

Janus
Provenance

Workflows for Information Integration in the Life Sciences

Part 2: Workflows in context

Paolo Missier

Information Management Group

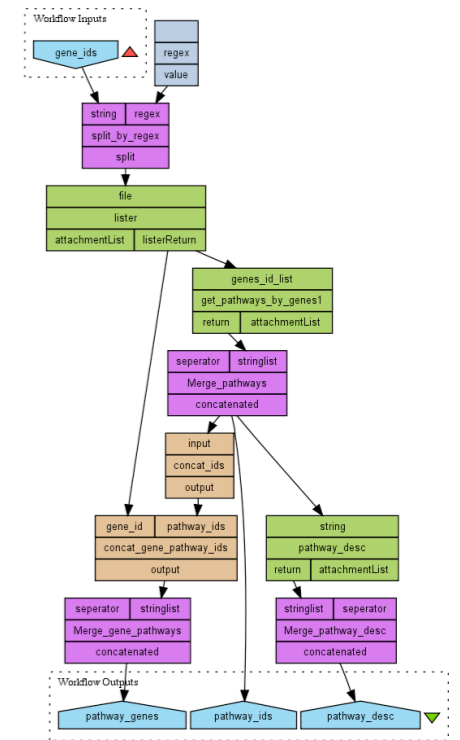
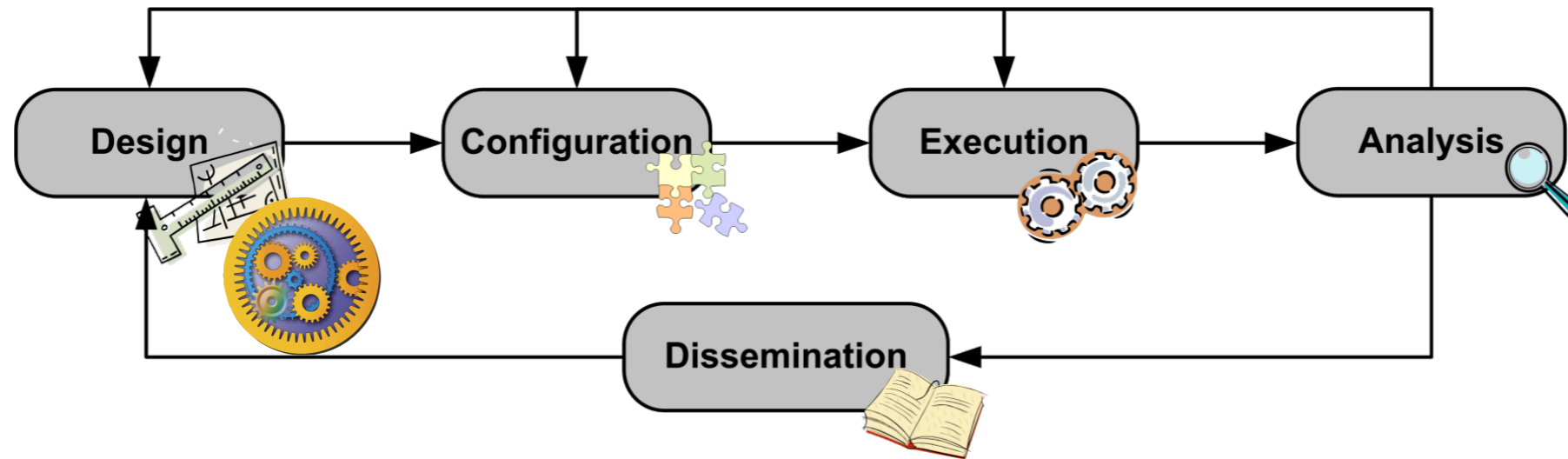
School of Computer Science, University of Manchester, UK

Search Computing workshop

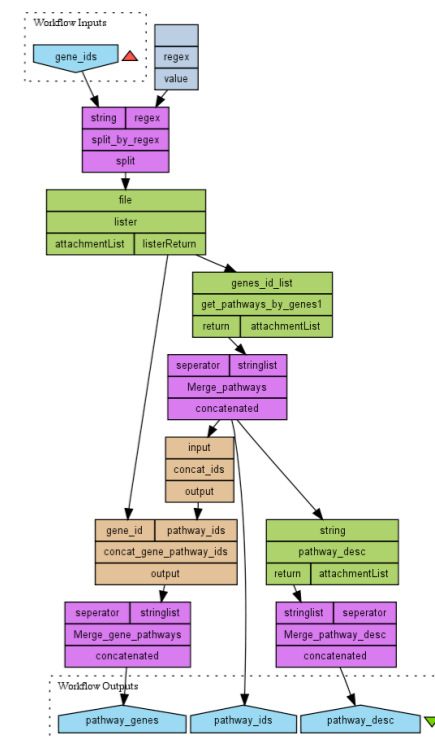
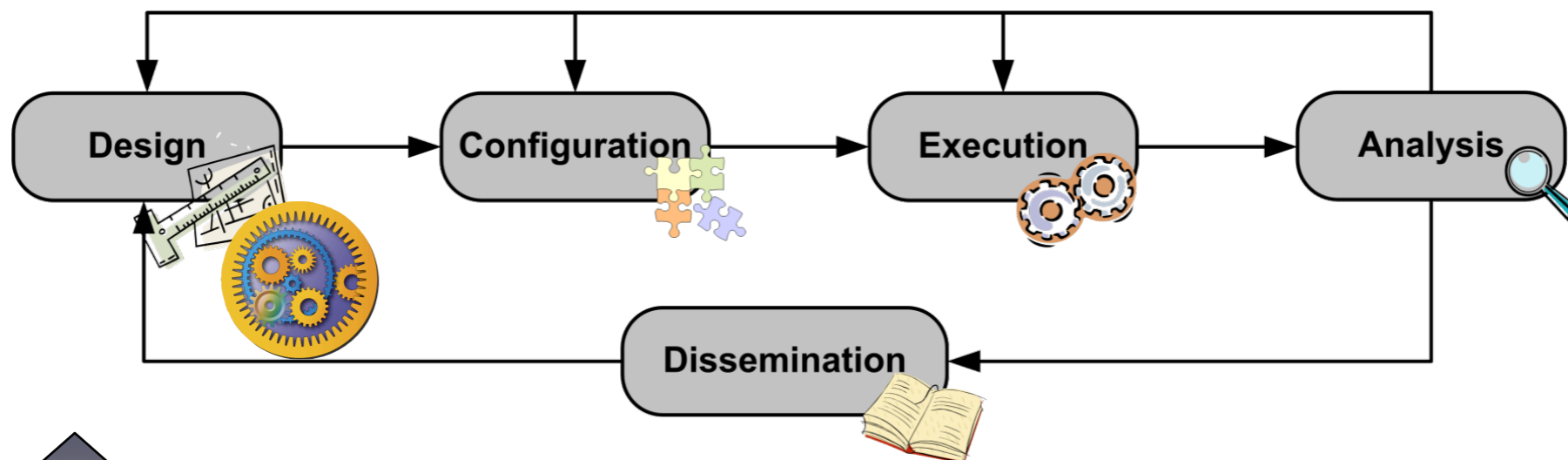
Como, Italy, May 28th, 2010

- In the first part we have seen a case study in system biology
 - showcasing scientific workflows at work
- Two key questions:
 1. Workflows live within an **eco-system of models, tools and technologies**. Can any of these benefit / complement the SeCo paradigm?
 2. What is the **relationship between a dataflow model (Taverna) and a SeCo query plan?**
- To inform the discussion, we focus of some elements of a **workflow's lifecycle**
 - importing services
 - benefits of domain-specific service collections
 - collecting and querying **provenance** traces

Process-centric science lifecycle




Process-centric science lifecycle

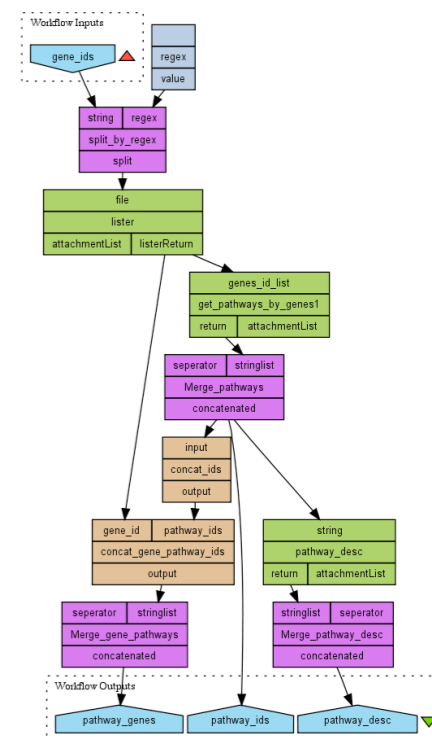
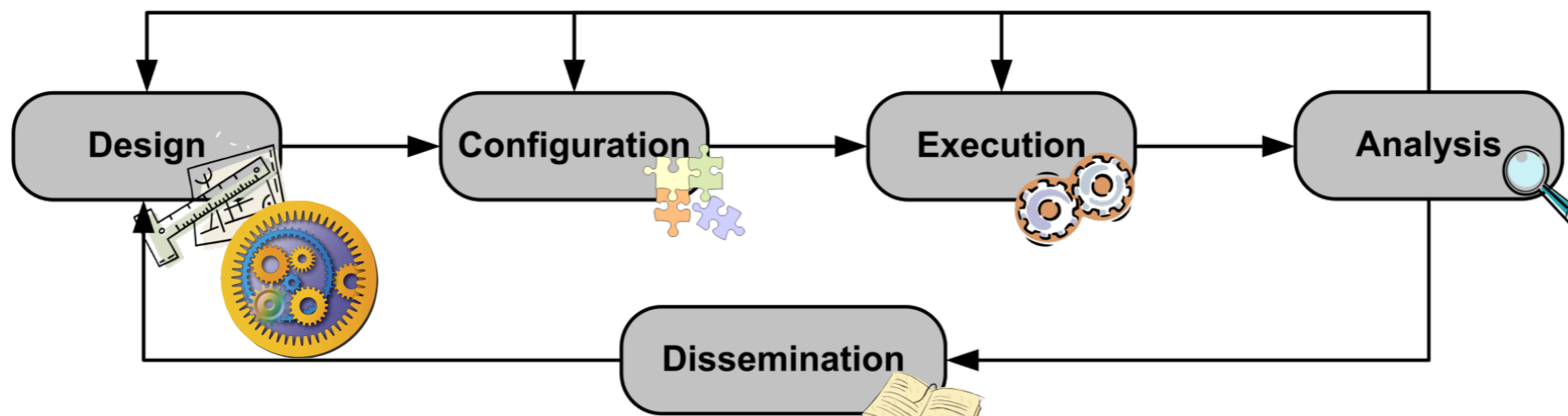


Service discovery
and import



BioCatalogue 

Process-centric science lifecycle



Service discovery and import

Data

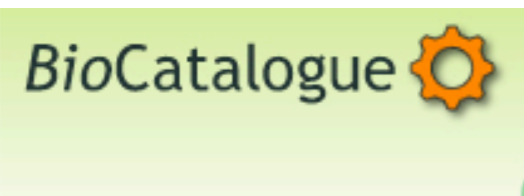
- inputs
- parameters
- results

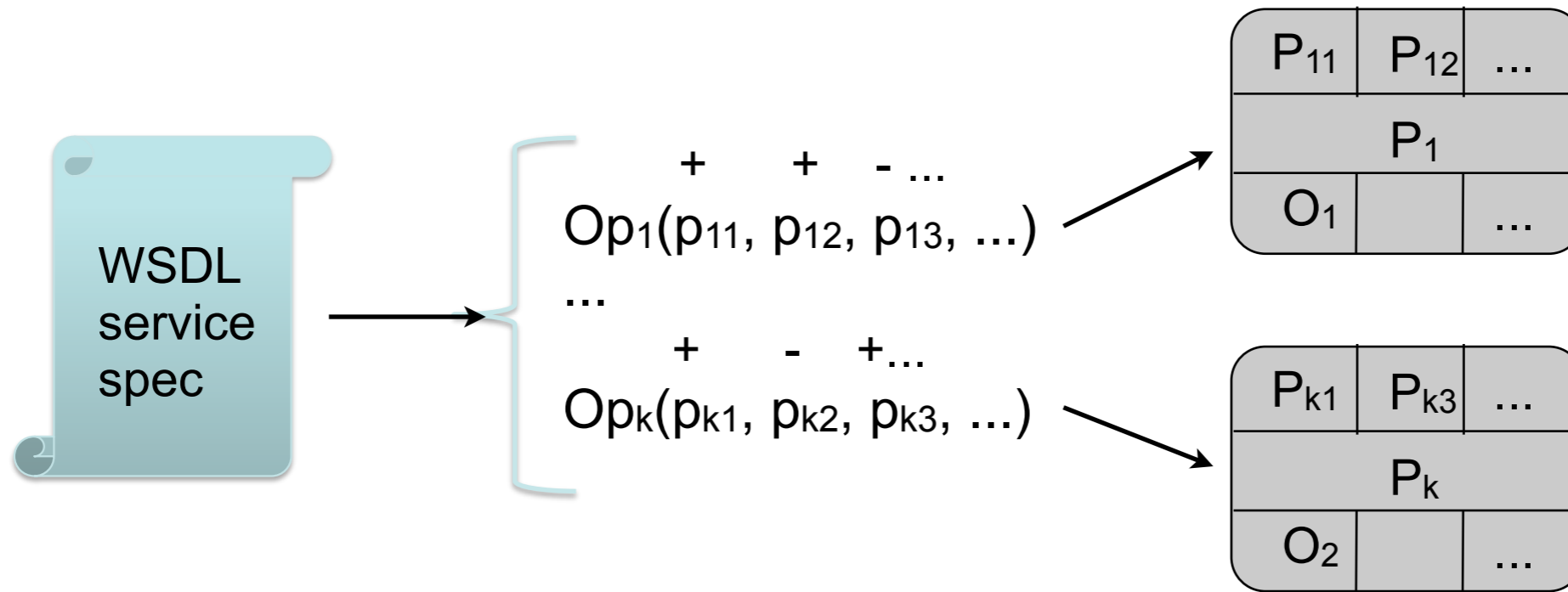
Metadata

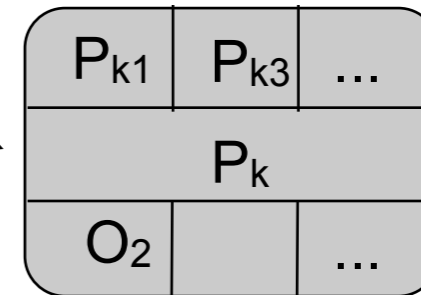
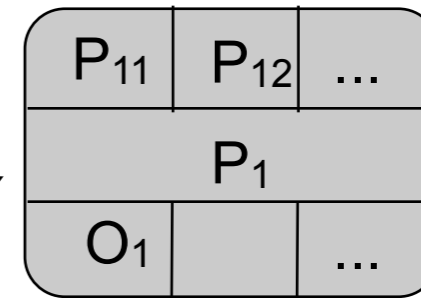
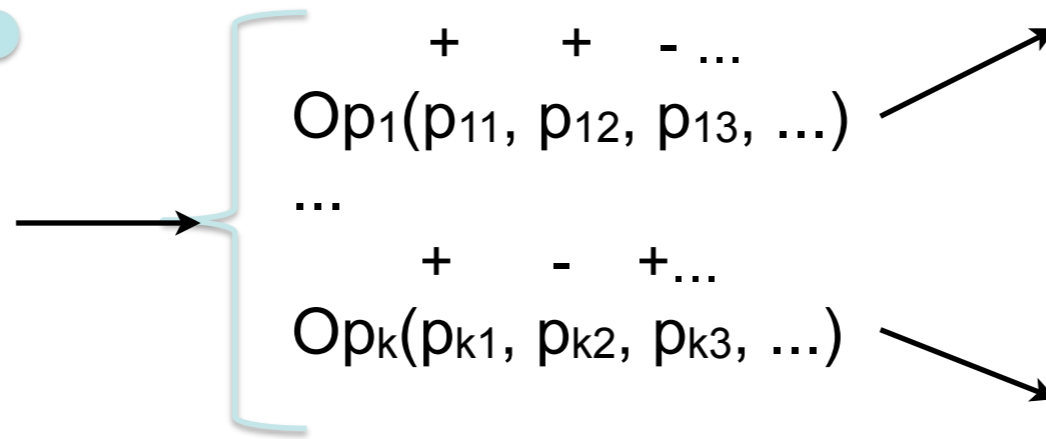
- provenance
- annotations

Methods

- the workflow







Home » Services » sabiorc

aka Reactions kinetic database

Categories: Systems Biology Pathways Ligand Interaction

Overview Operations (54) Monitoring

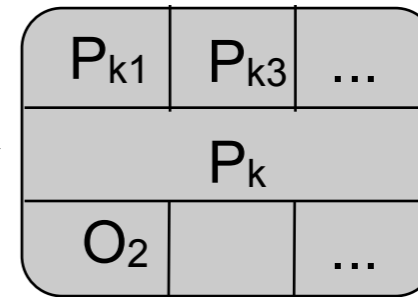
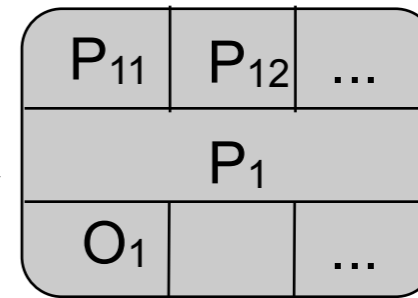
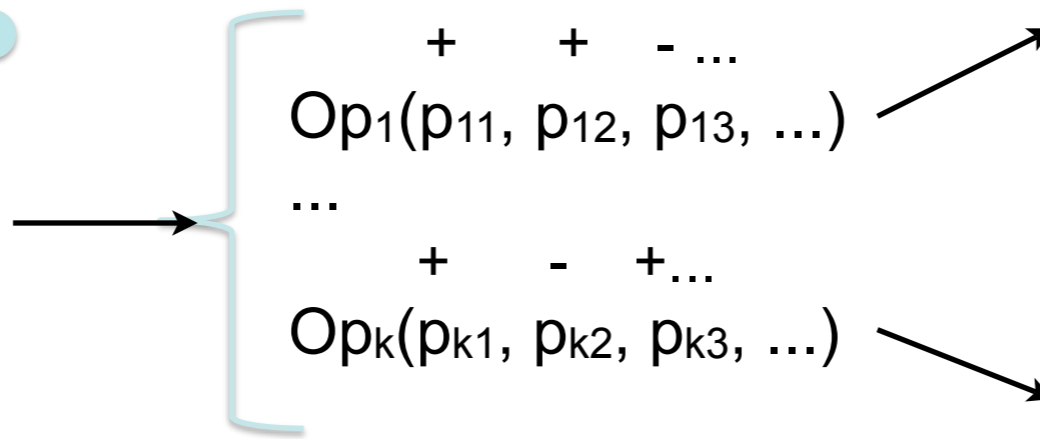
These are the SOAP operations available for the SOAP variant of this service. Click on each one to get more information.

Quick Browse | [getActivatorsSpeciesIDs](#) | [getAllCompoundIDs](#) | [getAllEnzymes](#) | [getAllPathways](#) | [getCHEBIID](#) | [getCofactorsSpeciesIDs](#) | [getCompoundID](#) | [getCompoundIDFromCHEBIID](#) | [getECByName](#) | [getECFromReactionID](#) | [getEnzymeProtein](#) | [getEnzymeVariant](#) | [getExpCor](#) | [getKEGGReactionID](#) | [getKineticLaw](#) | [getKinLawIDFromPubmed](#) | [getKinLawIDs](#) | [getKinLa](#) | [getNormalizedParametersXML](#) | [getOrganismFromKLID](#) | [getParametersXML](#) | [getPathwayName](#) | [getReactionIDByKEGG](#) | [getReactionIDFromCompound](#) | [getReactionIDFromEC](#) | [getReactionIDFromKEGG](#) | [getReactionInstanceIDs](#) | [getReactionInstanceIDsFromProtein](#) | [getReactionInstancesFromUniprot](#) | [getUnknownModifiersSpeciesIDs](#) | [searchCompounds](#) | [searchEnzymesByECNumber](#) | [searchPathways](#)

getActivatorsSpeciesIDs
Part of Service: [sabiorc](#)
1 input | 1 output
Accepts a reaction instance ID and returns the corresponding array of species IDs for the activators of that reaction.
Tags on this SOAP Operation: [activator](#) | [metabolic reaction](#) | [reaction activator](#)

getAllCompoundIDs
Part of Service: [sabiorc](#)
1 input | 1 output
Returns all Compound IDs stored in SABIO-RK

WSDL to Taverna processors



Service panel

Home » Services » sabiorc

sabiorc SOAP

aka Reactions kinetic database

Categories: Systems Biology Pathways Ligand Interaction

Overview Operations (54) Monitoring

These are the SOAP operations available for the SOAP variant of this service. Click on

Quick Browse | [getActivatorsSpeciesIDs](#) | [getAllCompoundIDs](#) | [getAllEnzyme](#)
[getCHEBIID](#) | [getCofactorsSpeciesIDs](#) | [getCompoundID](#) | [getCompoundID](#)
[getECByName](#) | [getECFromReactionID](#) | [getEnzymeProtein](#) | [getEnzymeVa](#)
[getKEGGReactionID](#) | [getKineticLaw](#) | [getKinLawIDFromPubmed](#) | [getKinL](#)
[getNormalizedParametersXML](#) | [getOrganismFromKLID](#) | [getParametersXML](#)
[getReactionIDByKEGG](#) | [getReactionIDFromCompound](#) | [getReactionIDFromE](#)
[getReactionInstanceIDs](#) | [getReactionInstanceIDsFromProtein](#) | [getReactionIn](#)
[getUnknownModifiersSpeciesIDs](#) | [searchCompounds](#) | [searchEnzymesByECI](#)

getActivatorsSpeciesIDs

Part of Service: [sabiorc](#)

1 input | 1 output

Accepts a reaction instance ID and returns the corresponding array of species IDs for the activators

Tags on this SOAP Operation: [activator](#) | [metabolic reaction](#) | [reaction activator](#)

getAllCompoundIDs

Part of Service: [sabiorc](#)

1 input | 1 output

Returns all Compound IDs stored in SABIO-RK

Filter:

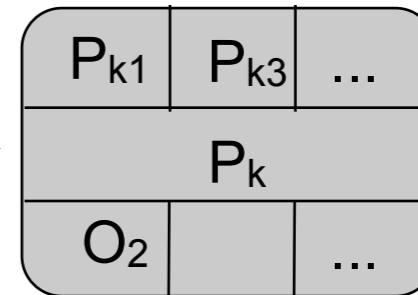
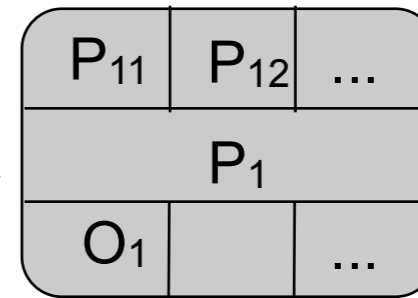
Import new services

- ▶ [Biomart @ http://www.biomart.org/biomart/martservice](http://www.biomart.org/biomart/martservice)
- ▶ [Biomoby @ http://moby.ucalgary.ca/moby/MOBY-Central.pl](http://moby.ucalgary.ca/moby/MOBY-Central.pl)
- ▶ [Soaplab @ http://www.ebi.ac.uk/soaplab/services/](http://www.ebi.ac.uk/soaplab/services/)
- ▶ [WSDL @ http://eutils.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils.wsdl](http://eutils.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils.wsdl)
- ▼ [WSDL @ http://sabio.villa-bosch.de/sabiorc?wsdl](http://sabio.villa-bosch.de/sabiorc?wsdl)
 - ⚙ [getActivatorsSpeciesIDs](#)
 - ⚙ [getAllCompoundIDs](#)
 - ⚙ [getAllEnzymes](#)
 - ⚙ [getAllPathways](#)
 - ⚙ [getAllReactionIDs](#)
 - ⚙ [getAllUniProtIDs](#)
 - ⚙ [getAllUnits](#)
 - ⚙ [getCatalystsSpeciesIDs](#)
 - ⚙ [getCHEBIID](#)
 - ⚙ [getCofactorsSpeciesIDs](#)
 - ⚙ [...](#)

WSDL to Taverna processors



$Op_1(p_{11}, p_{12}, p_{13}, \dots)$
 \dots
 $Op_k(p_{k1}, p_{k2}, p_{k3}, \dots)$



Service panel

Home » Services » sabiorc

sabiorc SOAP

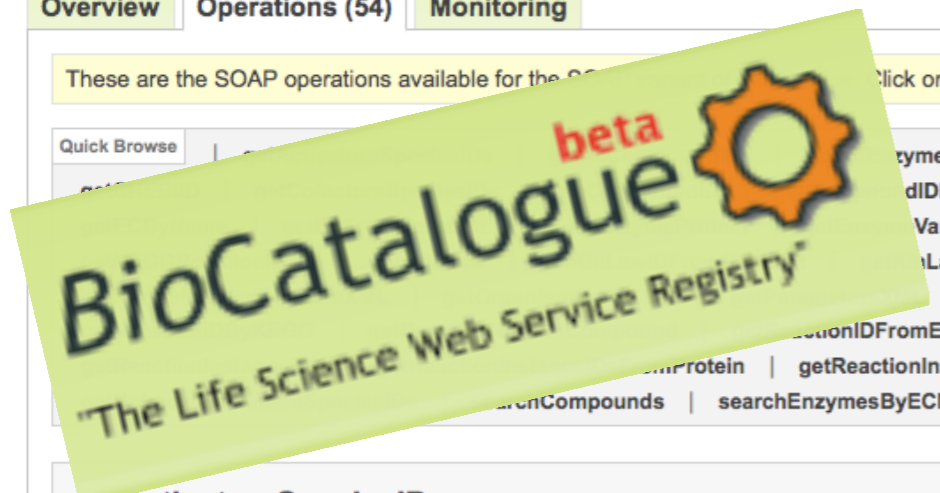
aka Reactions kinetic database

Categories: Systems Biology Pathways Ligand Interaction

Overview Operations (54) Monitoring

These are the SOAP operations available for the S...

Quick Browse



getActivatorsSpeciesIDs

Part of Service: [sabiorc](#)

1 input | 1 output

Accepts a reaction instance ID and returns the corresponding array of species IDs for the activators

Tags on this SOAP Operation: [activator](#) | [metabolic reaction](#) | [reaction activator](#)

getAllCompoundIDs

Part of Service: [sabiorc](#)

1 input | 1 output

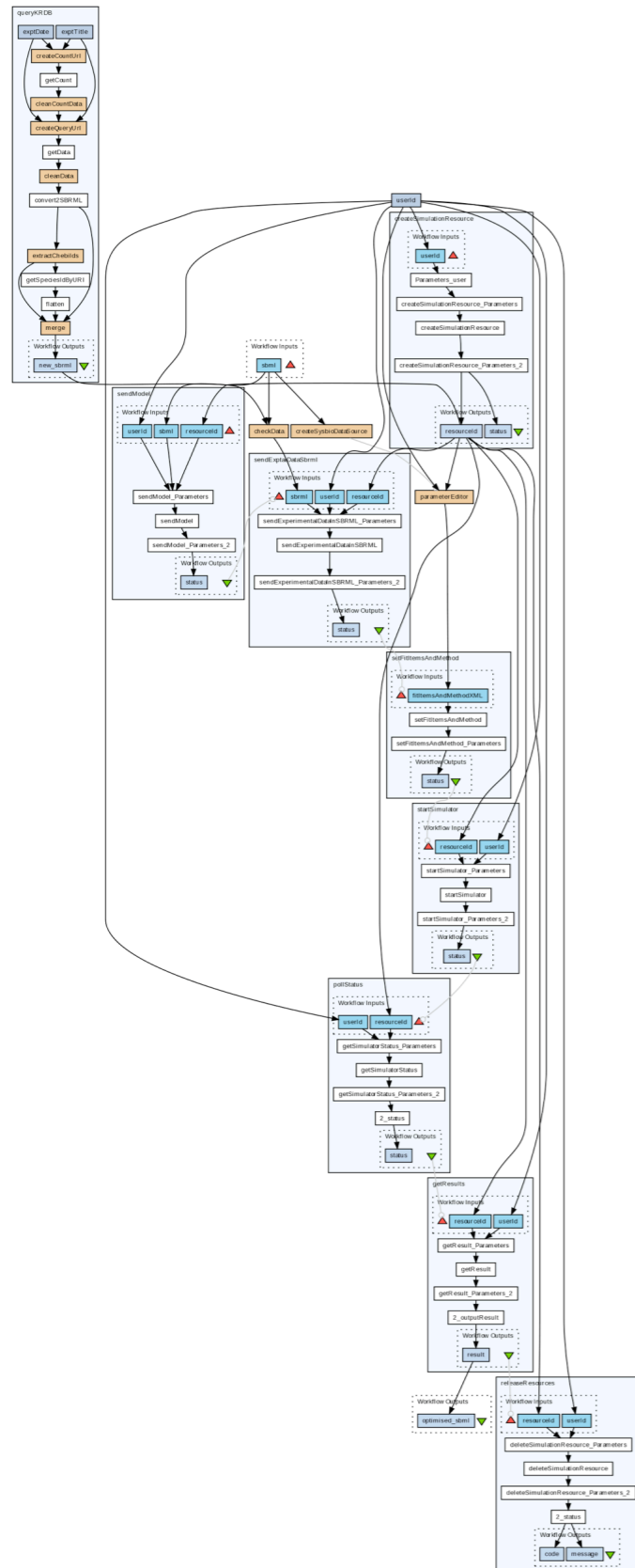
Returns all Compound IDs stored in SABIO-RK

Filter:

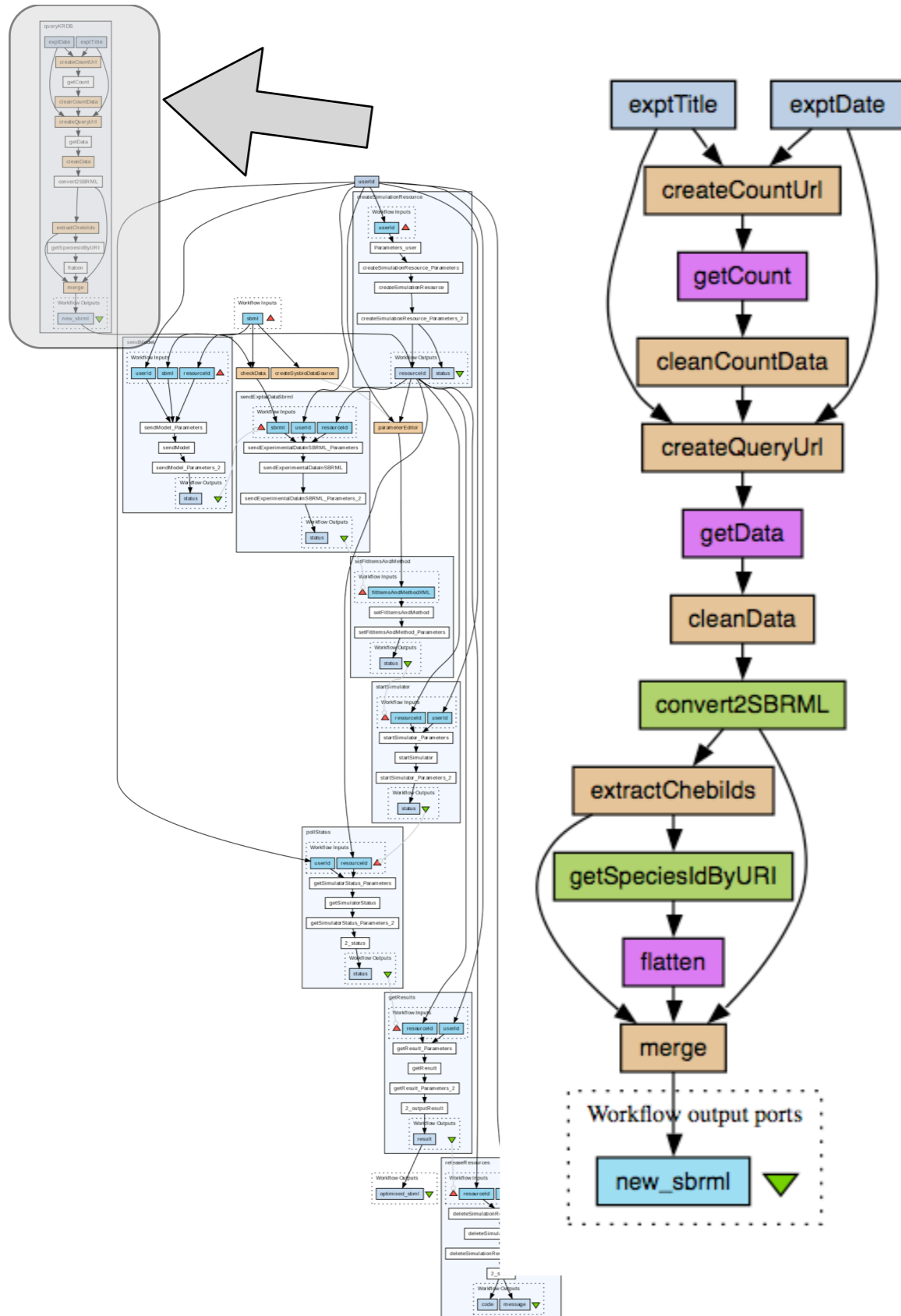
Import new services

- ▶ Biomart @ <http://www.biomart.org/biomart/martservice>
- ▶ Biomoby @ <http://moby.ucalgary.ca/moby/MOBY-Central.pl>
- ▶ Soaplab @ <http://www.ebi.ac.uk/soaplab/services/>
- ▶ WSDL @ <http://eutils.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils.wsdl>
- ▼ WSDL @ <http://sabio.villa-bosch.de/sabiorc?wsdl>
 - ⚙ getActivatorsSpeciesIDs
 - ⚙ getAllCompoundIDs
 - ⚙ getAllEnzymes
 - ⚙ getAllPathways
 - ⚙ getAllReactionIDs
 - ⚙ getAllUniProtIDs
 - ⚙ getAllUnits
 - ⚙ getCatalystsSpeciesIDs
 - ⚙ getCHEBIID
 - ⚙ getCofactorsSpeciesIDs

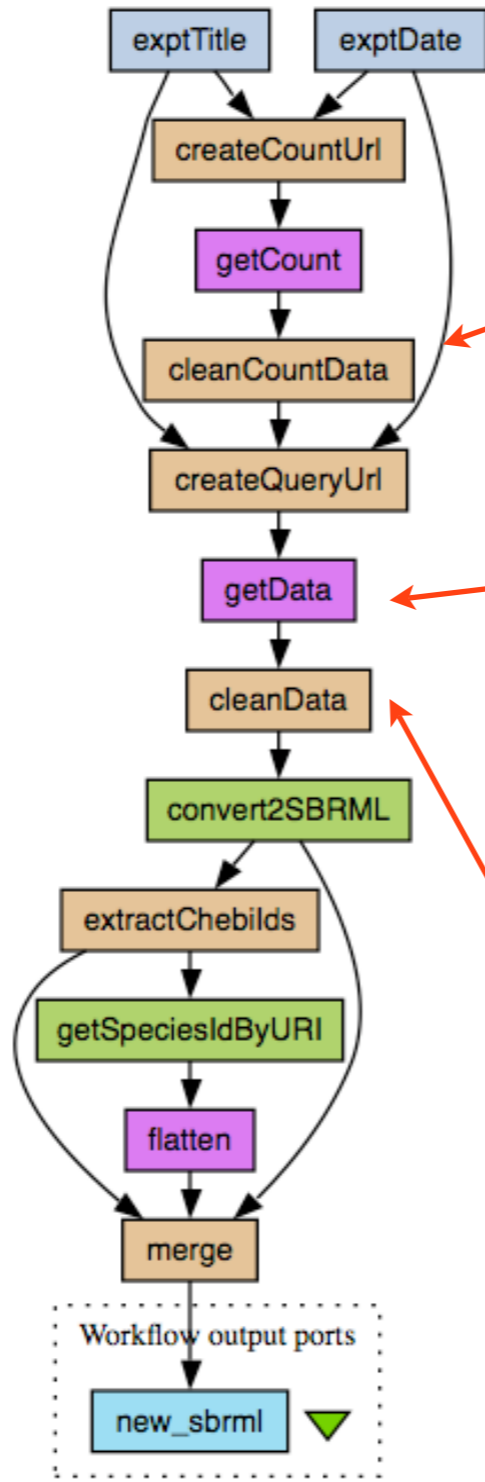
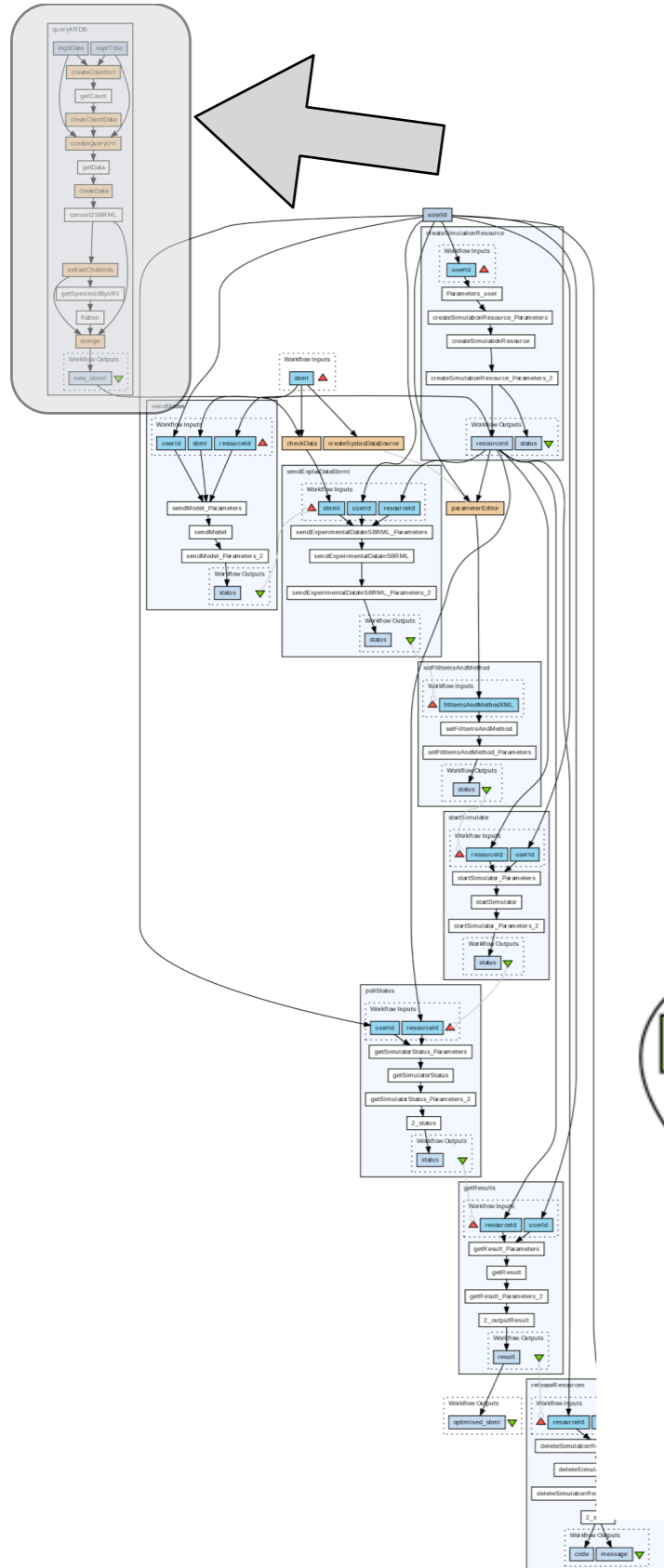
Example: SBML model optimisation workflow (see part I) -- designed by Peter Li
<http://www.myexperiment.org/workflows/1201>



Example: SBML model optimisation workflow (see part I) -- designed by Peter Li
<http://www.myexperiment.org/workflows/1201>



Example: SBML model optimisation workflow (see part I) -- designed by Peter Li
<http://www.myexperiment.org/workflows/1201>



```

String[] lines = inStr.split("\n");
StringBuffer sb = new StringBuffer();
for(i = 1; i < lines.length - 1; i++)
{
  String str = lines[i];
  str = str.replaceAll("<result>", "");
  str = str.replaceAll("</result>", "");
  sb.append(str.trim() + "\n");
}

String outStr = sb.toString();
  
```

Url -> content (built-in shell script)

```

import java.util.regex.Pattern;
import java.util.regex.Matcher;

sb = new StringBuffer();
p = "CHEBI:[0-9]+";

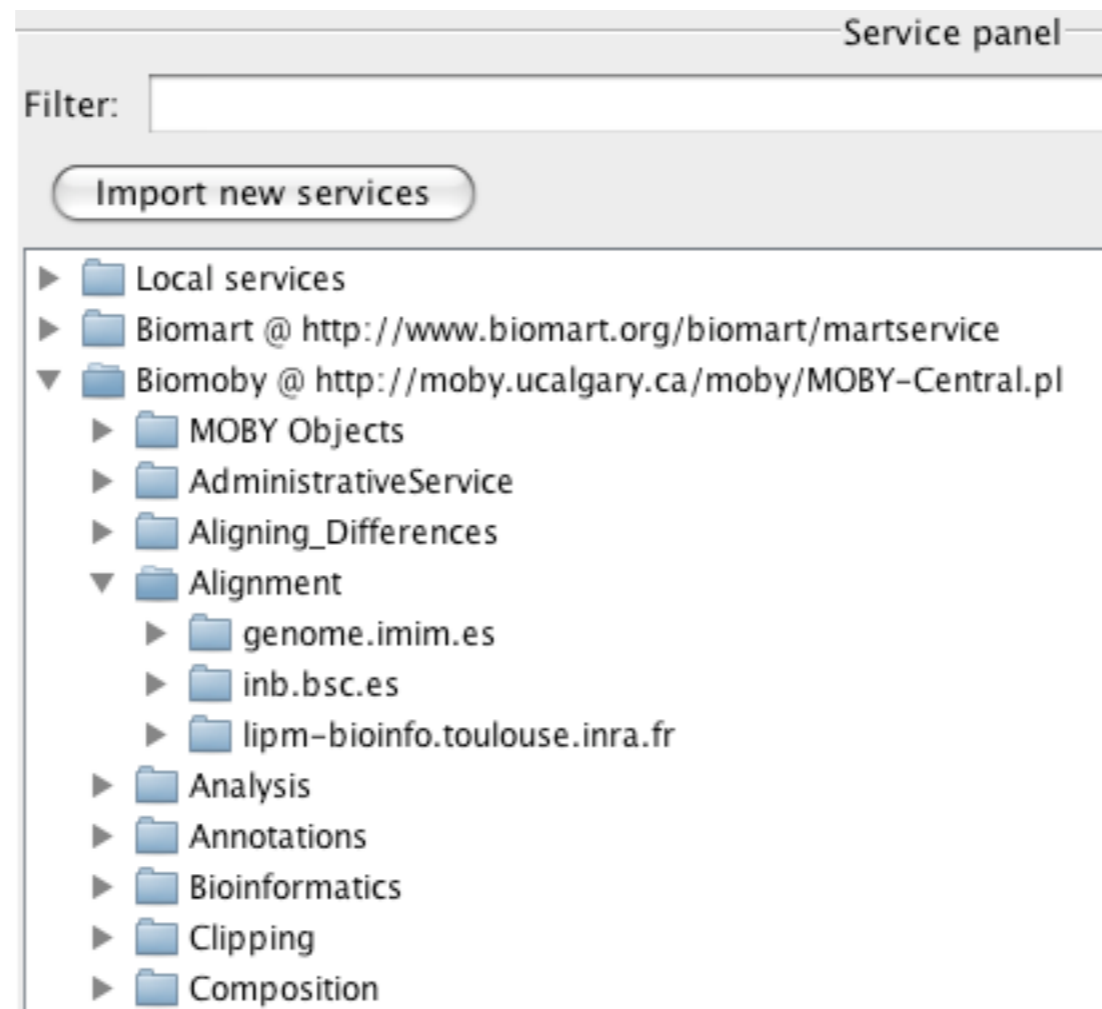
Pattern pattern = Pattern.compile(p);
Matcher matcher = pattern.matcher(sbrml);
while (matcher.find())
{
  sb.append("urn:miriam:obo.chebi:" + matcher.group() + ",");
}
String out = sb.toString();
//Clean up
if(out.endsWith(","))
  out = out.substring(0, out.length()-1);

chebids = out.split(",");
  
```

Example 1:

BioMoby / Taverna plugin

- provides a platform to
 - exchange common data representation formats
 - provide methods for service discovery
- offers more than 800 data retrieval and analysis services

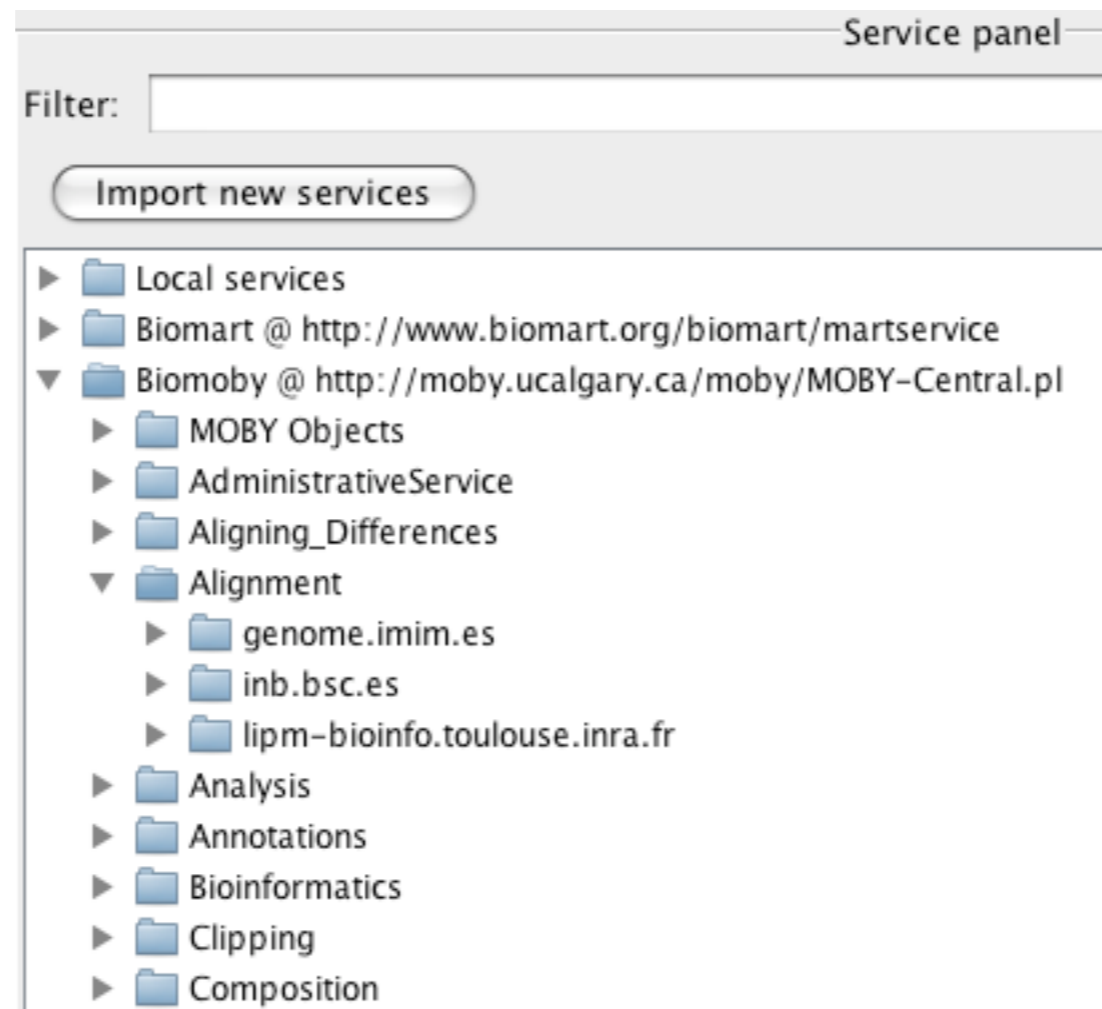


Example 1: BioMoby / Taverna plugin

- provides a platform to
 - exchange common data representation formats
 - provide methods for service discovery
- offers more than 800 data retrieval and analysis services

Example 2: caGrid

- part of caBIG (cancer Biomedical Informatics Grid)
- a US project to carry out eScience and bioinformatics in cancer research



Well-behaved service collections exist in specific domains

- Cost: may require dedicated access plugins
- Benefits:
 - well-curated → easier to discover
 - designed to work together → easier to compose

Targeting specific service collections

Well-behaved service collections exist in specific domains

- Cost: may require dedicated access plugins
- Benefits:
 - well-curated → easier to discover
 - designed to work together → easier to compose

ChemTaverna: a blend of generic + chemistry-specific components
required components reflect best practices in the specific domain

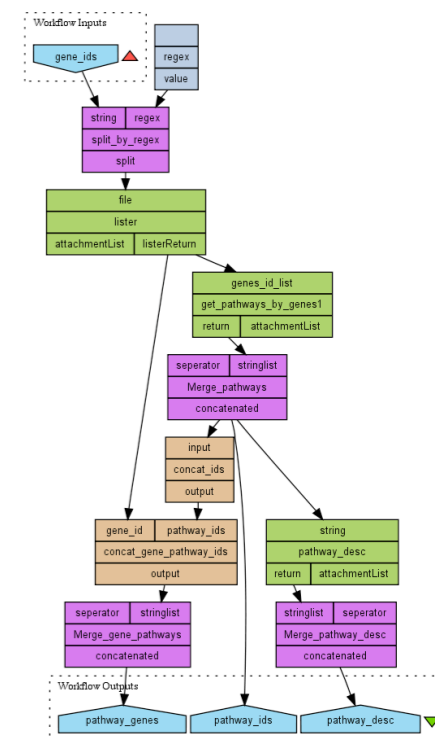
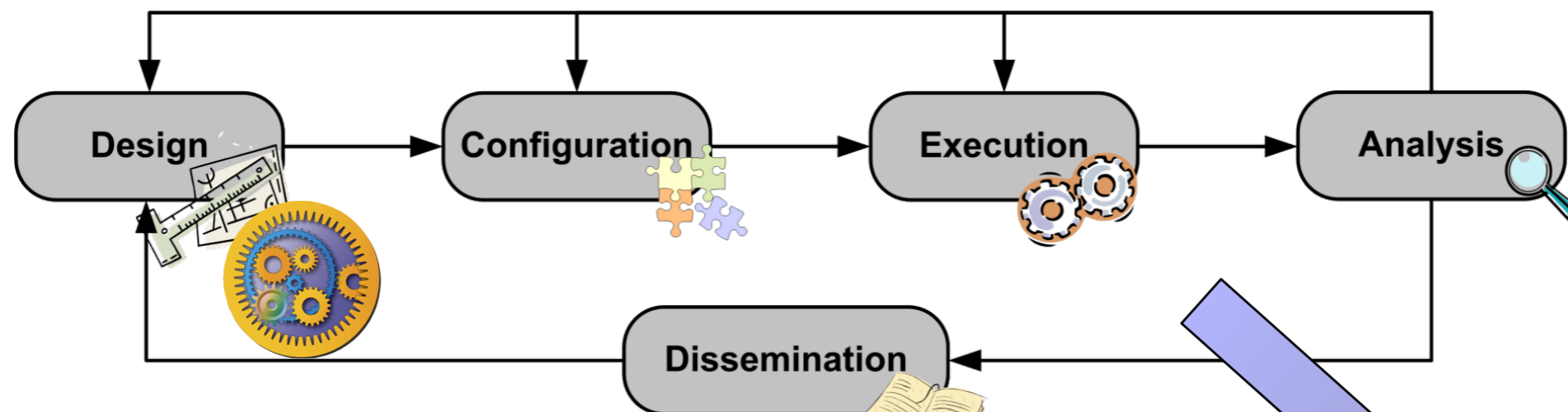
Targeting specific service collections

Well-behaved service collections exist in specific domains

- Cost: may require dedicated access plugins
- Benefits:
 - well-curated → easier to discover
 - designed to work together → easier to compose

ChemTaverna: a blend of generic + chemistry-specific components
required components reflect best practices in the specific domain

- Data I/O:
 - ▶ e.g. loading input from / writing results to spreadsheets
- Data visualisation library
- Data manipulation:
 - ▶ removal, filtering, splitting, merging, transposition
- Analysis and format transformation
 - ▶ (PubChem, KEGG, ..., R scripts)
- Service composition and seamless data flow:
 - ▶ dedicated library of reusable adapters (as opposed to *ad hoc*)



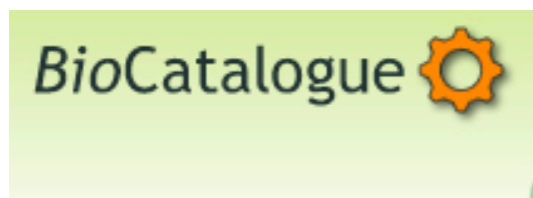
Service discovery and import

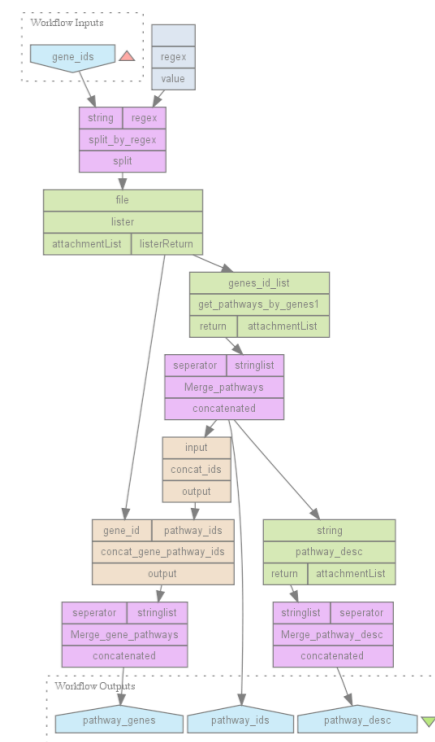
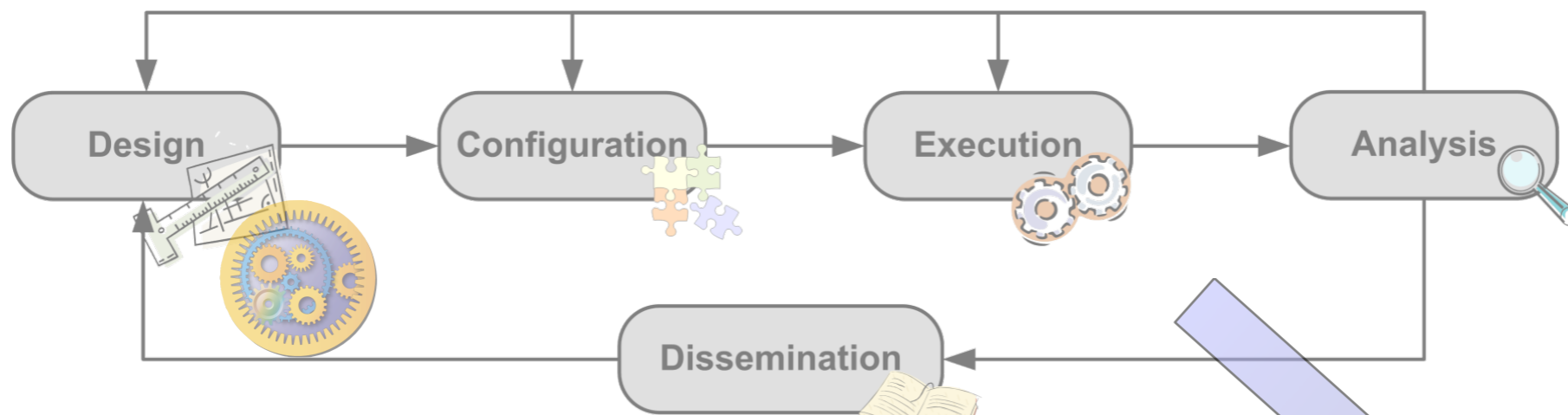
Data
- inputs
- parameters
- results

Metadata
- provenance
- annotations

Methods
- the workflow

Janus
Provenance management





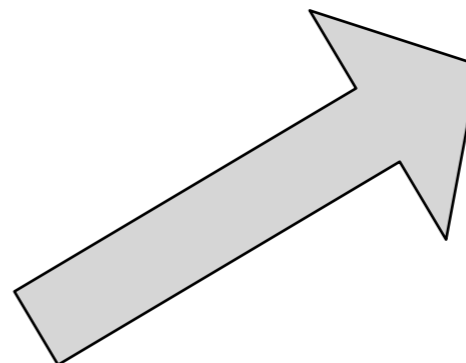
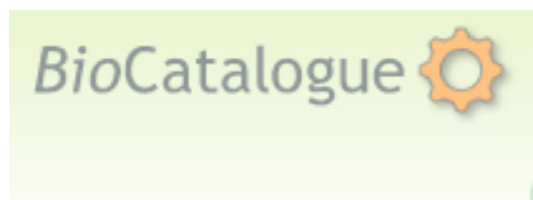
Service discovery and import

Data
- inputs
- parameters
- results

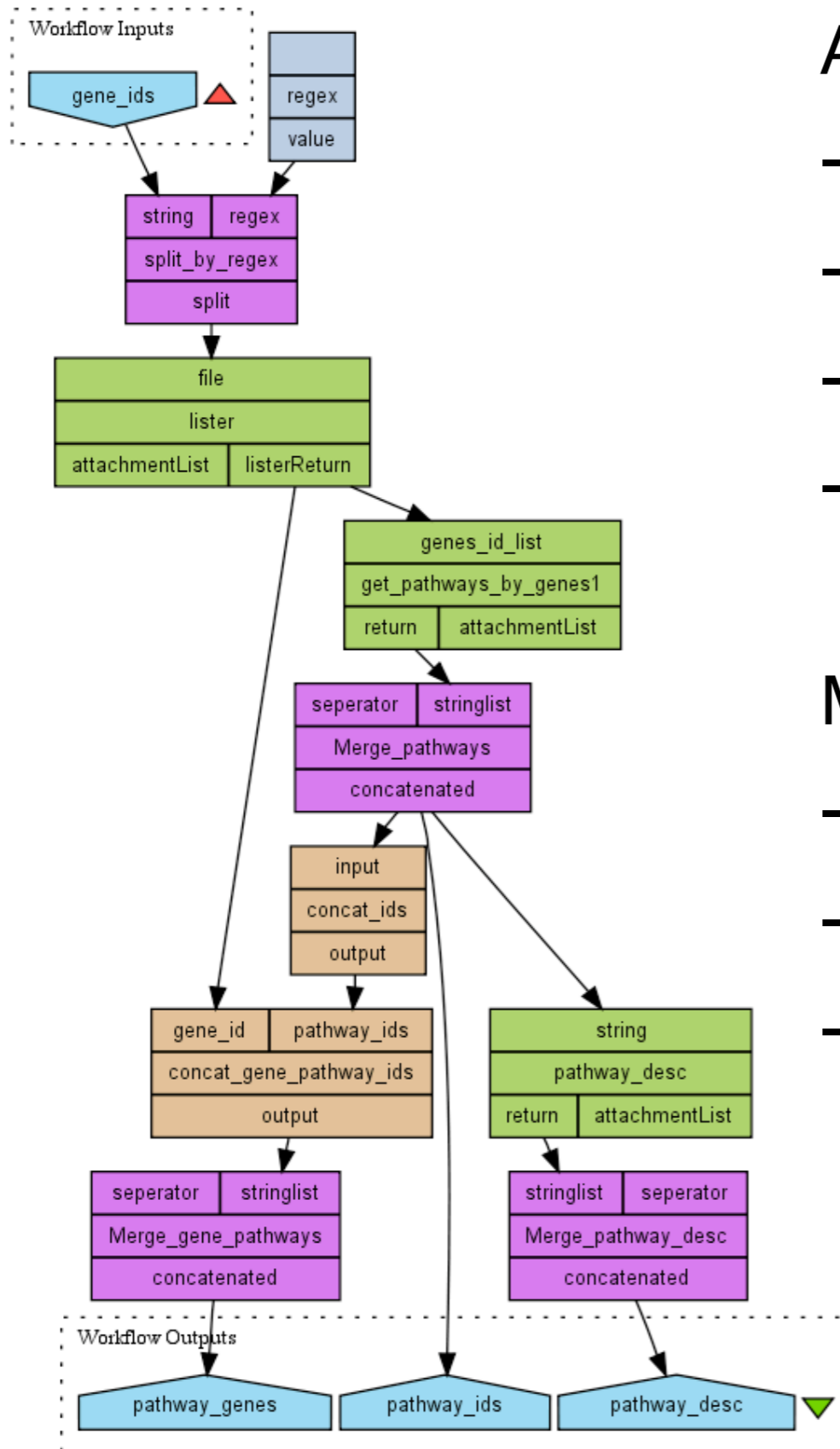
Metadata
- provenance
- annotations

Methods
- the workflow

Janus
Provenance
management



Taverna workflow provenance



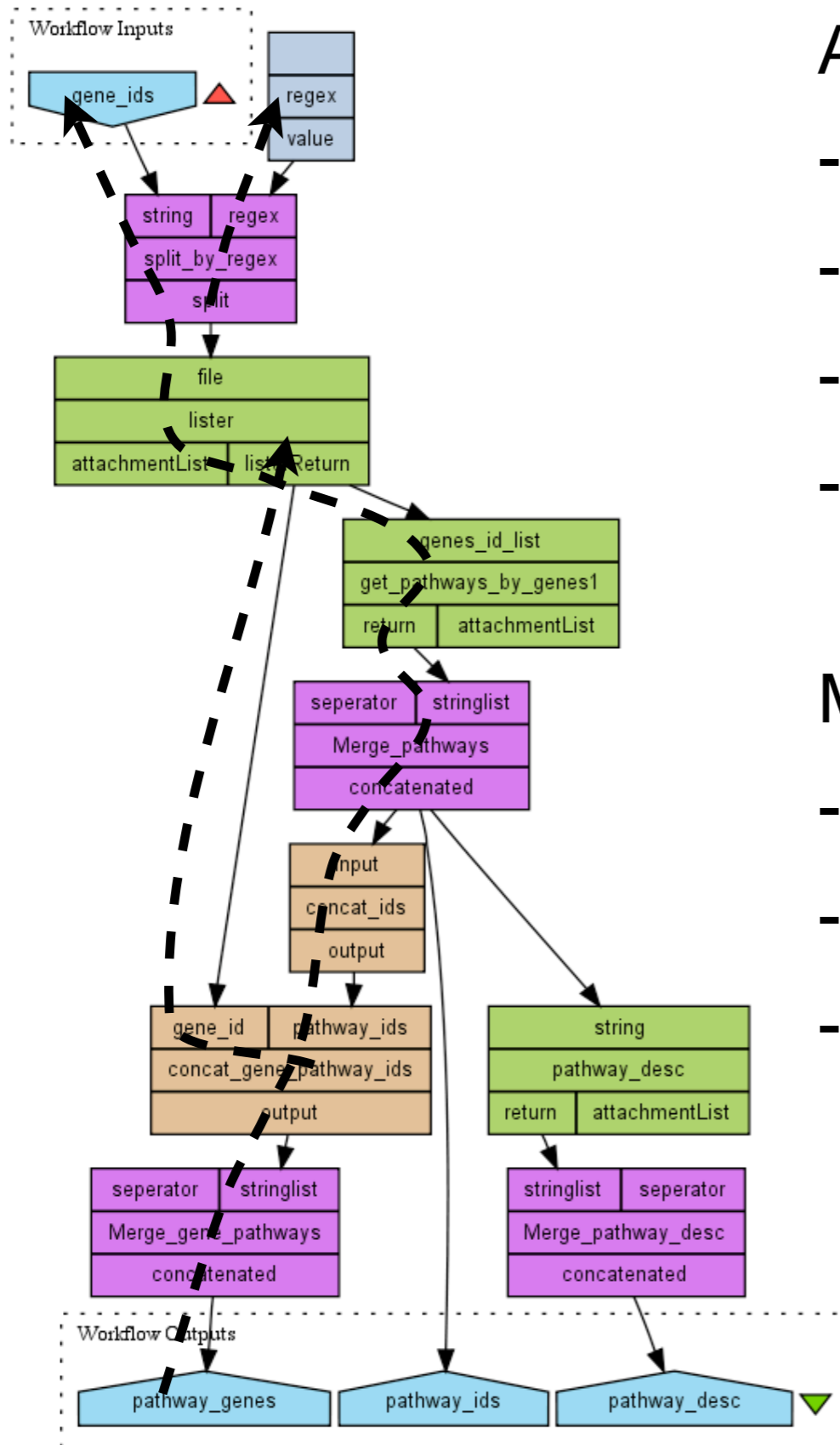
A detailed trace of workflow execution

- data dependencies
- order of processor execution
- processors' inputs/outputs
- union of these forms a DAG

Model realisation:

- relational (native)
- RDF (based on a provenance ontology)
- Open Provenance Model (XML or RDF)

Taverna workflow provenance



A detailed trace of workflow execution

- data dependencies
- order of processor execution
- processors' inputs/outputs
- union of these forms a DAG

Model realisation:

- relational (native)
- RDF (based on a provenance ontology)
- Open Provenance Model (XML or RDF)

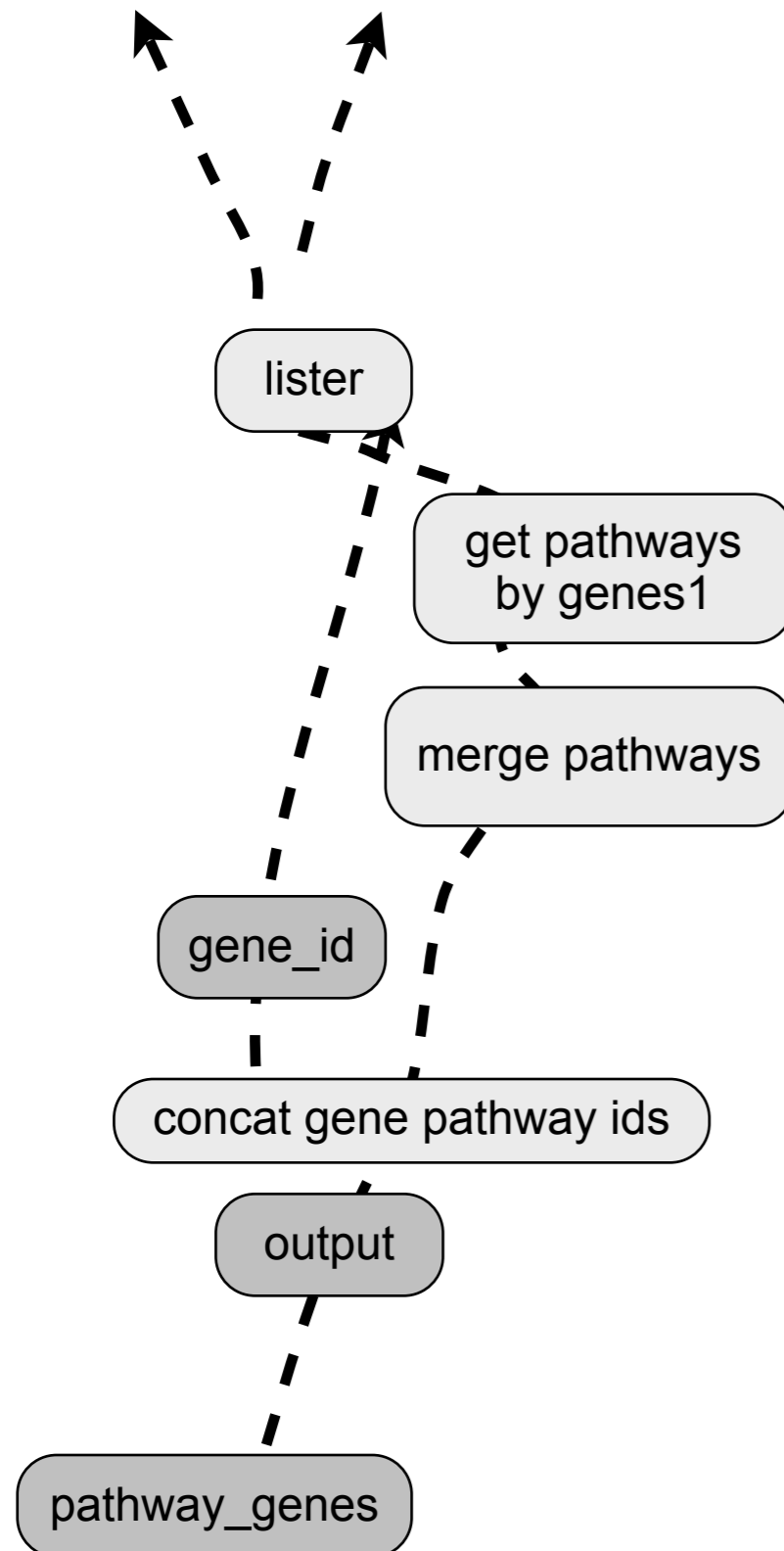
Taverna workflow provenance

A detailed trace of workflow execution

- data dependencies
- order of processor execution
- processors' inputs/outputs
- union of these forms a DAG

Model realisation:

- relational (native)
- RDF (based on a provenance ontology)
- Open Provenance Model (XML or RDF)



- To establish **quality, relevance, trust**
- To track **information attribution** through complex transformations
- To **describe** one's experiment to others, for understanding / reuse
- To **provide evidence** in support of scientific claims
- To enable *post hoc* process analysis for **improvement, re-design**

More specifically:

- **Causal** relations:
 - which input values contributed to computing an output value?
 - which process(es) caused data to be incorrect?
 - which data caused a process to fail?
- Process and data **analytics**:
 - analyze **variations** in output vs an input parameter sweep
 - **how often** has my favourite service been executed? on what inputs?
 - who produced this data?

See use cases and requirements from the [W3C Incubator on Provenance](http://www.w3.org/2005/Incubator/prov/wiki)
<http://www.w3.org/2005/Incubator/prov/wiki>

provenance access API (Java)

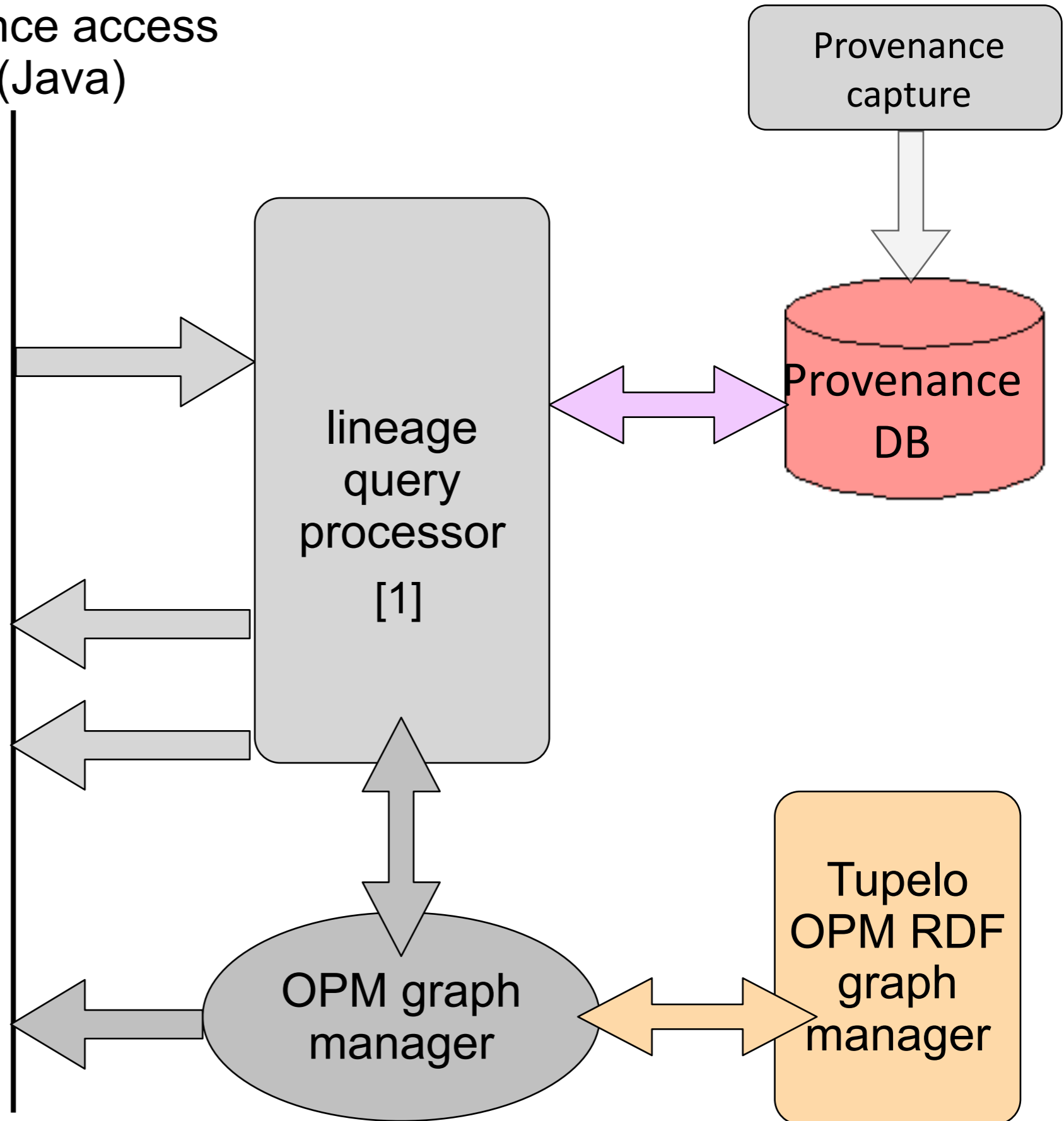
```

<select>
  <workflow name="lymphoma"> Query
    <processor name="fetch">
      <outputPort name="value"
        index="[1,2]" />
    </processor>
    ...
  </select>
  <focus>
    <workflow name="exprAnalysis">
      <processor name="p1" />
    </workflow>
    ...
  </focus>
  
```

Native query answer
(Java objects)

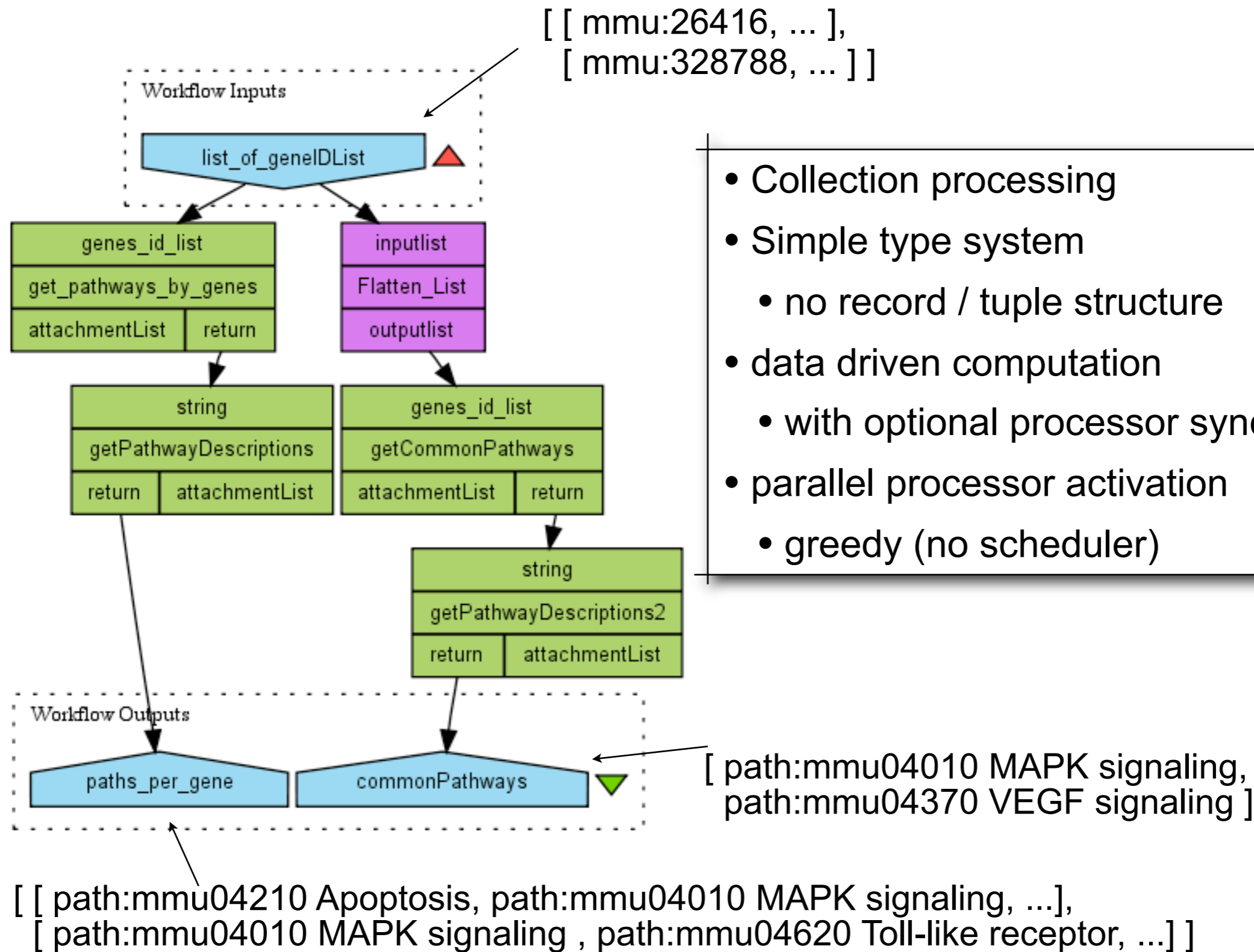
[2] RDF query answer
(Janus ontology)

[3] OPM graph
(RDF/XML)

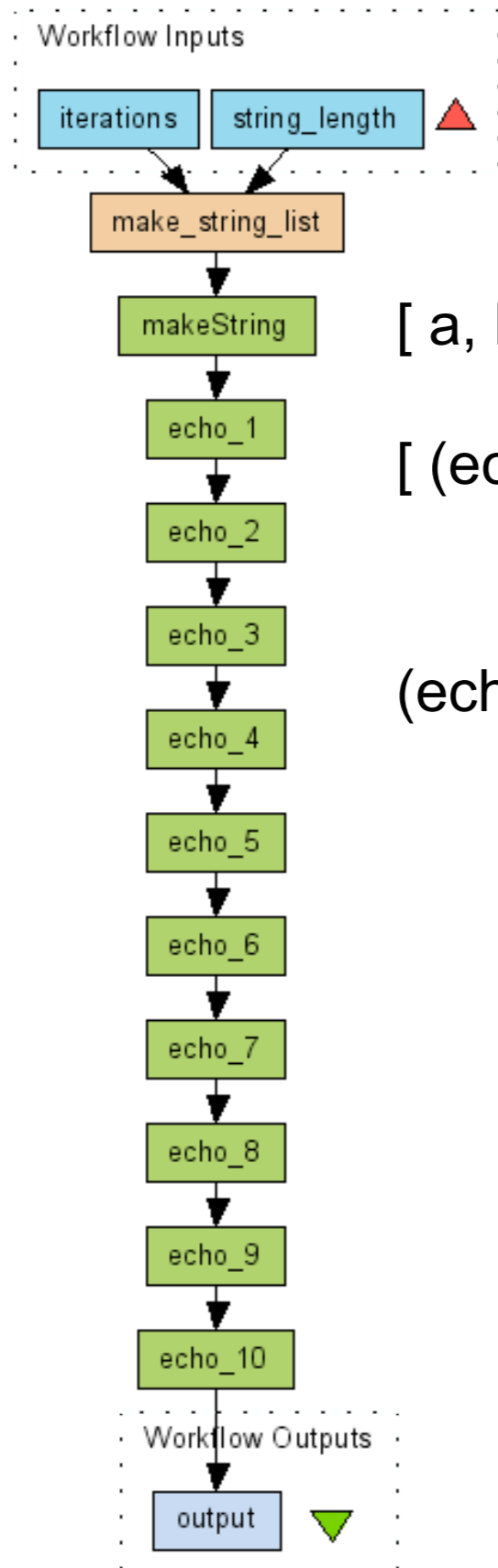


Is there a possible relationship between a dataflow model (Taverna) and a SeCo query plan?

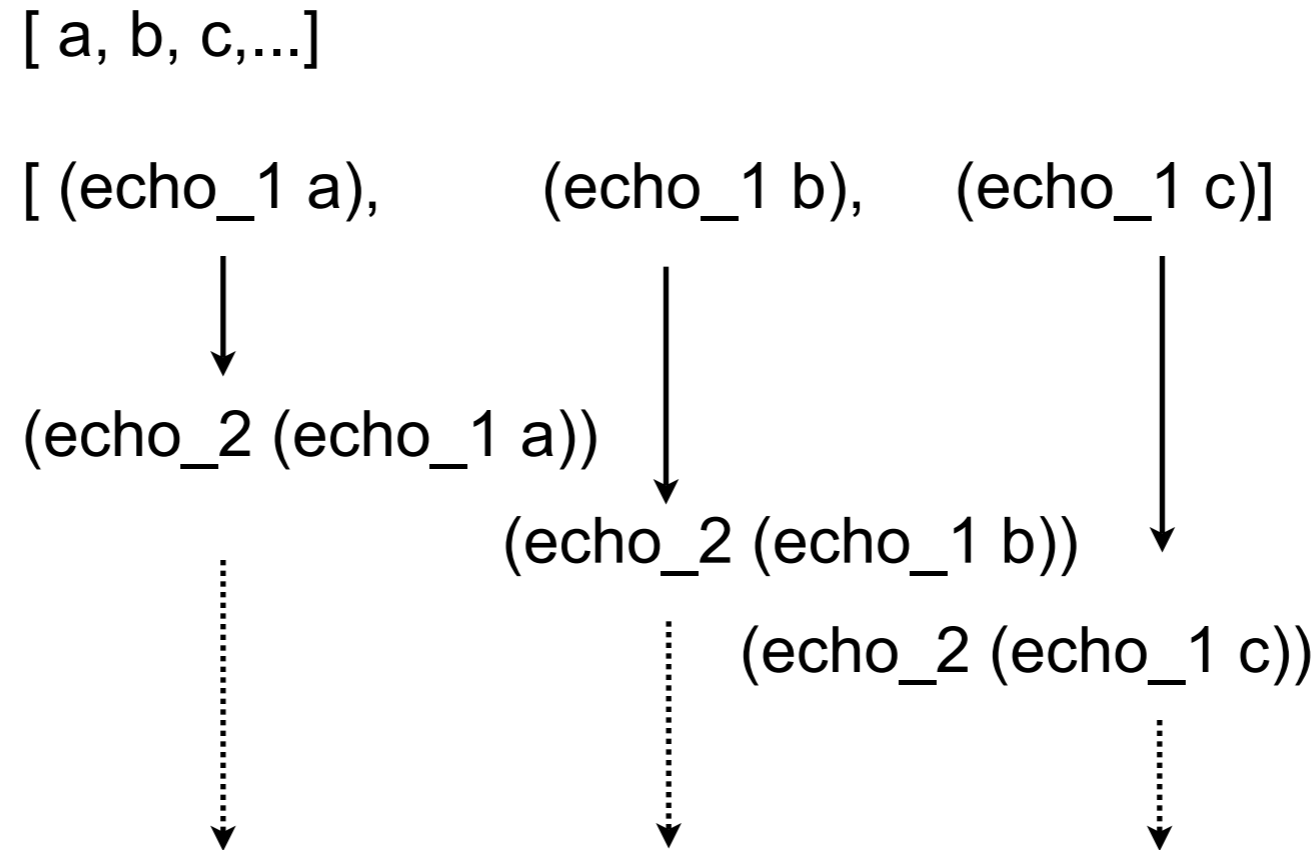
- Could any of the bioinformatics / systems workflows presented earlier have been expressed as SeCo queries?
- More generally:
 - under what assumptions can a hand-coded workflow be replaced by a query plan?
- In other words:
 - is the SeCo query model interesting from the point of view of system biology (or other bioinf domain) research?
 - would it let scientists express their service compositions at a higher level?



- Collection processing
- Simple type system
 - no record / tuple structure
- data driven computation
 - with optional processor synchronisation
- parallel processor activation
 - greedy (no scheduler)



- intra-processor: implicit iteration over collections
- inter-processor: pipelining



But:

no "chunking"
no repeated stateful calls to processors
no ranking

- Service composition requires adapters
- Target well-behaved collections of services
 - potentially low-hanging fruits
- The Taverna workflow enactor comes with a provenance capture and query component
 - Two key questions:
 1. Workflows live within eco-system of models, tools and technologies. Can any of these benefit / complement the SeCo paradigm?
 2. what is the relationship between a dataflow model (Taverna) and a SeCo query plan?
- how does the SeCo query model deal with data integration?
 - i.e., the adapters that account for the reality of data heterogeneity (in format and content)
- Is there a benefit in enhancing Taverna to support SeCo query plans?
- can the SeCo model take advantage of existing provenance models?

- (1) P. Missier, N. Paton, and K. Belhajjame, "Fine-grained and efficient lineage querying of collection-based workflow provenance," Procs. EDBT, Lausanne, Switzerland: 2010.
- (2) P. Missier, S.S. Sahoo, J. Zhao, A. Sheth, and C. Goble, "Janus: from Workflows to Semantic Provenance and Linked Open Data," Procs. IPAW 2010, Troy, NY: 2010.
- (3) P. Missier and C. Goble, "Workflows to Open Provenance Graphs, round-trip", Future Generation Computing Systems Journal, Special issue on the Open Provenance Model, *submitted*.
- (4) P. Missier, S. Soiland-Reyes, S. Owen, W. Tan, A. Nenadic, I. Dunlop, A. Williams, T. Oinn, and C. Goble, "Taverna, reloaded," Procs. SSDBM 2010, M. Gertz, T. Hey, and B. Ludaescher, Heidelberg, Germany: 2010.