

# Specification and verification of harmonized business-process collaborations

Alex Norta · Rik Eshuis

Published online: 1 April 2009  
© Springer Science + Business Media, LLC 2009

**Abstract** In the area of business-to-business (B2B) collaboration, original equipment manufacturers (OEMs) are confronted with the problem of spending a considerable time and effort on coordinating suppliers across multiple tiers of their supply chains. By supporting inter-organizational business-process collaborations with service-oriented technology, a scope for more efficient and effective supply-chain coordination is anticipated. This paper defines a formal framework, called *eSourcing*, for specifying structurally harmonized inter-organizational business-process collaborations. The framework permits verification of harmonized processes before their enactment. Moreover, the framework uses private and public layers to protect competitive knowledge of the individual partners. In the research project CrossWork, the eSourcing framework has been integral for harmonizing on an external level the intra-organizational business processes of a service-consuming and one or many service-providing organizations.

**Keywords** e-business · b2b · Collaboration · Service orientation · Supply chain · Inter-organizational · Business process · Visibilities · eSourcing

## 1 Introduction

Having explored and successfully applied workflow concepts for intra-organizational applications (van der Aalst and van Hee 2002; Leymann and Roller 1999), enterprises in the B2B domain are currently faced with the next challenge: achieving increased efficiency and effectiveness in the area of B2B collaborations. In the research project CrossWork (CrossWork 2009; Grefen et al. 2009; Mehandjiev and Grefen 2009), the problem was studied of how to establish inter-organizational B2B collaborations within the automotive industry, in which only essential business-process details are disclosed to other partners to enable collaboration, while the other details can remain hidden to safeguard competitive advantages. Establishing B2B collaborations is feasible by harmonizing the partner business processes. In this context, it is important to ensure that inter-organizational business-process harmonizing does not impose fixed standardized routing in the domain of a collaborating party. The harmonizing must allow a consumer, for CrossWork an automotive OEM, to ensure the presence of desired service content and behaviour in different degrees from a supplier.

As pointed out by Bussler (2002b), B2B collaboration is hampered if the parties involved share one common business-process definition or instance state that is split and shared for internal refinements, as this constitutes a violation of their competitive knowledge

---

A. Norta (✉)  
Department of Computer Science, University of Helsinki,  
P.O. Box 68 (Gustaf Hällströmin katu 2b),  
00014, Helsinki, Finland  
e-mail: alexander.norta@cs.helsinki.fi

R. Eshuis  
Faculty of Technology and Management,  
Department of Information Systems,  
Eindhoven University of Technology, P.O. Box 513,  
5600 MB, Eindhoven, The Netherlands  
e-mail: h.eshuis@tue.nl

protection, i.e., the business internals may be unintentionally revealed to collaborating counterparts. Furthermore, if process definitions are shared, correct message-exchange and message-transformation implementation, and the enactment of shared business rules becomes problematic, since adding a collaborating party quickly results in an explosion of the business-process definition.

To overcome these difficulties pointed out in Bussler (2002b), several authors have advocated the use of separate modelling levels (Chiu et al. 2004; Grefen et al. 2003). For this paper, we consider a simplified version of a specification framework for business process outsourcing proposed by Grefen et al. (2003) (see Fig. 1). An organisation, called the consumer, can outsource part of its business process to another organization, called the provider. The provider process interacts with the in-house consumer process. To specify such B2B collaborations, multiple modelling levels are needed. At the conceptual level, private business processes are specified independently from infrastructure and collaboration specifics. Conceptual processes are mapped to their respective technology-dependent internal level for enactment.

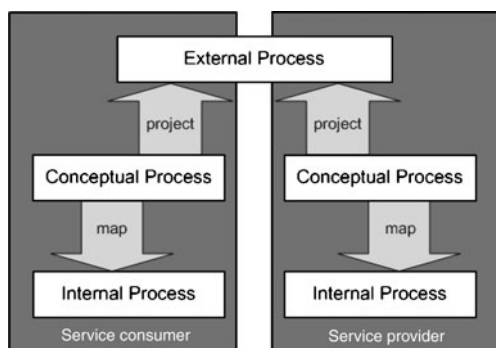
The internal level is not the focus of this paper, so we do not consider it in the sequel. External-level processes specify the public collaboration process between the two parties. Therefore, the external level stretches across the domains of the consumer and provider. Parts of the private conceptual processes are projected to the external level and compared by the collaborating parties to achieve a consensus. However, it is not a requirement to project the entire conceptual-level process to the external level, so an organization can choose to hide secret or competitive business internals at the conceptual level from its collaborating party.

While such specification frameworks (Chiu et al. 2004; Grefen et al. 2003) allow collaborating orga-

nizations to manage their process specifications at multiple levels, the frameworks do not define any concrete projection relations that can exist between conceptual and external-level business processes. Also, these frameworks do not address the question when the business processes of a consumer and a provider are harmonized, i.e., when the provider process realizes the requested consumer process and the provider process interacts correctly with the in-house consumer process, so no deadlock state is reached. For this paper, correctness refers to control-flow properties of the business processes; other perspectives such as data-flow or resources are not the focus of this paper. Note that business processes that perform correctly in isolation may, for example, contain a deadlock or livelock when linked together (van der Aalst 2002; Gomez et al. 2005). In B2B, if an inter-organizational business-process collaboration fails, the consequence is penalty payments, unsatisfied customers, lost time and money, and so on.

This paper defines a framework, called eSourcing, for specifying and verifying harmonized B2B process collaborations. Based on the three-level specification framework by Grefen et al. (2003), the framework distinguishes between conceptual and external-level processes. However, unlike the three-level framework, eSourcing uses a concrete formal language for modelling and analyzing business process, workflow (WF) nets. WF-nets have been used for modelling and analyzing business processes for correctness (van der Aalst 1998; Verbeek et al. 2001a). Based on WF-nets, the eSourcing framework formally defines different types of projections that can exist between conceptual and external level business processes. The availability of different types of projection gives flexibility to the collaborating parties with respect to how much business internals they want to disclose. One special type, grey box projection, allows the incorporation of a new private task in a conceptual provider process without violating the runtime behaviour at the external level. This gives providers flexibility in defining their own business processes while still realizing the external process agreed upon.

Next, the framework formally defines the notion of an eSourcing configuration, which is a set of conceptual-level and external-level models of consumer and provider with projection relations between them. We show that not every combination of projection types for consumer and provider-side is meaningful for eSourcing configurations. Certain types of eSourcing configurations are guaranteed to be harmonized, i.e., they are correct and realize the consumer's requested process, while others require any additional correctness check by a third party service.



**Fig. 1** A three-level specification framework for business process outsourcing

The structure of this paper is as follows. Based on case studies from CrossWork, the business context of eSourcing is informally introduced in Section 2 together with an example from CrossWork that shows how the processes of a consumer and a provider are inter-organizationally represented. As preliminaries, Section 3 presents existing theory on WF-nets that is used in the subsequent sections. Section 4 formalizes the eSourcing processes that are located on the conceptual and external levels of the three-level framework. Section 5 formally defines projection variations from the conceptual level to the inter-organizational level and describes different types of consensus constellations between collaborating parties. In Section 6, an eSourcing configuration is formally defined as a set of conceptual-level and external-level models of consumer and provider with consensus relations between them. In Section 7, a method is described to check the behaviour of an eSourcing configuration for correctness. Section 7 also shows that with certain types of inter-organizational process harmonizations, an established eSourcing configuration is guaranteed to be sound and to realize the consumer's process without requiring any additional check by a third party service. The architecture of a trusted third-party service is specified in Section 8. This trusted third-party service is needed for supporting the soundness-checking method of the previous section. Section 9 discusses related work from the domains of business-process formalizations and from B2B-collaboration research projects. Finally, Section 10 concludes the paper.

## 2 A motivating eSourcing example

To clarify the business context for eSourcing, we explain how it has been applied in the CrossWork project. The industrial partners in this project come from the automobile industry. In this industry, OEMs have several tiers of suppliers that agree to deliver systems collaboratively. For example, the OEM assembles cars with systems like a cockpit, or an engine, etc. These systems are manufactured by the second tier that receives components for those systems from their third-tier supplier.

The supply chain relationship between an OEM and suppliers resembles a pyramid where the OEM at the top spends considerable time and effort on aligning first- and second-tier suppliers for achieving the desired service provision. Additionally, the overall number of produced cars and also the number of variants is increasing while the lifetime of car-types is shortening, which means the number of cars per type is decreasing.

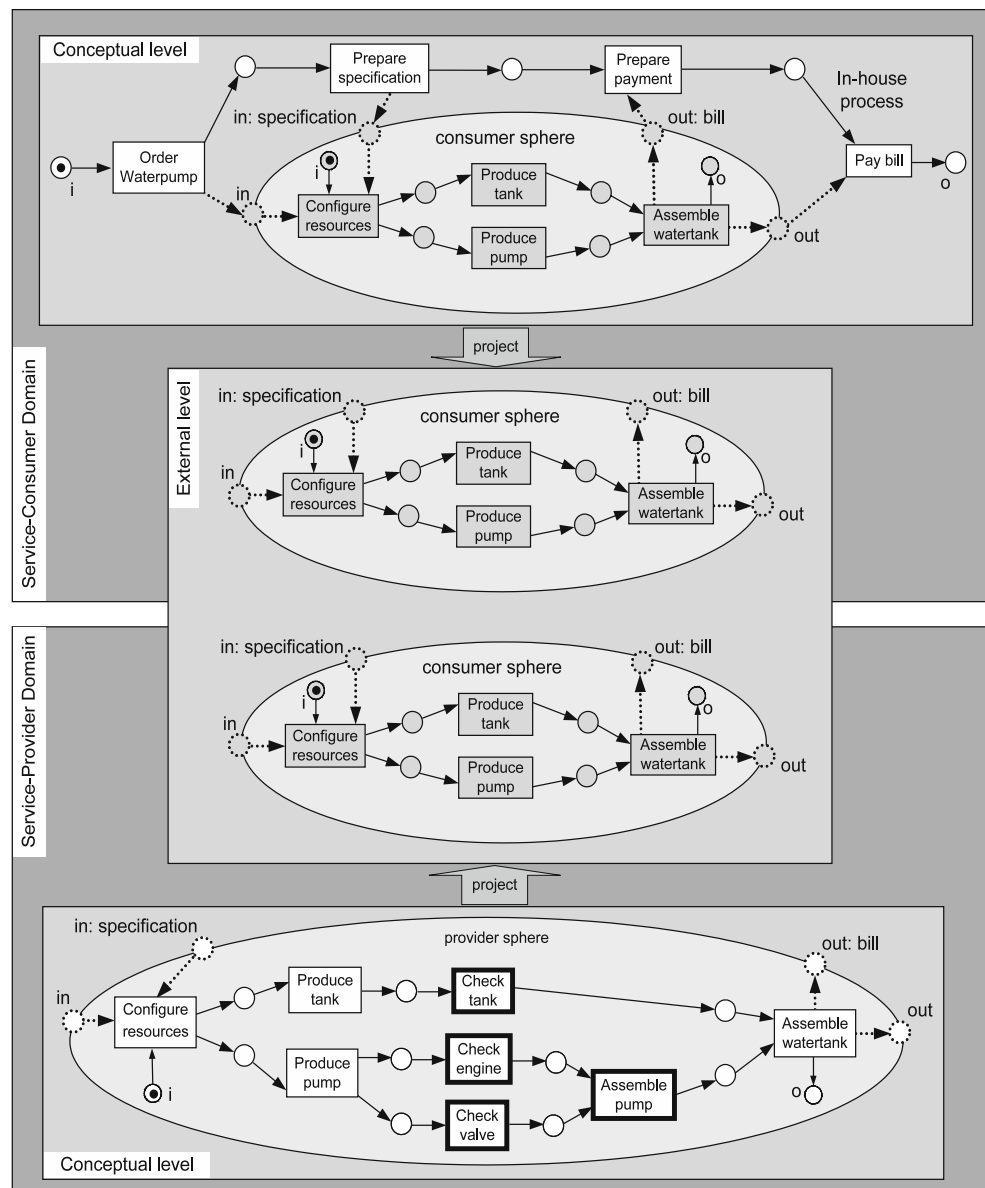
To deal with the resulting complexity in manufacturing as well as in design and development, OEMs are shifting parts of their activities down the organizational hierarchy. By applying eSourcing, specifying the inter-organizational process collaboration, this coordination effort between collaborating parties is relieved.

Based on a real-world scenario developed in the CrossWork project (CrossWork 2009) with industry partners from the automobile industry, the example is about an OEM sourcing a car water tank from a supplier. In Fig. 2, a corresponding eSourcing configuration is depicted, consisting of multiple processes. An eSourcing configuration comprises intra-organizational business processes of service-consuming and service-providing organizations that are harmonized dynamically on an external level into a B2B supply-chain collaboration. Here dynamically means that during process enactment collaborator organizations are found by searching business process market places and the subprocesses are integrated with the running process. Hence, a dynamic inter-organizational business process is formed dynamically by the (automatic) integration of the subprocesses of the involved organizations. Important elements of eSourcing are the support of different visibility levels of corporate process details for the collaborating counterpart and mechanisms for service monitoring and information exchange.

The processes in Fig. 2 are modelled in the WF-net formalism (van der Aalst 1998). Circles represent places and boxes represent transitions. Black dots represent the current active places. A process can change state by firing a transition that has all input places filled with a token; the transition then produces tokens on all output places (see Section 3 for formal details). In the center of Fig. 2 the external level is stretching across the respective domains of eSourcing parties where process harmonization takes place. Parts of the respective conceptual-level processes are projected to the external level for performing a harmonization to realize an automated and dynamically forged collaboration between partners.

Starting from the in-house process, the consumer orders a waterpump, which enables a consumer sphere that is sourced from a supplier. As we will explain in Section 4, a sphere is part of a conceptual-level process that is projected to the external-level. In a parallel branch the *specification* documentation and *payment* of the watertank are prepared. The consumer sphere in Fig. 2 is a subnet of the in-house process and it is entirely projected to the external level into a consumer contractual sphere. The projected contractual sphere of the service provider comprises matching content with equal transition labels. That way a consensus about the

**Fig. 2** An eSourcing-configuration example depicting the external and conceptual level



service content is established between the collaborating parties. On the conceptual level, a provider sphere is depicted that consists of the elements in the provider contractual sphere and additionally inserted refinement nodes, i.e., *check tank*, *check engine*, *check valve*, *assemble pump*. Note that the external level contains a process specification, not just a specification of the interface. The specification indicates what process the provider has to supply to the consumer.

The provider *receives* the watertank specification based on which an internal resource configuration takes place. In parallel branches the body of the water tank and the watertank pump are *produced*. These steps are followed by internally inserted *quality checks* that are not disclosed externally to the consumer. In Fig. 2, these

additional transitions are depicted with bolder lines. The watertank pump consists of separate parts that need to be assembled and finally the parallel branches are joined by a transition for assembling the overall watertank. During the assembly, a bill is retransmitted to the consumer where an output transition prepares the payment. While the enactment of the provider sphere is completed, the consumer's in-house process must still process the *payment* for completing the enactment of the overall eSourcing configuration.

The processes in Fig. 2 are not depicted with modelling constructs that indicate how the service consumer may monitor the enactment progress of the service provider and such monitoring is not the focus of this paper. However, in Norta (2007, b) so-called

monitorability constructs are described that bind process nodes from the external level and the respective conceptual levels with each other. This way, the consumer can monitor public tasks done by the provider.

The example assumes an out-sourcing scenario where during the setup time the consumer externalizes parts of the in-house process so that another party steps in as a provider. However, other interaction patterns are possible between a service consumer and provider during setup time (see Norta 2008 for more details). For example, internal-to-external sourcing means that the collaborating parties have internal processes that are only harmonized externally at the end of their setup interaction. In-sourcing means a service provider has a service that is subsequently integrated into the process of a service consumer. Thus, external harmonization is only performed at a later stage. Finally, external-to-internal means that externally harmonized processes are the starting point of interaction and the collaborating parties set up internal processes at a later stage just before enactment starts.

### 3 Preliminaries

We recall some preliminaries from Petri-net and workflow net theory that are used in the sequel.

#### 3.1 Petri nets

Workflow (WF) nets are defined as a subclass of a variant of Petri nets, labelled Place/Transition nets (Reisig and Rozenberg 1998). The definition we use here is taken from (Basten and van der Aalst 2001). Let  $U$  be a universe of identifiers; let  $AL$  be some set of *action labels* with  $\tau \in AL$  the silent action, whose role will be explained later. Let  $AL_v = AL \setminus \{\tau\}$  be the set of visible labels.

**Definition 1** (labelled P/T-net) A labelled Place/Transition net, or simply P/T-net, is a tuple  $(P, T, L, F, \ell)$  where

1.  $P \subseteq U$  is a finite set of places;
2.  $T \subseteq U$  is a finite set of transitions such that  $P \cap T = \emptyset$ ;
3.  $L \subseteq AL_v$  is a finite set of labels such that  $L \cap (P \cup T) = \emptyset$ ;
4.  $F \subseteq (P \times T) \cup (T \times P)$  is a set of directed arcs, called the flow relation;
5.  $\ell: T \rightarrow L \cup \{\tau\}$  is a labelling function.

A place  $p$  is called an *input place* of a transition  $t$  iff there exists a directed arc from  $p$  to  $t$ . Place  $p$  is called

an *output place* of transition  $t$  iff there exists a directed arc from  $t$  to  $p$ . Likewise, a transition  $t$  is called an *input transition* of a place  $p$  iff there exists a directed arc from  $t$  to  $p$ . Place  $t$  is called an *output transition* of place  $p$  iff there exists a directed arc from  $p$  to  $t$ . All places of a particular transition constitute the *preset* and all output places of a particular transition are called *postset*.

For the pre- and postsets an additional notation is relevant. Two auxiliary functions  $\bullet, \bullet: (P \cup T) \rightarrow \mathcal{P}(P \cup T)$  are defined that assign to each node its preset and postset, respectively. For any node  $x \in P \cup T$ ,  $\bullet x = \{y \mid yFx\}$ . To avoid confusion about which net a node belongs to, the preset and postset notation is augmented with the name of the net: Given a net  $N$ ,  $N \bullet x$  is the preset of node  $x$  in  $N$  and  $x \bullet N$  is the postset of node  $x$  in  $N$ .

**Definition 2** (Marked, labelled P/T-net) A marked, labelled P/T-net is a pair  $(N, s)$ , where  $N = (P, T, L, F, \ell)$  is a labelled P/T-net and where  $s$  is a bag over  $P$  denoting the marking (also called state) of the net.

At any time a place contains zero or more *tokens*, drawn as black dots. The *state*, often referred to as marking, is the distribution of tokens over places, i.e., a function  $s \in P \rightarrow \mathbb{N}$ . A Petri net  $PN$  and its initial marking  $s: P \rightarrow \mathbb{N}$  where for each  $p \in P$  there are  $n \in \mathbb{N}$  tokens, are denoted by  $(PN, s)$ . If confusion is possible, brackets are used to denote markings, e.g.,  $[p]$  is the marking with just a token in place  $p$ .

The number of tokens may change during the execution of the net. Transitions are the active components in a Petri net: they change the state of the net according to the following *firing rule*:

- (1) A transition  $t$  is said to be *enabled* iff each input place  $p$  of  $t$  contains at least one token.
- (2) An enabled transition may *fire*. If transition  $t$  fires, then  $t$  *consumes* one token from each input place  $p$  of  $t$  and *produces* one token for each output place  $p$  of  $t$ .

In Fig. 2, examples of Petri nets are depicted. The circles are places, the boxes are transitions, and the black dot in the  $i$ -labelled place of the in-house process is a token. Marking  $s'$  is reachable from  $s$  if there is a sequence of transitions such that, starting in  $s$ , firing the transitions results in state  $s'$ . For a formal definition, see Reisig and Rozenberg (1998). The following definitions represent a non-exhaustive selection of Petri-net properties that are sufficient for later sections. Let  $F^*$  be the reflexive-transitive closure of  $F$ .

**Definition 3** (Connectedness) A labelled P/T-net  $N = (P, T, L, F, \ell)$  is strongly connected iff for every two nodes  $x$  and  $y$  in  $P \cup T$ ,  $x F^* y$ .

The in-house process of Fig. 2 is not strongly connected because there is no directed path from  $o$  place to the  $i$  place. However, the in-house process would be strongly connected with an additional transition that connects the places with the  $i$  and  $o$  label.

**Definition 4** (Live) A marked, labelled P/T-net  $(N, s)$  is *live* iff, for every reachable state  $s'$  and every transition  $t$  there is a state  $s''$  reachable from  $s'$  which enables  $t$ .

The in-house process in Fig. 2 is live as no state is reachable where a transition would not be enabled to fire.

**Definition 5** (Bounded, safe) A marked, labelled P/T-net  $(N, s)$  is *bounded* iff for each place  $p$  there is a natural number  $n$  such that for every reachable state the number of tokens in  $p$  is less than  $n$ . The net is *safe* iff for each place the maximum number of tokens does not exceed 1.

Finally, it is possible that so-called dead transitions are contained in a P/T-net. A definition for dead transitions is given below.

**Definition 6** (Dead transition) Let  $(N, s)$  be a marked, labelled P/T-net. A transition  $t \in T$  is dead in  $(N, s)$  if and only if there is no marking  $s'$  reachable from  $s$ , such that  $s$  enables  $t$ .

The  $\beta$  operator (van der Aalst 2002) removes all dead transitions and corresponding places from the net.

**Definition 7** (Removing dead transitions:  $\beta$ ) Let  $(N, s)$  be a marked, labelled P/T net, with  $N = (P, T, L, F, \ell)$  and a set of dead transitions  $D \subseteq T$  such that  $T \setminus D$  does not contain dead transitions.  $\beta$  is a function such that it maps marked P/T-nets onto P/T nets:  $\beta(N, s) = (P', T', L', F', \ell')$  with  $T' = T \setminus D$ ,  $P' = \{p \in P | (\bullet p \cup p \bullet) \not\subseteq D\}$ ,  $F' = F \cap ((P' \times T') \cup (T' \times P'))$ ,  $dom(\ell') = T'$ , for  $t \in T' : \ell'(t) = \ell(t)$ , and  $L' = ran(\ell') \setminus \{\tau\}$ . If  $N$  is a WF-net with source place  $i$ , then  $\beta$  can also be applied without explicitly stating the initial marking, i.e.,  $\beta(N) = \beta(N, [i])$ .

The definition uses functions  $dom$  and  $ran$ , which return the domain and range of a function, respectively. So given a function  $f : X \rightarrow Y$ ,  $dom(f) = X$  while  $ran(X) = Y$ .

Petri-net formalisms are suitable for specifying the control-flow of tasks in a process. However, for the domain of business processes, simple Petri-nets are not sufficient; therefore, a subclass of Petri-nets has been developed.

*Workflow nets* Workflow is the operational aspect of a work procedure: how tasks are structured, who performs them, what their relative order is, how they are synchronized, how information flows to support the tasks and how tasks are tracked. A *Workflow* net (WF-net) (van der Aalst 1997, 1998; Ellis and Nutt 1993) models the control-flow dimension of a workflow. It should be noted that a WF-net specifies the dynamic behaviour of a single case in isolation. This means that every piece of work is executed for a specific case, which is also called a workflow instance. Examples of cases are handling an insurance claim, an order, a tax declaration, and so on. Different definitions of workflow nets exist, the one used here comes from van der Aalst (2002).

**Definition 8** (WF-net) van der Aalst (2002) Let  $N = (P, T, L, F, \ell)$  be a labelled P/T-net.  $N$  is a WF-net iff the following conditions are satisfied:

1. Instance creation:  $P$  contains an input (*source*) place  $i \in P$  such that  $\bullet i = \emptyset$ ;
2. Instance completion:  $P$  contains an output (*sink*) place  $o \in P$  such that  $o \bullet = \emptyset$ ;
3. Strongly connected:  $\tilde{N} = (P, T \cup \{\bar{i}\}, L, F \cup \{(o, \bar{i}), (\bar{i}, i)\}, \ell \cup \{(\bar{i}, \tau)\})$  is strongly connected ( $\bar{i} \notin T$ );
4. Label use:  $L = ran(\ell) \setminus \{\tau\}$ ;
5. Visible start: for any  $t \in T$  such that  $t \in i \bullet$ :  $\ell(t) \in AL_v$ , i.e.  $\ell(t) \neq \tau$ ;
6. Visible end: for any  $t \in T$  such that  $t \in \bullet o$ :  $\ell(t) \in AL_v$ , i.e.  $\ell(t) \neq \tau$ .

A WF-net has one input place ( $i$ ) and one output place ( $o$ ) because any case handled by the procedure represented by the WF-net is created when a token enters place  $i$  and ends when a token enters place  $o$ , i.e., the WF-net specifies the life-cycle of a case. The third requirement in Definition 8 has been added to avoid ‘dangling tasks and/or conditions’, i.e., tasks and conditions which do not contribute to the processing of cases. Note that transitions model tasks and places model conditions.

The in-house process of Fig. 2 fits the requirements of a WF-net as stated in Definition 8. To the left, the input place ( $i$ ) and to the right, the output place ( $o$ ) are depicted. Furthermore, the third condition is satisfied as all other nodes contribute to the processing of the WF-net.

The three requirements stated in Definition 8 can be verified statically, i.e., they only relate to the structure of the Petri net. However, there is another requirement which should be satisfied, namely, that the process will terminate eventually and the moment the procedure terminates there is a token in place  $o$  and all the other places are empty. Looking at the in-house process of Fig. 2, this requirement is fulfilled. This requirement is called the *soundness property*. Different definitions of soundness exist; the one used here is defined in van der Aalst (2002).

**Definition 9** (Soundness) A WF-net  $N$  is weakly sound iff the following conditions are satisfied:

- (i)  $(N, [i])$  is safe;
- (ii) for any marking  $s$  reachable from  $[i]$ ,  $o \in s$  implies  $s = [o]$ ;
- (iii) for any marking  $s$  reachable from  $[i]$ ,  $[o]$  is reachable from  $s$ .

$N$  is said to be strongly sound, or simply sound, if and only if, in addition there are no dead transitions, i.e.,  $(N, [i])$  contains no dead transitions.

The first condition of Definition 9 states that a sound WF-net is *safe*. The second condition focuses on the *proper completion* of a WF-net. If a marking in  $o$  is reached, all places are empty with the exception of place  $o$  that must contain one token. Finally, the third condition refers to a *completion option* that states that from the initial marking  $i$  that activates a case, it is always possible to reach the marking with one token in place  $o$  that results in a successful termination. The fourth condition about dead transitions that defines strong soundness, states that for each transition there is an execution sequence activating this transition. Removing all dead transitions and places connected to them from a weakly sound net results in a strongly sound net (van der Aalst 2002).

Note that the soundness property relates to the dynamics of a WF-net. Given a WF-net  $N = (P, T, L, F, \ell)$ , one wants to decide whether  $N$  is sound. In van der Aalst (1997) it is shown that soundness corresponds to liveness and boundedness. To link soundness to liveness and boundedness, an extended net  $\bar{N} = (\bar{P}, \bar{T}, \bar{L}, \bar{F}, \bar{\ell})$  is defined that is the P/T-net obtained by adding an extra transition  $\bar{t}$  (see

Definition 8) which connects  $o$  and  $i$ . Such an extended net is called the *short-circuited* net of  $N$  that allows for the formulation of the following theorem.

**Theorem 1** van der Aalst (2002) A WF-net  $N$  is sound iff  $(\bar{N}, [i])$  is live and safe.

This theorem shows that standard Petri-net-based analysis techniques can be used to verify soundness, which is of significant value for the area of intra-organizational business processes as a manual detection of control-flow problems such as deadlocks is difficult and time consuming. Hence, for the verification of complex WF-nets, tool support is available, e.g., Verbeek et al. (2001a, b).

When business processes need to be related inter-organizationally, it is desirable to establish a relationship that can be analyzed and checked for correctness. The following subsection presents such a relationship.

### 3.2 A notion of business-process inheritance

To express a client-server relationship between an original equipment manufacturer and suppliers, a special notion of business-process inheritance is used for eSourcing, namely the notion of *projection inheritance* (van der Aalst and Basten 2002; Basten and van der Aalst 2001) that can informally be described as follows:

*For two workflow process definitions  $A$  and  $B$ , where  $B$  contains all transitions in  $A$  and some additional ones, if it is not possible to distinguish between the behaviour of  $A$  and  $B$ , when the effects of the transitions that are in  $B$  but not in  $A$  are hidden (ignored), then  $B$  is a subclass of  $A$  under projection inheritance.*

Before projection inheritance can be defined formally, first an equivalence relation needs to be specified. This equivalence is based on the idea that a superprocess and a refined subprocess have the same (observable) behaviour. Concretely, *branching bisimilarity* (van Glabbeek and Weijland 1996) is such an equivalence.

The notion of a *silent action* is pivotal for branching bisimilarity and can be used to hide labels. Silent actions result from an abstraction that is defined as follows:

**Definition 10** (Abstraction) Let  $N = (P, T, L, F, \ell_0)$  be a labelled P/T-net. For any  $I \subseteq AL_v$ , the abstraction operator  $\tau_I$  is a function that renames all

transition labels in  $I$  to the silent action  $\tau$ . Formally  $\tau_I(N) = (P, T, L, F, \ell_1)$  such that, for any  $t \in T$ ,  $\ell_0(t) \in I$  implies  $\ell_1(t) = \tau$  and  $\ell_0(t) \notin I$  implies  $\ell_1(t) = \ell_0(t)$ .

Silent actions can not be observed and are denoted with the label  $\tau$ , i.e., only transitions in a Petri net with a label different than  $\tau$  are observable. Such a single label suffices as all internal actions are equal in the sense that they can not be observed by the collaborating counterpart.

Two marked, labelled P/T-nets are called *branching bisimilar*, denoted  $\sim_b$ , iff their observable behaviours coincide, i.e., abstracting from silent actions. For a formal definition, it is referred to van der Aalst (2002). Branching bisimilarity is an equivalence relation, i.e.,  $\sim_b$  is reflexive, symmetric, and transitive (see Basten 1998). Branching bisimilarity is used in the following definitions.

**Definition 11** (Behavioral equivalence of WF-nets) For any two sound WF-nets  $N_0$  and  $N_1$ ,  $N_0 \cong N_1$  iff  $(N_0, [i]) \sim_b (N_1, [i])$ .

After clarifying the notion of behavioural equivalence the initially presented notion of projection inheritance above can be defined formally. For that purpose the abstraction operator  $\tau_I$  of Definition 10 is useful for hiding labels. The definition of projection inheritance is presented as follows.

**Definition 12** (Projection inheritance) For any two weakly sound WF-nets  $N_0$  and  $N_1$ ,  $N_1$  is a subclass of  $N_0$  under projection inheritance, denoted  $N_1 \leq_{pj} N_0$ , iff there is an  $I \subseteq AL_v$  such that  $(\tau_I(N_1), [i]) \sim_b (N_0, [i])$ .

In Basten and van der Aalst (2001) details are contained about three projection-inheritance preserving refinement patterns, namely an inserted task, a loop, and a parallel branch. Examples of these refinement patterns are contained in following sections of this paper that explain formal properties of eSourcing configurations. After presenting the preliminaries, the subsequent sections explain properties of eSourcing collaboration.

## 4 Processes and spheres

This section formally defines the models used at the conceptual and external level of an eSourcing configuration. That way it is clarified how the respec-

tive processes and spheres relate to each other. The structure of this section is as follows. In Section 4.1, the conceptual-level process of the consumer and the provider are defined, followed by the definition of an operator that is instrumental for checking the correct termination of an in-house process. In order to determine the nature of correct termination, Section 4.1 gives a variation definition of the soundness property. Finally, Section 4.2 defines external-level models, called spheres.

### 4.1 Conceptual level

To specify conceptual-level processes, we use sound WF-nets as defined in the previous section. We use the term ‘sphere’ to denote a conceptual-level process that is mapped to the external-level. In Fig. 2 both conceptual levels contain spheres for the consumer and the provider.

**Definition 13** (Consumer sphere, provider sphere) A sphere  $N$  is a sound WF-net that is located at the conceptual level. If  $N$  is part of the consumer, it is called a consumer sphere  $CS$ . If  $N$  is part of the provider, it is called a provider sphere  $PS$ .

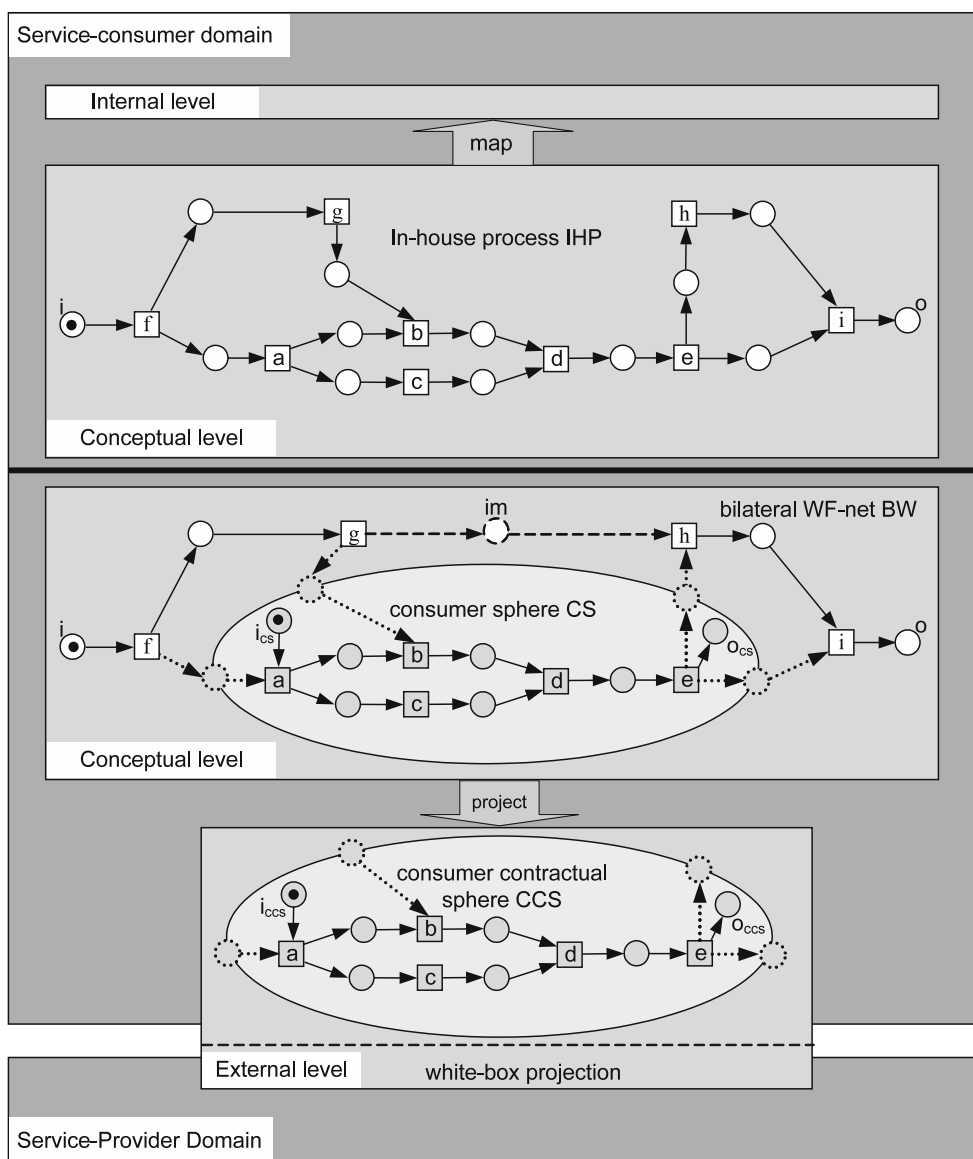
While the external level depicted in the abstract eSourcing example of Fig. 3 is explained in Section 5 where contractual spheres are investigated, this subsection focusses on the properties of the consumer’s conceptual level. The consumer sphere in Fig. 2 is contained in the in-house process of the consumer. When a consumer sphere is demarcated in an in-house process, gaps may occur, as illustrated by the in-house process of Fig. 3. The bottom conceptual level depicts that a consumer sphere is demarcated in the in-house process. Since this results in an unconnected remainder of the in-house process, it is considered invalid.

To resolve this issue, an extra place with the label  $im$  is introduced in the middle process of Fig. 3. Adding implicit place  $im$  results in a valid partitioning of the in-house process that results in a sound WF-net. Implicit places and their properties have been studied in Berthelot (1987), Colom and Silva (1990). Further details about implicit places and their use are contained in van der Aalst (2002). Adding the  $im$ -labelled place in Fig. 3 yields a P/T-net which is branching bisimilar to the original net.

The interface places depicted in Fig. 3 are located on the borders of the respective spheres and enable a systematic exchange of business-relevant information between the consumer sphere and the rest of the in-house process. To denote that the interface places and



**Fig. 3** The conceptual domain of the consumer



their connected arcs are separated from the consumer sphere in a valid bilateral WF-net, they are depicted with dotted lines. Furthermore, to support the clarity of the formalism in this paper, the interface places and connected arcs are also lined in a dotted way in depicted contractual spheres and provider spheres.

To formally model in-house processes, consumer spheres and their interaction, we introduce the notion of a bilateral WF-net, which consists of two interacting WF-nets. A bilateral WF-net is a simplification of an inter-organizational workflow net (IOWF-net) (van der Aalst 2002); Section 9 explains the differences between the two models. The definitions introduced in this subsection for bilateral WF-nets, such as activation safety, are also adapted from IOWF-nets.

**Definition 14** (Bilateral WF-net) A bilateral WF-net  $BW$  is a tuple  $(I, M, S, L, G)$  where:

1.  $I$  is a set of interface places;
2.  $M = (P_M, T_M, L_M, F_M, \ell_M)$ , the main process, is a sound WF-net, such that  $(P_M \cup T_M \cup L_M) \cap I = \emptyset$ ;
3.  $S = (P_S, T_S, L_S, F_S, \ell_S)$ , the subprocess, is a sound WF-net, such that  $(P_S \cup T_S \cup L_S) \cap I = \emptyset$  and  $(P_S \cup T_S \cup L_S) \cap (P_M \cup T_M \cup L_M) = \emptyset$ ;
4.  $L = L_M \cup L_S$  is the set of transition labels;
5.  $G \subseteq (I \times L) \cup (L \times I)$  is a set of directed arcs, called the interface flow relation.

Subprocess  $S$  corresponds to a sphere  $CS$  while  $M$  corresponds to an in-house process that activates and

deactivates  $S$ . The sphere can belong to either provider or consumer. The definition of the interface flow relation  $G$  connects interface places and transition labels. This facilitates the replacement of  $S$  by another process  $S'$  without changing  $G$ ; the binding of  $M$  to  $S'$  is achieved then by  $G$ , since  $S$  offers the same labels as  $S'$ . If  $G$  would connect interface places and transitions,  $S$  could not be replaced without changing  $G$  as well.

The definition of bilateral WF-net considers one subprocess only. However, this definition can be extended to multiple subprocesses by repeatedly partitioning the main process  $M$ . This way, collaborations between one consumer and several providers can be expressed. To define when a partitioning is valid, it is necessary to introduce a flattening operator that turns a bilateral WF-net into a P/T-net.

**Definition 15** ( $flat(BW)$ ) Let  $BW = (I, M, S, L, G)$  be a bilateral WF-net. The flattened P/T net  $flat(BW) = (P, T, L, F, \ell)$  where:

1.  $P = I \cup P_M \cup P_S \setminus \{i_S, o_S\}$ ;
2.  $T = T_M \cup T_S$ ;
3.  $L = L_M \cup L_S$ ;
4.  $\ell = \ell_M \cup \ell_S$ ;
5.  $F = F_M \cup F_S \cup \{(p, t) \in P \times T \mid (p, \ell(t)) \in G\} \cup \{(t, p) \in T \times P \mid (\ell(t), p) \in G\}$ .

Hence, the  $flat$  operator removes the  $i$  and  $o$ -labelled places from the sphere  $S$  that is contained in the main process  $M$  and creates a P/T-net.

A sphere  $S$  that is contained in a main process of a bilateral WF-net is activated if at least one of the places in the subflow  $S$  is marked (except the source and sink place). Since multiple activations of  $CS$  may lead to anomalies in an in-house process, the notion of activation safeness (van der Aalst 2002) is used.

**Definition 16** (Activation safeness) Let  $(N, s)$  be a marked, labelled P/T-net, where  $N = (P, T, L, F, \ell)$ . A subset of places  $P' \subseteq P$  is activation safe in  $(N, s)$  if and only if for any reachable state  $s'$  any transition  $t \in \bullet P' \setminus P' \bullet$ , and any place  $p \in P'$ : if  $s'$  enables  $t$ , then  $s'(p) = 0$ .

A set of places  $P'$  is activation safe if all transitions producing tokens for  $P'$  but not consuming tokens from  $P'$  are not enabled as long as there are tokens in  $P'$ . A sphere  $S$  that is a subflow of a bilateral WF-net is not activated multiple times if and only if the places of  $S$  are activation safe.

**Definition 17** (Soundness of bilateral WF-nets) Let  $BW = (I, M, S, L, G)$  be a bilateral WF-net and let  $N = (P, T, L, F, \ell)$  be the corresponding flattened net without dead transitions, i.e.,  $N = \beta(flat(BW))$ .  $BW$  is sound if and only if:

1. the flattened net  $N$  is a sound WF-net, and
2.  $P_S \setminus \{i_S, o_S\}$  is activation safe in  $(N, [i])$ .

Note that a flattened bilateral WF-net does not have dead transitions, i.e., the dead transitions are removed using  $\beta$ . The sphere contained in the main process of a bilateral WF-net must be activation safe.

The following definition states when a process  $N$  is validly partitioned by a bilateral WF-net  $BW$ .

**Definition 18** (Valid partitioning) Let  $N$  be a sound WF-net and let  $BW$  be a bilateral WF-net.  $BW$  is a valid partitioning of  $N$  if and only if  $BW$  is sound and  $N = \beta(flat(BW))$ .

The operator  $\beta$  removes dead transitions and places connected to them from a WF-net. For  $N$  to be validly partitioned by  $BW$ ,  $BW$  needs to be sound and the flattened net without dead transitions must equal  $N$ .

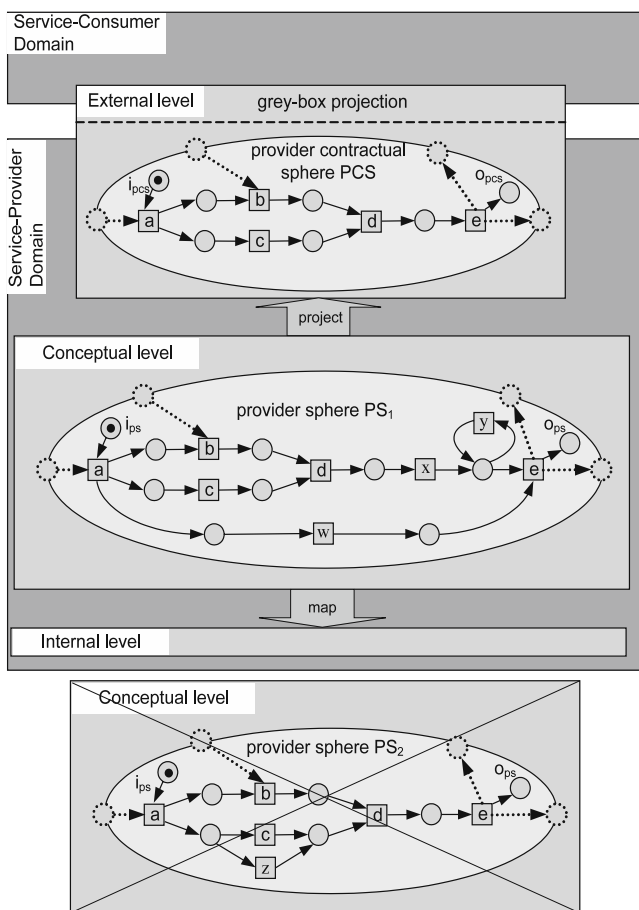
## 4.2 External level

Both the consumer and the provider have their own contractual spheres that belong to the external level of an eSourcing configuration.

**Definition 19** (Contractual sphere) A contractual sphere  $CS$  is a labelled P/T-net  $(P, T, L, F, \ell)$  such that  $\tau \notin ran(\ell)$ .

A contractual sphere does not use  $\tau$ -labels, since it does not make sense from a business point of view to outsource invisible actions. In an eSourcing configuration there are two contractual spheres, one for the consumer and one for the provider that are located on the external level. Having separate contractual spheres for the respective collaborating parties facilitates negotiations until a consensus is reached.

Examples of contractual spheres are depicted in Figs. 3 and 4. Note that a contractual sphere does not need to be a WF-net, which enables empty projections to the external level in the case of black-box projections, as is explained in the next section.



**Fig. 4** The provider contractual sphere and two provider spheres. The *bottom one* is illegal

### 5 Projections

The external level of an eSourcing configuration determines how much internal process details are exposed. The collaborating parties have the option of projecting different amounts of conceptual-level process content into their respective contractual spheres. To achieve a consensus about the nature of service provision, the respective contractual spheres must match in content. This subsection formally defines three projection options: white-box, grey-box, and black-box projection that differ in their level of projection abstraction.

#### 5.1 White-box projection

In the case of a white-box projection, the consumer or provider sphere is fully projected into the contractual sphere on the external level. This means the two spheres at the conceptual and external level must be identical. Figure 3 gives an example of white-box projection.

Mathematically, two nets are identical iff all their objects are pairwise identical. For P/T-nets the notion of an isomorphism is instrumental to express equality. The following definition is based on Reisig and Rozenberg (1998).

**Definition 20** (Isomorphism) Two nets  $N_0 = (P_0, T_0, L_0, F_0, \ell_0)$  and  $N_1 = (P_1, T_1, L_1, F_1, \ell_1)$  are isomorphic, denoted by  $N_0 \equiv N_1$  if there exist two bijections  $\alpha: P_0 \rightarrow P_1$  and  $\beta: T_0 \rightarrow T_1$  such that for every  $p \in P_0$  and  $t \in T_0$ ,

1.  $(p, t) \in F_0$  iff  $(\alpha(p), \beta(t)) \in F_1$ ;
2.  $(t, p) \in F_0$  iff  $(\beta(t), \alpha(p)) \in F_1$ ;
3.  $\ell(t) = \ell(\beta(t))$ ;
4.  $L_0 = L_1$ .

Using the notion of isomorphism, white-box projection is defined as follows:

**Definition 21** ( $\omega$ -projection) Let  $N_0 = (P_0, T_0, L_0, F_0, \ell_0)$  be a consumer or provider sphere and let  $N_1 = (P_1, T_1, L_1, F_1, \ell_1)$  be a contractual sphere. There is an  $\omega$ -projection from  $N_0$  to  $N_1$ , written  $N_0 \omega N_1$ , if and only if  $N_0 \equiv N_1$ , so  $N_0$  and  $N_1$  are isomorphic.

Theoretically, other equivalences like strong bisimulation (Milner 1989) are applicable as well. However, such equivalences allow syntactic modifications to the sphere, like creating duplicate branches. For example, strong bisimulation would allow to add another task *Produce pump* that is alternative to the existing task *Produce pump*. From a business point of view, such modifications do not make much sense.

#### 5.2 Grey-box projection

An example of grey-box projection is depicted in Fig. 4, for which a specific refinement relationship exists between the provider sphere and the corresponding contractual sphere. The provider sphere contains additional labels compared to the contractual sphere. However, during enactment the consumer only perceives process behaviour that is part of the external level and not what constitutes the provider’s refinement.

To realize such a refinement scenario as depicted in Fig. 4, *projection inheritance* (van der Aalst and Basten 2002; Basten and van der Aalst 2001) is employed. In Section 3.2, projection inheritance (see Definition 12) is explained together with the related notions of branching bisimilarity (van Glabbeek and Weijland 1996) and behavioural equivalence (see Definition 11). Details

about the external level follow in the sequel where contractual spheres are investigated.

The provider sphere  $PS_1$  of Fig. 4 is a subclass of the provider contractual sphere  $PCS$  according to projection inheritance. Hiding the inserted transitions does not violate the behaviour equivalence the consumer expects. Firstly, neither hiding the parallel branch with  $w$  nor hiding the execution of the inserted  $x$  violates the original behaviour of  $PCS$ . Secondly, the same holds for hiding the execution of  $y$  that merely postpones the execution of  $e$ . Figure 4 contains projection-inheritance preserving refinement patterns in  $PS_1$ , namely a parallel branch, inserted transition, and a loop. The parallel branch starts from the  $a$ -labelled transition and ends with the  $a$ -labelled transition containing a  $w$ -labelled transition. The inserted transition in Fig. 4 carries an  $x$  label and the loop example a  $y$  label.

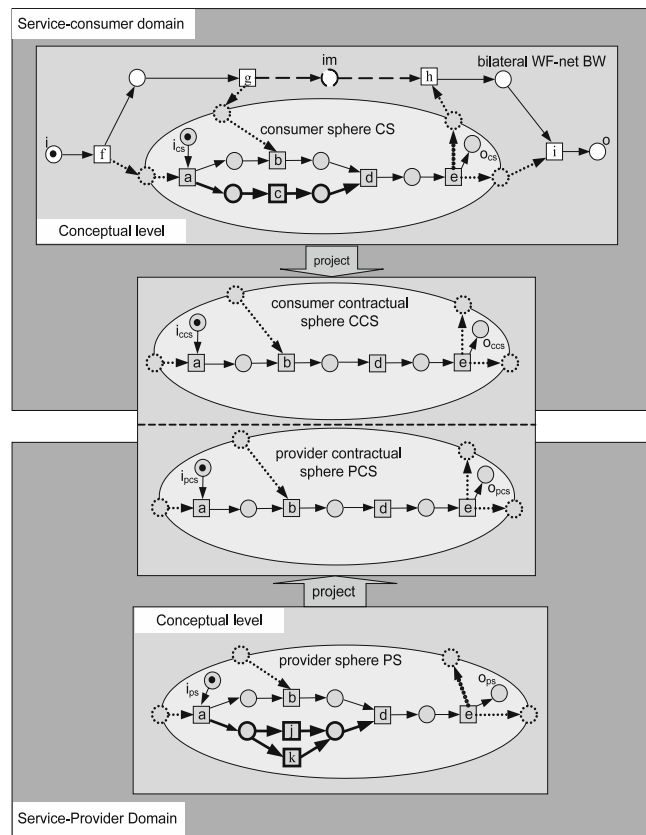
The sphere  $PS_2$  at the bottom of Fig. 4 shows a violation of projection inheritance in correlation to  $PCS$  because hiding the inserted newly labelled transitions results in a potential trace where  $a$  is followed by  $d$  without executing  $c$ . Hence, grey-box projection is defined as follows.

**Definition 22** ( $\gamma$ -projection) Let  $PS = (P_{PS}, T_{PS}, L_{PS}, F_{PS}, \ell_{PS})$  be a provider sphere and  $PCS = (P_{PCS}, T_{PCS}, L_{PCS}, F_{PCS}, \ell_{PCS})$  a provider contractual sphere. There is a  $\gamma$ -projection from  $PS$  to  $PCS$ , written  $PS \gamma PCS$ , if and only if:

- $PCS$  is a sound WF-net;
- $PS \leq_{pj} PCS$ ;
- $\{\ell(t) \mid t \in T_{PS} \wedge i_{PS} \in \bullet t\} = \{\ell(t) \mid t \in T_{PCS} \wedge i_{PCS} \in \bullet t\}$ ;
- $\{\ell(t) \mid t \in T_{PS} \wedge o_{PS} \in t \bullet\} = \{\ell(t) \mid t \in T_{PCS} \wedge o_{PCS} \in t \bullet\}$ .

For  $N_{PS}$  to be a projection-inheritance subclass of  $N_{PCS}$ , the latter must be a WF-net and both nets must be sound. Additionally, the labels of the starting transitions must be equal, and the labels of the ending transitions must be equal to support projection inheritance.

Note that  $\gamma$ -projection is limited to provider spheres and must not be performed with consumer spheres. The reason for this limitation is the non-adherence to projection inheritance that may occur during the enactment of an eSourcing configuration. In that case a consumer sphere and a provider sphere can have partly deviating control-flow constructs, leading to a violation of projection inheritance and therefore to a lack of contractual adherence. To see why, in Fig. 5 an example is depicted where both parties use grey-box projection. At the top of Fig. 5, the bold lined parallel branch is not projected to the external level. The provider sphere at the bottom of the figure depicts

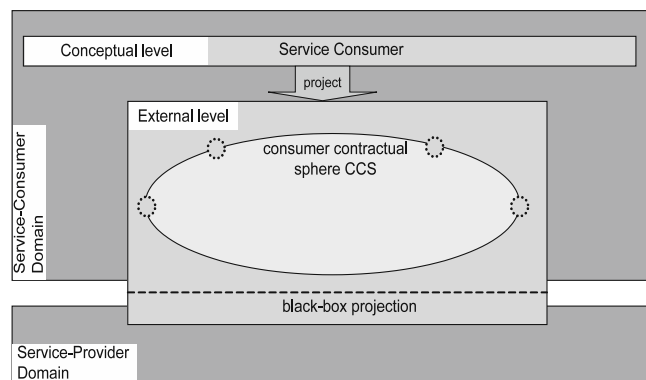


**Fig. 5** An example of both collaborating parties using grey-box projection

the bold lined refinement compared to the provider contractual sphere.

### 5.3 Black-box projection

The black-box projection does not project any content of the sphere to the external level. In Fig. 6, an example



**Fig. 6** A black-box projection to the external level

of  $\beta$ -projection is depicted. The notion of  $\beta$ -projection is defined as follows.

**Definition 23** ( $\beta$ -projection) Let  $N_0 = (P_0, T_0, L_0, F_0, \ell_0)$  be a consumer or a provider sphere, and let  $N_1 = (P_1, T_1, L_1, F_1, \ell_1)$  be a contractual sphere. There is a  $\beta$ -projection from  $N_0$  to  $N_1$ , written  $N_0 \beta N_1$ , if and only if  $N_1 = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ .

If the contractual spheres results from  $\beta$ -projection, nothing from the conceptual level is exposed. Still, the collaborating parties have to conjoin their spheres through the interface places. This can be solved in an architectural way, by allowing the consumer to inform the provider of interface specifics.

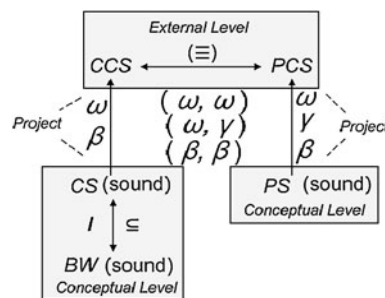
With  $\beta$ -projection it is not ensured that an eSourcing configuration is deadlock free. Instead the collaborating parties need to rely on a *collapsing method* for which an example is depicted in Fig. 8. Informally, the collapsing method replaces the consumer sphere of an in-house process with the provider sphere and the resulting net must be sound. For such a soundness check the tool Wolflan (Verbeek et al. 2001b) is instrumental.

For black-box projection the issue arises how a proper conjunction of the contractual spheres of a consumer and a provider is achievable. The problem occurs because for the formalization of eSourcing the labelling of interface places is omitted. To solve this problem, during the setup phase of an eSourcing configuration, the collaborating parties must inform each other about the conjunction labels of the channel flows in the consumer contractual sphere.

Black-box projection offers increased flexibility for external-level business process harmonization with the trade-off that harmonization is difficult to achieve as the business-process internals remain opaque. To alleviate this situation, it is necessary that collaborating parties have a mechanism available to support the checking of an eSourcing configuration realized with black-box projection. In Section 7.1 such a method is presented for checking the correctness of eSourcing configurations, which are formally defined in the following section.

### 6 eSourcing configurations

To realize a method for checking eSourcing configurations, it is relevant to first give a definition of an eSourcing configuration. This section presents a definition together with the accompanying properties of an eSourcing configuration. Figure 7 depicts a high-



**Fig. 7** A high-level overview of an eSourcing configuration

level overview of the different parts of an eSourcing configuration and how they relate to each other.

Shown at the left bottom of Fig. 7, a valid partitioned bilateral WF-net  $BW$  that is located on the conceptual level. The interface places  $I$  connect the consumer sphere  $CS$  as a subflow to the bilateral WF-net  $BW$ . On the right side of Fig. 7, the provider sphere  $PS$  is located on the conceptual level. The consumer can choose between an  $\omega$ -projection and  $\beta$ -projection from the consumer sphere to the consumer contractual sphere  $CCS$  on the external level. On the other hand, the provider can choose between an  $\omega$ -projection,  $\gamma$ -projection, and  $\beta$ -projection.

At the top of Fig. 7, the external level is depicted where the contractual spheres  $CCS$  and  $PCS$  are located. To have a contractual consensus between  $CCS$  and  $PCS$ , the respective contractual spheres need to be isomorphic, so  $CCS \equiv PCS$ . In the middle, the three projection tuples are depicted that are available for the collaborating parties to establish a contractual consensus between  $CCS$  and  $PCS$ . Either the consumer performs a white-box projection which the provider complements either with a white-box projection or a grey-box projection, or both parties use a black-box projection. At the left bottom of Fig. 7, the depicted conceptual level expresses that the consumer sphere  $CS$  is a subnet of the bilateral WF-net  $BW$  and the interface places  $I$  serve as connections. Based on these explanations, an eSourcing configuration is defined as follows:

**Definition 24** (eSourcing configuration) An eSourcing configuration is a tuple  $eSC = (IHP, BW, PS, CCS, PCS)$  such that:

1.  $IHP$ , the in-house process, is a sound WF-net;
2.  $BW = (I, M, CS, L, G)$  is a bilateral WF-net that is a valid partitioning of  $IHP$ , where  $CS$  is the consumer sphere (the subflow of  $BW$ );
3.  $PS$  is the provider sphere;
4.  $CCS$  is a consumer contractual sphere;

5.  $PCS$  is a provider contractual sphere;
6. there is a projection relation between  $CS$  and  $CCS$  on the one hand, and between  $PS$  and  $PCS$  on the other hand, such that either:
  - $CS\omega CCS$  and  $PS\omega PCS$ , or
  - $CS\omega CCS$  and  $PS\gamma PCS$ , or
  - $CS\beta CCS$  and  $PS\beta PCS$ ;
7. there is contractual consensus:  $CCS \equiv PCS$ .

The last requirement states that for contractual consensus, the contractual spheres of the collaborating parties must be isomorphic. An example of contractual consensus is given by Figs. 3 and 4. The consumer contractual sphere  $CCS$  of Fig. 3 and the provider contractual sphere  $PCS$  of Fig. 4 are isomorphic, although in the first case  $\omega$ -projection and in the latter case  $\gamma$ -projection results in the respective contractual sphere.

The definition states in which cases the bilateral WF-net, provider sphere, and the corresponding contractual spheres of the consumer and provider are properly related to each other. However, the eSourcing configuration can still be incorrect. For an eSourcing configuration, we define correctness as soundness of the bilateral WF-net containing the consumer main process and the provider sphere. The next section defines an operator that yields such a bilateral WF-net from an eSourcing configuration. The next section also introduces two methods that are instrumental for checking the correct termination and adherence of the provider to an agreed-upon service provision. In particular, we show that only for black-box eSourcing configurations an additional correctness check is needed. All other types of eSourcing configuration are guaranteed to be correct and the consumer main process together with the provider process are guaranteed to realize the original consumer in-house process.

### 7 Checking eSourcing configurations

An eSourcing configuration may contain errors, for example the collaboration of an in-house process and a provider sphere can result in a deadlock. To check eSourcing configurations for correctness, a verification method is used that takes advantage of the supporting theorems and their proofs from the domain of inter-organizational WF-nets (van der Aalst 2002). Firstly, it is shown how a *collapsing method* for eSourcing configurations can be specified by using the flattening function defined for bilateral WF-nets. Such a flattening is useful for checking soundness; however, it requires that a consumer and a provider expose their local

processes to a trusted third party. The architecture for this third party is painted in Section 8. Secondly, it is shown that grey-box and white-box eSourcing configurations are guaranteed to be correct (sound). Thus, for such eSourcing configurations, the collapsing method is not needed to check soundness, and the consumer and provider do not need to expose their processes to some trusted third party. Furthermore, for such eSourcing configurations it is ensured that the resulting inter-organizational business process realizes the in-house process.

Below, Section 7.1 presents a method that is necessary to ensure correct termination when the collaborating parties use black-box projections. When white-box or grey-box projections are used, the method of Section 7.1 is not applicable. Instead local termination checking of business processes inside the domains of collaborating parties suffices to ensure inter-organizational termination correctness, as Section 7.2 shows.

#### 7.1 Checking correct termination using collapsing

The practical method with which an eSourcing configuration is checked for control-flow problems and correct service provision is the collapsing method that is illustrated by means of an example in Fig. 8. Basically, the collapsing method replaces the consumer sphere of the in-house process with the provider sphere.

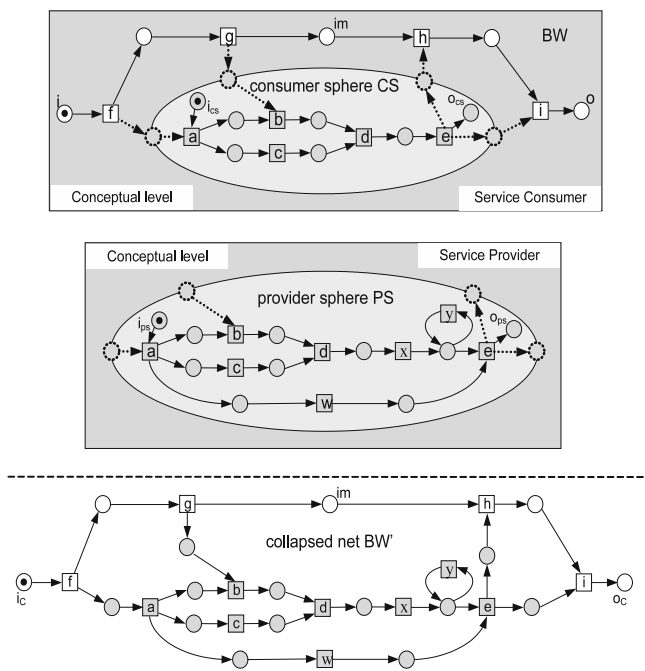


Fig. 8 A collapsed net

At the bottom of Fig. 8, the collapsed net for the running example is depicted, which can be verified for soundness with the tool Woflan (Verbeek and van der Aalst 2009). Moreover, Woflan checks whether the collapsed net is a subclass according to projection inheritance compared to the bilateral WF-net  $BW$  that is depicted in Fig. 3. If  $\beta$ -projection is used, the flattened net need not be a WF-net. In that case Woflan may still be used, although the tool then signals that the input is not a WF-net.

By applying the collapsing method, a bilateral WF-net  $BW'$  is yielded, i.e., a bilateral WF-net that specifies the enactment of inter-organizational collaboration between a consumer and a provider. To obtain  $BW'$ , an operator is defined for replacing a consumer sphere that is contained in the valid bilateral WF-net  $BW$  with a provider sphere. The resulting bilateral WF-net connects the internal process to the provider sphere so that  $BW'$  is linking the domains of a consumer and a provider for collaborative enactment.

**Definition 25** (*CSreplacePS*) Let  $eSC = (IHP, BW, PS, CCS, PCS)$  be an eSourcing configuration with  $BW = (I, M, CS, L, G)$ , then  $CSreplacePS(eSC)$  is defined as the bilateral WF-net  $BW' = (I, M, PS, L, G)$ .

Hence,  $CSreplacePS(eSC)$  is a bilateral WF-net in which the provider sphere replaces the consumer sphere, i.e., the provider sphere cooperates with the consumer's internal process. Figure 8 illustrates the  $CSreplacePS$  function. The process at the top is the bilateral WF-net  $BW$  and the bottom is the net  $BW'$  where the provider sphere  $PS$  replaces the consumer sphere contained in  $BW$ . Both processes are connected through interface places. Using  $CSreplacePS$ , we formally define the collapsing method. For the collapsing method the operator  $flat$  is used (see Definition 15).

**Definition 26** (*Collapsed net*) Let  $eSC = (IHP, BW, CS, PS, CCS, PCS)$  be an eSourcing configuration. The collapsed net is  $flat(CSreplacePS(eSC))$ .

The following section shows that for eSourcing configurations that do not use black-box projection, an established inter-organizational workflow is guaranteed to be sound.

### 7.2 Checking correct termination using projection inheritance

In this section, a theorem is introduced for white-box and grey-box eSourcing configurations. The theorem

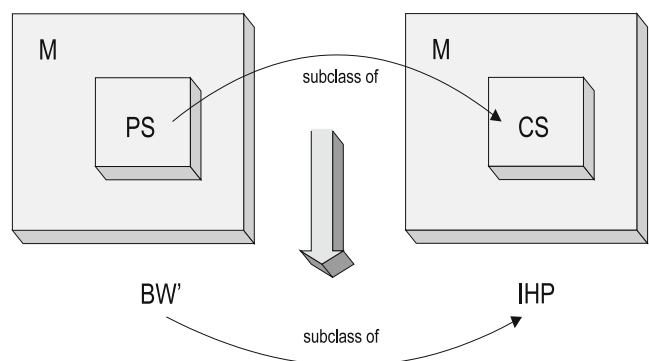
states that for such eSourcing configurations, the runtime bilateral WF-net is sound and the collapsed net is a subclass of the in-house process under projection inheritance. The main advantage of this result is that without the need for coordination among the collaborating parties, the resulting bilateral WF-net is guaranteed to be sound. Additionally, it is guaranteed that the eSourcing configuration realizes the in-house process, i.e., all the tasks specified in the in-house process are executed in the proper order. The proof is not difficult but relies on IOWF-nets (van der Aalst 2002); the proof is contained in Norta (2008).

**Theorem 2** (*Compositionality of eSourcing configurations*) Let  $eSC = (IHP, BW, PS, CCS, PCS)$  be an eSourcing configuration with  $BW = (I, M, CS, L, G)$  the bilateral WF-net,  $CS$  the consumer sphere,  $PS$  the provider sphere, and  $M$  the main process, with a projection relation between  $CS$  and  $CCS$  on the one hand, and between  $PS$  and  $PCS$  on the other hand, such that either  $CS\omega CCS$  and  $PS\omega PCS$ , or  $CS\omega CCS$  and  $PS\gamma PCS$ .

1.  $CSreplacePS(eSC)$  is sound, and
2.  $\beta(flat(CSreplacePS(eSC)))$  is a subclass of  $IHP$  under projection inheritance, i.e.,  $\beta(flat(CSreplacePS(eSC))) \leq_{pj} IHP$ .

The essence of the theorem about compositionality of projection inheritance is depicted in Fig. 9. It shows that the flattened net without dead transitions  $BW'$  is guaranteed to be a subclass of the in-house process  $IHP$  if the provider sphere  $PS$  is a subclass of the consumer sphere  $CS$  under projection inheritance. So  $BW'$  and  $IHP$  do not have to be checked explicitly for deciding whether  $BW'$  is a subclass of  $IHP$  under projection inheritance.

As a consequence of this theorem, it is possible to check the overall soundness of an eSourcing configura-



**Fig. 9** The essence of the compositionality of projection inheritance (van der Aalst 2002)

tion and the adherence of internal service provision to what is publicly agreed while maintaining independent and mutually opaque process domains of a consumer and a provider. The soundness of the eSourcing configuration is guaranteed without the need for any coordination among the provider and consumer. Hence, the employment of a trusted third party by the consumer and the provider is only required for checking contractual consensus. It is ensured that the tasks of the consumer sphere are executed in the proper order by the refined service provision.

For ensuring the overall correctness of an eSourcing configuration when  $\beta$ -projection is used, the architectural solution presented in the following section is instrumental.

### 8 Evaluation

In the previous section it is demonstrated how an eSourcing configuration is mapped to a bilateral WF-net which can be verified for correctness. First, the architecture of a corresponding verifier component is presented, followed by an introduction of XRL (eXchangable Routing Language) (Norta 2009) that shows how this language can be used to model the in-house process and provider process of an eSourcing config-

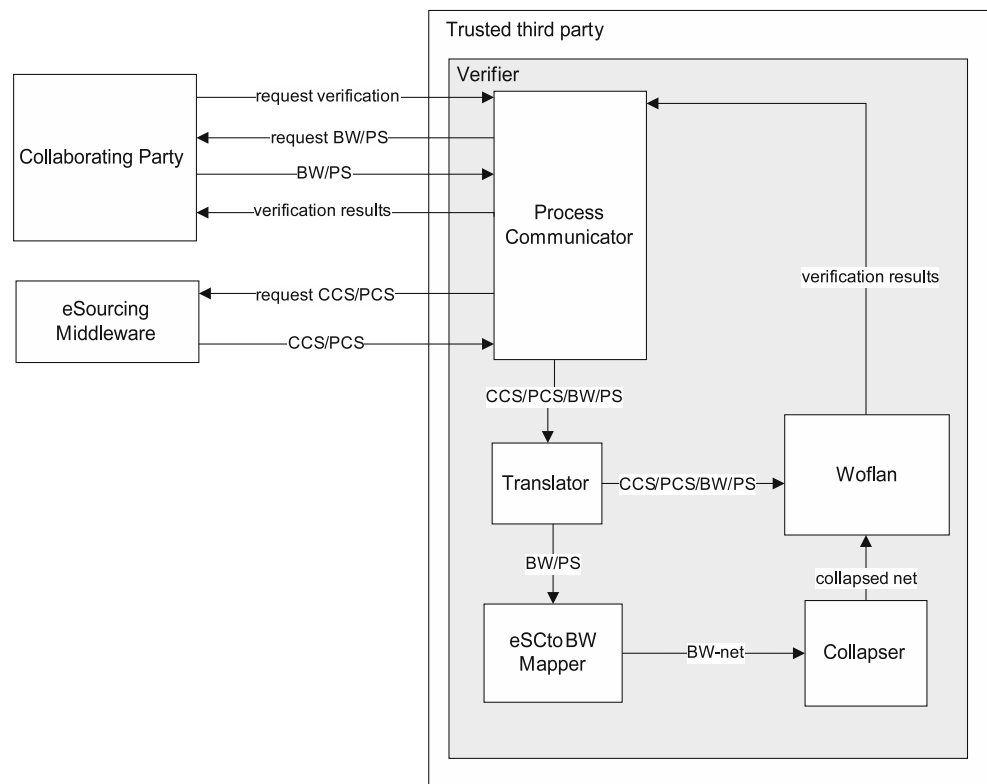
uration. As an enactment engine for XRL-formulated processes, the tool XRL/flower exists.

#### 8.1 A verifier component

Since neither consumer nor provider are willing to share its process definitions with each other, a trusted third-party service is needed to check the actual termination correctness. It needs to be stressed that Theorem 2 implies the verifier component in Fig. 10 is only needed for black-box eSourcing configurations, but not when a provider and a consumer agree on performing  $\omega$ -projection and/or  $\gamma$ -projection. Then the eSourcing configuration is either white-box or grey-box. In both cases, the third party only needs to check contractual consensus. All other checks are performed locally by the collaborating parties themselves. As a core part of the verifier component, the analysis tool Woflan (Verbeek and van der Aalst 2000, 2009; Verbeek et al. 2001a) checks control-flow abnormalities of submitted processes, e.g., deadlocks. Collaborating parties independently submit their conceptual processes for verification to this component without disclosing internal business details to each other.

Figure 10 shows the architecture of the trusted third-party verifier. A process-communicator component receives a request from the contracting client belonging

**Fig. 10** The trusted third-party verifier service in detail





to the domain of a collaborating party to perform a verification of a created eSourcing configuration. The process communicator requests the conceptual-level processes of all collaborating parties and the contractual spheres from the eSourcing middleware. Next, the collected processes from conceptual and external levels are delivered to a translator that converts the processes into a format the eSCtoBW-mapper component and Woflan can process. The first component delivers the resulting BW-net to a collapser component that creates a net, which Woflan verifies for soundness and projection inheritance. For the latter verification type, the BW-net is compared with the collapsed P/T-net. All processes delivered by the Translator component are separately verified by Woflan for control-flow problems.

## 8.2 XRL: An XML-based routing language

In CrossWork, the external- and conceptual-level business processes of an eSourcing configuration are formulated in XRL, which is an instance-based workflow language that uses XML for the representation of process definitions and WF-nets for its semantics. A catalogue of control-flow patterns (van der Aalst et al. 2000, 2007; Kiepuszewski 2002; Kiepuszewski et al. 2003) is contained in the definition of XRL (van der Aalst and Kumar 2003; Norta 2009) as routing elements that results in strong control-flow expressive power of XRL. These routing elements are equipped with WF-net semantics (van der Aalst et al. 2001), namely, every routing element stands for an equivalent WF-net snippet that can be connected with other routing elements into a bigger WF-net. Figure 11 shows extracts of an XRL code that are inspired by the provider process of Fig. 2, which depicts the watertank example.

The syntax of XRL is completely specified in a DTD and schema definition (Norta 2009). An XRL route is a consistent XML document, that is, a well-formed and valid XML file with top element route. The structure of any XML document forms a tree. In case of XRL, the root element of that tree is the route. This route contains exactly one so-called routing element. A routing element is an important building block of XRL as it can either be simple (no child routing elements) or complex (one or more child routing elements). A complex routing element specifies whether, when and in which order the child routing elements are carried out. In van der Aalst and Kumar (2003), Norta (2009) more details about the control-flow elements of XRL are contained.

Figure 12 shows the WF-net semantics of an XRL-task construct. Hence, since the semantics of XRL is expressed in terms of WF-nets (see Section 3.1), it permits the use of theoretical results and standard tools such as van der Aalst (1998); Verbeek et al. (2001a) for checking the notion of soundness and projection inheritance. The WF-net semantics of XRL is realized by mapping to PNML (Kindler et al. 2003a,b; Weber and Kindler 2003), an XML-based interchange format that permits the definition of Petri-net types. For that purpose a stylesheet translator is employed that contains mapping rules (van der Aalst et al. 2001) to PNML for every XRL control-flow construct.

For an evaluation and enactment application, the tool XRL/flower (Verbeek et al. 2002) is instrumental for XRL-modeled business processes. Figure 13 shows the enactment application of XRL/flower with a visual representation of enactment code resulting from XRL to PLMN mapping. Since XRL is based on both XML for syntax and WF-nets for semantics, standard XML tools can be deployed to parse, check, and handle XRL documents. The Petri-net representation allows

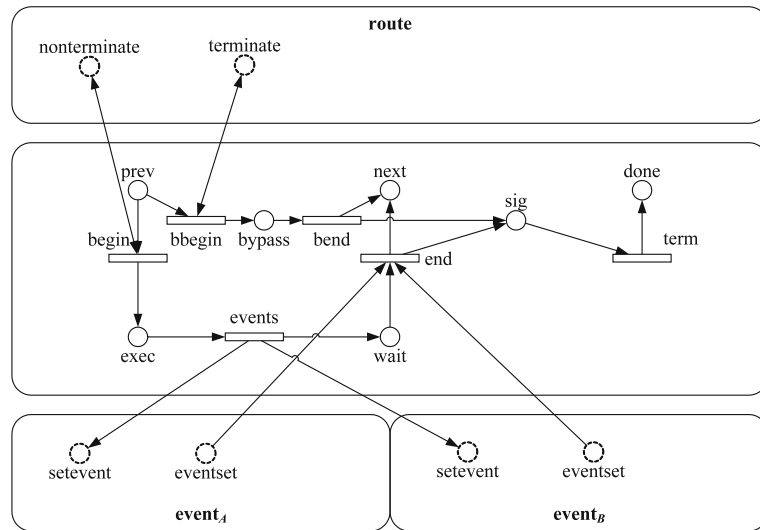
**Fig. 11** Extract of XRL-code example

```

01: <route name="purchase_1">
02:   <sequence>
03:     <task name="taskA" action="Configure resources." role="engineer"/>
04:     <parallel_sync>
05:       <sequence>
06:         <task name="taskB" action="Produce tank." role="warehouseworker"/>
07:         .....
08:       </sequence>
09:     </parallel_sync>
10:     <sequence>
11:       <task name="taskC" action="Produce pump." role="warehouseworker"/>
12:       .....
13:     </sequence>
14:   </parallel_sync>
15:   <task name="taskD" action="Assemble watertank." role="engineer"/>
16:   <while-do condition="$payment=5000">
17:     <true> ..... </true>
18:     <false> .... </false>
19:   </while-do>
20: </sequence>
21: </route>

```

**Fig. 12** The WF-net semantics of an XRL-task construct (van der Aalst et al. 2001)



for a straightforward implementation of the workflow enactment engine. XRL constructs are automatically transformed to Petri-net constructs. This allows for an efficient implementation and the system is easy to extend by employing an XSL translator for mapping routing elements to PNML. Thus, for supporting a new control flow primitive, only a transformation to the Petri-net format needs to be added and the engine itself does not need to change.

To model the contractual spheres of the external level, the XML-based markup language eSML (Norta 2005, 2008) was developed during the CrossWork project. eSML is instrumental as an external choreography language of the conceptual-level business processes and differs compared to other choreography languages such as AbstractBPEL (Alves et al. 2007) or WS-CDL (Jordan 2007). Firstly, in accordance with the high-level overview of an eSourcing configuration in Fig. 7, every service consumer and service provider specifies separate contractual spheres in one eSML

instantiation that are linked with additional language constructs to specify the degree of enactment observability for a service consumer. Secondly, the expressiveness of eSML permits its use as an electronic contract between collaborating parties that also specifies the collaborating parties, the service-reward, business-process rolebacks for a failed enactment, and so on.

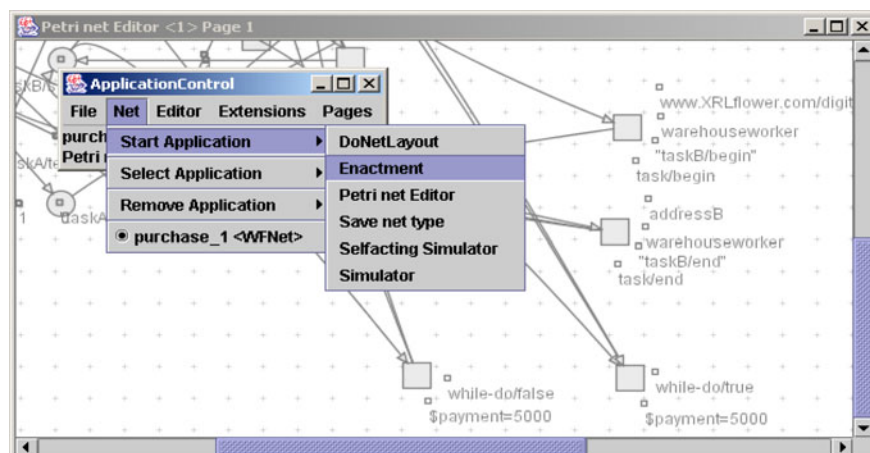
### 9 Related work

We first review existing formalizations for business process collaborations followed by a discussion of related research projects about inter-organizational business collaboration.

#### 9.1 Formalizations for business process collaborations

Bilateral WF-nets are a simplified version of IOWF-nets (van der Aalst 2002). First, an IOWF-net can

**Fig. 13** Enactment application of the Petri-net enactment module



reference an arbitrary number of WF-nets, while a bilateral WF-net only references two. More important, however, IOWF-nets are intended for modelling peer-to-peer collaborations, while a bilateral WF-net models a client/server collaboration. This difference is exemplified by the P2P (Public-To-Private) approach (van der Aalst 2002) in which IOWF-nets are used. In the first step of the approach, a publicly agreed WF-net is created. Secondly, the public WF-net is partitioned into domains for the collaborating parties. Finally, a private workflow is created for each domain such that the private workflow is a subclass of the corresponding part of the public workflow. Thus, by starting with a publicly agreed WF-net, the P2P approach implies an equal power constellation between collaborating business parties.

The above mentioned approach of IOWF-nets suits the current way of technically composing web services. However, from a business point of view, observing OEMs and their suppliers in CrossWork (CrossWork 2009) industry case studies shows that the business needs of B2B collaboration must be harmonized in a different way. Typically OEMs play a dominant role in B2B settings and try to exert tight control over their suppliers. Thus, OEMs impose a dominating client-server relationship on their suppliers that are usually tightly integrated into the OEM's in-house process. That way the OEM achieves fast production cycles, which is a competitive advantage. To support B2B collaboration in an electronic way, the client-server nature of inter-organizational business process management needs to be explicitly modeled. That is why we use bilateral WF-nets rather than IOWF-nets. Problems that occur in a technical realization of the IOWF-net approach are pointed out in Bussler (2002b) and discussed in Section 1 of this paper.

There are several other Petri net-based approaches in the literature that support service-based business process collaborations (Martens 2003a, b; Bonchi et al. 2007; Reisig et al. 2005; van der Aalst et al. 2008). In all these approaches, a service is represented by a net. Different services interact with each other by exchanging tokens through shared interface places, which models the asynchronous exchange of messages between the services. Given a service *A*, any other service *B* that can interact with *A* through the shared interface places without reaching a deadlock state is considered correct. These approaches resemble black-box projection. However, we use a collapsing method to check absence of deadlock, while the approaches focus on defining criteria or rules on the services that guarantee absence of deadlock, rendering an explicit check superfluous.

Regarding white-box and grey-box projection, these related approaches are completely different from ours. For these projection types, an interface of a service not only consists of places, but also of transitions and their labels (actions), and a service that replaces another service must not only agree on the interface places but also have similar behaviour. van der Aalst et al. (2008) claim this is too restrictive and give an example where a service with the same actions but different behaviour can still replace another service without leading to deadlock. However, the criterion they propose does not take transitions and their labelling into account. Consequently, using their criterion, a provider could perform actions that are different from the ones specified by the consumer. Clearly, this is not appropriate for outsourcing. Therefore, for grey-box and white-box projection, we have chosen interfaces that contain both transitions and their labels.

Next, there is other related work on process-based services. Some approaches focus on service contracts based on process algebra (Bravetti and Zavattaro 2007; Carbone et al. 2007). None of these approaches uses projection inheritance. Bravetti and Zavattaro (2007) use a testing preorder to check replacability, while Carbone et al. (2007) use a (bi)simulation approach. These process-algebraic approaches do not consider black-box projection. Finally, Benatallah et al. (2006) focus on service protocols, and analyse their behaviour to check their compatibility and replacability. However, service protocols are sequential. This makes their analysis considerably more simple than Petri nets, which are parallel. Also, Benatallah et al. (2006) do not consider different projection types, since these are not applicable to service protocols.

## 9.2 Research projects

The WISE project (Alonso 1999; Lazcano et al. 2001) resulted in a software platform for process-based B2B electronic commerce that focuses on support for a network of small and medium-sized enterprises. WISE relies on a central workflow engine to control inter-organizational processes that are termed virtual business processes. In WISE a virtual business process consists of a number of black-box services that are linked in a workflow process (Alonso 1999). A service is offered by an involved organization and can be a business process that is controlled by a local workflow management system. WISE does not support multiple levels of visibility, so there is no distinction between conceptual and external-level processes.

In the CrossFlow project (IBM Research 1999), inter-organizational business process collaboration was

investigated. In the context of this project, the formation of virtual enterprises is realized by dynamically out-sourcing a part of the consumer's process to a provider. A service matchmaker matches a service offering and a service request. The provider has adjustment flexibility as nodes of the assigned process can be internally refined on a lower process level. Based on the electronic contract, a service enactment infrastructure (Hoffner et al. 2005) is established dynamically, employing workflow technology. CrossFlow has an external level that spans across organizational domains where the process specification is part of a contract specification. The workflow specification language of the workflow management system IBM MQSeries Workflow (IBM 2009) forms the internal process level.

The integrated research project ATHENA (2009) investigates enterprise inter-operability in a holistic way with a technology-based approach that is guided by user requirements. It is the strategic objective of the project to enable networked businesses and governments. The expected results are of a technical-, business-, content-, and community-building nature where the topics of inter-organizational enterprise and process modelling, correspondingly required ontologies, service-composition frameworks, and enactment infrastructures are investigated. In the area of inter-organizational business process modelling, a business-level framework is proposed with a B2B process spanning across organizations, which is complemented by a public process and a private process in the domains of a collaborating organization. This business level is complemented with a technical and execution level that is responsible for enactment. For business-process modelling, an extension of event-process chain (EPC) formalism with so-called process-modules is proposed to achieve process abstraction. However, opting for EPC poses the difficulty that a verification of correct termination before enactment is not achieved. Furthermore, it is not possible to verify the extent to which collaborating parties adhere to their internal processes to what is externally promised.

Compared to these other research projects, the eSourcing framework offers a formal approach for specifying and verifying B2B process collaboration with multiple levels of visibility. Since CrossWork was a successor of CrossFlow, our work is naturally most closely related to CrossFlow. The main improvement over the CrossFlow approach is that eSourcing supports black-box and grey-box projections, whereas CrossFlow only considered white-box projections. However, CrossFlow did consider an alternative projection between conceptual and external-level processes: an external-level task can be decomposed into a set of conceptual-level tasks.

Elsewhere (Eshuis and Grefen 2008) we formalized aggregation in the context of process views. Adding aggregation to the eSourcing framework is straightforward, since it is a variant of transition refinement (replacing a transition by a subnet) which is well known from hierarchical Petri nets.

Additional approaches to inter-organizational process collaboration exist. In Bussler (2002a) the topic of B2B integration is dealt with in detail, ranging from differing integration concepts via required integration technology and their deployment to a discussion about integration standards, products, and ongoing research in the domain. These integration approaches deal predominantly with the technicalities involved and do not propose a collaboration model that is suitable for the way B2B collaboration between an OEM and suppliers unfolds. An advanced approach for enabling inter-organizational business collaboration (Bussler 2002b) investigates the use of public and private processes that solves the problems of message-exchange implementation, message transformation, and business rule handling between opposing parties. Open issues of Bussler (2002b) are how collaborating parties can regulate the degree of exposing their business internals to each other during setup and enactment time, which detailed options exist for binding the nodes of collaborating business processes, what issues arise for message exchanges between collaborating domains, which process properties must be adhered to for ensuring a smooth enactment phase, how can those properties be independently verified and evaluated during the setup phase without forcing the collaborating parties into exposing their business internals, and so on.

## 10 Conclusion

This paper focuses on control-flow issues that occur in inter-organizationally harmonized business processes where the collaborating parties disclose only as many business details as necessary. The presented formal eSourcing framework addresses the problem of overall termination correctness such as deadlocks or livelocks that may occur when business processes that correctly terminate on their own are linked together. To support the conceptual and external collaboration separation levels, the eSourcing framework caters for different projection methods of business-process details to the external level that result in variations of external business-process visibility, namely a white-box, grey-box and black-box visibility. Based on the adopted

pre-existing Petri-net theory, we define the control-flow properties for the processes of the conceptual and external levels of an eSourcing configuration. The paper also defines which conditions must hold in order to achieve contractual consensus between collaborating parties.

Based on a practical collapsing method the correct termination of a collapsed eSourcing configuration may be determined. The collapsing method is also suitable for verifying if the provider adheres to an agreed upon service request. Alternatively, we show that an eSourcing configuration that does not use black-box projection is guaranteed to be sound and moreover, the collaboration of the internal process with the provider process is guaranteed to realize the in-house process. The latter approach relies on a theorem about the compositionality of projection inheritance.

For the practical collapsing method, we propose a reference architecture of a trusted-third party to support the checking of an eSourcing configuration without forcing collaborating parties to reveal internal business secrets to each other. This reference architecture comprises the tool Woflan for checking the soundness of an eSourcing configuration before enactment. The trusted third party supports the collapsing method when the contractual parties use black-box projection. In this case, a check of the eSourcing configuration by a trusted-third-party service prevents the collaborating parties from revealing business internals to each other. For eSourcing configurations that use white-box projection and grey-box projection, a trusted third-party service may check contractual consensus. Instead, a local checking by the collaborating parties suffices for correct termination detection.

We mention several areas of future work. Firstly, it needs to be explored how the black-box projection can be supported in a better way. Hence a formalization extension should indicate the features of interface places without requiring the collaborating parties to exchange such information in an architectural solution. Despite using a black-box projection, the automated black-box projection should ensure that an eSourcing configuration still terminates correctly without forcing the collaborating parties into disclosing business-critical internals.

Future research must investigate the data-flow perspective in eSourcing configurations. For now we assume that data flows along a correctly terminating control flow. However, that does not solve all possible problems with respect to data flow, e.g., data mapping, data transformation, data consistency as part of long-lasting business transactions, and so on. Such data-flow issues particularly matter as eSourcing configurations

integrate heterogeneous legacy-system infrastructures of collaborating parties.

Further scope for future research results from experiences in evaluation studies of constructed application prototypes with CrossWork industry partners show that the expressiveness of available languages and applications for service-oriented computing do not sufficiently support eSourcing configurations. Hence, to enable eSourcing, future research needs to demarcate the existing gaps of support in service-oriented-computing languages and applications and explore extension options to fill them.

**Acknowledgements** We would like to thank the reviewers for their very valuable and helpful feedback. This research was partly conducted in the EU-FP6 research project CrossWork and partly in SOAMeS (Service Oriented Architecture in Multichannel e-Services). The latter project was financed by the Finnish Funding Agency for Technology and Innovation, VTT, Elisa Oyj, Kesko Oyj, Metsäteho Oy, and TietoEnator Processing & Network Oy.

## References

- Alonso, G., Fiedler, S.U., Hagen, C., Lazcano, A., Schuldt, H. & Weiler, N. (1999). WISE: business to business e-commerce. In *Proc. of the 9th international workshop on research issues on data engineering* (pp. 132–139). Sydney, Australia.
- Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., et al. (2007). Web services business process execution language version 2.0 (OASIS standard). WS-BPEL TC OASIS, <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>.
- ATHENA (2009) Advanced technologies for interoperability of heterogeneous enterprise networks and their application. <http://www.athena-ip.org/index.php>.
- Basten, T. (1998). *In terms of nets: System design with Petri nets and process algebra*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.
- Basten, T., & van der Aalst, W. M. P. (2001). Inheritance of behavior. *Journal of Logic and Algebraic Programming*, 47(2), 47–145.
- Benatallah, B., Casati, F., & Toumani, F. (2006) Representing, analysing and managing web service protocols. *Data & Knowledge Engineering*, 58(3), 327–357.
- Berthelot, G. (1987). Transformations and decompositions of nets. In W. Brauer, W. Reisig, & G. Rozenberg (Eds.), *Advances in Petri nets 1986 part I: Petri nets, central models and their properties. Lecture notes in computer science* (Vol. 254, pp. 360–376). Berlin: Springer.
- Bonchi, F., Brogi, A., Corfini, S., & Gadducci, F. (2007). A behavioural congruence for web services. In F. Arbab, & M. Sirjani (Eds.), *Proc. int. symposium on fundamentals of software engineering (FSEN 2007). Lecture notes in computer science* (Vol. 4767, pp. 240–256). Berlin: Springer.
- Bravetti, M., & Zavattaro, G. (2007). Contract based multi-party service composition. In F. Arbab, & M. Sirjani (Eds.), *Proc. int. symposium on fundamentals of software engineering (FSEN 2007)* (Vol 4767, pp. 207–222). Berlin: Springer.
- Bussler, C. (Ed.) (2002a). *B2B integration: Concepts and architecture*. Berlin: Springer.

- Bussler, C. (2002b). The application of workflow technology in semantic B2B integration. *Distributed and Parallel Databases*, 12, 163–191.
- Carbone, M., Honda, K., & Yoshida, N. (2007). Structured communication-centred programming for web services. In R. De Nicola (Ed.) *Proc. 16th European symposium on programming (ESOP 2007). Lecture notes in computer science* (Vol. 4421, pp. 2–17). Berlin: Springer.
- Chiu, D. K. W., Cheung, S. C., Till, S., Karlapalem, K., Li, Q., & Kafeza, E. (2004). Workflow view driven cross-organizational interoperability in a web service environment. *Information Technology Management*, 5(3–4), 221–250.
- Colom, J. M. & Silva, M. (1990). Improving the linearly based characterization of P/T nets. In G. Rozenberg (Ed.), *Advances in Petri nets 1990. Lecture notes in computer science* (Vol. 483, pp. 113–146). Berlin: Springer.
- CrossWork (2009) Cross-organisational workflow formation and enactment. IST no. 507590. <http://www.crosswork.info/>.
- IBM (2009) IBM MQSeries workflow. <http://www-4.ibm.com/software/mqseries/workflow>.
- IBM Research (1999). CrossFlow architecture description, Technical report, ESPRIT CrossFlow EP 28653.
- Ellis, C. A., & Nutt, G. J. (1993) Modelling and enactment of workflow systems. In M. A. Marsan (Ed.), *Application and theory of Petri nets 1993. Lecture notes in computer science* (Vol. 691, pp. 1–16). Berlin: Springer-Verlag.
- Eshuis, R., & Grefen, P. (2008). Constructing customized process views. *Data and Knowledge Engineering*, 64(2), 419–438.
- Gomez, J. M., Haller, A., & Bussler, C. (2005). A conversation oriented language for B2B integration based on semantic web services. web service semantics. In *Towards dynamic business integration workshop. In conjunction with the 14th international world wide web conference (WWW 2005), Chiba, Japan, May 2005*.
- Grefen, P., Ludwig, H., & Angelov, S. (2003). A three-level framework for process and data management of complex e-services. *International Journal of Cooperative Information Systems*, 12(4), 487–531.
- Grefen, P., Mehandjiev, N., Kouvas, G., Weichhart, G., & Eshuis, R. (2009). Dynamic business network process management in instant virtual enterprise. *Computers in Industry*, 60, 86–103.
- Hoffner, Y., Ludwig, H., C. Gülcü, & Grefen, P. (2005). Architecture for cross-organizational business processes. In *Proc. 2nd int. workshop on advanced issues of e-commerce and web-based Information systems* (pp. 2–11). Milpitas, CA, USA.
- Jordan, D., Evdemon, J., Alves, A., & Arkin, A. (2007). *Web services choreography description language 1.0*. <http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/>.
- Kiepuszewski, B. (2002). *Expressiveness and suitability of languages for control flow modelling in workflows*. PhD thesis, Queensland University of Technology, Queensland University of Technology, Brisbane, Australia.
- Kiepuszewski, B., ter Hofstede, A. H. M., & van der Aalst, W. M. P. (2003). Fundamentals of control flow in workflows. *Acta Informatica*, 39(3), 143–209.
- Kindler, E., Billington, J., & Christensen, S., et al. (2003a). The Petri net markup language: Concepts, technology, and tools. In W. M. P., van der Aalst, E. Best (Eds.), *Proc. of the 24th international conference, ICATPN 2003. Lecture notes in computer science* (Vol. 2679, pp. 483–505). Eindhoven: Springer.
- Kindler, E., et al. (2003b). *Petri net markup language (PNML) home page*. <http://www.informatik.hu-berlin.de/top/pnml/>.
- Lazcano, A., Schuldt, H., Alonso, G., & Schek, H. (2001). WISE: Process based e-commerce. *IEEE Data Engineering Bulletin*, 24(1), 46–51.
- Leymann, F., & Roller, D. (1999) *Production workflow: Concepts and techniques*. Upper Saddle River: Prentice-Hall PTR.
- Martens, A. (2003a). On compatibility of web services. *Petri Net Newsletter*, 65, 12–20.
- Martens, A. (2003b). On usability of web services. In C. Calero, O. Díaz, & M. Piattini (Eds.) *Proc. of 1st web services quality workshop (WQW 2003)*. Rome, Italy.
- Mehandjiev, N., & Grefen, P. (Eds.) (2009). *Dynamic business process formation for instant virtual enterprises*. Berlin: Springer (in press).
- Milner, R. (1989). *Communication and concurrency*. Englewood Cliffs: Prentice-Hall.
- Norta, A. (2005). eSML (electronic Sourcing Markup Language). In R. Eshuis, & I. Stalker (Eds.) *CrossWork: Requirements and concepts for agent-based workflow configuration, work package 2: Workflow modelling, task 2.2: Agent-based workflow configuration*.
- Norta, A. (2007). *eSourcing: Electronic sourcing for business to business*. <http://www.cs.helsinki.fi/u/anorta/research/eSourcing/>.
- Norta, A. (2008). *Exploring dynamic inter-organizational business process collaboration: Privacy protecting concepts for choreographing eSourcing in B2B with service-oriented computing*. VDM.
- Norta, A. (2009). XRL home page. <http://www.cs.helsinki.fi/u/anorta/research/XRL/xrlHome.html>.
- Reisig, W., & Rozenberg, G. (Eds.) (1998). *Lectures on Petri nets I: Advances in Petri nets. Lecture notes in computer science* (Vol. 1491). Berlin: Springer.
- Reisig, W., Schmidt, K., & Stahl, C. (2005). Kommunizierende Workflow-Services modellieren und analysieren. In *Informatik - Forschung und Entwicklung* (pp. 90–101).
- van der Aalst, W. M. P. (1997). Verification of workflow nets. In P. Azéma, & G. Balbo (Eds.), *Application and theory of Petri nets 1997. Lecture notes in computer science* (Vol. 1248, pp. 407–426). Berlin: Springer.
- van der Aalst, W. M. P. (1998). The application of Petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8(1), 21–66.
- van der Aalst, W. M. P. (2002). Inheritance of interorganizational workflows: How to agree to disagree without losing control? *Information Technology and Management*, 2(3).
- van der Aalst, W. M. P., & Basten, T. (2002). Inheritance of workflows: An approach to tackling problems related to change. *Theoretical Computer Science*, 270(1–2), 125–203.
- van der Aalst, W. M. P., & Kumar, A. (2003). XML based schema definition for support of inter-organizational workflow. *Information Systems Research*, 14(1), 23–47.
- van der Aalst, W. M. P., & van Hee, K. M. (2002). *Workflow management: Models, methods, and systems*. Cambridge: MIT.
- van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., & Barros, A. P. (2000). Advanced workflow patterns. In O. Etzion & P. Scheuermann (Eds.) *7th international conference on cooperative information systems (CoopIS 2000). Lecture notes in computer science* (Vol. 1901, pp. 18–29). Berlin: Springer.
- van der Aalst, W. M. P., Verbeek, H. M. W., & Kumar, A. (2001). XRL/Woflan: Verification of an XML/Petri-net based language for inter-organizational workflows (best paper award). In K. Altinkemer & K. Chari (Eds.) *Proceedings of the 6th informs conference on information systems and technology (CIST-2001)* (pp. 30–45). Linthicum: Informs.

- van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., & Barros, A. P. (2007). Workflow patterns home page. <http://www.workflowpatterns.com>.
- van der Aalst, W. M. P., Lohmann, N., Massuthe, P., Stahl, C., & Wolf, K. (2008). From public views to private views – Correctness-by-design for services. In M. Dumas & R. Heckel (Eds.) *Proc. WS-FM 2007. Lecture notes in computer science* (Vol. 4937, pp. 139–153). Berlin: Springer.
- van Glabbeek, R. J., & Weijland, W. P. (1996). Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43(3), 555–600.
- Verbeek, H. M. W., & van der Aalst, W. M. P. (2000). Woflan 2.0: A Petri-net-based workflow diagnosis tool. In M. Nielsen & D. Simpson (Eds.) *Application and theory of Petri nets 2000. Lecture notes in computer science* (Vol. 1825, pp. 475–484). Berlin: Springer.
- Verbeek, H. M. W., & van der Aalst, W. M. P. (2009). *Woflan home page*. Eindhoven, The Netherlands: Eindhoven University of Technology. <http://www.tm.tue.nl/it/woflan>.
- Verbeek, H. M. W., Basten, T., & van der Aalst, W. M. P. (2001a). Diagnosing workflow processes using Woflan. *The Computer Journal, British Computer Society*, 44(4), 246–279.
- Verbeek, H. M. W., Basten, T., & van der Aalst, W. M. P. (2001b). Diagnosing workflow processes using Woflan. *The Computer Journal*, 44(4), 246–279.
- Verbeek, H. M. W., Hirnschall, A., & van der Aalst, W. M. P. (2002). XRL/Flower: Supporting inter-organizational workflows using XML/Petri-net technology. In C. Bussler, R. Hull, S. McIlraith, M. E. Orłowska, B. Pernici, & J. Yang (Eds.) *Web services, e-business, and the semantic web. CAiSE 2002 international workshop, WES 2002, Toronto, Canada* (pp. 93–109). LNCS Springer, May 2002.
- Weber, M., & Kindler, E. (2003). The Petri net markup language. In H. Ehrig, W. Reisig, G. Rozenberg, & H. Weber (Eds.) *Petri net technology for communication-based systems advances in Petri nets, Lecture notes in computer science* (Vol. 2472, 455 p.). Berlin: Springer.
- Alex Norta** is a post-doctoral researcher at the University of Helsinki, Finland. He received his MSc degree (2001) from the Johannes Kepler University of Linz, Austria and his PhD degree (2007) from the Eindhoven University of Technology, The Netherlands. His PhD thesis was partly financed by the IST project CrossWork, in which he focused on developing the eSourcing concept for dynamic inter-organizational business process collaboration. His research interests include business process collaboration, e-business transactions, service-oriented computing, software architectures.
- Rik Eshuis** is an assistant professor at Eindhoven University of Technology, The Netherlands. He received an MSc degree (1998) and a PhD degree (2002) in Computer Science from the University of Twente. Afterwards, he spent one year as a visiting postdoctoral researcher at the CRP Henri Tudor and LIASIT in Luxembourg. He was involved in the IST project CrossWork, which focused on developing advanced process support for the automotive industry. He is on the editorial board of the Open Software Engineering Journal. His main research interest is in process-oriented information systems and service-oriented computing. This includes areas like process composition, process views, process integration, process modeling, process analysis, service composition, and service adaptation. He is member of ACM, IEEE, and IEEE Computer Society.