SECONDA UNIVERSITÀ DEGLI STUDI DI NAPOLI

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI
INGEGNERIA INDUSTRIALE E
DELL'INFORMAZIONE

A Semantic Model for Business Process Patterns to support Cloud Deployment

Beniamino Di Martino
Antonio Esposito
Stefania Nacchia
Salvatore Augusto Maisto

10th Symposium and Summer School On Service-Oriented Computing,
June 27 – July 1, 2016 Crete, Greece

Cloud Computing

- Shift to new Computing paradigms
  - Service Oriented Architecture
  - Grid Computing
  - **Cloud Computing**

- Cloud Computing **Opportunities** ...
  - Reduced up-front investment and maintenance costs
  - Better use of existing hardware
  - Enhanced resiliency and disaster recovery
- … and **challenges**
  - High variety of incompatible interfaces and data formats
  - Operations’ signatures vary while Parameters’ semantics is the same
  - Portability (to and across Cloud platforms)
  - Interoperability (also hybrid scenarios)

Business Processes

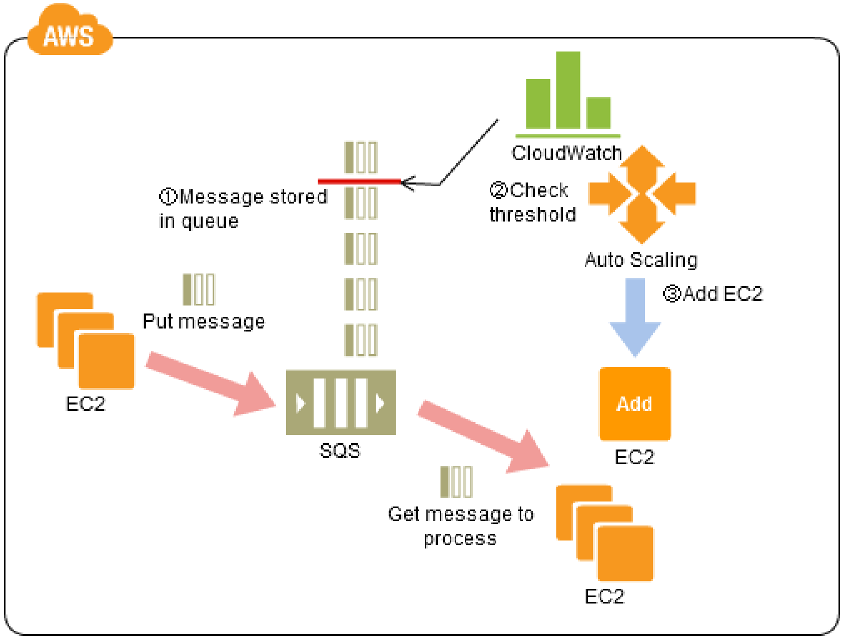
*"A **business process** consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal."*

- Great interest in Business Processes standardization
  - Homogenize formats used among and within companies
  - Focus on Business Process Model Notation (**BPMN**)
- A huge number of tools have been developed for BPMN
  - Camunda, Activiti, Bonita or jBPMN
- Different representations of the same Business Process are possible
  - Different semantics applied
  - Different design views of the same process

Why semantics?

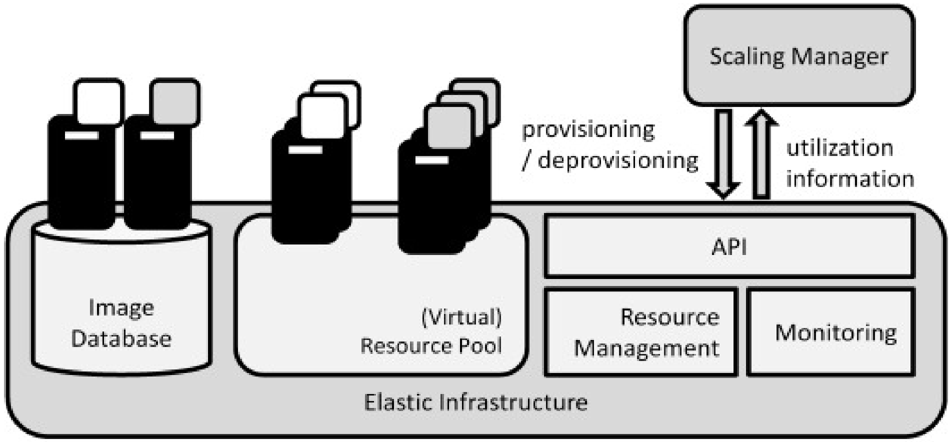
- **Roles compatibility**
- **Inspection/Process soundness**
- **Optimization/Process re-engineering**
- **Process re-use**
- **Support to Process Deployment**
  - Use cloud services to implement tasks
- **Enable model transformation**

Cloud Patterns



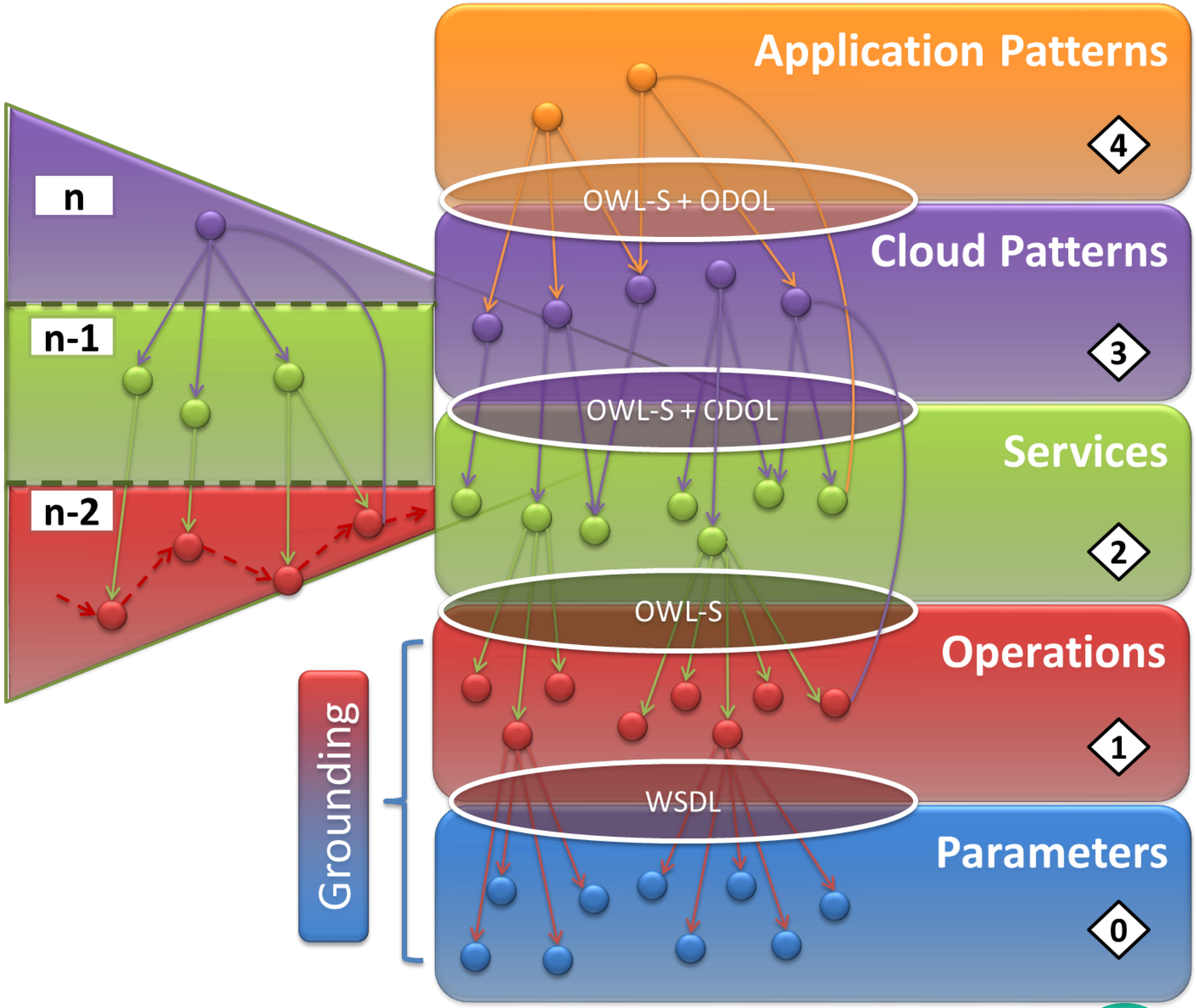
- **Published by Cloud Vendors**
  - More specific
  - Strongly tied to the reference Cloud Platform

- **Developed in academic environment**
  - High abstraction level
  - High flexibility

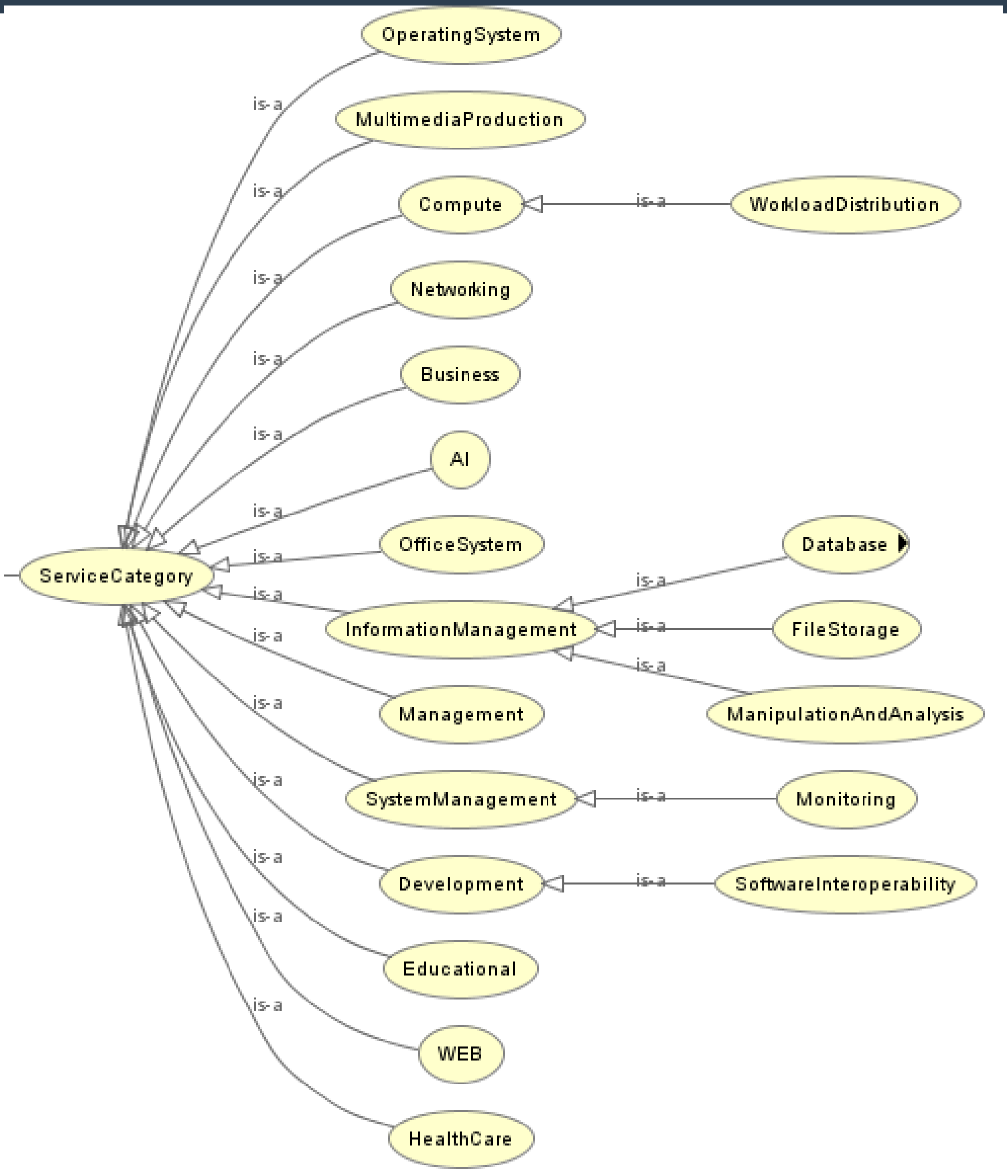


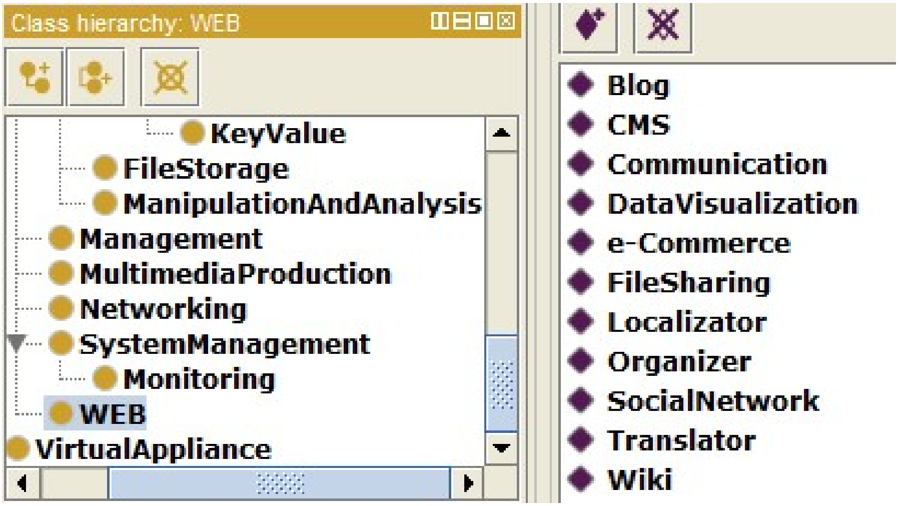
Cloud Patterns: a semantic representation

- Both agnostic and vendor specific
- Several abstraction layers:
  - Application and Cloud patterns for high level abstraction
  - Services and Operations level for generic services and related methods
  - Parameters level for lowel abstraction



Service Ontology for Cloud



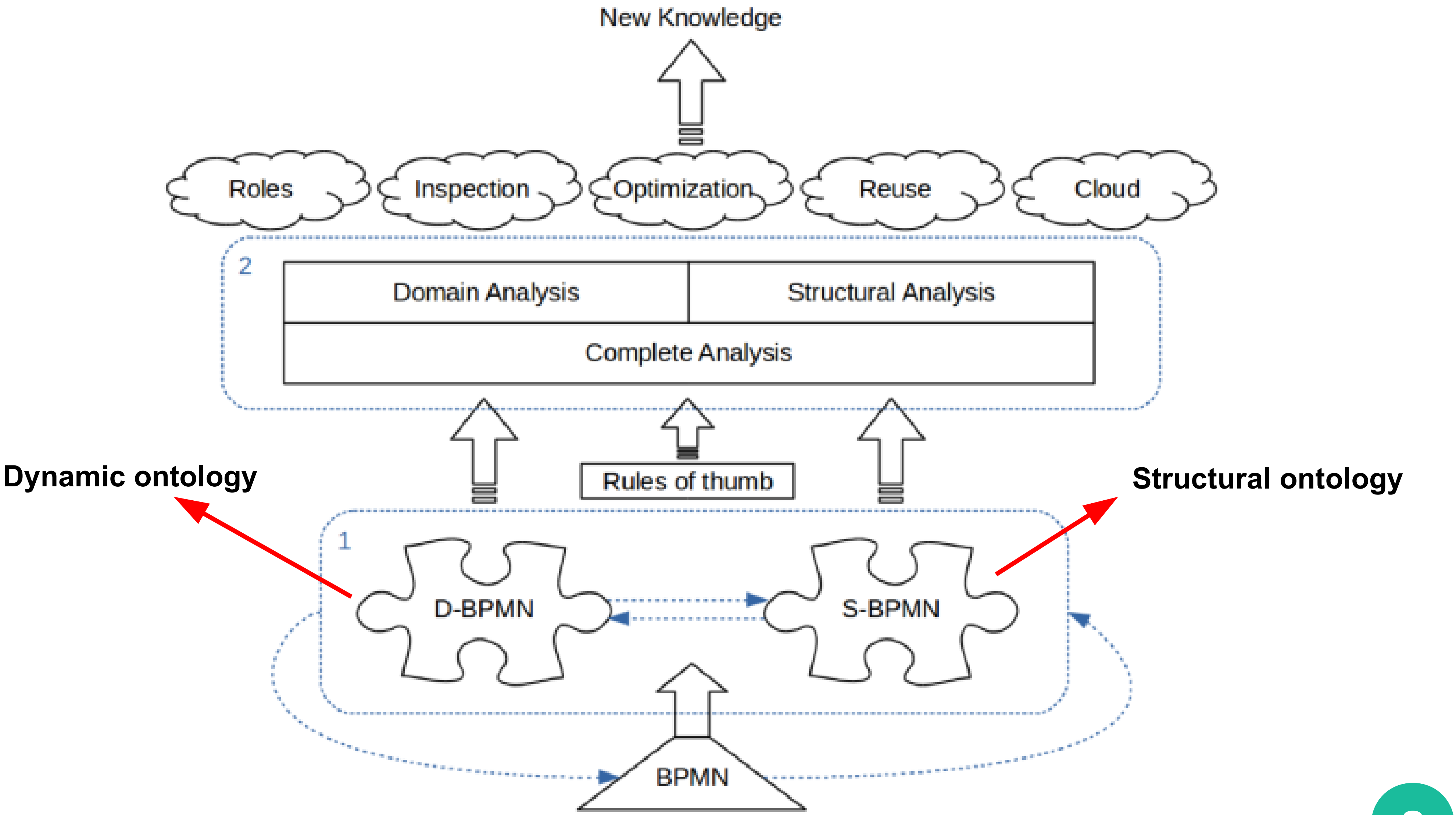

Class hierarchy: WEB

- KeyValue
- FileStorage
- ManipulationAndAnalysis
- Management
- MultimediaProduction
- Networking
- SystemManagement
  - Monitoring
- WEB
- VirtualAppliance

- Blog
- CMS
- Communication
- DataVisualization
- e-Commerce
- FileSharing
- Localizator
- Organizer
- SocialNetwork
- Translator
- Wiki

- ➢ Categorize tasks and actors
- ➢ Allow to identify similar tasks and sub-tasks
- ➢ Impose conditions and constraints on the involved entities

How to represent BPMN?



Dynamic ontology

Structural ontology

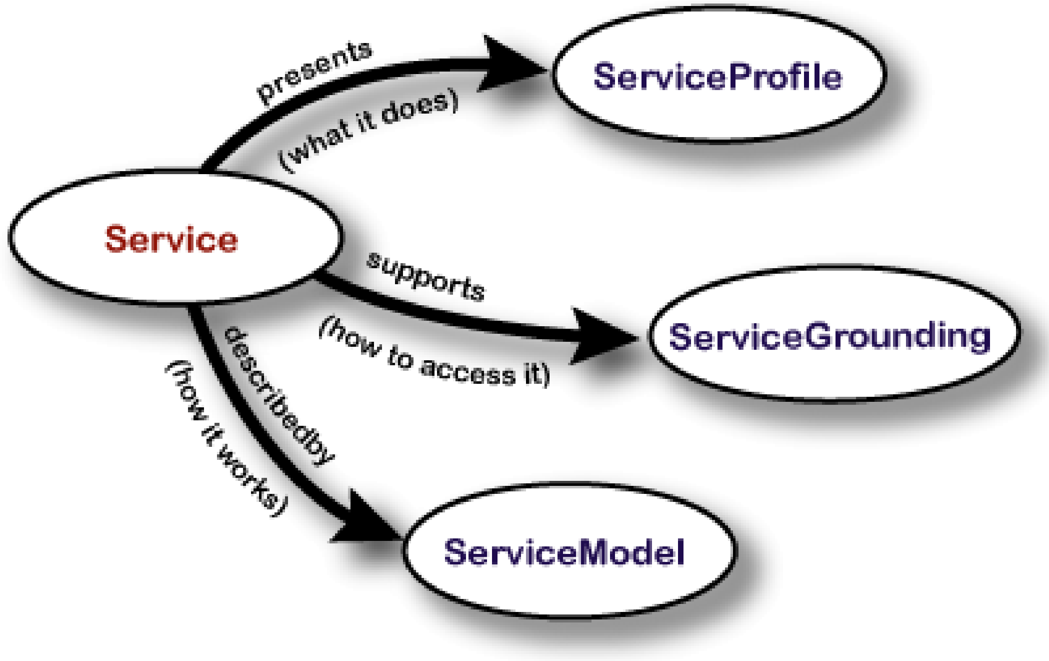
Our current approach

- No structural ontology
  - Map elements of the diagram to OWL-S concepts
  - Stress relations to a workflow meta-model

- Recognize Business Patterns in the semantic model
  - A repository of such patterns is being built
  - The same OWL-S representation is used for patterns

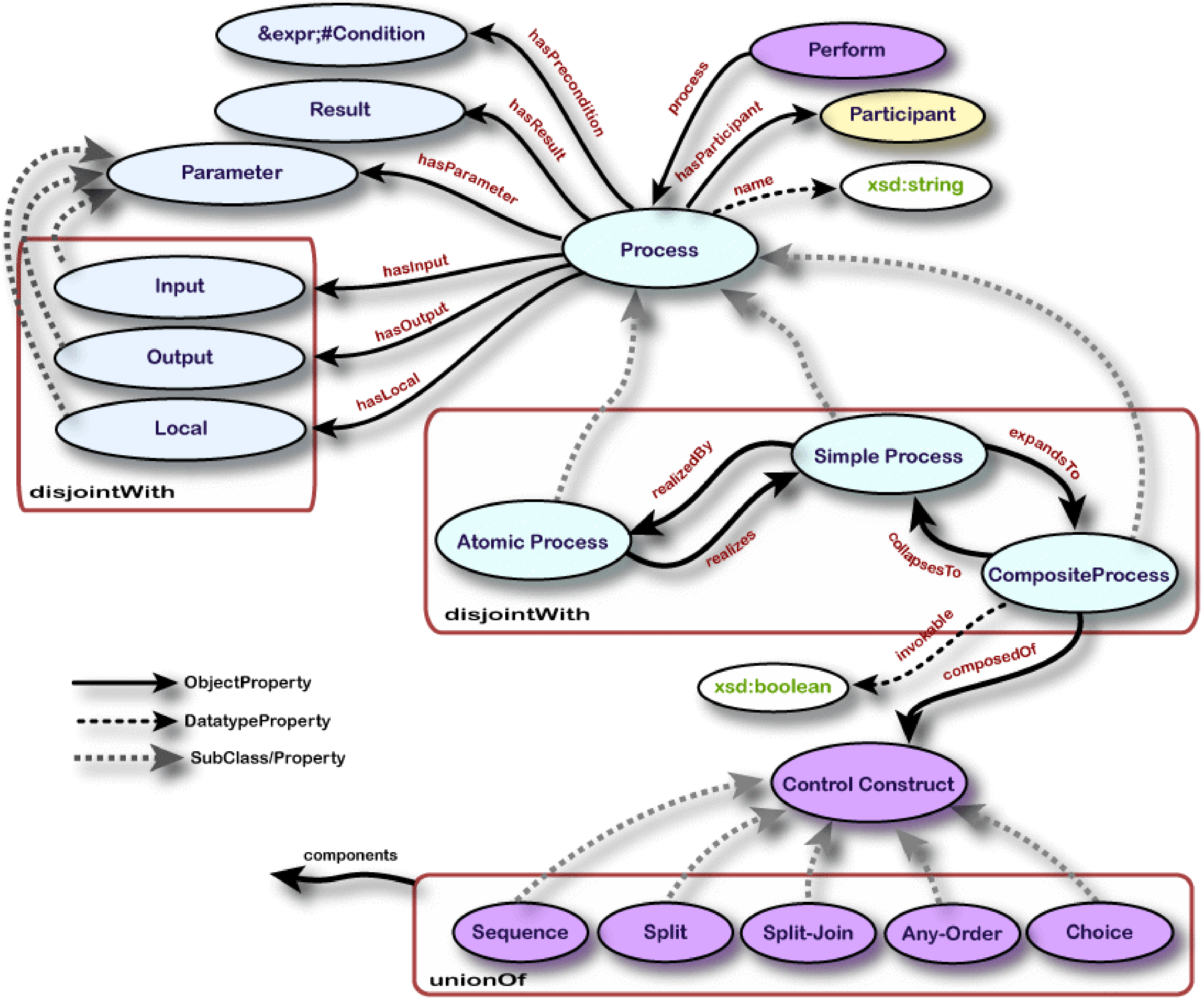
- Use the mapping between Business and Cloud Patterns to understand which services to deploy
  - Suggest suitable services with their operations\parameters
  - Mapping is currently hand-made

The OWL-S model

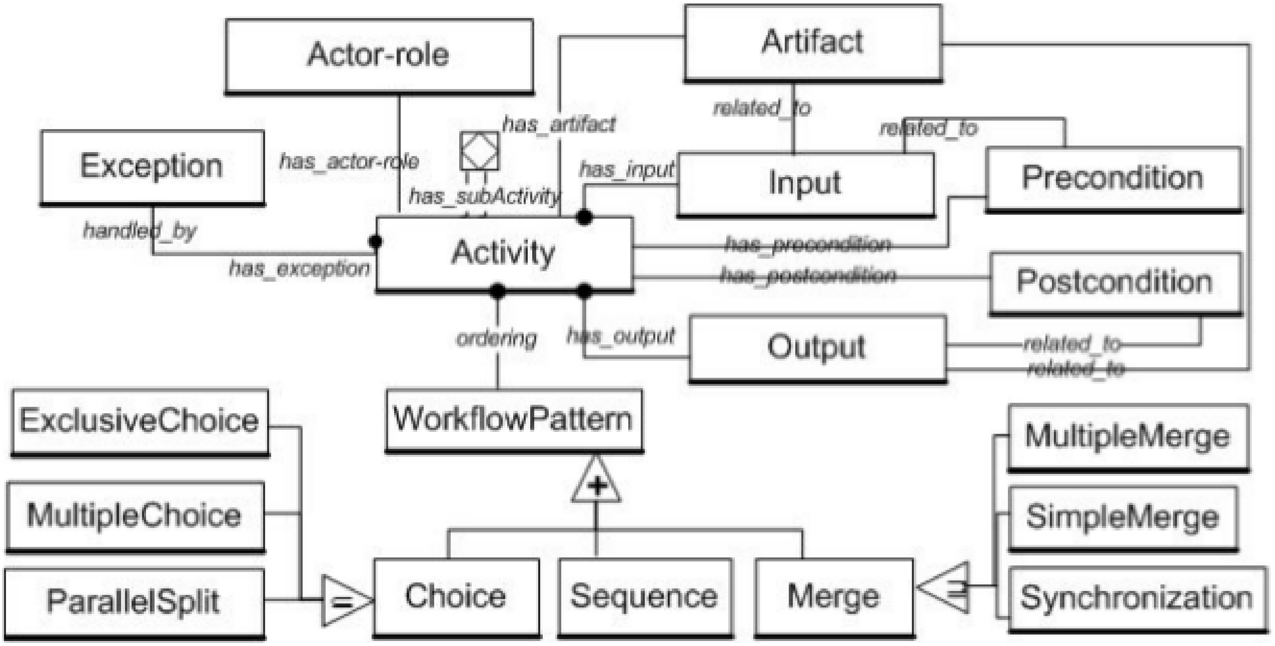


- Description of services' inputs and outputs
  - Definitions of contraints
- Connections to a WSDL definition
- Workflow definition of a service process

OWL-S Workflow



The adopted Meta-Model



Y. Lin and H. Ding. Ontology-based semantic annotation for semantic interoperability of process models.

Activity is a synonym of a Process, and can be atomic or composed by other activities.

Artifact is something involved in an Activity such as a tool or a software.

Actor-role represents the entity which interacts with or performs an AcTivity

Input and **Output** define the simple information needed by an Activity or produced by it.

Preconditions and **Postconditions** describe general representations of constraints.

Exception define not-ordinary situations which need to be addressed by ad-hoc Activities. **Workflow Patterns** represent orderings of different Activities.

Mapping of a BPMN to OWL-S

BPMN Element	OWL-S Concept
Task	Atomic Process
Task + Subtask	Composite Process
Series of two or more tasks	Sequence
Parallel tasks	Split/Split and Join
Gates	HasCondition → Condition
Start Event	HasCondition → Condition
End Event	HasEffect → Effect
Lane	HasClient

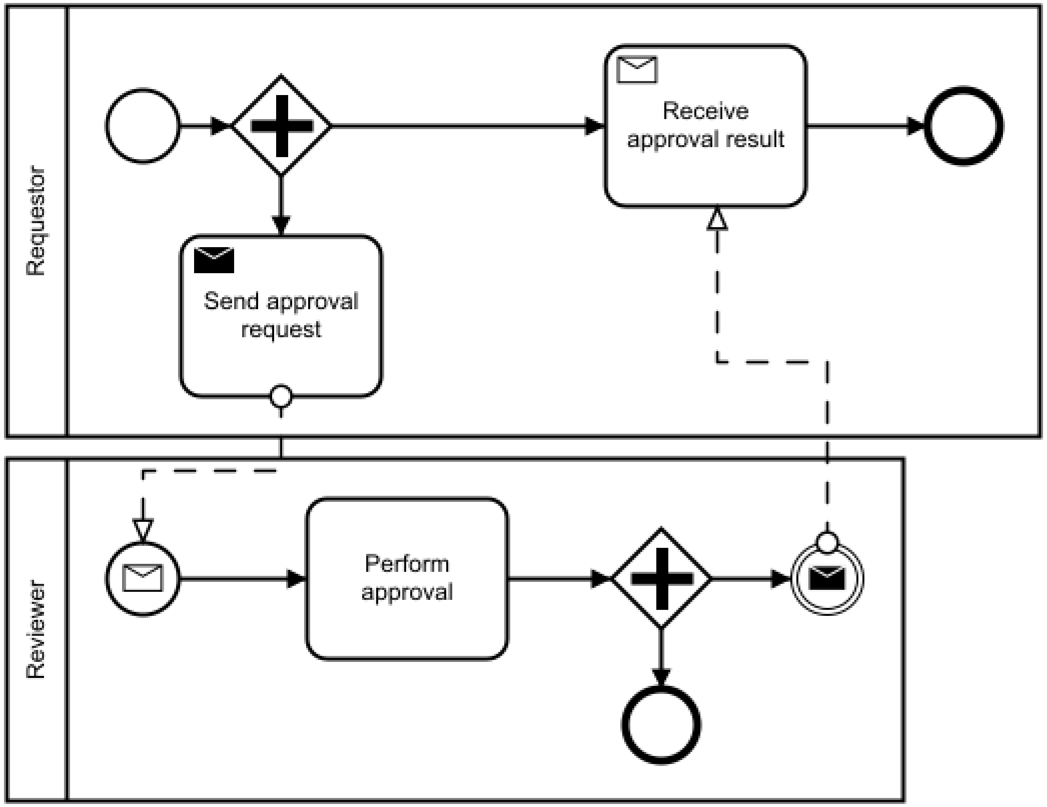
Mapping of a BPMN to OWL-S

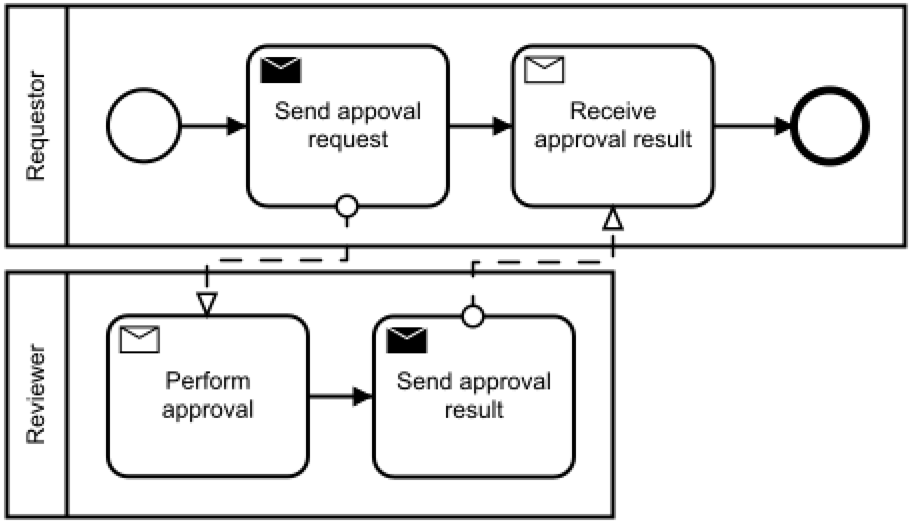
```
1.Create a process for each of the tasks described in the model.
    IF (task executed after gate AND gate has condition) THEN
        add pre-condition to the process
REPEAT
2.  Add consecutive tasks to Sequence structure
3. Include tasks connected to the outgoing flows of a gate in a split structure.
    IF (task is the head of a sequence) THEN
        the entire sequence becomes part of the split structure.
UNTIL All tasks have been included AND all gates have been evaluated
```

4. Start and end events with conditions and effects are eliminated, as conditions and events are associated to the first/last task of the process.

5. Intermediate events' conditions and effects are associated to the previous/next tasks.

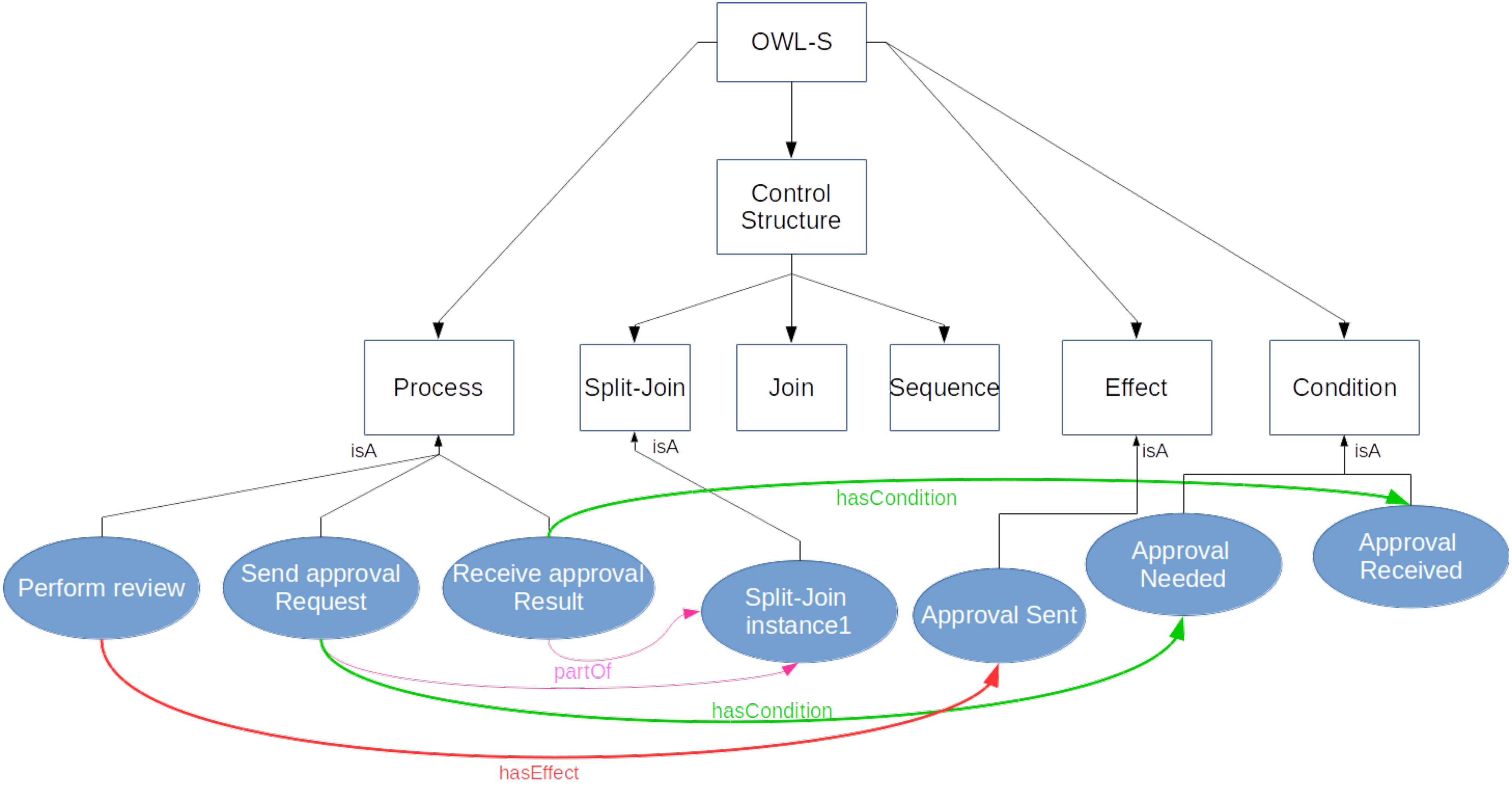
The case study: the Approval pattern



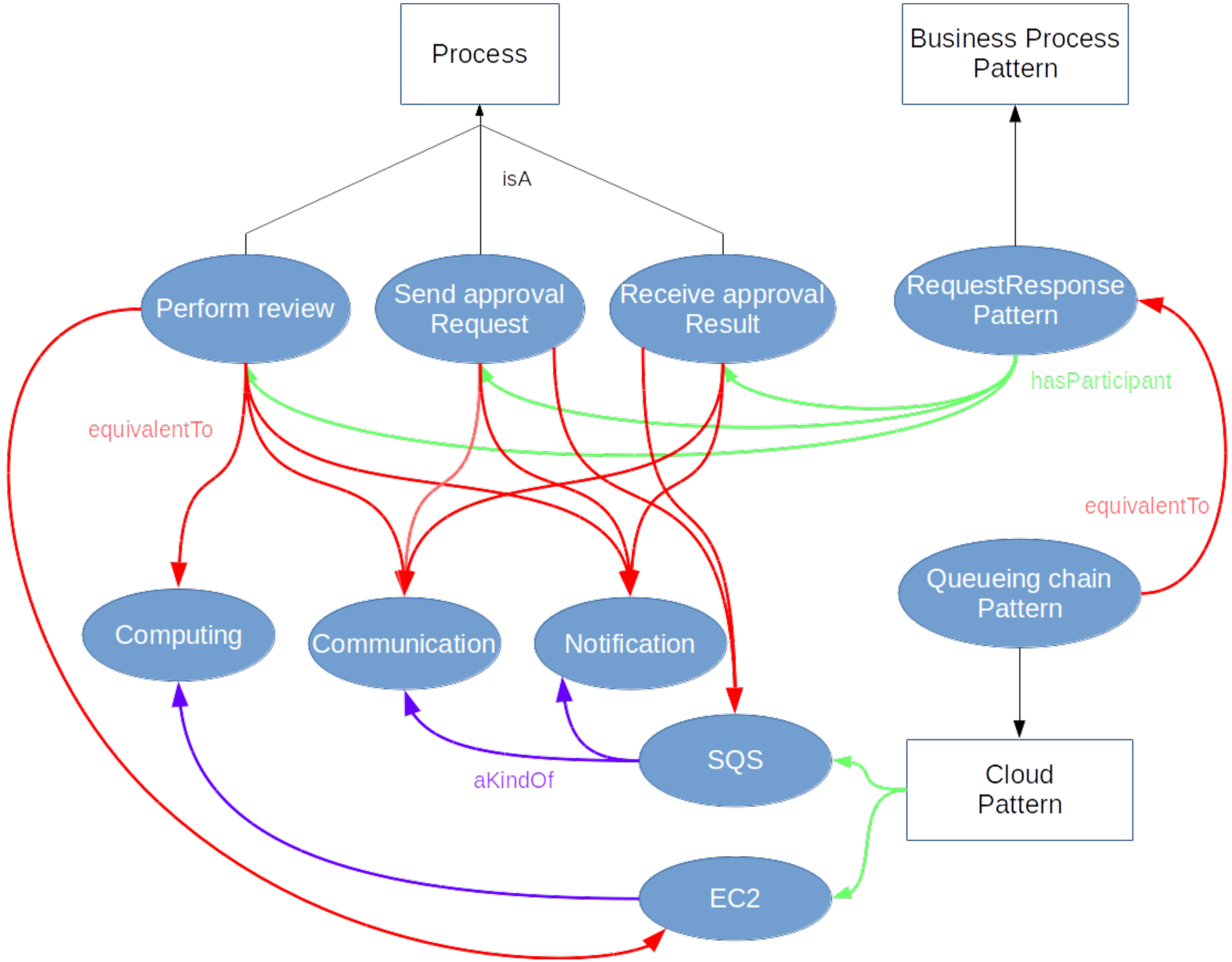


Lucinéia Heloisa Thom, Manfred Reichert, Cirano Iochpe
Activity patterns in process-aware information systems: basic concepts and empirical evidence.

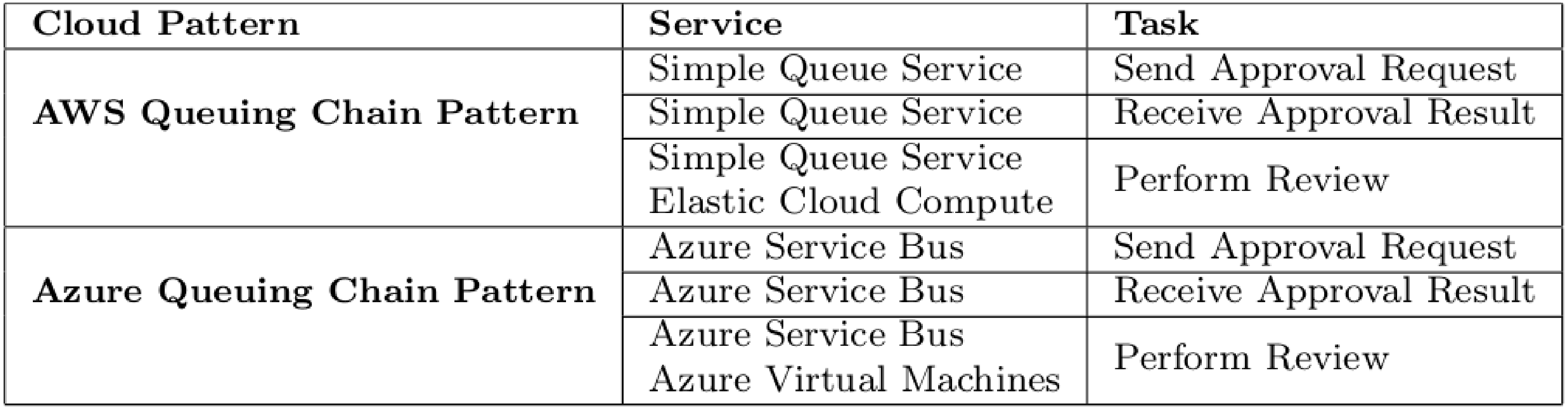
Description of the Process in OWL-S



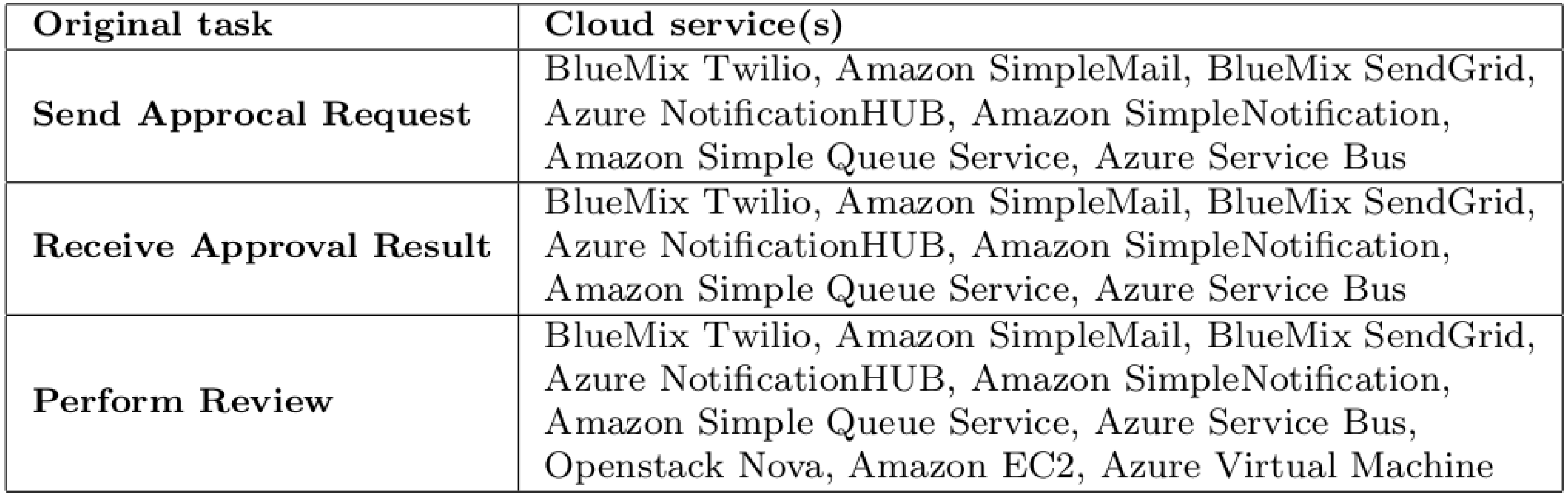
Connection to Patterns



Available services

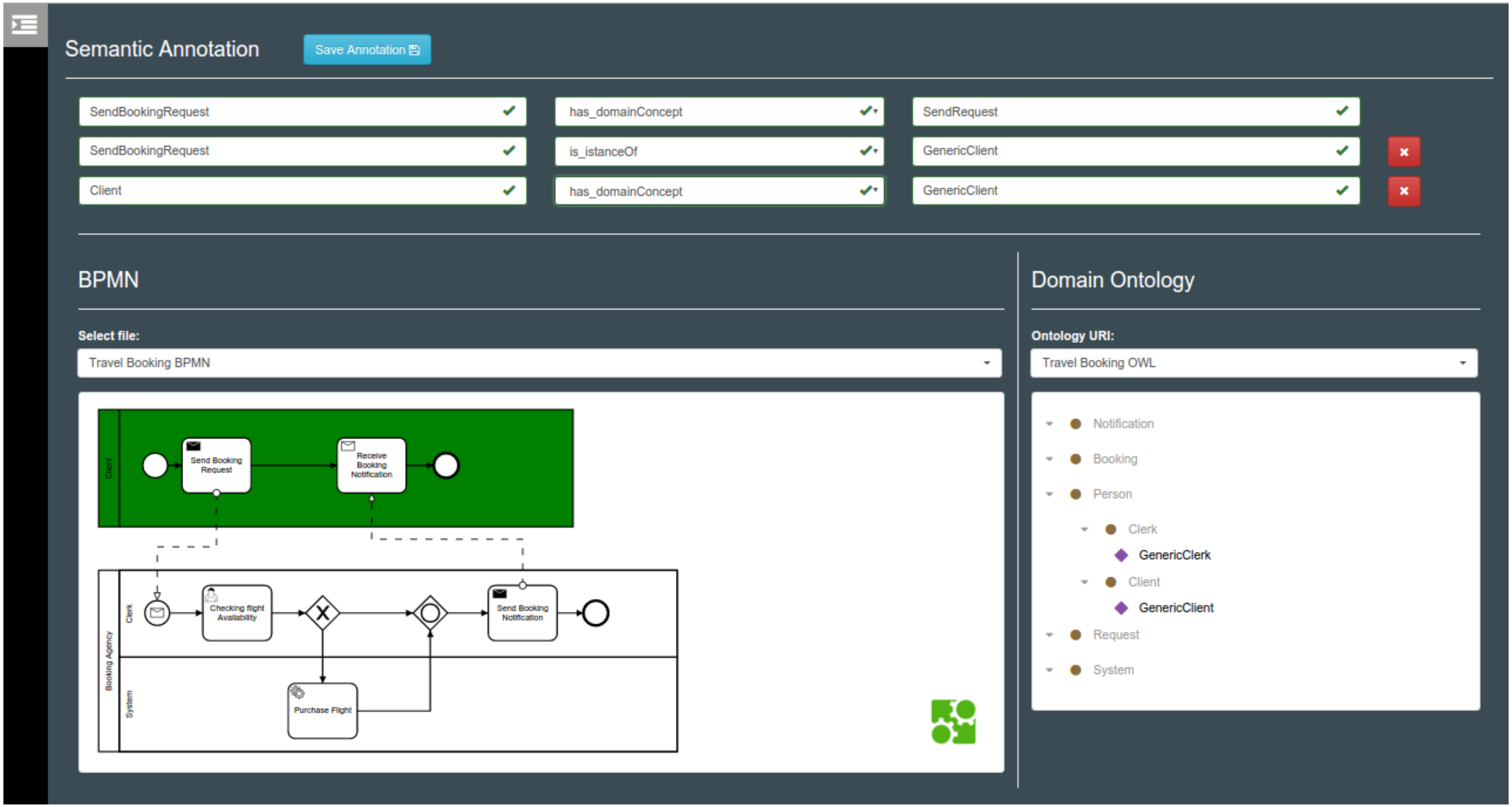


Cloud Pattern	Service	Task
AWS Queuing Chain Pattern	Simple Queue Service	Send Approval Request
	Simple Queue Service	Receive Approval Result
	Simple Queue Service Elastic Cloud Compute	Perform Review
Azure Queuing Chain Pattern	Azure Service Bus	Send Approval Request
	Azure Service Bus	Receive Approval Result
	Azure Service Bus Azure Virtual Machines	Perform Review



Original task	Cloud service(s)
Send Approcal Request	BlueMix Twilio, Amazon SimpleMail, BlueMix SendGrid, Azure NotificationHUB, Amazon SimpleNotification, Amazon Simple Queue Service, Azure Service Bus
Receive Approval Result	BlueMix Twilio, Amazon SimpleMail, BlueMix SendGrid, Azure NotificationHUB, Amazon SimpleNotification, Amazon Simple Queue Service, Azure Service Bus
Perform Review	BlueMix Twilio, Amazon SimpleMail, BlueMix SendGrid, Azure NotificationHUB, Amazon SimpleNotification, Amazon Simple Queue Service, Azure Service Bus, Openstack Nova, Amazon EC2, Azure Virtual Machine

The prototype tool interface



Semantic Annotation

SendBookingRequest ✓	has_domainConcept ✓	SendRequest ✓
SendBookingRequest ✓	is_istanceOf ✓	GenericClient ✓
Client ✓	has_domainConcept ✓	GenericClient ✓

BPMN

Select file: Travel Booking BPMN

Client
Send Booking Request
Receive Booking Notification
Booking Agency
Clerk
Checking flight Availability
Send Booking Notification
System
Purchase Flight

Domain Ontology

Ontology URI: Travel Booking OWL

- Notification
- Booking
- Person
  - Clerk
    - GenericClerk
  - Client
    - GenericClient
- Request
- System

Conclusion and Future work

- Owl-S based representation of BMPN
  - Allows for integration with our existing pattern ontology
  - Cloud patterns contains the information necessary to deploy the service
    - Skeletons and stubs can be produced [1][2], but more work needs to be done
- A prototype tool for loading and annotating BPMN files has been built
  - Needs support for pattern recognition
- The BPMN can be further annotated
  - Domain concepts can be added by users
- Need to work on the pattern recognition procedure
  - Past experience in design pattern recognition from source code and UML diagrams [2][3]

[1] Di Martino, Esposito, Cretella. Semantic Representation and Reasoning of Cloud Patterns and Services to support Cloud Portability, in IEEE Transactions on Cloud Computing. 2015
[2] Di Martino, Esposito. Automatic Dynamic Data Structures Recognition to Support the Migration of Applications to the Cloud. In International Journal of Grid and High Performance Computing (IJGHPC). 2015
[3] Di Martino, Esposito. Automatic Recognition of Design Patterns from UML-based Software Documentation. In Software Practice and Experience. 2013

Thanks for your attention