Using Concurrent Task Trees for Stakeholder-centered Modeling and Visualization of Business Processes

Jens Kolb[1], Manfred Reichert[1], and Barbara Weber[2]

[1] Institute of Databases and Information Systems, Ulm University, Germany
{jens.kolb,manfred.reichert}@uni-ulm.de
[2] Quality Engineering Research Group, University of Innsbruck, Austria
barbara.weber@uibk.ac.at

Abstract. The different stakeholders in Business Process Management have to deal with various process models in order to understand the business processes being relevant for them. Especially inexperienced stakeholders often have difficulties in comprehending large and complex process models. In this paper a stakeholder-centered approach for modeling, changing and visualizing business processes is introduced. It is based on the Concurrent Task Tree (CTT), which constitutes a task modeling language widely applied in the field of end-user development. In particular, CTT considers stakeholder needs in modeling the behaviour of user interfaces. In the context of our work we apply CTT for modeling, changing and visualizing business processes. To evaluate whether CTT is appropriate for stakeholder-centered process modeling we compare it with imperative process modeling, and introduce a mapping between CTT process models and imperative process models expressed in terms of the Business Process Modeling Notation (BPMN). Finally, we provide an advanced stakeholder-centered visualization concept based on CTT.

Key words: Stakeholder-centered Process Modeling, Process Visualization, Concurrent Task Tree

1 Introduction

When developing an information system a precise specification is required to describe the system's behaviour. For this purpose, requirements' engineers interview stakeholders and end-users, and try to capture elicitated requirements in these software specifications. As a common problem such specifications do not always meet the actual requirements of the stakeholders. In particular, this results from communication problems between stakeholders and software engineers due to the different backgrounds and goals these two groups have. *End-User Development (EUD)* has tackled this challenge by enabling stakeholders to create software specifications on their own [1]. Usually, this is addressed by providing easy to understand specification techniques. One of them is the *Concurrent Task*

Tree (CTT), which constitutes a widely applied task modeling language for describing the behaviour of end-user interfaces [2]. More precisely, a stakeholder may specify a *hierarchical task model*, where lower level tasks refine upper level ones. Furthermore, *temporal relations* between tasks being on the same level may be introduced specifying the order in which these tasks shall be executed. A corresponding development environment is provided by the *Concurrent Task Tree Environment (CTTE)* [3]. In addition to its CTT modeling component, CTTE comprises a simulation component enabling stakeholders to check whether or not a CTT-based task model behaves as desired.

In *Business Process Management (BPM)* or – to be more precise – in the field of *Business Process Modeling* one has to deal with similar problems. Typically, numerous stakeholders are involved in the performance of a business process (i.e., its process instances and business cases respectively), whereas only few stakeholders are actually able to define respective process models. Furthermore, in practice process models often become very large and complex, and are not understandable to non-experts [4, 5].

When executing process models using a *Business Process Management System (BPMS)* it is desirable that all stakeholders are able to understand these process models or at least selected views on them [6]. Further they should be able to (dynamically) adapt process models if required, e.g., to deal with a changing environment or exceptional situations [7, 8]. Partially, the *Business Process Modeling Notation (BPMN) 2.0* tries to foster model comprehensibility by establishing an industry-wide and well-founded standard for process modeling [9]. Furthermore, it introduces advanced modeling elements and patterns in order to reduce model complexity for non-experts as well. Still, the problems coming with large process models remain and advanced visualization concepts are missing [10]. Respective concepts are required since process-aware information systems become increasingly adaptive and stakeholders need assistance in adapting their processes and process models respectively [11].

This paper addresses these issues and introduces a CTT-based approach for

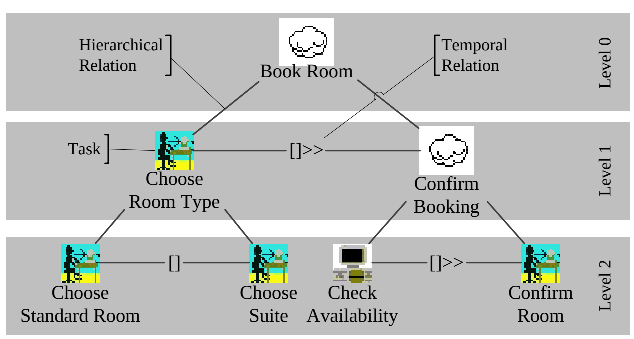

Fig. 1: Room Booking modeled in CTT [12]

specifying, visualizing and changing process models at a higher level of abstraction. It therefore provides a mapping of CTT modeling elements to BPMN elements and investigates how model changes can be accomplished in either one of these two formalisms. Further, a visualization concept is introduced to support stakeholders in viewing their process models.

This work was done in the context of the *proView* project, which enables stakeholders to adapt business process models based on personalized process views [13]. Such a personalized process view abstracts a business process model in a stakeholder-centered way, e.g. by only displaying those activities to a stakeholder he or she is involved in. Furthermore, *proView* provides alternative process representations to stakeholders (e.g., process graphs, forms or trees) such that they can choose the one being most favorable in their current work context. The overall goal is to assist stakeholders in understanding and adapting the business processes they are involved in.

The remainder of this paper is organized as follows: Section 2 introduces the structural modeling elements of CTT and then maps them to BPMN 2.0 elements. This mapping is required to support process modelers with a high-level specification approach on the one hand and to be still able to execute CTT process models in traditional BPMSs on the other hand. Opposed to [14] we do not provide a proprietary execution engine for CTT process models, but rather use CTT as stakeholder interface. Furthermore, we investigate what effects the changes of a CTT process model have on the corresponding BPMN model, and vice versa. Section 3 discusses how stakeholders may benefit from the use of CTT as process modeling language. In this context we also introduce CTT-specific visualization concepts. Section 4 discusses related work. The paper concludes with a summary and outlook in Section 5.

2 Using Concurrent Task Trees for Process Modeling

Section 2.1 first introduces basic modeling elements of CTT and shows how they can be mapped to BPMN elements. Section 2.2 then discusses how complex CTTs can be mapped to BPMN process models. Finally, Section 2.3 deals with changes on CTT process models.

2.1 Basic Modeling Elements

Concurrent Task Trees (CTTs) use *tasks* to describe changes of the state of the underlying information system. Thereby, a *task* represents work accomplished by the stakeholder or the underlying information system. Furthermore, tasks can be ordered and be structured hierarchically. Basically, there exist two kinds of relations between tasks: *Hierarchical Relations* and *Temporal Relations*. Figure 1 illustrates this using a room booking example. It describes the tasks to be accomplished in order to book a hotel room. Hierarchical relations connect tasks belonging to different tree levels, i.e., tasks having different granularity. Temporal relations, in turn, describe order and execution constraints of tasks belonging to the same level. We first consider tasks as basic building blocks of CTT. Generally, tasks are comparable to activities in a BPMN process model. As shown in Figure 2, CTT distinguishes between five different kinds of tasks.

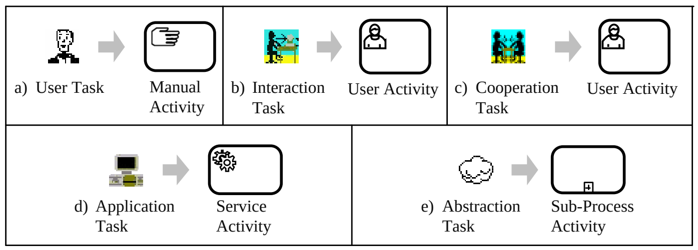

Fig. 2: CTT Task Types and their Mapping to BPMN

We describe these basic CTT task types as well as their mapping to BPMN 2.0 elements [9]. First, a *User Task* represents a task to be performed by a stakeholder without need for interacting with an information system (e.g., a clerk interviewing a customer). This kind of task can be directly mapped to a *Manual Activity* in BPMN. Second, if the stakeholder is interacting with an information system (e.g., filling data into a form), an *Interaction Task* can be used in CTT to express this. In particular, during process execution this kind of task has to be explicitly triggered by the stakeholder. Third, if more than one stakeholder is involved in a particular interaction, a *Cooperation Task* can be used. Both, Interaction and Cooperation Tasks can be mapped to a *User Activity* in BPMN. Note that BPMN does not distinguish between interactions of one or multiple stakeholders with the information system. Fourth, an *Application Task* describes a task to be executed by the information system without need for interacting with any stakeholder (e.g. storing data in a database). In BPMN this can be expressed by a *Service Activity*. Furthermore, BPMN offers specific alternatives to which a CTT Application Task may be mapped depending on what is done during the execution of the task. Either the BPMN *Send/Receive Activity* (e.g., if the task sends or receives messages) or the *Script Activity* (e.g., the task interprets a script) can be used. Fifth, an *Abstraction Task* covers activities not matching to any of the aforementioned task types. The semantics of an Abstraction Task is specified by refining it through child tasks. In BPMN this corresponds to a *Sub-Process Activity*. If an Abstraction Task has no child tasks, in turn, an *Abstract Activity* can be used in the corresponding BPMN process model instead.

Creating a task tree model based on CTT constitutes a top-down approach. The root task describes the name of the process and may be refined by lower-level tasks. The different levels are connected by *Hierarchical Relations*. In this context the type of a parent task is determined by the types of its child tasks. If all child tasks have the same type (e.g. Interaction Task) this type is assigned to the parent task as well. If the child tasks do not have the same type, however, the parent task will be an *Abstraction Task* (cf. *Confirm Booking* in Figure 1). We discuss how such task hierarchy can be mapped to a BPMN process model in Section 2.2.

After specifying the *Hierarchical Relations* between the tasks belonging to different tree levels, the *Temporal Relations* between the tasks referring to the same tree level have to be specified. Such relations describe the order in which the tasks of a particular level shall be executed. CTT offers eight kinds of temporal relations (cf. Figure 3). Relations *Enabling* and *Enabling with Information Passing* correspond to the *Sequence Flow* in BPMN. The latter relation sends additional information (i.e. data objects) to its target task (cf. Figure 3a+b). Furthermore, Figure 3c visualizes the temporal relation called *Iteration*, which can be mapped to a *Loop Activity* in the corresponding BPMN model. As specialization of Iteration, *Finite Iteration* can be used in CTT. It pre-specifies the number of iterations at design time.

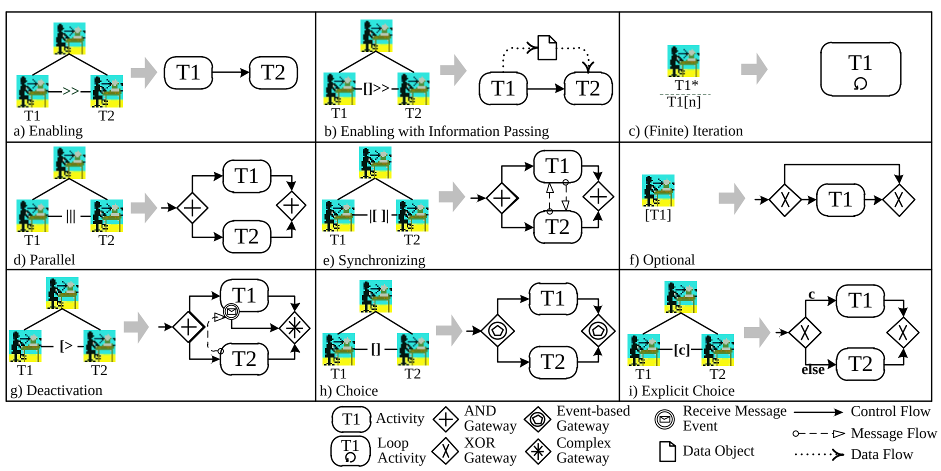

Fig. 3: Mapping CTT Temporal Relations to BPMN Model Elements

Three vertical lines between two tasks (cf. Figure 3d) express *Parallel* execution. Consequently, in the corresponding BPMN process model, the respective activities are surrounded by a splitting and joining AND-gateway. Since CTT realizes a task tree structure, CTT process models can be always mapped to a well-structured BPMN model, i.e., a BPMN process model for which every split gateway has a unique join gateway of the same type and vice versa. We denote such pair of gateways and their enclosed activities as *SESE* (Single-Entry-Single-Exit) block. SESE blocks may be nested, but must not overlap [15].

A special case of the aforementioned parallel relation is provided by the *Synchronizing* relation. It executes tasks in parallel, but these may exchange certain information with each other as well. In a BPMN process model this can be realized by sending messages from one activity to another.
An *Optional* CTT task (cf. Figure 3f) has to be mapped to a BPMN process fragment using an XOR-gateway, which then allows deciding whether or not the respective task shall be executed. *Deactivation*, in turn, executes the first task (cf. Figure 3g, task T1) as long as the second one is not started. However, when starting the second task the first one is aborted. A use case of this temporal relation is a user form that may be edited until the submit button is pressed. In a BPMN process model we can use an AND-gateway for expressing this scenario, i.e., both activities T1 and T2 become activated and hence may be started. Once T2 is started, it sends an event to T1 in order to abort its execution. Since it is *a-priori* not clear whether T1 is terminated normally or aborted, a complex gateway has to be used, allowing for the specification of respective join behaviour [9]. Finally, CTT offers a *Choice* relation between tasks; e.g. a user may decide whether T1 or T2 is executed. Once one of the two tasks is selected, the other one is no longer offered to the user. This behaviour can be mapped to an *Event-based* gateway in BPMN (cf. Figure 3h). Since this is the only way to express an alternative relation CTT offers, we extend the set of temporal relations by an *Explicit Choice* relation (cf. Figure 3i). The latter is needed in order to be able to express an alternative path based on data objects. In this context, we use a branching condition c within the square brackets between T1 and T2. The Explicit Choice relation can be mapped to an *Exclusive* gateway in BPMN.

2.2 Mapping Complex CTT Process Models to BPMN

Section 2.1 has shown how to map basic CTT elements to corresponding BPMN elements. Based on these elementary mappings, we informally show how CTT process models can be transformed into a behavior-equivalent[1] BPMN model. Obviously, the opposite mapping is not always possible, since not all BPMN models are well-structured or cannot be transformed into a behavior-equivalent, well-structured process model [16]. Besides, not all BPMN elements can be mapped to corresponding CTT elements (e.g. message flows, attached events).

When building and interpreting CTT process models, ambiguities might occur, particularly when introducing different kinds of temporal relations at the same level. As example assume that there are three tasks $T1$, $T2$ and $T3$ on the same tree level with: $T1$ [] $T2$ ||| $T3$; i.e., $T1$ and $T2$ are connected by a Choice relation, and $T2$ and $T3$ are connected by a Parallel relation. There are two ways

[1] A CTT process model and a BPMN model are denoted as *behavior-equivalent*, if they have exactly the same sets of producible execution traces. An execution trace reflects the temporal order of events (e.g. start, end activity) related to a process instance [11].

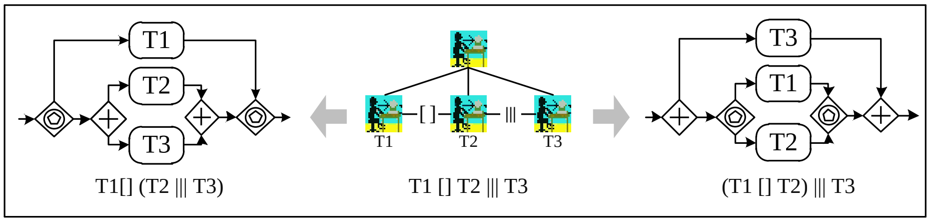

Fig. 4: Ambiguity in the Context of Temporal Relations

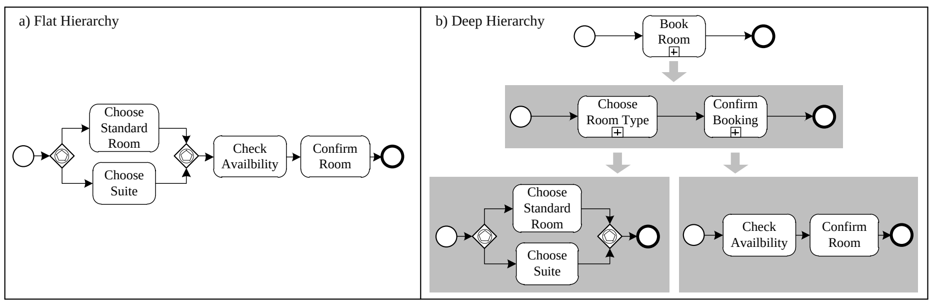

Fig. 5: Mapping the Room Booking CTT Process Model to a BPMN Model

to interpret these relations: $((T1 \; [] \; T2) \; ||| \; T3)$ or $(T1 \; [] \; (T2 \; ||| \; T3))$ (cf. Figure 4). This ambiguity has to be resolved since it affects both process semantics and the transformation of the CTT to the BPMN process model. CTT offers two options in this context [2]: First, the priority order defined in the ISO standard LOTOS may be used [17]; e.g., Choice > Parallel composition > Disabling > Enabling. Second, another tree level may be added to resolve the ambiguity, e.g. $T1 \; [] \; (T2 \; ||| \; T3)$. To support stakeholders in understanding CTT process models we prefer the second option since it requires no additional modeling knowledge.

When dealing with complex CTT process models the tree hierarchy needs to be transformed into the BPMN model. Generally, there exist several ways to accomplish this. The first one is denoted as **Flat Hierarchy**. Figure 5a shows the result of this transformation applied to the CTT process model from Figure 1. The tasks on the leaf level of the CTT process model are mapped to corresponding BPMN activities. Afterwards, the temporal relations between the tasks are transformed into the respective control flow patterns in the BPMN model. Finally, to connect the different branches the temporal relations of the parent task are used. The second way to map the tree hierarchy is to transform the CTT process model into a BPMN model using a **Deep Hierarchy** (cf. Figure 5b). Here, every tree level and every branch in the CTT process model is mapped to a separate BPMN model. In this context a task of the CTT process model

is mapped to a sub-process activity if the task has at least one child task (cf. Figure 3). Finally, the different models are assembled to a process hierarchy by connecting sub-process activities to the respective (sub-)process models.
Which of the two approaches for transforming a complex CTT process model into a BPMN model is more favorable depends on the concrete use case of the resulting BPMN model. For example, if one wants to integrate the model into a BPMS with the purpose to execute it, a flat hierarchy should be chosen. If the BPMN model is used to visualize the process for individual stakeholders, in turn, a deep hierarchy should be chosen. We discuss the second use case in more detail in Section 3.

2.3 Changing a CTT Process Model

Stakeholders should not only be supported in understanding process models, but also in changing and evolving them over time [8]. In this context, the use of CTT in the field of End-User Development has already shown promising perspectives. This section gives some insights into whether building and changing CTT process models is actually that easy as expected. Changes of a CTT process model can be categorized into three types. First, a change may concern *Temporal Relations*. Second, it may change the *Depth* of the CTT. Third, it may affect the *Breadth* of the tree. This is illustrated by Figure 6, which shows the progress during the modeling of a simple CTT process model as well as the respective modeling stages in the corresponding BPMN process model.

Changing the type of the **Temporal Relation** in the CTT process model influences the execution order of the respective tasks. This simple change of a CTT process model, however, might require complex adaptations of the corresponding BPMN model. In our example from Figure 6 the sequential ordering of tasks T1 and T2 is changed to a parallel one. While in the BPMN model this requires the insertion of AND-gateways for splitting and joining the control flow, in the CTT process model only the temporal relation is changed without need for modifying the tree structure.

The **Depth** of a CTT process model (i.e., the depth of its tree) may change when refining a task. More precisely, new child tasks may be added to a task and a new level be introduced in the tree. Depending on the level of the refined task, either the overall depth of the CTT process model or the depth of a specific subtree of the CTT process model increases. Generally, there are two alternatives to transfer a CTT process model change to the corresponding BPMN model depending on whether a flat or deep hierarchy is used (cf. Section 2.2). When using a flat hierarchy the activity of the refined task in the BPMN model is replaced by its newly added child tasks (cf. Figure 6). Using a deep hierarchy, in turn, the type of the refined task in the corresponding BPMN model is changed to a sub-process activity and a new sub-process is added to the BPMN process model hierarchy.

When deleting child tasks of a complex task in a CTT process model and only one or no child task remains, the respective child level is removed. Further-

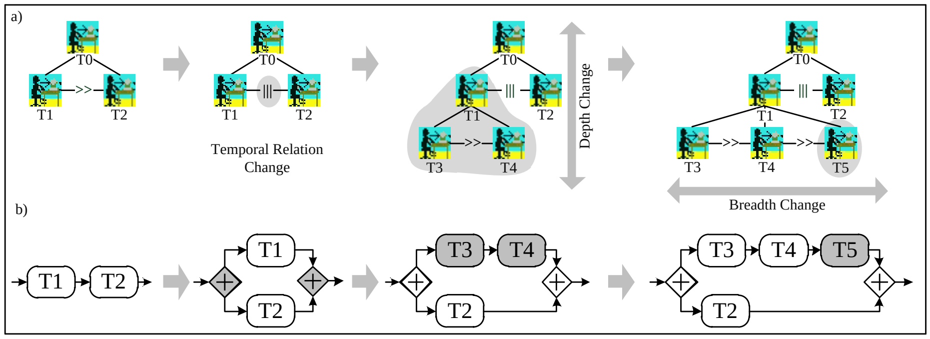

Fig. 6: Evolving Process Models in a) CTT and b) BPMN

more at least the depth of the affected sub-tree in the CTT process model is decreased by one. Depending on the chosen CTT-to-BPMN transformation, the sub-process activity will be removed or replaced by the parent task.

The **Breadth** of a sub-tree in a CTT process model changes when adding a new task to the respective level; e.g., when adding T5 to the CTT process model depicted in Figure 6. In general, adding a task requires adding a new hierarchical relation to its parent task as well as temporal relations to selected sibling tasks. To transfer the CTT process model change to a change of the corresponding BPMN process model, an activity needs to be added. Depending on the newly created temporal relations in the CTT process model, this activity is added sequentially or as new branch to existing split/join gateways. Obviously, when deleting a task in a sub-tree of a CTT process model, the breadth of this sub-tree is changed accordingly. If the deleted task has child tasks, in turn, these will be removed. In this case, the depth of the sub-tree is affected as well. As opposed to the deletion of a leaf task, deleting a non-leaf task requires the removal of a complete SESE block in the corresponding BPMN model. When using deep hierarchy (cf. Section 2.2) the sub-process activity as well as the underlying sub-process model are removed. If there is exactly one remaining task left on the tree level the deletion was applied, this level will be removed.

When mapping the effects of changes of a BPMN process model to the corresponding CTT process model, we need to consider whether a flat or a deep hierarchy is used (cf. Section 2.2). As advantage of a deep hierarchy the changes in the respective CTT process model can be kept local; note that each BPMN model represents a tree level in a specific branch of the CTT process model. Consequently, the change of the BPMN process model only concerns this rather small region of the CTT process model.

Using a flat hierarchy, in turn, even a simple movement of an activity from its current position to another one in the BPMN process model might require a complex restructuring of the CTT process model. For example, when moving

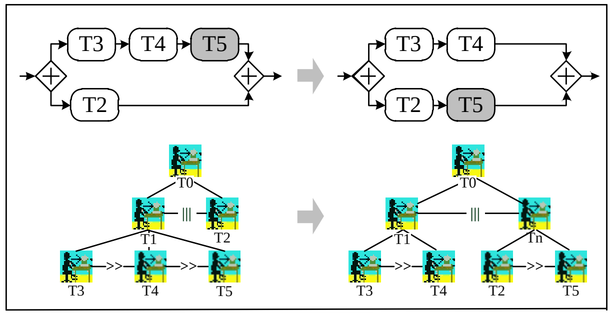

Fig. 7: Changes of a BPMN Process Model and Effects on the Corresponding CTT Process Model

task *T5* in Figure 7 from its current position to the one following task *T2*, a completely different tree structure results for the corresponding CTT process model. Obviously, the more the structure of a CTT process model is changed, the harder it will become for stakeholders to understand the effects of the change.

3 Supporting Stakeholders by Visualization Concepts

Section 2 discussed how models expressed in terms of activity-based process specification languages like BPMN can be mapped to CTT process models and vice versa. We now want to demonstrate how tree-based CTT process modeling supports stakeholders in understanding and interacting with a process model. Especially when dealing with large process models inexperienced stakeholders often have difficulties with comprehending and changing these models.

To foster model comprehension the tree-structure provided by CTT offers a top-down decomposition of the process model. In particular, any CTT process model can be visualized by folding/unfolding it as illustrated in Figure 8. Starting with Figure 8a the depicted CTT process model can be unfolded step-by-step by the stakeholder, e.g., using a zoom slider to refine the level of detail. Finally, when reaching the leaf level of the CTT process model (cf. Figure 8c), the most detailed level of the process model is shown to the stakeholder. We denote this unfolding as **Leveled Exploration**. In particular, this kind of refinement of CTT process models helps stakeholders in understanding or exploring process models step-by-step. When refining a BPMN process model by navigating to a sub-process, in turn, the related super-processes are no longer displayed to the stakeholder. Thus, the latter might loose the context of the sub-process. By contrast, this context is kept, when using Leveled Exploration in CTT. The second row in Figure 8 shows the BPMN process models corresponding to the respective CTT models. As opposed to other abstraction approaches (cf. Section

4) for each level of detail of the process model the respective sub-process activity labels are provided by the CTT process model; i.e., no label generation is required [18, 19].

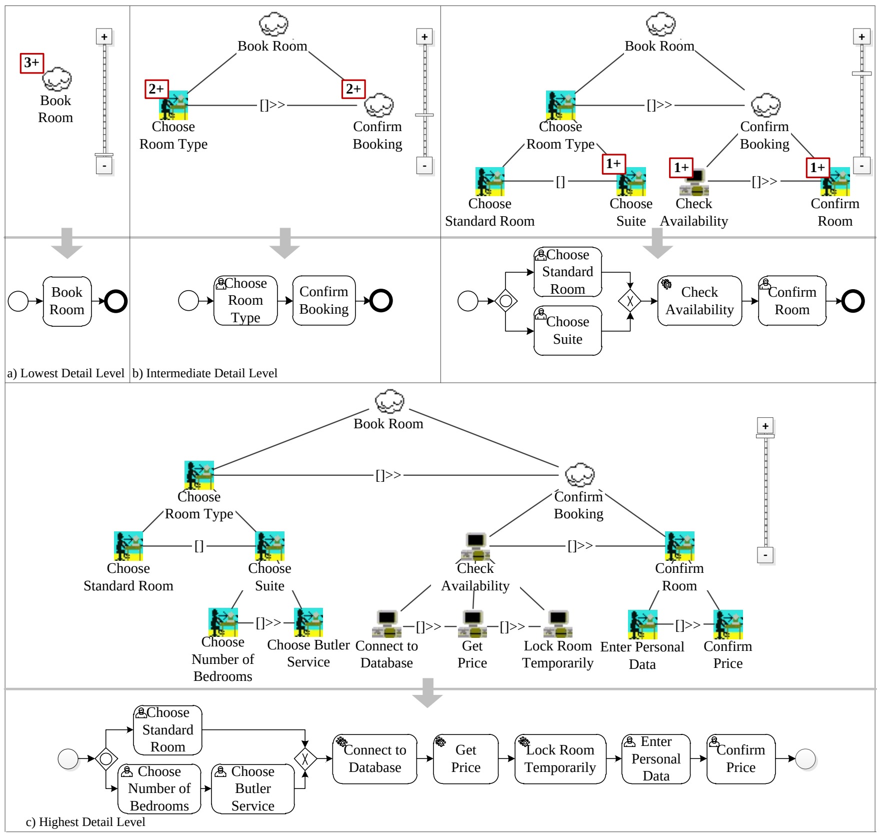

Fig. 8: Leveled Visualization of a CTT Process Model

A second method for exploring a CTT process model is **Depth Exploration**. In Figure 8b the stakeholder may click on one of the tasks *Choose Room Type* or *Confirm Booking*, and the corresponding sub-tree is then unfolded (cf. Figure 9a). Using this exploration method the stakeholder may interactively explore the CTT process model. Starting with an abstract (and small) CTT process model the stakeholder may decide which parts shall be refined. Clicking on the same task again will re-fold this sub-tree. This way the stakeholder may decide which region of the process model shall be explored at which level of granularity. Again a switch from the CTT process model to the corresponding BPMN process model

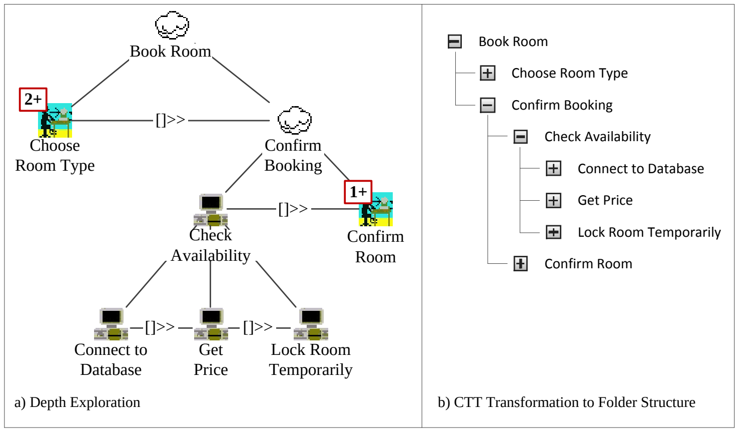

Fig. 9: CTT Process Model Visualization Concepts

is possible at every point in time during the exploration. As opposed to the use of traditional sub-processes in BPMN, one has the option to visualize selected sub-processes in more detail than others within one and the same process model. This, in turn, allows for a more flexible process visualization.

Our third exploration method provides a different representation of CTT process models to stakeholders. One already introduced example constitutes the **transformation** of a CTT process model to a BPMN process model (cf. Figure 5). As alternative representation of a CTT process model, a corresponding folder structure offers promising perspectives (cf. Figure 9b). In particular, stakeholders are familiar with respective folder structures.

4 Related Work

Similar to the approach desired in this paper, [20] introduces a mapping of CTT process models to UML 2.0 Activity Diagrams. Although UML provides a rich set of diagram types, none of them is supporting model-based design regarding the behavior of user interfaces. In order to bridge this gap the authors introduce a mapping of CTT modeling elements to UML 2.0 Activity Diagrams. Although UML Activity Diagrams can be used for modeling business processes, the suggested CTT mapping is more complex than in our approach; furthermore the resulting model would be too complex for stakeholders and end-users respectively. A similar approach showing the same drawbacks is introduced in [21].
The approach discussed in [22] uses BPMN for modeling to business processes. However, CTT is used to refine interactive activities in BPMN process models. More precisely, the stakeholder first models the business process using BPMN. Then, the activities requiring user interactions are refined similarly to the defi-

nition of a sub-process by defining a respective CTT process model.
Application Tasks (cf. Section 2.1) are mapped to Web Service calls and Interaction Tasks to a specific XML format enabling platform-independent user interfaces [23]. A similar approach is described in [24]. The authors suggest a four-step approach to derive user interfaces from business processes. As in the aforementioned approach, the refinement of an activity within a business process model is accomplished by defining a CTT process model. This CTT model and a domain model are then used to create an abstract user interface.
Another use case for CTT is presented in [14]. The authors introduce a method for modeling business processes based on CTTE (cf. Section 1). Afterwards the resulting CTT process model is transformed into an imperative process model which may be edited using a proprietary process editor (*Task Tree Workflow Management System Editor, TTMS Editor*). This way, the process model can be enriched with explicit choices and execution-relevant aspects. Finally, the process model is exported as XML file to a process engine (*TTMS WfMS*). This transformation described along the tool chain cannot be reversed. However, [14] neither provides an explicit mapping to an existing process modeling language nor does it support appropriate visualizing concepts.
In [25, 26] another tree structure, denoted as *Process Structure Tree* (PST), is introduced. On the one hand this tree type is used to analyze a process model in respect to control flow errors, on the other hand the PST structure is applied to detect SESE regions. The latter allows transforming certain classes of unstructured process models to well-structured ones [27]. However, stakeholder needs are not addressed by the PST notion.
Another way to support stakeholders in understanding and modeling business processes is illustrated in [13, 28, 29]: a powerful approach is introduced to change the visual appearance of a process model by replacing its visual representation by another one, e.g., by changing colors and adding pictographs to selected activities. Furthermore, advanced concepts for abstracting complex process models through hiding and aggregating process fragments is introduced [13, 30]. This allows for a more flexible visualization of processes, but also has its limitations. First, when aggregating process fragments, the resulting abstract activity requires an abstracted label, not provided by this approach. Second, the concept does not allow for the automated creation of abstract models as introduced in Section 3.

5 Summary and Outlook

In this paper we applied Concurrent Task Trees to Business Process Management. We showed that it is possible to map CTT to BPMN modeling elements. Especially, temporal relations can be transformed into behavior-equivalent BPMN process fragments. Complex CTT process models, in turn, can be transformed into flat or deep hierarchy process models in BPMN. We further discussed the effects of changes applied to a CTT process model and their translation into a behavior-equivalent BPMN process model. Finally, visualization concepts for

CTT process models were presented, which support stakeholders in understanding more complex process models. Since CTTs were designed with the goal to capture user interactions, we particularly suggest using CTTs in the context of human-centric business processes.
We developed a prototype to evaluate our concepts (cf. Figure 10). It extends the Cheetah Experimental Platform [31] and enables CTT-based process modeling. Cheetah is able to measure and log the order of arbitrary modeling steps and the time stakeholders need for accomplishing respective steps. Based on this, we will conduct user experiments for comparing CTT process modeling with BPMN-based modeling in a systematic way.

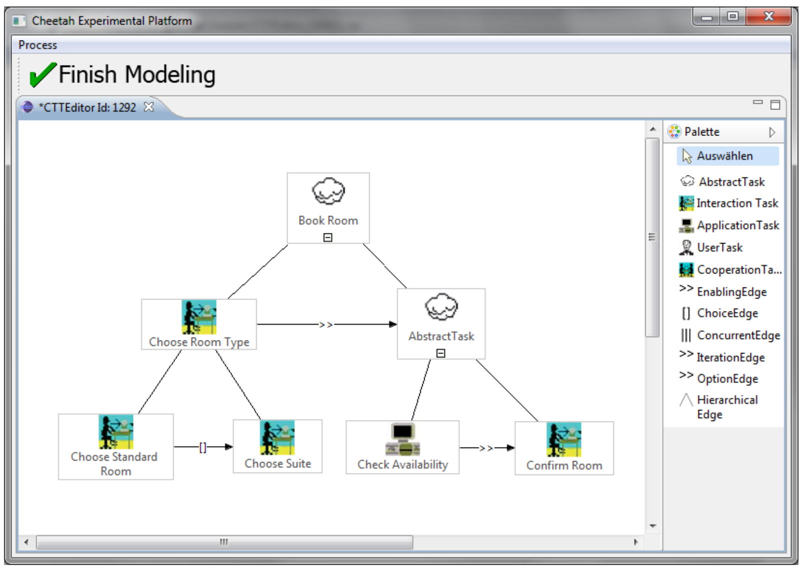

Fig. 10: Prototypical Implementation in Cheetah Experimental Platform

Acknowledgment

The authors would like to acknowledge and thank Thomas Grees for extending the Cheetah Experimental Platform by a CTT process modeling component as well as Yasemin Agbulak for providing first results on this topic.

References

1. Liebermann, H., Paternò, F., Wulf, V.: End-User Development (Human-Computer Interaction Series). Springer, Dordrecht (2006)

2. Paternò, F., Mancini, C., Meniconi, S., Maria, V.S.: ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In: Proc. IFIP TC13 Int'l Conf. on Human-Computer Interaction. (1997) 362–369
3. Mori, G., Paternò, F., Santoro, C.: CTTE : Support for Developing and Analyzing Task Models for Interactive System Design. IEEE ToSE **28** (2002) 797–813
4. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring Large Process Model Repositories. Computers in Industry **62** (2011) 467–486
5. Rinderle, S., Bobrik, R., Reichert, M., Bauer, T.: Businesss Process Visualization - Use Cases, Challenges, Solutions. In: Proc. 8th Int'l Conf. on Enterprise Information Systems (ICEIS'06). Volume 2006., Paphos, Cyprus (2006) 204–211
6. Weber, B., Mutschler, B., Reichert, M.: Investigating the Effort of Using Business Process Management Technology: Results from a Controlled Experiment. Science of Computer Programming **75** (2010) 292–310
7. Reichert, M., Rinderle-Ma, S., Dadam, P.: Flexibility in Process-aware Information Systems. LNCS Transactions on Petri Nets and Other Models of Concurrency (ToPNoC) **2** (2009) 115–135
8. Weber, B., Sadiq, S., Reichert, M.: Beyond Rigidity - Dynamic Process Lifecycle Support: A Survey on Dynamic Changes in Process-Aware Information Systems. Computer Science - Research and Development **23** (2009) 47–65
9. OMG: Business Process Management Notation (BPMN) 2.0 (2010)
10. Bobrik, R., Reichert, M., Bauer, T.: Requirements for the Visualization of System-Spanning Business Processes. Proc. DEXA'05 Workshops (2005) 948–954
11. Reichert, M., Weber, B.: Enabling Flexibility in Process-aware Information Systems - Challenges, Methods, Technologies (to appear). Springer (2012)
12. Paternò, F.: ConcurTaskTrees : An Engineered Approach to Model-based Design of Interactive Systems. The Handbook of Analysis for HumanComputer Interaction (1999) 1–18
13. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: Proc. 26th Symposium On Applied Computing (SAC'12), Riva del Garda (Trento), Italy (2012)
14. Brüning, J., Forbrig, P.: TTMS : A Task Tree Based Workflow Management System. In: Proc. BPMDS'11, Springer Berlin / Heidelberg (2011) 186–200
15. Reichert, M., Dadam, P.: ADEPTflex - Supporting Dynamic Changes of Workflows Without Losing Control. Journal of Intelligent Information Systems **10** (1998) 93–129
16. Liu, R., Kumar, A.: An Analysis and Taxonomy of Unstructured Workflows. In: Proc. 5th Int'l Conf. on Business Process Management. (2005) 268–284
17. ISO 8807: Information Processing Systems - Open Systems Interconnection - LOTOS: A Formal Description Technique based on the Temporal Ordering of Observational Behaviour. (1989)
18. Polyvyanyy, A., Smirnov, S., Weske, M.: Process Model Abstraction : A Slider Approach. In: 12th Int'l IEEE Enterprise Distributed Object Computing Conference, IEEE (2008) 325–331
19. Hipp, M., Mutschler, B., Reichert, M.: Navigating in Process Model Collections: A new Approach Inspired by Google Earth. In: Proc. BPM'11 Workshops, Clermont-Ferrand, France (2011)
20. Nóbrega, L., Nunes, N.J., Coelho, H.: Mapping ConcurTaskTrees into UML 2.0. In: Interactive Systems. Design, Specification, and Verification. (2006) 237 – 248
21. Brüning, J., Dittmar, A., Forbrig, P., Reichart, D.: Getting SW Engineers on Board: Task Modelling with Activity Diagrams. In: Proc of Engineering Interactive Systems 2007 (EIS'07), Salamanca, Spain (2008) 175–192

22. Pintus, A., Paternò, F., Santoro, C.: Modelling User Interactions in Web Service-based Business Processes. In: WEBIST'10. (2010) 175–180
23. Paternò, F., Santoro, C., Spano, L.D.: Model-Based Design of Multi-device Interactive Applications Based on Web Services. In: Human-Computer Interaction INTERACT 2009. (2009) 892–905
24. Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., Vandermeulen, J.: User Interface Derivation from Business Processes: A Model-Driven Approach for Organizational Engineering. In: SAC'08. (2008) 553–560
25. Vanhatalo, J., Hagen, V., Leymann, F.: Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. (2007) 43–55
26. Vanhatalo, J., Völzer, H., Koehler, J.: The Refined Process Structure Tree. Lecture Notes in Computer Science **5240** (2008) 100–115
27. Polyvyanyy, A., García-Bañuelos, L., Dumas, M.: Structuring Acyclic Process Models. In: Proc. 8th Int'l Conf. on Business Process Management, New York, USA (2010) 276–293
28. Bobrik, R., Bauer, T., Reichert, M.: Proviado - Personalized and Configurable Visualizations of Business Processes. In: Proc. 7th Int'l Conf. Electronic Commerce & Web Technology (EC-WEB'06), Krakow, Poland (2006) 61–71
29. Reichert, M., Bassil, S., Bobrik, R., Bauer, T.: The Proviado Access Control Model for Business Process Monitoring Components. Enterprise Modelling and Information Systems Architectures - An International Journal **5** (2010) 64–88
30. Bobrik, R., Reichert, M., Bauer, T.: View-Based Process Visualization. In: Proc. 5th Int'l Conf. on Business Process Management, Brisbane, Australia, Springer (2007) 88–95
31. Pinggera, J., Zugal, S., Weber, B.: Investigating the Process of Process Modeling with Cheetah Experimental Platform. In: Proc. 1st Int'l WS Empirical Research Proc.-Oriented Inf. Sys., Hammamet (2010)