Chapter 3

Scheduling and Release Planning

Additional Recommended Literature

- Joseph Y.-T. Leung: Handbook of Scheduling. CRC Press 2004.
- For Release Planning see the lectures of Dr. Ruhe, Calgary.

The Problem (1)

- Scheduling is the problem of assigning a set of tasks (sometimes also called features or requirements) to a set of resources subject to a set of constraints.
- Examples of scheduling constraints:
  - deadlines (e.g., job i must be completed by time t)
  - precedence constraints on the order of tasks (e.g., a design must be done before implementation)
  - Coupling constraints: Task1 and task2 are coupled (either by technical reasons or by customer demands)
- These constraints are of local character, i.e. they are defined between a fixed number of arguments (e.g. unary or binary constraints)

The Problem (2)

- The tasks require resources.
- Resource capacities are usually limited, e.g., there are only four experts for image processing.
- This limits the parallel execution of tasks.
- Special resources are time slots.
- These types of constraints are called global because they can refer to an arbitrary number of arguments

The Problem (3)

- Schedules that satisfy the constraints are called feasible solutions or simply solutions.
- Feasible solutions may have different qualities, e.g.:
  - priorities on tasks (e.g., finish job j as soon as possible) are more or less met
  - costs are more or less high
  - benefits are more or less high
- This leads to a multi-dimensional optimization problem

Participants

- The following human agents participate in scheduling:
  - The project manager: responsible for the schedule
  - The customers
  - The stakeholders: Persons who give advice and opinions about importances of tasks and customers. They have contact to the customers.
- The project manager may be supported by software agents like
  - Scheduling and/or release planners
  - Optimizations programs
  - Software for determining or estimating durations, costs
  - etc.

Releases (1)

- Release planning is a special kind of scheduling. Two steps take place:
  - The tasks are sequentially ordered
  - The ordered tasks are cut into intervals.
- A release is a package of tasks. The definition of releases is done in terms of available resources, it then defines an interval of tasks.
- The company considers the release as a set, all tasks in the release are fulfilled in a certain amount of time. Inside of a release an additional scheduling will take place. The number of releases to be considered in advance may vary from case to case.
- The definition of the releases is called release planning.

Releases (2)

- Releases are defined incrementally in order to keep up with unforeseen events: Release planning can be performed interactively!
- For simplicity and without loss of generality, we will only consider two releases in advance. Consequently, as a result of release planning, each requirement is assigned to exactly one of three possible cases:
  - next release (option 1),
  - next but one release (option 2), or
  - postponed or not (yet) considered for implementation (option 3).

Releases (3)

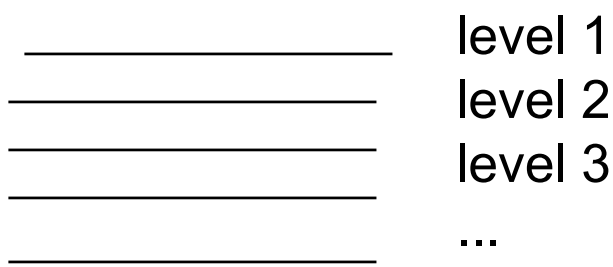
- Releases have a starting date and an end date.
- For each task in a release the deadline of a task is met if it is after the end date of the release.
- Release planning has to respect the constraints; in particular: if there is a coupling constraint between tasks they have to be in the same release.
- Solutions to release planning have again different qualities.

Hard and Weak Constraints

- CSP's are often over specified, i.e. they have no solution
- In many situations where no solution exists one is willing to remove certain constraints in order to obtain a solution. This leads to the following definitions:
- Def.: A constraint is
  - *hard* if it cannot be violated
  - *weak* if it may be violated.
- Weak constraint are desirable properties which are not absolutely necessary.
- The task is to satisfy "as many weak constraints as possible" we need precisions of what "*more or less weak*" means: This results in an optimization problem

Weak Constraints

Hierarchies



Constraint on the same level have the same degree of importance.
Each level on a higher degree is more important than all constraints of lower degrees.

Point evaluations

Each constraint has an integer associated to it. These values are added up: Higher values mean higher importance.

Fuzzy constraints

Values for the variables may satisfy a constraint to some degree (a real number between 0 and 1). Higher degrees mean better fulfillment. These constraints can be interpreted as cost functions.

Constraints Revisited (1)

- A constraint, hard or weak, may be a general law or fact, e.g.
  - If an object oriented language is required then COBOL is excluded .
  - The complexity of a certain problem is quadratic.
- A constraint, hard or weak, may be normative, i.e. It is the consequence of a certain decisions of someone, e.g.
  - The importance of a task is very high.
- From the term "hard" one cannot see whether it is normative or factual.
- Some constraint may be a consequence of another constraint. This defines a dependency relation among constraints and leads to the notion of a *dependency graph*:
  - The nodes are constraints and the edges indicate dependencies.

Constraints Revisited (2)

- Hard factual constraints can never be omitted.
- Normative constraints (i.e. decisions) may be revised:
  - They may be omitted
  - They may be converted into weak constraints.
- In order to remove a hard constraint one has therefore backtrack in the dependency graph until one reaches normative constraints.
- Such a constraint may be discussed with the user.
- It may happen that one such constraint is again a consequence of other constraints. Then the search has to be continued. This corresponds to the search for an “authority” which is willing or able for a constraint revision.

Estimates

- Attributes values may be based on
  - objective facts, e.g.
    - A certain task requires a specific data base
  - estimates, e.g.
    - The implementation of some GUI requires three person weeks.
- An objective fact cannot be changed but an estimate is always up to revision.
- Such a revision will lead to a change in a process or in a plan.
- Revisions often take place if some constraints are violated.

Resources (1)

- Resources are prerequisites for actions and processes like money, persons, tools etc.
- The management for resources is a planning task in itself.
- Actions g:
  - The needed resources are listed in the preconditions of g.
  - Each action undertakes acquisition steps for acquiring the needed resources.
  - The manager distributes the resources
- At each time there is a subset of all resources, the *available resources*; all other resources are *blocked*.
-

Resources (2)

- There are different kinds of resources:
  - Usable resources: They can be used by other actions when they are released after a blockade:
    - Typically: Humans
    - These resources are *released* after the action is executed.
  - Consumable resources: They cannot be reused because they are consumed during the execution of an action.
    - Typically: Money
  - Resources which can be shared: They are not blocked by acquisition actions.
    - Typically: Software tools

Resources (3)

- The resources are grouped into types.
- Human agents have abilities and skills; once they are assigned to a process they are blocked. Each skill is associated to a resource type. For each skill there is a fixed number of agents: They determine the availavle resources.
- Available resources give rise to the definition of global constraints.
- Software agents can usually be shared by different processes. They do not generate global constraints.
- However, if software agents are not present and must be purchased they contribute to the budget and this is limited.

Resource Planning

- Suppose a set of tasks is given.
- Determine for each task t the needed resources R(t).
- The available resources have to be assigned to the needed set R(t).
- Optimal resource planning makes best use of the available resources.
- Resource Planning is part of scheduling and release planning.

Optimization

- Optimization is a mathematical problem of the form

  Maximize a real valued function f
  Subject to a set of constraints C

- For planning f is defined on plans P.
- The function have has to reflect the interest of the user, in particular the weak constraints.
- A plan that just satisfies the constraints will be called a solution to the planning problem that is, howver, in general not optimal.

Constraints in Release Planning and Scheduling (1)

- We consider the following local constraints:
- 1) Precedence constraints: X before Y
    - hard, factual
- 2) Coupling constraints: X and Y in the same release
  - if factual then hard
  - if normative hard or weak
- 3) Pre-assignments: The project manager pre-assigns X in release i or to a certain place in the schedule
  - normative, usually hard
- 4) Deadlines: Have to be respected by the schedule or after the end of the release
  - normative, hard or weak, may be connected with additional costs

Constraints in Release Planning and Scheduling (2)

- We consider some global constraints like:
  - Budget amount
  - Total amount of a specific resource („we have two specialists for image processing“)
  - Total amount of human resources
- In a simplified version we consider a magnitude called ***effort***. The effort for a certain release cannot be overstepped. Usually we can think of „available human resources“ or „available person days for the release time“.

Stakeholders and Voting

- Stakeholders have opinions about importance of tasks.
- This partially a consequence of there own involvement („my stuff is important“) and partially a consequence of their knowledge („how long does it take“? , „When is it needed“?)
- In particular, often stakeholders think in terms of costs (positive and negative ones). These costs cannot always be specifies numerically, e.g. „this is a reference customer“.
- The stakeholders express their opinions in terms of votes where they give a number as an evaluation to the different alternatives.
- The voting gives rise to some objective functions.

The Objective Function: Version 1

- Stakeholders vote on the importance of tasks.
- Example: Stakeholders are given the option of rating each task from 0-9.
- Each release R has an importance imp(R)
- Each stakeholder s_j is a given a weight from 0-9. This weight determines how much influence the stakeholders' votes have.

The objective function to be maximized is:

$$f(P) = \sum_{i,j} \text{weight}(s_j) \cdot value(s_j, task_i) \cdot imp(R(task_i))$$

Here P denotes the release plan and $R(task_i)$ is the release of $task_i$.
Remark: The objective functions are often normalized to the unit interval.

Example

Imp(R1) = 0.7
imp(R2) = 0.3
R1= {task1}, R2 = {task2, task3}
One stakeholder s with weight 0.5

	Value
Task 1	9
Task 2	0
Task 3	4

f(P) = 0.5*(0.7*9+0.3(0+4)) = 3.75

The Objective Function: Version 2

- Here the voters are interested in urgency. They vote for priorities. The vote says how satisfied they are if the tasks ended up in a certain release.
- A vote v of stakeholder s can be represented in the form $task_i$ = (s, v(Release1) = n, v(s, Release2) = m)
- The objective function then is

$$f(P) = \sum_{i,j} \text{weight}(s_j) \cdot v(s_j, R(task_i)) \cdot imp(R(task_i))$$

Example (extended)

- Stakeholder s votes:
- v(task1, release1) = 1, v(task1, release2) = 8
- v(task2, release1) = 3, v(task2, release2) = 6,
- v(task3, release1) = 0, v(task3, release2) = 9

Imp(R1) = 0.7
imp(R2) = 0.3
R1= {task1}, R2 = {task2, task3}
One stakeholder s with weight 0.5

f(P) = 0.5(8*0.7+6*0.3+9*0.3) = 5.05

The Objective Function: Version 3

- This objective function is concerned with the costs for overstepping deadlines.
- Each release R has an end date e(R) and each task t has a deadline d(t). Here the objective function has to be minimized:
- Define cost(s_j, R($task_i$)) = 0 for e(R($task_i$) < d($task_i$)
  and cost(s_j, R($task_i$)) = value(s_j,$task_i$)*(e(R($task_i$) - d($task_i$))
  otherwise (time is measured in days) and:

$$f(P) = \sum_{i,j} \text{weight}(s_j) \cdot \text{cost}(s_j, R(task_i))$$

This means: Costs increase with value of the task and delay time. Of course, one can think of more involved cost functions.

Combining Objective Functions

- Often, more then one objective function has to be considered.
- Then these function have to be amalgamated into one function.
- There are various ways to do it. A simple example is a weighted sum:

$$F(f1,f2,\ldots,fp)(P) = \sum_{j=1}^{p} \omega_j \cdot f_j(P)$$

Here the weights reflect the importance of the individual objective functions for the overall problem.
This is acceptable if the functions are independent.
However, there may be dependencies among the functions.

Example

Requirements	Developer(6)	Project Manager(4)	User(7)
R 1 - customer care system enters customer details into its customer database	2, 4, 3	0, 5, 4	4, 3, 2
R 2 - order management system interrogates its customer database	6, 3, 0	4, 5, 0	2, 7, 0
R 3 - order management system scans order database	4, 4, 1	5, 4, 0	5, 2, 2
R 4 - order management system looks up customer information from its database	7, 0, 2	3, 5, 1	4, 3, 2
R 5 - order management system makes entries into database in the billing system	6, 3, 0	6, 1, 2	8, 1, 0
R 6 - billing system gets billiing history	7, 2, 0	6, 3, 0	2, 2, 5
R 7 - billing system retrieves billing records for customers	8, 1, 0	6, 3, 0	1, 2, 6
R 8 - customer care translator gets customer details	9, 0, 0	7, 2, 0	2, 1, 6
R 9 - customer care translator takes user details and updates to its database	5, 3, 1	3, 5, 1	4, 5, 0
R 10 - order management translator retrieves new customers that have placed order	7, 2, 0	9, 0, 0	2, 5, 2
R 11 - order management translator updates its customer records	7, 1, 1	3, 4, 2	2, 5, 2
R 12 - order management translator retrieves all the order for given customer	6, 3, 0	6, 2, 1	7, 1, 1
R 13 - billing translator is asked to start a new billing	7, 1, 1	7, 2, 0	3, 2, 4
R 14 - billing translator asks the billing system to get billing history	7, 1, 1	6, 2, 1	1, 3, 5
R 15 - the system should be only used by the users with defined privileges	1, 3, 5	0, 3, 6	7, 0, 2
R 16 - the system should be consistent in the user interface	3, 6, 0	3, 5, 1	3, 6, 0
R 17 - the system should provide online and context-sensitive help	6, 2, 1	6, 3, 0	5, 2, 2
R 18 - the system should generate error messages when invalid data is entered	7, 0, 2	3, 6, 0	6, 3, 0
R 19 - the system should have 99% reliability during operating hours	7, 1, 1	6, 1, 2	6, 2, 1
R 20 - system's response time should be less than 5 seconds	5, 2, 2	2, 3, 4	1, 3, 5
R 21 - system's recovery time should be less than 10 minutes	1, 6, 2	3, 5, 1	1, 2, 6
R 22 - Easy to maintain customer information in customer database	5, 4, 0	2, 7, 0	5, 3, 1
R 23 - the system should be easy to install	5, 3, 1	3, 6, 0	2, 2, 5

Project Constraints

Requirement		Requirement
R 3 - order management system scans order database	must precede	R 12 - order management translator retrieves all the order for given customer
R 7 - billing system retrieves billing records for customers	must precede	R 14 - billing translator asks the billing system to get billing history
R 9 - customer care translator takes user details and updates to its database	must precede	R 1 - customer care system enters customer details into its customer database
R 4 - order management system looks up customer information from its database	must precede	R 11 - order management translator updates its customer records
R 5 - order management system makes entries into database in the billing system	must precede	R 13 - billing translator is asked to start a new billing
R 1 - customer care system enters customer details into its customer database	is coupled to	R 22 - Easy to maintain customer information in customer database

Requirement Pre-assignment

Requirement		Increment
R 1 - customer care system enters customer details into its customer database	is assigned to	1

Plan "EAI test Plan 1" - Generated on 2004-04-07 14:20:36

ncrement 1	Effort
1 - customer care system enters customer details into its customer database	25.00
5 - order management system makes entries into database in the billing system	30.00
6 - billing system gets billing history	20.00
7 - billing system retrieves billing records for customers	20.00
8 - customer care translator gets customer details	25.00
9 - customer care translator takes user details and updates to its database	45.00
11 - order management translator updates its customer records	20.00
13 - billing translator is asked to start a new billing	15.00
14 - billing translator asks the billing system to get billing history	20.00
18 - the system should generate error messages when invalid data is entered	25.00
22 - Easy to maintain customer information in customer database	55.00
Effort Used: 300.00 / 300.00	

ncrement 2	Effort
2 - order management system interrogates its customer database	35.00
3 - order management system scans order database	45.00
10 - order management translator retrieves new customers that have placed order	40.00
12 - order management translator retrieves all the order for given customer	35.00
15 - the system should be only used by the users with defined privileges	20.00
16 - the system should be consistent in the user interface	25.00
17 - the system should provide online and context-sensitive help	30.00
20 - system's response time should be less than 5 seconds	30.00
23 - the system should be easy to install	40.00
Effort Used: 300.00 / 300.00	

ostponed Requirements	
4 - order management system looks up customer information from its database	40.00
19 - the system should have 99% reliability during operating hours	65.00
21 - system's recovery time should be less than 10 minutes	60.00
Effort remaining: 165.00	

Involved Uncertainties

- The basis of the optimization procedure has several uncertainties involved, e.g.:
  - The voting of the stakeholders is based on personal opinions
  - The estimates on the duration of the tasks (i.e. the time needed) is often a pure guess.
- There are two major ways to deal with these aspects:
  - Give upper and lower bounds for the uncertain values
  - Interact and change the plan if new insights come up.
- The last way is in the spirit of interactive planning. It requires a notification to the responsible agent if an event with such a new insight comes up.

Optimization Revisited

- 1) The first aspect of a problem is the naïve goal, i.e. in which aspects are we primarily interested? (the ”Importance”). This goal can be formulated as:

Achieve an optimal degree of satisfaction for the customer.

- 2) Question: Can we formulate for a customer an objective function such that their optimal realization provides this degree of satisfaction?
- This is usually not possible and even the customer cannot do it precisely because the problem is too complex and difficult to overlook.
- Mostly, little attention is paid to this problem. People soon arrive at the mathematical problem and the original intentions are forgotten or neglected.

Interactive Scheduling

- As in general process planning for scheduling and release planning the planning and the execution can be interleaved.
- An example is incremental release planning: Only one or two releases are executed, the rest of the planning is post poned.
- Interactive release planning allows to respect new event, e.g. one can correct estimates of needed resources (e.g. time needed) and an adaptation of the objective function.
- This requires some form of communication between the planner and the project manager.
- We will return to this in the chapter on explanation.