of aggregation which exist in real life applications are studied in this work.

Also,in order to show that the ontology proposed in the study is applicable to modeling real life problems, integration scenarios between plan and scheduling for companies from different sectors are modeled.

## ÖZET

# HİYERARŞİ İÇEREN BÜTÜNLEŞİK PLANLAMA VE ÇİZELGELEME SİSTEMLERİ İÇİN BİR ONTOLOJİ

Bu çalışmada planlama ve detaylı planlama sistemlerinin bütünleşik modellemesi sağlamak amacıyla bir ontoloji geliştirilmiştir. Bu ontoloji kullanılarak, değişik seviyelerdeki planlama ve çizelgeleme problemleri modellenmiştir. Kısa dönem çizelgelemedeki detaylı bilgiyi içeren uzun dönemli planlamayı kapsayan bir kompleks sistemin çözümü çok uzun zaman almakta ve pek çok durumda çözümü bulmak çok zor olmaktadır. Bu nedenle ana problemi küçük problemler halinde ayrıştırarak çözmeyi içeren hiyerarşik planlama sistemleri daha faydalıdır.

Hiyerarşik planlama sisteminin ana elemanları planlama ve çizelgeleme modelleridir. Planlama sistemi uzun dönemli ve taktiksel kararların alındığı bir sistem iken çizelgeleme sistemi kısa vadeli ve operasyonel kararların alındığı bir sistemdir. Her iki planlama sistemi bu çalışmada modellenmiştir.

Hiyerarşik planlama sisteminin varlığı, bütünleştirme gerekliliğini ortaya koyar. Bu birleştirmeye neden ise planlama sisteminin geniş kapsamlı ve az detaylı olması fakat çizelgeleme sisteminin dar kapsamlı ve detaylı olmasıdır. Hiyerarşik planlama sistemlerinde alt seviyeden üst seviyeye doğru bir bütünleşme mevcuttur. Farklı tipte bütünleşme çeşitleri bu çalışmada incelenmiştir.

Ayrıca ontolojinin gerçek hayat problemlerini modellemek için kullanılabileceğini göstermek amacıyla farklı sektörlerden şirketler için geliştirilmiş orta seviye plan ve detaylı plan bütünleştirme senaryoları modellenmiştir.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	iii
ABSTRACT . . . . .	iv
ÖZET . . . . .	v
LIST OF FIGURES . . . . .	viii
1. INTRODUCTION . . . . .	1
2. LITERATURE REVIEW . . . . .	4
3. METHODOLOGY . . . . .	16
4. GENERAL MODEL . . . . .	18
4.1. Structure . . . . .	18
4.2. Definitions . . . . .	21
5. ADVANCED PLANNING SCHEDULING MODEL . . . . .	24
5.1. Introduction . . . . .	24
5.2. Structure . . . . .	27
5.3. Definitions . . . . .	27
6. AGGREGATION SCENARIOS AND AGGREGATE MODEL . . . . .	29
6.1. Aggregation Scenarios . . . . .	29
6.1.1. Time Aggregation . . . . .	29
6.1.2. Part Aggregation . . . . .	30
6.1.3. Demand Aggregation . . . . .	31
6.1.4. Resource Aggregation . . . . .	32
6.1.5. Process Aggregation . . . . .	34
6.1.5.1. Aggregation on Operation Level . . . . .	34
6.1.5.2. Aggregation on Job Level . . . . .	37
6.2. Aggregate Model . . . . .	40
6.2.1. Introduction . . . . .	40
6.2.2. Structure . . . . .	44
6.2.3. Definitions . . . . .	45
6.3. Constraints . . . . .	49
7. INTEGRATION SCENARIOS . . . . .	50

7.1. Integration Between CP and APS . . . . .	50
7.2. Integration Between CP and CP . . . . .	53
7.3. Integration Between APS and APS . . . . .	54
7.4. Integration Between APS and CP . . . . .	57
8. REAL LIFE APPLICATIONS . . . . .	59
8.1. A Pharmaceutical Company Case . . . . .	59
8.2. Trakya Cam Case . . . . .	59
8.3. Brisa Case . . . . .	61
9. CONCLUSIONS . . . . .	62
REFERENCES . . . . .	66

## LIST OF FIGURES

Figure 4.1.	General Model Structure-1 . . . . .	18
Figure 4.2.	General Model Structure-2 . . . . .	19
Figure 4.3.	General Model Structure-3 . . . . .	20
Figure 4.4.	General Model Definitions-1 . . . . .	21
Figure 4.5.	General Model Definitions-2 . . . . .	22
Figure 4.6.	Class Diagram of General Model . . . . .	23
Figure 5.1.	Class Diagram of APS . . . . .	26
Figure 5.2.	Structure of APS Model . . . . .	27
Figure 5.3.	APS Model Definitions-1 . . . . .	27
Figure 5.4.	APS Model Definitions-2 . . . . .	28
Figure 6.1.	Period Object . . . . .	29
Figure 6.2.	Aggregation of time . . . . .	30
Figure 6.3.	Aggregation of part . . . . .	30
Figure 6.4.	Demand-Delivery Objects . . . . .	31
Figure 6.5.	Aggregation of demand . . . . .	32

Figure 6.6.	Aggregation of resources . . . . .	33
Figure 6.7.	Aggregation of resources-2 . . . . .	33
Figure 6.8.	Process Object . . . . .	34
Figure 6.9.	Process Aggregation on Operation Level . . . . .	35
Figure 6.10.	Aggregation Scenario . . . . .	36
Figure 6.11.	Aggregation of operations . . . . .	37
Figure 6.12.	Process Aggregation on Job Level . . . . .	38
Figure 6.13.	Supply chain structure . . . . .	41
Figure 6.14.	Relation between demand, delivery, inventory, and process . . . . .	42
Figure 6.15.	Class Diagram of CP . . . . .	43
Figure 6.16.	CP Structure-1 . . . . .	44
Figure 6.17.	CP Structure-2 . . . . .	45
Figure 6.18.	CP Model Definitions-1 . . . . .	45
Figure 6.19.	CP Model Definitions-2 . . . . .	46
Figure 6.20.	CP Model Definitions-3 . . . . .	47
Figure 6.21.	CP Model Definitions-4 . . . . .	48

Figure 6.22.	Constraints of CP Model . . . . .	49
Figure 7.1.	Integration between top and base level . . . . .	50
Figure 7.2.	Integration of CP and APS . . . . .	51
Figure 7.3.	Case for Integration of CP and APS . . . . .	52
Figure 7.4.	Integration of CP and CP . . . . .	53
Figure 7.5.	Case for Integration of CP-CP . . . . .	54
Figure 7.6.	Integration of APS-APS . . . . .	55
Figure 7.7.	Case for Integration of APS-APS . . . . .	56
Figure 7.8.	Case for Integration of APS-APS-2 . . . . .	56
Figure 7.9.	Integration of APS-CP . . . . .	58
Figure 8.1.	Trakya Cam - Integration of CP-CP . . . . .	60

## 1. INTRODUCTION

In today's complex systems, we can no longer control all the system by one centralized decision making. Since there is diversity of problems in complex systems, these individual problems are solved by separate decision making units rather than one centralized decision making unit. Companies divide their units and each unit has its own separate decision making.

When we look from the systems perspective, these separated units of a company should be in coordination with each other. So we come to a new issue called 'Hierarchies in distributed decision making' which includes coordination of separate decision making units. These separated decision making units talk with each other in order to be coordinated or there exist a higher level power that can get information from these decision making units and can coordinate them. In order to sum up, distributed decision making refers to different decisions being made in different parts of a general system and after that there exists the coordination of these decisions.

In this study, we consider hierarchies in distributed decision making which is 'Hierarchies of planning' as the hierarchy in distributed planning between CP (Capacity Planning) and APS (Advanced Planning and Scheduling) and also between more than one APS and CP. We can see different usages of CP and APS in many companies both from the manufacturing and service sector and those companies face with the need for integration of capacity planning and advanced planning and scheduling.

With the help of capacity planning, companies are able to achieve best using of any type of resources such as manpower, machines and facilities efficiently while considering some constraints due to capacity. Capacity Planning provides decisions on where to produce what and when. It supports decisions on driving acquisition plans for critical raw material with long lead times, mid-term investment decisions based on forecasts on high level product families, handling capacity allocations based on mid and long-term customer orders where accurate product definitions and/or specific shipment

details are not known at the time of decision, where and when additional capacity should be created or capacity should be removed based on market expectations and expected costs and profits.

With the help of Advanced Planning and Scheduling (APS) companies can decide on distribution of jobs according to capacity planning. After capacity planning, companies can know how much capacity to allocate in each period, and advanced planning and scheduling supports the planner's decisions on how to allocate this capacity between jobs. Capacity planning and advanced scheduling are concerned with the acquisition, utilization, and allocation of production resources to best satisfy customer requirements at minimum usage.

There are implementation areas of APS and CP individually and there exist a need for using both of them in coordination with each other. Although there is a need for such an integration, there does not exist a work on how to integrate APS and CP and more than one APS and more than one CP, how to manage data transfers between them, which informations to transfer and how to implement APS and CP integration.

In this study, we consider APS and CP, more than one APS and more than one CP integration in ICRON. APS and CP are decision making units considering different problems and after information transfer between them, their decisions will turn out to be more accurate and more near to actual life situation. In this work, we study the hierarchies in planning between APS and CP.

ICRON is an optimization system developed in  $C^{++}$  to provide Advanced Planning and Scheduling (APS) solutions. Its architecture is based on object orientation. It has a unique environment called Graphical Scheduling Algorithm Modeling System (GSAMS), where user of the system can construct and customize generic algorithms. These algorithms are generated in order to be executed as a solution procedure.

Graphical scheduling algorithm generation process is based on visualization with node and link structures. User does not have to know any coding language and ICRON

requires no experience in software development from the user. ICRON provides an environment for users to develop algorithmic modeling of variety of problems due to its generic system architecture and also modeling of mathematical programming problems.

Although many authors claim that there are ontologies, representing a general model, are applicable in many industries, many of them are only focused on one or two industries for instance only food industry. Some of them are only task based and some of them are constraint based ontologies. And detail of ontologies are not enough to understand and model a planning model. Many of ontologies in literature only describes only class diagrams for modelling a planning system but fields and relations between other classes are not explicitly given in the articles. In this study, we consider APS and CP, more than one APS and more than one CP integration, detailed concepts of this integration like roles of these two components, information sharing details and implementation details. An ontology is proposed which consists of general model, advanced planning and scheduling model and capacity planning model. The basic aim for developing the ontology is generating a general modelling tool which is applicable. With the help of the proposed ontology, it will be more easy to apply aggregation and integration between advanced planning and scheduling model and capacity planning model.

In chapter 2, literature review on hierarchical planning and ontology is given, in chapter three, methodology used in the study is explained, in chapter four, general model is given and in chapter five, continuous time advanced planning and scheduling model is given. In chapter six, aggregate scenarios and aggregate model are studied, in chapter seven, integration scenarios are given and finally in chapter eight, real life applications are given. In chapter nine, conclusion and future studies are given.

## 2. LITERATURE REVIEW

Although hierarchies in distributed decision making is a significant issue since it becomes more difficult to handle today's very complex systems, there are not so many articles and books related to hierarchies in distributed decision making and especially hierarchies in distributed planning. In this section, a brief review of the existing literature is provided.

Schneeweiss and Zimmer (2004) studied hierarchical relation between a producer and a supplier. Operational coordination mechanism between producer and supplier is analyzed where both of them have private local information. Information sharing between a producer and a supplier is important for the overall performance of the supply chain and also for design of a supply contract. So it should be clear enough for all participants to know which information should be transferred and which should not.

Authors analyzed three different types of integration in hierarchies where the top-level represents the producer and the base-level represents the supplier. In the pure top-down hierarchy, the base level features are not taken into account at all. So producer's features are seen as important, and only instruction related to producer is exerted. In the non-reactive anticipation type of integration, only important features of the supplier are taken into account such as issues related to capacity. In the reactive anticipation, information from supplier is fully taken into account as the information related to producer. When results of these three integration types are compared, it is seen that pure top-down hierarchy is the one with the lowest result where the result of non-reactive anticipation is not much better than the pure top-down hierarchy. But reactive anticipation results with an outcome near to the ideal situation. It is concluded that reactive anticipation is logical, since it comes with results near to ideal situation, and it does not need too much information transfer as in the ideal situation.

Schneeweiss (1999) worked on hierarchies in distributed decision making which comprises areas like hierarchical optimization, hierarchical production planning, multi-agent systems, principal agent theory, behavioral cost accounting, hierarchical negotiations and group decision making. The author handled distributed decision making with its all aspects and clarified distributed decision making as not only the problem of separating a system into subsystems but more importantly as the coordination task. Coordination is covered as four different levels which are: data integration, integrating systems through negotiations, integration through planning activities and integration through leadership activities. When we come to types of hierarchies, hierarchies is characterized as constructional and organizational hierarchies where constructional hierarchies is the symmetry of information status which results from imposing a hierarchical structure on a non-structured system and organizational hierarchies is the asymmetry of information status which can be grouped into two as decision time hierarchy with one decision maker and leadership hierarchy with each level having its own decision maker. In hierarchical distributed decision making, there are two levels which are base level and top level where top level can be seen as the leading level and base level as the following level. Author considered hierarchies in distributed decision making in detail and give examples and explained relations between the top and base level by introducing some important notions.

Pocket and Wolsey (2006) considered production planning by mixed integer programming. As our problem is hierarchies in distributed planning which contains capacity planning, production planning by mixed integer programming is a related issue. Authors handled production planning as the planning of acquisition of the resources and raw materials and planning production activities in the most efficient and economical manner. Production planning is the problem of trying to make planning decisions optimizing the trade off between economic objectives and customer satisfaction.

Production planning is rarely seen as mixed integer programming since production planning features such as setup, start up and machine assignment decisions require binary or integer variables which makes problem harder for large scaled problem instances. Authors give model of the production planning problem and used reformu-

lations from the literature. Also they described all the reformulations and examined results of reformulations and techniques. Moreover production planning models like Enterprise Resource Planning (ERP) and Materials Requirement Planning (MRP) are reviewed. Finally authors presented an algorithmic approach to solve production planning problem.

Voss and Woodruff (2002) analyzed tactical production planning and developed models for integrated and optimized planning and scheduling. Optimization models which are used for planning are studied and specific examples and computer implementations are also provided. These models can be used in many planning implementations by making small changes and also they remain valid as the details change over time.

Graves(1982) studied production planning problem which is modeled as a mixed integer linear program. Authors formulated aggregate planning problem as well as detailed scheduling problem as a mixed integer problem. After that formulation, this problem is relaxed by Lagrangian relaxation technique and from this relaxation both lower bounds and upper bounds on the optimal solution value are obtained. Lagrangean function is defined which can be optimized by a subgradient procedure in order to obtain a lower bound and authors proposed a heuristic for converting lagrangean solutions to feasible solution to get upper bound value. They used hybrid approach for production planning in which two subproblems aggregate planning and scheduling are worked and feedback is provided between these two subproblems. These two subsystems working together can be viewed as a hierarchical system.

Federgruen et. al (2007) studied capacitated dynamic lot-sizing models which are joint setup cost and joint and item dependent setup cost models. They worked on a problem where there is a family consisting of items from the same production facility. Demands are known in advance since they are deterministic. Setup cost exists when an order is placed and there exists a constraint on the order size due to capacity limit. In the models, the objective is to find a proper lot-sizing strategy. In the the joint setup cost model which is the basic model, setup cost due to an order does not depend on the order. The setup cost for an order in any given period only depends on the period

object. But in the joint and item-dependent setup cost model, setup costs depend on the order and on the period object like in the joint setup costs. Progressive interval heuristics are generated for these models. After authors developed progressive interval heuristics, they demonstrate that a progressive interval heuristic generates close-to-optimal solutions with modest computational effort, in not only small sized problems, even for large-scale problems.

Ozdamar et. al. (1998) generated a hierarchical decision supporting system for production planning so that handling of complex systems can be done more easily and in a consistent way with interactive sub-systems. The system they proposed is composed of aggregate planning with integration with production schedule. Authors mention that important features of their proposed system is that it enables customer to manipulate planning and scheduling systems and also feasibility of planning levels are also can be manipulated through database manipulations.

The hierarchical system considered in the article is composed of four hierarchies, aggregate planning at the top, family disaggregation, item disaggregation and master production schedule at the lowest level. At each level, information is manipulated by the data management system and each level has different level of detail since higher level is with scarce detail and lower level of hierarchy has higher detail. Lower level data is added cumulatively at higher levels and is disaggregated on the reverse direction. The proposed system is implemented to develop an annual production plan for the manufacturing company.

Hax and Meal (1973) considered hierarchical integration of hierarchical planning and scheduling. They presented a hierarchical planning and scheduling system which is also applied in a factory with more than one plant producing more than one type of product. The hierarchical planning they proposed is a hierarchical system which makes decisions in sequence, in the higher level of the hierarchy decisions is made at an aggregate level which is providing constraints within low level of hierarchy which is a more detailed decision making unit.

The model they proposed is flexible to make corrections at each level in hierarchy, which is significant for satisfying private goals of each hierarchy at an acceptable level and also for coordination among the decisions made at each level. Authors stated that this flexibility of the system is the essential characteristic of hierarchical planning. They modeled hierarchical system and moreover implemented the model on a company.

Tan and Khoshnevis (2000), studied on making a review on articles about integration of planning and scheduling. Hierarchical integration between planning and scheduling is a significant for successful overall system and due to that reason; this subject takes attention of many researchers. With integration between these two hierarchy levels, it has been reported that overall system performance like minimization of makespan and maximization of equipment utilization is increased in many firms.

In many applications of planning and scheduling systems, planning and scheduling systems work independently from each other. Planning is a less detailed system and can not be fed by changes on resource maintenances and other unpredictable events which are subject of scheduling system. Due to that reason, planning system gives results which can not be applicable to the current system. Planning system usually try to find the best way of production under making the assumption that resources are available at all times. On the other hand, generating a schedule is a complex process which leads to not considering alternative plans.

Due to difficulty of planning and scheduling systems working independent from each other, there have been many studies on achieving this integration. Integrations based on five solution methods have been reviewed in the study. These solution methods are concurrent assignment, methods like simulated annealing, knowledge-based planning and scheduling, decision matrix, non-linear process planning and mathematical programming. In those studies, it has been shown that this integration have benefits like improvements in many performance measures. Also obstacles in integration between these two hierarchies is described with application examples.

Ozdamar and Birbil (1999) studied hierarchical production planning with application to a tile manufacturing factory. Basic problem is divided into two sub problems one of which is an aggregate production planning and the other one is a detailed disaggregated problem. In the aggregate production planning, products and capacity are aggregated and in the other sub problem, detailed lot sizing and loading solution is solved. Aggregate planning problem considers the whole planning horizon where period consider in the more detailed problem is a much shorter one.

There are some authors working on the area of constructing ontology for hierarchical planning. In their enlightening study, Smith and Becker (1997), studied constructing a new ontology for scheduling and planning. An ontology can be viewed as a conceptualization of general knowledge into general readable and understandable format. Ontology is constituted by describing a set of classes, their definitions, property definitions and axioms about these classes. The classes, properties and the axioms are related to each other and all these are combined to form a model.

The “ozone ontology” is based on authors experience on planning projects that they preformed. Their study has opened a new way of thinking for later studies. The ontology they constructed in their work is called ozone ontology which is mainly based on scheduling ontology describing and representing scheduling problems and constraints. Ozone ontology is used by further researchers in constructing their scheduling and planning models since ozone ontology constitutes a framework for these models.

In ozone ontology, there are five basic concepts which are demand, activity, resource, product and constraint representing a constraint based scheduling. Scheduling is defined as a process of feasibly synchronizing the use of resources by activities to satisfy demands over time. In this ontology, demand is a request for one or more products. A product is a good or service that is required by demand. For production of a product, there are a set of resource requirements which are the activities. Resource usages and execution of activities is restricted by a set of constraints which can be resource capacity, restrictions due to product characteristics, technological restrictions of resources and due date of demand .

Generally scheduling ontology available in the literature may be classified into two as ontology which is based on constraints specification and as ontology which is based on task specification. Ozone ontology is in the group based on constraints specification. The Ozone ontology constitutes a constraint based solution framework and scheduling system architecture.

The ozone ontology mainly has a scheduling perspective and some perspectives that are needed for a complete model are absent. These issues include goal representation and expansion, activity network synthesis, agency and ownership.

In ozone ontology, relationships between classes are not represented explicitly since describing those relations means defining extra terminology in ontology. But the relationships are defined as properties inside classes. So, if there is a relationship between two classes, then this relationship is represented as property of these two classes so that they can know that they are related to each other.

The article by Motta et al. (2002) in which they studied an ontology which can be viewed as an ontology that is included in the group of task ontologies rather than constraint ontologies. Their task ontology is not for a special application domain or a special problem. Their task ontology is application, method and also domain independent which gives a freedom to apply this ontology on various application domains.

Authors conclude that their task ontology has two main benefits. These benefits are: since ontology is domain independent, it provides a generic model and it has applicability to different domains with necessary domain and application knowledge. This property of the ontology to be applied in different application areas gives an applicability freedom without need to any particular domain specific problem solving approach. Also the task ontology provides a formal description to scheduling systems which can be used for understanding various scheduling problems.

The task ontology proposed in their article is mainly based on two ontologies existing in the literature which are called Base Ontology and Simple Time Ontology.

Base Ontology is an ontology which describes basic elements of a model like resource, task and relations between those basic concepts. The other ontology that the task ontology refers to is simple time ontology which has descriptions related to time concept like time interval, time point and time range and also time relations.

For validation of the benefit that authors claim about the applicability of their ontology in different application domains, they tested the generality of the ontology on two different scheduling applications. First one is the domain of satellite scheduling and the other one is a real-life resource allocation problem in the context of a large research project. As a result of this work, they see that their ontology is good at knowledge acquisition and modeling. Also problem specifications from different domains are fully characterized with the ontology.

For making comparison between existing ontologies and their proposed ontology, they compare the ontology with a task ontology called Multis and constraint based ontology the Ozone ontology. In the other task ontology, detailed description about tasks is not given in detail like it is in their task ontology and also the proposed task ontology provides a more detailed framework for representing task knowledge for scheduling problems by distinguishing between constraints, requirements, preferences and cost function.

Also there are some descriptions that are missing in the other task ontology like resource capacity which is an important lack since it is needed for avoiding cases of overlapping jobs. Moreover, some attributes of classes are not described like capacity of a resource class which is a significant attribute.

In comparison between ozone ontology and this task ontology, ozone ontology doesn't consider cost and preference issues so it is not possible with ozone ontology to compare different preferences. Since ozone ontology is a constraint based scheduling ontology, it includes descriptions related to constraint based problems. In ozone ontology, type of problem-solving approaches that can be used to solve problem are assumed, on the other hand in this task ontology, there is not a distinction between

application types.

Since the task ontology does not belong to any specific problem solving approach it is more generic than other ontologies existing in the literature and it provides an ontological foundation for many classes of scheduling problems. Also the task ontology is appropriate for specialization for different classes of applications.

Frankovic et. al. (2002) looked ontology from a more general perspective since they did not try to generate an ontology for a special domain but they worked for presenting general problems and methods of creation ontologies for different domains. Due to this reason, their article constitutes a basis for researchers working on ontologies.

Authors described the basic steps in constructing an ontology and with the help of their study, they aim new ontologies to be formed in a more generic way which can be applicable in more specific and generic systems. Moreover, their concept is also usable in integration of more than one ontologies. In their article, authors mention that their concept has applicability to all classes of scheduling problems.

For understanding relationships and integration between more than one sub systems, it is important to note study of Merdan et. al. (2007) since they worked on multi-agent assembly domain and they generate an ontology for their integration. Each agent acts as an individual system like it is capacity planning and advanced planning and scheduling in our case. Since agents communicate and integrate with each other in order to perform operations of the parent system, there is a need for a general ontology for their integration. Rajpathak et. al. (2001) studied on generating an ontology which is a task based ontology in their article. They presented an ontology that they refer as task ontology for scheduling. Authors emphasized that although there are some ontological studies on task ontology, these ontologies don't include cost related issues and also some of them are only addressing special domains rather than being generic. Moreover, preference related issues are also lacking in those ontologies.

Their task ontology is based on definitions containing job class, job type class, resource class, resource type class and also cost function and preference classes. They presented the properties and relations between those classes. There are some distinctions between that ontology and other ontologies existing in the literature.

The task ontology presented in that article is different from ozone ontology since basic issues are involved in the task ontology where other environmental issues like demand and product take place in ozone ontology. Also the task ontology considers cost related issues and also preference issues which are important in scheduling systems whereas ozone ontology doesn't include those items.

Another ontology that the task ontology is compared with is CommonKADS ontology. In that ontology, cost related issues are missing also definitions related to classes are also inadequate. In the task ontology, properties of classes are given in detail whereas CommonKADS ontology, main concepts are given in less detail and necessary descriptions are not presented with enough detail. Job Assignment ontology is another ontology that the task ontology is compared with. In this comparison, important things to mention are that job assignment ontology has less detail and some descriptions are missing where task ontology is a more detailed ontology. Also cost and preference related items are also missing elements in job assignment ontology.

An example to researchers who proceeded from Smith's Ozone ontology is Houba et. al. (2000) since they worked on generating a modeling method for food industry based on ozone ontology. In addition to ozone ontology which contains concepts like activity, resource and demand, the authors presented a new concept. Since each product has some ingredients and also process steps, they added a recipe class in their ontology which is composed of activities, durations of the activities, resource and also sequence constraints.

By proposing the recipe notion, authors aim that usability of scheduling in food industry increases, since recipe concept is a domain specific terminology. There are some specifications related to recipe concept: a recipe is defined for only one product

but also a product can be produced by more than one recipes. So there is a one to many relationship between product and recipe classes.

Gil and Blythe (1999) proposed a new ontology for representation of plans which they called Planet. There are some works on integrating planning tools which are costly since many of them has separate tools for making interchange between two planning tool and also they need some other additions. With the help of planet ontology, a common plan representation for systems will result with more easy interaction between systems by easily exchanging information about plan.

Different ontologies are developed for different planning systems and Planet ontology is developed for integrating those different ontologies. Planet ontology is developed as a combination of Ozone ontology, time relations, and also improvements done by authors containing plan evaluations. In order to test applicability of their ontology, Planet ontology is applied successfully on three different domains. Mönch and Stehli (2003) described an ontology for production control system in semiconductor manufacturing. The system they considered is a complex one since it is composed of sub-systems which work independently on their own and communicate with each other in order to form the main system. For making the interaction between these sub systems more easily and also easy control of the whole system, there is a need for generality of conceptualization between these subsystems which turns out to be the Planet ontology that the authors proposed.

Production control system in semiconductor manufacturing is an example to complex systems. The considered manufacturing system consists of different work areas each of which includes several work centers. The ontology proposed consists of concepts like resource, activity, tool, order, product, event, plan and schedule. Basic concepts of the ontology are mainly the concepts of the Enterprise Ontology and the Ozone ontology. But for applicability of the ontology to the domain, they also add several other concepts related with the domain and also the hierarchical production control approach.

Although there exist ontologies in the literature, application of these ontologies is limited. It is hard to find real life application examples to the ontologies. But in their study, Smith et. al. (1996) gave an example project to the application of ozone ontology. They worked on fastening the construction and introduction of planning and scheduling systems in complex application domains and also they mention some obstacles they see in application of those ontologies on real life. For overcoming these obstacles, they suggested some solutions.

### 3. METHODOLOGY

For the design and implementation of this study, object oriented methodology, object oriented mathematical programming and Graphical Scheduling Algorithm Modeling System (GSAMS) module of ICRON software will be used. ICRON is an optimization system developed in  $C^{++}$  to provide Advanced Planning and Scheduling (APS) and Capacity Planning (CP) solutions. Object oriented mathematical programming has the main idea of making mathematical model abstractions based on real life objects. In object oriented systems we can handle data in a structured way since objects contain the actual data to be manipulated in the optimization system. So we can divide the complex data into these objects which are corresponding to real life entities. All objects are represented by class definitions. The class definition with attributes and functions provides an abstract representation of a conceptual or real object. A class definition can be done based on other class definitions with gathering the content of the previously defined classes.

ICRON is based on object oriented data models which involves classes and their relationships. In ICRON, user can make any class definition having attributes that may also refer to other objects in the system, graphically by interacting via GUI (Graphical User Interface). In GSAMS user can construct algorithms based on these class definitions which are methods of the associated class in order to model the system and execute as a solution procedure.

GSAMS is an environment designed for modeling algorithms by using abstractions such as nodes and links. Algorithms modeled in ICRON are logical sequences of processes performed on the input data in order to reach the result. Nodes and links exist in algorithms where nodes represent data processing and links represent the object flow between the nodes. Algorithms generated in ICRON can be used to manipulate objects of the related classes, and user does not have to have any programming knowledge. Algorithms can be run in order to produce solution of the problem of concern in an appropriate format, and it is easy for the user to debug the algorithm for correcting

errors. Reports and charts can be given as examples of outputs of these algorithms.

In ICRON mathematical programming module, user can formulate mathematical programming with mathematical programming nodes, which are the basic constructs of object oriented mathematical programming. User interactively formulates the optimization problem in GSAMS environment with mathematical programming nodes and with other related nodes for algorithmic manipulations, if necessary. Mathematical programming is completely based on object oriented concepts like algorithm generation in APS (Advanced Planning and Scheduling).

ICRON user is supposed to formulate expressions for objective function and constraints and give the decision variables of the mathematical program. ICRON currently supports linear programming, integer programming and mixed integer programming due to the capabilities of the solvers. ICRON has accessibility to different solvers. The system has four embedded solvers for linear programming, integer programming and mixed integer programming which are LP\_Solve, GNU\_LPK, COIN and CPLEX.

## 4. GENERAL MODEL

In this study, an ontology is proposed for hierarchical detailed scheduling and planning systems. Proposed ontology consists of items which are a general model, bucket based capacity planning model and continuous time advanced planning and scheduling model. Bucket based capacity planning model defines only objects and relations for constructing capacity planning model. Advanced planning and scheduling model defines only objects and relations for constructing advanced planning and scheduling.

### 4.1. Structure

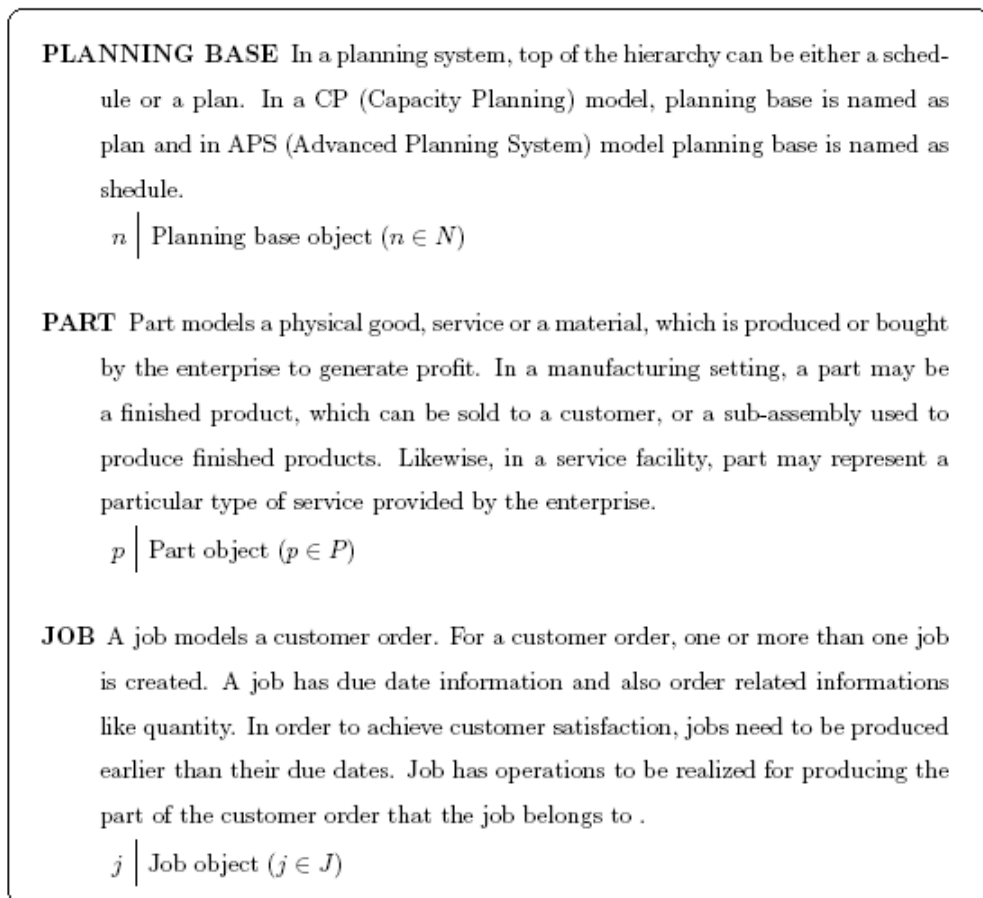


Figure 4.1. General Model Structure-1

**RESOURCE** A resource is required to carry out the jobs. A resource models a resource which can be renewed for each time period. Examples to a resource can be machines, or human since we can use a machine again on another operation, after it is used by an operation. An operation acquires some amount from capacity of the resource and after operation is finished, it releases the capacity that it takes from the resource. So, resource can be renewed after it is being used by an operation.

$r$  | Resource object ( $r \in R$ )

**RESOURCE GROUP** Resources can be grouped according to any logical criteria. A resource group models a group which is formed of resources. In most of the cases a resource group may contain identical resources and also a resource group may form a hierarchy between resources.

$rg$  | Resource group object ( $rg \in RG$ )

**ROUTING** Routing is a detailed description of different phases of production of a part. Each phase of production is an operation. A routing defines a specific set of operations that must be performed to produce a particular part. There may be many feasible routings to produce a particular part.

$rt$  | Routing object ( $rt \in RT$ )

**OPERATION** In order to become a finished product, a part undergoes certain operations that are described in its routing. When we have a job to produce a part, each production process that it should undergo are modeled as an operation.

$o$  | Operation object ( $o \in O$ )

**SPLIT** When an operation is divided, an operation can be thought of the undivided operation that is the parent of the split brothers. So; for example when an operation is splitted into three split operations, then there exists a split consisting of these three split operations. A split holds a list of split operations which are split from a common parent operation. Also each operation has a split operation

Figure 4.2. General Model Structure-2

even if it is not split. The split operation list of the split of an unsplit operation contains only one operation (i.e., the operation itself).

$m$  | Split object ( $m \in M$ )

**ALTERNATIVE** Each different way of realization of an operation is modeled as an alternative for that operation. Due to this reason, an alternative can model a choice to produce an operation on a resource. For instance; operation can be produced using the resource on that alternative. Alternative keeps the list of resource requirements and list of part requirements specific to that alternative.

$al$  | Alternative object ( $al \in AL$ )

**INVENTORY** Inventories are the places where both finished and semi finished parts are stored.

$i$  | Inventory object  $i$

**PATH** Parts are moved between inventories through defined paths. Path definitions are input to the model.

$h$  | Path object ( $h \in H$ )

**ACQUISITION** An acquisition object models a procurement process of a part. Acquisition object holds information of time when procurement is going to occur and number of units that is going to be received. A part may have more than one acquisitions.

$aq$  | Acquisition object ( $aq \in AQ$ )

**RESOURCE REQUIREMENT** Requirement information regarding the usage of a resource to accomplish an alternative is modeled by the resource requirement.

$rr$  | Resource requirement object ( $rr \in RR$ )

**PART REQUIREMENT** Requirement information regarding the part is modeled by the part requirement. Part requirement is related to the usage of a part to accomplish an alternative.

$pr$  | Part requirement object ( $pr \in PR$ )

Figure 4.3. General Model Structure-3

## 4.2. Definitions

<b>PLANNING BASE</b>	
$P$	Set o parts of schedule $n$
$J$	Set of jobs of schedule $n$
$R$	Set of renewable resources of schedule $n$
$P_t$	Set o parts of plan $n$
$D_t$	Set of demands of plan $n$
$RG_t$	Set of renewable resource groups of plan $n$
$A_t$	Set of processes of plan $n$
$H_t$	Set of paths of plan $n$
$I_t$	Set of inventories of plan $n$
$T$	Set of periods of plan $n$
<b>JOB</b>	
$d$	Due date of job $j$
$b$	Batch size of job $j$
$r$	Release time of job $j$
$p$	Part of job $j$
$rt$	Routing of job $j$
$O$	Operations of job $j$
$M$	Set of splits of job $j$
<b>SPLIT</b>	
$O$	Operations of split $m$
<b>ROUTING</b>	
$l$	Lead time of routing $rt$
$i$	Inventory of routing $rt$
$P$	Parts of routing $rt$
$J$	Jobs of routing $rt$
$O$	Operations of routing $rt$

Figure 4.4. General Model Definitions-1

In order to define relations between objects in general model proposed in this study, class diagram is used to draw the big picture of the architecture and class diagram also provides a formal basis for understanding general model. This UML diagram can be used for mapping the relations of objects in APS. Class diagram of general model is shown in figure 4.6.

<b>ACQUISITION</b>	
$t$	Acquisition time of acquisition $aq$
$q$	Quantity of acquisition $aq$
<b>PART</b>	
$p$	$p$ is a purchased part
	If $\sum_{r \in p.R} r.O = \phi$ otw. $p$ is a manufactured part
$z$	Onhand inventory of part $p$
	$\sum_{i \in p.I} i.q = p.z$
$J$	Set of jobs of part $p$
$R$	Set of routings of part $p$
$I$	Set of inventories of part $p$
$W$	Set of acquisitions of purchased part $p$
	$\forall a \in W, a.t = j.d, a.q = j.b$
	where $j \in p.J$ and $j.O = \phi$
<b>INVENTORY</b>	
$p$	Part of inventory $i$
$RT$	Routings of inventory $i$
<b>OPERATION</b>	
$r$	Split ratio of operation $o$
	$\sum_{o \in o.m.O} o.r = 1$
$j$	Job of operation $o$
$m$	Split of operation $o$
$AL$	Alternatives of operation $o$
$P$	Predecessor operations of operation $o$
$S$	Successor operations of operation $o$
<b>ALTERNATIVE</b>	
$ot$	Operation time per unit of alternative $al$
$o$	Operation of alternative $al$
$S$	Operation schedules of alternative $al$
$P$	Part requirements of alternative $al$
$R$	Resource requirements of alternative $al$

Figure 4.5. General Model Definitions-2



## 5. ADVANCED PLANNING SCHEDULING MODEL

### 5.1. Introduction

Advanced planning and scheduling system is more advanced than other scheduling models since APS simultaneously makes schedules based on availability of parts and resource capacity. Advanced planning and scheduling aims at planning and controlling execution of the operations.

With the help of APS, operation is scheduled on which resource and start and end times of the operation are obtained. While making a schedule generation, continuous time advanced planning and scheduling model takes into consideration of capacity of resources and availability of part requirements needed for the operations.

Continuous time advanced planning and scheduling model is based a number of key concepts like part, job, operation and alternative. A customer order is mainly modeled as a job  $j \in J$  and for satisfying the customer order, jobs have to be fulfilled. Customer order also has the part information in it where a part  $p \in P$  is an output of the company for generating profit which can be a physical good or service.

For modelling customer orders in scheduling concept, jobs are created. A job can have more than one sub works inside it where each of these sub works corresponds to an operation  $o \in O$ . An operation is a small section of a job and a job is a combination of operations. This combination of operations which is a sequence of operations that are needed for producing a part is modeled by the routing object  $rt \in RT$ . For finishing a job, all operations inside that job should be realized. In advanced planning and scheduling, we try to schedule those operations for meeting customer demand with giving care to some limitations like resource capacity and part requirements.

An operation can be processed by using different renewable resources. Renewable resource  $r \in R$  is a physical asset of a company which is required to carry out the jobs.

Machines can be examples to renewable resource since they can be renewed after they are utilized during operations.

Different ways to realize an operation is modeled by an alternative  $alt \in Alt$ . An alternative contains related information about processing an operation on a resource like resource information and part requirement information. Since different resources can be different from each other, they have different processing times for an operation.

In advanced planning and scheduling, operation is scheduled on each of its alternatives and best one is selected among these. Each of these is an operation schedule object  $s \in S$ . By an operation schedule, we can get information on the alternative an operation is scheduled, start and finish times of the operation, setup times and also resource and part requirements.

In order to define relations between objects in APS, class diagram is used to draw the big picture of the architecture and class diagram also provides a formal basis for understanding model of APS. This UML diagram can be used for mapping the relations of objects in APS.

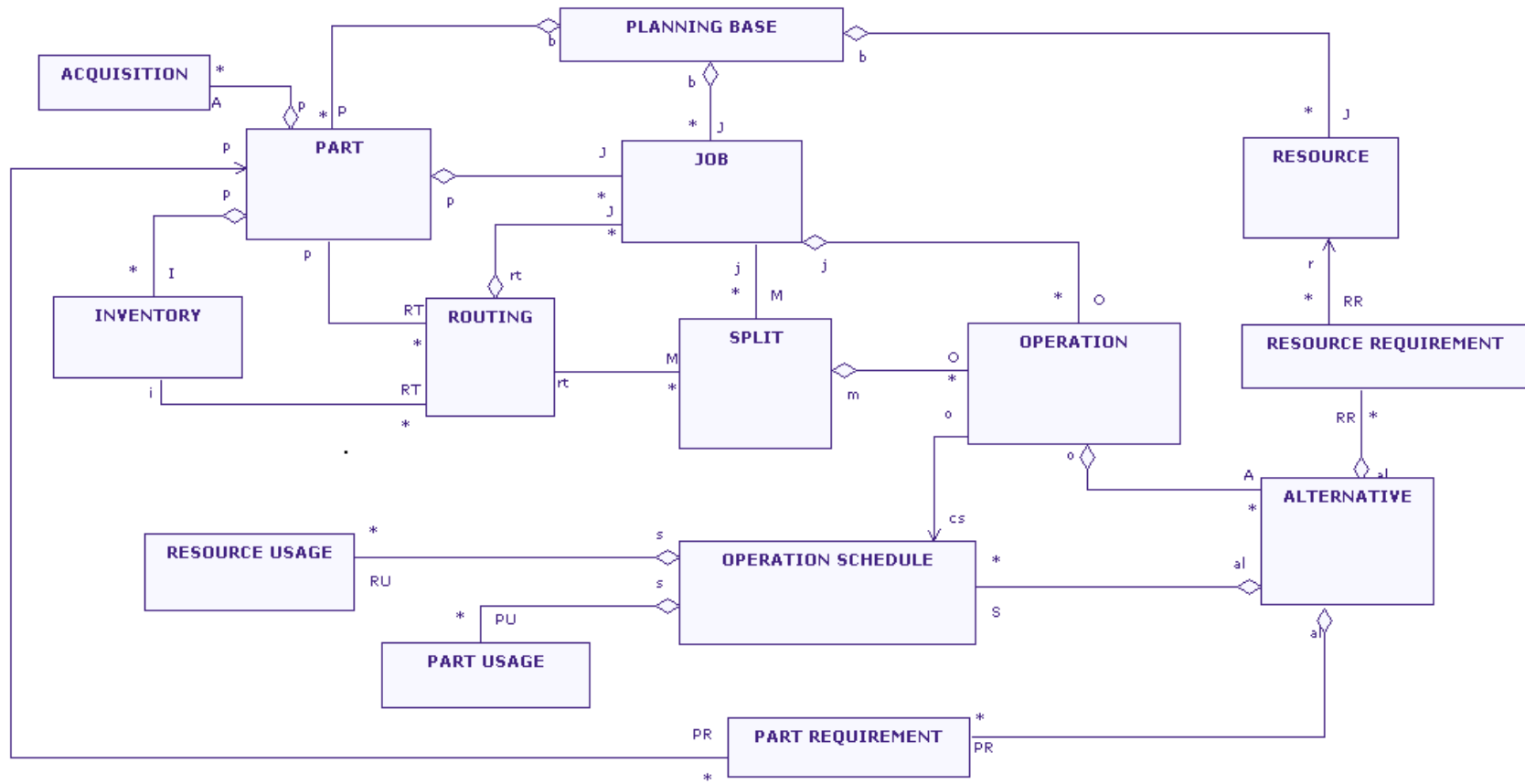


Figure 5.1. Class Diagram of APS

## 5.2. Structure

**OPERATION SCHEDULE** An operation schedule models the schedule information which includes start and completion times and the scheduled alternative for an operation. Since an operation can be scheduled at different places on the schedule, an operation can have more than one operation schedules. An operation schedule is like a template which is gathering information about which resource an operation is scheduled and setup time required before an operation and start and completion times of the operation. There may be many operation schedules created for an operation and each one is reflecting various concerns and performance expectations. So, a selection can be made between operation schedules using different performance evaluations. For instance, if the objective is minimization of setup, the operation schedule with the minimum setup time is selected or if there exists an aim with minimum completion time, then the operation schedule with minimum duration is selected. When an operation is scheduled, it means that schedule of the operation is fixed and all scheduling related information of the operation is set to the informations belonging to the selected operation schedule.

$s$  | Operation schedule object ( $s \in S$ )

Figure 5.2. Structure of APS Model

## 5.3. Definitions

### OPERATION

$cs$  | Current schedule of operation  $o$

### ALTERNATIVE

$S$  | Operation schedules of alternative  $al$

### RESOURCE

$k$  | Available capacity of resource  $r$

$e$  | Efficiency of resource  $r$

$w$  | Working pattern of resource  $r$

Figure 5.3. APS Model Definitions-1

$w.f_1(t_1, t_2)$	Working time difference function of working pattern $w$ (Working time difference function finds working time as number of seconds between $t_1$ and $t_2$ )
$w.f_2(t, n)$	Add time with shift function of working pattern $w$ (Add time with shift function calculates ending time by using starting time $t$ and $n$ working time seconds, regarding shift assignments of resource)
<b>OPERATION SCHEDULE</b>	
$st$	Start time of operation schedule $s$
$ct$	Completion time of operation schedule $s$
$ot$	Operation time of operation schedule $s$
$sst$	Setup start time of operation schedule $s$
$sct$	Setup completion time of operation schedule $s$
$set$	Setup time of operation schedule $s$
$al$	Alternative of operation schedule $s$
$RU$	Resource usages of operation schedule $s$
$PU$	Part usages of operation schedule $s$
<b>PART REQUIREMENT</b>	
$n$	Number of units required for part requirement $pr$
$a$	Alternative of part requirement $pr$
$p$	Part of part requirement $pr$
	where $\sum_{r \in p.R} r.O = \phi$
<b>RESOURCE REQUIREMENT</b>	
$q$	Required capacity of resource requirement $rr$
$a$	Alternative of resource requirement $rr$
$r$	Resource of resource requirement $rr$
<b>PART USAGE</b>	
$y$	Part requirement of part usage $pu$
$s$	Operation schedule of part usage $pu$
<b>RESOURCE USAGE</b>	
$r$	Resource requirement of resource usage $ru$
$s$	Operation schedule of resource usage $ru$

Figure 5.4. APS Model Definitions-2

## 6. AGGREGATION SCENARIOS AND AGGREGATE MODEL

### 6.1. Aggregation Scenarios

In this section, aggregation scenarios and related objects for modelling an aggregate model are defined. Typical aggregation scenarios are time, part, demand, resource and process aggregations.

#### 6.1.1. Time Aggregation

Time is represented as time-buckets. That is, time is not a continuous variable, but it is a collection of constant sized time periods (buckets), such as day, week, month, etc. In general model proposed in this study, time buckets are defined as periods. PERIOD models a time period with start and end dates.

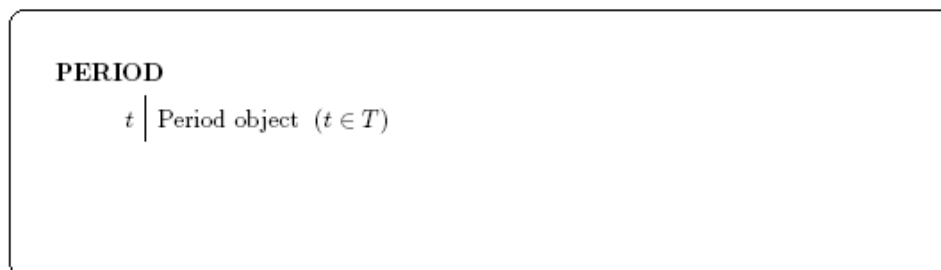


Figure 6.1. Period Object

Planning is dependent on time and time unit used by the planner. Due to this reason, planning time buckets are important. So, time aggregation is a significant aggregation type for hierarchical planning.

Depending on environment of production, market conditions and corporate culture, time buckets can be defined as day, week, month or year. In general model proposed in this study, in order to change periods i.e. week to month, only start date and end date of period have to be changed.

Period : day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Period : week	Week 1			Week 2			Week 3			.....						Week 1			Week 2			Week 3			.....																																			
Period : month	MONTH 1															MONTH 2																																												

Figure 6.2. Aggregation of time

### 6.1.2. Part Aggregation

Parts can be aggregated according to a feature which is common enough to aggregate.

For instance;

Let  $p_1$ ,  $p_2$  and  $p_3$  be parts and in order to produce these parts,  $p_{10}$  and  $p_{20}$  are required.

If critical part for  $p_1$ ,  $p_2$  and  $p_3$  is  $p_{10}$  and main concern of planning jobs of  $p_1$ ,  $p_2$  and  $p_3$  is determining requirements and fulfilling requirements.

Then parts  $p_1$ ,  $p_2$  and  $p_3$  can be grouped to an aggregate part  $pg$  and  $p_{10}$  is the only required part for producing  $pg$ .

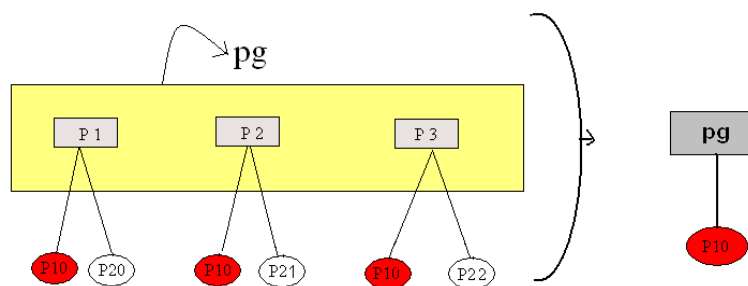


Figure 6.3. Aggregation of part

So, for this case, parts requiring same part are grouped. There are possible cases for aggregation of parts depending on the production environment characteristics, market conditions, etc. For some companies, there is no information of parts in detail, only information of part groups are at hand. And all information in company (i.e. forecast, sales, etc.) is about part groups. Then for these companies also aggregation of part is unavoidable.

In ontology proposed in this study, both part and part group are represented as a part. So, an aggregated part namely part group represent also a PART.

### 6.1.3. Demand Aggregation

A JOB models a customer order for a part. A JOB has due date information and also order related information like quantity, customer, shipment location, etc. In order to achieve customer satisfaction, jobs need to be produced earlier than their due dates.

Based on information of job, demand aggregations can be defined. Jobs can be aggregated based on part, customer, shipment location and period of due date, etc. Simplest aggregation based on DEMAND is combining jobs in a time period for same part. Due to time aggregation, part aggregation or both, demand aggregation occurs.

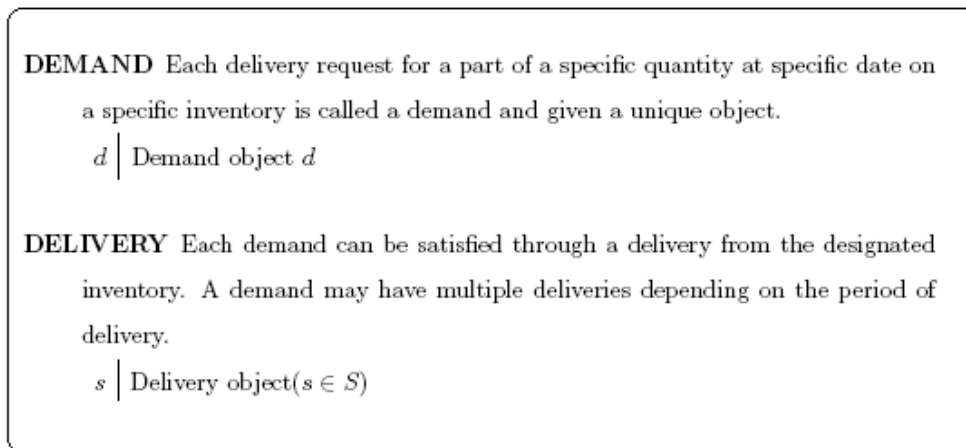


Figure 6.4. Demand-Delivery Objects

Depending on availability of information, demand aggregation is a necessity. Because creating info or gathering info may be not so easy. Sometimes, only demand information is forecasts. So, reason of using forecast as aggregate demands is lack of information of jobs. Also according to engineering choice, demand aggregations may be defined in order to simplify model.

For instance;

Let  $j1$  be job for part  $p$  and batchsize of job  $j1$  is  $j1.b$

Let  $j2$  be job for part  $p$  and batchsize of job  $j2$  is  $j2.b$

If due dates of jobs  $j1$  and  $j2$  are between start time and end time of the period  $t$

$(t.b < j1.d < t.f \text{ and } t.b < j2.d < t.f)$

Then quantity of demand  $d_1$  is summation of batchsizes of jobs  $j1$  and  $j2$

$(d_1.q = j1.b + j2.b)$

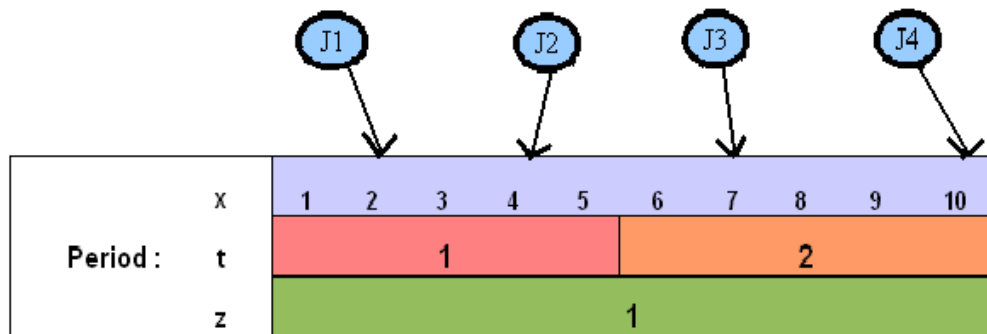


Figure 6.5. Aggregation of demand

#### 6.1.4. Resource Aggregation

A resource group is an aggregation of resources. When resources are similar to each other (for ex. parallel identical machines), which have common properties, they can be grouped into a resource group and has a capacity which is summation of resource capacities inside that particular resource group. Resources which are not similar (for ex. machines which are in series in a production line) also can be grouped and capacity of resource group can defined as capacity of resource(s) which is classified as bottleneck resource(s) in the group.

In the ontology proposed in this study, aggregation of resources are represented as resource groups. And a single RESOURCE can also be modeled as a RESOURCE GROUP aggregating only itself.

For instance;

Let cell 1 contains resources  $R1$  and  $R2$

Let  $R1$  and  $R2$  be similar resources

Let cell 2 contains resources  $R3$  and  $R4$

Let  $R3$  and  $R4$  be different type of resources

Let cell 3 contains resources  $R5$ ,  $R6$  and  $R7$

Let  $R5$ ,  $R6$  and  $R7$  be similar resources

Then there may exist possible resource aggregations. One of the possible scenario is aggregation of resources of cell 3 on a resource group.

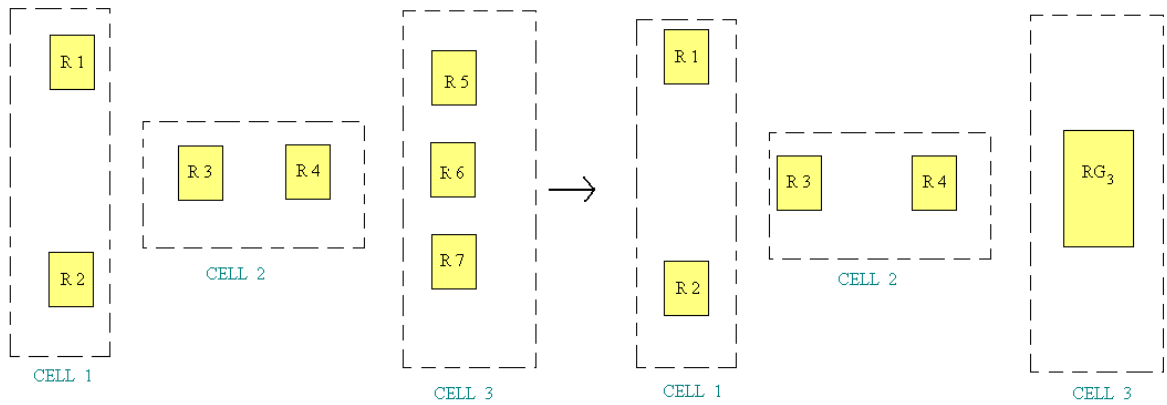


Figure 6.6. Aggregation of resources

Capacity of resource group  $RG_3$  is sum of capacities of  $R5$ ,  $R6$  and  $R7$ .

The other possible scenario is aggregation of all resources in all cells.

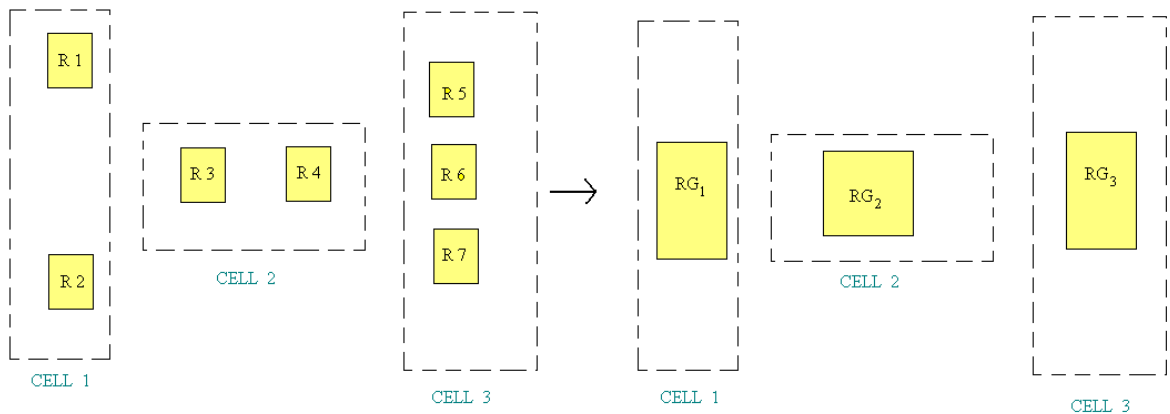


Figure 6.7. Aggregation of resources-2

Since  $R3$  and  $R4$  are different type of resources and  $R4$  is the bottleneck resource. So, capacity of resource group  $RG_2$  is capacity of  $R4$ .

### 6.1.5. Process Aggregation

In a hierarchical planning system, more than one planning models may exist as top and base level. Most of the time, models makes plans using different level of details. Due to that reason, it may be needed to make aggregation of processes belonging to base level when defining processes of top level. Depending on the problem and production environment, different process aggregations can be defined. Aggregating processes provides decrease in detail on top level.

**PROCESS** Parts enter the system through a process which may model either a supply process or production process. Output of a process is a part being introduced into the system and stored in an inventory. Processes (supply or production) have related lead times, which indicates the time it may take to complete the process after its started. We assume that all the parts generated during the process are entered into the output inventory at the completion period of the process. During the process lead time, a number of resources may be utilized in different rates. A production process may also have a defined BOM structure, i.e., other parts (component items or materials) in specified quantities may need to be consumed during the process. Processes may also consume capacity of resource groups. A process may consume its resources in different volumes during the lead time of the process.

$a$  | Process object  $a$

Figure 6.8. Process Object

6.1.5.1. Aggregation on Operation Level. Process aggregation is needed when giving tactical and strategic decisions at OPERATION level. For instance for making a decision to plan an operation on a resource by considering resource capacity constraints, PROCESS aggregation is required.

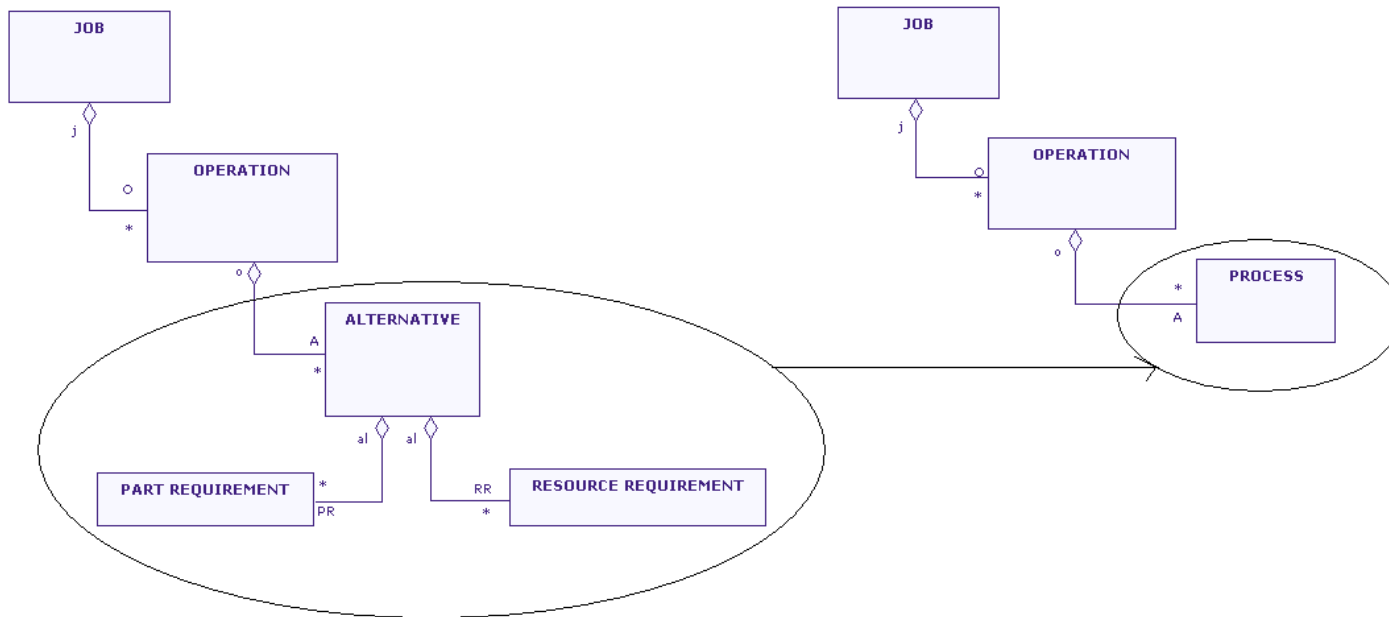


Figure 6.9. Process Aggregation on Operation Level

For instance;

Let  $o$  be an operation with two alternatives such as;  $k, l$ .

$$o.al = \{k, l\}$$

Then there are two processes such as;  $k, l$

$$A = \{k, l\}$$

Let there are three periods such as; 1, 2, 3.

$$T = \{1, 2, 3\}$$

then there are six process per periods such as;  $k_1, k_2, k_3, l_1, l_2, l_3$ .

$$A_t = \{k_1, k_2, k_3, l_1, l_2, l_3\}$$

So, an operation has  $x$  alternatives so it has  $x$  processes. If there exists  $y$  periods then number of process per periods is  $x * y$ .

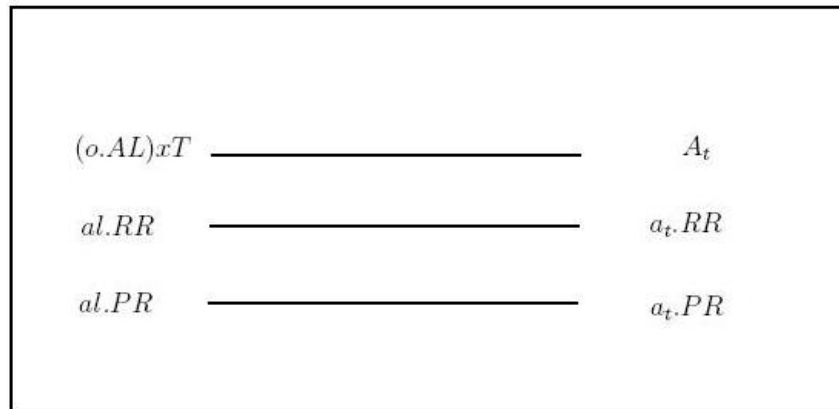


Figure 6.10. Aggregation Scenario

In a production environment, in order to produce a part, there may be many operations to be realized. In hierarchical planning systems, operations can be aggregated according to resource requirements, part requirements, etc. Aggregation of operations can be named as an aggregate operation. Depending on the production environment characteristic, market conditions, etc. resource requirement and part requirement of alternative of aggregate operation can be defined in different ways.

As seen in figure 6.2, Production of a part starts at  $Op1$ .  $Op2$ ,  $Op3$  and  $Op4$ ,  $Op5$ ,  $Op6$  can be processed simultaneously. After completion of both  $Op3$  and  $Op6$ ,

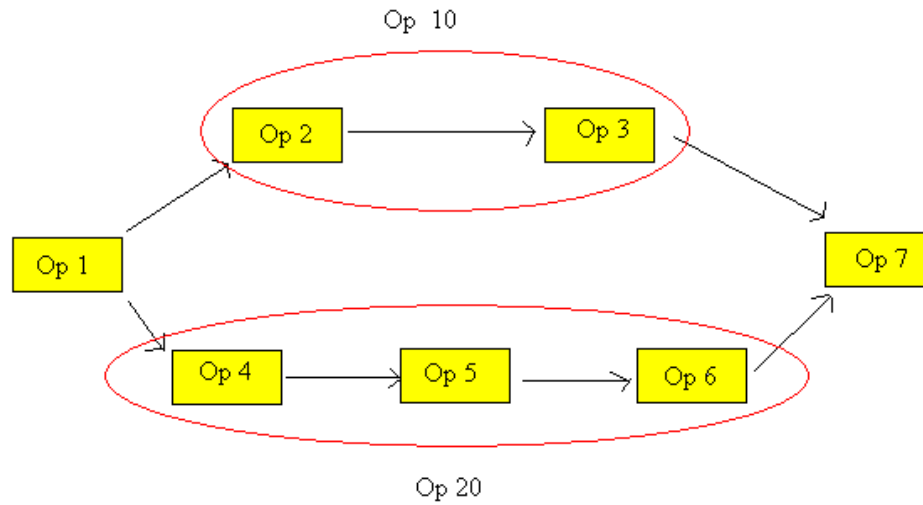


Figure 6.11. Aggregation of operations

$Op7$  can start.  $Op4$ ,  $Op5$ ,  $Op6$  can be aggregated as  $Op20$ . After aggregating operations, depending on the production environment characteristics, market conditions, etc. resource requirements can be aggregated. Resource requirements can be defined as summation of resource requirements of alternatives or summation of resource requirements consisting requirements for bottleneck resources.

Let  $Op20$  be aggregate operation created using  $Op4$ ,  $Op5$  and  $Op6$ .

Then, processes for operation  $Op20$  can be defined as;

$$((Op4).AL) \times ((Op5).AL) \times ((Op6).AL)$$

Then, processes per periods can be defined as;

$$((Op4).AL) \times ((Op5).AL) \times ((Op6).AL) \times T$$

But If resources, required for alternatives of  $Op5$ , are bottleneck resources, then processes for operation  $Op20$  can be defined as;

$$(Op5).AL$$

Then, processes per periods can be defined as;

$$((Op5).AL) \times T$$

**6.1.5.2. Aggregation on Job Level.** OPERATION's ALTERNATIVEs concept becomes process concept. This formation is performed by combining operation alternatives. Each combination of operation alternatives of the same JOB constitute a PROCESS.

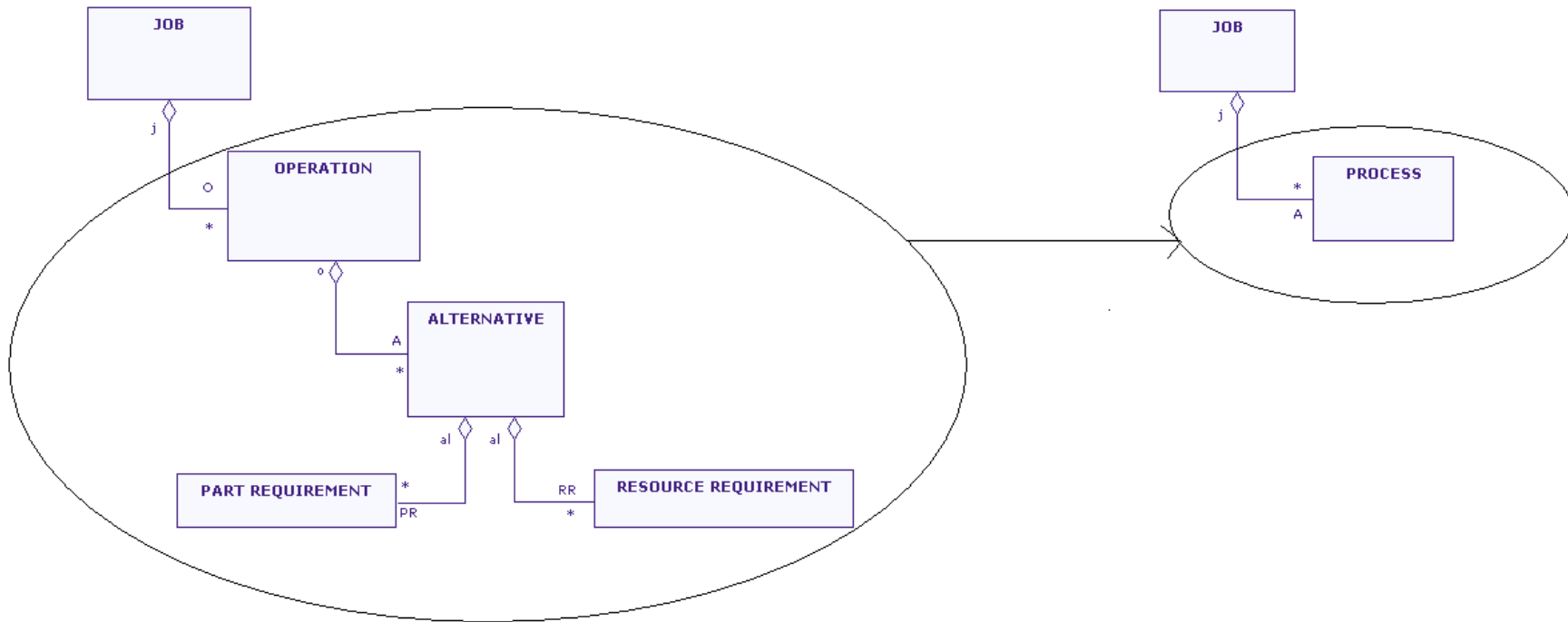


Figure 6.12. Process Aggregation on Job Level

For instance;

Let  $j$  be a job with two operations such as;  $m, n$ .

$$j.O = \{m, n\}$$

Let operation  $m$  has two alternatives such as;  $k, l$ .

$$m.al = \{k, l\}$$

Let operation  $n$  has two alternatives such as;  $b, c$ .

$$n.al = \{b, c\}$$

then there are four processes such as;  $(kb), (kc), (lb), (lc)$

$$A = \{(kb), (kc), (lb), (lc)\}$$

Let there are three periods such as; 1, 2, 3.

$$T = \{1, 2, 3\}$$

then there are twelve process per periods such as;  $(kb)_1, (kc)_1, (lb)_1, (lc)_1, (kb)_2, (kc)_2,$   
 $(lb)_2, (lc)_2, (kb)_3, (kc)_3, (lb)_3, (lc)_3$

$$A_t = \{(kb)_1, (kc)_1, (lb)_1, (lc)_1, (kb)_2, (kc)_2, (lb)_2, (lc)_2, (kb)_3, (kc)_3, (lb)_3, (lc)_3\}$$

So, a job with two operations, where the first operation has  $x$  alternatives and the second one has  $y$  alternatives, then the job has  $x * y$  different processes. If there exists  $z$  periods then number of process per periods is  $x * y * z$ .

Routing is a detailed description of different phases of production of a part. Each phase of production is an operation. A ROUTING defines a specific set of operations that must be performed to produce a particular part. There may be many feasible routings to produce a particular part. A part may have different routings which a part may be produced in different processes. Since routing is an aggregation of OPERATIONS for a part, routing in general model also can be classified as a process aggregation. So, routing can be defined as a PROCESS in aggregate model.

## 6.2. Aggregate Model

### 6.2.1. Introduction

Aggregate model is based a number of key concepts as depicted in Figure 6.13. A *part*  $p_t \in P_t$ , models any item that can be purchased or produced. Parts enter the system through a *process* ( $a_t \in A_t$ ), which may model either a supply process or production process. Eventually, output of a process is a part being introduced into the system and stored in an *inventory* ( $i_t \in I_t$ ). A part stored in an inventory is removed from the inventory either to satisfy a customer order (which is called a *delivery*  $s \in S$ ) or to be shipped to another inventory through a *path* ( $h_t \in H_t$ ).

Processes (supply or production) have related lead times, which indicates the time it may take to complete the process after its started. We assume that all the parts generated during the process are entered into the output inventory at the completion period of the process. During the process lead time, a number of resources may be utilized in different rates. A production process may also have a defined BOM structure, i.e., other parts (component items or materials) in specified quantities may need to be consumed during the process. This drives *dependent demand*  $d_t \in DD_t$  for parts. We assume that component parts are to be delivered fully before the start of the process (completion time of the process - process lead time).

Processes may also consume capacity of resource groups. These are modeled as *resource group usages*  $u_t \in U_t$ , i.e., percentage of the resource usage at a particular period within the lead time. A process may consume its resources in different volumes during the lead time of the process.

Capacity constraints such as machines, production facilities, labor, tools, and suppliers are modeled as (renewable) *resource groups*. A resource group may be viewed as an aggregation of a group of resources. A single resource is also modeled as a resource group aggregating only itself. Resource groups may also have a hierarchy of aggregation. That is, a group can be defined as the aggregation of child resource

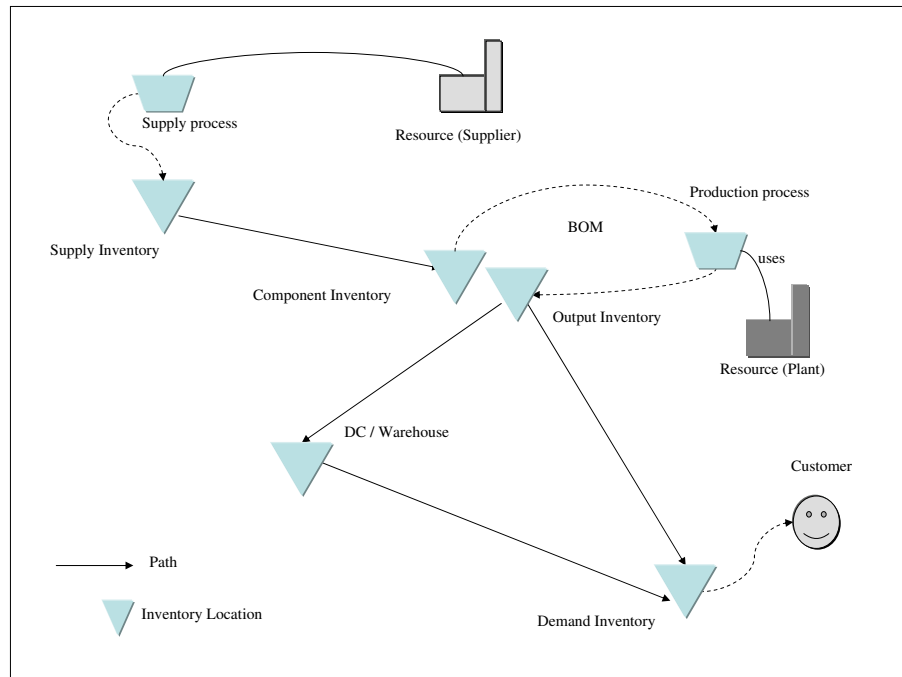


Figure 6.13. Supply chain structure

groups, and a resource group can be a child of many resource groups. In this respect, only resource groups with no children in the hierarchy represent real capacity, whereas other resource groups are abstract groupings.

In the aggregate model, *independent demand*  $d_t \in ID_t$  models either a customer order, or a sales forecast for a part. In either case, demand is manifestation of an independent request for a quantity of part to be delivered to a specific location (inventory) by a specific time.

Relation between, demand, delivery, inventory and process can be viewed in Figure 6.14. Deliveries of independent demands can either be done on the same date as the period of the demand, or later (backorder, tardiness). In either case, the source for the delivery is an inventory. Each demand is expected to indicate a specific delivery inventory. Parts can be input into an inventory by either performing one of the processes (produce or purchase) that stores its output to that inventory, or by moving parts from other inventories. Dependent demands have the same delivery structure as independent demands and treated similarly. The major difference between a dependent

and independent demand is that, delivering dependent demand after the demand date need not be considered.

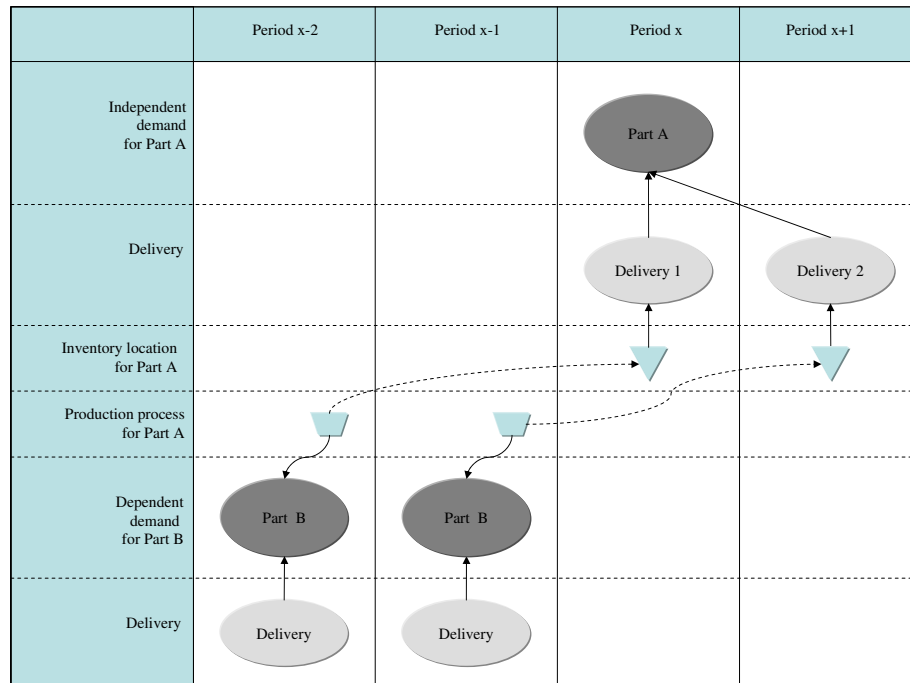


Figure 6.14. Relation between demand, delivery, inventory, and process

In order to define relations between objects in CP, class diagram is used to draw the big picture of the architecture and class diagram also provides a formal basis for understanding model of CP. This UML diagram can be used for mapping the relations of objects in CP.

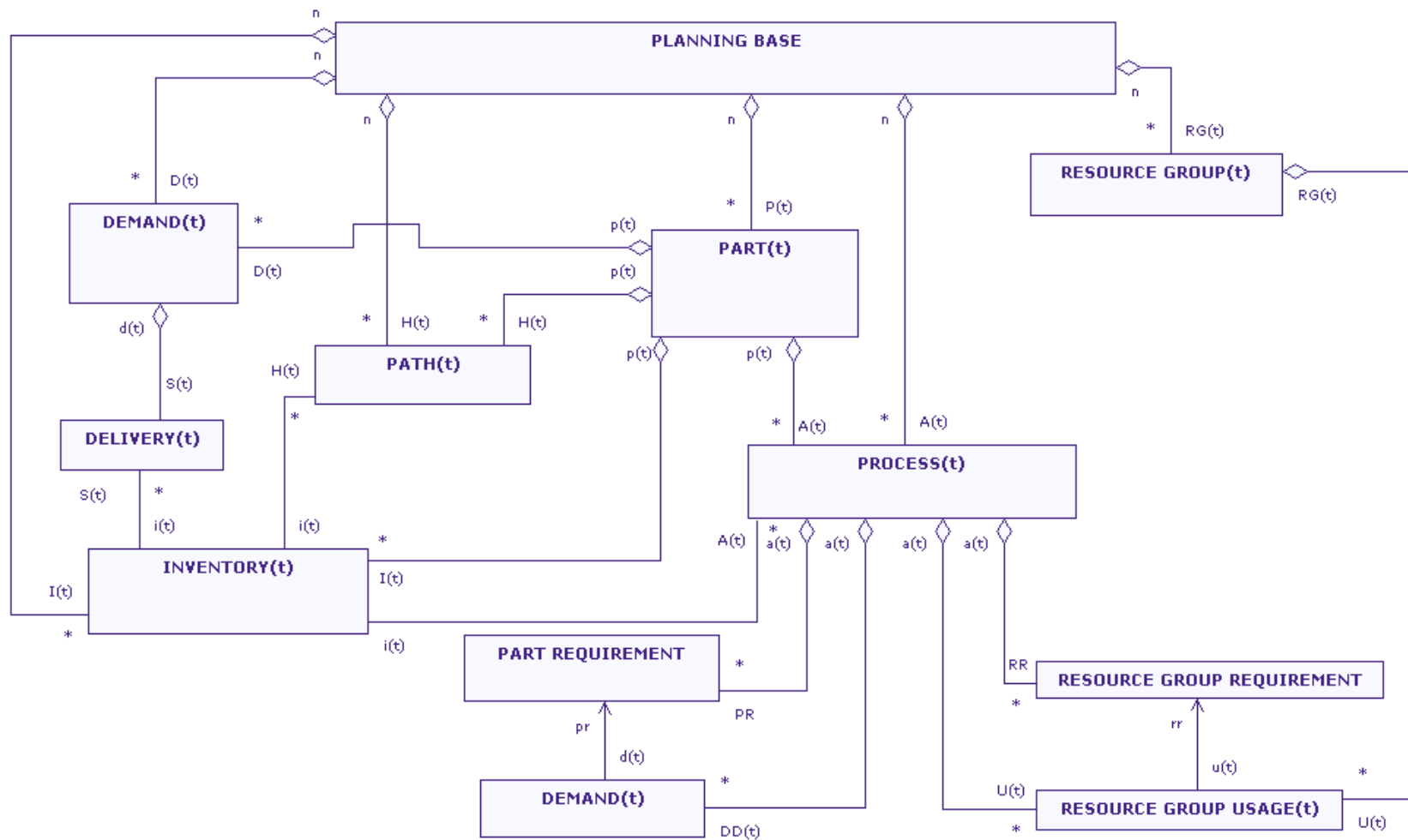


Figure 6.15. Class Diagram of CP

### 6.2.2. Structure

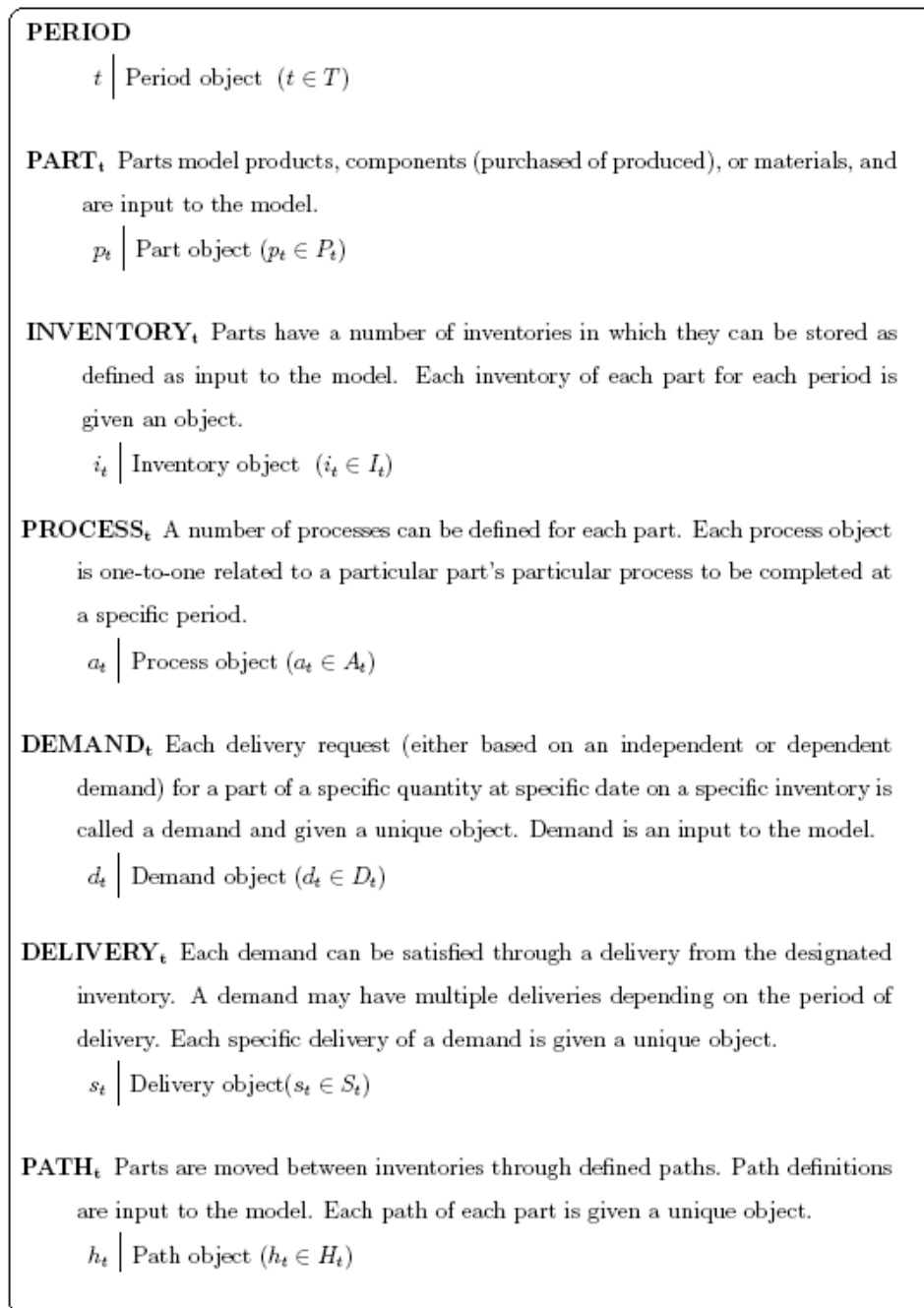


Figure 6.16. CP Structure-1

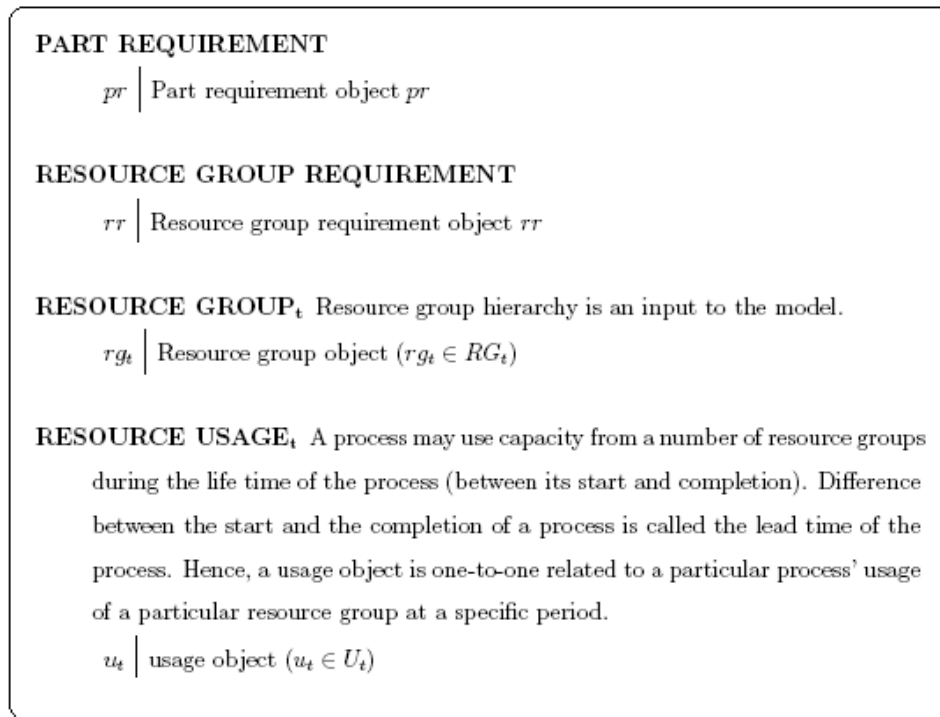


Figure 6.17. CP Structure-2

### 6.2.3. Definitions

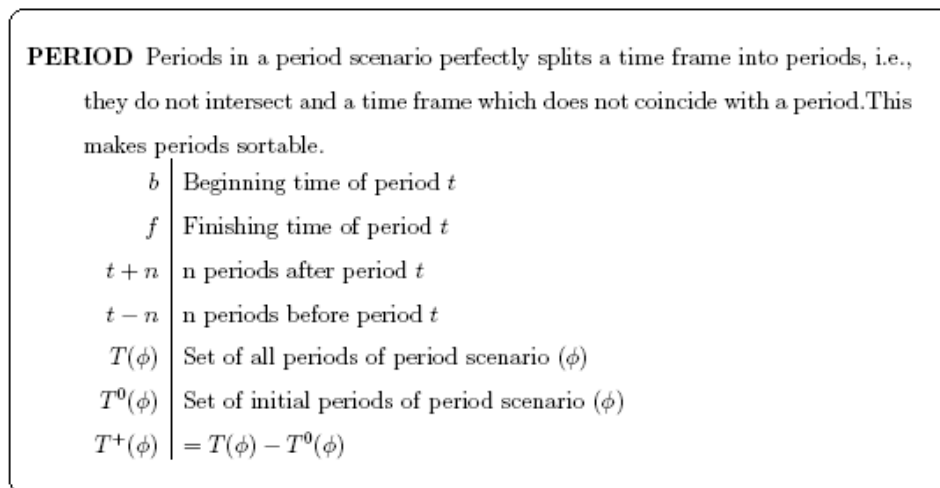


Figure 6.18. CP Model Definitions-1

**PART<sub>t</sub>**

$A_t$	Set of processes of part $p_t$
$H_t$	Set of paths of part $p_t$
$I_t$	Set of inventories of part $p_t$

**INVENTORY<sub>t</sub>**

$z$	Initial inventory stored in $i_t, i_t.t \in T^0(\phi)$
$y$	Percent yield of inventory $i_t$
$q$	Quantity stored in inventory $i_t$ at the end of the period
$p_t$	Part of inventory $i_t$ ( $i_t.p_t \in P_t$ )
$t$	Period of inventory $i$ ( $i_t.t \in T(\phi)$ )
$S_t$	$\{s_t : s_t.d_t.i_t = i_t, s_t.t = i_t.t\}$ Set of deliveries from inventory $i_t$
$A_t$	$\{a_t : a_t.i_t = i_t\}$ Set of processes feeding inventory $i_t$
$O_t$	$\{h_t : h_t.it = i_t\}$ Set of paths with destination inventory $i_t$
$I_t$	$\{h_t : h_t.if = i_t\}$ Set of paths leaving inventory $i_t$

**PROCESS<sub>t</sub>**

$lt$	Lead time of process $a_t$
$y$	Percent yield of process $a_t$
$q$	Quantity of process $a_t$
$i_t$	Output inventory of process $a_t$ ( $a_t.i_t \in I_t$ )
$p_t$	Part of process $a_t$ ( $a_t.p_t \in P_t$ )
$t$	Completion period of process $a_t$ ( $a_t.t \in T^+(\phi)$ )
$T$	Periods within the lead time of process $a_t$ $ a_t.T  = a_t.lt$
$PR$	Set of part requirements of process $a_t$

Figure 6.19. CP Model Definitions-2

$RR$	Set of resource group requirements of process $a_t$
$DD_t$	Set of dependent demands derived by process $a_t$ $\forall d_t \in a_t.DD_t, d_t.t = \max(0, a_t.t - a_t.lt)$
$U_t$	Set of resource group usages during process $a_t$ . There is one usage $u_t$ for each $rg_t$ such that $rg_t.t = \max(t \in a_t.T)$
<b>DEMAND<sub>t</sub></b>	
$q$	Quantity of demand $d_t$ ( $d_t \in D_t$ )
$m$	BOM multiplier: Quantity of part $d_t.p_t$ required for unit $d_t.a_t$ ( $d_t \in DD_t$ )
$pr$	Demand $d_t$ which is derived according to part requirement $pr$ ( $d_t \in DD_t$ )
$p_t$	Part of demand $d_t$ ( $d_t.p_t \in P_t$ )
$ID_t$	Set of independent demands ( $ID_t \subset D_t$ )
$DD_t$	Set of dependent demands ( $DD_t \subset D_t$ )
$S_t$	$\{s : s_t.d_t = d_t\}$ Set of all deliveries that can satisfy demand $d_t$ For any $d_t \in ID_t$ , there is one $s_t$ with $s_t.t = t$ for each $\{t : t \geq d_t.t\}$ For any $d_t \in DD_t$ , there is one $s_t$ with $s_t.t = d_t.t$
<b>PART REQUIREMENT</b>	
$n$	Number of units required for part requirement $pr$
$p_t$	Part of part requirement $pr$
$a_t$	Process of part requirement $pr$
<b>RESOURCE GROUP REQUIREMENT</b>	
$q$	Required capacity of resource group requirement $rr$
$a_t$	Process of resource group requirement $rr$
$r_t$	Resource group of resource group requirement $rr$
<b>DELIVERY<sub>t</sub></b>	
$q$	Quantity of delivery $s_t$
$t$	Period of delivery $s_t$ ( $s_t.t \in T$ )
$d_t$	The demand which is satisfied by delivery $s_t$ ( $s_t.d_t \in D_t$ )

Figure 6.20. CP Model Definitions-3

<b>PATH<sub>t</sub></b>	
$q$	Quantity delivered over path $h_t$
$f$	Starting (from) inventory of path $h_t$ ( $f \in I_t$ )
$t$	Ending (to) inventory of path $h_t$ ( $t \in I_t$ )
$p_t$	Part of path $h_t$ ( $h_t.p_t \in P_t$ )
<b>RESOURCE GROUP<sub>t</sub></b>	
$t$	Period of resource group $rg_t$ ( $rg_t.t \in T^+$ )
$P_t$	Set of parent resource groups of resource group $rg_t$ ( $rg_t \in RG_t$ )
$C_t$	Set of child resource groups of resource group $rg_t$ ( $rg_t \in RG_t$ )
$U_t$	Set of usages of resource group $rg_t$
$k$	Independent capacity of resource group $rg_t$ ( $rg_t \in RG_t$ )
$F_t$	Capacity transfers from resource group $rg_t$ ( $c_t \in F_t$ where $c_t.f = rg_t$ )
$T_t$	Capacity transfers to resource group $rg_t$ ( $c_t \in T_t$ where $c_t.t = rg_t$ )
<b>RESOURCE GROUP USAGE<sub>t</sub></b>	
$m$	Capacity of resource group $u_t.rg_t$ to be consumed per unit process $u_t.a_t$
$q$	Already completed quantity of the resource group consumption ( $u_t.t = 0$ )
$q$	Quantity of resource usage ( $u_t.t > 0$ )
$a_t$	Process of resource group usage $u_t$ ( $u_t.a_t \in A_t$ )
$rg_t$	Resource group of usage $u_t$ ( $u_t.rg_t \in RG_t$ )
$rr$	Resource group requirement of resource group usage $u_t$
<b>CAPACITY TRANSFER<sub>t</sub></b>	
$q$	Amount of capacity transferred
$c_t$	Capacity transfer object
$f$	Resource group, capacity transferred from
$t$	Resource group, capacity transferred to where $c_t.t \in c_t.f.P$

Figure 6.21. CP Model Definitions-4

### 6.3. Constraints

Resource usage

$$\sum_{u_t \in U} u_{t,q} / u_{t,m} = a_{t,q}, \forall a_t \in A_t, r_{g,t} \in RG_t$$

Inventory balance

$$i_{t,y} - i_{t-1,q} + \sum_{a_t \in i_{t,A}} a_{t,y} - \sum_{s_t \in i_{t,S}} s_{t,q} + \sum_{h_t \in i_{t,I}} h_{t,q} - \sum_{h_t \in i_{t,O}} h_{t,q} = i_{t,q}, \forall i_t \in I_t$$

$$i_{t,z} - \sum_{s_t \in i_{t,S}} s_{t,q} = i_{t,q}, \forall i_t \in I_t, i_{t,t} \in T^0$$

Demand - delivery

$$\sum_{s_t \in d_{t,S_t}} s_{t,q} = d_{t,q}, \forall d_t \in D_t$$

Resource group capacity

$$\sum_{u_t \in r_{g_t,U_t}} u_{t,q} \leq r_{g_t,k} - \sum_{c_t \in r_{g_t,F_t}} c_{t,q} + \sum_{c_t \in r_{g_t,T}} c_{t,q}, \forall r_{g_t} \in RG_t$$

BOM

$$d_{t,q} = d_{t,m} - d_{t,a_t,q}, \forall d_t \in DD_t$$

Figure 6.22. Constraints of CP Model

## 7. INTEGRATION SCENARIOS

In hierarchical planning, there are levels such as top and base levels. Objective and level of details in top and base level are different. Although top and base levels have different objectives and level of details, both top level and base level are built for solving a planning problem in a company.

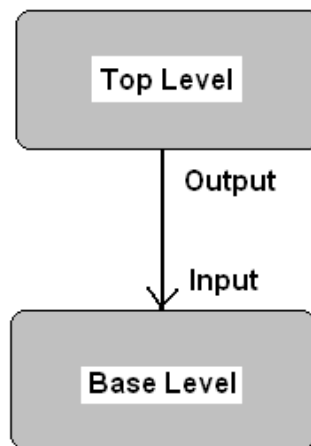


Figure 7.1. Integration between top and base level

As shown in figure 7.1, top level and base are linked to each other with output and input relation. Depending on the model for top level and base level used, different output and input relations can be classified into integration scenarios.

### 7.1. Integration Between CP and APS

In a company, in order to make decisions about long term or medium term issues, capacity optimization should be used. After solving the capacity optimization model, tactical or strategical decisions are obtained. An advanced planning and scheduling model should be used in order to make decisions about short term issues according to results obtained from the CP model. After solving advanced planning and scheduling model, operational decisions are obtained. Thereby, a hierarchical planning cycle is completed.

Depending on environment of production and company, there are different planning problems. Complexity of problems are also different. Since there are different planning problems, output of capacity planning and usage of output of capacity planning as input for advanced planning and scheduling differs.

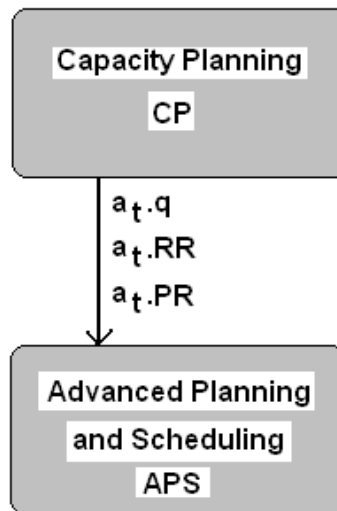


Figure 7.2. Integration of CP and APS

For instance; a company is producing only two parts  $p1$  and  $p2$ , needed operations for producing  $p1$  and  $p2$  are only operations  $o1$  and  $o2$ . Resources for producing  $p1$  and  $p2$  are very expensive so company wants to maximize production and profit with the existing resources  $r1$ ,  $r2$  and  $r3$ . Since the company producing only two parts, objective of capacity optimization model is determination of which operations or splits of operations are going to be scheduled on which resource. Output of capacity planning is used as an input for advanced planning and scheduling. As shown in figure 7.2, output of CP includes information of  $a_t$ ,  $a_t.RR$  and  $a_t.PR$ . For this case,  $A_t = (o.AL)xT$ , so predetermined alternatives for operations or splits of operations are going to be used while scheduling. For this case, there is an aggregation of processes on operation level. As shown in figure 6.9, details of operations are aggregated.

For instance; a company is producing parts  $p1$ ,  $p2$ ,  $p3$ ,  $p4$  and  $p5$ . There are forecasts in high quantities. Forecasts are defined as demands in capacity planning model

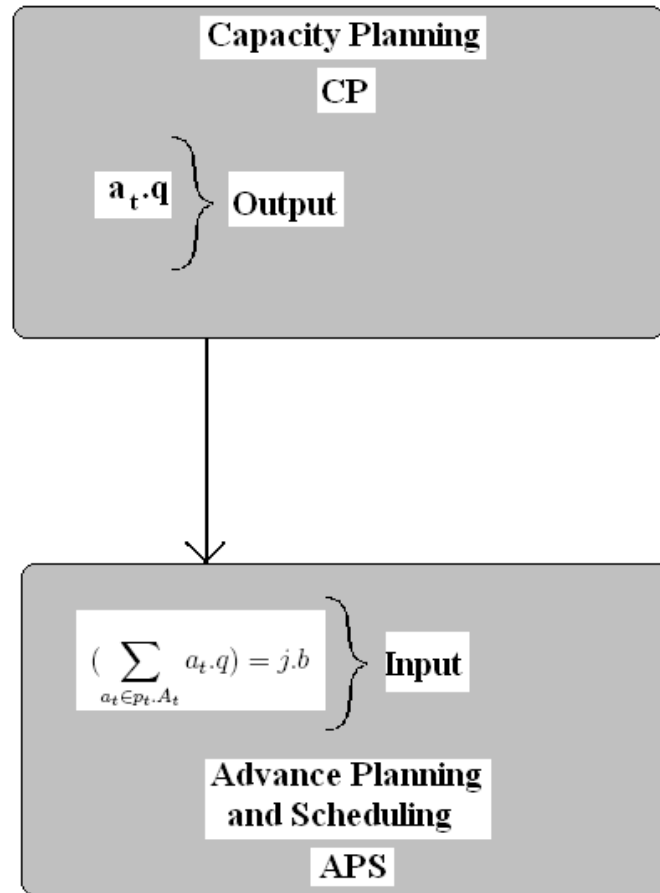


Figure 7.3. Case for Integration of CP and APS

and in order to satisfy in the following periods, demands should be divided into smaller portions in order to create jobs in advanced planning and scheduling model. During division of demands, capacity constraints should be considered. There is an aggregation of process on job level in this case (6.1.5.2 defines processes of capacity planning model.) After solving capacity planning model, information of demand portions can be obtained from processes in capacity planning. Information includes determination of period and amount of production for parts. As shown in figure 7.3, output of capacity planning model should be used as input to the advanced planning and scheduling model. Output of capacity planning is  $a_t \cdot q$ . Since sum of  $a_t \cdot q$  for a part i.e.  $p1$  is going to be a job having  $p_t.t.f$  period's end time as  $j.d$  due date, sum of  $a_t \cdot q$  for part  $p1$  as  $j.b$  and  $j.p = p1$ .

## 7.2. Integration Between CP and CP

In a company, in order to make decisions about long term or medium term issues, capacity optimization should be used. After solving capacity optimization model, tactical or strategical decisions are obtained. But for instance; for large scale firms having more than one factory, there is an extra level of hierarchy in planning so need of more than one capacity optimization model is unavoidable.

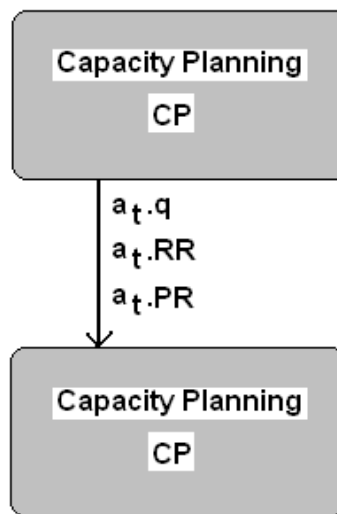


Figure 7.4. Integration of CP and CP

For instance; a company is producing parts  $p1$  and  $p2$ . Company has two factories  $f1$  and  $f2$  in different locations. For the next periods, there are many demands requesting parts  $p1$  and  $p2$  in different locations where customers are placed. So, satisfying a demand from a factory is a crucial decision, because time constraints, transportation costs, production costs and resource capacity of factories has to be considered during making decision. In capacity optimization model at the top level, objective is decreasing transportation costs, production costs and increasing customer satisfaction by delivering demands on time in order to make profit at the whole company not at one of the factories.

Solution of capacity optimization model at top level is going to be used as input for the capacity optimization model at the base level. Solution of CP at top level

determines which demand is going to be delivered by which factory. So there are four process such as producing  $p1$  at  $f1$ ,  $p1$  at  $f2$ ,  $p2$  at  $f1$  and  $p2$  at  $f2$ . After solving CP at top level, demands are going to be assigned to factories.

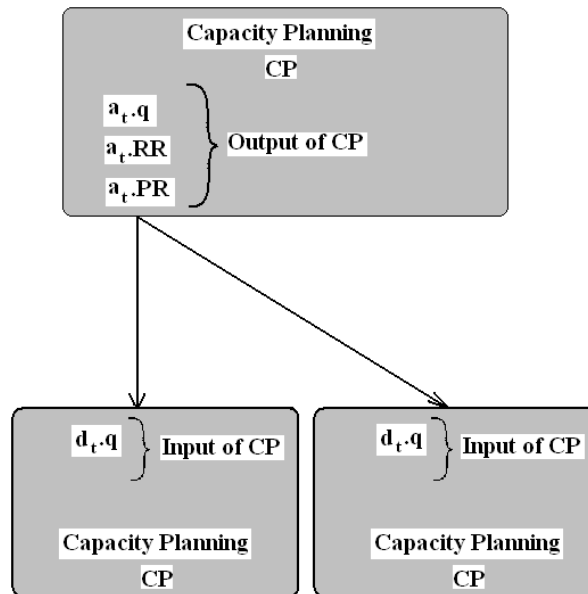


Figure 7.5. Case for Integration of CP-CP

Then capacity optimization model at base level turns out to a standard capacity optimization model. In this capacity optimization model, solution is going to guide planners about quantity of producing on which resource at which period and in order to satisfy demands. Quantity of process obtained from top level  $a_t.q$  is an input to base level capacity optimization model as  $d_t.q$  where  $a_t.q = d_t.q$  and  $a_t.p_t = d_t.p_t$  (part of process  $a_t.p_t$  and part of demand  $d_t.p_t$  is same).

### 7.3. Integration Between APS and APS

In order to produce and satisfy customer requirements on time with high utilization of resources, advanced planning and scheduling model is a detailed planning tool to overcome planning problems.

In some cases, not only one advanced planning and scheduling model is enough for making plans. Large scaled companies has many departments in order to divide planning problems into small problems to deal with. Departments are linked to each

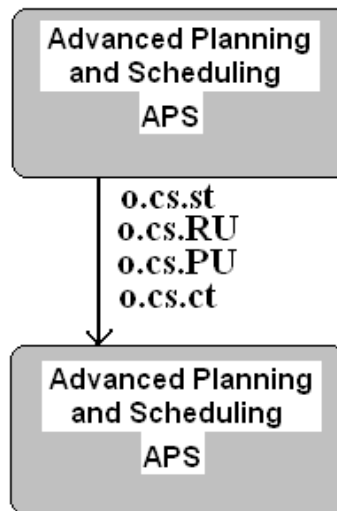


Figure 7.6. Integration of APS-APS

other as internal customers. Although they are linked to each other, they have different objectives to achieve. So, need of more than one advanced planning and scheduling model is unavoidable. Not only in large scaled companies but also in companies, linked to each other based on satisfying requirements of the company at top level in hierarchy, need of more than one advanced planning and scheduling model is unavoidable.

For instance; company W is producing products  $p1$  and  $p2$ . In order to produce one unit of  $p1$ ,  $m$  units of  $p10$  and  $n$  units of  $p20$  are needed. For  $p2$ ,  $x$  units of  $p10$  and  $y$  units of  $p21$  are needed. Procurement of  $p10$ ,  $p20$ ,  $p21$  is important. Company W uses APS for scheduling. While scheduling, in order to obtain required amount of  $p10$ ,  $p20$ ,  $p21$ , on hand inventory of  $p10$ ,  $p20$ ,  $p21$  are not taken into account, on hand inventory of both  $p10$ ,  $p20$ ,  $p21$  are infinite. Then according to schedule, required amount of  $p10$ ,  $p20$ ,  $p21$  are obtained in order to satisfy jobs of  $p1$  and  $p2$ . In order to apply released schedule, required amount of  $p10$ ,  $p20$ ,  $p21$  have to be on hand at right time. Company Z is a supplier of  $p10$ . Company Z also uses APS for scheduling. So, part usages of company W calculated from released schedule are inputs for company Z. From the schedule released by company W, start time of operations and part usages are calculated in APS. So, part requirements of company W are taken as jobs for company Z and due date of jobs are start time of operations in released schedule.

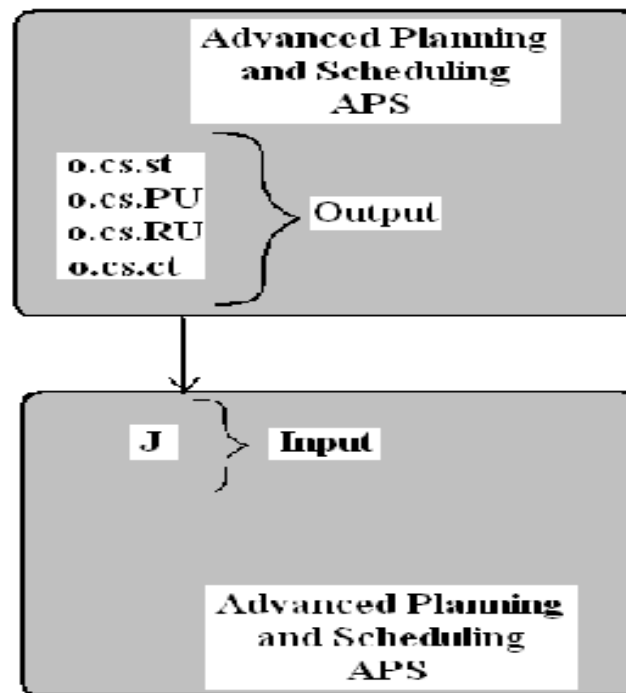


Figure 7.7. Case for Integration of APS-APS

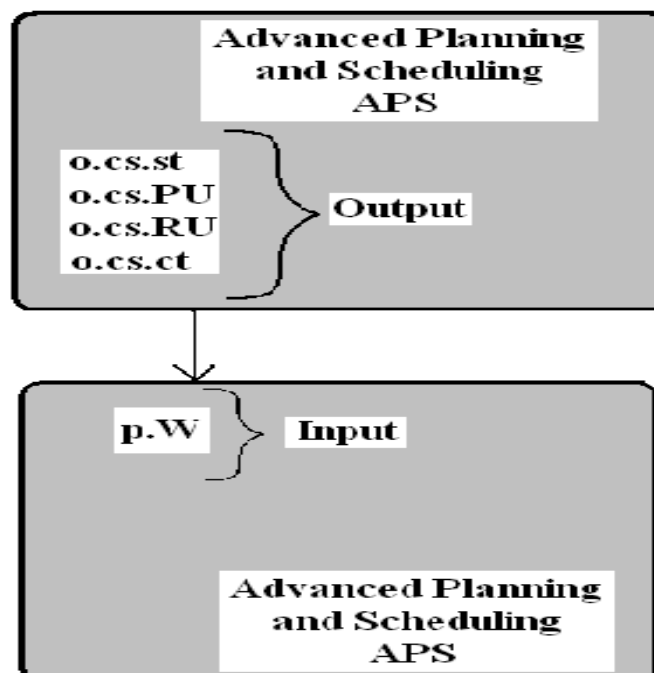


Figure 7.8. Case for Integration of APS-APS-2

For this case, most of the time, customer has the power but in some cases suppliers may have power for instance if there is monopoly or company W and company Z may

be included in a group of company thereby company Z is the only choice as a supplier for company W. Then as defined in figure 7.8 top level and base level changes vice versa. According to schedule obtained from company Z, acquisition plan of  $p10$  is at hand. Then according to acquisition plan of part  $p10$ , company W makes schedules.

#### **7.4. Integration Between APS and CP**

Objective of CP is making feasible plans in order to satisfy demands in planning periods and making profit. While satisfying demands, there are some constraints that has to be taken into consideration such as; resource capacity, inventory balance, etc. Since environment of production and market conditions are dynamic, although there is not enough raw material inventory, new demands may arise and some of demands may drop although you started producing to satisfy those demands. Everytime CP model runs, output of CP models is going to be different. In APS perspective, adaptation of APS to results of CP is not so easy because changing plan at the APS side may be too expensive. Changing plan may cause setups and change in requirements for raw materials for that period because according to plan released before, there was no need for that parts to have. But according to latest released plan, there are required parts. Sometimes depending of power of company, needed parts can be supplied very quickly but this is not the case everytime for all companies.

In order to make fewer changes on first schedule obtained from advanced planning and scheduling model, schedule obtained from APS somehow should be considered. So another cost may be defined; such as cost of revision of plan and added to objective function or another constraint, quantities obtained from first plan can be defined as minimum production quantities, can be added to the model.

So, there has to exist an integration between APS and CP. Schedule obtained from advanced planning and scheduling should be taken into consideration as an input to capacity planning model.

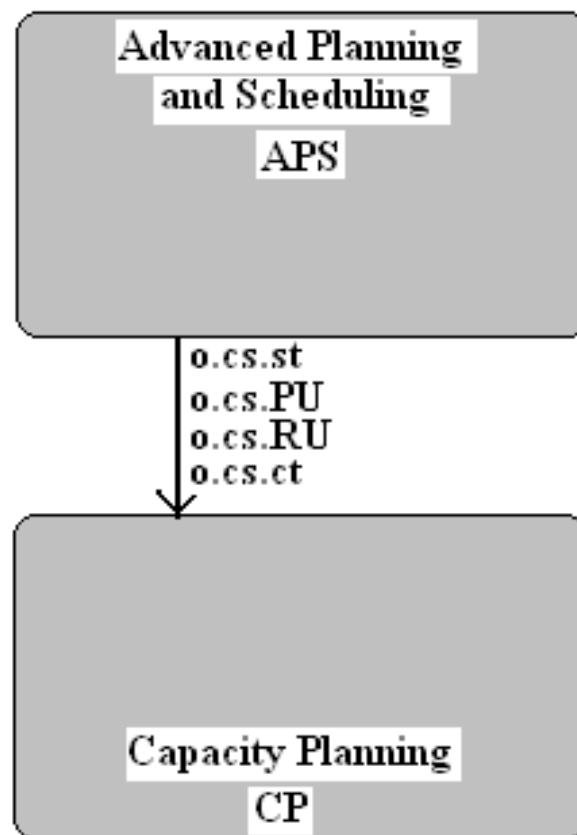


Figure 7.9. Integration of APS-CP

## 8. REAL LIFE APPLICATIONS

### 8.1. A Pharmaceutical Company Case

Integration scenario between continuous time advanced planning and scheduling and bucket based capacity planning is the one in which there is APS on base level and CP on top level. A real life example to this kind of integration scenario is the ICRON project that is implemented in a pharmaceutical company.

In this project, sales department prepares budgets and forecasts for a year. In order to make detailed schedules, forecasts for a year is long period. So in capacity planning model, according to capacity restrictions, forecasts should be divided into portions for finding monthly demands.

In section 6.1.5.2, aggregation on job level is explained in detail.

There is an aggregation of process on job level in this case (6.1.5.2 defines processes of capacity planning model. In section 6.1.5.2, aggregation on job level is explained in detail. There is a demand aggregation in this case also. After solving capacity planning model, information of demand portions can be obtained from processes in capacity planning. Information includes determination of period and amount of production for parts. As shown in figure 7.3, output of capacity planning model should be used as input to advanced planning and scheduling model. Output of capacity planning is  $a_{t.q}$ . Since sum of  $a_{t.q}$  for a part i.e.  $p1$  is going to be a job having  $p_{t.t.f}$  period's end time as  $j.d$  due date, sum of  $a_{t.q}$  for part  $p1$  as  $j.b$  and  $j.p = p1$ .

### 8.2. Trakya Cam Case

Integration scenario between bucket based capacity planning and bucket based capacity planning is the one in which there is CP on base level and also CP on top level. A real life example to this kind of integration scenario is the ICRON project

that is implemented in Trakya Cam Sanayi A.Ş. (Taskın and Ünal (2008))

Trakya Cam Sanayi A.Ş., which was founded by Türkiye Şişe Cam Fabrikalar A.Ş. in 1978, is the leading company and the pioneer of the flat glass industry in Turkey.

Trakya Cam that employs the latest technologies in production operates in the fields of basic glass (flat glass, patterned glass, mirror, laminated glass, coated glass, glass for architectural applications), automotive glass and glass for other vehicles, energy glass, glass for home appliances and provides input for various sectors including construction, automotive, home appliances, furniture, energy, and agriculture.

There are many companies requires float glass, both companies connected to Trakya Cam Sanayi A.Ş. and not related companies requires flat glass. There is a CP-CP integration in Trakya Cam. Since flat glass is required for laminated glass, coated glass etc. For a company producing any of products that requires flat glass as raw material, output of capacity planning model determines needed flat glass for company.

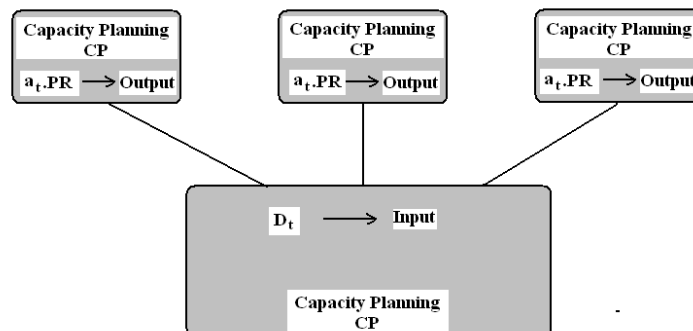


Figure 8.1. Trakya Cam - Integration of CP-CP

In order to satisfy demands from both connected or not related companies using flat glass as raw material, a capacity planning model has to determine period and quantity of float glass that can be satisfied in that period.

There is no customized objects and relations in this project for building both any

aggregation and integration scenario.

### 8.3. Brisa Case

Integration scenario between advanced planning and scheduling and advanced planning and scheduling is the one in which there is APS on base level and also APS on top level. A real life example to this kind of integration scenario is the ICRON project that is implemented in Brisa. (Tacettin and Unal (2008))

Brisa is the number one tire manufacturer in Turkey and the sixth biggest tire producer in Europe. Brisa manufactures a wide range of tires to meet domestic and international demand under the Lassa and Bridgestone brands. The product range of Brisa includes the following; Ultra high performance tires, high performance tires, standard car tires, winter tires, 4x4 vehicle tires, light duty commercial vehicle tires, steel radial van/light truck tires, bias-ply van/light truck tires, steel radial bus/truck tires, bias-ply bus/truck tires, agricultural tires, off the road tires.

In order to manufacture tire, there are raw material needs to be supplied. In Brisa, there are processes linked to each other based on raw material requirements. Last process which is called as curing is producing final products; tires. Raw material requirements of curing are satisfied from process which is called assembly. Advanced planning and scheduling (APS) is used for both curing and assembly processes in order to make schedules. Since both processes use APS for planning and processes are linked to each other based on raw material requirements, there has to exist an integration between APS at top level and APS at base level. APS model of assembly is at base level and APS model of curing is at top level.

As seen in 7.7, output of APS model at top level is schedule. Schedule gives detailed information of start time, finish time, resource requirements and part requirements of operations. And part requirements of APS model at top level are defined as jobs for APS model at base level.

## 9. CONCLUSIONS

Carrying out control of today's complex systems from one centralized decision making unit is almost impossible. This condition comes from the fact that there exist diverse problems in companies which are solved by separate decision making units. So it is very time consuming and also very hard to achieve control of a system from one center without dividing the whole problem into sub problems.

This situation gave a direction to emerging of a new issue which is called hierarchies in distributed decision making. This new issue explains coordination of separate decision making units which have relation with each other in order to solve problems of the complex system that they belong to.

In this study, hierarchies of planning is being studied which is a hierarchical decision making system including sub planning models. These sub planning models can be bucket based capacity planning and continuous time advanced planning and scheduling. There exist many companies which can be included in the examples applying hierarchical planning system including capacity planning and advanced planning and scheduling.

Bucket based capacity planning is a planning system considering capacity constraints while achieving best usage of resources. By using bucket based capacity planning, companies are able to give significant decisions like where to produce and when. Bucket based capacity planning has a large scope but with less consideration on details.

Continuous time advanced planning and scheduling is a more detailed but less scoped planning in which companies can make decisions on distribution of jobs . While giving this decision, some clever methods can be used like heuristics and algorithms.

Both bucket based capacity planning and continuous time advanced planning and scheduling are modeled in our study. Object structure is given in detail and also

relations between objects are clarified. With the help of class diagrams given in the study, it is more easy to understand the overall system, objects and relations between them. By stating all these items, modeling of continuous time advanced planning and scheduling and also bucket based capacity planning are given. Continuous time advanced planning and scheduling and bucket based capacity planning models are given in an ontology.

Since there are some differences between capacity planning and advanced planning and scheduling in areas like scope and level of detail, there should be some additional work on making these sub planning systems compatible with each other. This is a requirement for aggregation. In order to ensure that capacity planning and advanced planning and scheduling form a hierarchical planning system which are working coherent with an interchange of solutions, aggregation is a need. In a hierarchical planning system, there is an aggregation from base level to top level in the hierarchy.

There are kinds of aggregation which are used by companies using hierarchical planning systems. In this study, we also give aggregation types and give their definitions by giving examples. The aggregation types include process aggregation , resource aggregation , time aggregation , demand aggregation and part aggregation .

Process aggregation is mainly aggregating small processes of base level as bigger processes of top level like it is in aggregating jobs of advanced planning and scheduling as a demand for the planning period of capacity planning.

Resource aggregation includes aggregation of resources that are belonging to base level in the hierarchical planning system as resource groups of the top level in the hierarchy. Combination of base level resources constitute resource groups of the top level.

Time aggregation implies making plan in shorter time buckets with more detail and making plan with less detail by aggregating these small buckets as a longer period of top level. This aggregation can be between continuous time advanced planning

and scheduling and bucket based capacity planning and also between more than one bucket based capacity planning which includes aggregating shorter time intervals of one capacity planning on another capacity planning.

Like other aggregation types, demand aggregation is aggregating small demand amounts of base level as larger demand values of top level in the hierarchy. This aggregation between continuous time advanced planning and scheduling and bucket based capacity planning is made by aggregating jobs of advanced planning and scheduling and combining them into larger demands on the capacity planning side.

The last type of aggregation, part aggregation, is mainly aggregating parts of the base level according to a feature which is common enough to aggregate and forming them as a part of top level of the hierarchy .

Moreover; in order to show that the ontology proposed in the study is applicable to modeling real life problems, integration scenarios of bucket based capacity planning and continuous time advanced planning and scheduling for companies from different sectors are modeled.

With the help of this study, generalization of bucket based capacity planning and continuous time advanced planning and scheduling systems is achieved by modeling them and also an ontology is being generated by describing object structure of these models and relations between those objects. Aggregation types and also integration of capacity planning and advanced planning and scheduling is also given in detail.

This generalization of bucket based capacity planning and advanced planning and scheduling gives applicability of hierarchical planning systems in companies having complex systems. By using proposed study, an analysis for aggregation and integration can easily be done on a company with hierarchical planning system.

As guide to further researches, it may be useful for modeling the proposed ontology in real life problems and examining the results. A work on modeling companies

which have hierarchical planning systems with the proposed ontology will be exiting and enlightening.

Also it will be very beneficial for making aggregation and integration of hierarchical planning systems in a rule based perspective. This rule based aspect will also have significant utility for new researches on hierarchies in planning systems.

## REFERENCES

- Blythe J. and Y.Gil, 1996 , “Planet: A shareable and reusable ontology for representing plans” , *Technical report*.
- Federgruen A., J. Meissner and M. Tzur, 2007 , “Progressive Interval Heuristics for Multi-Item Capacitated Lot-Sizing Problems” , *In Operations Research*, Vol. 55, pp. 490-502.
- Frankovic, B., I. Budinska and T. Dang, 2002 , “Creation of Ontology for Planning and Scheduling” , *In Proceedings of the 3 International Symposium on Computational Intelligence*, Vol. 12, pp. 73-84.
- Gil Y. and J. Blythe, 1999, “PLANET: A Shareable and Reusable Ontology for Representing Plans” , *Representational Issues for Real-world Planning Systems*, Vol. 28, pp. 28-33.
- Graves S.C., 1982 , “Using Lagrangean techniques to solve hierarchical production planning problems” , *Management Science*, Vol. 28, pp. 260-275.
- Hax A. C. and M. Harlan, 1973 , “ Hierarchical integration of production planning and scheduling” , *Massachusetts Institute of Technology, Sloan School of Management*, Vol.20, pp. 656-673.
- Houba H.G. , R. Hartog, T. Jan L. Top ,A. Beulens and L. Berkel, 2000 , “Using recipe classes for supporting detailed planning in food industry: A case study” , *European Journal of Operational Research*, Vol. 122, 367- 373.
- Merdan M, G. Koppensteiner, A. Zoitl and B. Favre-Bulle, 2007 , “Distributed Agents Architecture Applied in Assembly Domain” , *International Symposium on Knowledge and Systems Sciences*, Ishikawa, Japan.

- Mönch L. and M. Stehli, 2003 , “An Ontology for Production Control of Semiconductor Manufacturing Processes”, *Mates*, pp. 156167.
- Motta E., D. Rajpathak, Z. Zdrahal and R. Roy, 2002 , “The Epistemology of Scheduling Problems”, *In Proc. Os the 15 European Conference on Artificial Intelligence*.
- Ozdamar L., M. A. Bozyel and S. Birbil, 1998 , “A hierarchical decision support system for production planning (with case study)”, *European Journal of Operational Research*, Vol. 104, pp. 403-422.
- Ozdamar L. and S. Birbil, 1999 , “A hierarchical planning system for energy intensive production environments”, *Int. J. Production Economics*, Vol. 58, pp. 115- 129.
- Rajpathak, D., Motta, E. and Roy, R., 2001 , “The Epistemology of Scheduling Problems”, *In Proceedings of the International Conference on Artificial Intelligence*, Nevada, Las Vegas.
- Smith, S. F., O. Lassila and M. Becker, 1996 , “A Generic Task Ontology for Scheduling Applications”, *International Conference on Artificial Intelligence*, Toledo, Spain.
- Smith, S. F., O. Lassila and M. Becker, 1996 , “Congurable, mixed-initiative systems for planning and scheduling”, *Advanced Planning Technology*, Vol. 98, pp. 235-241.
- Smith, S. F. and M. A. Becker, 1997 , “An Ontology for Constructing Scheduling Systems”, *Proceeding of AAAI-97 Proceeding of AAAI-97* , Stockholm, Sweden.
- Voss., S. and L. Woodruff, 2002 , *Introduction to Computational Optimization Models for Production Planning in a Supply Chaine*, Springer, New York.
- Tacettin, M. and A. T. Unal, 2008 , “Brisa Presentation”, *YAEM 2008*.
- Tan W. and B. Khoshnevis, 2000 , “Integration of process planning and scheduling - a review”, *Journal of Intelligent Manufacturing*, Vol. 11, pp. 51-63

Taskin, Z. C. and A. T. Unal, 2008 , “Tactical Level Planning in Float Glass Manufacturing with Co-production Random Yields and Substitutable Products”, *European Journal of Operational Research* 2008 doi: 10.1016 / j.ejor.2008.11.024.