# ANALYZING OBSERVABLE BEHAVIOURS OF DEVICE ECOLOGY WORKFLOWS

Seng W. Loke and Sea Ling
*School of Computer Science and Software Engineering*
*Monash University, Caulfield East, VIC 3145, Australia*

Keywords: Device ecology, workflow, Petri net modelling, analysis.

Abstract: We envision an Internet computational platform of the 21st century that will include device ecologies consisting of collections of devices interacting synergistically with one another, with users, and with Internet resources. We consider *device ecology workflows* as a type of workflow describing how such devices work together. It would be ideal if one can model the devices in a computer and analyze the effects when such workflows are executed in the device ecology. This paper provides a Petri Net model in terms of workflow nets for analyzing the observable effects of device ecology workflows.

## 1 INTRODUCTION

A new era of Internet Computing is emerging where not just every person has the potential of being networked with every other person but even devices and things might be networked. New devices and things are emerging that will be capable of interacting meaningfully, exhibiting smart behaviour. Such smart devices will need to work effectively with each other and with Web services. The devices might interact across the living room or across nations in order to fulfill the goals of users. The American Heritage Dictionary defines the word "ecology" as "the relationship between organisms and their environment." We envision a computing platform of the 21st century that takes the form of device ecologies consisting of collections of devices (in the environment and on users) interacting synergistically with one another, with users, and with Internet resources, undergirded by appropriate software and communication infrastructures that range from Internet-scale to very short range wireless networks. These devices will perform tasks and work together perhaps autonomously but will need to interact with the user from time to time.

There has been significant work in building the networking and integrative infrastructure for such devices, within the home, the office, and other environments and linking them to the global Internet. For example, AutoHan (Saif et al., 2001), UPnP (Corporation, 2002), OSGI (Marples and Kriens, 2001), Jini (Waldo, 1999), and SIDRAH (with short range networking and failure detection) (Durand et al., 2003) define infrastructure and mechanisms at different levels (from networking to services) for devices to be inter-connected, find each other, and utilize each other's capabilities. Embedded Web Servers (Bentham, 2002) are able to expose the functionality of devices as Web services. Embedding micro-servers into physical objects is considered in (Nakajima, 2003). Approaches to modelling and programming such devices for the home have been investigated, where devices have been modelled as software components (Jahnke et al., 2002), as collections of objects (AHAM, 2002), as Web services (Matsuura et al., 2003), and as agents (Ramparany et al., 2003; Carabelea and O.Boissier, 2003). However, there has been little work on specifying at a high level of abstraction (and representing this specification explicitly) how such devices would work together at the user-task or application level, and how such work can be managed.

Our earlier work (Loke, 2003) explored the workflow (which we call *device ecology workflows*) as a metaphor for thinking about how collections of these devices (or devices in a device ecology) can work together to accomplish a purpose. It would be ideal if one can model (and simulate aspects of) the devices in a computer and analyze the effects when workflows are executed in the device ecology. Such analysis is useful in order to predict the effects of workflows be-


W. Loke S. and Ling S. (2004).
ANALYZING OBSERVABLE BEHAVIOURS OF DEVICE ECOLOGY WORKFLOWS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 78-83
DOI: 10.5220/0002625800780083

fore they are actually executed (e.g., to see if too much resources will be used or undesirable effects will occur) which will then feedback into the design of the workflow. In order to predict the behaviours of these workflows, we might require detailed knowledge of the device. Realistically, we can only assume some knowledge about the device such as some of their observable properties. Our more recent work (Loke, 2004) provided a formalization of what it means to say that the behaviour of devices and device ecology workflows are predictable: the behaviour of a device ecology workflow is predictable if the behaviour of the individual devices are predictable, and the more predictable each device is, the more predictable the device ecology workflow will be - our formalization tells us precisely how much more predictable the workflow will be. However, we have not considered in detail how properties of these workflows can be represented and analyzed.

In this paper, we investigate an alternative modelling using Petri nets for the observable states of devices and device ecology workflows. We show how existing Petri net theory can be used to analyze the behaviour of the devices and their workflows.

The next section explains the notion of workflows in device ecologies. Then, section 3 explains a Petri net model of devices and device ecology workflows. In section 4, we identify useful properties of device ecology workflows and then consider how Petri net analysis techniques can be applied to analyse these properties. We conclude in section 5 with future work.

## 2 WORKFLOWS IN DEVICE ECOLOGY

The environment of a device is not only other devices but also human users and the Web resources that it can connect to. We consider several examples of devices working together and interacting with Web resources (e.g., Web services).

Mentioned in the UPnP whitepaper is a script that, on detecting that a master switch has changed to "on", will turn the heater on to a preset temperature, start the answering machine playing new messages, turn the stereo system on set to a favourite station, raise the window blinds, turn the TV to the news station, and turn on the light in the foyer. In this case, the devices effectively work together, as orchestrated by some central coordinator, in order to create the right atmosphere and provide the right services (of informing the user of the new phone messages and news), even though the individual devices might not be aware of this global goal.

Devices might interact directly with one another and with the user. An example is quoted from Berger [1] concerning Thalia appliances, short for Thinking and Linking Intelligent Appliances: "When you set your Thalia alarm clock to wake you at a certain time, it will notify the coffee maker to adjust the time for your morning cup of java. The alarm clock will let you know if you forgot to put the water in the coffee maker. It will also tell the blanket with a brain when to turn off. In the morning, the alarm clock will greet you with the current news and weather. As you are making the pancakes, the kitchen console will automatically adjust the recipe for the number of portions you need. Your HomeHelper kitchen console will store shopping lists, calendars, and telephone numbers that can be downloaded to your HandHelper PDA."

Smart appliances might interact with applications and services across the Internet. For example, the fridge can order food that has run out by connecting to a Web service or negotiate with other appliances about resource (e.g., power and networking) consumption. Devices might seek human approval for more critical tasks.

Devices can work together with other devices, the user or Web resources in accomplishing its goals, either as initiated by users or by proactive smart devices. Such workflows might or might not involve devices, users and Web resources, depending on the application semantics. The simpler workflows might involve only one device or two devices, but larger workflows can involve a larger number of devices, as we saw in the examples above.

As an illustration, we consider a workflow device ecology workflow involving a television, a coffee-boiler, bedroom lights, bathroom lights, and a news Web service accessed over the Internet. Figure 1 describes this workflow. The dashed arrows represent sequencing, the boxes are tasks, the solid arrow represents a control link for synchronization across concurrent activities, and free grouping of sequences (i.e., the boxes grouped into the large box) represents concurrent sequences.

This workflow is initiated by a wake-up notice from Janes alarm clock which we assume here is issued to the Device Ecology Workflow Engine (which we call the Decoflow Engine) when the alarm clock rings. This notice initiates the entire workflow. Subsequent to receiving this notice, five activities are concurrently started: retrieve news from the Internet and display is on the television, switch on the television, boil coffee, switch on the bedroom lights, and switch on the bathroom lights. Note the synchronization arrow from Switch On TV to Display News on TV, which en-

[1] http://www.aarp.org/computers-features/Articles/a2002-07-10-computers_features_appliances.html

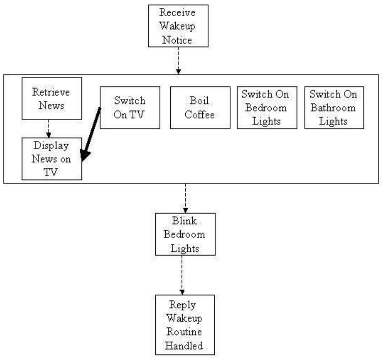


Figure 1: An Example Device Ecology Workflow

sures that the television must be switched on before the news can be displayed on it. After all the concurrent activities have completed, the final task is to blink the bedroom lights, in order to indicate to Jane that the workflow tasks have completed. This scenario was inspired by and extends that by Berger. This workflow can be described using BPEL4WS and executed using a corresponding workflow engine as outlined in (Loke, 2003).

# 3 PETRI NET MODELLING OF DEVICES'S OBSERVABLE STATES AND DEVICE ECOLOGY WORKFLOWS

Device ecology workflows can be decentralized (ad hoc and peer-to-peer between devices), coordinated by a central engine or hybrid, initiated by a user or a device. Regardless of the architecture or the language used to express the workflow, we have, in essence, a workflow and a device ecology in which the workflow is executed. Before describing the Petri net model of devices and device ecology workflows, we first provide some background on Petri nets.

## 3.1 Preliminaries

**Petri Nets.** Petri nets (Murata, 1989) have been widely used for process specification and verification. A standard Petri net graph consists of *places* (represented by circles), *transitions* (represented by rectangles) and *arcs* (represented by arrows). Given a set of identifiers $U$, a *Petri net structure* or simply *net* $N$ is a triple $(P, T, F)$, where $P \subseteq U$ and $T \subseteq U$ are nonempty, finite disjoint sets of places and transitions respectively, and $F \subseteq (P \times T) \cup (T \times P)$, is the set of arcs (flow relation). The components of a net $N$ are also denoted by $P_N$, $T_N$ and $F_N$. A *path* of a net is a nonempty sequence $x_1x_2 \,..\, x_k (k \in \mathbb{N})$ of net elements which satisfies $(x_1, x_2), .., (x_{k-1}, x_k) \in F$. It is *strongly connected* iff for every two net elements $x, y$ there is a path leading from $x$ to $y$. For some $x \in P \cup T$, the set ${}^\bullet x = \{y \mid (y, x) \in F\}$ is the *preset* of $x$ and the set $x^\bullet = \{y \mid (x, y) \in F\}$ is the *postset* of $x$.

We can view places as describing the possible local system states or conditions, transitions as events which may modify the system state and arcs simply link a place to a transition or a transition to a place. In other words, an arc linking a place to a transition indicates from which local state can the event occur next, and an arc linking a transition to a place indicates the local state transformation induced by the event occurrence, if any.

At any time, a place contains zero or more *tokens*, drawn as black dots. The state $M$, referred to as *marking*, is the distribution of tokens over places. It is a mapping $M : P \rightarrow \mathbb{N}$. It will be represented as follows: for example, $2p_1 + p_2 + 3p_3$ denotes the state with 2 tokens in place $p_1$, 1 token in place $p_2$ and 3 tokens in $p_3$. If a place (condition) is marked with a token, the condition is true. A *Petri net system* is a pair $(N, M_0)$ where $N$ is a net structure and $M_0$ is a marking called the initial marking (initial state) of the system.

The dynamic behaviour of a Petri net is controlled by the firing rule. A transition can *fire* or an event can occur if there is at least one token in each of the transition's input places (i.e. the event's pre-conditions are true). This transition is then said to be *enabled*. An enabled transition fires by removing one token from all of its input places (pre-conditions) and depositing one token in each of its output places (post-conditions). This movement of tokens from place(s) to place(s) indicates a change of system states after the occurrence of the event. The notation $M_1 \xrightarrow{t} M_2$ denotes a transition $t$ enabled in state $M_1$ and firing $t$ in $M_1$ results in a new state $M_2$. If $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} M_n$ are transition occurrences, then $\sigma = t_1t_2t_3 \cdots t_n$ is an occurrence sequence leading from $M$ to $M_n$ and we write $M \xrightarrow{\sigma} M_n$. $M_n$ is *reachable* from $M$ (we write $M \rightarrow^* M_n$) iff there is some firing sequence $\sigma$ such that $M \xrightarrow{\sigma} M_n$. $[M\rangle$ represents the set of all reachable markings from $M$.

Some useful system properties have also been defined for Petri nets. A Petri net system is *live* if, for every marking $M$ and every transition $t$, there is a marking $M' \in [M\rangle$ which enables $t$. The system is *bounded* if for every place $p$, there is a natural number $n$ such that $M(p) \leq n$ for every reachable marking $M$. The system is *1-bounded* or *safe* if $n = 1$. Petri net analysis methods (Murata, 1989; Desel and Esparza, 1995) have been devised to check for these properties in system modelling.

**Workflow Nets** A workflow (or business process) is a set of coordinated tasks to fulfill a specific business purpose (Workflow Management Coalition, 1999). Figure 1 is an example of a workflow. In a special class of Petri nets, called Workflow nets (WF-nets) (Aalst, 1998), workflow concepts are modelled by Petri net elements: workflow *activities or tasks* are modelled by the transitions, *conditions* (precedence relations) modelled by the places, flow directions modelled by flow relations (directed arcs) and *cases* modelled by tokens.

A Petri net $N = (P,T,F)$ is a *Workflow net (WF-net)* if and only if $N$ has two special places $i$ called *source* and $o$ called *sink*, where $^\bullet i = \varnothing$ and $o^\bullet = \varnothing$; and by adding a transition $t^*$ to $N$, the short-circuited net $(P, T \cup \{t^*\}, F \cup \{(o,t^*),(t^*,i)\})$ must be strongly connected (Aalst, 1998).

Using the WF-net model, the behavioural correctness of the workflow can be analysed by examining its *soundness* (Aalst, 1998), which is defined based on the following expected behaviour: that a workflow should always be able to complete a case; every case should be completed properly, with no more work in progress after completion; and every task should be executed by the workflow execution of some case. In essence, a case starting in the initial state $i$ must be able to reach the only final state $o$ in the model. The soundness property can then be expressed in terms of Petri net's liveness and boundedness properties: that the workflow is sound if and only if the short-circuited net is live and bounded. Hence, existing Petri net analysis methods can be employed to check for workflow soundness (Aalst, 1998). In (Aalst, 1999), the concepts of WF-nets have been extended to interorganizational workflows, in which several business partners are involved in shared workflow processes.

## 3.2 Modelling Devices: Describing Observable Effects of Device Ecology Workflows

The observable states of each device and how a device transitions between such states in response to operations can be represented by a Petri net. For example, figure 2 depicts the *minimal* behaviour of the devices participating in the ecology workflow in Figure 1. The TV, bath room light and the kettle can exist in the on or off state while the bedroom light has an additional blinking state. Note that the behavioural models are effectively finite state machines.

Recalling the definition of state machine S-net (Desel and Esparza, 1995), an S-net is a Petri net $(P,T,F)$ such that each transition $t \in T$ has exactly one input and output S-element, i.e., $|\, ^\bullet t \,|=|\, t^\bullet \,|= 1$. It is also a well-known fact that a Petri net system, called S-system $(N, M_0)$ is live if and only if $N$ is a strongly connected S-net and $M_0$ marks at least one token. In our approach, we assume that every device behaves like a strongly connected S-system marked by exactly one token, (i.e., at any time, the device can only exists in exactly one single (finite) state) and it is assumed to be correct at all times (i.e., the S-system is live).

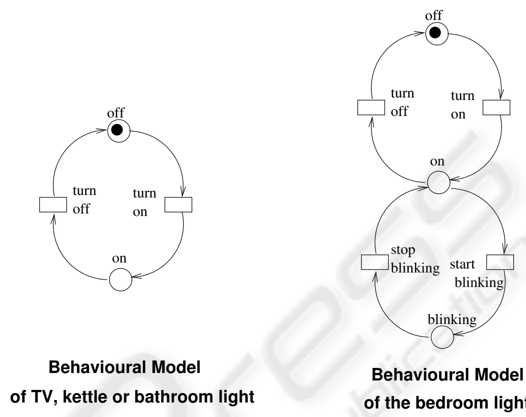


Figure 2: S-systems for Devices

Each device will also have an internal state - we are only concern with the observable properties of the devices here. Moreover, we assume that each device has a set of operations which can be performed on it, each of which might update the device's observable state (and internal state). The choice of properties modelled of the device might depend on the application at hand and so, it is meant to be an abstraction of the device.

## 3.3 Describing Device Ecology Workflows

Any device ecology workflow can be transformed into a WF-net. The dashed box in Figure 3 shows a WF-net equivalent of the device ecology workflow example of Figure 1. The WF-net is synchronizing with the observable behavioural (finite state machine) models of the devices to form a Device Ecology Behaviour Model (DEB-model), defined below.

**Definition 3.1**
*A* **Device Ecology Behaviour Model** *is a tuple* $DEB = (WF_1, WF_2, \cdots, WF_n, D_1, D_2, \cdots, D_k, T_{SC}, SC)$, *such that:*

1. $n, k \in \mathbb{N}$, *where* $n$ *is the number of device ecology workflow nets and k is the number of S-systems each representing a device;*
2. *for each* $i \in \{1, \cdots, n\}$ : $WF_i$ *is a WF-net with a start place* $in_{WF_i}$ *and an end place* $out_{WF_i}$;

3. *for each $i \in \{1, \cdots, k\}$ : $D_i$ is an S-system of a device;*
4. *$T_{SC}$ is the set of synchronous communication elements (fusion sets);*
5. *$SC \subseteq T_{SC} \times \mathbb{P}(T^{\Box}) \times \mathbb{P}(T^{\diamond})$ is the synchronous communication relationship, where $T^{\Box} = \bigcup_{i \in \{1, \cdots, n\}} T_{WF_i}$, and $T^{\diamond} = \bigcup_{i \in \{1, \cdots, k\}} T_{D_i}$;*
6. *for all $t \in T_{SC}$, $\{(t', x, y) \in SC \mid t' = t\}$ is a singleton;*
7. *for all $(t_1, x_1, y_1), (t_2, x_2, y_2) \in SC$: if $t_1 \neq t_2$, then $x_1 \neq x_2 \wedge y_1 \neq y_2$.*

The above defines a system which combines device ecology workflows (WF-nets) with device behaviours (finite state machines). Figure 3 depicts an example with only one WF-net synchronising with a number of devices. Requirement 1 states that potentially a device user can issue more than one device ecology workflows (WF-nets) to the ecology of devices for the system to do work.

A transition $t$ in the set $T_{SC}$ is the result of synchronising or "fusing" two transitions - one from the WF-net and one from the device - called a synchronous communication element. The relationship is defined by *SC* which is a set of triples, each consisting of the synchronous communication element and a pair of "fused" transitions. In Figure 3, the relationship is denoted by the dotted line labelled *SC*.

Requirements 6 and 7 state that for every synchronous communication element, there is only one unique element in *SC* and the pair of fused transitions are also unique. In other words, no already fused transition can be involved in other synchronous relationship.

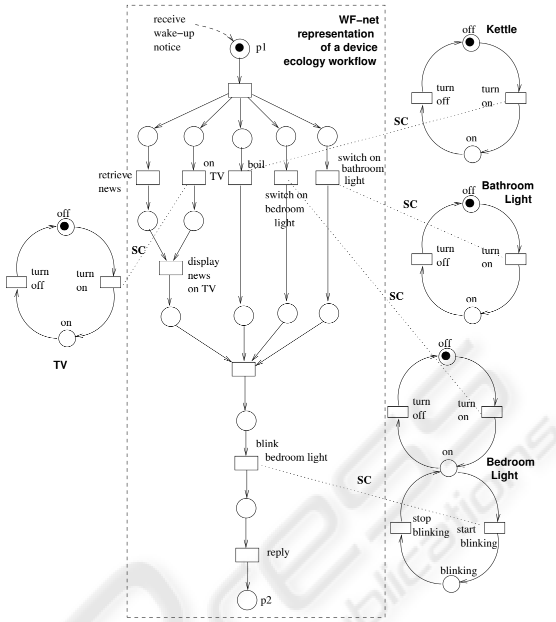


Figure 3: Device ecology workflow interacting with devices

## 4 ANALYSIS OF DEVICE ECOLOGY WORKFLOWS

Since we model device ecology workflows as WF-nets, for a device ecology behaviour model (*DEB*) with one WF-net, we can use existing WF-net analysis methods (Aalst, 1998) to verify the model. For a *DEB* with multiple WF-nets ($i > 1$), we can use the analysis methods for Interorganizational Workflow (IOWF) (Aalst, 1999).

These methods are concerned with detecting the soundness property of the workflow for behavioural correctness, which is defined in terms of the liveness and boundedness properties of Petri nets, as mentioned in section 3. The soundness property is consistent with device ecology workflows because we also want the *DEB* to complete the workflow execution. For example, it is desirable for the workflow in Figure 3 to be able to execute to completion such that the token can move from the source place `p1` to the sink place `p2`.

For each device, it is crucial to determine whether a particular device state is reachable. For example, in Figure 3, by executing the workflow, will the kettle reach the "on" state? Such questions can be answered by performing traditional reachability analysis (Aalst, 1998; Murata, 1989) on the Petri net model. A related question that can be answered is "Given the current state of a device, can the workflow execute to completion?". For instance, if the kettle is already in the "on" state before execution, is the workflow still executable? These questions and more can be answered by applying Petri net analysis methods on the DEB model.

Using reachability analysis, one can determine if certain desirable or undesirable states will be reached on execution of a workflow. Such states map to markings in a DEB. For example, at any point of the execution of a workflow, one might ask "will there be a state where all the lights in my house are off?" or "will there be a state where my front door is left unlocked?" (in the context of a home security workflow, for example).

## 5 CONCLUSION AND FUTURE WORK

We have described a Petri net model of device ecology workflows and of the observable behaviour of devices. A futuristic scenario of application of this work is where before one buys a device, he or she can verify if the device will work with the other devices he or she has, and whether existing workflows will be compatible with a new device which replaces an old one. There is still work to be done in applying our model to specific scenarios in order to evaluate the practicality of the proposed approach. We contend that our approach is practical given that the S-system of each device we use will not be overly complex, being an abstraction of only some aspect of the devices. In this paper, we do not deal with the full complexity of a million networked devices and things as this was not our goal, but we consider device ecologies where devices are of a granularity with some computational functionality (offering Web services, for example at the appliance level) and collections of devices that are user conceivable. Another avenue of research is to consider semantics-based matching of device ecology workflow tasks with devices in a device ecology.

## ACKNOWLEDGEMENTS

We would like to thank the Australian Research Council for partial funding of this work.

## REFERENCES

Aalst, W. (1999). Interorganizational workflows: An approach based on message sequence charts and Petri nets. *Systems Analysis - Modelling - Simulation*, 34(3):335–67.

Aalst, W. v. (1998). The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66.

AHAM (2002). Connected Home Appliances Object Modelling, AHAM CHA-1-2002.

Bentham, J. (2002). *TCP/IP Lean: Web Servers for Embedded Systems (2nd Edition)*. CMP Books.

Carabelea, C. and O.Boissier (2003). Multi-agent Platforms for Smart Devices: Dream or Reality? In *Proceedings of the Smart Objects Conference (SOC'03)*, Grenoble.

Corporation, M. (2002). Understanding *UPnP$^{TM}$*: A White Paper. *Software - Practice and Experience*.

Desel, J. and Esparza, J. (1995). *Free Choice Petri Nets*. Cambridge University Press.

Durand, Y., Vincent, S., Marchand, C., Ottogalli, F., Olive, V., Martin, S., Dumant, B., and Chambon, S. (2003). SIDRAH: A Software Infrastructure for a Resilient Community of Wireless Devices. In *Proceedings of the Smart Objects Conference (SOC'03)*, Grenoble.

Jahnke, J., DEntremont, M., and Stier, J. (2002). Facilitating the Programming of the Smart Home. *IEEE Wireless Communications*, pages 70–76.

Loke, S. (2003). Service-Oriented Device Ecology Workflows. In *Proceedings of the International Conference on Service-Oriented Computing*, pages 559–74.

Loke, S. (2004). Device ecology workflows with predictable observable effects. In *Proceedings of the Formal Foundations of Embedded Software and Component-Based Software Architectures (FESCA) Workshop (to appear)*.

Marples, D. and Kriens, P. (2001). The Open Services Gateway Initiative: An Introductory Overview. *IEEE Communications*, pages 2–6.

Matsuura, K., Haraa, T., Watanabe, A., and Nakajima, T. (2003). A New Architecture for Home Computing. In *Proceedings of the IEEE Workshop on Software Technologies for Future Embedded Ssytems (WSTFES03)*, pages 71–74.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–80.

Nakajima, T. (2003). Pervasive Servers: A Framework for Creating a Society of Appliances. In *Proceedings of the 1AD: First International Conference on Appliance Design*, pages 57–63.

Ramparany, F., Boissier, O., and Brouchoud, H. (2003). Cooperating Autonomous Smart Devices. In *Proceedings of the Smart Objects Conference (SOC'03)*, Grenoble.

Saif, U., Gordon, D., and Greaves, D. (2001). Internet Access to a Home Area Network. *IEEE Internet Computing*, pages 54–63.

Waldo, J. (1999). The Jini Architecture for Network-Centric Computing. *Communications of the ACM*, pages 76–82.

Workflow Management Coalition (1999). Workflow management coalition terminology and glossary (WFMC-TC-1011). Technical report, Workflow Management Coalition, Brussals.