May 31, 2011
DIA@RomaTRE

Semantic Technologies For Business Process Management

Fabrizio Smith (IASI-CNR, Univaq)
Maurizio Proietti (IASI-CNR)
Michele Missikoff (IASI-CNR)
Outline

(Semantic) Business Process Management
- Introducing Business Process Management
- Business Process Interoperability
- Semantic Approaches for Enterprise Modeling

Reasoning on Semantically Enriched Business Processes
- BPAL Framework
- Querying BPAL Business Process Knowledge Bases

Students’ Project Proposals
(Semantic) Business Process Management
A Business Process (BP) consists of a set of activities that are performed in coordination in an organizational and technical environments. These activities jointly realize a business goal. Each BP is enacted by a single organization, but it may interact with BPs performed by other organizations [1].

A Workflow system automates a BP, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action.

A workflow typically involves:
- Activities
- Decision Points
- Routes (e.g., sequential, parallel)
- Resources
- Roles
Business Process Management (BPM)

BP Management is a collection of methods and techniques to assist business practitioners and employees in the management of business processes along their entire life cycle [2].

• BPM views processes as central in an organization
• Potential for substantial cost & time savings (e.g. business process improvement)
• Managerial and technical ramifications
On the role of models (from [2])
BP Modelling Languages

The Business Process Modelling Notation (BPMN)
- Graphical notation for process modelling
- OMG standard (current version 2.0)
- Not executable, designed for conceptual modelling (semantics loosely defined)
- XPDL 2.0 as XML serialization

Business Process Execution Language (BPEL)
- Derived from IBM’s WSFL and Microsoft’s XLANG
- Define executable Web service orchestrations (XML, SOAP, WSDL)

Yet Another Workflow Language (YAWL)
- Formal Semantics (Petri-nets)
- Comprehensive support for the workflow patterns
BPMN Overview
**Cross-Enterprise Business Processes**

- **Strong acceleration towards new forms of cooperation among enterprises**
  - A networked enterprise integrates the resources of the participating organizations allowing them to pursue shared objectives in a tightly coordinated fashion, operating as a unique (virtual) organization.

- **BP Integration** deals with the definition of a global commonly accepted BP that specifies execution logic and the information exchange between multiple interconnected enterprises to form a *single logical system.*
The Role of SOA

Service Oriented Architecture is a design philosophy where a system is designed to provide services to either end-user applications or other services distributed in a network, via published and discoverable interfaces.

The implementation of SOA through Web Services, allow packaged functionalities to be offered as a suite of “interoperable” services, with interfaces defined independently from the underlying technologies.

Business Processes defined as Service orchestrations

- built from a logically interrelated collection of services, possibly implemented and deployed as web services;
- composite services can be defined by combining existing elementary or composite services, that are in turn offered as high-level services or processes.
The Semantic Interoperability Problem

Service-oriented computing offers a technical means for building cross-enterprise processes.

However, the semantic interoperability problem arises both at a data level and at a process level.

- BPs are built by using different tools, according to different business logics, and using different labels and terminology to denote activities and resources.
- A common view of the business domain should be used…
- …together with descriptions of the local BPs according to such an agreed common view.
Semantic BPM

Promising solutions for the enhancement of BP management tools from the area of the Knowledge Representation to ease:

- interoperability between software applications and reuse/exchange of knowledge between human actors
- automation in the specification, analysis, implementation and monitoring
- same principles at the basis of the Semantic Web vision
- see, e.g., the SUPER (http://www.ip-super.org/), COIN (http://www.coin-ip.eu/) and PLUG-IT (http://plug-it.org/) European projects.

Shared vocabularies (aka Ontologies) to describe processes

- Input / output
- Involved resources (data models, organizational models)
- Goal
- Pre-conditions and effects
- Non-functional information (e.g., QoS)
On the role of models…again

"world"
- business processes
- people
- organizations
- machines
- components

Ontologies
- Formalizes (shared) Conceptualizations

(models analyses)
- (process) model

information system
- supports/controls

-specifies configures implements analyses

records events, e.g. messages, transactions, etc.

 discovers

check conformance
What is an Ontology?

 Ontology: the philosophical discipline
  • Study of *what there is*
  • Study of the *nature* and *structure* of “reality”

   **A (philosophical) ontology is** a structured system of entities assumed to exists, organized in categories and relations.

 Computational ontologies:

   Specific (theoretical or computational) artifacts expressing the *intended meaning* of a *vocabulary* in terms of *primitive* categories and relations describing the *nature* and *structure* of a *domain of discourse*

   ...in order to account for the competent use of vocabulary in real situations!

   Gruber: “Explicit and formal specifications of a conceptualization”
Ontology Languages in the Semantic Web

- **RDF**
  - Generic mechanism for describing resources
  - Data Model based on *statements* in the form of triples \(<subject,predicate,object>\)

- **RDFS**
  - An RDF vocabulary to describe other RDF vocabularies
  - Introduces hierarchies and domain and range restrictions on properties

- **OWL**
  - Ideally, an extension of RDFS with new modeling primitives (strong relationship with Description Logics)
  - Several sub-language. Among them, **OWL-RL:**
    - subset designed for practical implementations using rule-based techniques
    - semantics is an upward-compatible extension of RDF and RDFS
    - axiomatisation of the OWL 2 RDF-Based Semantics (OWL 2 RL/RDF rules) in the form of first-order implications

\[
<?o,rdf:type,?x>, <?x,rdfs:subClassOf,?z> \rightarrow <?o,rdf:type,?z>
\]
Ontologies… for what?

Reasoning with BPs through a suitable semantic model used for the annotation of BP Repositories

- **Query** & retrieval
- Similarity reasoning
- Composition
- …

Semantic Model

Global BP

LBPR_1

... 

LBPR_x

\[ \Sigma_1 \]

\[ \Sigma_x \]
Reasoning on Semantically Enriched Business Processes
Motivations and Goal

Several open issues regarding

- the synergy between workflow languages and ontologies:
  - execution semantics (enactment, correctness of BP executions,…)
  - verification (deadlocks, livelocks, …)
  - taxonomic reasoning
  - …
- the accomplishment of reasoning services involving both these components:
  - business rules (modeling and compliance verification)
  - querying & retrieval (reuse, composition, …)
  - …

Our goal is to provide a uniform and formal representation of both the workflow-related and the ontology-related (domain) knowledge.
BPAL Framework

BPAL (Business Process Abstract Language)

- A declarative modeling method to capture the procedural knowledge of a BP
- Based on widely accepted modeling languages (i.e., BPMN)
- Provided with an explicit formalization of the meta-model and execution semantics (suited for automated reasoning)

Semantic Enrichment through Reference Ontologies

- OWL-RL support (more or less Description Logic Programs)

Formalization in Logic Programming

- Uniform and formal representation
- Support by tools developed in the area of logic programming
- A BPAL specification (meta-model, schema and trace semantics) can be queried by any Prolog system (no need for ad-hoc compiler)
Business Process Knowledge Base

Business Reference Ontology

Semantic Annotation

BPAL Meta-Model

sequence_flow
source
target
flow_element
gateway
activity
event

BPAL BP Schemas

S1
S2
S3
S4

Execution Traces

s1
s2
s3
BPAL

BPAL [3] is a logic-based language (i.e., a set of predicates and constants) conceived to provide a declarative modeling method capable of fully capturing procedural knowledge in a business process.

- its core is based on BPMN 2.0 specification

A BPAL business process schema (BPS) is specified by a set of ground facts of the form $p(C_1, ..., C_n)$, where $C_1, ..., C_n$ are constants denoting BPS elements (e.g., business activities, events, and gateways) and $p$ is a BPAL predicate.
BPAL Overview

Input(RejectPO, ClientDetails)

Flow(RejectPO, G4)

Participant(Supplier)
Assigned(ReceivePO, Supplier)

Item(PO_items)
Item(PO)

Exc_branch(G3)
Par_branch(G5)

Activity(ShipRequest)
Activity(BillClient)

End_event(E1)
BPAL Meta-model

The BPAL Meta-model theory ($M$) formalizes a set of structural properties of a BPS, that at this level is regarded as a labeled graph:

- **Well-formedness**, defining how the constructs provided by the BPAL language can be used to build BPS correct from a syntactical point of view:
  - *local properties*, e.g., every activity must have at most one incoming and at most one outgoing sequence flow;
  - *global properties*, e.g., every flow element must lie on a path from the *start* to the *end*.

- **Structuredness**, of a (sub-)process:
  - each branch point is matched with a merge point of the same type, and such branch-merge pairs are also properly nested.

- **Reachability** between nodes;

- **Containment** of an element within a (sub-process);

- …
Execution Semantics

A process trace represents a possible execution of a business process, reflecting a sequence of token-transitions:

- Start(FulfillPO)
- Complete(ReceivePO)
- Complete(CheckInventory)
- Choice(G3,ShipRequest)
- Complete(ShipRequest)
- Fork(ShipRequest,BillClient)
- Complete(ShipRequest)
- Complete(BillClient)
- Synch(ShipRequest,BillClient)
- Merge(G6)
- Complete(ClosePO)
- End(FulfillPO)
The execution semantics of a BPS is formalized by the theory $T$, in terms of state transitions:

$$\text{trans}(S, \text{complete}(A), Z) \leftarrow \text{activity}(A) \land \text{holds}(\text{token}(A1,A), S) \land \text{flow}(A,B) \land Z = (S - \{\text{token}(A1,A)\}) \cup \text{token}(A,B).$$

- $TR$ is trace from $S$ to $Z$ if $\text{trans}(S, TR_1, Z_1) \land \text{trans}(Z_1, TR_2, Z_2) \land \ldots \land \text{trans}(Z_n, TR_{n+1}, Z)$ and $TR = [TR_1, TR_2, \ldots, TR_{n+1}]$.

$T$ can be used to:

- verify the correctness of a trace w.r.t. a given BPS
- generate the set of correct traces given a BPS
- verify ordering properties (i.e., dependency constraints)
Behavioral Properties

In theory

• Exploration of the state space
• Model Checking (CTL)
• Very Expensive

In practice

• Some relevant properties can be verified with ad-hoc strategies, without state space exploration
  • \textit{precedence}(a,b) if b is executed then a has been previously executed
  • \textit{response}(a,b) if a is executed then b will be executed after
  • \textit{mutex}(a,b) a and b are never both executed
  • \textit{coexistence}(a,b) neither a nor b are executed, or they are both executed
Semantic Annotation

BP Schema

Business Reference Ontology (BRO)
Annotated Business Process

```xml
<rdf:Description rdf:about="bps:ReceivePO">
  <rdf:type rdf:resource="bpal:Activity"/>
  <bpal:input rdf:resource="bps:PO"/>
  <bpal:model_ref>http://acme/ACME.xpdl#_123</bpal:model_ref>
  ....
</rdf:Description>

<rdf:Description rdf:about="bps:BillClient">
  <rdf:type rdf:resource="bpal:Activity"/>
  <bpal:sigma rdf:resource="bro:Invoicing"/>
  <bpal:assigned rdf:resource="bps:Supplier"/>
  ....
</rdf:Description>

<rdf:Description rdf:about="bps:PO">
  <rdf:type rdf:resource="bpal:Item"/>
  <bpal:sigma rdf:resource="bro:PurchaseOrder"/>
  ....
</rdf:Description>

<rdf:Description rdf:about="bps:Supplier">
  <rdf:type rdf:resource="bpal:Participant"/>
  ....
</rdf:Description>
```
The BPKB [4] can be translated to a Logic Program to model and reason with:

- **process structure** (BPAL schema + Meta-model): flow elements (activities, events, gateways) and their relationships (sequence flows);
- **process executions** (BPAL schema + Trace semantics): behavior at execution time and properties regarding the temporal sequencing of activities in the set of correct traces;
- **domain knowledge** (Ontology + Semantic Annotation): domain entities (processes, actors, object) and their relationships.

The BPAL framework provides:

- a **reasoning engine**, based on Logic Programming, that operates on a BPKB;
- a **BP query language**, developed on top of the reasoning engine;
- a **verification mechanism**, tightly connected to the latter.
BPAL Reasoner

Commercial BPM Tool
Ontology Management System

XPDL2BPAL
OWL2LP

QuBPAL2Prolog

XPDLWriter

BPKB

M
Meta-model
OWLRL
OWL rule-set
T
Trace theory

B
BP Schema
BRO \cup \Sigma

p = ‘eProcurement’
s = ‘ReceivingPO’
e = ‘SendingGoods’
QuBPAL

A SELECT-FROM-WHERE query language for the BPKB
- translation into logic programs, nested and disjunctive queries

SELECT statement: output of the query evaluation
- a sequence $?x*$ of variables occurring in the WHERE statement
- a BPS, denoted by the process identifier $<?bpld>$
- a sub-process of a BPS, denoted by the triple $<?bpld,?start,?end>$

FROM statement: the process(es) from which data is to be retrieved
- a non empty sequence of BP schemas or sub-processes ($<bpld> | <bpld,start,end>)*$

WHERE statement: a sentence built from
- the set of the predicates defined in the BPKB;
- the connectives AND, OR, NOT, and the predicate $=$ with the standard logic semantics;
- another QuBPAL query, to allow nested queries.

For the arguments used in the query, $bpsEl::conc$ stands for “$bpsEl$ is annotated with $conc$”
Query Example

Retrieve any process where a purchase order is processed and that provides services for the invoicing and the delivering of goods.

```
SELECT ?p
FROM *
WHERE belongs(?a1,?p) AND input(?a1,?po::bro:PurchaseOrder) AND belongs(?a2::bro:Invoicing,?p) AND belongs(?a3::bro:Delivering,?p)
```
Query Example

Retrieve any sub-process that i) starts with an activity that processes a purchase order, and ii) both a delivering and an invoicing are eventually executed.

```
SELECT <p,s,e>
FROM *
WHERE input(s,po::bro:PurchaseOrder) AND response(s,a1::bro:Invoicing,p,s,e) AND response(s,a2::bro:Delivering,p,s,e)
```
Students’ Project Proposals

- Development of an Eclipse Plug-in for the management of the BPKB
  - Current work-in-progress

- Similarity Reasoning
  - Automatic support for annotation
  - Similarity measures for processes
  - Adopt/adapt ontology matching techniques

- Query processing / optimization
  - Currently a simple algorithm for the re-ordering of literals is implemented

- Graphical editor for QuBPAL queries

- Support for other standard BP modeling languages (e.g., BPEL)
  - Mapping from/to BPAL
References

Thanks for your attention, any question?