Chapter 3. Planning and Planning Support

3.1. Introduction
The previous chapter provides an overview of various characteristics of food processing industries, including a description of how planning and production control takes place. We conclude there that the planning requirements from the market in some aspects oppose the planning requirements from the production system. If we look at our research question from this perspective, we can state that we seek ways how the planning can deal with the opposing requirements.

In this chapter, we explore literature that deals with the planning task of human planners. In this perspective, the starting point is the way in which the planning task is actually performed, rather than the view on planning as a collection of activities that together form an organizational control mechanism.

We start with a description of the task of human planners. We show that algorithms from Artificial Intelligence that are based on cognitive models of planning need careful consideration before they can be applied in planning support in organizations. After that, we will discuss planning from the perspective of the entities that are planned. Lastly, computerized planning tools are discussed. There, we discuss that algorithms for planning support need to be attuned to the subtasks of human planners. The chapter ends with a conclusion.

3.2. The role of human planners
Since we regard planning as a task that must be performed in an organization, humans play an important role. In literature, however, the word ‘planner’ has two meanings. The first meaning refers to the profession of a member of an organization. Organizational processes must be planned so people know what work they are going to do. An example is a production planner in a factory. The second meaning of planner refers to an actor that must think about its actions before he performs them, for example making a shopping list. Roughly speaking, the difference between the two is whether the planner makes a plan that is executed by others or by himself. Both connotations of planner have their own research fields. The profession of a planner is mainly analyzed in the context of computer support. Example research areas are knowledge acquisition, task modeling, decision support, constraint modeling and solving, and operations research. The second meaning of planning is dealt with, for example, by cognitive psychology, semiotics, Artificial Intelligence, and robotics. Apparently, both planning worlds have their own research methodologies, languages, ontologies, and models. Clearly, research from the first

---

field is applicable to our study. Whether this also holds for the second meaning of planner, is not trivial to answer. Therefore, we will first pinpoint where the differences lie in both forms of planning.

3.2.1. The recursive planning paradox

The question whether “planning for yourself” (or “life planning”) is the same as “planning for others” (or “organization planning”) is an important one for our purpose. After all, if they are truly the same, then it should be possible to apply research for the one to the domain of the other and vice versa without further consideration. If, on the other hand, they are not equal but merely similar, then a further analysis should be made to figure out what aspects are the same and what aspects are different, and how theories and mechanisms from one research field can be applied to the other. In this section, we will show with a conceptual line of reasoning why both research fields differ and how this affects our analysis. Henceforth, the term ‘planner’ denotes the profession kind unless explicitly stated otherwise.

Analyses of planners in organizations show that there are a number of recurring task components. Mietus (1994), for example, mentions the subtasks information collection, counting, attuning, adaptation, and negotiation for nurse schedulers in Dutch hospitals. Somehow, there is a mechanism that determines the order of these subtasks. The order of some subtasks is logical (e.g., first collect information, then count), but other orderings such as the moment to negotiate, and loops of subtasks (assign, negotiate, assign again) are somewhat more intricate.

A trivial but interesting remark is that the planner must determine when he/she starts doing what subtask. In other words, the planner must plan the planning task. It is this kind of planning (“life planning”) that is analyzed by cognitive sciences and Artificial Intelligence (AI). Our statement to show that planning for yourself differs from planning for others is based on this ‘planning for the planning’. For the sake of clarity, we call the task to make a plan in an organization ‘planning 1’, and we call the planning of the planning ‘planning 2’. Thus, ‘planning 2’ is making a plan for the task ‘planning 1’. Now suppose that planning for yourself is the same as planning for others. Then all characteristics of ‘planning 1’ would also apply to ‘planning 2’. Of course, this includes the necessity to make a plan for the task at hand. Because we must make a plan that determines how ‘planning 1’ will be done, and ‘planning 1’ is the same kind of activity as ‘planning 2’, we end up with the notion that we must make a plan for ‘planning 2’. Thus, we must make a plan for making a plan for making a plan. Let’s call it ‘planning 3’. Here the arguments repeat. We must make a plan for ‘planning 3’, which is ‘planning 4’. ‘Planning 5’ is making a plan how to make ‘planning 4’, etc. This recursion is unavoidable if one assumes that ‘planning 1’ is the same as ‘planning 2’ (Figure 3.1). Of course, practice proofs that plans are made without endless recursion. The only conclusion can be that planning for yourself differs from planning for others. This raises the question how they differ, and to what extent they are similar.
Thus, there are at least two distinct kinds of planning. The first kind is to make a plan for, e.g., the production process. The second kind is to plan the planning, since each task must be planned. Because this second level is not seen as a task anymore, there is no recursion and literature from cognitive sciences can be used to analyze this kind of planning, just like it is used to analyze the planning of other (cognitive) tasks (e.g., Hayes-Roth & Hayes-Roth (1979), and Newell & Simon (1972)). According to Das et al. (1996), there is a third level, which they call ‘meta-planning’ (Figure 3.2). This kind of planning is again different from the first and second level. In contrast to the second kind, which has a functional nature, this third kind has an identifiable location in the brain. As we will point out in the following sections, there do not seem to be more levels that are approached as planning.

Until now we have used the twosome planning for yourself and planning for others to denote two research schools. Actual research, however, is more complex (and for that matter, so is practice). Sometimes there is only a remote link between planning problems in practice and the planning problems that are addressed in research. In other cases, research might possess characteristics that occur in both research schools
so it is not unequivocally clear to what research school it belongs (in which case, of course, the statement that there are two disjunctive research schools proofs wrong). Nevertheless, the following sections provide a short overview of dominant streams within the two paradigms.

3.2.2. Planning for yourself: a cognitive viewpoint
Planning in cognitive sciences deals with the functioning of individuals. There are several directions of research that focus on this type of planning. We will respectively discuss planning from a neuropsychological perspective, the relation between planning and problem solving, and Artificial Intelligence approaches to planning.

Neuropsychological aspects of planning
Although neuropsychological research in planning is quite remote from planning as a coordination mechanism in organizations, it can be worthwhile to investigate how the functioning of a human individual relates to physical planning mechanisms, and how this compares to the aforementioned organizational planning.

Das et al. (1996) provide an extensive overview of the processes of the human brain that relate to planning. They describe that planning is one of three functional components of cognitive processes: (1) arousal and attention, (2) information reception, coding, and storage, and (3) planning, self-monitoring, and structuring of cognitive activities. These units are sequential: information can only be received after arousal-attention, and only then can it be synthesized and acted upon in the third component. Still, they are also tightly coupled since processes in the third component direct resources for attention.

The third functional component is the one that gives humans their intelligence. It is located in the prefrontal areas of the frontal lobes. This area of the brain is intimately connected with the motor cortex, which indicates that planning can directly control actions. Anticipation and temporal structures of behavior are located in this component, so it is the primary part of the brain for goal directed behavior of humans.

The considerations about neuropsychological aspects of planning are used by researchers in cognitive psychology to build models that can describe and explain human behavior. The idea behind it is that neuropsychological structures set boundary conditions for the functional level of description (Newell & Simon, 1976). The following subsection discusses how planning is taken up in this kind of research.

Planning and problem solving
Cognitive models of planning show how problem solving, planning, and information processing relate. Miller et al. (1960, p. 16) define planning as “any hierarchical process in the organism that can control the order in which a sequence of operations is to be performed”. Das et al. (1996, p. 27) state about this that “it is the plan that controls human information processing and supplies patterns for essential connections between knowledge, evaluation, and action”. This very generic
description can be extended by the approach of Newell & Simon (1972). They describe planning as a system of heuristics that is used by their General Problem Solver (GPS) “to construct a proposed solution in general terms before working out the details. This procedure acts as an antidote to the limitation of means-ends analysis in seeing only one step ahead.” (op. cit., p. 428). Planning heuristics are used to guide action when a problem is too difficult to solve by means-end analysis. Newell & Simon assume the following steps in planning: “(1) abstracting by omitting certain details of the original objects and operators, (2) forming the corresponding problem in the abstract problem space, (3) when the abstract problem has been solved, using its solution to provide a plan for solving the original problem, (4) translating the plan back into the original problem space and executing it.” (op. cit., p 429). Complexity is reduced by leaving out details and reasoning by analogy. In this sense, planning is a way of problem solving.

Early models of planning presume that planning is always a hierarchical process that proceeds according to successive refinement. Sacerdoti (1975) implemented such an approach in his computer program NOAH. In this view, planning is performed by decomposing goals in sub goals recursively, until a sub goal can be reached by elementary actions. This paradigm is contradicted by Hayes-Roth & Hayes-Roth (1979). In their line of reasoning, they first argue “that planning processes operate in a two-dimensional planning space defined on time and abstraction” (op cit., p. 312). In those terms, successive refinement would always work top-down from high to low abstraction and forward in the timeframe of the plan. Thinking aloud protocols from different subjects that perform planning tasks show that this is not always the case. Hayes-Roth & Hayes-Roth found planning actions that they call ‘opportunistic planning’. The subjects do not work solely linear but appear to switch in levels of abstraction and move both forward and backward in time in successive reasoning steps. Hayes-Roth & Hayes-Roth (1979) propose a theoretical framework for cognitive planning that incorporates this behavior. Some behavior that can be explained by their model is multi-directional processing in addition to top-down processing, incremental planning, and heterarchical (i.e., network) plan structures. According to Hayes-Roth & Hayes-Roth, the choice of a planning strategy depends on three variables: the problem characteristics, individual differences, and expertise. Task strategies within a domain depend on individual differences and change over time if experience increases.

Riesbeck & Schank (1989) argue that planning is based on scripts. Instead of thinking up a new plan for each problem, humans try to find a plan that is used for a previously solved comparable planning problem. Then, the basic planning activity is more adaptation than it is construction. In this paradigm, planning is about memory, indexing and learning (Hammond, 1989; Veloso, 1996). These issues are very much interrelated. Plans should be stored in such a way that it becomes easy to find an existing plan on the basis of a comparison of the new goal with already handled goals. There are two senses of learning in the case-based planning paradigm. First, solutions must be remembered so they can be used for new problems. Second, a failure to execute the plan provides an indication that the knowledge that the planner
has of the execution world could be faulty. Thus, script models can be seen as adding learning to the paradigms already discussed. Together, the three paradigms that were discussed provide various interpretations of a cognitive approach to human planning. In this approach, planning is about how to find the actions that solve a problem or, more general, reach a goal. The process of planning is not neatly hierarchical but switches in level of abstraction and in the time frame under consideration. The process itself is about formulating goals, finding similar solved goals, finding existing plans, adapting plans, and storing plans in such a way that they can easily be found for future reference.

**Artificial Intelligence**

Planning in artificial intelligence is very much related to the problem-solving approaches as described in the previous section. The contribution of Artificial Intelligence is that it introduces computer programs that simulate and test problem-solving strategies of humans. Although the distinction between problem-solving approaches and Artificial Intelligence is somewhat blurring, it is certainly true that contemporary methods in Artificial Intelligence relate less to human problem solving than to finding technological solutions for intricate problems. Therefore, we will discuss several Artificial Intelligence search techniques that probably do not resemble the way in which humans reason, but all the same provide problem solving approaches for planning problems.

Much of the planning research in Artificial Intelligence stems from the wish to let autonomous agents (such as robots) perform tasks without prescribing how the task should be carried out. For this, planning is an important part, and it is no surprise that cognitive models of human planning are used as exemplars. Most Artificial Intelligence methods (also called algorithms, procedures, or heuristics in this context) are based on state space descriptions. An agent or actor finds itself in a state, in which it can perform a limited number of actions. An action changes the state, after which it can again perform a number of actions. The agent keeps on choosing and performing actions until the state it gets in somehow satisfies its goal. Planning is one way in which the agent can reach its goal (other ways are, for example, trial and error or full search). To make a plan, an agent somehow simulates the actions he will make beforehand. The original link to physical entities has been relinquished somewhat so planning agents are now often only computer programs that find or make a plan but not necessarily execute it. Interestingly, in Artificial Intelligence planning literature, planning agents are called planners. Of course, this causes quite some confusion of tongues if Artificial Intelligence researchers talk with or read from organization scientists.

The disadvantage of a state space approach is that it requires a lot of information storage because for each operation in each state, the resulting state must be known. The approach STRIPS (Fikes & Nilsson, 1971) resolves this issue by separating actions explicitly from state descriptions, and adding a condition to each action. The model then consists of a world that is described as a number of characteristics that can be true or false in any specific state (for example, a door is open or closed), and a
number of actions that can be applied (for example, open or close the door). In each state, all rules must be considered to determine what actions can be taken in the current state. An action can be taken if its corresponding condition is true. The action is described as the characteristics that it changes. For example, we have the rule: if the door is open (condition), then close it (action): the door becomes closed (which is a change in the world so the state of the agent changes). In this modeling paradigm, planning is searching for a sequence of actions that will bring the agent from its current state in the goal state. The way in which a planner finds this sequence is the paramount activity of Artificial Intelligence planning approaches. Rich & Knight (1991, p. 333) describe five functions that a planner of this kind must be able to perform:

- “Choose the best rule to apply next based on the best available heuristic information.
- Apply the chosen rule to compute the new problem state that arises from its application.
- Detect when a solution has been found.
- Detect dead ends so that they can be abandoned and the system’s effort directed in more fruitful directions.
- Detect when an almost correct solution has been found and employ special techniques to make it totally correct.”

Models of actions and states can not only be used to describe the behavior of artificial agents but also of human problem solvers. Models of human problem solving, that were discussed in the previous subsection, have provided researchers in Artificial Intelligence with starting points for the planning functions of their artificial agents. Examples are the General Problem Solver (GPS), which constructs a proposed solution in general terms before working out the details, the opportunistic planning paradigm, and script-based planning. Here it becomes clear that models of human problems solving are closely related to Artificial Intelligence, because it is no exception that research in human problem solving is initiated from the wish to make intelligent artificial agents. There have been numerous extensions and elaborations of the original STRIPS paradigm, for example, where actions and events take time, actions may overlap, incomplete knowledge is allowed, and the plan deals with multiple agents (e.g., Allen et al, 1996; Dean & Kirman, 1996; Wilkins, 1990). Other search methods are for example nonlinear planning (a plan is nonlinear if it has multiple sub-plans that are worked on simultaneously instead of sequentially) and constraint posting. In the latter search method, a state is not a description of the ‘world’ but a collection of operators that can be used to calculate the ‘world state’. Such a collection of operators represents a partial or complete plan. In this model, operators to change from one state to another are the addition or deletion of plan operators.
Artificial Intelligence models of planning are based on (cognitive) models of human decision making. In this way, the functioning of artificial agents is based on the way that humans plan. The recursive planning paradox states that this kind of planning differs from planning as it is performed in organizations. This is discussed in the next section.

3.2.3. Planning for others: the task of human planners

Several analyses of the planning task are described in literature (examples will be given shortly), but general agreement about the merits of the planning profession is missing. It is rather strange to note that most professions have their own educational program, but that planning is not recognized as such. Perhaps this is due to the large amount of case specific knowledge that is needed in the planning task, though we argue that the same goes for other complex professions. Still, some generic knowledge about the planning task can be abstracted from the task analyses that are described in literature. Such generic knowledge can aid several purposes. First, it can provide basic knowledge about planning. Although there probably is no best way of planning, the learning process of planners can be quickened by providing generic knowledge. Second, it provides a way to compare and communicate about planning situations. Third, it can help in assessments of the task performance of a human planner, and fourth, it can provide a link to functional requirements of decision support. Most task analyses that are described in literature are performed with the prospect of computer support. This indicates that important other aspects of planning tasks do not get much attention, for example task improvement in itself and organizational aspects of planning. In this subsection, we will first discuss planning task characteristics that are found in literature. Then, we will discuss several task models of planning.

Determinants of task performance

In order to make generic statements about the planning task, it is important to know what the task performance depends upon (notice that by performance we mean execution without a qualitative connotation). According to Hayes-Roth & Hayes-Roth (1979), the determinants of the planning task are problem characteristics, individual differences, and expertise. That the task performance depends on individual differences and expertise is no surprise. It goes for all tasks. But the fact that the task performance also depends on problem characteristics leads to the statement that it is possible to describe a planning problem at least partly independent from the planner. Unfortunately, Hayes-Roth & Hayes-Roth do not elaborate on this. The analysis of Wiers (1996) describes four schedulers that seem to have the same type of problem but show a large difference in their task execution. He made performance criteria, actions, and disturbances operational in 16 variables and measured these variables during 4 months in a truck manufacturing company. Wiers states that the results consist of a “spaghetti of relationships that is not easy to interpret” (p. 390).
McKay et al. (1995, p. 1601) discuss a number of measurable attributes that can say something about the quality of a human planner in situations of instability:

- **Accuracy** – the ability to predict [...] can be measured, and as the scheduler’s ability increases, the horizon should extend;
- **Span** – how far into the future a scheduler can accurately predict is related to planning and this can be measured – as the scheduler’s ability increases, the horizon should extend;
- **Decision timing** – it is possible to measure the timing of short, mid, and long term decisions – when they occur during the decision process. It seems that most schedulers have multiple decision modes that work with varying degrees of aggregation and time horizons (e.g. similar to hierarchical production planning levels). Based on our preliminary observations, it appears that good schedulers do not artificially restrict the different levels to specific times of the day, week, or month, while novice schedulers use regimented boundaries;
- **Decision domain** – the ability of the scheduler to identify and suggest changes to the processes, procedures, and capacity of the plant can be captured. Coupled with prediction tracking, this last trait can illustrate how much the scheduler understands about the situation and contributes to manufacturing improvement. An experienced scheduler should be expected to participate in the complete system and put their knowledge and skill to good use”

Although these points were not meant exhaustively, they provide some directions to what constitutes the individual characteristics and expertise mentioned by Hayes-Roth & Hayes-Roth.

Mietus (1994) analyzed thinking aloud protocols of three groups of nurse schedulers: experts, novices with practical training, and novices without practical training. She concludes that due to the lack of domain knowledge, novices without practical training are unable “to perform problem solving adequately” (op cit., p. 85). According to Mietus, novices do not look at the domain that the schedule represents (in this case nurses and shifts), but they look at the schedule as a puzzle to be solved without being able to interpret the schedule in the wider context. The goals that the novices tried to achieve were clearly linkable to a domain object (e.g., honoring a wish of a nurse), whereas experts also used goals that were based on relations between objects (e.g., continuity in the schedule). Mietus found that the novices use an opportunistic planning approach, whereas experts appear to use scripts that determine a more or less fixed sequence of actions (e.g., see Figure 3.3).

To conclude, there is not much known about what factors determine the way in which the planning task is performed. There are, however, several initiatives to model the task of human planners. We will continue by discussing some examples of such models.
Task models of planning

Although theory on planning has shown that planning problems are NP-hard from a mathematical perspective, workable plans are created in practice. Some sort of trade-off must be made to allow for the difference between the (perhaps not even definable) optimality and the solution that is achievable within the boundaries of time and people that are available in practice. The way in which human planners create and choose alternatives is at the core of their task. We will shortly discuss some task analyses in order to clarify the way in which the planner’s task can be depicted.

An often used way to describe how planners make decisions is heuristics or rules of thumb. Such rules describe what action is taken if some conditions are met. McKay et al. (1995), for example, studied extensively the decision behavior of Ralph, a planner at a printed circuit board factory. They found 128 policies and heuristics that Ralph uses to perform his task. These were separated in routine heuristics for standard situations (e.g., determination of the batch size and prioritize orders) and exceptions (e.g., how to deal with products that are not made in a long time). The latter category comprised more than a hundred rules.

Heuristics can be used to create computer models of decision behavior (Dutton, 1964). They are often implemented in so called rule-based systems, that use the heuristics to draw conclusions by deduction about a given situation. Sanderson (1989) discusses a number of studies where heuristics are modeled by a linear model instead of a rule-base. Some authors describe the task at a somewhat more abstract level with flowcharts. Such diagrams often consist of (a) subtasks and (b) a strategy that denotes the order in which the subtasks are carried out. In this sense, subtasks depict heuristics that decompose a task into independently solvable sub-problems. For example, the heuristic ‘first schedule the products of category A, then the products of category B’ is used in a task model to denote two subtasks. A sample task model in Figure 3.3 depicts the subtasks of nurse scheduling (Mietus, 1994). With the analysis Mietus was able to make a generic model of the nurse scheduling task. Such more or less ad hoc modeling activities provide detailed insight into specific problem domains. Although they can probably be generalized to some extent by determining which parts seem generic and which parts appear very domain specific, a more generic view about the planning task is missing. Breuker & Van de Velde (1994) propose to deal with tasks in a more structured way. They describe Kads, a generic taxonomy of task elements with which various tasks and domains can be described. Several planning and scheduling task models are described with Kads, two of which will be discussed here (Dorn (1993) and Sundin (1994)).

Dorn (1993) applies Kads to two planning problems in the steelmaking industry. He provides a generic domain model, inference structures, and task models. He acknowledges that Kads can be used as a methodology for designing task oriented planning support, but points out that Kads lacks mechanisms for temporal reasoning,
and that it is difficult to model reactive planning because events can occur that change the domain state during problem solving.

Figure 3.3. Division of the nurse scheduling task (Mietus, 1994, p. 93)

Sundin (1994) describes problem solving methods for assignment and scheduling problems for the CommonKads library (Breuker & Van der Velde, 1994). Like Schreiber et al. (1993), he treats scheduling as a special case of the assignment task. The solution of an assignment task is a set of relations between entities of the different sets in such a way that the constraints are not violated. The roles of the objects differ in different domains. According to Sundin, the entities of scheduling problems are respectively activities and time-slots. In order to depict the more generic class of assignment problems, however, Sundin models one entity set as resources and the other entity set as components. Scheduling problems with more than two entity sets, e.g., the classroom assignment problem with classrooms, teachers, classes, and topics, are decomposed into several assignment problems. In his models, he specializes the propose and revise problem solving methods which are general for all synthetic tasks in Kads. He creates a number of models in which subtasks are called functions (Figure 3.4 and Figure 3.5 contain examples of such functions). These models can be combined to depict a specific scheduling task. He is able to translate the task model of a personnel scheduling problem to the generic model that is based on propose and revise.
An interesting aspect is that task models usually deal with the generation of a plan, while in practice, planners often spend much time on other tasks than plan generation. Sanderson (1989, p. 661) notes about this that “in industrial environments, schedulers generally perform their scheduling activities in the context of a wide variety of other responsibilities”. Several authors report how much time is invested in subtasks of planning. Fox and Smith (1984) report that 80% to 90% of the time was used in identifying constraints and only little time in dispatching, Mietus (1994) indicates that clerical work (e.g., data collection) often accounts for 40% to 80% of the planners time, Allen et al. (1996) found that 23% of an hour of human-human problem solving dialogues dealt with planning, and Verbraeck (1991) even reports that (for his case) the actual planning task only accounted for 3% of the planner’s time. Bakker (1995) surveyed several planners in different situations Table 3.1. contains her data about the time that is needed for the several subtasks.


<table>
<thead>
<tr>
<th>Subtask</th>
<th>Starting from scratch</th>
<th>Using a previous schedule as a basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clerical work</td>
<td>21.2%</td>
<td>42.8%</td>
</tr>
<tr>
<td>Adjustments</td>
<td>18.5%</td>
<td>38.6%</td>
</tr>
<tr>
<td>Generation</td>
<td>55.9%</td>
<td>36.1%</td>
</tr>
<tr>
<td>Counting</td>
<td>10.4%</td>
<td>10.0%</td>
</tr>
</tbody>
</table>

Table 3.1. Average percentage of time spent on subtasks (Bakker, 1994, p. 19)

These figures indicate that task models should also incorporate other tasks than plan generation.

Organizational aspects of planning
The previous subsection discusses a number of task models. Such models provide a static picture of the task performance. Jorna et al. (1996, p. 74) describe a number of organizational aspects of the planning that can be used to complement the description of a task as a collection of subtasks (the word ‘organizational’ here refers to the organization of the task and therefore relates to the task strategy and not to the organization in which the planner works).

A first distinction deals with the temporal relation between planning and execution. The planning horizon can be fixed or rolling, and both planning and plan execution can be organized in time buckets (Table 3.2 on page 48). A second distinction looks at the content of the planning. It can be based on patterns. An example is a fixed sequence of production. Another example is that a train always takes the same route when it is shunted from one track to another. The organizational aspects of planning are summarized in Table 3.3 on page 48. These organizational aspects are meant as examples, and do not provide a complete classification for task strategies.

3.2.4. Comparison of planning for yourself and planning for others
As we have tried to show with the recursive planning paradox, planning for yourself differs from planning for others. Still, both deal with trying to determine a future course of actions. Therefore, research results from one area can possibly be used in the other. Models and algorithms that are created for planning as a cognitive activity can be used in two ways. First, they can assist in the planning part of the planning task. Second, they can be used as a paradigm for the planning task as a whole. The differences between planning for yourself and planning for others are summarized in Table 3.4 on page 49. The most apparent difference between planning for yourself and planning for others is the decoupling of making the plan and executing the plan. Two reasons for this decoupling are that planning is a difficult job that requires expertise and experience, and that planner must be able to weigh the interests of multiple parties and therefore must have knowledge about things that extend over the limits of the individual tasks that are planned. The separation of planning and execution can cause several problems that do not appear if one makes a plan for oneself.
Planning of a period takes place before the execution of that plan starts. The plan covers a time bucket, both during the planning process and during the execution. Changes to the plan are not made by the planner but by the one who executes it.

A plan is made before the execution of that plan starts, but it can be adjusted during the time of execution by the planner. There are time buckets, so adjustments to the plan for one time bucket can overlap with the creation of the plan for the following time bucket.

The plans cover distinguishable time buckets (for example, a week), but the horizon of the planning extends over the current and the next bucket. The relation between the moment of the plan decision and the time of execution it relates to is relinquished.

There are no time buckets.

**Table 3.2. Time frames of planning and plan execution**

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of the planning</td>
<td>The planning can be made once in a period, for example, each fortnight, weekly, or daily. In the extreme, planners work continuously.</td>
</tr>
<tr>
<td>Horizon of the planning</td>
<td>The horizon can be fixed or variable.</td>
</tr>
<tr>
<td>Repeatability of the planning</td>
<td>The planner can use previously created (partial) plans as a starting point</td>
</tr>
<tr>
<td>Pattern based planning</td>
<td>The planner can use patterns, e.g., fixed sequences of production or fixed shift patterns in personnel planning.</td>
</tr>
</tbody>
</table>

**Table 3.3. Organizational aspects of planning (Jorna et al., 1996, p. 74)**
<table>
<thead>
<tr>
<th>Entity kind (i.e. the entity that makes the plan)</th>
<th>Human that plans for himself</th>
<th>Human that plans in an organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alone/Group</td>
<td>Alone</td>
<td>Alone and Group</td>
</tr>
<tr>
<td>Natural/Artificial</td>
<td>Natural</td>
<td>Natural and Artificial</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Entity/process Characteristics</th>
<th>Human that plans for himself</th>
<th>Human that plans in an organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Processing</td>
<td>Internal</td>
<td>Internal and External</td>
</tr>
<tr>
<td>Representation</td>
<td>Internal: hidden and mental</td>
<td>External: various and coded</td>
</tr>
<tr>
<td>Communication</td>
<td>Internal: hidden and mental</td>
<td>Internal and External: mental and coded</td>
</tr>
<tr>
<td>Modeling</td>
<td>AI-models (temporal, case based reasoning)</td>
<td>OR-models</td>
</tr>
<tr>
<td>Relation planning, execution, and control</td>
<td>Intertwined; flexible adaptation after unforeseen Events</td>
<td>Decoupled; inflexible with respect to adaptation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Domain characteristics</th>
<th>Human that plans for himself</th>
<th>Human that plans in an organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem space</td>
<td>Ill-defined</td>
<td>Strive towards well-defined</td>
</tr>
<tr>
<td>Planned entities</td>
<td>Sequence of own activities</td>
<td>Alignment between other's activities, capacity, orders</td>
</tr>
<tr>
<td>Constraints/Goal Functions</td>
<td>Self-paced; self-imposed; easily revisable</td>
<td>Externally imposed, non-paced and difficult to change</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Research aimed at</th>
<th>Human that plans for himself</th>
<th>Human that plans in an organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulating planning process; operating autonomous agent</td>
<td>Support / completion of the planning process</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3.4.** Summary of differences between planning for yourself and planning for others (Van Wezel & Jorna, 2001)

The planner can have a different view of the entities that he plans than the people who must execute the plan. For example, although the real world is full of uncertainties, planners must make assumptions without the detailed knowledge that, for example, human operators have about the machines they operate. Furthermore, adjustment of the plan after unexpected events is troublesome at the least. This is because events must be communicated, and subsequent changes to the plan must be weighed and communicated back to the ones who execute. The strong integration of
the goal state, the expected execution path, and the execution of the plan that occurs during planning for yourself makes that one’s own plan is easily adjustable.

A second difference is the kind of entities that are planned. Planning for yourself is mainly restricted to ordering and possibly determining time boxes for future actions. The end product is an ordered list with actions, and an scheme or idea which resources will be used to perform the actions. In other words, the plan is a process description. For example, my plan to buy ingredients for my dinner could state that I first go with my bicycle to the grocery, then I go by foot to the bakery, etc. In contrast, the end product in organizational planning is often a future state description without the path that leads to it. In such a plan, the path that leads to the state as specified in the plan is determined by the person who executes the plan.

This difference is of course gradual, because an organizational plan could also specify the path explicitly (e.g., a transportation plan), or be based on an implicit agreement about which path ought to be followed.

A third difference is that the number of dimensions and the number of entities that are attuned in organizations are typically higher. This, together with the fact that organizational plans must always be communicated, makes that the information carrier for organizational planning is always external, e.g., paper or a computer file.

In our research question, we ask how day-to-day planning can contribute to flexible product replenishment. As noted, humans that make and execute plans for themselves are better able to react on unexpected events by adjusting plans than humans in organizations, because there is less communication and coordination involved. In other words, they are more flexible.

The literature review started with the role of human planners because we regard human planners as the starting point of analyses and design. The planning task has to do with the attunement of entities, e.g., nurses and shifts, orders and machines, or trucks and cargo. Because the kinds of entities that the planner deals with can influence the way in which the task is carried out, we will continue with a discussion about the entities that are planned by human planners.

3.3. The entities in the plan
A plan is a statement about the future. Such a statement does not necessarily imply a rigid determination of a future state, since unexpected events can lead to deviations of assumptions. Rather, it is a statement of an expected or desired future state. Since planning deals with the future, time is an important aspect. Most plans describe a future course of action, i.e., a number of tasks that have to be performed in a future time frame. A plan can also describe a static future state, for example allocation of employees to rooms. In this section, we will discuss ways to deal with modeling issues of the entities for which the future state is determined. We provide an abstract overview of types of entities that can be involved in planning. In chapter 5, we will provide a description of the entities that play a role in planning in food processing industries.
The descriptions of entities in a plan that are found in literature are roughly distinguishable in three types. First, there are models that describe the entities in the way that they are commonly known. The entities in such models are referred to as objects. These models describe mainly the function that the objects have in the execution of a plan, i.e., their function in the real world. Examples are machines that perform product transformations, personnel that perform tasks, locations at which goods are stored, vehicles that move products from one location to another, etc. Second, there are models that describe the role that the entities have in the task strategies of the planner. Examples are resources (whether it be personnel or machine), activities, supply, demand, etc. Third, quantitative models describe the entities in mathematical variables or equations. The different ways to model the planned entities each serve their own purpose:

- **Objects**: The description of entities as a reference to real world objects can be used (a) for communication between planners and others, and (b) for an easy identification of the information that is needed in the planning process. Such objects can provide a view of a planning problem that is independent of the interpretation and task performance of the human planner. In other words, this description should be the same for different planners.

- **Roles**: The description of entities as their role in the planning problem can be used to identify the way how planners reason in their task performance. This construct is used as a translation between the models of the analyst and the models of the human planner. This is closely linked to the discussion about the task performance of human planners. A model of the roles provides the ontological primitives that are referred to in a task description.

- **Mathematical constructs**: Quantitative models are used to solve scheduling problems mathematically. Mathematical formulations are often geared to algorithms that can find solutions. These kinds of models are not regarded in this research.

Because each kind of description ultimately deals with the same real-world objects, it must be possible to map different descriptions. An advantage of the possibility to map descriptions is that it becomes clear what information is lost in what description, since each modeling language has its own abstraction, and it becomes possible to compare the references to the domain or task models of different planners. In the remainder of this section, we will give some examples of models of planned entities.

Pillutla & Nag (1996) provide an object-oriented framework in which “real-world” objects of manufacturing scheduling problems such as machines, products, and raw materials are modeled with part-whole and generalization-specialization relationships (Figure 3.6). Information about object types is stored in class templates, and the set of class templates constitutes the model dictionary. A model for a new planning problem is either derived from existing models in the model base, or created from scratch. In both cases, the new models can be added to the model-base
to be reused in the future. Pillutla & Nag restrict themselves to the conceptual modeling of manufacturing problems but they acknowledge the need to enlarge the scope to other problem domains. Figure 3.6 contains some classes in a generalization-specialization hierarchy.

![Diagram of a generalization-specialization hierarchy]

**Figure 3.6.** Classes in a generalization-specialization hierarchy (Pillutla & Nag, 1996)

Smith & Becker (1997) describe an ontology with five basic concepts: Demand, Activity, Resource, Product, and Constraint. These concepts can be specialized to define models for sub-domains, e.g., manufacturing or transportation (Figure 3.7). According to Smith & Becker, the abstract model can be used to define generic solution methods. Such solution methods can be used for different situations, e.g., it would not make a difference for the solution method whether it concerns production planning or transportation planning as long as the methods only refer to characteristics of the basic concepts.

The models of Pillutla & Nag and Smith & Becker have many things in common, although the model of Pillutla & Nag is based on production problems whereas the basic model of Smith & Becker is open to more problem types. The basic concepts in the model of Smith & Becker seem to define roles of entities, but one level deeper, these are specialized as real world objects. This results in an interesting problem. At a truck manufacturer, a vehicle is the product of a production activity. If this is modeled with the scheme of Smith & Becker, then Vehicle would be a sub-type of the basic concept Product. Are we still talking about the same Vehicle as the one that is used as a resource in transportation? Or will it have other characteristics now that it inherits from another basic concept? In any case, the same real-world object would occur twice in the taxonomy of Smith & Becker. This supports our postulation that objects and roles should not be mixed in one model.
Figure 3.7. Abstract model and model with sub-domains (Smith & Becker, 1997)

In Chapter 4 we will describe an object oriented modeling language that is based on the same principles as the taxonomy of Pillutla & Nag (1996), without the restriction for the production planning domain. In the following section, the literature review will be continued by describing how planning can be supported by the computer.

3.4. Computerized planning support

3.4.1. Introduction

The previous sections dealt with the planning task and the entities that are planned. These aspects of planning can be accounted for in planning support. The amount of
information that is involved in planning problems makes that external tools for
information storage or manipulation are necessary. Pen and paper (“the backside of a
cigar box”) or a planning board are still no exception, but computerized support is
common nowadays. In this section, we will discuss computerized planning tools.

Roughly, there are three layers of sophistication in such tools. First, there are
general tools such as word processors and spreadsheets, in which the planning can be
typed and printed. These tools enable planners to view, search, and manipulate the
planning easily. For example, parts of an existing planning can be copied to a new
one, which saves time. Second, spreadsheets can be tailored to a specific planning
situation. Then, it can automatically make calculations and check constraints, for
example on capacity usage. The disadvantage of general tools is that it is hard to
integrate them in the corporate information structure. Although it could technically
be possible to link a spreadsheet to a corporate database in order to update the
planning, this is not something that a planner would be expected to know how to do.
Third, there are specialized tools for planning. Such systems offer integrated
functionality for data retrieval and maintenance, constraint checking, and plan
generation. Several authors note that spreadsheets do not suffice for even slightly
complex planning situations (Green, 1996; Kondili et al., 1993; Jakeman, 1994). One
of the main themes in our research is a faster reaction to events that can influence the
planning. To provide this, a planning tool should always be up to date with respect to
stock positions, order portfolio, etc. Therefore, we state a priori that planning support
should be of this third category and the literature review will focus on such
scheduling systems. Literature shows variety in functionality that is applied for
scheduling systems. We will describe some systems to cover the available
functionality.

Woerlee (1991) uses a traditional DSS architecture: a database, user-interface,
and model base. The database stores all information, the user-interface shows the
stored information and provides possibilities to update and manipulate the
information, and the model base contains generation algorithms. The system that
Woerlee specifies is meant to be somewhat generic, with a modeling language to
specify cases. Other systems do not really deviate from the traditional DSS
architecture, but rather provide a more detailed description of the three components.
Prietula et al. (1994) extend the DSS architecture in three ways. First, they explicitly
include a constraint checker and a critiquer. The constraint checker checks the
validity of the schedule, and the critiquer (or evaluator) provides an analysis in
which the number of satisfied constraints, preferences, and scheduled high-priority
jobs are taken into account. Second, the model base explicitly separates generative
scheduling from reactive scheduling. The latter is invoked when the generative
scheduler or a human scheduler makes a proposition and a constraint violation is
detected. Third, the manual scheduler explicitly allows interactions between
algorithms and the human user by cooperative problem solving. Hildum et al. (1997)
add a blackboard to their system. At the blackboard, multiple solutions and partial
solutions can be stored for later retrieval. Everyone that is working on the problem
can view solutions that others have proposed. This allows a human user and an
algorithm to work simultaneously on the solutions of a problem, and the planner or algorithm can easily try solutions paths and switch between partial solutions. In addition, Hildum et al. pay attention to integration with production process. In this view, the database should not be the end point of the scheduling system. Close integration with information systems of, e.g., suppliers and the shop floor, enables real-time reactivity. Figure 3.8 shows an architecture that includes these components.

Not all existing systems provide all the functionality, and the way in which the functionality is implemented also differs. The following sub-sections will describe the main functions in more detail.

![Figure 3.8. Scheduling system architecture](image)

### 3.4.2. Information retrieval

Before a planner can start planning, he needs to collect the information on which the planning decisions will be based. In general, this amounts to information about the entities that are going to be planned, the constraints with which the plan must comply, the goals of the plan, and feedback of the execution of a previous plan. The following information holders can be distinguished:

- **Computer systems**, for example a central database. It can be retrieved easily if the right software is available.
- **Paper.** Examples are personnel records, old plans, orders on faxes, and printouts of computer systems.
- **People.** Information can be held by people who communicate to the planner.

Information collection can be a very time consuming task. Table 3.1 showed figures of roughly twenty and forty percent. Information that is needed for planning but that
can not be retrieved from a database must be manually entered, which is slow and error prone. Information can be described on various aspects:

- **Ease of acquirement.** Not all information is readily available so it must be gathered and interpreted, e.g., counting pallets in the warehouse to know the stock positions. In addition, available information might not be in the right format, so the planner must transform it, e.g., aggregation of stock positions to product families.

- **Timeliness.** Preferably, the information must be available before the planning process starts. In practice, however, this is not always the case. Of course, it is not efficient to change a plan because new information becomes available.

- **Certainty.** Information is often not certain. Examples are processing times, machine performance, stock positions, order portfolio, etc. Two types of uncertainty can be distinguished. First, the uncertainty distribution can be known. Then it is possible to perform calculations in order to decide on the basis of an expected value. Second, the distribution can be unavailable. Then it is known that the information is uncertain, but not to which degree.

- **Stability.** The information can change while the planner is making a plan. Possibly, parts of plans can become invalid and decisions must be reconsidered.

In an ideal situation information is easily acquired, always up-to-date, with a high degree of certainty, and fairly stable. More and more, computer systems integrate the required information, and standardization of database exchange protocols brings real-time use of information in reach of scheduling systems. Real-time links with order processing, the warehouse, and the shop floor increase the ease of acquirement and the timeliness.

### 3.4.3. Schedule editor: show and manipulate the plan

The schedule editor is the part of the user-interface that deals with manipulation of the plan (next to other parts of the user-interface, e.g., configuration, data entry, constraints, etc.). It represents the communication interface about the plan between the human planner and the computer. The editor shows the plan and the planner must be able to manipulate the plan with it. If we formulate the goal of the editor in abstract terms, we can say that it must show the entities that are going to be assigned, it must show the assignment somehow, and it must allow the planner to make assignments. As an example, Figure 3.9 contains a screen of a Gantt-chart with which the human planner can manipulate the plan by dragging orders to machines on a time-line.

Usually, a display that shows a plan contains two kinds of elements. First, there are the entities that are going to be assigned. It is often possible to show details of the entities and to order them according to criteria such as earliest due date, color,
largest batch size, etc. Second, we see the assignments themselves. In production planning, the machines and time buckets are usually fixed beforehand and the orders are the entities that are put in.

Pinedo & Yen (1997, p. 374) describe a number of ways to display information about production schedules and manipulate schedules:

1. “The order and Job Information Interface. This interface consists of multiple messages regarding the attributes of the order objects and the job objects. This interface often uses menus, dialogue boxes and text windows.
2. The Gantt Chart interface. The Gantt chart interface is a canvas with methods for drawing Gantt charts, with function buttons, and with messages regarding the status of the schedule.
3. The Machine Dispatch List interface. This interface has to be a multiple text window (it has to display for each machine a list of the jobs in the order in which they are to be processed), with function buttons and with messages for the display of the schedule status.
4. The Capacity Buckets interface. The construction of this interface is similar to the construction of the Gantt chart interface.
5. The Throughput Diagram interface. This interface depicts, usually in a graphical way, the number of jobs that have not started with their processing yet, the number of jobs currently in the system and the number of jobs which have already completed their processing.
6. The Performance Evaluation interface. This interface displays the values of all the performance measures of interest. These measures may be displayed either in alphanumeric characters or in the form of bars.”
In addition to these displays, Pinedo & Yen (op. cit.) provide some examples of interfaces that display data about orders, jobs, and machines independently of the schedule:

1. “*The Plant Layout interface.* This interface may depict graphically the work centers and the machines in the plant as well as the possible routes between the work centers.

2. *The Resource Calendar interface.* The resource calendar displays shift schedules, holidays, preventive maintenance schedules of the machines, and so on.

3. *The Routing Table interface.* The routing table depicts static data (which are independent of the schedule) that are heavily job and machine dependent.”

It is not always clear what kind of editor should be used for what kind of planning problem. First, there is the question whether to use graphical displays or not. For production planning this amounts to the choice between a Gantt-chart and a machine dispatch list. A Gantt-chart is a matrix display with time and resources along the axis and production orders in the cells. The dispatch list is a textual overview of the jobs that have to be processed on a machine. In other words, the dispatch list provides a detailed overview of one row of a Gantt-chart. This is suitable if separate machines have no relations. Sharit (1985) shows that graphical displays do not have to lead to better performance of supervisory control in a flexible manufacturing system. Danek & Koubeek (1995) relate the type of task, the level of cognitive processing, and the type of graphical display to the performance of the scheduling task. They found empirical evidence that an integrated graphical display of performance measures facilitates better performance than a display that has a graph for each performance measure individually.

Second, literature does not provide guidelines how to deal with the integration of automatic and manual scheduling. The kinds of interfaces that we just described provide adequate means do display a schedule and to manually manipulate it. If a schedule is generated by a black box kind of algorithm, the outcome can be shown in such an interface. However, it is not trivial how it should be shown what steps a generative algorithms makes, how the planner can react on that, and what constraints are violated. In addition, a blackboard architecture with multiple partially solved plans is probably convenient in the planning process, but how it should be presented to the human planner is not a standardized issue yet.

3.5. **Schedule generation: The imitate/replace debate and mixed initiative support**

Most research on day-to-day planning deals with automating the quest for good schedules. Automated schedule generation, however, often leaves little room for human control in the search process. The optimal balance between human control versus automatic plan generation is not clear from literature. Advocates of analytical
models argue that humans do a poor job at planning. Advocates of a more human centered perspective, however, state that analytical models can not deal with uncertainty and instability of the real world (McKay et al., 1988). Specifically, the latter state that the lack of application of scheduling systems in practice is due to the black box nature of such systems (McKay et al., 1989; Sanderson, 1989).

We distinguish three main philosophies of schedule generation. The first two categories focus on mere generation. The first category of techniques only looks at characteristics of the domain. The second category only looks at the way that the problems is solved by humans. Techniques in the third category base themselves on support requirements that stem from decision support theory. Techniques in this last category try to find a balance between efficiency that can be reached by using the computer and the fact that the solution must be understood by the human planner. The three categories will be discussed respectively.

**Domain oriented generation techniques**

First, there are approaches that focus mainly on the domain without analyzing the way in which the problems are solved by the human planner. The possibilities of the computer are then not restricted by the human planner. In such approaches, characteristics of domain entities and their relations are analyzed (for example, capacity of machines, shift requirements, historical data of working hours, etc.) and an algorithm is formulated to efficiently find a schedule that does not violate constraints. Thus, these approaches focus on schedule generation. There are several lines of approach to generate a schedule. Operations Research techniques are available for all domains of plan generation such as job shop, flow shop, routing, etc. (Baker, 1974). Such techniques, however, operate under strict restrictions and limitations of the amount of variables, making them often unsuitable to optimize real-world problems. Another critique on these approaches is that they are not understandable by human planners and therefore an often neglected part of a scheduling support system. Artificial Intelligence and Computational Intelligence techniques are proposed to solve this problem (Brown et al., 1995). The so-called intelligent scheduling or knowledge based scheduling approaches are presumed to cohere more to the problem solving steps of the human planner, and the techniques appear more easily adaptable to changed circumstances. Examples of knowledge based schedule generation are constraint based search, simulation of heuristics, activity-based scheduling, and fuzzy scheduling (Smith, 1992). Still, these approaches are made to the analogy of humans that plan for themselves, and not humans that plan for organizations (compare Sections 3.2.2 and 3.2.3).

The domain oriented approaches can yield powerful algorithmic support, but the amount of coherence with the cognitive processes of the human planner is probably small, even with the knowledge based techniques. Note that this is not necessarily a hindrance for application in practice. Rather, it depends on the task environment characteristics if such an approach is worthwhile. If much can be gained by optimization then this approach can be beneficiary. Limiting factors are that the constraints must be clear and exhaustive, and the optimization goal must be clear. In
addition, the situation must be rather stable since changes in the domain induce changes in the algorithms, and fine-tuning algorithms is time consuming.

Problem solving oriented generation techniques
Second, approaches can focus on imitating the human problem solving processes in so called rule bases or expert systems, also called the transfer view (Schreiber et al., 1993) because the knowledge is extracted from a human and transferred into a computer program. For this approach, the problem solving approach of the human scheduler must be analyzed. In terms of the human problem solver (Newell & Simon, 1972), this means that the problem space and operators must be traced and implemented. As with the domain oriented approach, the distribution of tasks between the computer and the user is mainly towards the computer, but the available computational capacity is not used since the computer is used as a symbolic processor. It is, however, understandable for the human planner why a generated plan looks as it looks, because he would have processed the symbols in more or less the same way. This approach can be used if a planner is satisfied with a reduction of the efforts without much improvement of the solution, e.g., if planning is a secondary task of the one who makes it. The main disadvantage of this approach is that the system not only inherits the capacity of abstract reasoning that is so typical of humans, but also the myopic fire fighting tactics that human schedulers practice (Smith, 1992). In addition, the resulting algorithms are highly specific for the individual human planner.

Task oriented generation techniques
Third, the task oriented or mixed-initiative approach combines the first two approaches. Domain oriented and expert system approaches both focus on computerized schedule generation. In the mixed initiative approach, the support approach focuses on improvement of the solution by establishing a coalition between the computer and the user. Hereby, not the domain or the problem solving process is the main focal point, but the task of the human planner. This implements the common DSS view that both human and computer should do the tasks they are best at. This is called Knowledge-based decision support (KB-DSS). KB-DSS approaches are equipped to analyze the task and problem solving behavior of human decision makers. In this paradigm, decision support is based on the decision processes of the human decision maker. Thereby, the aim is not to accurately mimic the human problem solving process, but to provide support for the problem solving process. In addition, by focusing on the knowledge level, it is not necessary to use techniques that come from Artificial Intelligence to provide knowledge based support (Newell, 1981). The focus in a KB-DSS is on the level at which the system and the user communicate. Since such a system will change the current problem space of the human planner, it is not straightforward which decisions should be taken by the user, which decisions should be taken by the computer, and for which decisions the user and the computer should cooperate. Benbasat & Todd (1996) have shown that the decision strategy that is chosen by a human problem solver is contingent upon
environmental demands and decision aids. They pose that human problem solvers weigh the effort that a strategy will cost against the accuracy of the expected solution. By introducing decision aids, the effort can be reduced or the accuracy (or quality) of the solution can be increased. Normative decision models can be applied if they will reduce the effort of existing strategies that the human problem solver can apply. This implies that decision support does not have to focus solely on an individual user. If a system provides enough benefits, the user will be inclined to change his own decision behavior. Benbasat & Todd (following Newell & Simon, 1972) use elementary information processes (EIP’s) as the smallest task elements. Decision aids should substitute such information processes so that other EIP’s (either performed by man or machine) can use the output of these small elements. In this way, intermediate results are understandable by the decision maker. Note that what is elementary is not defined and contingent upon the goal of the analysis and the characteristics of the domain.

Task oriented scheduling support focuses on both the domain characteristics and computational advances that have been made to solve scheduling problems in the domain, and on the problem solving processes of the human planner. Changes in the task that are forced by introduction of scheduling support do not pose a problem as long as the planner feels that either the task strain is reduced or the quality of the outcome is increased. This approach can be applied if the quality of the outcome is important, but the task environment is unstable and requires human judgment. Several authors report about the use of the task oriented approach.

McKay et al. (1995) describe an approach for field research of scheduling problems in which methods from social sciences are used to gather and analyze the field research data. Thereby, organizational and cognitive issues get explicit attention. Both Sundin (1994) and Dorn (1993) use the Kads approach (Breuker & Van de Velde, 1994) to model the task of schedulers. Dorn notes that it is necessary to have task libraries and an indexing scheme for scheduling systems in order to find an appropriate task for the purpose of reuse. Several authors state that they will extend their work with more explicit attention for task modeling (e.g., Lui, 1993; Smith & Becker, 1997).

Prietula et al. (1994) describe how they applied the concept of ‘coincident problem spaces’ to the scheduling system McMerl, with the following proposition: “To configure effectively a support system that can exploit the knowledge of the scheduling expert, it is important to direct the behavior of the system to function in a manner that is consistent with the key problem spaces of the scheduler; that is, the system and the scheduler should be problem solving in coincident problem spaces.” (p. 660). They acknowledged the restricted computational abilities of human planners, and designed a system that could augment the search of the human with search by the computer. In addition, they used a mixed-initiative approach for the system, because “the system could not anticipate the entire set of parameters that define acceptable solutions” (p. 657). In such an approach, both the human planner and the computer can take initiative in the proposition of schedule decisions. The human planner, though, always has the last word. To overcome the problem that the
problem space of a human planner can change after the introduction of scheduling support, Prietula et al. (1993) apply the Soar architecture. This architecture allows (symbolic based) learning, so changes in the task that occur after the introduction of the scheduling support system can be depicted in the reasoning mechanism of the system.

Both Benbasat & Todd (1996) and Prietula et al. (1994) propose to provide decision support at the knowledge level. They aim at a division of tasks that are done by the computer and tasks that are done by the human. Important to note here is that the paradigm of human problem solving as symbol processing is used to define the level of communication, but not necessarily the way of reasoning by the computer.

Scheduling systems often use heuristics that compromise optimality for performance reasons. Hofstede (1992) gives some prerequisites to which heuristics must comply if they are to be used in interactive support at the knowledge level. First, the user must be able to interact during operation. Second, the problem representation must consist of objects that are meaningful for the planner and it must be possible to show the progress of the heuristic to the user. Third, the operators or transitions in the heuristic must refer to actions in the real world. Fourth, the control mechanism must allow the user to alter the current state during execution of the heuristic, and must provide a way for the user to make a trade-off between the quality of the solution and the time spent in generating it.

The research of Mietus (1994) shows the distinction between subtasks that can be performed by the computer (such as counting) and the subtasks that need human judgment (such as scheduling a shift). For the latter, the level at which the human planner must be able to control the planning process is of importance. For subtasks that do not need interference by the planner, the computer can apply whatever algorithms are most efficient. For sub-problems that need interference, however, intermediate results must be communicated to the human planner for he must be able to interfere in the search process. Then, the search operators that can be used by the computer are limited by the state space of the human planner.

The mixed-initiative approach opens another way in which algorithms can be used. Because the human is always there to make the decisions, the algorithms can focus on analyzing the problem space and giving feedback to the planner about the decisions he makes. In this way, the planner gets feedback about the choices that he makes with functions or algorithms that can analyze partial plans.

*The paradox of task support*

The discussion about whether or not to support a subtask by generation techniques circles around the cognitive limitations of the human planner, the need for control of the human planner, and the semantic distance between a new way and the current way of working. The argument of task support contains a paradox. Computer programs should be adjusted to the task performance of a planner, but the task performance is not static. Computer support changes the task of the planner. This has an important implication. If task support changes the task, it has to be redesigned to adapt to the new way in which the task is performed. For example, if task support
relieves the planner of some time intensive tasks, the planner will be able to spend more time on other components of his task. The changes in the task performance will outdate the previous appropriate way to support the task. Appropriate task support would have to incorporate the new way in which the task is carried out. If learning strategies of planners that are caused by the introduction of task support would be known, the planning system could incorporate the various stages that planners go through when they start using task support. Unfortunately, theories or models to incorporate this kind of learning in task support beforehand do not exist yet.

Assessment of the paradigms
We outlined three categories of generative scheduling support. We can not unequivocally say which approach is best. Much depends on characteristics of the problem domain, the frequency of planning decisions, time pressure on planners, the stability of the environment, etc. There has been some experimental research to determine the performance (e.g., number of due date violations, idle time of machines) of human versus computer with respect to the scheduling task. Sanderson (1989, p. 661) states that “many studies have compared human scheduling performance with that of simple priority rules. This usually makes human superiority easy to demonstrate”. Of course, schedule generation can be more sophisticated than the use of simple priority rules, but there are more authors that report the superiority of human planners. Experiments of Schartner & Pruett (1991) show that (in their setting) an approach in which the human scheduler makes the decisions yields better results than approaches where the computer makes some or all of the decisions.

Still, there are some generic guidelines when the computer can be used without human interference. Tasks with the following characteristics are appropriate to be performed by an algorithm: routine tasks with little uncertainty and a clear and unequivocal goal. Evident examples are counting, sorting, and calculating. Actual assignment tasks can also partly or totally comply with these characteristics. Tasks that require expertise, judgment, or communication, require human interference. Analysis of the tasks should show the decisions that the computer can make on its own, the decisions in which the computer has no part, and everything in between. From the perspective of the division of tasks between the human and the computer and the resulting requirements of communication, there are three levels:

- Level of human decision making: tasks that are done solely by the human planner
- Level of solitary search: tasks that are done solely by the computer
- Level of mixed initiative: tasks that are done by both the human and the computer

For the first two categories, only the end results of the tasks have to be communicated from human to system and vice versa. For the latter category, however, intermediate results should also be transferred because both the computer
and the human planner need to understand the reasoning process that lies behind a decision.

Benbasat & Todd (1996) and Prietula et al. (1994) show that generation techniques can be understandable and acceptable to the user even if they are not based on symbolic information processing. Therefore, domain based techniques seem to be at an advantage because they can compute better solutions in most cases. But such techniques should only be used for tasks of which the human planner does not need intermediate results. Generation algorithms can also be applied at subtasks on the level of mixed initiative. Preferably, algorithms at the level of mixed initiative need to be able to communicate about the search process to the planner in a meaningful way. This can be reached if an algorithm to perform a subtask at the level of mixed initiative uses algorithms that are made for subtasks of the subtask under consideration. Then, the algorithm uses the same partitioning of the task as the planner and intermediate solutions are always understandable by the human user. So, in a way, generation tasks that are performed by both human and computer are decomposed until only subtasks remain that either can function as a black box for the human or are performed without the help of the computer.

3.6. Conclusion

In this chapter, we provided a review of a number of aspects that are involved in planning problems in organizations. The nature of this review is rather abstract in the sense that it does not provide many specific pointers to planning in the food processing industries. Chapter 5 and 6, however, will apply the generic observations to our research domain.

The review in this chapter contains two somewhat integrated purposes. First, a literature overview describes various aspects of planning and planning support from a task oriented perspective. Second, we have tried to fill some deficiencies in literature. Although these two are intermingled in the contents of the chapter, they will be discussed separately here.

Figure 3.10 contains an overview of the elements that were discussed in this chapter. Although all elements are related somehow, the arrows in the figure express the most important relations. There is great variety in the amount that is known about the various planning aspects that we discussed. Generation techniques have had a great deal of attention of researchers, most notably in operations research. Discussions about planning domain ontologies, mixed initiative approaches, and user interfaces for scheduling support also got a fair deal of attention. Planning from a task perspective, organization of the planning, and integrated approaches for planning organization and planning support are clearly lacking behind. Especially causal relations between aspects of planning are missing from theory. In other words, we can not base statements about organizational arrangements, task performance, and planning support on an integrated theory about planning. This will be taken up in the following chapters.
As a step towards integration of the planning approaches, we have analyzed how task models can be used to determine where generation algorithms can be used. The following findings in this chapter contribute to this subject:

- The recursive planning paradox makes clear that algorithms from Artificial Intelligence that are based on cognitive models of planning need careful consideration before they can be applied in planning support.
- A planning task consists of subtasks. Each subtask needs its own form of generation support.
- Three kinds of generation techniques can be distinguished: domain oriented, problem solving oriented, and task oriented.
- There are three ways to provide generation support for a planning subtask: solitary search by the computer, human decision making, and mixed initiative plan generation.
- A subtask in a hierarchy of subtasks at which the human planner needs control of partial solutions, must not be supported by a solitary search algorithm.

A complicating factor in the formulation of these issues is that a task (i.e., the division of the task in subtasks and the task strategy) changes due to the introduction of planning support. The paradox of task support states that the introduction of
planning support is not a one time activity but that *learning* should be taken into account.

The integration issue will be taken up in the next chapter, where a framework with models is constructed with which the various elements of planning and their relations can be described.