Chapter 15

Advanced Concepts in Systems Design

15.1 Introduction ........................................... 373
15.2 Robustness and Reactive Decision Making .... 374
15.3 Machine Learning Mechanisms .................. 379
15.4 Design of Planning and Scheduling Engines
and Algorithm Libraries ............................. 385
15.5 Reconfigurable Systems ......................... 388
15.6 Web-Based Planning and Scheduling Systems .. 390
15.7 Discussion ........................................... 393

15.1 Introduction

This chapter focuses on a number of issues that have come up in recent years in the design, development, and implementation of planning and scheduling systems. The next section discusses issues concerning uncertainty, robustness and reactive decision making. In practice, plans or schedules often have to be changed because of random events. The more robust the original plan or schedule is, the easier the replanning or rescheduling process is. This section focuses on the generation of robust plans and schedules as well as the measurement of their robustness. The third section considers machine learning mechanisms. A system cannot consistently generate good solutions that are to the liking of the user. The decision-maker often has to tweak the plan or schedule generated by the system in order to make it usable. A well-designed system can learn from adjustments made by the user in the past; the mechanism that allows the system to do so is typically referred to as a learning mechanism. The fourth section focuses on the design of planning and scheduling engines. An engine often contains a library of algorithms and routines. One procedure may be more appropriate for one type of instance or data
set, while another procedure may be more appropriate for another type of instance. The user should be able to select, for each instance, which procedure to apply. It may even be the case that a user would like to tackle an instance using a combination of various procedures. This fourth section discusses how a planning or scheduling engine should be designed in order to enable the user to adapt and combine algorithms in order to achieve maximum effectiveness. The fifth section focuses on reconfigurable systems. Experience has shown that the development and implementation of systems is very time consuming and costly. In order to reduce the costs, efforts should be made to maintain a high degree of modularity in the design of the system. If the modules are well designed and sufficiently flexible, they can be used over and over again without any major changes. The sixth section focuses on the design aspects of web-based planning and scheduling systems. This section discusses the effects of networking on the design of such systems. The seventh and last section discusses a number of other issues and presents a view on how planning and scheduling systems may evolve in the future.

15.2 Robustness and Reactive Decision Making

In practice, it often happens that soon after a plan or schedule has been generated, an unexpected event happens that forces the decision-maker to make changes. Such an event may, for example, be a machine breakdown or a rush job that suddenly has to be inserted. Many planners and schedulers believe that in practice, most of the time, the decision making process is a reactive process. In a reactive process, the planner or scheduler tries to accomplish a number of objectives. He tries to accommodate the original objectives, and also tries to make the new plan or schedule look, as much as possible, like the original one in order to minimize confusion.

The remaining part of this section focuses primarily on reactive decision making in short term scheduling processes. The number of random events that can occur in a short term may, in certain environments, be very high. Rescheduling is in many environments a way of life. One way of doing the rescheduling is to put all the operations not yet started back in the hopper, and generate a new schedule from scratch while taking into account the disruptions that just occurred. The danger is that the new schedule may be completely different from the original schedule, and a big difference may cause some confusion.

If the disruption is minor, e.g., the arrival of just one unexpected job, then a simple change may suffice. For example, the scheduler may insert the unexpected arrival in the current schedule in such a way that the total additional setup is minimized and no other high priority job is delayed. A major disruption, like the breakdown of an important machine, often requires substantial changes in the schedule. If a machine goes down for an extended period of