http://www.gridworkflow.org/

http://users.cs.cf.ac.uk/O.F.Rana/icsoc08-workflow.ppt

Distributed Scientific Workflows: Techniques, Tools and Applications

Omer F. Rana

School of Computer Science and

Welsh eScience Centre

Cardiff University, UK

o.f.rana@cs.cardiff.ac.uk

Thanks to:

- David Walker, Ian Taylor, Matthew Shields, Lican Huang, Ali Shaikh Ali, Richard White at Cardiff
- Bertram Ludaescher at UC Davis
- Cecilia Gomes at UNL-Lisbon
- John Domingue at Open University
- Steve McGough, John Darlington at Imperial College
- Luc Moreau and Terry Payne, University of Southampton
- Zhiming Zhao, University of Amsterdam

Some material contained in this tutorial has been obtained from the individuals mentioned above.

Overview

- Introduction + Examples of Scientific Workflows
- Constructing and Managing Workflow
- Application Example: Distributed Data Mining using FAEHIM
- Adaptive Workflows
- Workflow-related research themes

Time Division

13:00 – 13:50 Workflow examples, types, general issues, scientific vs. business workflows

13:50 – 14:00 Break

14:00 – 14:50 Workflow Optimisation, Dynamic Adaptation, Semantics, Provenance

14:50 – 15:00 Triana Demonstration

15:00 – 15:10 Discussion + Research Directions

Workflow

Adapted From:
Aleksander Slominski

- '70s: Skip Ellis And Gary Nutt (OfficeTalk)
  - Xerox Parc "Office Automation Systems"
  - "to reduce the complexity of the user's interface to the [office information] system, control the flow of information, and enhance the overall efficiency of the office." (Ellis, Nutt 1980)
- "Representation, Specification, and Automation of Office Procedures" (Michael D. Zisman, PhD Thesis, University of Pennsylvania, *Warton School of Business,* **1977**)
  - **Often seen as a technique to automate existing "processes"**
  - **Very popular in the business world**
- **Over 20 years gap:**
  - Availability of Computer Networks

Ellis, C. A.; Nutt, G. J. Office Information Systems and Computer Science. In *ACM Computing Surveys*, 12 (1980) 1, pp. 27-60.

Historical Perspective

From:
Aleksander Slominski

- '65-'75 Decompose Applications
  - Data And Code Separated
- '75-'85 Database Management
  - DBMS Used To Share Data
- '85-'95 User Interface Management
  - User Interface Separated
- '95-'08 Workflow Management
  - Isolate Business Process
  - Emerging standards – such as those based on the Service Oriented Architecture
  - Use of Service Mashups

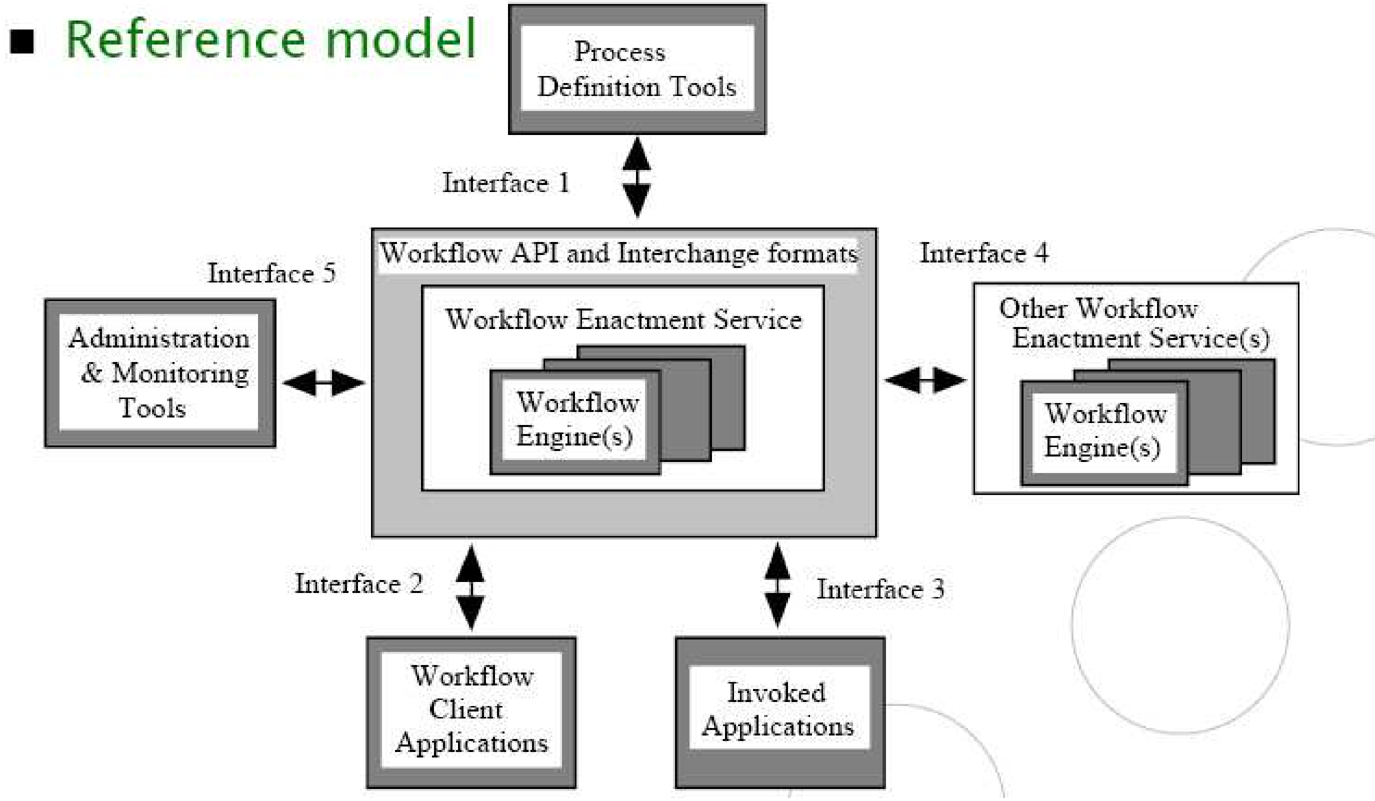
"Workflow Management" Aalst, van Hee

Workflow

From:
Aleksander Slominski

"The automation of a business process, in whole or part, where documents, information or tasks are passed from one participant to another to be processed, according to a set of procedural rules "

Workflow Management Coalition (WfMC)



WFMS And WF Engine

From:
Aleksander Slominski

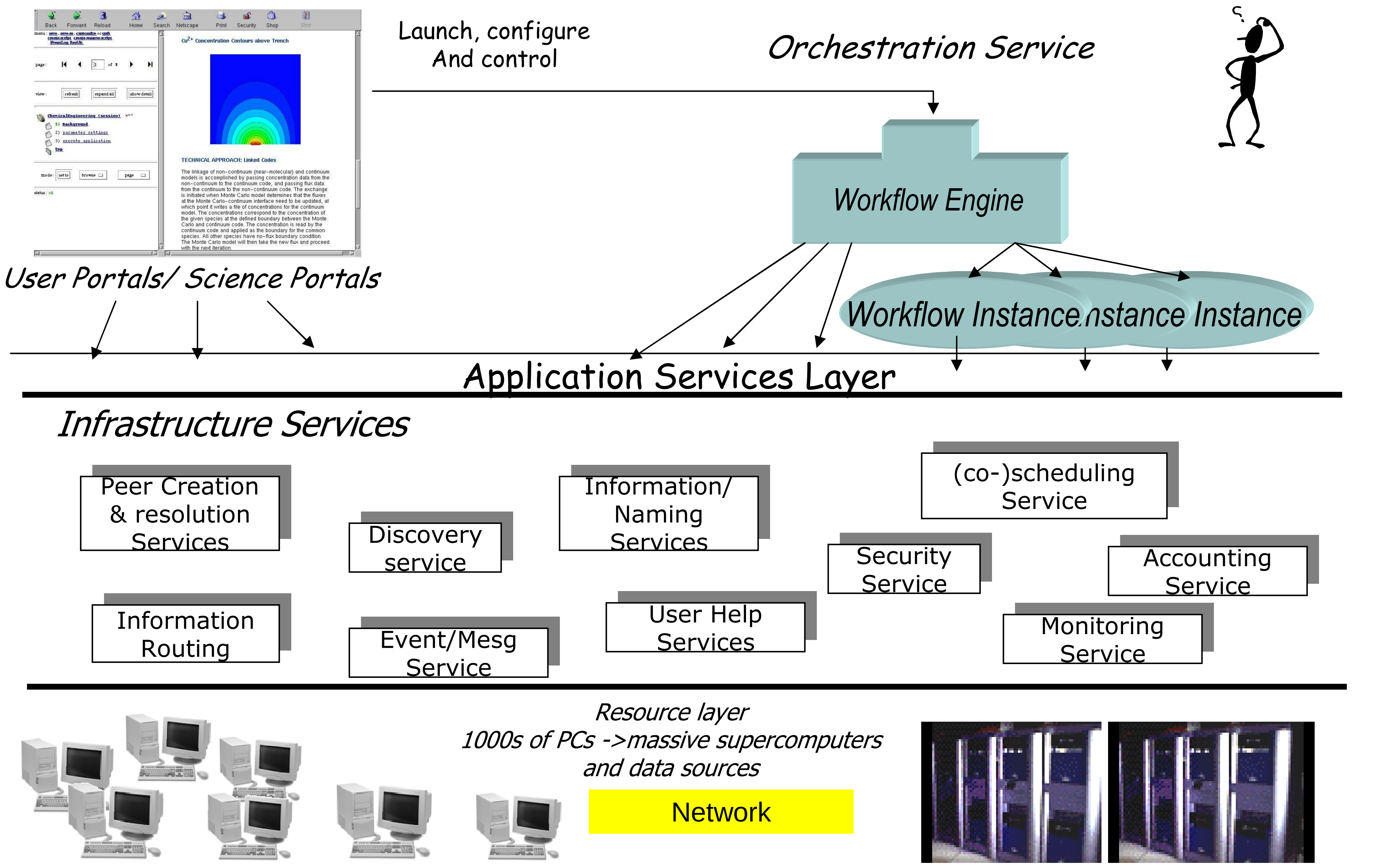
- Workflow Management System (WFMS)
  - "A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications."
- Workflow Engine
  - "A software service or `engine' that provides the run time execution environment for a process instance."

A representation that shows
- precedence relationship (links) between activities

May be
- Acyclic (no loops) or cyclic
- Contain annotations associate with activities or links

Workflow (+ Enactment)

From:
Aleksander Slominski



Scientific Workflows

- What makes it different (how it is applied)?
  - Support for large data flows
  - Need to do parameterized execution of large number of jobs
  - Need to monitor and control workflow execution including ad-hoc changes
  - Need to execute in a dynamic environment where resources are not known a priori and may need to adapt to changes
  - Hierarchical execution with sub-workflows created and destroyed when necessary
- Science Domain specific requirements.

Problems with “Predictability”

- Triana
- Taverna/SCUFL
- GridAnt
- Condor DAG
- CoG DAG
- SWFL
- BioOpera/JOpera
- BEPL4WS
- OASIS WSBEPL
- YAWL
- GSFL
- Askalon
- OMII-BPEL, etc

Origin (?):
Problem Solving Environments
(MatLab, Mathematica, SciRun, NetSolve, Ninf, Nimrod etc)

Workflow World

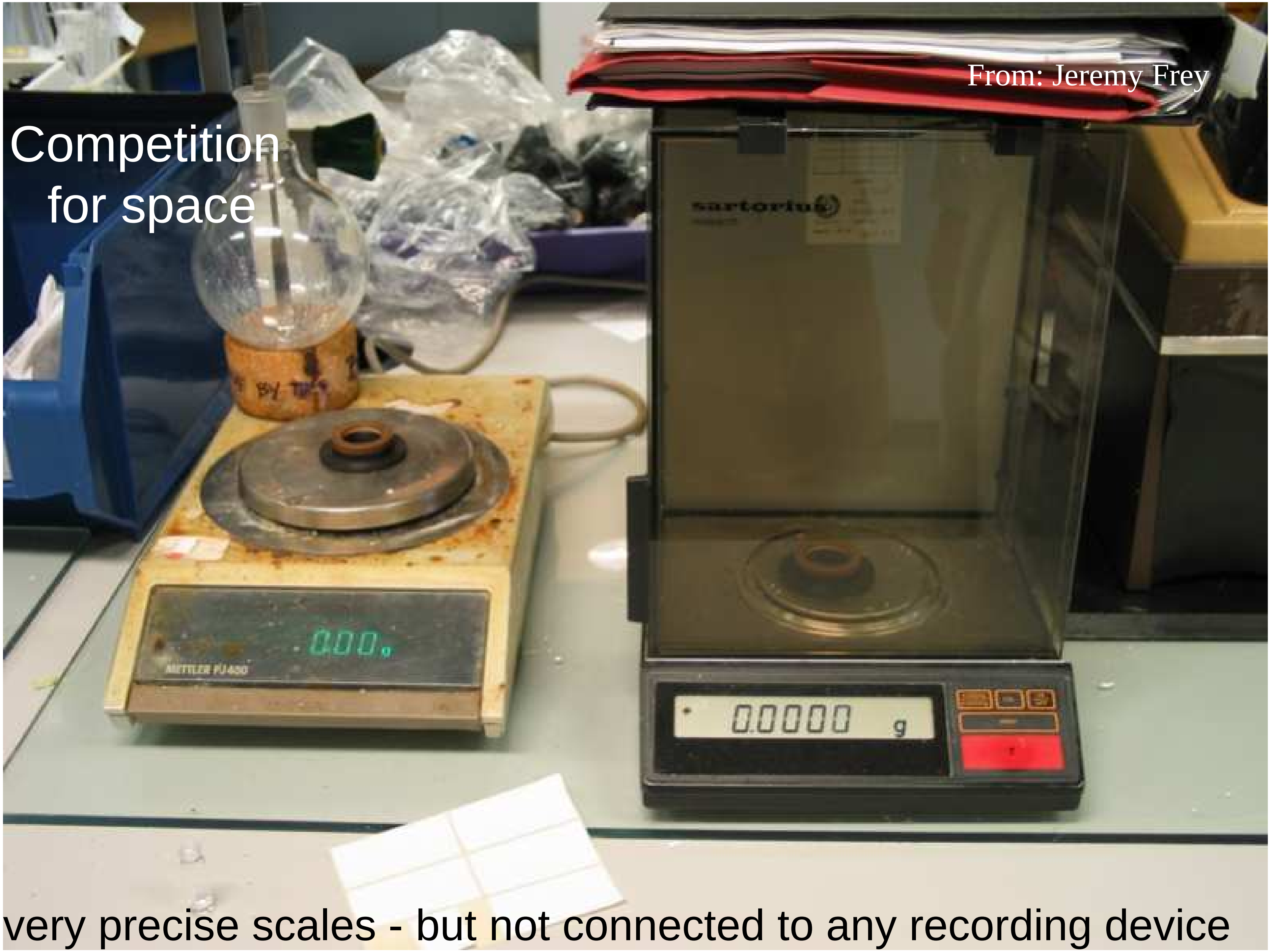
http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=303

http://www.extreme.indiana.edu/swf-survey/

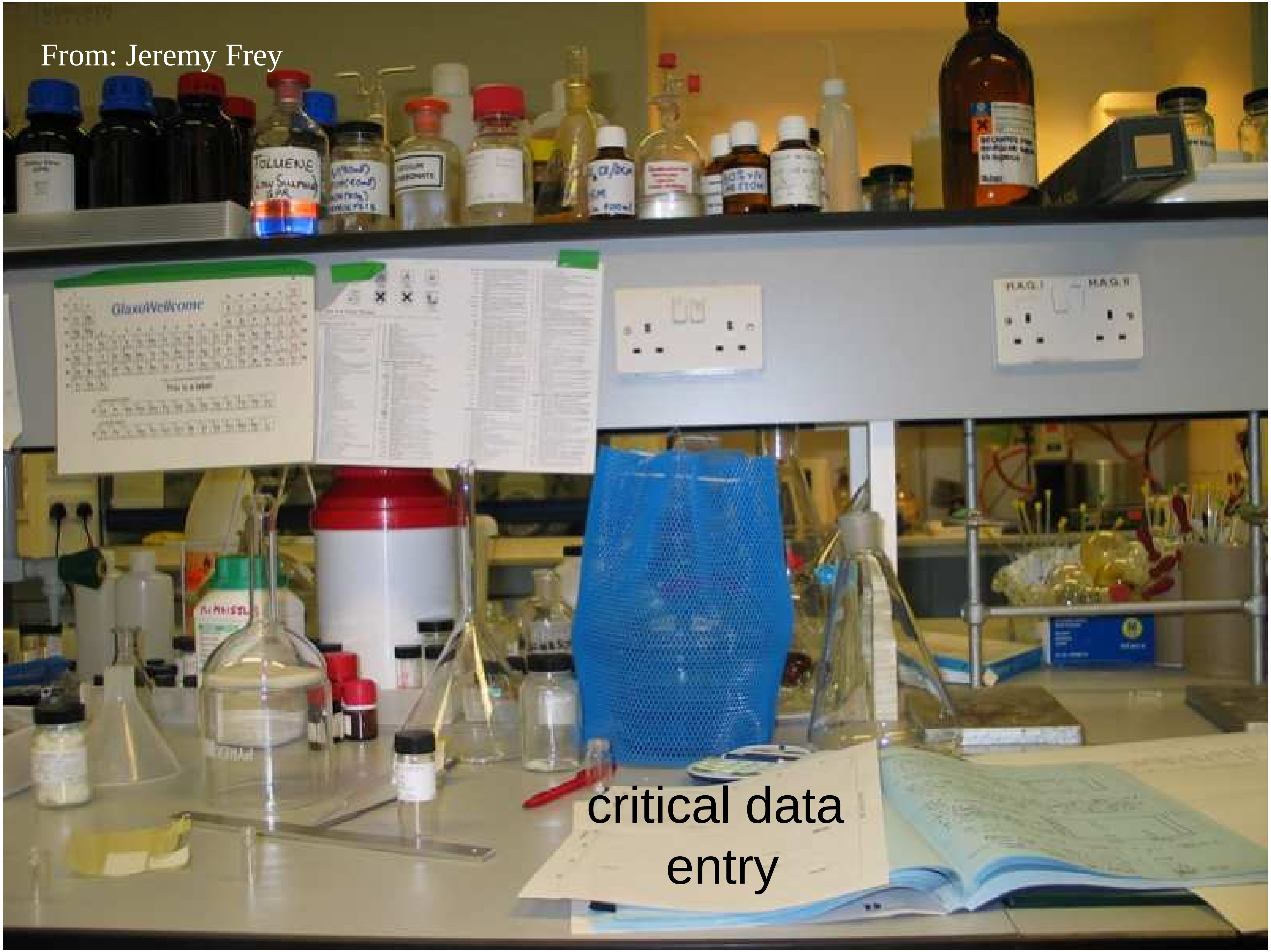

A chemistry lab is a hostile environment
without much room to maneuver
what can be captured automatically with
sensors?
what must rely on manual annotation?
The chemist
The fume
cupboard
From: Jeremy Frey

Competition for space

From: Jeremy Frey

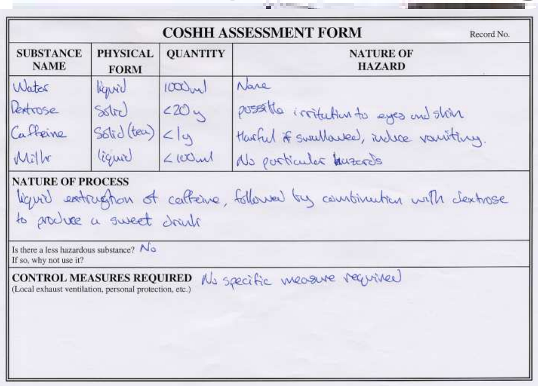


very precise scales - but not connected to any recording device


From: Jeremy Frey
critical data
entry

By Making Tea!

From: Jeremy Frey



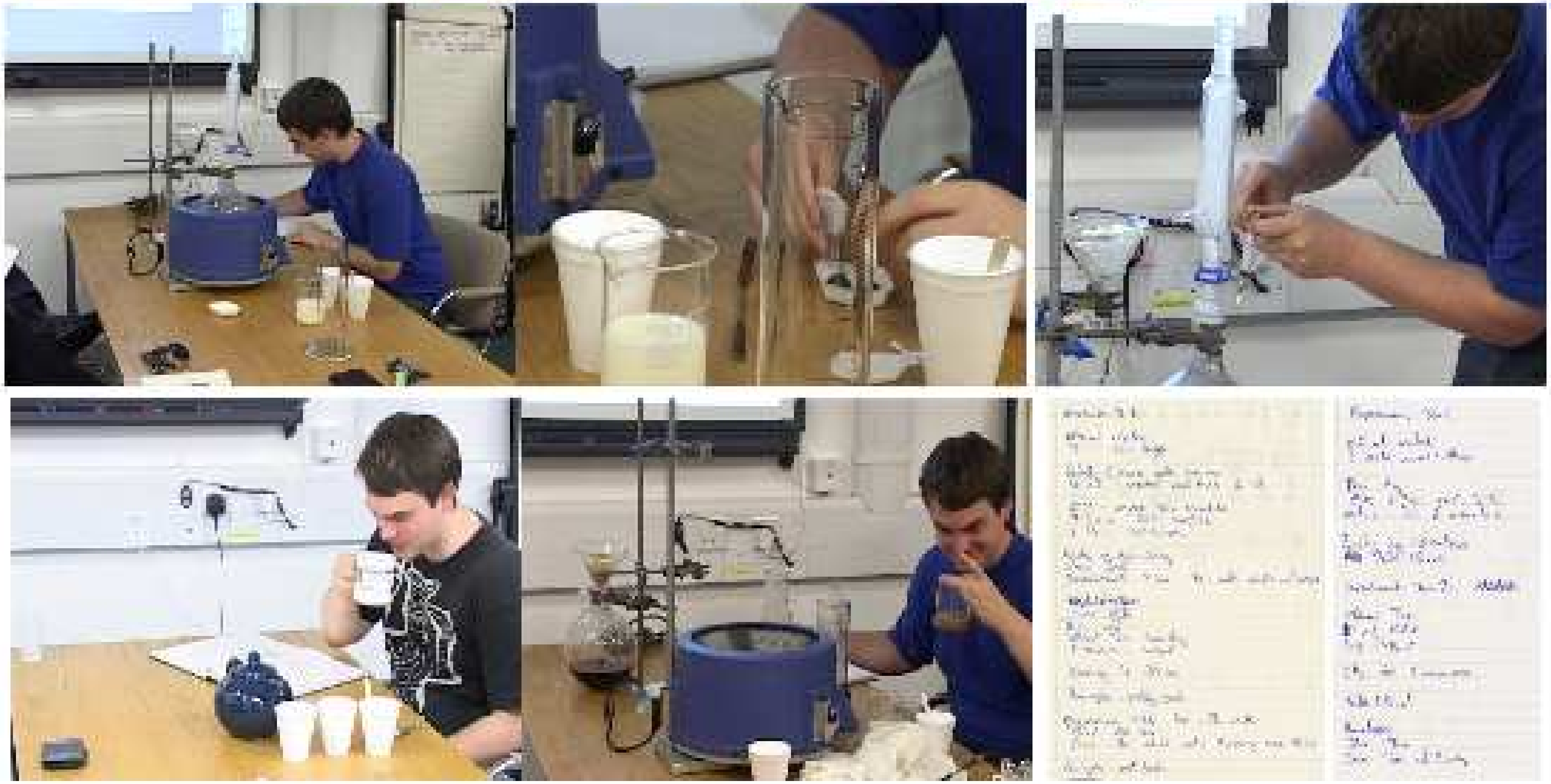
COSHH ASSESSMENT FORM Record No.

SUBSTANCE NAME	PHYSICAL FORM	QUANTITY	NATURE OF HAZARD
Water	liquid	1000ml	None
Dextrose	Solid	<20 g	possible irritation to eyes and skin
Caffeine	Solid (tea)	<1g	Harmful if swallowed, induce vomiting.
Milk	liquid	<100ml	No particular hazards

NATURE OF PROCESS
liquid extraction of caffeine, followed by combination with dextrose to produce a sweet drink

Is there a less hazardous substance? No
If so, why not use it?

CONTROL MEASURES REQUIRED No specific measure required
(Local exhaust ventilation, personal protection, etc.)



Getting not just the what and how, but the *why*

By Making Tea!

From: Jeremy Frey

COSHH ASSESSMENT FORM Record No.

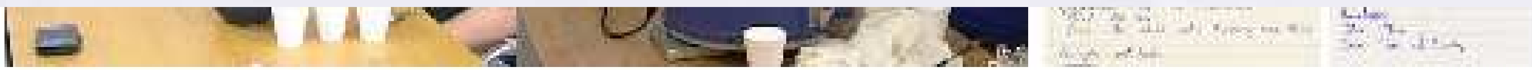
SUBSTANCE NAME	PHYSICAL FORM	QUANTITY	NATURE OF HAZARD
Water	liquid	1000ml	None
Dextrose	Solid	<20g	possible irritation to eyes and skin
Caffeine	Solid (tea)	<1g	Harmful if swallowed, induce vomiting.
Milk	liquid	<100ml	No particular hazards

NATURE OF PROCESS

liquid extraction of caffeine, followed by combination with dextrose to produce a sweet drink

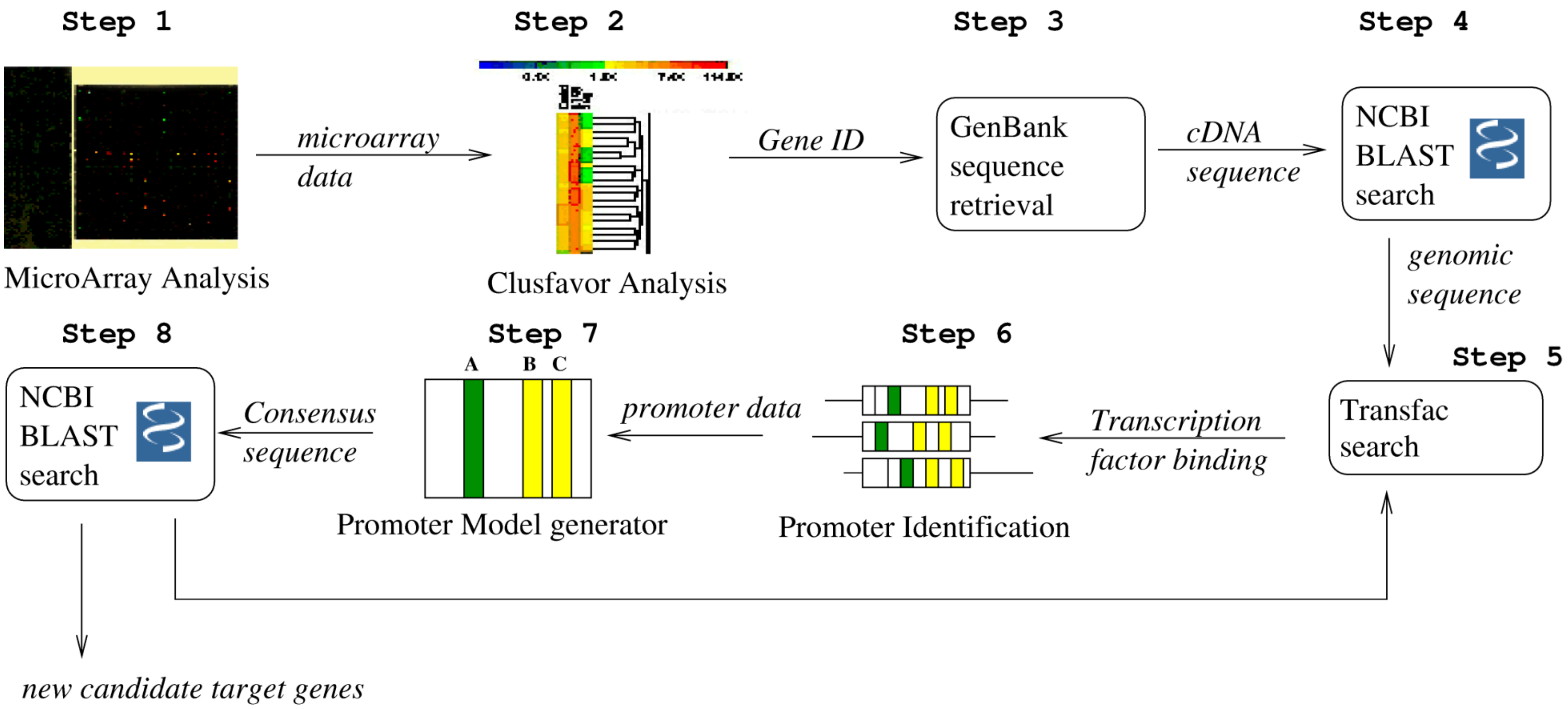
Is there a less hazardous substance? No
If so, why not use it?

CONTROL MEASURES REQUIRED No specific measure required
(Local exhaust ventilation, personal protection, etc.)



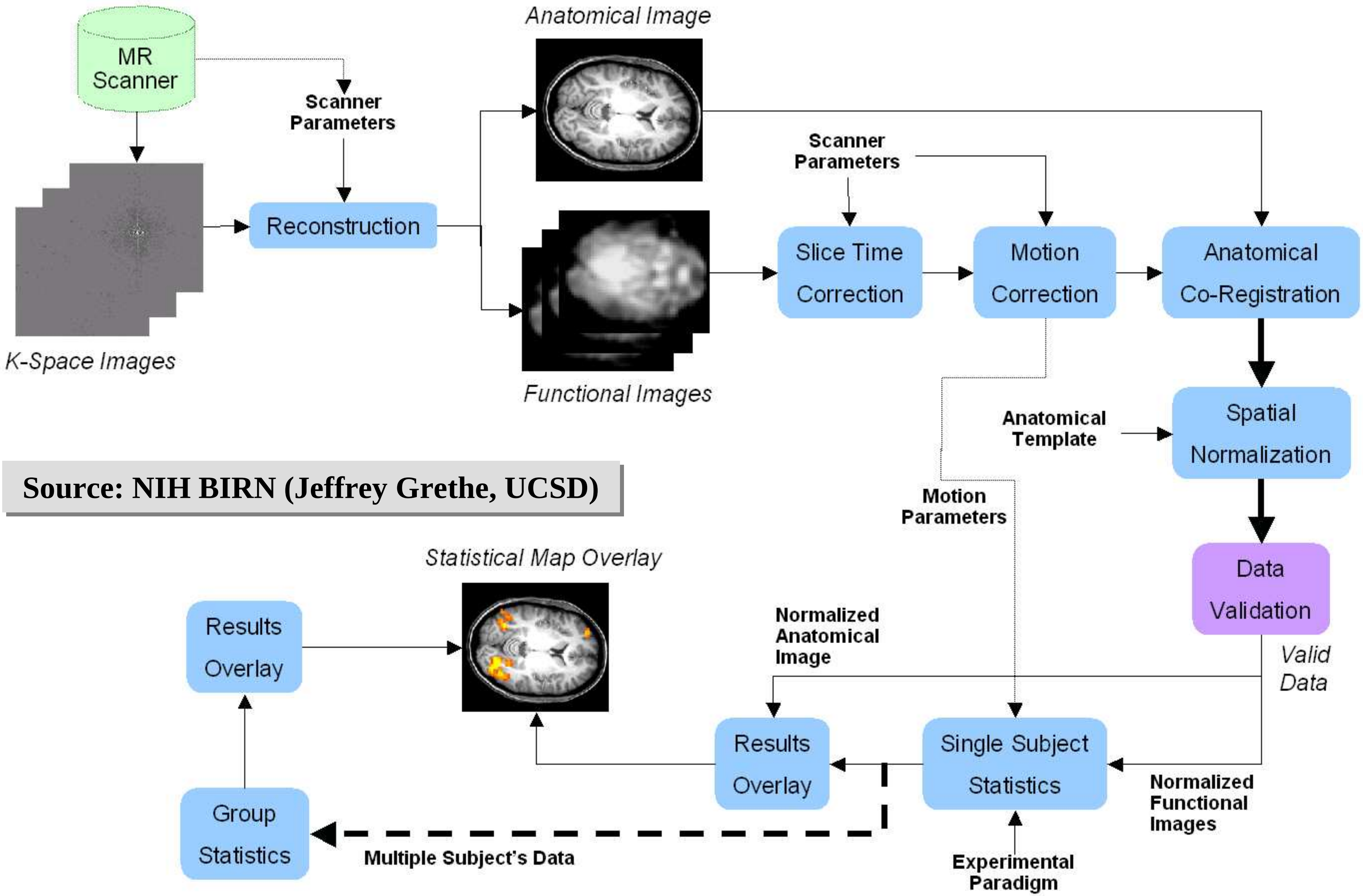
Getting not just the what and how, but the *why*

Promoter Identification Workflow



Source: Matt Coleman (LLNL)

Functional MRI Analysis Workflow



Source: NIH BIRN (Jeffrey Grethe, UCSD)

Montage (http://montage.ipac.caltech.edu/)

- Deliver science-grade custom mosaics on demand
  - Produce mosaics from a wide range of data sources (possibly in different spectra)
  - User-specified parameters of projection, coordinates, size, rotation and spatial sampling.

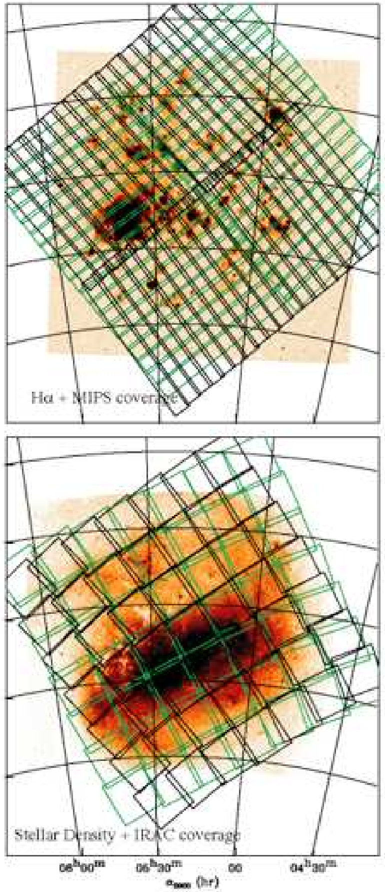
The **Large Magellanic Cloud** (LMC) is a nearby satellite galaxy of our own galaxy, the Milky Way. At a distance of slightly less than 50 kiloparsecs (≈160,000 light-years), the LMC is the third closest galaxy to the Milky Way. It has a mass equivalent to approximately 10 billion times the mass of our Sun (1010 solar masses), making it roughly 1/10 as massive as the Milky Way. The LMC is the Fourth largest galaxy in the Local Group, the first, second and third largest places being taken by Andromeda Galaxy (M31), our own Milky Way Galaxy, and the Triangulum Galaxy (M33).

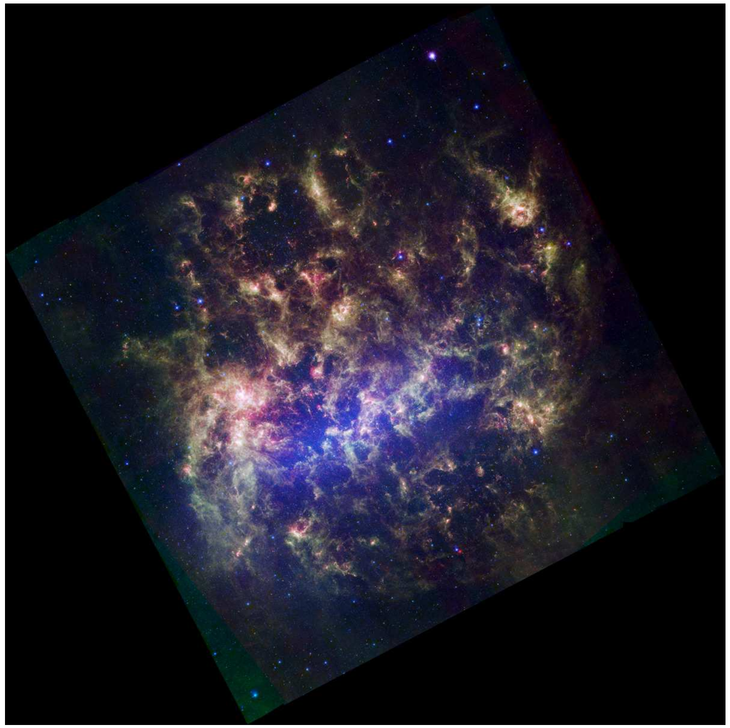
SAGE: Spitzer Survey of the Large Magellanic Cloud

Instrument	Bands (μm)	Field-of-View (arcmin)
IRAC	3.5, 4.5, 5.8, 8.0	5.2. x 5.2
MIPS	24	5.4 x 5.4
	70	5.25 x 2.6
	160	0.5 x 0.5

IRAC: Infrared Array Camera

MIPS: Multiband Imaging Photometry





IRAC 3.6 μm

IRAC 8.0 μm

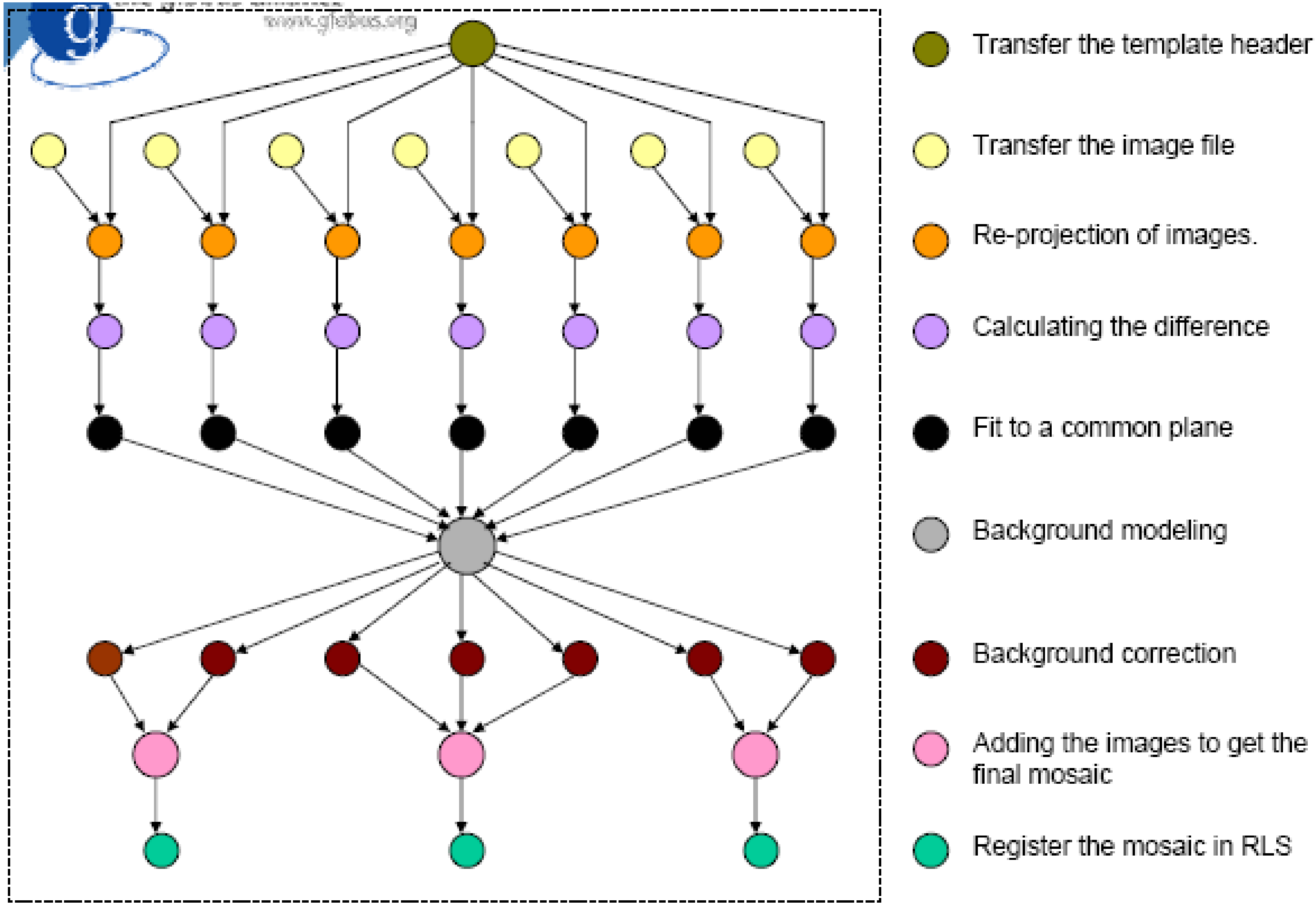
MIPS 24 μm

Two epochs:
Jul/Aug 05 & Oct/Nov 05

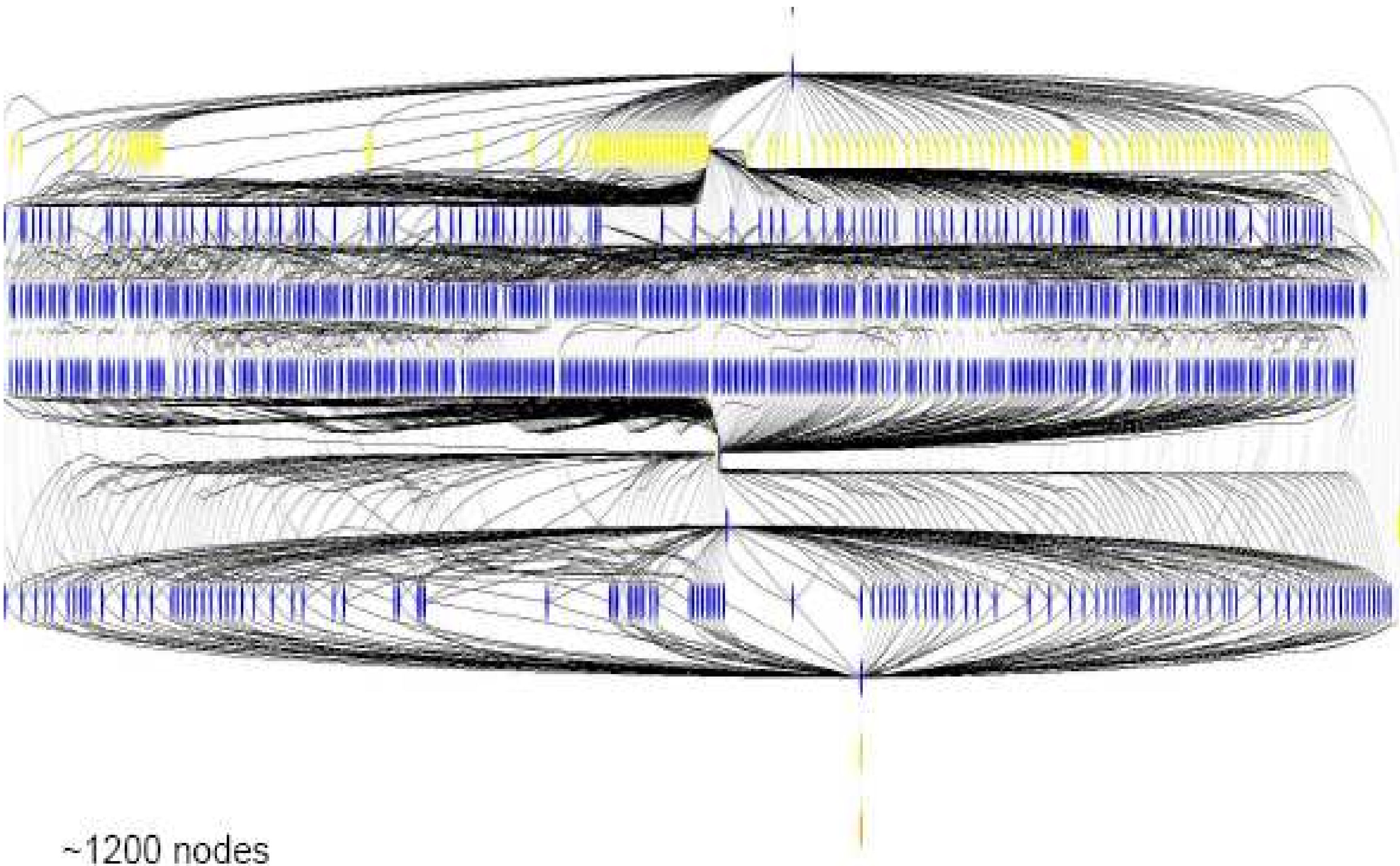
Images Courtesy Margaret Meixner (PI)

From: G. Bruce Berriman

Montage Workflow (from Ewa Deelman)



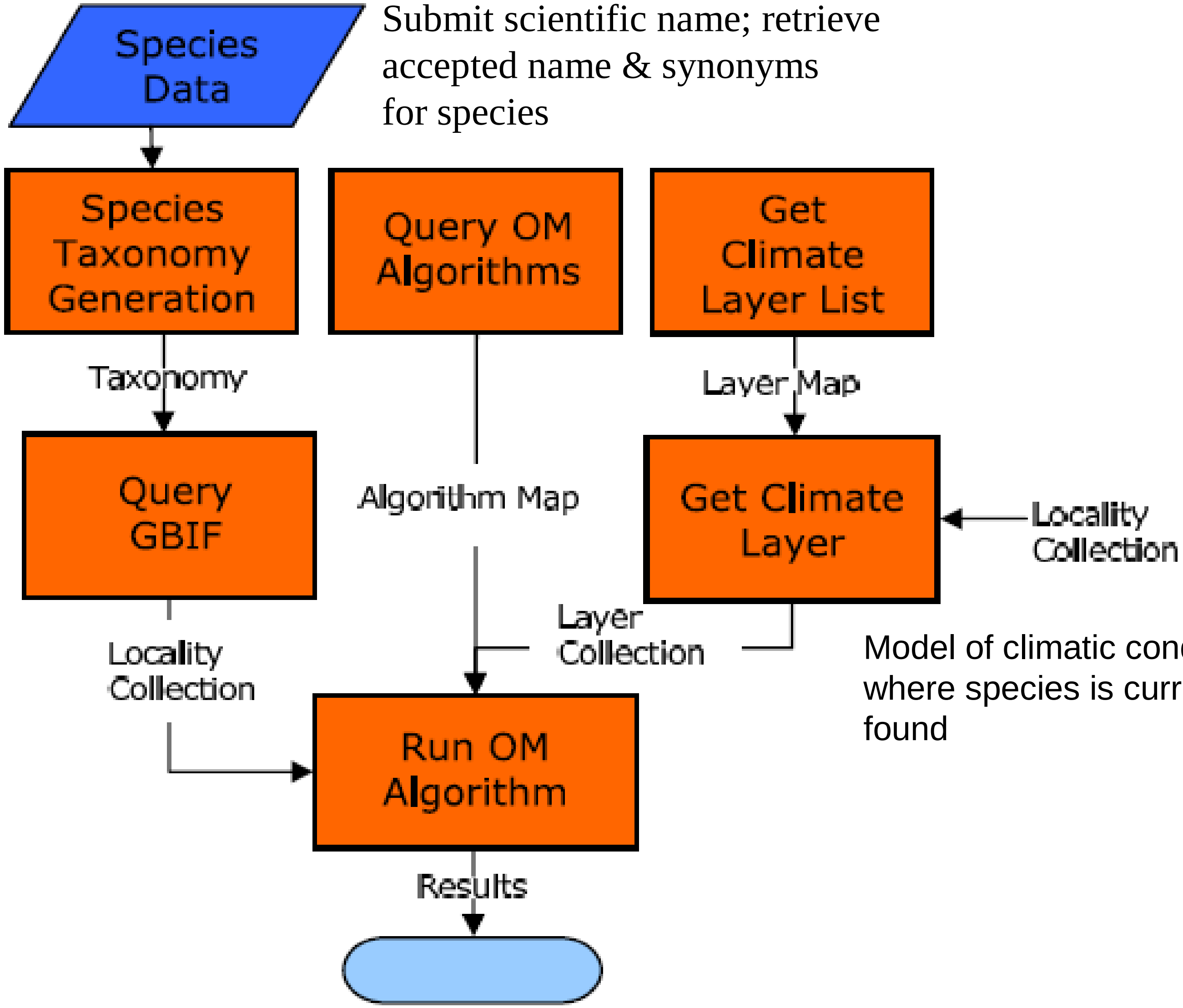
Montage workflow



~1200 nodes

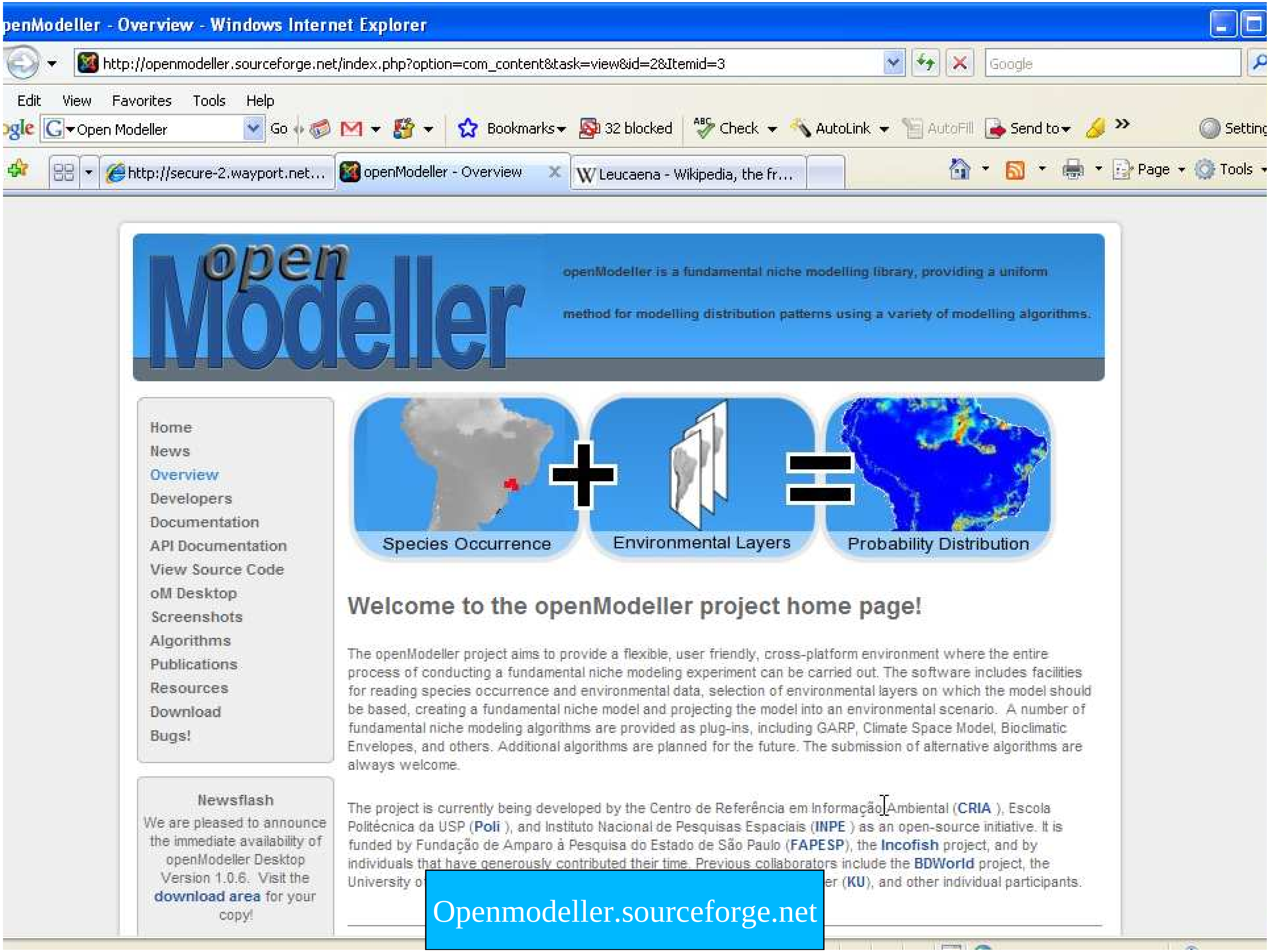
BDWorld Components

Submit scientific name; retrieve accepted name & synonyms for species



Model of climatic conditions where species is currently found

ite



Openmodeller.sourceforge.net

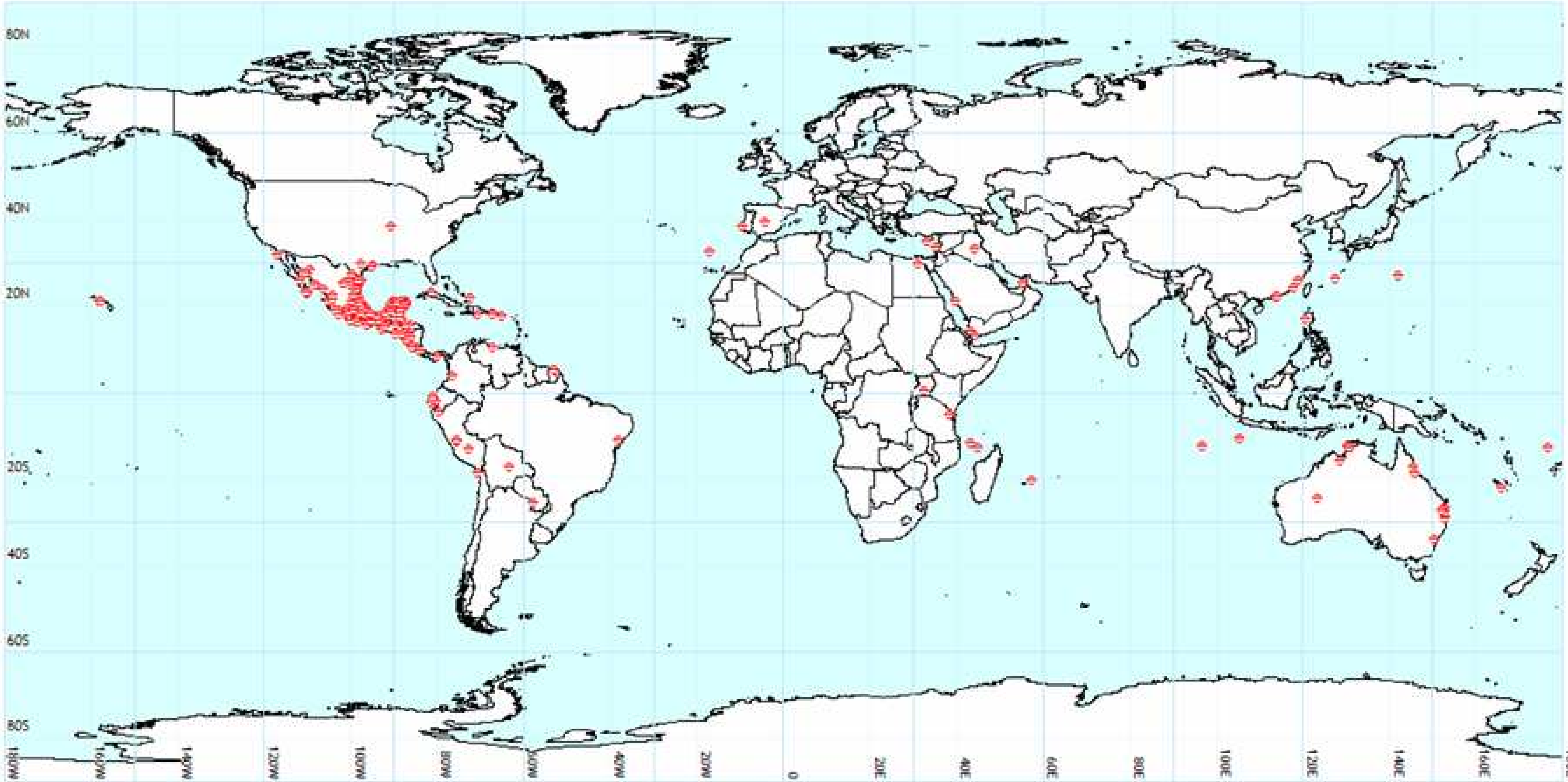
BioDiversity Questions

- How should conservation efforts be concentrated?
  - (example of Biodiversity Richness & Conservation Evaluation)
- Where might a species be expected to occur, under present or predicted climatic conditions?
  - (example of Bioclimatic & Ecological Niche Modelling)
- How can geographical information assist in inferring possible evolutionary pathways?
  - (example of Phylogenetic Analysis & Palaeoclimate Modelling)

Resource used in these biodiversity studies

- Data sources:
  - Catalogue of Life (names of species: Species 2000, GBIF)
  - Biodiversity data
    - Descriptive data
    - Distribution of specimens and observations
  - Geographical data
    - Boundaries of geographical & political units
    - Climate surfaces
  - Genetic sequences
- Analytic tools:
  - Biodiversity richness assessment – various metrics
  - Bioclimatic modelling – bioclimatic 'envelope' generation
  - Phylogenetic analysis (generation of phylogenetic trees)

Leucaena leucocephala (Lam.) De Wit.

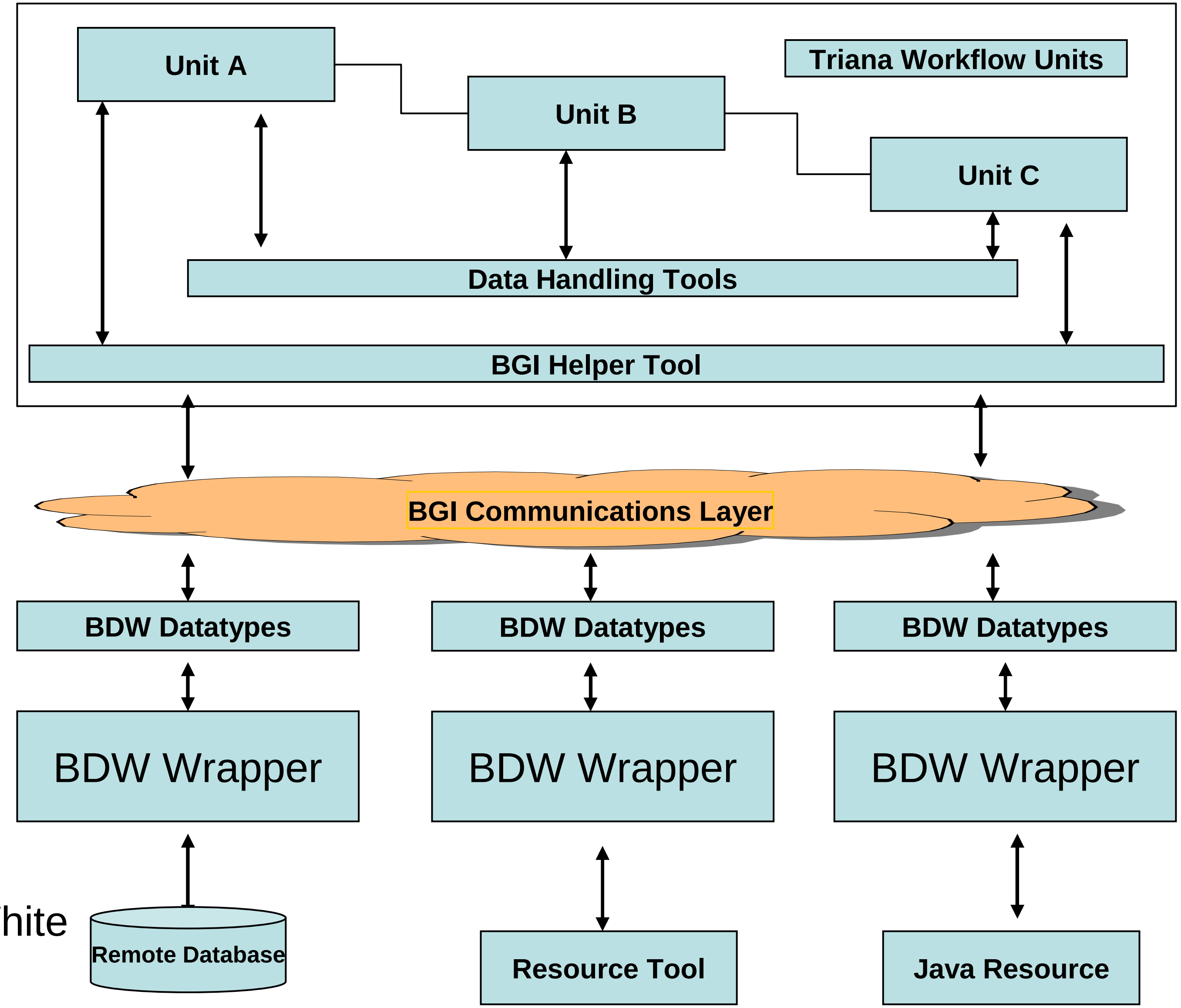


Point data from various herbaria

Taxonomic Database Working Group (TDWG)
WorldSat International, Inc.
Environmental Systems Research Institute, Inc. (ESRI)

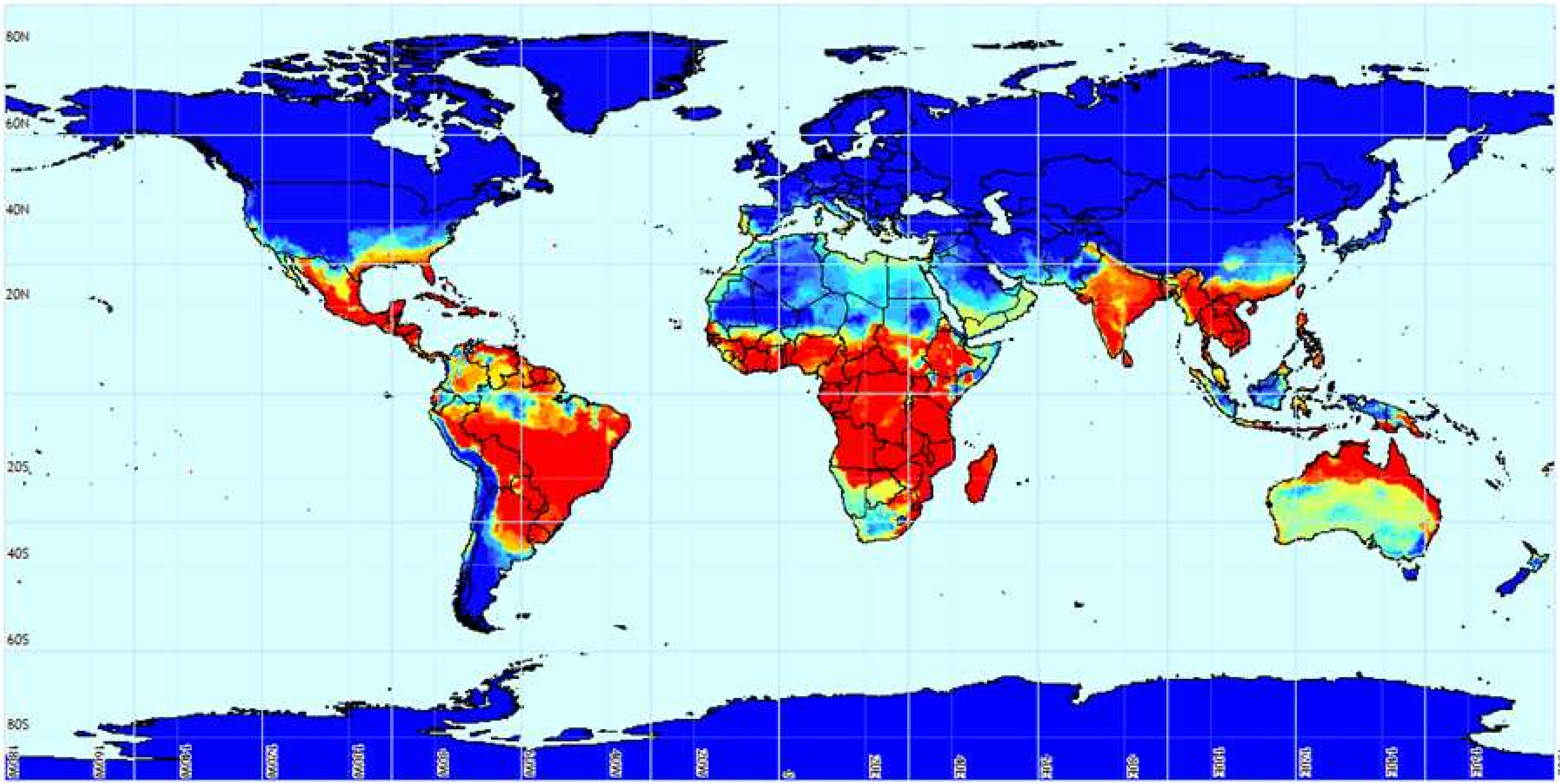
Specimen record

BDWorld Architecture



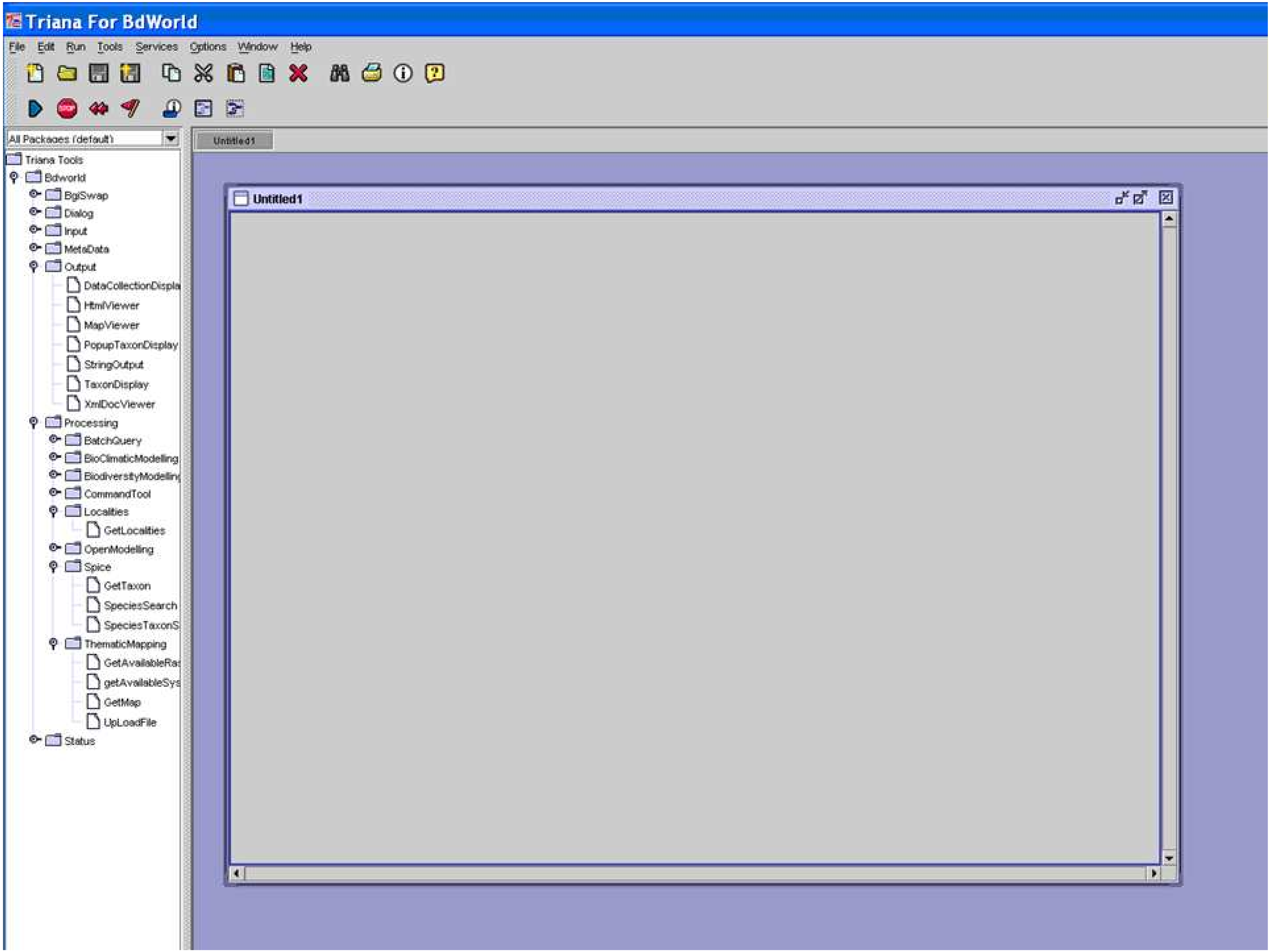
From: Richard White

Leucaena leucocephala (Lam.) De Wit.

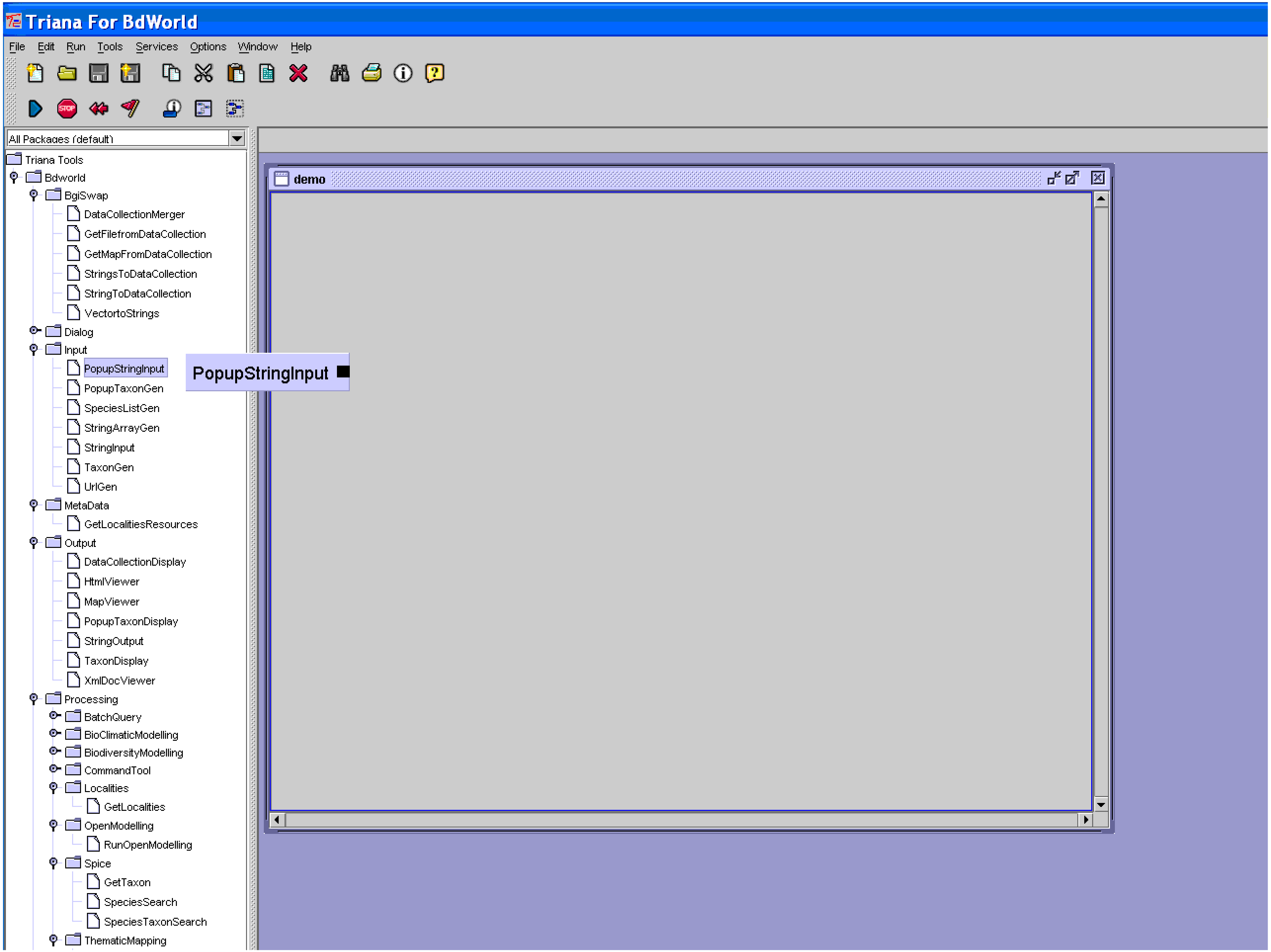


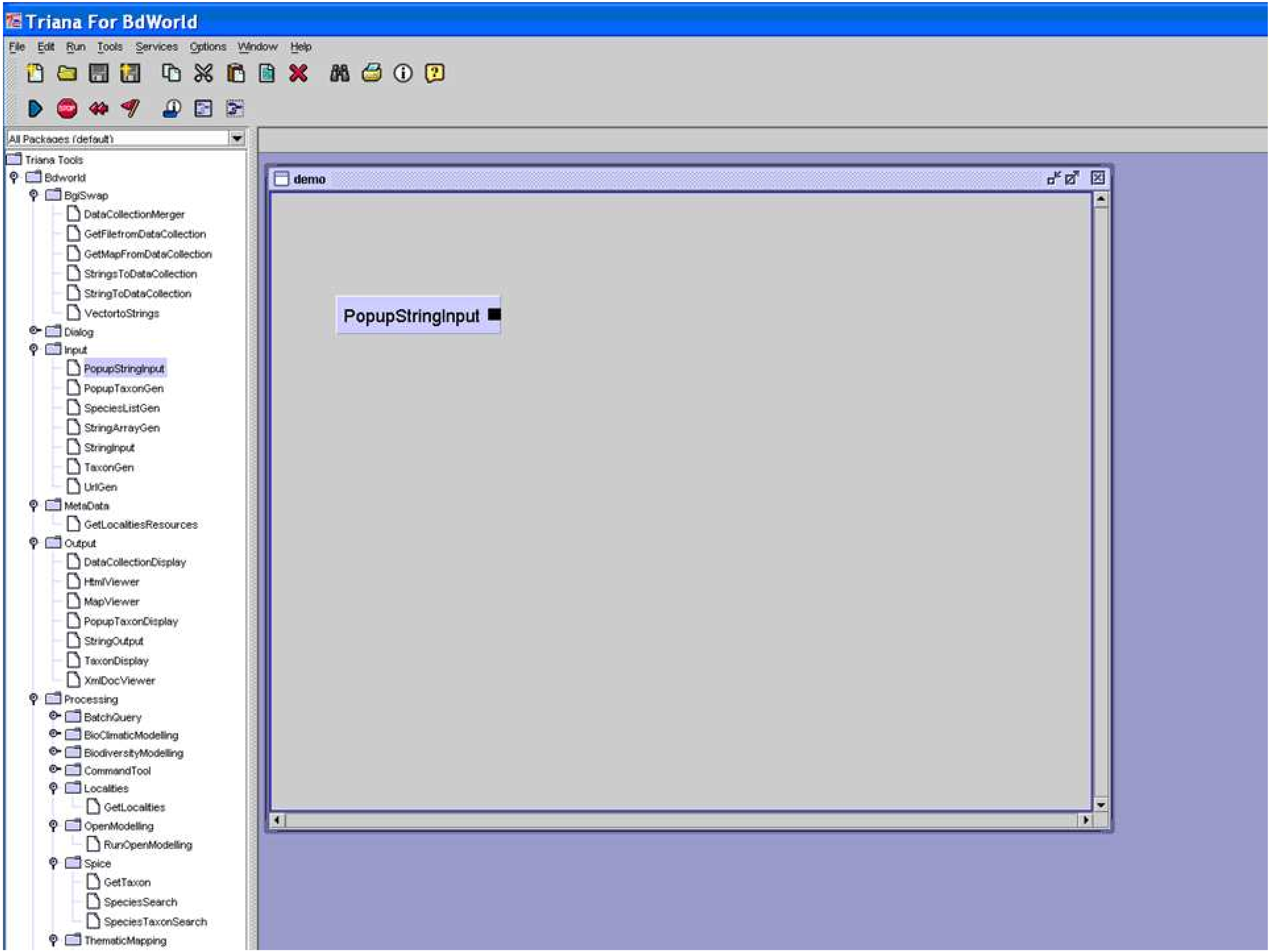
GARP prediction of climatic suitability

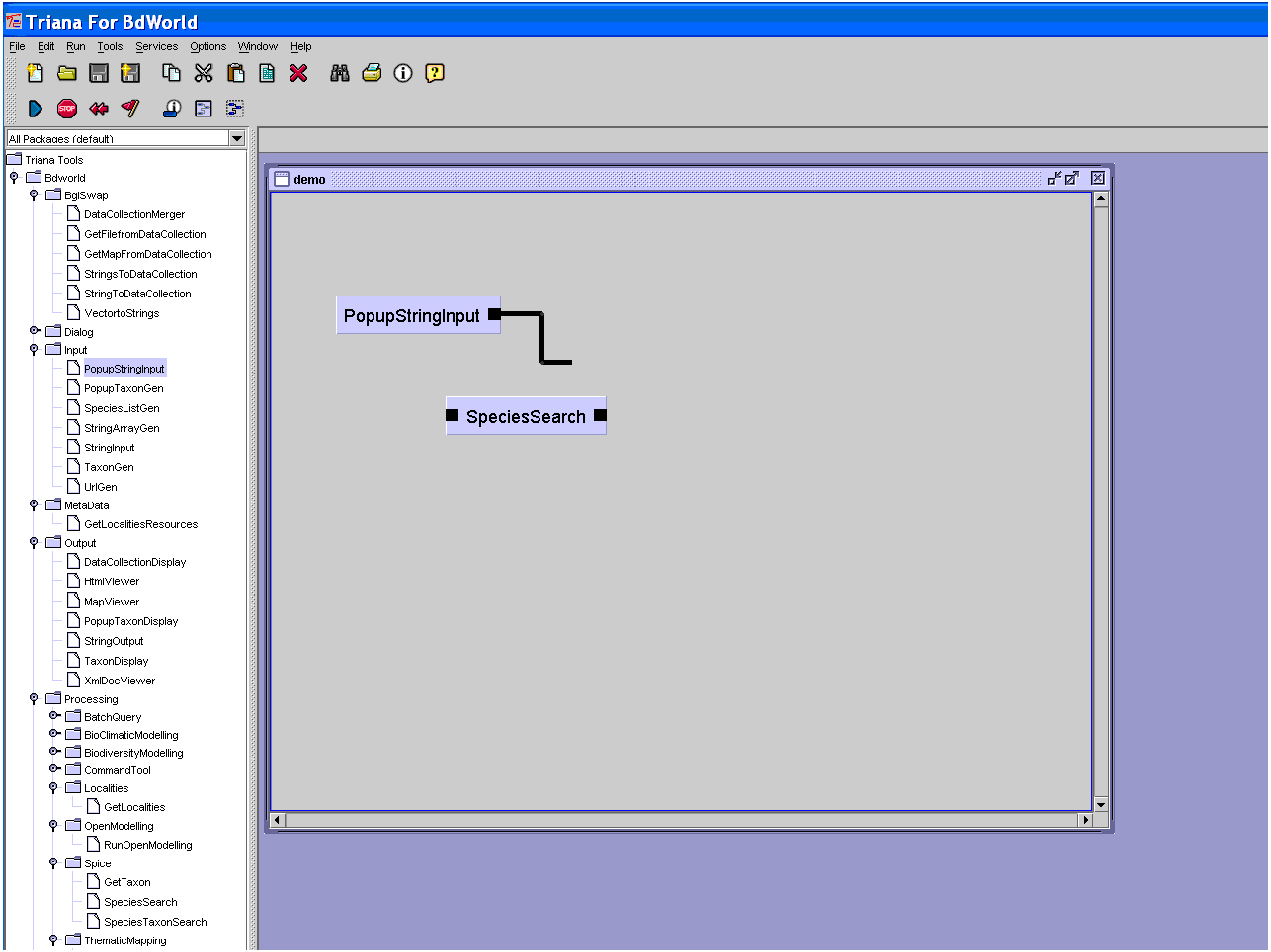
Taxonomic Database Working Group (TDWG)
WorldSat International, Inc.
Environmental Systems Research Institute, Inc. (ESRI)

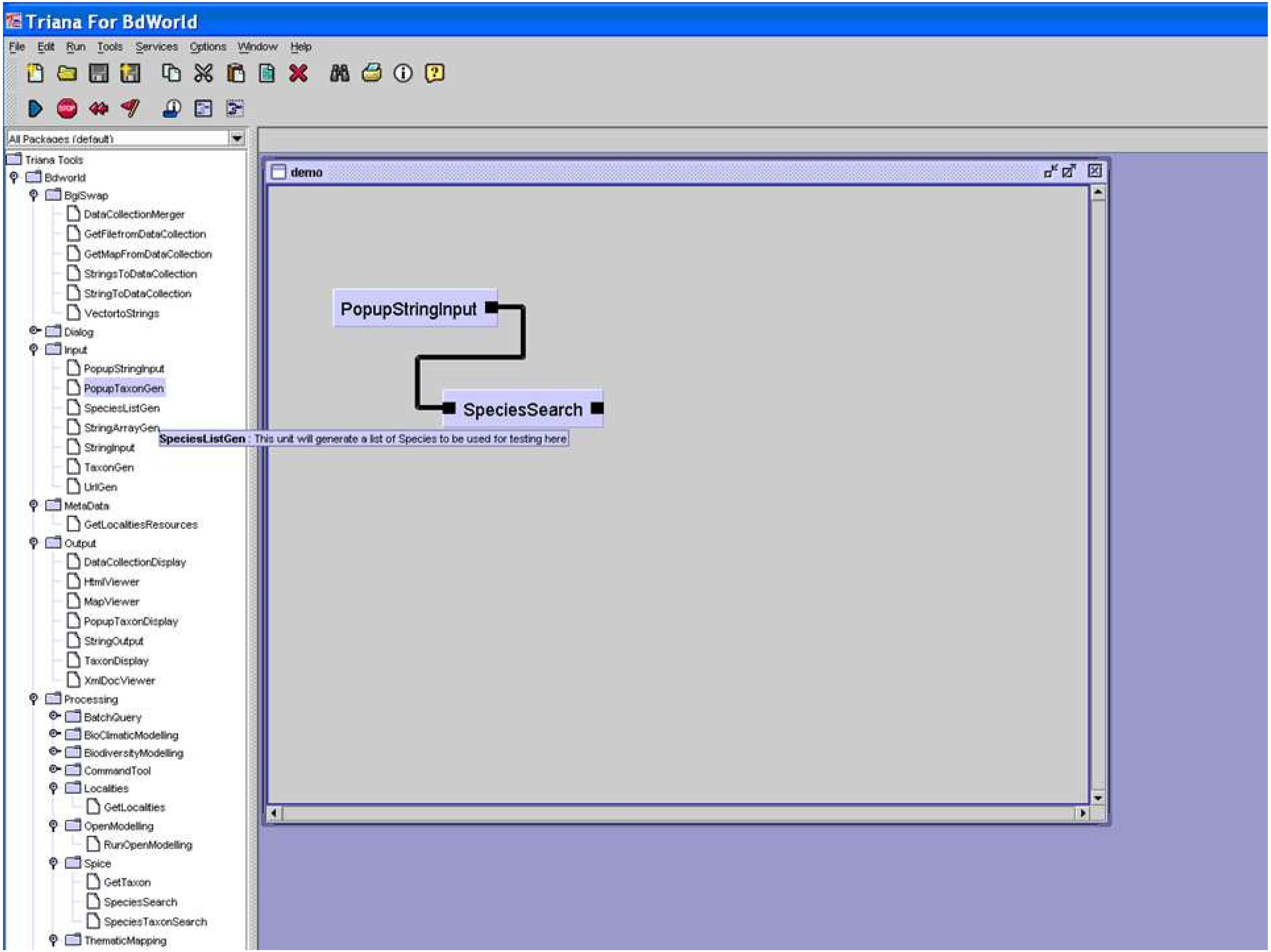


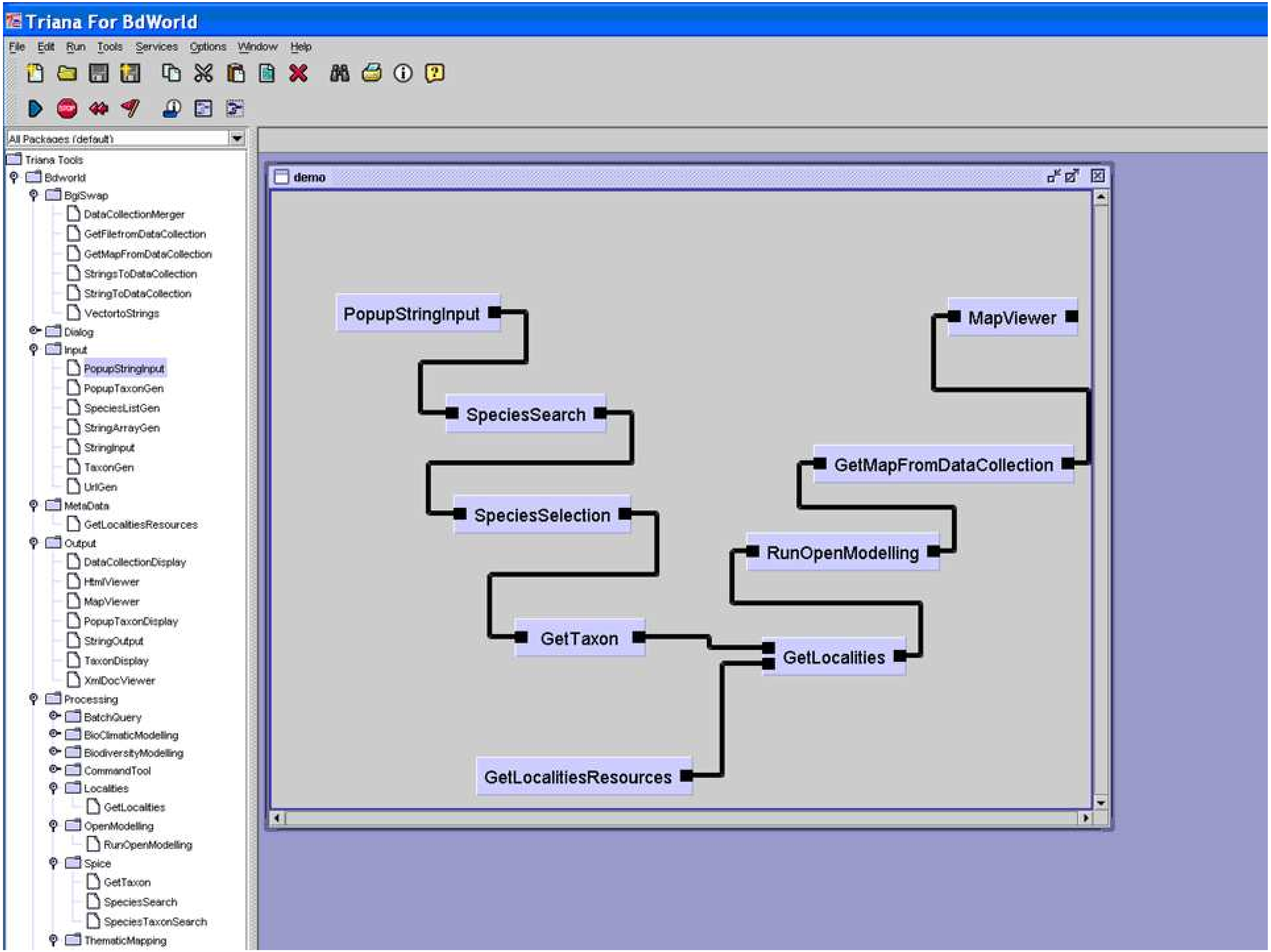
Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Untitled1
Triana Tools
Bdworld
BgiSwap
Dialog
Input
MetaData
Output
DataCollectionDispla
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModellin
CommandTool
Localities
GetLocalities
OpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonS
ThematicMapping
GetAvailableRa
getAvailableSys
GetMap
UpLoadFile
Status
Untitled1

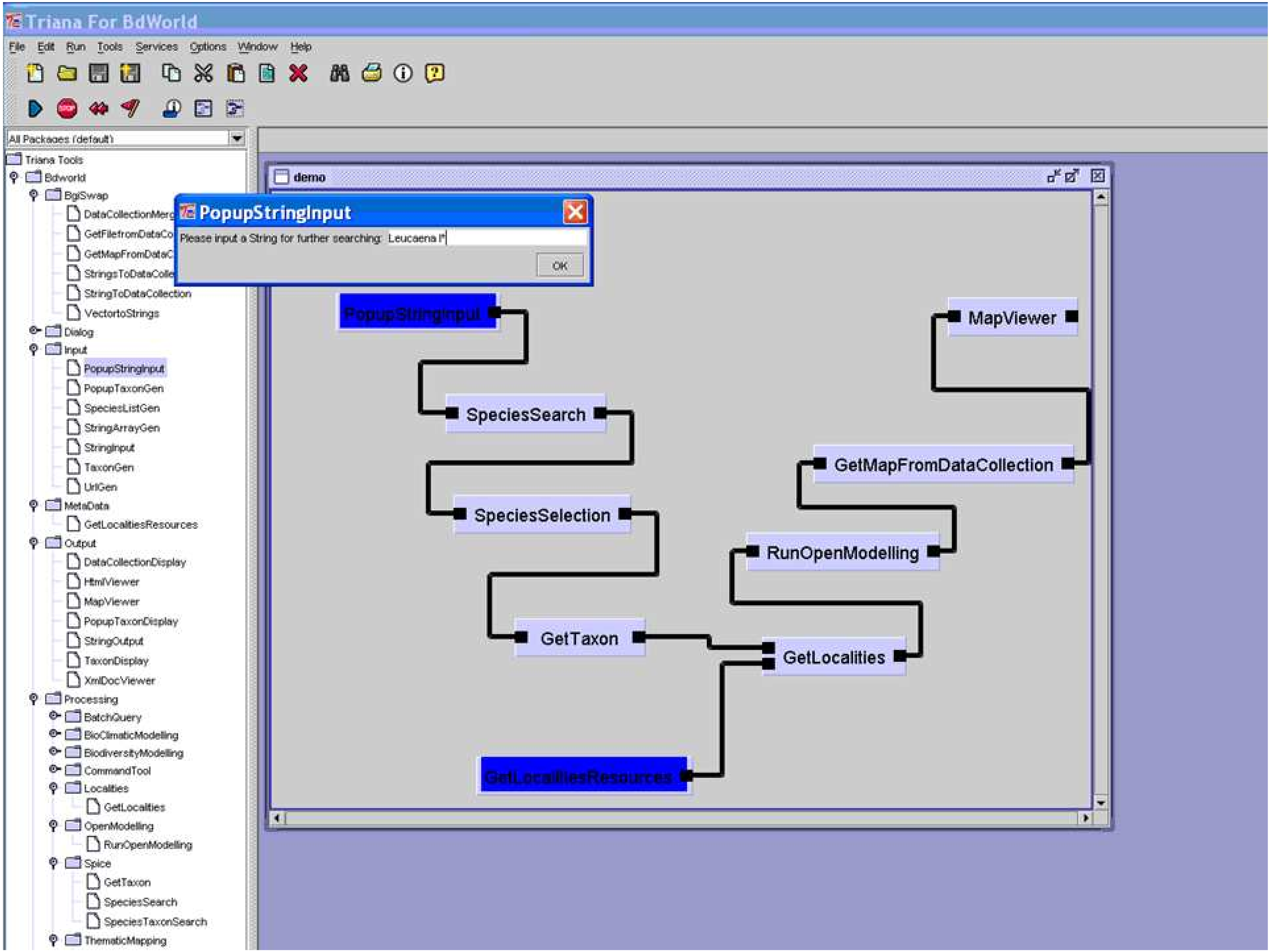

Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupStringInput

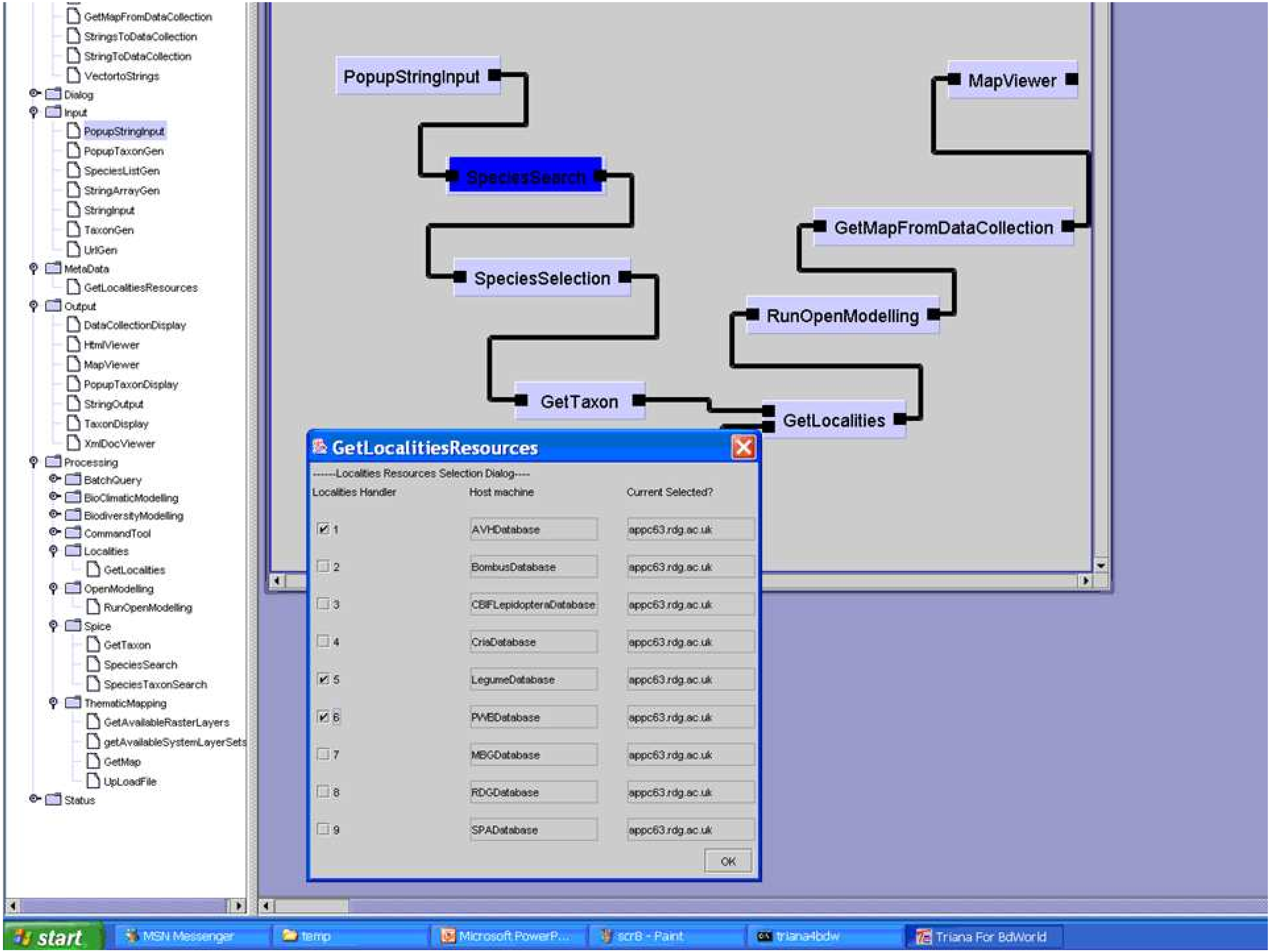

Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupStringInput


Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupStringInput
SpeciesSearch

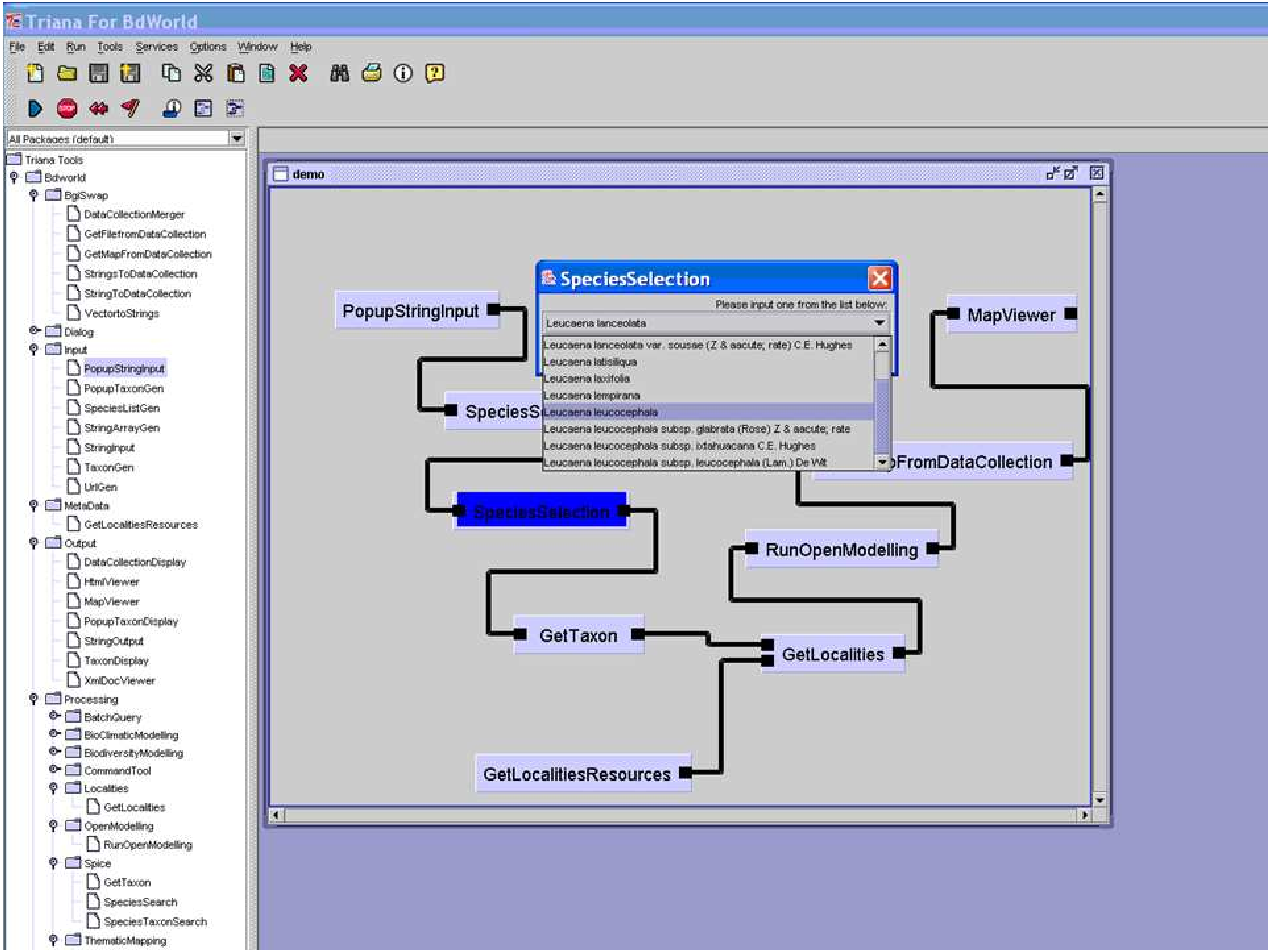

Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupStringInput
SpeciesSearch
SpeciesListGen : This unit will generate a list of Species to be used for testing here

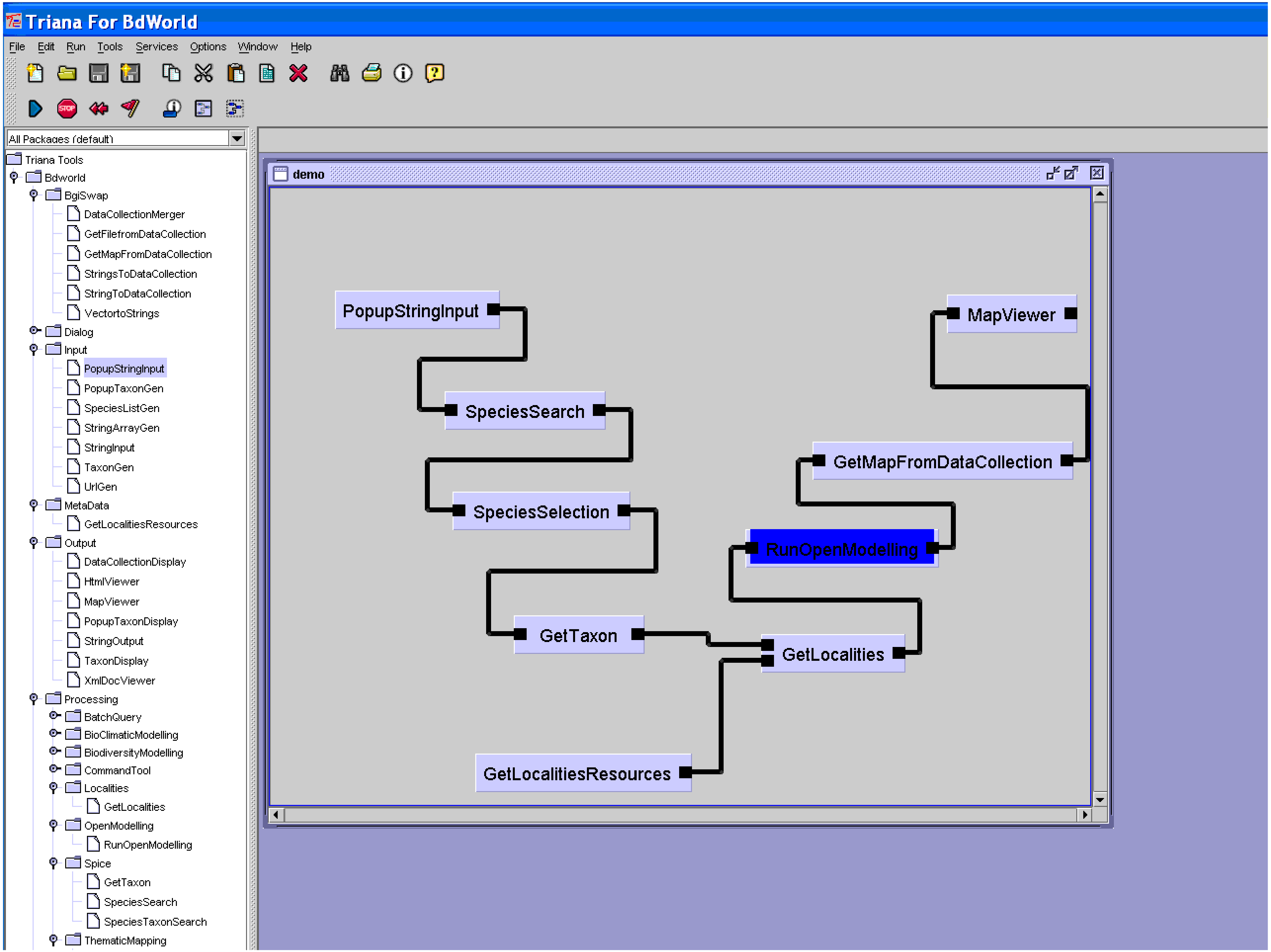

Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupStringInput
SpeciesSearch
SpeciesSelection
GetTaxon
GetLocalitiesResources
MapViewer
GetMapFromDataCollection
RunOpenModelling
GetLocalities


Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupStringInput
Please input a String for further searching: Leucaena l*
OK
PopupStringInput
SpeciesSearch
SpeciesSelection
GetTaxon
GetLocalities
GetLocalitiesResources
RunOpenModelling
GetMapFromDataCollection
MapViewer

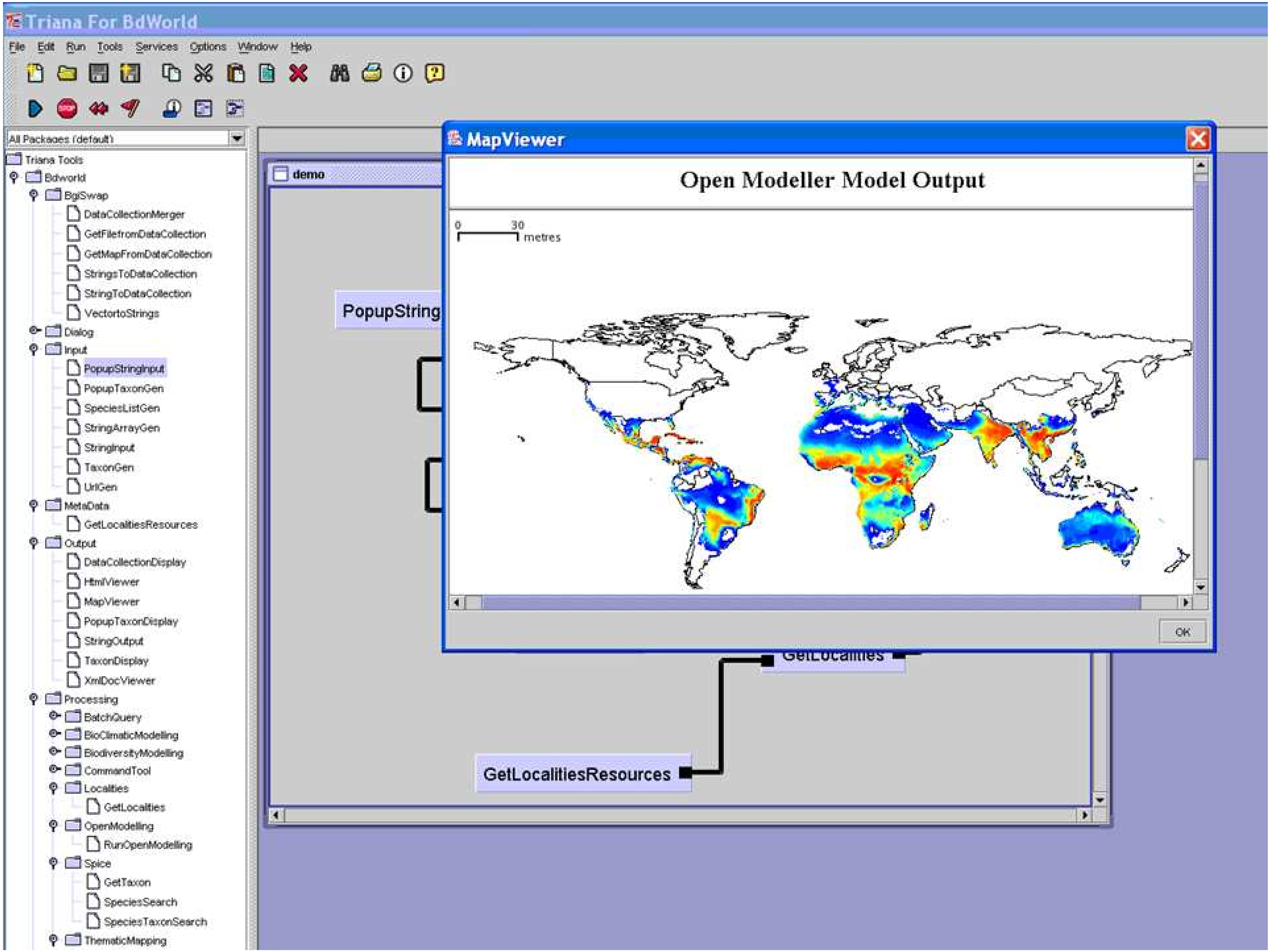


GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
GetAvailableRasterLayers
getAvailableSystemLayerSets
GetMap
UpLoadFile
Status
PopupStringInput
SpeciesSearch
SpeciesSelection
GetTaxon
MapViewer
GetMapFromDataCollection
RunOpenModelling
GetLocalities
GetLocalitiesResources
------Localities Resources Selection Dialog-----
Localities Handler
Host machine
Current Selected?
1 AVHDatabase appc63.rdg.ac.uk
2 BombusDatabase appc63.rdg.ac.uk
3 CBFLepidopteraDatabase appc63.rdg.ac.uk
4 CriaDatabase appc63.rdg.ac.uk
5 LegumeDatabase appc63.rdg.ac.uk
6 PWBDatabase appc63.rdg.ac.uk
7 MBGDatabase appc63.rdg.ac.uk
8 RDGDatabase appc63.rdg.ac.uk
9 SPADatabase appc63.rdg.ac.uk
OK
start
MSN Messenger
temp
Microsoft PowerP...
scr8 - Paint
triana4bdw
Triana For BdWorld

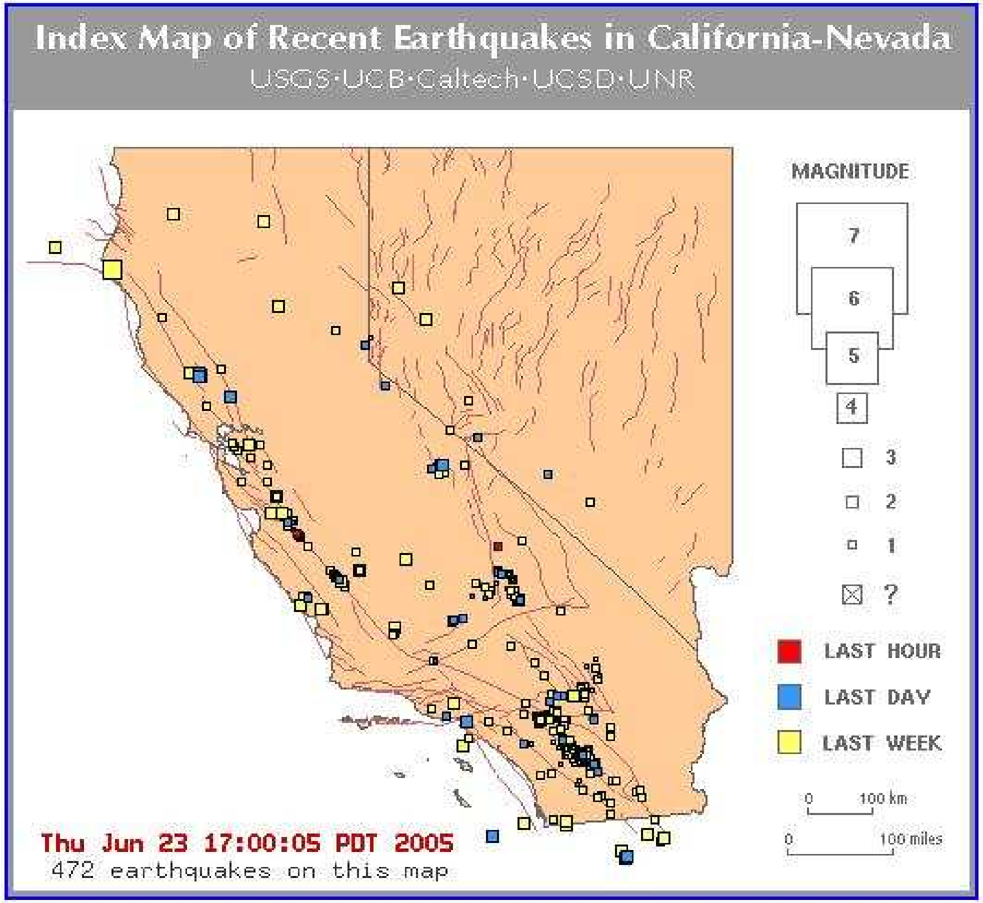

Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
SpeciesSelection
Please input one from the list below:
Leucaena lanceolata
Leucaena lanceolata var. sousae (Z & aacute; rate) C.E. Hughes
Leucaena latisiliqua
Leucaena laxifolia
Leucaena lempirana
Leucaena leucocephala
Leucaena leucocephala subsp. glabrata (Rose) Z & aacute; rate
Leucaena leucocephala subsp. ixtahuacana C.E. Hughes
Leucaena leucocephala subsp. leucocephala (Lam.) De Wit
PopupStringInput
MapViewer
SpeciesS
pFromDataCollection
SpeciesSelection
RunOpenModelling
GetTaxon
GetLocalities
GetLocalitiesResources



Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupStringInput
SpeciesSearch
SpeciesSelection
GetTaxon
GetLocalitiesResources
GetLocalities
RunOpenModelling
GetMapFromDataCollection
MapViewer


Triana For BdWorld
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Bdworld
BgiSwap
DataCollectionMerger
GetFilefromDataCollection
GetMapFromDataCollection
StringsToDataCollection
StringToDataCollection
VectortoStrings
Dialog
Input
PopupStringInput
PopupTaxonGen
SpeciesListGen
StringArrayGen
StringInput
TaxonGen
UrlGen
MetaData
GetLocalitiesResources
Output
DataCollectionDisplay
HtmlViewer
MapViewer
PopupTaxonDisplay
StringOutput
TaxonDisplay
XmlDocViewer
Processing
BatchQuery
BioClimaticModelling
BiodiversityModelling
CommandTool
Localities
GetLocalities
OpenModelling
RunOpenModelling
Spice
GetTaxon
SpeciesSearch
SpeciesTaxonSearch
ThematicMapping
demo
PopupString
GetLocalities
GetLocalitiesResources
MapViewer
Open Modeller Model Output
0
30
metres
OK

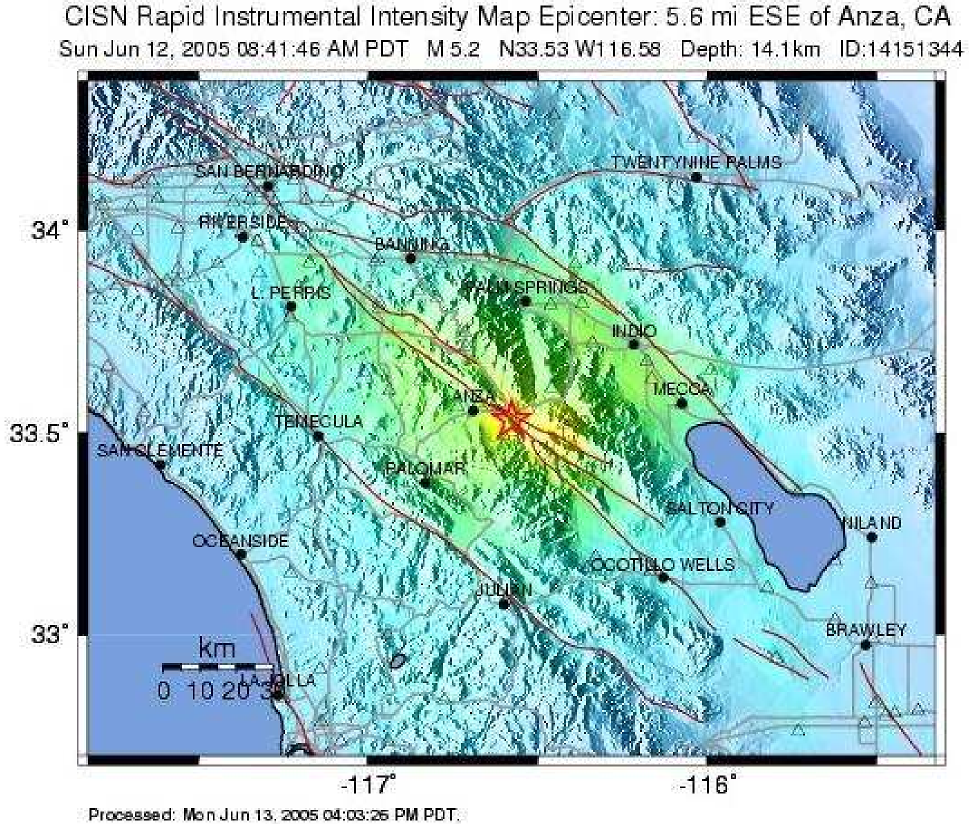
Workflows in Earthquake Engineering



Click on an earthquake on the map above to zoom in.

From: Philip Maechling

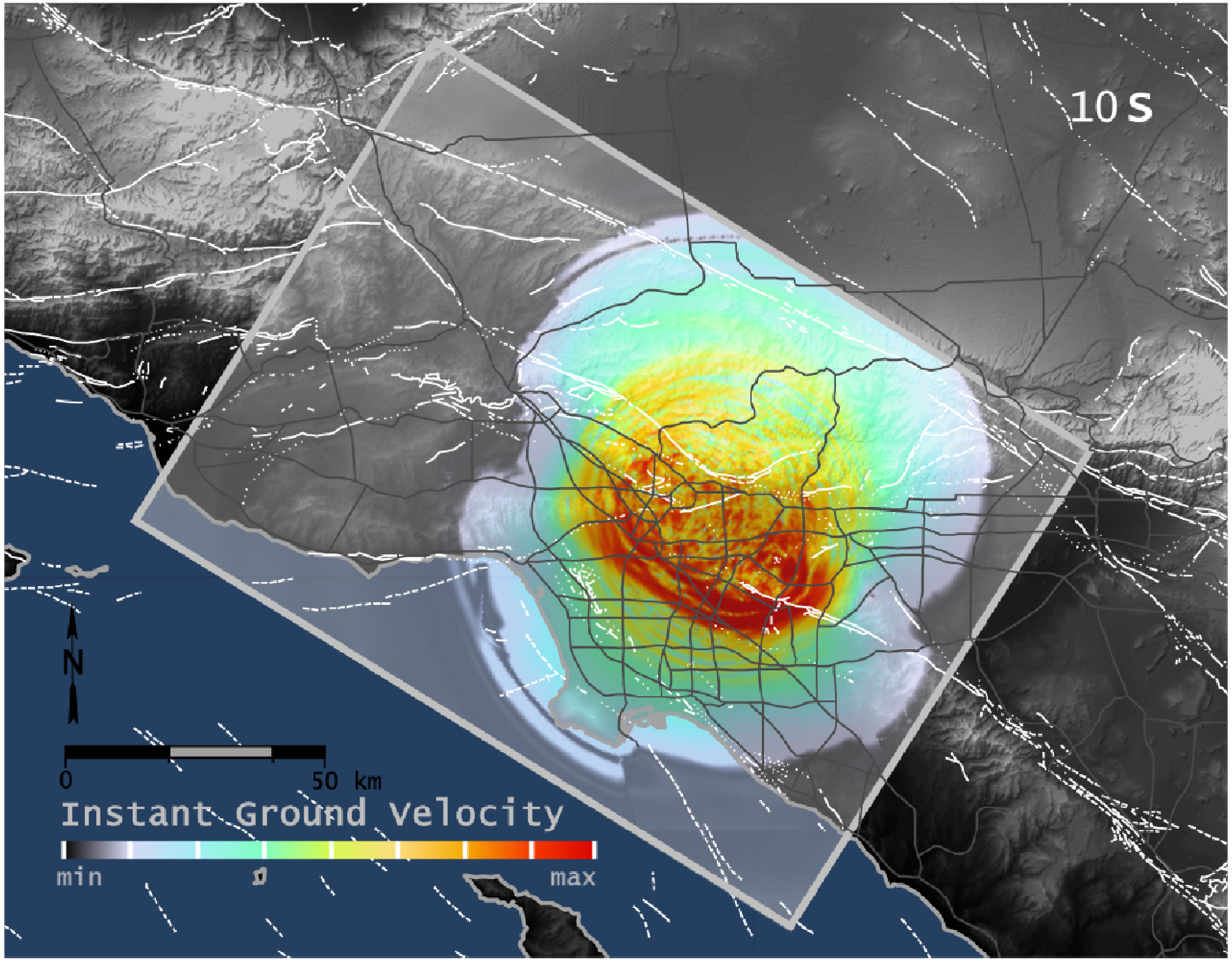
Observed Areas of Strong Ground Motion



PERCEIVED SHAKING	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
POTENTIAL DAMAGE	none	none	none	Very light	Light	Moderate	Moderate/Heavy	Heavy	Very Heavy
PEAK ACC.(%g)	<.17	.17-1.4	1.4-3.9	3.9-9.2	9.2-18	18-34	34-65	65-124	>124
PEAK VEL.(cm/s)	<0.1	0.1-1.1	1.1-3.4	3.4-8.1	8.1-16	16-31	31-60	60-116	>116
INSTRUMENTAL INTENSITY	I	II-III	IV	V	VI	VII	VIII	IX	X+

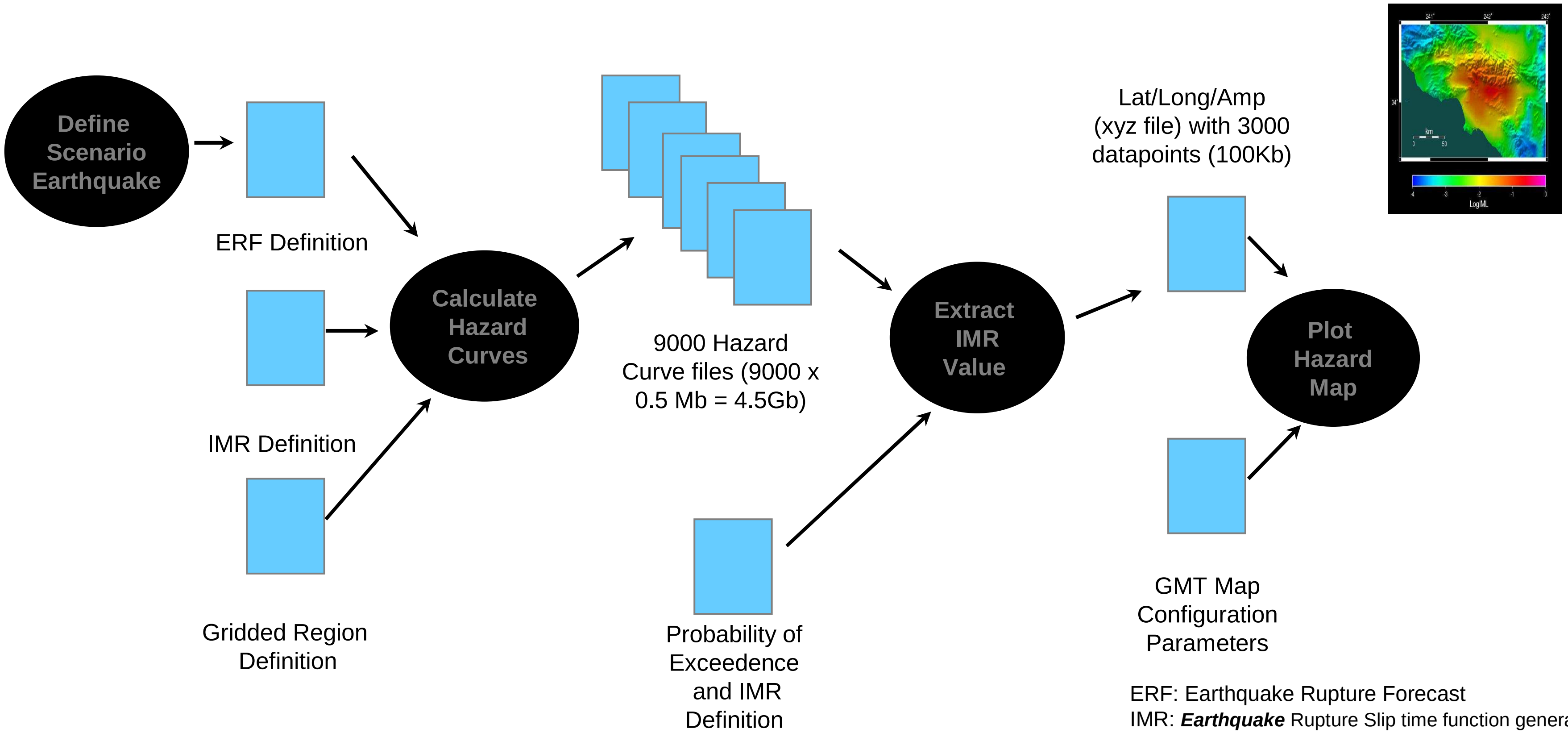
From: Philip Maechling

Simulations Supplement Observed Data



From: Philip Maechling

SCEC/CME Scientific Workflow Construction



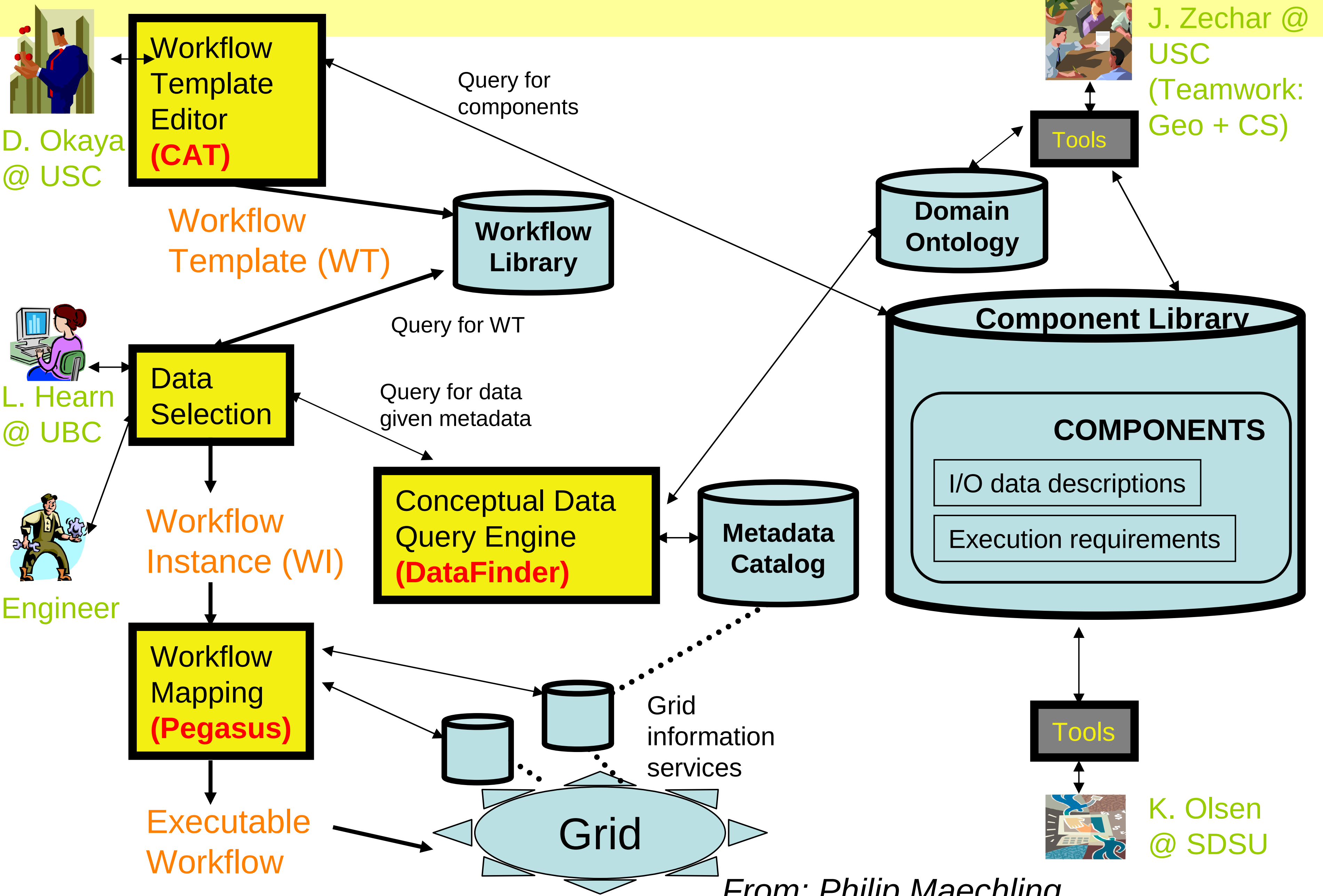
ERF: Earthquake Rupture Forecast
IMR: ***Earthquake*** Rupture Slip time function generator -

Intensity Measure Relationship
OpenSHA: Open Seismic Hazard Analysis
(http://www.opensha.org/index.html)

From: Philip Maechling

- In the first year, a Pathway-1 team led by Ned Field erected a new object-oriented architecture for seismic hazard analysis, dubbed "OpenSHA" (http://www.opensha.org/). This Java-based code implements a number of SHA conceptual objects, such as earthquake forecast models (EFM), intensity measure relationships (IMR), and intensity measure types (IMT). The team has thus far incorporated seven different IMR's that are applicable to Southern California and has developed an analysis tool that lets users explore the implications of the IMR's via a Web-enabled graphical user interface. The API between the IMR's and the analysis tool is very general and flexible, so that any new models can be plugged into the framework without having to change existing code. Along with the codes that calculate seismic hazard analysis curves, we have developed Web-based analysis tools that allow the user to explore the implications of combining various EFM's with a number of possible IMR's. OpenSHA will thus provide the platform for integrating the research done by the SCEC working group on Regional Earthquake Likelihood Models (RELM) (http://www.relm.org/). An overview of the OpenSHA architecture will be presented in a paper by N. Field, T. H. Jordan, and C. A. Cornell to be published soon in *SRL*.
- The OpenSHA framework has provided an interesting and challenging initial application of our KR&R technology. A Pathway-1 team comprising SCEC scientists and AI researchers from ISI has developed an initial knowledge base for SHA objects such as EFM's and IMR's using a powerful KR&R inference engine named PowerLoom (http://www.isi.edu/isd/LOOM/PowerLoom/). A Web-based tool called DOCKER (Distributed Operations of Code with Knowledge-based descriptions for Earthquake Research) was developed to allow users to define and perform Pathway-1 calculations by accessing the SHA knowledge base. As the user sets up a computational pathway by specifying the hazard-curve variables, DOCKER checks the user's selections for consistency by querying the SHA knowledge base and warns the user of inconsistencies. Moving the SHA system into the calculation of hazard maps will significantly increase the execution time, so this type of consistency checking should prove useful

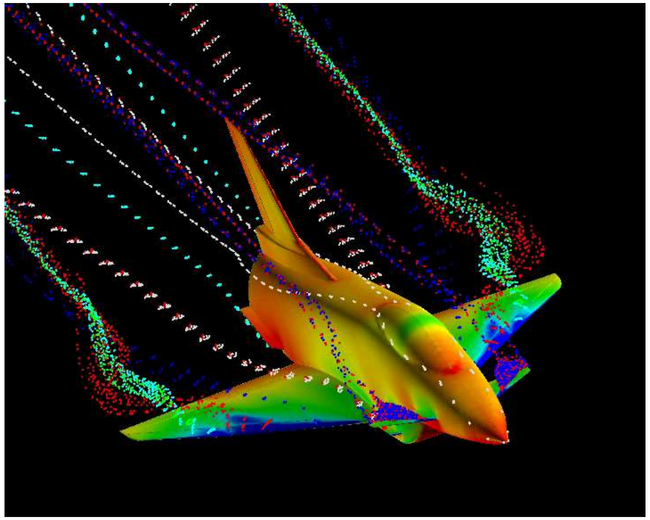
INTEGRATED WORKFLOW ARCHITECTURE



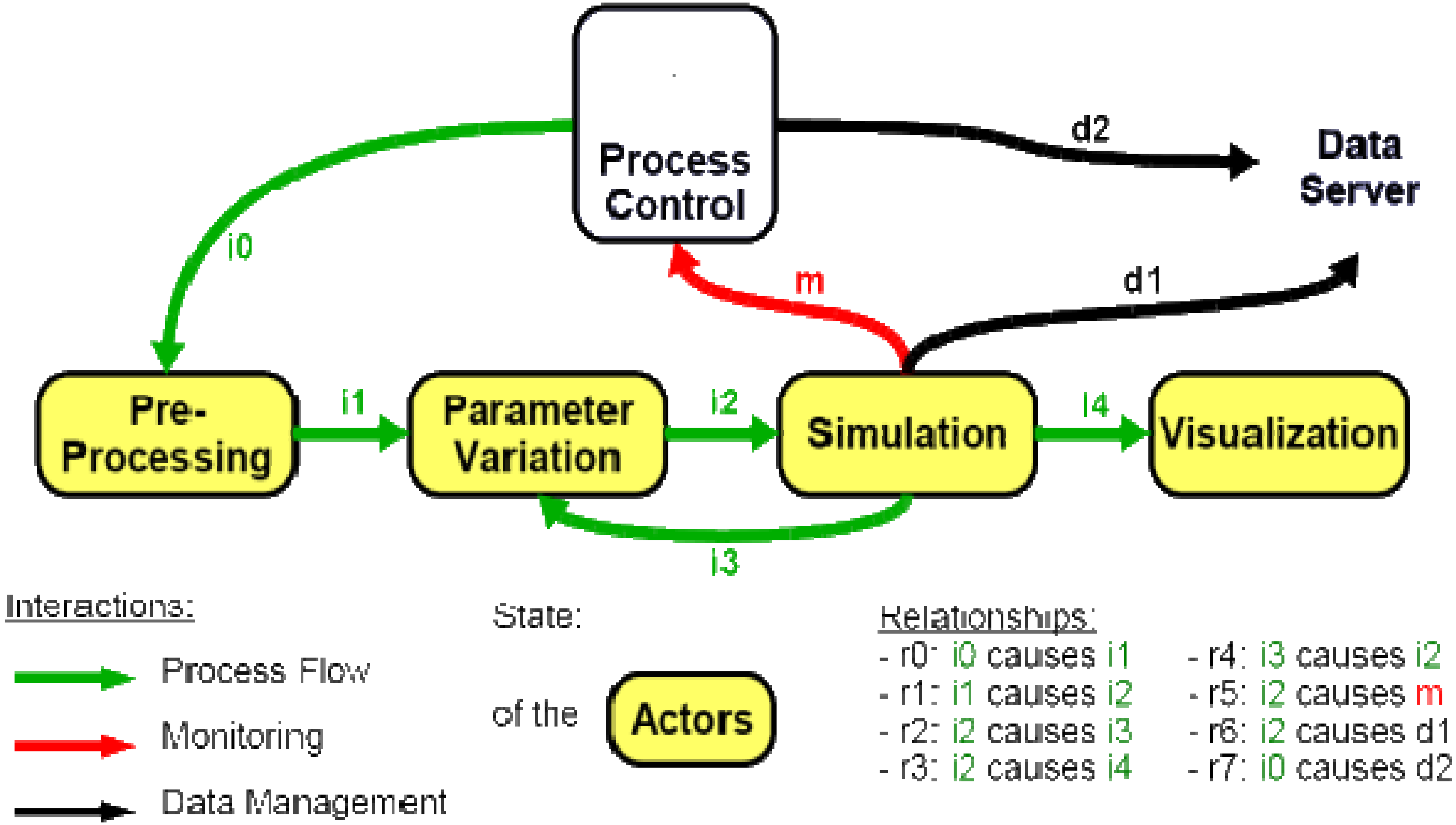
From: Philip Maechling

Flight Maneuver Simulation

- Project SikMa
  - Interactive simulation of a freely flying, fully configured, elastic warplane
- SikMa partners provided end-user requirements



Mapping: Workflow Example

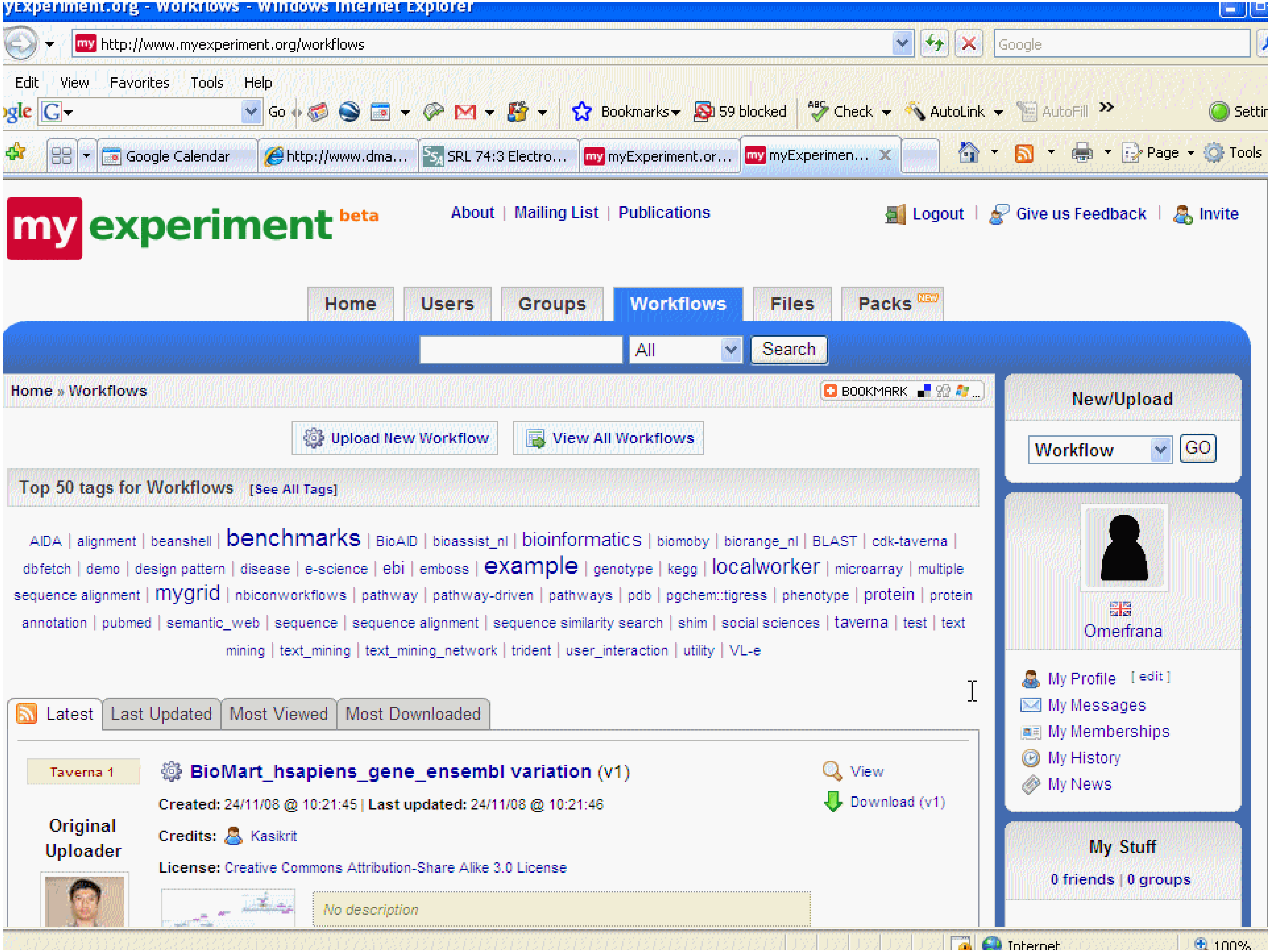


Scientific vs. Business Workflow

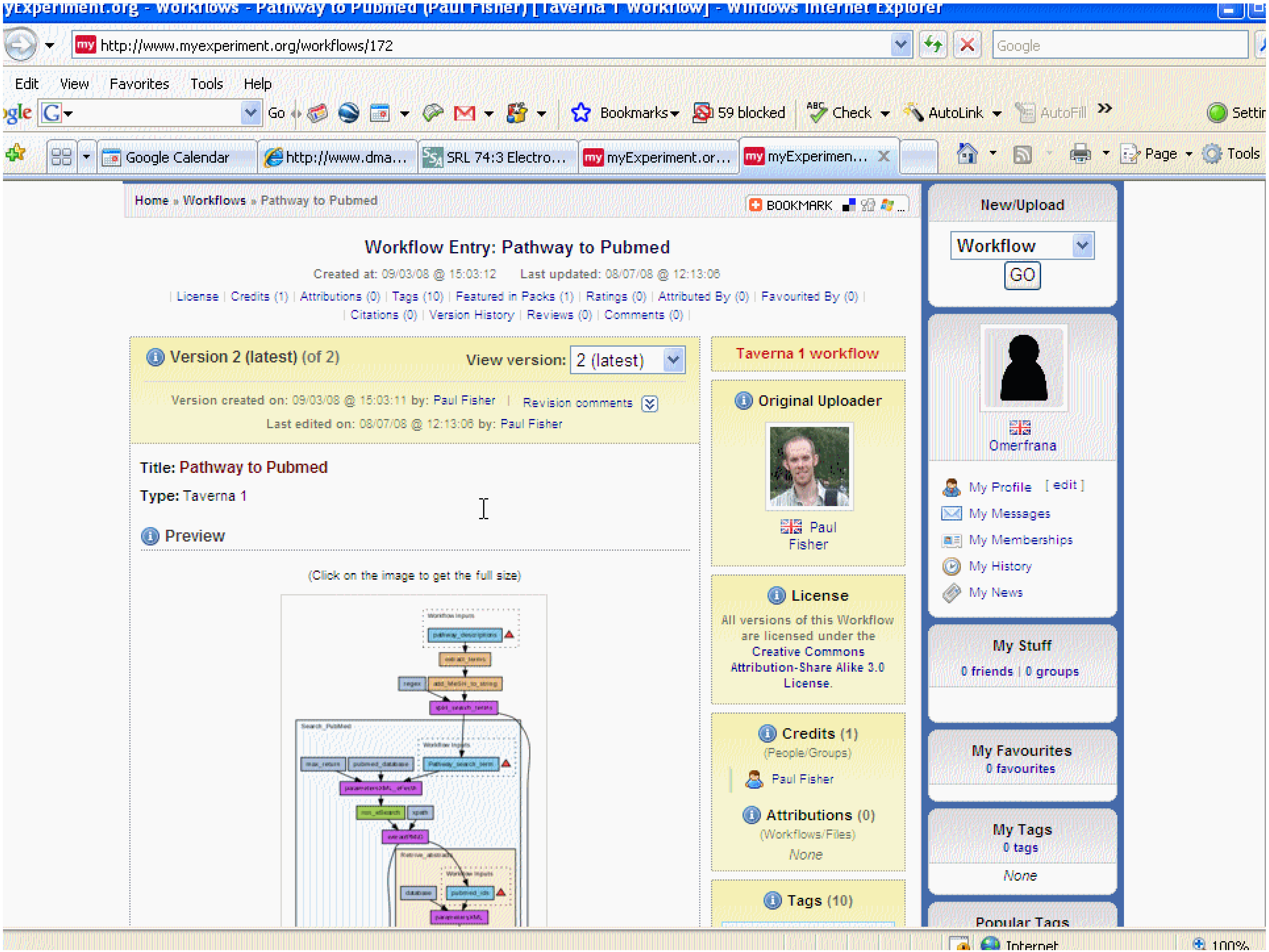
- **Reproducability** of results at the core of the scientific method
  - Create, manage and capture dynamic w/flows
- Scientific Exploration and User-Steering
  - Flexibility of design + **exploration** capabilities
  - Need to represent workflow variants (different workflow configurations and settings) – support complex scientific exploratory processes
  - Support for "user-steered" workflows
  - vs. more prescriptive use in business computing
- Emphasis on Data
  - **Heterogeneous** with different **access patterns**
  - Domain specific formats (textual, semi-structured, visual + varying degrees of annotations)

Scientific vs. Business Workflow … 2

- Common Characteristics
  - Repetitive cycle of data analysis and data migration
  - Parameter sweep or "range search" operations (can lead to creation of multiple jobs)
  - Dependencies between partial results generated through a workflow
  - Data aggregation across different repositories – often in different formats
  - Composition of complex data capture + simulation engines in a single (often linear) pipeline



yExperiment.org - Workflows - Windows Internet Explorer
http://www.myexperiment.org/workflows
Google
Edit View Favorites Tools Help
Go
Bookmarks
59 blocked
Check
AutoLink
AutoFill
Google Calendar
http://www.dma...
SRL 74:3 Electro...
myExperiment.or...
myExperimen...
Page
Tools
my experiment beta
About | Mailing List | Publications
Logout | Give us Feedback | Invite
Home
Users
Groups
Workflows
Files
Packs NEW
All
Search
Home » Workflows
BOOKMARK
Upload New Workflow
View All Workflows
Top 50 tags for Workflows [See All Tags]
AIDA | alignment | beanshell | benchmarks | BioAID | bioassist_nl | bioinformatics | biomoby | biorange_nl | BLAST | cdk-taverna | dbfetch | demo | design pattern | disease | e-science | ebi | emboss | example | genotype | kegg | localworker | microarray | multiple sequence alignment | mygrid | nbiconworkflows | pathway | pathway-driven | pathways | pdb | pgchem::tigress | phenotype | protein | protein annotation | pubmed | semantic_web | sequence | sequence alignment | sequence similarity search | shim | social sciences | taverna | test | text mining | text_mining | text_mining_network | trident | user_interaction | utility | VL-e
Latest
Last Updated
Most Viewed
Most Downloaded
Taverna 1
BioMart_hsapiens_gene_ensembl variation (v1)
View
Download (v1)
Created: 24/11/08 @ 10:21:45 | Last updated: 24/11/08 @ 10:21:46
Credits: Kasikrit
License: Creative Commons Attribution-Share Alike 3.0 License
Original Uploader
No description
New/Upload
Workflow
GO
Omerfrana
My Profile [edit]
My Messages
My Memberships
My History
My News
My Stuff
0 friends | 0 groups
Internet
100%

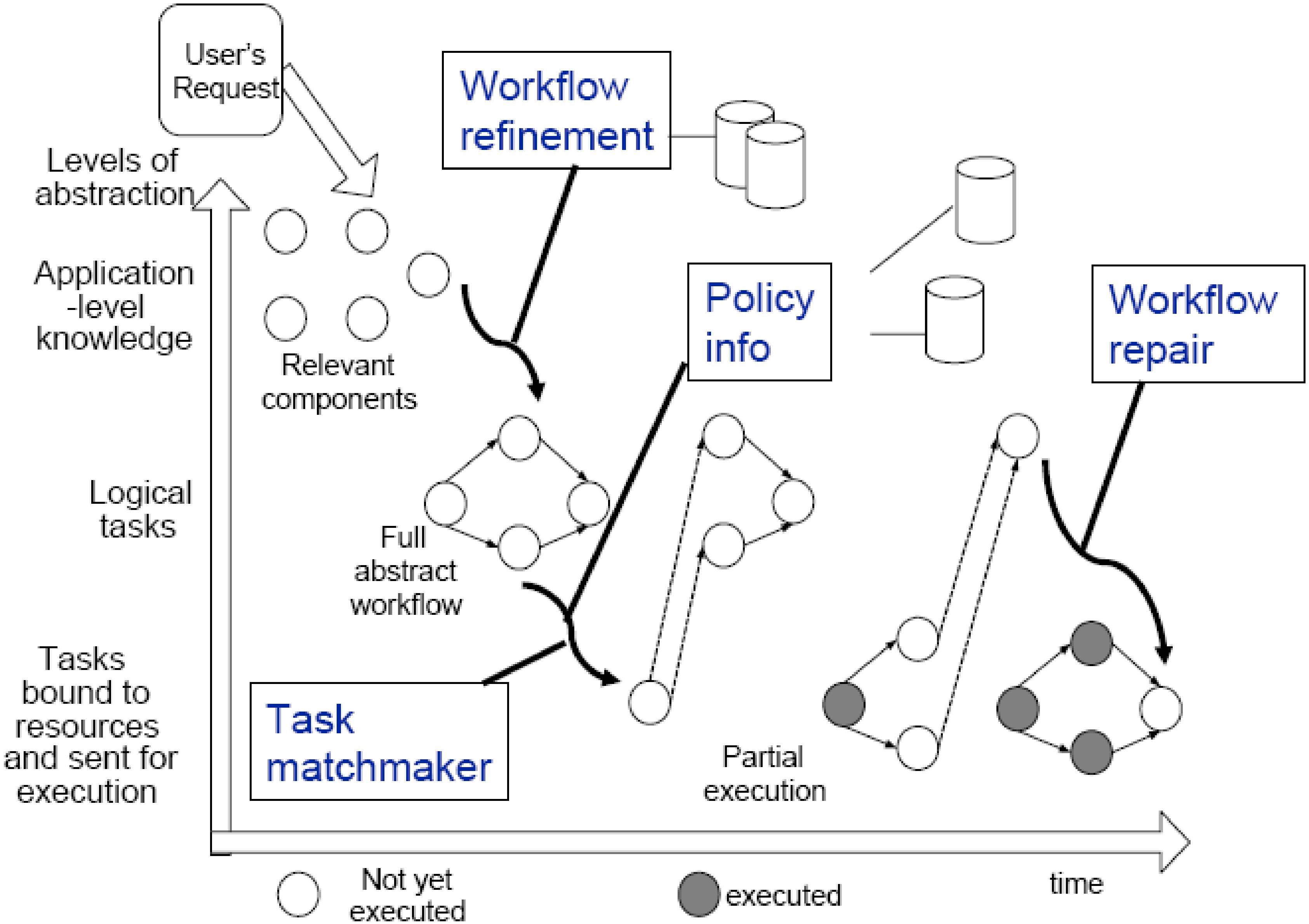

myExperiment.org - Workflows - Pathway to Pubmed (Paul Fisher) [Taverna 1 Workflow] - Windows Internet Explorer
http://www.myexperiment.org/workflows/172
Edit View Favorites Tools Help
Go Bookmarks 59 blocked Check AutoLink AutoFill
Google Calendar
http://www.dma...
SRL 74:3 Electro...
myExperiment.or...
myExperimen...
Page Tools
Home » Workflows » Pathway to Pubmed
BOOKMARK
Workflow Entry: Pathway to Pubmed
Created at: 09/03/08 @ 15:03:12 Last updated: 08/07/08 @ 12:13:06
License | Credits (1) | Attributions (0) | Tags (10) | Featured in Packs (1) | Ratings (0) | Attributed By (0) | Favourited By (0)
Citations (0) | Version History | Reviews (0) | Comments (0)
Version 2 (latest) (of 2)
View version: 2 (latest)
Version created on: 09/03/08 @ 15:03:11 by: Paul Fisher | Revision comments
Last edited on: 08/07/08 @ 12:13:06 by: Paul Fisher
Title: Pathway to Pubmed
Type: Taverna 1
Preview
(Click on the image to get the full size)
Taverna 1 workflow
Original Uploader
Paul Fisher
License
All versions of this Workflow are licensed under the Creative Commons Attribution-Share Alike 3.0 License.
Credits (1)
(People/Groups)
Paul Fisher
Attributions (0)
(Workflows/Files)
None
Tags (10)
New/Upload
Workflow
GO
Omerfrana
My Profile [edit]
My Messages
My Memberships
My History
My News
My Stuff
0 friends | 0 groups
My Favourites
0 favourites
My Tags
0 tags
None
Popular Tags
Internet
100%

Workflow Lifecycle

From: Aleksander Slominski

- Design
  - Typical workflow is graph oriented (ease of use)
  - Language: how expressive is workflow
  - GUI: Visual Service Composition Environment
- Deployment
  - Workflow Description is sent to Workflow Engine
  - Possibly validated and compiled
- Execution
  - Workflow Engine enacts Workflow Description
- Monitoring
  - Events reflecting from workflow and services execution
- Refinement

Workflow refinement and execution (Ewa Deelman)

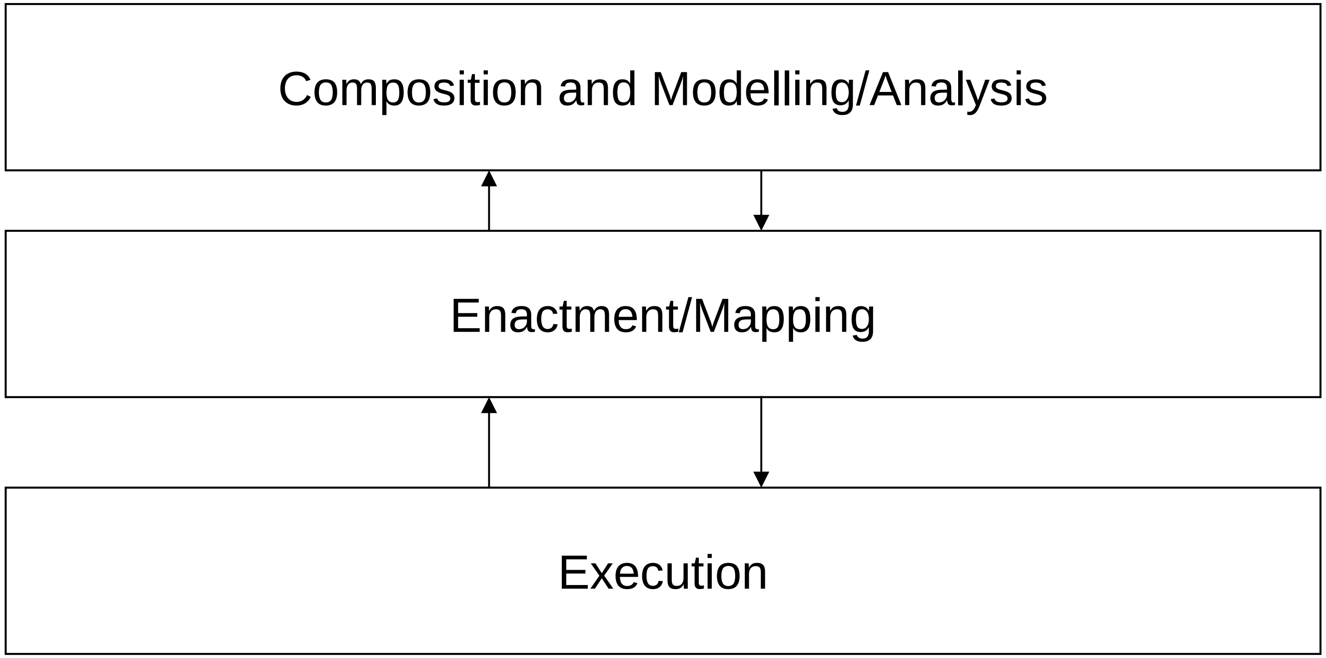


Workflow Representation

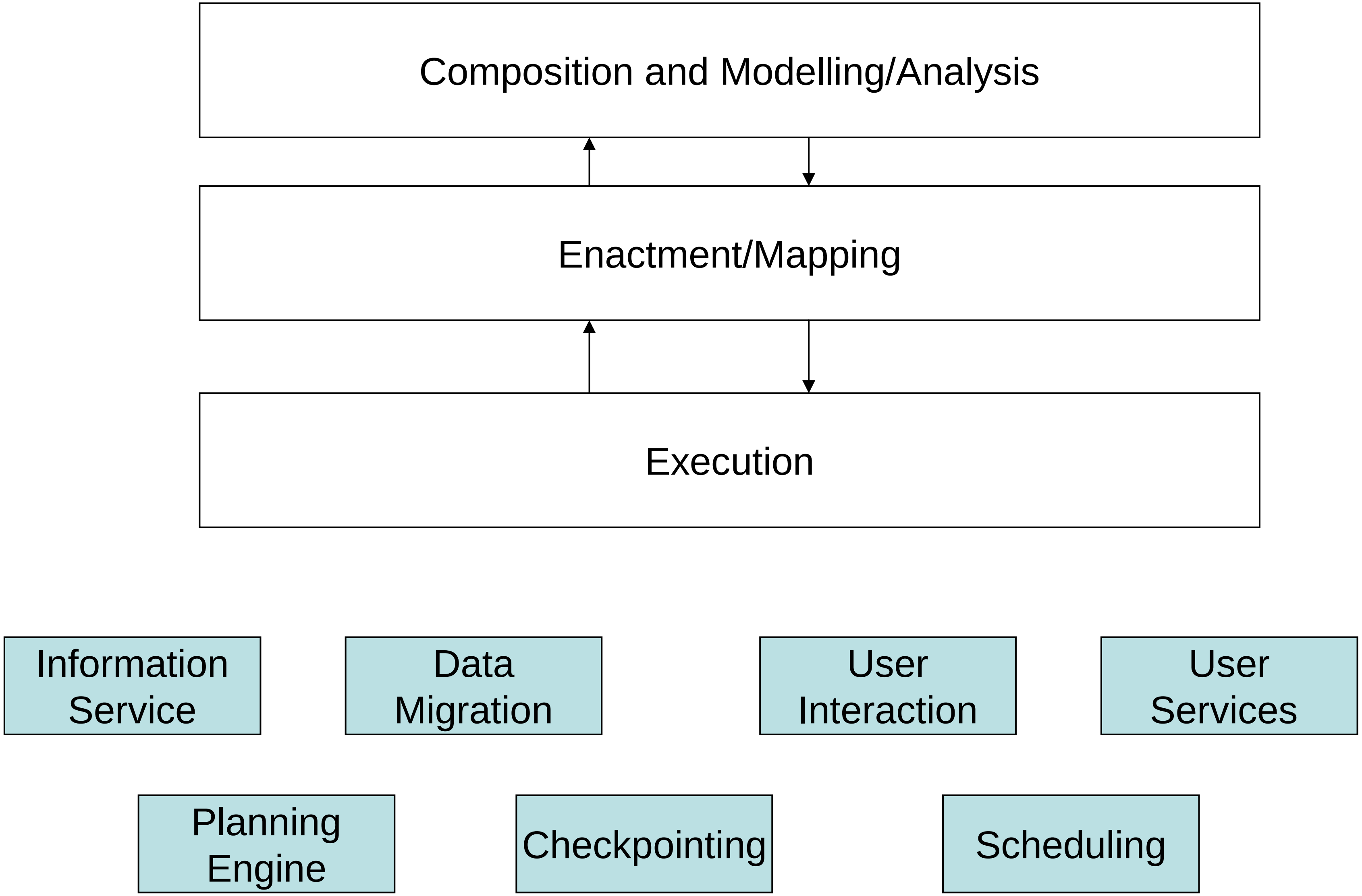
- Abstract ("design time") workflow
  - Task graph encoding data flow or control flow dependencies
  - "Scientific" reproducability
- "Concrete" (run time) workflow
  - Service bindings to an abstract workflow graph
  - "Engineering" reproducability

- Sharing of graph structures, rather than just services
  - Limited case: "composite services"

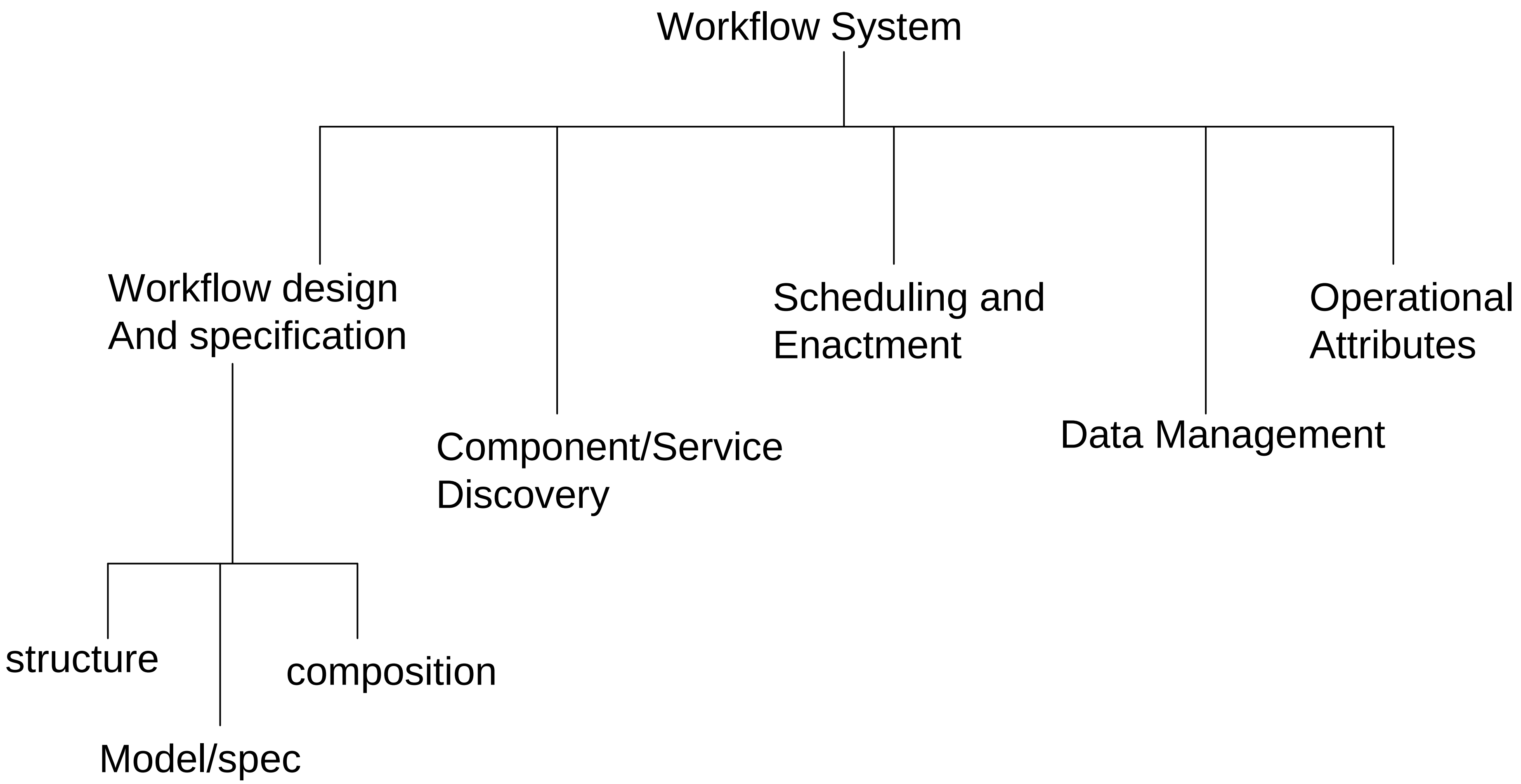
Workflow System Architecture



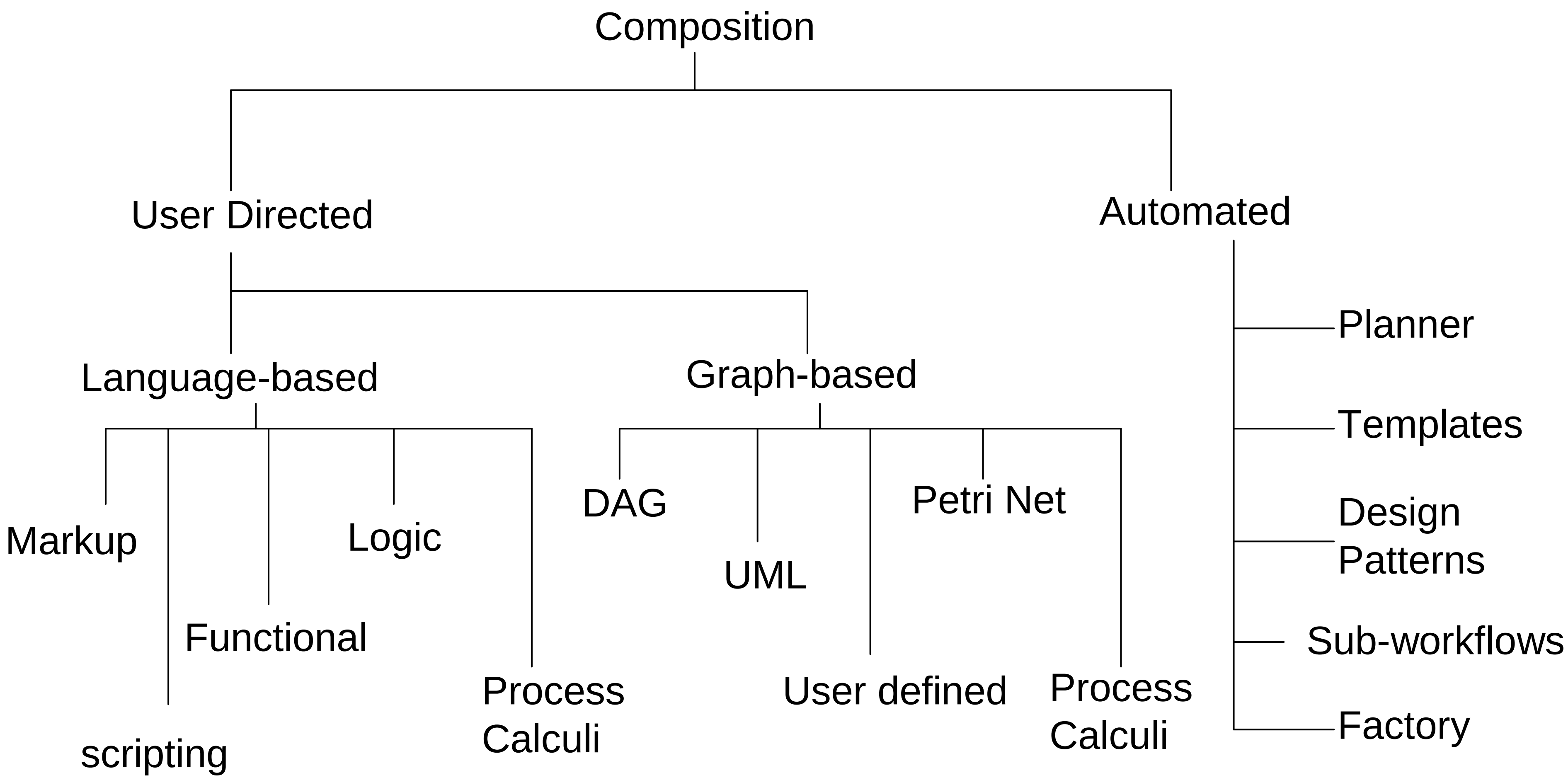
Workflow System Architecture



Workflow Taxonomy



Workflow Composition

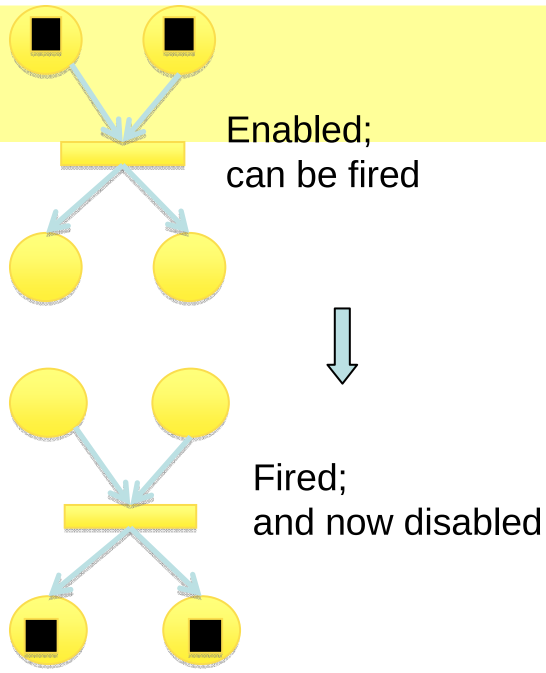


Textual: BPEL, SCUFL (Taverna), DAGMan, DAX
Graphical: Taverna, Triana, Kepler, SciRun, VisTrails
Planner/Semantics: Wings/Pegasus, IXI, Taverna/FETA

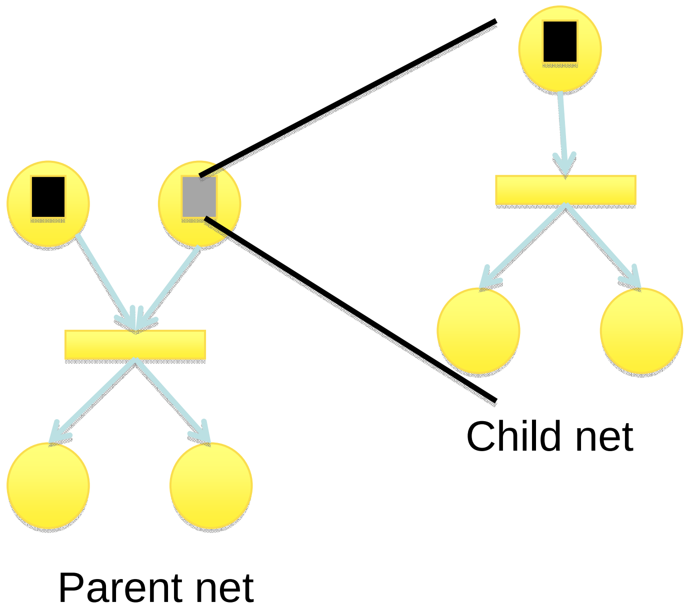
UML: Askalon
(UML Activity Dia.)

Petri nets

- Petri nets (informally)
  - Directed cyclic graph
  - 2 types of nodes: places and transitions
  - Arcs: place-transition, transition-place
  - Tokens: move on the graph, enable/fire transitions

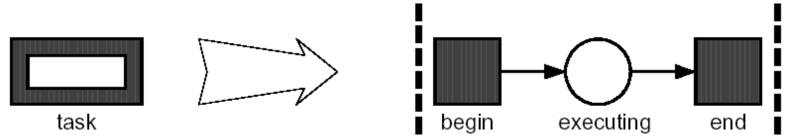


- **Reference nets**
  - Tokens can be nets
  - Nested structures: Parent and child nets
  - Dynamic creation of tokens
  - Synchronous channels

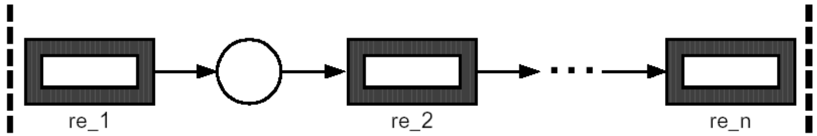


Petri Nets
(from *van der Aalst und Kumar, 2000*)

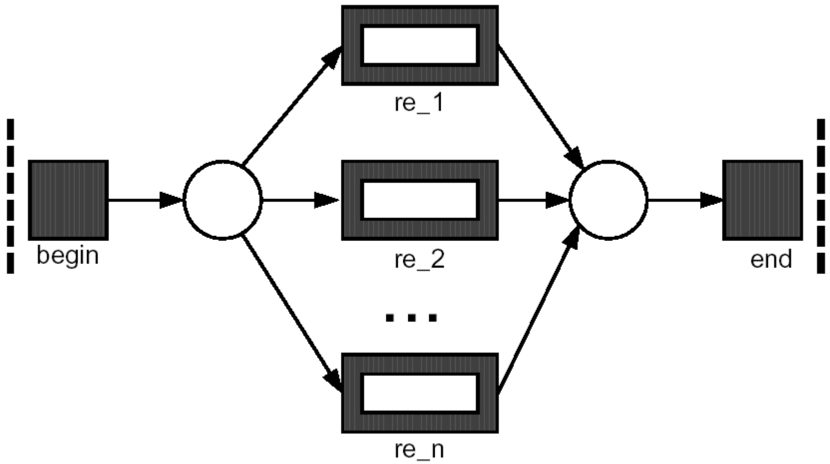
Task



Sequence

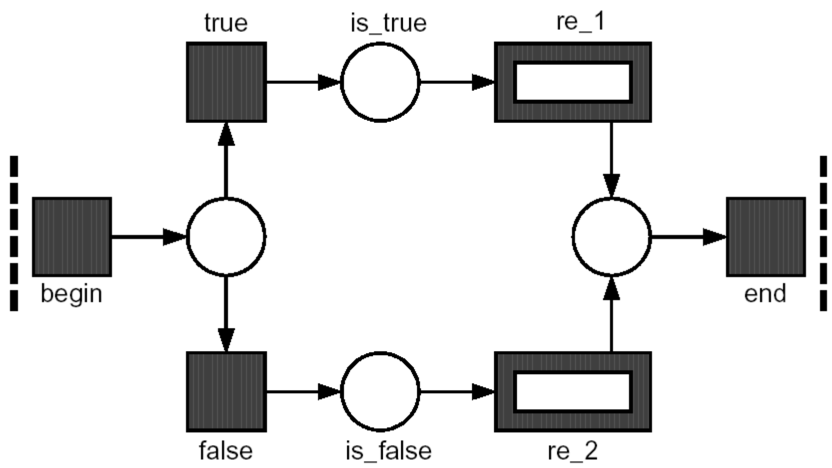


Choice



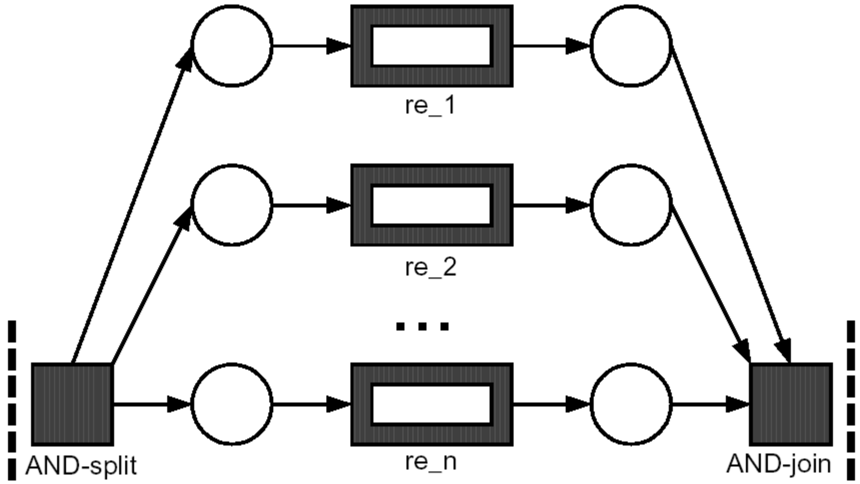
Petri Nets

Condition



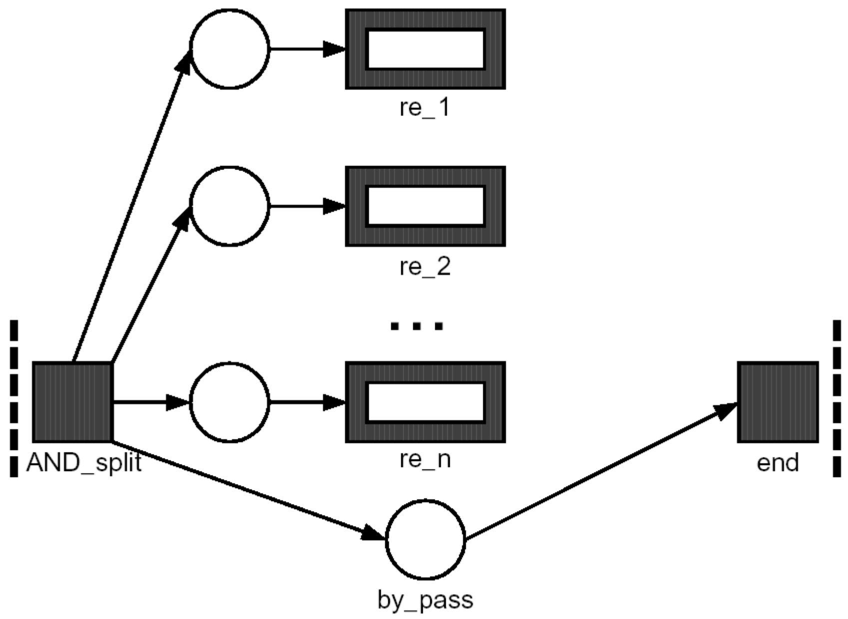
Petri Nets

Parallel execution with synchronization



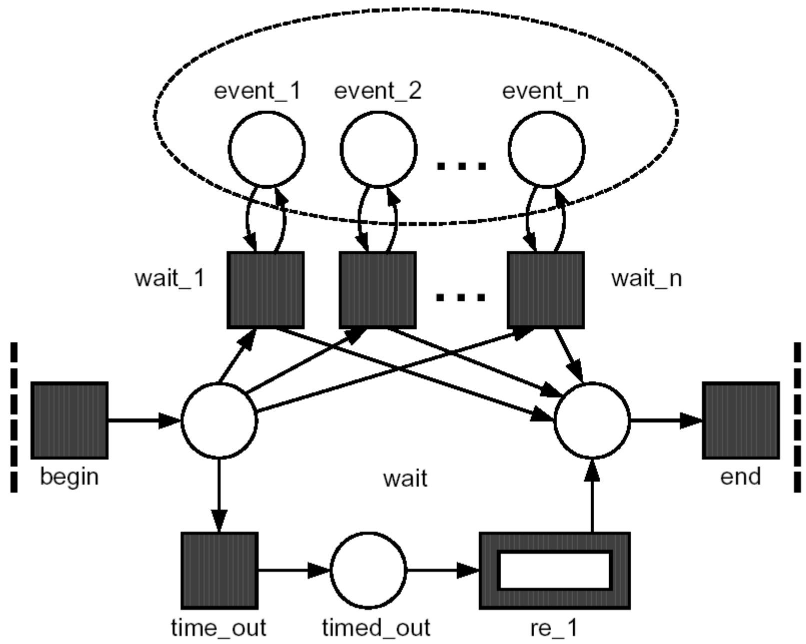
Petri Nets

Parallel execution without synchronization



Petri Nets

Wait any with time out



Process Markup Languages (**http://www.ebpml.org/status.htm**)

Tools
Blogroll

XML RSD

See also the summary from the Cover pages.

The meta model of each language vary quite a bit from one specification to another. BMPL, XLANG and WSFL are all relying on the concept of Web Services. They also clearly define a data flow (as XML documents), a control flow (block structured or transition based), a message flow (web services) and transaction flow. However, they do not spend much time on specifying how users may interact with a BPMS. The WfMC has mostly focused on that problem in the past (see "user friendly" column below). On the other hand, the WfMC specification does not support a real message flow and only a very limited data flow (process variables). The following table summarizes the differences between each data model. *(Please note that I do not have the UML 2.0 data points yet)*

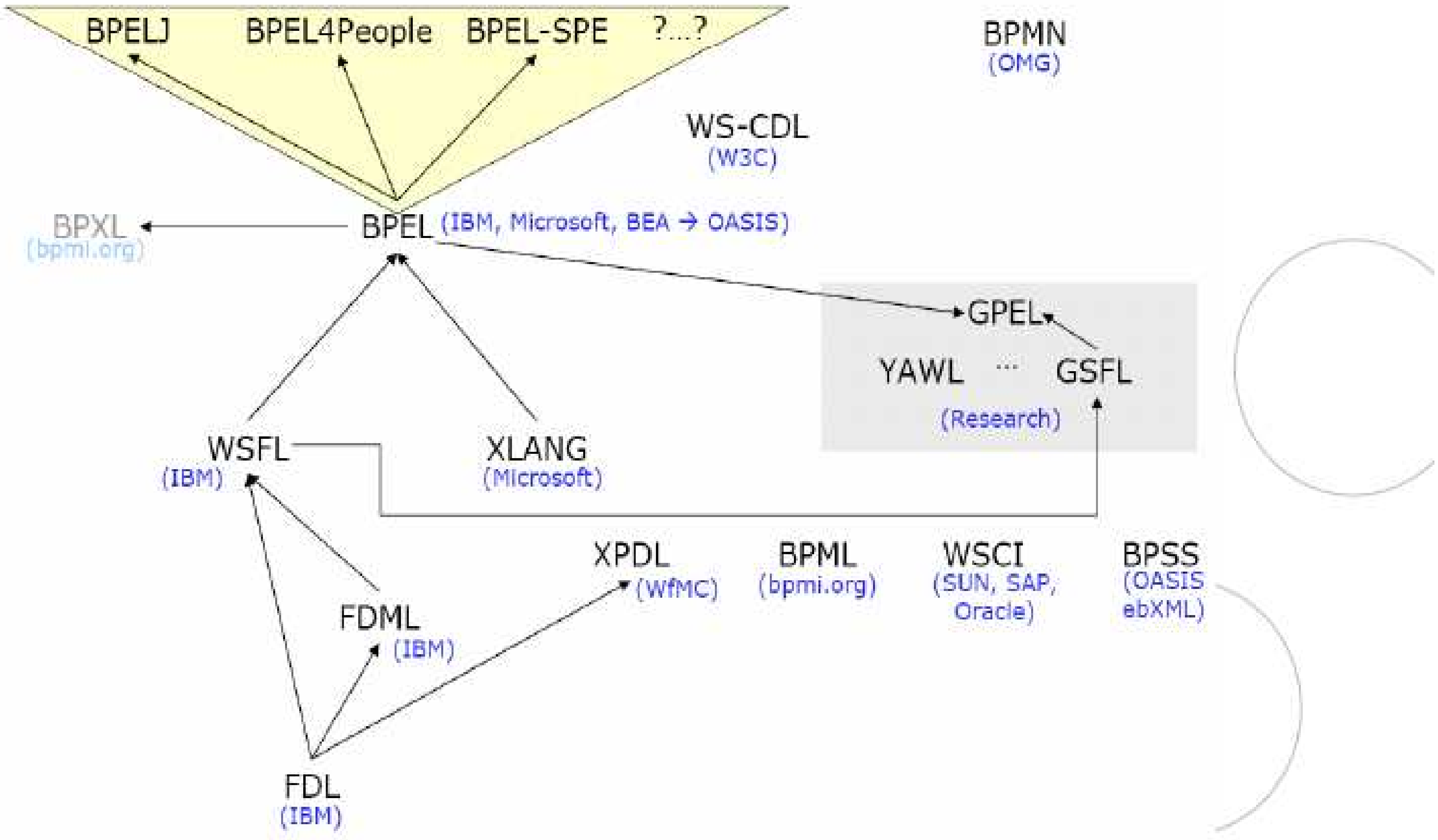
Note that Stefan Haberl provides another view of this matrix which I like better.

Specification	Control flow	Data flow	Message flow	Transaction	"EAI friendly"	"B2B friendly"	"User friendly"
BPML 1.0	Block structured	XML	Web services & Global model	yes	no	no	no
BPML 0,4	Block structured	XML	Web services	yes	yes	no	no
XLANG	Block structured	XML	Web services & Contracts	yes	no	no	no
WSFL	Transitions	XML	Web services & Global model	yes	no	no	no
BPEL4WS	Block structured	XML	Web services	yes	yes	kind of	no
EDOC	Event/Notification	Entities	Events	no	yes	yes	no
XPDL	Transitions	Process variables	Nested and chained processes	no	no	no	yes
UML 2.0	Transitions	Transitions and data buffers	collaboration model	no	?	?	?

This table makes it pretty clear: the foundations of a modern BPMS are XML and Web Services and I totally agree with this. However, this is far from enough.

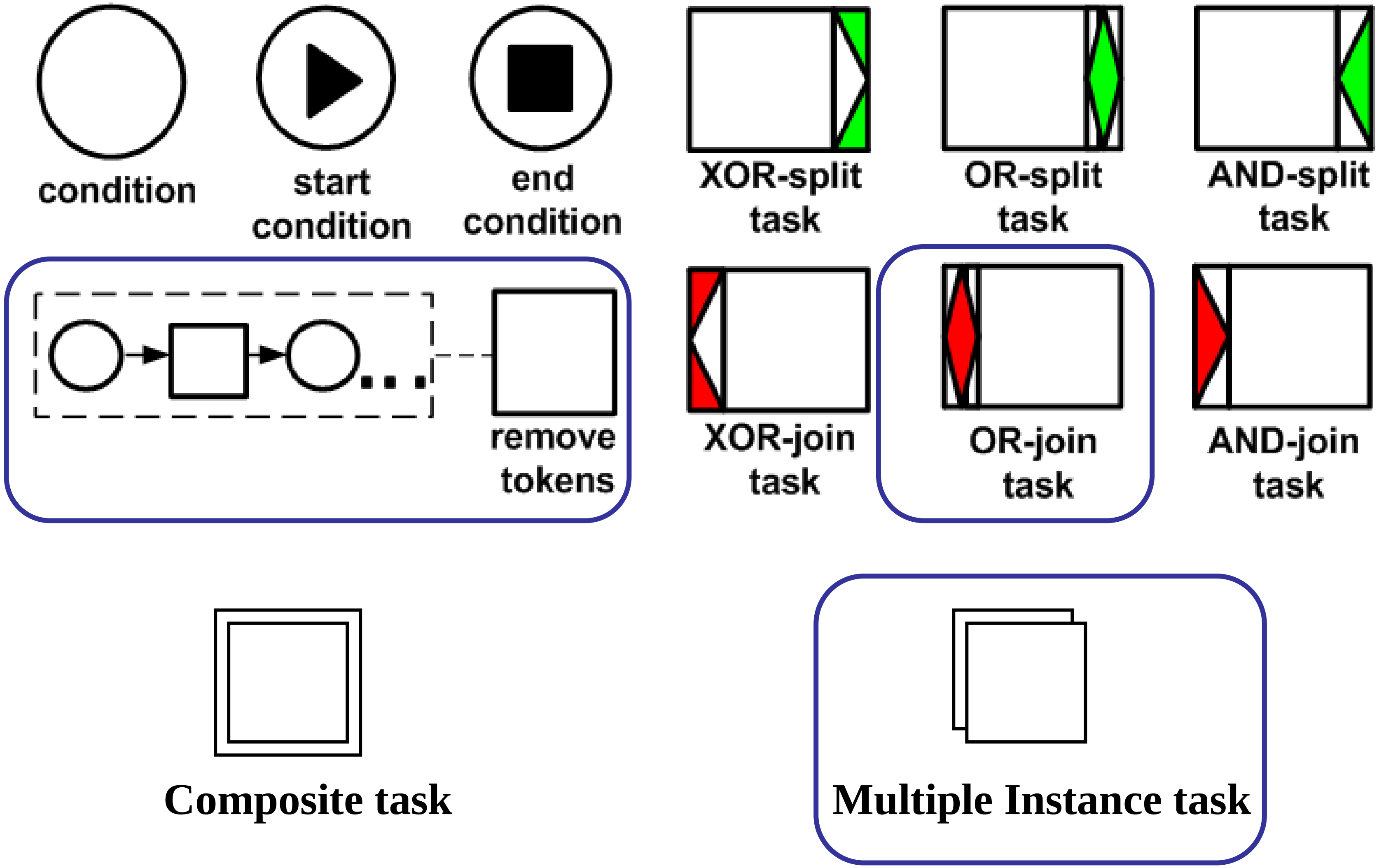
I have added "B2B EAI and User friendly" columns to measure how serious each specification is in dealing with the 3 main axes of a BPMS. As you can see, none of the new generation of PML is taking into account user interactions, though they are essential to resolve business process exceptions. None of them is considering

Process Descriptions



[Leymann, F., Nitzsche, J.: SUPER Tutorial Slides. 2007.]

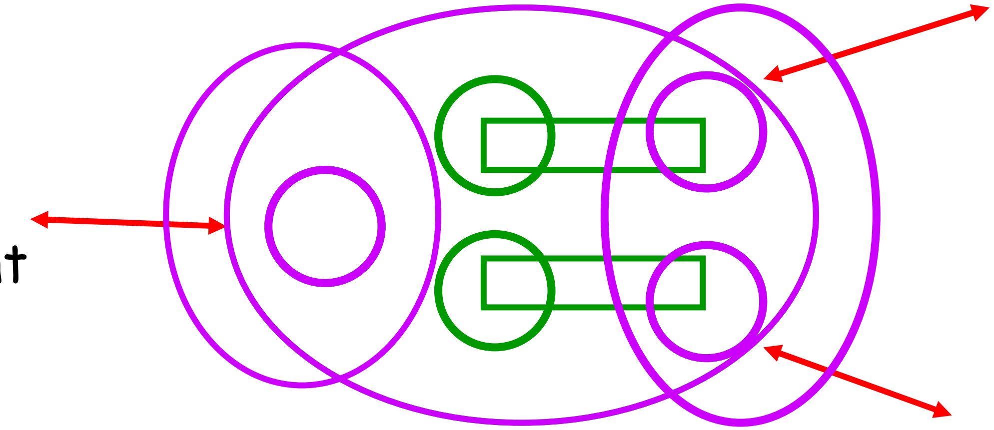
YAWL notation



Abstract vs. Executable

- abstract processes
  - public behaviour
  - define "business protocols"
  - hide things that do not affect partner
    - constrain only the message exchange
    - what the possible replies are, not why one is chosen

- executable processes
  - private behaviour
  - fully define behaviour
  - portable between compliant environments



- WS-Choreography
  - Defines abstract behaviour

- BPEL_A hides parts that exist in BPEL_B

From: Peter Furniss, Choreology

BPEL Concept: Basic Activities

Do a blocking wait for a matching message to arrive / send a message in reply

Invoke a one-way or request-response operation

Update the values of variables or partner links with new data

Validate XML data stored in variables

Generate a fault from inside the business process

Forward a fault from inside a fault handler

Immediately terminate execution of a business process instance

Invoke compensation on all completed child scopes in default order

Invoke compensation on one completed child scope

Wait for a given time period or until a certain time has passed

No-op instruction for a business process

Wrapper for language extensions

From: Michael Illiger and Simon Moser

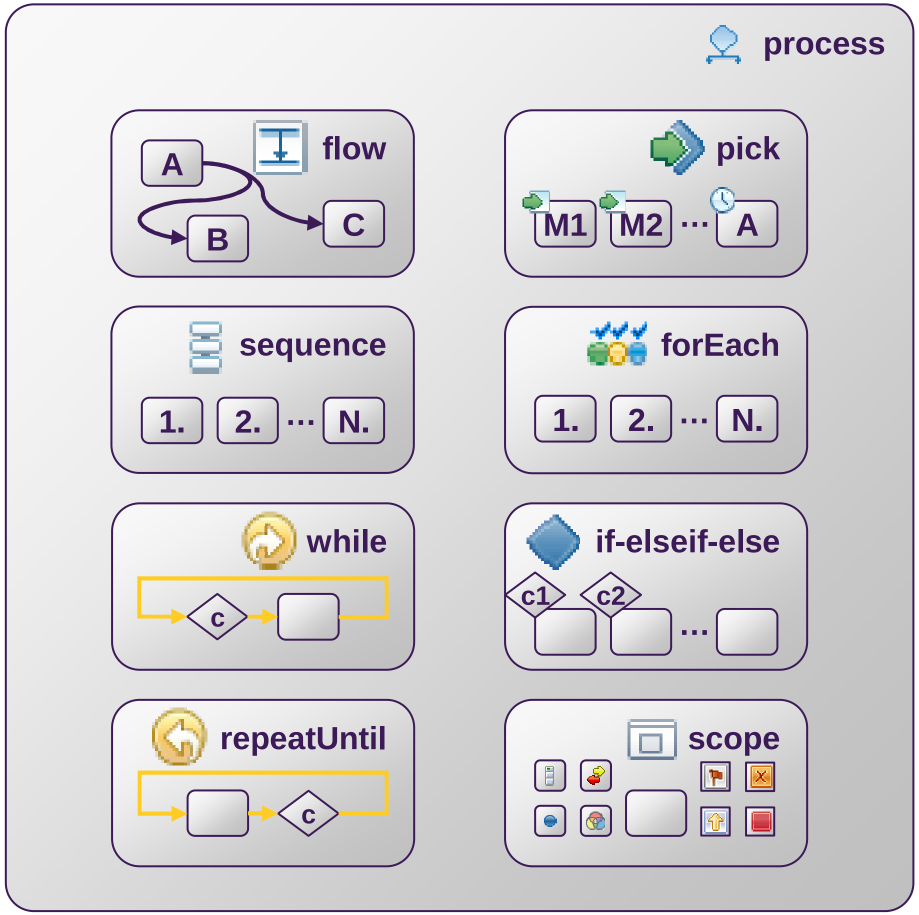
BPEL Concept: Structured Activities

Contained activities are executed in parallel, partially ordered through control links

Contained activities are performed sequentially in lexical order

Contained activity is repeated while a predicate holds

Contained activity is repeated until a predicate holds



Block and wait for a suitable message to arrive (or time out)

Contained activity is performed sequentially or in parallel, controlled by a specified counter variable

Select exactly one branch of activity from a set of choices

Associate contained activity with its own local variables, partner links, etc., and handlers

From: Michael Illiger and Simon Moser

activities

- ***structured activities – can contain other activities***

<sequence>	one after the other
<flow>	in parallel
<pick>	choose by inbound message
<switch>	choose by expression evaluation
<while>	iteration
<scope>	nest, with declarations and handlers, synchronize

- ***communication***

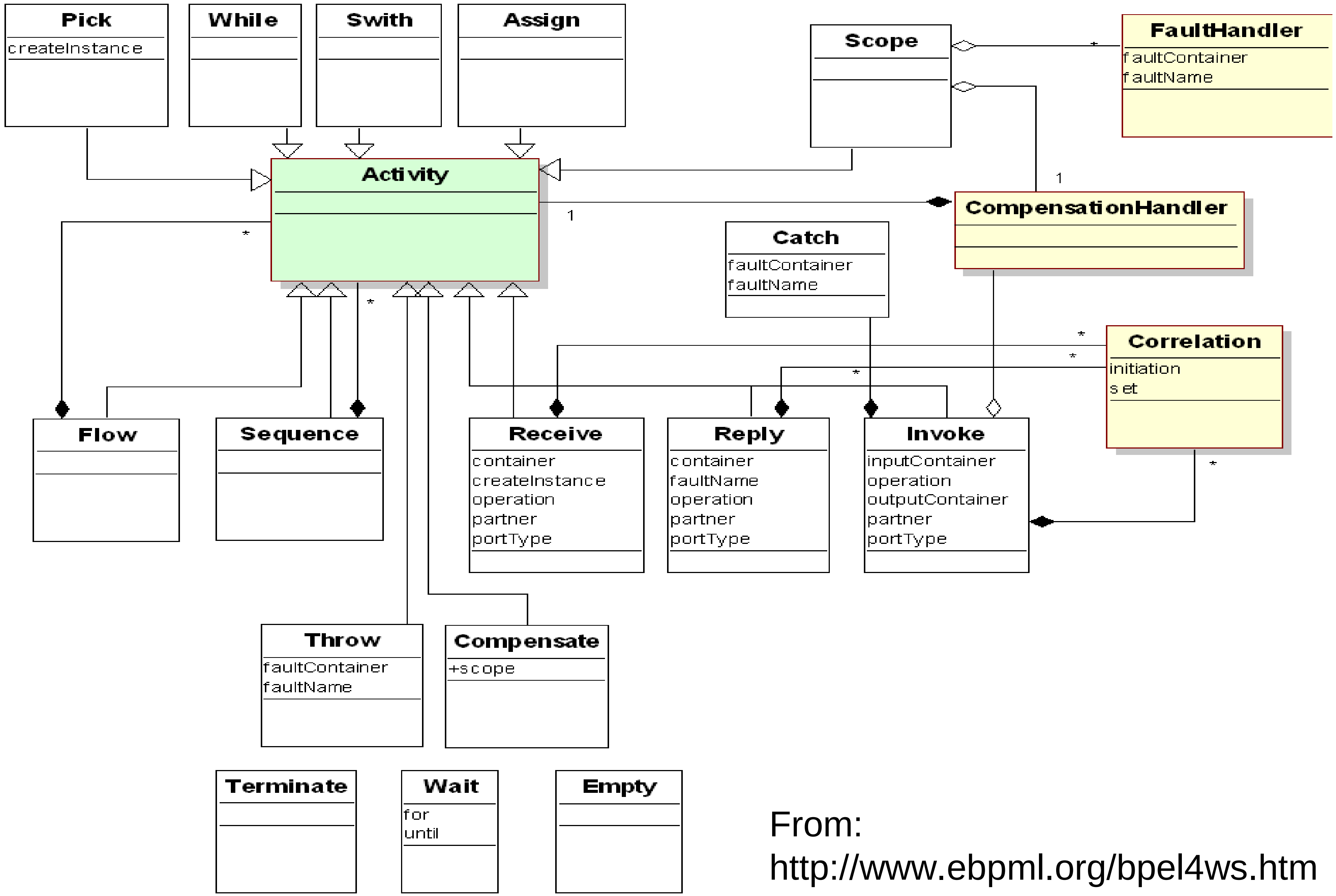
<invoke>	send msg to partner; possibly receive response
<receive>	accept msg from partner
<reply>	send msg to partner as response to <receive>

- ***other***

<assign>	manipulate variables
<wait>	for duration / until time
<terminate>	end the process
<compensate>	run compensation handler of inner scope
<throw>	exit with fault to outer scope
<empty>	do nothing

From: Peter Furniss, Choreology

BPEL Activity



From:
http://www.ebpml.org/bpel4ws.htm

"Legacy" Code Handling

Pre-existing codes, mostly in Fortran

- Generally domain-specific
- Hard to re-use in other applications
- They are still useful
- They are often large, complex monoliths with little structure.

- *Support Re-use*
- *Support Remote Execution*
- *Support Remote Discovery*
- *Support Remote Data Input/Output*

Re-write? - try convincing App Scientists

Wrapping Approaches

- Wrapping executables - "As-Is" Approach
  - No source available (or provided)
  - Maintain execution environment

Similar name in DBs, but different approach

- Wrapping Source - "Source-Update" Approach
  - Some source provided (generally I/O)
  - Executable can relinquish some control
  - Data type conversions
- Source split Wrapping - "Unit-Mapping" Approach
  - Split source into units -- wrap units
  - Maintain unit execution environment + overall manager
- Application Supported Wrapping - "App-Wrap"
  - Steering support
  - Data management support

Wrapping Approaches

Similar name in DBs, but different approach

- Wrapping executables - "As-Is" Approach
  - No source available (or provided)

•
•
•

- *Provide Isolation between existing code, in its present form, and need to re-use and execute code remotely*
- *Enable properties of code to be specified (in terms, perhaps of its interface), to enable a discovery mechanism to utilise in, say, a particular application.*
- *Sustain performance, correctness of results, ownership, and availability*

Automating Wrapping

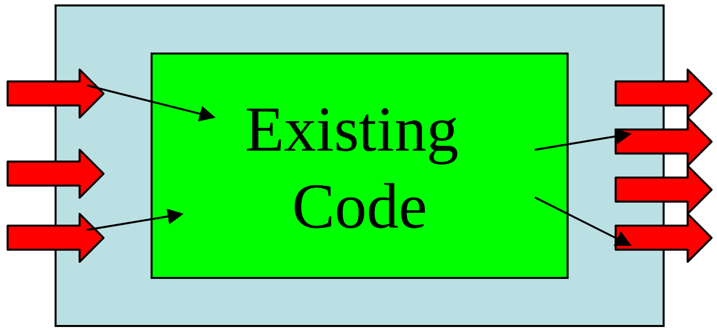
- Time consuming and error prone process
- Automate the implementation of interfaces to access code
  - via a system wide data model
- Automate interactions between wrapped components
  - via a discovery service
  - Registry/Lookup service
- Can have
  - same interface, different implementation

Component Model and Extensions

Existing
Code

Component Model and Extensions

Data Type
Translation

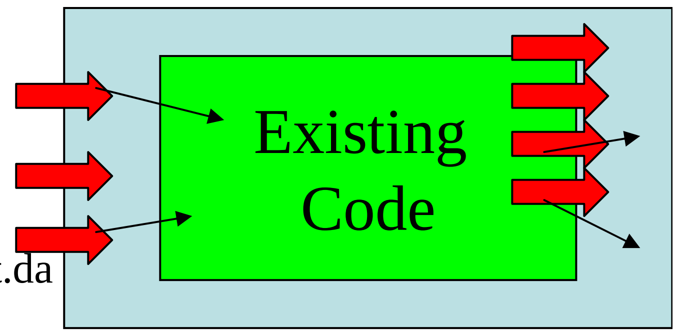


Component Model and Extensions

```
<pse-def>
  <preface>
    <name alt="MD1" id="MD01"> MDComponent</name>
    <pse-type> Molecular Dynamics </pse-type>
    <component-directory>/home/scmlm1/wgen/Component</component-directory>
    <legacy-code>/home/scmlm1/md/moldyn</legacy-code>
    <ORB-Compiler>idl2java</ORB-Compiler>
    <processors>8</processors>
    <host-name>sapphire.cs.cf.ac.uk</host-name>
  </preface>
```

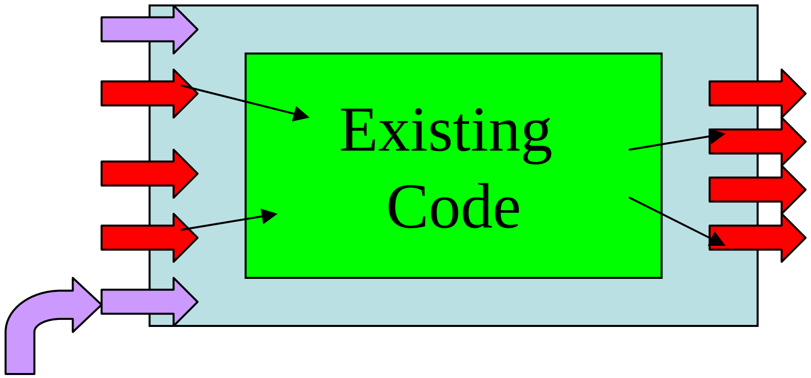
```
<outports>
   <outportnum> 6 </outportnum>
   <outport id="1"> int </outport>
   <outport id="2"> float </outport>
   <outport id="3"> float </outport>
   <outport id="4"> float </outport>
   <outport id="5"> float </outport>
   <outport id="6"> float </outport>
   <href
name="file:/home/scmlm1/wgen/Component/output.da
ta" value="output" />
   </outports>
  </ports>
```

XML Data Model



Component Model and Extensions

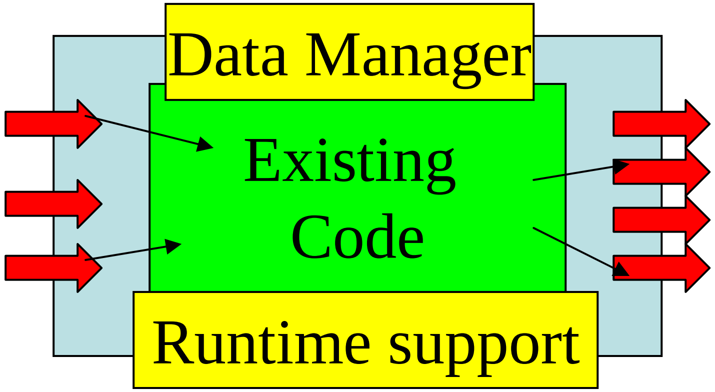
Adding additional control inputs



External Control Input (for Steering)

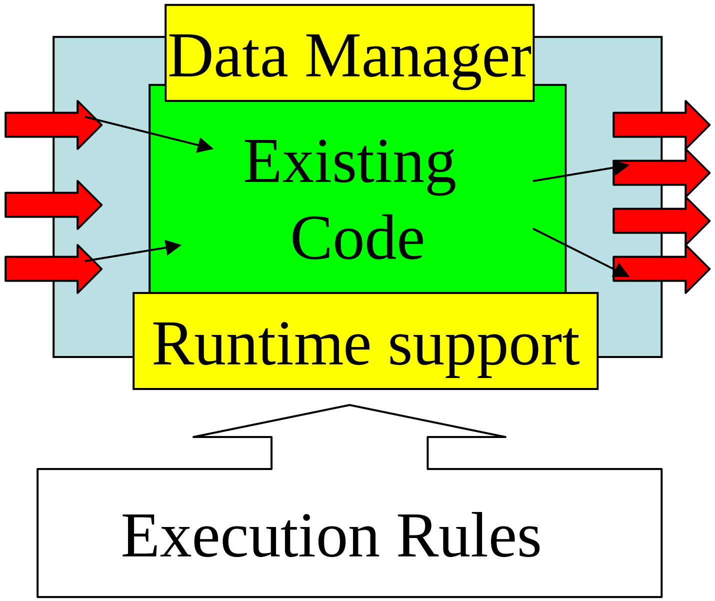
Component Model and Extensions

Adding "container" services



Component Model and Extensions

Adding an execution policy



Control Flow vs. Data Flow

- Control Flow
  - Managed via use of specialist control constructs (conditions – may be simple conjunction/disjunction, or more complex operators)
  - Unit/component execution managed through these control constructs
  - Types
    - Transition only
    - Switch, flow, while, etc
- Data Flow
  - Execution managed via transmission of data

Data/Control-Flow Spectrum

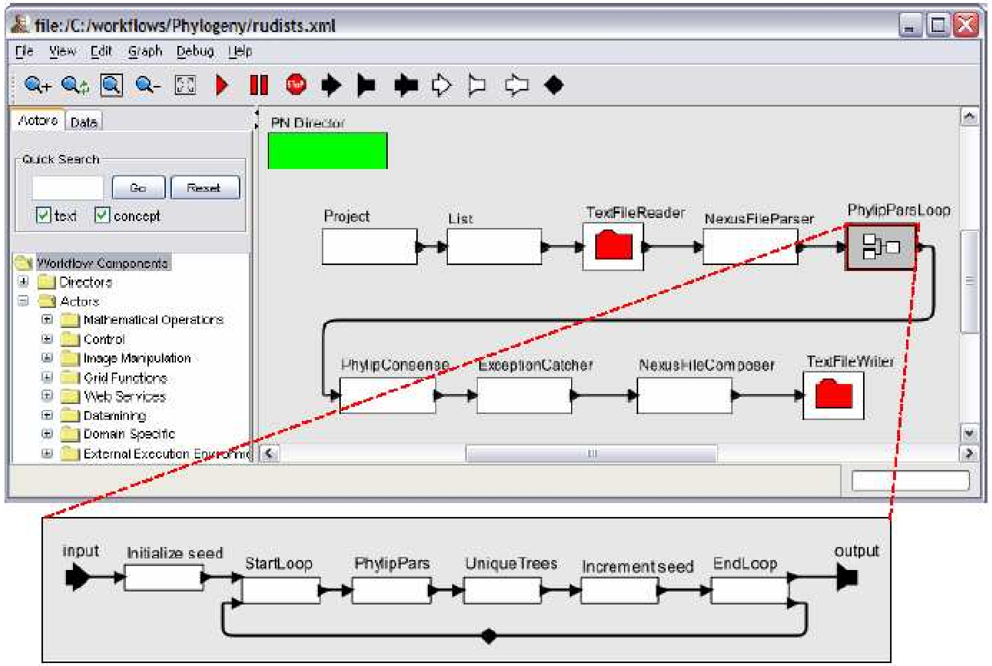
"clean" data(=ctl)-flow special tokens flow message passing, control flow

- Data (tokens) flow
  - (almost) no other side effects
  - WYSIWYG (usually)
- References flow
  - token reference type may be "http-get", "ftp-get"
  - generic handling still possible
- Application specific tokens flow
  - e.g. current Nimrod job management in Resurgence
  - "invisible contract" between components
  - Director (Kepler) is unaware of what's going on
- Specific messages passing protocols (e.g., CSP, MPI)
  - for systems of tightly coupled components

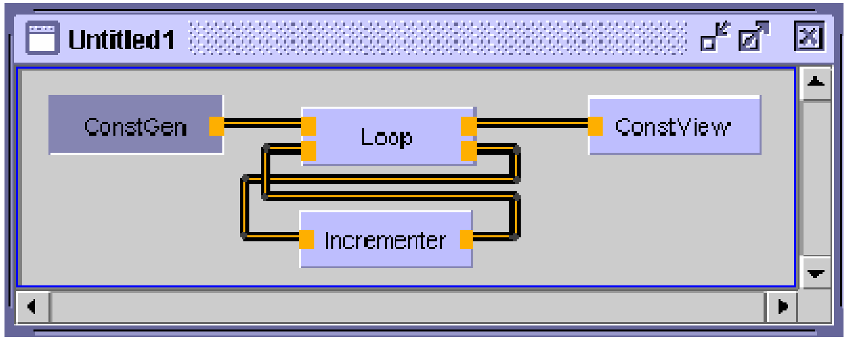
Dealing with Loops and Conditionals

- Often difficult to achieve – often ignored
- Conditional
  – Specified as control-blocks
  – Implemented through the use of scripts
- Loops
  – Specified as "meta-blocks" – blocks implemented over sub-workflows
  – Implemented through the use of scripts
- Must be supported in the Enactment Engine
- YAWL → defines the concept of "worklets" – sub-workflows over which control constructs can be applied

Loops … 2



KEPLER

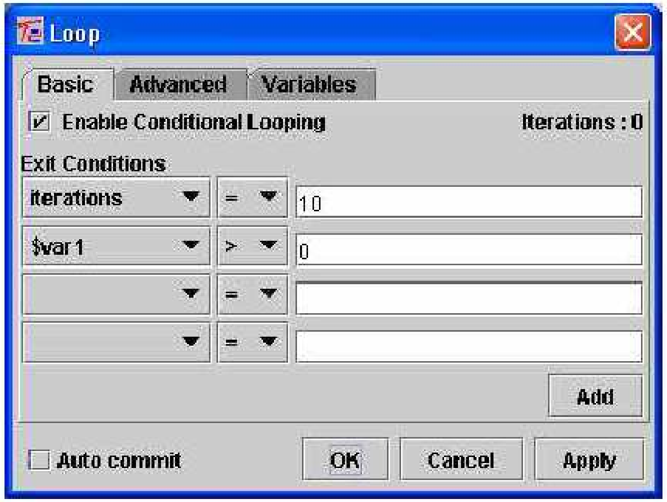


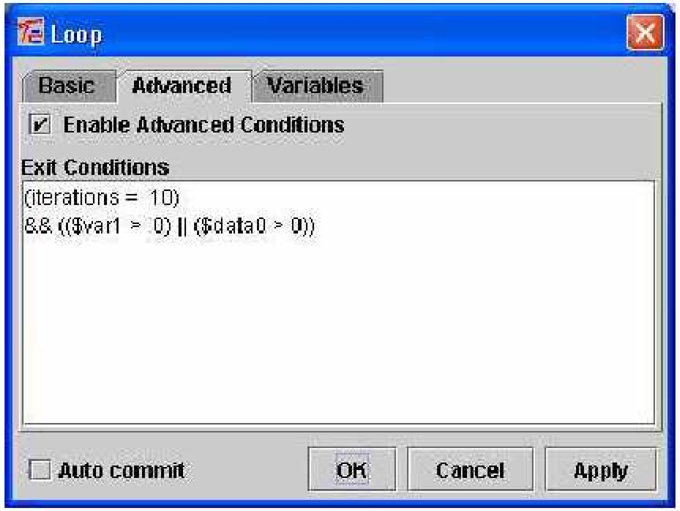
Triana

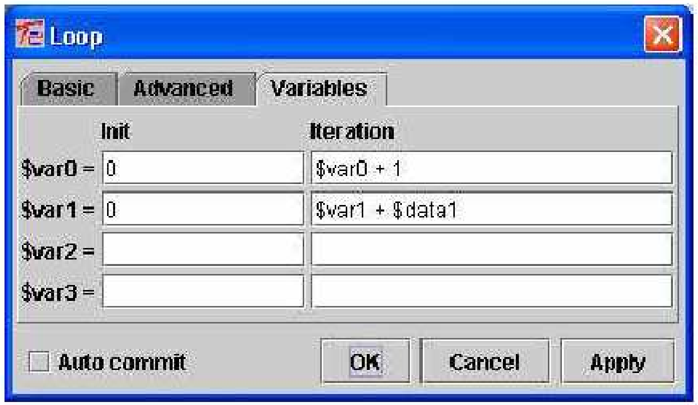
In Triana and Kepler – use of specialist "Loop" components

- Components can be explicit
- Implemented as "hidden" command

Loops … 3








```
iterations >= total iterations / 2
ConstGen1.constant < ConstGen2.constant + 2 / 3
$data0 = iterations % (10 + $var2)
$var1 = $data0 + $data1
```

Init()
Iteration()
isExitLoop(Object[] data)
(Allows for user defined objects to
specify loop exit condition)

Control-flow Patterns

- **Basic Control-flow Patterns**
  capture elementary aspects of control-flow (similar to the concepts provided by the WFMC).

- **Advanced Branching and Synchronization Patterns**
  describe more complex branching and synchronization scenarios.

- **Iteration Patterns**
  describe various ways in which iteration may be specified.

- **Termination Patterns**
  address the issue of when the execution of a workflow is considered to be finished.

- **Multiple Instances (MI) Patterns**
  delineate situations with multiple threads of execution in a workflow which relate to the same activity.

- **State-based Patterns**
  reflect situations which are most easily modelled in WF languages with an explicit notion of state.

- **Cancellation Patterns**
  categorise the various cancellation scenarios that may be relevant for a workflow specification.

- **Trigger Patterns**
  catalogue the different triggering mechanisms appearing in a process context.

Data Pattern Categories

- **Data Visibility:** The extent and manner in which data elements can be viewed and utilised by workflow components.
- **Internal Data Interaction:** Data communication between active elements within a workflow.
- **External Data Interaction:** Data communication between active elements within a workflow and the external operating environment.
- **Data Transfer:** Data element transfer across the interface of a workflow component.
- **Data Routing:** The manner in which data elements can influence the operation of the workflow.

Workflow Resource Patterns

- Focus on the manner in which work is offered to, allocated to and managed by workflow participants
- Consider both the system and resource perspectives
- Assume the existence of a process model and related organisational model
- Take into account differing workflow paradigms:
  - richness of process model (esp. allocation directives)
  - autonomy of resources
  - alternate routing mechanisms
  - work management facilities

The Workflow Patterns Framework

time

2000 2003	Jun 2005	Oct 2005	
Control-flow P:s 20	Resource P:s - 43	Data P:s - 40	Ex
COSA FLOWer Eastman Meteor Mobile I-Flow Staffware InConcert Domino Workflow Visual Workflow Forte Conductor MQSeries/Workflow SAR R/3 Workflow Verve Workflow Changengine	Staffware WebSphere MQ FLOWer COSA iPlanet jBPM OpenWFE Enhydra Shark	Staffware MQSeries FLOWer COSA jBPM OpenWFE Enhydra Shark	Sta We FLC CO iPla
XPDL, BPEL4WS, BPML, WSFL, XLANG, WSCI, UML AD 1.4 UML AD 2.0, BPMN	BPEL4WS UML AD 2.0 BPMN	XPDL, BPEL4WS UML AD 2.0, BPMN	XPD BPE

Evaluations

Language Development: YAWL/newYAWL

Control-Flow Perspective: Evaluation

1 – BPMN
2 – UML AD
3 – BPEL

		1	2	3			1	2	3
Basic Control-flow					**Termination**				
1	Sequence	+	+	+	11	Implicit Termination	+	+	+
2	Parallel Split	+	+	+	43	Explicit Termination	+	+	-
3	Synchronisation	+	+	+	**Multiple Instances**				
4	Exclusive Choice	+	+	+	12	MI without Synchronisation	+	+	+
5	Simple Merge	+	+	+	13	MI with a priory Design Time Knlg	+	+	+
Advanced Synchronisation					14	MI with a priory Runtime Knlg	+	+	-
6	Multiple Choice	+	+	+	15	MI without a priory Runtime Knlg	-	-	-
7	Str. Synchronising Merge	+/-	-	+	27	Complete MI Activity	-	-	-
8	Multiple Merge	+	+	-	34	Static Partial Join for MI	+/-	-	-
9	Discriminator	+/-	+	-	35	Cancelling Partial Join for MI	+/-	-	-
28	Blocking Discriminator	+/-	+/-	-	36	Dynamic Partial Join for MI	-	-	-
29	Cancelling Discriminator	+	+	-	**State-based**				
30	Structured Partial Join	+/-	+/-	-	16	Deferred Choice	+	+	+
31	Blocking Partial Join	+/-	+/-	-	39	Critical Section	-	-	+
32	Cancelling Partial Join	+/-	+	-	17	Interleaved Parallel Routing	+/-	-	+/-
33	Generalised AND-Join	+	-	-	40	Interleaved Routing	+/-	-	+
37	Acyclic Synchronizing Merge	-	+/-	+	18	Milestone	-	-	-
38	General Synchronizing Merge	-	-	-	**Cancellation**				
41	Thread Merge	+	+	+/-	19	Cancel Activity	+	+	+
42	Thread Split	+	+	+/-	20	Cancel Case	+	+	+
Iteration					25	Cancel Region	+/-	+	-
10	Arbitrary Cycles	+	+	-	26	Cancel MI Activity	+	+	-
21	Structured Loop	+	+	+	**Trigger**				
22	Recursion	-	-	-	23	Transient Trigger	-	+	-
					24	Persistent Trigger	+	+	+

Data Perspective: Evaluation

1 – BPMN
2 – UML AD
3 – BPEL

		1	2	3
Data Visibility				
1	Task Data	+	+/-	+/-
2	Block Data	+	+	-
3	Scope Data	-	-	+
4	MI Data	+/-	+	-
5	Case Data	+	-	+
6	Folder Data	-	-	-
7	Workflow Data	-	+	-
8	Environment Data	-	-	+
Data Interaction (Internal)				
9	between Tasks	+	+	+
10	Task to Sub-workflow Decomp.	+	+	-
11	Sub-workflow Decomp. to Task	+	+	-
12	to MI Task	-	+	-
13	from MI Task	-	+	-
14	Case to Case	-	-	+/-
Data Interaction (External)				
15	Task to Env - Push-Oriented	+	-	+
16	Env. to Task - Pull-Oriented	+	-	+
17	Env. to Task - Push-Oriented	+	-	+/-
18	Task to Env - Pull-Oriented	+	-	+/-
19	Case to Env. - Push-Oriented	-	-	-
20	Env. to Case - Pull-Oriented	-	-	-

		1	2	3
Data Interaction (External), cont.				
21	Env. to Case - Push-Oriented	-	-	-
22	Case to Env. - Pull-Oriented	-	-	-
23	Workflow to Env. - Push-Oriented	-	-	-
24	Env. to Workflow - Pull-Oriented	-	-	-
25	Env. to Workflow - Push-Oriented	-	-	-
26	Workflow to Env. - Pull-Oriented	-	-	-
Data Transfer				
27	by Value - Incoming	+	-	+
28	by Value - Outgoing	+	-	+
29	Copy In/Copy Out	+/-	-	-
30	by Reference - Unlocked	-	-	+
31	by Reference - Locked	+	+	+/-
32	Data Transformation - Input	+/-	+	-
33	Data Transformation - Output	+/-	+	-
Data-based Routing				
34	Task Precondition Data Exist.	+	+	+/-
35	Task Precondition Data Value	-	+	+
36	Task Postcondition Data Exist.	+	+	-
37	Task Postcondition Data Value	-	+	-
38	Event-based Task Trigger	+	+	+
39	Data-based Task Trigger	+	-	+/-
40	Data-based Routing	+	+	+

Resource Patterns Classes リソースパターン分類

- **Creation patterns:** design-time work allocation directives
  生成パターン：設計時でのリソース割り当て
- **Push patterns:** workflow system proactively distributes work items
  プッシュパターン：ワークフローシステムが積極的に作業を提供
- **Pull patterns:** resources proactively identify and commit to work items
  プルパターン：リソース（人など）が積極的に作業をコミットする
- **Detour patterns:** re-routing of work items
  回り道パターン：作業の流れを変える
- **Auto-start patterns:** automated commencement
  自動スタートパターン：自動開始のパターン
- **Visibility patterns:** observability of workflow activities
  可視化パターン：作業の監視性
- **Multiple resource patterns:** work allocation involving multiple participants or resources
  複数リソースパターン：複数リソースにまたがる作業の割り当て

Click here for a FLASH animation of Delegation Pattern

Resource perspective: Evaluation

1 – BPMN,
2 – UML AD,
3 – Oracle BPEL PM
(from [Mulyar 2005])

		1	2	3			1	2	3
Creation Patterns					**Pull Patterns (cont.)**				
1	Direct Allocation	+	+	+	24	System-Determ. Work Queue Cont.	-	-	-
2	Role-Based Allocation	+	+	+	25	Resource-Determ. Work Queue Cont.	-	-	+
3	Deferredc Allocation	-	-	+	26	Selection Autonomy	-	-	+
4	Authorization	-	-	-	**Detour Patterns**				
5	Separation of Duties	-	-	-	27	Delegation	-	-	+
6	Case Handling	-	-	+	28	Escalation	-	-	+
7	Retain Familiar	-	-	+	29	Deallocation	-	-	+
8	Capacity-based Allocation	-	-	+	30	Stateful Reallocation	-	-	+
9	History-based Allocation	-	-	+/-	31	Stateless Reallocation	-	-	-
10	Organizational Allocation	-	-	+/-	32	Suspension/Resumption	-	-	+
11	Automatic Execution	+	+	+	33	Skip	-	-	+
Push Patterns					34	Redo	-	-	-
12	Distritubtion by Offer-Single Resource	-	-	+	35	Pre-do	-	-	-
13	Distritubtion by Offer-Multiple Resources	-	-	+	**Auto-start Patterns**				
14	Distritubtion by Allocation-Single Resource	+	+	+	36	Commencement on Creation	+	+	-
15	Random Allocation	-	-	+/-	37	Commencement on Allocation	-	-	-
16	Round Robin Allocation	-	-	+/-	38	Piled Execution	-	-	-
17	Shortest Queue	-	-	+/-	39	Chained Execution	+	+	-
18	Early Distribution	-	-	-	**Visibility Patterns**				
19	Distribution on Enablement	+	+	+	40	Config. Unallocated WI Visibility	-	-	-
20	Lata Distribution	-	-	-	41	Config. Allocated WI Visibility	-	-	-
Pull Patterns					**Multiple Resource Patterns**				
21	Resource-Init. Allocation	-	-	-	42	Simultaneous Execution	+	+	+
22	Resource-Init. Exec. - Allocated WI	-	-	+	43	Additional Resources	-	-	+
23	Resource-Init. Exec. - Offered WI	-	-	+					

Enactment Engines

- Employ a variety of techniques for enactment
- Integrated with a Portal – others based on a command line interface (some also provide a scripting language)
- Generally for constructing graphs – others also support execution of components within a graph
- Support for third-party services
  - Monitoring, Registry, etc
- Can take workflow as input, process this, and return another workflow (equivalent to treating workflow graphs as data)

Enactment Strategies … I

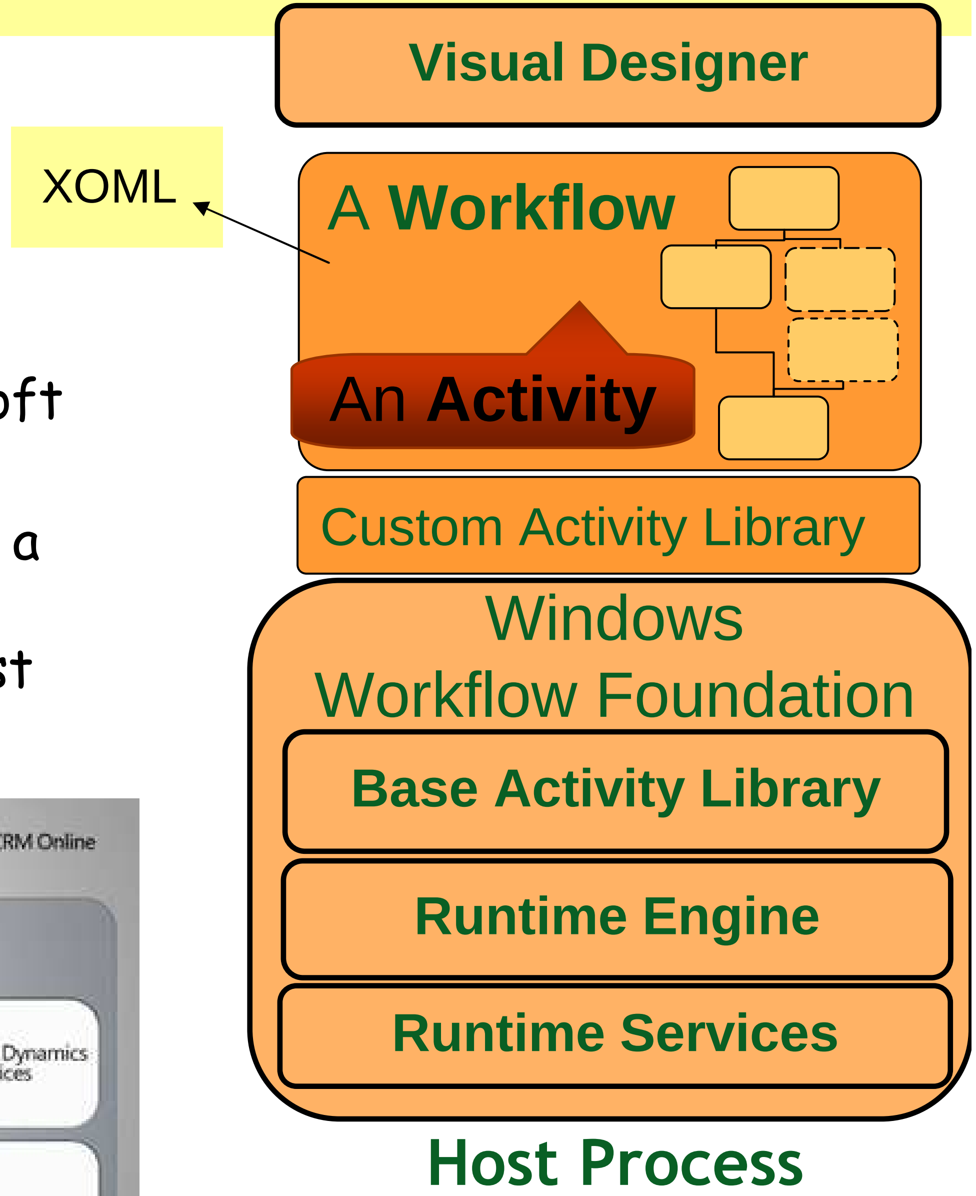
- Centralised Enactor
  - Single graph coordinated through a centralised enactor
  - The enactor manages execution of components in some sequence
- Distributed Enactors
  - Graph divided into sub-graphs and handed to different enactors
  - Each enactor responsible for executing local graph
  - Divide graph across enactors
    - Undertaken by a user
    - Undertaken automatically using rules (more later)

Enactment Strategies … II

- Event-based
  - Each component on completion generates an event
  - Use of publish-subscribe mechanism
  - Each component also activated through the generation of an event
  - Can have multiple event types
- Blackboard/Shared memory
  - Component/Enactor writes to a shared space
  - Monitored by components/enactor
  - Blocks on availability of particular data items in shared space

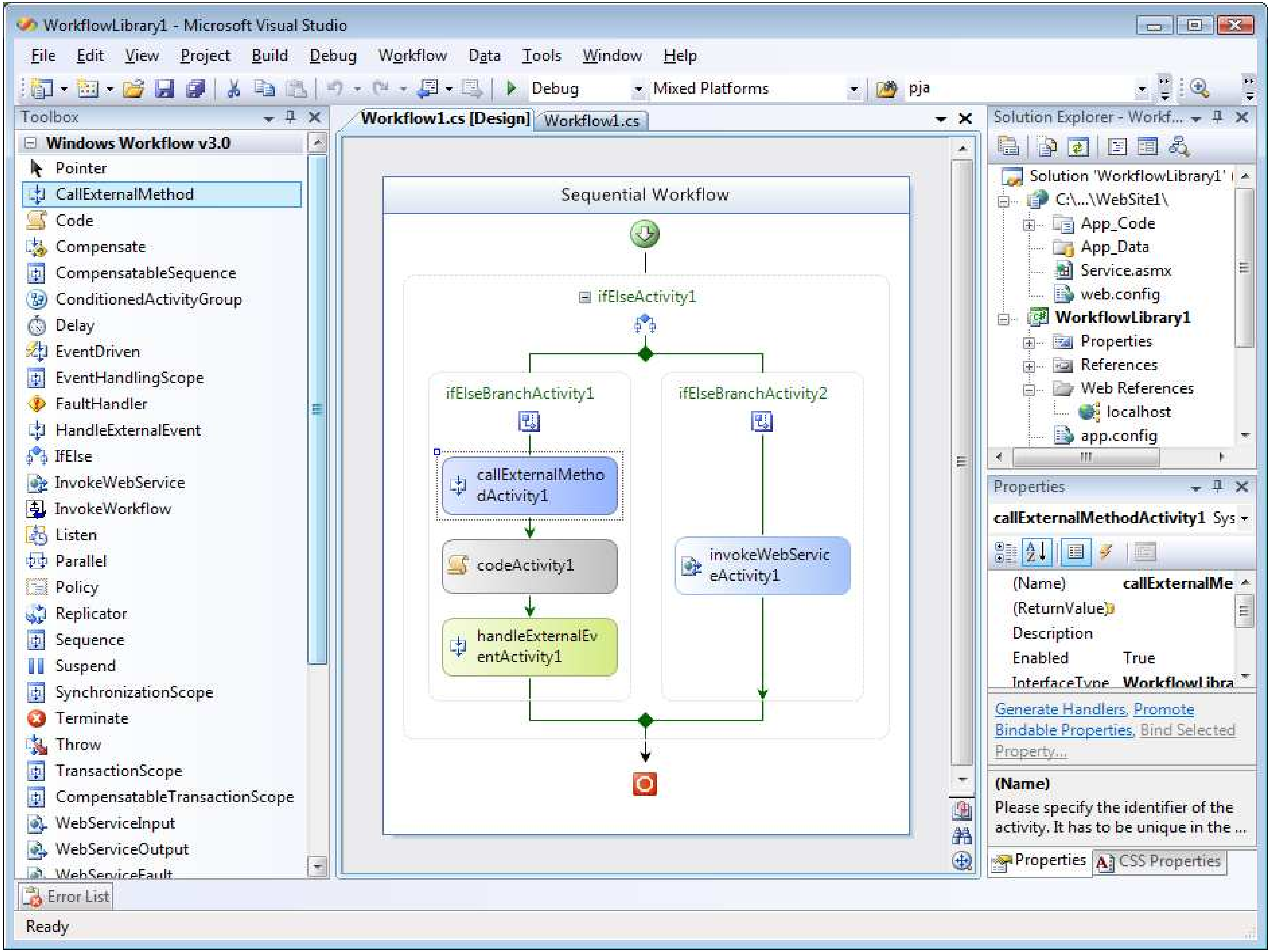
.NET Services – Windows Workflow Foundation

- Hosting of Workflow:
  - Own Host
  - “Dublin”
  - .NET workflow Services
- Hosting supported in a Microsoft Cloud (via Microsoft Azure)
- Supports multiple instances of a workflow instance (for fault tolerance) – through a multicast Service Bus



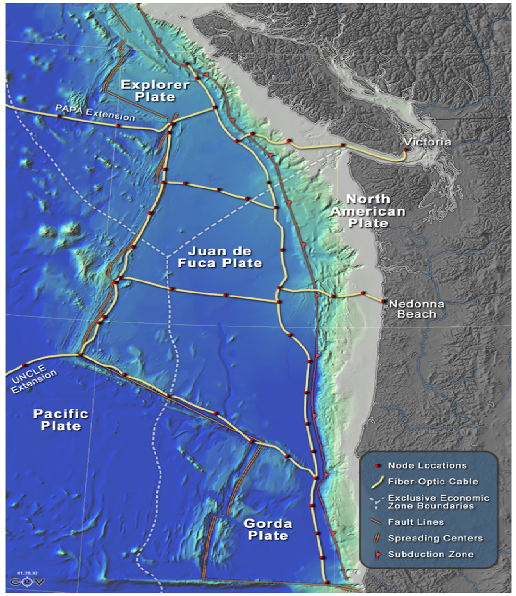
Activities in Microsoft Workflows

- Out-of-Box WF Activities
  - IfElse, <u>Sequence</u>, Suspend, Terminate, <u>While</u>
- .NET Workflow Service Activities (for Azure)
  - Delay, HTTPSend, HTTPReceive, ServiceBusSend (for Events), XPathRead, XPathUpdate (content-based routing)
- Execution supported through a .NET execution engine
  - Workflow status (terminated, suspended, running, etc)
  - Can use portal to change workflow definition


WorkflowLibrary1 - Microsoft Visual Studio
File Edit View Project Build Debug Workflow Data Tools Window Help
Debug
Mixed Platforms
pja
Toolbox
Windows Workflow v3.0
Pointer
CallExternalMethod
Code
Compensate
CompensatableSequence
ConditionedActivityGroup
Delay
EventDriven
EventHandlingScope
FaultHandler
HandleExternalEvent
IfElse
InvokeWebService
InvokeWorkflow
Listen
Parallel
Policy
Replicator
Sequence
Suspend
SynchronizationScope
Terminate
Throw
TransactionScope
CompensatableTransactionScope
WebServiceInput
WebServiceOutput
Error List
Ready
Workflow1.cs [Design]
Workflow1.cs
Sequential Workflow
ifElseActivity1
ifElseBranchActivity1
ifElseBranchActivity2
callExternalMethodActivity1
codeActivity1
handleExternalEventActivity1
invokeWebServiceActivity1
Solution Explorer - Workf...
Solution 'WorkflowLibrary1'
C:\...\WebSite1\
App_Code
App_Data
Service.asmx
web.config
WorkflowLibrary1
Properties
References
Web References
localhost
app.config
Properties
callExternalMethodActivity1
(Name)
callExternalMe
(ReturnValue)
Description
Enabled
True
Generate Handlers, Promote Bindable Properties, Bind Selected Property...
(Name)
Please specify the identifier of the activity. It has to be unique in the ...
Properties
CSS Properties

Trident & NEPTUNE (Roger Barga, Microsoft)

- Trident: Scientific workflow tools using Microsoft Workflow Foundation
- Distributed registry service for sensor + simulation/model data
- Trident enables:
  - Automate tedious data cleaning and analysis pipelines.
  - Explore and visualize data, regardless of source.
  - Compose, run and catalog experiments, save results.
  - Explore and visualize ocean & model data.
- Also, utilize collaboration facility in MSW
  - through the use of .NET Services portal



Trident & NEPTUNE (Roger Barga, Microsoft)

Populate Windows WF with custom activities

- Introduce gridded data structures;
- Define basic operators (data transformations);
- Implemented as custom activities;

Introduce parameterized activities

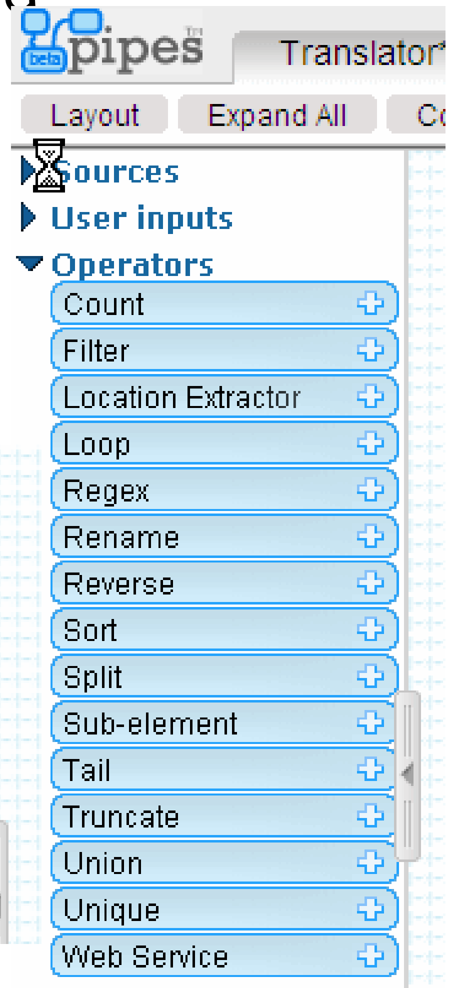
- Easier for users to design workflows
- Tool to convert custom to parameterized activities

Invoke and author workflows via web browser

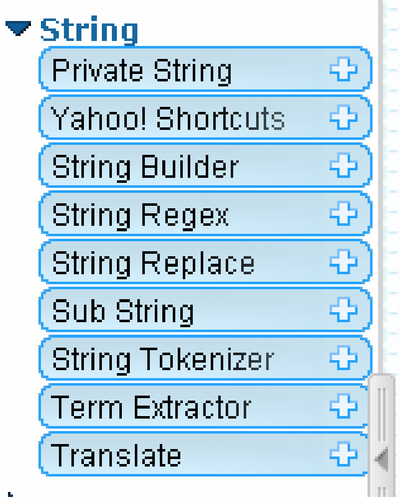
Persistent workflows, checkpoints (*stop-revise-rerun*)

Yahoo! Pipes

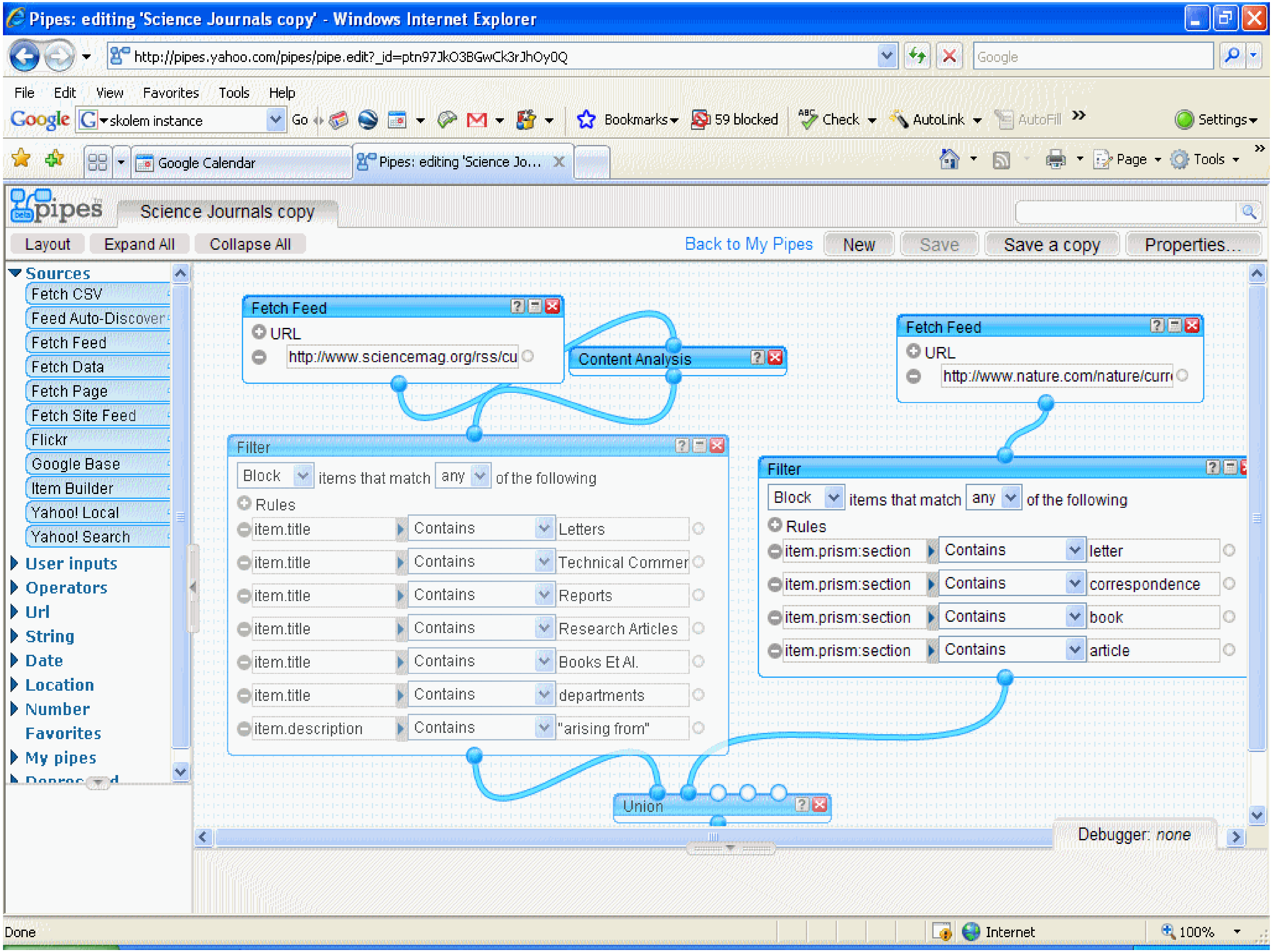
- Exports data to RSS, XML and JSON (data aggregation)
- Mainly provides support for aggregating and manipulating RSS feeds
  - Feeds can come from Google Base, Flickr, Yahoo! Local, CSV files etc.
- Provides a variety of functions for this

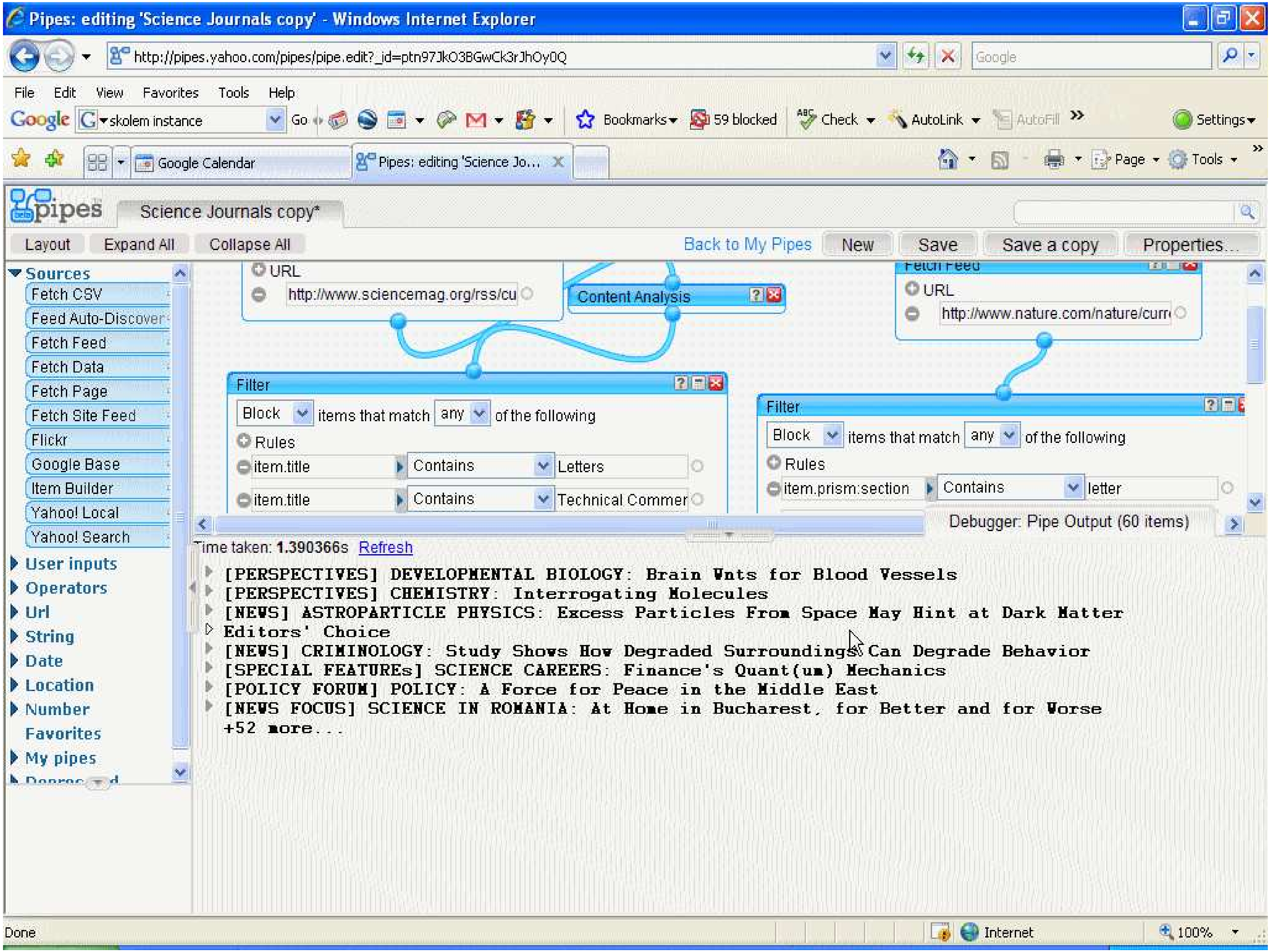


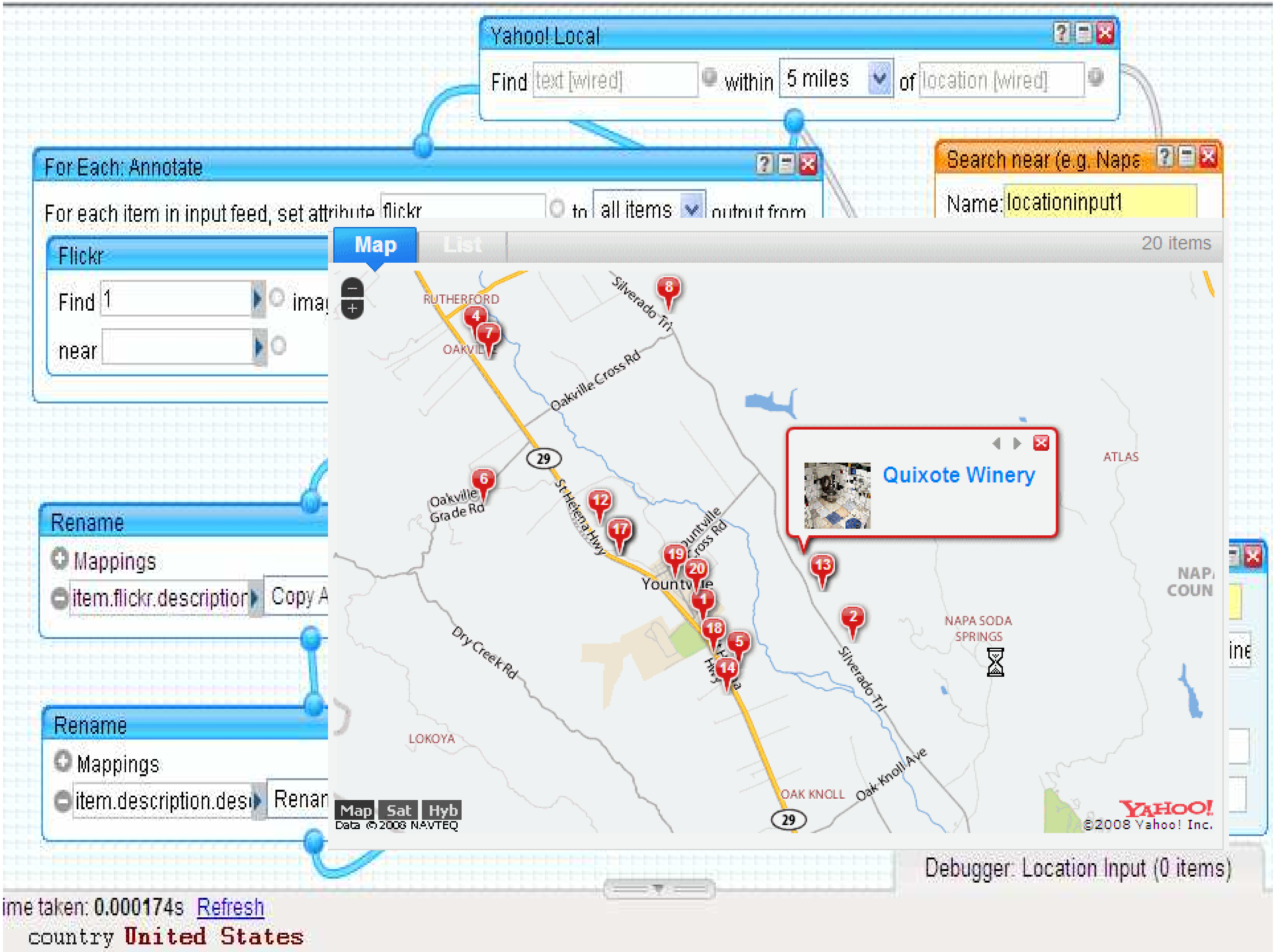
- Allows
  - Translation between feeds
  - Aggregation of feeds
  - Integration with map

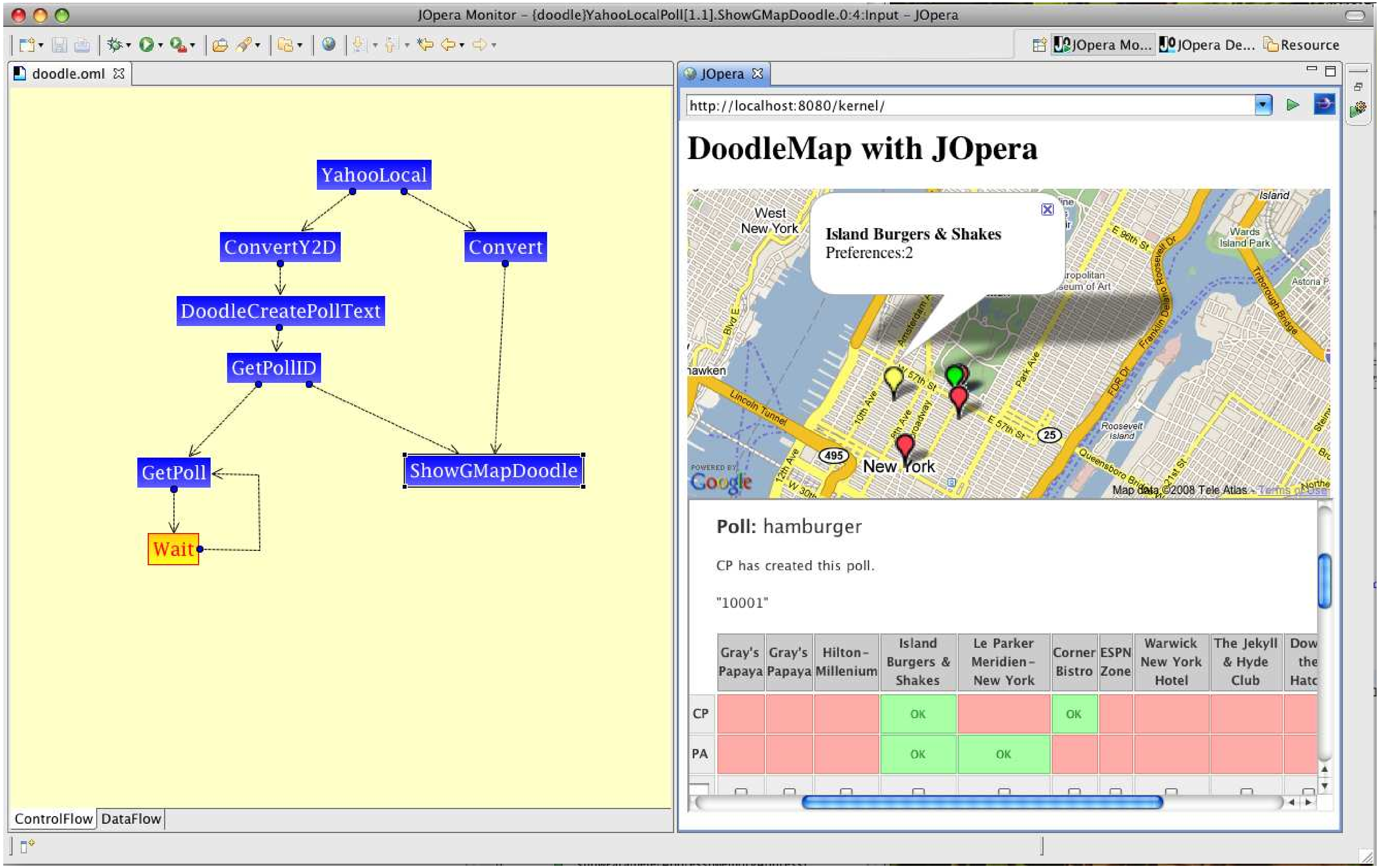


- Focus primarily on a data driven approach


Pipes: editing 'Science Journals copy' - Windows Internet Explorer
http://pipes.yahoo.com/pipes/pipe.edit?_id=ptn97JkO3BGwCk3rJhOy0Q
File Edit View Favorites Tools Help
skolem instance
Go
Bookmarks
59 blocked
Check
AutoLink
AutoFill
Settings
Google Calendar
Pipes: editing 'Science Jo...
Page
Tools
pipes
Science Journals copy
Layout
Expand All
Collapse All
Back to My Pipes
New
Save
Save a copy
Properties...
Sources
Fetch CSV
Feed Auto-Discovery
Fetch Feed
Fetch Data
Fetch Page
Fetch Site Feed
Flickr
Google Base
Item Builder
Yahoo! Local
Yahoo! Search
User inputs
Operators
Url
String
Date
Location
Number
Favorites
My pipes
Fetch Feed
URL
http://www.sciencemag.org/rss/cu
Content Analysis
Fetch Feed
URL
http://www.nature.com/nature/curr
Filter
Block items that match any of the following
Rules
item.title Contains Letters
item.title Contains Technical Commer
item.title Contains Reports
item.title Contains Research Articles
item.title Contains Books Et Al.
item.title Contains departments
item.description Contains "arising from"
Filter
Block items that match any of the following
Rules
item.prism:section Contains letter
item.prism:section Contains correspondence
item.prism:section Contains book
item.prism:section Contains article
Union
Debugger: none
Done
Internet
100%


Pipes: editing 'Science Journals copy' - Windows Internet Explorer
http://pipes.yahoo.com/pipes/pipe.edit?_id=ptn97JkO3BGwCk3rJhOy0Q
File Edit View Favorites Tools Help
Google Calendar
Pipes: editing 'Science Jo...
Science Journals copy*
Layout Expand All Collapse All
Back to My Pipes New Save Save a copy Properties...
Sources
Fetch CSV
Feed Auto-Discover
Fetch Feed
Fetch Data
Fetch Page
Fetch Site Feed
Flickr
Google Base
Item Builder
Yahoo! Local
Yahoo! Search
User inputs
Operators
Url
String
Date
Location
Number
Favorites
My pipes
URL
http://www.sciencemag.org/rss/cu
Content Analysis
Fetch Feed
URL
http://www.nature.com/nature/curr
Filter
Block items that match any of the following
Rules
item.title Contains Letters
item.title Contains Technical Commer
Filter
Block items that match any of the following
Rules
item.prism:section Contains letter
Debugger: Pipe Output (60 items)
Time taken: 1.390366s Refresh
[PERSPECTIVES] DEVELOPMENTAL BIOLOGY: Brain Wnts for Blood Vessels
[PERSPECTIVES] CHEMISTRY: Interrogating Molecules
[NEWS] ASTROPARTICLE PHYSICS: Excess Particles From Space May Hint at Dark Matter
Editors' Choice
[NEWS] CRIMINOLOGY: Study Shows How Degraded Surroundings Can Degrade Behavior
[SPECIAL FEATUREs] SCIENCE CAREERS: Finance's Quant(um) Mechanics
[POLICY FORUM] POLICY: A Force for Peace in the Middle East
[NEWS FOCUS] SCIENCE IN ROMANIA: At Home in Bucharest, for Better and for Worse
+52 more...
Done
Internet
100%


Yahoo! Local
Find text [wired] within 5 miles of location [wired]
For Each: Annotate
For each item in input feed, set attribute flickr to all items output from
Search near (e.g. Napa
Name: locationinput1
Flickr
Find 1 imag
near
Map
List
20 items
RUTHERFORD
OAKVILLE
Silverado Trl
Oakville Cross Rd
29
Oakville Grade Rd
St Helena Hwy
Yountville Cross Rd
Yountville
Quixote Winery
ATLAS
NAPA SODA SPRINGS
Dry Creek Rd
LOKOYA
OAK KNOLL
Oak Knoll Ave
Map Sat Hyb
Data ©2008 NAVTEQ
©2008 Yahoo! Inc.
Rename
Mappings
item.flickr.description Copy A
Rename
Mappings
item.description.desc Renan
Debugger: Location Input (0 items)
ime taken: 0.000174s Refresh
country United States



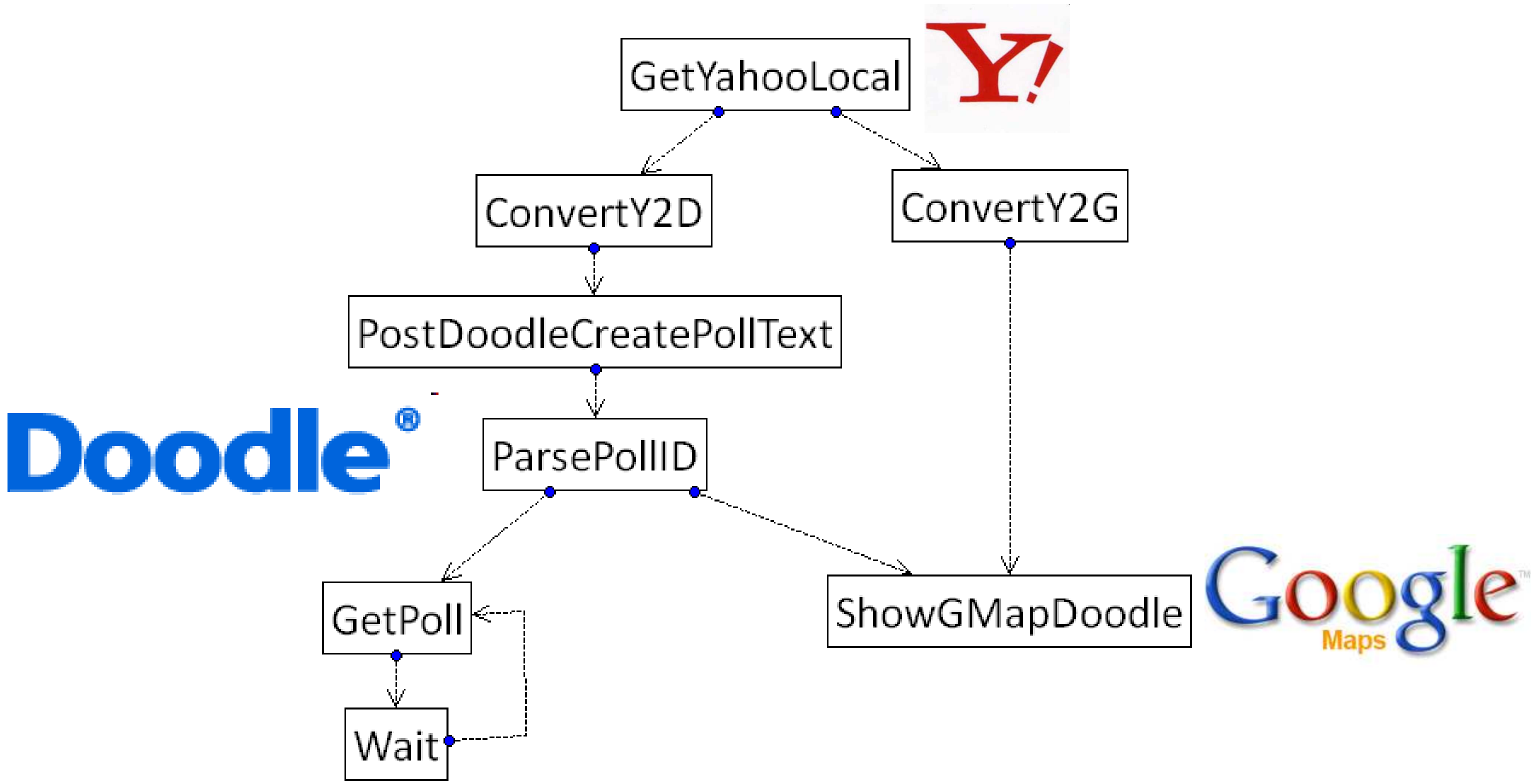
www.jopera.org

Powered by

From: Cesare Pautasso

JOpera Example: Doodle Map Mashup

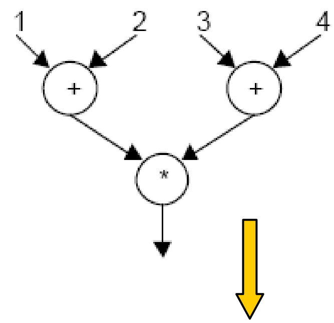
- Setup a Doodle with Yahoo! Local search and visualize the results of the poll on Google Maps



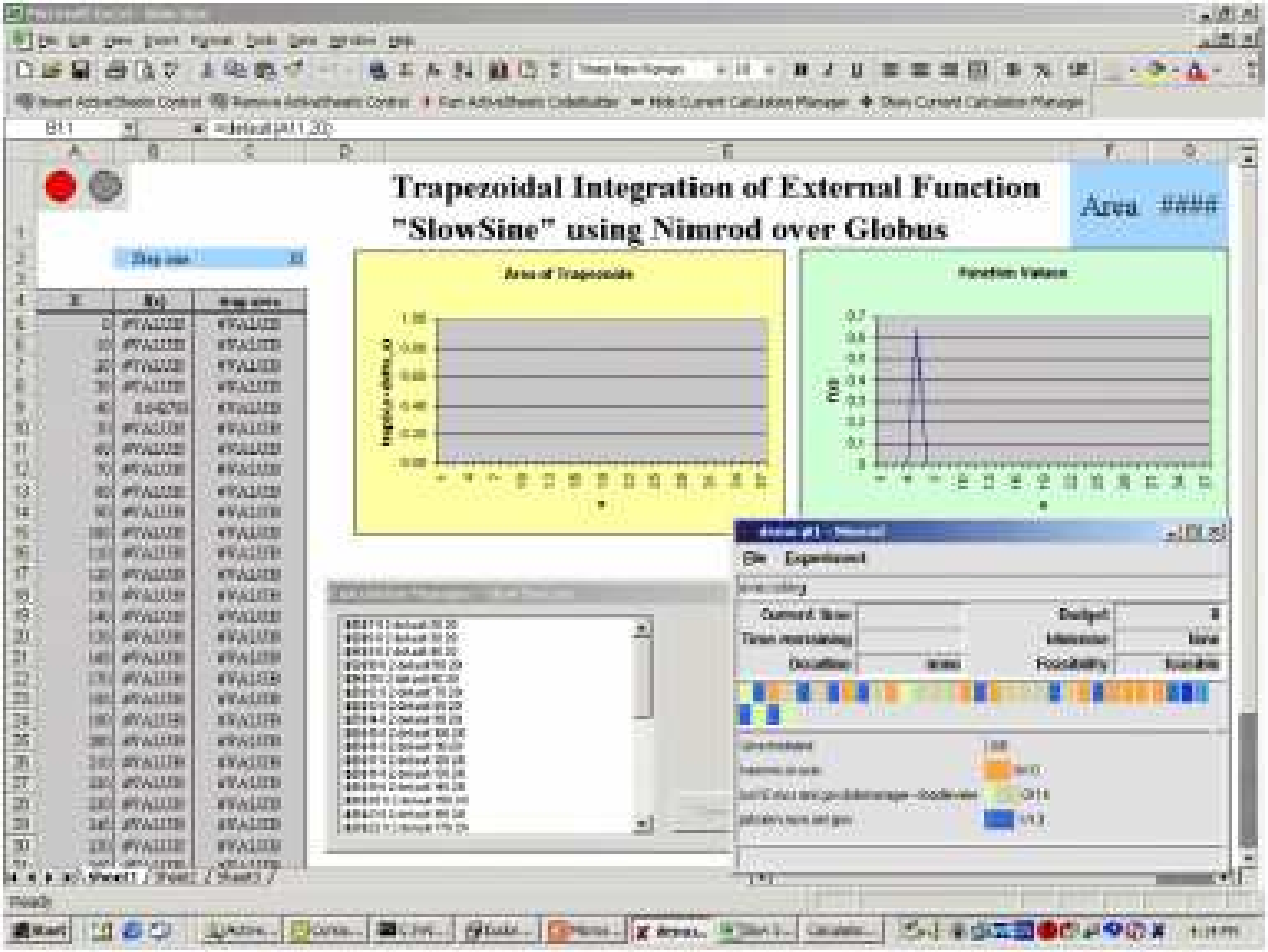
From: Cesare Pautasso

ActiveSheets/EnFuzion

- Extend MS-Excel to support execution of functions
  - Excel → Nimrod Cache → Nimrod-based execution
  - Use OLE extensions (VB + ActiveX DLL)
- Support for parallel evaluation of cells within a spreadsheet
  - Results of one cell may feed as input into another
- Use of custom functions (rather than built in ones in Excel) – evaluate cells in a data flow manner
  - Cells must be "functionally" independent
- Table used to maintain current state:
  - not evaluated, under evaluation, evaluated
- Custom function returns "before" evaluation completes – causing other functions to be evaluated in parallel



	A	B	C	D
1	1	2	3	4
2	=A1+B1		=C1+D1	
3	=A2+C2			


Trapezoidal Integration of External Function "SlowSine" using Nimrod over Globus
Area ####

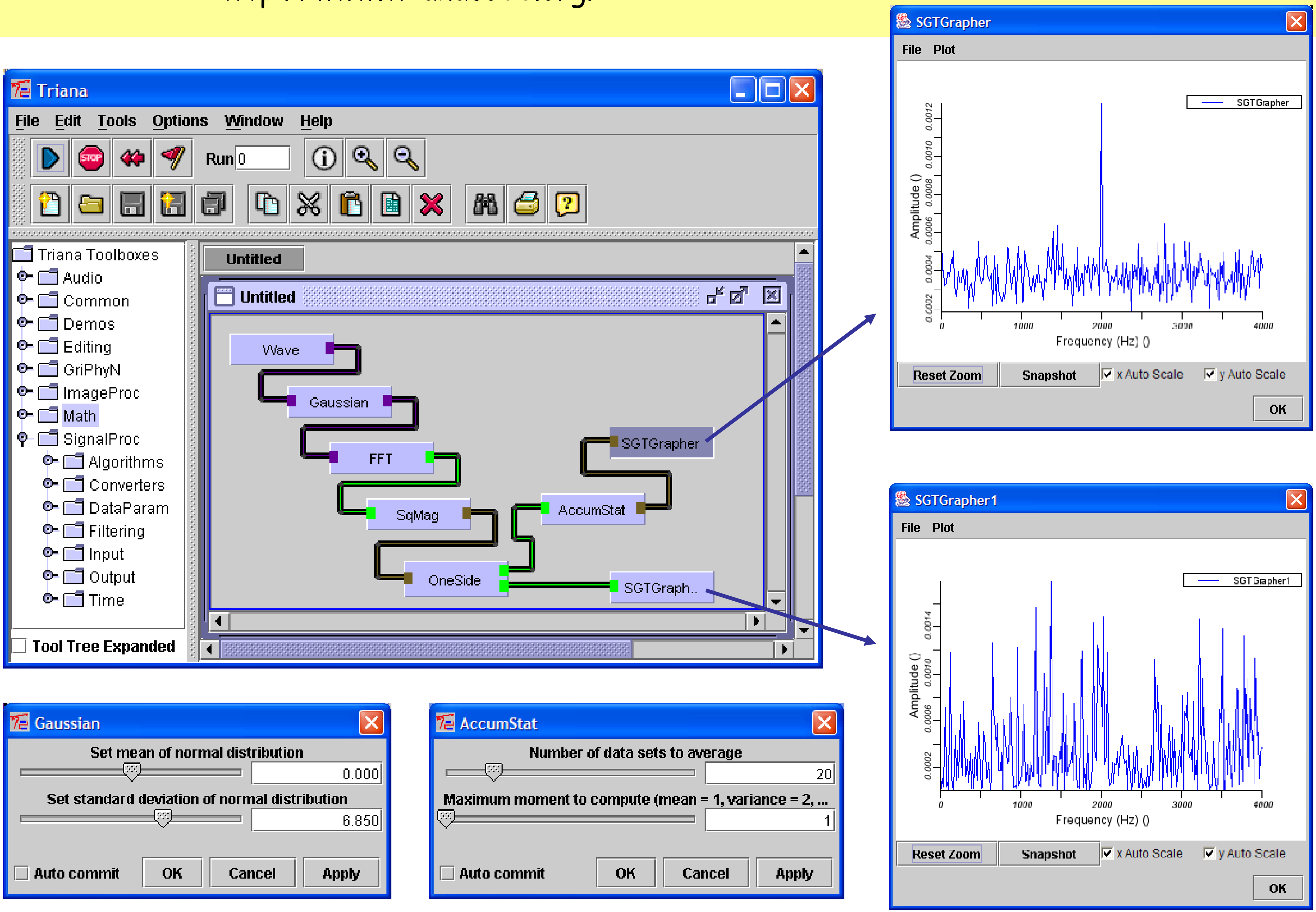
Enactment for Automated Composition (more later)

- Enactment engine enlists use of other components
  - Discovery Service
  - Planning Engine
- Enactment may be "goal-oriented"
  - Define requirement, rather than components
  - Conflict detection support
  - Mechanisms to chose between alternatives (constraints)

Difficult to do in practice

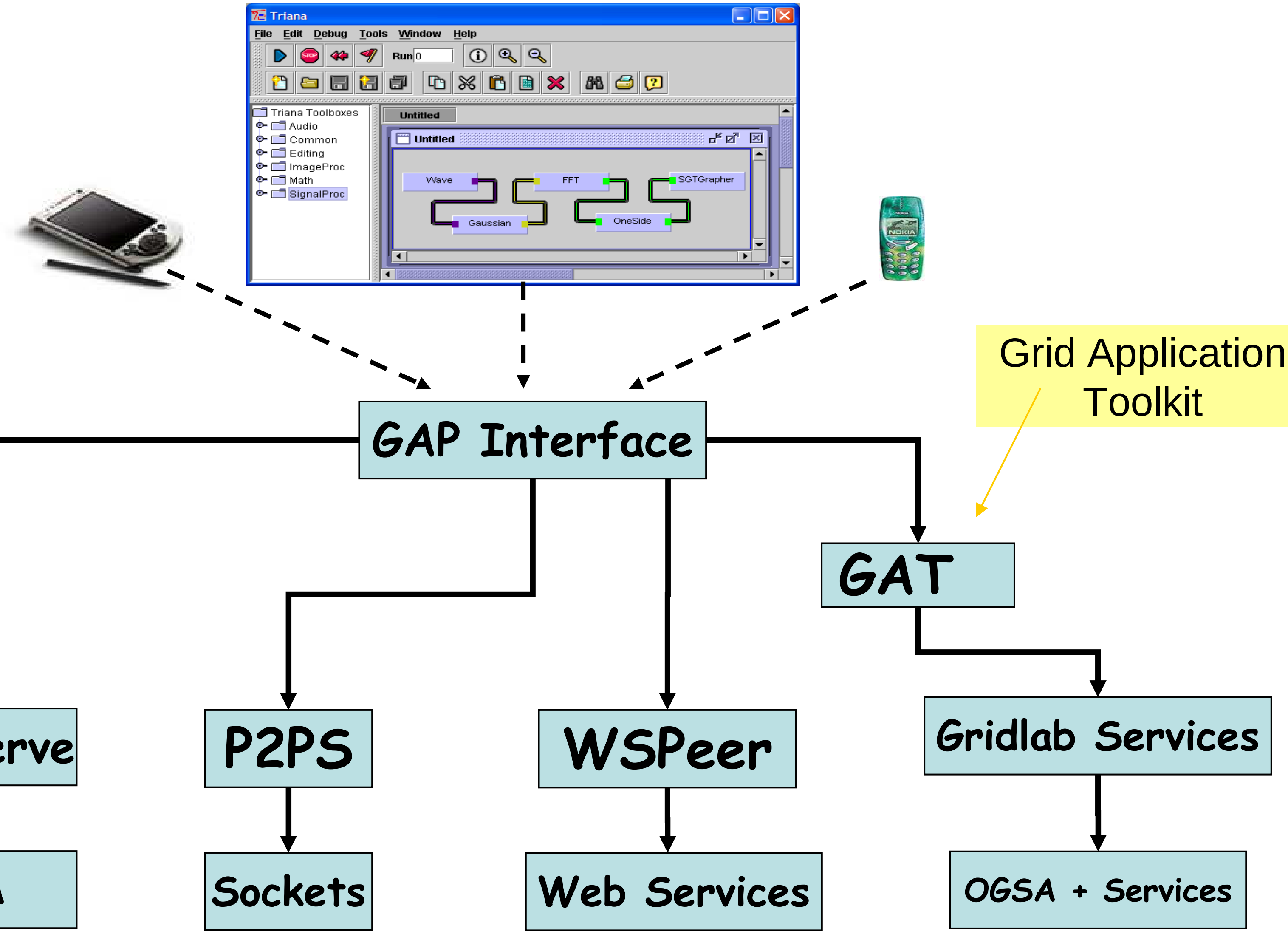
http://www.gridlab.org/

http://www.trianacode.org/



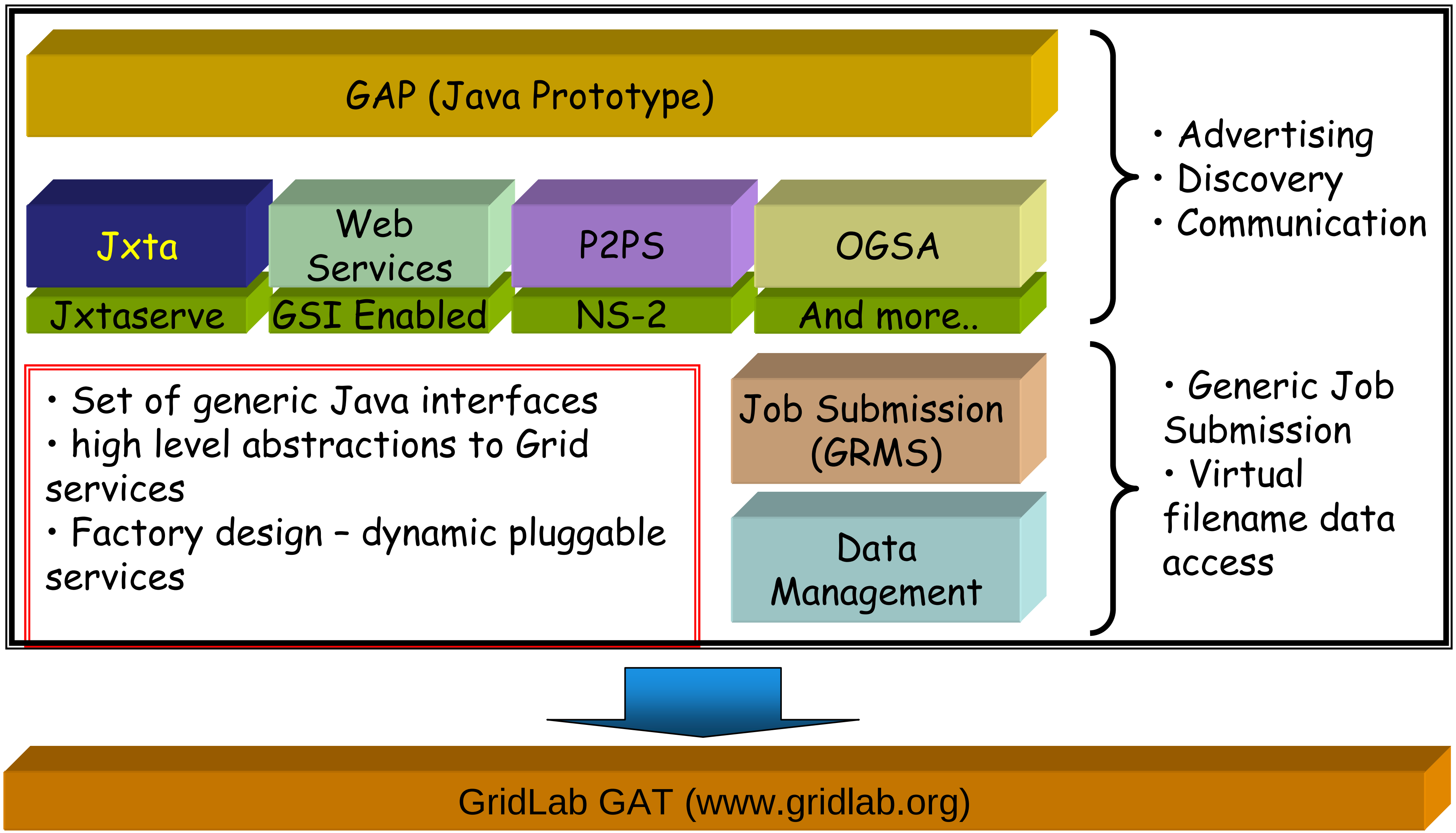
GridLab Implementation

http://www.trianacode.org/

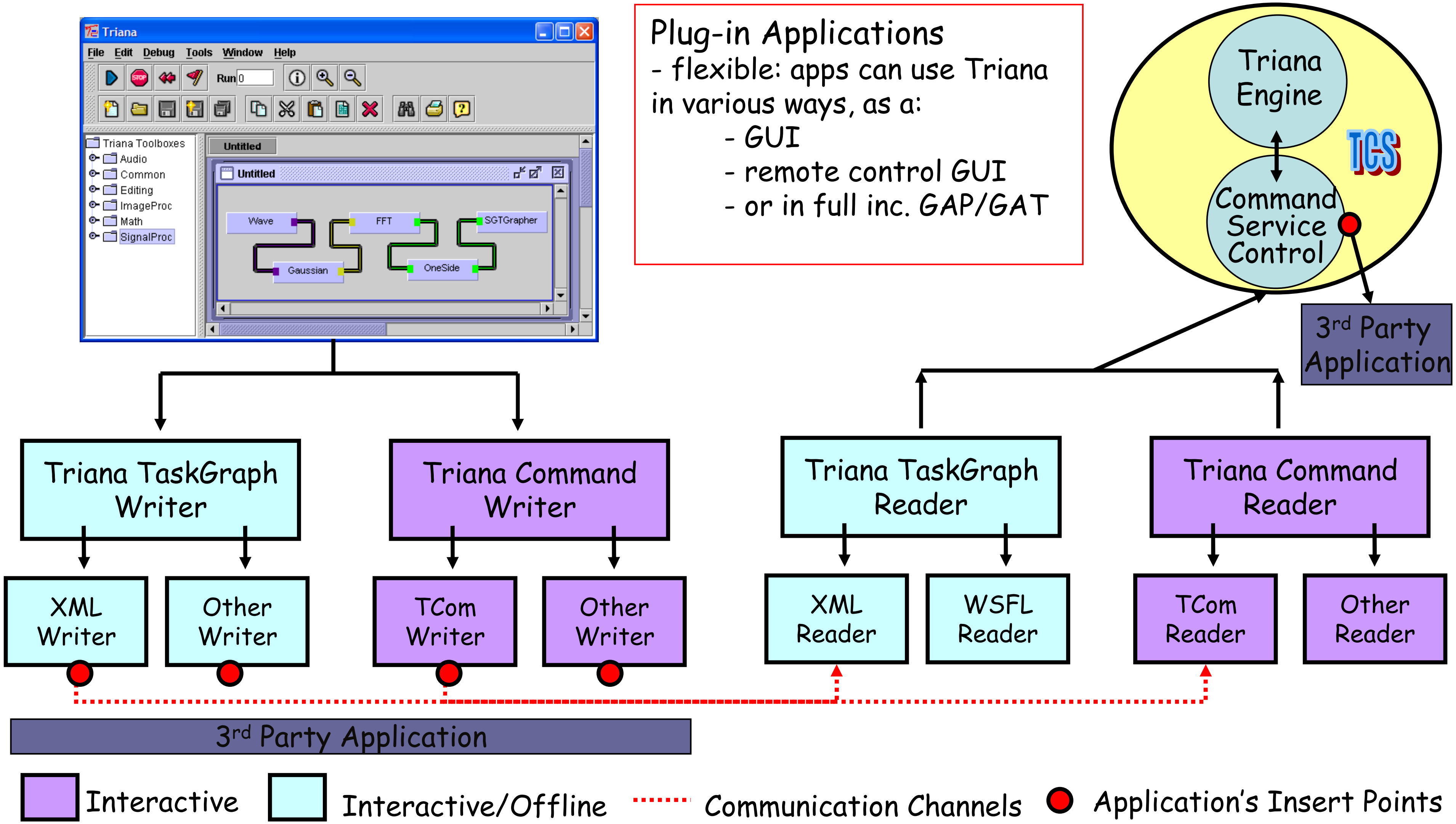


Grid Application Toolkit

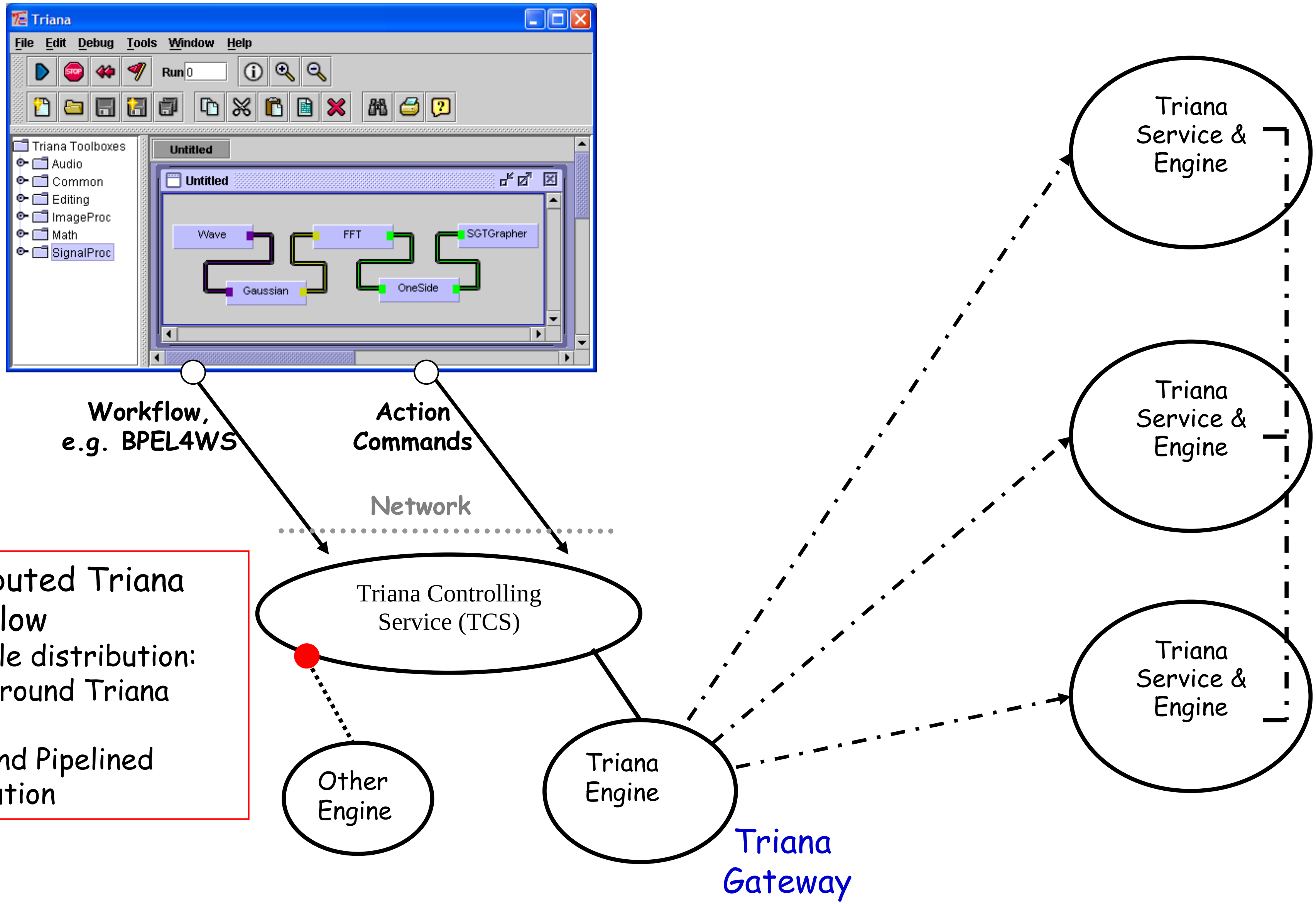
Java GAT Prototype



Triana Architecture



Triana Distributed Workflow

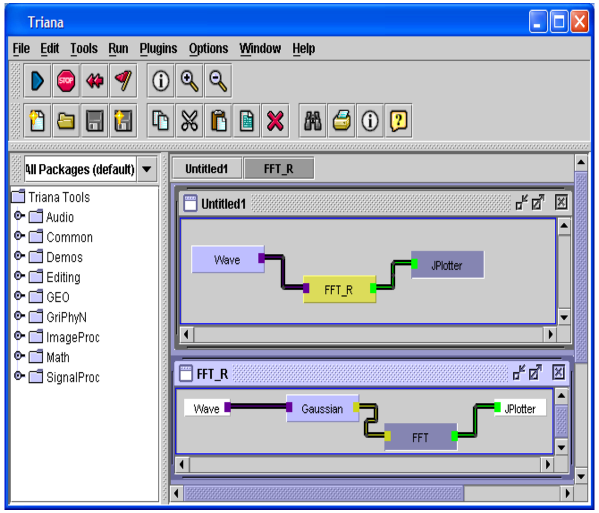


Distributed Triana Workflow
- flexible distribution: based around Triana Groups
- HPC and Pipelined distribution

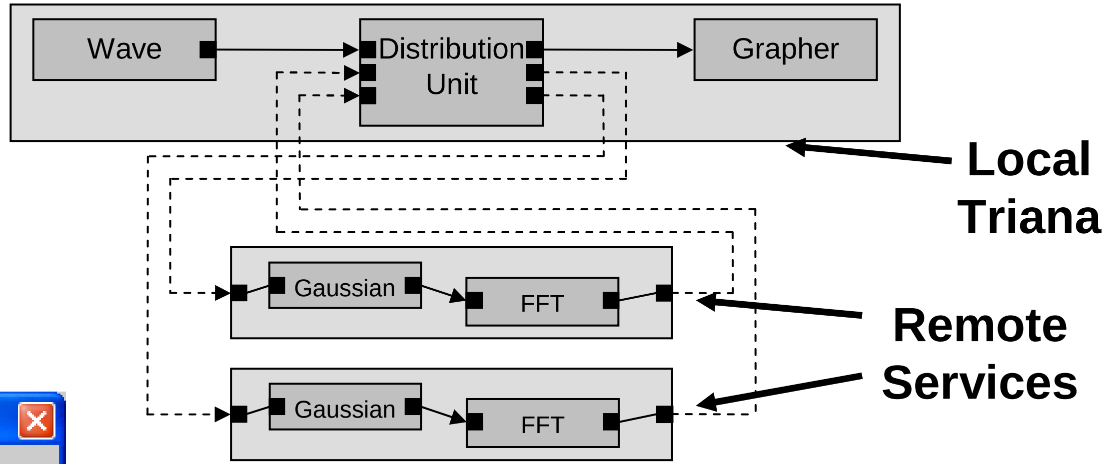
Distributing Triana Taskgraphs

- Mapping tasks or groups of tasks to resources
- Two stages:
  - Taskgraph annotation, XML definition for each task or group of tasks
  - extended to specify resources and message channels
  - Data distribution, annotated sub-sections of taskgraph passed to resources

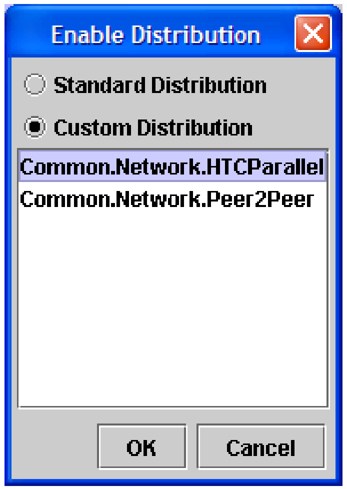
Custom Distribution



- The workflow is cloned/split/rewired to achieve the required distribution topology



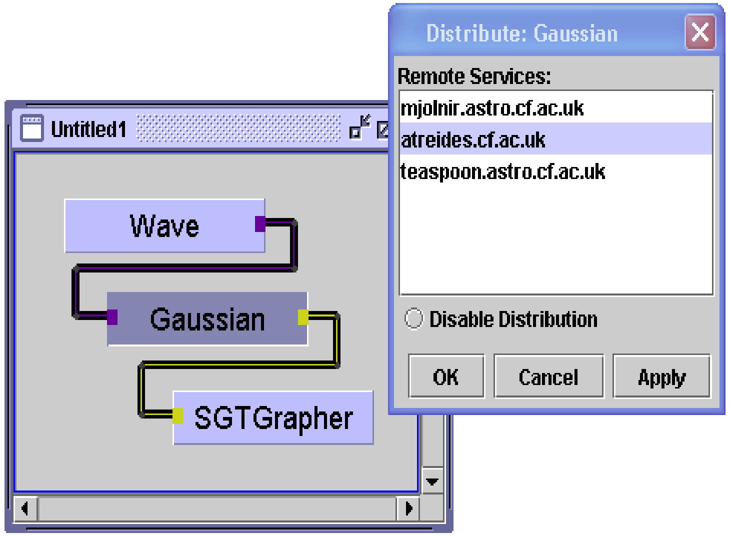
- Custom distribution units allow sub-workflows to be distributed in parallel or pipelined

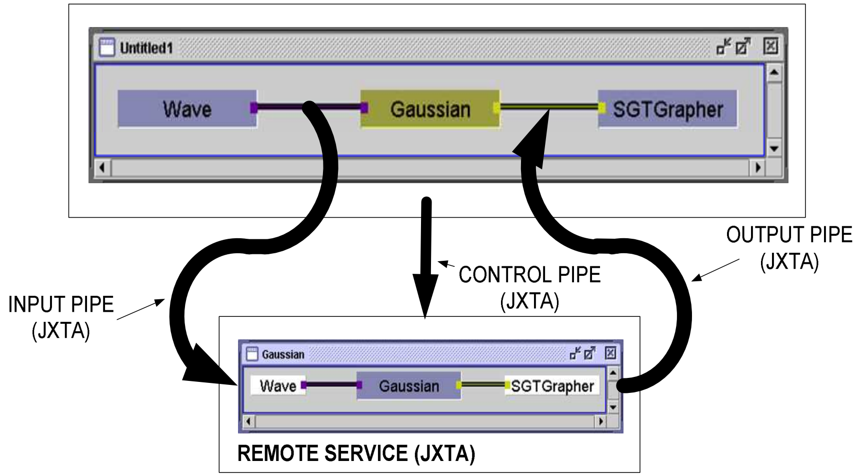


- Distribution units are standard Triana tools, enabling users to create their own custom distributions

Remote Deployment

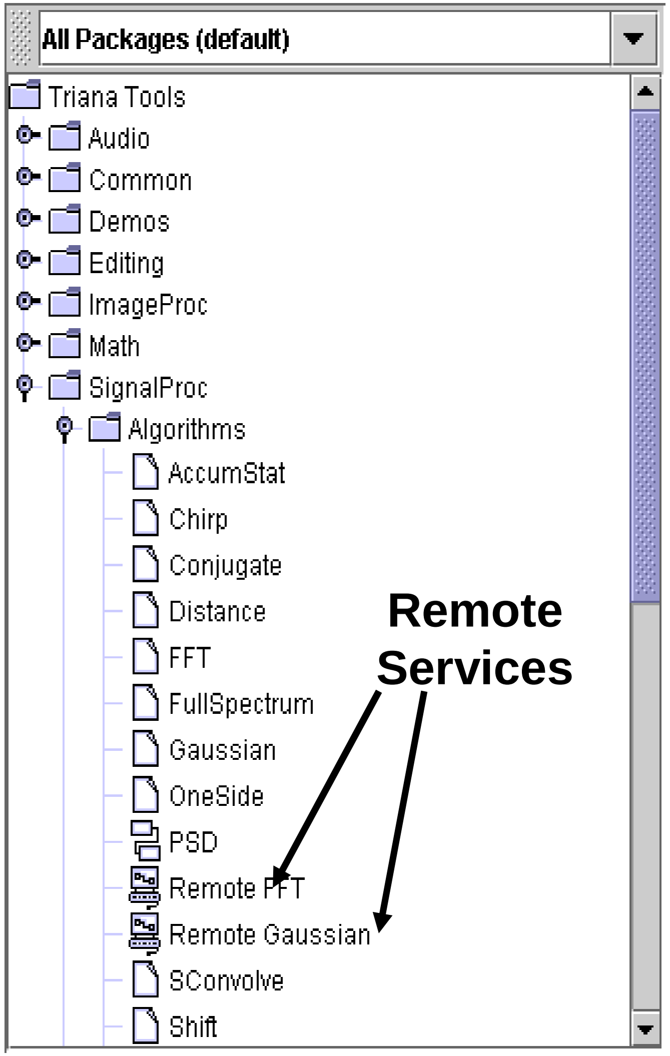
- User can distribute any task or group of tasks (sub-workflow)
- Using the GAP Interface, Triana automatically launches a remote service providing that sub-workflow.
- Input, Output and Control Pipes are connected using the current GAP binding (e.g. JXTA Pipes)





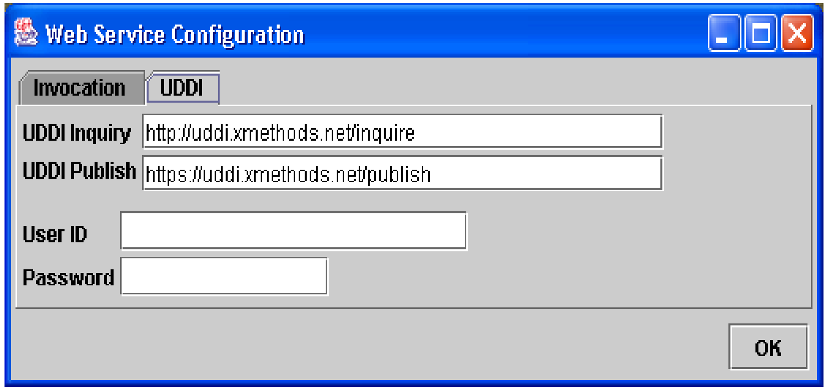
Deploying and Connecting To Remote Services

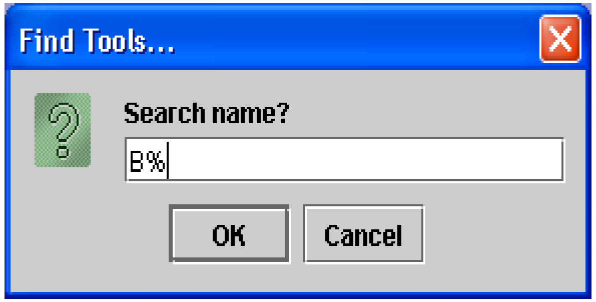
- Running services are automatically discovered via the GAP Interface, and appear in the tool tree
- User can drag remote services onto the workspace and connect cables to them like standard tools (except the cables represent actual JXTA/P2PS pipes)



Web Service Discovery 1

- Triana allows users to query UDDI repositories

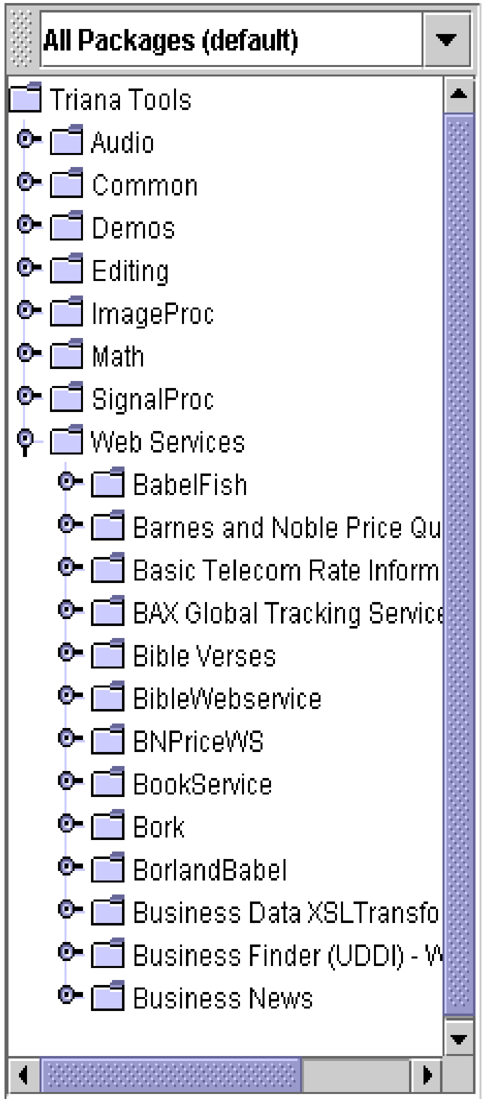




- Alternatively, users can import services directly from WSDL

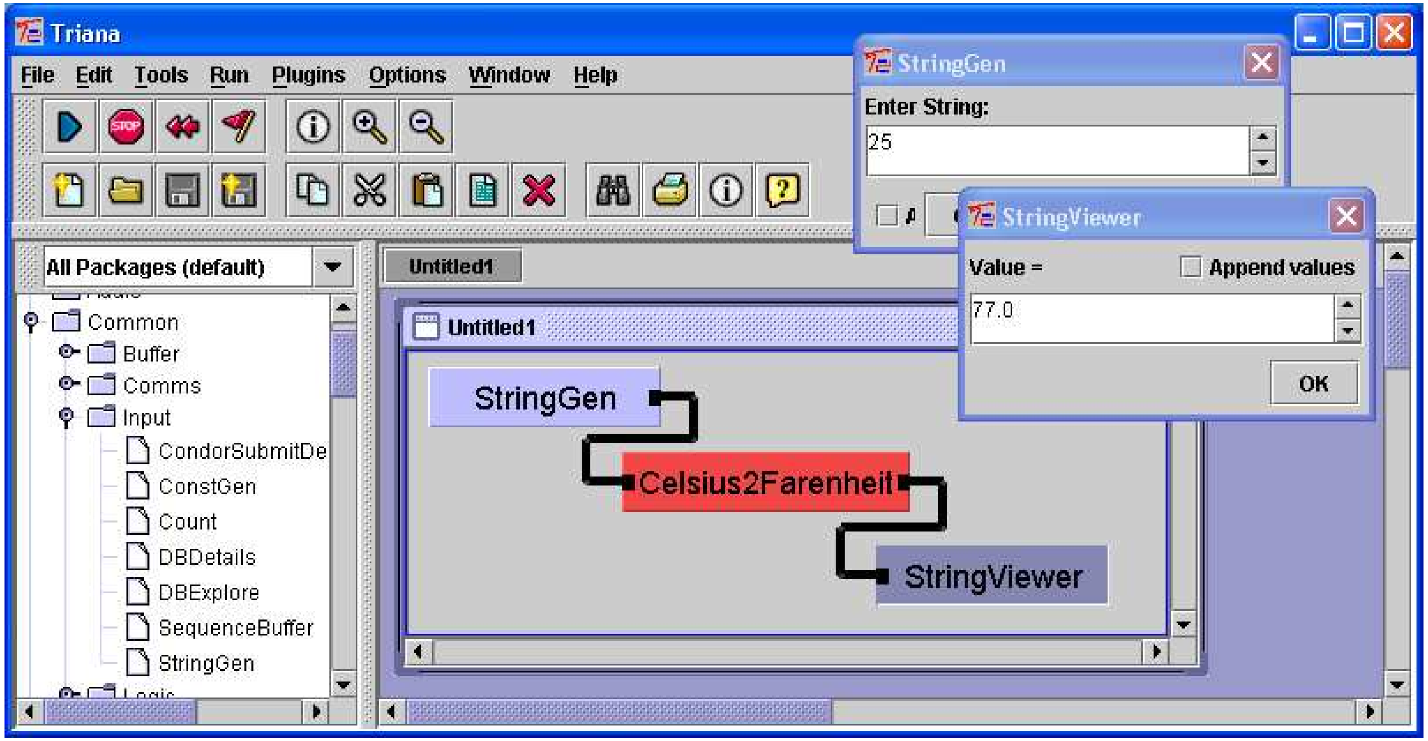
Web Service Discovery 2

- Discovered/Imported Web Services are converted into Triana tools
  - (service name = tool name)
  - (input message parts = in nodes)
  - (output message parts = out nodes)
  - etc…
- Web Service tools are displayed in the user's Tool Tree (alongside local tools)



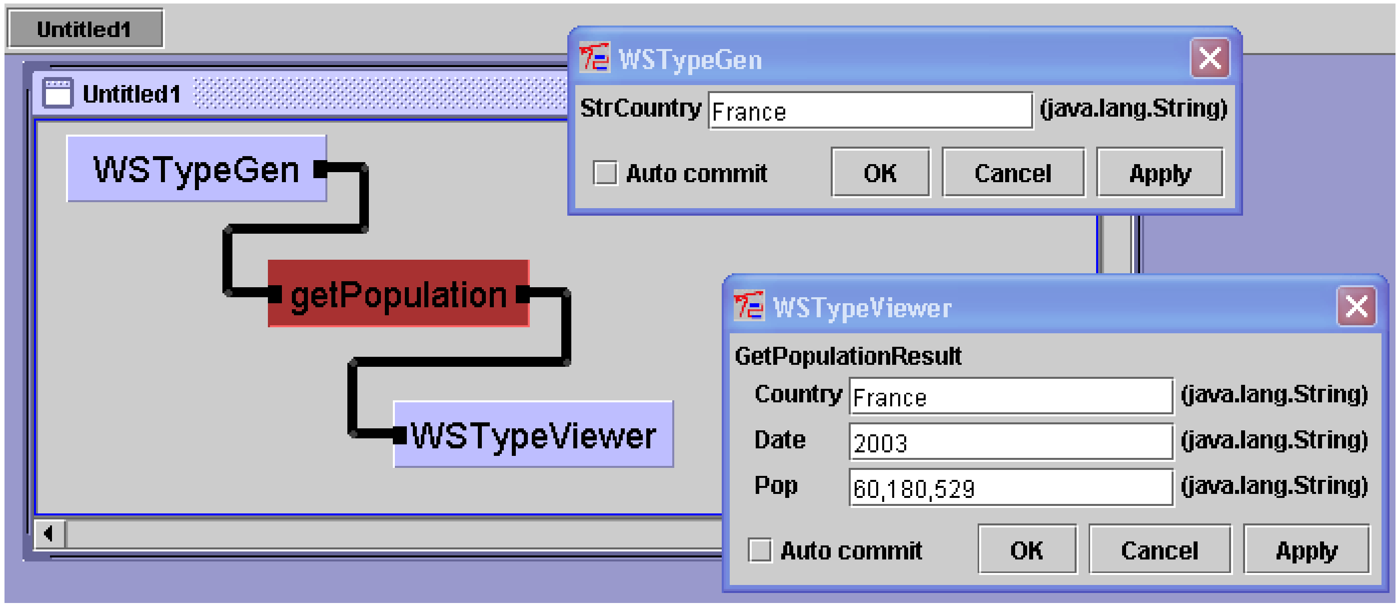
Connecting Workflows

- Web Service tools can be dropped onto the user's workspace and connected like local tools
- A workflow can contain both local and Web Service tools

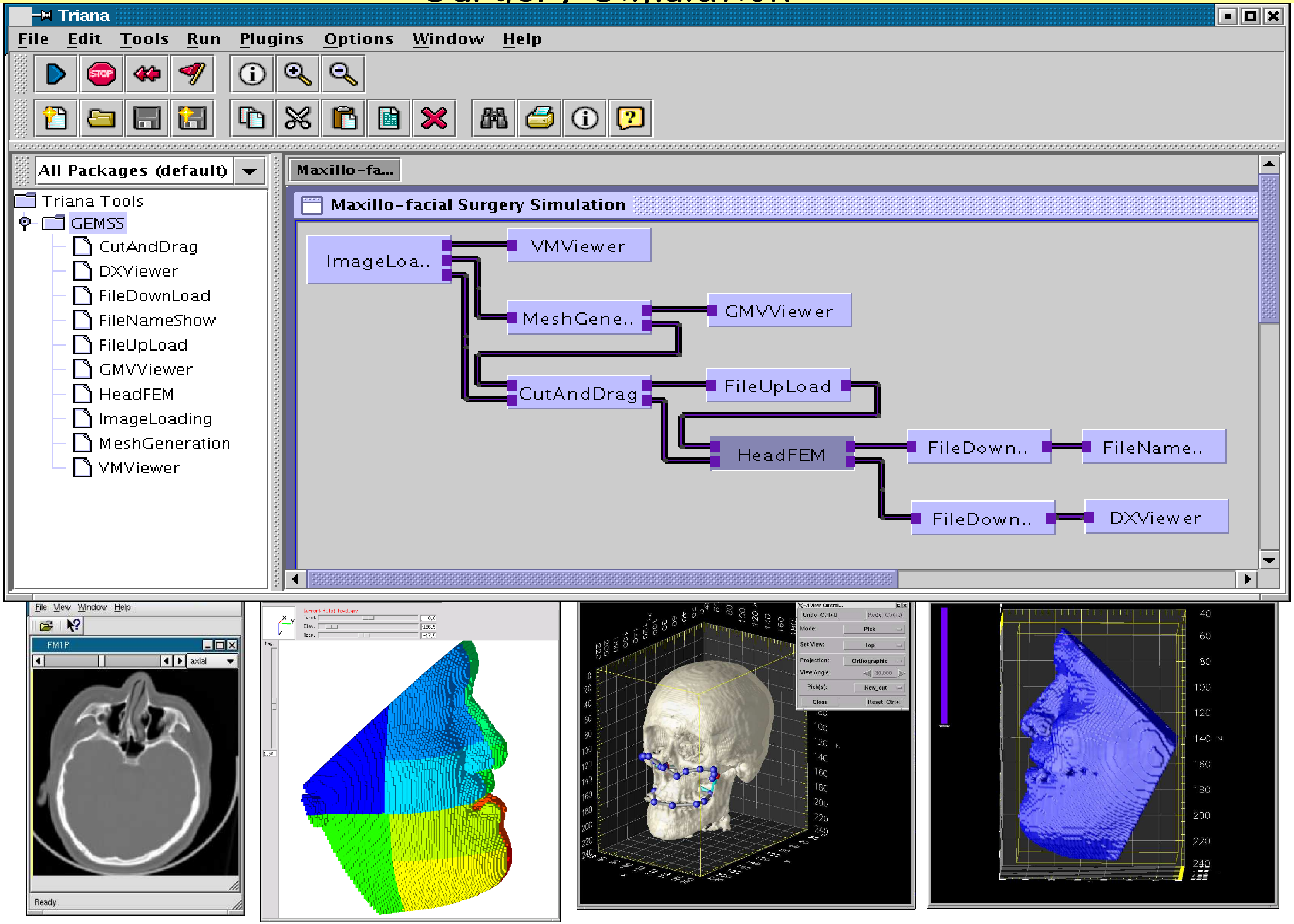


Complex Data Types

- Users can build their own interface for creating/mediating between complex types
- Alternatively, Triana can dynamically generate an interface from the WSDL2Java generated bean class



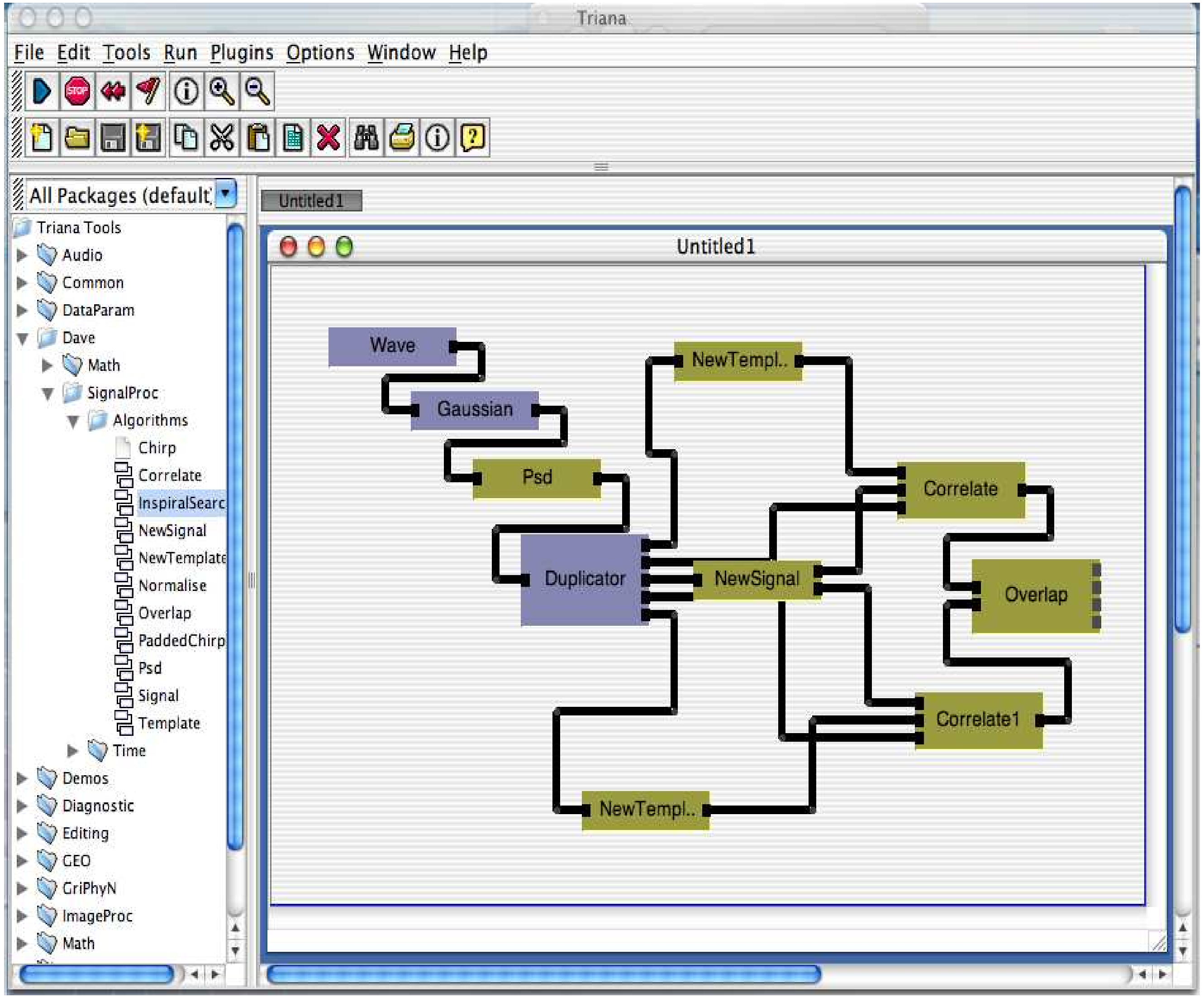
GEMSS: Maxillo-facial Surgery Simulation



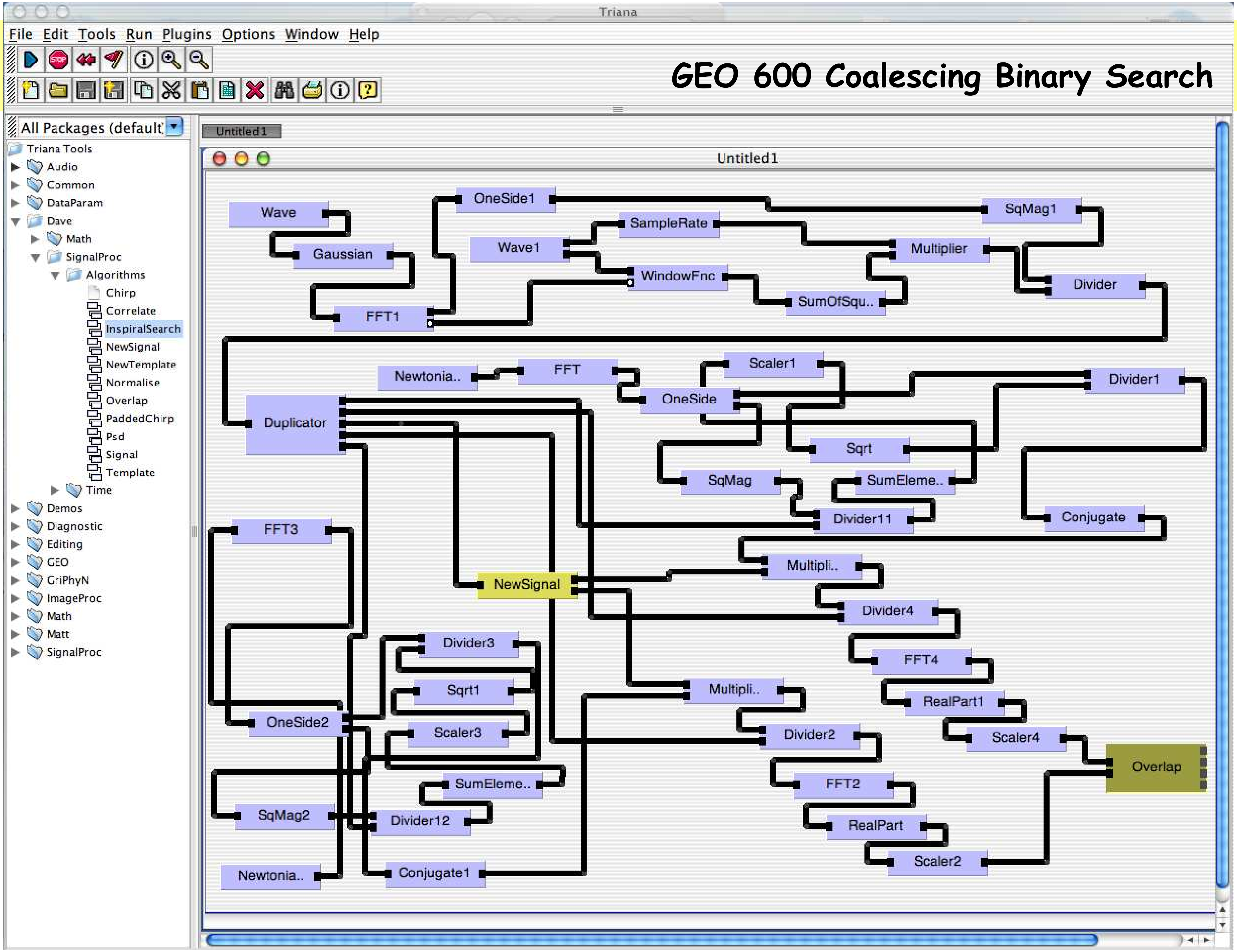
GEO 600 Inspiral Search

- Background
  - Compact binary stars orbiting each other in a close orbit
    - among the most powerful sources of gravitational waves
  - As the orbital radius decreases a characteristic chirp waveform is produced - amplitude and frequency increase with time until eventually the two bodies merge together
- Computing
  - Need 10 Gigaflops to keep up with real time data (modest search..)
    - Data 8kHz in 24-bit resolution (stored in 4 bytes) -> Signal contained within 1 kHz = 2000 samples/second
    - divided into chunks of 15 minutes in duration (i.e. 900 seconds) = 8MB
- Algorithm
  - Data is transmitted to a node
  - Node initialises i.e. generates its templates (around 10000)
  - fast correlates its templates with data

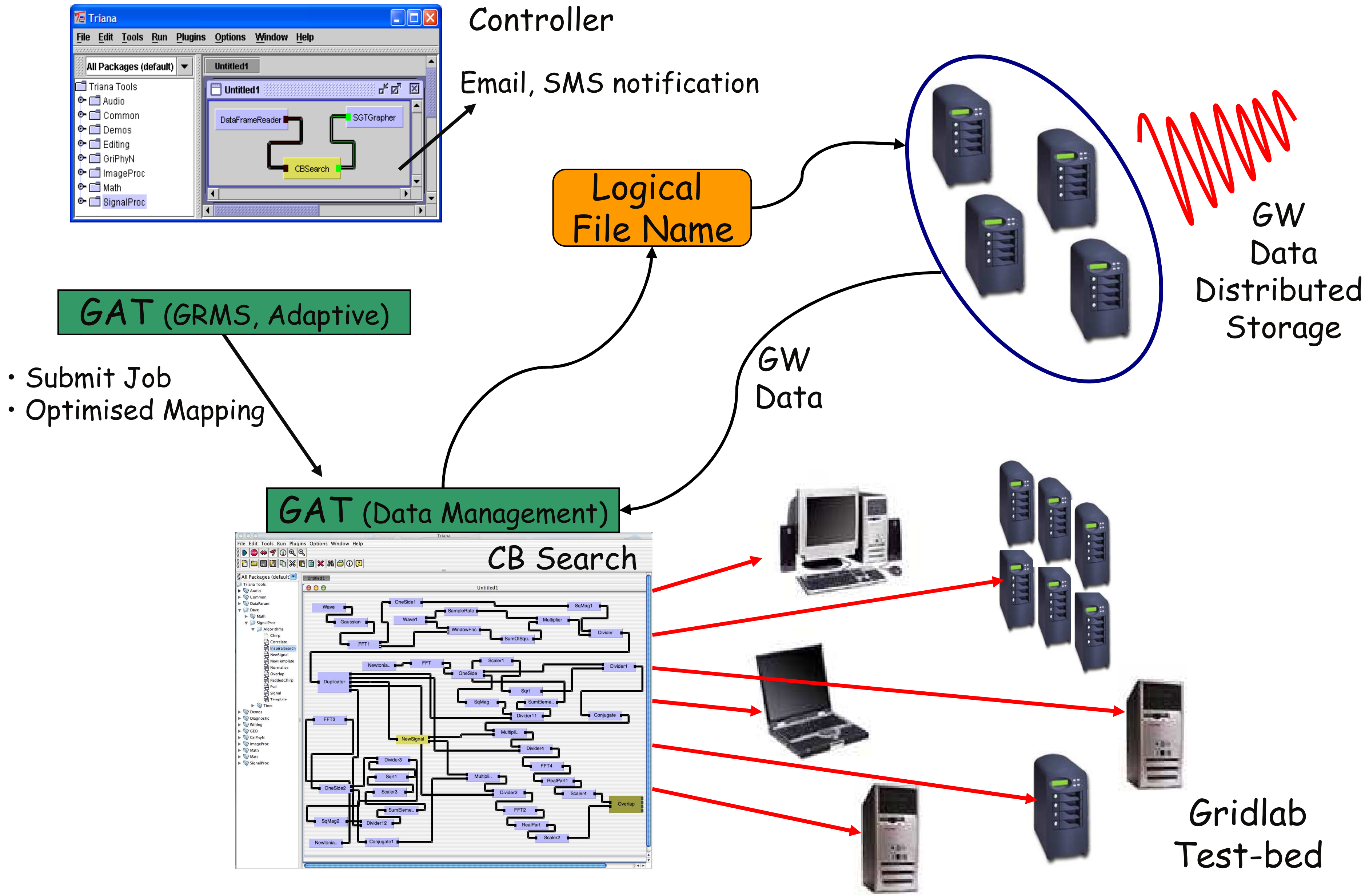
Coalescing Binary Search



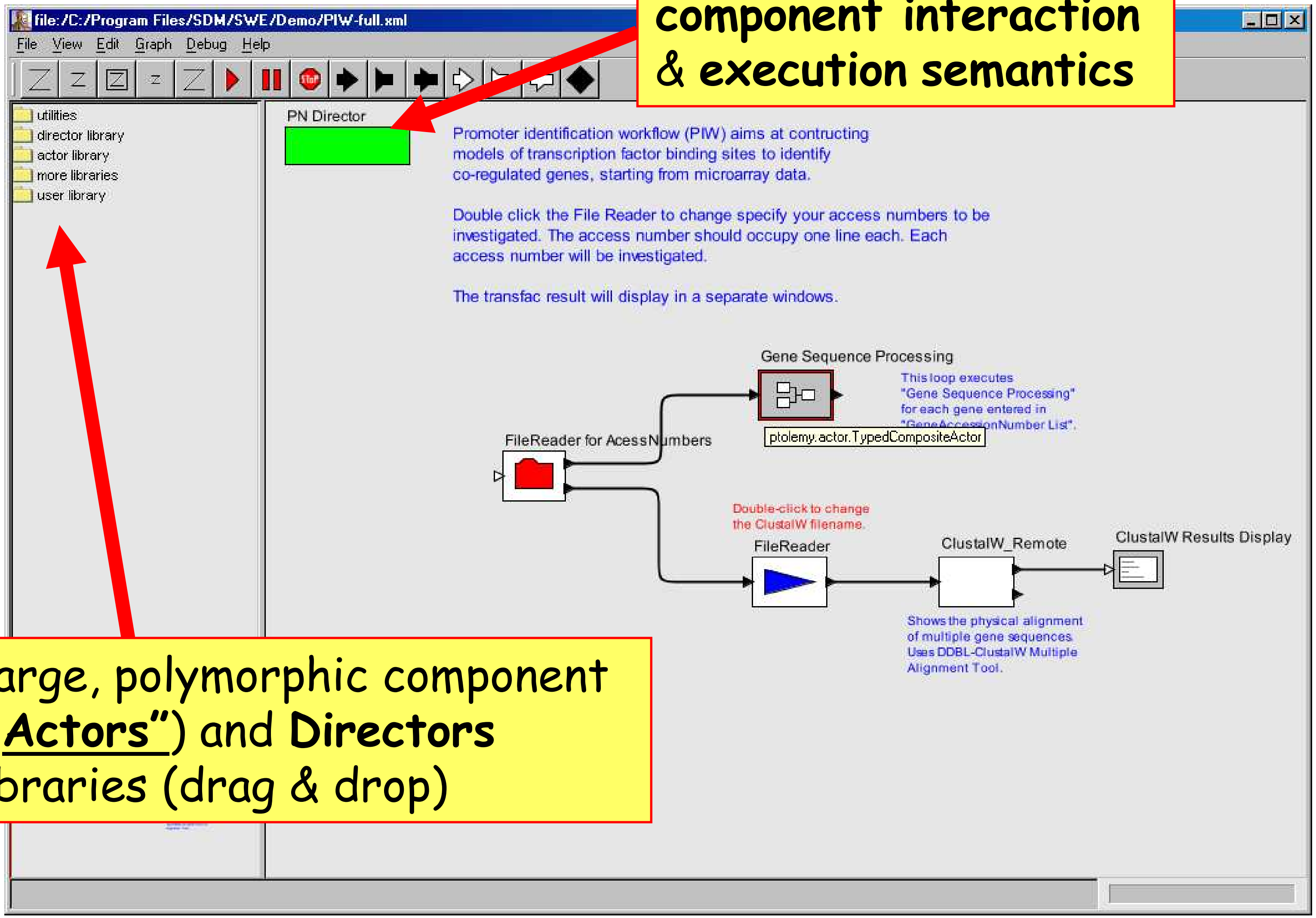
GEO 600 Coalescing Binary Search



Coalescing Binary Scenario



The KEPLER/Ptolemy II GUI (Vergil)

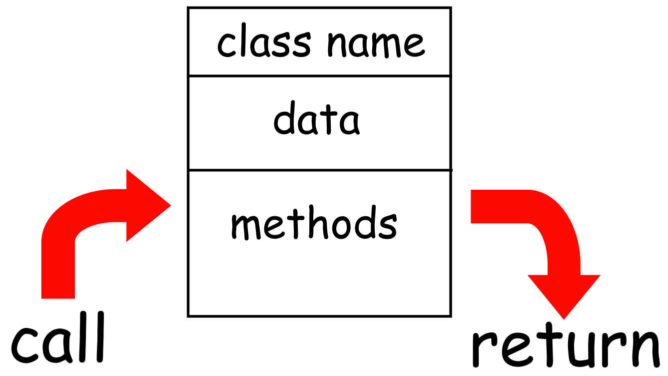


"Directors" define the **component interaction & execution semantics**

Large, polymorphic component (**"Actors"**) and **Directors** libraries (drag & drop)

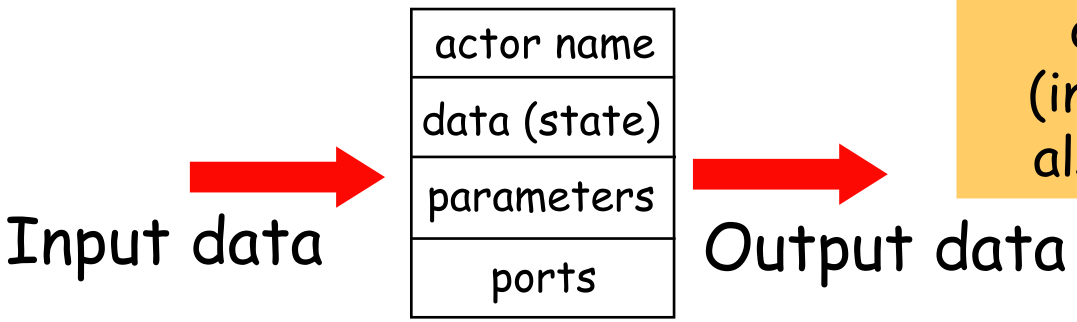
Actor-Oriented Design

- Object orientation:



What flows through an object is sequential control (cf. CCA, MPI)

- ***Actor/Dataflow orientation:***



What flows through an object is a stream of data tokens (in SWFs/KEPLER also **references!!**)

Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/

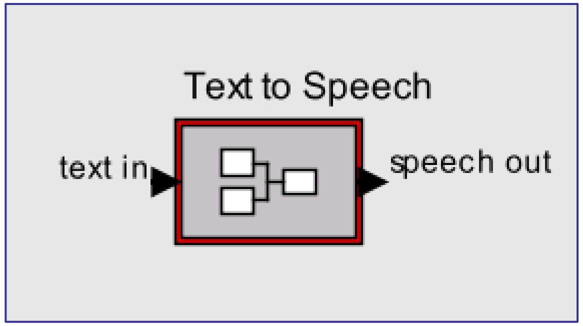
Object-Oriented vs. Actor-Oriented Interfaces

Object Oriented

TextToSpeech
initialize(): void notify(): void isReady(): boolean getSpeech(): double[]

OO interface gives procedures that have to be **invoked in an order not specified as part of the interface definition**.

Actor/Dataflow Oriented



AO interface definition says "Give me text and I'll give you speech"

Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/

Ptolemy II: Actor-Oriented Modeling

- “Director” acts as an enactor
  - In this instance, interaction semantics are not maintained within a component
  - This is equivalent to having a centralised enactor

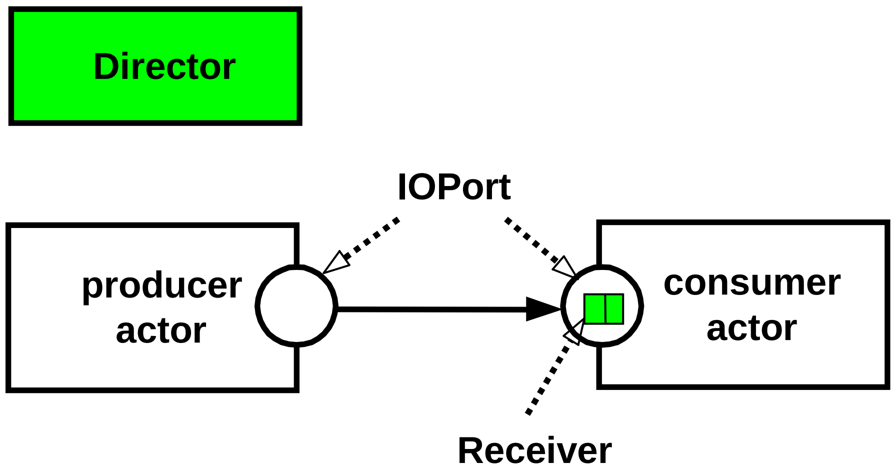
- Different directors for different modeling and execution needs
  - Hence, a variety of directors can operate on the same components

➔ Better **abstraction**, **modeling**, **component reuse**, …

Behavioral Polymorphism **in Ptolemy**

«Interface» ***Receiver***
+get() : Token *+getContainer() : IOPort* *+hasRoom() : boolean* *+hasToken() : boolean* *+put(t : Token)* *+setContainer(port : IOPort)*

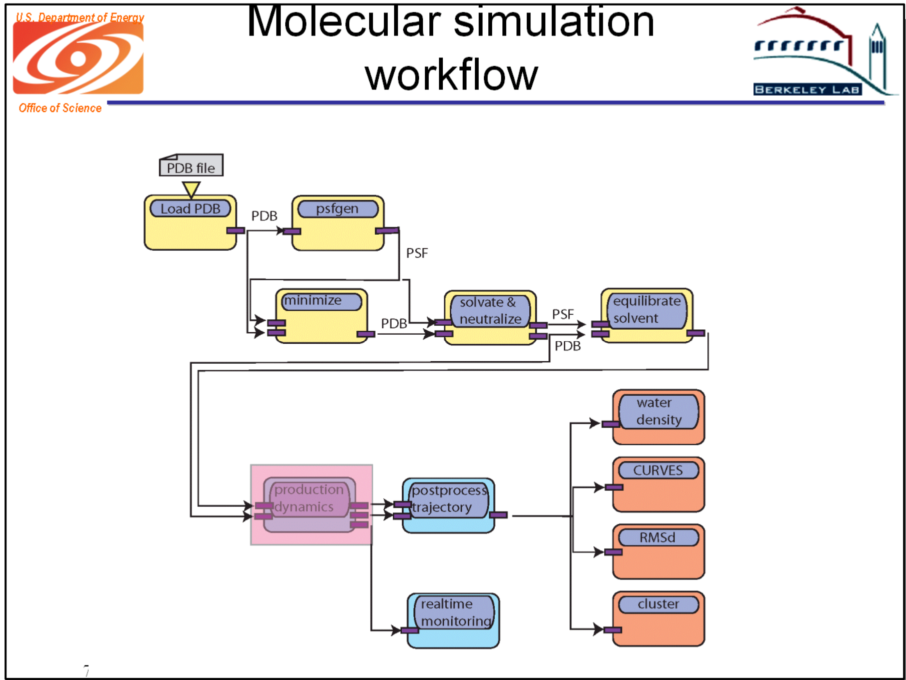
These polymorphic methods implement the **communication semantics** of a **domain** in Ptolemy II. The **receiver instance** used in communication is **supplied by the director, not by the component**.
(cf. CCA, WS-??, [G]BPL4??, … !)

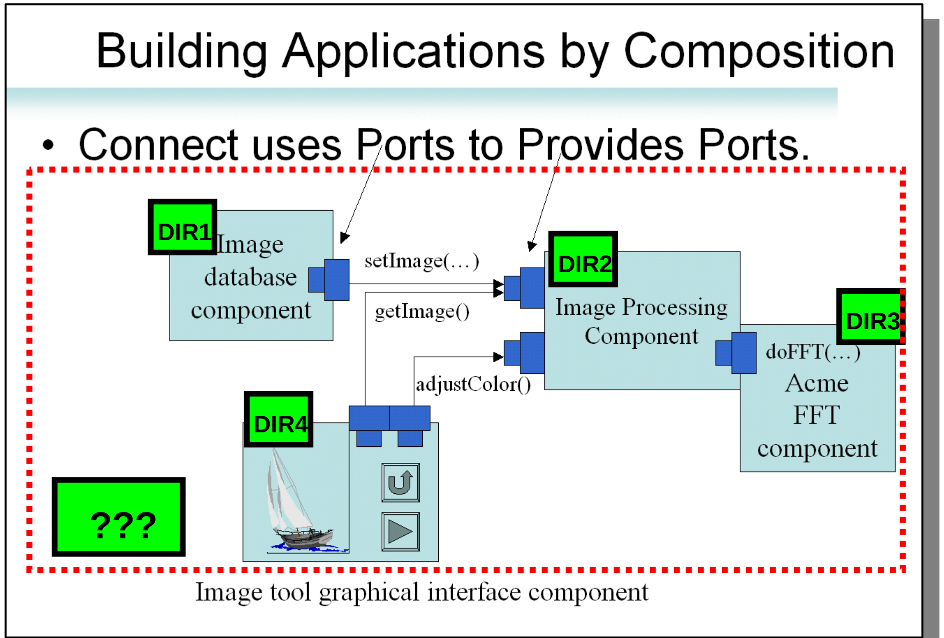


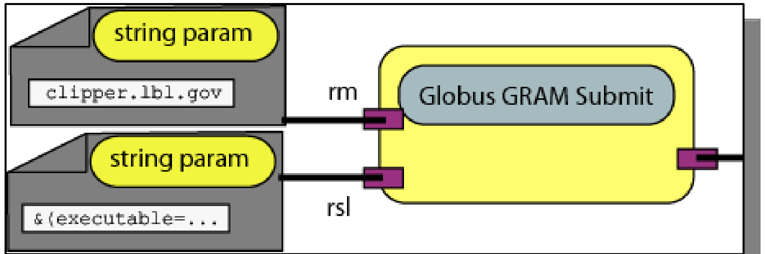
Behavioral polymorphism is the idea that components can be defined to operate with **multiple models of computation** and **multiple middleware frameworks**.

Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/

Component Composition & Interaction







- Components linked via ports
- Dataflow (and msg/ctl-flow)
- **Where is the component interaction semantics defined??**
  - each component is its own director!
- But still useful for special applications, e.g. parallel programs (MPI, …)

Source: GRIST/SC4DEVO workshop, July 2004, Caltech

Domains **and** Directors: **Semantics for** Component Interaction

- CI – Push/pull component interaction
- CSP – concurrent threads with rendezvous
- CT – continuous-time modeling
- DE – discrete-event systems
- DDE – distributed discrete events
- FSM – finite state machines
- DT – discrete time (cycle driven)
- Giotto – synchronous periodic
- GR – 2-D and 3-D graphics
- **PN – process networks**
- **SDF – synchronous dataflow**
- SR – synchronous/reactive
- TM – timed multitasking

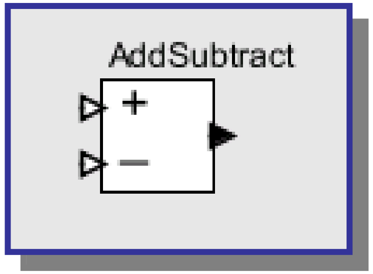
For (**finer-grained**) **concurrent jobs**!?

For (**coarse grained**) **Scientific Workflows**!

Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/

Polymorphic Actor Components Working Across Data Types and Domains

- Actor **Data Polymorphism**:
  - Add ***numbers*** (int, float, double, Complex)
  - Add ***strings*** (concatenation)
  - Add ***complex types*** (arrays, records, matrices)
  - Add ***user-defined types***

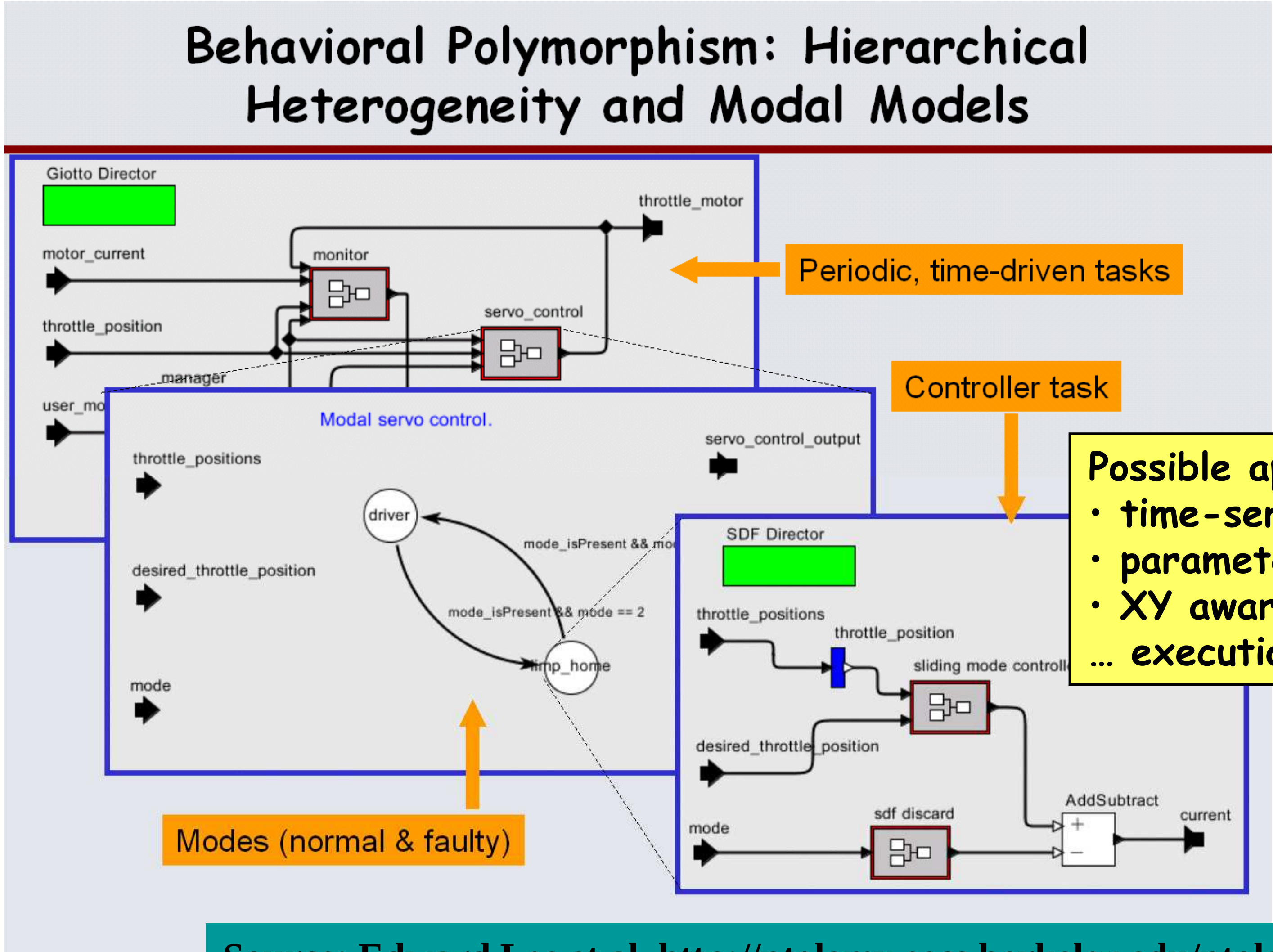


- Actor **Behavioral Polymorphism**:
  - In ***dataflow***, add when all connected inputs have data
  - In a ***time-triggered model***, add when the clock ticks
  - In ***discrete-event***, add when any connected input has data, and add in zero time
  - In ***process networks***, execute an infinite loop in a thread that blocks when reading empty inputs
  - In ***CSP***, execute an infinite loop that performs rendezvous on input or output
  - In ***push/pull***, ports are push or pull (declared or inferred) and behave accordingly
  - In ***real-time CORBA***, priorities are associated with ports and a dispatcher determines when to add

By not choosing among these when defining the component, we get a huge increment in component re-usability. But how do we ensure that the component will work in all these circumstances?

Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/

Directors and Combining Different Component Interaction Semantics

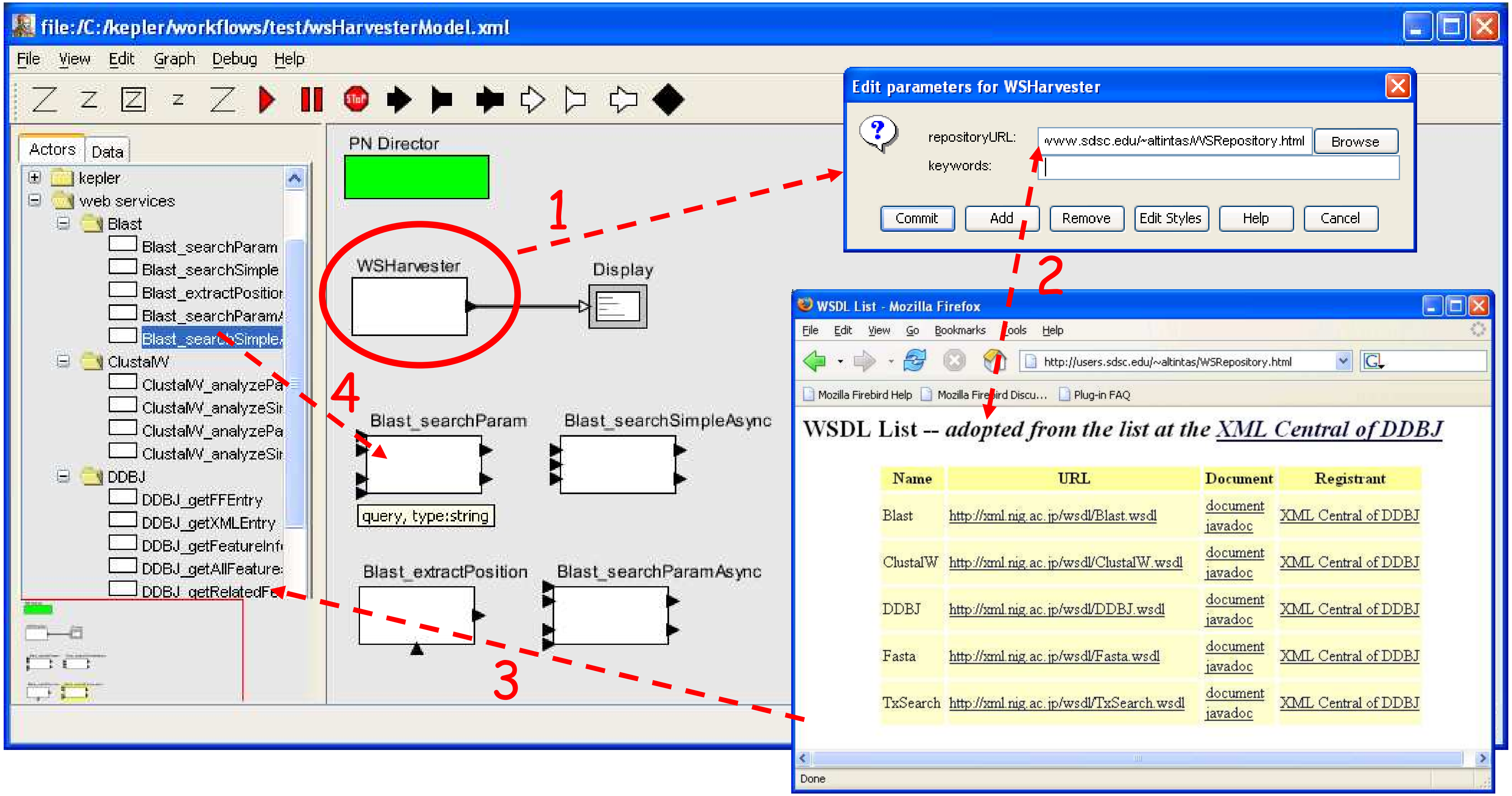


Possible app. in SWF:
- **time-series aware …**
- **parameter-sweep aware …**
- **XY aware …**

… execution models

Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/ptolemyII/

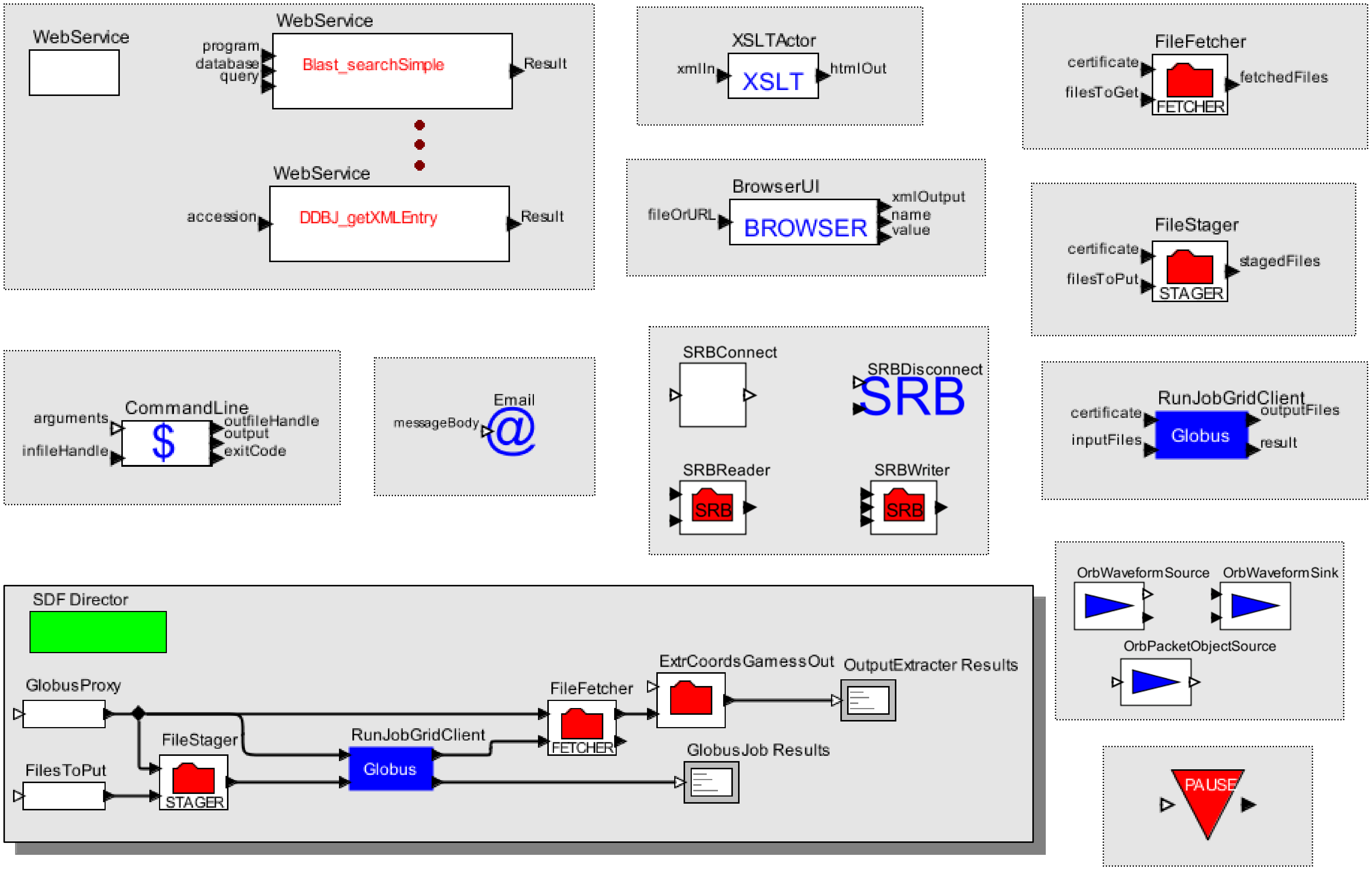
Web Services ➔ Actors (WS Harvester)



➔ "Minute-made" (MM) WS-based application integration

- Similarly: MM workflow design & sharing w/o implemented components

KEPLER: Actors

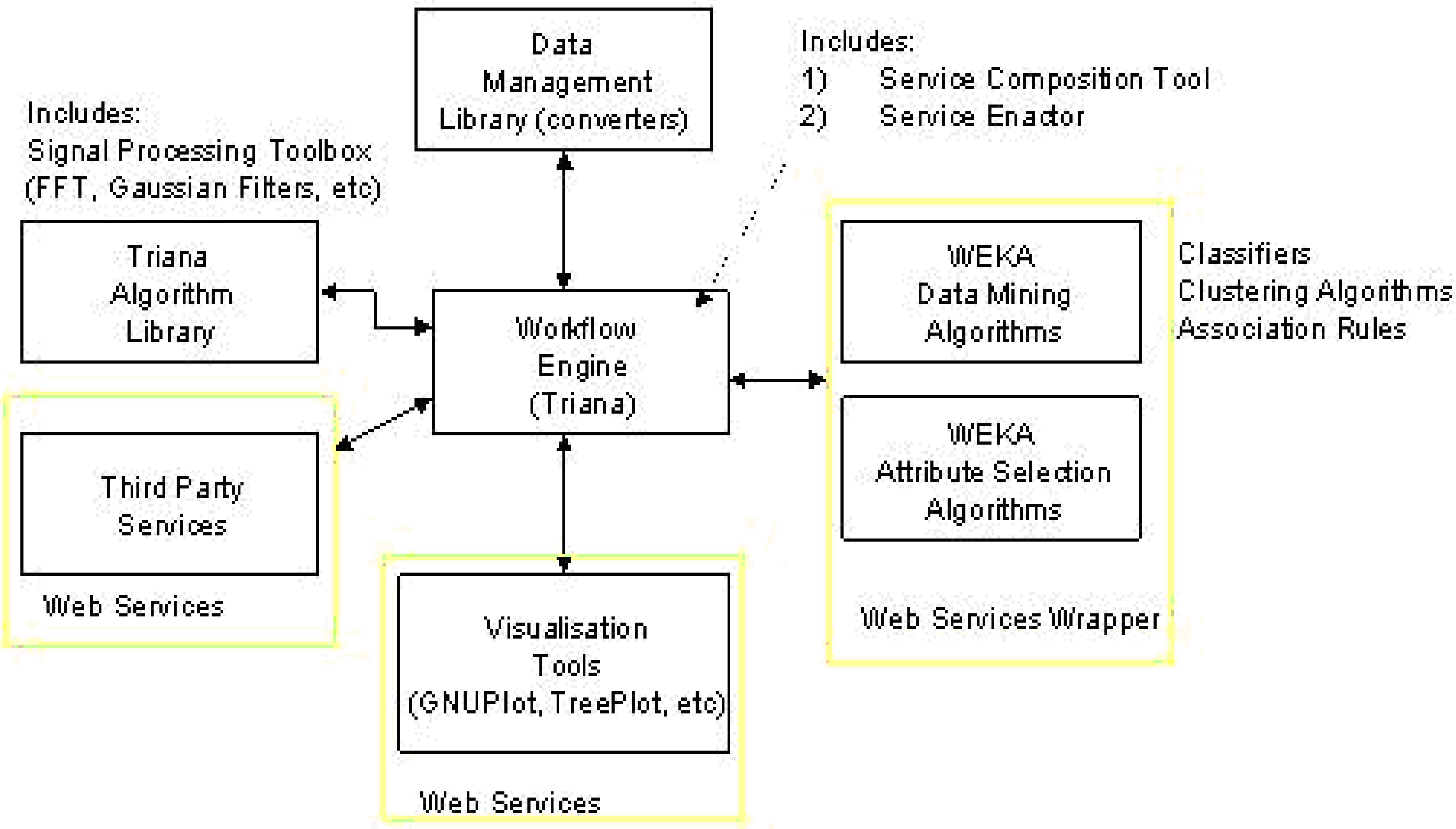


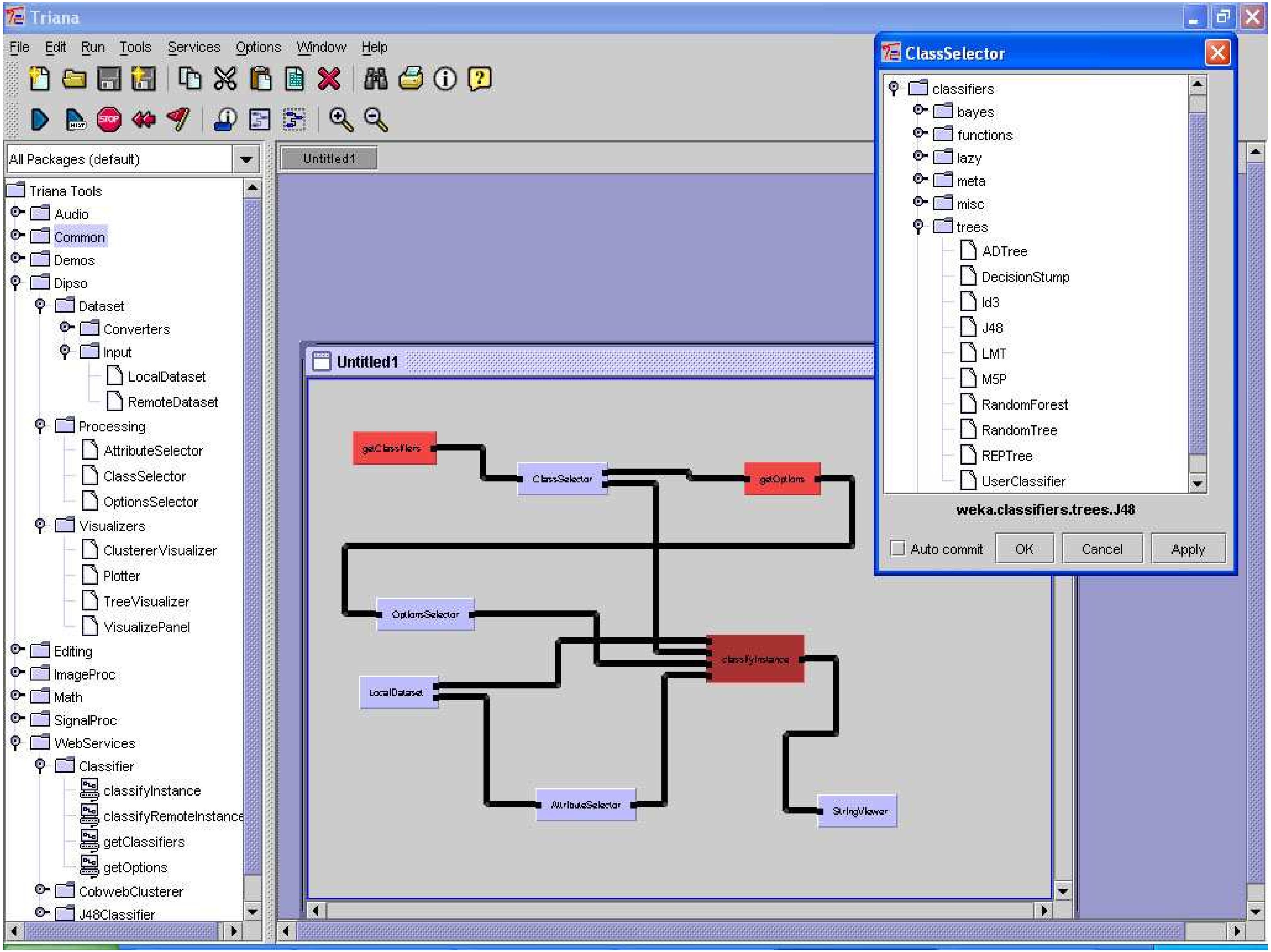
FAEHIM

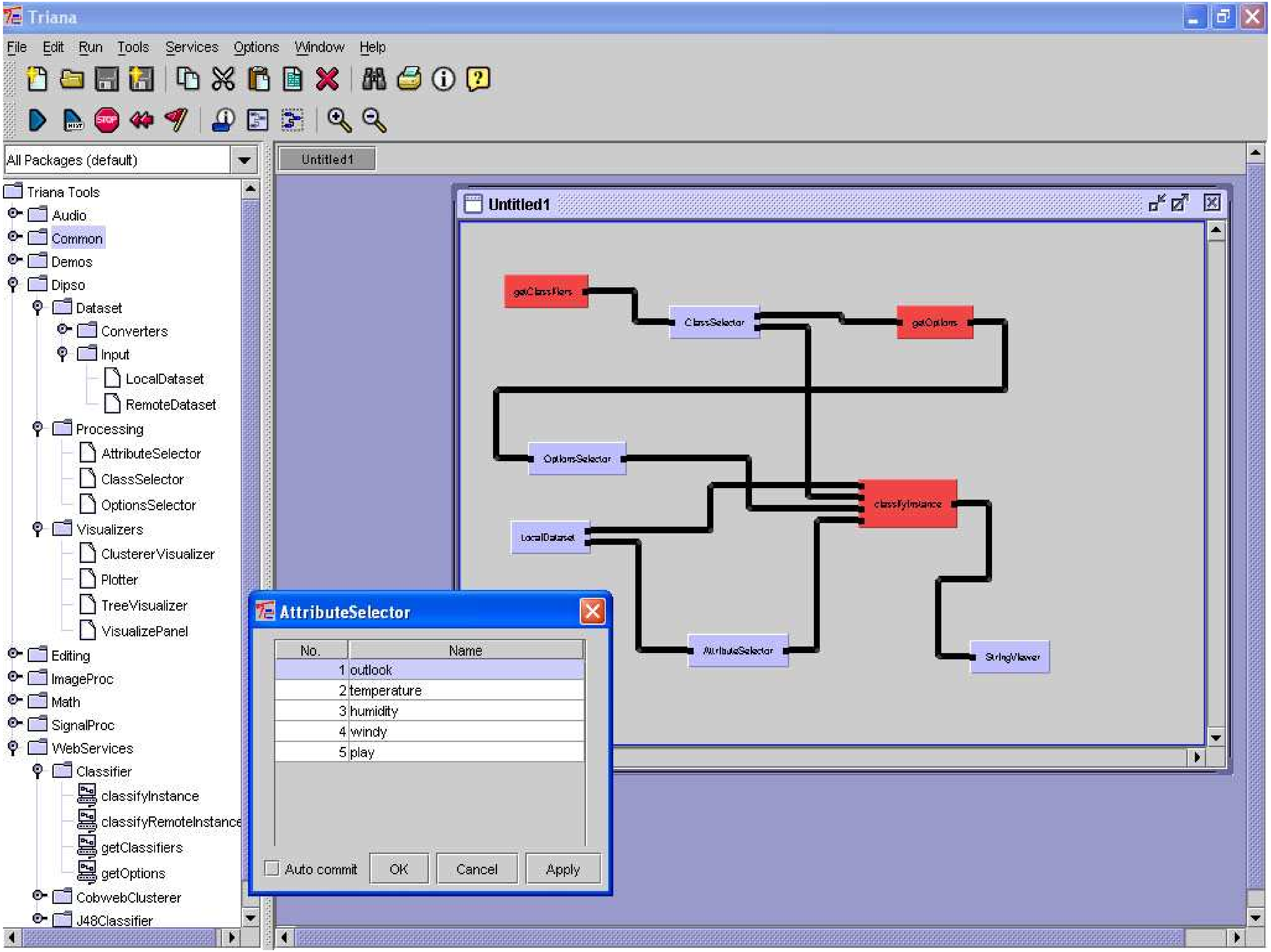
- Use of Web Services composition – with distributed services
  - Wrap third party services (Mathematica, GNUPlot)
  - WEKA Service template
  - Triana Workflow
- Services provided by third parties
  - WSDL interfaces (avoid use of specialist languages – unless really necessary)
  - SOAP-based message exchange
    - Use of attachments
- Access to local and remote data sets
  - Support for data streaming
- Wrapping of existing algorithms (important requirement)

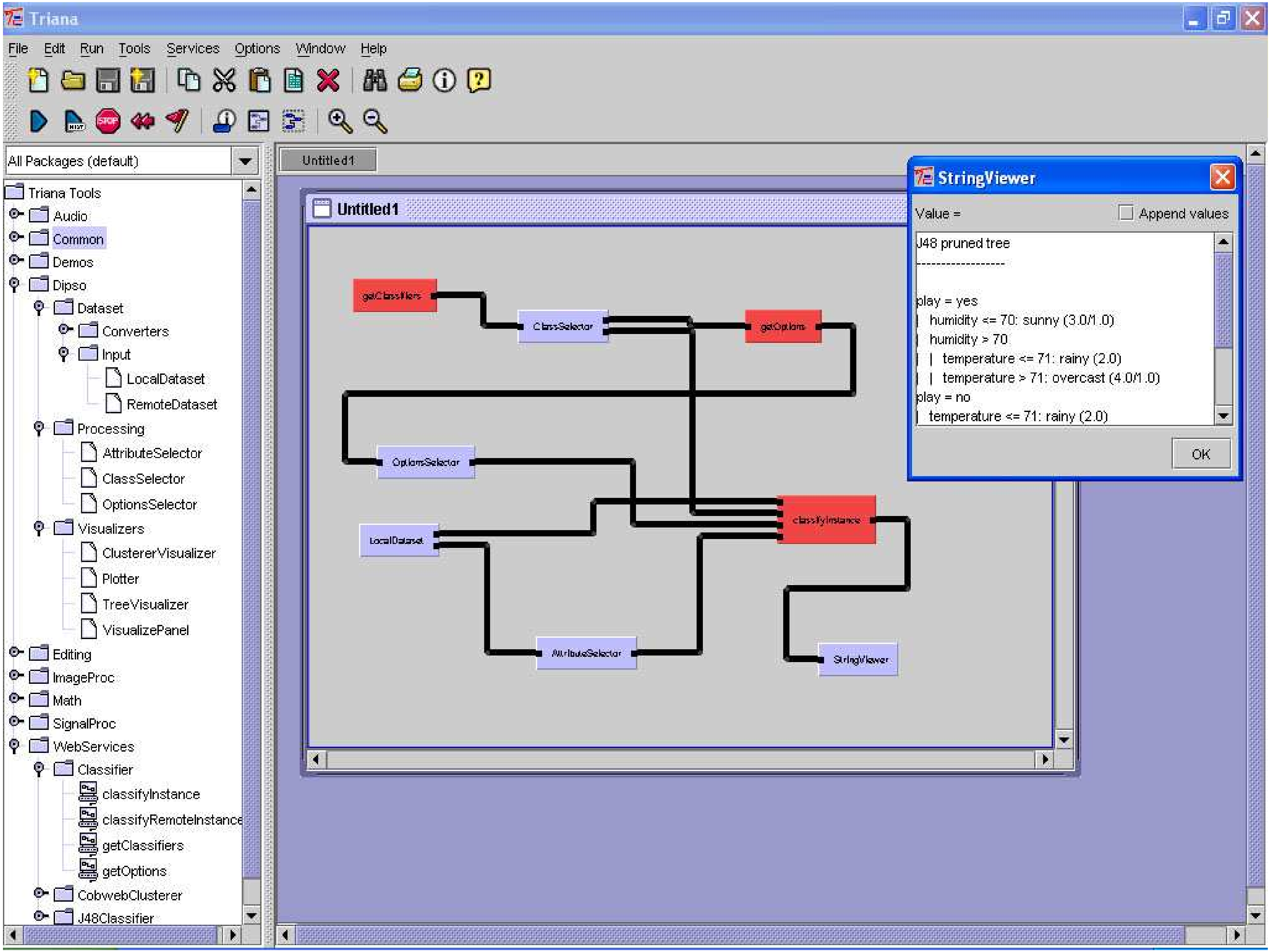
http://users.cs.cf.ac.uk/Ali.Shaikhali/faehim/

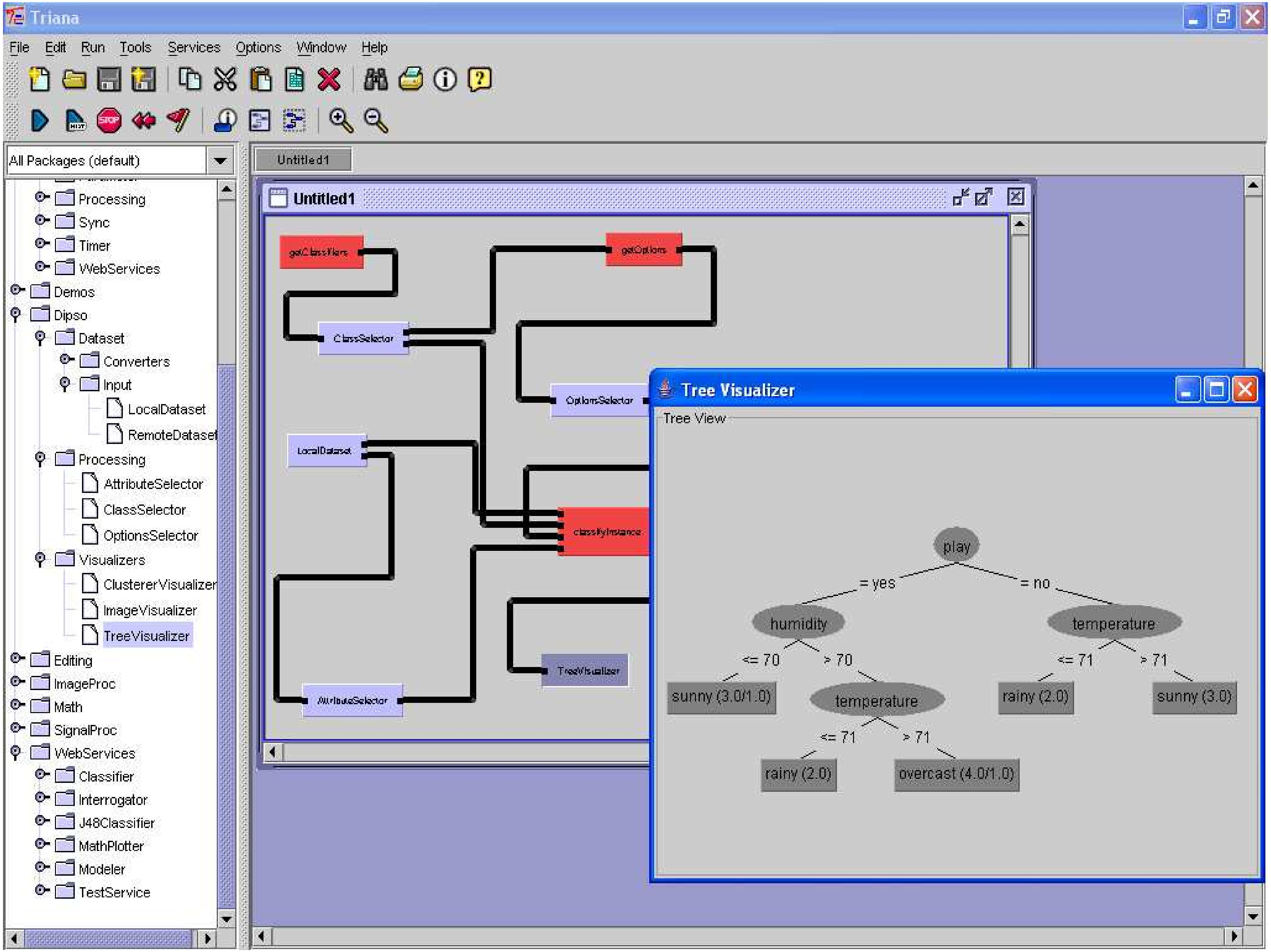
FAEHIM Architecture

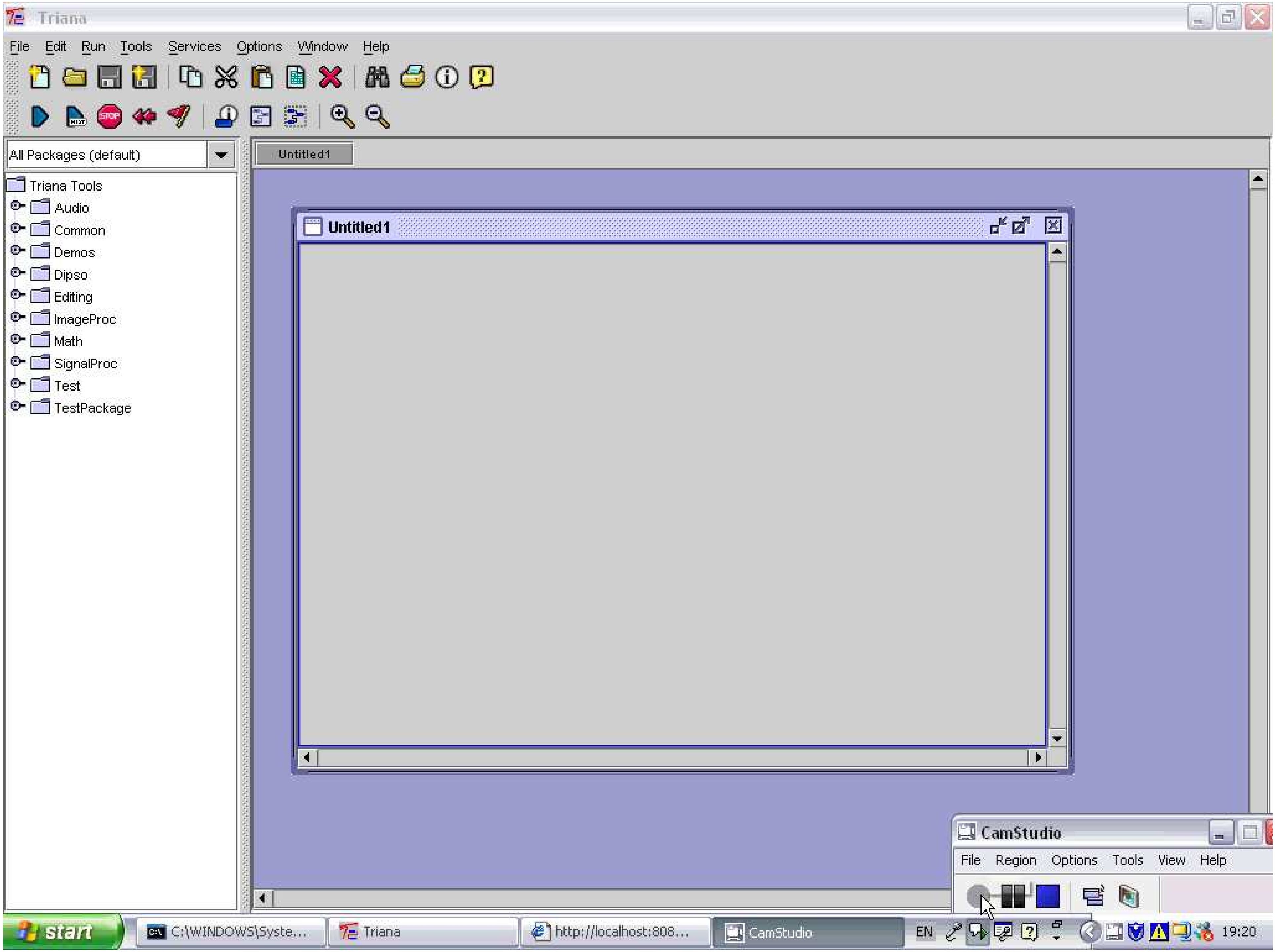


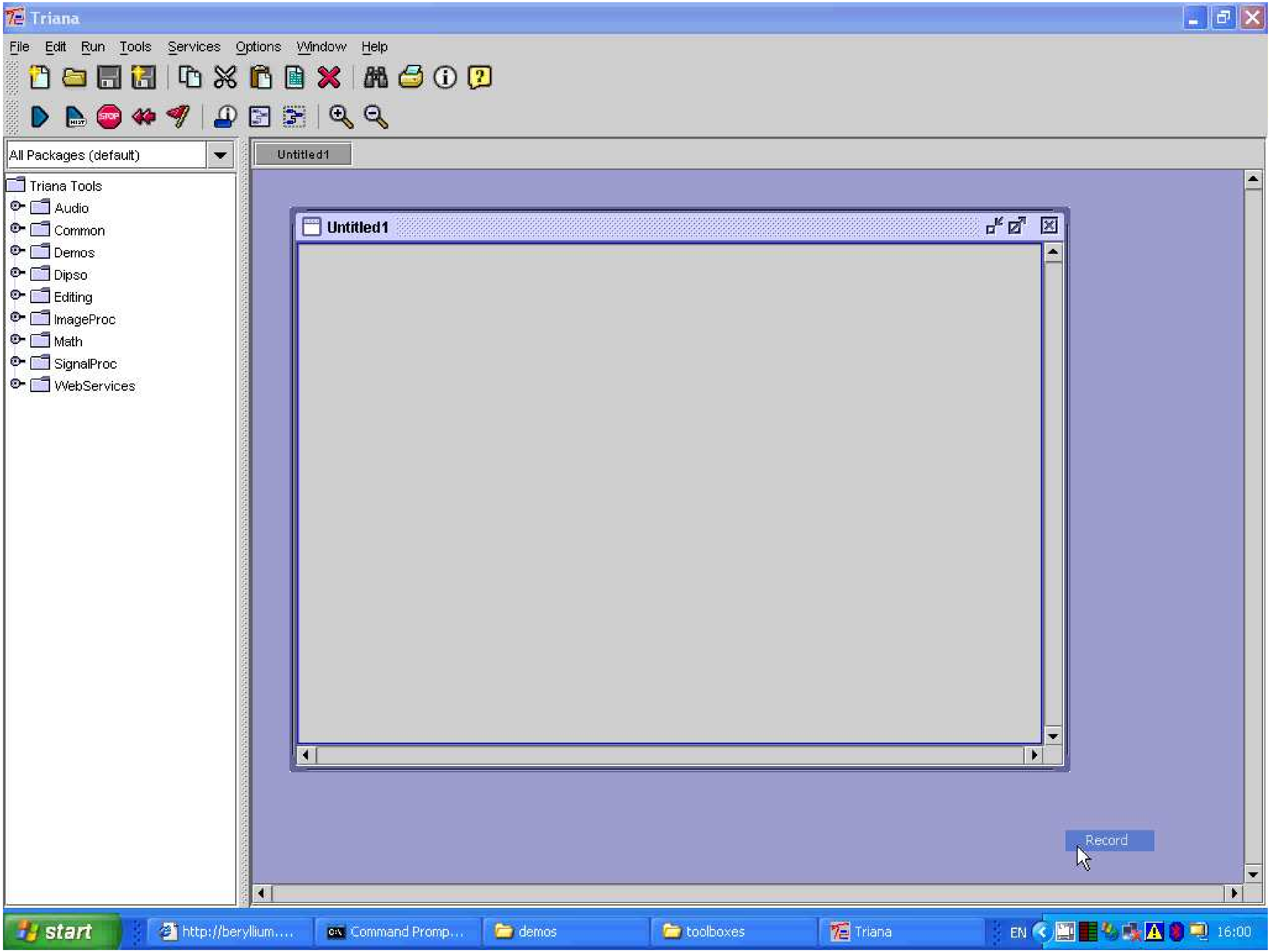

Triana
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Audio
Common
Demos
Dipso
Dataset
Converters
Input
LocalDataset
RemoteDataset
Processing
AttributeSelector
ClassSelector
OptionsSelector
Visualizers
ClustererVisualizer
Plotter
TreeVisualizer
VisualizePanel
Editing
ImageProc
Math
SignalProc
WebServices
Classifier
classifyInstance
classifyRemoteInstance
getClassifiers
getOptions
CobwebClusterer
J48Classifier
Untitled1
Untitled1
ClassSelector
classifiers
bayes
functions
lazy
meta
misc
trees
ADTree
DecisionStump
Id3
J48
LMT
M5P
RandomForest
RandomTree
REPTree
UserClassifier
weka.classifiers.trees.J48
Auto commit
OK
Cancel
Apply


Triana
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Audio
Common
Demos
Dipso
Dataset
Converters
Input
LocalDataset
RemoteDataset
Processing
AttributeSelector
ClassSelector
OptionsSelector
Visualizers
ClustererVisualizer
Plotter
TreeVisualizer
VisualizePanel
Editing
ImageProc
Math
SignalProc
WebServices
Classifier
classifyInstance
classifyRemoteInstance
getClassifiers
getOptions
CobwebClusterer
J48Classifier
Untitled1
Untitled1
AttributeSelector
No. Name
1 outlook
2 temperature
3 humidity
4 windy
5 play
Auto commit
OK
Cancel
Apply

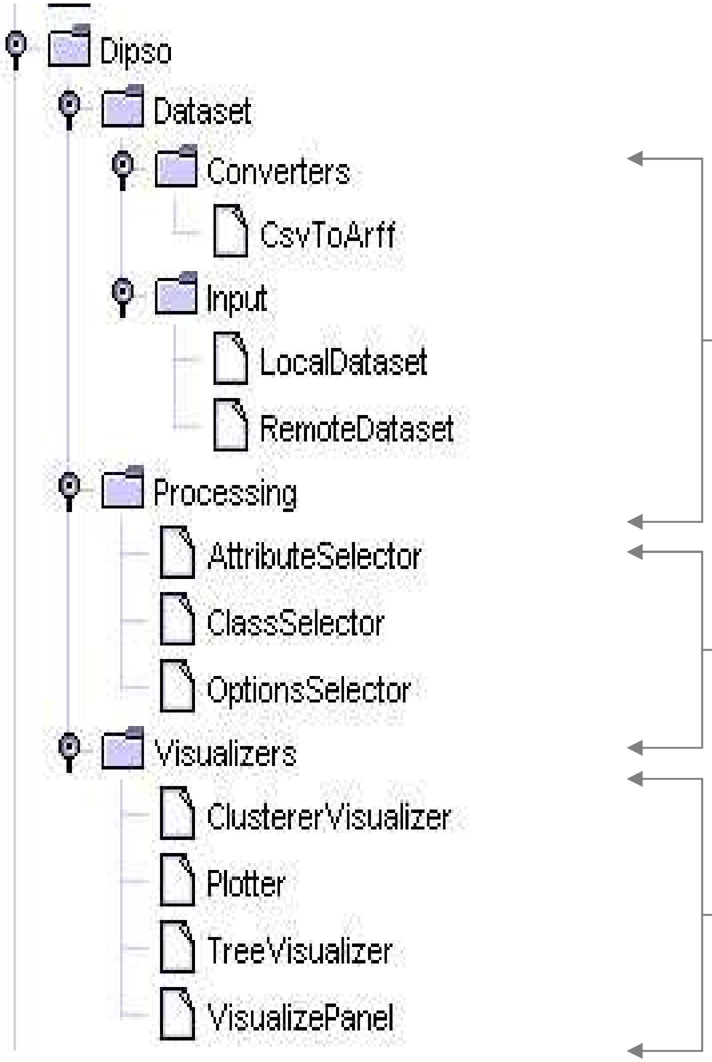

Triana
File Edit Run Tools Services Options Window Help
All Packages (default)
Triana Tools
Audio
Common
Demos
Dipso
Dataset
Converters
Input
LocalDataset
RemoteDataset
Processing
AttributeSelector
ClassSelector
OptionsSelector
Visualizers
ClustererVisualizer
Plotter
TreeVisualizer
VisualizePanel
Editing
ImageProc
Math
SignalProc
WebServices
Classifier
classifyInstance
classifyRemoteInstance
getClassifiers
getOptions
CobwebClusterer
J48Classifier
Untitled1
Untitled1
getClassifiers
ClassSelector
getOptions
OptionsSelector
classifyInstance
LocalDataset
AttributeSelector
StringViewer
StringViewer
Value =
Append values
J48 pruned tree
----
play = yes
| humidity <= 70: sunny (3.0/1.0)
| humidity > 70
| | temperature <= 71: rainy (2.0)
| | temperature > 71: overcast (4.0/1.0)
play = no
| temperature <= 71: rainy (2.0)
OK


Triana
File Edit Run Tools Services Options Window Help
All Packages (default)
Processing
Sync
Timer
WebServices
Demos
Dipso
Dataset
Converters
Input
LocalDataset
RemoteDataset
Processing
AttributeSelector
ClassSelector
OptionsSelector
Visualizers
ClustererVisualizer
ImageVisualizer
TreeVisualizer
Editing
ImageProc
Math
SignalProc
WebServices
Classifier
Interrogator
J48Classifier
MathPlotter
Modeler
TestService
Untitled1
getClassifiers
getOptions
ClassSelector
OptionsSelector
LocalDataset
classifyInstance
TreeVisualizer
AttributeSelector
Tree Visualizer
Tree View
play
= yes
= no
humidity
temperature
<= 70
> 70
<= 71
> 71
sunny (3.0/1.0)
temperature
rainy (2.0)
sunny (3.0)
<= 71
> 71
rainy (2.0)
overcast (4.0/1.0)


Triana
File Edit Run Tools Services Options Window Help
All Packages (default)
Untitled1
Triana Tools
Audio
Common
Demos
Dipso
Editing
ImageProc
Math
SignalProc
Test
TestPackage
Untitled1
CamStudio
File Region Options Tools View Help
start
C:\WINDOWS\Syste...
Triana
http://localhost:808...
CamStudio
EN
19:20


Triana
File Edit Run Tools Services Options Window Help
All Packages (default)
Untitled1
Triana Tools
Audio
Common
Demos
Dipso
Editing
ImageProc
Math
SignalProc
WebServices
Untitled1
Record
start
http://beryllium....
Command Promp...
demos
toolboxes
Triana
EN
16:00

Inside the FAEHIM Toolbox

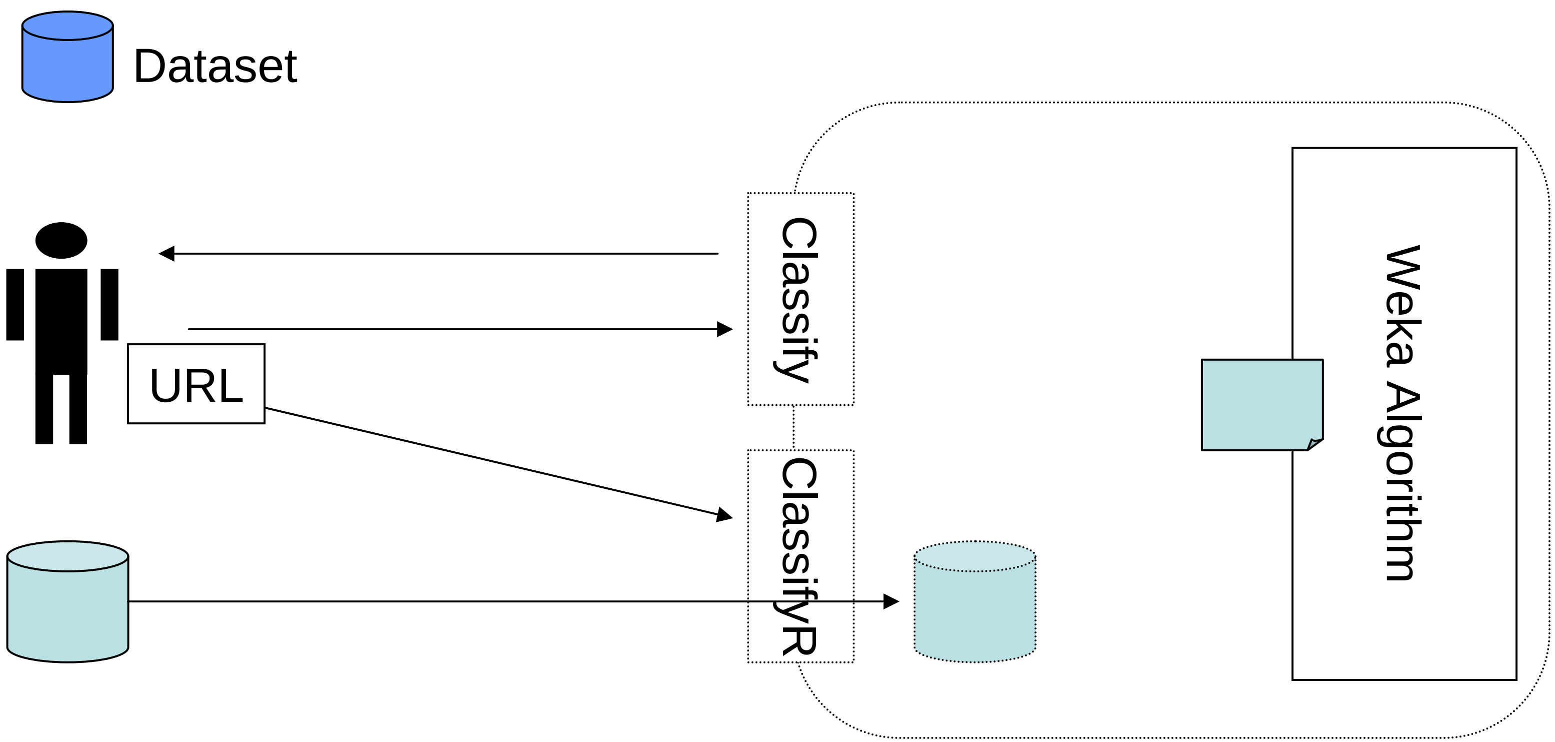


Deals with Dataset files
e.g. load datasets,
converts dataset formats

Deals with data that need manipulation

Visualize data

Usage Overview



Classifier

- **J48 Classifier**

  Class for generating an (un)pruned C4.5 decision tree. For more information, see *Ross Quinlan (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA.*

- **Operations**
  classify( )
  Input: DataHandler *dataset*, String *attributeName*
  output: DataHandler *decisionTree*

- **classifyRemoteDataset( )**
  Input: String *url*, String *attributeName*
  output: DataHandler *decisionTree*

Clustering Support

Cobweb Web Service

Operations

cluster( )
Input: DataHandler *dataset*
output: String *result*

clusterRemoteInstance( )
Input: String datasetURL
output: String *result*

clusterByPercentage( )
Input: DataHandler *dataset*, int *percentage*
output: String *result*

Graph Plotting Service

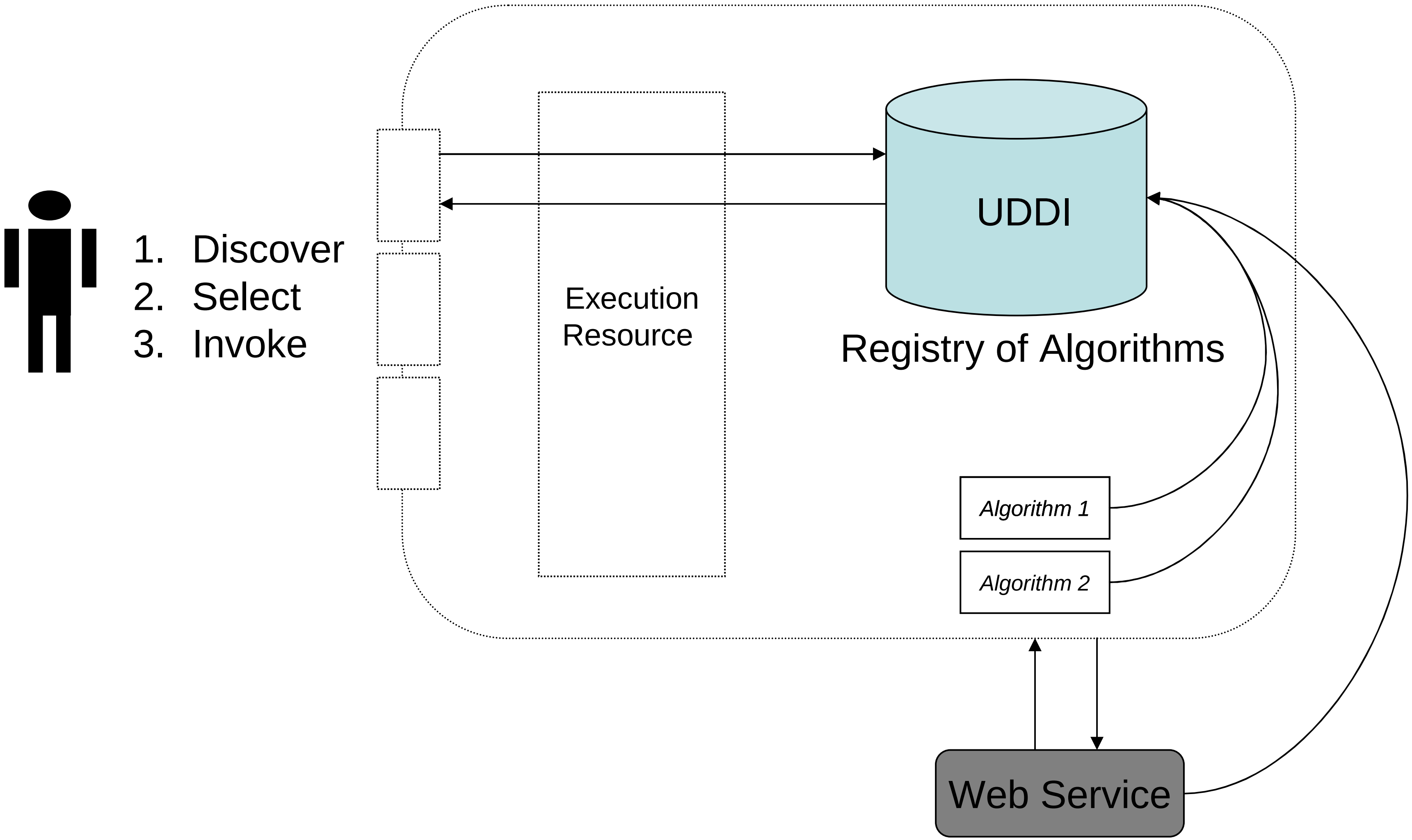
Plotting Web Service

Operations

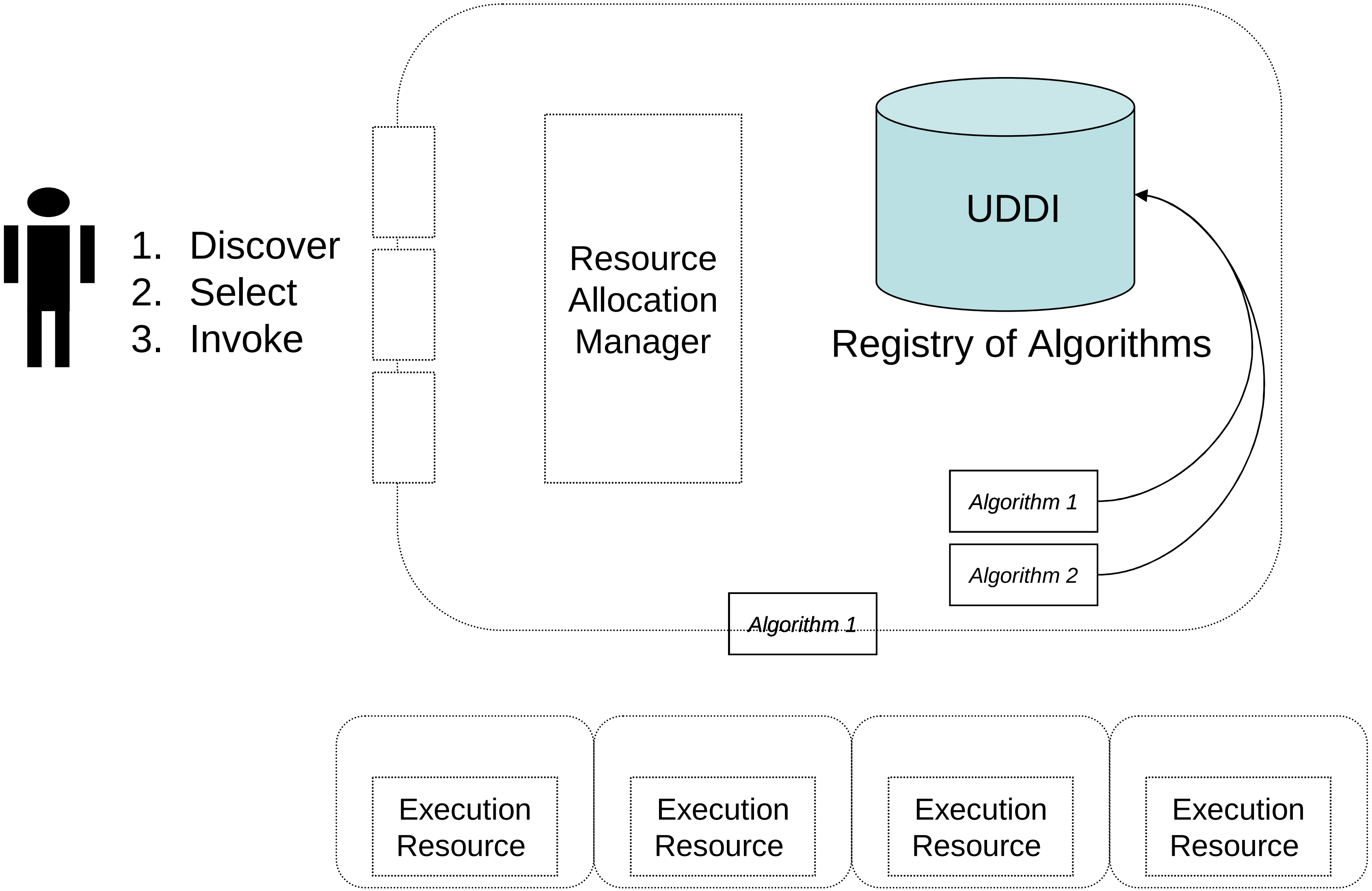
plot3D( )
Input: DataHandler *data*, String *plotType*
output: DataHandler *graph*

getPlotTypes( )
Input: *null*
output: String *plotTypes*

Registry Usage



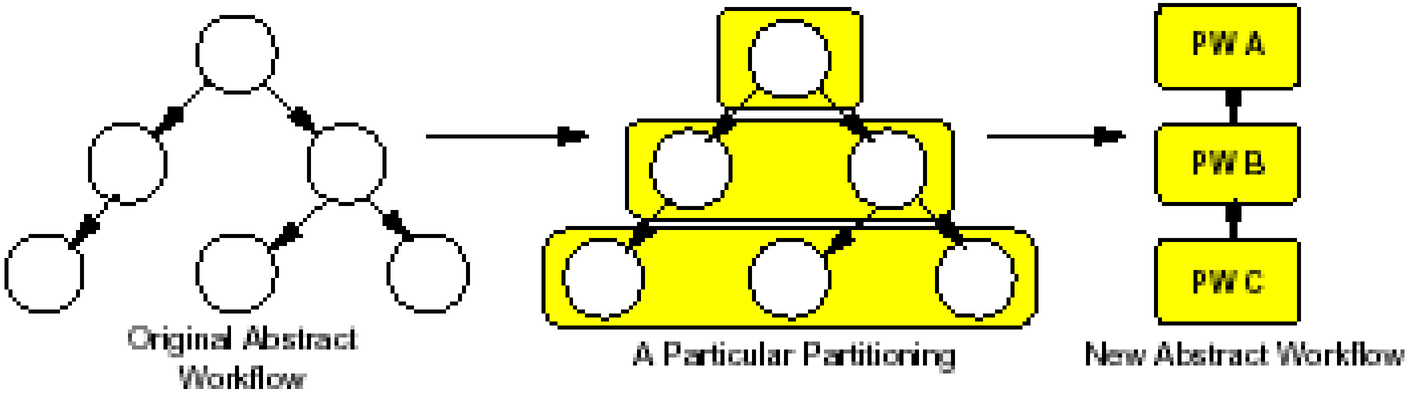
Parallel Execution



Workflow Optimisation

- Types of workflow optimisation
  - Through service selection
  - Through workflow re-ordering
  - Through exploitation of parallelism
- When is optimisation performed?
  - At design time (early binding)
  - Upon submission (intermediate binding)
  - At runtime (late binding)

Workflow Partitioning (Pegasus)



Full Graph vs Partial Graph Scheduling

Schedule

- Total workflow Graph
- Sub-graph
- Each node

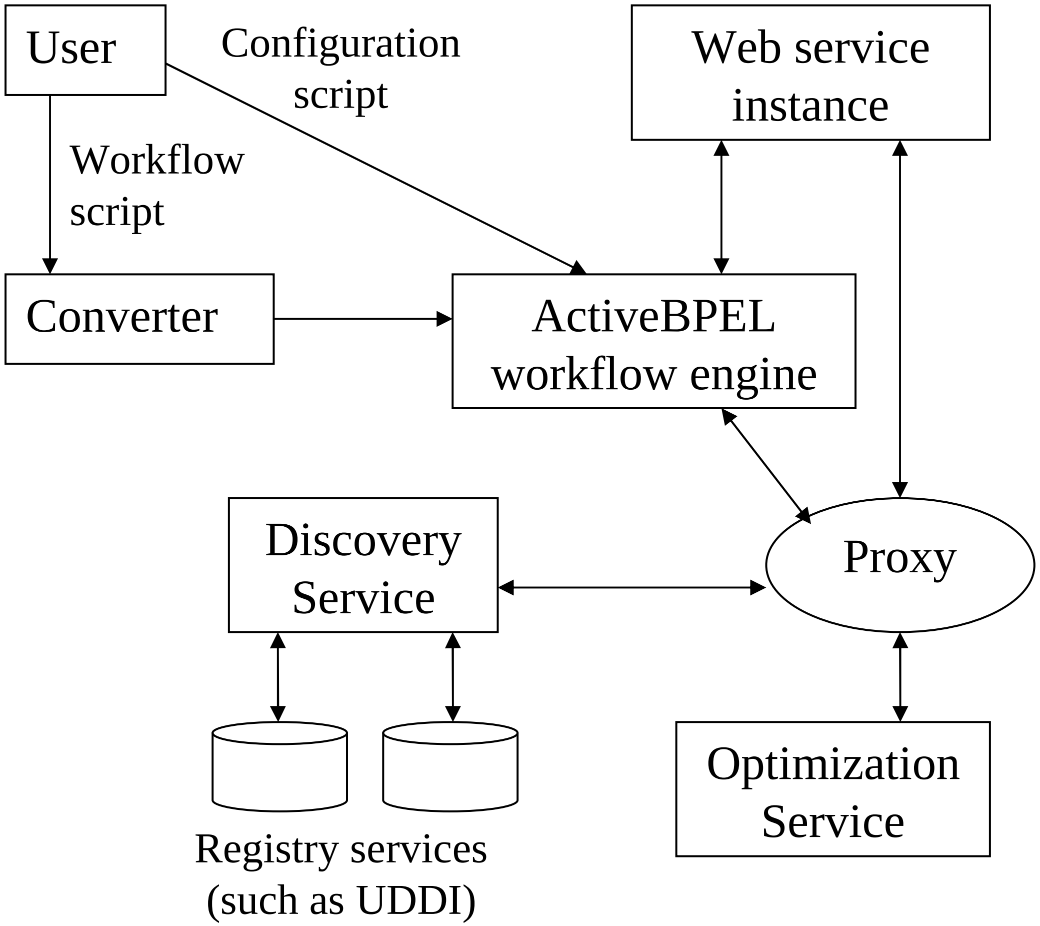
Service Binding Models

- Late binding of abstract service to concrete service instance means:
  - We use up-to-date information to decide which service to use when there are multiple semantically equivalent services
  - We are less likely to try to use a service that is unavailable.

Late Binding Case

- Search registry for all services that are consistent with abstract service description.
- Select optimal service based on current information, e.g, host load, etc.
- Execute this service.
- Doesn't take into account time to transfer inputs to the service.
- In early and late binding cases we can optimise overall workflow.

WOSE Architecture



Work at Cardiff has focused on implementing a late binding model for dynamic service discovery, based on a generic service proxy, and service discovery and optimisation services.

Service Discovery Issues

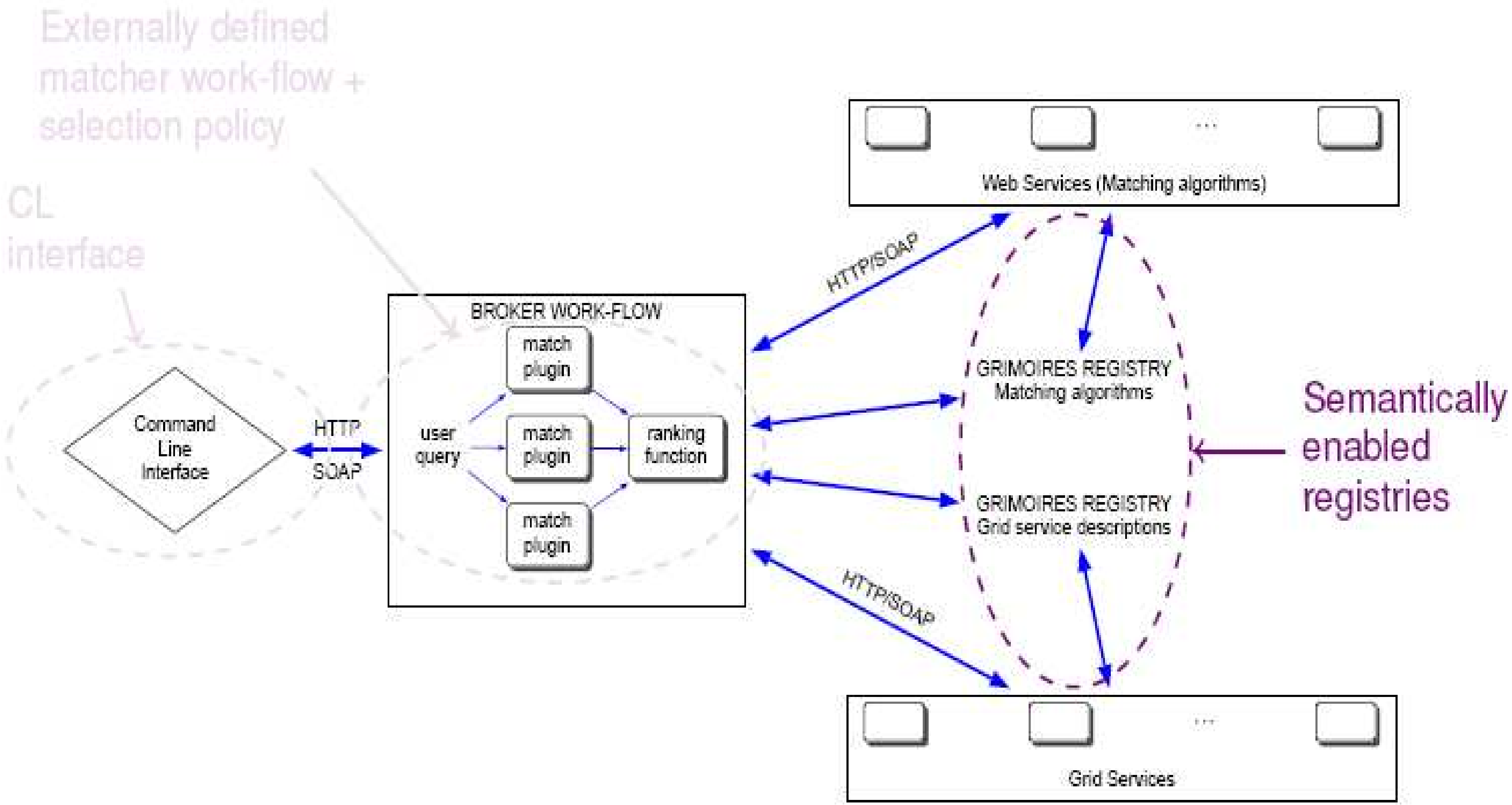
- Service discovery and optimisation is based on service metadata.
- Could store in a database.
- Could obtain by interrogating service.

Use of Registry Services

Specialization vs. late-binding of function creates three options:

- No fixed actions
- One fixed action: the matching service comes with
  - A set of registries or
  - A set of matchers or
  - A selection function
- Two fixed actions: the matching service comes with
  - A set of registries and a set of matchers or
  - A set of registries and a selection function or
  - A set of matchers and a selection function
- Three fixed actions: a set of registries and a set of matchers and a selection function

KNOOGLE



Externally defined matcher work-flow + selection policy
CL interface
Command Line Interface
HTTP
SOAP
BROKER WORK-FLOW
user query
match plugin
match plugin
match plugin
ranking function
HTTP/SOAP
HTTP/SOAP
Web Services (Matching algorithms)
GRIMOIRES REGISTRY
Matching algorithms
GRIMOIRES REGISTRY
Grid service descriptions
Grid Services
Semantically enabled registries

- User writes a `knoogle script`
- commands:
  - submit-query: submit a query to be processed by the broker
  - get-query-status: get the identities of the query jobs currently being run by the broker
  - get-IDs: get the String id of all the queries being currently processed by the broker
  - triple-store: get the status of a query
  - terminate-query: terminate a query
  - validate-broker: validate the given broker
  - broker-status: return the status of a broker
- Accessible also as an API – enabling embedding in applications.
- Keep the API simple.

Broker Configuration

```
<querySubmission>
<!– the query document –>
<query>
<repositories>
<!– the repositories to be used –>
<repository>* <!– repository URL –>
<matchers>?
<!– the matchers to be used –>
<matcher>* <!– match service URL –>
<!– the selection policy script to use –>
<selectionPolicy>?
<!– selection policy script language –>
<selectionPolicyLanguage>?
...
```

Taverna Configuration

- All inputs specified in knoogle.properties File

  `knoogle.wsdlBroker =`

  `http://alis.cs.bath.ac.uk:9050/axis/services/Broker`

- `knoogle.wsdlRepository =`

  `http://chost-comsc.grid.cf.ac.uk:8080/grimoires/services`

  `knoogle.wsdlMatcher =`

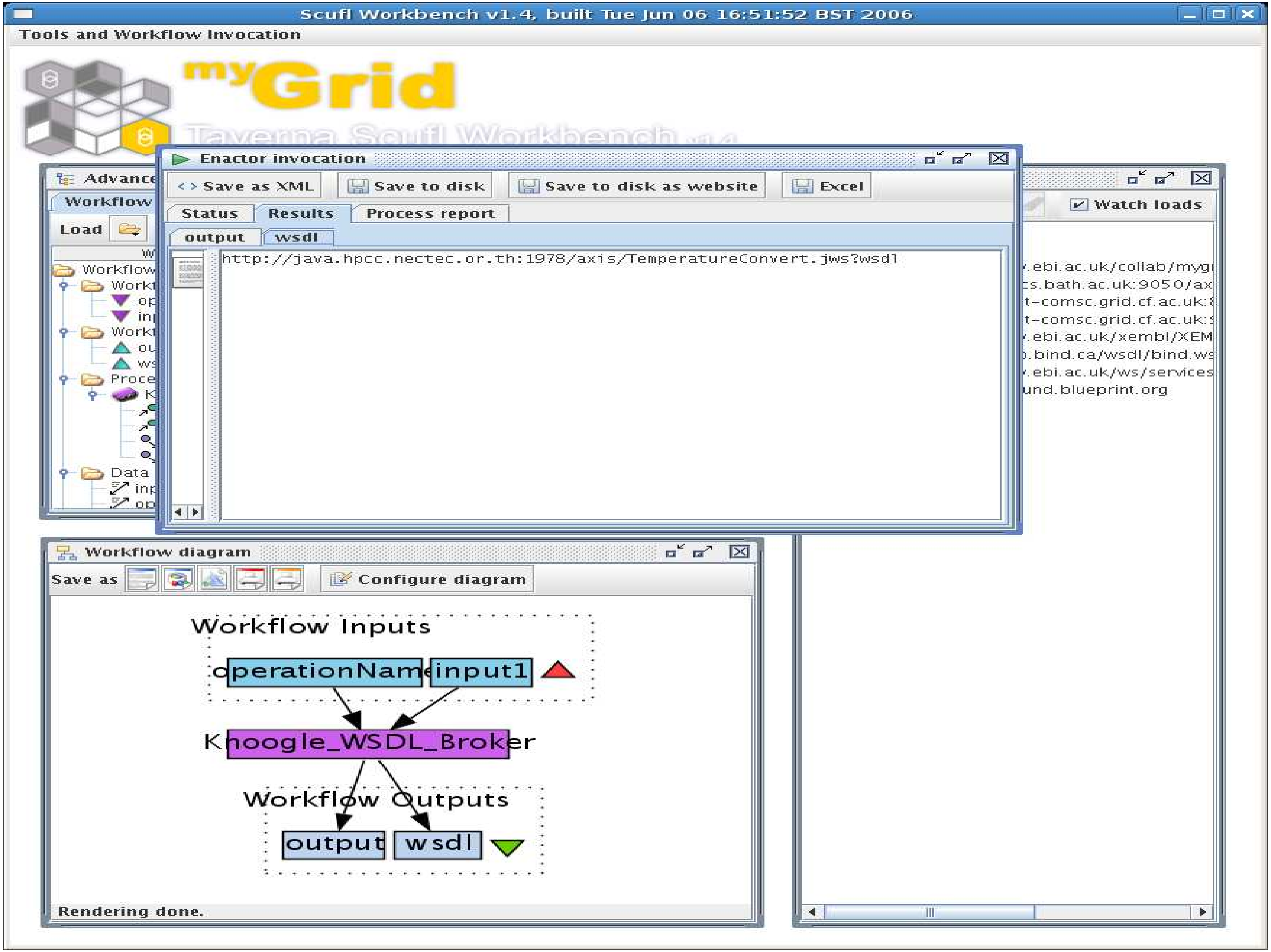
  `http://host1:9050/axis/services/basicMatcher,`

  `http://host2:9050/axis/services/gridSAMmatcher`

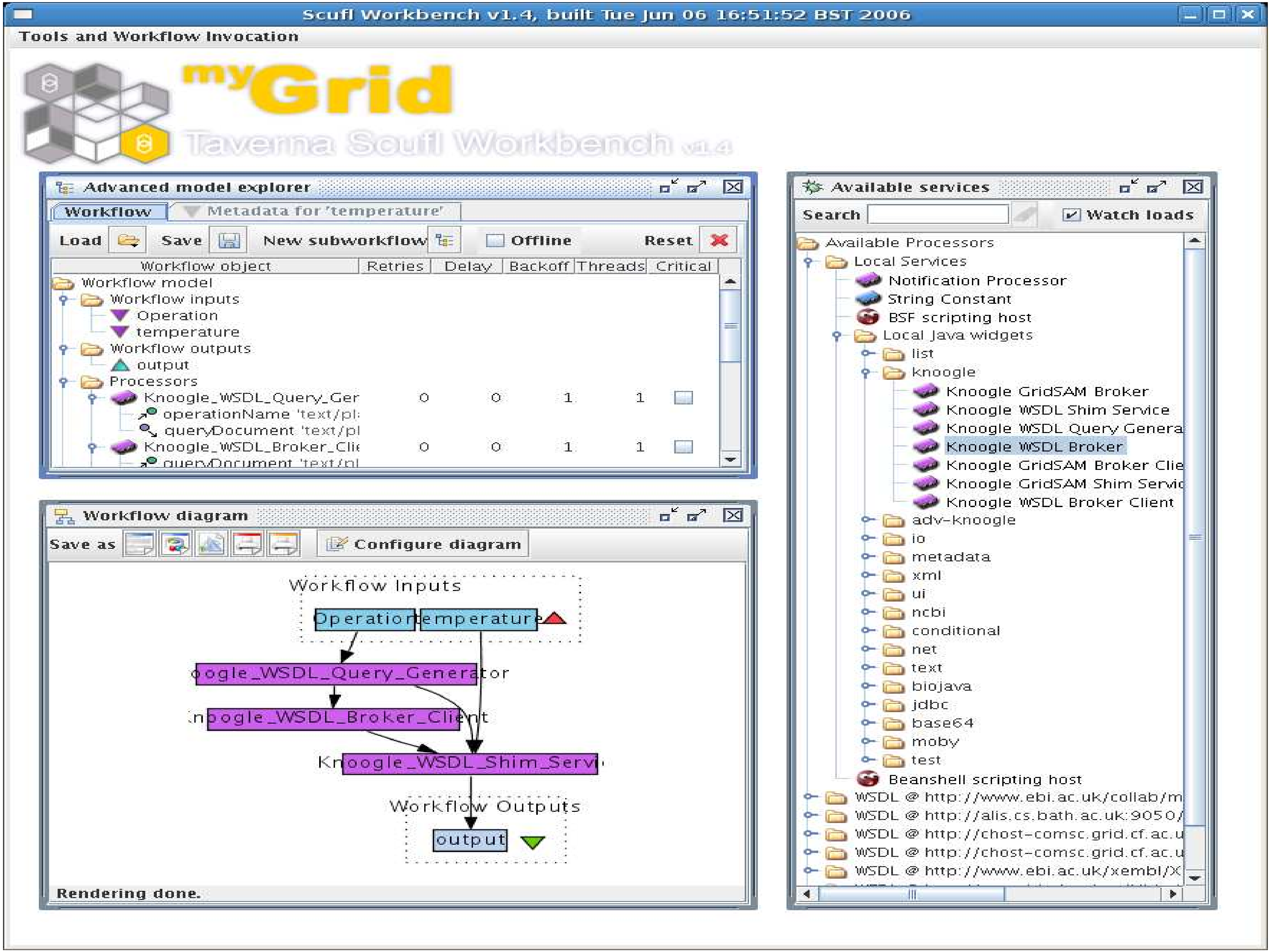
- `knoogle.wsdlSPScriptName = SeRQL`

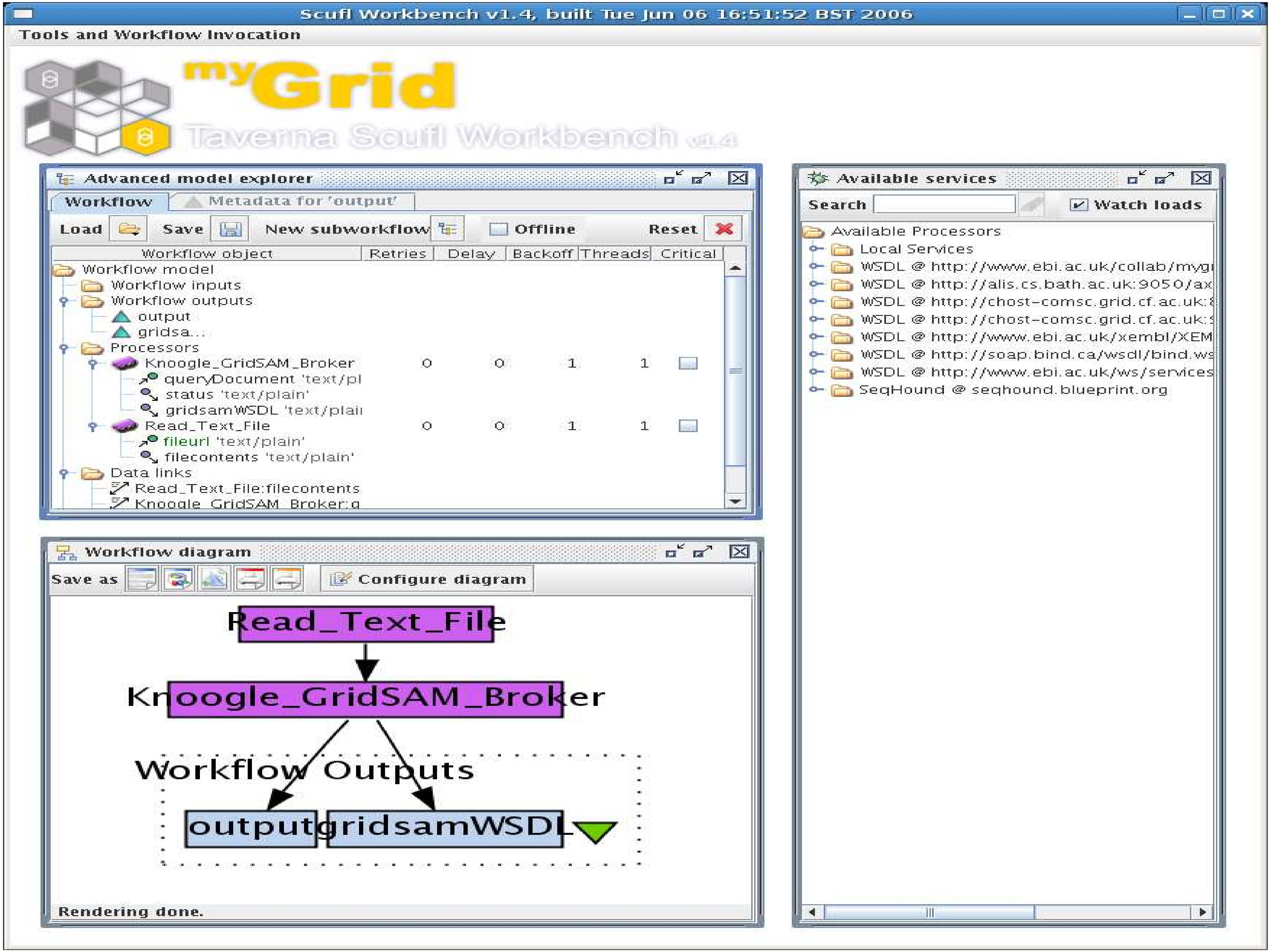
  `knoogle.wsdlSPScriptFile = SP1.serql`

- Provision for multiple Repositories and Matchers



Scufl Workbench v1.4, built Tue Jun 06 16:51:52 BST 2006
Tools and Workflow Invocation
myGrid
Taverna Scufl Workbench
Enactor invocation
Save as XML
Save to disk
Save to disk as website
Excel
Status
Results
Process report
output
wsdl
http://java.hpcc.nectec.or.th:1978/axis/TemperatureConvert.jws?wsdl
Watch loads
Workflow diagram
Save as
Configure diagram
Workflow Inputs
operationName
input1
Khoogle_WSDL_Broker
Workflow Outputs
output
wsdl
Rendering done.


Scufl Workbench v1.4, built Tue Jun 06 16:51:52 BST 2006
Tools and Workflow Invocation
myGrid
Taverna Scufl Workbench v1.4
Advanced model explorer
Workflow
Metadata for 'temperature'
Load
Save
New subworkflow
Offline
Reset
Workflow object
Retries
Delay
Backoff
Threads
Critical
Workflow model
Workflow inputs
Operation
temperature
Workflow outputs
output
Processors
Knoogle_WSDL_Query_Ger
operationName 'text/pl
queryDocument 'text/pl
Knoogle_WSDL_Broker_Clie
Workflow diagram
Save as
Configure diagram
Workflow Inputs
Operation
temperature
Knoogle_WSDL_Query_Generator
Knoogle_WSDL_Broker_Client
Knoogle_WSDL_Shim_Servi
Workflow Outputs
output
Rendering done.
Available services
Search
Watch loads
Available Processors
Local Services
Notification Processor
String Constant
BSF scripting host
Local Java widgets
list
knoogle
Knoogle GridSAM Broker
Knoogle WSDL Shim Service
Knoogle WSDL Query Genera
Knoogle WSDL Broker
Knoogle GridSAM Broker Clie
Knoogle GridSAM Shim Servic
Knoogle WSDL Broker Client
adv-knoogle
io
metadata
xml
ui
ncbi
conditional
net
text
biojava
jdbc
base64
moby
test
Beanshell scripting host
WSDL @ http://www.ebi.ac.uk/collab/m
WSDL @ http://alis.cs.bath.ac.uk:9050/
WSDL @ http://chost-comsc.grid.cf.ac.u
WSDL @ http://chost-comsc.grid.cf.ac.u
WSDL @ http://www.ebi.ac.uk/xembl/X


Scufl Workbench v1.4, built Tue Jun 06 16:51:52 BST 2006
Tools and Workflow Invocation
myGrid
Taverna Scufl Workbench v1.4
Advanced model explorer
Workflow
Metadata for 'output'
Load
Save
New subworkflow
Offline
Reset
Workflow object
Retries
Delay
Backoff
Threads
Critical
Workflow model
Workflow inputs
Workflow outputs
output
gridsa...
Processors
Knoogle_GridSAM_Broker
queryDocument 'text/pl
status 'text/plain'
gridsamWSDL 'text/plai
Read_Text_File
fileurl 'text/plain'
filecontents 'text/plain'
Data links
Read_Text_File:filecontents
Available services
Search
Watch loads
Available Processors
Local Services
WSDL @ http://www.ebi.ac.uk/collab/mygi
WSDL @ http://alis.cs.bath.ac.uk:9050/ax
WSDL @ http://www.ebi.ac.uk/xembl/XEM
WSDL @ http://soap.bind.ca/wsdl/bind.ws
WSDL @ http://www.ebi.ac.uk/ws/services
SeqHound @ seqhound.blueprint.org
Workflow diagram
Save as
Configure diagram
Read_Text_File
Knoogle_GridSAM_Broker
Workflow Outputs
output
gridsamWSDL
Rendering done.

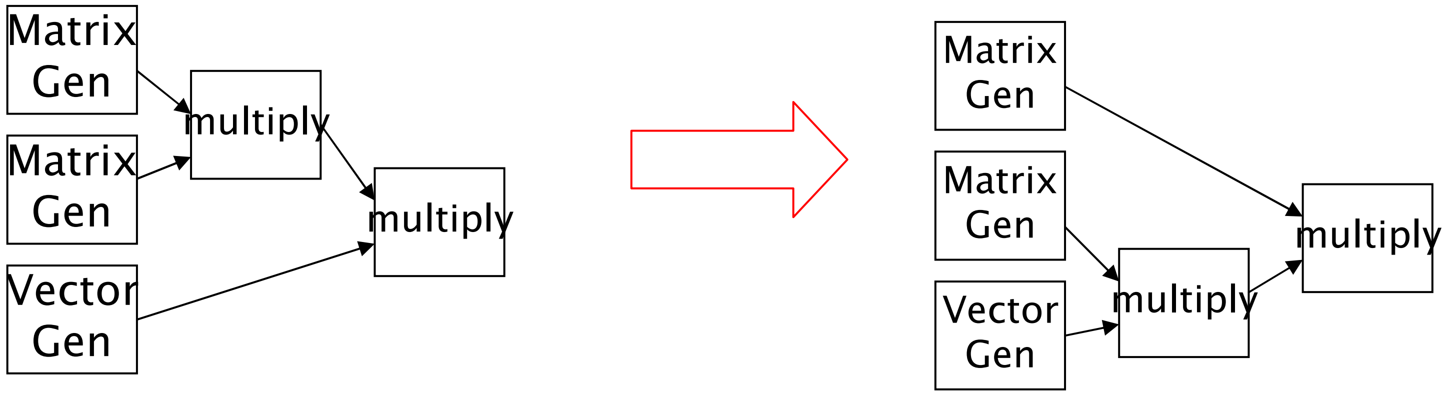
Optimisation by Re-Ordering

- Optimise the runtime execution of workflow before it is executed
- Achieves the goal through:
  - Re-ordering of components
  - Addition of components
  - Substitution of components
  - Pruning of the workflow

- Performance and workflow aware Scheduling
- Runtime Optimisation
  - through monitoring, check-pointing and migration

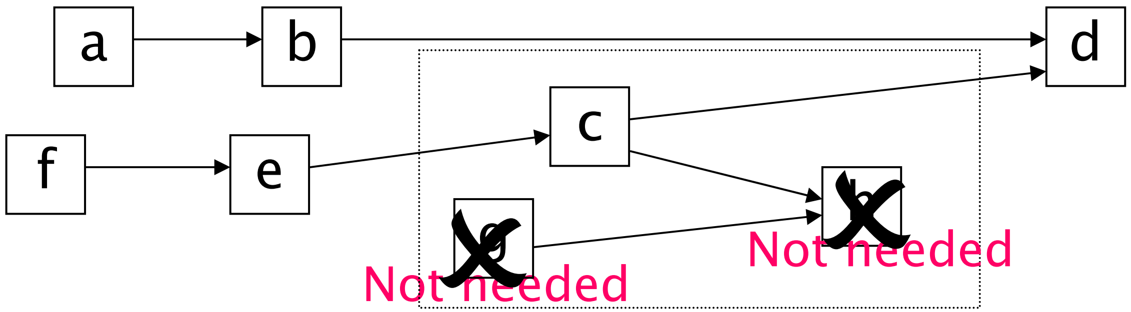
Component Manipulation

- Re-ordering: Workflows (often composed from composite workflows) may contain non-optimal ordering of components
  - Use re-ordering to improve performance

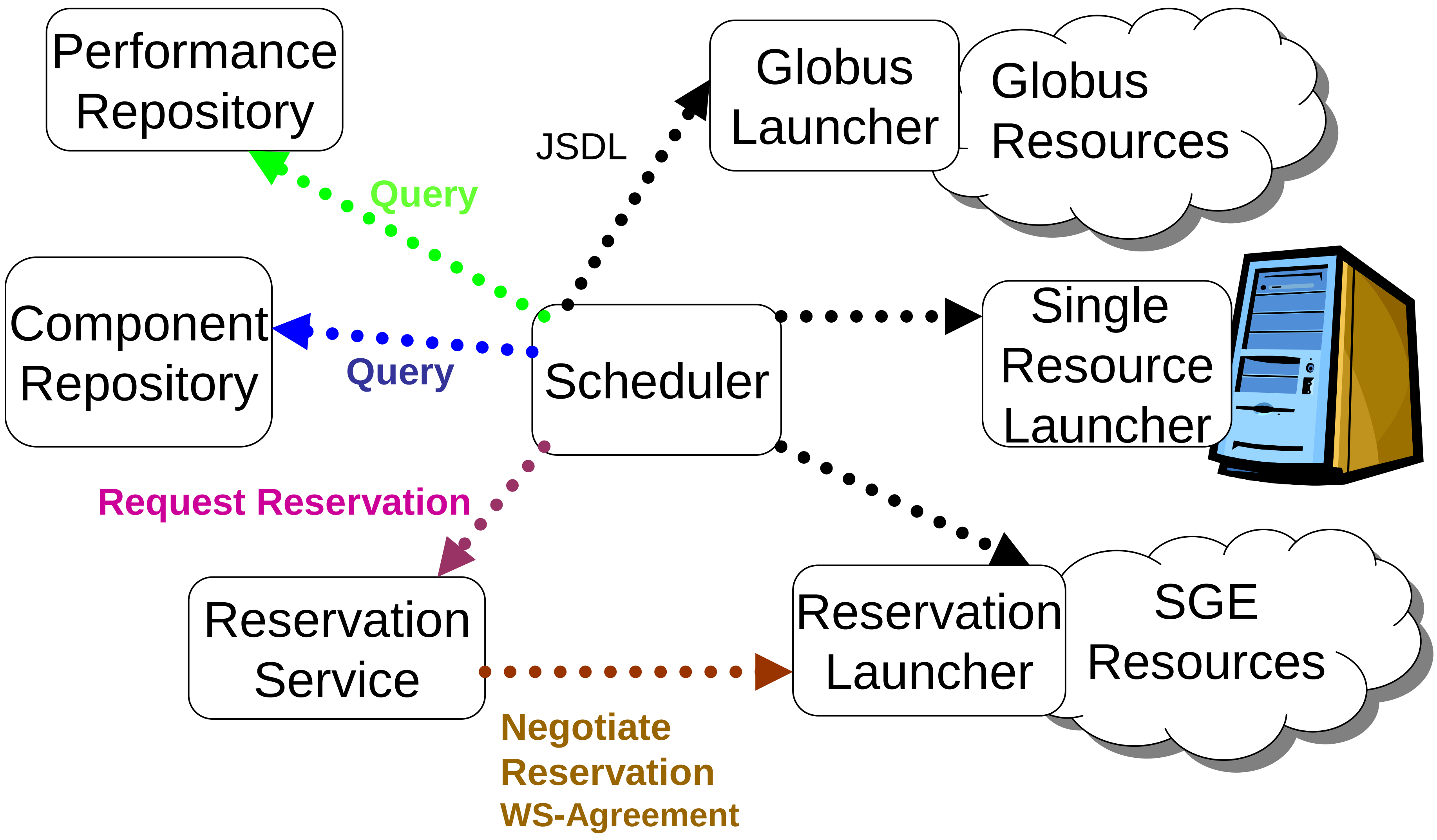


Pruning

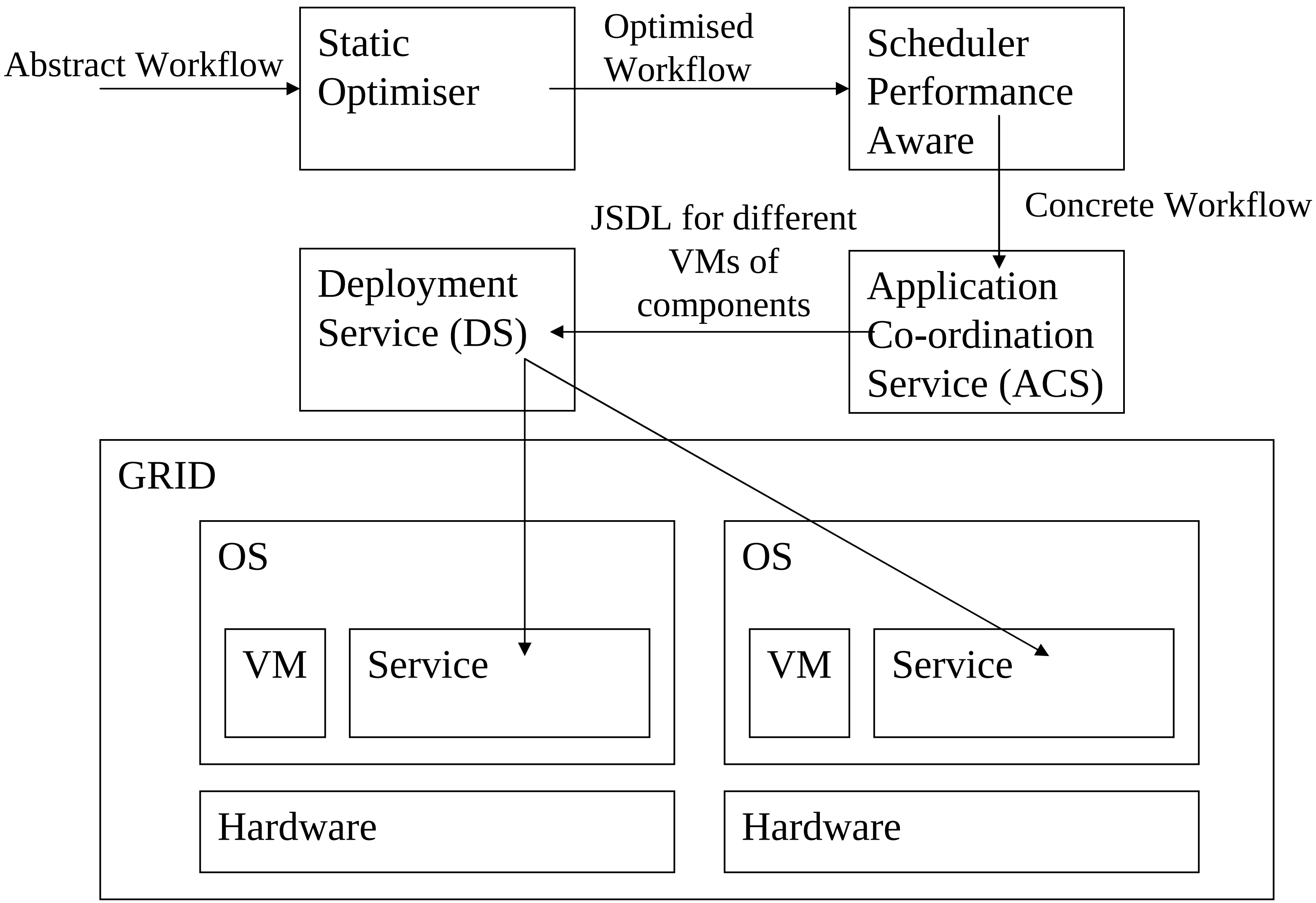
- Workflow Pruning:
  - Workflows may contain unused components. Especially when composed from other sub-workflows
- Remove redundant components



Performance Aware Scheduling



Execution Pipeline



Workflow Patterns

- Identify and reuse common "idioms" in some scientific domain and across different scientific domains.
- An "idiom" captures common knowledge and experience and describe how a similar set of experiments are to be set-up and managed.

From: Cecilia Gomes

Usage

1. To allow computational scientists and developers to capture *design patterns* that express common usage of software infrastructure within scientific domains
2. To provide a software engineering tool that supports:
   - application configuration,
   - execution control, and
   - reconfiguration of software services

From: Cecilia Gomes

Approach

- Patterns are divided in two categories for flexibility:
  - Co-ordination (Behavioural) patterns
    - Capture interactions between software sub-systems
  - Structural patterns
    - Capture connectivity between particular types of Grid software/hardware components

From: Cecilia Gomes

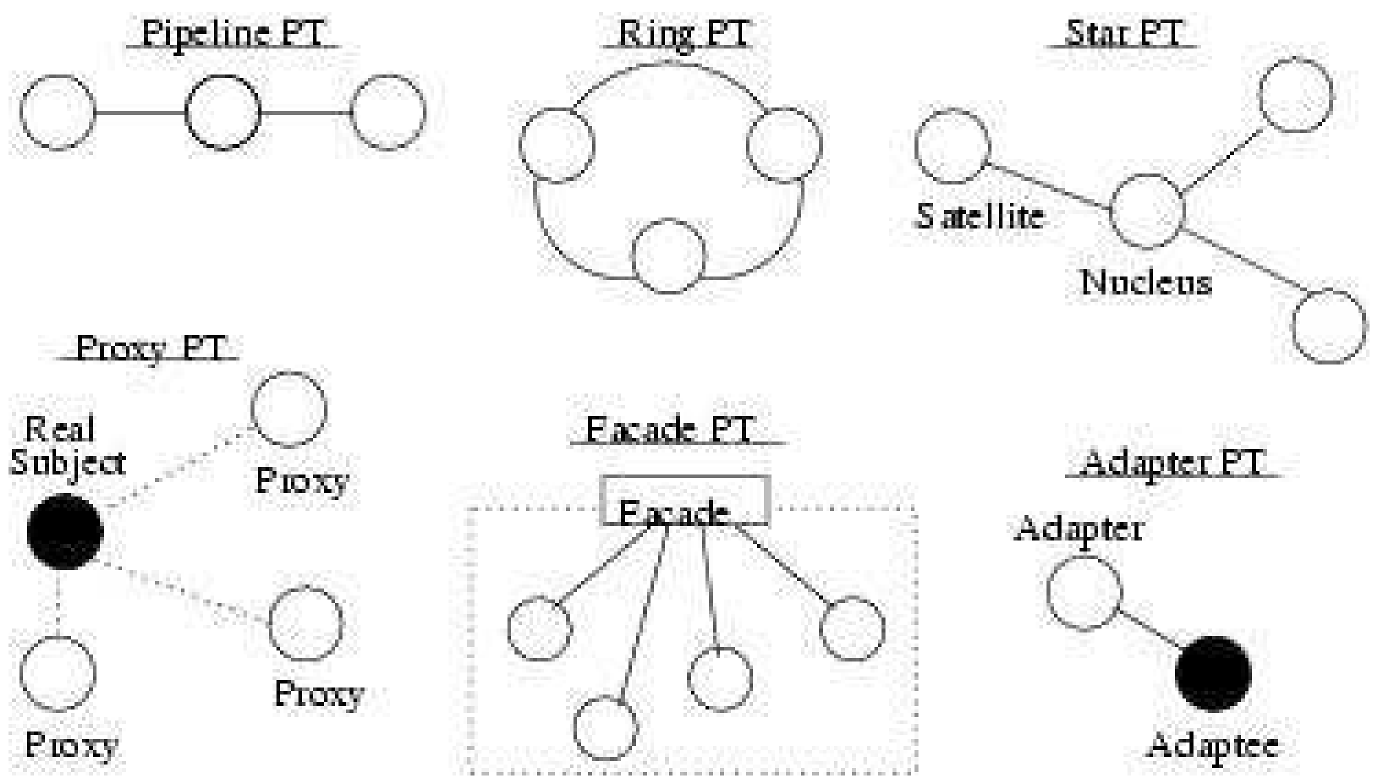
Approach

- Patterns as first class entities both at design, execution, and reconfiguration times
- Pattern templates are manipulated through Pattern Operators:
  - Structural operators
  - Behavioural operators

From: Cecilia Gomes

Structural Pattern Templates

- Encode component connectivity. Ex: Pipeline, Ring, Star, Façade, Adapter, Proxy.



From: Cecilia Gomes

Structural Operators

- Manipulate structural patterns keeping their structural constraints.
- Examples:
  - Increase, Decrease,
  - Extend, Reduce,
  - Embed, Extract,
  - Group,
  - Rename/Reshape, …

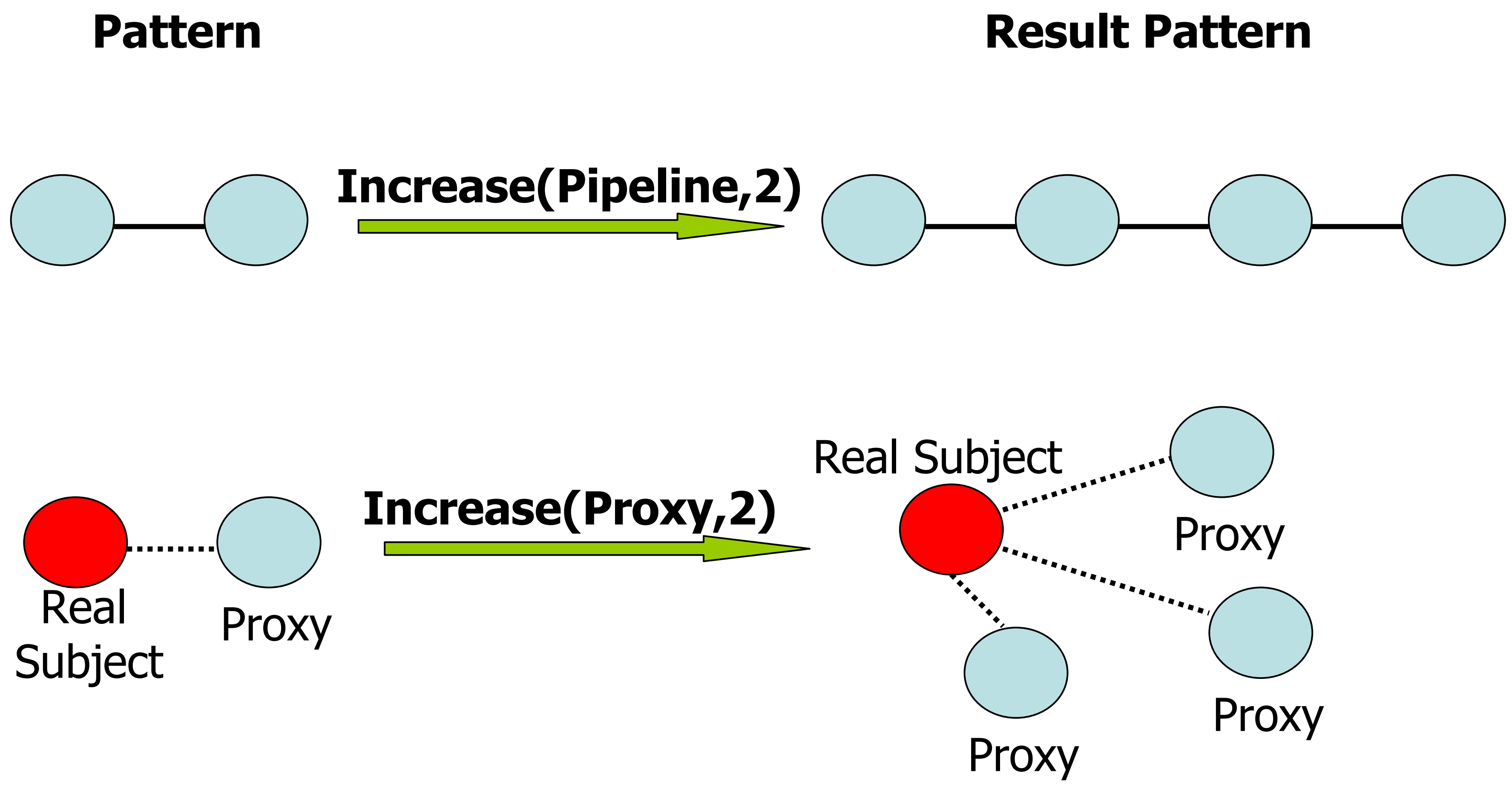
From: Cecilia Gomes

Structural Operators

- Manipulate structural patterns keeping their structural constraints.
- Examples:
  - Increase, Decrease,
  - Extend, Reduce,
  - Embed, Extract,
  - Group,
  - Rename/Reshape, …

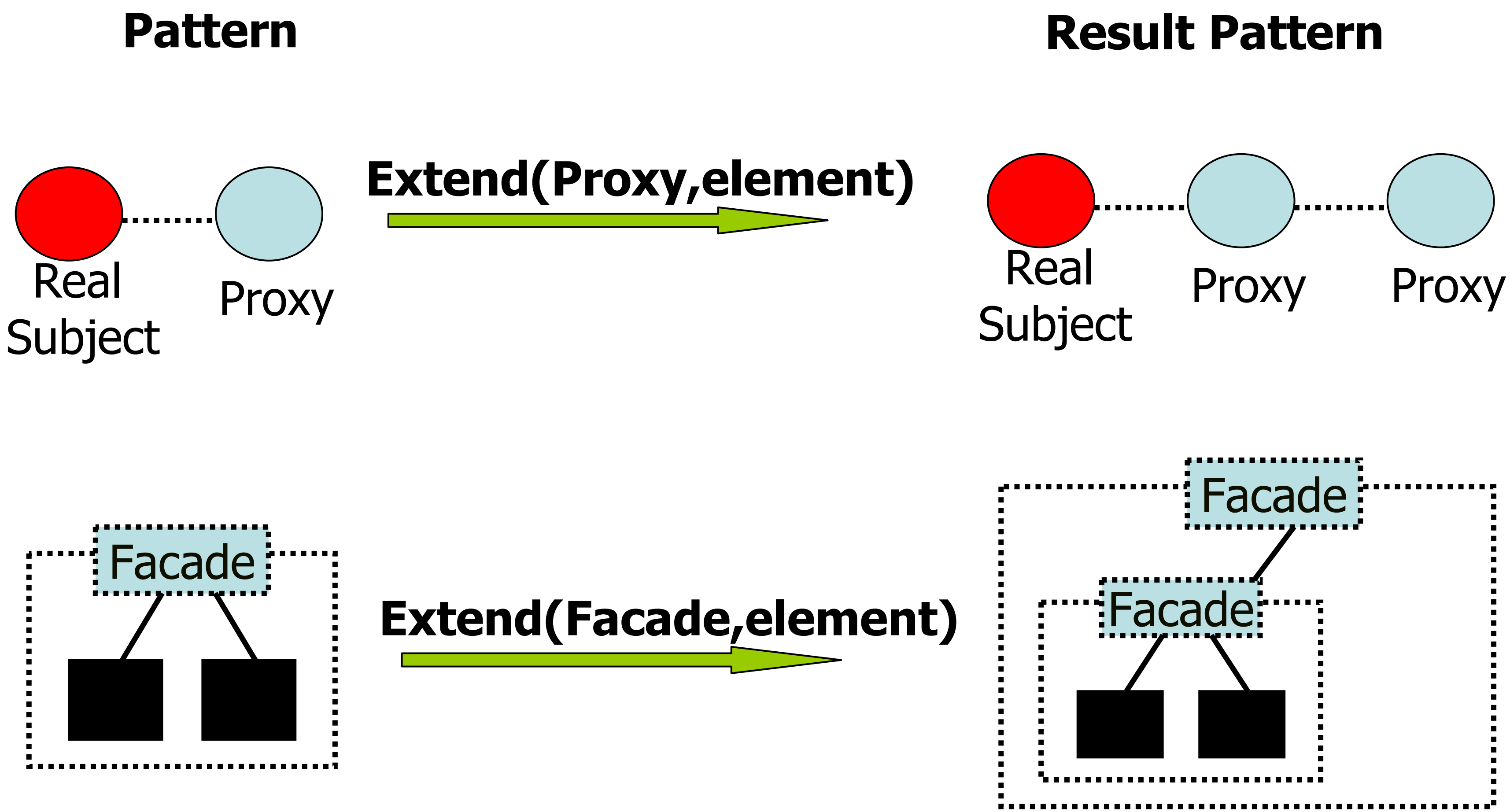
From: Cecilia Gomes

Increase Structural Operator



From: Cecilia Gomes

Extend Structural Operator



From: Cecilia Gomes

Behavioural Pattern Templates

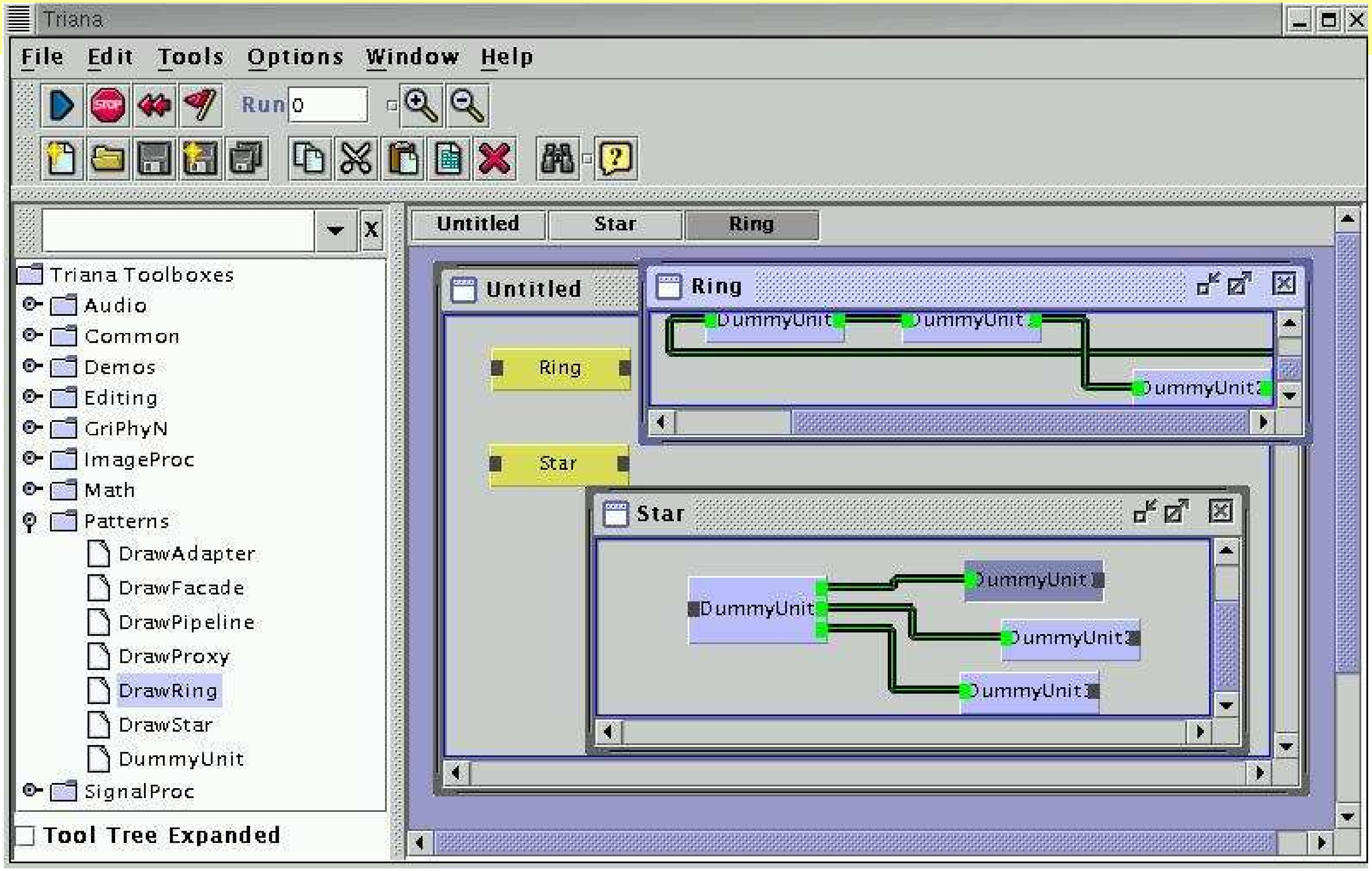
- Capture temporal or (data/control) flow dependencies between components.
- Examples:
  - Client/Server,
  - Master/Slave,
  - Streaming,
  - Service Adapter,
  - Service Migration,
  - Broker Service
  - Service Aggregator/Decomposer, …

From: Cecilia Gomes

Behavioural Operators

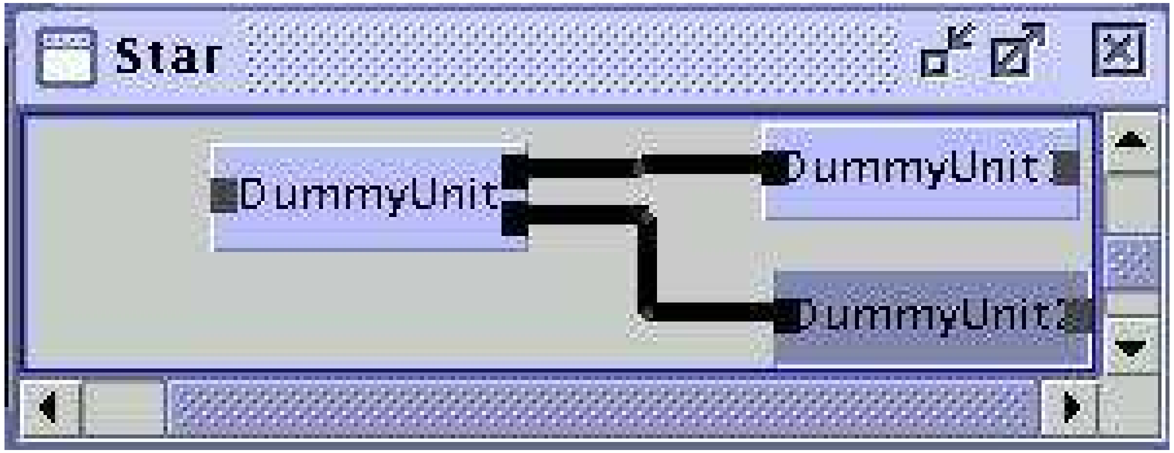
- Act over the temporal or flow dependencies for execution control and reconfiguration.
- Examples:
  - Start, Terminate,
  - Log,
  - Stop, Resume,
  - Restart, Limit,
  - Repeat, …

From: Cecilia Gomes

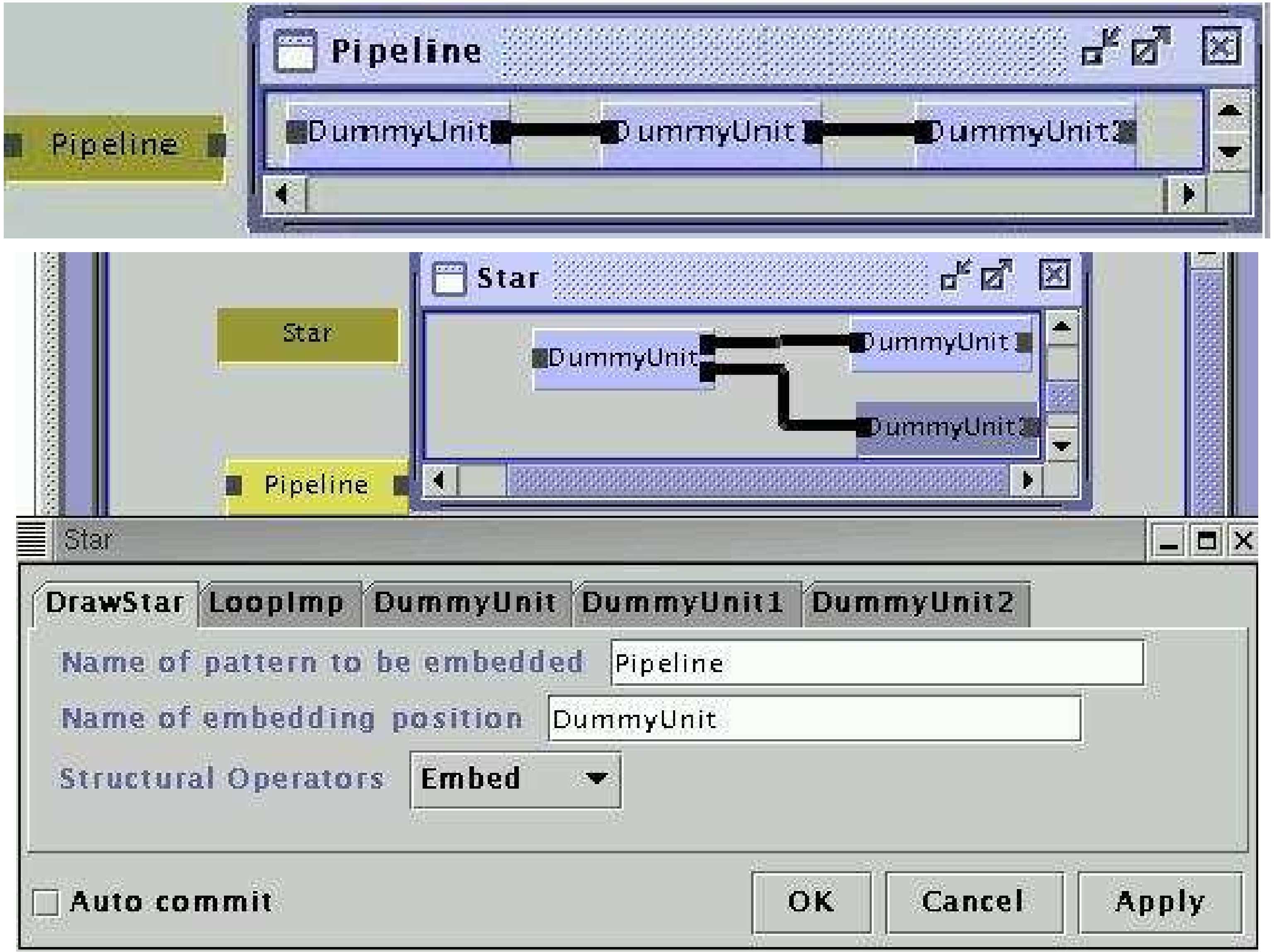


- Applications are built by connecting services available in a toolbox
- The execution follows the dataflow model

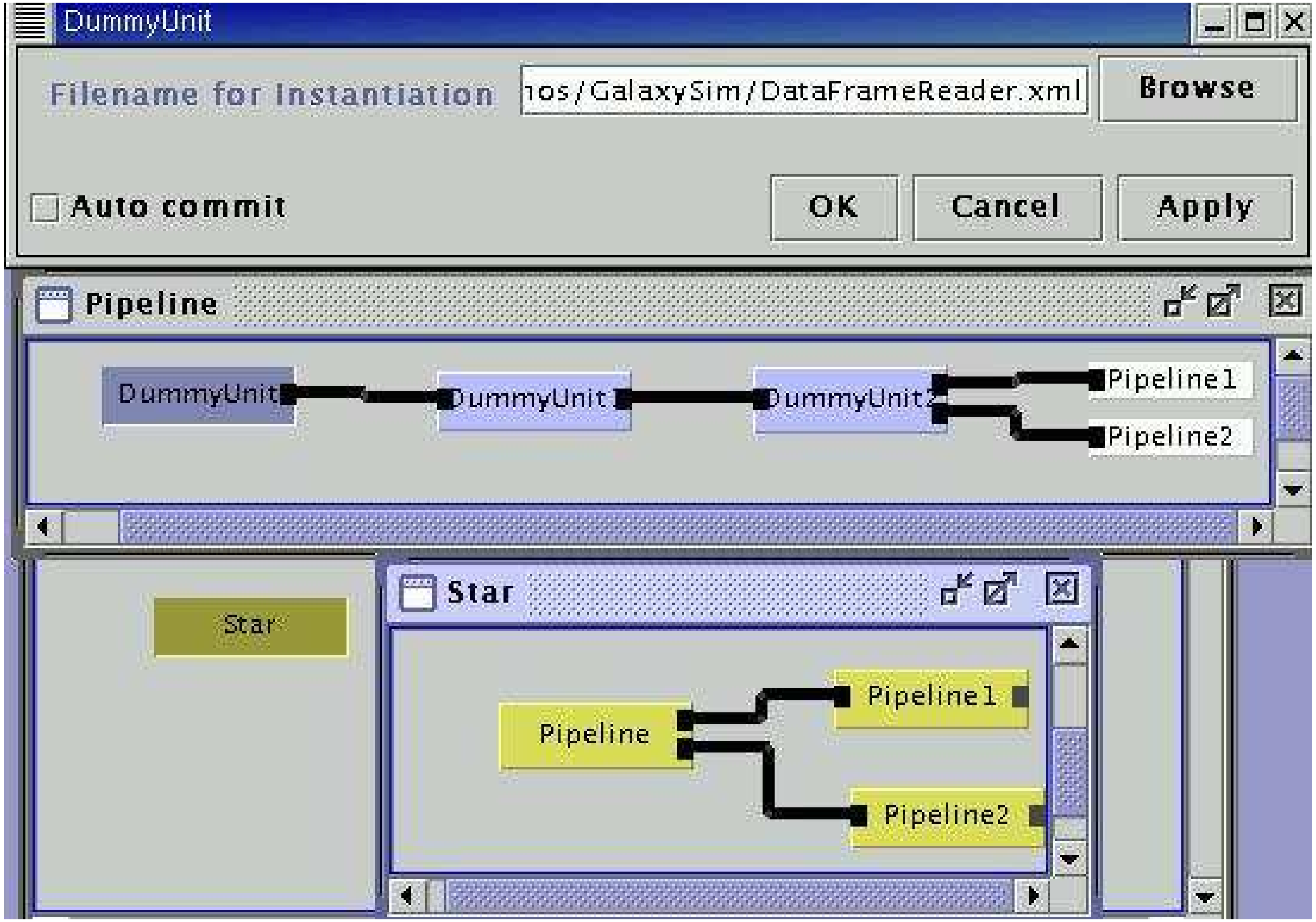
3- Implementation over Triana – example

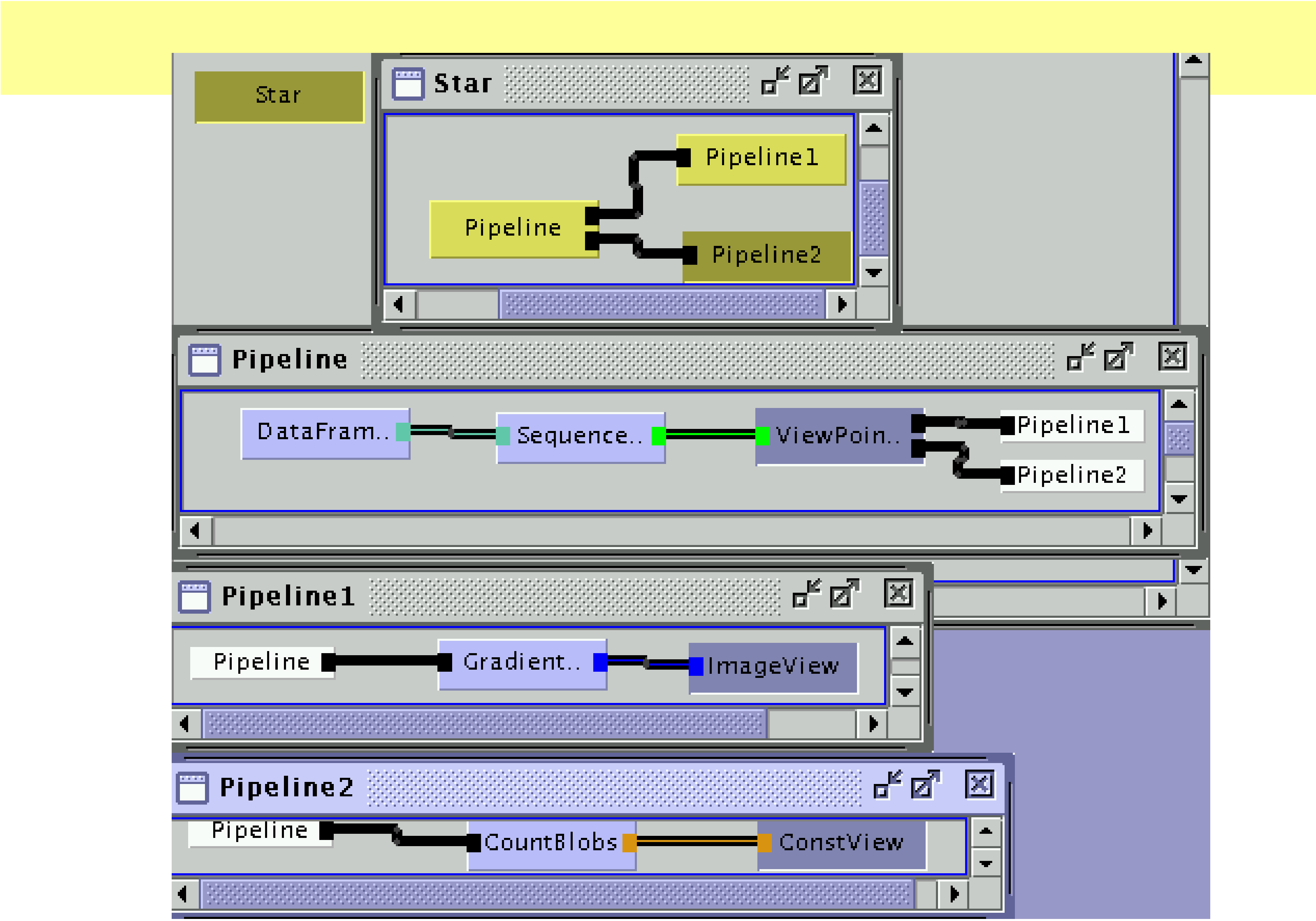


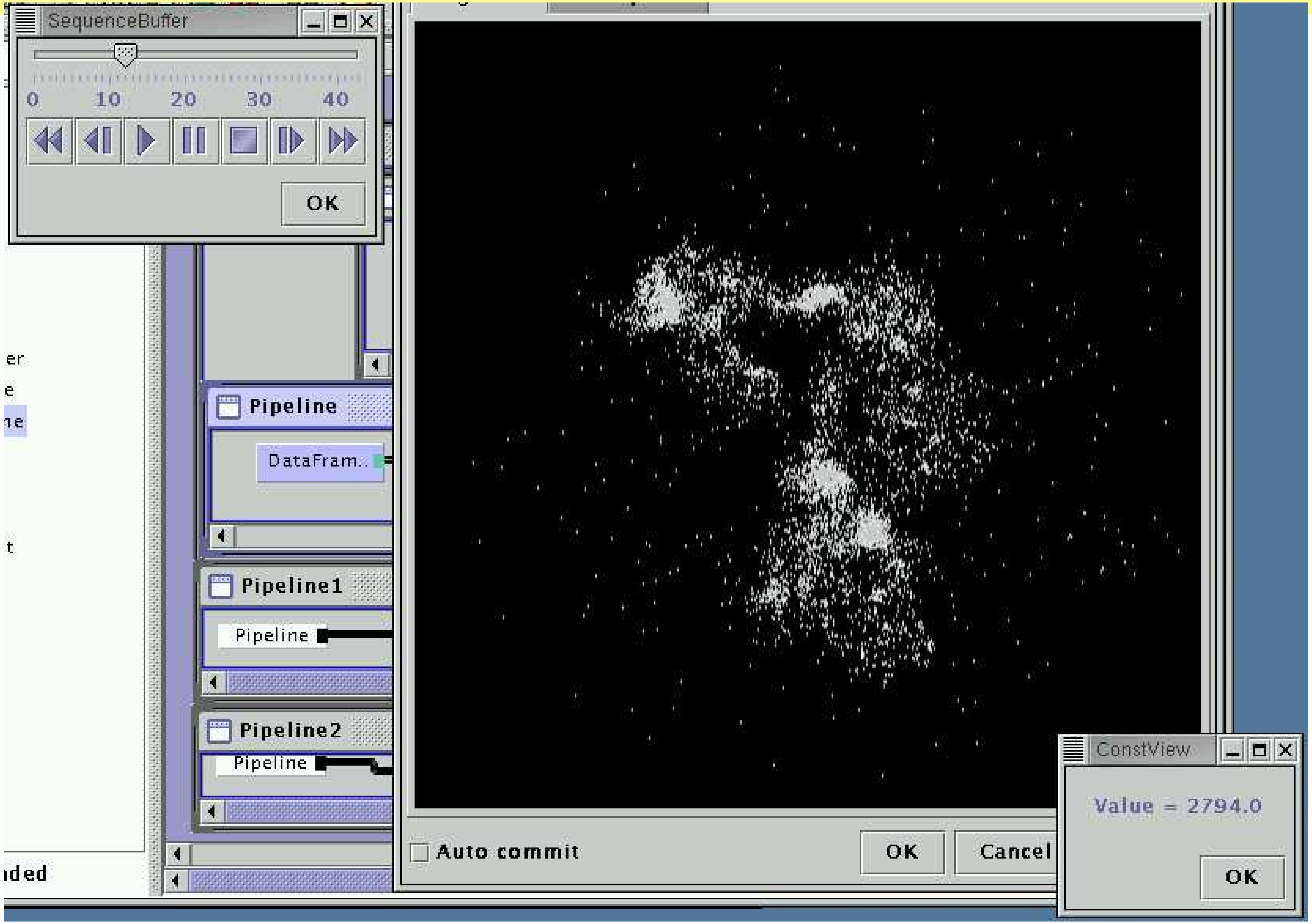
example


Pipeline
DummyUnit
DummyUnit1
DummyUnit2
Star
Pipeline
DrawStar
LoopImp
DummyUnit
DummyUnit1
DummyUnit2
Name of pattern to be embedded
Pipeline
Name of embedding position
DummyUnit
Structural Operators
Embed
Auto commit
OK
Cancel
Apply

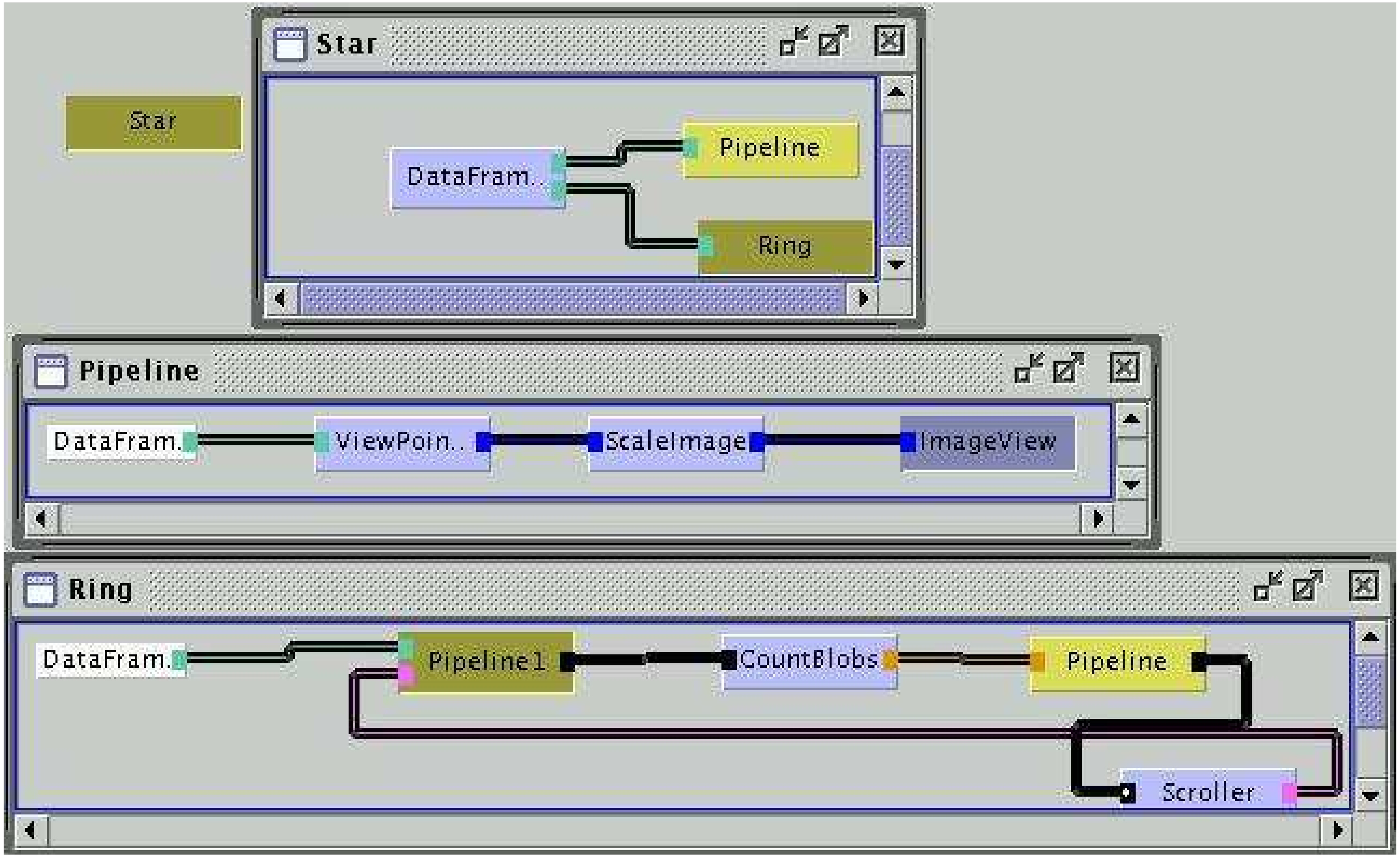
example


DummyUnit
Filename for Instantiation
ıos/GalaxySim/DataFrameReader.xml
Browse
Auto commit
OK
Cancel
Apply
Pipeline
DummyUnit
DummyUnit1
DummyUnit2
Pipeline1
Pipeline2
Star
Star
Pipeline
Pipeline1
Pipeline2


Star
Star
Pipeline1
Pipeline
Pipeline2
Pipeline
DataFram..
Sequence..
ViewPoin..
Pipeline1
Pipeline2
Pipeline1
Pipeline
Gradient..
ImageView
Pipeline2
Pipeline
CountBlobs
ConstView


SequenceBuffer
0 10 20 30 40
OK
Pipeline
DataFram..
Pipeline1
Pipeline
Pipeline2
Pipeline
Auto commit
OK
Cancel
ConstView
Value = 2794.0
OK

a second configuration

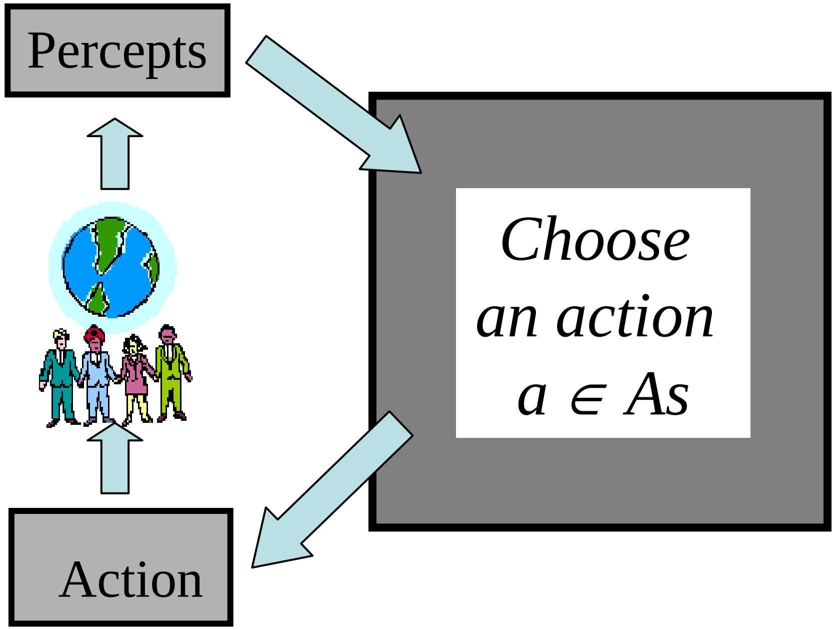


Workflow Planning/Adaptation

- Goal-oriented
- Abstract → Concrete workflow translation
  - May utilise a number of different infrastructure services (Pegasus)
- Level of automation can vary
  - Find components
  - Find sub-workflows
  - Find infrastructure services
  - Publish output data at specific locations

Planning

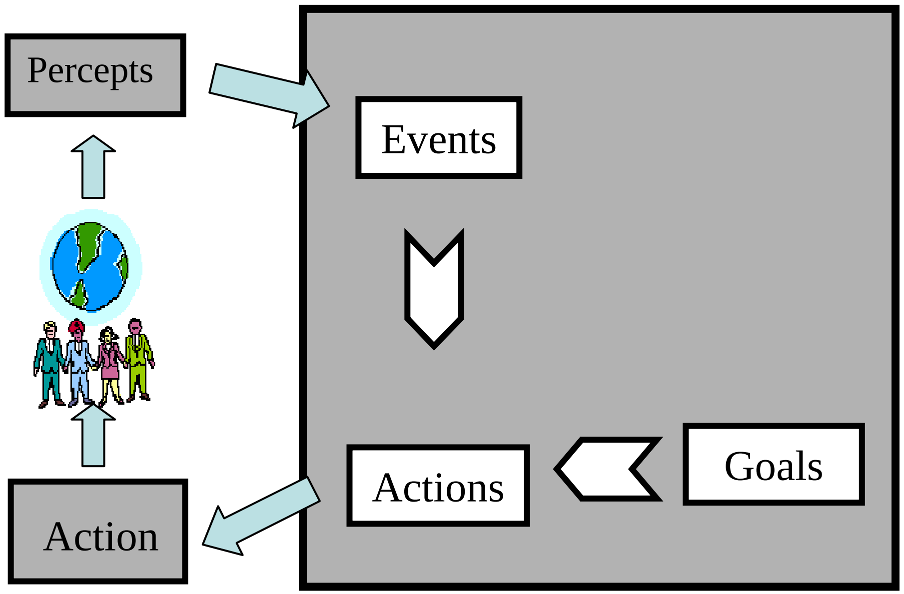
- *Situated* so **actions**, **percepts**, time
- An enactment process:
  - **Monitors "Percepts"**
  - **Executes one or more "Plans"**
  - **Leading to "Actions"**
  - **Leading to new "Percepts"**



From: Michael Winikoff, RMIT

Planning … 2

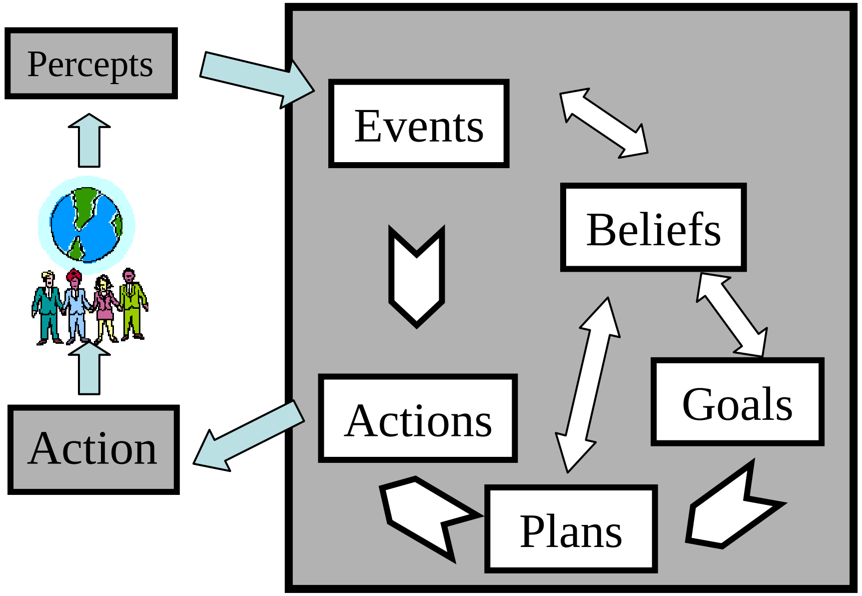
- *Reactive* so **events** (significant occurrence)
  - **Percepts lead to internal events**
  - **Events need to be monitored with reference to "Goals"**
  - **Goals act as filters to decide "Actions"**



From: Michael Winikoff, RMIT

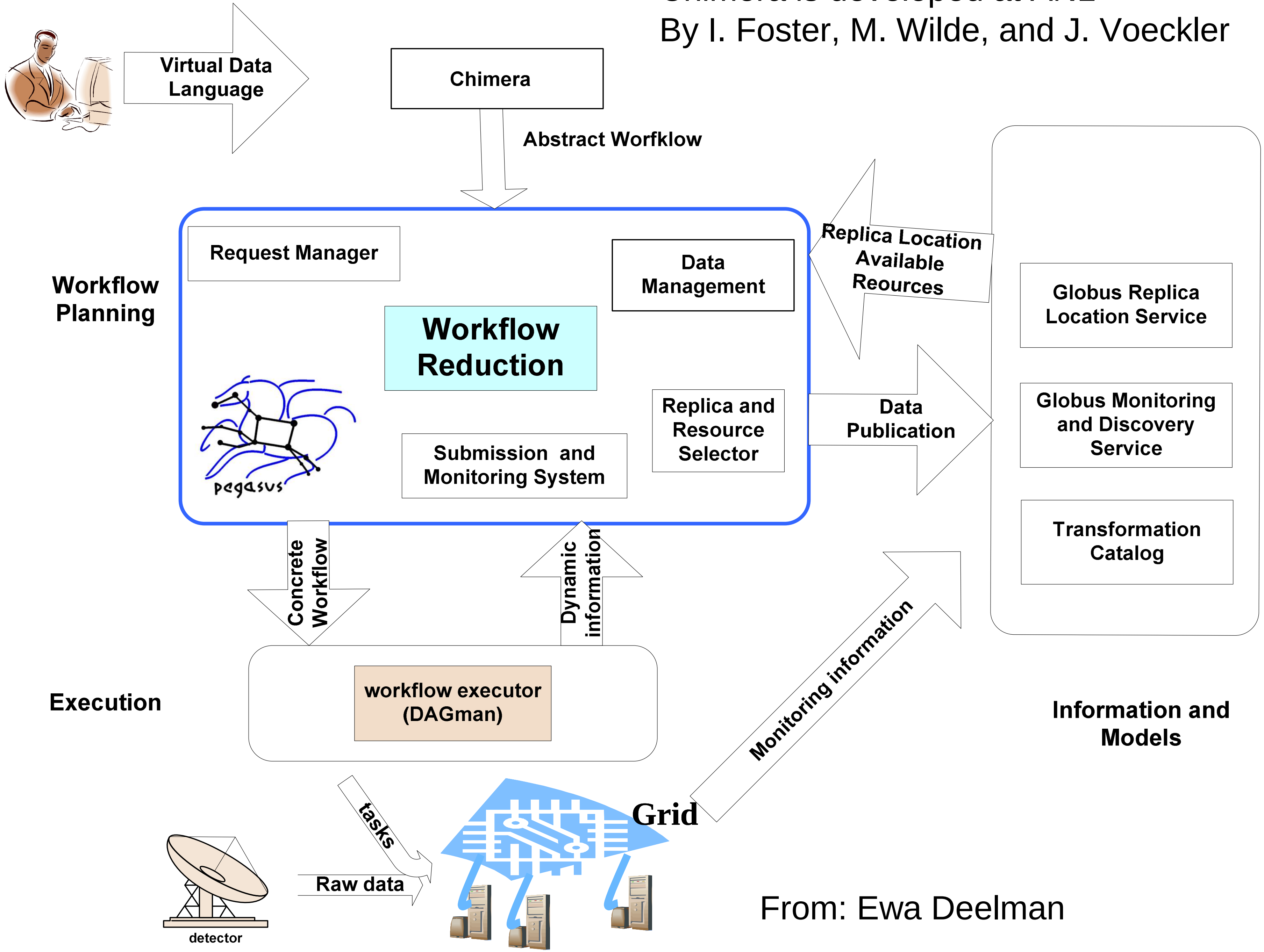
Planning … 3

- Implementation uses **plans** and **beliefs**
  - ➲ Cache for means, and world information respectively
  - ➢ *Beliefs: contains information about current state of resources*
  - ➢ *Plans: chose a schedule to meet a specific deadline*



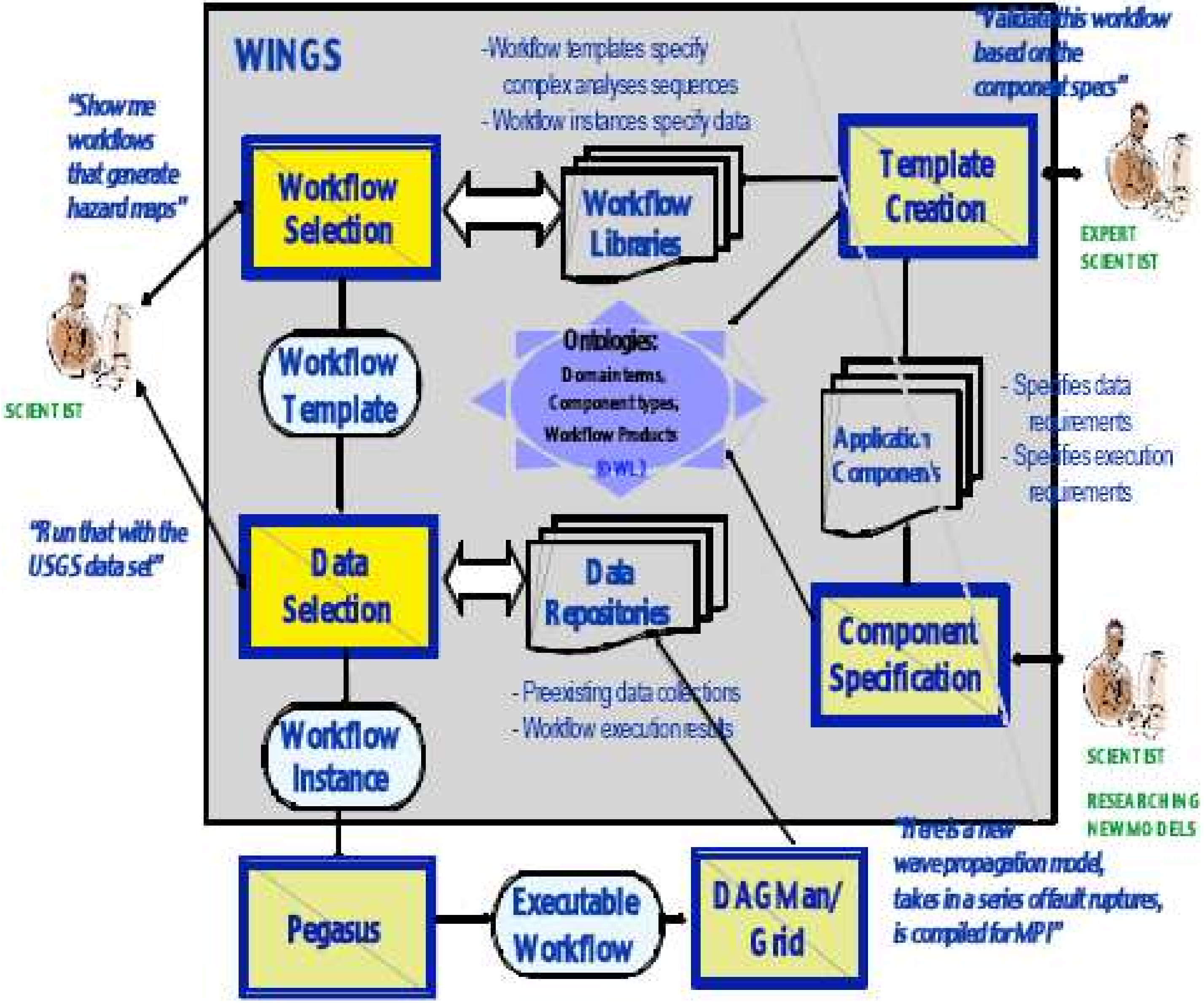
From: Michael Winikoff, RMIT

Chimera is developed at ANL
By I. Foster, M. Wilde, and J. Voeckler



From: Ewa Deelman

Wings for Pegasus

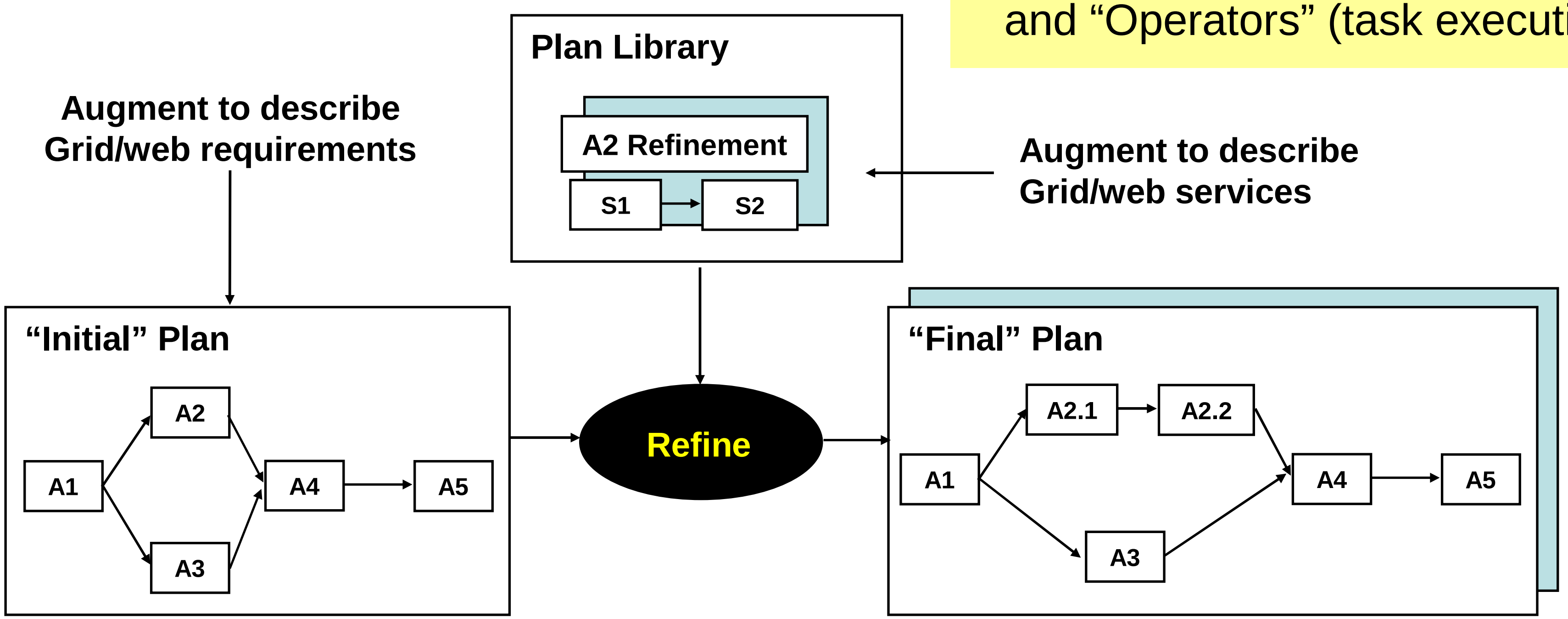


Wings … 2

- Uses: workflow templates, workflow instances and executable workflows
- Data
  - *File*
  - *DataCollection* (objects or files)
- Computation
  - *ComponentType*
  - *ComponentCollection* (hasComponentType property)
- Node
  - Node in a workflow
  - uses *hasComponent* to specify contained component
- Link
  - *hasDestinationNode* and *hasOriginNode*
  - *hasDestinationFileDescription* and *hasOriginFileDescription*
  - Subclasses: *InputLink, InOutLink, OutputLink*
  - Data collections carried in links with *Skolem* instances (stand in for actual data to be used in the instance)

HTN Planning (Activity Composition)

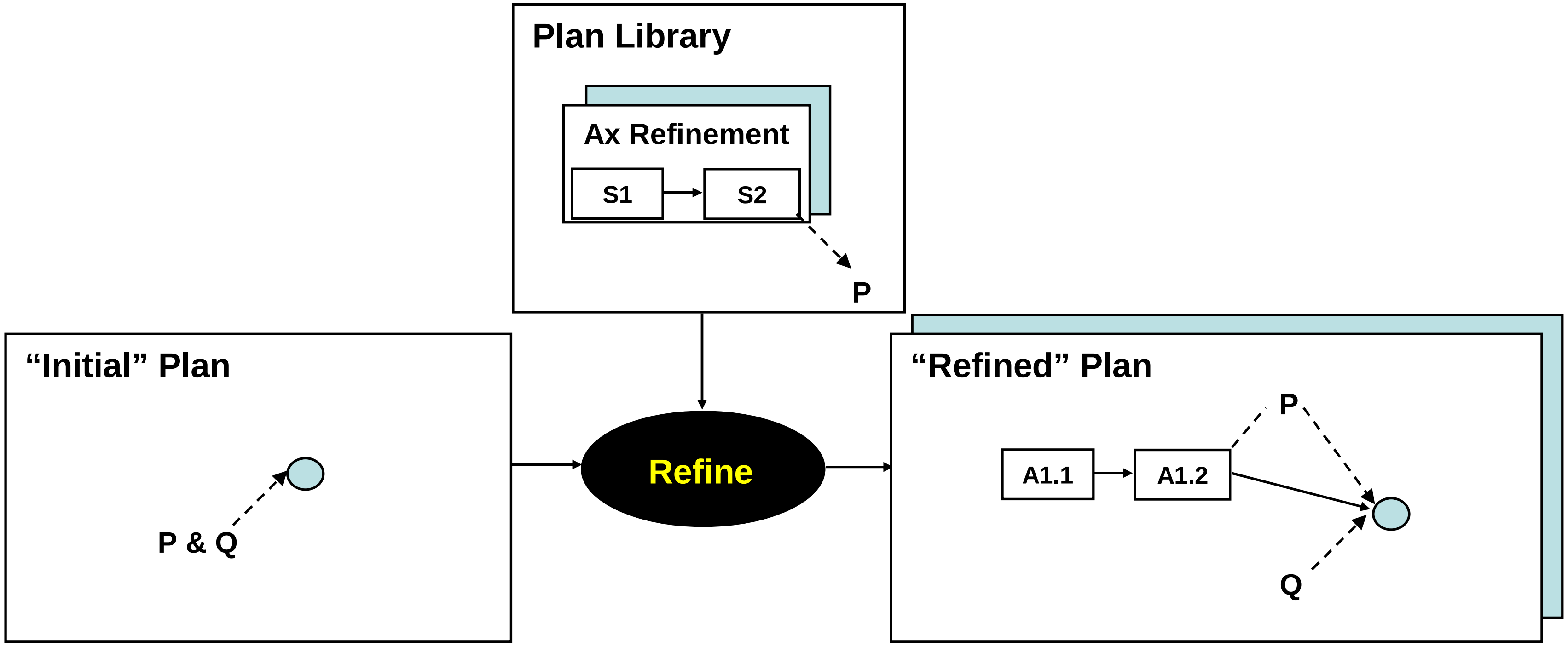
HTN Planning:
Use of "Methods" (task decomp)
and "Operators" (task execution)



Introduce activities to achieve preconditions
Resolve interactions between conditions and effects
Handle constraints (e.g. world state, resource, spatial, etc.)

From: Austin Tate (Edinburgh)

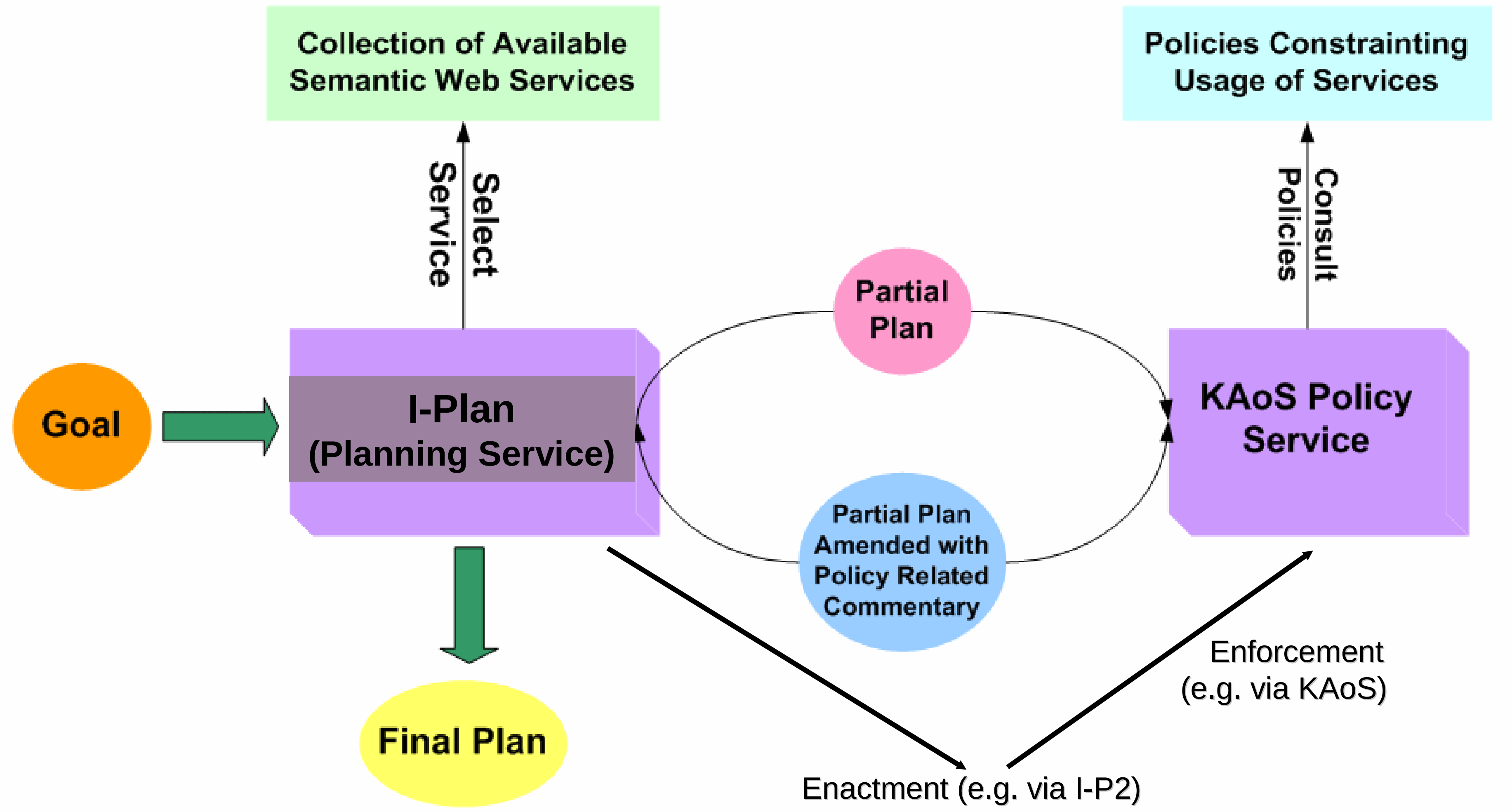
HTN Planning (Initial Plan Stated as “Goals”)



Initial Plan can be any combination of Activities and Constraints

From: Austin Tate (Edinburgh)

Composer & Enactor



From: Austin Tate (Edinburgh)

BDI agents (based on AgentSpeak(L))

$$goal : B_1 \wedge \ldots \wedge B_n \leftarrow S_1; \ldots; S_m$$

where each B_i is a belief, and each S_i is either an action (a), or a subgoal (α_{sub}).

The execution model of AgentSpeak consists of the following steps:

1. The agent selects an event e (note that goals are an event type)
2. The agent generates all plans with matching invocation conditions
3. From these relevant plans the agent identifies those with satisfied preconditions
4. If there are several plans, one is chosen nondeterministically

- Chosen plan added to "intention" stack (can be either an event (posted) or action (executed))

BDI-based Enactor

- Enactor can maintain local plan library
  - update of plan library as new conditions are detected
  - Useful in a dynamic environment (Grid) -- as agents are goal directed
- Execution of a plan leads to update of beliefs
  - useful mechanism to adapt agent behaviour in a Grid context
- Potentially useful to allow detection of plan conflicts
- Traditional approach:
  - number of tasks fixed, resources identical
  - fixed number of resources, tasks pre-defined
- Delegate scheduling priorities to each resource and task agent (no central schedulers)

Planning as Model Checking

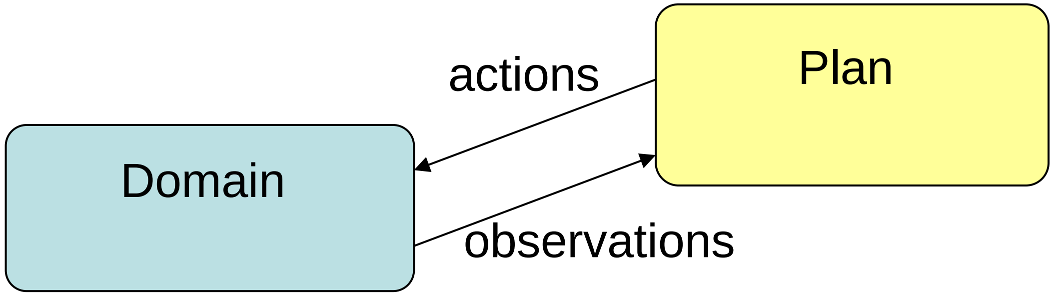
- Planning based on:
  - Non-determinism: cannot predict interactions with external processes – i.e. cannot predict whether answer to a request for availability will be positive or negative
  - Partial Observability: can only observe external interactions (as BPEL) not internal status
  - Extended Goals: behaviour of the "process" is important, and not just the final goal
  - Conditional Preferences: may require multiple conditions to hold for goal to be satisfied

- Given current state, evaluate possible likely states (may require an exhaustive checking of possibilities)

Planning as Model Checking

- Planning under uncertainty
- Support different degrees of “run-time observability”
  - domain state partially visible via sensing
- “Temporally extended planning” goals
  - conditions on states that arise when a plan is executed
  - “goal” specifies conditions/constraints on intermediate states, and not just on the final outcome

Planning Domains



- Domain ➔ Model of generic system
- Plan ➔ monitors evolution of domain via "observations"
  - Controls evolution of domain via "actions"
- Planning domain defined in terms of:
  - States, Actions (it accepts), Observations (domain can exhibit)
  - Transition function: action execution changes domain state
  - Observation function: observations associated with each state

Planning

Definition 1 (planning domain). *A* nondeterministic planning domain with partial observability *is a tuple* $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{T}, \mathcal{X} \rangle$*, where:*

- $\mathcal{S}$ *is the set of* states.
- $\mathcal{A}$ *is the set of* actions.
- $\mathcal{O}$ *is the set of* observations.
- $\mathcal{I} \subseteq \mathcal{S}$ *is the set of* initial states*; we require* $\mathcal{I} \neq \emptyset$.
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to 2^{\mathcal{S}}$ *is the* transition function*; it associates to each current state* $s \in \mathcal{S}$ *and to each action* $a \in \mathcal{A}$ *the set* $\mathcal{T}(s, a) \subseteq \mathcal{S}$ *of next states.*
- $\mathcal{X} : \mathcal{S} \to \mathcal{O}$ *is the* observation function.

Definition 2 (plan). *A* plan *for domain* $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{T}, \mathcal{X} \rangle$ *is a tuple* $\Pi = \langle \mathcal{C}, c_0, \alpha, \epsilon \rangle$*, where:*

- $\mathcal{C}$ *is the set of* plan contexts.
- $c_0 \in \mathcal{C}$ *is the* initial context.
- $\alpha : \mathcal{C} \times \mathcal{O} \rightharpoonup \mathcal{A}$ *is the* action function*; it associates to a plan context* c *and an observation* o *an action* $a = \alpha(c, o)$ *to be executed.*
- $\epsilon : \mathcal{C} \times \mathcal{O} \rightharpoonup \mathcal{C}$ *is the* context evolutions function*; it associates to a plan context* c *and an observation* o *a new plan context* $c' = \epsilon(c, o)$.

Context: captures state

Action Execution and Beliefs

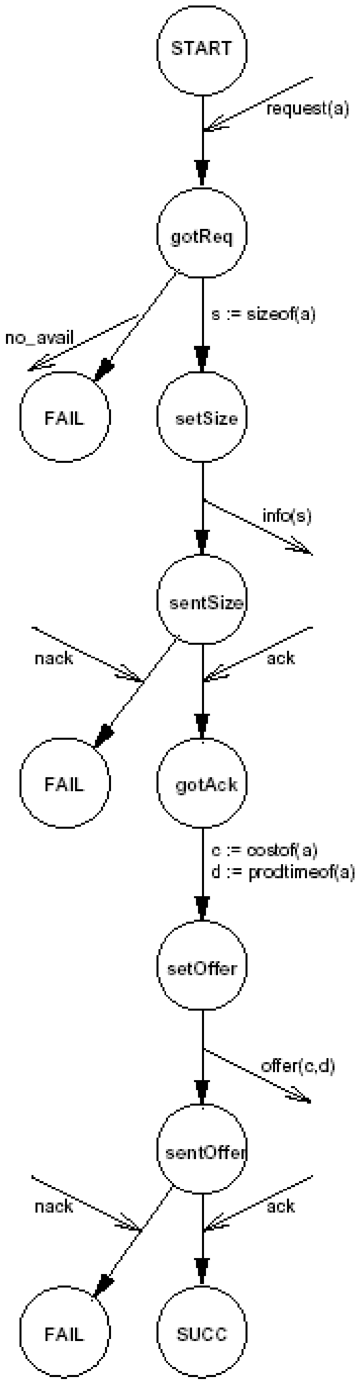
- Context: internal state of plan
  - Account for knowledge gathered during previous steps
  - Actions: depend on observation and on the context
- Due to partial observability, a set of domain states need to be considered (given initial knowledge and current plan state)
  - Executing an action "a" evolves B→B' (contains all possible states that can be reached through "a" from "B")
  - If after executing "a" observation "o" still holds, then filter out states for which "o" is not valid

$$Evolve(B, a, o) = \{s' : \exists s \in B . s' \in \mathcal{T}(s, a) \wedge \mathcal{X}(s') = o\}.$$

Application to Web Services Composition

- First model the process undertaken within each service involved
- Synthesise, using planning, a process that interacts with the three processes (each service) in order to reach a particular state

- Aim to reach some “ideal” state – defined as an overall goal



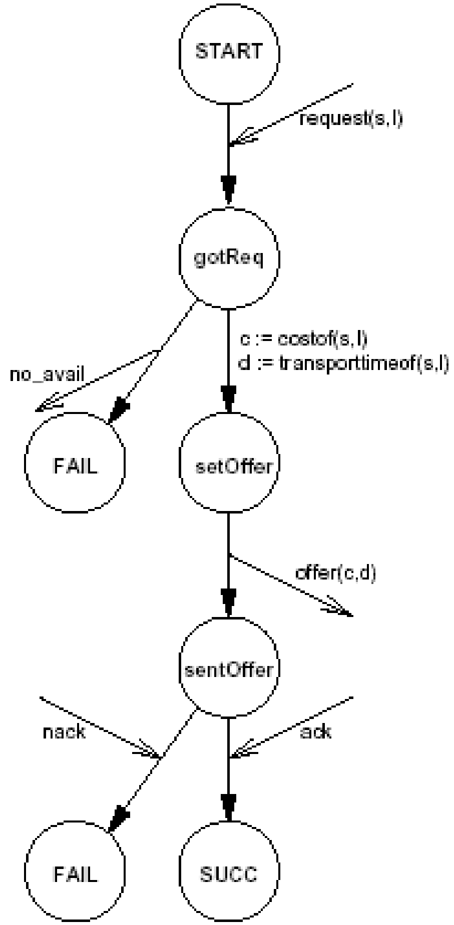


Fig.1.2: The shipper protocol

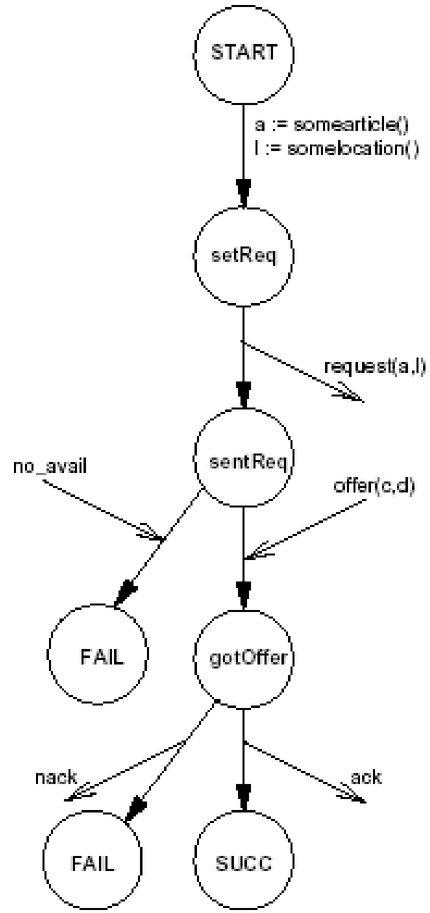
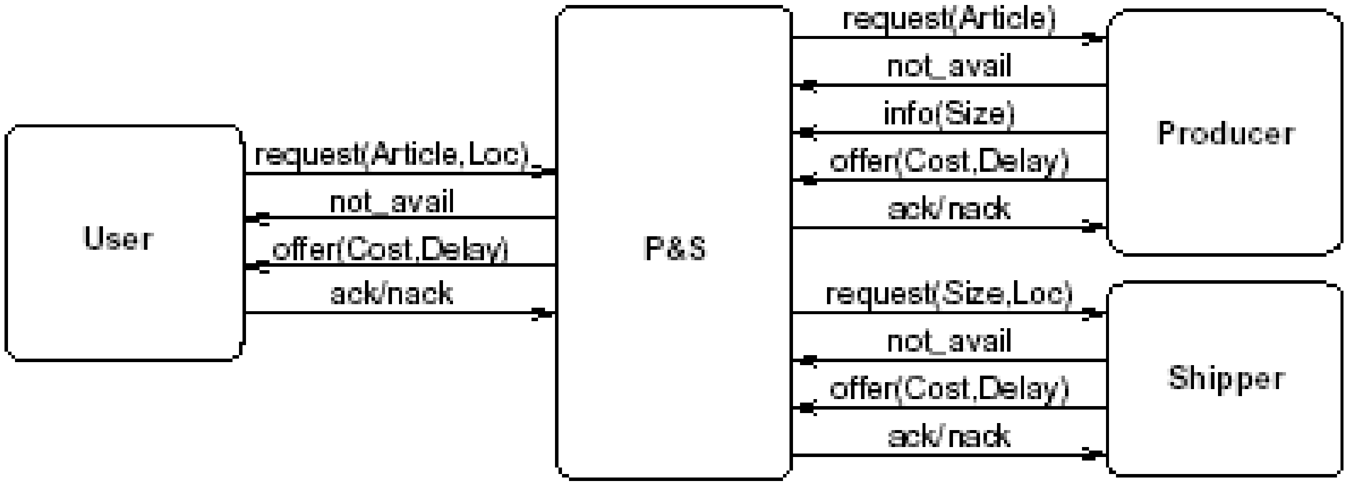


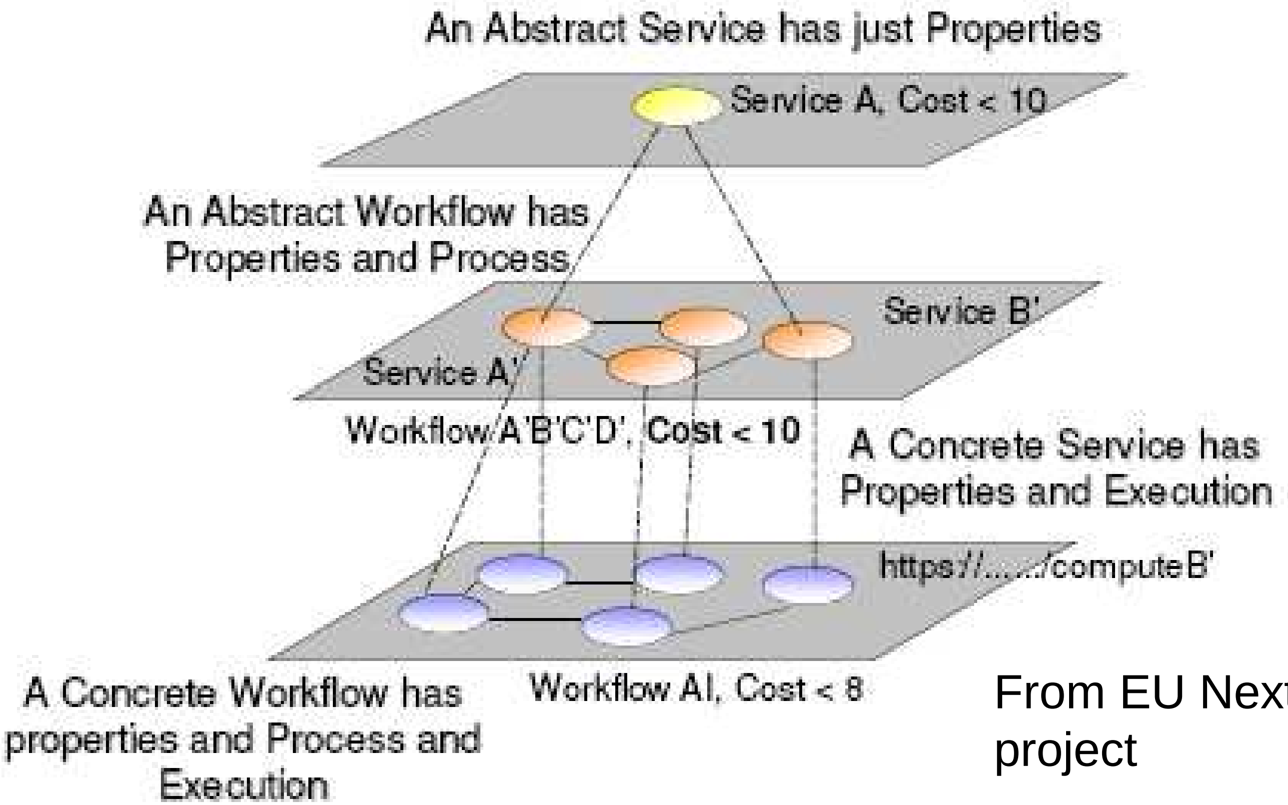
Fig.1.1: The user protocol

Composed Web Service



Semantic Approaches

- Component/Services have "rich" annotations to aid discovery
- Descriptions also contain support for composition of components



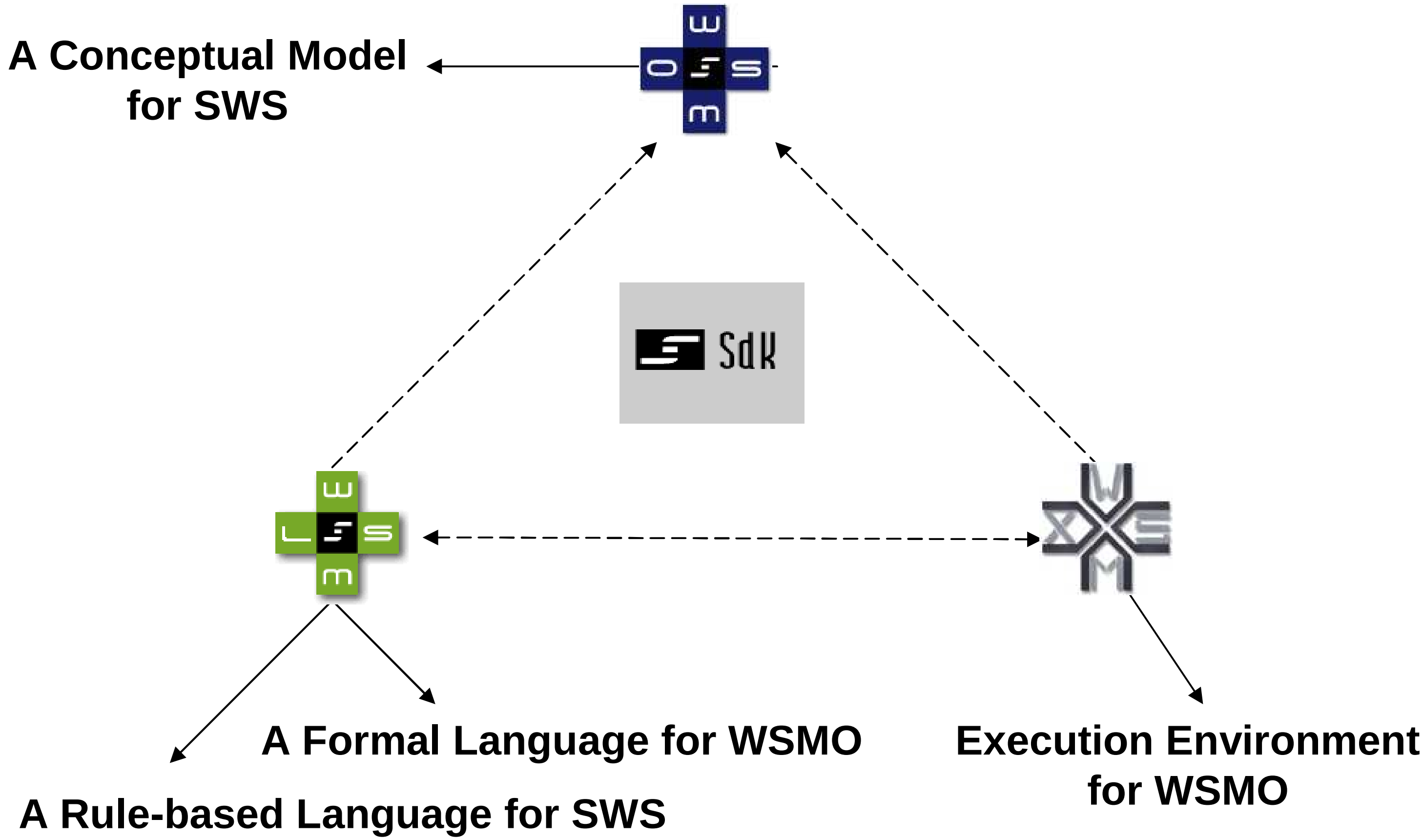
From EU NextGRID project

Web Services Modelling Ontology (WSMO)

- Use of Semantic Web Services to aid automated composition
- Given a goal, identify how services could be composed to achieve the goal
- Specifies a complete set of infrastructure that is necessary to achieve this
- Provides three main components:
  - Web Services Modelling Ontology
  - Web Services Modelling Language
  - Execution Environment

From: John Domingue, Open University

WSMO Working Groups

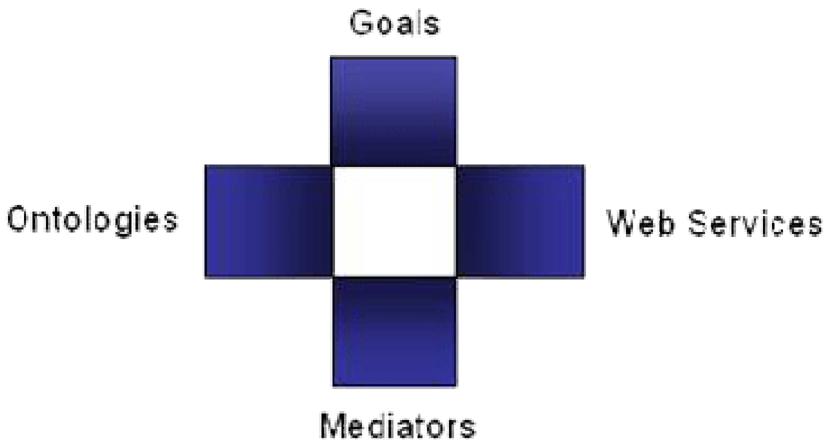


From: John Domingue, Open University

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components



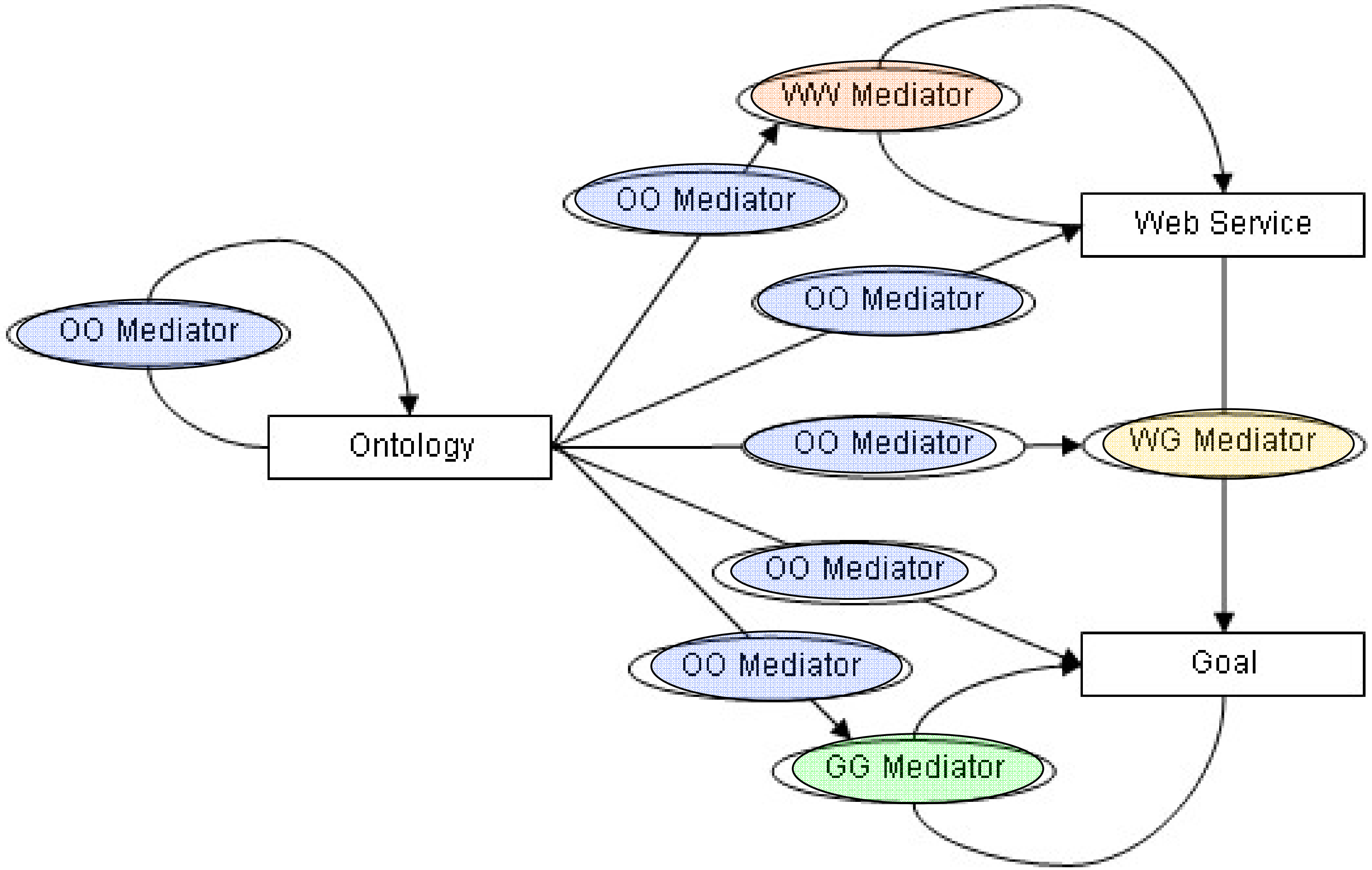
Semantic description of Web Services:

- **Capability** *(functional)*
- **Interfaces** *(usage)*

Connectors between components with mediation facilities for handling heterogeneities

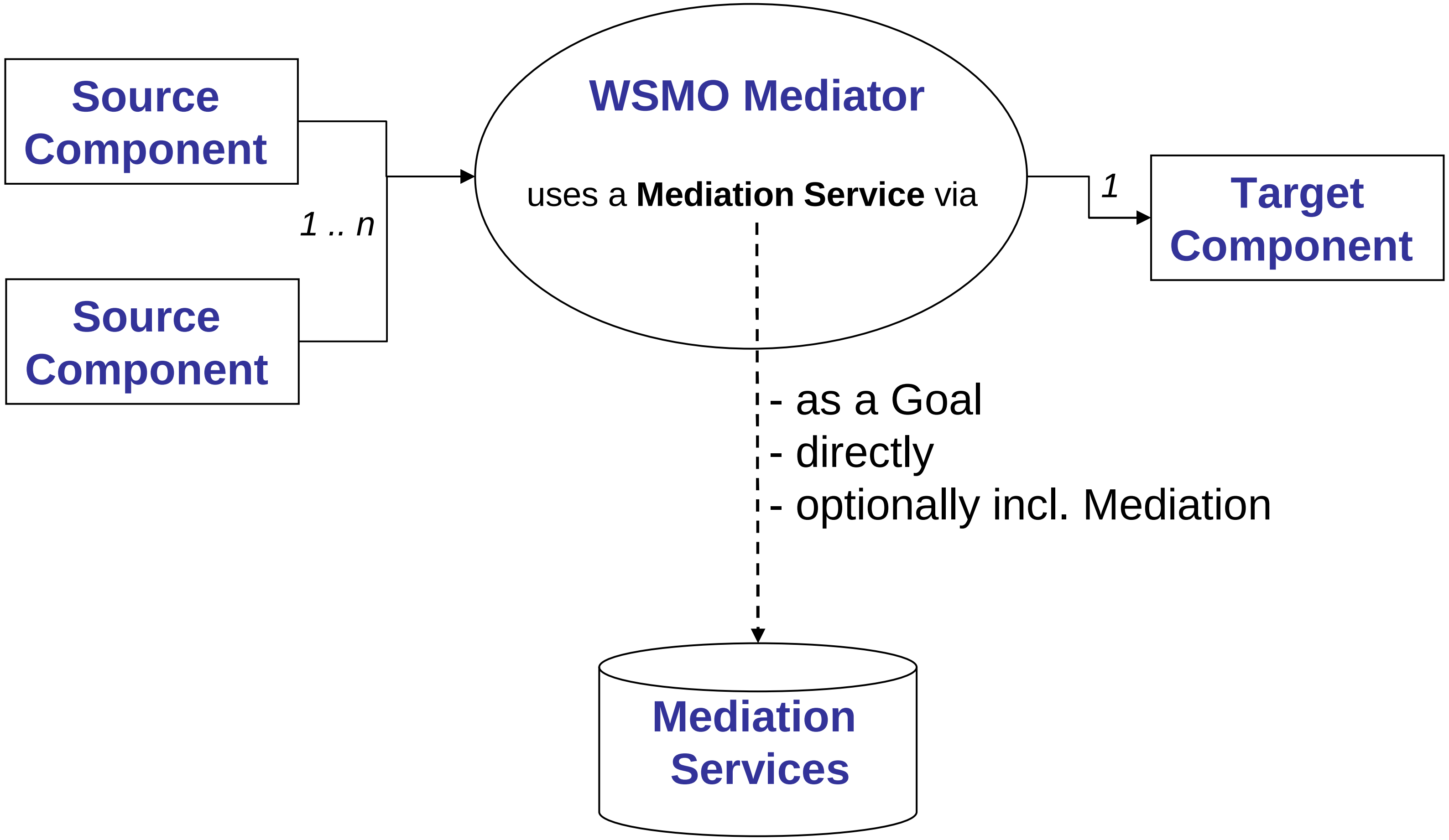
From: John Domingue, Open University

WSMO Mediators Overview



From: John Domingue, Open University

Mediator Structure



From: John Domingue, Open University

DAML-S (Similar to OWL-S)

- Primarily aimed at Software Agents community
- Enable reasoning/planning about services
  - With particular support for automated composition
  - Integration with other information services

- Key aspect:
  - Notion of a service "profile"
  - Used to register service + support automated discovery via a matchmaking infrastructure
  - Use of "service advertisement"
  - Specifies inputs, outputs, pre-conditions, effects (post-conditions)

Provenance Definition

The provenance of a piece of data is the <u>process</u> that led to that piece of data

- We represent the provenance of some data by *documenting* the process that led to the data:
  - documentation can be complete or partial;
  - it can be accurate or inaccurate;
  - it can present conflicting or consensual views of the actors involved;
  - it can provide operational details of execution or it can be abstract.

From: Luc Moreau
(U Southampton)

Provenance constituents

- The provenance of a data item is composed of several elements:
  - Interaction provenance: the set of all interactions between actors involved in the computation of the data
  - Actor provenance: the documentation provided by a particular actor pertaining to an interaction
  - Grouping: notion that allow us to give a scope, in terms of execution semantics and application's needs.

Provenance Questions

- After completion of workflow:
  1. Did the services I use actually fulfil my overall application requirement?
  2. Two of the analysis were performed on the same initial data but have different results – did I alter the services between these experiments?
  3. Did I perform each service on the type of data that the service was intended to analyse, i.e. were the inputs and outputs of each activity compatible?
  4. Did I use data sources from the same site?
  5. Why did it take much longer to run the analysis in the second instance?

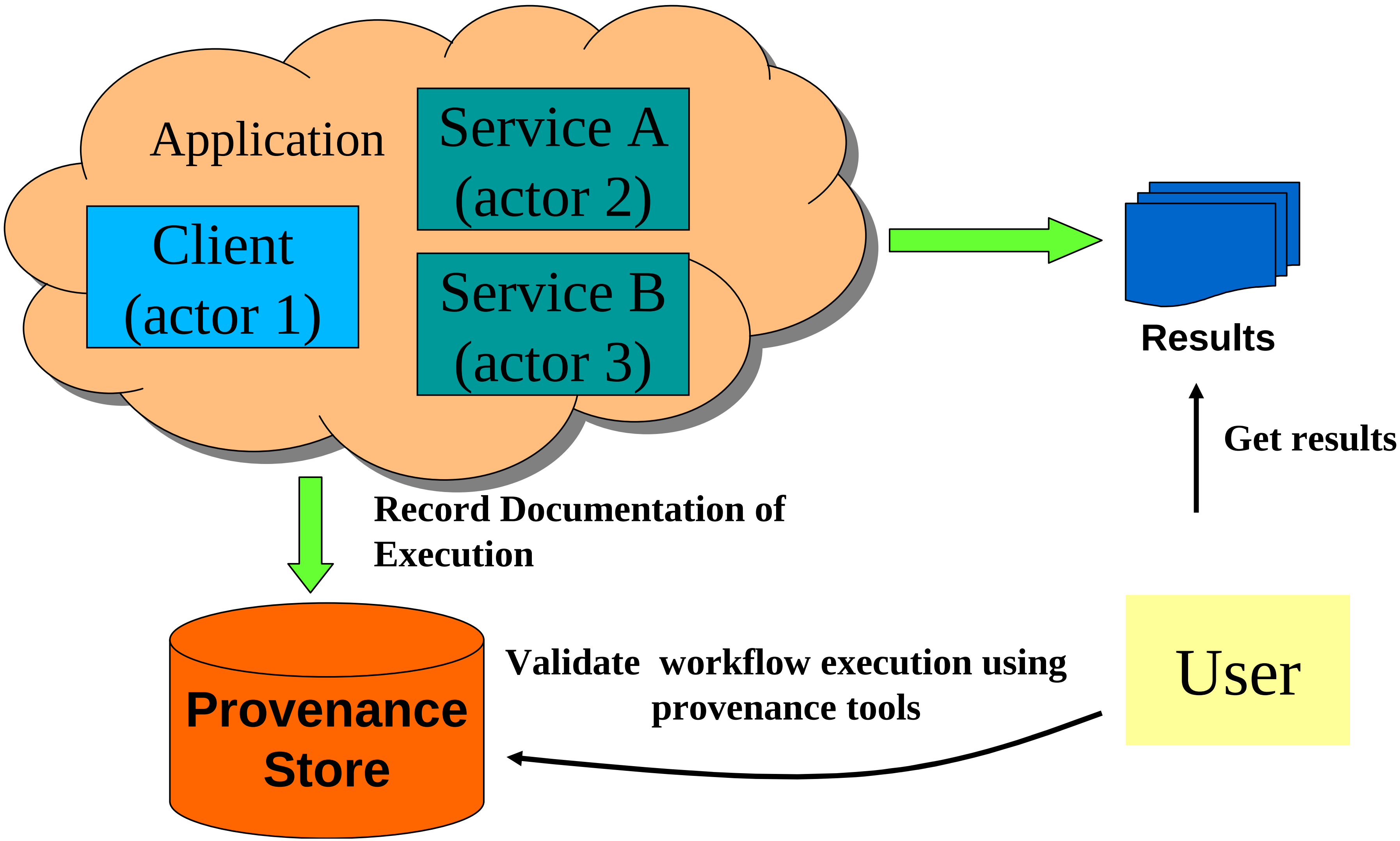
Particularly significant in the context of Distributed Services

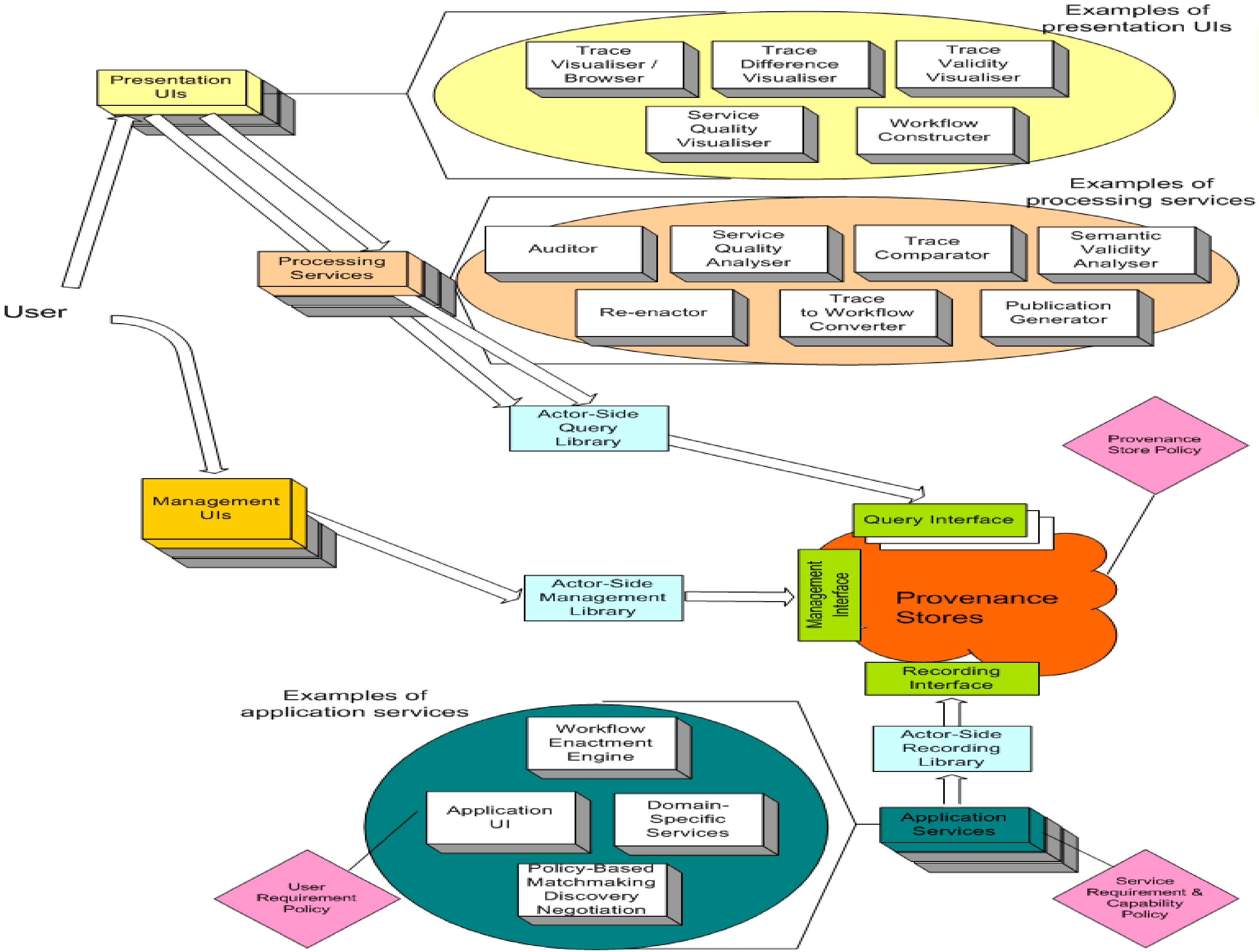
p-assertion

- A given element of process documentation referred to as a p-assertion
  - p-assertion: is an assertion that is made by an actor and pertains to a process.
- Types
  - Interaction p-assertion
    - relates to content of received/sent message
  - Actor p-assertion
    - Relationships between actors
    - State of an actor

From: Luc Moreau
(U Southampton)

Provenance architecture





Examples of presentation UIs
Trace Visualiser / Browser
Trace Difference Visualiser
Trace Validity Visualiser
Service Quality Visualiser
Workflow Constructer
Presentation UIs
Examples of processing services
Auditor
Service Quality Analyser
Trace Comparator
Semantic Validity Analyser
Re-enactor
Trace to Workflow Converter
Publication Generator
Processing Services
User
Actor-Side Query Library
Provenance Store Policy
Query Interface
Management UIs
Actor-Side Management Library
Management Interface
Provenance Stores
Recording Interface
Examples of application services
Workflow Enactment Engine
Application UI
Domain-Specific Services
Policy-Based Matchmaking Discovery Negotiation
Actor-Side Recording Library
Application Services
User Requirement Policy
Service Requirement & Capability Policy

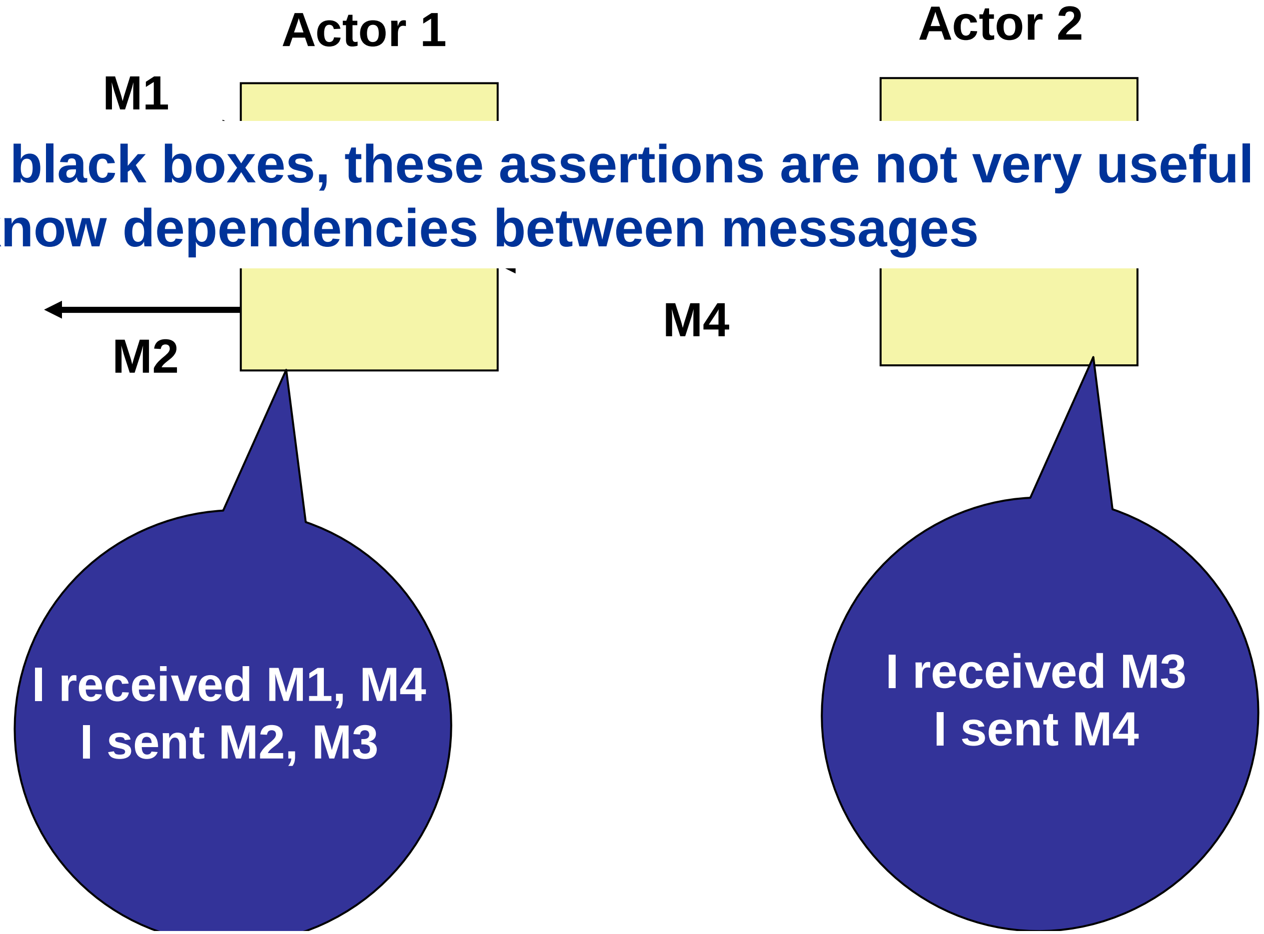
p-assertion

- A given element of process documentation referred to as a p-assertion
  - p-assertion: is an assertion that is made by an actor and pertains to a process.
- Types
  - Interaction p-assertion
    - relates to content of received/sent message
  - Actor State p-assertion
    - State of an actor
  - Relationship p-assertion
    - Relationships between interaction p-assertions

From: Luc Moreau (U Southampton)

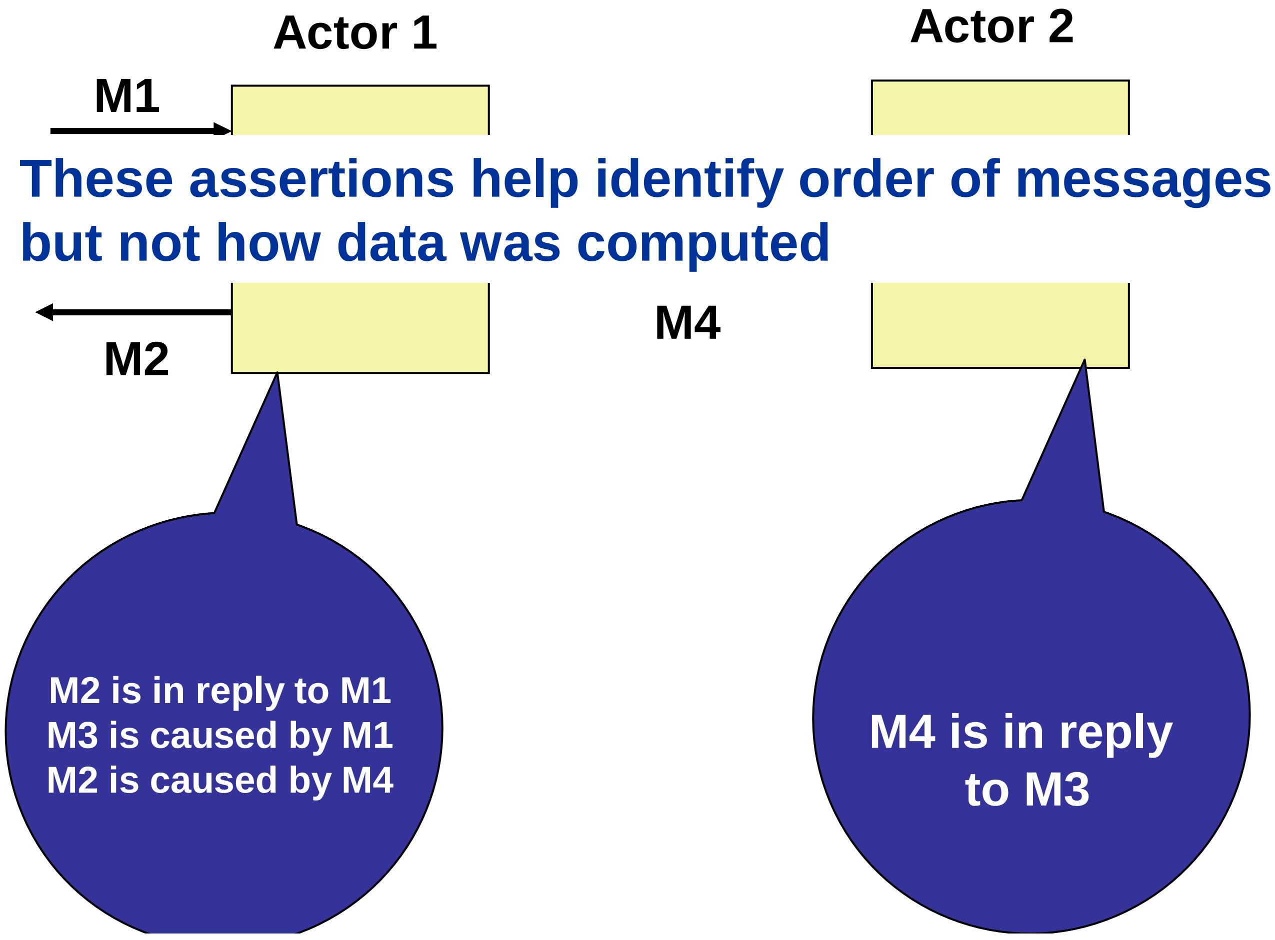
Process Documentation (1)

From these p-assertions, we can derive that M3 was sent by Actor 1 and received by Actor 2 (and likewise for M4)

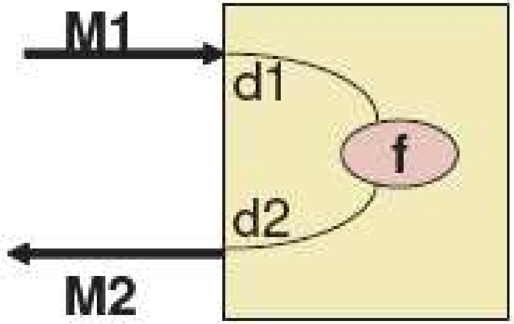


If actors are black boxes, these assertions are not very useful because we do not know dependencies between messages

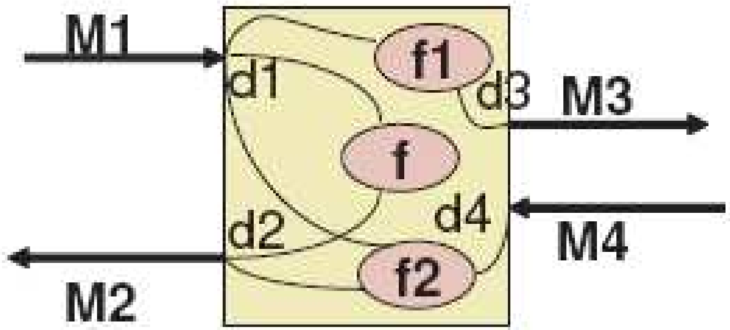
Process Documentation (2)



These assertions help identify order of messages, but not how data was computed

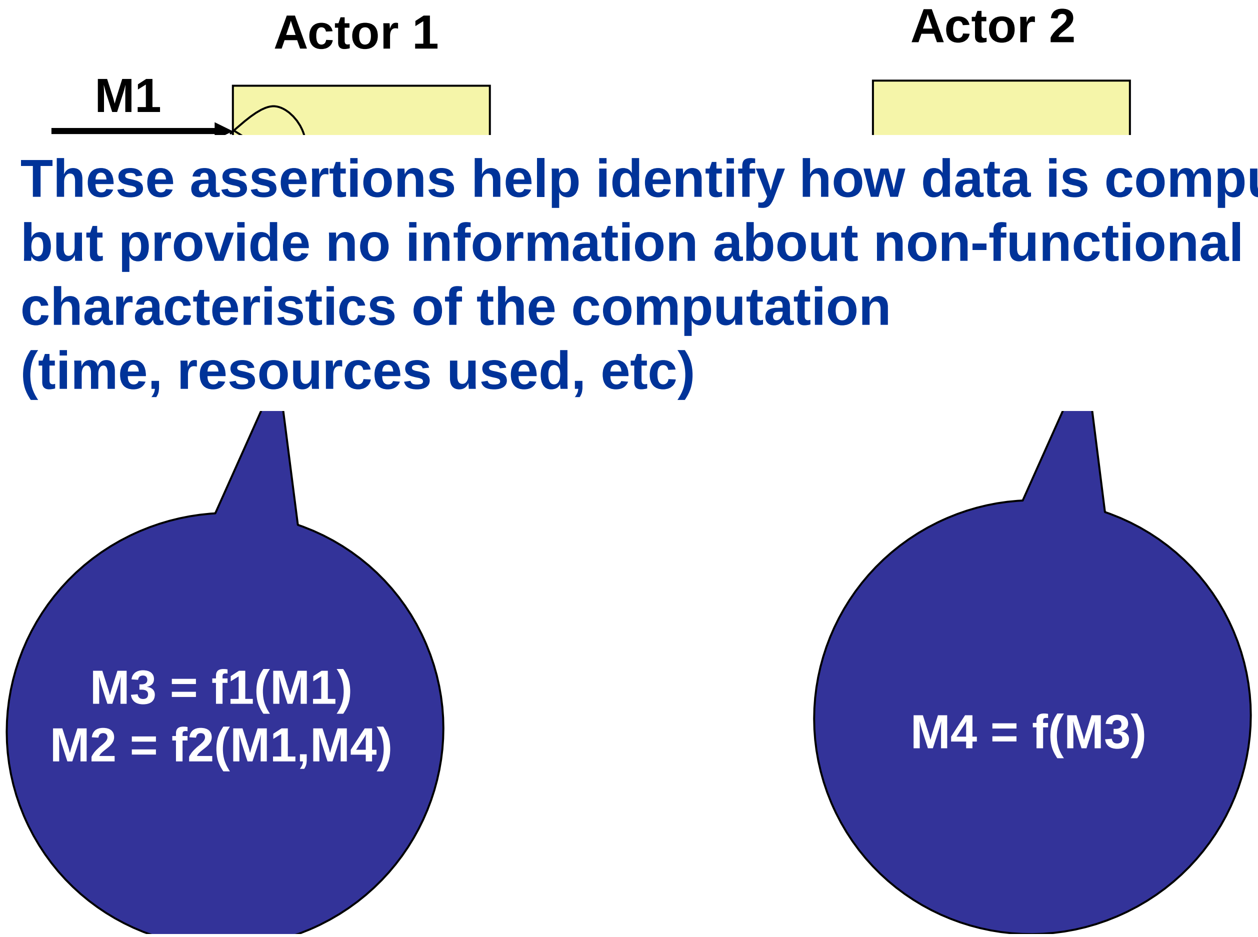


interaction key	p-assertion type	p-assertion content
1	interaction	M1
2	interaction	M2
2	relationship	d2=f(d1)



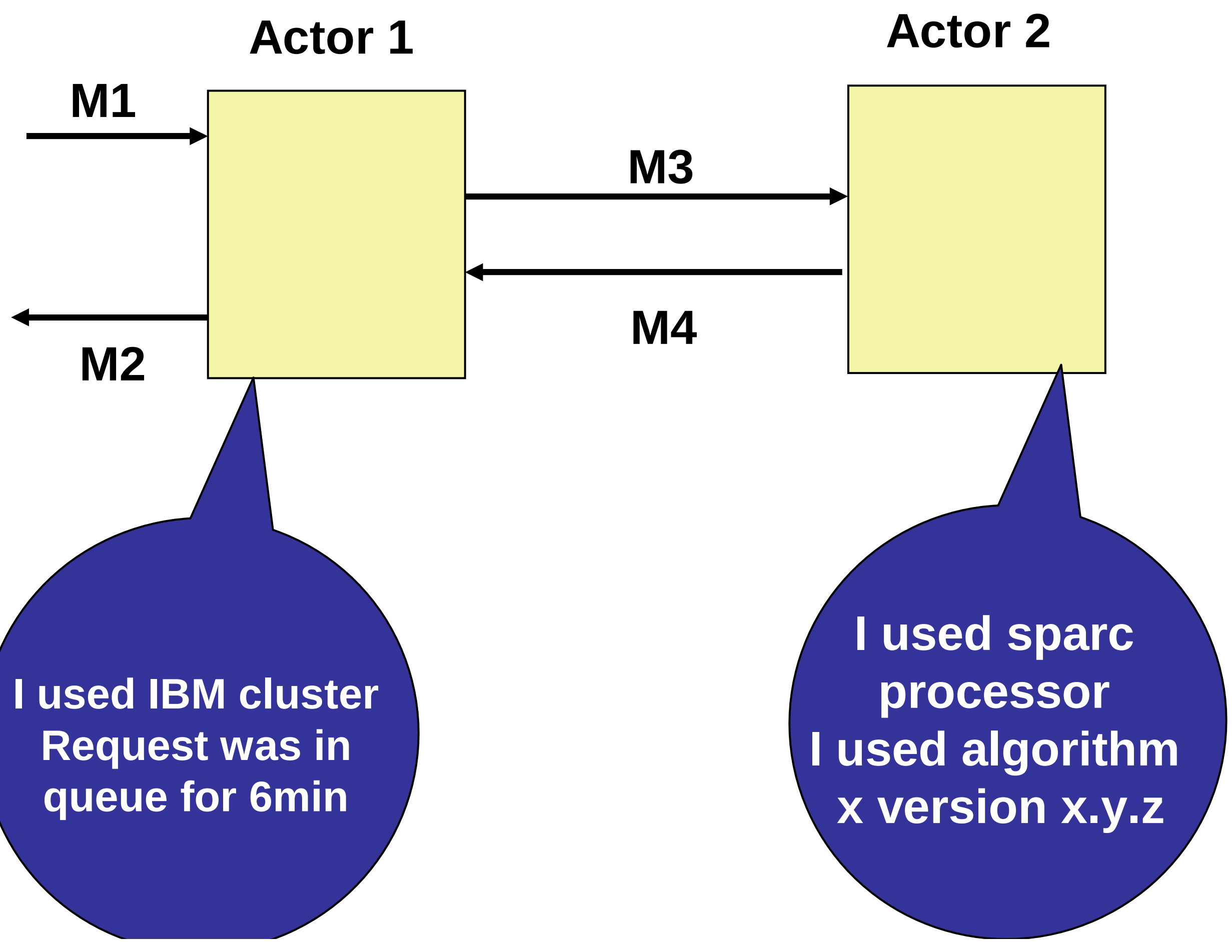
interaction key	p-assertion type	p-assertion content
1	interaction	M1
2	interaction	M2
3	interaction	M3
4	interaction	M4
2	relationship	d2=f(d1)
3	relationship	d3=f1(d1)
2	relationship	d2=f2(d4,d1)

Process Documentation (3)



These assertions help identify how data is computed, but provide no information about non-functional characteristics of the computation (time, resources used, etc)

Process Documentation (4)



Types of p-assertions (1)

- Interaction p-assertion: is an assertion of the contents of a message by an actor that has sent or received that message

Types of p-assertions (2)

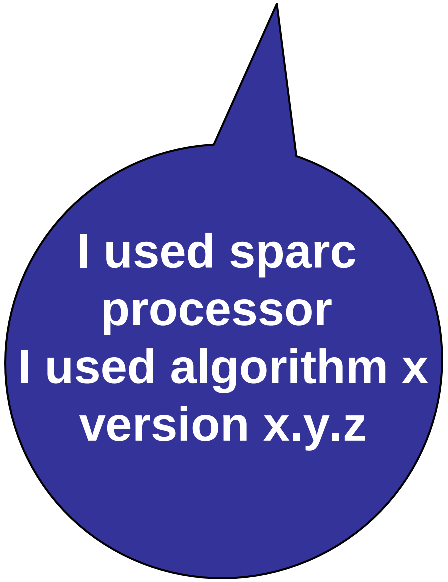
- Relationship p-assertion: is an assertion, made by an actor, that describes how the actor obtained an output message sent in an interaction by applying some function to input messages from other interactions (likewise for data)

M2 is in reply to M1
M3 is caused by M1
M2 is caused by M4

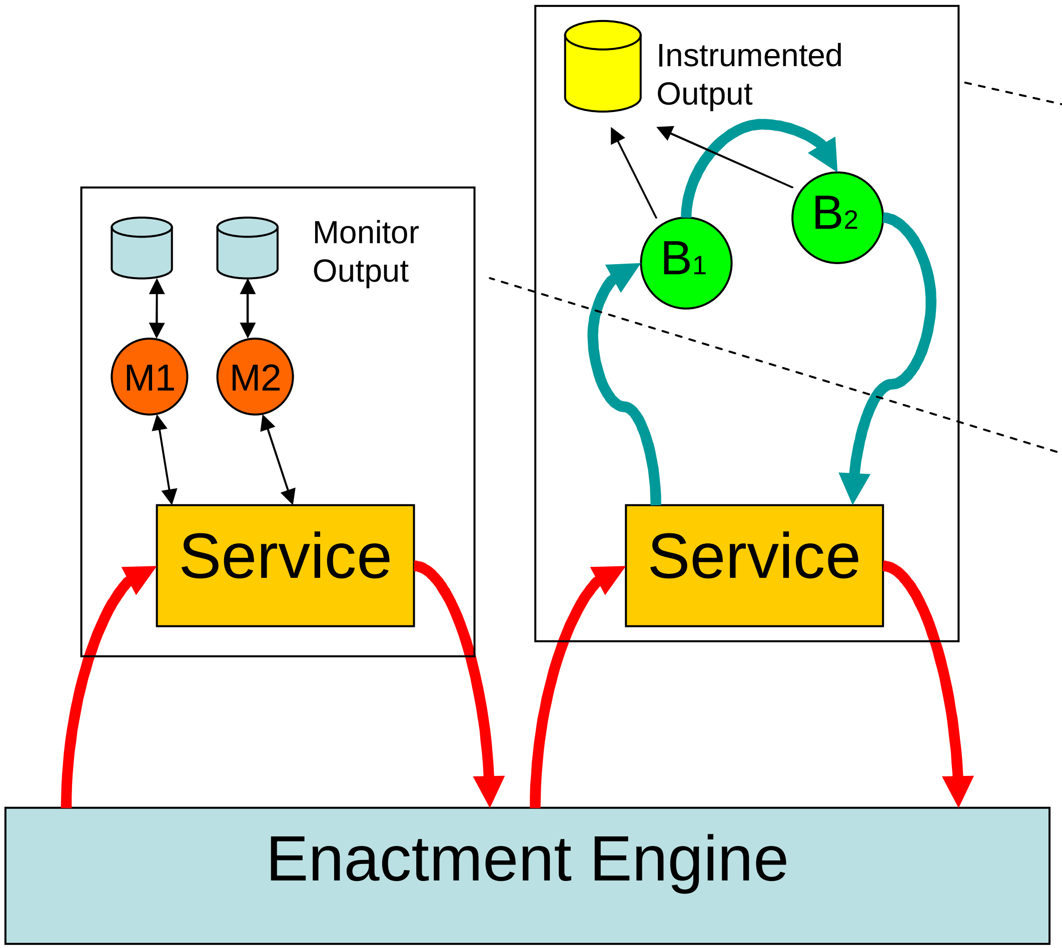
M3 = f1(M1)
M2 = f2(M1,M4)

Types of p-assertions (3)

- Actor state p-assertion: assertion made by an actor about its internal state in the context of a specific interaction



Actor State Capture



Instrumented Actor: Service information obtained from instrumented points within an actor.

Monitoring Sources: Service information derived from hosting platform via monitoring sources (eg Ganglia)

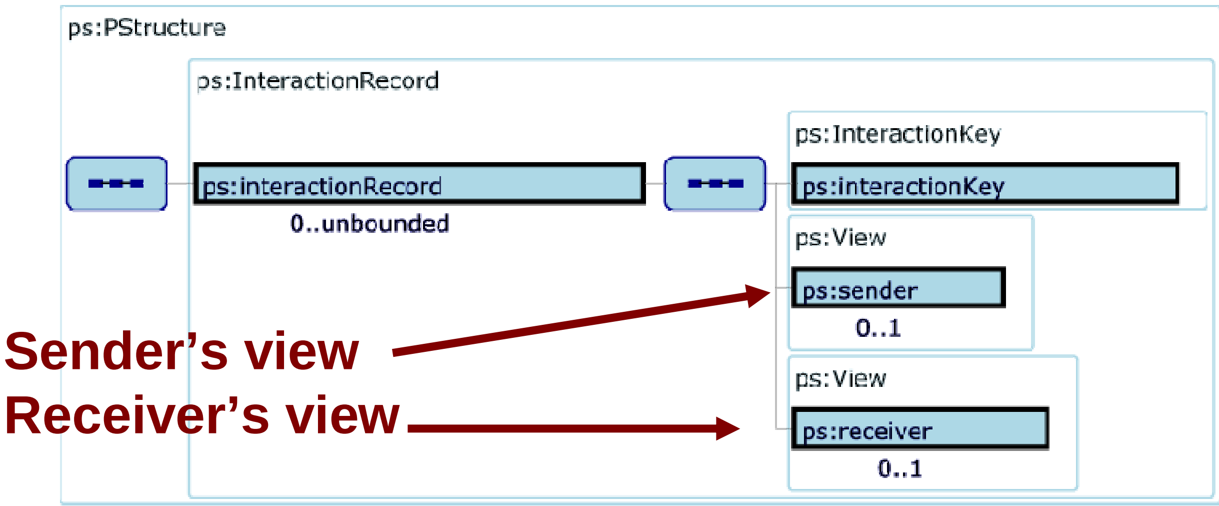
Metrics for Actor State Assertion

- Static
  - No variation in value over actor lifetime
    - Per Node - Node identity, Operating system
    - Per Actor - Actor identity, Name, Owner, Version
- Dynamic
  - Variation in value over actor lifetime
    - Per Node - Memory usage, Network traffic
    - Per Actor - Execution Time, Availability
- Instrumented
  - Actor is 'Instrumented' at Key Points in its Execution
    - Description of internal data flow
  - Eg. Completion states for action events and file transfers

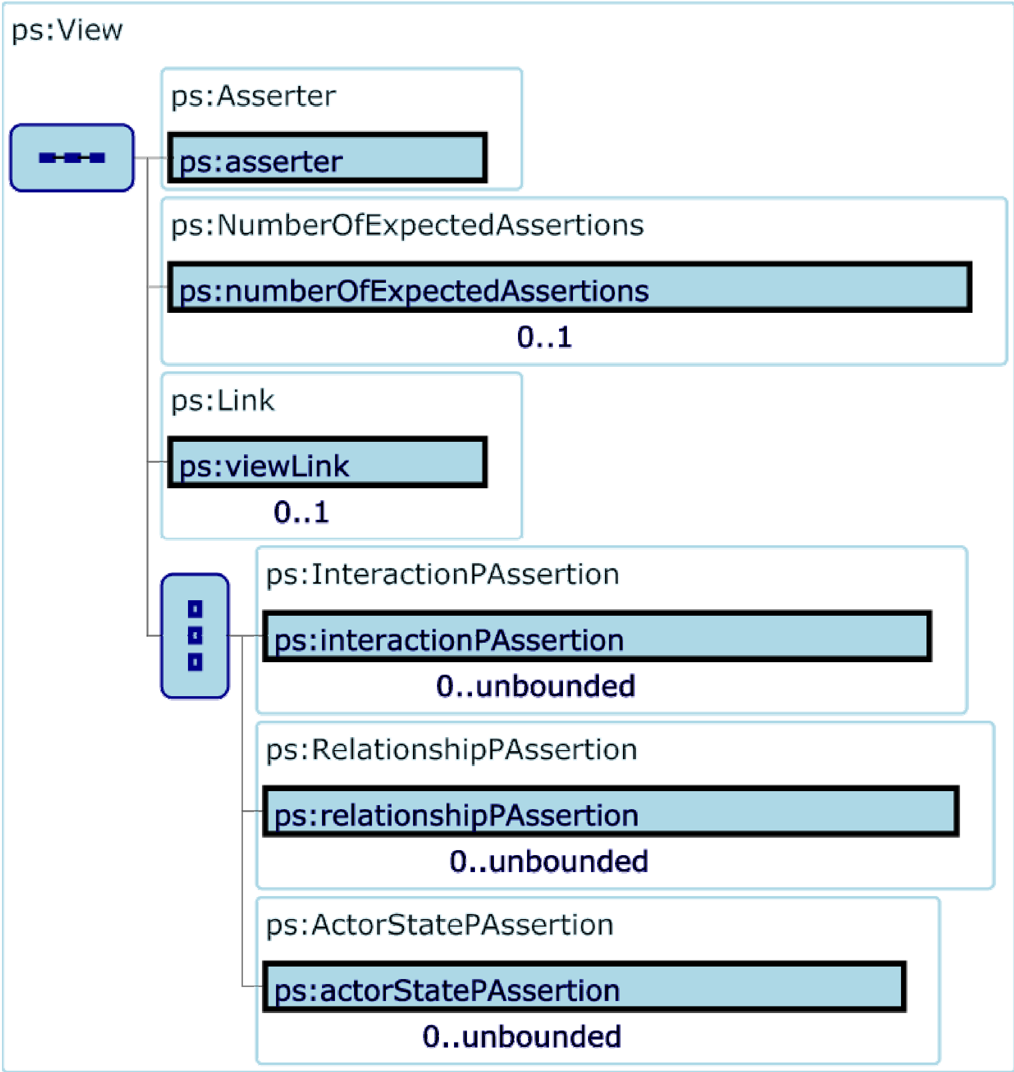
The p-structure (1)

- The p-structure is a common logical structure of the provenance store shared by all asserting and querying actors
- Hierarchical
- Indexed by interactions (interaction= 1 message exchange)

  ➔ Now part of the Open Provenance Model



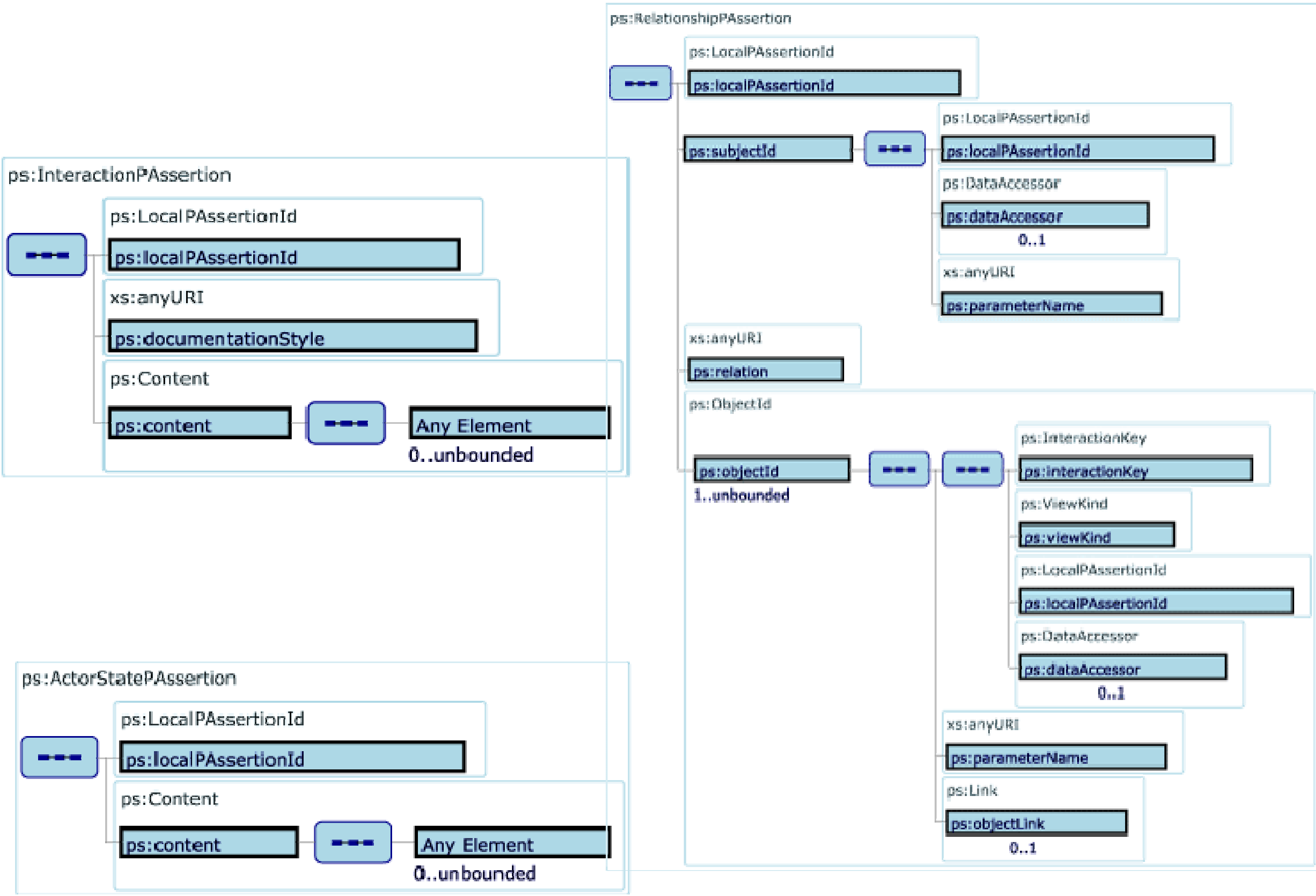
The p-structure (2)



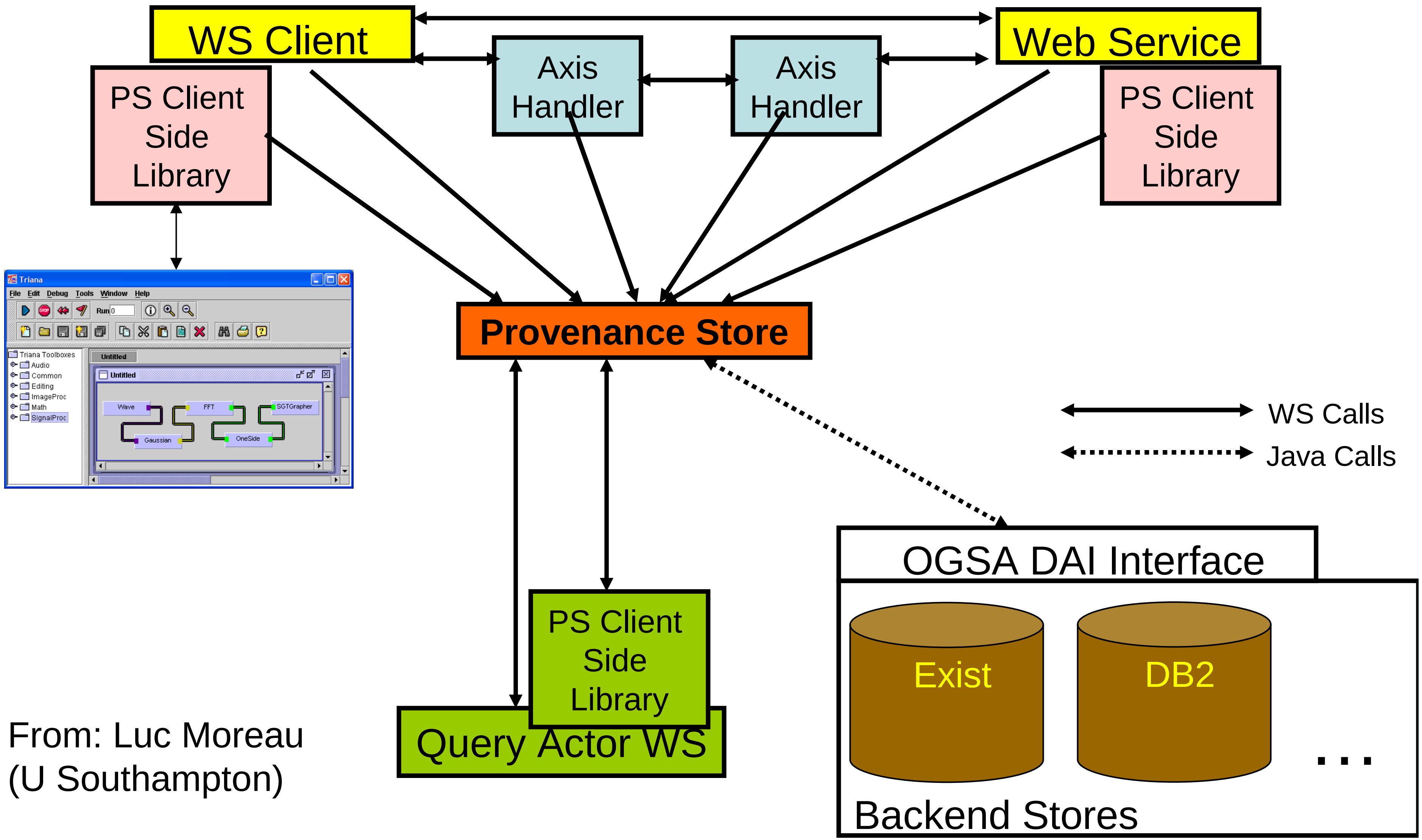
Asserter identity

All p-assertions asserted by a given actor participating in an interaction

P-Assertion schemas

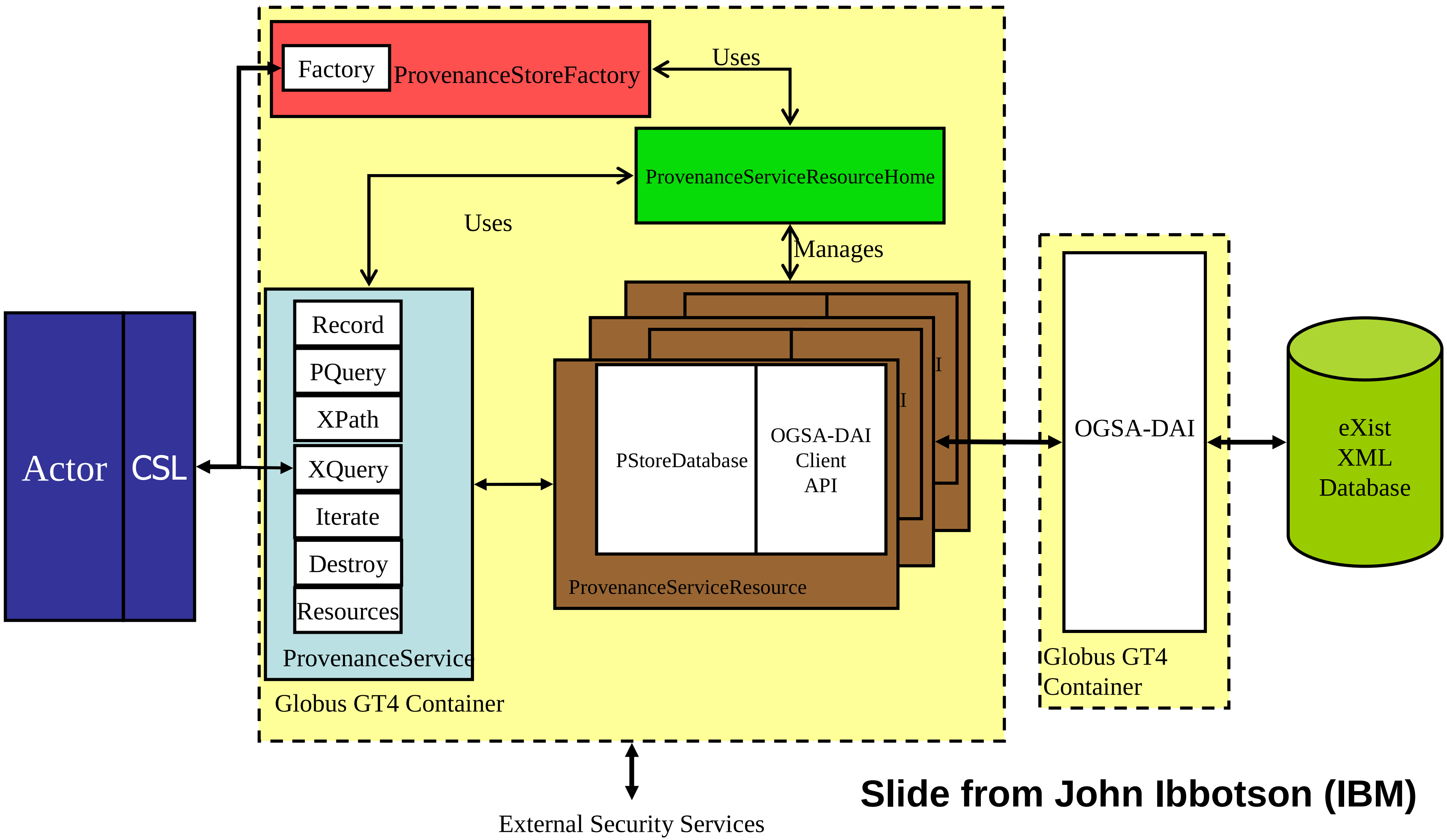


Implementation Diagram



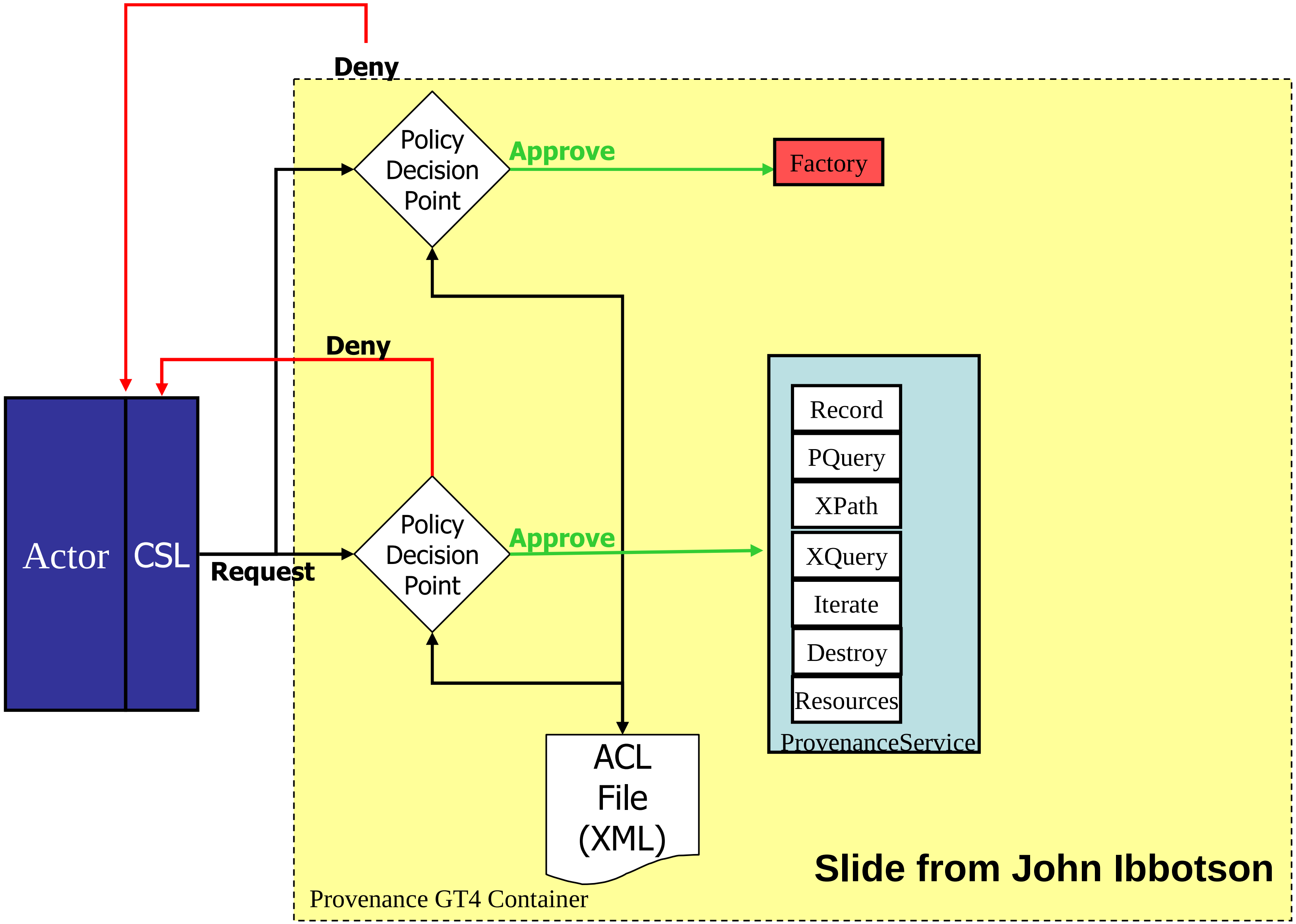
From: Luc Moreau
(U Southampton)

Provenance Store Components



Slide from John Ibbotson (IBM)

Provenance Store Security



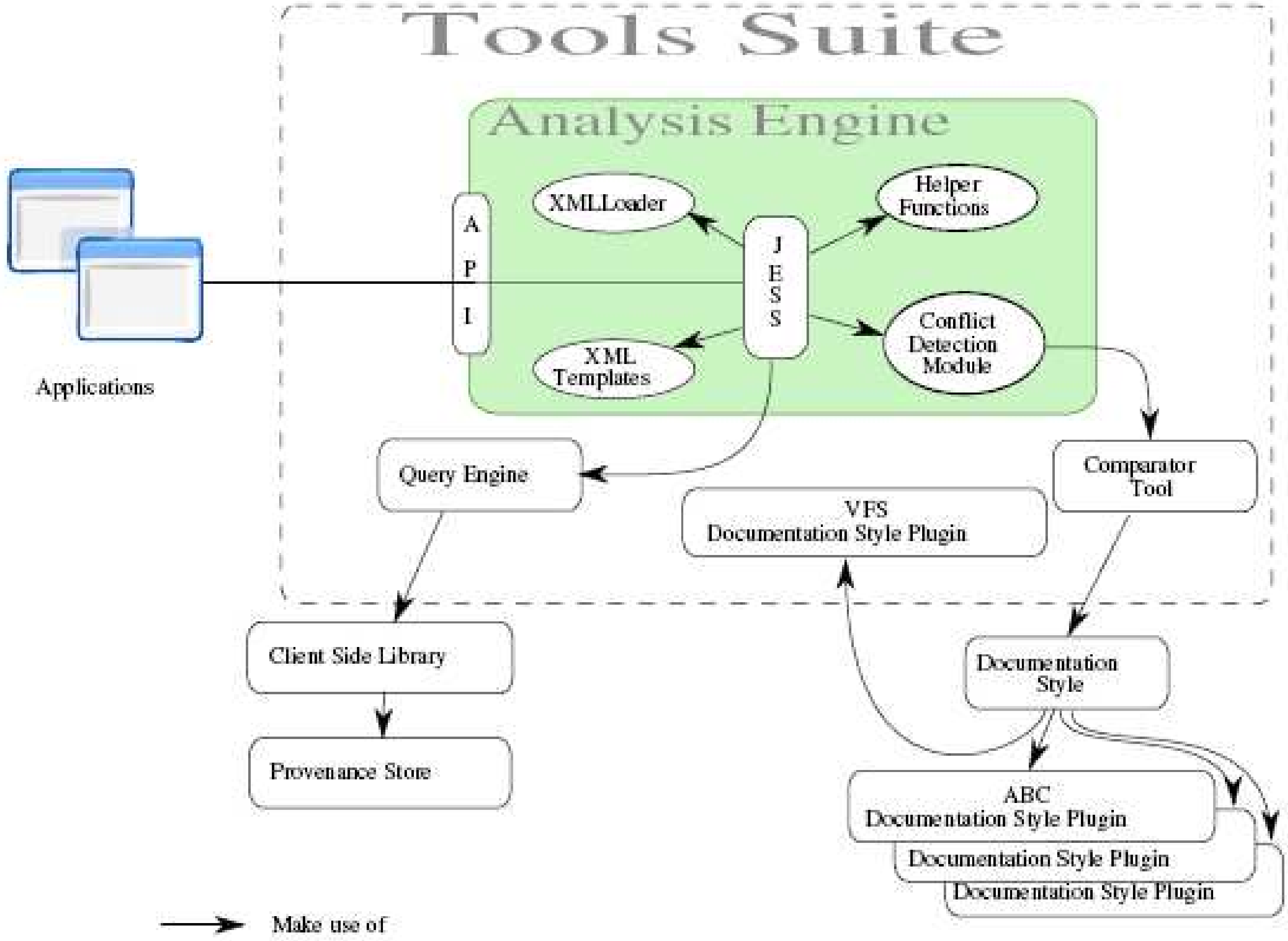
Slide from John Ibbotson

Portal – Tools Architecture

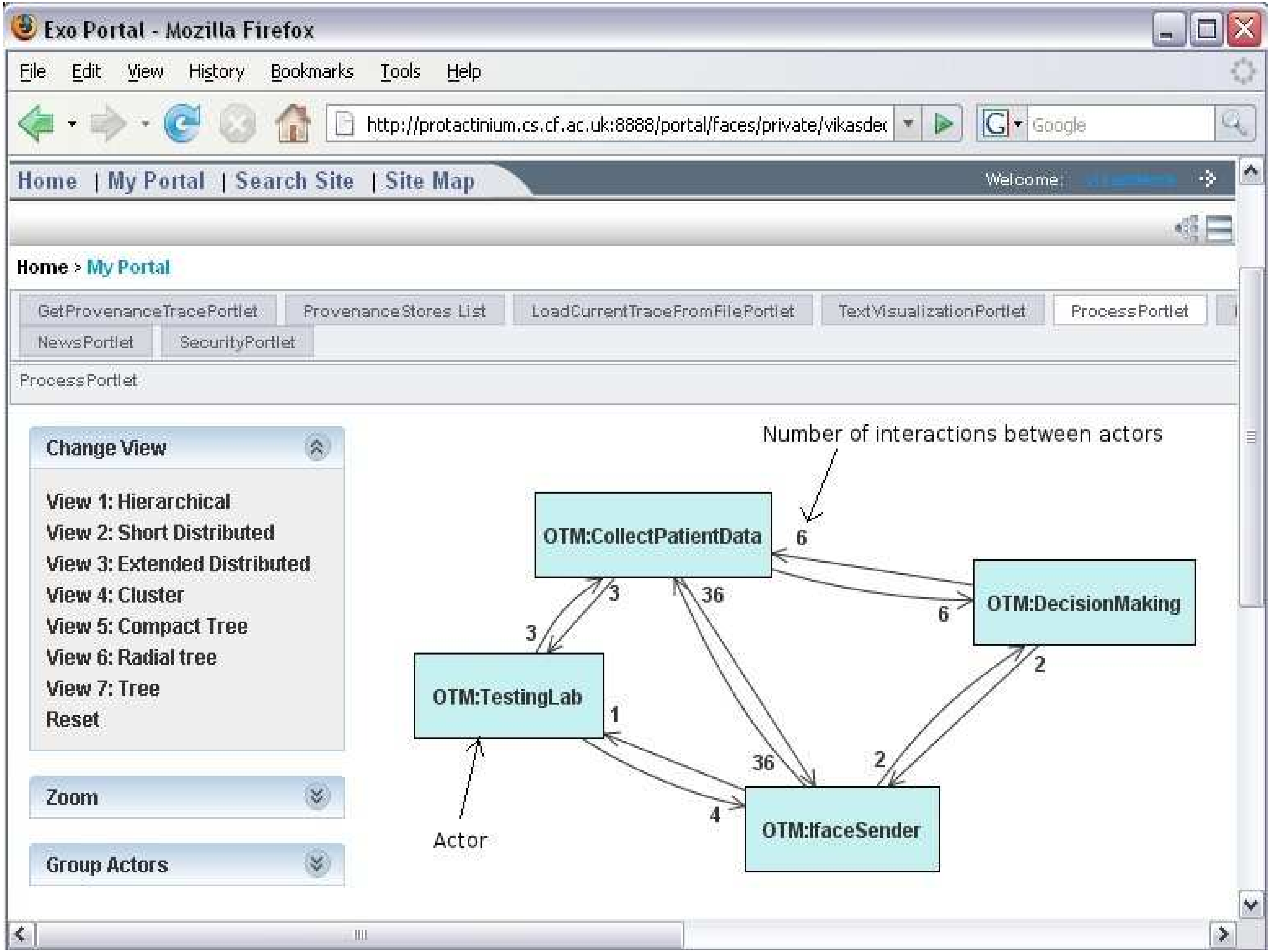
- Tool suite allow users of the tool to navigate and visualize provenance information beyond the capabilities provided by the Client Side Library.
  - **The Visualisation Tools:** these tools provide Graphical User Interfaces (GUI) for visualizing p-assertions that have been submitted by an application.

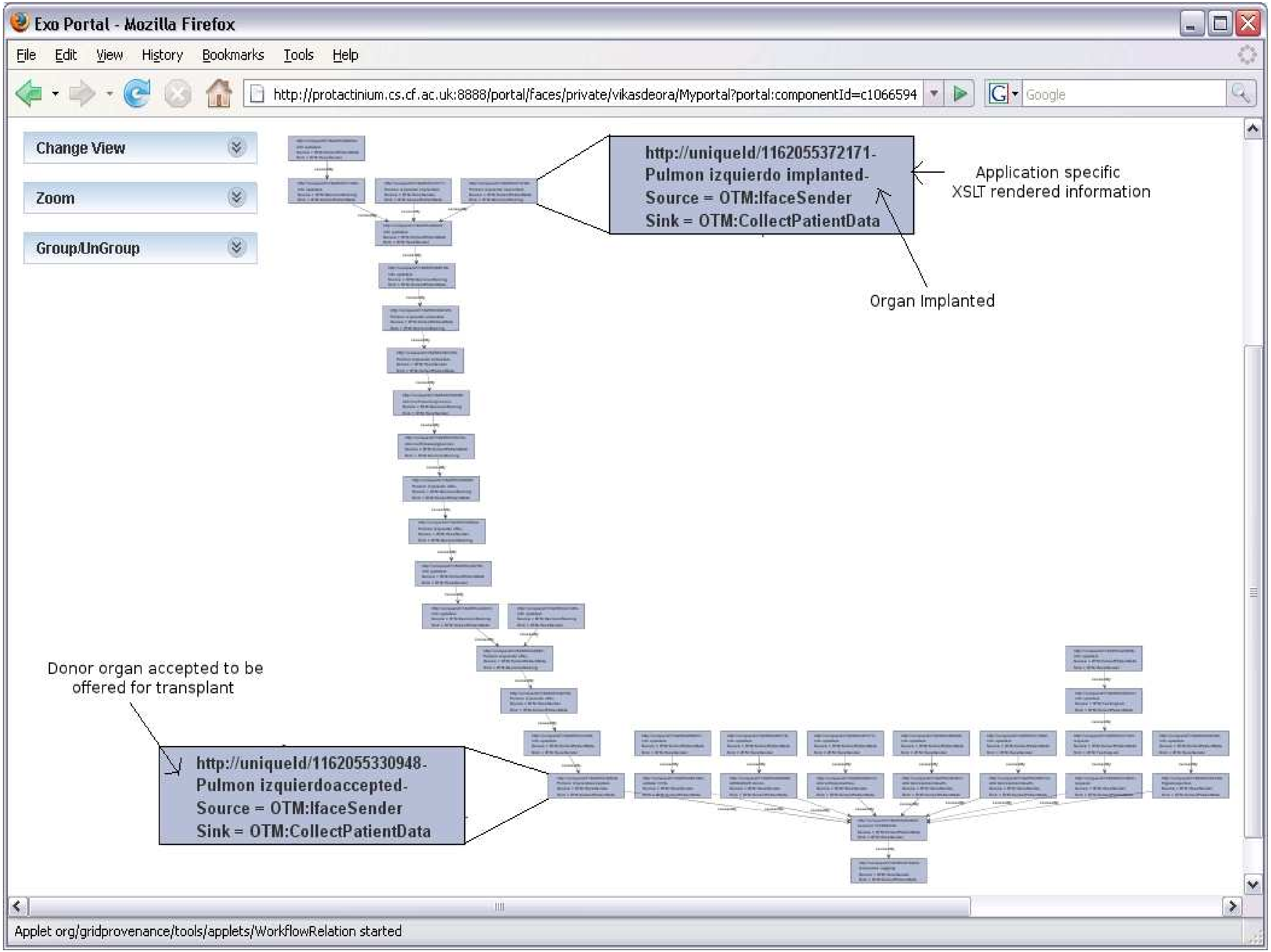
Portal – Tools Architecture

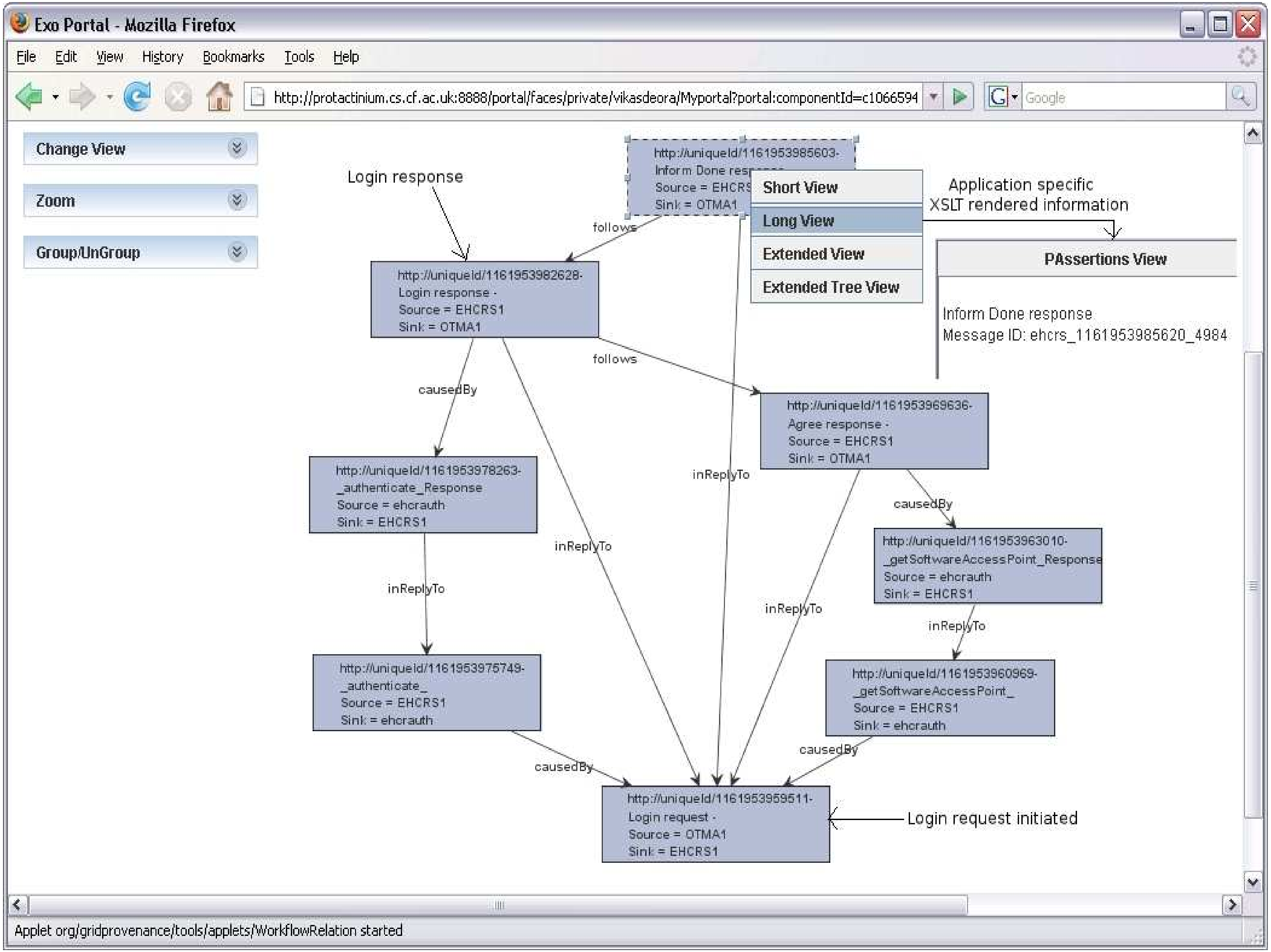
- **The Processing Tools:** provide features accessible through an Application Programming Interface (API). The processing tools include the following:
  - The Analysis Engine provides reasoning capabilities over a set of passertions,
  - The Comparator Tool may be used to compare p-assertions that have been submitted by an application,
  - the Query Tool makes use of the Client Side Library to query one or more Provenance Store(s).



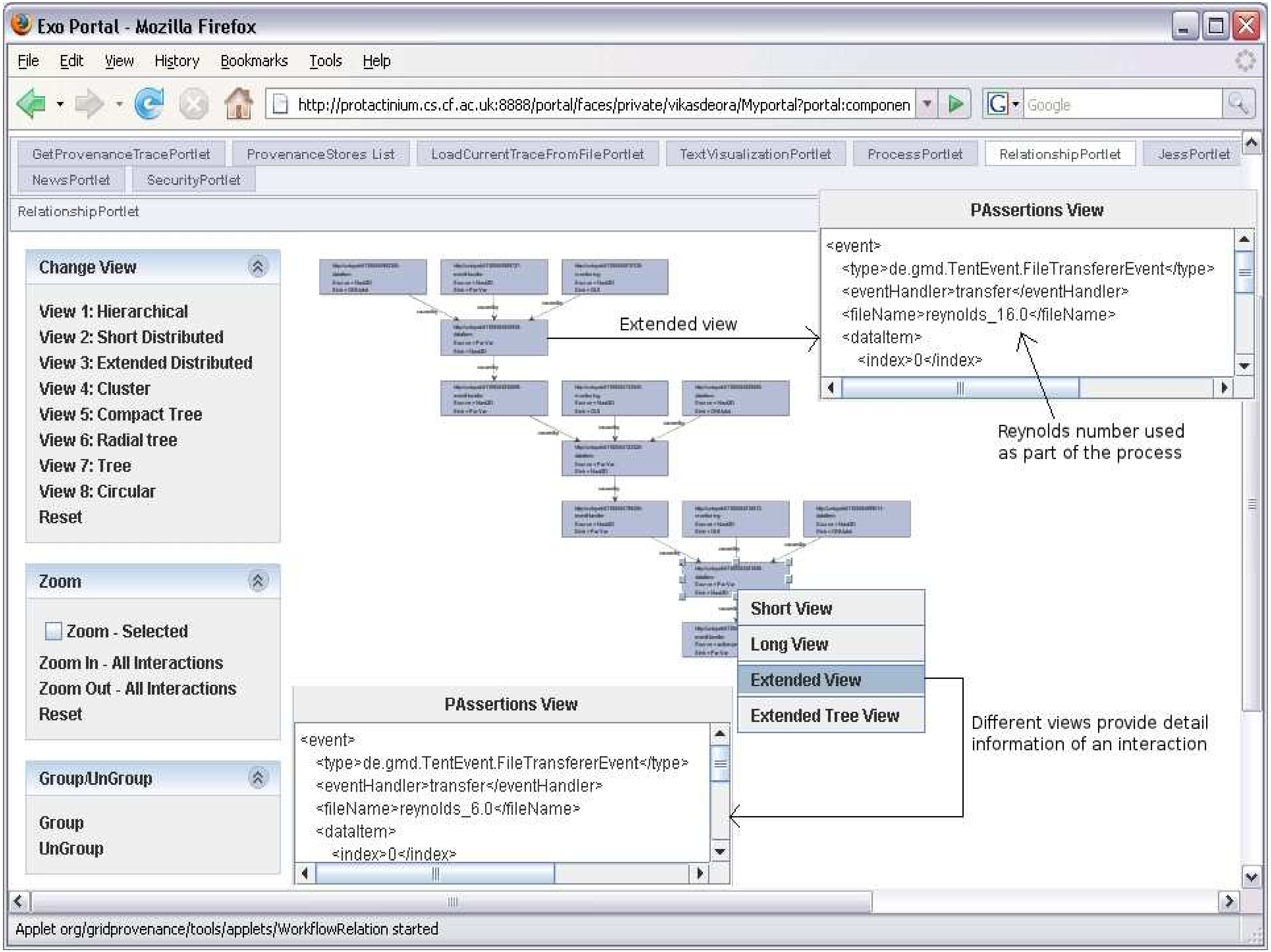
Tools Suite
Analysis Engine
A P I
XMLLoader
Helper Functions
J E S S
Conflict Detection Module
XML Templates
Applications
Query Engine
Comparator Tool
VFS Documentation Style Plugin
Client Side Library
Documentation Style
Provenance Store
ABC Documentation Style Plugin
Documentation Style Plugin
Documentation Style Plugin
Make use of


Exo Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://protactinium.cs.cf.ac.uk:8888/portal/faces/private/vikasdec
Google
Home | My Portal | Search Site | Site Map
Welcome:
Home > My Portal
GetProvenanceTracePortlet
ProvenanceStores List
LoadCurrentTraceFromFilePortlet
TextVisualizationPortlet
ProcessPortlet
NewsPortlet
SecurityPortlet
ProcessPortlet
Change View
View 1: Hierarchical
View 2: Short Distributed
View 3: Extended Distributed
View 4: Cluster
View 5: Compact Tree
View 6: Radial tree
View 7: Tree
Reset
Zoom
Group Actors
Number of interactions between actors
OTM:CollectPatientData
OTM:DecisionMaking
OTM:TestingLab
OTM:IfaceSender
Actor
6
6
3
3
36
36
2
2
1
4

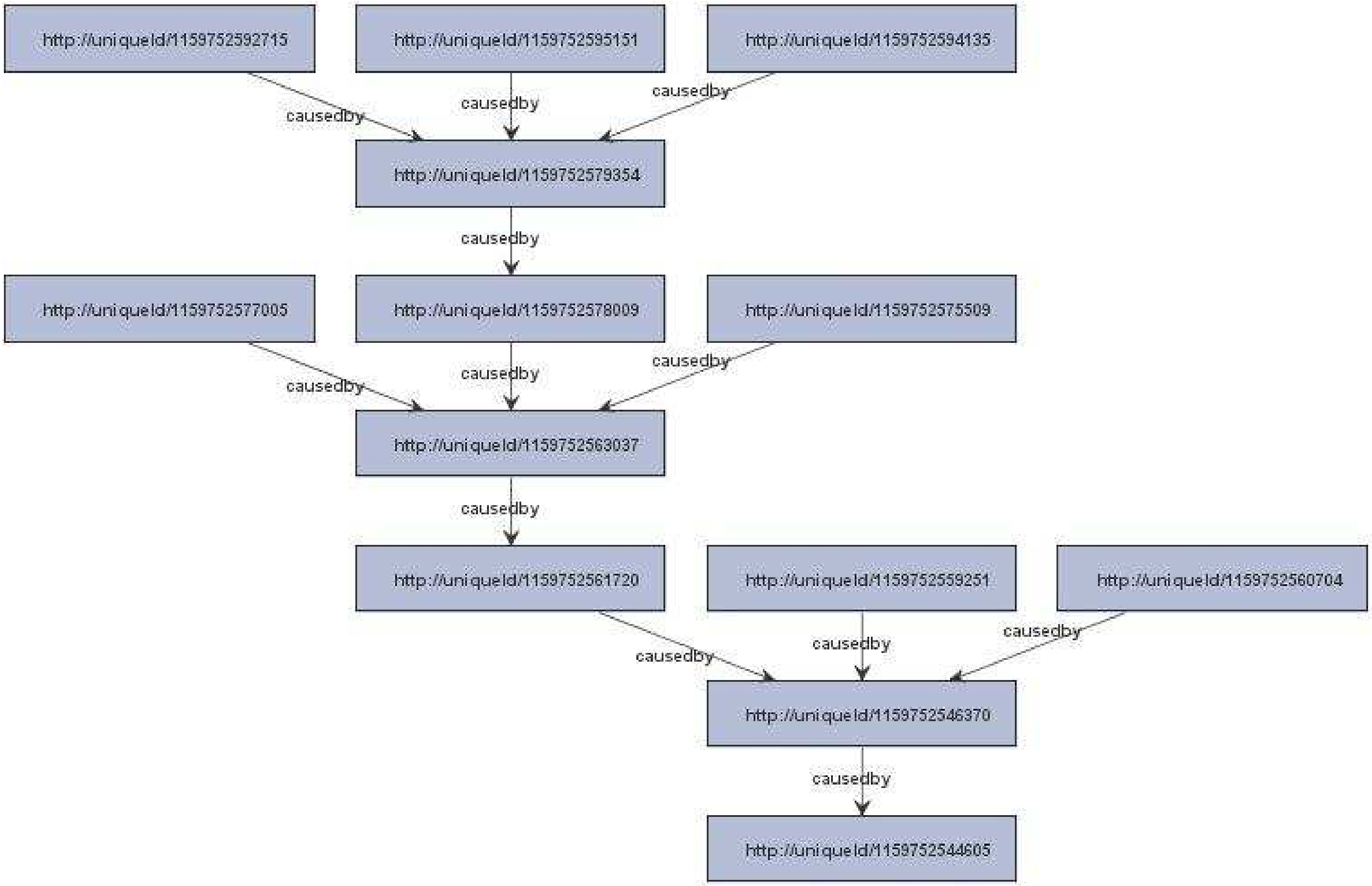

Exo Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://protactinium.cs.cf.ac.uk:8888/portal/faces/private/vikasdeora/Myportal?portal:componentId=c1066594
Google
Change View
Zoom
Group/UnGroup
http://uniqueId/1162055372171-
Pulmon izquierdo implanted-
Source = OTM:IfaceSender
Sink = OTM:CollectPatientData
Application specific
XSLT rendered information
Organ Implanted
Donor organ accepted to be
offered for transplant
http://uniqueId/1162055330948-
Pulmon izquierdoaccepted-
Source = OTM:IfaceSender
Sink = OTM:CollectPatientData
Applet org/gridprovenance/tools/applets/WorkflowRelation started



Exo Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://protactinium.cs.cf.ac.uk:8888/portal/faces/private/vikasdeora/Myportal?portal:componentId=c1066594
Google
Change View
Zoom
Group/UnGroup
Login response
http://uniqueId/1161953985603-
Inform Done res
Source = EHCRS
Sink = OTMA1
Short View
Long View
Extended View
Extended Tree View
Application specific
XSLT rendered information
PAssertions View
Inform Done response
Message ID: ehcrs_1161953985620_4984
follows
http://uniqueId/1161953982628-
Login response -
Source = EHCRS1
Sink = OTMA1
follows
causedBy
http://uniqueId/1161953969636-
Agree response -
Source = EHCRS1
Sink = OTMA1
http://uniqueId/1161953978263-
_authenticate_Response
Source = ehcrauth
Sink = EHCRS1
inReplyTo
inReplyTo
causedBy
http://uniqueId/1161953963010-
_getSoftwareAccessPoint_Response
Source = ehcrauth
Sink = EHCRS1
inReplyTo
inReplyTo
inReplyTo
http://uniqueId/1161953975749-
authenticate
Source = EHCRS1
Sink = ehcrauth
http://uniqueId/1161953960969-
getSoftwareAccessPoint
Source = EHCRS1
Sink = ehcrauth
causedBy
causedBy
http://uniqueId/1161953959511-
Login request -
Source = OTMA1
Sink = EHCRS1
Login request initiated
Applet org/gridprovenance/tools/applets/WorkflowRelation started


Exo Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://protactinium.cs.cf.ac.uk:8888/portal/faces/private/vikasdeora/Myportal?portal:componen
Google
GetProvenanceTracePortlet
ProvenanceStores List
LoadCurrentTraceFromFilePortlet
TextVisualizationPortlet
ProcessPortlet
RelationshipPortlet
JessPortlet
NewsPortlet
SecurityPortlet
RelationshipPortlet
Change View
View 1: Hierarchical
View 2: Short Distributed
View 3: Extended Distributed
View 4: Cluster
View 5: Compact Tree
View 6: Radial tree
View 7: Tree
View 8: Circular
Reset
Zoom
Zoom - Selected
Zoom In - All Interactions
Zoom Out - All Interactions
Reset
Group/UnGroup
Group
UnGroup
Extended view
PAssertions View
<event>
<type>de.gmd.TentEvent.FileTransfererEvent</type>
<eventHandler>transfer</eventHandler>
<fileName>reynolds_16.0</fileName>
<dataItem>
<index>0</index>
Reynolds number used
as part of the process
Short View
Long View
Extended View
Extended Tree View
Different views provide detail
information of an interaction
PAssertions View
<event>
<type>de.gmd.TentEvent.FileTransfererEvent</type>
<eventHandler>transfer</eventHandler>
<fileName>reynolds_6.0</fileName>
<dataItem>
<index>0</index>
Applet org/gridprovenance/tools/applets/WorkflowRelation started

Relationship Tool



References

- “A Taxonomy of Workflow Management Systems for Grid Computing”, Jia Yu and Rajkumar Buyya, GRIDS Lab, University of Melbourne
- “Examining the Challenges of Scientific Workflows”, Y Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau and J. Myers, IEEE Computer, December 2007, pp 26—34
- “Putting Semantics in Grid Workflow Management: the OWL-WS approach”, Stefano Beco, Barbara Cantalupo, Nikolaos Matskanis, Mike Surridge, EU NextGRID paper
- “Planning and Monitoring Web Service Composition”, M. Pistore, F. Barbon, P. Bertoli, D. Shaparau, and P. Traverso (ITC/IRST, Italy)

References … 2

- G. Fox and D. Gennon, Special Issue "Workflow Systems", Concurrency and Computation: Practice and Experience, 18(10): 1009-1019. 2006
- Ewa Deelman, Dennis Gannon, Matthew Shields and Ian Taylor, "Workflows and e-Science: An overview of Workflow System Features and Capabilities", FGCS, 2008
- Y. Gil et al., "Wings for Pegasus: Creating Large-Scale Scientific Applications Using Semantic Representations of Computational Workflows", 19th IAAI Conference, Vancouver, BC, Canada, June 2007
- **I. J. Taylor, E. Deelman, D. Gannon and M. Shields (Eds), Workflows for eScience, Springer 2007**