Inferring, Validating, and Coordinating the Commitments in a Workflow

Jiangbo Dang
University of South Carolina
dangj@engr.sc.edu

Michael N. Huhns
University of South Carolina
huhns@sc.edu

Abstract

A workflow can be represented as a set of Web services and a specification for the control and data flows among these services. It can also be represented as a colored Petri net (CPN), which is a graphical and mathematical modeling tool. In multiagent systems (MAS), a workflow is a dynamic set of tasks performed by a set of agents to reach a shared goal. We show herein that commitments among agents can be used to model a workflow and coordinate their execution of it. This paper presents methodologies to map an OWL-S model for a workflow to a CPN, and then to infer commitments and causal relationships from the CPN graph. With our methodologies, agents can collaboratively enact a workflow through commitment-based formalisms.

1. Introduction

As more complex business operations become candidates for automation, it is difficult to find a single service to fulfill a business need completely, and a combination of several services from different enterprises are required instead. The combinations are organized into workflows, which are becoming ubiquitous in business applications.

A commitment is a binary relationship binding two participants. It is a well-defined data structure with an algebra of operations that have a formal semantics. The agent that is bound to fulfilling the commitment is called the debtor. The agent that is the beneficiary of the commitment is called the creditor. A commitment has the form $C(a; b; q)$, where a is its creditor, b is its debtor, and q is the condition the debtor will bring about. A conditional commitment $C(a; b; p \rightarrow q)$ denotes that if a condition p is brought about, then the commitment $C(a; b; q)$ will hold. Commitments capture the dependencies among the agents with regard to the workflow.

As shown in Figure 1, this paper presents the relationships among an OWL-S workflow, a PNML colored Petri net, and agents' commitments in a MAS, and present methodologies to infer commitments from a workflow. Most existing workflow technologies apply only centralized methods to coordinate and monitor the execution of a workflow through its procedural specifications. In contrast, this paper advances the state of the art by describing how to (1) convert a Semantic Web service model (in OWL-S) to a business process model (in CPN); (2) infer the commitments of service agents involved in a workflow; (3) explore the use of Colored Petri Nets in workflow verification; (4) allow flexible workflow coordination through commitments.

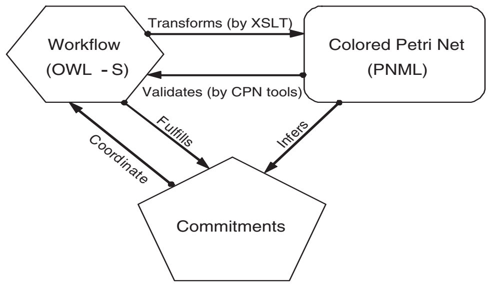

Figure 1. The relationships among a OWL-S workflow, a CPN, and commitments for service agents in an MAS implementation

2. A Motivating Scenario

In order to illustrate our methodology, we present a workflow scenario where several parties work together to produce a product. In Figure 2, *ProductRequestor* agent A initiates this workflow by sending a product requirement to *ProductMaker* agent B. To meet A's requirement, B designs this product and send its design to the third party *Analyzer* C. C performs some specific tests to ensure this design will meet the requirements. Once the product design is approved, B will generate the requirements for different parts of this product and send them to *PartsMaker* agent D. *PartsMaker* D will design these parts and send the design to C. If C approves the parts design, D will produce the parts for the product. In addition, if the design requires a specific treatment like drilling, a *Driller* agent E will drill the

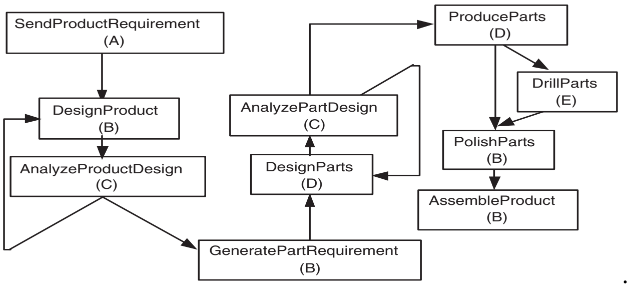

Figure 2. A Workflow Example

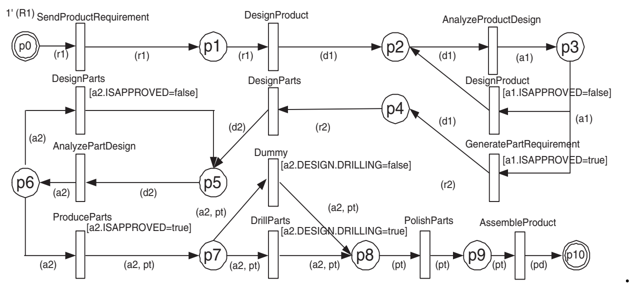

Figure 3. A ProduceProduct Petri Net

parts. Finally, *ProductMaker* B will polish the parts and assemble the product to finish this workflow.

We describe this workflow as a composite service, with its behavior described in terms of its inputs, outputs, preconditions, and results (IOPRs). Current workflow coordination mechanisms cannot deal with this scenario properly due to its dynamic nature. We believe that commitments are an appropriate abstraction to characterize and coordinate collaborative service agents in a workflow.

3. Inferring,Validating, and Coordinating the Commitments in a Workflow

A Petri net $N = (P, T, F)$ consists of a set of *transitions* T, a set of *places* P, and a flow relation F(*arcs*). *Place* is used to describe possible states of a process. The actions of a process are described by *transitions*. *Arcs* are used to connect places and transitions. They are indicated by ellipses, rectangles, and directed lines respectively. There are *Tokens* in places. A transition is enabled if there is at least one token in every place connected to a transition.

In a CPN, each token has a value referred to as *color*, which can be a schema or type specification. Transitions determine the values of the produced tokens based on the values of the consumed tokens. It is possible to specify a *guard* of a transition, which takes the colors of tokens to be consumed into account. These values match the inputs of a process, the outputs and results of a process, and the preconditions of a process from an OWL-S definition, respectively.

We use the Petri net markup language (PNML) to represent this workflow shown in Figure 3. Our mapping algorithm is based on a depth-first search and yields valuable information about the structure of the workflow. The input, output, precondition, and result of the OWL-S process are stored in the inscription of the ingoing arc, the inscription of the outgoing arc, the guard of the transition, and the inscription of the outgoing arc, respectively.

The coherent behavior of an MAS system is governed by interactions among the agents, and commitments are the proper abstraction to characterize the interactions for monitoring and control of the systems. Commitment $C(a; b; q)$ can be represented in terms of the IOPRs of q. Therefore, we can rewrite it as $C(a; b; (IOPR)_q)$ and a conditional commitment $C(a; b; p \to q)$ as $C(a; b; (I'OP'R)_q)$, where $I'_q = I_q \wedge O_p$, $P'_q = P_q \wedge R_p$.

Given a CPN workflow as the input, our inference algorithm produces a set of commitments for service agents involved in a workflow. In our example scenario. For *PartsMaker* that owns task *DesignParts*.

[DesignParts]	
Input:	PartRequirements
Output:	PartDesign
Pre-conditions:	Completed(GeneratePartRequirement)
	$\wedge$ ISAPPROVED=false

Given a CPN, we might be interested in (1) validity, i.e., is this a correct workflow with no error or conflict? (2) reachability, i.e., does the initial marking result in the correct result? and (3) liveness, i.e., does the workflow enter a "dead" state in which no further activity can occur? CPNs are accompanied by numerous techniques and tools that can provide formal verifications of these properties.

After deriving the commitments from a workflow, the participating agents involved in the workflow can be monitored and coordinated. These commitments can be used in two ways: coordinating and guiding the interactions among service agents in a competitive service-oriented environment, and monitoring and controlling the debtor agents to fulfill the workflow by fulfilling their committed tasks. Therefore, a centralized workflow execution engine is not necessary for coordinating and monitoring the execution of the workflows and for verifying the output of the workflow.

4. Conclusions

This paper presents methodologies to infer commitments from a workflow. CPN can be analyzed for validity, deadlocks, liveness, and other faults by a variety of CPN tools. More importantly, agents can collaboratively enact a workflow through commitment-based formalisms.