Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

# Semantic reengineering of business processes

Violeta Damjanović *

*Salzburg Research Forschungsgesellschaft, Knowledge based IS, Jakob Haringer Strasse 5-II, 5020 Salzburg, Austria*

ARTICLE INFO

*Keywords:*
Business process modelling
Model transformations
Process algebra
Model-driven architecture

ABSTRACT

This paper discusses transforming ontological models into non-ontological models of business processes, when the process of articulating different data models is known as reengineering domains. As a crucial factor in achieving interoperability and semantic reengineering of the domains with the different levels of semantic representation (expressiveness), we point out the role of foundational ontology that serves to enable global meaning of the process knowledge and that is, in this paper, additionally connected with the process theory (process algebra). The main focus of the process theory is on the system that interacts with one another, such as the business processes, whereas the main idea of semantic reengineering is transforming ontological models into the semantic business processes that can be (semi-)automatically executed via a workflow engine. Therefore, as a promising solution in achieving interoperability between the real enterprise needs and the business process models, we involve the Pi-Calculus as a process theory to provide semantics between the ontological models based on the DOLCE Description and Situation (D&S) Plan and Tasks Ontology (DDPO) and the business process models that are expressed in Business Process Execution Language (BPEL).

## 1. Introduction

The future business process domains will be certainly grounded in ontological knowledge with the ability to reuse existing knowledge, articulate the new business processes based on them and provide ontological explication of activities, steps and procedures involved in creating software solutions for the business processes. In this paper, we use the DOLCE Description and Situation (D&S) Plan and Tasks Ontology (DDPO) model as a semantic data model to help us get a good understanding of the task and plan models and provide implicit rules for concepts that explain behaviour as well as structure of both executable and abstract business processes. In other words, we use the main DDPO ontological concepts such as *non-physical*, *social*, *non-agentive* [1] for representing the process knowledge.

Nowadays, for the sake of combining the ontological resources with the non-ontological resources such as business processes, new tools and methods to support process knowledge creation are needed. Hence, we propose semantic reengineering of the ontological knowledge domains to semantically support (semi-)automatic creation of business processes via process knowledge that is additionally expressed in the form of Pi-Calculus. Our approach also builds on related work in this area connecting ontologies and business processes, such as [2] that describes an ontology for executable business processes using the formalism of the Web Service Modelling Language (WSML); [3] that proposes a set of ontologies and formalisms, and defines the scope of these ontologies by giving competency questions that is a common approach in the ontology engineering; [4] that explores the behavioural aspects of Web services and

* Tel.: +43 699 10955020.
*E-mail address:* violeta.damjanovic@salzburgresearch.at

0306-4379/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2009.06.003

proposes using Description Logics (DL) for their formalization.

As the main motivation of this paper, we address the problem of exchanging the information between the semantic data models (based on the DDPO and expressed in Web Ontology Language (OWL)) and the real business models (expressed in Business Process Execution Language (BPEL)). The business processes from the domain of mechatronic engineering are chosen to explicate the main challenges in semantic reengineering of business processes.

## 2. Description of a business process domain

Mechatronic engineering is one of the most recent branches of engineering, which has increasing impact on many sectors of the economy and on the society overall [5]. The mechatronic engineering processes cover an interdisciplinary combination of different domains comprising of mechanical engineering, electrical engineering, and software engineering. Each domain covers a specific mechatronic field, while the intersection of knowledge models between these different engineering domains is connected through the mechatronic processes. The ability to integrate various mechatronic knowledge models and to articulate business processes based on them are central key to effectively manipulating knowledge resources. Hence, the DOLCE foundational ontology has been chosen as the working hypothesis from which the modelling of the mechatronic domain and process ontologies started [5].

The DOLCE ontology aims at capturing the main cognitive categories, such as *endurants* and *perdurants*, underlying existing ontologies and human commonsense that appears even in the domain of mechatronics. The main difference between *endurants* and *perdurants* is related to their behaviour in time, e.g. *endurants* can change in time, while *perdurants* cannot. Furthermore, the DDPO module specialises the concepts and relations defined in DOLCE, and extends Descriptions and Situations (D&S) module that involves a representation language for the tasks and processes [6]. In this paper, the ability to transform the DDPO theoretical model into the BPEL business processes is represented as an essential task towards semantically meaningful business process execution via the BPEL workflow engine. As BPEL is not equipped with the formal semantics, exchanging the information between semantics and non-semantics data models needs a mechanism that is able to couple semantic expressions of the DDPO process models with the business process models. Therefore, we have involved the Pi-Calculus process algebra to provide accurate transformations between the different levels of semantic expressiveness of ontological models (expressed in OWL and based on the DDPO) and business process models (expressed in BPEL). In this way, different levels of semantic expressiveness enable the original semantic definitions to take a part of the business processes implementation, and to reduce the possible mistakes in semantic interpretation of the different domains.

Here, we represent a business process trinity mechanism that translates semantic expressions of the DDPO theoretical models into the Pi-Calculus and additionally provides the underlying semantics for the BPEL executable processes.

## 3. Business process trinity: DDPO, Pi-Calculus, BPEL

In this section, a brief description of the following three elements of the business process trinity mechanism is given: (a) the syntax and semantics of the DDPO theoretical model, (b) the Pi-Calculus syntax and operational semantics, and (c) the syntax and semantics of the BPEL executable processes.

### 3.1. DDPO

The purpose of the DDPO is to specify DOLCE plans at the abstract level by using First Order Logic (FOL).

**Table 1**
The syntax of the DDPO theoretical model: basic concepts and relations [6].

| Syntax of the DDPO theoretical model | Semantics |
|---|---|
| $\mathbf{Plan}(\boldsymbol{x})\rightarrow\text{Description}(x)$ | Plan is a description of *x* |
| $\text{Plan}(\boldsymbol{x})\rightarrow\exists t.\ \text{Task}(t)\ \wedge\text{Uses}(x,t)$ | … that uses at least one task *t*… |
| $\text{Plan}(\boldsymbol{x})\rightarrow\exists c.\ ((\text{AgentiveRole}(c)\ \vee\text{Figure (c)})\ \wedge\text{Uses}(x,c)$ | … and at least one agentive role *c* or Figure c… |
| $\text{Plan}(x)\rightarrow\exists g.\ \text{Goal}(g)\ \wedge\text{ProperPart}(x,g)$ | … and that has at least one goal *g* as a proper part |
| $\mathbf{Subplan}(\boldsymbol{x}) = df\ \text{Plan}(x)\ \wedge\exists y.\ \text{Plan}(y)\ \wedge\text{ProperPart}(y,x)$ | A subplan *x* can have varied proper parts, including other plans *y* |
| $\mathbf{Goal}(\boldsymbol{x}) = df\ \text{Desire}(x)\ \wedge\exists \text{p}.\ \text{Plan}(p)\ \wedge\ \text{ProperPart}(p,x)$ | A goal *x* is defined as a desire that is a proper part of a plan *p* |
| $\mathbf{PlanExecution}(\boldsymbol{x}) = df\ \text{Situation}(x)\ \wedge\exists y.\ \text{Plan}(y)\ \wedge\text{P-SAT}(x,y)$ | Plan execution (execution of a plan *x*) is defined as a situation that proactively satisfy a plan *y* (P-SAT) |
| $\mathbf{GoalSituation}(\boldsymbol{x}) = df\ \text{Situation}(x)\ \wedge\exists y.\ \text{Goal}(y)\ \wedge\text{SAT}(x,y)$ | A goal situation *x* is defined as a situation that satisfy a goal *y* |
| $\text{GoalSituation}(x)\rightarrow\forall y,p,s.\ (\text{Goal}(y)\ \wedge\text{SAT}(x,y)\ \wedge\text{Plan}(p)\ \wedge\text{ProperPart}(p,y)\ \wedge\text{P-SAT}(s,p))\rightarrow\neg\text{ProperPart}(s,x)$ | A goal situation *x* is not a proper part of the execution of a plan *x* |
| $\mathbf{Precondition}(\boldsymbol{p},\boldsymbol{s})\rightarrow\text{Plan}(p)\ \wedge\text{Situation}(s)$<br>$\text{Precondition}(p,s)\rightarrow\forall \text{s}1.\ (\text{PlanExecution}(s1)\ \wedge\text{P-SAT}(s1,p))\rightarrow(\exists \text{d}\cdot\text{SAT}(s,d)\ \wedge\text{Precedes}(s,s1))$ | A precondition for a plan *p* can be defined as a relation between a situation *s* and a plan *p*, implying that for all plan executions of that plan to occur, a situation should preliminary satisfy some description |
| $\mathbf{Postcondition}(\boldsymbol{p},\boldsymbol{s})\rightarrow\text{Plan}(x)\ \wedge\text{Situation}(s)$<br>$\text{Postcondition}(p,s)\rightarrow\forall s1.\ (\text{PlanExecution}(s1)\ \wedge\text{P-SAT}(s1,p))\rightarrow(\exists \text{d}\cdot\text{SAT}(s,d)\ \wedge\text{Precedes}(s1,s))$ | A postcondition for a plan *p* can be defined as a relation between a situation *s* and a plan *p*, implying that after plan executions of that plan to occur, a situation should satisfy some description |