Specifying Web Workflow Services for Finding Partners in the Context of Loose Inter-Organizational Workflow

Eric Andonoff, Lotfi Bouzguenda (Phd), Chihab Hanachi

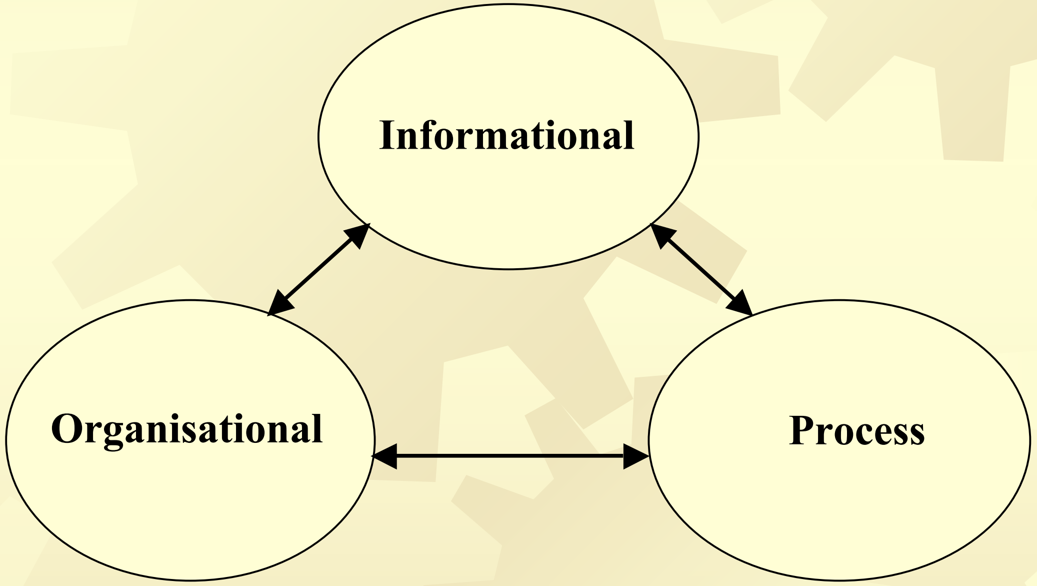
IRIT Laboratory, Toulouse, France

Outline

- Context and definition of the problem
- Requirements for a Workflow Web Service description language
- Limitations of current languages
- Our approach: from Petri Nets with Objects to OWL-S
- Implementation: matchflow
- Conclusion and future work

1. Context : from workflow to inter-organizational workflow

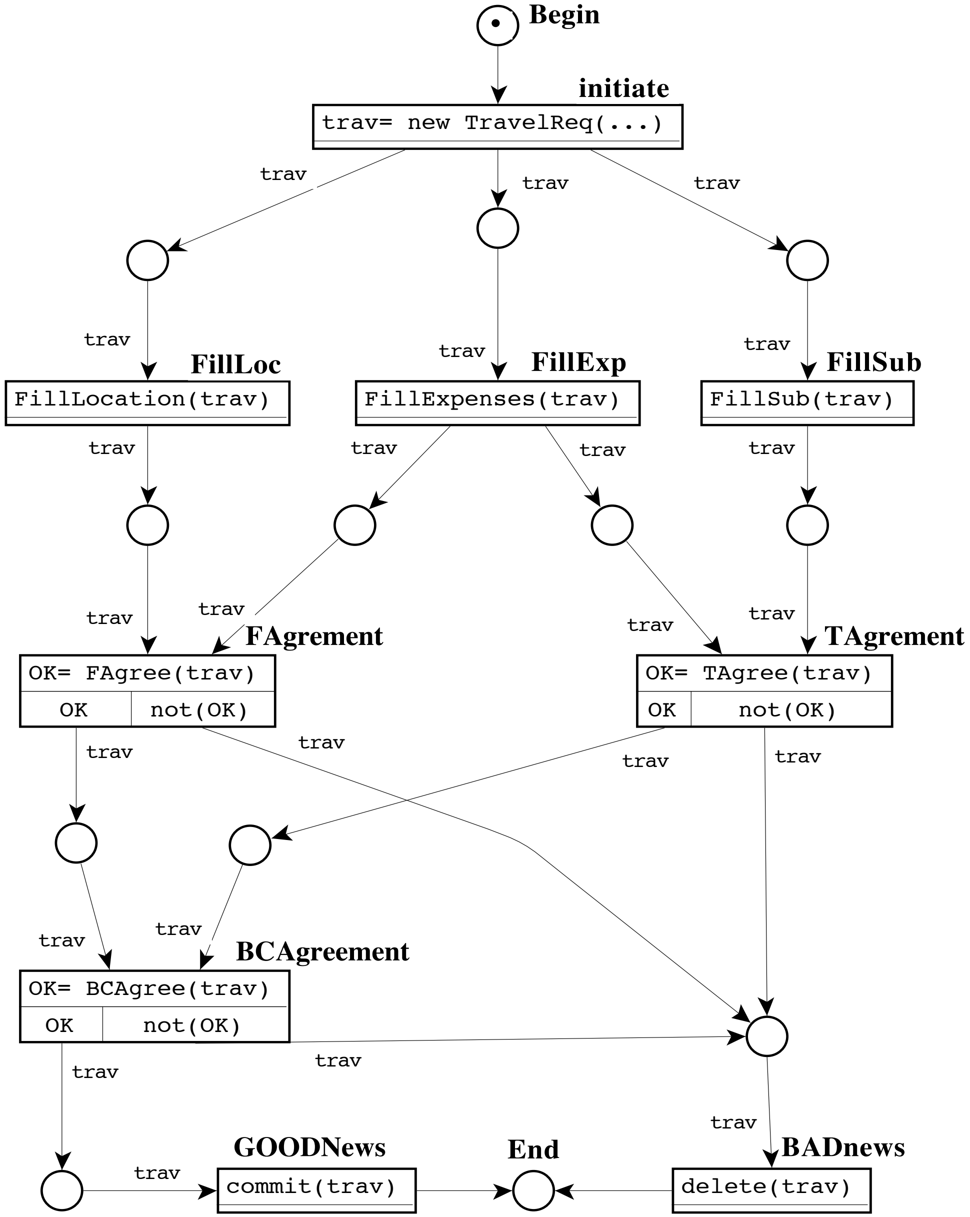
- ***Workflow*** : Automation of a business process within an organization.
- ***Workflow models :***



Context : from workflow to inter-organizational workflow

- Transition= task
- Input Place = required resource (info, performer)
- Output Place = result produced
- PN Structure : coordination of tasks

- Token= available resource
- Distribution of tokens=state of the process.



Context : Inter-organizational workflow

- N business partners put in common their workflow ⇒ Value Added Service
- IOW = n local Wf + A coordination model
- *Coordination model :*
  - *To rule/manage the interactions between local Wf.*
  - Constraints : heterogeneity, distribution, autonomy, confidentiality.
  - *Solutions:* composition, event publish/subscribe models, contract net allocation protocol, mediator, …
  - *Remains an Open issue notably with the emergence of semantic web-based technology.*

Context : 2 possible scenarios to study coordination in IOW [Divitini 01]

- Tight IOW :
  - Structural cooperation between organizations
  - Well-identified partners
  - Well-established coordination rules.
- *Loose IOW :*
  - Occasional cooperation
  - Free of structural constraints
  - Organizations involved and their number are not pre-defined.

Context : Coordination issues in Loose IOW

- Research of Partners:
  - Description, Publication of workflow services offers and requests
- Selection of partners:
  - Preferences, Matching mechanisms, Mediator.
- Negotiation with partners:
  - Protocols to reach agreement and establish contracts.
- Monitoring Execution and Managing Contracts.

Remark: amenable to multi-agent system

Problem being addressed

- Context :
  - Research of partners in Loose IOW
- Question :
  - How to describe workflow services through the web, in the same way as web service, in order to enable their <u>publication</u>, <u>discovering</u>, invocation and composition ?
  - What language for Workflow Web Services (W2S) description: should we define a new language or should we choose an existing one?

2. Requirements for Workflow Service Description Languages.

- *Appropriate expressive power:*
  - Description the three Wf aspects and their interactions.
  - Representing most of the « *control patterns* » involved in a process definition
- *To ease syntactic and semantic interoperability:*
  - Accessible via the Web ⇒ XML-like syntax
  - Context representation, semantic conflicts solving, matching process easing ⇒ Ontologies
- *Formal + operational semantics :*
  - Non ambiguous language
  - Analyses and simulation to validate and verify services ⇒ guaranteeing good properties before their publication.

4. Limitations existing languages: WSDL, BPEL4WS, WSFL, YAWL and OWL-S

	WSDL	BPEL4WS	WSFL	YAWL	OWL-S
An appropriate expressive power	-	+	+	+	**++**
Semantic Interoperability	–	–	–	–	**++**
Syntactic Interoperability	++	++	++	++	**++**
Formal with operational semantics	-	-	-	++	**-**

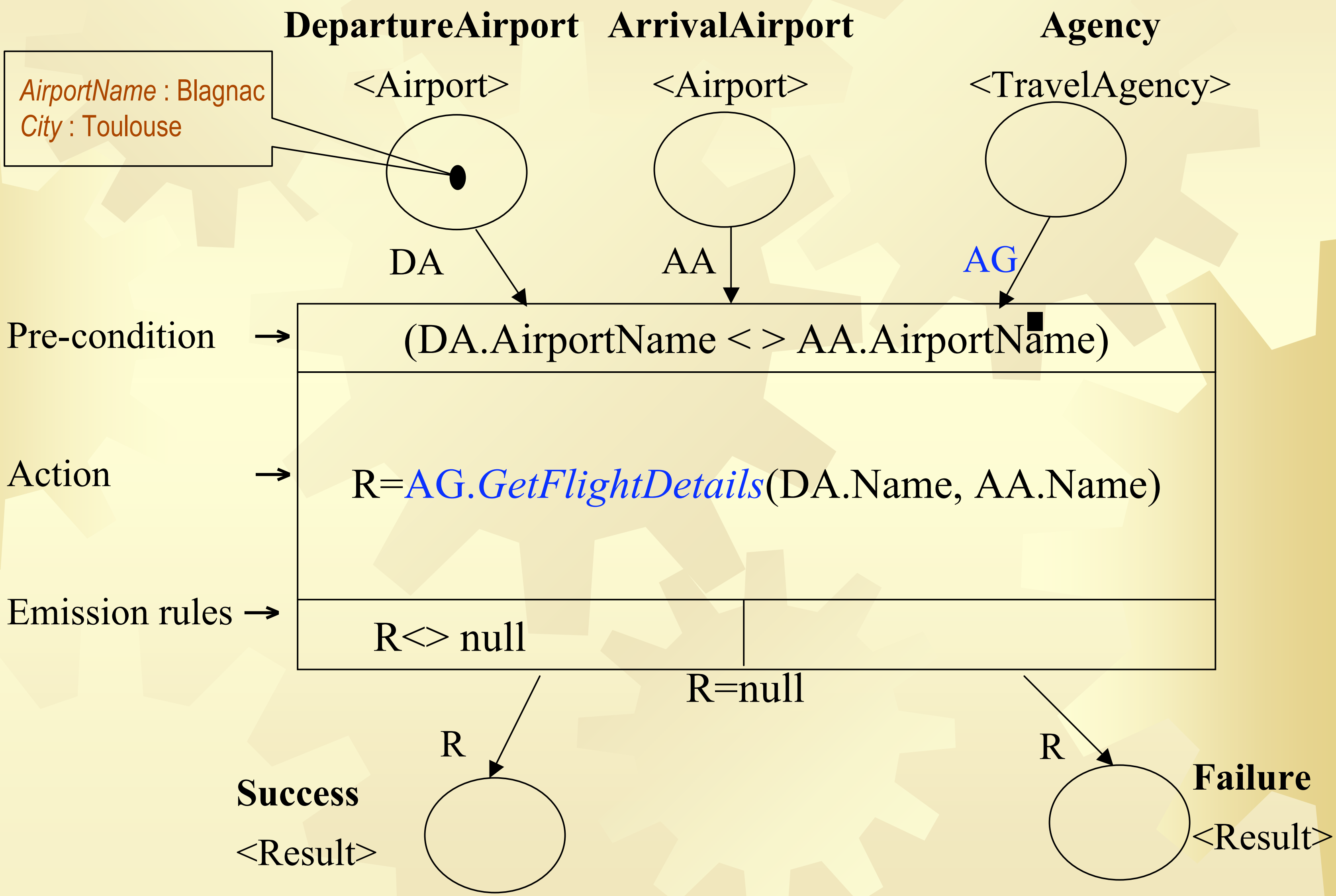
4. Our approach

1. Specification of workflow services with **Petri Nets with Objects** (PNO) :
   - Formal and graphic
   - With an Operational semantics : executable specifications
   - Integrating the three aspects of worklfow
   - Capturing all the OWL-S (control) patterns
2. Analyze, Simulation, Checking and Validation of the workflow service behaviour.
3. **Automatic derivation** of the previous workflow specification onto OWL-S specification (rules and algorithms)
4. Publication of the workflow services by means of **OWL-S**.

OWL-S Specification

- **OWL-S**
  - Semantic markup language
  - Refers to an ontology of services organized as hierarchy of classes, extensible according to the business domain considered.
- **Service Profile** (Interface level : info. needed to discover, compare and select services).
  - Attributes identifying the service : *serviceName, TextDescription, contactInformation*
  - Attributes describing the service capacity : *inputs, outputs, preconditions and effects*
  - Attributes classifying the service : *serviceCategory, qualityRating, serviceParameter*
- **Service Model** (Process/Operational level: how does it work?)
  - Atomic processes and composite processes thanks to constructors (*sequence, iterate, choice, split, split-join, …)*
  - For each process : *inputs, outputs, preconditions*, and *effects*
- **Service Grounding** (Exploitation level: how to access to it?)

Petri Nets with Objects through an example [Sibertin 1985]



Formal definition of PNO

A PNO is defined as a 9-uplet (***C***,*P, T, V, PreCond, A*, ***,*** *EmR, Pre, Post*) as follows :

C is a set of object classes,

P is a set of places, typed by a function P→C*,

T is a set of transitions, each transition being identified by a name,

V is a set of object variables, typed by a type function V→C,

PreCond is a set of preconditions, each one being necessary to trigger a transition,

A is a set of actions, each action being triggered by a transition,

EmR is a set of emission rules, each one corresponding to a logical expression

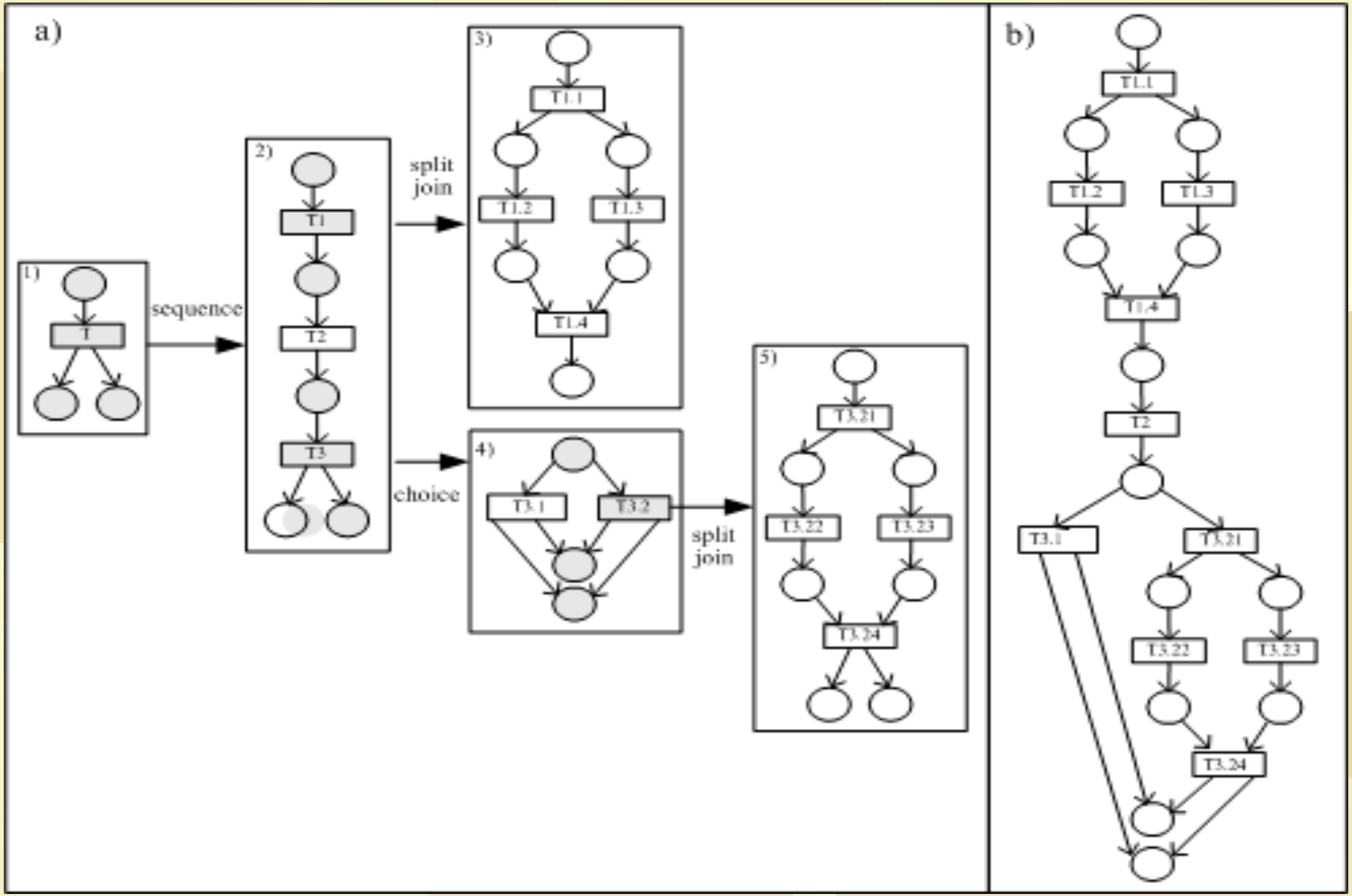
Pre is the forward incidence function: PxT→MultiSet(V*); Pre associates a multi-set of object variables to a (place, transition) couple,

Post is the backward incidence function: PxTxEmR→MultiSet(V*); Post associates a multi-set of object variables to a (place, transition, emission rule) triplet.

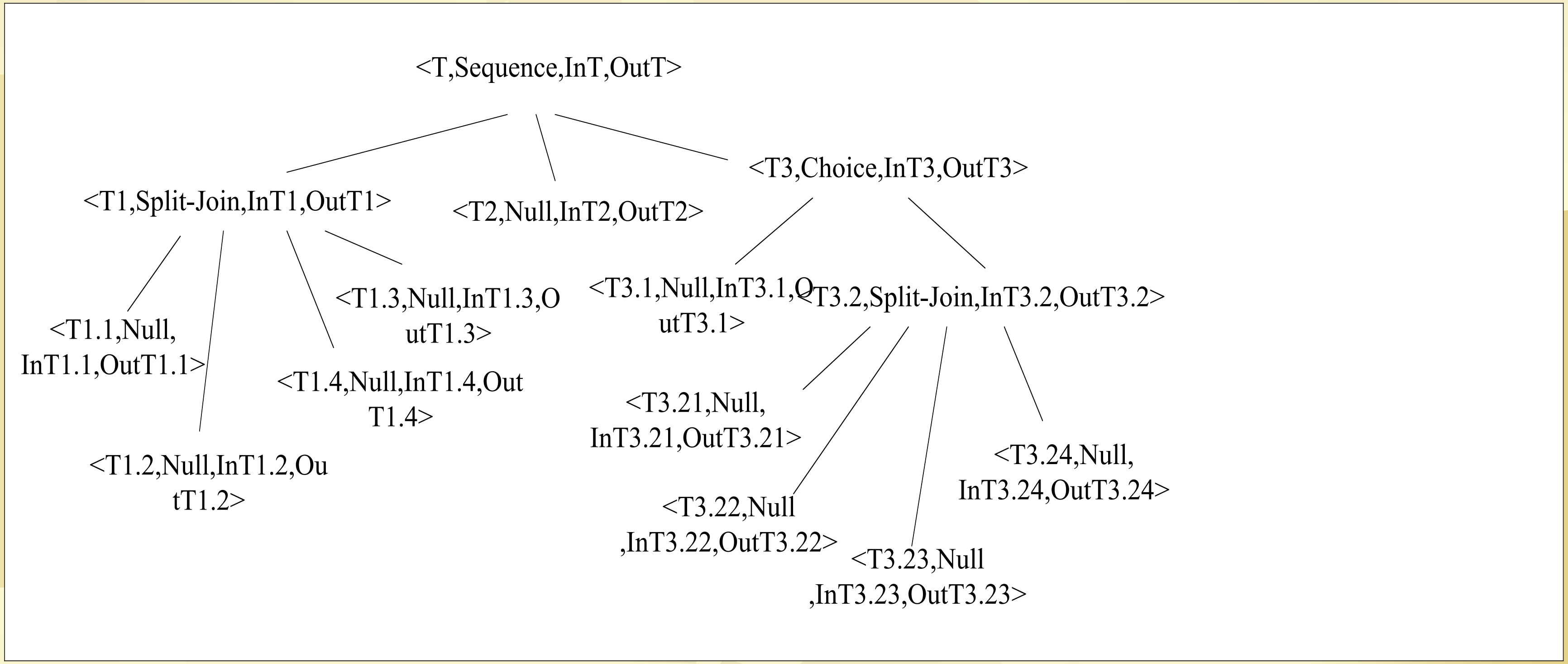
Advantages of using PNO for workflow description

- Advantages of using PN [Van Der Aalst 98] :
  - Adequate Expressiveness (patterns description).
  - Graphical representation
  - Operational semantics: simulation, execution.
  - Theoretical foundations ⇒ *analyse*
    - *Verification of behavioural properties* (ending, accessibility, liveness),
    - *performance evaluation (*average waiting time, occupation of resources, …)*.*
- Specific advantages of PNO:
  - Coherent description of the 3 workflow models;
  - May refer classes (of on ontology).

Hierarchical Specification of a Workflow Service using PNO



The corresponding PNO tree



Node {Transition
Pattern
InT {In,PreCdt},
OutT{Out,PostCdt}
}

Mapping PNO with OWL-S *Service Profile*

PNO	***OWL-S Service Profile***
source place : I-(I∩O))	Parameter Name of an Input <profile:input> … </profile:input>
sink place : O-(O∩I)	Parameter Name of an Output
Precondition associated to a source	Parameter Name of a Precondition
Emission rule associated to a sink place	Parameter Name of an Effect

Mapping PNO Tree with OWL-S *Service Process*

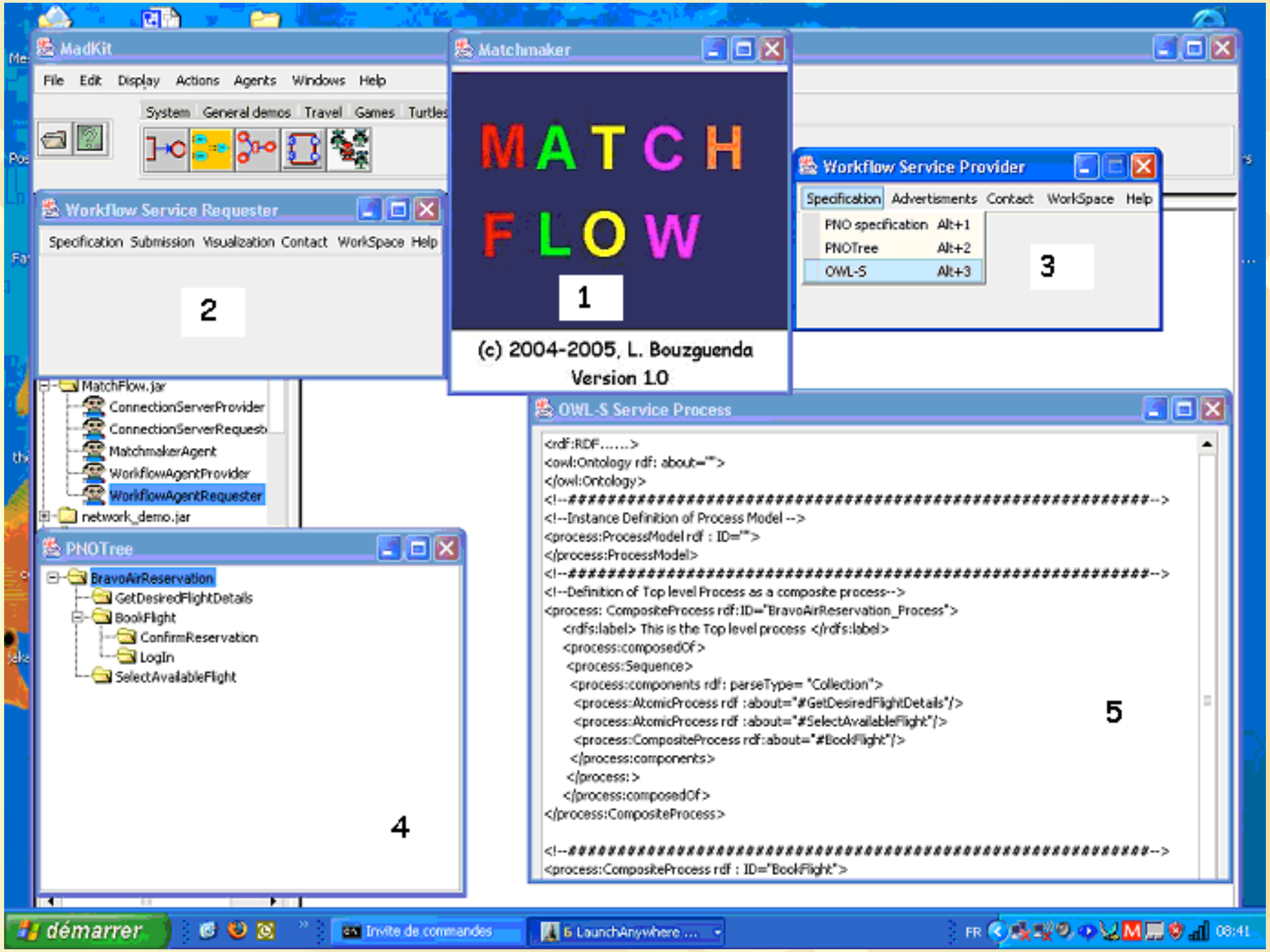
PNO tree	***OWL-S Service Process***
Name of a node	Name of a Process
(InputName, PreCondition) of a node	Input of a Process Precondition associated to the Input
(OutputName, EmRule) of a node	Output of a Process Effect associated to the Output
Terminal node (leaves)	Atomic Process
Non Terminal node	Composite Process

Implementation: MatchFlow

- Matchmaker :
  - connecting workflow service requesters and workflow service providers.
  - Offers and requests are specified using PNO and stored in OWL-S format.
  - Different comparison modes: exact, relaxed.
- Implemented with MADKIT:
  - Multi-Agent platform : java, distributed mode.
  - Based on an organizational abstractions (agent, role, group)

  ⇒ Good abstractions to deal with autonomy, distribution, heterogeneity and coordination.

Partial implementation: MatchFlow



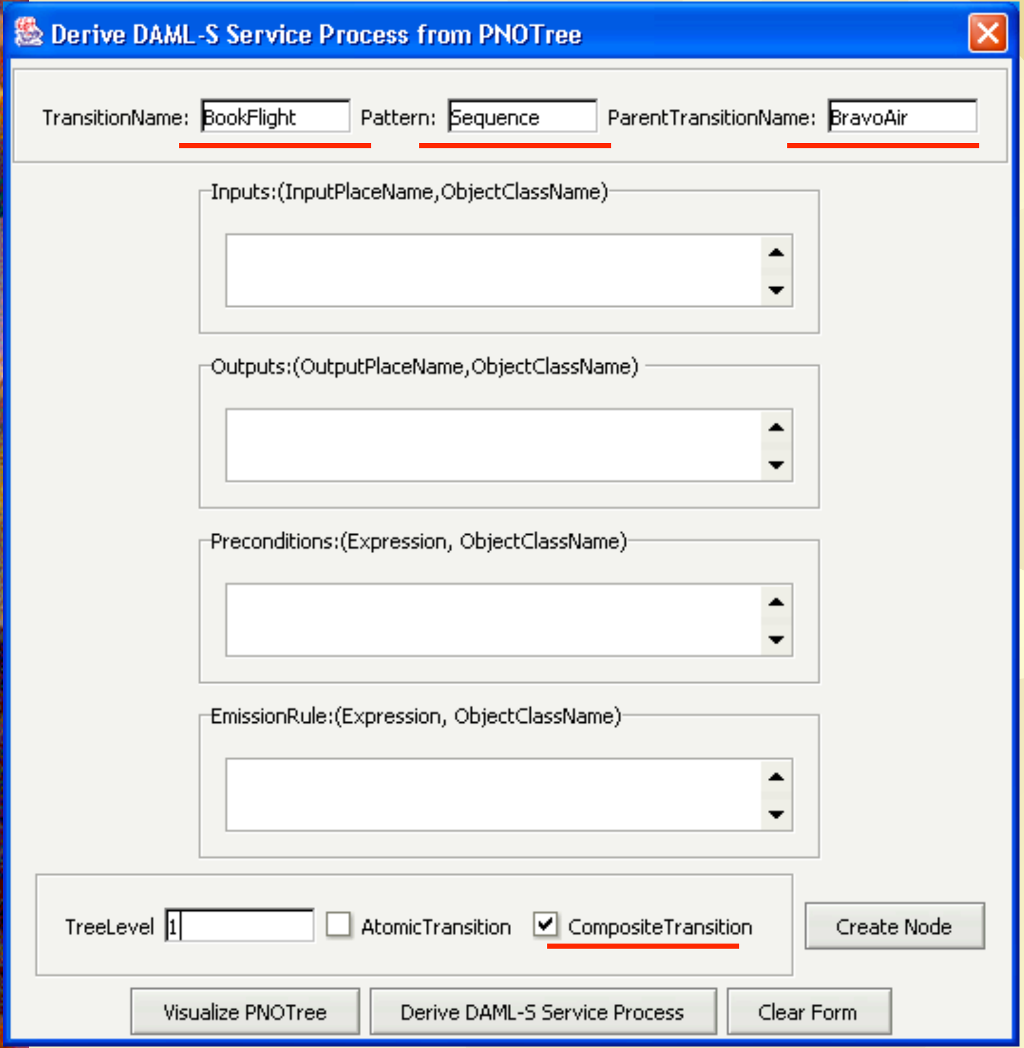
```
<rdf:RDF......>
<owl:Ontology rdf: about="">
</owl:Ontology>
<!--############################################################-->
<!--Instance Definition of Process Model -->
<process:ProcessModel rdf : ID="">
</process:ProcessModel>
<!--############################################################-->
<!--Definition of Top level Process as a composite process-->
<process: CompositeProcess rdf:ID="BravoAirReservation_Process">
    <rdfs:label> This is the Top level process </rdfs:label>
    <process:composedOf>
     <process:Sequence>
      <process:components rdf: parseType= "Collection">
       <process:AtomicProcess rdf :about="#GetDesiredFlightDetails"/>
       <process:AtomicProcess rdf :about="#SelectAvailableFlight"/>
       <process:CompositeProcess rdf:about="#BookFlight"/>
      </process:components>
     </process: >
    </process:composedOf>
</process:CompositeProcess>

<!--############################################################-->
<process:CompositeProcess rdf : ID="BookFlight">
```

Conclusion and future work

- OWL-S is convenient for workflow web service publication:
  - Appropriate expressive power;
  - Includes ontology that eases *semantic interoperability*, matchmaking mechanism;
  - describes reasonably workflow services
  - No guarantee of their correct execution.
- PNOs are convenient for workflow specification:
  - *Glue* between the different workflow models;
  - *Formal and executable* specifications, simulation and validation;
  - Not web oriented
- An Appropriate combination of PNO and OWL-S compensates these drawbacks.
- Automatic derivation of PNO specification onto OWL-S.
- Future work:
  - refining OWL-S ontology to integrate workflow properties and performance evaluations checked on the PNO.
  - Described as a sub-class of the process properties.

Exemple: BravoAirReservation

OWL-S Service Process



Derive DAML-S Service Process from PNOTree

TransitionName: BookFlight Pattern: Sequence ParentTransitionName: BravoAir

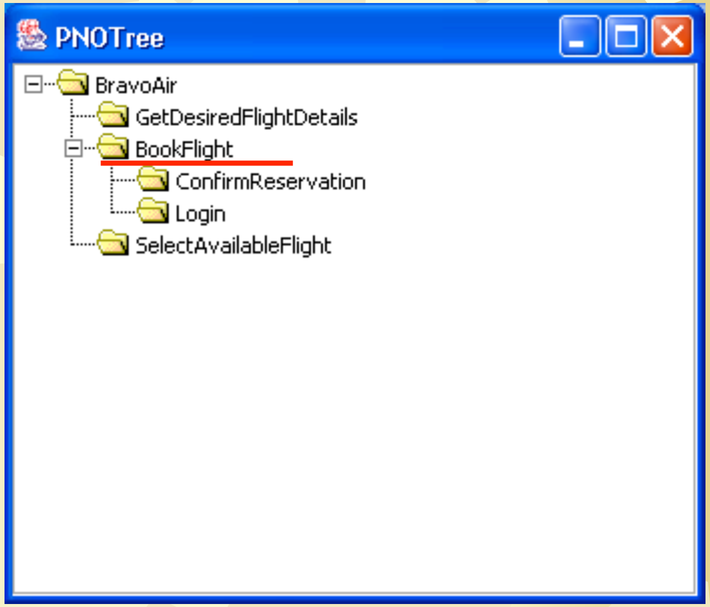
Inputs:(InputPlaceName,ObjectClassName)

Outputs:(OutputPlaceName,ObjectClassName)

Preconditions:(Expression, ObjectClassName)

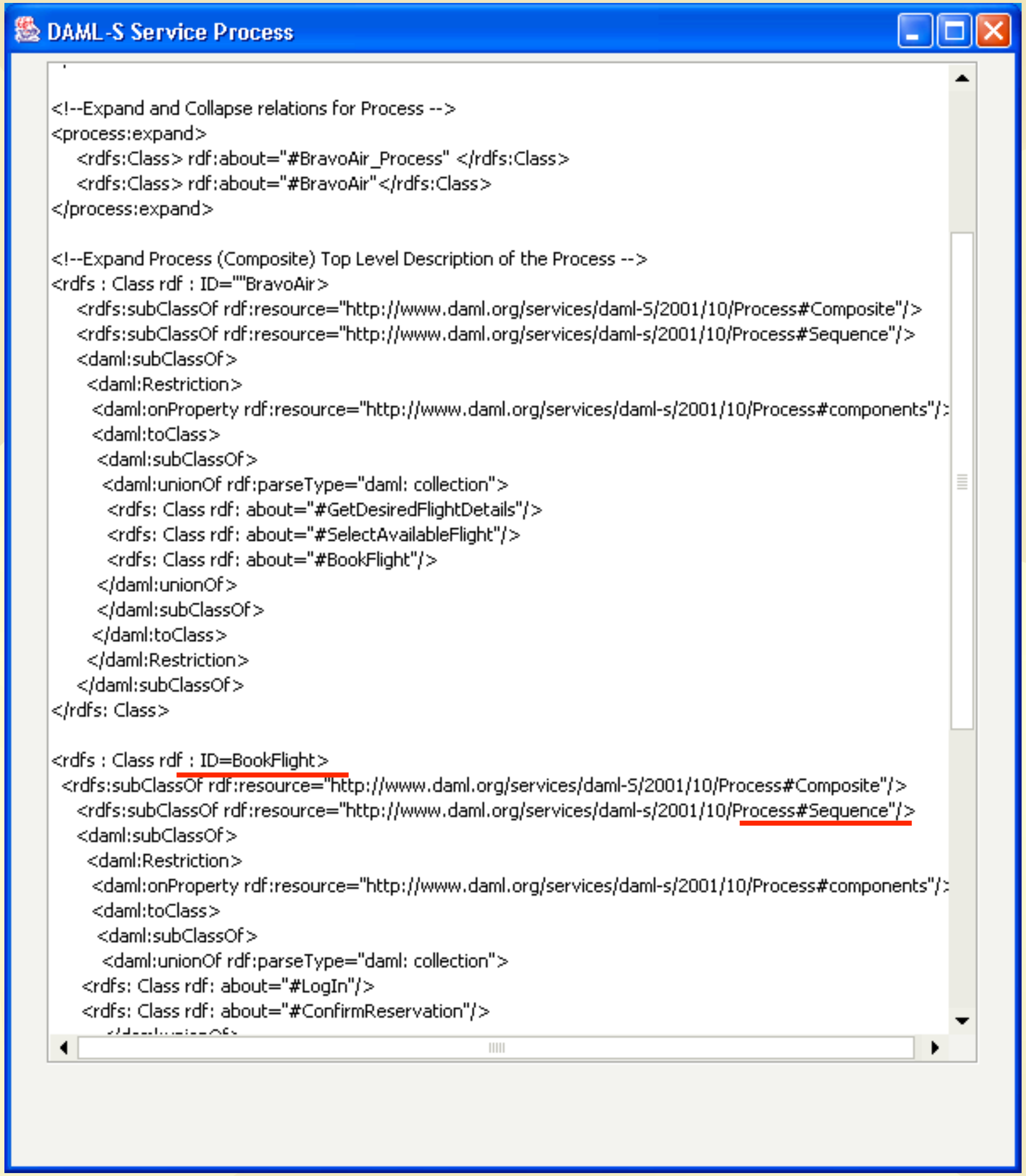
EmissionRule:(Expression, ObjectClassName)

TreeLevel 1 □ AtomicTransition ☑ CompositeTransition



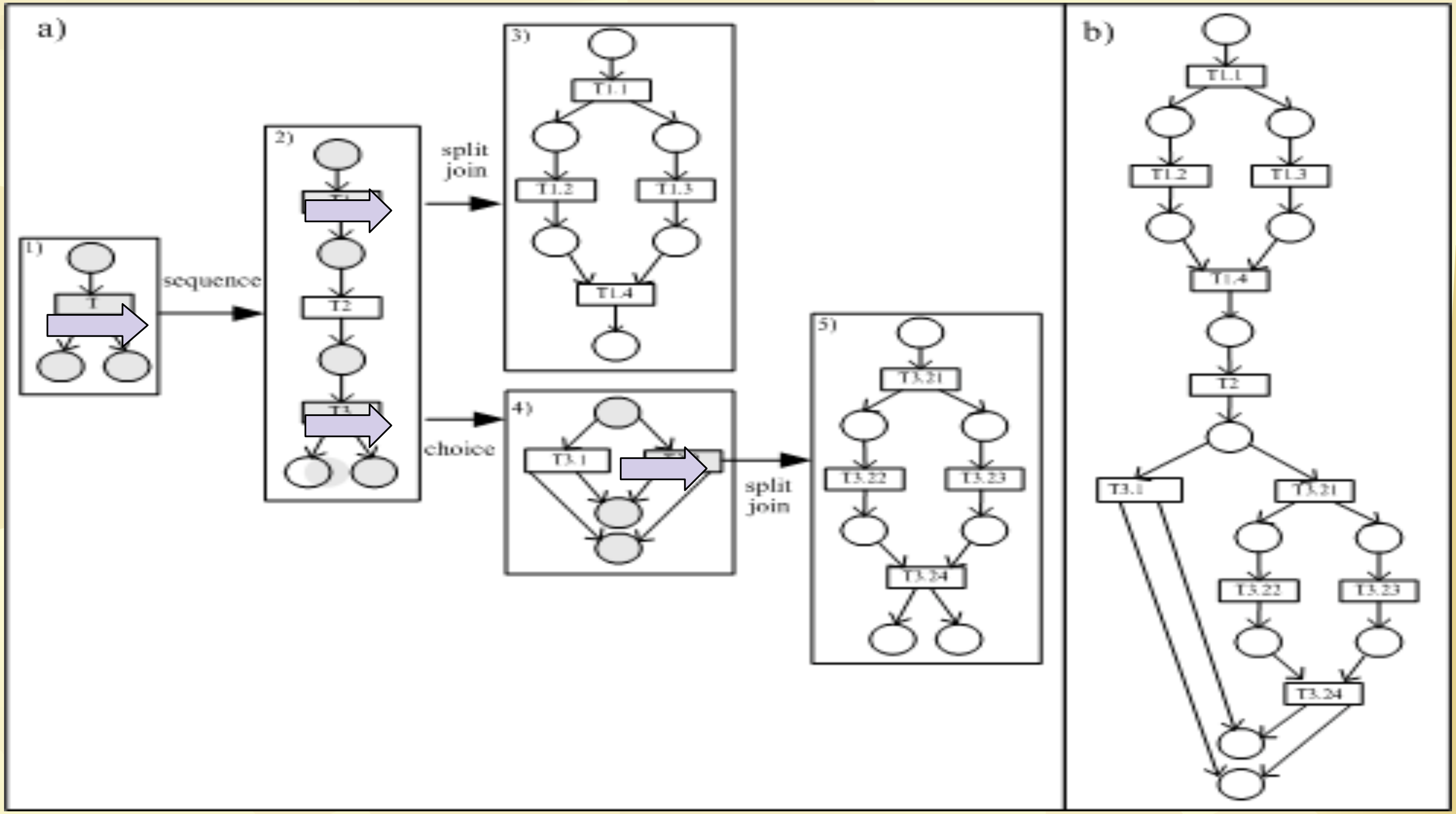
PNOTree

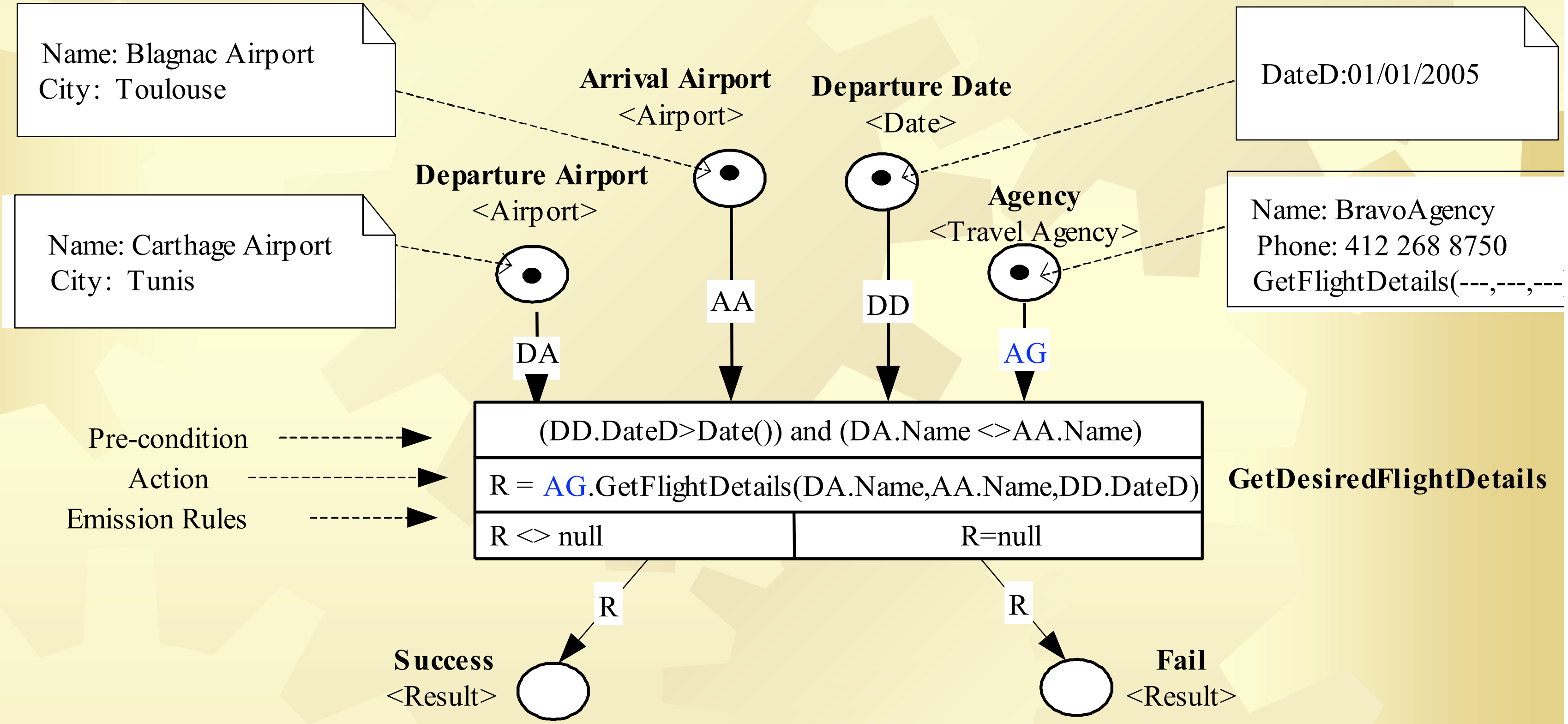
- BravoAir
  - GetDesiredFlightDetails
  - BookFlight
    - ConfirmReservation
    - Login
  - SelectAvailableFlight



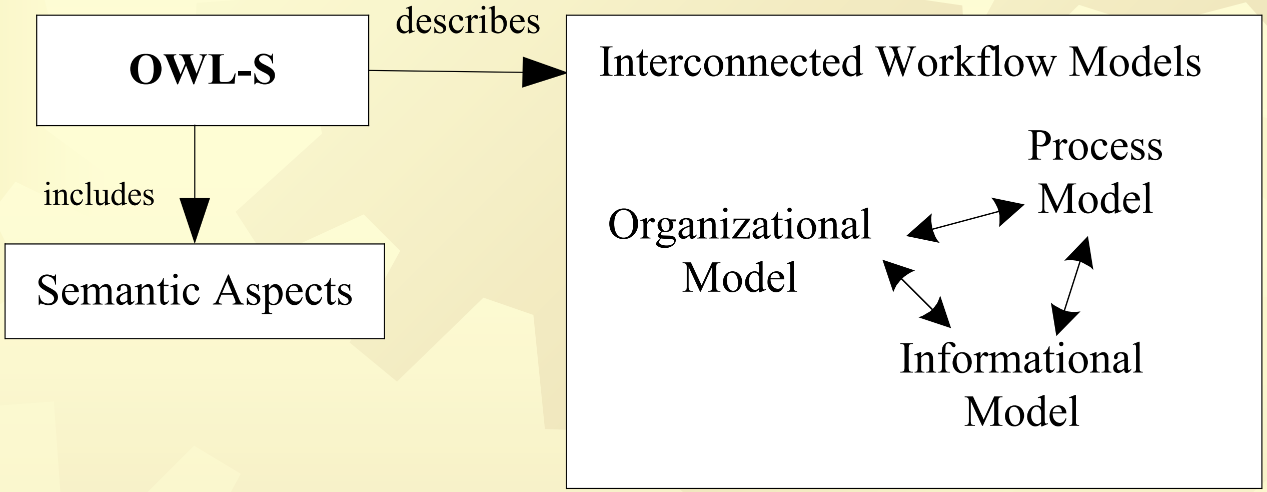
```
<!--Expand and Collapse relations for Process -->
<process:expand>
    <rdfs:Class> rdf:about="#BravoAir_Process" </rdfs:Class>
    <rdfs:Class> rdf:about="#BravoAir"</rdfs:Class>
</process:expand>

<!--Expand Process (Composite) Top Level Description of the Process -->
<rdfs : Class rdf : ID=""BravoAir>
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/daml-S/2001/10/Process#Composite"/>
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/daml-s/2001/10/Process#Sequence"/>
    <daml:subClassOf>
      <daml:Restriction>
       <daml:onProperty rdf:resource="http://www.daml.org/services/daml-s/2001/10/Process#components"/>
       <daml:toClass>
        <daml:subClassOf>
         <daml:unionOf rdf:parseType="daml: collection">
          <rdfs: Class rdf: about="#GetDesiredFlightDetails"/>
          <rdfs: Class rdf: about="#SelectAvailableFlight"/>
          <rdfs: Class rdf: about="#BookFlight"/>
        </daml:unionOf>
        </daml:subClassOf>
       </daml:toClass>
      </daml:Restriction>
    </daml:subClassOf>
</rdfs: Class>

<rdfs : Class rdf : ID=BookFlight>
  <rdfs:subClassOf rdf:resource="http://www.daml.org/services/daml-S/2001/10/Process#Composite"/>
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/daml-s/2001/10/Process#Sequence"/>
    <daml:subClassOf>
      <daml:Restriction>
       <daml:onProperty rdf:resource="http://www.daml.org/services/daml-s/2001/10/Process#components"/>
       <daml:toClass>
        <daml:subClassOf>
         <daml:unionOf rdf:parseType="daml: collection">
     <rdfs: Class rdf: about="#LogIn"/>
     <rdfs: Class rdf: about="#ConfirmReservation"/>
        </daml:unionOf>
```

Hierarchical Design of a Workflow Service using PNO





Name: Blagnac Airport
City: Toulouse
Arrival Airport
<Airport>
Departure Date
<Date>
DateD:01/01/2005
Departure Airport
<Airport>
Agency
<Travel Agency>
Name: Carthage Airport
City: Tunis
Name: BravoAgency
Phone: 412 268 8750
GetFlightDetails(---,---,---
AA
DD
DA
AG
Pre-condition
Action
Emission Rules
(DD.DateD>Date()) and (DA.Name <>AA.Name)
R = AG.GetFlightDetails(DA.Name,AA.Name,DD.DateD)
R <> null
R=null
GetDesiredFlightDetails
R
R
Success
<Result>
Fail
<Result>


OWL-S
describes
includes
Semantic Aspects
Interconnected Workflow Models
Process Model
Organizational Model
Informational Model

Properties of PN

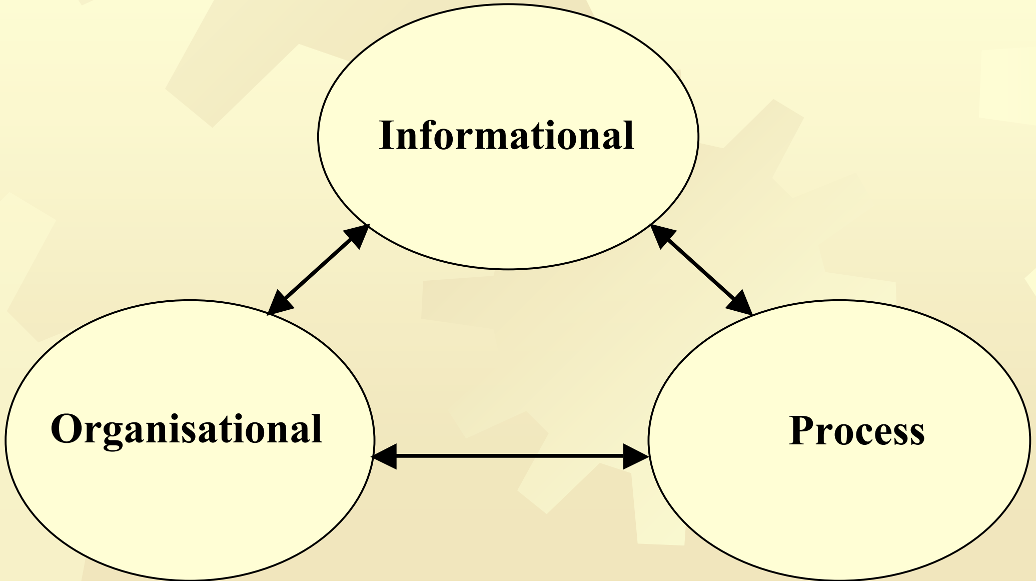
- Ending: does a process effectively end?)
- Liveness: is a given task (transition) always possible?
- Boundedness: is the number of possible configurations of a process finite?
- Reachability: is there an evolution in the process leading to a given configuration (desired or not)?)
- Quasi-Liveness: does a configuration exist where a given task is possible?.

Performance evaluations

- Average throughput time;
- Average waiting time;
- Occupation rates of resources.

1. Context : from workflow to inter-organizational workflow

- ***Business Process*** : a set of coordinated tasks, within an organization, to achieve a well-defined business outcome.
- ***Workflow*** :
  - technology for understanding, modelling and automating business processes.
  - Automation of a business process
- ***Workflow models***


Informational
Organisational
Process