

Planning Dynamic Activity and Resource Allocations Using a Risk-Aware Business Process Management Approach

Simon Tjoa
St. Poelten University of Applied Sciences
St. Poelten, Austria
simon.tjoa@fhstp.ac.at

Stefan Jakoubi, Sigrun Goluch, Gerhard Kitzler
Secure Business Austria
Vienna, Austria
sjakoubi, sgoluch, gkitzler@sba-research.org

Abstract—The planning and effective usage of resources is a major challenge for organizations. More than ever through the global interconnections and the current economic situation the domain of resource allocation gains importance. Thus, within this paper we contribute to this research field by introducing a novel approach for dynamic activity and resource allocation using risk-aware business process simulations in order to facilitate and enhance planning and analysis activities. To clarify our work we show the application using a stylized business case.

Index Terms—resource allocation, simulation, risk, business process management

I. INTRODUCTION

The management of resources is indispensable for every company in order to reliably perform their business processes. It is undeniable that companies highly depend on the efficient and moreover continuous operation of their critical business processes. Within the course of this paper the management and allocation of resources include all resources considered within business process management. This includes amongst others technical IT infrastructure (e.g. servers or workstations), applications (e.g. essential business applications such as SAP), human resources (e.g. key employees) and premises (e.g. server room).

Techniques supporting proper resource allocation and planning from an economical viewpoint have been addressed by business process management discipline for many years. Thus it is no big surprise that there exist a huge variety of business process management [1], [2] approaches coping with the effective and efficient design of business processes. These approaches are often supported by business process simulations which enable the process architect to validate the design beforehand. When analyzing the approaches in the literature we identified a shortcoming regarding the integration of security aspects which can play an essential part when assigning resources as requirements can change rapidly when facing a threat. Security related aspects and resource requirements are mostly addressed separately, often by different departments and responsible persons. Risk Management [3], [4] Business Continuity Management [5],

[6], [7], [8], [9] and Incident/Problem Management [10] are representative disciplines coping with requirements during negative events (e.g. virus attacks). Similar to the business process management domain exhaustive research has been performed to best possible protect organizations. However, a common reasoning and information basis that links both worlds is still missing. Therefore, we recently introduced a novel approach integrating a risk view into business process management which subsequently enables the simulation of risks within business process simulations [11]. This extension of "traditional" business process management enabled us to consider the analysis of certain threats and their impacts on the organization.

Within this paper we extended the capabilities of our formal modeling and simulation approach in order to represent and include the perspective of cross functional resource allocation. We further added the possibility to simulate workarounds by introducing the concept of activity reallocation. The major contribution is the extension of our formal model and the introduction of additional simulation capabilities that allow the abovementioned dynamic allocations. We tested our approach with a stylized business case using a proof-of-concept prototype implemented in the professional simulation toolkit Simulink®.

Sample Scenario

As guidance for the presentation of our formal model and as basis for our evaluation, we introduce the following demonstrative sample scenario. We assume the call center of the company ACME. The main activities are first level support (Act1), second level support (Act2) and third level support (Act3). Historical data of ACME shows that the first level support can handle 2/3 of incoming calls whereas 1/3 of the calls have to be escalated to the second level support. The second level support can handle 1/2 of the forwarded requests. For adequately performing the activities, Act1 requires three persons with skill level A, Act2 two with skill level B and Act3 2 with skill level C. Due to cross skill trainings, persons with skill level B can - if needed - perform the work of skill level A and persons with skill level C the work of skill level

B. The sample scenario is schematically shown in Figure 1.

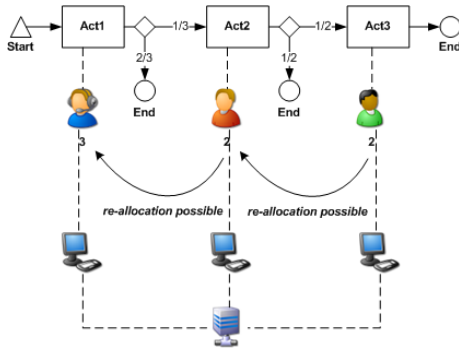


Fig. 1. Sample Process

The rest of this paper is structured as follows: Section 2 outlines related research approaches in the field of business process security and resource allocation as well as our previous work in the field of risk-aware business process management. Section 3 introduces our formal model that enables us to conduct dynamic activity and resource allocations during our risk-aware business process simulation. In section 4 we present our proof-of-concept implementation of the formal model. Section 5 illustrates the results of a sample scenario to show the capabilities of the approach. Section 6 concludes the paper by briefly summarizing the major benefits of the extension.

II. RELATED RESEARCH

Within this section, we give a brief overview firstly about research results aiming at enhancing resource allocation and secondly at the incorporation of risk aspects into business process modeling and analyses. For detailed information on the included approaches, we kindly refer to the according references.

Resource Allocation

Korvin et al. [12] introduce an approach how to assign tasks to resource pools. Each task has specific resource requirements and each pool can provide resources with varying capacities. The importance of a resource for executing a specific task is determined via its membership. A membership value of zero indicates the lowest importance level (no use) and a membership value of one reflects the highest importance, dependency respectively. Task and resource pools are organized as finite fuzzy sets. In the case of a resource pool, the membership attributes indicate the pool's possibility to provide resources. Exemplarily, a membership value of zero states that the pool is not able to provide the needed resource. The authors derive a so called measure of compatibility between task and resource pools. This compatibility is also organized as fuzzy set. The center of area method is used to enable a comparison of the sets' defuzzifications. Furthermore, an algorithm is described how to recursively assign tasks to resource pools and to handle violations (i.e. no assignment possible). For decision making purposes, the authors consider in their approach costs of using a pool and

a flexible budget so that if the need is accordingly large, the initial budget limit may be exceeded for some extent. To reflect importance issues, the approach allows the weighting of compatibility and costs.

Netjes et al. [2] introduce an approach addressing the challenge to improve two best practices identified in [3]: specialist-generalist trade-off and the flexible assignment policy. The specialist-generalist trade-off has the objective to identify the optimal ratio between specialists which are capable to perform one task and generalists which can perform more than one task. Furthermore, it is assumed that specialists can do their task faster than generalists. The usage of the flexible assignment policy should guarantee that the best possible flexibility is preserved for the future. This implies that if two resources for a task are available (one specialist and one generalist), the specialist would be assigned to carry out the task to maintain a better flexibility. Within the scope of the paper business processes consist of inter-related tasks. A task is carried out by one resource. A task has the restriction that it can be fully performed or not at all. Resources have different roles to facilitate the mapping between resources and tasks. For modeling and analyzing of such a resource-constraint process the authors use colored petri nets (CPN). Within these CPNs tasks are connected to generic resource modules which contain the role capable to fulfill the task. The allocation strategies that can be used by resource modules to assign resources to tasks are priority-based or random. If the priority based allocation strategy is used pre-defined priorities are used to allocate the resources. In the second case resources are allocated by randomly drawing a resource that is capable to perform a task out of the resource pool. In order to clarify the approach the authors present a short sample bank process modeled in CPN using their allocation approach.

Business Process Security

Neiger et al. [13] address the need for a holistic business view on risk management in the enterprise systems space. Therefore the authors introduce a framework by applying value-focused process engineering principles to risk management models, which enables risk-oriented process management, incorporating a multi-disciplinary view of risk. Milanovic et al. [14] present in their work a framework modeling business process availability taking services, underlying ICT infrastructure and human resources into account. This framework uses a fault-model with two failure modes (i.e. temporal, value) for its analytical assessment procedure.

Sackmann [15], [16] introduces an IT risk reference model which bridges the economic and more technical layers with vulnerabilities. The model consists of four interconnected layers: (1) business process layer; (2) IT applications / IT infrastructure layer; (3) vulnerabilities layer; (4) threats layer. For expressing relations between these layers (i.e. the searched cause-effect relations) a matrix-based description is proposed.

Neubauer et al. propose an IT-Security Valuation Framework which aims at establishing the connection between a

company's core business processes, IT-processes and security levels [17]. Core business processes are used to determine the external value regarding the valuation of security measures. This valuation bases on the determination of downtime costs in the case of a system's unavailability. IT processes are used to measure "the costs needed for implementing and keeping a defined level of security". On the basis of the information gained from the analysis of the core business processes and IT processes valuation models such as ALE (Annual Loss Expectancy) can be used for calculating the expected loss and defining the optimal level of security.

CORAS [18] is a method for conducting security risk analysis. It provides a customized language for threat and risk modeling, and comes with detailed guidelines explaining how the language should be used to capture and model relevant information during the various stages of the security analysis. The Unified Modeling Language (UML) is used and extended to model the target of the analysis.

Jensen and Feja [19] introduce in their work a new approach to integrate security aspects into model driven software development. The authors therefore introduce a new model which they call an EPC Security Model View. This model view is a new perspective within the ARIS SOA architect. This extended view is used to specify security properties. The specifications are later on considered when generating the WS Security Policy and BPEL processes.

For further approaches in the field of Business Process Security, we kindly refer the reader to [20].

Previous Work

In this section we present our previous work in the domain of risk-aware business process management in order to facilitate our contribution's discussion in the following sections.

In literature, one can see that there is a growing need in integrating risk and business process management [20] whereas according simulations are the next logical step. This led us to the development and introduction of our novel reference model enabling risk-aware business process management [11]. This reference model serves as basis for our contribution within this paper - i.e. the planning and analysis of dynamic activity and resource allocation - and is thus in this chapter briefly summarized.

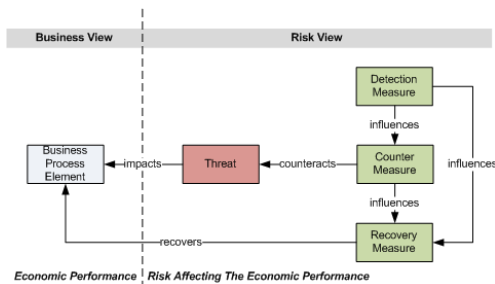


Fig. 2. General Reference Model Enabling Risk-Aware Business Process Management [11]

Conceptually, threats affect business process elements such as activities (e.g. delay in execution time) or resources

(e.g. confidentiality breach). Detection measures influence the invocation of counter measures and/or recovery measures. Counter measures influence the invocation of recovery measures. Counter measures counteract threats (until an occurred threat cannot impact a business process element any more) and recovery measures try to re-establish the intended functionality of a potentially affected business process element. For a detailed description of the reference model, we kindly refer to [11].

III. FORMAL MODEL

In this section we present our formal model enabling resource and activity allocation within business process management system. We therefore formally describe the mathematical assembly of such a system concentrating on business process activities and their resources.

Formal Description of Business Process Elements

Let $\mathcal{P} := \{process_1, \dots, process_k\}$ be the set of all business processes. Every business process consists of business process elements, such as activities, resources. These elements possess attributes. In order to formally describe business process elements and their relations we introduce the following sets. Let $\mathcal{Act} = \{Act_1, \dots, Act_l\}$ be the set of activities, $\mathcal{R} = \{R_1, \dots, R_n\}$ the set of resources and $\mathcal{A} = \{A_1, \dots, A_m\}$ the set of attributes. For a comprehensive description of a business process we have to meet several requirements, such as following: Let $AsR_i := \{R_j \in \mathcal{R} | R_j \text{ is assigned to } Act_i\}$ be the set of assigned resources of activity i and $AsAct_k := \{Act_l \in \mathcal{Act} | Act_l \text{ is assigned to } process_k\}$ be the set of assigned activities of process k .

Every activity has certain requirements concerning their assigned resources. Those requirements are retained in the resource requirements:

$$RR_i := \{Act_i; \{(R_j, A_m, Z_j, f_{j,m}(t)), \dots | R_j \in AsR_i\}\} \quad (1)$$

where Act_i is the certain activity, R_j is the required resource, A_m the needed attribute of resource R_j , Z_j a threshold for the attribute's condition (e.g. availability = 100%). If the attribute falls below Z_j (e.g., threat impact) and no redundant resources are available, then the corresponding activity is impacted. The function $f_{j,m}(t)$ describes how the alteration of the resource's attribute condition changes the activities' condition itself.

It is obvious that an alteration of a resource attribute's condition has an indirect influence on the corresponding activity itself. The impact of a threat on the attacked resource's attribute is defined in the threat's impact function itself ($\omega(t)$). In order to establish the link between the impact on the resource and the activity we use function $f_{j,m}(t)$ which is independent of $\omega(t)$ but dependent on the resource's condition.

To understand how an activity is structured we combine the quadruples of RR_i (i.e. $(R_j, A_m, Z_j, f_{j,m}(t))$) with the

logical operators \wedge/\vee and obtain the activity assembly:

$$As(Act_i) := q_k \wedge / \vee q_l \wedge / \vee q_m \cdots \quad (2)$$

where the $q_{(k,l,m)}$ are quadruples of the form contained in RR_i . In $As(Act_i)$ all quadruples of RR_i are used to obtain an explicit description of the activities' structure.

For each activity Act_i there exists one exact set of assigned resources AsR_i and one activity assembly $As(Act_i)$ (assumption 3). The same can be said about each process (*process_j*). There is one exact set of assigned activities for each process $AsAct_j$ (assumption 4). Therefore we define two assumptions:

$$\forall Act_i, i = 1, \dots, l : \exists! AsR_i \wedge \exists! As(Act_i) \quad (3)$$

$$\forall process_j, j = 1, \dots, k : \exists! AsAct_j \quad (4)$$

Formal Description of Resource Allocation Elements A resource allocation is needed when there appears to be a change in the process-, activity- or resource attribute's condition. We define three archetype sets for resources, activities and processes. Every set contains one or more indicators and one or more subsets of possible resources for a successful allocation.

$$RAI(R_i) := \{ \underbrace{(A_j, \leq / \geq W_{i,j}, [t_n, t_m], PR_{i,j}, \dots)}_{\text{indicator sequence}} \} \quad (5)$$

$$RAI(Act_k) := \{ \underbrace{(A_l, \leq / \geq V_{k,l}, [t_n, t_m], \bigcup_{i,j} PR_{i,j} : \forall R_i \in AsR_k, \dots)}_{\text{indicator sequence}} \} \quad (6)$$

$$RAI(process_g) := \{ \underbrace{(A_h, \leq / \geq U_{g,h}, [t_n, t_m], \dots)}_{\text{indicator sequence}} \} \quad (7)$$

where in equation (5) $R_i \in \mathcal{R}$ is the certain resource, A_j one of its attributes. $W_{i,j}$ is the threshold for the attribute's A_j condition ($z(A_j)$), it is directly obtained from the corresponding resource requirements RR_n . Therefore $W_{i,j} \hat{=} Z_k$, where Z_k is the threshold from resource R_i and attribute A_j from the corresponding $RR_n | R_i \in AsR_n$. The relational operators \leq / \geq determine if the threshold has to be undershot / overshot for a successful impact of a threat. $[t_n, t_m]$ is the time interval in which the threshold is applying. $PR_{i,j}$ is a set of possible resources that are able to successfully proceed the original resource's (R_i) task. The elements of $PR_{i,j}$ are tuples of the form: (R_m, p_m) , where R_m is the resource that could replace the original resource and p_m the corresponding percentage to which the resource's attribute is needed for the specific activity. It is possible to combine two or more such tuples with the logical operator \wedge . In equation (6), Act_k is the certain activity, A_l one of its attributes, $V_{k,l}$ the threshold for the attribute's condition as was described above. To get all possible allocation resources for an activity, we unite every $PR_{i,j}$ of the activity's resources. In equation (7), $process_g$ is the certain process, A_h one of its attributes, $U_{g,h}$ the threshold for the attribute's condition. We assume that an affected process has to have an affected activity as well. Therefore it has only the indicator

sequence and no set of possible resources for each attribute.

The sample scenario (Fig. (1)):

The sample scenario described with the definitions established above: ($A_1 \hat{=} availability$, $A_2 \hat{=} backlog$, $A_3 \hat{=} proceeds$)

$$AsR_1 := \{R_1, R_2, R_3\}$$

$$AsR_2 := \{R_4, R_5\}$$

$$AsR_3 := \{R_6, R_7\}$$

$$AsAct_1 := \{Act_1, Act_2, Act_3\}$$

$$RR_1 := \{Act_1; \underbrace{\{(R_1, A_1, 70\%, f_{1,1}(t)), \dots\}}_{q_1}$$

$$\underbrace{\{(R_2, A_1, 70\%, f_{2,1}(t)), (R_3, A_1, 70\%, f_{3,1}(t))\}}_{q_2, q_3}$$

$$RR_2 := \{Act_2; \underbrace{\{(R_4, A_1, 80\%, f_{4,1}(t)), \dots\}}_{q_4}$$

$$\underbrace{\{(R_5, A_1, 80\%, f_{5,1}(t))\}}_{q_5}$$

$$RR_3 := \{Act_3; \underbrace{\{(R_6, A_1, 80\%, f_{6,1}(t)), \dots\}}_{q_6}$$

$$\underbrace{\{(R_7, A_1, 80\%, f_{7,1}(t))\}}_{q_7}$$

$$As(Act_1) := q_1 \vee q_2 \vee q_3$$

$$As(Act_2) := q_4 \vee q_5$$

$$As(Act_3) := q_6 \vee q_7$$

$$RAI(R_1) := \{(A_1, \leq, \overbrace{W_{1,1}}^{=70\%}, [t_0, t_{end}], PR_{1,1})\}$$

$$PR_{1,1} := \{(R_4, 100\%), (R_5, 100\%)\}$$

$$RAI(R_2) := \{(A_1, \leq, 70\%, [t_0, t_{end}], PR_{2,1})\}$$

$$PR_{2,1} := \{(R_4, 100\%), (R_5, 100\%)\}$$

⋮

$$RAI(R_5) := \{(A_1, \leq, \overbrace{W_{5,1}}^{=80\%}, [t_0, t_{end}], PR_{5,1})\}$$

$$PR_{2,1} := \{(R_6, 100\%), (R_7, 100\%)\}$$

⋮

$$RAI(R_7) := \{(A_1, \leq, 80\%, [t_0, t_{end}], \emptyset)\}$$

$$RAI(Act_1) := \{(A_2, \geq, \underbrace{V_{1,2}}_{=5}, [t_0, t_{end}], PR_{1,1} \cup PR_{2,1} \cup PR_{3,1})\}$$

⋮

$$RAI(Act_3) := \{(A_2, \geq, \overbrace{V_{3,2}}^{=2}, [t_0, t_{end}], PR_{6,1} \cup PR_{7,1})\}$$

$$RAI(process_1) := \{(A_3, \geq, \underbrace{U_{1,1}}_{50}, [t_0, t_{end}])\}$$

Formal Description of Activity Allocation Elements

It is also possible to substitute an affected activity for another

activity or a subprocess. The indicator sequence is built in the same way as in the formulas (5)(6)(7). The difference related to the resource allocation is that the set of possible substitutes $PAL_{k,l}$ contain activities and/or subprocesses instead of resources.

$$AAI(Act_k) := \underbrace{\{(A_l, \leq / \geq Z_{k,l}, [t_n, t_m], PAL_{k,l}), \dots\}}_{\text{indicator sequence}} \quad (8)$$

with $PAL_{k,l} := \{pal_1, \dots, pal_n\}, n \in \mathbb{N}$

and $pal := \begin{cases} Act_k \in \mathcal{A}ct \\ process_l \in \mathcal{P} \end{cases}$

$$AAI(process_g) := \underbrace{\{(A_h, \leq / \geq K_{g,h}, [t_n, t_m]), \dots\}}_{\text{indicator sequence}} \quad (9)$$

$$\bigcup_{k,l} PAL_{k,l} : \forall Act_k \in AsAct_g, \dots$$

Allocation algorithm

In the previous sections we declared the necessary sets and indicators for a resource allocation within a system of business process elements. To understand how we use those sets of possible resources and indicators we present the algorithm which is later implemented with the toolkit Simulink[®]:

For each process attribute ($\forall A_h \in RAI(process_g)$)

If indicator sequence is true

($z(A_h) \leq / \geq U_{g,h}$ in interval $[t_n, t_m]$)

For each process activity attribute

($\forall A_l \in RAI(Act_k) | Act_k \in AsAct_g$)

If indicator sequence is true

($z(A_l) \leq / \geq V_{k,l}$ in interval $[t_n, t_m]$)

For each process activity resource attribute

($\forall A_j \in RAI(R_i) | R_i \in AsR_k$)

If indicator sequence is true

($z(A_j) \leq / \geq W_{i,j}$ in interval $[t_n, t_m]$)

Choose a resource from the set of possible resources for R_i
($R_i \in PR_{i,j}$)

Else

Choose a resource from the set of possible resources for Act_k
($R_i \in \bigcup_{i,j} PR_{i,j}$)

Assign resource R_i to activity Act_k

If the resource allocation is not effective enough to reduce the impact on the process, or the activity, we introduced the possibility of an activity allocation. It operates similar to the above described algorithm:

For each process attribute ($\forall A_h \in AAI(process_g)$)

If indicator sequence is true

($z(A_h) \wedge / \vee K_{g,h}$ in interval $[t_n, t_m]$)

For each process activity attribute

($\forall A_l \in AAI(Act_k) | Act_k \in AsAct_g$)

If indicator sequence is true

($z(A_l) \leq / \geq Z_{k,l}$ in interval $[t_n, t_m]$)

Choose a subprocess from the set of possible substitutes for Act_k
($pal_n \in PAL_{k,l}$)

Else

Choose a subprocess from the set of possible substitutes for $process_g$
($pal_n \in \bigcup_{k,l} PAL_{k,l}$)

Assign subprocess pal_n to process $process_g$

In this chapter we introduced our novel formal extension of our method enabling the dynamic allocation of resources and activities. We further sketched how our allocation algorithm works.

IV. IMPLEMENTATION

This section introduces the implementation of our formal model enabling resource allocation. We demonstrate how we simulate resource allocation using the professional toolkit Simulink[®]. The implementation follows the sample scenario of the Introduction section. In Figure 3 one can see the main

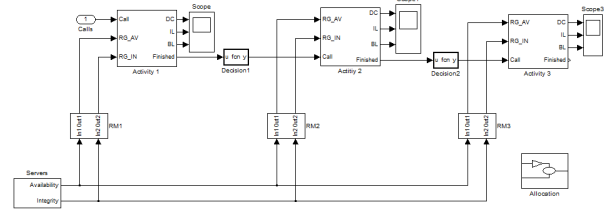


Fig. 3. Simulation Model: Main Layer

layer of our Simulink[®] model. It represents a whole business process with its business process elements. The process consists of three activity subsystems (*Activity 1*, *Activity 2*, *Activity 3*), for each activity a resource management subsystem (*RM1*, *RM2*, *RM3*), one server resource block for all activities (*Servers*) and two decisions between the activities (*Decision1*, *Decision2*). Within the resource management subsystem we realized the assigned PCs. Additionally in the **main layer** we find the allocation subsystem (*Allocation*) which is realized as described above. We assume our process to have one financial attribute which is *proceeds*, our activities have two attributes: *degree of completion (DC)* and *integrity loss (IL)*. The DC attribute depends on the availability attribute of the needed resources respectively the IL attribute depends on the corresponding resources integrity attribute. In our sample case we consider the integrity attribute only for the server. Figure 4 shows an activity layer. The main block in this system is realized as an embedded Matlab[®] function (*Embedded MATLAB Function*) which simulates the activity. The input signals are: the assured resources (*RG_AV*, *RG_IN*), these are the maximal useable resources. The next two input signals (*DC_old*, *IL_old*) represent the DC and IL attribute of the previous time step. The signal *time_unit* serves as a help signal, used to determine the length of the actual time step, which is a necessary information to determine the integration

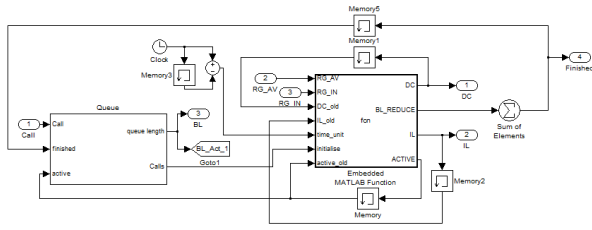


Fig. 4. Simulation Model: Activity Layer

intervals. The *initialise* signal from the Queue block is a Boolean signal determining when to start the activity resp. one of its instances. The last remaining input signal *active_old* holds information whether one of the activity’s instances was active in the previous time step.

The Embedded Matlab Function is structured as follows: In the first step the difference between needed and assured (*RG_AV*, *RG_IN*) resources is calculated to obtain actual completion speed resp. integrity mismatch. The following step is the calculation of the DC and IL attribute. In the last step it is determined if one instance has finished the execution of the activity resp. how many instances are active. In each time step the input signal *active_old* gets updated and acts as the output signal *ACTIVE* which is needed in the Queue block. The *BL_REDUCE* signal denotes if an instance has finished a task in the actual time step.

The Queue block on the left side manages the incoming calls. It has the input signals *Call*, *finished* and *active*. The block calculates the value of the queue length (*queue length*) for the corresponding activity and the initialise indicator (*Calls*) for the main block *Embedded MATLAB function*. Whenever an incoming call appears (*Call=1*), the length of the queue is increased by 1. Whenever an instance finishes a task (*finished=1*) the queue is decreased by 1. The *active* signal gives information about the amount of active or nonactive instances, to decide whether a call gets forwarded to the Embedded MATLAB Function block. In Figure 5 we see the

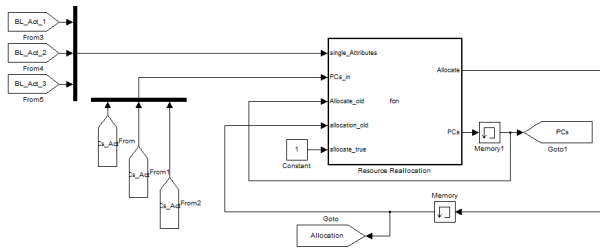


Fig. 5. Simulation Model: Allocation Layer

allocation layer. It acts as a subsystem in the main layer. It contains one main block realized as an embedded Matlab function (*Resource Reallocation*). This block has five input signals. The *single_Attributes* input represents the backlog of each activity (*BL_Act_1*, *BL_Act_2*, *BL_Act_3*). It is used for the calculation of the total backlog. The *PCs_in* signal gives information about the availability attribute condition of

each PC. The *Allocate_old* signal holds information about the previous permutation of the PCs, *allocation_old* is a Boolean signal which is set to true whenever an allocation took place in the previous time step. On the basis of our formal model and the above described input signals, the embedded Matlab function determines if and how an allocation is needed. It has the output signals *Allocate* which is the update signal of *allocation_old*. PCs is the actual permutation of the PCs.

V. RESULTS AND DISCUSSION

This section discusses the results gained by our simulation of the sample scenario. We compare two different settings: (1) no re-allocation considered; (2) resource allocation available. Regarding scenario 1 (Figure 6), as the threat attacks one instance in activity 1 around time step 400, the instance is not able to proceed its given task. This results in an ample increase of activity 1’s backlog (c). It is not possible for the two left instances to reduce this backlog, therefore the backlog increases until the attacked instance is fully recovered and able to work properly again. This is around time step 980. It is apparent that the instance starts to work sooner, around time step 730. But in a distinctly inefficiently manner. The obvious lesser completion rate of activity one results in an almost stagnancy of the process’ financial attribute proceeds. As there is no apparent threat on activity 2 and 3 there is no increase of its backlogs. Regarding scenario 2 (Figure 7), one can see

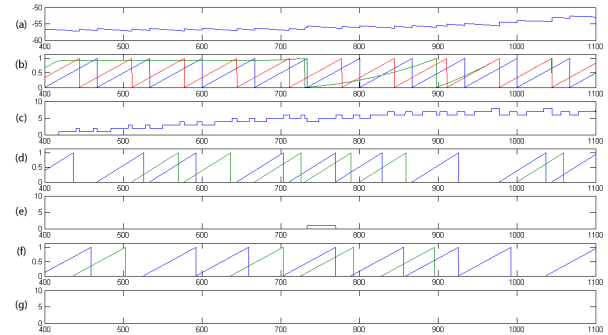


Fig. 6. Scenario: No Resource Allocation

the simulation results for a business process with enabled resource allocation. Subplot (a) represents the process level with its financial attribute proceeds. (b), (d) and (f) represent the activity level with its instances. In activity 1 (b) there are three instances, in activity 2 and 3 (d)(f) there are only 2. The subplots (c),(e) and (g) represents the corresponding backlogs for each activity. We chose to display from time step 400 to 1100 for better understanding. We see that one instance in activity 1 (b) gets attacked by a threat around time step 400, this results in the idleness of its degree of completion and therefore in an increase of activity’s 1 backlog (c). As the backlog reaches the defined threshold (=5) at time step 510 an instance - and therefore its resources - from activity 2 (d) gets reallocated to activity 1. It stays in activity 1 until it gets called back to its assigned activity due to the increase of activity’s 2 backlog (e) to the defined threshold. Soon after the backlog

of activity 2 is reduced, the same instance gets allocated to activity 1 again. As it is not possible for one instance in activity 2 to reduce the backlog an instance from activity 3 (f) gets allocated to activity 1 around time step 750. This results in an increase of activity 3's backlog (g) and in a decrease of activity 2's backlog. The original instance of activity 1 is fully recovered around time step 980. The allocated resources are going back to their original assigned activities and therefore are able to reduce the emerged backlogs.

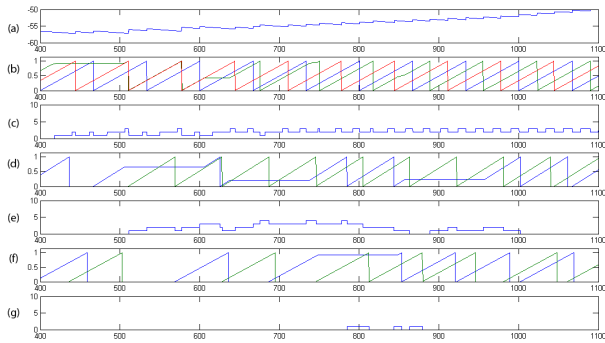


Fig. 7. Scenario: Resource Allocation

When comparing the two scenarios we clearly see that in scenario 2 there is almost no impact on the process' attribute proceeds. Due to the effective resource allocation it is possible to work as if there is no threat at all. In scenario "No Resource Allocation" the threat on activity 1 has an obvious impact on the process' attribute, as there is a major downtime in activity 1's completion rate.

VI. CONCLUSION

Effective activity and resource allocation delivers enormous advantages for organizations competing in today's globally connected markets. Cross-skill trainings of human resources and suitable resource allocation strategies for all kind of resources are the key to an improved utilization of resources and to the continuous execution of business processes. Additionally, workarounds are a significant part of incident response for most organizations and thus deserve proper planning and evaluation.

Within this paper we provided the reader a novel extension of our approach enabling companies to dynamically simulate their activity and resource allocation strategies in the case of negative events (e.g. loss of personnel). The advantages of the approach are obvious. Through the risk-aware business process simulation, effects of threats can be shown. At the same time through the extension of the simulation capabilities, activity and resource allocation as a reaction against certain threats can be evaluated providing a reasonable basis for the selection of cross-skilled trainings and workaround procedures. Furthermore the graphical representations of the simulation results provide a user-friendly possibility to show decision makers the effects of planned allocations.

ACKNOWLEDGEMENTS

This work was performed at the Research Center Secure Business Austria funded by the Federal Ministry of Economy, Family and Youth of the Republic of Austria (BMWFJ) and the federal province of Vienna.

REFERENCES

- [1] D. Karagiannis, S. Junginger, and R. Strobl, *Business Process Modelling*. Springer, Berlin, 1996, ch. Introduction to Business Process Management Systems Concepts, pp. 81–106.
- [2] A. W. Scheer and M. Nüttgens, "Aris architecture and reference models for business process management," in *BPM*, 2000.
- [3] *ISO/IEC 27005:2008 Information technology - Security techniques - Information security risk management*, ISO/IEC Std., 2008.
- [4] National Institute of Standards and Technology, "Nist special publication 800-30, risk management guide for information technology systems," National Institute of Standards and Technology (NIST), 2002.
- [5] British Standard Institute (BSI), "British standard bs25999-1:2006: Business continuity management - part 1: Code of practice," British Standard Institute (BSI), 2006.
- [6] —, "British standard bs25999-2:2007: Business continuity management - part 2: Specification," British Standard Institute (BSI), 2007.
- [7] Business Continuity Institute, "Good Practice Guidelines," 2008. [Online]. Available: <http://www.thebci.org/gpgdownloadpage.htm>
- [8] International Organization for Standardization, "Iso/iec 24762:2008 information technology - security techniques - guidelines for information and communications technology disaster recovery services," ISO/IEC, 2008.
- [9] National Institute of Standards and Technology, "Nist sp800-34: Contingency planning guide for information technology systems," 2002.
- [10] —, "Nist sp800-61: Computer security incident handling guide," National Institute of Standards and Technology, 2004.
- [11] S. Jakoubi and S. Tjoa, "A reference model for risk-aware business process management," in *International Conference on Risks and Security of Internet and Systems*. IEEE, 2009.
- [12] A. De Korvin, S. Hashemi, G. Quirchmayr, and K. R., "Assigning tasks to resource pools: A fuzzy set approach," in *LNCS of the DEXA Conference (DEXA2000)*, 2000.
- [13] D. Neiger, L. Churilov, M. zur Muehlen, and M. Rosemann, "Integrating risks in business process models with value focused process engineering," in *European Conference on Information Systems (ECIS 2006)*, 2006.
- [14] N. Milanovic, B. Milic, and M. Malek, "Modeling business process availability," in *IEEE International Conference on Services Computing (SCC 2008)*, 2008, pp. 315–321.
- [15] S. Sackmann, "A reference model for process-oriented it risk management," in *16th European Conference on Information Systems*, 2008.
- [16] S. Sackmann, L. Lowis, and K. Kittel, "Selecting services in business process execution - a risk-based approach," in *Business Services: Konzepte, Technologien, Anwendungen, Tagung Wirtschaftsinformatik (WI09)*, 2009.
- [17] T. Neubauer, M. Klemen, and S. Biffel, "Business process-based valuation of it-security," in *Economics-driven software engineering research (EDSER 2005)*, 2005.
- [18] F. Braber, I. Hogganvik, M. Lund, and F. V. K. Stolen, "Model-based security analysis in seven steps - a guided tour to the coras method," *BT Technology Journal*, vol. 25, pp. 101–117, 2007.
- [19] M. Jensen and S. Feja, "A security modeling approach for web-service-based business processes," in *16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2009)*, 2009.
- [20] S. Jakoubi, S. Tjoa, G. Goluch, and Q. G., "A survey of scientific approaches considering the integration of security and risk aspects into business process management," in *International Workshop on Business Processes Security (BPS2009)*. IEEE, 2009.