Sven Tackenberg, Sönke Duckwitz, Christopher M. Schlick

An Actor-Oriented Model of a Service Provision

The development of valid and efficient plans for service provision is a critical success factor for companies. Adequate planning assures the optimal use of resources, customer satisfaction, and the attainment of service targets in an acceptable period of time. Where these plans are incorrect or provide only a small contribution to the objectives of the company, resources are wasted and opportunities are missed. The problem of planning an efficient service provision can be described in the form of tasks that must be scheduled subject to precedence and resource constraints. Consequently, an analysis of existing scheduling models from the perspective of service providers is necessary to use them in the field of service management. The purpose of this paper is to provide a formal description of an actor-oriented model of a service provision that can be used for person-centered simulation. Such a model defines the decision variables and constraints to be determined by a person or a software tool during service management. Finally, we provide insight into the use of a formal model in case of a simulation study.

1 Introduction

The development of valid and efficient plans is an important function for improving the quality of service provision. The number of tasks for a service provision, the presence of weakly structured precedence relations between these tasks and the inclusion of the customer as well as many other actors in the service process make service planning difficult. But, a predominated situation-based service provision without a prospective consideration of the direct and indirect effects of assigning tasks to working persons and the characteristic of processing these tasks appears to be inefficient and generates additional workload. Consequently, one of the key challenges is the development of tools to support the planning of a service provision.

In the discipline of project management (Shtub et al. 2005) algorithms were developed to effectively solve the so-called Resource-Constrained Project Scheduling Problem (RCPSP). The RCPSP represents a planning problem with a quantity of tasks (activities) that must be scheduled subject to precedence constraints and resources with the objective of minimising the makespan. The foundation of such exact and heuristic scheduling procedures is a formal model that prospectively describes the execution of a project by a set of scheduling constraints and an objective function (Hartmann and Briskorn 2010). However, the basic RCPSP model oversimplifies the planning and management of a project. Accordingly, extensions and variants of the RCPSP have been developed, but a transfer and evaluation of these extended models for the specific characteristics of services have not been forthcoming.

Therefore, this paper makes two contributions. First, prior to any simulation studies we present an actor-oriented model of a service provision, which is based on the simulated decision behaviour of working persons during a service provision and the resulting processing of tasks. Since services are mostly immaterial, and multiple stakeholders work together, a realistic modelling of communication, interaction and integration of internal and external working persons as well as the customer becomes the most promising approach for a realistic service planning (Rosenkranz 2008). Second, to improve service planning, we present a Petri net model to describe the dynamic of services and to derive the decision variables of a service provision.

The paper is organised as follows. In the next section, we review the RCPSP and give extensions of modelling a task or an activity, the allocation of resources to tasks and precedence constraints. The model formulation is given in Sect. 3. The model allows a service provision to be defined by a set of decision variables and scheduling constraints. This model is mapped onto a Petri net to simulate the different states of a service provision. Section 4 deals with a simulation study derived from real service projects which represents different variants of the RCPSP.

2 Literature review

During the last two decades, resource constrained scheduling problems have become a standard problem in the field of operations research (Artigues et al. 2008; Firat and Hurkens 2011; Hartmann and Briskorn 2010; Reyck and Herroelen 1999). However, all these models are very generic and limited with regard to their validity in service management. Thus, methods for planning an optimal service provision and their application to real service processes are missing.

Service provision can be seen as a number of interconnected and independent work processes. Simulation provides a way of experimenting with a model of a service provision in order to understand its dynamics under heterogeneous settings of decision variables and constraints. The simulation method usually refers to both the process of developing the model and setting up a simulation study for that model (Greasley 2004). A simulation study usually consists of iteratively processed simulation runs in order to generate data for statistical analysis. Taxonomy for classifying organisational models and the corresponding simulation approaches is subject of the German standard VDI 3633, Part 6. According to this standard, the pivotal point is the simulated level of individual activities within an organisation. In activity-oriented models the simulated dynamic is determined by the modeled activities, whereas in actor-oriented models the simulated behaviour of persons causes the dynamic of the model. Both approaches can be further differentiated according to the level of human behaviour represented in the simulation model: task-centered, personnel-integrated, and person-centered (Tab. 1).

Table 1: Classification of organisational simulation approaches

	Activity-oriented In activity-oriented simulation models the model dynamics are determined by activities.	**Actor-oriented** In actor-oriented simulation models the model dynamics are determined by persons.
Activity-centered	Persons are not explicitly taken into account.	Persons are only considered in an undifferentiated manner.
Personnel-integrated	Persons are considered as trivial resource; an activity can only be processed if the required persons are available.	Persons are considered in the form of queuing models; the model includes basic selection procedures for competing activities.
Person-centered	Persons are considered more detailed; aspects of human behavior such as human errors and skills are considered for processing activities.	Activity processing is controlled by the persons; persons can make autonomous decisions on the basis of behavioral model and properties; skills of the persons are explicitly considered.

2.1 Variants of organisational resource-constrained scheduling problems

The RCPSP is defined as follows: The problem is described by a project which consists of a set of activities $i = \{1, ..., n\}$ to be processed. The duration of an activity i is denoted by d_i. The precedence relations between activities are defined by the set of immediate predecessors of an activity $j \in P_i$. Only if the predecessors are fully processed, i can start without time lag. Each activity i requires r_{ik} units of renewable resource k during each period of processing. The availability of each resource type k in each period is R_k units, $k = \{1, ..., K\}$. The objective of the RCPSP is to find a schedule S, which consists of a set of starting times $(S_1, S_2, ..., S_n)$ for the activities, where $S_1 = 0$ and the precedence and resource-constraints are satisfied in such a way that the schedule length $T(S) = S_n$ is minimised. The activities 1 and n represent 'dummy activities' with the makespan 0. From a computational point of view, the RCPSP is a difficult planning problem because the most problem instances are NP-hard (Bartusch et al. 1988).

The RCPSP makes a substantial contribution to describing the predecessor/successor relationships between activities. This ensures that the chronological order of activities corresponds to the function-logic requirements of the service outcome and process.

The multi-mode RCPSP (MM-RCPSP) is an extension of the RCPSP and allows heterogeneous modes $m = 1, ..., M_i$ to process a specific task (Kolisch and Drexl 1997). Due to different modes, the duration of an activity d_{im} as well as the resource consumption r_{ikm} of a resource k depends on the chosen mode m. An activity i must be processed in one mode and without preemption. Once an activity has started, a changing of the mode is not allowed. Modes are used in the literature to consider different resources, e.g., work equipment, valid for processing the activity. Newer approaches for the solution of the MM-RCPSPs are found by Santos und Tereso; Kadrou und Youness (Kadrou Youness and Najid 2006; Santos and Tereso 2011).

The Multi-Skill Project Scheduling Problem (MSPSP) was originally published by Néron and Baptista (2002). The model was expanded through the introduction of working persons with heterogeneous skills. The requirements of an activity are given by the required abilities and capabilities of working persons for processing an activity. Therefore, all subsets of persons have to be identified that can carry out the activity with regard to the required skill levels (Bellenguez-Morineau and Néron 2008).

Li et al. follow a similar problem setting and use a linear model to solve the MSPSP (Li and Womer 2009; Li et al. 2004). For each activity, requirements regarding heterogeneous skills are defined in the model. However, there is the restriction that a working person can fulfill only one condition (skill requirement) at a time. The objective of the approach is to find an optimal schedule to minimise the makespan and cost due to an optimal assignment of activities to working persons in the short-term and to build up an efficient personnel structure in an organisation in the long-term (Li and Womer 2009).

The Weighted-Multi-Skill Project Scheduling Problem (WMSPSP) of Al-Anzi et al. (2010) consists of working persons with several skills and different proficiencies. Thereby, an activity has specific skill requirements that must be satisfied by the working persons, and the duration d_i depends on the staff's individual skill.

Heimerl and Kolisch (2010) consider the problem of simultaneously scheduling projects of a project portfolio and assigning multi-skilled humans with resource-specific efficiencies to the work. For the processing of the projects, external and internal resources with different skills and different unit costs have to be used. Therefore, the scheduling problem can be separated into the planning levels of project selection, project scheduling, and project staffing. The problem is modeled as a mixed-integer linear program and the objective is to minimise labor costs.

Firat and Hurkens (2011) employ the Multi-Level Skill Requirements Problem (MLSRP) to schedule complex tasks with an inhomogeneous set of resources. The problem being investigated is to assign working persons who possess the necessary capabilities to tasks with multi-level skill requirements. To solve the MLSRP, Firat and Hurkens used a hybrid combination of mixed integer programming models and applied it on maintenance instances provided by France Telecom. The investigated case study leads to the restriction that a set of working persons and activities have to be assigned to teams and these teams must stay together for the duration of a working day. The scheduled non-preemptive activities of a team have precedence constraints and heterogeneous priorities. Further restrictions to be addressed are the limited availability of a working person, the fixed compilation of a team during a working day, and that a team can process only one activity at a time.

All introduced approaches assume that the availability of the resources is constant in each period. Due to restricted working hours, Franck et al. introduce a calendar concept for the RCPSP with activities which can be interrupted (Franck et al. 2001). They use a break calendar, which is described by a binary function $b : \mathbb{R}_{\geq 0} \to \{0, 1\}$. The binary variable shows the value '1' if the next period t is a working period and allows the processing of an activity. Otherwise, $b(t) = 0$ and the activity cannot be continued or started. To handle the extended planning problem, Franck et al. assume that an activity cannot be interrupted during a working period and that an interrupted activity has to be processed in the next available working period (Franck et al. 2001). Furthermore, an activity must always have a minimum processing time before a non-working period can occur. Buddhakulsomsiria and Kim (2006) follow a similar problem setting and propose a calendar concept for the Multi-Mode-RCPSP that permits activity splitting due to a pre-defined vacation schedule for resources. Knechtel and Kempkes (2006) use an ant algorithm to solve the RCPSP with calendar and time lags. A calendar for each renewable resource and the minimal and maximal time lags determine the admissible processing periods of activities. Capacities of renewable resources varying with time to capture the availability of working persons and machines were investigated by Hartmann and Schwindt (Hartmann 1999; Schwindt and Trautmann 2000).

2.2 Evaluation from the perspective of service science

In the previous section we have reviewed the essential variants and extensions of the RCPSP to identify aspects to be used in a service-oriented extension of the RCPSP. The most popular extensions of the RCPSP in the literature are multi-modes, generalised time lags, and objectives based on the net present value (Hartmann and Briskorn 2010).

Specific characteristics of services have been discussed in Fitzsimmons and Fitzsimmons (1999). While the authors fail to agree on a single theory for 'best' work processes as a benchmark for services, the following characteristics of services are generally regarded as important for scheduling: A service process is described by sequences of tasks which are predetermined due to technical and organisational restrictions. Such precedence constraints are covered by the RCPSP model. However, the persons involved in the service process usually have their own preferences regarding the exact sequences. Therefore, functional relations and dependencies of tasks are often refined with respect to the situation and the person. Thereby, the provision of a service heavily relies on individual behaviour. The multi-mode concept can distinguish between a task and different modes of task processing (activities). Therefore, a detailed description of the potential influence of working persons as well as renewable and non-renewable resources on the duration of an activity is provided. The critical element for a realistic model of a service provision is the valid and optimal assignment of tasks to internal and external working persons with specialised expert knowledge and only temporal availability. The multi-skill concept of Firat and Hurkens (2011) can be extended to further describe the interrelationship between working persons and the characteristic of a task-oriented cooperative processing. Due to the time-limited availability of working persons during a working day and the direct impact on providing a service, the multi-mode and the multi-skill concept should be extended by a calendar concept. Furthermore, the amount of time for processing a task cannot be accurately predicted due to the strong influence of situation-based decision making. Therefore, a service manager might be interested in a schedule in which unforeseen events have only a limited effect and overlapping of activities can reduce service time.

The above evaluation shows that the RCPSP and the extensions of the RCPSP are powerful models. But, these separate models have to be integrated into a service-specific model to fully describe the provision of services. Thereby, the modeled service provision must be understood as the result of decisions and actions of working persons. A scheduling of abstract activities is not sufficient. We hypothesise that an integration of the variants of the RCPSP into one universal actor-oriented service model will support the application of the algorithms originally designed for the RCPSP and to improve the planning of a service provision.

3 Formal model for service provision

Based on the criticism expressed in Sect. 2.2, we have integrated and further developed the introduced variants of the RCPSP into an actor-oriented model of a service provision. The proposed model allows the simultaneous scheduling and staffing of different services with multi-skilled internal and external working persons, and an activity processing which is based on the bounded rational behaviour of humans. Our approach differs from the papers reviewed in Sect. 2 as follows: In contrast to the approaches for the MSPSP (Al-Anzi et al. 2010; Bellenguez-Morineau and Néron 2008; Li et al. 2004; Néron and Baptista 2002) we distinguish between a task and an activity. The individual processing of a task is based on qualifications and skill levels. Therefore, it can result in different modes of activities for this specific task. This $1:n$ relation between a task and several activities, combined with effects of heterogeneous skill levels of the working persons on activity processing, leads to a high level of individualisation of services. Our model describes the dynamics of a service provision by the simulation of the behaviour of the involved internal and external working persons, and is a combination of actor- and activitiy-oriented models. Consequently, the representation of the dynamics differentiates our model from all other variants and extensions of the RCPSP. Through the consideration of the behaviour of internal working persons as well as the customer and the possibility to model a participative activity processing, the diversity and interactivity of a service can be completely described. A fundamental assumption of our service model is that each working person can work in each period t only on one task. But, one working person can cover more than one qualification and one competence requirement of a task. This assumption differs from those of Al-Anzi et al. (2010); Li et al. (2004); Néron and Baptista (2002).

A service processing is given by a set of tasks, $\boldsymbol{A}$. We denote by d_i the nominal processing time of task i. All further used symbols and their definitions are presented in Tab. 2.

3.1 Partial model of a working person

The sub-model of a working person covers all persons participating in a service provision, such as the working persons of the service provider and of the subcontractors (internal factor of production) as well as the consumer of a service (external factor of production). In this context, the term ‘qualification’ of a working person describes the correlation between a certified training of a working person (job title) and the formal requirements of processing a task. In addition, the term ‘competence’ is used to describe the cognitive abilities and skills for a situation-specific problem solving. Therefore, ‘competence’ only refers to the technical and methodical expertise and the selection of a course of action as well

Table 2: Symbols and definitions

Symbol	Definition	
$\boldsymbol{A}$	set of tasks $i \in \boldsymbol{A}$	Sets
$\boldsymbol{A_{ap}}, \boldsymbol{A_{aq}}$	set of tasks which must be processed	Sets
$\boldsymbol{A_{op}}, \boldsymbol{A_{oq}}$	set of tasks which may be processed	Sets
$\boldsymbol{A_t}$	set of tasks which can be processed at time t	Sets
$\boldsymbol{AP}$	set of working persons	Sets
$\boldsymbol{K}$	set of competences	Sets
$\boldsymbol{L}$	set of levels of all competences	Sets
$\boldsymbol{M_i}$	set of permissible modes m of task i	Sets
$\boldsymbol{O_i}$	set of overlapping modes for task i	Sets
$\boldsymbol{P_i}$	set of predecessors of task i	Sets
$\boldsymbol{Q}$	set of qualifications	Sets
$\boldsymbol{R}$	set of renewable resources r	Sets
$\boldsymbol{\bar{R}}$	set of non-renewable resources $\bar{r}$	Sets
$\boldsymbol{U_i}$	set of interruptions of task i	Sets
δ_i	degree of completion of task i	Variables
δ_{STpi}	level of adjustment of working person p to a task i	Variables
a_{imb}	estimated time to process task i in mode m (minimum requirements of competences fulfilled)	Variables
a_{STi}	time required to adjust to the task i	Variables
$AK_{im}^{(l,k)}$	matrix, to define the level l of competence k required for processing task i in mode m	Variables
al_{pt}	binary variable, "1" if the person p can organize his or her task pool at time t	Variables
ap_{pt}	binary variable, "1" if the assigned person p can process a task i at time t	Variables
ap_{APit}	binary variable, "1" if all assigned working persons $p \in \boldsymbol{AP}$ can process a task i at time t	Variables
$AQ_i^{(m,q)}$	matrix, to define the qualifications q required for processing task i in mode m	Variables
d_i	duration for processing task i	Variables
d_{ui}	duration of breaks during processing task i	Variables
I_{ip}	importance of a task i for working person $p \in \boldsymbol{AP}$	Variables
k	competence	Variables
K_{KDp}	individual weighting factor of a working person $p \in \boldsymbol{AP}$ if a competence gap exists	Variables
K_p	individual weighting factor of a working person $p \in \boldsymbol{AP}$ for the importance of a task	Variables
K_{STp}	individual weighting factor of a person $p \in \boldsymbol{AP}$ to adjust to a task	Variables
KD_{pi}	competence gap if working person p processes task i	Variables
kk_{pk}	binary variable, 1 if the working person p has the competence k	Variables
$KM_p^{(l,k)}$	matrix, to define the level $l \in \boldsymbol{L}$ of competence $k \in \boldsymbol{K}$ of a person $p \in \boldsymbol{AP}$	Variables
$KV_p^{(1,k)}$	vector, to describe the level $l \in \boldsymbol{L}$ of competence $k \in \boldsymbol{K}$ of a working person $p \in \boldsymbol{AP}$	Variables
l	level of competence	Variables
$l_{imk}^{(min)}$	minimum required competence k for processing task i in mode m	Variables
p	working person	Variables
q	qualification	Variables
qr_{pq}	binary variable, "1" if the working person p has the qualification q	Variables
$QM^{(p,q)}$	matrix, to define the qualifications $q \in \boldsymbol{Q}$ of the working persons $\boldsymbol{AP}$	Variables
r	renewable resource r	Variables
$\bar{r}$	non-renewable resource $\bar{r}$	Variables
r_{irm}	level of consumption of resource r for processing task i in mode m	Variables
$r_{i\bar{r}m}$	level of consumption of non-renewable resource $\bar{r}$ for processing task i in mode m	Variables
RL_r	capacity limit of renewable resource r	Variables
$RL_{\bar{r}}$	capacity limit of non-renewable resource $\bar{r}$	Variables
s_i	binary variable, "1" if all predecessors of task i are sufficiently processed	Variables
t	current time	Variables
T_{day}	working day $T_{day} \in T$	Variables
t_{AAp}, t_{AEp}	start of work and end of work for each working day T_{day} of working person p	Variables
t_{PAp}, t_{Pep}	start of a break and end of a break for each working day T_{day} of person p	Variables
t_i	time t, when $\frac{1}{a_{im}}\sum_{t=1}^{t'} y_{imt} = 1$	Variables
t_w	period until the deadline t_{i_dead}	Variables
u	interruption	Variables
$v_{o_{ij}}$	minimum time lag between task i and task j (start-start relation)	Variables
w_i	urgency of task i	Variables
$\boldsymbol{AP_i}$	set of working persons assigned to task i	Decision Variables
a_{im}	time required to process task i in mode m	Decision Variables
ak_{ipm}	binary variable, "1" if the task i in mode m is assigned to person $p \in \boldsymbol{AP}$	Decision Variables
AP_i^{min}	number of minimum required working persons for task i	Decision Variables
AP_i^{max}	number of maximum required working persons for task i	Decision Variables
I_i	importance of a task i	Decision Variables
m	mode of a task	Decision Variables
o_{ij}	overlapping mode for tasks i and j	Decision Variables
t_{i_dead}	deadline of a task i	Decision Variables
x_{imt}	binary variable, "1" if the task i in mode m is assigned to the task pool of a working person at time t	Decision Variables
y_{imt}	binary variable, "1" if the task i in mode m is processed at time t	Decision Variables
z_{imt}	binary variable, "1" if a task i in mode m can be processed at time t	Decision Variables

as the processing of activities during a service provision. We assume that a set $\boldsymbol{AP}$ of working persons to process the tasks is given:

$$\sum_{q \in \boldsymbol{Q}} qr_{pq} \geq 1, \quad \forall\, p \in \boldsymbol{AP} \quad (1)$$

$$QM^{(p,q)} = \begin{Bmatrix} q_{11} & \cdots & q_{1v} \\ \vdots & \vdots & \vdots \\ q_{n1} & \cdots & q_{nv} \end{Bmatrix}, \quad q_{nv} \in \{0,1\} \quad \forall\, p \in \boldsymbol{AP}, q \in \boldsymbol{Q} \quad (2)$$

$$\sum_{k \in \boldsymbol{K}} kk_{pk} \geq 1, \quad \forall\, p \in \boldsymbol{AP} \quad (3)$$

$$KV_p^k \in \{0, 1, \dots, |\boldsymbol{L}|\}^{|\boldsymbol{K}|} \qquad \forall\, p \in \boldsymbol{AP} \quad (4)$$

$$KM_p^{(l,k)} = \begin{cases} 0, & \text{if} \quad l \leq KV_p^k \\ 1, & \text{else} \end{cases}, \quad KM_p^{(l,k)} \in \{0,1\}^{\mathbb{L} \times \mathbb{K}}$$

$$\forall\, k \in \boldsymbol{K}, l \in \boldsymbol{L}, p \in \boldsymbol{AP} \quad (5)$$

Each working person p has at least one qualification $q \in \mathbf{Q}$ and one competence $k \in \mathbf{K}$ (1, 3). The qualifications and the competences of a working person are given by a qualification matrix (2) and a competence vector as well as a competence matrix (4, 5). The entries of the vector KV_p^k for a working person p describe the level for the competences $k \in \boldsymbol{K}$. The values of KV_p^k are mapped to the matrix $KM_p^{(l,k)}$ to simplify a statement which levels of heterogeneous competences are fulfilled or even exceeded by assigning tasks to one working person or a team of working persons.

For processing a service, tasks have to be assigned to working persons of service providers and their suppliers as well as to the persons of the customer. We assume that the tasks are assigned to a working person if the conditions of a sufficient execution of the predecessor tasks are fully met. This event leads to an appearance of the task in the *task pool of a working person.* A task pool may contain several tasks with varying processing statuses due to a preemptive task processing and uncoupled activities. We assume that a working person can process only one task (activity) at a time, so that a working person has to organise his or her individual task pool.

A working person does not always make rational decisions during the delivery of a service. Working persons of the service provider as well as the customer are prone to regarding short-term tasks to be more important than long-term ones due to the demand of the day-to-day business in an organisation. A higher priority is often assigned when the time to desired task completion continues to greatly decrease. This behaviour is referred to in literature as 'bounded rational behavior' (Tversky and Kahneman 1992). In order to take this behaviour into consideration, the time factor must be included in a prioritisation algorithm. The Temporal Motivational Theory (TMT) of Steel and König (2006) was the foundation for the development of the prioritisation algorithm. The priority that a working person assigns to a task consists of several aspects (Tackenberg et al. 2010a).

First, a working person $p \in \boldsymbol{AP}$ determines the individual value for the 'importance' of a task I_{ip} that results from the reported significance of the task I_i by a supervisor and a person specific weighting factor:

$$I_{ip} = I_i \cdot K_p \qquad (6)$$

If the working person and the supervisor is the same person, I_{ip} equals I_i. Moreover, the temporal aspect during the priority calculation is also considered. The positive effect of task processing is realised when one of the permitted activities for this specific task is fully executed by the task's particular deadline t_{i_dead}. In addition to the time span until the deadline, the makespan still needed for the task must also be considered, since the urgency w_i of a task is determined by the task's already attained degree of processing. The urgency of a task i at time t results from the quotients of the time that must still be invested and the time remaining until the deadline. The remaining processing time can be calculated by the time required a_{im}, and the already reached degree of processing δ_i:

$$w_i = \frac{a_{im}(1-\delta_i)}{t_w} \quad (7)$$

$$t_w = \begin{cases} |t_{i\_dead} - t|, & \text{for} \quad t_{i\_dead} > t \\ 1, & \text{for} \quad t_{i\_dead} = 1 \\ \frac{1}{|t_{i\_dead} - t|}, & \text{for} \quad t_{i\_dead} < t \end{cases} \quad (8)$$

Organising the individual task pool by a working person is based on evaluating the positive and negative aspects of processing a specific task. The negative aspects are measured by the familiarisation of the working person with task δ_{STpi} and the preparation time a_{STi} for processing the task. The individual level of familiarisation increases during the execution of activities and decreases during breaks. The priority of a task is expressed as follows (9):

$$Pr_i = \frac{I_i \cdot K_p}{1 + \Gamma\left(\frac{t_{i_{dead}} - t}{a_{im}(1-\delta_i)}\right)} - \left((1-\delta_{STpi})a_{STi}K_{STp} + KD_{pi}K_{KDp}\right)$$

The processing of a service has a chronological and a chronometric dimension. The former specifies the starting times and the distribution of task processing over the planning horizon and the latter describes the time period of processing each task. Thereby, the starting time of a task processing is determined by the weekly working hours and the availability of a working person. To describe the working and non-working periods of a person, a calendar concept is introduced:

$$ap_{pt} = \begin{cases} 1, \text{ if} & t_{AAp} \le t < t_{PAp} \lor t_{PEp} \le t < t_{AEp} \\ 0, \text{ otherwise} \end{cases} \quad (10)$$

The daily working hours $T_{day} \in T$ of a person are given by the period between the start t_{AAp} and lunchtime t_{PAp} as well as between the end of the break t_{PEp} and the start of the leisure-time t_{AEp} (10).

3.2 Partial model of a task

The service outcome is the result of processing a set of tasks $\boldsymbol{A}$:

$$\sum_{t=1}^{T} \sum_{m \in \boldsymbol{M_i}} x_{imt} = 1, \quad \forall\, i \in \boldsymbol{A_{ap}} \cup \boldsymbol{A_{aq}} \quad (11)$$

$$\sum_{t=1}^{T} \sum_{m \in \boldsymbol{M_i}} x_{imt} \le 1, \quad \forall\, i \in \boldsymbol{A_{op}} \cup \boldsymbol{A_{oq}} \quad (12)$$

The set of tasks $\boldsymbol{A}$ is divided into the subset of tasks $\boldsymbol{A_{ap}} \cup \boldsymbol{A_{aq}}$ that must be processed (11) and the subset of tasks $\boldsymbol{A_{op}} \cup \boldsymbol{A_{oq}}$, which may be processed (12). A task is in the set $\boldsymbol{A_{ap}} \cup \boldsymbol{A_{op}}$ if a preemption of a task processing is allowed at any time. Otherwise, a task is in the set $\boldsymbol{A_{aq}} \cup \boldsymbol{A_{aq}}$ and can be interrupted only by a non-working period of the working persons assigned to this specific task.

The working and non-working periods for a task i are given by a calendar:

$$z_{imt} = \begin{cases} ap_{APit}\, s_i, & if \quad \sum_{t \in T} \sum_{m \in \boldsymbol{M_i}} x_{imt} > 0 \\ 0, & else \end{cases}, \forall i \in \boldsymbol{A} \quad (13)$$

$$ap_{APit} = \begin{cases} 1, if \sum_{p \in \boldsymbol{AP}} \sum_{m \in \boldsymbol{M_i}} ap_{pt}\, al_{pt}\, ak_{ipm} = \sum_{p \in \boldsymbol{AP}} \sum_{m \in \boldsymbol{M_i}} ak_{ipm}, \\ 0, if \sum_{p \in \boldsymbol{AP}} \sum_{m \in \boldsymbol{M_i}} ap_{pt}\, al_{pt}\, ak_{ipm} < \sum_{p \in \boldsymbol{AP}} \sum_{m \in \boldsymbol{M_i}} ak_{ipm}, \end{cases}$$
$$\forall\, t \in T, i \in \boldsymbol{A} \quad (14)$$

$$al_{pt} = \begin{cases} 1, if & \sum_{i \in \boldsymbol{A}} \sum_{m \in \boldsymbol{M_i}} y_{imt}\, ak_{ipm} = 0, \\ 1, if & \sum_{i \in \boldsymbol{A_{ap}} \cup \boldsymbol{A_{op}}} \sum_{m \in \boldsymbol{M_i}} y_{imt}\, ak_{ipm} = 1, \\ 0, if & \sum_{i \in \boldsymbol{A} \setminus \boldsymbol{A_{ap}} \cup \boldsymbol{A_{op}}} \sum_{m \in \boldsymbol{M_i}} y_{imt}\, ak_{ipm} = 1, \end{cases}$$
$$\forall\, p \in \boldsymbol{AP}, t \in T \quad (15)$$

$$A_t = \{i \in \boldsymbol{A} | z_{imt} = 1\} \quad (16)$$

A task i can be processed at time t if the set of predecessors $\boldsymbol{P_i}$ are adequately processed and all required working persons are available: $ap_{APit} = 1$ (13, 14). The availability of a working person p at a time t is determined by the individual working hours ap_{pt} and the current status al_{pt} of p. Therefore, a task can be processed at time t if a working person p does not work on a task $y_{imt}ak_{ipm} = 0$ or the working person processes a task which is allowed to be preempted. If the person processes a task which is not allowed to be interrupted $\boldsymbol{A_{aq}} \cup \boldsymbol{A_{oq}}, al_{pt} = 0$, the person cannot organise his or her task pool, and therefore $z_{imt} = 0$.

The assignment of a task i to a task pool of a working person at time t ($x_{imt} = 1$) and the feasibility of a task i at time t, ($z_{imt} = 1$) are not the same as the processing of task i at time ($y_{imt} = 1$). The point in time for starting the processing is solely determined by the decision-making of the assigned person (9) (see also Sect. 3.1). Therefore, if a task i must be processed by a team of working persons, all persons have to be available at $t : al_{pt}ap_{pt} = 1, \forall p \in \boldsymbol{AP}_i$ and all persons have to select this task for processing. Accordingly, the scope of decision-making of a person is limited to the period of a feasible working period and the individual preferences regarding the tasks within his or her task pool.

To consider renewable resources r and non-renewable resources $\bar{r}$, the concept of modes $m \in \boldsymbol{M_i}$ for processing a task i is used (17, 18). The non-renewable resources have a certain capacity for the planning horizon T, whereas the capacity of a renewable resource is limited to a point of time t:

$$\sum_{i \in \boldsymbol{A}} \sum_{m \in \boldsymbol{M_i}} r_{irm}\, x_{imt} \leq RL_r, \forall\, r \in \boldsymbol{R}, t \in T \quad (17)$$

$$\sum_{t=1}^{T} \sum_{i \in \boldsymbol{A}} \sum_{m \in \boldsymbol{M_i}} r_{i\bar{r}m}\, x_{imt} \leq RL_{\bar{r}}, \quad \forall \bar{r} \in \bar{\boldsymbol{R}} \quad (18)$$

Processing a task i requires qualifications and a certain level of competences of the assigned working person. The qualifications and competence levels of a task are given by a matrix of qualification $AQ_i^{(m,q)} \in \mathbb{R}^{|M||x||Q|}$ and a matrix $AK_{im}^{(l,k)} \in \mathbb{R}^{|L||x||K|}$ for the competence requirements. Entries $AQ_i^{(m,q)} \neq 0$ determine the required number of working persons with the specific qualification $q \in \mathbf{Q}$ to process the task i in mode m. Besides the assignment of resources r to a task i, the selected mode m determines which feasible combination of heterogeneous qualifications and number of working persons is used to process the task i. For example, $AQ_i^{(1,2)} = 3$ describes that a minimum of three working persons each with the qualification q_2, have to process task i in mode m_1.

$$l_{imk}^{(min)} := \begin{cases} \min\{l \mid AK_{im}^{(l,k)} = 1\} \\ 0 \end{cases}, \;\forall\, k \in \boldsymbol{K}, m \in \boldsymbol{M_i}, i \in \boldsymbol{A} \quad (19)$$

$$\sum_{p \in \boldsymbol{AP}} QM^{(p,q)}\, ak_{ipm} \geq AQ_{\mathrm{i}}^{(m,q)}, \forall\, m \mid x_{imt} = 1\,;\; i \in \boldsymbol{A}, q \in \boldsymbol{Q} \quad (20)$$

$$KM_{\boldsymbol{AP_i}}^{\left(l_{imk}^{(min)},k\right)} \geq AK_{im}^{\left(l_{imk}^{(min)},k\right)},$$
$$\forall\, (l,k) \in \boldsymbol{L} \times \boldsymbol{K}, \forall i \in \boldsymbol{A}, m \in \boldsymbol{M_i}, k \in \boldsymbol{K} \quad (21)$$

$$AP_i^{min} \leq \sum_{m \in \boldsymbol{M_i}} \sum_{p \in \boldsymbol{AP}} ak_{ipm}\, x_{imt} \leq AP_i^{max}, \qquad \forall t \in T \quad (22)$$

An entry of $AK_{im}^{(l,k)} \in (0,1]$ represents the minimum required level $l_{imk}^{min} \in \boldsymbol{L}$ of a competence $k \in \boldsymbol{K}$ if the task i is processed in mode m (19). The constraint (20) ensures that the qualification requirements of a task are met without violating the minimum and maximum number of working persons (22). Furthermore, it is evaluated whether the working persons assigned to task i, $p \in \boldsymbol{AP_i}$ fulfill the required level l of each competence k (21).

$$\{l_{imk}^{(max)} = max\{ak_{ipm} \cdot KV_p^{(1,k)}\}, \;\forall\, p \in \boldsymbol{AP}\}, \forall\, k \in \boldsymbol{K} \quad (23)$$

$$a_{im} = \sum_{m \in \boldsymbol{M_i}} a_{imb} \prod_{k \in \boldsymbol{K}} AK_{im}^{\left(l_{imk}^{(max)},k\right)}, \;\forall\, l_{imk}^{(max)} \in \boldsymbol{L} \quad (24)$$

The time needed to process task i in mode m is denoted by a_{im}. The time of each task and mode is not fixed, and varies therefore with the number and expertise of the working persons assigned (24). Exceeding the minimum required level of competence $l_{imk}^{(min)}$ reduces the time a_{imb}. a_{imb} represents the time for processing task i if it is only processed by one or several sufficiently qualified working person with $l_{imk}^{(min)}$ for all required competences. The extent of the reduction results from the contributions of the matrix $AK_{im}^{(l,k)}$. To calculate a_{im}, the maximum level $l_{imk}^{(max)}$ of a competence k of all assigned working persons $p \in \boldsymbol{AP_i}$ is concluded (23).

$$AQ_i = \begin{pmatrix} 1 & 3 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 2 \end{pmatrix} \quad AK_{i3} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0.7 \\ 0 & 0.8 & 0{,}7 \end{pmatrix}$$

$$\sum_{p \in \boldsymbol{AP}} ak_{ip3} x_{i3t} = 2$$

The matrix element $AQ_i^{(3,3)} = 2$ describes that two working persons, each with the qualification q_3 are necessary to process the task i in mode 3. At least one of the two working persons must possess the qualifications q_2 to satisfy the condition (22). The minimum required competences are $AK_{i3}^{(2,2)} = 1$ and $AK_{i3}^{(1,3)} = 1$. Analogously to the description of the requirements for working persons, capacity and the functions of resources can be modeled.

3.3 Precedence constraints

Precedence relations of task i enforce that all tasks in $\boldsymbol{P_i}$ must be completed before the task i is assigned to a task pool of a working person. A task $i-1 \in \boldsymbol{P_i}$ is said to be a predecessor of task i and their relation is denoted by $i-1 \rightarrow i$. Moreover, task i is also said to be a successor of task $i-1$. For processing a sequence of tasks, the following assumptions are made:

- Multiple tasks can be assigned to the task pool of one working person simultaneously. Each working person organises the individual task pool and selects a task for processing in a specific mode.
- A working person can process only one task at time instant t.
- An assignment of a task to a working person remains until the task is fully processed.

Therefore, the sequence of activities in a plan is the result of the fulfillment of precedence constraints and the priority of a task, which is calculated by the working person when organising the individual task pool. Besides the highest priority, the following constraints must be satisfied to include an activity j in the plan:

$$s_j = \begin{cases} 1, if\ v_{o_{ij}} \leq \displaystyle\sum_{m \in \boldsymbol{M_i}} \sum_{t=1}^{t'} \frac{1}{a_{im} x_{imt}} y_{imt} & \forall\ v_{o_{ij}} \leq 1 \\ 1, if\ v_{o_{ij}} \leq \left(\displaystyle\sum_{t=1}^{t'} \sum_{m \in \boldsymbol{M_i}} \frac{1}{a_{im} x_{imt}} y_{imt} + (t' - t_i) \right) & \forall\ v_{o_{ij}} > 1 \\ 0, & otherwise \end{cases}$$

$$\forall\ i,j\ \in \boldsymbol{A},\ i\ \in \boldsymbol{P_j},\ t \in T \quad (25)$$

$$d_i = \sum_{m \in \boldsymbol{M_i}} \sum_{u \in \boldsymbol{U_i}} a_{im} + d_{ui}, \qquad \forall\ i\ \in \boldsymbol{A} \quad (26)$$

The precedence constraints between two tasks are defined by (25). At the time instant t', the binary variable s_j determines whether the task j can be assigned to a working person's task pool $s_j = x_{jmt} = 1$. v_{ijo} represents the minimal time lag between i and j. A value of 0 represents a start-start relation between the tasks and a value larger than one describes a time lag between both tasks. Due to the decision making of working persons and a preemptive task processing, the makespan of a complete processing of task i is given by (26).

If $v_{o_{ij}} < 1$, both tasks overlap, although $i \in \boldsymbol{P_j}$. Overlapping describes the parallel execution of two activities i and j by allowing the activity j to start before the end of its predecessor i based on preliminary information which can cause rework. The assignment of an successor j to a task pool of a working person, which can overlap, is restricted to a finite number of instants $v_{o_{ij}}$ corresponding to the processing stage of the predecessors $i \in \boldsymbol{P_j}$. Each instant refers to an overlapping mode $o_{ij} \in \boldsymbol{O_j}$. Thereby, an overlapping mode describes the rework caused by an overlapped execution of task i and j. Thus, the amount of rework and the point of time for $s_j = 1$ are determined by the assignment of working persons to task i and j, the selected mode m_i as well as the start time of i.

3.4 States of service provision

As already pointed out in the introduction, due to the weakly structured work processes, the provision of a service often cannot be predicted. For a definition of the individual states, the Petri net method is used. A Petri net is a graphical and mathematical modelling method which can be used for a visualisation of a system status. The primary difference between Petri nets and other modelling methods is the presence of tokens which are used to simulate concurrent and asynchronous status changes in a system. The dynamic based on status changes can be represented in a Petri net by setting up state equations, algebraic equations and similar mathematical models (Prashant Reddy et al. 2001). A Petri net is a directed bipartite graph with nodes and arcs (Fig. 1). The nodes represent transitions that can be activated according to the system status and places. A directed arc connects one place with one transition. A direct connection between places or transitions is not allowed. Places in a Petri net may contain a discrete number of marks (tokens). A transition of a Petri net fires, whenever there are sufficient tokens in the input places. When a transition fires, it 'consumes' a predefined quantity of tokens, and places tokens in the output places. A firing of a transition cannot be interrupted.

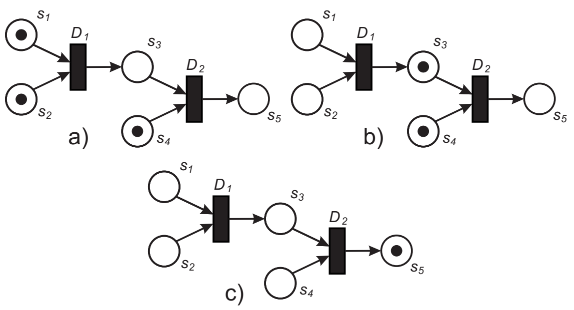

Figure 1: Petri net: a) initial marking, b) marking after firing of D_1, c) marking after firing of D_2

Petri nets have been applied successfully in the areas of business process modelling (Aalst and Stahl 2011; Adam et al. 1998) and service management. Heterogeneous variants of Petri nets are known in the modelling literature: untimed; timed; coloured; stochastic; predicate; priority etc. (Prashant Reddy et al. 2001). Recently, many complex workflows have been analysed using high-level Petri nets involving extensions with 'colour' and 'time' (Kausch et al. 2008). For a further introduction to graphical modelling of workflows using Petri nets, the reader is referred to van der Aalst and Stahl (Aalst and Stahl 2011).

The arcs connecting a place and a transition can represent the precedence constraints between the tasks of a service provision. Each individual change of the state of a Petri net (firing of at least one transition) represents an instance of the incremental production of a service. The introduced constraints (11-26) limit the firing of the transitions and therefore the service provision. To ensure a simple mapping of a scheduling problem to a formal description (Tab. 3), base models with timed Petri net were developed. These models are used in this section to derive the decision variables of a service oriented scheduling problem (SOSP).

The sub-model of task processing (Fig. 2) consists of four transitions (D_1, D_2, A_i, T_1) and five places (s_1, s_2, s_3, s_4, s_5). The arcs $V(f)$ and $L(f)$ link the places and transitions of the model. The place s_1 serves as an interface with all predecessors of the modeled task. If all the predecessors are sufficiently processed, the initial state M_0 is reached and a token is placed in s_1. The transition D_1 is activated and triggers the decrement of a timer value. The initial timer value is specified by the decision variable $v_{0_{ij}}$ and determines when the transition D_1 fires. The token of s_1 is consumed and a new token is set in s_2. The latter represents the assignment of a task to one or several task pools of the involved working persons $|\boldsymbol{AP_i}| = \sum_{p\in \boldsymbol{AP}} \sum_{m\in \boldsymbol{M}_i} ak_{ipm}$ and therefore the conditions of a task processing are created.

Table 3: Parameters of a timed Petri net

Symbol	Definition
$\mathbb{C}(M)$	set of all valid timer-vectors at marking M
d	activation time of a transition
$F(\bullet; M', M, \hat{T})$	distributive function of the timer for the transition t' activated, when a transition in $\hat{T}$ causes a change in marker from M to M'
$N(M', M, \hat{T})$	set of new activated transitions at M', when a transition in $\hat{T}$ causes a change in marker from M to M'
$O(M', M, \hat{T})$	set of activated transitions at M' which has already been activated at M, when a transition in $\hat{T}$ causes a change in marker from M to M'
$S = \{s_1, \ldots, s_n\}$	set of places
$\mathbb{T}$	set of transitions
T'	set of immediate transitions
$T(M)$	set of transitions which are activated at M

If all working persons assigned to a task are available $z_{imt} = 1$ (13, 14) and they all intend to process the task at this stage (9), a token is set to place s_5. A parameter may be assigned to the token which represents $\boldsymbol{AP_i}$. The decision of all members of $\boldsymbol{AP_i}$ regarding a processing of the task leads to a consumption of the token in s_2 and s_5 and a firing of D_2. The token set in place s_3 leads to an activation of the transition A_i. The timer of the activation of A_i is set to the value of a_{im}. The activation of A_i remains until the timer value is reached (full processing of the task) or the token in s_3 is consumed due to a firing of T_1. However, the latter represents an interruption of a task processing due to a non-working period or a modified task selection (due to a modified prioritisation of the task pool) by at least one working person. If the timer of A_i corresponds to the value of aim, A_i fires and a new token is set in s_4. Due to the backward oriented arcs $L(s_2, D_1), L(s_3, D_1)$ as well as $L(s_4, D_1)$ a new activation of D_1 is disabled.

The processing of a task is therefore given by the Petri net:

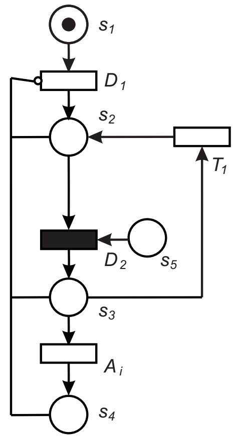

Figure 2: Sub-model of a task processing

$$N = (S, \mathbb{T}, F, V, L, M_0, G);$$

$$S = \{s_1, s_2, s_3, s_4, s_5\}; \quad \mathbb{T} = \{D_1, D_2, A_i, T_1\}; \quad T' = \{D_2\}$$

The weight of each arc is defined by:

$$V(s_1, D_1) = 1;\ V(D_1, s_2) = 1;\ V(s_2, D_2) = 1;\ V(D_2, s_3) = 1;$$
$$V(s_5, D_2) = 1;\ V(T_1, s_2) = 1;\ V(s_3, T_1) = 1;\ V(s_3, A_i) = 1;$$
$$V(A_i, s_4) = 1; L(s_2, D_1) = 1;\ L(s_3, D_1) = 1;\ L(s_4, D_1) = 1$$

$G_1, G_2, G_3 \subset G$ represent the probability distribution of the timers of D_1, A_i and T_1. The firing of these transitions yields the following markings:

$$M_0 = \{1,0,0,0,0\}; \quad M_1 = \{0,1,0,0,0\}; \quad M_2 = \{0,1,0,0,1\};$$
$$M_3 = \{0,0,1,0,0\}; \quad M_4 = \{0,0,0,1,0\}; \quad M'_4 = \{0,1,0,0,0\}$$

The amount of all previously and new activated transitions (definition of the change of status) are defined by the set of activated transitions $T(M)$ which have already been activated at M, and the set of valid timer readings:

$$T(M_0) = \{D_1\}; T(M_1) = \emptyset; T(M_2) = \{D_2\};$$
$$T(M_3) = \{A_i, T_1\}; T(M_4) = \emptyset; T(M_4') = \emptyset$$

$$\mathbb{C}(M_0) = \mathbb{T} = \mathbb{C}(M_1) = \mathbb{C}(M_2) = \mathbb{C}(M_3) = \mathbb{C}(M_4);$$
$$\mathbb{C}(M'_4) = \mathbb{T}/\{A_i\}$$

The processing of a task is specified by the firing of at least one transition and a state change of markers $M \to M'$. Therefore, the state change $M_0 \to M_1$ and all new activated transitions at M_1 is described as follows:

$$N(M_1; M_0, \hat{T} = D_1) = \left(T(M_1) \backslash \left(T(M_0) \backslash \hat{T}\right)\right) \cap \mathbb{C}(M_1)$$
$$= (\emptyset \backslash (\{D_1\} \backslash \{D_1\})) \cap \{D_1, D_2, A_1, T_1\}$$
$$= \emptyset$$

$$N(M_2; M_1, \hat{T} = T_1) = \{D_2\}; \quad N(M_3; M_2, \hat{T} = D_2) = \{A_i, T_1\};$$
$$N(M_4; M_3, \hat{T} = A_i) = \emptyset; \quad N(M'_4; M_3, \hat{T} = T_2) = \emptyset$$

The definition of a change of status requires the set of activated transitions at M_1 which have been activated at M_0:

$$O(M_1; M_0, \hat{T} = D_1) = \left(T(M_1) \cap \left(T(M_0) \backslash \hat{T}\right)\right) \cap \left(\hat{T} \backslash \mathbb{C}(M_1)\right)$$
$$= (\emptyset \cap (\{D_1\} \backslash \{D_1\})) \cap \emptyset = \emptyset$$

$$O(M_2; M_1, \hat{T} = \emptyset) = \emptyset; \quad O(M_3; M_2, \hat{T} = D_2) = \emptyset;$$
$$O(M_4; M_3, \hat{T} = A_i, T_1) = \emptyset; \quad O(M'_4; M_3, \hat{T} = A_i, T_1) = \emptyset$$

The time required for processing a task is defined by the timer value t' and represents the change of markings $M_3 \to M_4$ and $M_3 \to M'_4$ respectively:

$$F(\bullet, M_4, A_i, M_1, \hat{T}) = G_2\,(A_i)(\bullet)$$

The opportunity of an interruption of a task processing is defined by transition probabilities:

$$p(M_4; M_3, \{A_i, T_1\}) \coloneqq p \in [0,1]; \; p(M_4'; M_3, \{A_i, T_1\}) \coloneqq 1 - p$$

The probability distribution is determined by the simulated rational decision behaviour of the working persons. The formal modelling of the change of states of a task processing opens up the decision variables for the simulation model: activation time of transition D_1 (time of assigning a task to at least one person's task pool); time of firing D_2 (fulfillment of the competence requirements of a task processing and selection of this task by the assigned working person due to the results of prioritisation); the time of activating A_i and the duration of an activation (determination of the start time and the makespan of a task processing) as well as the time of the consumption of the token in s_3 (interruption of a task processing due to the decision making of the involved working persons). The precedence constraints with and without overlapping of a sequence of tasks are represented by the combination of several models of a task processing and the specification of the activation time of D_1. The place s_4 of a task equates to the place s_1 of all of its successors.

For the modelling of an alternative (XOR) or a simultaneous task processing (AND) further Petri-net sub-models were developed. These models have to be combined with the Petri net model of task processing. The resulting nets are shown in Fig. 3.

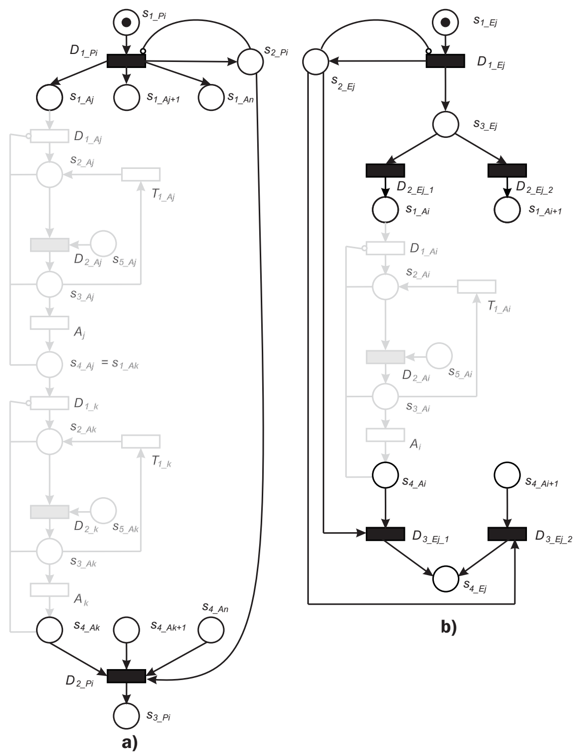

Figure 3: Sub-models of logical dependencies

To model independent paths of different tasks which all have necessarily to be processed, an immediate transition D_{1_Pi} is used (Fig. 3a). This transition is connected with all places $s_{1\_Aj}, j \in \{1, ..., n\}$ of each first task in a path. Therefore, $\boldsymbol{P_j}$ must be identical for all tasks which directly follow an AND branch. The immediate firing of D_{1_Pi} consumes the token in s_{1_Pi} and activates $D_{1\_Aj}, j \in \{1, ..., n\}$ of the respective first task in each path. The adjustment of the timer of each $D_{1\_Aj}, j \in \{1, ..., n\}$ is crucial for modelling an overlapped task processing. If $v_{o_{ij}} = 1$, all tasks j are assigned to a task pool of a working person at the same time. Therefore, a simultaneous processing of these tasks may occur. If the tasks i and j are assigned to the task pools of different sets of working persons $\boldsymbol{AP_i} \cap \boldsymbol{AP_j} = \varnothing$ at heterogeneous times, an overlapped task processing may occur. It should be noted that an overlapped task processing results exclusively from the decision making and the prioritisation of the task pool of the working persons. The sub-model of a simultaneous and overlapped processing of several tasks is given by:

$$
\begin{gathered}
N = (S, \mathbb{T}, V, L, M_0) \\
S = \{S_{1\_Pi}, S_{2\_Pi}, S_{1\_Aj}, ...S_{4\_Ak}, S_{3\_Pi}, \}; \\
\mathbb{T} = \{D_{1\_Pi}, D_{2\_Pi}\} = T' \\
M_0 = \{1, 0, 0, ..., 0, 0\}; \qquad M_1 = \{0, 1, 1, ..., 0, 0\} \\
M_2 = \{0, 1, 0, ..., 1, 0\}; \qquad M_3 = \{0, 0, 0, ..., 0, 1\} \\
T(M_0) = \{D_{1\_Pi}\}; T(M_1) = \{D_{1\_Aj}\}; \\
T(M_2) = \{D_{2\_Pi}\}; T(M_3) = \varnothing
\end{gathered}
$$

The change of markings is given by:

$$
\begin{gathered}
N(M_1; M_0, \widehat{T} = D_{1\_Pi}) \\
= (T(M_1) \backslash (T(M_0) \backslash \widehat{T})) \cap \mathbb{C}(M_1) \\
= (\{D_{1\_Aj}\} \backslash (\{D_{1\_Pi}\} \backslash \{D_{1\_Pi}\})) \cap \varnothing = \{D_{1\_Aj}\} \forall j \in \boldsymbol{A}
\end{gathered}
$$

$$
\begin{gathered}
N(M_2; M^*, \widehat{T} = A_k) = \{D_{2\_Pi}\}, \\
\forall k \in \{1, 2, ..., n\}, k \neq j \\
N(M_3; M_2, \widehat{T} = D_{2\_Pi}) = \varnothing
\end{gathered}
$$

$\mathbb{T} = \{ D_{1\_Pi}; D_{2\_Pi} \} = T'$ proves, that no decision variables are necessary to describe a simultaneous and overlapped processing of tasks assigned to different paths. Thereby, the precedence constraints of tasks within a path are exclusively described by the combination of the sub-models *task processing*.

An overlapped processing of tasks is modeled by the assignment of the tasks to different paths. Each path is connected with the transition D_{1_Pi} and the level of maximum overlapping $d_{D1\_Aj}, j \in \{1, ..., n\}$.

A decision between different paths of tasks which have to be alternatively processed is modeled in Fig. 3b. The competing firing of transitions $D_{2\_Ej\_1}$ and $D_{2\_Ej\_2}$ describes a decision and is defined by:

$$N = (S, \mathbb{T}, F, V, L, M_0, G)$$

$$
\begin{gathered}
S = \{s_{1\_Ej}, s_{2\_Ej}, s_{3\_Ej}, s_{1\_Ai}, s_{1\_Ai+1}, \dots, s_{4\_Ai}, s_{4\_Ai+1}, s_{4\_Ej}\}; \\
\mathbb{T} = \{D_{1\_Ej}, D_{2\_Ej\_1}, D_{2\_Ej\_2}, \dots, D_{3\_Ej\_1}, D_{3\_Ej\_2}, \} = T'
\end{gathered}
$$

$$
\begin{array}{ll}
M_0 = \{1,0,0,0,0 \dots ,0,0,0\}; & M_1 = \{0,1,1,0,0 \dots ,0,0,0\}; \\
M_2 = \{0,1,0,1,0 \dots ,0,0,0\}; & M'_2 = \{0,1,0,0,1 \dots ,0,0,0\}; \\
M_3 = \{0,1,0,0,0 \dots ,1,0,0\}; & M'_3 = \{0,1,0,0,0 \dots ,0,1,0\}; \\
M_4 = \{0,0,0,0,0 \dots ,0,0,1\} &
\end{array}
$$

$$
\begin{gathered}
T(M_0) = \{D_{1\_Ej}\}; T(M_1) = \{D_{2\_Ej\_1}; D_{2\_Ej\_2}\}; \\
T(M_3) = \{D_{3\_Ej\_1}\}; T(M'_3) = \{D_{3\_Ej\_2}\}; T(M_4) = \emptyset
\end{gathered}
$$

The change of markings for an XOR decision with two alternatives is given by:

$$
\begin{gathered}
N(M_1; M_0, \widehat{T} = D_{1\_Ej}) \\
= (T(M_1) \backslash (T(M_0) \backslash \widehat{T})) \cap \mathbb{C}(M_1) \\
= (\{D_{2\_Ej\_1}; D_{2\_Ej\_2}\} \backslash (\{D_{1\_Ej}\} \backslash \{D_{1\_Ej}\})) \cap \varnothing \\
= \{D_{2\_Ej\_1}; D_{2\_Ej\_2}\}
\end{gathered}
$$

$$
\begin{gathered}
N(M_3; M_2, \widehat{T} = A_i) = \{D_{3\_Ej\_1}\}; \\
N(M'_3; M_2, \widehat{T} = A_i) = \{D_{3\_Ej\_2}\}; \\
N(M_4; M_3, \widehat{T}) = \{D_{3\_Ej\_1}\}; = \varnothing \\
N(M'_4; M_3, \widehat{T} = D_{3\_Ej\_2}) = \varnothing
\end{gathered}
$$

The transitions $D_{2\_Ej\_1}$ and $D_{2\_Ej\_2}$ are both immediate transitions. Therefore, both transition compete for the token in place s_{3_Ej}. The conflict is solved by the assignment of priorities to the transitions. Such a priority value can be assigned prior to the previous system states or stochastically. Both forms of assignment can be represented by a random variable which can be interpreted as a stochastic decision variable for the service model:

$$p\left(M_2; M_1, \left\{D_{2_{Ej_1}}; D_{2_{Ej_2}}\right\}\right) := q;$$
$$p\left(M'_2; M_1, \left\{D_{2_{Ej_1}}; D_{2_{Ej_2}}\right\}\right) := r$$
$$mit\ p = q + r = 1$$

3.5 Decision variables

The substantial difference between the introduced variants of the RCPSP and our actor-oriented service model (Tackenberg et al. 2011) is that the priority in which the activities are scheduled depends on the simulated bounded rational decision making of the persons involved in the service process. Consequently, the start time of an activity is determined by the fulfillment of precedence constraints and the individual priority of a task (see Sect. 3.1). Consequently, a scheduling order of tasks cannot be determined before the predecessors of a task have been processed and the task has been placed in the task pool of a working person. Therefore, the indication of relative relations between tasks is necessary. In this paper we use a vector λ of independent random numbers (configuration bank) to describe a blueprint of a plan. The length and the structure of the vector (quantity and position of entries) are the same across all stochastic simulation runs for a specific scheduling problem:

$$\lambda = \left(v_{o_{ij}}\ \xi_j\ \mu_j\ \kappa_j\ I_j\ \theta_j\ m_j\ \dots\ v_{n_{ij}}\ \xi_n\ \mu_n\ \kappa_n\ \theta_n\ I_n\ m_n\ D_A\ D_Z\right)$$

The vector λ includes two classes of configurations with different random variables:

Activity configuration:

- The relative starting time $v_{0_{ij}}$ defines the starting time of activity j in relation to the degree of completion of $\boldsymbol{P_j}$. The definition of the random number is restricted to a permitted range or discrete values.
- The variable ξ_j is used to model uncertainty involved in time and effort estimation for processing task j. If a triangular distribution is used, ξ_j references to a time value for the correction of a_{jmb}.
- The variable μ_j refers to the number of working persons for j and is restricted to a given range: $AP_j^{min} \leq \mu \leq AP_j^{max}$.
- The variable κ_j refers to a unique set of working persons $\boldsymbol{AP_j}$. $\boldsymbol{AP_j}$ is a subset of all feasible combinations of persons for a given μ_j to work on the task j. $\left|\boldsymbol{AP_j}\right|$ corresponds to the value of μ_j.
- The variable I_j can accommodate values between 0 and 1 and represents the importance of the task and the service process communicated to $\boldsymbol{AP_j}$.
- The entry θ_j determines the deviation from a date t and the outcome is the stated deadline for task j.
- The variable m_i corresponds to a mode of a task processing and describes a specific consumption of non-renewable and renewable resources. The occurrence of the variable is restricted to a given set of modes $\boldsymbol{M_i}$ for each task i.

Decision configuration: A configuration D consists of a random number and characterises the outcome of an XOR or OR branch node. D_A is the outcome of a decision regarding branch node A and refers to at least one path of tasks (direct successors of the node) which have to be processed. The specification of the value can depend on rules or a random distribution.

4 Case study

We generated two test instances with 15 tasks each inspired by data from an engineering service provider. The cases describe the preparation to develop and install a small decentralised hydropower plant (Task A), and the dealing with technical questions or claims of customers (Task B). Figure 4 and Tab. 4 (see Appendix) contain the precedence constraints and the parameters of a Multi-Skill Project Scheduling Problem without and with overlapping modes combined with time and personnel expenditure.

4.1 Scheduling Problem

The scheduling task is indicated by a graphical model which represents the precedence con straints between tasks and the logical dependencies. The basic routing elements applied are similar to an UML activity diagram. To model a planning task a service manager can use an editor which is a component of the developed simulation tool. The tool allows for the efficient graphical modelling of precedence constraints between tasks, a standardised requirement definition of task processing as well as the modelling and assignment of working persons to a specific service project. For a further introduction to graphical modelling of planning tasks and the editor, the reader is referred to Tackenberg et al. (2011).

Both investigated cases of service provision consist of 15 tasks, which were combined into three separate 'building blocks' with identical $\boldsymbol{P_i}, \forall i \in \boldsymbol{A}$. Therefore, they have same level of difficulty (size, number of precedence constraints, and boundary conditions by assigning persons to task). The building blocks are only combined in different order (Fig. 5).

We expanded the MSPSP to allow an alternative sequencing of activities (Task A: *A9* to *A12* OR *A13* AND *A14*; Task B: *A3* to *A6* OR *A7* AND *A8*). For the MSPSP, the assignment of tasks to working persons has many degrees of freedom and leads to a large number of activity-processing modes. Each mode requires a minimum and maximum capacity of personnel (e.g., $[2, 2]$) with a certain level l of competence k (e.g., competence 1 with the level: 'medium': K1$[m]$). The requirements for all three planning problems can be obtained from Tab. 4.

We restrict ourselves in this study to the non-preemptive Resource-Constrained Project Scheduling Problem, but we allow an overlapping of activities (Task B: MSPSP-ov). MSPSP-ov is dedicated here to the determination of the optimal overlap amount of two activities under consideration of resource constraints. The start time of an overlapped successor i is restricted to a finite number of instants corresponding to the start time of the predecessor $\boldsymbol{P_i}$ of i, which constitutes different feasible modes for the execution of the overlapping tasks $A8$, $A10$, and $A14$. Each overlapping mode of an activity i is determined by the time period between the start of all tasks in $\boldsymbol{P_i}$ and the start of i, the assigned working persons to i and $\boldsymbol{P_i}$ as well as the time for processing i and its predecessors (e.g., Start-Start: $SS(+1)$; Working person 1: 4h, 440 €) (Tab. 4).

Table 5 (see Appendix) describes the internal and external working persons for processing the service. Both scheduling problems have an identical set of working persons regarding the required competence and the level of competence. The working hours of the employees were set for the study to 8:00-12:00 and 13:00-17:00.

4.2 Results

Both scheduling problems from the previous section can be transferred into a set of combined Petri net modules (Fig. 6). The Petri-net model represents the essential features of the actor-oriented stochastic simulation software for service provision introduced in (Tackenberg et al. 2010b,c).

A solution of the scheduling problem is achieved by a stochastic definition of each decision variable of λ and the mapping of λ to a detailed plan during one simulation run. Therefore, a valid solution of the scheduling problem is completely described by the decision variables of the introduced vector λ. A value of an entry of λ is

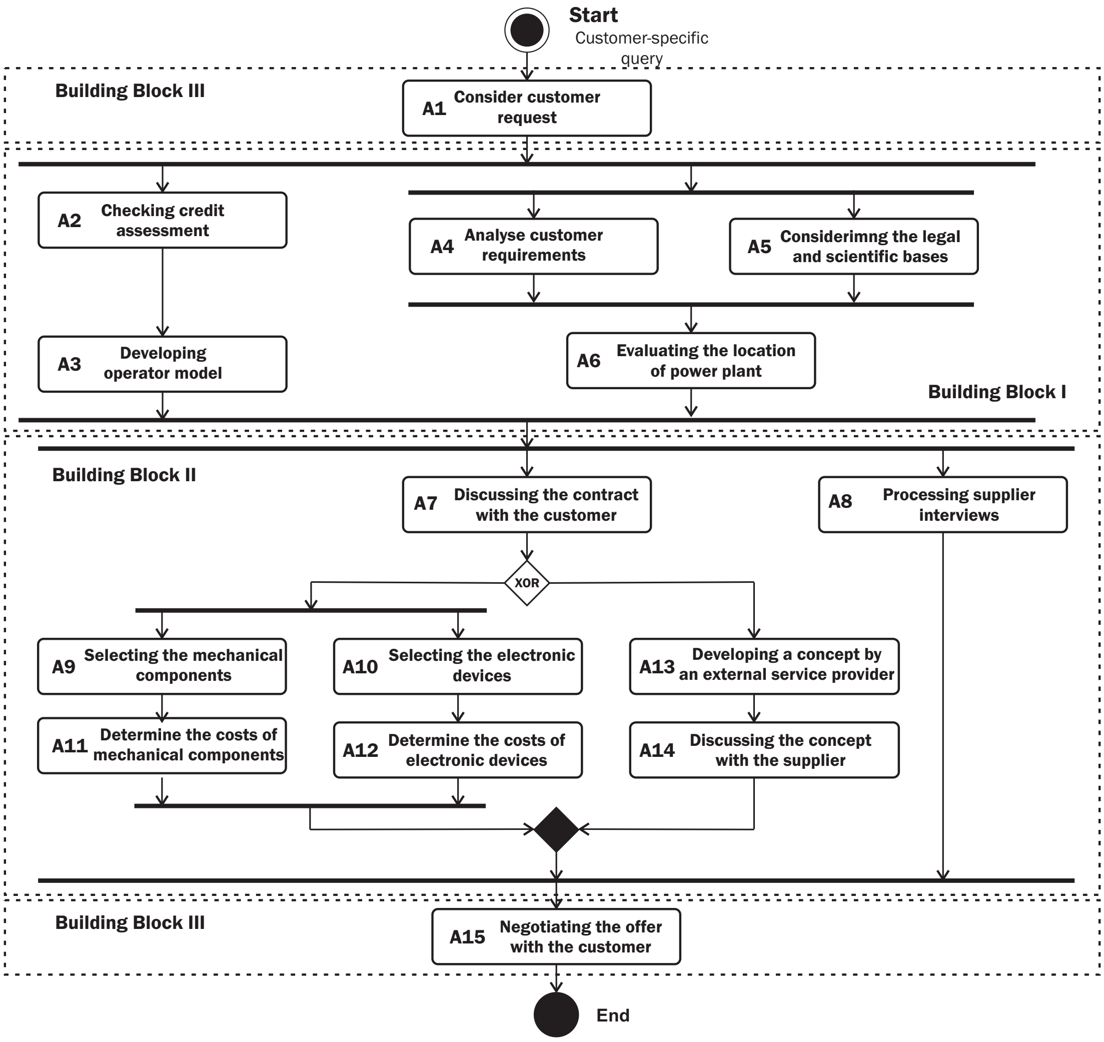

Figure 4: Graphical model of the scheduling problem (Task A)

assigned with the same probability from all valid entries of this parameter.

For the description of the solution space, terminating experiments with discrete-event system simulation are performed (Banks et al. 2010). Due to the experiment design for terminating simulation and the stochastic definition of the entries of λ, multiple replications of a simulation run have to be made (David and Nelson 1998). Therefore, the assignment of one random number seed to λ represents one simulation run. Assigning different seeds to λ, guarantees that the outputs from these simulation runs will be statistically independent. For a fair comparison among both planning tasks (Task A and Task B) the same source of randomness (characteristics of seeds) is used.

Due to independent simulation runs and a parameterisation of each activity configuration without taking into account prior parameterisations, a vector can occur multiple times. To evaluate the quality of λ, during a simulation run λ is mapped under consideration of the restrictions into a detailed plan. To ensure the comparability of λ, the mapping of a seed or a specific vector must always lead to the same plan and therefore

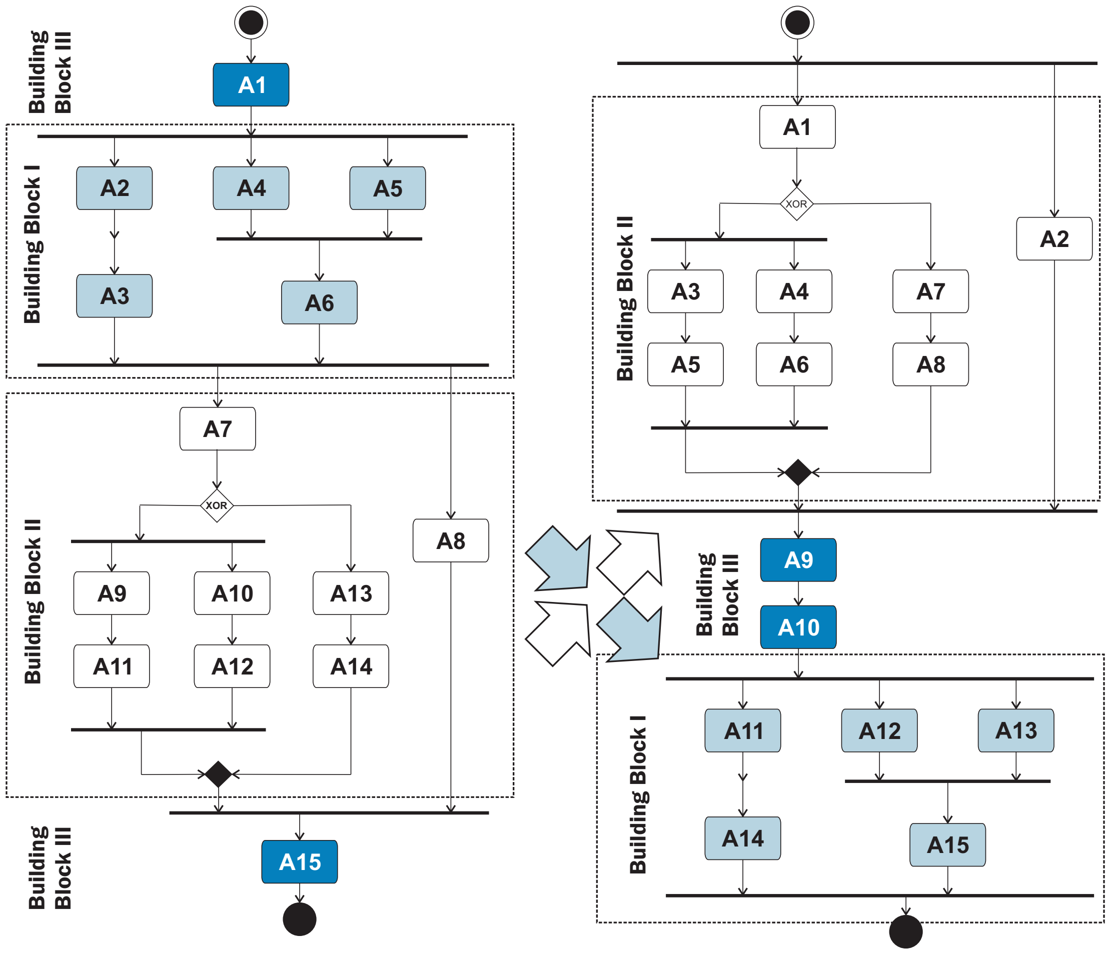

Figure 5: Scheduling tasks A (left) and B (right)

guarantee reproducibility. Thereby, it is to be noted that different λ may lead to the same plan of a service provision.

To solve the MSPSP and the MSPSP-ov, for each problem the evaluation was stopped after 100,000 simulation runs in which an equal number of schedules were generated. There was no weighting of the planning objectives (service time, service costs) to be minimised. This yields a Pareto front that contains a set of non-dominated solutions. Figure 7 shows the Pareto front of both scheduling tasks. All non-dominated (optimal) solutions are completely known due to the use of a multi objective evolutionary algorithm and an exact CPLEX solver. For the MSPSP the discrete event simulation model allowed to find only 12 of the 14 non-dominated solutions for the MSPSP and 13 of 15 for the MSPSP-ov. The Pareto front is therefore not completely approximated by the discrete event simulation experiment. A follow-up survey indicated that an increase of the numbers of simulation up to 1,000,000 runs does not necessarily lead to a comprehensive description of the Pareto-front.

The frequency of a solution with a specific setting of service cost and service time for *Task A* is shown in Fig. 7. The figure shows the identified solution space after 100,000 simulation runs. The results affirm that clusters of solution exist, which were frequently identified as solutions on

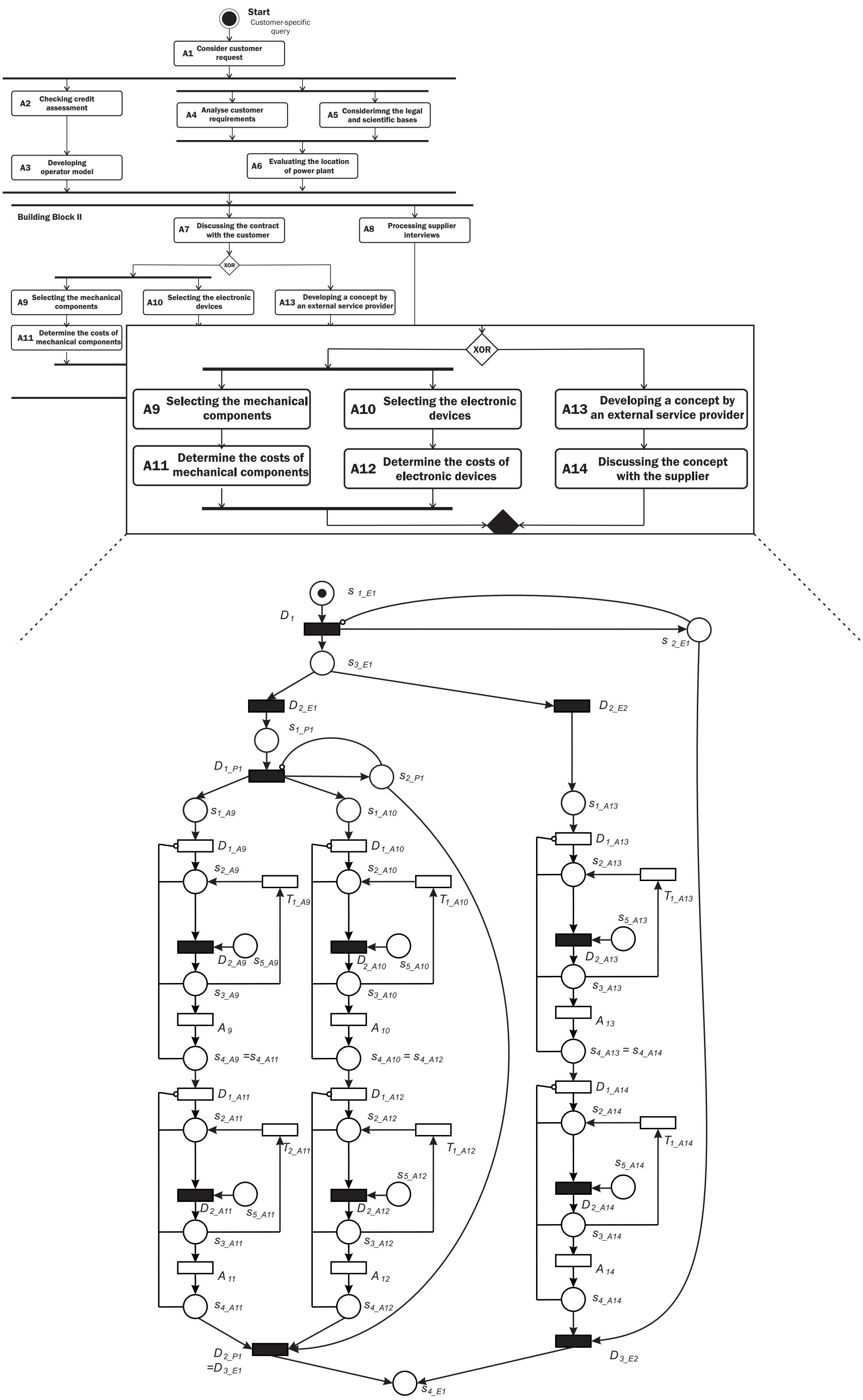

Figure 6: Timed Petri net model for the represented section of the planning task

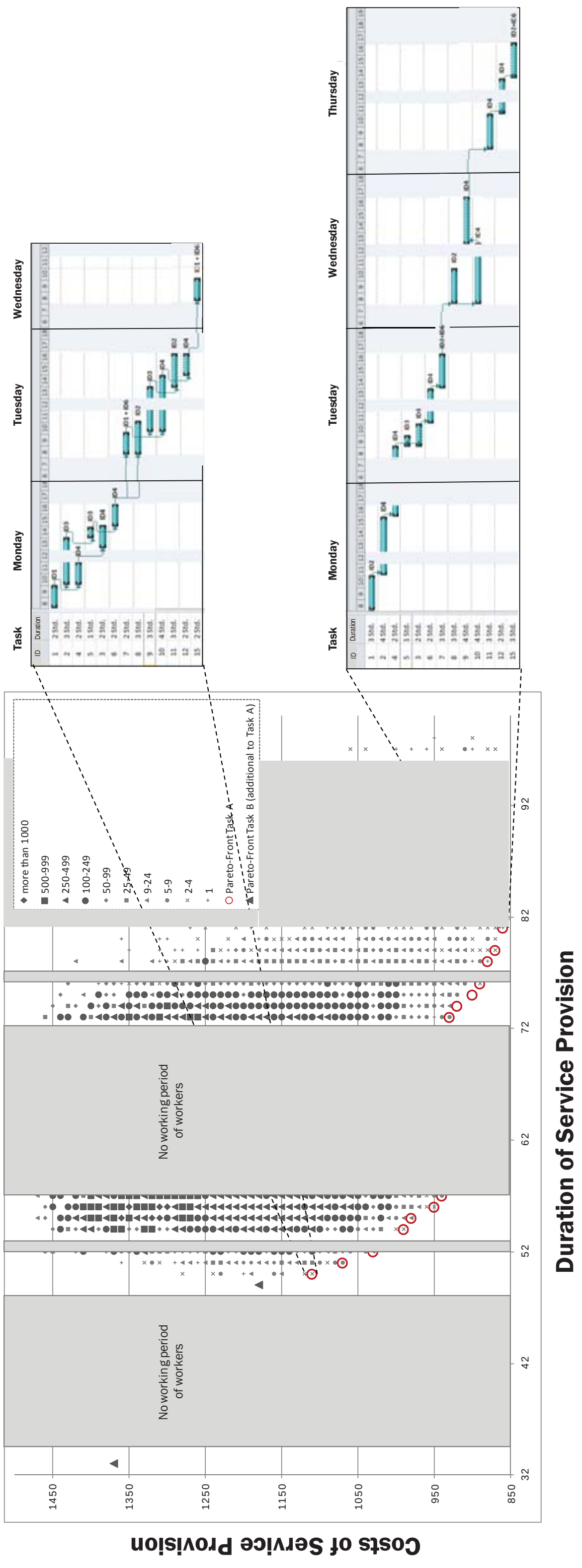

Figure 7: Simulation results for the MSPSP (Task A)

or nearby the Pareto-front. This is explained by the existence of a larger group of valid combinations of different activity configurations which lead to one specific setting of service time and cost. Therefore, the probability of identifying such a point in the solution space is far higher.

For the scheduling problem with resource constraints and overlapping (MSPSP-ov), the overlapping of tasks does not necessarily lead to a non-dominated solution. Therefore, only a task i on the critical path should be overlapped if the reduction has an impact on the total makespan of the service. The MSPSP and the MSPSP-ov differ only in the combination of the building blocks and the overlapping modes of tasks $A8$, $A10$, and $A14$. If an overlapping occurs, the simultaneous processing of a task leads to additional workload and to more working hours of the involved working persons. Therefore, overlapping is less attractive if the working person assigned to the overlapped task has a higher hourly wage or both tasks are assigned to the same working person. For the illustrated scheduling problem with overlapping two further solutions of the Pareto front compared to the MSPSP were identified (Fig. 7). Due to the consideration of the working hours of the working person the overlapping cuts the minimum possible makespan of the service by 34%. This confirms that overlapping and the availability of competences with a specific level are closely interrelated.

5 Conclusion and future work

In the previous sections we addressed the scheduling of a service provision with heterogeneous qualifications and competencies of internal and external working persons. The problem is typical for the provision of engineering and maintenance services as well as IT services. We introduced an actor-oriented service model to formalise the problem of planning a service provision. The model was heavily influenced by the RCPSP (Artigues et al. 2008) and the MSPSP (Bellenguez-Morineau and Néron 2008; Firat and Hurkens 2011; Li and Womer 2009; Néron and Baptista 2002) in the operations research domain. The introduced model of a service provision offers a novel concept that is able to cope with individual decision making and weakly structured precedence constraints of cooperative tasks as a decision vector λ. The model allows to identify optimal solutions of real-world problems with a stochastic simulation model. To evaluate the performance of stochastic simulation and to substantiate the results, we have prescinded the service model to solve two scheduling problems with known Pareto-front. The objective was to minimise the service time and costs through the improvement of task sequences, assignment of working persons, under certain constraints (availability, qualification, skills, level of overlapping tasks etc.). A simulation experiment was carried out for two small service projects. Due to an existing documentation, detailed information about the task processing and the amount of work was available. Comparing the known Pareto front with the set of non-dominated solutions identified by discrete event simulation with multiple replications, we observed that already for very small scheduling problems the Petri net based simulation logic was not able to fully identify the Pareto set of an actor oriented service model. Although, the Pareto front was not completely identified; we showed that the approach offers managers of service companies a suitable technique for the quantitative comparison of alternative service provision scenarios at an early planning stage. An additional comparison of the simulation results with plans created by humans in a laboratory study demonstrated moreover that it is difficult for humans to identify Pareto-optimal solutions for service-oriented scheduling problems (Tackenberg et al. 2012).

In future papers we will present a refined multi objective evolutionary algorithm for actor-oriented service models (Tackenberg et al. 2011). The novel algorithm will be able to cope with more complex service provisions, an iterative processing of tasks and working persons with heterogeneous working hours. In addition, further objectives of services will be identified and integrated into the service model to improve statements on the quality of plans.

Acknowledgement

This work was supported by the German Federal Ministry of Education and Research according to Grant Nos. 01FL10011 (WiDiPro, `www.widipro.de`), and is administered by PT-DLR. This support is gratefully acknowledged.

References

van der Aalst W., Stahl C. (2011) Modeling Business Processes: A Petri Net-Oriented Approach. MIT Press, Cambridge Mass.

Adam N. R., Atluri V., Huang W.-K (1998) Modeling and Analysis of Workflows Using Petri Nets. In: Journal of Intelligent Information Systems 10(2), pp. 131–158

Al-Anzi F. S., Al-Zamel K., Allahverdi A. (2010) Weighted Multi-Skill Resources Project Scheduling. In: International Software Engineering & Applications (3), pp. 1125–1130

Artigues C., Demassey S., Néron E. (eds.) Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications. Wiley, Hoboken, NJ

Banks J., Carson J. S., Nelson B. L., Nicol D. M. (2010) Discrete-event system simulation, 5th ed. Prentice Hall, Upper Saddle River, N.J.

Bartusch M., Möhring R., Radermacher F. J. (1988) Scheduling project networks with resource constraints and time windows. In: Annals of Operations Research 16(1), pp. 201–240

Bellenguez-Morineau O., Néron E. (2008) Multi-Mode and Multi-Skill Project Scheduling Problem. In: Artigues C., Demassey S., Néron E. (eds.) Resource-Constrained Project Scheduling. Wiley, Hoboken, NJ, pp. 149–160

Buddhakulsomsiria J., Kim D. S. (2006) Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. In: European Journal of Operational Research (175), pp. 279–295

David G., Nelson B. L. (1998) Comparing systems via simulation. In: Banks J. (ed.) Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. John Wiley & Sons, Berlin, pp. 272–306

Firat M., Hurkens C. (2011) An improved MIP-based approach for a multi-skill workforce scheduling problem. In: Journal of Scheduling 15(3), pp. 1–18

Fitzsimmons J. A., Fitzsimmons M. J. (1999) Service management: Operations, strategy, and information technology, 2nd ed. Irwin and McGraw-Hill, Boston, Mass

Franck B., Neumann K., Schwindt C. (2001) Project scheduling with calendars. In: OR Spektrum 23(3), pp. 325–334

Greasley A. (2004) Simulation Modelling for Business. Ashgate, Hants

Hartmann S. (1999) Project scheduling under limited resources: Models, methods, and applications. Springer, Berlin

Hartmann S., Briskorn D. (2010) A survey of variants and extensions of the resource-constrained project scheduling problem. In: European Journal of Operational Research 207(1), pp. 1–14

Heimerl C., Kolisch R. (2010) Scheduling and staffing multiple projects with a multi-skilled workforce. In: OR Spectrum 32(2), pp. 343–368

Kadrou Youness, Najid N. M. (2006) A new heuristic to solve RCPSP with multiple execution modes and Multi-Skilled Labor. In: IMACS Multiconference on Computational Engineering in System Applications 2006. Beijing, pp. 1302–1309

Kausch B., Schneider N., Tackenberg S., Schlick C. M., Luczak H. (2008) Integrative Simulation of Work Processes. In: Nagl M., Marquardt W. (eds.) Collaborative and distributed chemical engineering. Springer, Berlin, pp. 451–476

Knechtel T., Kempkes J. P. (2006) Ein Ameisenalgorithmus für die ressourcenbeschränkte Projektplanung mit Zeitfenstern und Kalendern. In: Haasis, Hans-Dietrich, Kopfer H., Schönberger J. (eds.) Operations research proceedings 2005. Springer, Berlin, pp. 691–696

Kolisch R., Drexl A. (1997) Local search for non-preemptive multi-mode resource-constrained project scheduling. In: IIE Transactions (29), pp. 987–999

Li H., Womer K. (2009) Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. In: Journal of Scheduling 12, pp. 281–298

Li H., Dula J., Lewis K., Womer K. (2004) Using Constraint Programming to Solve a Project Scheduling Problem with Skilled Labor. In: Boddy M. C. A., Smith Stephen (eds.) ICAPS-04 Workshop on Integrating Planning into Scheduling. 14th International Conference on Automated Planning and Scheduling, pp. 53–59

Néron E., Baptista D. (2002) Heuristics for the multi-skill project scheduling problem: International Symposium on Combinatorial Optimization (CO'2002)

Prashant Reddy J., Kumanan S., Krishnaiah Chetty O. V. (2001) Application of Petri Nets and a Genetic Algorithm to Multi-Mode Multi-Resource Constrained Project Scheduling. In: The International Journal of Advanced Manufacturing Technology 17(4), pp. 305–314

Reyck B. d., Herroelen W. (1999) The Multi-Mode Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations. In: European Journal of Operational Research 119(2), pp. 538–556

Rosenkranz C. (2008) Analyzing Information Flows in Service Networks. In: Thomas O., Nüttgens M. (eds.) Dienstleistungsmodellierung: Methoden, Werkzeuge und Branchenlösungen. Physica-Verlag HD, Berlin, pp. 35–51

Santos M. A., Tereso A. P. (2011) On The Multi-Mode, Multi-Skill Resource Constrained Project Scheduling Problem – A Software Application. In: Cunha A. G. L. d. (ed.) Soft computing in industrial applications. Springer, Berlin, pp. 239–248

Schwindt C., Trautmann N. (2000) Batch-Scheduling in der Prozeßindustrie: eine Anwendung der ressourcenbeschränkten Projektplanung. In: OR Spectrum (22), pp. 501–524

Shtub A., Bard J. F., Globerson S. (2005) Project Management: Processes, Methodologies, and Economics, 2nd ed. Pearson, Prentice Hall

Steel P., König C. J. (2006) Integrating Theories of Motivation. In: The Academy of Management Review 31(4), pp. 889–913

Tackenberg S., Duckwitz S., Karahance S., Schlick C. M. (2010a) A Meta-Model for Actor-Oriented, Person-Centered Simulation for the Management of Development Projects. In: Proceedings of the 2010 IEEE 17th International Conference on Industrial Engineering and Engineering Management, Xiamen

Tackenberg S., Duckwitz S., Schlick C. M. (2010b) Activity- and actor-oriented simulation approach for the management of development projects. In: International Journal of Computer Aided Engineering and Technology (IJCAET) 2(4), pp. 414–435

Tackenberg S., Gärtner T., Duckwitz S., Schlick C. M. (2010c) Simulation Based Evaluation of Service Science Productivity for Solution Providers. In: International Journal of Service Science, Management, Engineering, and Technology (IJSSMET) 1(4), pp. 35–52

Tackenberg S., Schmitz P., Duckwitz S., Schlick C. M. (2011) Multi-criteria optimization of service productivity using evolutionary algorithm. In: Ganz W., Kicherer F., Schletz A. (eds.) Productivity of services nextgen-beyond output/input. Fraunhofer Verlag, Stuttgart, pp. 1–22

Tackenberg S., Rieder S., Duckwitz S., Schlick C. M. (2012) Resource Constrained Service Scheduling Problem: Analysis of Human Planning Performance. In: Proceedings of the International Conference on Management and Service Science (MASS 2012), August 10-12, Shanghai, China

Tversky A., Kahneman D. (1992) Prospect Theory: Cumulative Representation of Uncertainty. In: Journal of Risk and Uncertainty 5, pp. 297–323

Sven Tackenberg, Sönke Duckwitz,
Christoph M. Schlick

RWTH Aachen University
Institute of Industrial Engineering and
Ergonomics
Bergdriesch 27
D-52062 Aachen
{s.tackenberg | s.duckwitz | c.schlick}
@iaw.rwth-aachen.de

Appendix

		MSPSP				MSPSP-ov							
i	P_i	M_i	AP_i^{min} AP_i^{max}	AK	a_{im}	i	P_i	M_i	AP_i^{min} AP_i^{max}	AK	a_{im}	Overlapping O_j	 p: a_{im}, c_i
A1	9	1 2	[1,1] [1,1]	K1[m] K1[h]	3 2	A1	--	1 2	[2,2] [2,2]	K1[m], K5[h] K1[h], K5[h]	3 2	--	--
A2	10	1 2	[1,1] [1,1]	K3[m] K3[h]	4 3	A2	--	1 2	[1,1] [1,1]	K1[m] K1[h]	3 2	--	--
A3	11	1 2	[1,1] [1,1]	K2[m] K2[h]	2 2	A3	1	1 2	[1,1] [1,1]	K3[m] K3[h]	4 3	--	--
A4	10	1 2	[1,1] [1,1]	K3[m] K3[h]	2 1	A4	1	1 2	[1,1] [1,1]	K3[m] K3[h]	4 3	--	--
A5	10	1	[1,1]	K3[h]	1	A5	3	1 2	[1,1] [1,1]	K2[m] K2[h]	4 3	--	--
A6	12,1 3	1 2	[1,1] [2,2]	K2[h] K2[h]	2 1	A6	4	1 2	[1,1] [2,2]	K2[h] K2[h]	2 1	--	--
A7	--	1 2	[2,2] [2,2]	K1[m], K5[h] K1[h], K5[h]	3 2	A7	1	1	[1,1]	K4[h]	4	--	--
A8	--	1 2	[1,1] [1,1]	K1[m] K1[h]	3 2	A8	7	1 2	[2,2] [2,2]	K1[m], K3[h] K1[h], K3[h]	4 3	1 2 3 4	1,3: 4, 440 2,3: 6, 480 1,3: 3, 330 2,3: 4, 320
A9	1	1 2	[1,1] [1,1]	K3[m] K3[h]	4 3	A9	2,5, 6,8	1 2	[1,1] [1,1]	K1[m] K1[h]	3 2	--	--
A10	1	1 2	[1,1] [1,1]	K3[m] K3[h]	4 3	A10	9	1 2	[1,1] [1,1]	K1[m] K1[h]	3 2	1 2 3 4	1: 3, 240 2: 4, 200 1: 3, 160 2: 4, 200
A11	3	1 2	[1,1] [1,1]	K2[m] K2[h]	4 3	A11	10	1 2	[1,1] [1,1]	K3[m] K3[h]	4 3	--	--
A12	4	1 2	[1,1] [2,2]	K2[h] K2[h]	2 1	A12	10		[1,1] [1,1]	K3[m] K3[h]	2 1	--	--
A13	1	1	[1,1]	K4[h]		A13	10	1 2	[1,1]	K3[h]	1	--	--
A14	7	1 2	[2,2] [2,2]	K1[m], K3[h] K1[h], K3[h]	4 3	A14	11	1 2	[1,1] [1,1]	K2[m] K2[h]	2 2	1 2 3 4 5 6	2: 3, 150 3: 3, 90 4: 3,...30 2: 3, 150 3: 3, 90 4: 3,...30
A15	2,5, 6,8	1 2	[1,1] [1,1]	K1[m] K1[h]	3 2	A15	12,1 3	1 2	[1,1] [2,2]	K2[h] K2[h]	2 1	--	--

Table 4: Parameters for the MSPSP and MSPSP-ov

	Head of Department (ID 1)		Engineer 1 (ID 2)		Engineer 2 (ID 3)		Student trainee (ID4)		External service provider (ID5)		Customer (ID6)	
	Competence	Cost [€/h]	Competence	Cost [€/h]	Competence	Cost [€/h]	Competence	Cost [€/h]	Competence	Cost [€/h]	Competence	Cost [€/h]
MSPSP	K1[h]	80	K1[m] K2[h]	50	K2[m] K3[h]	30	K2[h] K3[m]	10	K4[h]	30	K5[h]	0
MSPSP-ov	K1[h]	80	K1[m] K2[h]	50	K2[m] K3[h]	30	K2[h] K3[m]	10	K4[h]	40	K5[h]	0
Skill: [h]: high; [m] medium; [l] low; Working hours:8:00–12:00; 13:00–17:00												

Table 5: Working persons for the MSPSP and MSPSP-ov