Workflow Resource Pattern Modelling and Visualization

Hanwen Guo, Ross Brown, Rune Rasmussen
Information Systems School, Science and Engineering Faculty
Queensland University of Technology
GPO 2343, Brisbane 4001, QLD

hanwen.guo@student.qut.edu.au, r.brown@qut.edu.au, r.rasmussen@qut.edu.au

Abstract
Workflow patterns have been recognized as the theoretical basis to modeling recurring problems in workflow systems. A form of workflow patterns, known as the resource patterns, characterise the behaviour of resources in workflow systems. Despite the fact that many resource patterns have been discovered, people still preclude them from many workflow system implementations. One of reasons could be obscurity in the behaviour of and interaction between resources and a workflow management system. Thus, we provide a modelling and visualization approach for the resource patterns, enabling a resource behaviour modeller to intuitively see the specific resource patterns involved in the lifecycle of a workitem. We believe this research can be extended to benefit not only workflow modelling, but also other applications, such as model validation, human resource behaviour modelling, and workflow model visualization.

Keywords: Workflow Resource Patterns, Modelling, Visualization.

1 Introduction
Presently, people often use workflow modeling languages to describe their business environment (van der Aalst and Hofstede 2005). Conventionally, a workflow system can be understood from the control, resource and data perspective (van der Aalst, Hee et al. 1994). The resource perspective represents responsibilities, behaviour and the organizational structure of workflow resources within a business environment.

Human resource behaviour is one important component in the resource perspective. This is because the behaviour of resources can affect the efficiency of an organization (Moore 2002; zur Muehlen 2004). In the workflow domain, people have already indentified some patterns to describe behaviour of human resource, and have used these patterns to solve human resource behaviour related problems in a workflow system (Russell, van der Aalst et al. 2005).

Russell et al. have defined a group of resource patterns (Russell, van der Aalst et al. 2005), describing various human resource task allocation, execution manner and interaction mechanisms between human resources. These descriptions can be grouped into two categories, that is, “single resource to single task” and “many resources to many tasks”.

A modelling approach that can represent this resource-task logic or resource patterns is quite necessary. This is because such a modelling approach can provide people with a clear view about the relationship between resources and tasks, that is, resource-task logic. With such a logic representation, modern workflow management systems, such as YAWL (van der Aalst, Aldred et al. 2004), can support and automate such resource patterns.

In the modelling domain, there are various modelling approaches, such as Petri net (Pesic and van der Aalst 2007) and BPEL4PEOPLE (an extension of BPEL) (Russell and van der Aalst 2008) that can be used in representing such resource logic. These modelling approaches usually have visual representations. These visual representations usually employ 2D conceptual shapes, such as circles, arrows and rectangles to indicate workflow logic.

Indeed, these visual representations can impede the communication between business analysts and stakeholders (Shannon and Weaver 1963; Sadiq, Indulska et al. 2007; Moody 2009). This is because stakeholders usually don’t hold necessary knowledge about modelling grammars (V. Khatri, I. Vessey et al. 2006), and some empirical studies show that such a simple representation can reduce the cognitive load required for understanding in the human brain, especially for naïve stakeholders who are not quite familiar with specialist visual grammars (J. Parsons and Cole 2005; Burton-Jones, Wand et al. 2009).

According to relevant research, the communication between business analysts and stakeholder has been recognized as a key critical success factor in success of business process management projects (Nah, Lau et al. 2001; Trkman 2010). This implies that an ineffective communication approach may results in the failure of a process modeling and improvement exercise, as a workflow management solution may be implemented sub optimally, resulting in an inefficient organization.

From the discussion above, we can say that a well established resource behaviour modelling approach with an easily understood visual representation can not only enable managers to understand the relationships between workflow resources and tasks in the workflow system and facilitate workflow management system development, but it can also benefit communication between business analysts and stakeholders, improving implementation outcomes.

Therefore, in this paper we propose a resource behaviour oriented modelling and visualization approach for resource patterns (Russell, van der Aalst et al. 2005). The modelling approach is based on an automated planning technique, known an Hierarchical Task Network (HTN) (Erol, Hendler et al. 1994). Such a modelling approach supports a decomposition mechanism, whereby some simple resource patterns can be automatically assembled up to represent some complex resource patterns, enabling a “many workitem to many resources” relationship to be modelled. We employ the virtual world as our visualization approach. Virtual worlds are a synthetic replication of the real world (Burdea and Coiffet

2003). With a mapping mechanism between HTN modelling results and a virtual world, the modelling results can be represented as an intuitive, easy to understand animation. Such an intuitive visualization enables people to recall and cognate about conceptual and concrete content in the business process system, facilitating the communication process with regards to analysing, modelling and validating human resource behaviours. We show a simple example, visualizing a push resource pattern in Figure 1, below.

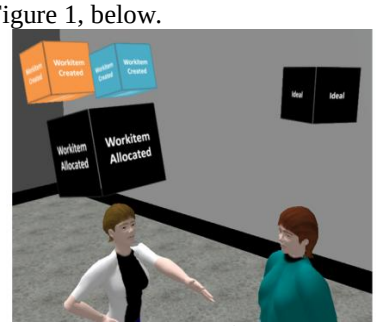


Figure 1 visualization representation of a push resource pattern. The cubes above the avatars' heads indicate the current states of workitems, and the allocation task is illustrated with the body movement of the avatar, indicating a state transition of a workitem.

This paper is organized as follows: Section 2 discusses related work within the control, resource and data perspectives. Section 3 discusses our HTN based resource pattern modelling and visualization approach. Section 4 utilizes a health care scenario to demonstrate the modelling and visualization ability of our approach with multiple resources and workitems. Finally, Section 5 discusses our further work.

2 Related Work

2.1 Control Perspective

Regarding the control perspective, some researchers (Lu, Bernstein et al. 2006; Rasmussen and Brown 2012) model tasks in the workflow system as an operator with the form $op = <p, q, v>$. Item v is list of parameters, while items p and q are two assertions indicating the execution of operator op must satisfy the p (pre-conditions) and can establish the post-conditions q, invoking the state transition from the state containing p to the state containing q in the workflow system. In their approach, search algorithms are used to continually select a suitable operator whose pre-conditions are compatible with current state. With several iterations, the search algorithms will find a serial list of sequenced operators, and the execution of operators can lead the workflow system to transition from an initial state to a goal state. Modellers can use these sequenced tasks to represent the workflow model.

2.2 Data Perspective

A similar approach has been used within the data perspective. Some researchers (Nigam and Caswell 2003; Wang and Kumar 2005; Bhattacharya, Gerede et al. 2007) utilize business artefacts or documents in the workflow system to automatically construct workflow models. These document centric approaches recognize that tasks in the workflow system are services requiring input data from the task executors which generate related output data. These inputs and outputs are regarded as clues of task execution records, being used to derive the temporal and logic ordering of tasks. For example, an online shopping activity may contain an order list, a confirmation letter, an invoice and a confirmation slip of reception. Based on their occurrence, the ordering logics of item selection, payment and delivery can be derived. With these relations, the occurrence ordering and dependency of these tasks can be derived, and linkages between these tasks can be used to represent a workflow model.

2.3 Data Perspective

In a resource perspective, most of the research work focuses on resource modelling and resource utilization issues. Zur Muehlen (zur Muehlen 1999) states that a resource model usually contains two parts: assignment policies and resource details. He points out that most modelling approaches do not consider that the resource details should facilitate the assignment policies on the one hand, and ignore the importance of non-human resources in the workflow system on the other hand. Therefore, he proposes a generic meta-model that can not only represent any resources in the workflow activity, but also facilitates the assignment policy implementation and execution.

Pesic and van der Aalst (Pesic and van der Aalst 2007) focus on task distribution issues in the workflow system. They proposed a basic model which contains the work distribution and work list model. These two modules can interact with each other to simulate the process of workitem distribution, and the internal mechanisms of these two modules are modelled using Petri nets.

It can be concluded that these works (Nigam and Caswell 2003; Wang and Kumar 2005; Lu, Bernstein et al. 2006; Bhattacharya, Gerede et al. 2007; Rasmussen and Brown 2012) focus on the automated model construction mechanism in the workflow system from different perspectives. Zur Muehlen (zur Muehlen 1999) deals with the static structural description of resource properties, while Pesic and van der Aalst (Pesic and van der Aalst 2007) describe the dynamics aspect of work distribution.

In this paper, we intend to automate the execution of a single workitem, rather than the entire workflow model construction approaches which have been proposed in the papers (Nigam and Caswell 2003; Wang and Kumar 2005; Lu, Bernstein et al. 2006; Bhattacharya, Gerede et al. 2007; Rasmussen and Brown 2012). Similar with the work done by Pesic and van der Aalst (Pesic and van der Aalst 2007), we focus on the dynamics aspects of a workitem, describing resource interactions around an allocated workitem. The difference between our approach and others (Pesic and van der Aalst 2007) is that we employ an HTN to automatically model interaction mechanisms involved in a workitem execution. The benefit of utilizing an HTN as a modelling approach is that an HTN can automatically generate rational interactions between human resources, if pre-conditions and post-conditions of each interaction are provided. In addition, we address the visualization issues that have not been addressed in these papers (zur Muehlen 1999; Nigam and Caswell 2003; Wang and Kumar 2005; Lu, Bernstein et al. 2006; Bhattacharya, Gerede et al. 2007; Pesic and van der Aalst 2007). These papers only consider modelling aspects, rather than communication aspects of the modelling approach. Their graphical representations are in a highly abstracted 2D diagram, which will likely puzzle naïve

stakeholders who have less professional knowledge about specific modelling languages. In contrast, we provide a "hands on" representation manner for model readers, by defining a mapping system between modelling results and a virtual world. Such a manner is expected to be intuitive and easily understood by stakeholders.

3 Resource Pattern Modelling and Visualization

We are going to describe an HTN based modelling and visualization approach for resource patterns. We briefly introduce resource patterns in Section 3.1, and discuses how to use the Hierarchical Task Network (HTN) to model resource patterns in Section 3.2. We then continue with a brief introduction of virtual worlds in Section 3.3, followed by the mapping mechanism between the HTN modelling results and a virtual world. Lastly, we discuss implementation issues with this approach in Section 3.4.

3.1 Resource Pattern as State-transitions

Resource patterns (Russell, van der Aalst et al. 2005) can be categorized into seven groups, namely: Creation, Push, Pull, Detour, Auto-start, Visibility and Multiple Pattern, respectively, see brief description in Table 1.

Pattern Category	*Brief Description*
Creation Pattern	Workitem creation mechanism in a workflow management system
Push Pattern	Workitem allocation mechanism in workflow management system
Pull Pattern	Workitem acquisition mechanism in workflow management system
Detour Pattern	How a workitem is related to another resource
Auto-start Pattern	How one workitem can trigger the execution of other workitems
Visibility Pattern	Visibility of committed workitems with respect to other resources
Multiple resource Patterns	Coordination mechanism between multiple resource execution

Table 1 The brief description of pattern category.

These patterns can characterize the behaviour of workflow management systems and workflow resources in the lifecycle of a workitem, in Russell et al. these patterns belong to two relationship groups, viz., "single workitem to single resource" and "many workitems to many resources". Some evidence (Pesic and van der Aalst 2007) utilizing Petri-net to model resource patterns shows that resource patterns are state-transitions. Here, we also consider resource patterns as state-transitions, but we will use different mechanism to model them.

The "single workitem to single resource" relationship involves the Creation Pattern, Push Pattern, Pull Pattern, Detour Pattern, and Auto-start Pattern. The selected visualizations of these patterns are available in Figure 2, and readers who are interest in resource pattern visualizations are suggested to read their original paper (Russell, Hofstede et al. 2004). Within these five resource pattern categories, the lifecycle of a workitem begins at the created state and ends at a failed or completed state. For example, the push patterns can be represented via three state transitions, which are state transitions from created state to offered state to single resource, created state to allocated state to a single resource, and created state to offered to multiple resources (see the three red dash lines with arrows in Figure 2). It can be said that the essence of resource patterns are actual state transitions in the lifecycle of a workitem. The life cycle of a workitem can be viewed as a sequence of resource patterns, transiting a workitem from the created state to completed or failed state (see the dash rectangle in Figure 2, where a creation pattern, push pattern and pull pattern occur, consequently). Therefore, patterns in the "single workitem to single resource" relationship can be modelled as state-transitions.

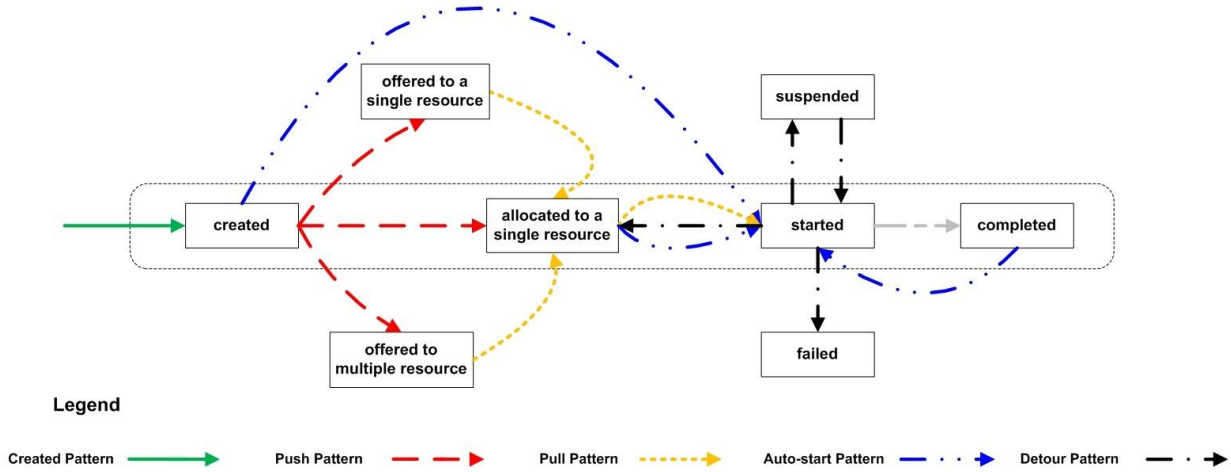


Figure 2 is a visual representation of the resource pattern. The rectangles are used for representing the states, and arrows are used for the transitions. Some of the resource patterns have been omitted for clarity.

Similar to patterns in the "single workitem to single resource", we consider resource patterns in the "many workitems to many resources" relationship can be modelled as state-transitions. Russel et al. (Russell, Hofstede et al. 2004) discuses two multiple resource patterns, Additional Resources Pattern (Pattern R-AR) and Simultaneous Execution Pattern (Pattern R-SE). The Pattern R-AR describes the behaviour of a resource requiring assistance from additional resources when this resource is dealing with a workitem, while Pattern R-SE describes the behaviour of several resources processing the same workitem at the same.

We utilize these two patterns placing them into Scenario 1, and then use this scenario to demonstrate the applicability of using a state transition mechanism to model the Multiple Resource Pattern.

Scenario 1 *There is a workitem W_o. It is created and allocated to a resource R_a by a workflow engine. Resource R_a started the workitem, and then divided Wo as three sub-workitems W_a, W_b and W_c to himself and two subordinates R_a, R_b and R_c, respectively. The execution of sub-workitem W_a is dependent on the results of W_b and W_c. R_a allocates the W_b to R_b without negotiation, R_b has to executed W_b immediately. R_a allocates the W_c to R_c with negotiation, R_c can select an appropriate time to execute it. As these two sub-workitems have been completed and reported back to R_a, R_a can start to execute the W_a. When W_a is finished, the original workitem W_o can be accomplished and checked back to workflow engine.*

Scenario 1 describes four nested workitems. Their state transitions are different from flat ones we discussed previously, that is, these state transitions are in a hierarchical structure.

At a very high level, the lifecycle of this workitem W_o can be understood as two states (initial and finial) with an execution phase. The execution phase also involves several states. That is, workitem W_o is created and allocated to the resource R_a, and then it is the resource R_a and the other two additional resources R_b and R_c that jointly complete it. The state transition transiting the workitem W_c from to the completed state can be further investigated. According to the description, workitem W_o can be divided as three sub-workitems W_a, W_b and W_c. The life cycles of these sub-workitems consist of different state-transitions or resource patterns. For example, the life cycle of workitem W_c involves two state-transitions or two resource patterns, that is, a creation pattern transiting workitem W_c from created state to started state, and an auto-state pattern transiting workitem W_c from started state to completed state. This will be true when considering lifecycles of W_a and W_b. The execution of W_a and W_b can be started simultaneously. In particular, the execution of W_a and W_b can be the Pattern R-SE, if we recognize R_a and R_b are the same resource.

We illustrate these state-transitions in a top-bottom view, see Figure 4. It can say that a decomposition mechanism enables us to analyse the state-transitions of nested workitems. In other words, patterns in the Multiple Resource Pattern category can be represented as state-transitions in a hierarchical structure.

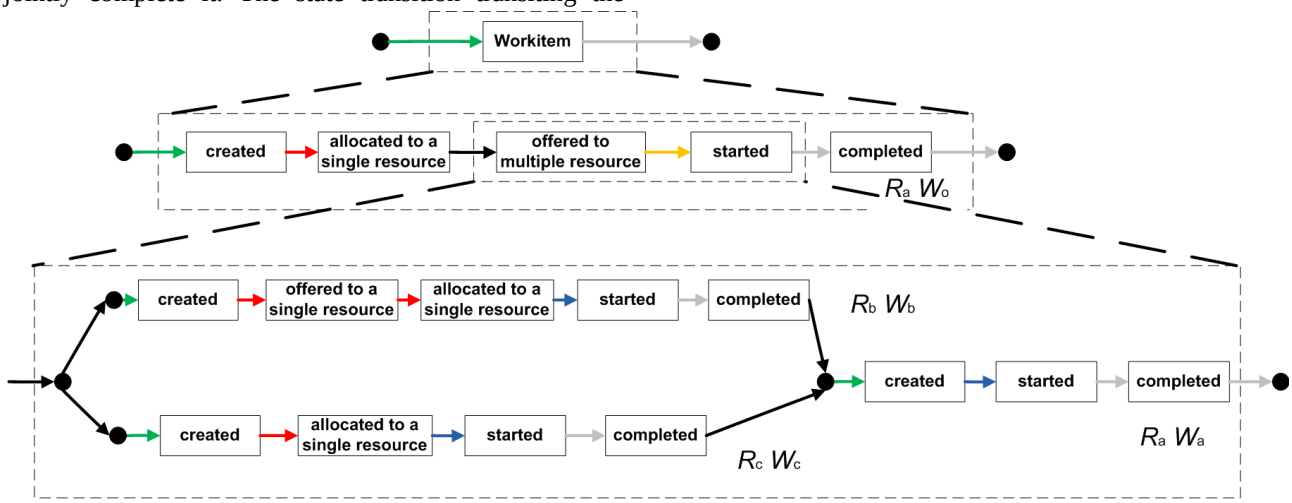


Figure 3 a top-bottom view of the state transitions in the workitem W_o.

3.2 HTN Modelling Approach

Conventionally, it is believed that Erol et al. (Erol, Hendler et al. 1994) first provided a clear theoretical framework for an HTN. There are two types of tasks in their HTN framework, namely complex and primitive tasks. The execution of a primitive task or a complex task can lead the system transit from a state to another, but the execution mechanism of these two types of tasks are different. In practice, the executions ordering of primitive and complex task are constrained by a task network, and a decomposition mechanism may involved substantial computational effort. We believe these two types of tasks, task network and decomposition mechanism can be used to represent resource patterns. Table 2 indicates a mapping mechanism between resource patterns and an HTN.

Resource Pattern	***HTN framework***
"single resource to single workitem" resource patterns	Primitive task
"multiple resources to multiple workitems" resource patterns	Complex task
Workitem life cycle	Task network

Table 2 mapping mechanism between Resource Pattern and HTN framework.

In the following we select some necessary concepts for introduction. Readers who are interest in the full syntax and semantics of HTNs are suggested to read the original paper (Erol, Hendler et al. 1994).

A primitive task is a task that can be directly solved by the task execution. It can be modelled within a form $op = <p, q, v>$. The satisfaction of pre-conditions p enables operator execution, and the operator execution enables the establishment of post-conditions q. This means the operator execution enables a state transition from the state containing pre-conditions p to the state containing post-conditions q.

A complex task can be recognized as the aggregation of primitive tasks. Such a complex task cannot be solved by task execution directly, but by requiring the decomposition before the execution. That is, using a set of primitive tasks to represent this complex task, the execution of the complex task is equivalent to execution of all primitive tasks. The state transition triggered by the execution of a complex task is equivalent to the aggregation of state transitions of selected primitive tasks. For example, a complex task ct is a complex task, being composited by three primitive tasks pt_1, pt_2 and pt_3. The execution of the ct is the execution of pt_1, pt_2 and pt_3. The

pre-conditions of the firstly executed primitive task or primitive tasks should not violate the state s_i before the execution of *ct*, and the state s_t after the execution of *ct* is depended by the post-conditions of finally executed primitive task or tasks.

In short, the complex task needs to be resolved by a task network. The task network is an array where some states and task sets are alternatively placed. It can be modelled in a form $[tn\_name, [(t_1, tl_1) \dots\dots (t_n, tl_n)], \mu]$, where tn_name is the name of this task network, t_i and tl_i are the name and label of a task, μ is a formula defining the partial ordering of tasks and states. Usually, a task network has two functionalities, decomposing a complex task and defining logical ordering of tasks.

A complex task can be modelled in a form $me = < ct, tn, mp, mq >$, where ct is the name of the complex task, tn is the corresponding task network, and mp and mq are high-level pre- and post-conditions of the sequenced primitive tasks in the task network tn, respectively. That is, a method $me = < ct, tn, mp, mq >$ can be selected for the complex task ct, if and only if the current and target state contains the mp and mq, respectively.

3.3 Modelling Resource Patterns with HTN

We will enumerate a number of HTN modelled resource patterns to prove the possibility of using HTN as the modelling approach for resource patterns in this section.

Patterns in the Creation Pattern, Push Pattern and Pull Pattern are relatively simple. The common feature of these patterns is that they can transit one state to another without further decomposition. For example, pull patterns characterize the transition from the allocated or offered state to started state, characterising the proactive behaviour of resources selecting a suitable workitem to execute. Thus, we provide basic HTN modelling results for them, see Table 3.

Task Name	***Task Network Details***
basic_task	$[basic\_task\_network, [(t, tl)], \mu]$ $\mu = \{(S_I \prec tl \prec S_T)\}$
REMARKS	
The *basic_task* is a primitive task that can be used to represent the resource patterns with two states and one transition. The primitive task *t* in it can be implemented as requirements mentioned in the Creation Pattern, Push Pattern, and Pull Pattern.	

Table 3 the basic task and task network that can be used to model patterns in Creation Pattern, Push Pattern, Pull Pattern.

Most patterns, for example, the Detour Pattern and Auto-start Pattern, transit a workitem from one state to another. They can be modelled by the basic task network in Table 3. However, there are some patterns, in these two categories, requiring a decomposition mechanism. We have to model these patterns individually. These patterns are the Stateful Reallocation Pattern (Pattern R-PR) and Stateless Reallocation Pattern P-UR (Pattern P-UR) in the Detour Pattern group, Piled Execution (Pattern P-PR) in the Auto-start Pattern group, as well as the Simultaneous Execution Pattern (Pattern R-SE) and the Additional Resources Pattern (Pattern R-AR) in the Multiple Resources Pattern. These patterns should be modelled by complex tasks and decomposition mechanisms.

Pattern R-PR (Stateful Reallocation) and Pattern P-UR (Stateless Reallocation) are different in functionality. Pattern R-PR requires the state information of a workitem being kept when this workitem is reallocated to another resource, while Pattern P-UR doesn't have such a rule. However, their modelling result can be illustrated in a similar manner. Tasking Pattern R-PR (Stateful Reallocation) as an example, at the top level, a task *reall_task* is needed to transit workitem from the started state back to the allocated state. It should be noticed that this task network contains a primitive task *exe_task* and a complex task *next_task*. The primitive task *exe_task* enables the execution situation to be recorded, the complex task *next_task* can be interpreted by a task network as *next_step_a* or *next_step_b*, see details in Table 4. In particular, modelling results in Table 4 can be used as a reference to model the Pattern P-UR (Stateless Reallocation) by implementing *exe_task* as a function that doesn't record the execution information.

Task Name	***Task Network Details***
reall_task	$[reall\_with\_state, [(exe\_task, tl_e), (next\_task, tl_n)], \mu]$ $\mu = \{(S_S \prec tl_e \prec S), (S \prec tl_n \prec S_A)\}$
next_task	$[next\_step\_a, [(exe\_task, tl_e), (next\_step, tl_n)], \mu]$ $\mu = \{(S_I \prec tl_e \prec S), (S \prec tl_n \prec S_A)\}$
next_task	$[next\_step\_b, [(all\_task, tl_a)], \mu]$ $\mu = \{(S \prec tl_a \prec S_A)\}$
REMARKS	
reall_task is a task network that can reallocate workitems from one one resource to another, involving one primitive task *exe_task* and a complex *next_task*. The *exe_task* is an executable function that can records the execution state information, while *next_step* can be interoperated by two different task networks, *next_step_a* and *next_step_b*. the *next_step_a* enables a resource to further execute the workitem, *next_step_b* enables a resource to reallocate the workitem to another resource.	

Table 4 the modelling result of Pattern R-PR (Stateful Reallocation).

Pattern P-PE (Piled Execution) in the Auto-start Pattern is a pattern that enables a resource to execute workitems in batch. A HTN modelling results of Pattern P-PR is available in Table 5. In this modelling result, there are two tasks. The task *pile_all* enables a resource to recognize the incoming tasks, while the task *pile_cpl* enables the resource to start processing and complete these allocated workitem.

Task Name	***Task Network Details***
pile_task	$[pile\_network, [(pile\_all, tl_a), (pile\_cpl, tl_c)], \mu]$ $\mu = \{(S_I \prec tl_a \prec S), (S \prec tl_c \prec S_C)\}$
pile_all	$[pile\_all\_network, [(t_1, tl_1), (t_2, tl_2) \dots\dots (t_m, tl_m)], \mu]$ $\mu = \{(S_I \prec tl_i), (tl_i \prec S_S)\}$, where $i = 1,2 \cdots\cdots m$
pile_cpl	$[pile\_cpl\_network, [(t_{m+1}, tl_{m+1}) \dots\dots (t_{2m}, tl_{2m})], \mu]$ $\mu = \{(S_S \prec tl_i), (tl_i \prec S_C)\}$, where $i = 1,2 \cdots\cdots m$
REMARKS	
plie_task being interoperated by *pile_network* is the task that enables a resource to execute workitems in a batch. Such a task can be divided into two parts, namely *pile_all* and *pile_cpl*. The task *pile_all* enables all involved workitems transit from some state S_I to the started state S_S in a partial order, while *pile_cpl* enables all involved workitems in the started state S_S to the state S_C where all workitems are completed.	

Table 5 the modelling result of Pattern R-PE (Piled Execution).

Pattern R-SE (Simultaneous Execution) and Pattern R-AR (Additional Resources) are two patterns in the Multiple Resource Pattern, characterizing the "many workitems to many resources" relationship.

Pattern R-SE (Simultaneous Execution) requires that one single resource can manipulate multiple workitems in a period. We believe the Pattern P-PE (Piled Execution) is a particular type of Pattern R-SE. This is because those two patterns require that one single resource can deal with multiple workitems at the same time. The difference is that Pattern P-RP constrains a resource to complete workitems in batch, while Pattern R-SE doesn't have such a strong constraint. Thus, the modelling results of Pattern R-SE (Simultaneous Execution), as a simple version of Pattern R-PE, is available in Table 6 .

Task Name	*Task Network Details*
sim_task	$[sim\_network, [(t_1, tl_1), (t_2, tl_2) \dots \dots (t_m, tl_m)], \mu]$ $\mu = \{(S_I \prec tl_i \prec S_S)\}$, where $i = 1,2 \cdots \cdots m$
REMARKS	
sim_task is a complex task involving many workitems. The μ doesn't put execution ordering in a strict manner. It puts every workitem t_i in a context that every task should be executed between states S_I and S_S.	

Table 6 the modelling result of Pattern R-SE (Simultaneous Execution).

Pattern-AR (Additional Resource) characterizes that one resource can request additional resources to assist in the process of a workitem. One possible solution is to divide the workitem into several sub workitems, and allocate these sub workitems to additional resources. Then, these additional resources can start to process sub workitems, individually. As all the sub workitems have been completed, then the original workitem is completed (van der Aalst and Kumar 2001). We model this pattern in Table 7 .

Task Name	*Task Network Details*
add_res_task	$[add\_network, [(div\_and\_dis, tl_d), (cpl, tl_c)], \mu]$ $\mu = \{(S_I \prec tl_d \prec S_S), (S_S \prec tl_c \prec S_C)\}$ where $i = 1,2 \cdots \cdots m$
div_and_dis_task	$[dd\_network, [(t_1, tl_1), (t_2, tl_2) \dots \dots (t_m, tl_m)], \mu]$ $\mu = \{(S_I \prec tl_i \prec S_S)\}$, where $i = 1,2 \cdots \cdots m$
REMARKS	
add_res_task is the task being interpreted by *add_network* containing complex task *div_and_dis_task* and primitive task *cpl*. The complex task *div_and_dis_task* can be used to decompose a workitem into a set of sub-workitems (task) t_i, and these sub-workitems should be completed before final *completion*, see the constrains $(tl_d \prec S_S \prec tl_c)$. The decomposition details about t_i are not shown in this modelling results, but can take the *basic_task_network* in Table 3 as reference.	

Table 7 the modelling result of Pattern R-AR (Additional Resource).

4 Resource Pattern Visualization in the Virtual World

A virtual world is a network-based, computer synthesized dynamic environment, where participants can observe and interact with computer-generated objects (Burdea and Coiffet 2003). The modelling results of the resource patterns previously detailed, basically involves two entities, state and transition. A state means a unique configuration of the system, indicating the static aspects of a workitem. A transition means a process where a system moves from one state to other, describing the dynamics aspects of the workitem. We believe that appropriate geometry and an animated avatar, as features of a virtual world, can be used to satisfied static and dynamic workitem aspect visualization.

Geometry in a virtual world can be shaped and decorated with different textures to represent different material. These representations are an integration of visual singles (structure and spectrum). According to cognitive theory, the working memory in human can distinguish the features of visual singles (Lohse 1997). In the context of our research, we can use these visual features to represent the different states of a workitem. For example, the green colour can suggest a workitem is in started state, while a red arrow can suggest a workitem is being handed over from one human resource to another.

Avatars, in general, are a representation of the self in a given environment, enabling its host to sense and react on events happened in the environment, and to change the given environment (Castronova 2003). In the context of a 3D virtual world, an avatar can be a humanoid 3D representation, driven by a virtual world participant (a human or an intelligent software agent). It can be used to replicate the behaviour of human resource in the workflow system. For example, the hand shaking of two avatars can be used to represent reallocation of a workitem, the keyboard tapping of an avatar can be used to represent a human resource is dealing with a workitem. Such an animated behaviour can intuitively suggest the transitions happened in the system (Tverskyand and Morrison 2002).

With the discussion above, if a resource pattern can be modelled and mapped into a virtual world appropriately, participants such as business analysts and stakeholders can observe resource patterns in an intuitive manner. We already demonstrated the modelling applicability of an HTN for resource patterns in Section 3, a mapping mechanism between modelled resource patterns and virtual world features, geometry and avatar, will be necessary for us to establish this visualization system. Therefore, we demonstrate the mapping between resource patterns and virtual world features in Table 8

Resource Pattern Element	Visual Representation	Description
State		A cube with different texture can be used to represent the state of the workitem. The texture with words and colourful background can be used to indicate the name and statue of the workitem, respectively. By attaching the texture on the cube, it enables people to observers the state information of the workitem from different angles, and different colour can make it distinguishable in different state.

Transition		The animation and postures of avatars are used to represent the dynamic aspect of the workitem. The avatar taking a blood indicates may indicate the blood transition is in progress.

Table 8 the visual representation of states and transitions in the resource patterns.

5 Detailed Medical Example

The previous section has established a mapping mechanism between resource patterns and a virtual world. Here, we use a medical example to illustrate our modelling and visualization approach in detail. Section 5.1 introduces the background of workflow applications in the medical domain, indicating a potential visualization needs in this field. Section 5.2 uses a fabricated scenario to demonstrate applicability of virtual world visualization.

5.1 Background

Treatment processes in the medical domain have been investigate by many workflow experts, and these experts (Mans, M.H.Schonenberg et al. 2008) recognized treatment processes as "careflows", which are a type of ad-hoc workflows. Workitems involved in such workflows require resources to be highly participative, interactive and collaborative, therefore it is evident that, numerous resource patterns occur in the lifecycle of one workitem in such scenarios. A resource modelling component would be useful to clarify the participation, interaction and collaboration mechanisms in these careflows (Richard and Manfred 2007). Animation has a strong ability to explain the dynamics aspect of a system (Tverskyand and Morrison 2002), therefore, a visualization component will be necessary to reduce the cognitive overhead in understanding underlying participation, interaction and collaboration mechanisms.

5.2 Resource Pattern Visualization Example

We adapt the Scenario 1 in Section 3.1 into a medical context for demonstrating modelling and visualization results. In an adapted scenario, four resources are involved in accomplishing a complex workitem, containing three primitive tasks. The example involves a creation pattern, pull pattern, push pattern, detour and an auto-start pattern, with the example itself as a multiple resource pattern representing the relationship "many workitems to many resources", see the details below:

> ***Scenario 2.*** *The trauma team lead R_1 is executing a workitem W_X, the "Medical Case Review". At that time, the workflow engine creates a workitem called "Surgery Preparation" W_0 for the resource R_1. Thus, resource R_1 reallocates workitem W_X to another R_X with current execution information of the workitem W_X. After the reallocation, resource R_1 accepts this workitem W_0 and starts to divide W_0 as three sub-workitems, "Retrieve Patient Information" workitem W_1, "Aesthetic Preparation" workitem W_2 and "Instrument Preparation" workitem W_3, which are going to allocate to herself, surgery assistants R_2 and R_3, respectively. R_2 should passively wait for the allocation, while R_3 can actively commit to the workitem. Unless resources R_2 and R_3 confirm the sub-workitems, they are entitled to execute these three workitems. The execution of sub-workitem W_1 should be started immediately after the accomplishment of W_2 and W_3. When W_1 is finished, the original workitem W_0 can be comcluded and checked back to the workflow engine."*

The scenario above implicitly contains several resource patterns (Russell, Hofstede et al. 2004). For example, the detour pattern (Pattern R-PR,) between Resource R_1 and R_X, as they are dealing with the workitem W_X. The pull pattern (Pattern R-SA, see) between Resource R_1 and R_3, as resource R_3 is actively requiring the commitment of sub-workitem W_3. The lifecycle of workitem W_0 can be modelled by an HTN. We show the final modelling results in Figure 4, where the relationships between the tasks are represented, but we omit the task network construction and execution processes. With the mapping mechanism we defined in Section 4, the visualization results can be shown in Figure 4.

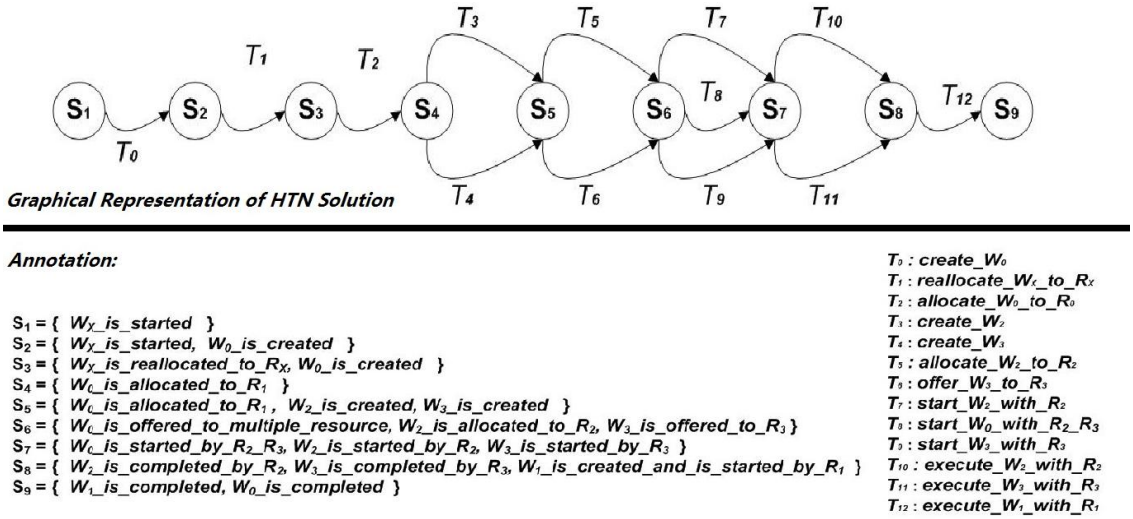
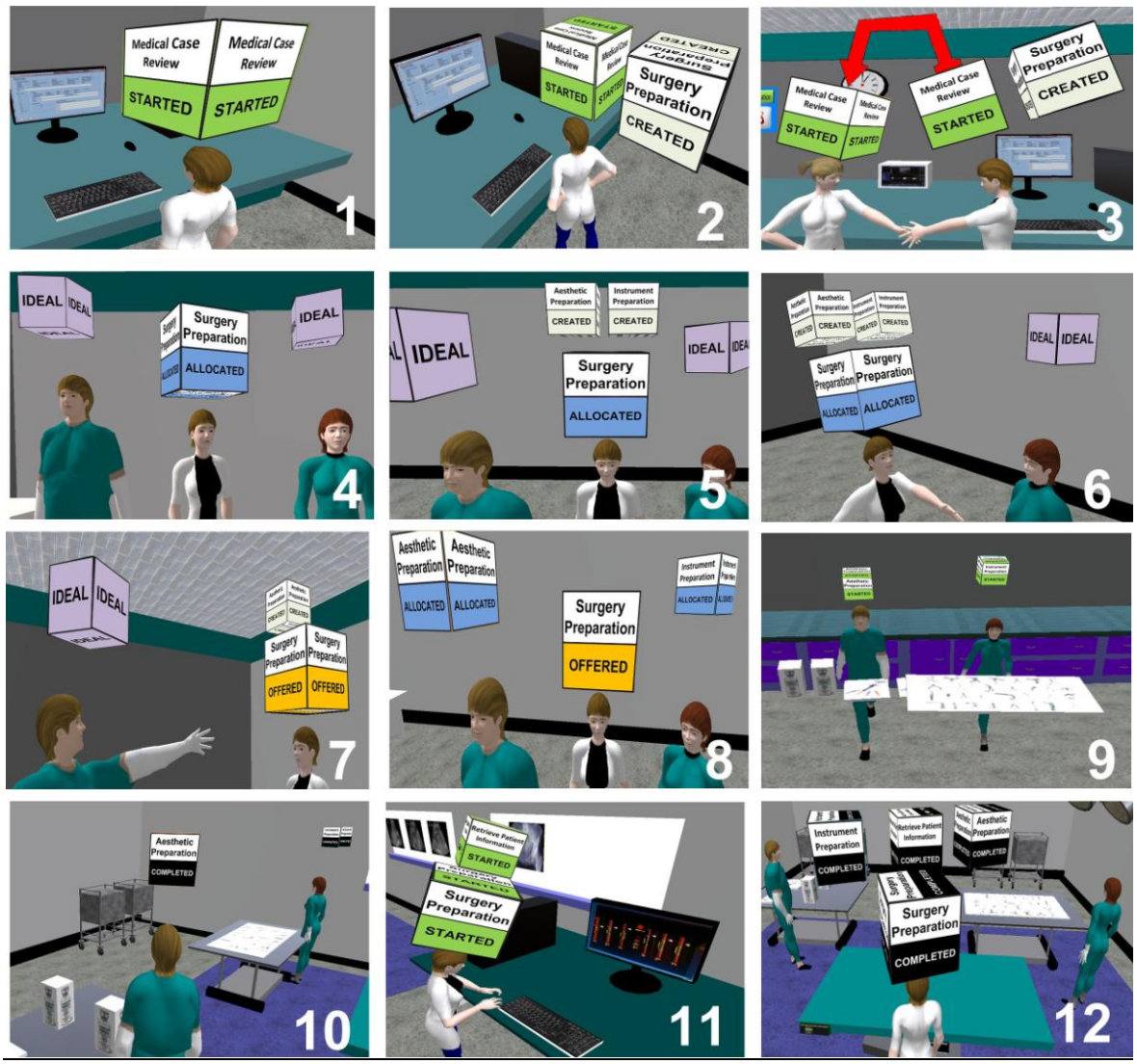


Figure 4 The HTN solution of surgery preparation, representing the lifecycle of the workitem "surgery". The circle is the representation of state, while the curve arrow is the representation of transition.



Picture ID	*HTN modelling result*	*Picture ID*	*HTN modelling result*	*Picture ID*	*HTN modelling result*
1	S_1	*2*	S_2	*3*	T_1
4	S_4	*5*	S_5	*6*	T_5
7	T_6	*8*	S_6	*9*	T_{10}, T_{11}
10	S_8	*11*	S_8 , T_{12}	*12*	S_9

Pattern Category	*Pattern Name*	*Picture ID*	*Remark*
Creation Pattern	Pattern R-DA	4,5	A resource is creating three sub-workitems, and going to allocate these sub-workitems.
Push Pattern	Pattern R-DBOS	6	A resource is trying to allocate a workitem to her subordinates.
Pull Pattern	Pattern R-SA	7	A resource is actively asking for workitem commitment.
Auto-Start Pattern	Pattern R-CC	10,11	As two resources completed their workitems in picture 10. Consequently, a resource can start workitem in picture 11.
Detour Pattern	Pattern R-PR	3	One resource is reallocating her workitem to another resource, the other resource can continue her work
Multiple Resource Pattern	Pattern R-AR	5-12	A resource needs two extra resources to assist her to accomplish surgery preparation.

Figure 5 The resource behaviour visualization in 9 pictures. These 9 pictures describe the responsibilities of different resources in the workitem surgery preparation (the black cube). Such a workitem is divided into three sub-workitems (the orange, green and blue cubes) that are allocated to three different resources. The combination of these 9 pictures reflects the relationship of many workitems to many resources, that is, the multiple resource pattern.

In Figure 5, we visualize six categories of resource patterns, except the Visibility Pattern. According to the statement in paper (Russell, van der Aalst et al. 2005), Visibility Pattern mainly deals with relationship among the availability and commitment of workitems and attributes of resources. This is a problem of authorization rather than state transition. Thus, we don't visualize patterns belonging to this category. Despite the fact the visualization case does not involve the visualized visibility patterns, we still can visualize them by modifying the property of cube hovering above heads of avatars. The cube is the indication of state of a workitem being processed by an avatar. The solid, semi-transparent and fully-transparent appearance of the cube can be mapped to indicate that the workitem in the different state.

6 Conclusion

Reviewing the state of the art of knowledge in the field of workflow, only a few researchers have started to explore the resource pattern modelling issue in the workflow domain. Few researchers have thought fully about how to utilize a virtual world to visualize the behaviour of human resources at an operational level.

With this in mind, we propose that an HTN can be used to model to model the resource patterns occurred in the lifecycle of workitem. The major advantage of such a mathematical tool is that it can represent all resource patterns in detail, as we demonstrated. We hope the modelling approach we discussed in this paper can inspire more research works in the multiple resource pattern field.

In addition, we discussed the visual mapping mechanism between the resource pattern and virtual world. The conceptual resource pattern can be turned into an intuitive animation. This will be useful for the naïve stakeholders who have less knowledge in conceptual modelling terminology, enabling them to more easily engage in resource model validation activities with business analysts.

Presently, our approach can translate resource pattern into an intuitive animation, however, subjective evaluation tests need to be performed to indicate its capacity as a visualization approach. To our best knowledge, less attention has been made to resource model visualisation as a research question, though some are investigating the perception and comprehension of 2D process models (Recker, Rosemann et al. 2009), no work has been performed in the validity of 3D process model representations.

7 References

Bhattacharya, K., C. E. Gerede, et al. (2007). Towards Formal Analysis of Artifact-Centric Business Process Models. Conference on Business Process Management.

Burdea, G. C. and P. Coiffet (2003). Virtual Reality, Wiley.

Burton-Jones, A., Y. Wand, et al. (2009). "Guidelines for Empirical Evaluations of Conceptual Modeling Grammars." Journal of the Association for Information Systems **10**(6): 495-532.

Castronova, E. (2003). Theory of the Avatar, Department of Telecommunications , Indiana University Bloomington.

Erol, K., J. Hendler, et al. (1994). Semantics for hierarchical task-network planning, University of Maryland at College Park**:** 28.

J. Parsons and L. Cole (2005). "What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques." Data & Knowledge Engineering **55**(3): 327-342.

Lohse, G. L. (1997). "The Role of Working Memory in Graphical Information Processing." Behaviour and Information Technology **16**(6): 297-308.

Lu, S., A. Bernstein, et al. (2006). "Automatic workflow verification and generation." Theoretical Computer Science **353**: 71-92.

Mans, R. S., M.H.Schonenberg, et al. (2008). Application of Process Mining in Healthcare - A Case Study in Dutch Hospital. Biomedical Engineering Systems and Technologies International Joint Conference**:** 425-438.

Moody, D. L. (2009). "The "Physics" of Notations: Toward Scientific Basis for Constructing Visual Notations in Software Engineering." IEEE Transctions On Software Engineering **35**(6): 756-779.

Moore, C. (2002). Common mistakes in workflow implementations, Cambridge, MA: Giga Information Group.

Nah, F., J. Lau, et al. (2001). "Critical factors for successful implementation of enterprise systems." Business Process Management Journal **7**(3): 286-296.

Nigam, A. and N. S. Caswell (2003). "Business artifacts: An approach to operational specification." IBM Systems Journal **42**(3): 428-445.

Pesic, M. and W. M. P. van der Aalst (2007). "Modelling Work Distribution Mechanisms Using Colored Petri Nets." INTERNATIONAL JOURNAL ON SOFTWARE TOOLS FOR TECHNOLOGY TRANSFER **9**(3-4): 327-352.

Rasmussen, R. and R. Brown (2012). "A deductive system for proving workflow models from operational procedures." Future Generation Computer Systems **28**(5): 732-742.

Recker, J., M. Rosemann, et al. (2009). "Business Process Modeling: A Comparative Analysis." Journal of the Association for Information Systems **10**(4): 333-363.

Richard, L. and R. Manfred (2007). "IT support for healthcare processes– premises, challenges, perspectives." Data & Knowledge Engineering **61**: 39-58.

Russell, N., A. H. M. t. Hofstede, et al. (2004). Workflow Resource Patterns. BETA Working Paper Series. Eindhoven, Eindhoven University of Technology. **WP 127**.

Russell, N. and W. M. P. van der Aalst (2008). Work Distribution and Resource Management in BPEL4People: Capabilities and Opportunities. Advanced Information Systems Engineering.

Russell, N., W. M. P. van der Aalst, et al. (2005). Workflow Resource Patterns: Identification, Representation and Tool Support Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05).

Sadiq, S., M. Indulska, et al. (2007). Major Issues in Business Process Management: A Vendor Perspective. 11th Pacific-Asia Conference on Information Systems:Managing Diversity in Digital Enterprises**:** 40-47.

Shannon, C. E. and W. Weaver (1963). The Mathematical Theory of Communication, University of Illinois Press.

Trkman, P. (2010). "The critical success factors of business process management." International Journal of Information Management **30**(2): 125-134.

Tverskyand, B. and J. B. Morrison (2002). "Animation: can it facilitate?" Human-Computer Studies **57**: 247-262.

V. Khatri, I. Vessey, et al. (2006). "Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge." information Systems Research **17**(1): 81-99.

van der Aalst, W. M. P., L. Aldred, et al. (2004). Design and implementation of the YAWL system. Proceedings of The 16th International Conference on Advanced Information Systems Engineering (CAiSE 04).

van der Aalst, W. M. P., K. M. v. Hee, et al. (1994). Modelling and Analyzing Workflow using a Petri-net based approach. Proceedings of Second Workshop on Computer-supported Cooperative Work, Petri nets related formalisms.

van der Aalst, W. M. P. and A. H. M. t. Hofstede (2005). "YAWL: Yet Another Workflow Language." Inofrmation Systems **30**(4): 245-275.

van der Aalst, W. M. P. and A. Kumar (2001). "A reference model for team-enabled workflow management systems." Data and Knowledge Engineering **38**(3): 335 - 363.

Wang, J. and A. Kumar (2005). A Framework for Document-Driven Workflow Systems. Business Process Management : 3rd International Conference.

zur Muehlen, M. (1999). Resource Modeling in Workflow Applications. Workflow Management Conference.

zur Muehlen, M. (2004). "Organizational Management in Workflow Applications – Issues and Perspectives." Information Technology and Management **5**: 271-291.