Supporting Software Services Discovery and Sharing in Collaborative Networks

Alexandre Perin-Souza; Ricardo J. Rabelo

GSIGMA - Intelligent Manufacturing Systems Group
DAS - Department of Automation and Systems
UFSC - Federal University of Santa Catarina
Brazil

Table of Contents

- Introduction & Motivation
- Research Problem & Objective
- Basic Literature Overview
- Proposed Model
- Prototype Implementation & Preliminary Results
- Final Considerations
- Next Steps

Introduction

Collaborative Networks (CN) paradigm relies on the existence of three essential pre-conditions:

- *Collaboration*: among involved partners.
- *Trust*: partners shall trust to one another.
- *Digital transactions*: (most of the) activities carried out via computer networks.

In essence, all this aims at enabling organizations to *agilely and adaptively define, set up relations and operate with other organizations*.

Introduction

Supporting this requires several issues to cope with in businesses and collaborations, such as:

- Business processes effectiveness;
- Interoperation;
- Value added perception;
- Trustworthiness;
- Cooperation;
- adequate Business and Legal Models;
- People empowerment;
- …
- and *Flexible and Advanced ICT infrastructures*.

Prominent approaches:

SOA (Service Oriented Architecture) ,

Utility ,

Cloud Computing ,

and *SaaS* (Software as a Service)

paradigms.

Flexible and Advanced ICT Infrastructures

- SOA: design model where systems' functions are viewed as small, distributed and loose-coupled / composable software modules called *software **services***.
- Utility: principle that a software service should be **discoverable and usable** anytime, anywhere, like electricity.
- Cloud computing: resources (e.g. software services) should be **promptly available** in a computing "cloud" accessed via Internet.
- SaaS: architectural and business models where services are accessed **on-demand and paid-per-used** through the Internet, in most of the cases as a strategy to **outsource** the TI area.

Flexible and Advanced ICT Infrastructures

These approaches & technologies are a clear trend (in several cases they are a reality already) and they have the potential to support some perspectives of agility in a CNO whereas decrease ICT costs.

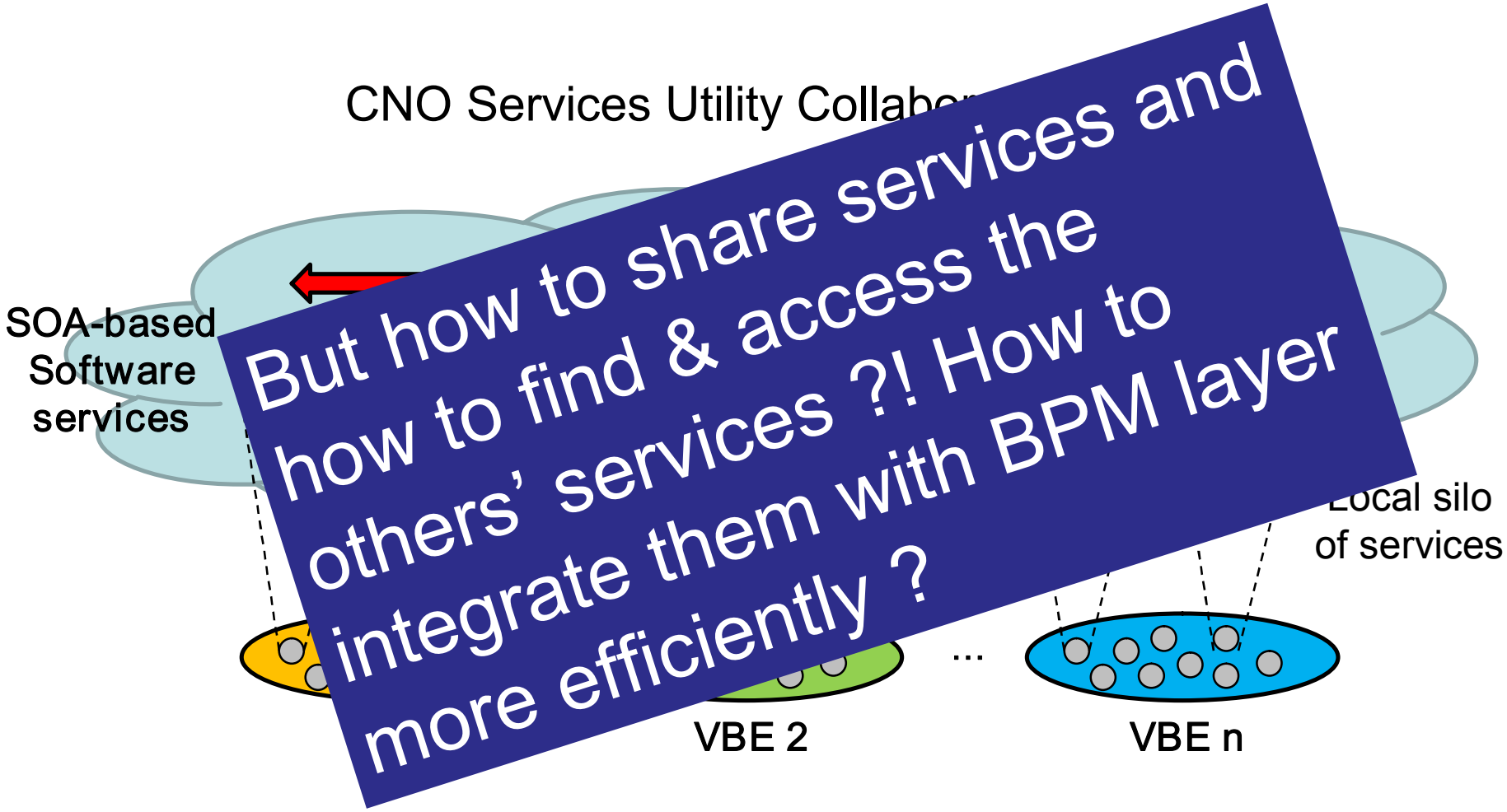
Most of CNO members are composed of MSMEs (Micro, Small and Medium size Enterprises), without much conditions to maintain ICT infrastructures and costly staff.

Evidences and Motivation

- Several companies **already use** SOA-based solutions and this will increase;
- Several companies develop their own software services but keep them in their **isolated silos**;
- Several (and increasingly) applications are nowadays **already offered** under the SaaS model;
- Most of SOA implementations are **poorly connected** with the Business Process management layer.

Besides sharing business, knowledge, practices, training, etc., ***why do not CNO members share software services too*** if they are a CNO ?

Exploratory Research Scenario



Objective

In this wider and advanced context, this exploratory work tackles one of the involved problems, which is the …

service discovery.

Services Discovery

In general, the problem of (software) **services discovery can be defined as the activity of finding and selecting services**. It presumes the existence of (local or largely distributed) repositories (where services are registered and deployed) over which the search will be made.

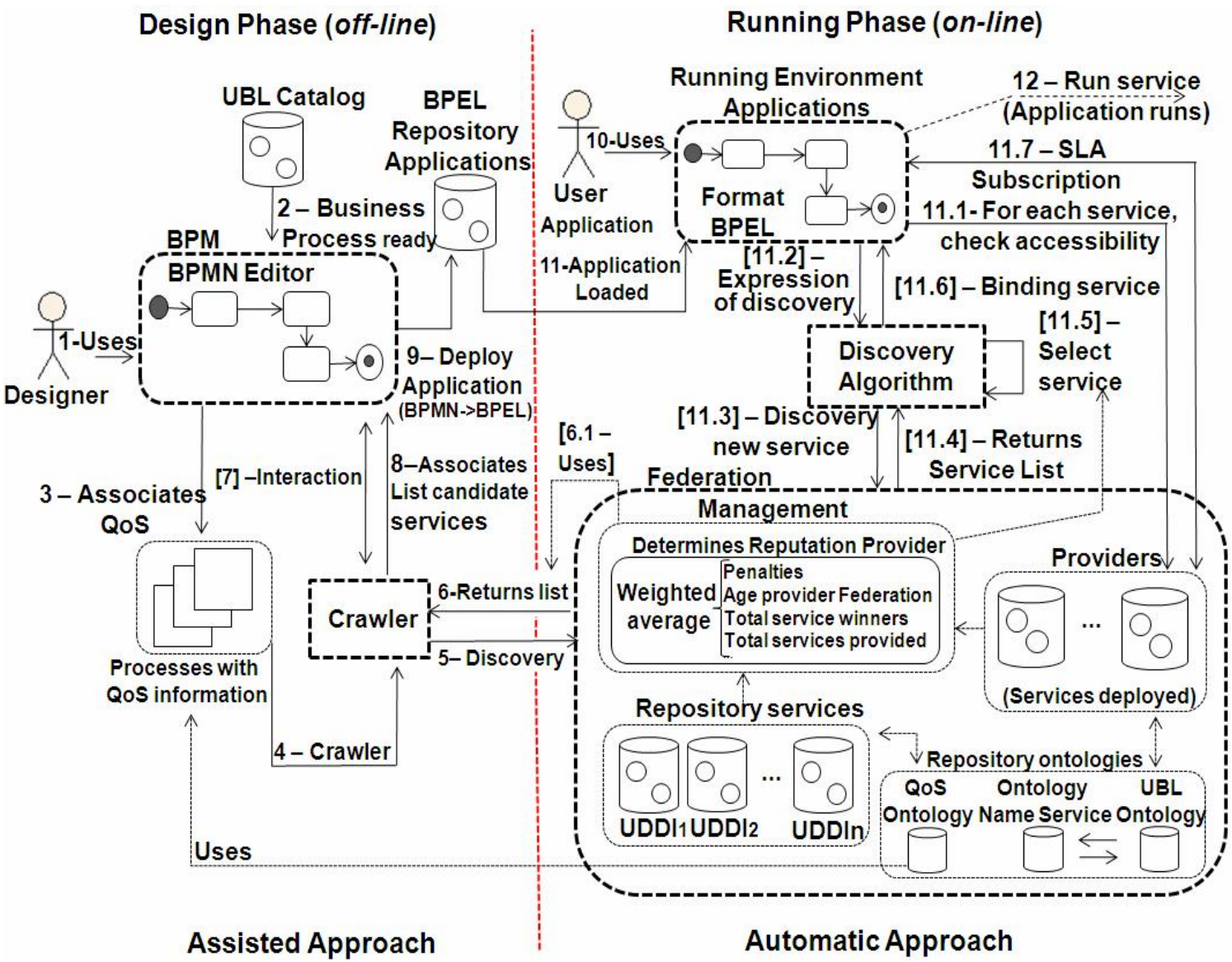
Proposed Services Discovery

It is proposed a novel approach for **dynamic** services discovery, strongly based on **ICT standards**, considering **QoS** and **BP context** in **largely distributed** services repositories (CNO members' repositories) and integrated with **BPM level**, also giving some support for SOA-based applications **composition** and **SLA** signature.

Basic Model's Rationale

- **BPM&SOA integration:**
  - BPM allows managers to express business processes' rules;
  - SOA acts as the link with the IT level, binding BPM processes to software services;
- Managers (at BPM level) can specify processes considering business process models **standards**, e.g. *UBL* or *RosettaNet*, which defines BPs' contexts.
- CNO members (including small software providers) share their services under the ***SaaS* model**.
- Composed SOA-based application gets ready to run (**BPEL**) as soon as the required services are discovered.

Proposed Model (Operation view)



Expression of Discovery

- It is composed of a 2-tuple $\mathbf{WebService_{wanted}} = (F,Q)$.

F: Required functionality

Sample:

Name: UBL/…/AcceptOrderBuyerParty

UBL ontology defines the capabilities of the service required

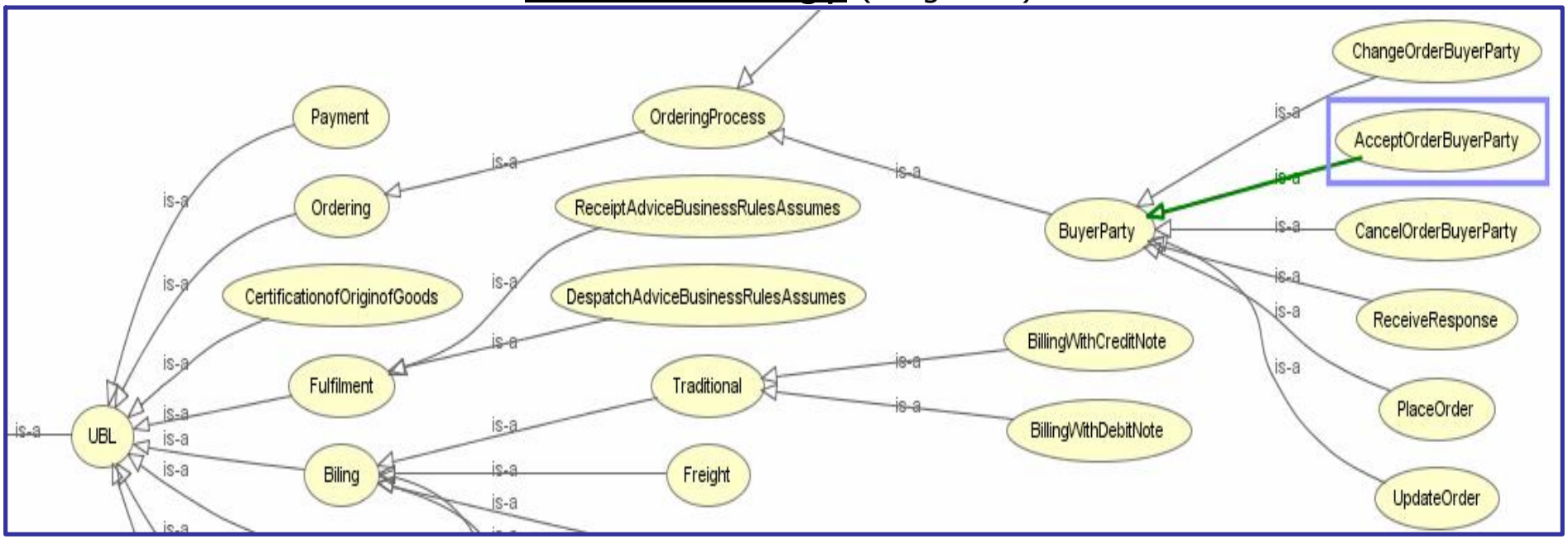
Q: QoS

Sample:

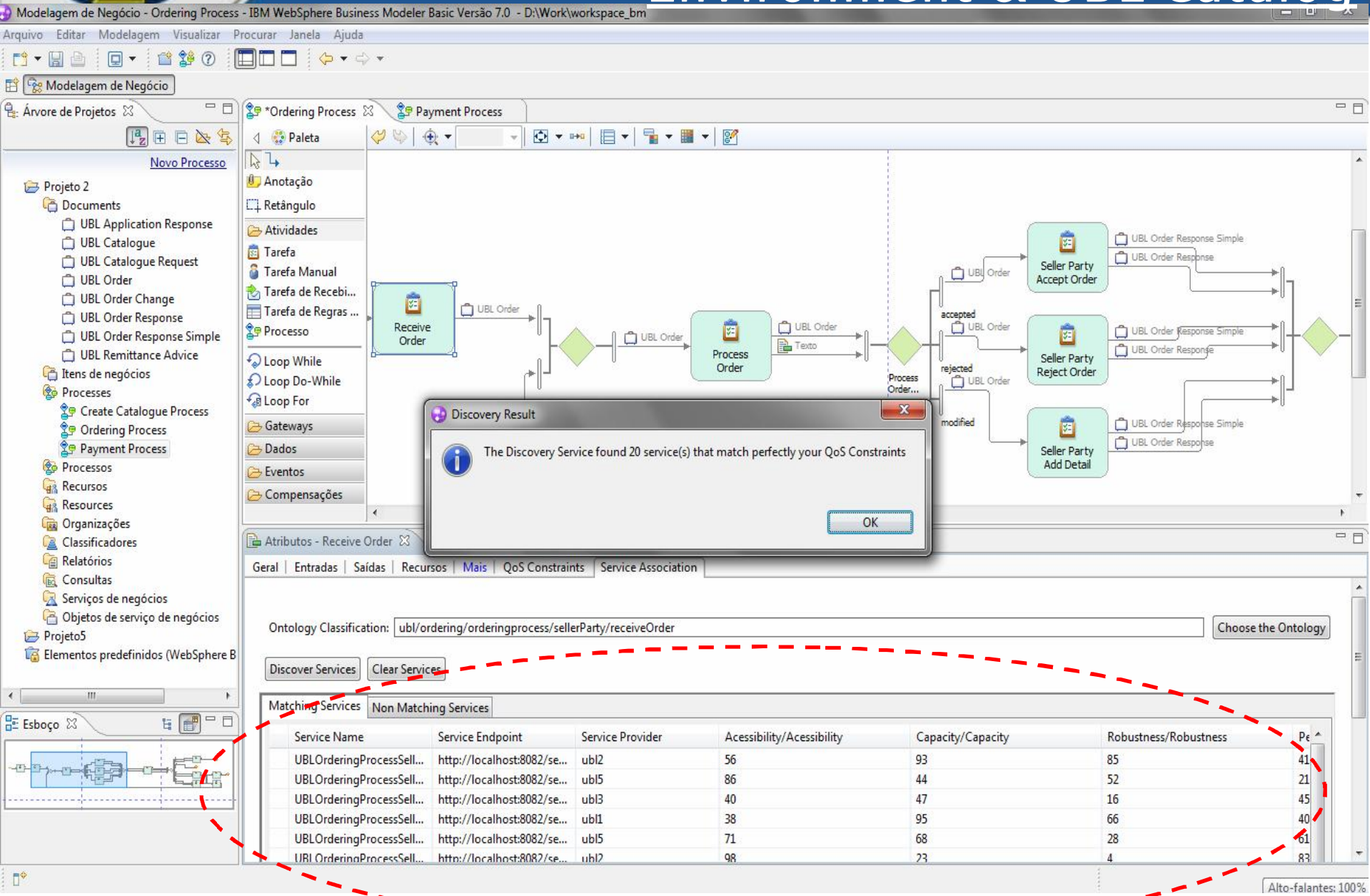
It expressed by a pair (name, value):

name = Response Time	value = 1s
name = Scalability	value = 500
…	…

O: UBL ontology (Fragment)



Initial Results: The BPM Environment & UBL Catalog

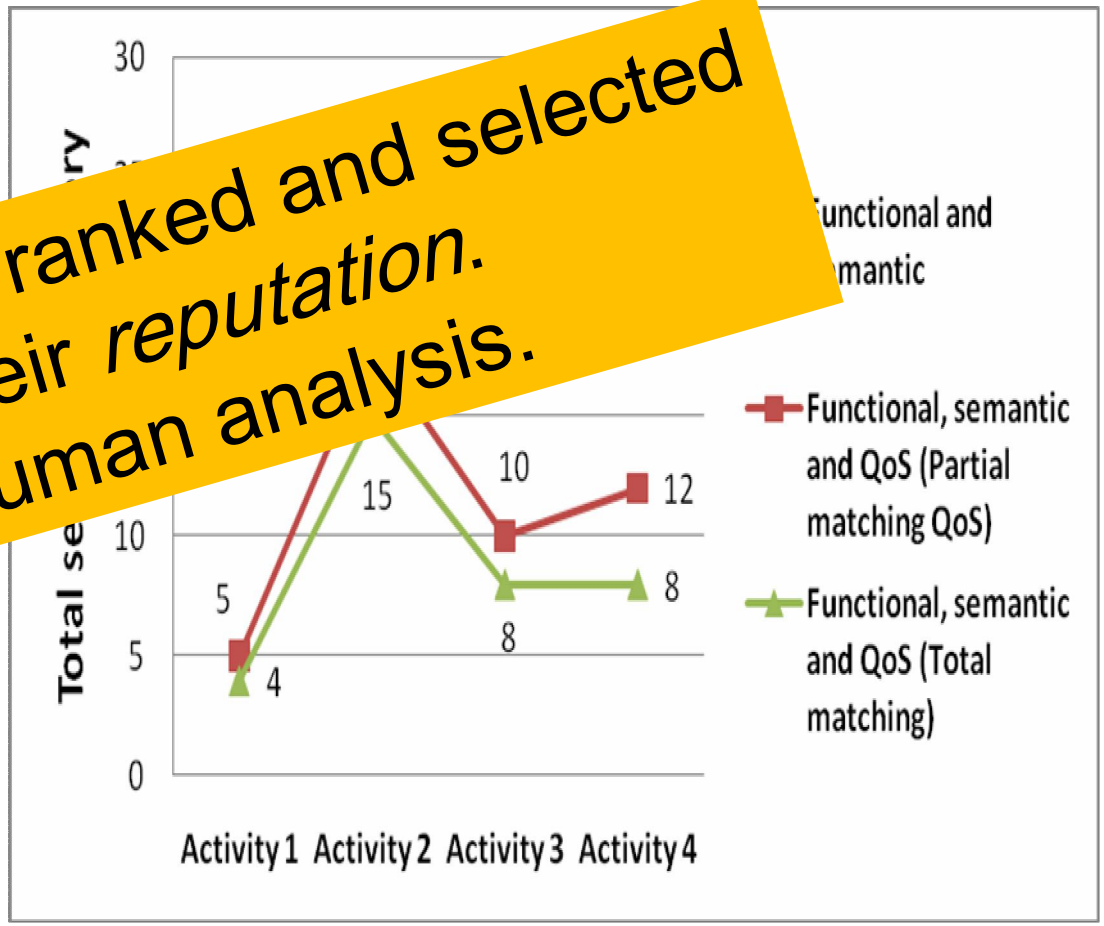


Initial Experimental Results

- Scenario
  - Publication of 100 web services related to the *Purchasing* UBL process:
    - 25 *Ordering*
    - 25 *Sourcing*
    - 25 *Payment*
    - 25 *Billing*
  - One local UDDI
  - Random QoS

- Experim
  - Query
    - Fu
  - Query
    - Func /Semantic
    - Partial QoS matching
    - QoS attributes relaxation
  - Query 3 (green line)
    - Functional/Semantic
    - Total QoS matching
    - QoS attributes relaxation

Services are ranked and selected based on their *reputation*.
Room for human analysis.



Final Considerations

- A **novel discovery model** has been presented as an initial contribution to the (very complex) problem of supporting CNO members to share software services, providing a concrete, integrated (BPM/SOA), standard-based, SaaS-based approaches.
- A **UBL catalog** within a BPM environment has been devised and implemented as well as **QoS and UBL ontologies**, besides the **discovery algorithm** itself and the **SLA model**.
- The discovery algorithm was able to get **only the most suitable services** for the given moment of a certain BP, supporting both QoS current requirements and flexibility from software providers.
- The model / exploratory research showed that it is possible to share services, hence **enlarging CNO collaboration**.
- The model does **not cover all issues** involved in services discovery and in this new scenario of services sharing among CNO members. A number of **assumptions** has been taken and the need for some improvements in the model has been identified.
- …

Issues supported by the Proposed Model

Business & Applications

- ✓ Contract services (SLA)
- ✓ Business Model (SaaS)
- ✓ Support for application design
- ✓ Supporting catalog of processes
- ✓ Support for services execution

Enabled activities via Discovery

- ✓ Integration of BPM & SOA, also regarding composition and interoperability.

Efficiency

- ✓ Support Accelerators
- ✓ Ranking criteria
- ✓ Computational Complexity
- • Support Queries (volatile/persistent)
- ✓ Binding Services (static/dynamic)

Architecture

- ✓ Implementation strategy
  - Classic SOA life cycle services
  - New elements
  - Distributed
- ✓ Repository (local/distributed)
- ✓ Following standards

Thank you !

UFSC - GSIGMA

www.gsigma.ufsc.br

Alexandre Perin-Souza, M.Sc.
perin@das.ufsc.br

Prof. Ricardo J. Rabelo
rabelo@das.ufsc.br

Main Next Steps

- Ongoing work (PhD Thesis).
- Expand the discovery to several and largely distributed services repositories (UDDIs) for doing a more robust and rigorous evaluation.
- Complexity of the current discovery algorithm grows linearly, which is a problem when having hundreds of providers. Some improvements are necessary, possibly inspired in P2P algorithms.
- Legal frameworks, security platforms, economic models, cultural aspects, among other, were not treated in this exploratory work.

Proposed Services Discovery

A number of sub-problems should be handled to support services discovery, such as: *how to express the desired service, how services should be published and expressed, which criteria should be applied to selected the "best" service, how security should be designed, how to bind services to processes and put them in execution*, etc.

Research Question

How to dynamically discover services over CNO's repositories that are better aligned to business processes respecting intrinsic QoS restrictions ?

Proposed Model: Main Assumptions

- Content Search:
  - **Web services.**
- Service Providers:
  - Services are implemented following **UBL standard specification (and ontology** !) and services **providers publish** their services accordingly.
  - There will be one (1) implemented service for each UBL activity.
  - A **common QoS ontology** is used by all providers and clients.
  - Providers can use their own ontologies and map them to the standard one.
- Discovery place:
  - "Cloud" of CNO members' services repositories.
  - There are potentially several services functionally equivalent available in the cloud and hence discoverable for each UBL business process' activity.
- Service composition:
  - **For each (1) individual UBL activity (within a SOA application) there will be one (1) corresponding web service to be bind to it.**
- BPM Environment:
  - Security, Availability, Versions control, services trustworthiness, services quality, "cloud" management, etc. : supported by "someone else".
- Access & Business Models:
  - CNO members will share & access services under SaaS.

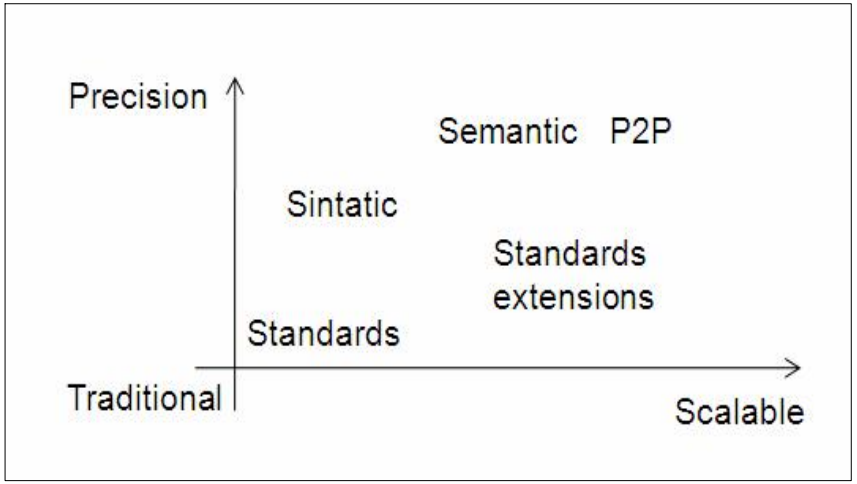
Research Problem

- Discovery:
  - is a step primary/essential life-cycle services.
  - is the act of discovering services that meet a set of criteria.
  - can be:
    - Static;
    - Dynamic.
  - has (intrinsically) several challenges:
    - Technological heterogeneity;
    - Problems of ambiguity;
    - Interoperability issues.

This study aims to answer the following question: how to support that members CN (considering CN characteristics and the modern requirements of a collaborative scenario) can leverage web services discovery and sharing?

Basic Literature Overview

- In essence, discovery is an issue complex and rich in alternatives:
  - Increase precision
    - Associate semantic descriptions of services;
    - Use QoS (Quality of Service) to allow selection according to quality aspects.
  - Improving the performance
    - Use mechanisms that accelerate the discovery response (caching).
  - Making the discovery highly available, scalable, robust, and other
    - Proposal for a differentiated architectures (P2P)
  - Binding at design time or runtime
    - Static binding
    - Dynamic binding
  - Achieving highly interoperable
    - Recommendations and standards
    - Ignore/extends standards



Basic Literature Overview

Issues addressed in discovery initiatives

Business & Applications

- Contract services (SLA)
- Model of Commercialization
- Approach to application design
- Support catalog of processes
- Support the run of services

Activity improved by Discovery

- Application composition
- Scalability
- Availability
- Other

Efficiency

- Support Accelerators
- Ranking criteria
- Computational Complexity
- Support Queries (volatile/persistent)
- Binding Services (static/dynamic)

Architecture

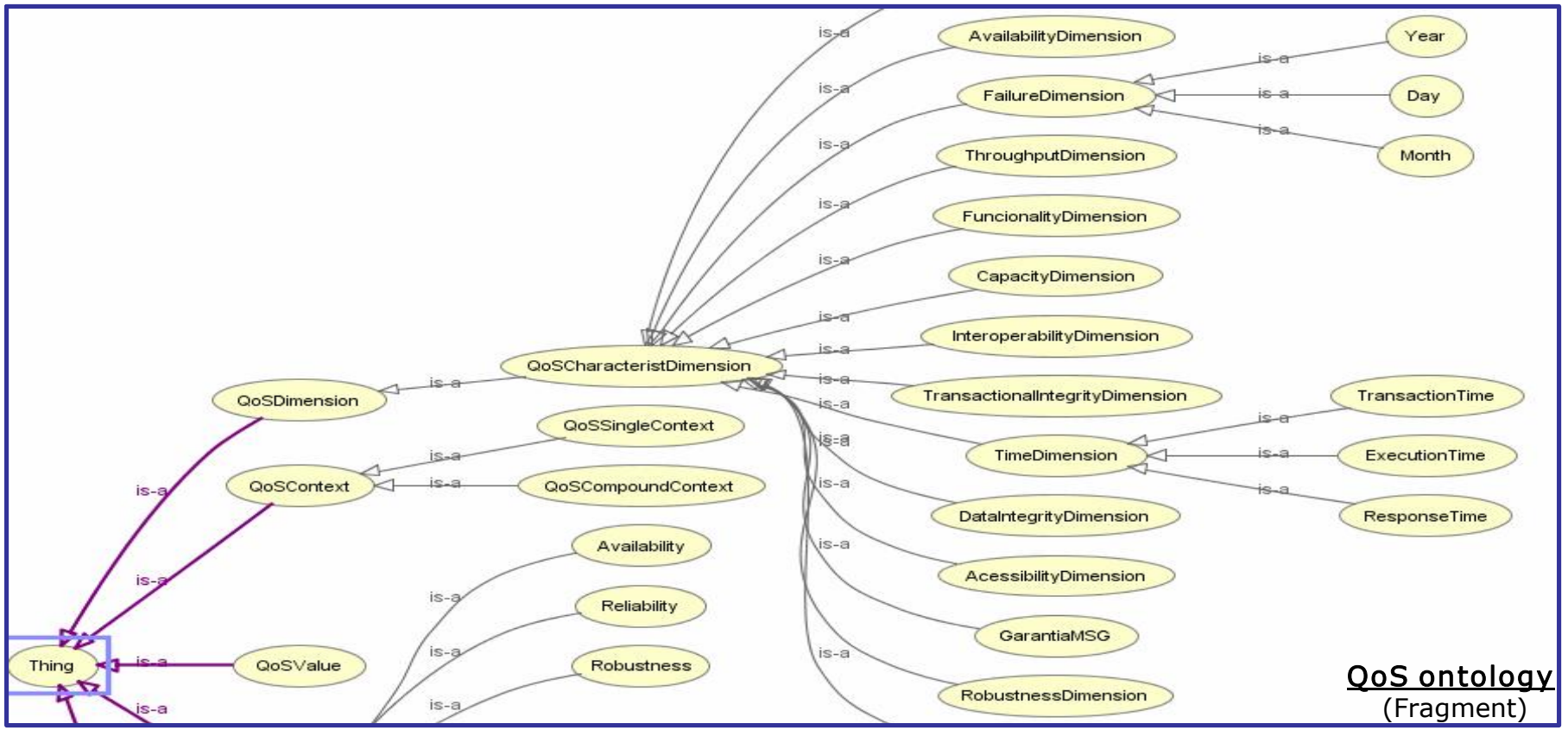
- Implementation strategy
  - Traditional life cycle services
  - Incorporates elements to the traditional cycle
  - Distributed
- Repository (local/distributed)
- Following standards

Prototype Implementation

- Development strategy
  - Reuse and development of other conceptual elements
    - Interactive and incremental approach
  - Tools
    - Java and IDE NetBeans v 6.7.1 with Tomcat v 6
    - MySQL v 5.1
    - MS Windows XP
- Several prototypes
  - First version
    - Classical discovery
  - Second version
    - Crawler, QoS, ...
  - Third version (still under development)
    - Discovery process in distributed UDDIs

QoS Ontology

- It was developed based on:
  - A high-level ontology proposed by Tondello (2008);
  - QoS attributes defined / used by OMG (2006) and W3C (2003).
  - Implemented in Protégé tool v4.0.2



QoS ontology
(Fragment)

A SOA application as a VE ?!