Work at ISI, Current Status, Next Steps

Daniel Garijo Verdejo,
Oscar Corcho,
Yolanda Gil

Ontology Engineering Group. Laboratorio de Inteligencia Artificial
Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid

Date: 29/11/2012

What am I working on?

- Creation of abstractions in scientific workflows
    - Workflow Traces and template representation
        - Provenance representation
        - Plan representation
    - Abstraction catalog
    - Find ways to link the definitions to the provenance traces automatically
- Understandability and reuse of scientific workflows

Outline

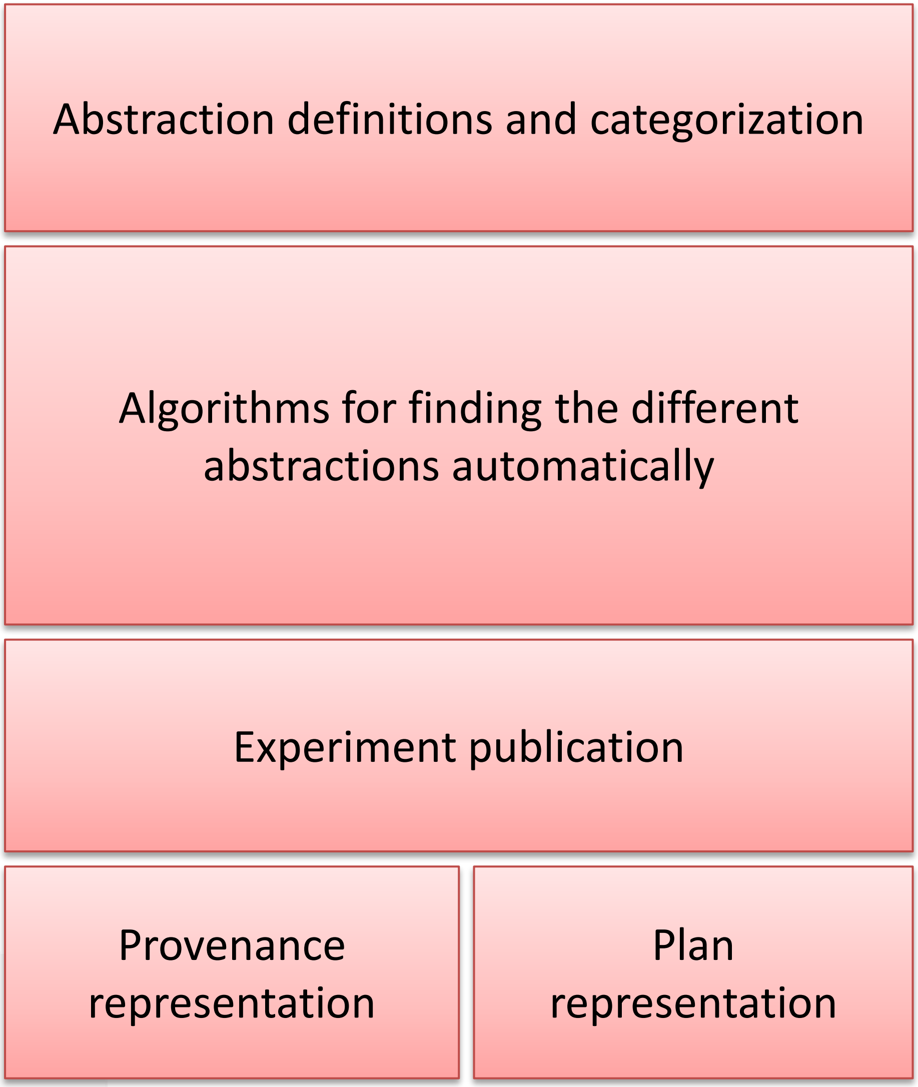
Index

1. Motivation
2. Overview
3. Workflow systems used
4. Summary of work done in my previous visit to ISI
   - OPMW and provenance publishing
5. Summary of work done before second visit to ISI
   - Workflow motif catalog
6. Summary of work done in my second visit to ISI
   - OPMW-PROV and P-PLAN
   - Automatic macro abstraction detection
7. Next Steps
8. Future work

Motivation

- As a designer: Discovery
  - Workflows with similar functionality fragments/methods
  - Design based in previous templates.
- As user/reuser: Understandability
  - Search workflows by functionality
  - Commonalities between execution runs
  - Component categorization

Overview



Abstraction definitions and categorization

Algorithms for finding the different abstractions automatically

Experiment publication

Provenance representation

Plan representation

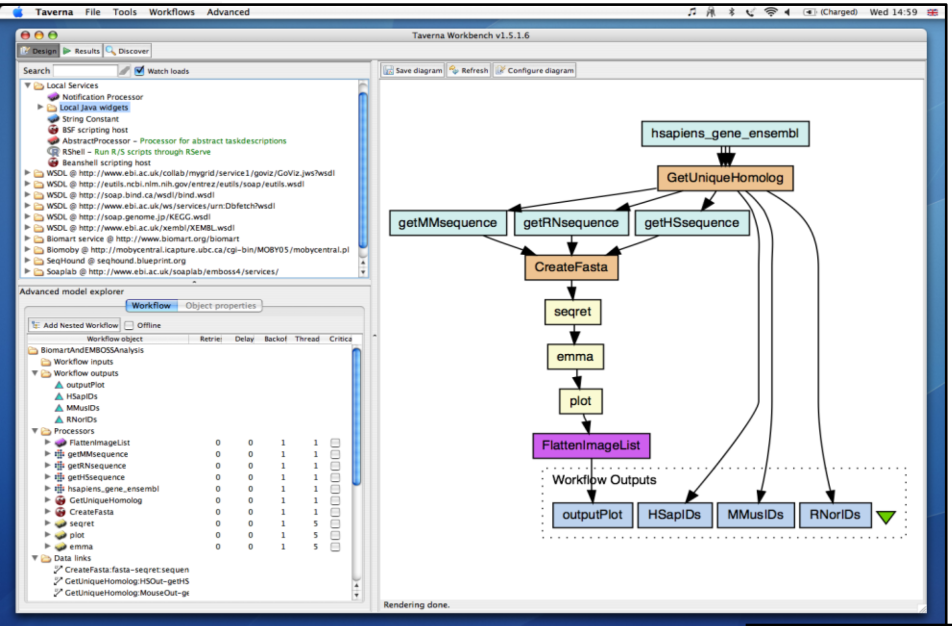
Descriptions/ PSMS/Ontologies

Data mining tools, graph analysis, etc.

RDF Stores

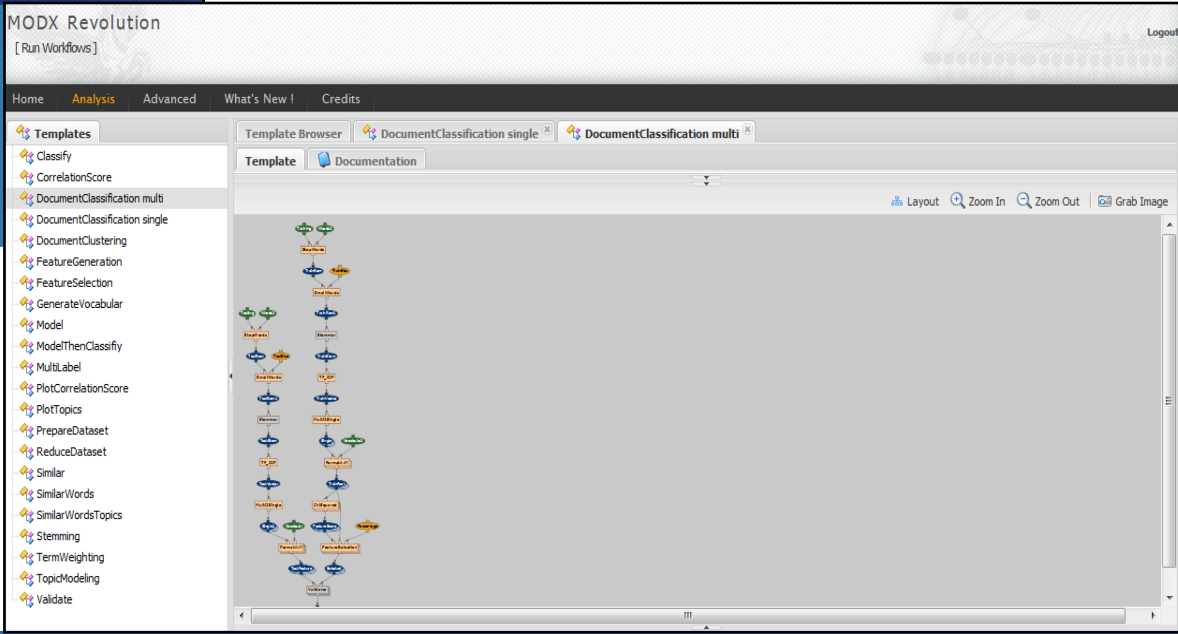
Vocabularies

Taverna and Wings

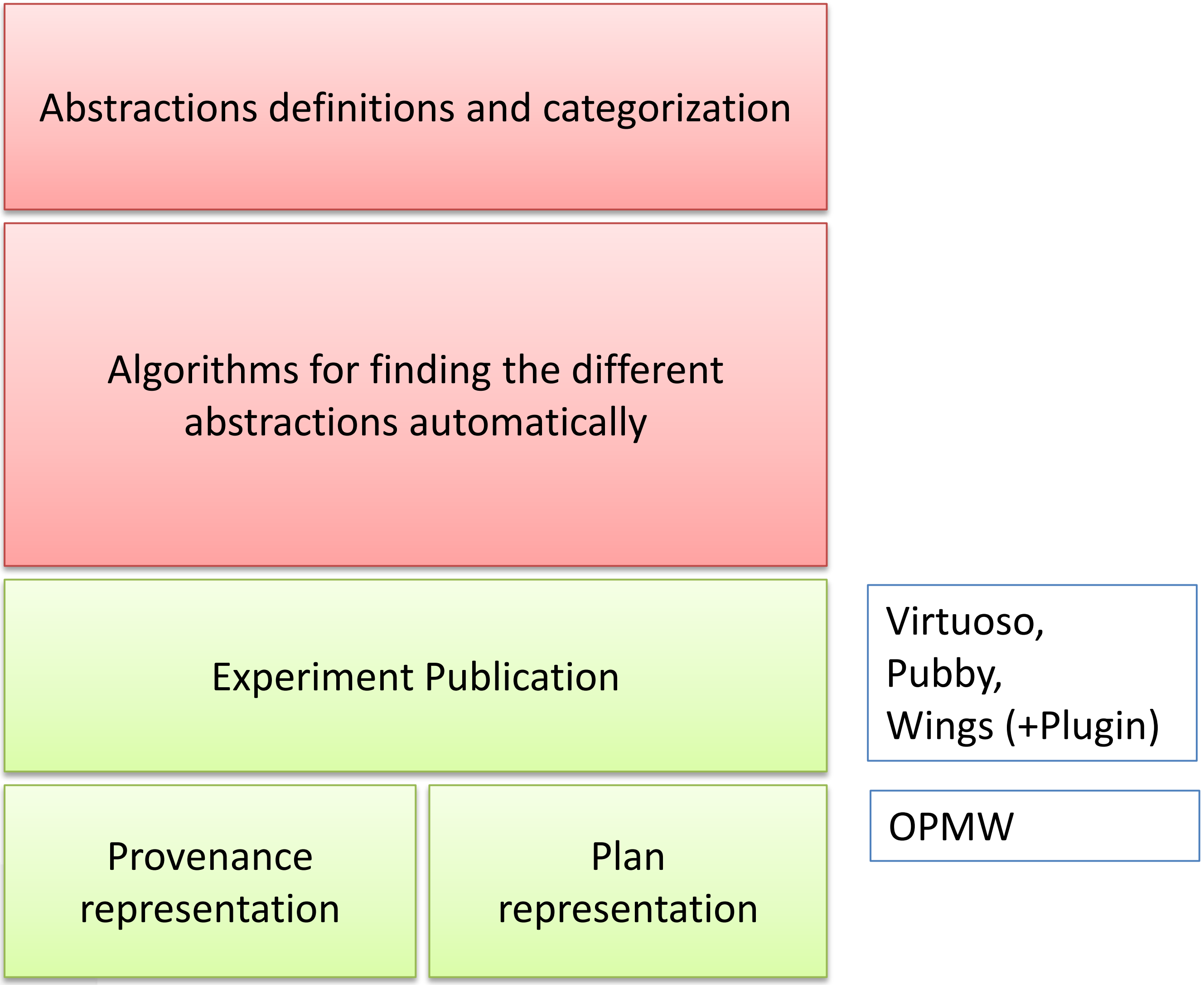


http://www.taverna.org.uk/

http://www.wings-workflows.org/



Summary: Previous Work at ISI



Abstractions definitions and categorization

Algorithms for finding the different abstractions automatically

Experiment Publication

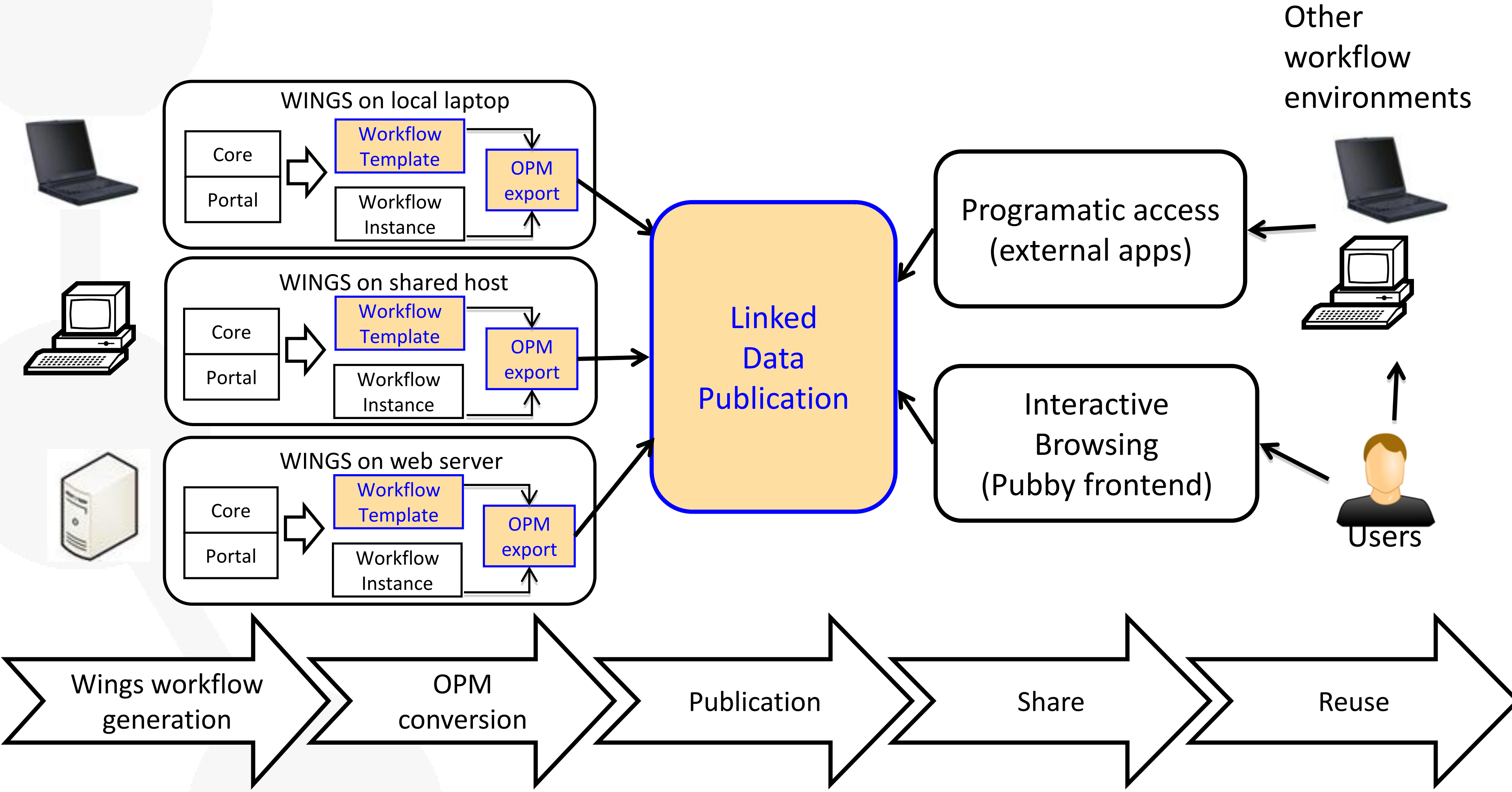
Provenance representation

Plan representation

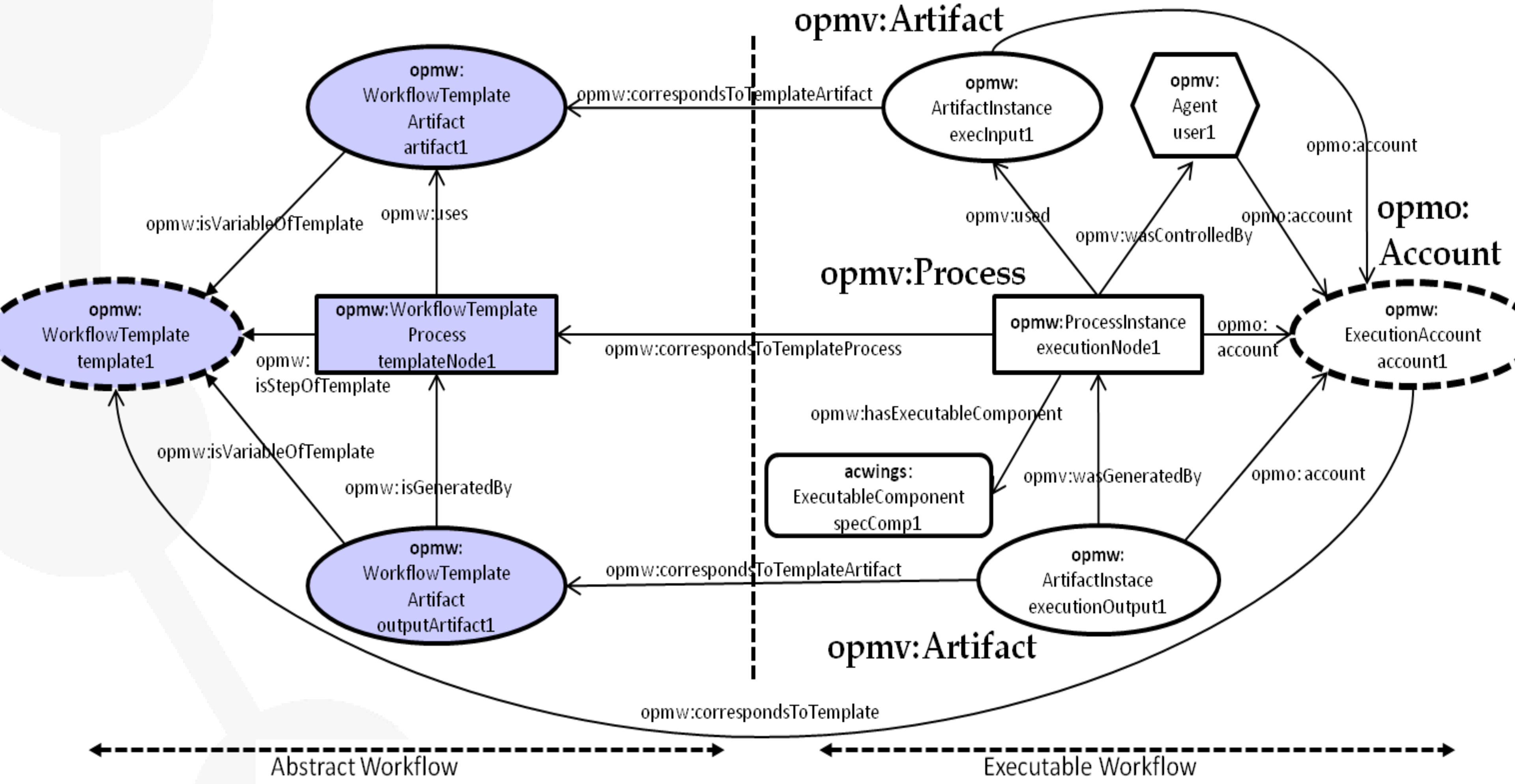
Virtuoso,
Pubby,
Wings (+Plugin)

OPMW

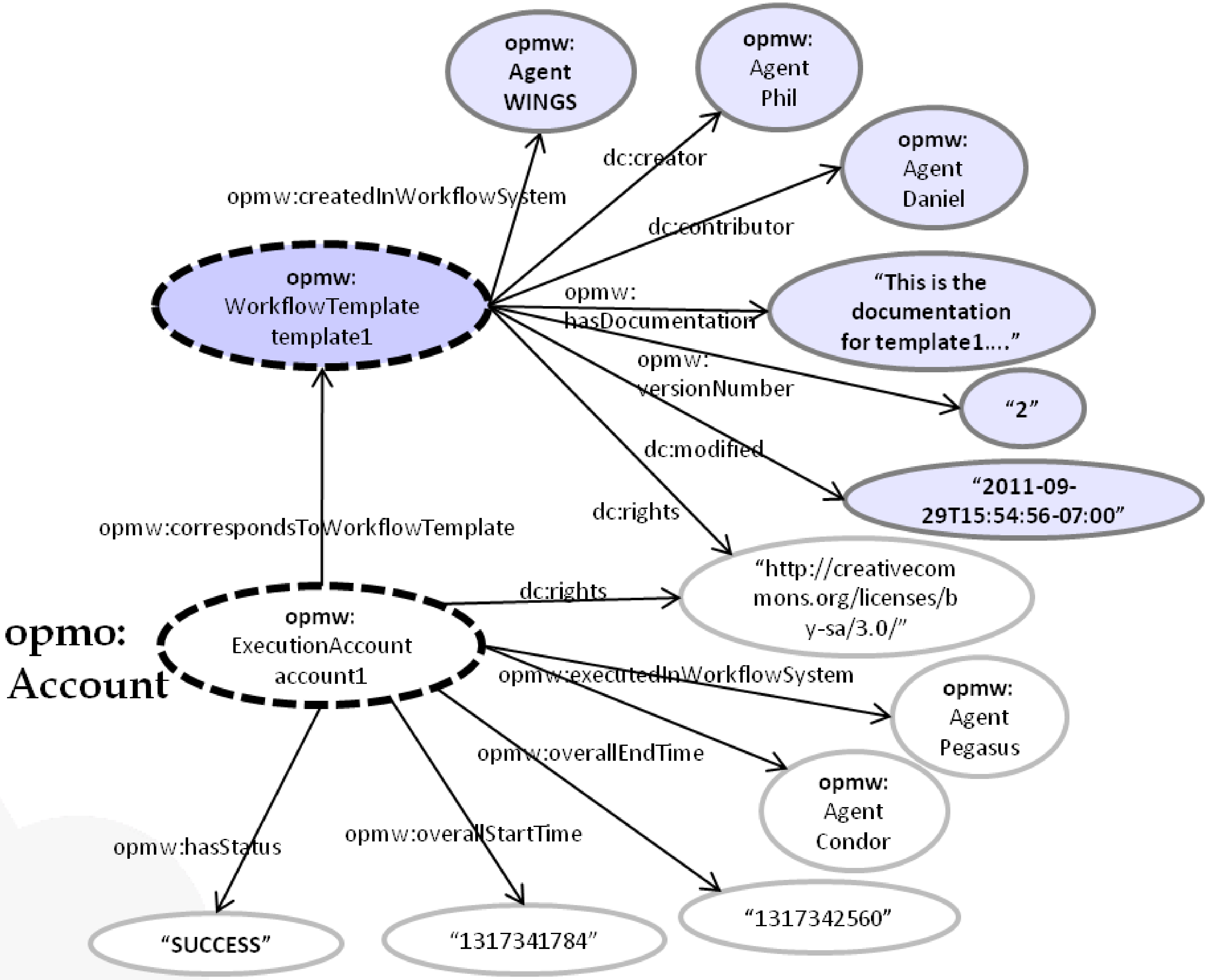
High level architecture



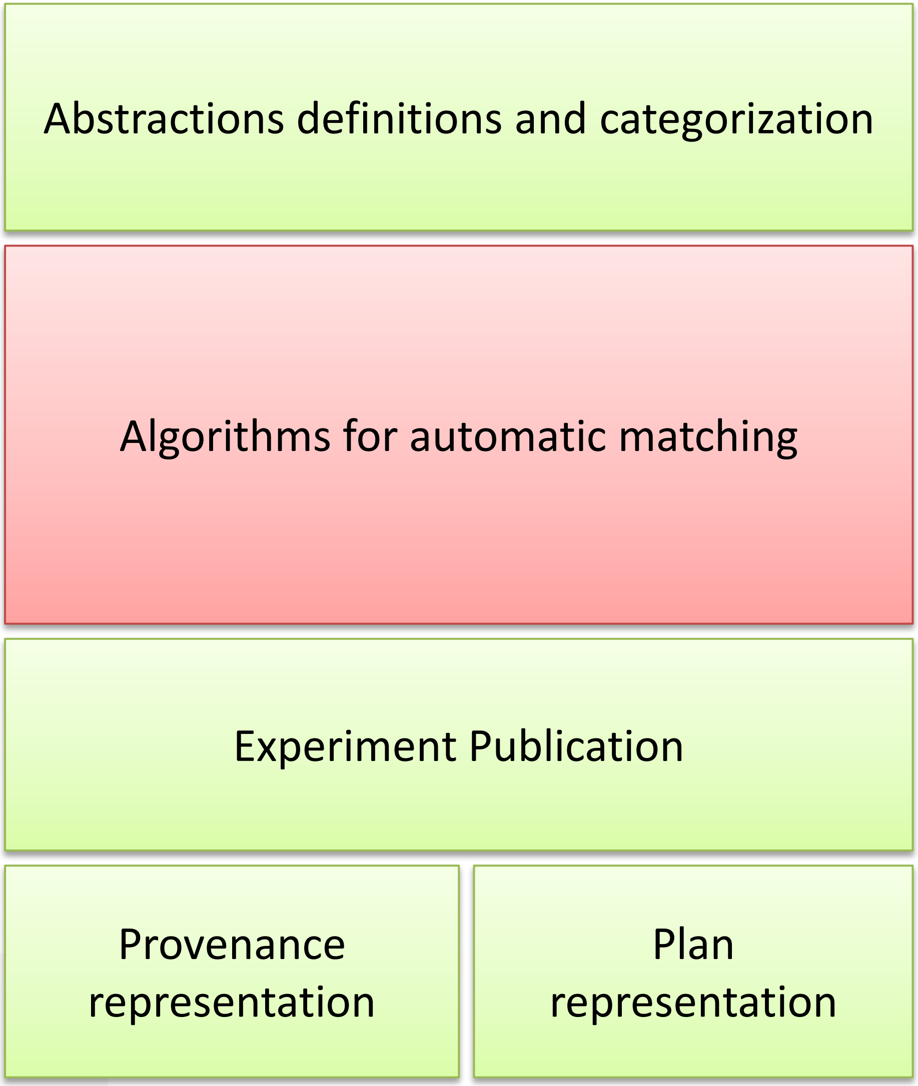
OPMW: Process view



OPMW: Attribution view



Work previous to second visit to ISI



Abstractions definitions and categorization

Algorithms for automatic matching

Experiment Publication

Provenance representation

Plan representation

Motif Detection

Virtuoso,
Pubby,
Wings (+Plugin)

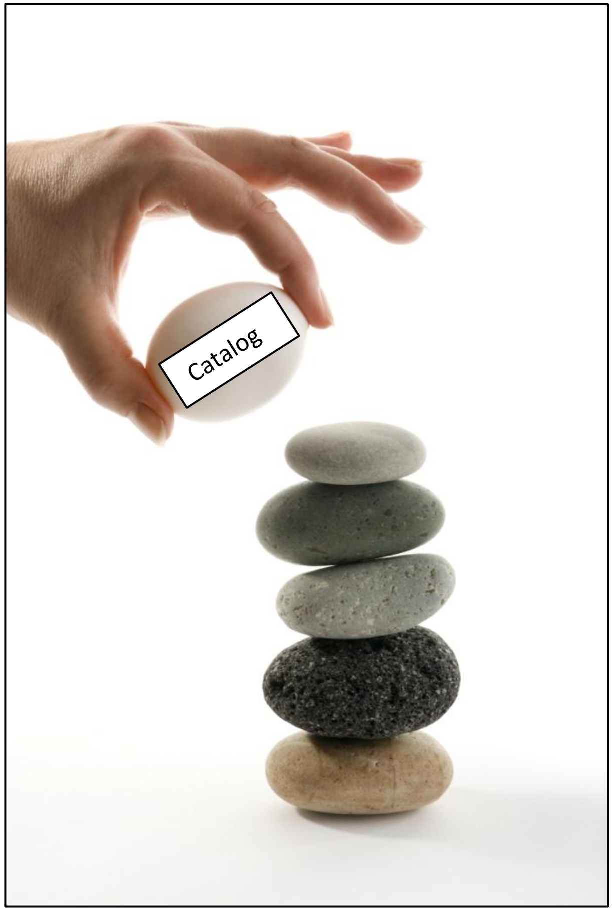
OPMW

Overview

- Empirical analysis on 177 workflow templates from Taverna and Wings

- Catalog of recurring patterns: scientific workflow *motifs*.
  - Data Oriented Motifs
  - Workflow Oriented Motifs

- Understandability and reuse

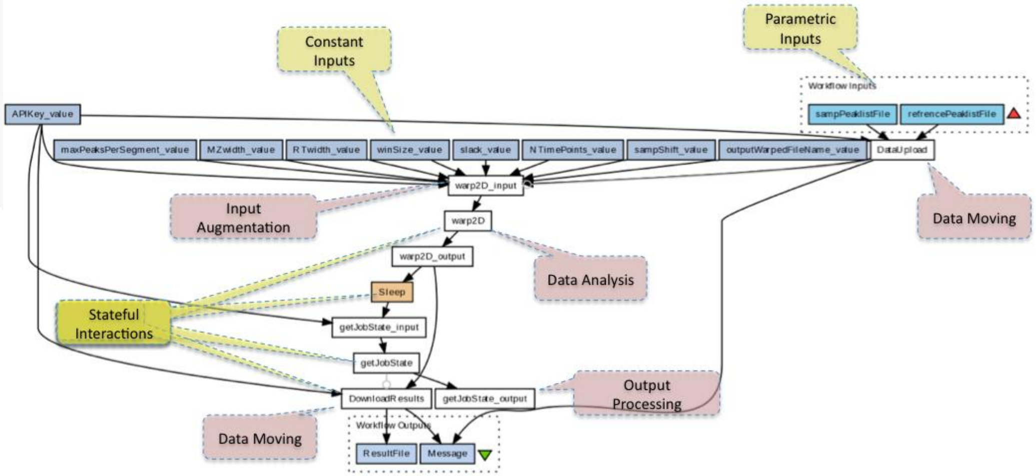


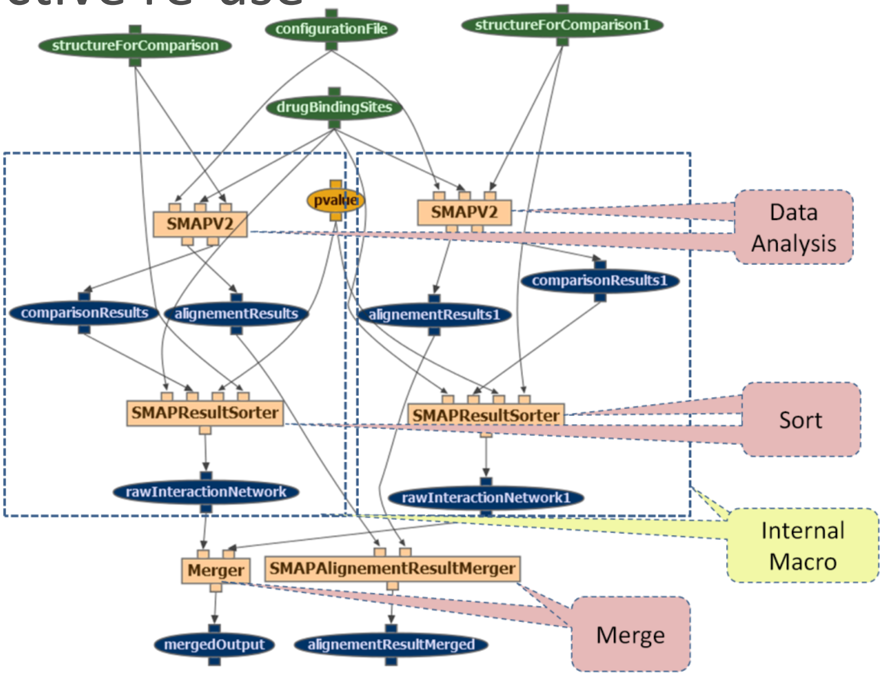
http://sensefinancial.com/wp-content/uploads/2012/02/contribution.jpg

Approach

- Reverse-engineer the set of current practices in workflow development through an analysis of empirical evidence

- Identify workflow abstractions that would facilitate understandability and therefore effective re-use





Workflow Motifs

- Workflow motif: Domain independent conceptual abstraction on the workflow steps.

1. Data-oriented motifs: What kind of manipulations does the workflow have?

   - E.g.:
     - Data retrieval
     - Data preparation
     - etc.

WHAT?

2. Workflow-oriented motifs: How does the workflow perform its operations?

   - E.g.:
     - Stateful steps
     - Stateless steps
     - Human interactions
     - etc.

HOW?

Motif Catalog

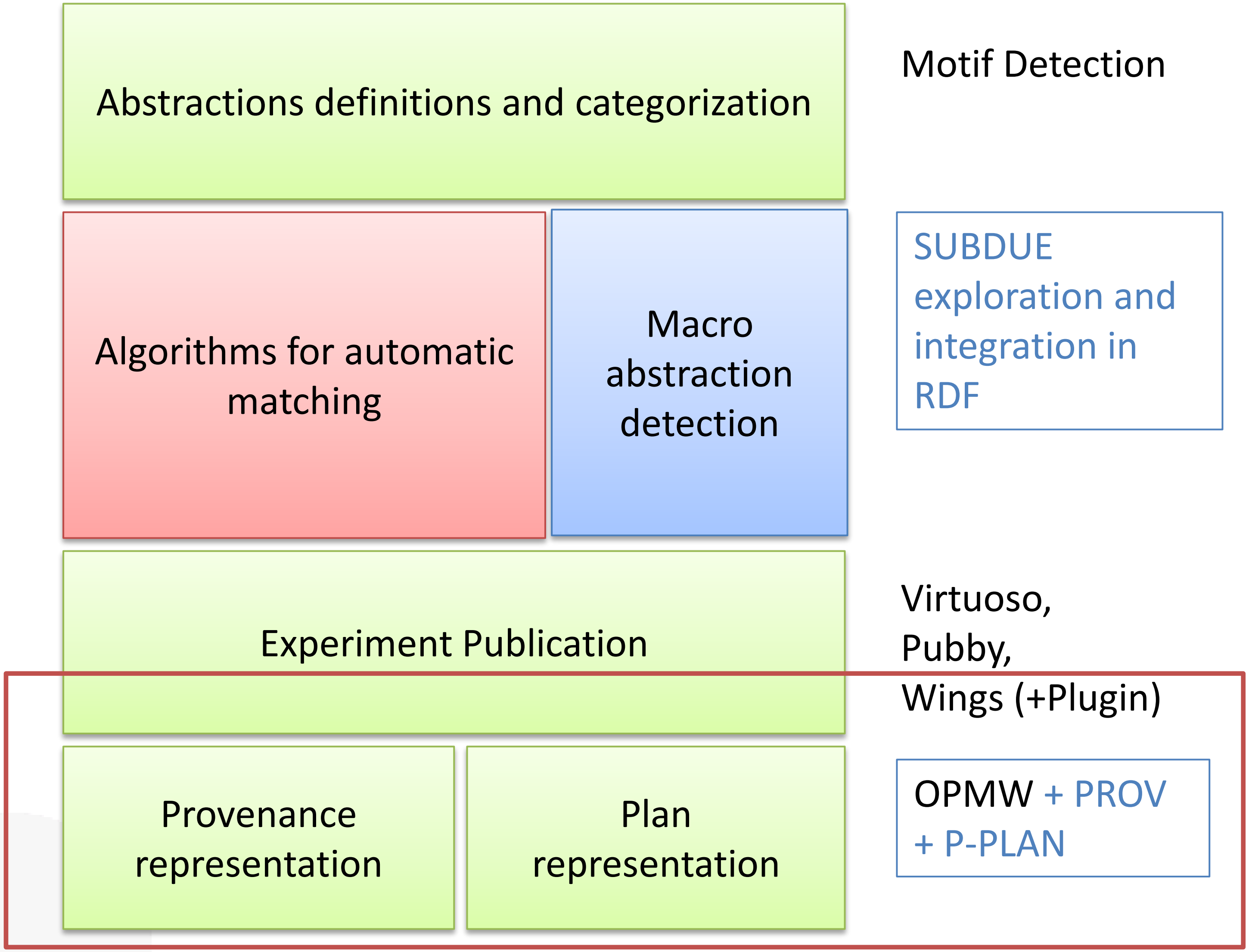
Data-Oriented Motifs

- Data Retrieval
- Data Preparation
  - Format Transformation
  - Input Augmentation and Output Splitting
  - Data Organisation
- Data Analysis
- Data Curation/Cleaning
- Data Moving
- Data Visualisation

Workflow-Oriented Motifs

- Intra-Workflow Motifs
  - Stateful (Asynchronous) Invocations
  - Stateless (Synchronous) Invocations
  - Internal Macros
  - Human Interactions
- Inter-Workflow Motifs
  - Atomic Workflows
  - Composite Workflows
  - Workflow Overloading

Summary: Work done at ISI



Abstractions definitions and categorization

Motif Detection

Algorithms for automatic matching

Macro abstraction detection

SUBDUE exploration and integration in RDF

Experiment Publication

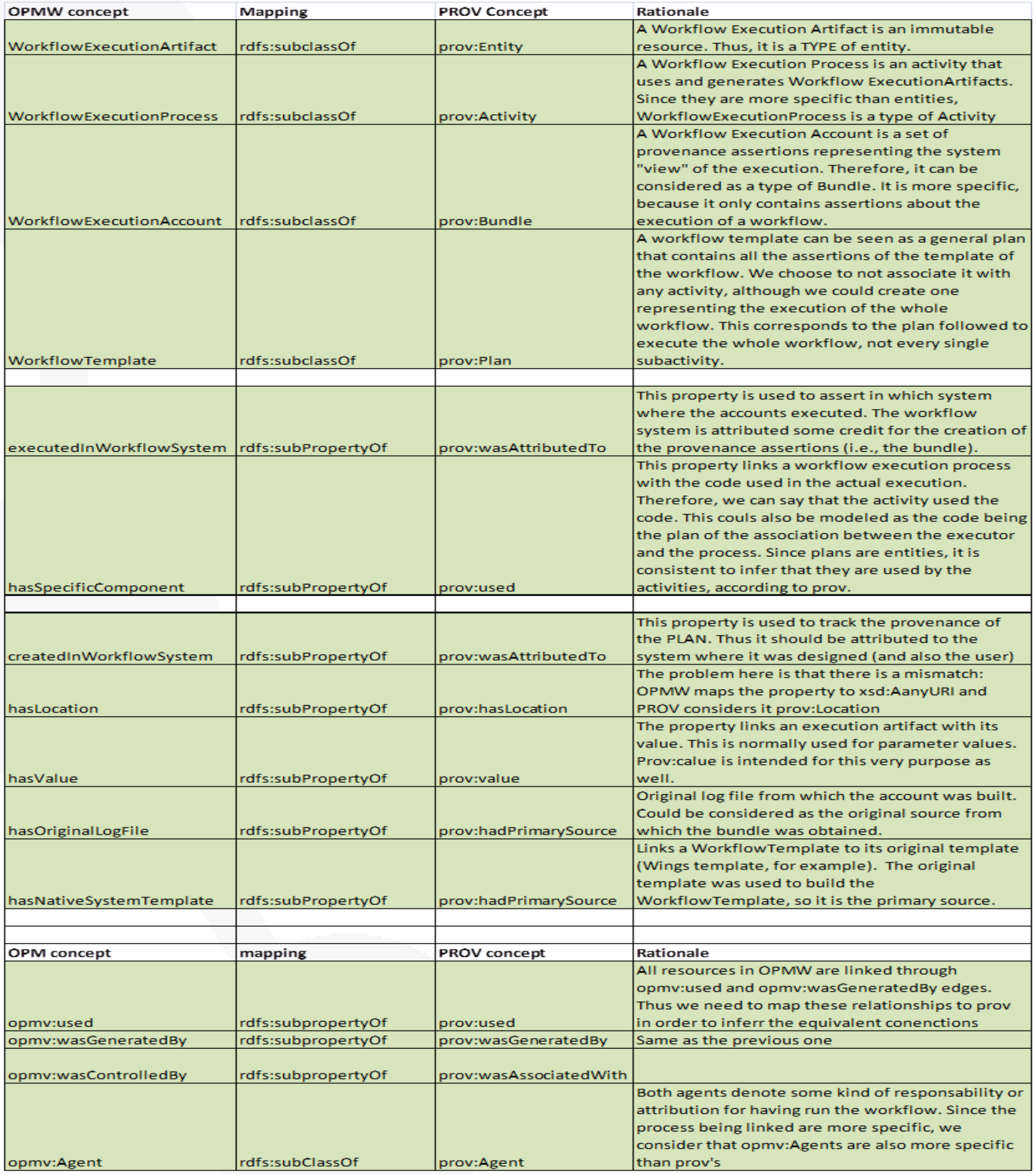
Virtuoso,
Pubby,
Wings (+Plugin)

Provenance representation

Plan representation

OPMW + PROV + P-PLAN

PROV Compatibility



OPMW concept	Mapping	PROV Concept	Rationale
WorkflowExecutionArtifact	rdfs:subclassOf	prov:Entity	A Workflow Execution Artifact is an immutable resource. Thus, it is a TYPE of entity.
WorkflowExecutionProcess	rdfs:subclassOf	prov:Activity	A Workflow Execution Process is an activity that uses and generates Workflow ExecutionArtifacts. Since they are more specific than entities, WorkflowExecutionProcess is a type of Activity
WorkflowExecutionAccount	rdfs:subclassOf	prov:Bundle	A Workflow Execution Account is a set of provenance assertions representing the system "view" of the execution. Therefore, it can be considered as a type of Bundle. It is more specific, because it only contains assertions about the execution of a workflow.
WorkflowTemplate	rdfs:subclassOf	prov:Plan	A workflow template can be seen as a general plan that contains all the assertions of the template of the workflow. We choose to not associate it with any activity, although we could create one representing the execution of the whole workflow. This corresponds to the plan followed to execute the whole workflow, not every single subactivity.
executedInWorkflowSystem	rdfs:subPropertyOf	prov:wasAttributedTo	This property is used to assert in which system where the accounts executed. The workflow system is attributed some credit for the creation of the provenance assertions (i.e., the bundle).
hasSpecificComponent	rdfs:subPropertyOf	prov:used	This property links a workflow execution process with the code used in the actual execution. Therefore, we can say that the activity used the code. This couls also be modeled as the code being the plan of the association between the executor and the process. Since plans are entities, it is consistent to infer that they are used by the activities, according to prov.
createdInWorkflowSystem	rdfs:subPropertyOf	prov:wasAttributedTo	This property is used to track the provenance of the PLAN. Thus it should be attributed to the system where it was designed (and also the user)
hasLocation	rdfs:subPropertyOf	prov:hasLocation	The problem here is that there is a mismatch: OPMW maps the property to xsd:AanyURI and PROV considers it prov:Location
hasValue	rdfs:subPropertyOf	prov:value	The property links an execution artifact with its value. This is normally used for parameter values. Prov:calue is intended for this very purpose as well.
hasOriginalLogFile	rdfs:subPropertyOf	prov:hadPrimarySource	Original log file from which the account was built. Could be considered as the original source from which the bundle was obtained.
hasNativeSystemTemplate	rdfs:subPropertyOf	prov:hadPrimarySource	Links a WorkflowTemplate to its original template (Wings template, for example). The original template was used to build the WorkflowTemplate, so it is the primary source.
OPM concept	**mapping**	**PROV concept**	**Rationale**
opmv:used	rdfs:subpropertyOf	prov:used	All resources in OPMW are linked through opmv:used and opmv:wasGeneratedBy edges. Thus we need to map these relationships to prov in order to inferr the equivalent conenctions
opmv:wasGeneratedBy	rdfs:subpropertyOf	prov:wasGeneratedBy	Same as the previous one
opmv:wasControlledBy	rdfs:subpropertyOf	prov:wasAssociatedWith	
opmv:Agent	rdfs:subClassOf	prov:Agent	Both agents denote some kind of responsability or attribution for having run the workflow. Since the process being linked are more specific, we consider that opmv:Agents are also more specific than prov's

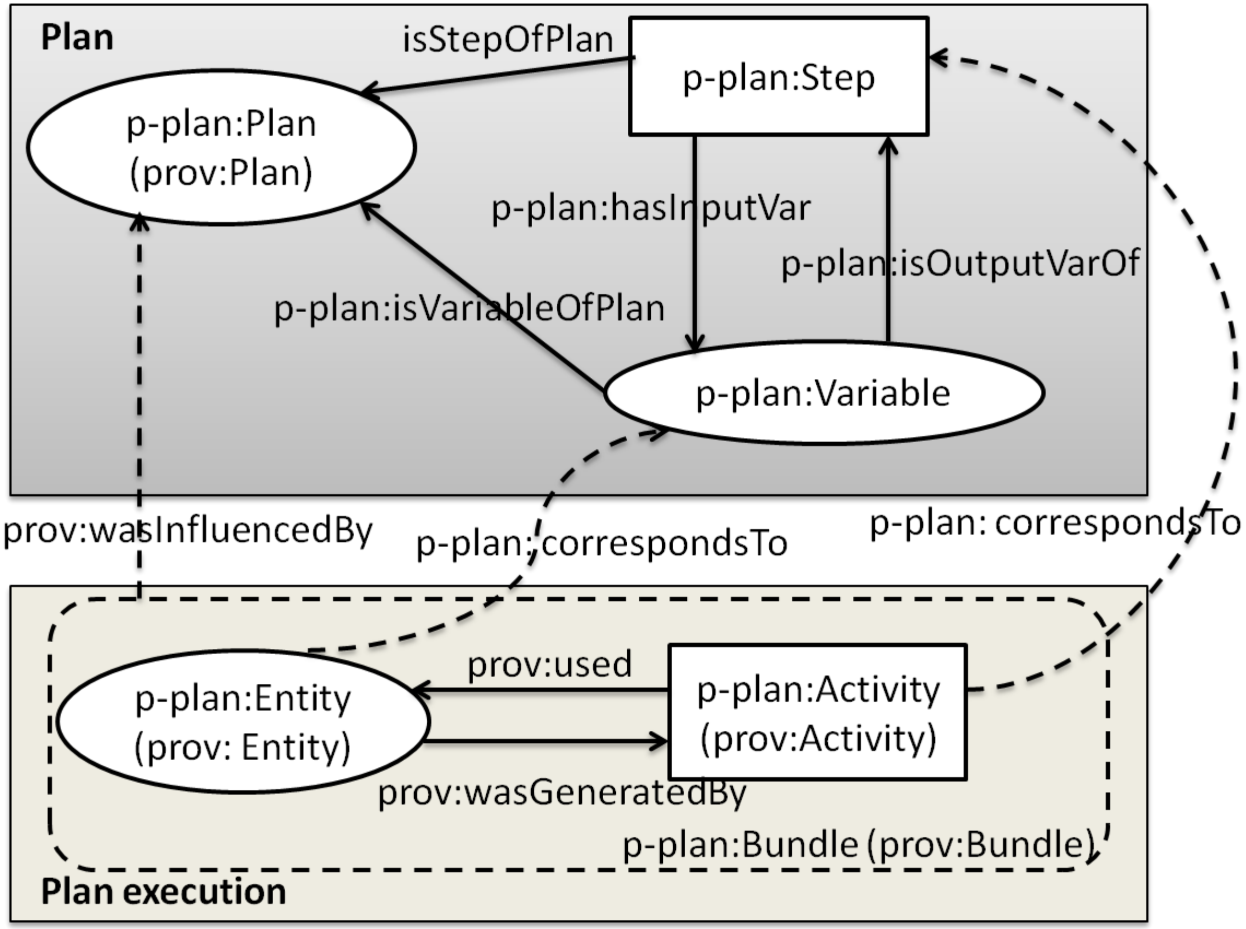
- OPMW fits naturally into PROV
  - Same usage-generation structure
  - Extension for the scientific workflow with PROV

- Binary relationships (no n-ary patterns used).
  - Simplicity

- Publication of PROV as well as OPMW.
  - Queries can be answered in both languages.
  - Flexibility.

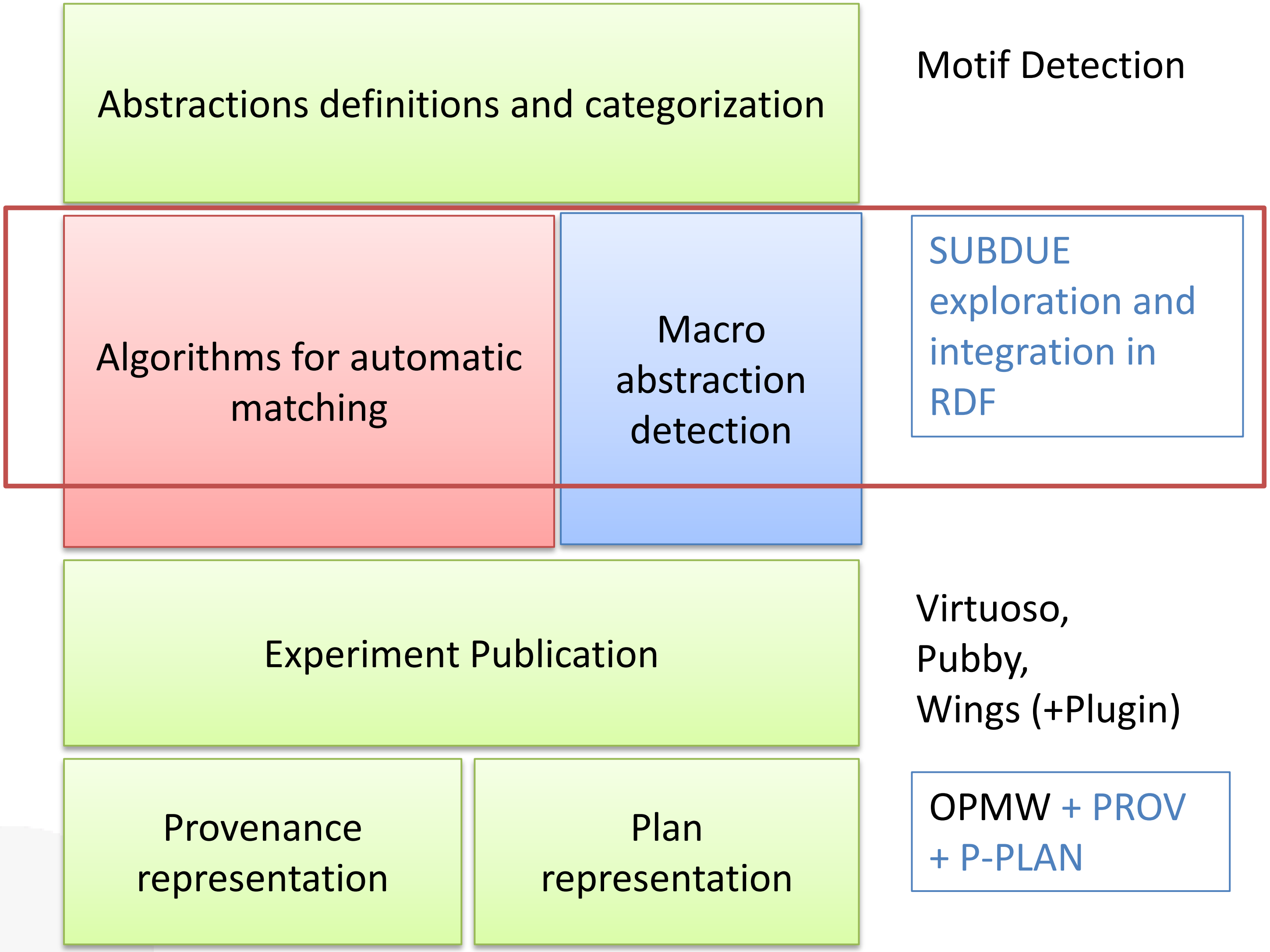
- http://www.opmw.org/node/8

P-PLAN



- Plans are not provenance
- P-PLAN: Simple plan model for binding traces to template representations
- Aligned with OPMW and PROV
- Documentation in progress

Summary: Work done at ISI



Abstractions definitions and categorization

Motif Detection

Algorithms for automatic matching

Macro abstraction detection

SUBDUE exploration and integration in RDF

Experiment Publication

Virtuoso,
Pubby,
Wings (+Plugin)

Provenance representation

Plan representation

OPMW + PROV + P-PLAN

Macro abstraction detection

Problem statement:

Given a ***repository of workflow templates (either abstract or specific) or workflow execution traces, what are the workflow fragments I can deduce from it?***

Useful for:

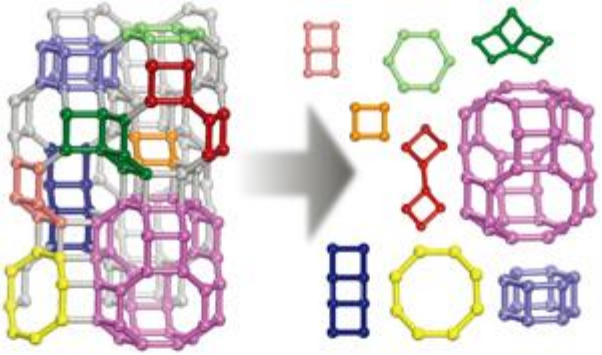
•Systems like Taverna and Wings: (Many templates, little annotation to relate them)

  •Finding relationships between workflows and sub-workflows.

    •Most used fragments, most executed, etc.

  •Systems like GenePattern and Galaxy: (Many runs, nearly no templates published)

    •Proposing new templates with the popular fragments.



Macro abstraction detection

- Work in Progress (implementation and evaluation)
  - WINGS traces

- Similar to Sub-graph Isomorphism
- Kind of “Graph Clustering”

- Early results
  - Tool for finding common sub-graphs
    - Sequential graphs
    - Efficient
    - Scalable.

  - Integration with RDF (by me)

- TO DO:
  - Finish implementation: inference.
  - Evaluation!!

Next Steps

Next Steps

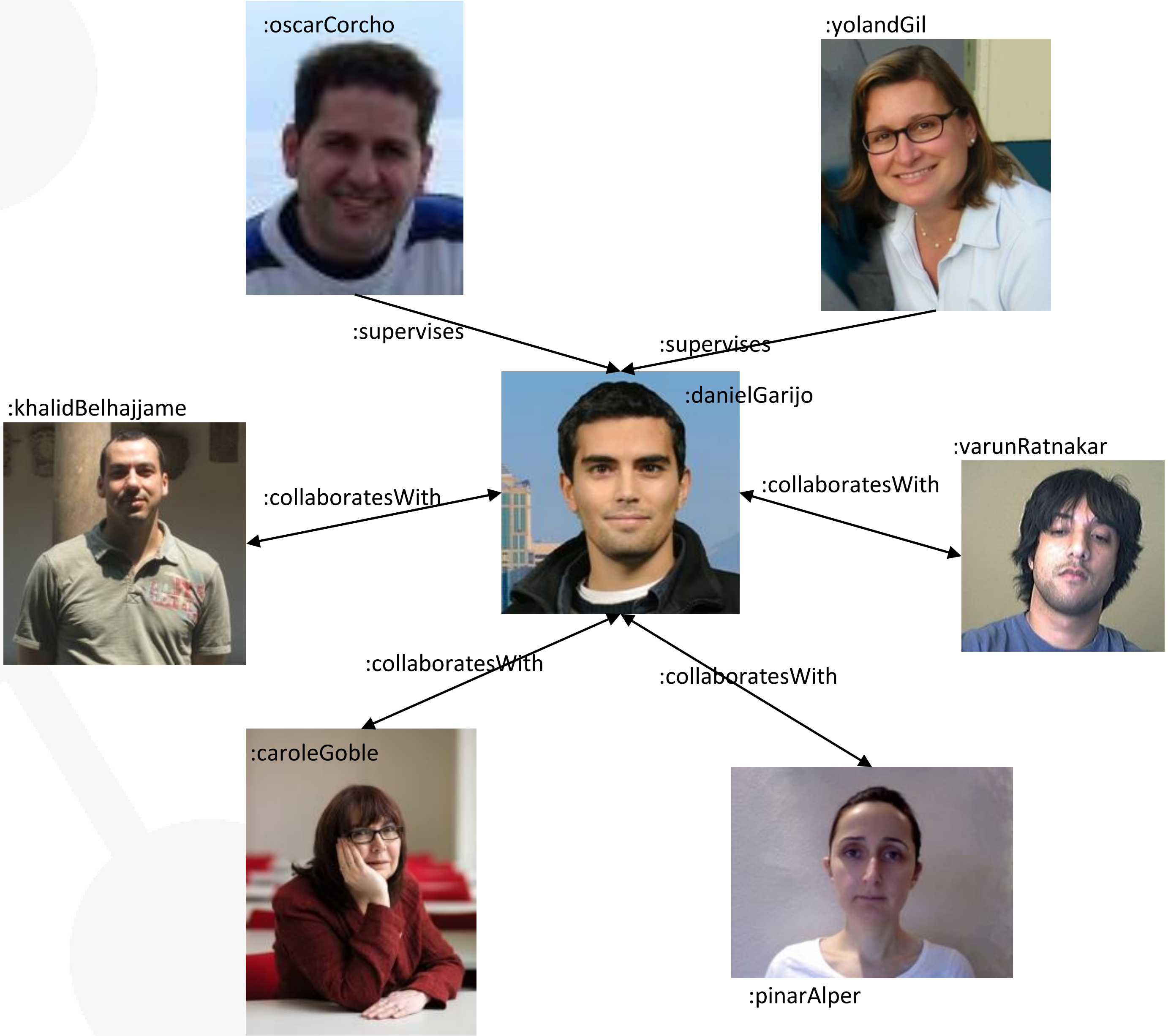
- Thesis:
  - Finish up implementation.
  - How to evaluate results?

- Publications:
  - Workshop:
    - Provenance Corpus (with Taverna Team). To have something citable
  - Conference:
    - KCAP: Macro detection implementation and evaluation.
  - Journal
    - Decay analysis publication in journal (January)
    - OPMW - PROV -P-PLAN publication in journal (December)
    - Motif extension publication in journal (Invited by special issue) (Now)

Future work

- Thesis:
  - Other methods for detecting workflow abstraction automatically
    - Metadata and file analysis (diff, etc.): Filter, merge, etc.
    - Provenance reconstruction.
- Project:
  - RO model specifications
  - Testcases
  - Workflow abstraction with Isoco

Thanks !



:oscarCorcho

:yolandGil

:supervises

:supervises

:danielGarijo

:khalidBelhajjame

:collaboratesWith

:varunRatnakar

:collaboratesWith

:collaboratesWith

:collaboratesWith

:caroleGoble

:pinarAlper

Work at ISI, relation with wf4Ever, future steps

Daniel Garijo Verdejo

Ontology Engineering Group. Laboratorio de Inteligencia Artificial
Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid

Date: 03/10/2011