Towards a Reference Model for Work Distribution in Workflow Management Systems

M. Pesic and W.M.P. van der Aalst

Department of Technology Management, Eindhoven University of Technology,
P.O.Box 513, NL-5600 MB, Eindhoven, The Netherlands.
m.pesic@tm.tue.nl, w.m.p.v.d.aalst@tm.tue.nl

Abstract. Reference models such as the well-known SAP reference models tend to focus on the control-flow perspective. Although the languages typically used to capture reference models (e.g., EPCs) allow for the modeling of the resource or data perspectives, reference models tend to oversimplify these other perspectives. This paper focusses on the *resource perspective* in the context of workflow management systems. The aim is to develop a reference model for work distribution, i.e., how should the system distribute work based on the structure of the organization, capabilities/qualifications of people, and characteristics of the process. This paper reports on our first results based on a detailed analysis of contemporary workflow management systems (Staffware, FileNet, and FLOWer), supported by *Colored Petri Nets* (CPNs) to model work distribution mechanisms and *resource patterns* to identify key functionalities.

Key words: Work distribution, reference models, workflow management, business process management, resource patterns, colored Petri nets.

1 Introduction

Reference models are generic conceptual models that formalize recommended practices for a certain domain. Often labelled with the term "best practice" reference models claim to capture reusable state-of-the-art practices. Reference models typically focus on a specific application domain. For example, The Dutch NVVB (http://www.nvvb.nl/) offers a set of reference models for local governments modeled using the Petri-net-based tool Protos [31]. Other reference models are more general, moreover, the term reference model is also used for models describing the structure and functionality of business applications. One could argue that the SAP reference model actually describes the R/3 system rather than "best practices" in some domain. We will interpret reference models in the more system-oriented sense. However, instead of building a system-specific reference model, we would like to generalize over a range of systems, i.e., existing and future workflow management systems.

Workflow management systems are process-aware information systems [1, 12], which are used in companies as a means for the computerized structuring and driving of complex business processes. Workflow management systems implement business process models and use them for driving the flow of work by allocating the right employees to the right tasks at the right times. The system *manages the work of employees.* It will determine which tasks an employee has to execute and when, which documents will be used, which information will be available during work, etc. Typically, a workflow management system offers several mechanisms to distribute work. Nevertheless, we believe that existing systems are too limited in this respect. The goal of this paper is not to propose advanced work distribution mechanisms. Instead we focus on the analysis of functionality in existing systems. The goal is not to evaluate these systems, but to understand *how* they offer specific functionality. Since work distribution defines the quality of work, it is important to consider research from the field of social sciences, e.g., social-technical design [43, 13, 7, 10]. We believe that only by combining both *technical* and *social* approaches, one can truly grasp certain phenomena. A *deeper understanding* of particular aspects of work distribution is essential for developing a new breed of more user-centric systems.

The work reported in this paper can be seen as an extension of the *workflow patterns initiative* [2] (cf. www.workflowpatterns.com). Within the context of this initiative 43 resource patterns [39, 37] have been defined. Using a patterns approach, work distribution is evaluated from the perspective of the end-user as a dynamic property of workflow management systems. The work reported in this paper adds to a better understanding of these mechanisms by providing explicit process models for these patterns, i.e., the descriptive models are augmented with executable models. Note that most work reported in literature (cf. Section 5) uses static models to describe work distribution. Consider for example the meta modeling approaches presented in [3, 27–29, 36]. These approaches use static models (e.g., Unified Modeling Language class diagrams) to discuss work distribution concepts. This paper takes a truly dynamic model – a *Colored Petri Net* model – as a starting point, thus clearly differentiating our contribution from existing work reported in literature.

Colored Petri Nets (CPNs) [18, 23] are a natural extension of the classical Petri net [33]. There are several reasons for selecting CPNs as the language for modeling work distribution in the context of workflow management.First of all, CPNs have formal semantics and allow for different types of analysis, e.g., state-space analysis and invariants [19]. Second, CPNs are executable and allow for rapid prototyping, gaming, and simulation. Third, CPNs are graphical and their notation is similar to existing workflow languages. Finally, the CPN language is supported by CPN Tools[1] – a graphical environment to model, enact and analyze CPNs.

In this paper, we provide a basic CPN model that can be seen as the "greatest common denominator" of existing workflow management systems. The model will incorporate concepts of task, case, user, work item, role and group. This model should be seen as a *starting point* towards a more *comprehensive reference model for work distribution.* The basic CPN model is extended and specialized

[1] CPN Tools can be downloaded from wiki.daimi.au.dk/cpntools/.

for three specific systems: Staffware [42], FileNet [15], and FLOWer [30]. The latter three models are used to investigate differences and similarities as aid in a deeper understanding of work distribution mechanisms. In addition, advanced resource patterns that are not supported by these three systems are modeled by extending the basic CPN model.

The remainder of this paper is organized as follows. Section 2 discusses the various types of reference models and how our work can be positioned in a wider range of reference models. Section 3 presents the basic CPN model which should be considered as the "greatest common denominator" of existing workflow management systems. Section 4 extends this model in two directions:(1) Section 4.1 discusses the model in the context of three different systems (i.e., Staffware, FileNet, and FLOWer), and (2) Section 4.2 reflects on the basic model from the perspective of the so-called "resource patterns". An overview of related work is given in Section 5. Section 6 concludes the paper.

2 Reference Models

As indicated in the introduction, we can distinguish at least two types of reference models: (1) "best practice" reference models that aim at capturing domain-specific practices, and (2) "system oriented" reference models that aim at capturing the structure and functionality of a software system [9]. Although the focus of this paper is on the latter class of reference models, we first discuss characteristics of reference models in a broader context.

The main objective of reference models is to streamline the design of particular models by providing a generic solution [35]. The application of reference models is motivated by the "Design by Reuse" paradigm. Reference models accelerate the modeling process by providing a repository of potentially relevant models. These models are ideally "plug and play" but often require some custimization/configuration [6]. Reference models can be differentiated along the following main criteria [35]: scope of the model (e.g., functional areas covered), granularity of the model (e.g., number of levels of decomposition detail), views (e.g., process, data, objects, organization) that are depicted in the model, degree of integration between the views, purposes supported, user groups addressed, internal or external (commercial) use, availability of the model (e.g., paper, tool-based, Web-based), availability of further textual explanation of the model, explicit inclusion of alternative business scenarios, existence of guidelines on how to use these models, and availability of relevant quantitative benchmarking data. A further and more comprehensive differentiation based upon the domain that underlies the reference model can be found in [5, 9, 21, 34]. In this paper, we look at a reference model focusing on the resource perspective (i.e., the scope is work distribution and the view is the interaction between the process and the organization) at a finer level of granularity.

One of the most comprehensive models is the SAP reference model [9, 21]. Its data model includes more than 4000 entity types and the reference process models cover more than 1000 business processes and inter-organizational business scenarios [35]. Most of the other dominant ERP vendors have similar or alternative approaches towards reference models. Foundational conceptual work for the SAP reference model had been conducted in the years 1990-1992 [20]. The outcome of this project was the process modeling language Event-driven Process Chains (EPCs) [20, 22], which has been used for the design of the reference process models in SAP. EPCs also became the core modeling language in the Architecture of Integrated Information Systems (ARIS) [40, 41]. It is now one of the most popular reference modeling languages and has also been used for the design of many SAP-independent reference models (e.g., the ARIS-based reference model for Siebel CRM or industry models for banking, retail, insurance, telecommunication, etc.).

Reference models such as the SAP reference model provide for modelling various perspectives (e.g., the process perspective, the data perspective, etc.). However, existing languages for representing reference models (e.g., EPCs and UML activity diagrams) tend to oversimplify the resource/organizational perspective. When it comes to work distribution there are subtle but very important differences between mechanisms. A badly chosen work distribution mechanism may be very disruptive, and have dramatic effects on the performance of a business process. In the remainder, we will investigate the possibility of a comprehensive CPN-based reference model to overcome these problems.

3 Towards a Reference Model for Work Distribution

Different workflow management systems tend to use not only different work distribution concepts, but also completely different terminologies. This makes it difficult to compare these systems. Therefore, we will not start by developing CPN models for different systems and see how these can be unified, but, instead, start with modeling the "greatest common denominator" of existing systems. This model can assist in comparing systems and unifying concepts and terminology. We will use the term *Basic Model* to refer to this "greatest common denominator" and represent it in terms of a CPN model.

In the introduction we already motivated the use of CPNs as a modeling language [18, 23]. A CPN consists of *places* and *transitions* connected by *arcs*. The network structure is static but places can hold *tokens* thus representing the state of the model. The number of tokens per place can vary over time. Moreover, unlike the classical Petri net, tokens can have both a value and a timestamp. The timestamps indicate the availability of tokens and can be used to model delays, processing times, timeouts, etc. The value of a token indicates the properties of the object represented by this token. Places (represented by ovals) are typed, i.e., the tokens in a place have values of a particular type (or color in CPN jargon). These types are a subset of the data types in Standard ML such as the primitive types integer and string and compositional types such as tuple, list and record. Each place can hold tokens with values of a certain type. Transitions (represented by rectangles) may consume and produce tokens. Since tokens have values, *arc inscriptions* are needed to specify the input-output relations.

Besides the extension with token colors and timestamps, CPN models allow for hierarchy. Complex models may be decomposed into subpages, also referred to as subprocesses or modules, to obtain a layered hierarchical description. A more detailed discussion of the CPN concepts is beyond the scope of this paper. In the remainder, we assume that the reader is familiar with the CPN language and refer to [18, 23] for more details.

The Basic Model represents a workflow management system that enables the following concepts: The business *process* is defined as a set of *tasks*. Before the process can be executed, it has to be instantiated. One (executable) instance of a process is referred to as a *case*. Each case traverses the process. If a task is enabled for a specific case, a *work item*, i.e., a concrete piece of work, is created. There is a set of *users* that can execute work items. The users are embedded in the organizational structure on the basis of their *roles*, and the *groups* they belong to. A group is an organizational unit (e.g., 'sales', 'purchasing', 'production', etc.), while a role represents a capability of the user (e.g., 'manager', 'software developer', 'accountant', etc.). These concepts are mapped onto CPN types as shown in Table 1. As indicated, CPN uses Standard ML types (e.g., *string* and *int*) and type constructors such as *product* to create pairs and other complex constructs (e.g., *(1,"taskA")* represents a value of type *WI*).

During the distribution, work items change state, which determines the next actions the users and the distribution mechanism can perform. The Basic Model uses a simple model of the life cycle of work items as shown in Figure 1. After the *new* work item has arrived, it is assumed that it is also *enabled* in order to be taken into distribution (i.e., state *initiated*). The Basic Model assumes that a work item becomes enabled at the moment of creation (arrival). Next, the work item is *offered* to the user(s). Once a user *selects* the work item, it is *assigned* to him/her, and he/she can *start executing* it. After the execution, the work item is considered *completed*, and the user can continue working on the next work item. Note that this description covers only the general, rather simplified, behavior of workflow management systems (e.g., errors and aborts are not considered).

color Task = string;
color Case = int;
color User = string;
color WI = product Case * Task;
color Role = string;
color Group = string;

Table 1. Basic Workflow Concepts Represented in CPN.

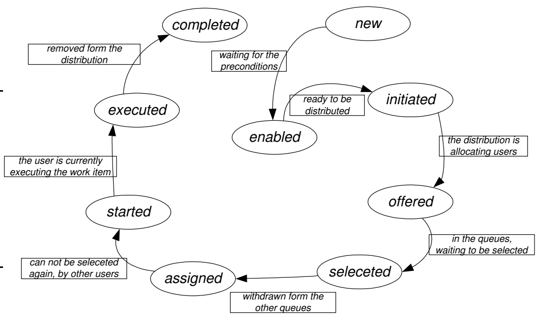
completed
new
removed form the distribution
waiting for the preconditions
ready to be distributed
initiated
executed
enabled
the distribution is allocating users
the user is currently executing the work item
offered
started
in the queues, waiting to be selected
can not be seleceted again, by other users
assigned
seleceted
withdrawn form the other queues

Fig. 1. Basic Model - Life Cycle of a Work Item

Before starting the model, it is necessary to provide the description of a concrete situation that is to be executed. This is done by defining the value of input elements as shown in Table 2.

Table 2. Input For The Basic Model

name	color	description
new work items	color WI = product Case * Task;	work items that have arrived and are ready to be distributed to users;
system users	color Users = list User;	a set of available users;
task maps	color TMap = product Task * Role * Group;	decision about which work items can be executed by which users is made based on the authorizations given in the process definition, for every task;
user maps	color UMap = product User * Roles * Groups;	the organizational structure is used to map users to the authorization of tasks;

As a model of an abstract workflow management system, the Basic Model is made on the basis of predefined assumptions: (1) we abstract from the process perspective (i.e., splits, joins, creation of work items), (2) we only consider the "normal" behavior (i.e., work items are completed successfully; errors and aborts are not included), and (3) we abstract from the user interface.

The model structure is organized into two sub-systems as shown in Figure 2. The CPN language allows for the decomposition of complex nets into subpages. These subpages are also referred to as sub-systems, sub-processes or modules. Using such modules we obtain a layered hierarchical description.

The two modules shown in Figure 2 communicate by exchanging messages via six places. The messages contain information about a user and a work item. Each of the six message places is of the type *color UWI = product User * WI*, i.e., each token represents a "user work item" – a combination of a work item and a user (cf. Table 3).

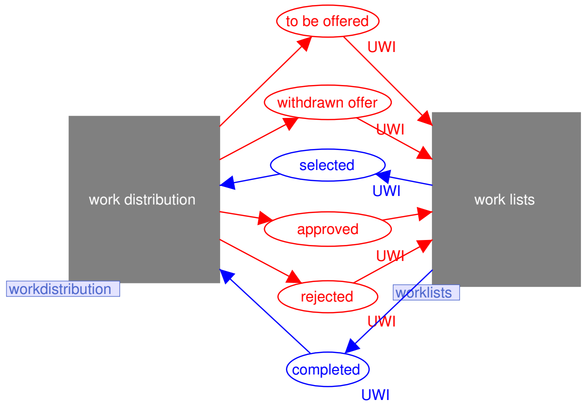

Fig. 2. Basic Model - Main

Table 3. Messages Between Modules (All of type *color UWI = product User * WI*)

Place	Message
to be offered	The work item is offered to the user.
withdrawn offer	Withdraw the offered work item from the user.
selected	The user requests to select the work item.
approved	Allow the user to select the work item.
rejected	Do not allow the user to select the work item.
completed	The user has completed executing the work item

Work Distribution. Figure 3 shows the Work Distribution module. This module manages the distribution of work items. It allocates users to which the work items should be offered, based on authorization (*TMap*) and organization (*UMap*) data. It should also manage the process of work execution, and make sure that work items are executed correctly. The variables used in this module are shown in Table 4.

The allocation function *offer* contains allocation rules of the specific distribution mechanism. Work items that are offered to users are stored in the place *offered work items*. After receiving a request from the user to select the work item, the decision is made whether to allow the user to select the item (and thus to execute it), or to reject this request. This decision is based on the assumption that at one moment, only one user can work on the work item. If the work item has already been selected (i.e., it is not in the place *offered work items*), then the request is rejected. Otherwise, the approval is sent to the user and the work item is moved to the place *assigned work items*, and, therefore, it cannot be selected again.

Table 4. Basic Model - Variables in Work Distribution Module

var tmaps: TMaps;
var umaps: UMaps;
var wi: WI;
var wis:WIs; (color WIs = list WI;)
var uwi: UWI;

Work Lists. Figure 4 shows the Work Lists module. This module receives messages from the Work Distribution module about which work items are to be offered to which users. The Work Lists module further manages events associated with the activities of users. It is decomposed into three units, which correspond to three basic actions users can make: *log on and off* (cf. Figure 5) in the system, *select work* (cf. Figure 6), *start work* (cf. Figure 7), and *stop work* (cf. Figure 8). Once the work item has been *offered* to users, they can *select* it. When a user selects the work item, the *request* is sent to the Work Distribution module. If the request is *rejected*, the action is *aborted.* If the Work Distribution Module *approves* the request, the user can start working on the work item. Once the user has started working, the work item is considered to be in *progress.* Next, the user can stop working, and the work item is *completed.* In order to perform any of these actions, it is necessary that the user is *logged on* in the system.

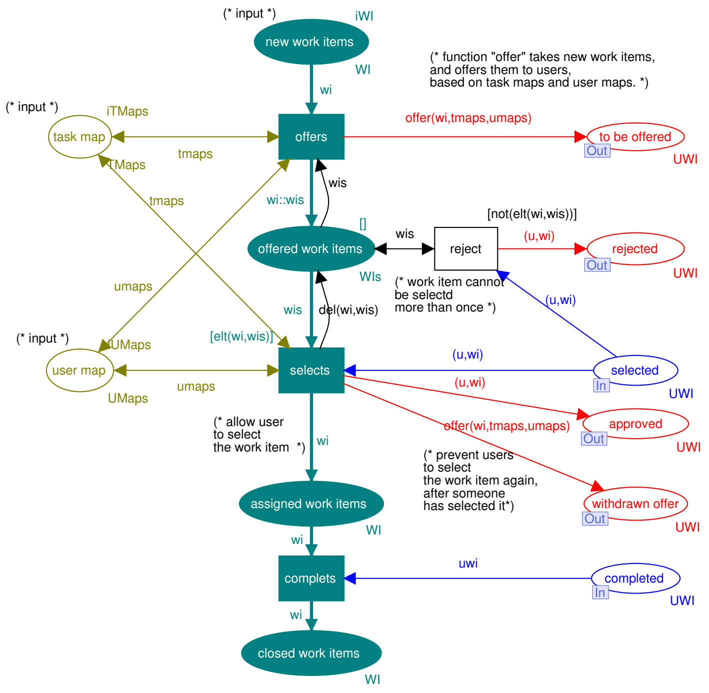

Fig. 3. Basic Model - Work Distribution

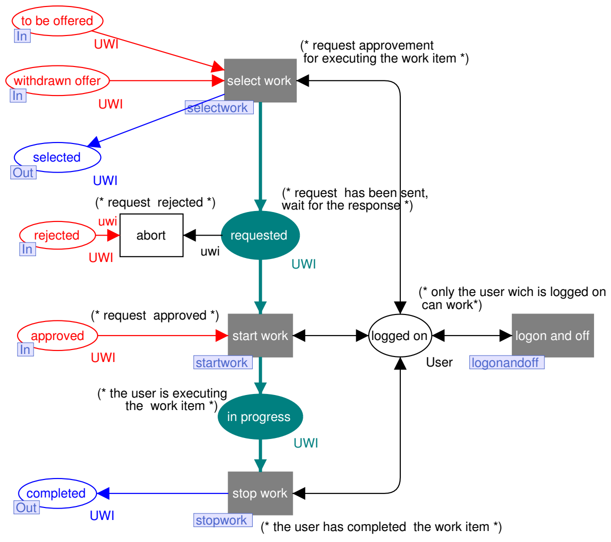

Fig. 4. Basic Model - Work Lists

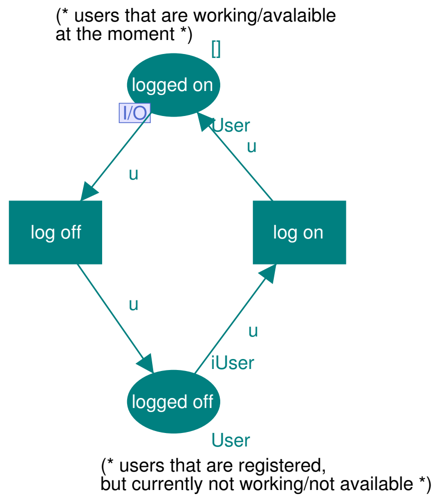
(* users that are working/avalaible at the moment *)
[]
logged on
I/O
User
u
u
log off
log on
u
u
iUser
logged off
User
(* users that are registered, but currently not working/not available *)

Fig. 5. Basic Model - Log On and Off

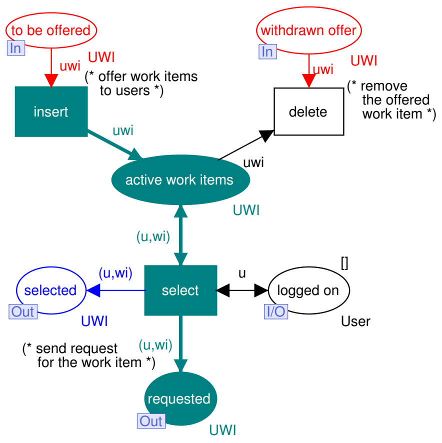

Fig. 6. Basic Model - Select Work

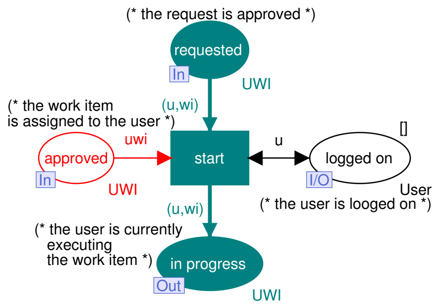
(* the request is approved *)
requested
In
UWI
(* the work item is assigned to the user *)
(u,wi)
uwi
approved
In
UWI
start
u
[]
logged on
I/O
User
(* the user is loged on *)
(u,wi)
(* the user is currently executing the work item *)
in progress
Out
UWI

Fig. 7. Basic Model - Start Work

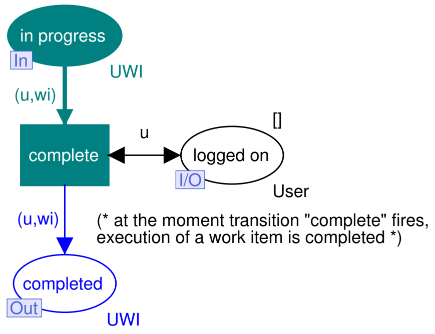
in progress
In
UWI
(u,wi)
complete
u
[]
logged on
I/O
User
(u,wi)
(* at the moment transition "complete" fires, execution of a work item is completed *)
completed
Out
UWI

Fig. 8. Basic - Stop Work

4 Evaluation of the Basic Model

The Basic Model presented in the previous section is used as a kind of reference for different extensions and specializations of work distribution. We have extended and specialized the Basic Model for three concrete systems (Staffware, FileNet and FLOWer) [32]. In this section, we evaluate the basic model by discussing differences between and commonalities among Staffware, FileNet, FLOWer and the Basic Model. Moreover, in Section 4.2, we discuss the relation between the resource patterns reported in [39, 37] and the Basic Model.

4.1 Three Workflow Management Systems: Staffware, FileNet and FLOWer

Staffware and FileNet are two "traditional" workflow management systems. FLOWer can be characterized as a case handling system [4] allowing for more flexibility. We have developed three dedicated CPN models for these three systems. We are not able to show these models here and need to refer to a technical report of this [32]. However, we are able to report on our experiences.

To model the functionality of Staffware, the concept of *work queues* is introduced in the CPN model. In Staffware there are personal queues and group queues. If the same work item is offered to *multiple* work queues, it is executed multiple times. Staffware also allows for *allocation at run-time* based on the attributes of a case. Moreover, Staffware also allows for *forward* and *suspend*, i.e., a user can put a work item on hold (suspend) or forward it to the another user.

FileNet allows for two ways of grouping people in organizational entities: *work queues* and *workflow groups*. Similar to Staffware, FileNet allows for personal queues and group queues. Workflow groups offer a completely different functionality: multiple people can work on the same work item and the group structure may change at run-time. Similar to Staffware, FileNet allows for *forward* and *suspend*.

FLOWer is quite different from Staffware and FileNet because it is data-driven and allows for all kinds of case-handling functionality [4]. This implies several extensions of the Basic Model. FLOWer allows users to *skip* or *redo* activities in addition to simply executing them. Unlike most systems, FLOWer *separates authorization* ("can do") from *distribution* ("should do").

Since we cannot show the full CPN models, we limit ourselves to some general conclusions. First of all, the Basic Model is a good basis for building system-specific models. The extensions are typically straightforward and consistent with the core structure of the Basic Model. Second, systems have surprisingly strange limitations, e.g., Staffware supports the role concept, but roles need to be associated to a single user (i.e., no two users can have the same role). Third, systems offer very different functionalities when it comes to work distribution. Finally, in each of the systems the basic concepts are presented and named differently, although a close observation often shows that these system-specific concepts are actually identical.

4.2 Resource Patterns

Instead of extending the Basic Model for more systems, we also looked at a more systematic way of work distribution. As indicated, similar concepts are often named and presented differently. Therefore, it is interesting to define these concepts in a system-independent manner. Therefore, we have used 43 documented *resource patterns* [39, 37]. These patterns are grouped into a number of categories: *creation patterns*, *push patterns*, *pull patterns*, *detour patterns*, *auto-start patterns*, *visibility patterns*, and *multiple resource patterns*. Each of these patterns can be modeled in terms of a CPN model.

Table 5 shows an overview of the patterns. It also shows whether a pattern is directly supported by the three systems and the Basic Model. We cannot elaborate on each of the patterns, but we will discuss one to illustrate our work. None of the systems supports *Pattern 17: R-SHQ (Shortest Queue)*. Pattern 17 is one of the push patterns, i.e., a pattern to push work to a specific user. For

Table 5. Support for Resource Patterns the Three Workflow Systems and the Basic Model (+ = direct support, – = no direct support, +/– = partial support, o = out-of-scope)

Nr	**Pattern**	**Staffware**	**FileNet**	**FLOWer**	**Basic Model**
1	R-DA (Direct Allocation)	+	+	+	+/–
2	R-RBA (Role-based Allocation)	+	+/–	+	+
3	R-FBA (Deferred Allocation)	+	+	–	–
4	R-RA (Authorization)	–	–	+	–
5	R-SOD (Separation of Duties)	–	–	+	–
6	R-CH (Case Handling)	–	–	+	–
7	R-RF (Retain Familiar)	–	–	+	–
8	R-CBA (Capability-based Allocation)	–	–	+	–
9	R-HBA (History-based Allocation)	–	–	–	–
10	R-OA (Organizational Allocation)	+/–	+/–	+/–	+/–
11	R-AE (Automatic Execution)	+	+	+	o
12	R-DBOS (Distribution by Offer – Single Resource)	–	–	–	–
13	R-DBOM (Distribution by Offer – Multiple Resources)	+	+	+	+
14	R-DBAS (Distribution by Allocation – Single Resource)	+	+	+	–
15	R-RMA (Random Allocation)	–	–	–	+
16	R-RRA (Round Robin Allocation)	–	–	–	–
17	R-SHQ (Shortest Queue)	–	–	–	–
18	R-ED (Early Distribution)	–	–	+	–
19	R-DE (Distribution on Enablement)	+	+	+	+
20	R-LD (Late Distribution)	–	–	–	–
21	R-RIA (Resource-Initiated Allocation)	–	–	+	+
22	R-RIEA (Resource-Initiated Execution – Allocated Work Item)	+	+	+	+
23	R-RIEO (Resource-Initiated Execution – Offered Work Item)	+	+	–	–
24	R-OBS (System-Determined Work List Management)	+	+	+	o
25	R-OBR (Resource-Determined Work List Management)	+	+	+	o
26	R-SA (Selection Autonomy)	+	+	+	+
27	R-D (Delegation)	+	+	–	–
28	R-E (Escalation)	+	+	–	–
29	R-SD (Deallocation)	–	–	–	–
30	R-PR (Stateful Reallocation)	+/–	+	–	–
31	R-UR (Stateless Reallocation)	–	–	–	–
32	R-SR (Suspension/Resumption)	+/–	+/–	–	–
33	R-SK (Skip)	–	–	+	o
34	R-REDO (Redo)	–	–	+	o
35	R-PRE (Pre-Do)	–	–	+	o
36	R-CC (Commencement on Creation)	–	–	–	–
37	R-CA (Commencement on Allocation)	–	–	–	–
38	R-PE (Piled Execution)	–	–	–	–
39	R-CE (Chained Execution)	–	–	+	–
40	R-CUWV (Configurable Unallocated Work Item Visibility)	–	–	–	o
41	R-CAWV (Configurable Allocated Work Item Visibility)	–	–	+	o
42	R-SE (Simultaneous Execution)	+	+	+/–	+
43	R-AR (Additional Resources)	–	–	–	–

this pattern, a new work item is pushed to the user with the shortest queue of all users that qualify. This implies that each user has a counter to count the number of pending work items. Based on this counter, the work is distributed. As figure 9 shows, the required changes to the Basic Model are minimal. A counter is introduced for each user (token in place *available*) and function *shortest_queue* is used to select one user from the set of possible users based on these counters. Similarly, most of the other patterns can be realized quite easily.

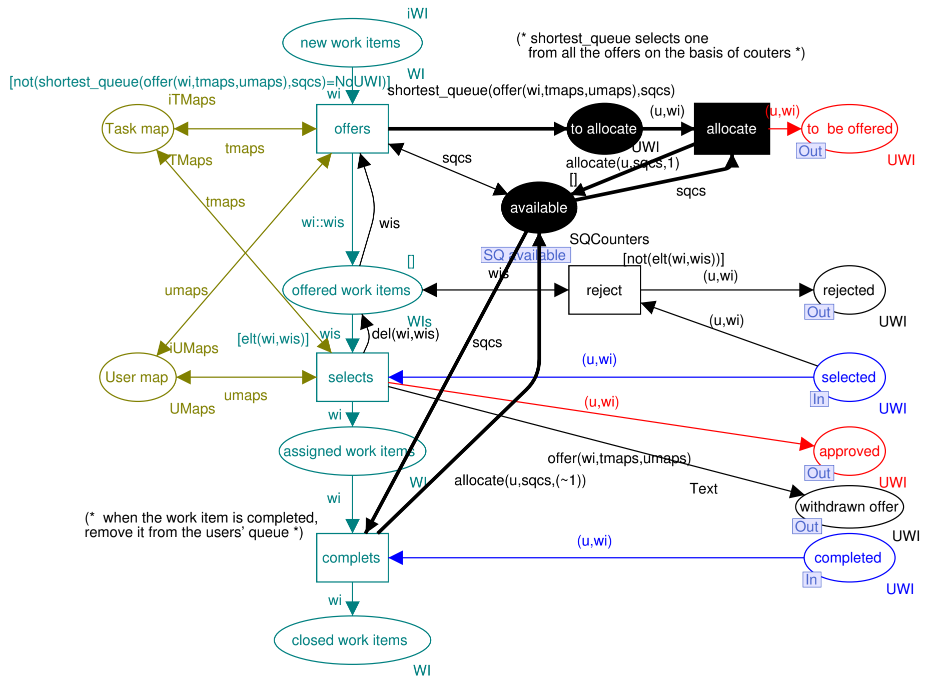

Fig. 9. Push Patterns - Shortest Queue

Table 5 shows that the Basic Model supports less patterns than any of the three systems. This makes sense since each of the system-specific models can be seen as an extension of the Basic Model. It is interesting to see that existing systems typically support less than half of the patterns directly. This reveals typical limitations of contemporary products. Some of the patterns are considered out-of-scope for the reference model we are aiming at (marked with "o"). These are typically patterns directly depending on control-flow functionality, while we prefer to focus exclusively on work distribution. Each of the patterns not marked with "o" can easily be added to the Basic Model separately. However, the patterns tend to interact. For example, what does "Shortest Queue" (Pattern 17) mean if multiple resources work on the same item (Pattern 43)? Therefore, we are still looking for a suitable CPN model that captures many patterns while still being intuitive and relatively simple, i.e., a more comprehensive reference model for work distribution. For this quest we want to use the results presented in this paper.

5 Related Work

Since the early nineties workflow technology has matured and several textbooks have been published, e.g., [1, 12, 17, 26, 28]. During this period many languages for modeling workflows have been proposed. These languages range from generic Petri-net-based languages to tailor-made domain-specific languages.

Despite the central role that resources play in workflow management systems, there is a surprisingly small body of research into resource and organizational modeling in a workflow context [24]. In their early work, Bussler and Jablonski [8] identified a number of shortcomings of workflow management systems when modeling organizational and policy issues. In subsequent work [17], they presented one of the first broad attempts to model the various perspectives of workflow management systems in an integrated manner including detailed consideration of the organizational/resource view.

One line of research into resource modeling and enactment in a workflow context has focused on the characterization of resource managers that can manage organizational resources and enforce resource policies. In [11], the design of a resource manager is presented for a workflow management system. It includes a high level resource model together with proposals for resource definition, query and policy languages. Similarly in [25], an abstract resource model is presented in the context of a workflow management system although the focus is more on the efficient management of resources in a workflow context than the specific ways in which work is allocated to them. In [16], a proposal is presented for handling resource policies in a workflow context. Three types of policy – qualification, requirement and substitution – are described together with a means for efficiently implementing them when allocating resources to activities.

Another area of investigation has been into ensuring that only suitable and authorized users are selected to execute a given work item. The RBAC (Role-Based Access Control) model [14] presents an approach for doing this. Whilst effective, RBAC models tend to focus on security considerations and neglect work distribution aspects.

Several researchers have developed meta-models, i.e., object models describing the relation between workflow concepts, which include work allocation aspects, cf. [3, 27–29, 36]. However, these meta-models tend to focus on the structural description of resource properties and typically do not describe the dynamical aspects of work distribution.

The work reported in this paper can be seen as an extension of the *workflow patterns initiative* (cf. www.workflowpatterns.com). Besides a variety of control-flow [2] and data [38] patterns, 43 resource patterns [39, 37] have been defined. This paper complements the work of resource patterns [39, 37] by providing executable models for work distribution mechanisms.

6 Conclusions

This paper is a first step towards a comprehensive reference model for work distribution in process-aware information systems (i.e., workflow management systems and beyond). To assist in understanding work distribution better, we used the CPN language and CPN Tools to model and analyze different mechanisms. To serve as a reference, we provided a basic model that can be seen as the "greatest common denominator" of existing workflow management systems. This model was extended and specialized for three specific systems (Staffware, FileNet, and FLOWer). The basic model already captures many of the so-called *resource patterns* defined earlier. However, we also modeled more advanced patterns by extending the basic model. In contrast to existing research mainly using static models (e.g., UML class diagrams), we focused on the dynamics of work distribution.

Our experiences revealed that it is relatively easy to model and analyze the systems and patterns using CPN Tools. This suggests that CPN language and the basic CPN model are a good basis for future research. We plan to extend the Basic Model into a more *comprehensive reference model for work distribution*. First, we want the model to be able to capture the typical functionality offered by existing systems. One can think of this as the "Least Common Multiple" of existing functionality. The corresponding CPN model will be much more complicated than the Basic Model used now. However, it can serve as a reference for organizations that want to implement more advanced functionality. The goal is to design and implement distribution mechanisms that overcome the limitations of existing systems. An important ingredient will be to use insights from socio-technical design [43, 13, 7, 10] as mentioned in the introduction.

References

1. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems.* MIT press, Cambridge, MA, 2002.
2. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.
3. W.M.P. van der Aalst and A. Kumar. Team-Enabled Workflow Management Systems. *Data and Knowledge Engineering*, 38(3):335–363, 2001.
4. W.M.P. van der Aalst, M. Weske, and D. Grünbauer. Case Handling: A New Paradigm for Business Process Support. *Data and Knowledge Engineering*, 53(2):129–162, 2005.
5. J. Becker, M. Kugeler, and M. Rosemann, editors. *Process Management: A Guide for the Design of Business Processes.* Springer-Verlag, Berlin, 2003.
6. P. Bernus. Generalised Enterprise Reference Architecture and Methodology, Version 1.6.3. IFIPIFAC Task Force on Architectures for Enterprise Integration, March 1999.
7. J. Bowers, G. Button, and W. Sharrock. Workflow From Within and Without: Technology and Cooperative Work on the Print Industry Shopfloor. In *The Fourth European Conference on Computer-Supported Cooperative Work (ECSCW 95)*, pages 51–66, Stockholm, September 1995. Kluwer Academic Publishers, Dordrecht, The Netherlands.

8. C. Bussler and S. Jablonski. Policy Resolution for Workflow Management Systems. In *Proceedings of the 28th Hawaii International Conference on System Sciences*, page 831. IEEE Computer Society, 1995.
9. T. Curran and G. Keller. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*. Upper Saddle River, 1997.
10. L. U. de Sitter, J. F. den Hertog, and B. Dankbaar. From complex organiations with simple jobs to simple organizations wiht complex jobs. *Human Relations*, 510(5):497–534, 1997.
11. W. Du and M.C. Shan. Enterprise Workflow Resource Management. In *Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE'99)*, pages 108–115, Sydney, Australia, 1999. IEEE Computer Society Press.
12. M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. *Process-Aware Information Systems*. Wiley & Sons, 2005.
13. F.M. van Eijnatten and A.H. van der Zwaan. The Dutch IOR approach to organisation design. An alternative to business process re-engineering? *Human Relations*, 51(3):289–318, 1998.
14. D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli. Proposed NIST Standard for Role-Based Access Control. *ACM Transactions on Information and System Security*, 4(3):224–274, 2001.
15. FileNET. *FileNet Business Process Manager 3.0*. FileNET Corporation, Costa Mesa, CA, USA, June 2004.
16. Y.N. Huang and M.C. Shan. Policies in a Resource Manager of Workflow Systems: Modeling, Enforcement and Management. Technical Report HP Tech. Report, HPL-98-156, Palo Alto, CA, USA, 1999. Accessed at http://www.hpl.hp.com/techreports/98/HPL-98-156.pdf on 20 March 2005.
17. S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation*. International Thomson Computer Press, London, UK, 1996.
18. K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1*. EATCS monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1997.
19. K. Jensen and G. Rozenberg, editors. *High-level Petri Nets: Theory and Application*. Springer-Verlag, Berlin, 1991.
20. G. Keller, M. Nüttgens, and A.W. Scheer. Semantische Processmodellierung auf der Grundlage Ereignisgesteuerter Processketten (EPK). Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89 (in German), University of Saarland, Saarbrücken, 1992.
21. G. Keller and T. Teufel. *SAP R/3 Process Oriented Implementation*. Addison-Wesley, Reading MA, 1998.
22. E. Kindler. On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. In J. Desel, B. Pernici, and M. Weske, editors, *International Conference on Business Process Management (BPM 2004)*, volume 3080 of *Lecture Notes in Computer Science*, pages 82–97. Springer-Verlag, Berlin, 2004.
23. L.M. Kristensen, S. Christensen, and K. Jensen. The Practitioner's Guide to Coloured Petri Nets. *International Journal on Software Tools for Technology Transfer*, 2(2):98–132, 1998.
24. A. Kumar, W.M.P. van der Aalst, and H.M.W. Verbeek. Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance? *Journal of Management Information Systems*, 18(3):157–193, 2002.

25. B.S. Lerner, A.G. Ninan, L.J. Osterweil, and R.M. Podorozhny. Modeling and Managing Resource Utilization in Process, Workflow, and Activity Coordination. Technical Report UM-CS-2000-058, Department of Computer Science, University of Massachusetts, August 2000. Accessed at http://laser.cs.umass.edu/publications/?category=PROC on 20 March 2005.
26. F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques*. Prentice-Hall PTR, Upper Saddle River, New Jersey, USA, 1999.
27. M. Zur Muehlen. Evaluation of Workflow management Systems Using Meta Models. In *Proceedings of the 32nd Hawaii International Conference on System Sciences - HICSS'99*, pages 1–11, 1999.
28. M. Zur Muehlen. *Workflow-based Process Controlling: Foundation, Design and Application of workflow-driven Process Information Systems*. Logos, Berlin, 2004.
29. M. zur Muhlen. Organizational Management in Workflow Applications Issues and Perspectives. *Information Technology and Management*, 5(3–4):271–291, July-October 2004.
30. Pallas Athena. *Flower User Manual*. Pallas Athena BV, Apeldoorn, The Netherlands, 2002.
31. Pallas Athena. *Protos User Manual*. Pallas Athena BV, Plasmolen, The Netherlands, 2004.
32. M. Pesic and W.M.P. van der Aalst. Modeling Work Distribution Mechanisms using Colored Petri Nets. BETA Working Paper Series, Eindhoven University of Technology, Eindhoven, 2005.
33. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.
34. M. Rosemann. Application Reference Models and Building Blocks for Management and Control (ERP Systems). In P. Bernus, L. Nemes, and G. Schmidt, editors, *Handbook on Enterprise Architecture*, pages 596–616. Springer-Verlag, Berlin, 2003.
35. M. Rosemann and W.M.P. van der Aalst. A Configurable Reference Modelling Language. *Information Systems*, pages (accepted for publication, can be obtained via BPMcenter.org), 2005.
36. M. Rosemann and M. Zur Muehlen. Evaluation of Workflow Management Systems - a Meta Model Approach. *Australian Journal of Information Systems*, 6(1):103–116, 1998.
37. N. Russell, W.M.P.van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. In O. Pastor and J. Falcao e Cunha, editors, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)* , volume 3520 of *Lecture Notes in Computer Science*, pages 216–232. Springer-Verlag, Berlin, 2005.
38. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Data Patterns. QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane, 2004.
39. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Resource Patterns. BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven, 2004.
40. A.W. Scheer. *Business Process Engineering, Reference Models for Industrial Enterprises*. Springer-Verlag, Berlin, 1994.
41. A.W. Scheer. *ARIS: Business Process Modelling*. Springer-Verlag, Berlin, 2000.
42. Staffware. *Using the Staffware Process Client*. Staffware, plc, Berkshire, United Kingdom, May 2002.
43. F. M. van Eijnatten. *The Paradigm that Changed the Work Place*. Van Gorcum, Assen, The Netherlands, 1993.