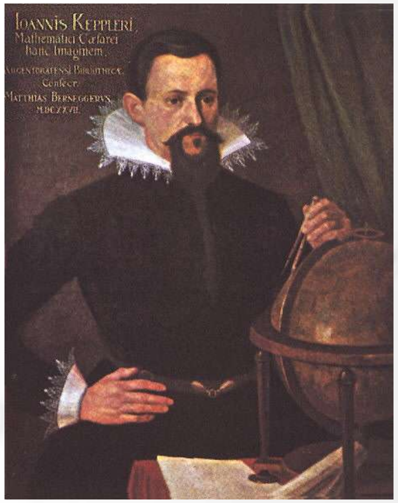
KEPLER Scientific Workflow System



Bertram Ludäscher

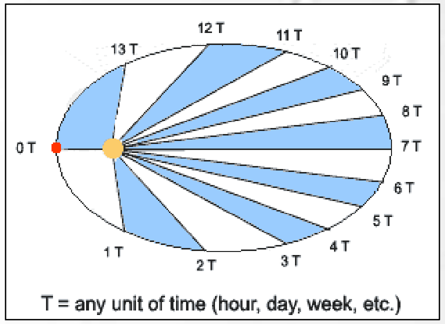
Knowledge-Based Information Systems Lab

San Diego Supercomputer Center

&

Dept. of Computer Science & Engineering

University of California, San Diego



GRIST Workshop, July 13-14, 2004, Caltech

Overview

- *Motivation/Examples: Scientific Workflows*
- *Ptolemy II Goodies*
- *Technical Issues and KEPLER extensions*
- *Ongoing and future plans*
- *Getting Involved*

Why Web Services are so important!

- *??? (beats me …)*
- *Never mind …*
- *What you probably really care about:*

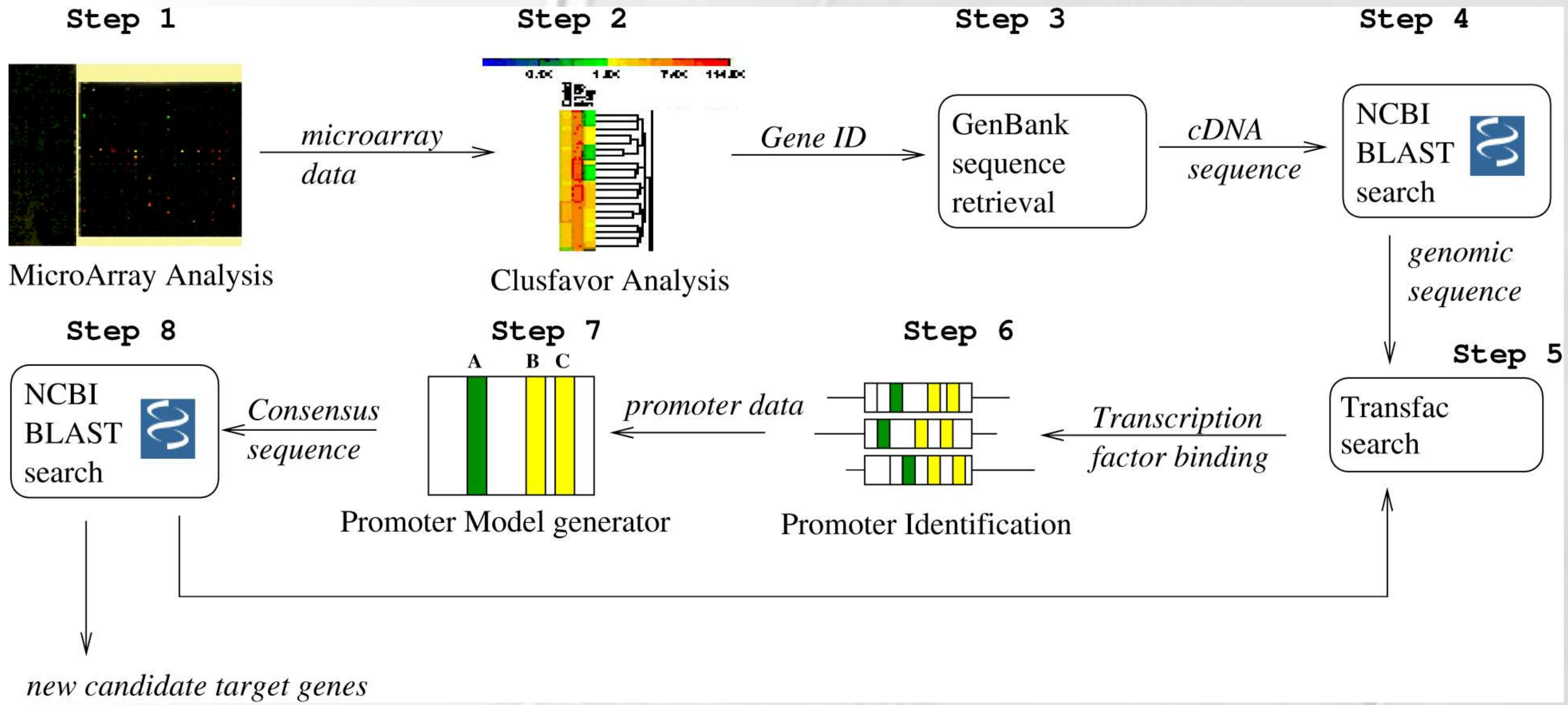
- *How to design, annotate, plan, query, schedule, optimize, execute, monitor, reuse, share, archive, …*

… **Scientific Workflows!**

(and the data that goes with them)

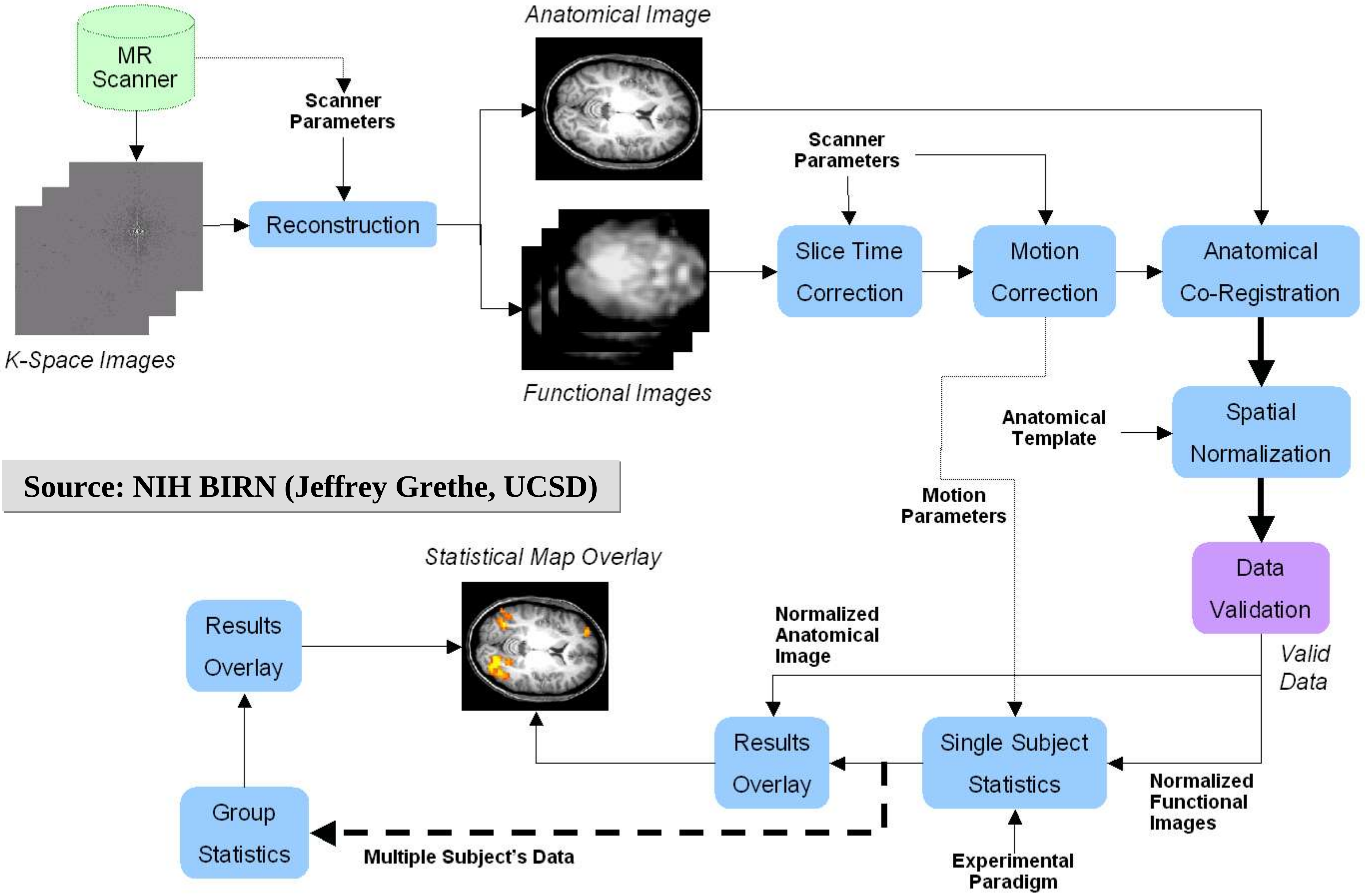
- *aka:* **Getting the job (science) done!**

Promoter Identification Workflow (PIW)



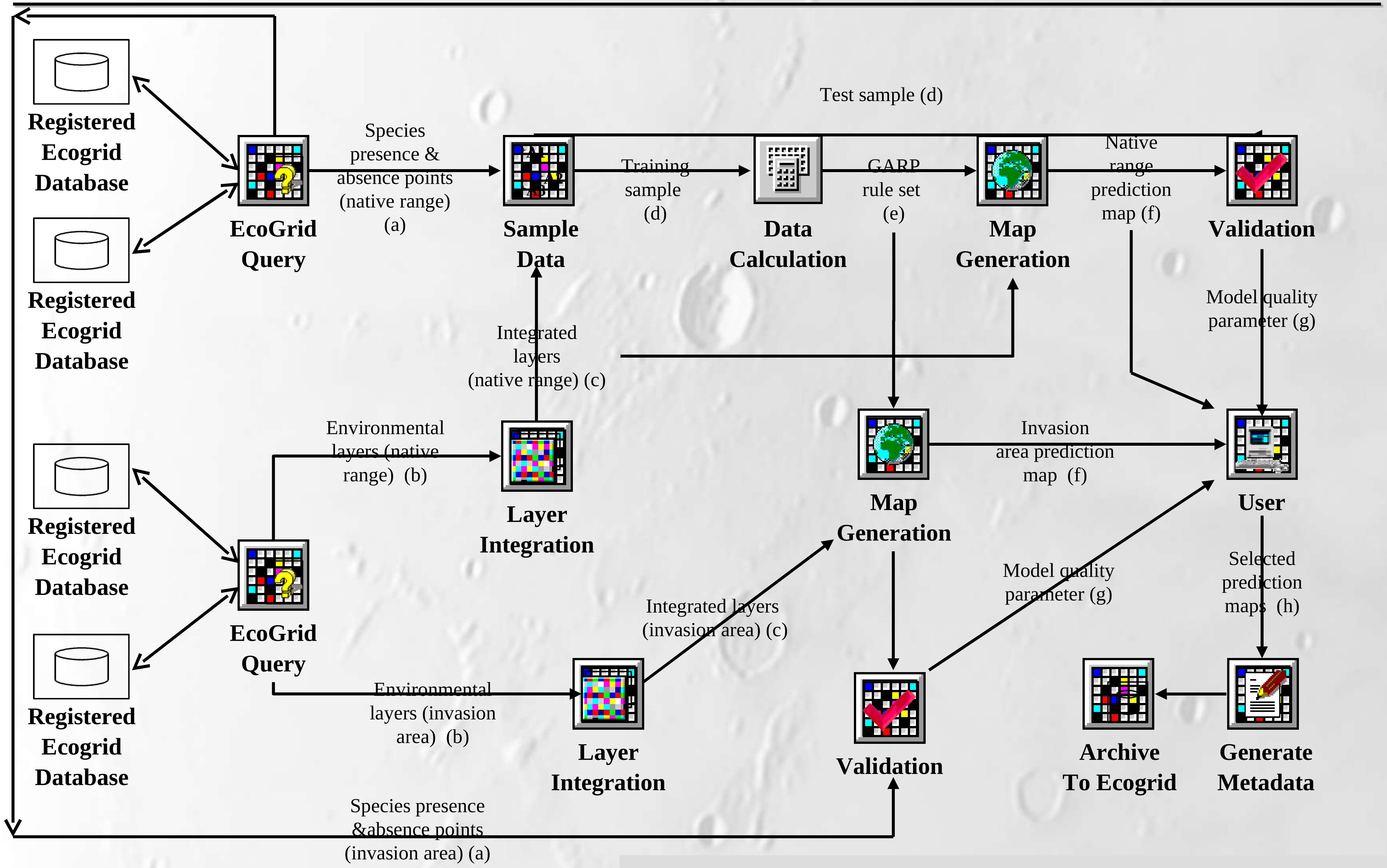
Source: Matt Coleman (LLNL)

Functional MRI Analysis Workflow



Source: NIH BIRN (Jeffrey Grethe, UCSD)

Ecology: GARP Analysis Pipeline for Invasive Species Prediction



Registered Ecogrid Database
Registered Ecogrid Database
EcoGrid Query
Species presence & absence points (native range) (a)
Sample Data
Test sample (d)
Training sample (d)
Data Calculation
GARP rule set (e)
Map Generation
Native range prediction map (f)
Validation
Model quality parameter (g)
Integrated layers (native range) (c)
Environmental layers (native range) (b)
Layer Integration
Registered Ecogrid Database
Registered Ecogrid Database
EcoGrid Query
Map Generation
Invasion area prediction map (f)
User
Model quality parameter (g)
Selected prediction maps (h)
Integrated layers (invasion area) (c)
Environmental layers (invasion area) (b)
Layer Integration
Validation
Archive To Ecogrid
Generate Metadata
Species presence &absence points (invasion area) (a)

Source: NSF SEEK (Deana Pennington et. al, UNM)

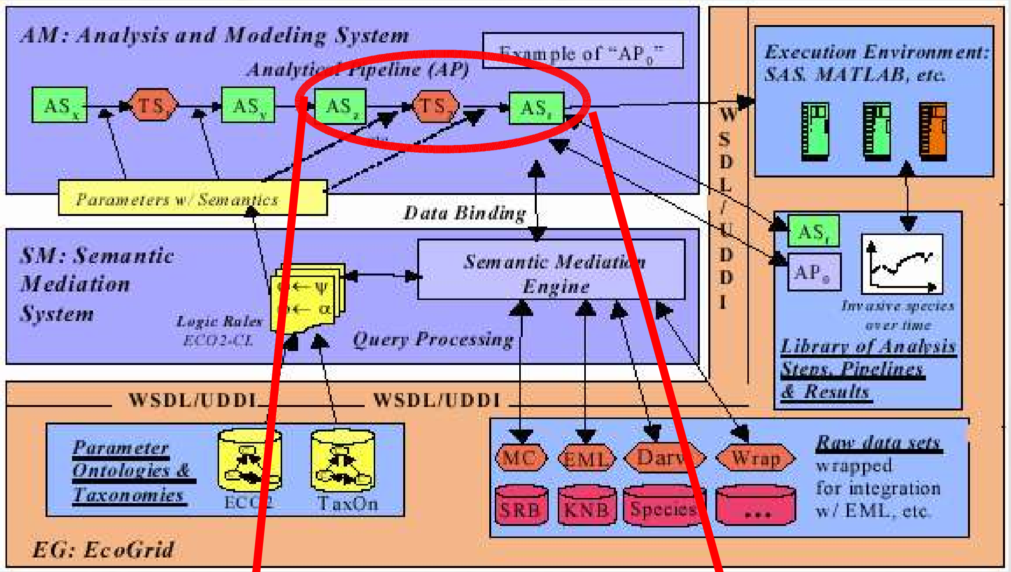
B. Ludäscher, SDSC

NSF/ITR Science Environment for Ecological Knowledge

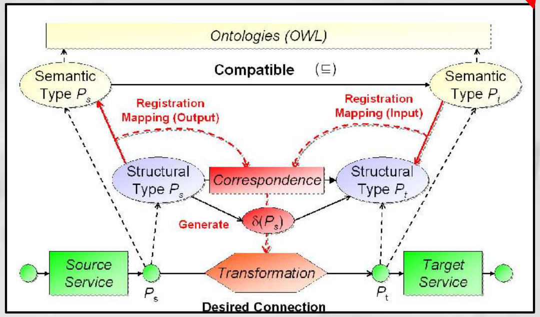
- **Domain Science Driver**
  - *Ecology (LTER), biodiversity, …*
- **Analysis & Modeling System**
  - *Design & execution of ecological models & analysis*
  - *End (&power) user focus*
  - **{application,upper}-ware**

  ➔ **KEPLER system**
- **Semantic Mediation System**
  - *Data Integration of hard-to-relate sources and processes*
  - *Semantic Types and Ontologies*
  - **upper middleware**

  ➔ **SPARROW toolkit**
- **EcoGrid**
  - *Access to ecology data and tools*
  - **{middle,under}-ware**


AM: Analysis and Modeling System
Example of "AP0"
Analytical Pipeline (AP)
ASx TSx ASy ASz TSy ASr
Parameters w/ Semantics
Data Binding
Execution Environment: SAS, MATLAB, etc.
WSDL/UDDI
SM: Semantic Mediation System
Logic Rules ECO2-CL
Semantic Mediation Engine
Query Processing
ASr APo
Invasive species over time
Library of Analysis Steps, Pipelines & Results
WSDL/UDDI
WSDL/UDDI
Parameter Ontologies & Taxonomies
ECO2
TaxOn
MC EML Darw Wrap
SRB KNB Species ...
Raw data sets wrapped for integration w/ EML, etc.
EG: EcoGrid

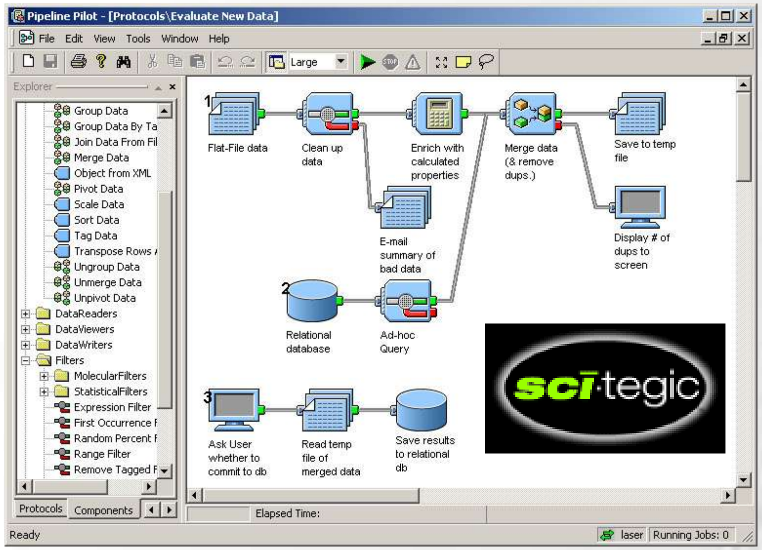
SEEK Architecture

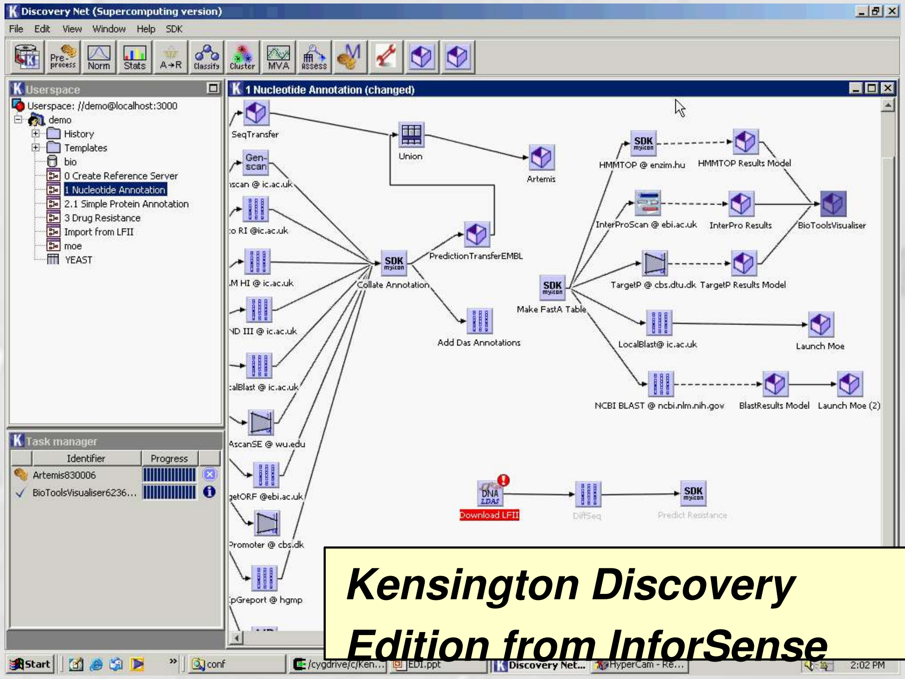


one specific problem [DILS'04]

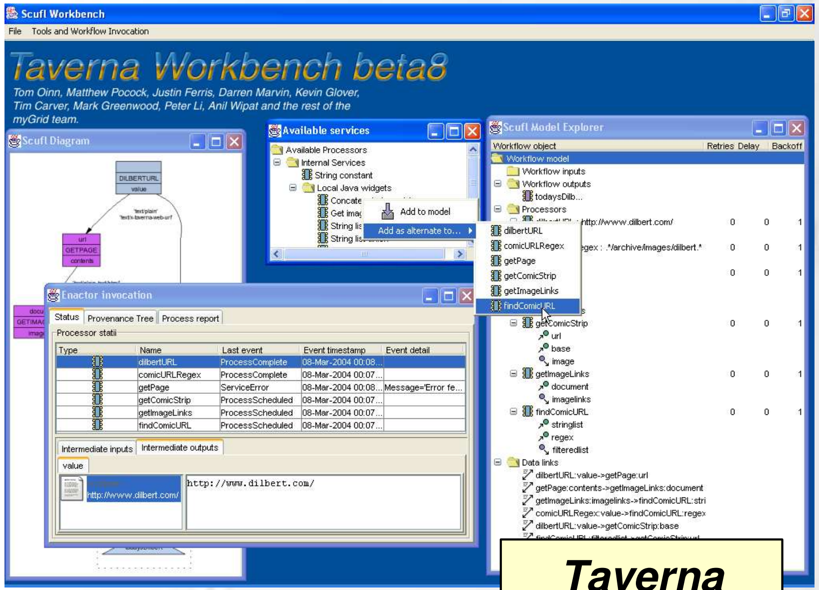
Commercial & Open Source Scientific "Workflow" (well **Dataflow***) Systems*

Kepler





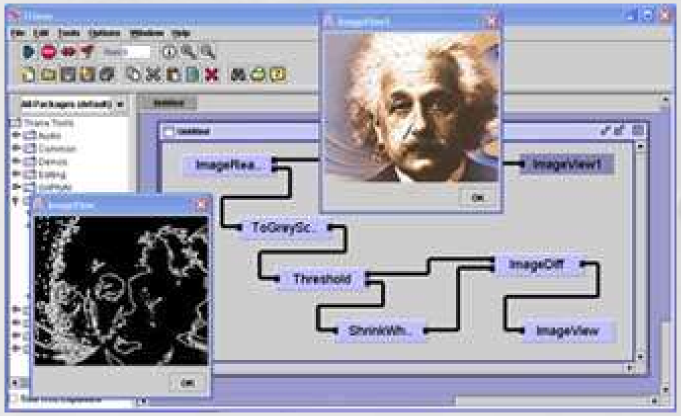
Kensington Discovery Edition from InforSense



Type	Name	Last event	Event timestamp	Event detail
	dilbertURL	ProcessComplete	08-Mar-2004 00:08...	
	comicURLRegex	ProcessComplete	08-Mar-2004 00:07...	
	getPage	ServiceError	08-Mar-2004 00:08...	Message='Error fe...
	getComicStrip	ProcessScheduled	08-Mar-2004 00:07...	
	getImageLinks	ProcessScheduled	08-Mar-2004 00:07...	
	findComicURL	ProcessScheduled	08-Mar-2004 00:07...	

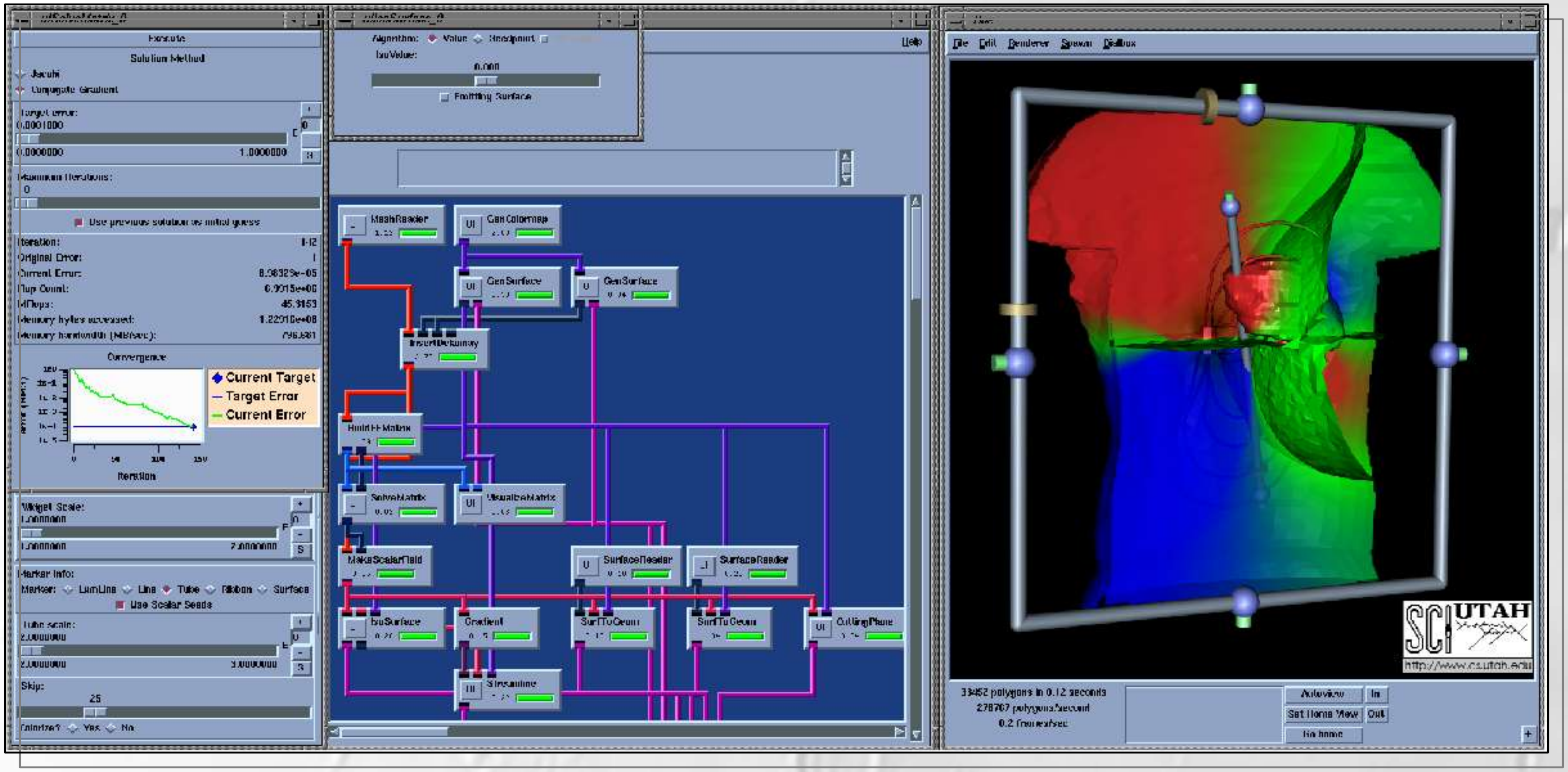
```
http://www.dilbert.com/
```

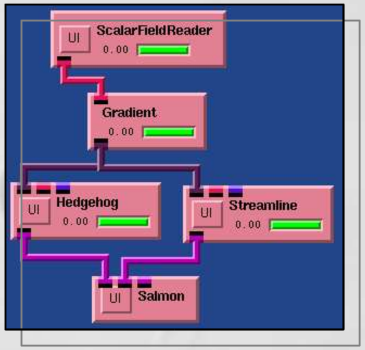
Taverna



Triana

SCIRun: Problem Solving Environments for Large-Scale Scientific Computing

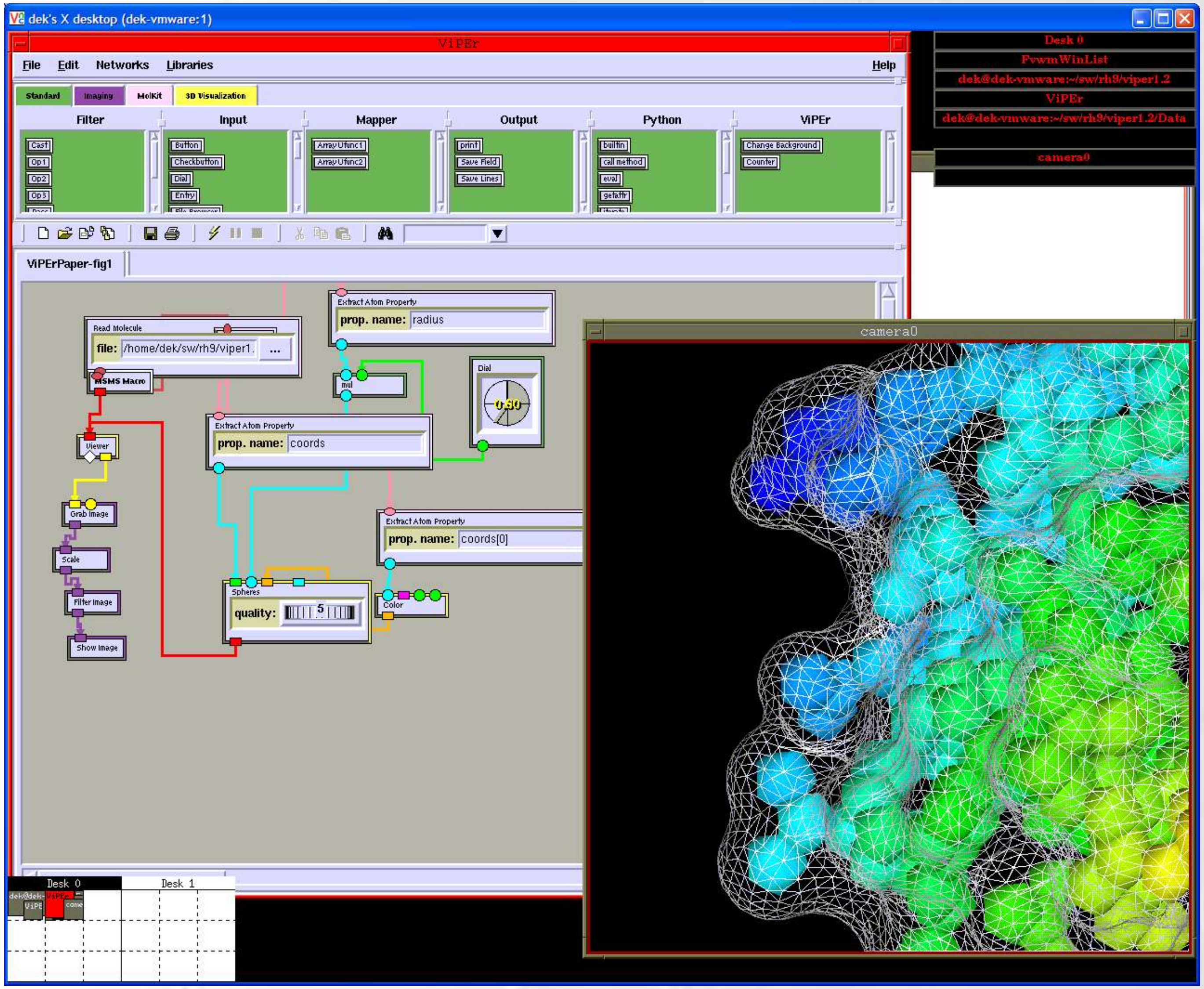




- *SCIRun: PSE for interactive construction, debugging, and steering of large-scale scientific computations*
- *Component model, based on generalized dataflow programming*

Steve Parker (cs.utah.edu)

Viper/Vision/VIPUS



dek's X desktop (dek-vmware:1)
ViPEr
File Edit Networks Libraries Help
Standard Imaging MolKit 3D Visualization
Filter Input Mapper Output Python ViPEr
Cast Op1 Op2 Op3
Button Checkbutton Dial Entry
Array Ufunc1 Array Ufunc2
print Save Field Save Lines
builtin call method eval getattr
Change Background Counter
ViPErPaper-fig1
Read Molecule
file: /home/dek/sw/rh9/viper1.
MSMS Macro
Viewer
Grab Image
Scale
Filter Image
Show Image
Extract Atom Property
prop. name: radius
mul
Dial
0.60
Extract Atom Property
prop. name: coords
Extract Atom Property
prop. name: coords[0]
Spheres
quality: 5
Color
camera0
Desk 0
FvwmWinList
dek@dek-vmware:~/sw/rh9/viper1.2
ViPEr
dek@dek-vmware:~/sw/rh9/viper1.2/Data
camera0
Desk 0
Desk 1

Keith Jackson, David Konerding, Michel Sanner

Scientific "Workflows": Some Findings

- More dataflow than (business control-/) workflow
  - DiscoveryNet, Kepler, SCIRun, Scitegic, Triana, Taverna, …,
- Need for "programming extensions"
  - Iterations over lists (foreach); filtering; functional composition; generic & higher-order operations (zip, map (f), …)
- Need for abstraction and nested workflows
- Need for data transformations (WS1➔DT➔WS2)
- Need for rich user interaction & workflow steering:
  - pause / revise / resume
  - select & branch; e.g., web browser capability at specific steps as part of a coordinated SWF
- Need for high-throughput data transfers and CPU cyles: "(Data-)Grid-enabling", "streaming"
- Need for persistence of intermediate products and provenance

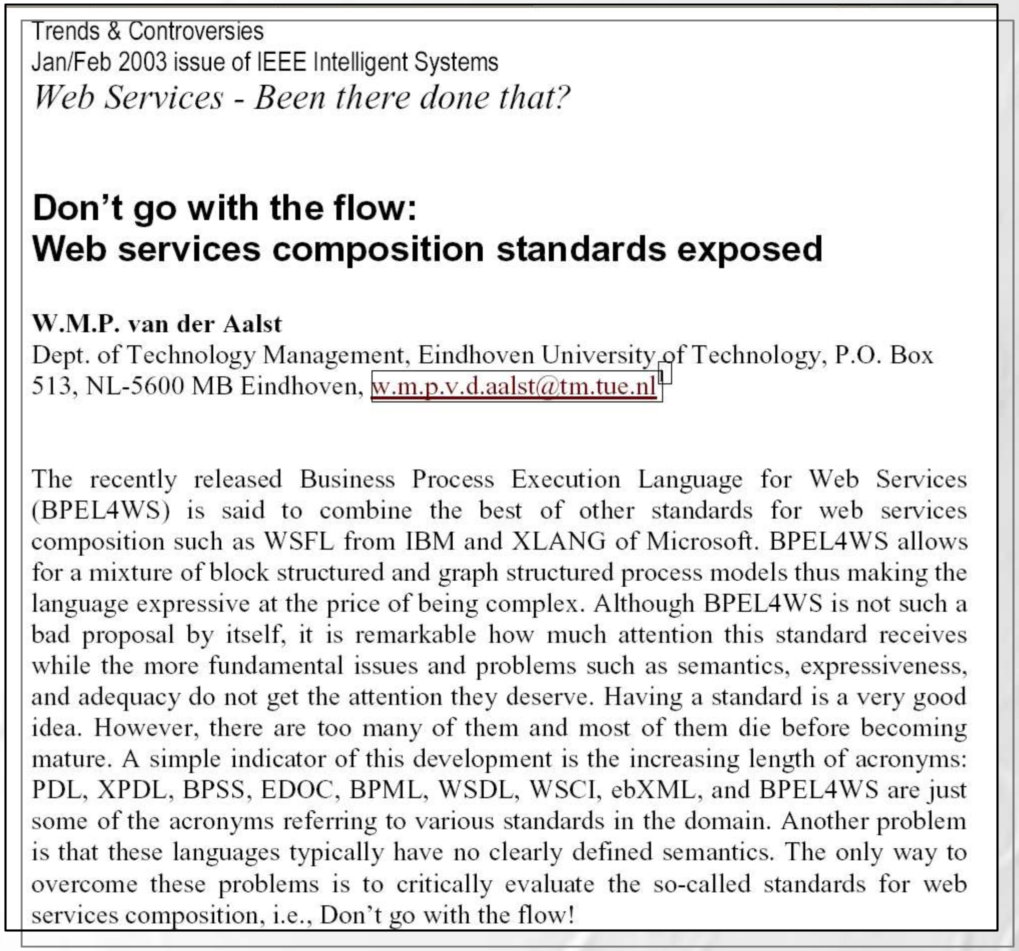
Scientific "Workflows" vs Business Workflows

- Scientific "Workflows"
  - Dataflow and data transformations
  - Data problems: volume, complexity, heterogeneity
  - Grid-aspects
    - Distributed computation
    - Distributed data
  - User-interactions/WF steering
  - Data, tool, and analysis integration

  ➔ Dataflow and control-flow are often **married!**
- Business Workflows (BPEL4WS …)
  - Task-orientation: travel reservations; credit approval; BPM; …
  - Tasks, documents, etc. undergo modifications (e.g., flight reservation from reserved to ticketed), but modified WF objects still identifiable throughout
  - Complex control flow, complex process composition (danger of control flow/dataflow "spaghetti")

  ➔ Dataflow and control-flow are **divorced!**

In a Flux: WS-“Standards” Quicksand

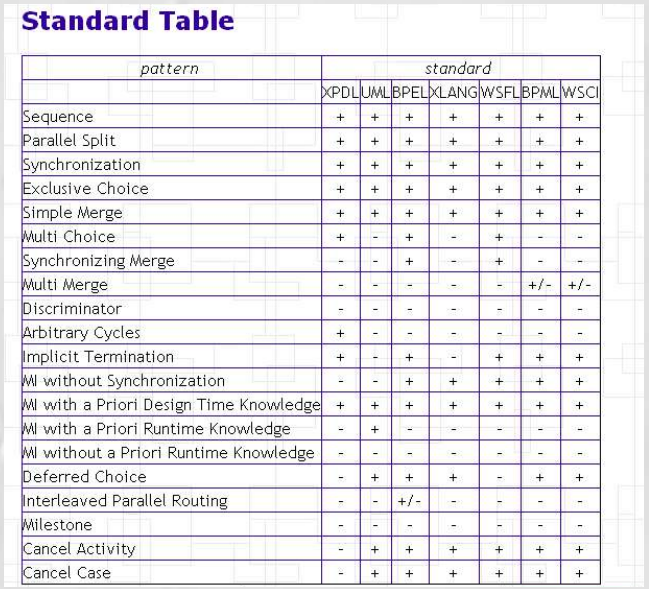


Trends & Controversies
Jan/Feb 2003 issue of IEEE Intelligent Systems
Web Services - Been there done that?

Don’t go with the flow:
Web services composition standards exposed

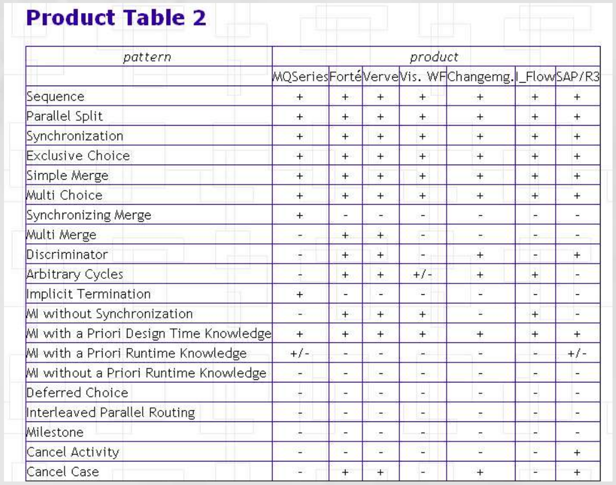
W.M.P. van der Aalst
Dept. of Technology Management, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, w.m.p.v.d.aalst@tm.tue.nl

The recently released Business Process Execution Language for Web Services (BPEL4WS) is said to combine the best of other standards for web services composition such as WSFL from IBM and XLANG of Microsoft. BPEL4WS allows for a mixture of block structured and graph structured process models thus making the language expressive at the price of being complex. Although BPEL4WS is not such a bad proposal by itself, it is remarkable how much attention this standard receives while the more fundamental issues and problems such as semantics, expressiveness, and adequacy do not get the attention they deserve. Having a standard is a very good idea. However, there are too many of them and most of them die before becoming mature. A simple indicator of this development is the increasing length of acronyms: PDL, XPDL, BPSS, EDOC, BPML, WSDL, WSCI, ebXML, and BPEL4WS are just some of the acronyms referring to various standards in the domain. Another problem is that these languages typically have no clearly defined semantics. The only way to overcome these problems is to critically evaluate the so-called standards for web services composition, i.e., Don’t go with the flow!



Standard Table

pattern	*standard*						
	XPDL	UML	BPEL	XLANG	WSFL	BPML	WSCI
Sequence	+	+	+	+	+	+	+
Parallel Split	+	+	+	+	+	+	+
Synchronization	+	+	+	+	+	+	+
Exclusive Choice	+	+	+	+	+	+	+
Simple Merge	+	+	+	+	+	+	+
Multi Choice	+	-	+	-	+	-	-
Synchronizing Merge	-	-	+	-	+	-	-
Multi Merge	-	-	-	-	-	+/-	+/-
Discriminator	-	-	-	-	-	-	-
Arbitrary Cycles	+	-	-	-	-	-	-
Implicit Termination	+	-	+	-	+	+	+
MI without Synchronization	-	-	+	+	+	+	+
MI with a Priori Design Time Knowledge	+	+	+	+	+	+	+
MI with a Priori Runtime Knowledge	-	+	-	-	-	-	-
MI without a Priori Runtime Knowledge	-	-	-	-	-	-	-
Deferred Choice	-	+	+	+	-	+	+
Interleaved Parallel Routing	-	-	+/-	-	-	-	-
Milestone	-	-	-	-	-	-	-
Cancel Activity	-	+	+	+	+	+	+
Cancel Case	-	+	+	+	+	+	+



Product Table 2

pattern	*product*						
	MQSeries	Forté	Verve	Vis. WF	Changemg.	I_Flow	SAP/R3
Sequence	+	+	+	+	+	+	+
Parallel Split	+	+	+	+	+	+	+
Synchronization	+	+	+	+	+	+	+
Exclusive Choice	+	+	+	+	+	+	+
Simple Merge	+	+	+	+	+	+	+
Multi Choice	+	+	+	+	+	+	+
Synchronizing Merge	+	-	-	-	-	-	-
Multi Merge	-	+	+	-	-	-	-
Discriminator	-	+	+	-	+	-	+
Arbitrary Cycles	-	+	+	+/-	+	+	-
Implicit Termination	+	-	-	-	-	-	-
MI without Synchronization	-	+	+	+	-	+	-
MI with a Priori Design Time Knowledge	+	+	+	+	+	+	+
MI with a Priori Runtime Knowledge	+/-	-	-	-	-	-	+/-
MI without a Priori Runtime Knowledge	-	-	-	-	-	-	-
Deferred Choice	-	-	-	-	-	-	-
Interleaved Parallel Routing	-	-	-	-	-	-	-
Milestone	-	-	-	-	-	-	-
Cancel Activity	-	-	-	-	-	-	+
Cancel Case	-	+	+	-	+	-	+

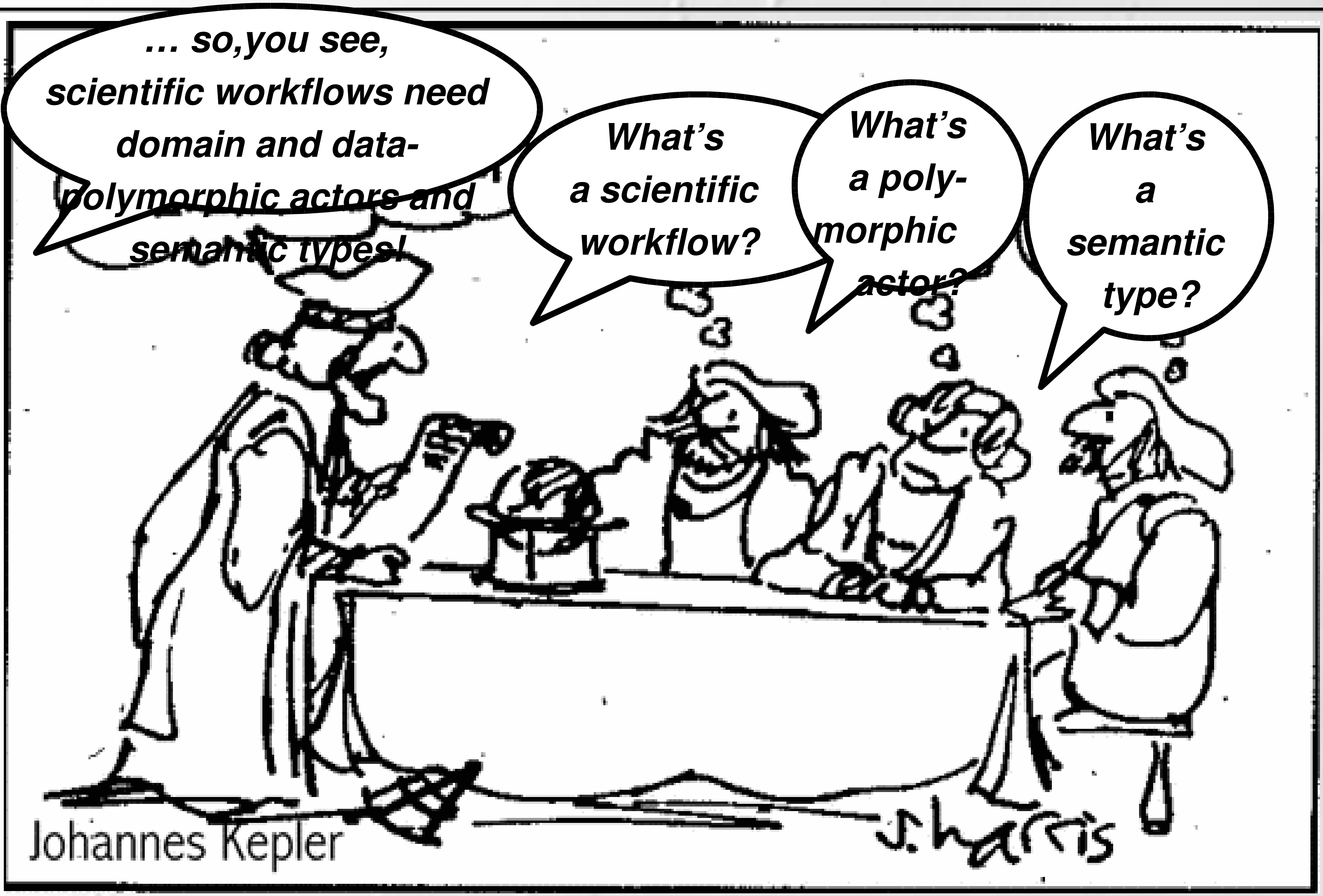
Source: W.M.P. van der Aalst et al. http://tmitwww.tm.tue.nl/research/patterns/
http://tmitwww.tm.tue.nl/staff/wvdaalst/Publications/publications.html

Some Rules of Thumb

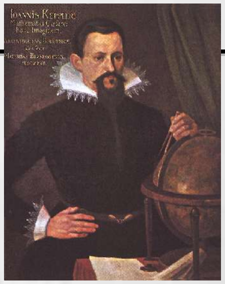
- *Ask yourself: What exists?*
  - *Planets, stars, galaxies, dark matter, …*
  - *Natural numbers, sets, graphs, trees, relations, functions, abstract data types, …*
  - *(Standards are a means to an end. Ask: What end?)*
- *… and what is known about it? What can be done w/ it?*
  - *Universe (your turn)*
  - *Maths & CS (Petri nets, deadlock analysis, query optimization/rewriting, job scheduling, …)*
  - *WS-<huh>?*
- *What is your problem/goal/interest?*
- *Time shall be consumed (no matter what) – your pick:*
  - *Reinvent (… hopefully only good ideas)*
  - *Rediscover; adapt; leverage (… good ideas)*

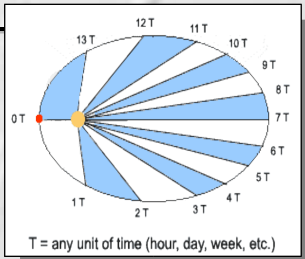
Back to KEPLER…who was ahead of his time …

… but such is life … ;-)


… so,you see, scientific workflows need domain and data-polymorphic actors and semantic types!
What's a scientific workflow?
What's a poly-morphic actor?
What's a semantic type?
Johannes Kepler
S. harris

KEPLER Team, Projects, Sponsors





Ptolemy II

Ilkay Altintas SDM

Chad Berkley SEEK

Shawn Bowers SEEK

Jeffrey Grethe BIRN

Christopher H. Brooks Ptolemy II

Zhengang Cheng SDM

Dan Higgins SEEK

Efrat Jaeger GEON

Matt Jones SEEK

Edward Lee Ptolemy II

Kai Lin GEON

Ashraf Memon GEON

Bertram Ludaescher BIRN, GEON, SDM, SEEK

Steve Mock NMI

Steve Neuendorffer Ptolemy II

Mladen Vouk SDM

Yang Zhao Ptolemy II

…

ptIIp
lane

SEEK

GEON
The Geosciences Network

BIRN
Biomedical
Informatics
Research Network

SciDAC
Scientific Discovery through Advanced Computing

SDM

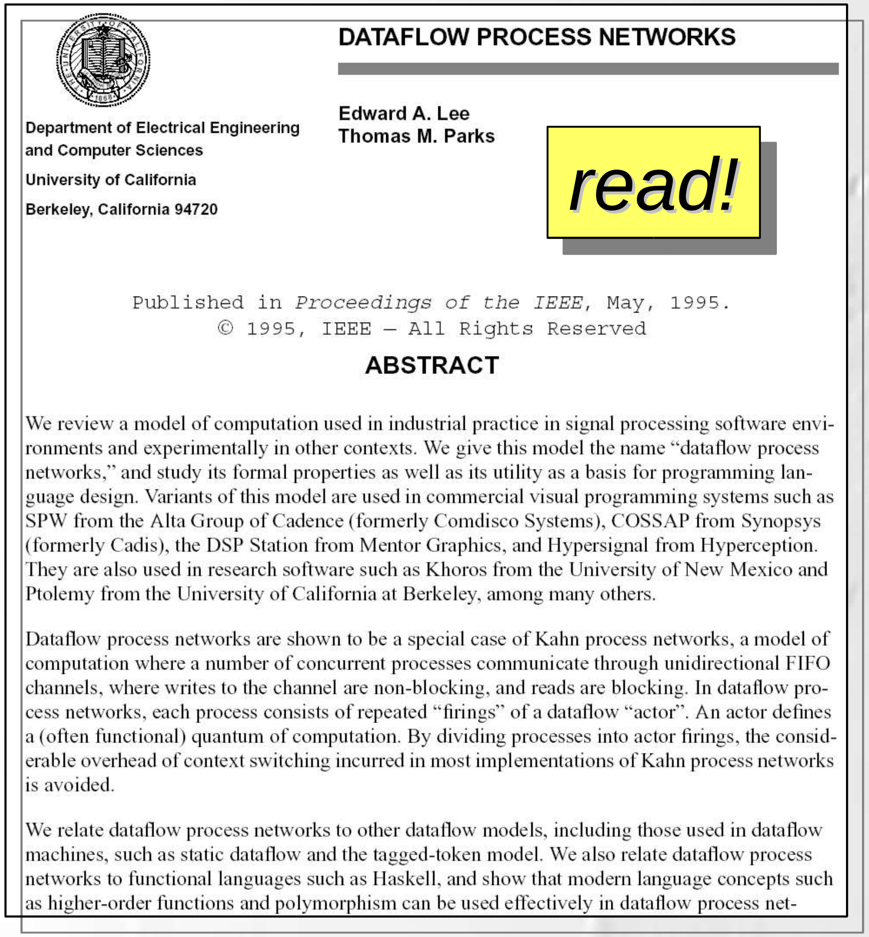
DARPA

NSF

KEPLER: An Open Collaboration

- *Open Source (BSD-style license)*
- *Communications: Mailing lists, IRC*
- *Co-development:*
  - *Via CVS repository*
  - *Becoming a co-developer (currently):*
    - *get a CVS account (read-only)*
    - *contribute via existing KEPLER member*
    - *be voted "in" as a member/co-developer*
- *Software and social engineering:*
  - *How to scale to many new groups?*
  - *How to accommodate different usage/contribution models (core dev … special purpose extender … user)?*

Our Starting Point: Ptolemy II



DATAFLOW PROCESS NETWORKS

Department of Electrical Engineering and Computer Sciences

University of California

Berkeley, California 94720

Edward A. Lee
Thomas M. Parks

Published in *Proceedings of the IEEE*, May, 1995.
© 1995, IEEE — All Rights Reserved

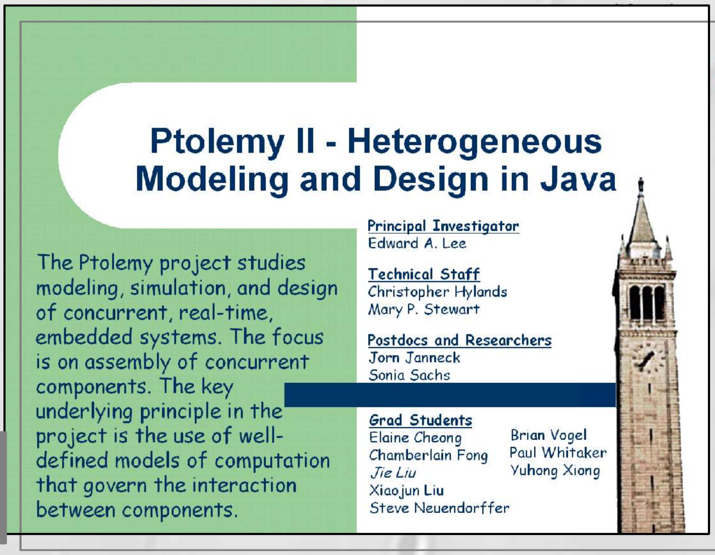
ABSTRACT

We review a model of computation used in industrial practice in signal processing software environments and experimentally in other contexts. We give this model the name “dataflow process networks,” and study its formal properties as well as its utility as a basis for programming language design. Variants of this model are used in commercial visual programming systems such as SPW from the Alta Group of Cadence (formerly Comdisco Systems), COSSAP from Synopsys (formerly Cadis), the DSP Station from Mentor Graphics, and Hypersignal from Hyperception. They are also used in research software such as Khoros from the University of New Mexico and Ptolemy from the University of California at Berkeley, among many others.

Dataflow process networks are shown to be a special case of Kahn process networks, a model of computation where a number of concurrent processes communicate through unidirectional FIFO channels, where writes to the channel are non-blocking, and reads are blocking. In dataflow process networks, each process consists of repeated “firings” of a dataflow “actor”. An actor defines a (often functional) quantum of computation. By dividing processes into actor firings, the considerable overhead of context switching incurred in most implementations of Kahn process networks is avoided.

We relate dataflow process networks to other dataflow models, including those used in dataflow machines, such as static dataflow and the tagged-token model. We also relate dataflow process networks to functional languages such as Haskell, and show that modern language concepts such as higher-order functions and polymorphism can be used effectively in dataflow process net-

read!



Ptolemy II - Heterogeneous Modeling and Design in Java

The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined models of computation that govern the interaction between components.

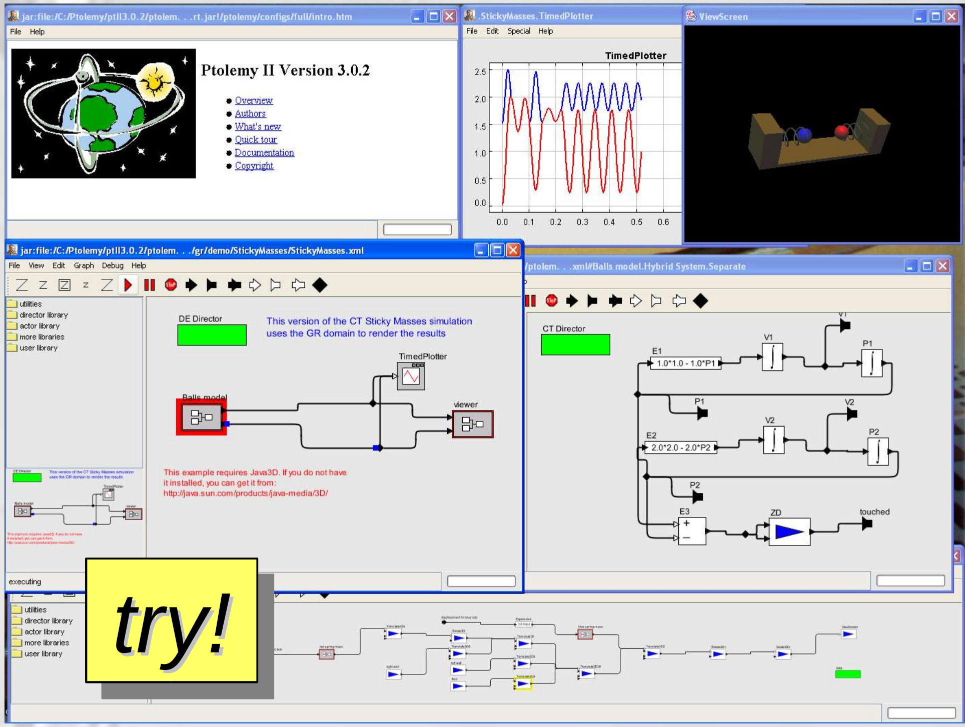
Principal Investigator
Edward A. Lee

Technical Staff
Christopher Hylands
Mary P. Stewart

Postdocs and Researchers
Jorn Janneck
Sonia Sachs

Grad Students
Elaine Cheong
Chamberlain Fong
Jie Liu
Xiaojun Liu
Steve Neuendorffer
Brian Vogel
Paul Whitaker
Yuhong Xiong

see!



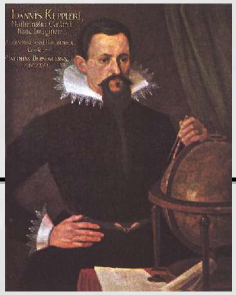
This version of the CT Sticky Masses simulation uses the GR domain to render the results

This example requires Java3D. If you do not have it installed, you can get it from: http://java.sun.com/products/java-media/3D/

try!

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

Some History



- *Gabriel (1986-1991)*
  - *Written in Lisp*
  - *Aimed at signal processing*
  - *Synchronous dataflow (SDF) block diagrams*
  - *Parallel schedulers*
  - *Code generators for DSPs*
  - *Hardware/software co-simulators*
- *Ptolemy Classic (1990-1997)*
  - *Written in C++*
  - *Multiple models of computation*
  - *Hierarchical heterogeneity*
  - *Dataflow variants: BDF, DDF, PN*
  - *C/VHDL/DSP code generators*
  - *Optimizing SDF schedulers*
  - *Higher-order components*
- **Ptolemy II (1996-2022)**
  - *Written in Java*
  - *Domain polymorphism*
  - *Multithreaded*
  - *Network integrated*
  - *Modal models*
  - *Sophisticated type system*
  - *CT, HDF, CI, GR, etc.*

- *PtPlot (1997-??)*
  - *Java plotting package*
- *Tycho (1996-1998)*
  - *Itcl/Tk GUI framework*
- *Diva (1998-2000)*
  - *Java GUI framework*
- **KEPLER (2003-2028)**
  - *scientific workflow extensions*

Ptolemy II: *A laboratory for investigating design*
KEPLER: *A problem-solving environment for Scientific Workflows*

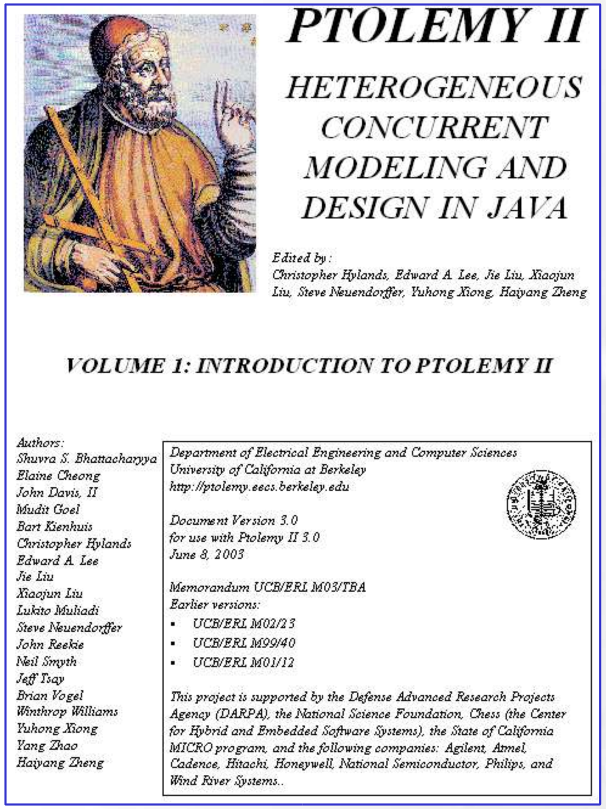
KEPLER = *"Ptolemy II++" for Scientific Workflows*

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

Why Ptolemy II (and thus KEPLER)?

- Ptolemy II Objective:
  - "The focus is on **assembly of concurrent components**. The key underlying principle in the project is the use of **well-defined models of computation** that govern the interaction between components. A major problem area being addressed is the use of **heterogeneous mixtures of models of computation**."
- Data & Process oriented: Dataflow Process Networks
- Natural Data Streaming Support
- User-Orientation
  - "application-ware" (not middle-/under-ware)
  - Workflow design & exec console (Vergil GUI)
- PRAGMATICS
  - Ptolemy II is mature, continuously extended & improved, well-documented (500+pp), …
  - open source system
  - ➔ KEPLER developed across multiple projects (NSF/ITRs SEEK and GEON, DOE SciDAC SDM, …); easy to join the action (open collaboration)

Ptolemy Design Documents


PTOLEMY II

HETEROGENEOUS CONCURRENT MODELING AND DESIGN IN JAVA

Edited by:
Christopher Hylands, Edward A. Lee, Jie Liu, Xiaojun Liu, Steve Neuendorffer, Yuhong Xiong, Haiyang Zheng

VOLUME 1: INTRODUCTION TO PTOLEMY II

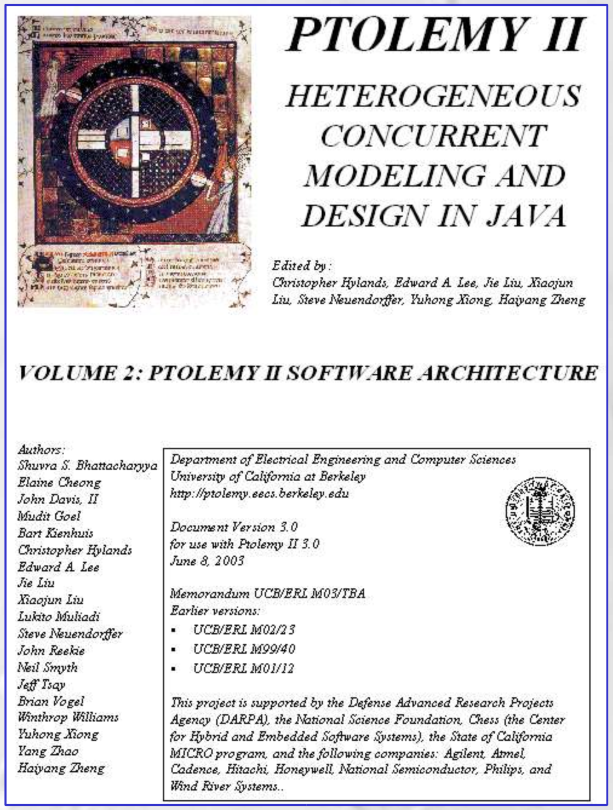
Authors:
Shuvra S. Bhattacharyya
Elaine Cheong
John Davis, II
Mudit Goel
Bart Kienhuis
Christopher Hylands
Edward A. Lee
Jie Liu
Xiaojun Liu
Lukito Muliadi
Steve Neuendorffer
John Reekie
Neil Smyth
Jeff Tsay
Brian Vogel
Winthrop Williams
Yuhong Xiong
Yang Zhao
Haiyang Zheng

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
http://ptolemy.eecs.berkeley.edu

Document Version 3.0
for use with Ptolemy II 3.0
June 8, 2003

Memorandum UCB/ERL M03/TBA
Earlier versions:
- UCB/ERL M02/23
- UCB/ERL M99/40
- UCB/ERL M01/12

This project is supported by the Defense Advanced Research Projects Agency (DARPA), the National Science Foundation, Chess (the Center for Hybrid and Embedded Software Systems), the State of California MICRO program, and the following companies: Agilent, Atmel, Cadence, Hitachi, Honeywell, National Semiconductor, Philips, and Wind River Systems..


PTOLEMY II

HETEROGENEOUS CONCURRENT MODELING AND DESIGN IN JAVA

Edited by:
Christopher Hylands, Edward A. Lee, Jie Liu, Xiaojun Liu, Steve Neuendorffer, Yuhong Xiong, Haiyang Zheng

VOLUME 2: PTOLEMY II SOFTWARE ARCHITECTURE

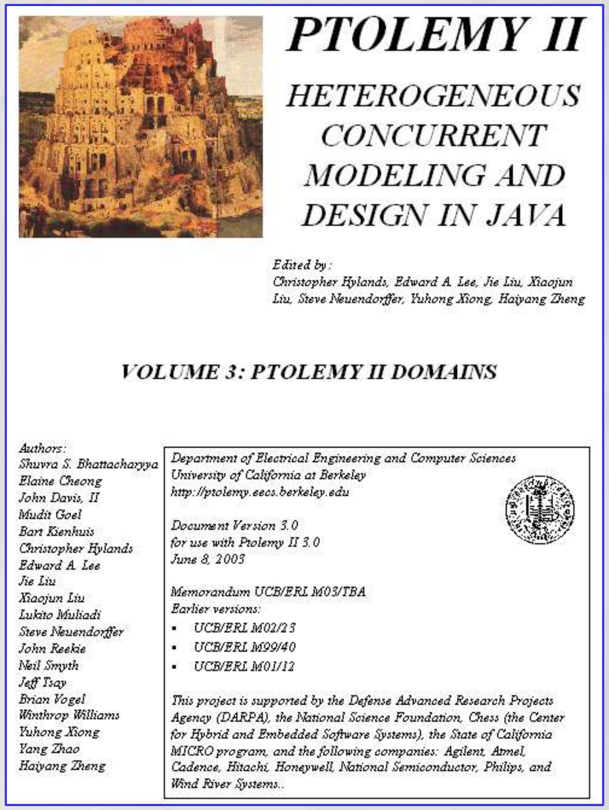
Authors:
Shuvra S. Bhattacharyya
Elaine Cheong
John Davis, II
Mudit Goel
Bart Kienhuis
Christopher Hylands
Edward A. Lee
Jie Liu
Xiaojun Liu
Lukito Muliadi
Steve Neuendorffer
John Reekie
Neil Smyth
Jeff Tsay
Brian Vogel
Winthrop Williams
Yuhong Xiong
Yang Zhao
Haiyang Zheng

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
http://ptolemy.eecs.berkeley.edu

Document Version 3.0
for use with Ptolemy II 3.0
June 8, 2003

Memorandum UCB/ERL M03/TBA
Earlier versions:
- UCB/ERL M02/23
- UCB/ERL M99/40
- UCB/ERL M01/12

This project is supported by the Defense Advanced Research Projects Agency (DARPA), the National Science Foundation, Chess (the Center for Hybrid and Embedded Software Systems), the State of California MICRO program, and the following companies: Agilent, Atmel, Cadence, Hitachi, Honeywell, National Semiconductor, Philips, and Wind River Systems..


PTOLEMY II

HETEROGENEOUS CONCURRENT MODELING AND DESIGN IN JAVA

Edited by:
Christopher Hylands, Edward A. Lee, Jie Liu, Xiaojun Liu, Steve Neuendorffer, Yuhong Xiong, Haiyang Zheng

VOLUME 3: PTOLEMY II DOMAINS

Authors:
Shuvra S. Bhattacharyya
Elaine Cheong
John Davis, II
Mudit Goel
Bart Kienhuis
Christopher Hylands
Edward A. Lee
Jie Liu
Xiaojun Liu
Lukito Muliadi
Steve Neuendorffer
John Reekie
Neil Smyth
Jeff Tsay
Brian Vogel
Winthrop Williams
Yuhong Xiong
Yang Zhao
Haiyang Zheng

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
http://ptolemy.eecs.berkeley.edu

Document Version 3.0
for use with Ptolemy II 3.0
June 8, 2003

Memorandum UCB/ERL M03/TBA
Earlier versions:
- UCB/ERL M02/23
- UCB/ERL M99/40
- UCB/ERL M01/12

This project is supported by the Defense Advanced Research Projects Agency (DARPA), the National Science Foundation, Chess (the Center for Hybrid and Embedded Software Systems), the State of California MICRO program, and the following companies: Agilent, Atmel, Cadence, Hitachi, Honeywell, National Semiconductor, Philips, and Wind River Systems..

Volume 1:

User-Oriented

Volume 2:

Developer-Oriented

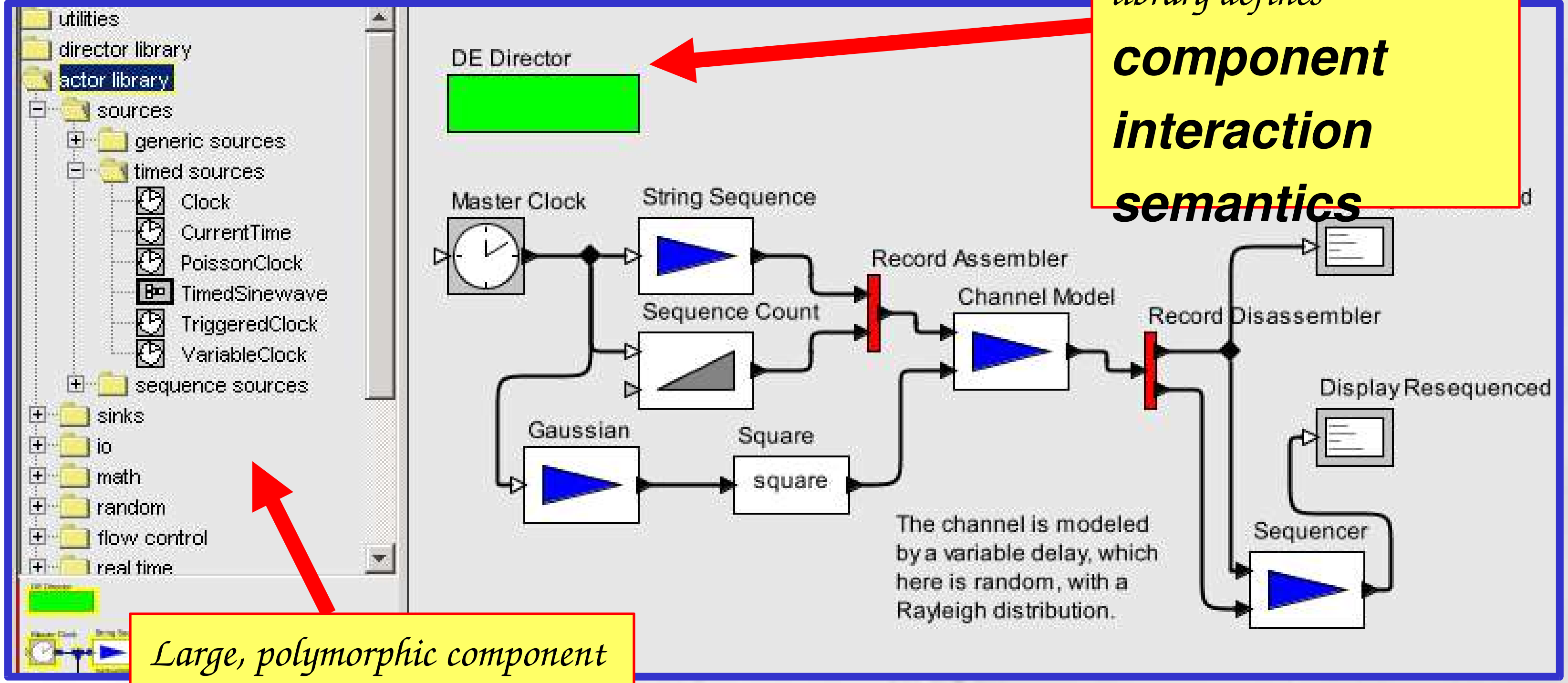
Volume 3:

Researcher-Oriented

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

Ptolemy Principles

Basic Ptolemy II infrastructure:



The channel is modeled by a variable delay, which here is random, with a Rayleigh distribution.

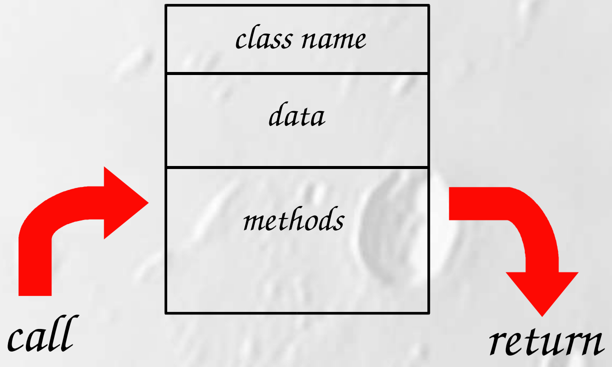
Director *from a library defines* ***component interaction semantics***

Large, polymorphic component library.

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

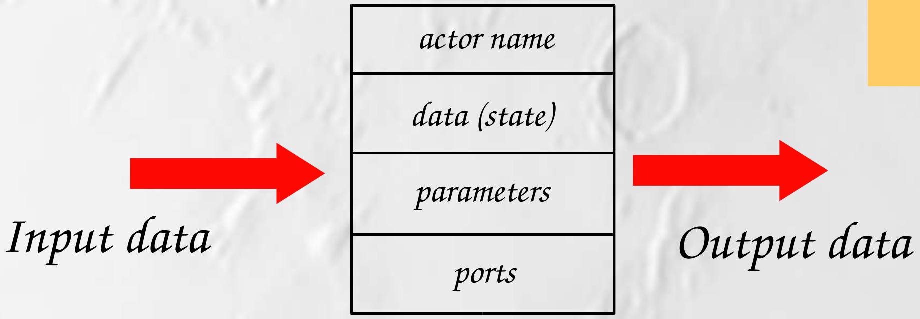
Focus on Actor-Oriented Design

- *Object orientation:*



What flows through an object is sequential control

- **Actor orientation:**



What flows through an object is streams of data

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

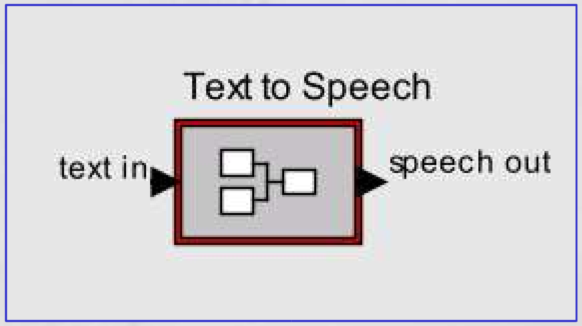
Object-Oriented vs. Actor-Oriented Interface Definitions

Object Oriented

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

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

Actor Oriented



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

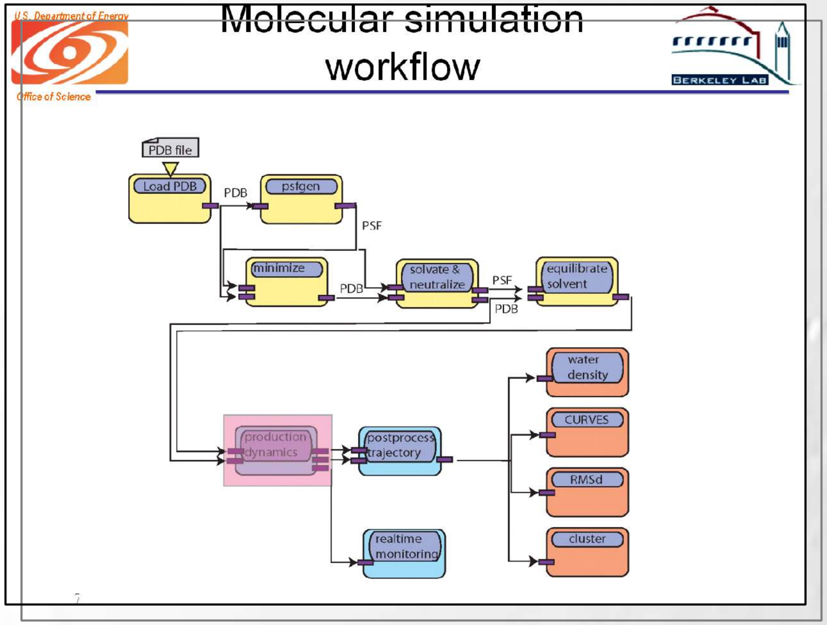
Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/

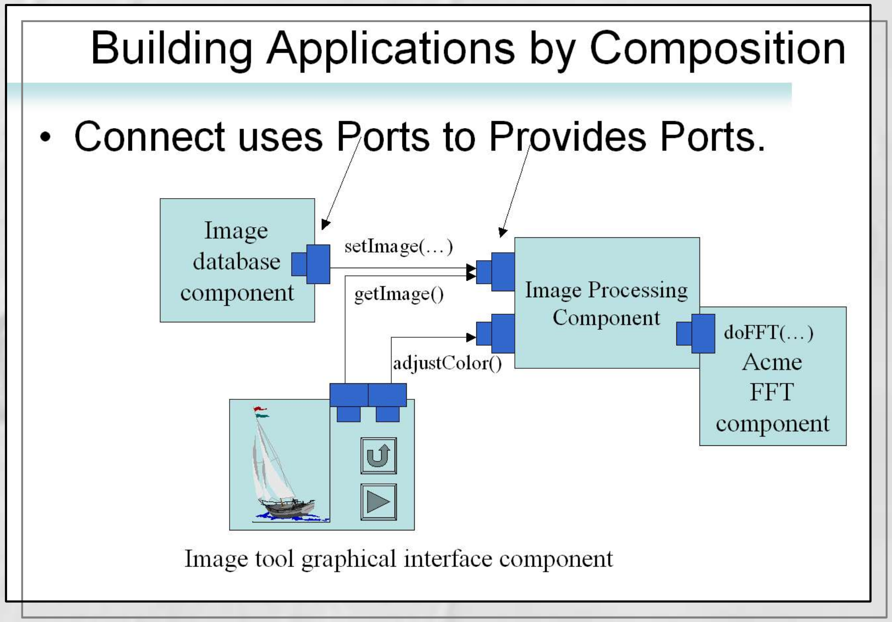
Examples of Actor-Oriented Component Frameworks

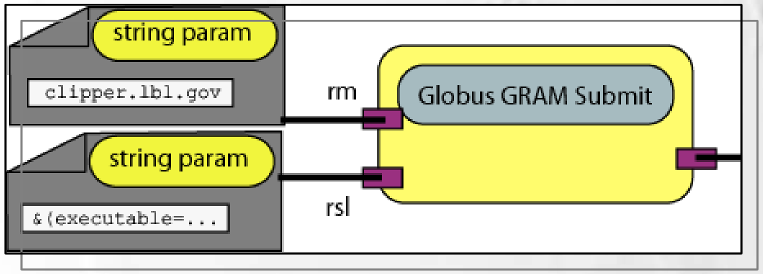
- *Simulink (The MathWorks)*
- *Labview (National Instruments)*
- *Modelica (Linkoping)*
- *OCP, open control platform (Boeing)*
- *GME, actor-oriented meta-modeling (Vanderbilt)*
- *Easy5 (Boeing)*
- *SPW, signal processing worksystem (Cadence)*
- *System studio (Synopsys)*
- *ROOM, real-time object-oriented modeling (Rational)*
- *Port-based objects (U of Maryland)*
- *I/O automata (MIT)*
- *VHDL, Verilog, SystemC (Various)*
- *Polis & Metropolis (UC Berkeley)*
- *Ptolemy & Ptolemy II (UC Berkeley)*
- …

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

Component Composition & Interaction



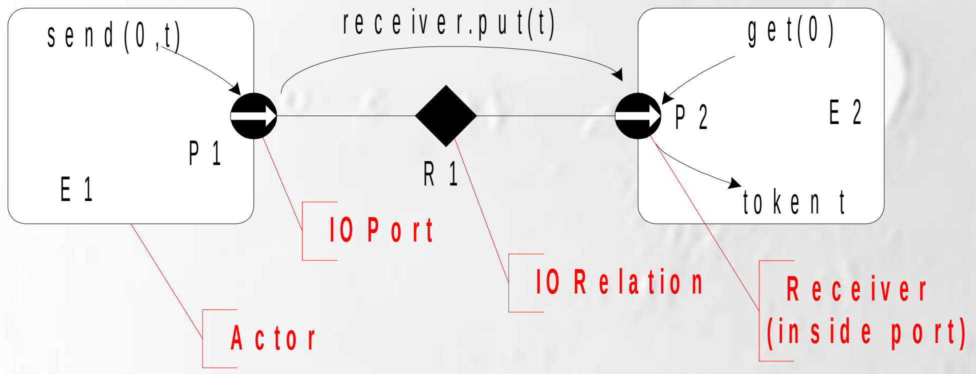

Building Applications by Composition
• Connect uses Ports to Provides Ports.
Image database component
setImage(…)
getImage()
Image Processing Component
adjustColor()
doFFT(…)
Acme FFT component
Image tool graphical interface component



- *Components linked via ports*
- *Dataflow (and msg/ctl-flow)*
- ***But where is the component interaction semantics defined??***
- *cf. WS composition, orchestration, …*

ACTOR Package
Supports Producer/Consumer Components

Basic Transport

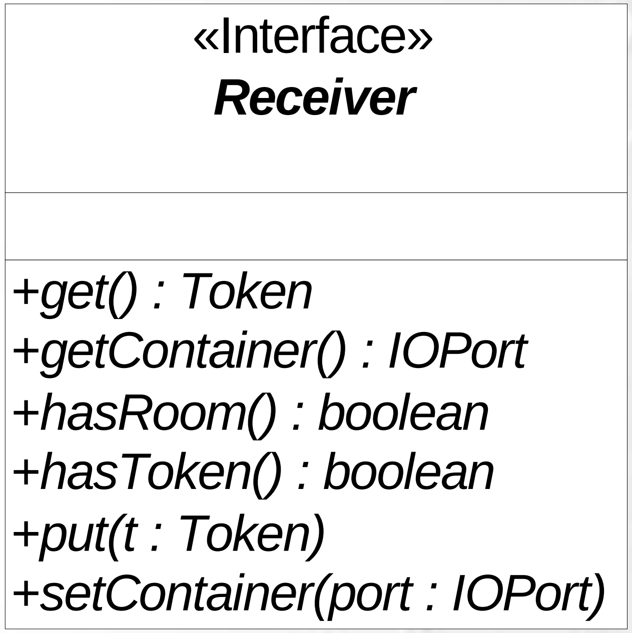


Services in the Infrastructure:

- *broadcast*
- *multicast*
- *busses*
- *mutations*
- *clustering*
- *parameterization*
- *typing*
- *polymorphism*

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

Component Interaction *and* Behavioral Polymorphism *in Ptolemy II*

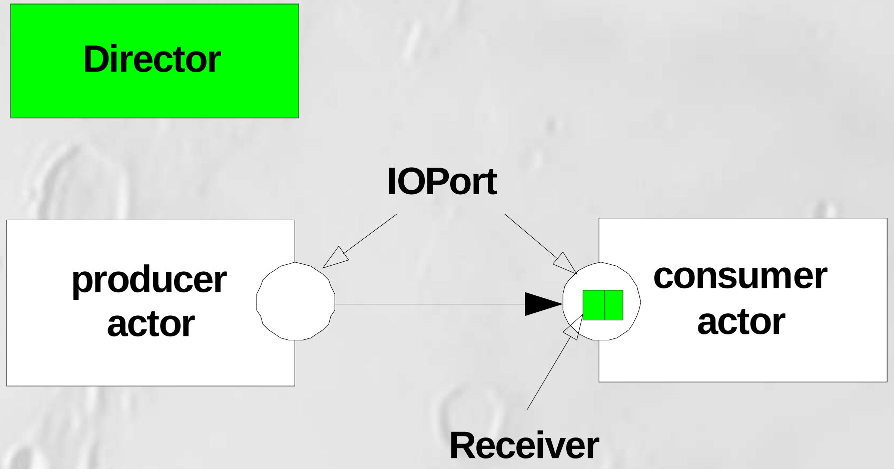


«Interface»
Receiver

+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/

Domains: *Semantics for* Component Interaction

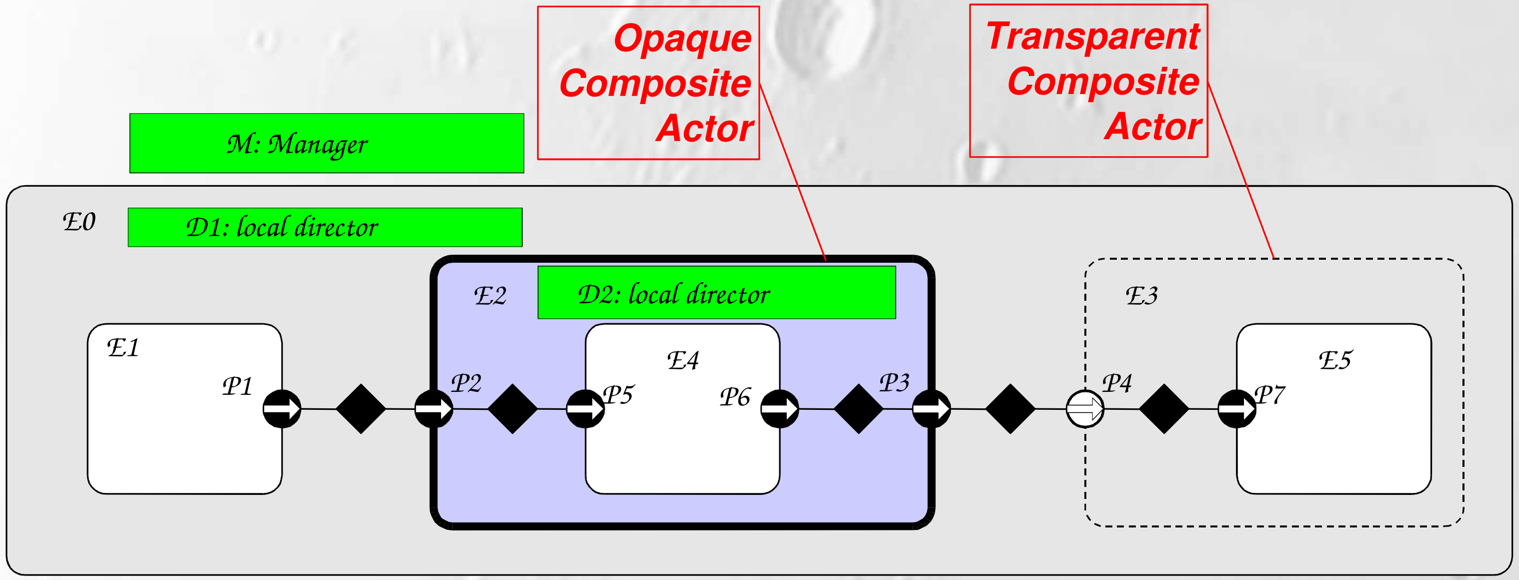
- *CI – Push/pull component interaction*
- *CSP – concurrent threads with rendezvous*
- *CT – continuous-time modeling*
- *DE – discrete-event systems*
- *DDE – distributed discrete events*
- *FSM – finite state machines*
- *DT – discrete time (cycle driven)*
- *Giotto – synchronous periodic*
- *GR –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

For (coarse grained) Scientific Workflows!

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

Hierarchical Heterogeneity

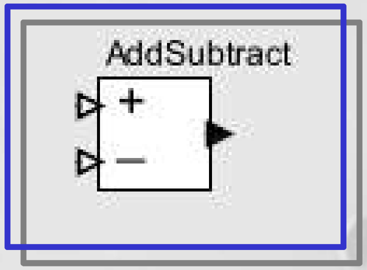
Directors *are* **domain***-specific. A composite actor with a director becomes opaque. The Manager is domain-independent.*



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

Polymorphic Actors: 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*

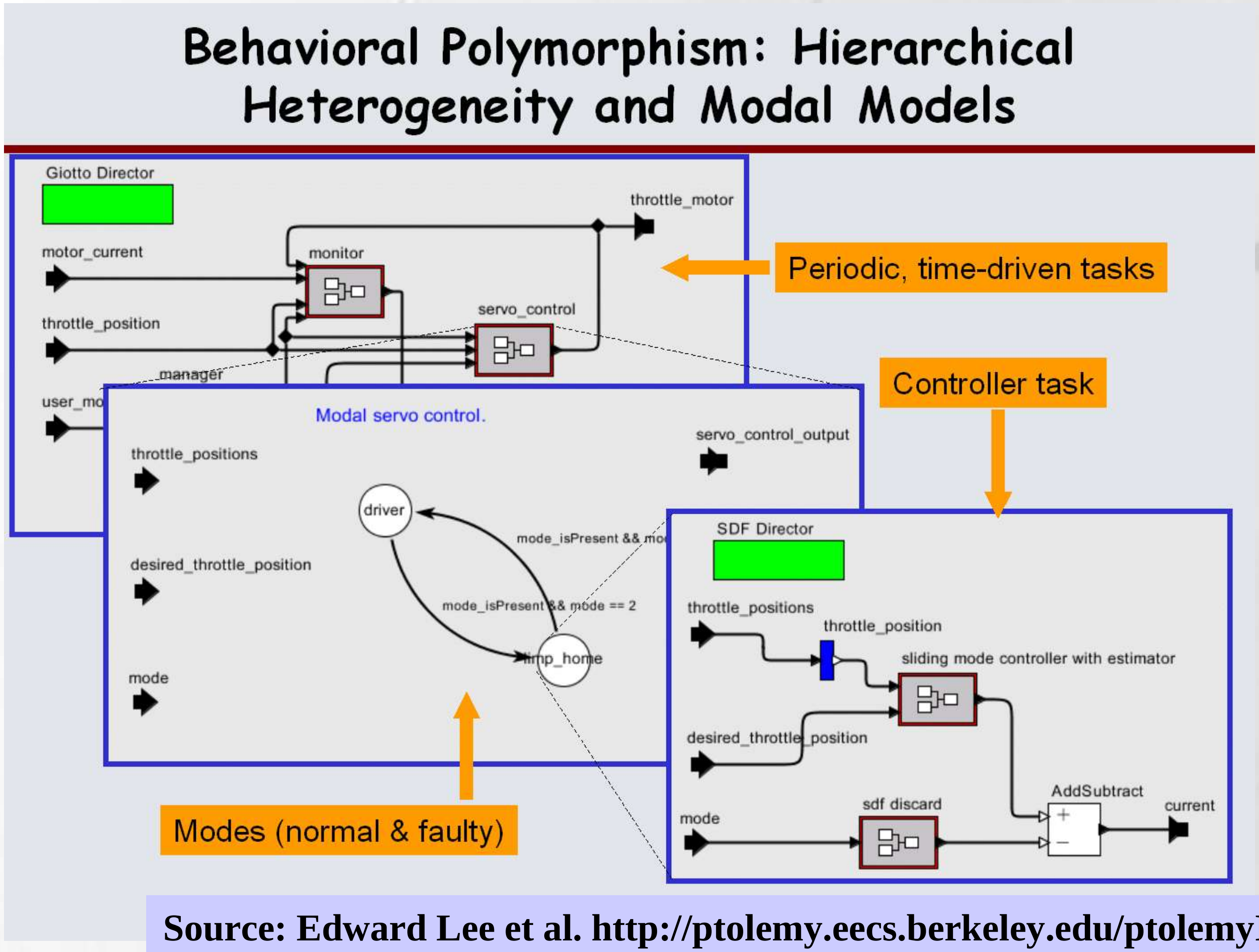


By not choosing among these when defining the component, we get a huge increment in component re-usability. But how do we ensure that the component will work in all these circumstances?

**hey, Ptolemy has been out for long!*

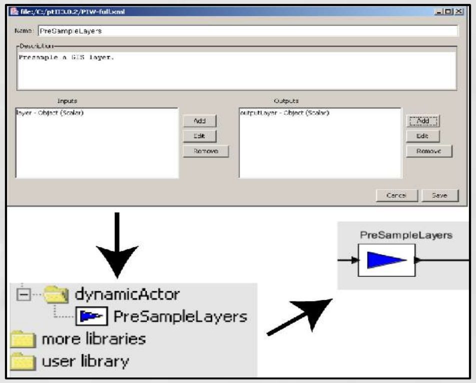
Source: Edward Lee et al. http://ptolemy.eecs.berkeley.edu/

Directors *and Combining Different Component Interaction Semantics*



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

Scientific Workflows in KEPLER


file:/C:/ptII3.0.2/PTW-full.xml

Name: PreSampleLayers

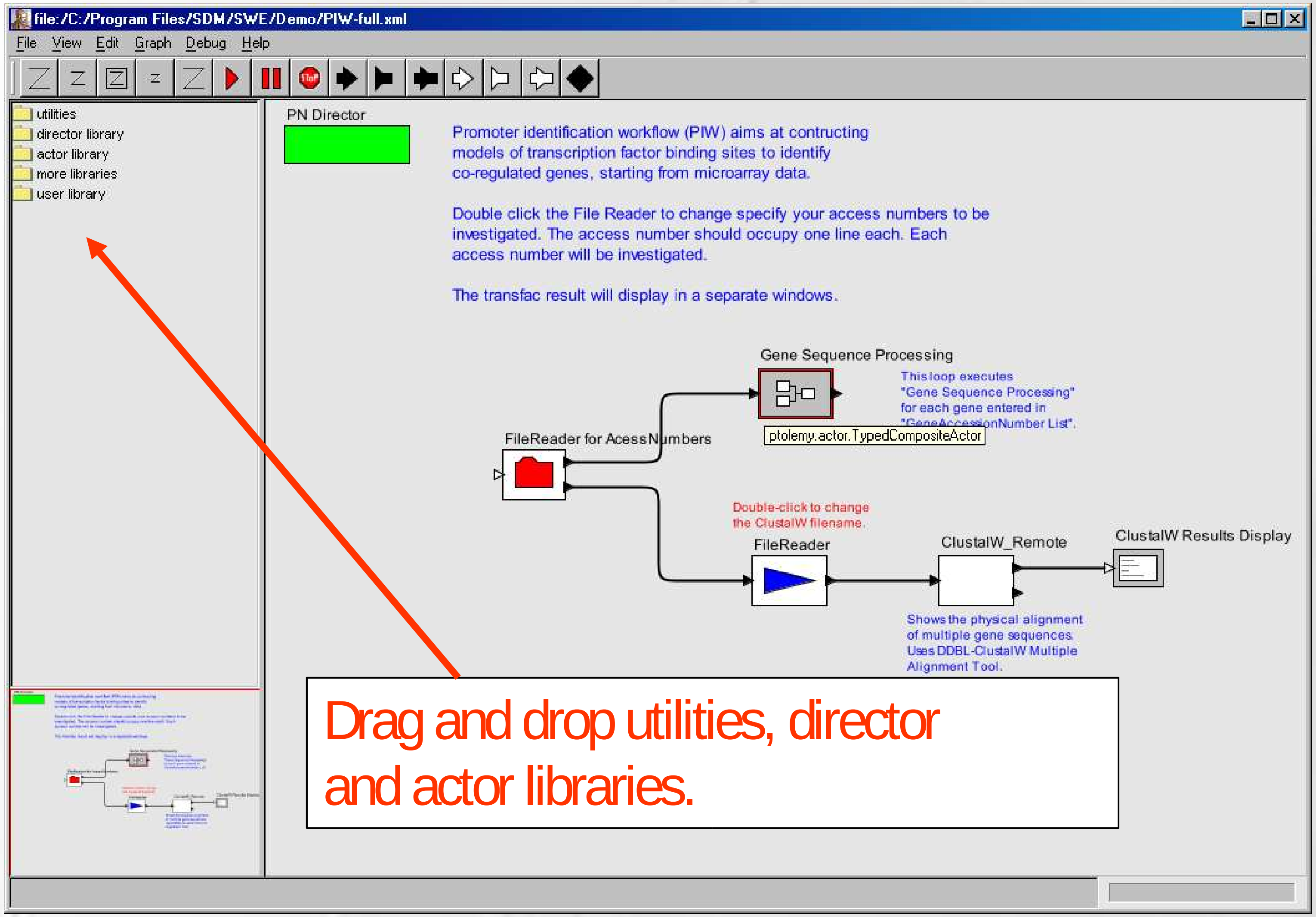
Description: Presample a GIS layer.

Inputs: layer - Object (Scalar)

Outputs: outputLayer - Object (Scalar)

- *Modeling and Workflow Design*
- *Web services = individual components ("actors")*
- *"Minute-Made" Application Integration:*
  - *Plugging-in and harvesting web service components is easy, fast*
- *Rich SWF modeling semantics ("**directors**"):*
  - *Different and precise dataflow models of computation*
  - *Clear and composable component interaction semantics*
  - ➔ *Web service composition and application integration tool*
- *Coming soon:*
  - *Structural and semantic typing (better design support)*
  - *Grid-enabled web services (for big data, big computations,…)*
  - *Different deployment models (web service, web site, applet, …)*

The KEPLER (=Ptolemy II) GUI: Vergil
(Steve Neuendorffer, Ptolemy II)


file:/C:/Program Files/SDM/SWE/Demo/PIW-full.xml
File View Edit Graph Debug Help
utilities
director library
actor library
more libraries
user library
PN Director
Gene Sequence Processing
ptolemy.actor.TypedCompositeActor
FileReader for AcessNumbers
Double-click to change the ClustalW filename.
FileReader
ClustalW_Remote
ClustalW Results Display

Promoter identification workflow (PIW) aims at contructing models of transcription factor binding sites to identify co-regulated genes, starting from microarray data.

Double click the File Reader to change specify your access numbers to be investigated. The access number should occupy one line each. Each access number will be investigated.

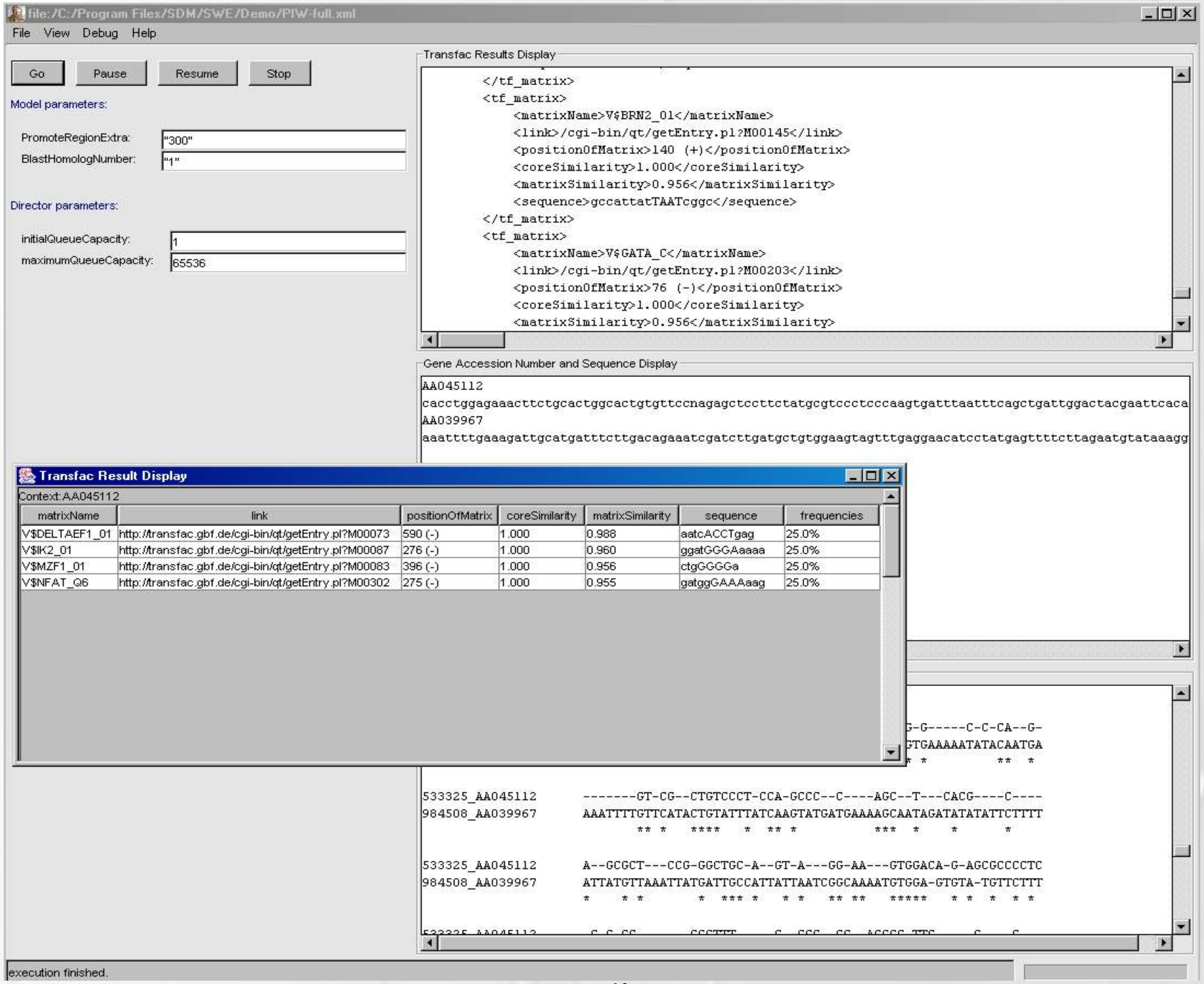
The transfac result will display in a separate windows.

This loop executes "Gene Sequence Processing" for each gene entered in "GeneAccessionNumber List".

Shows the physical alignment of multiple gene sequences. Uses DDBL-ClustalW Multiple Alignment Tool.

Drag and drop utilities, director and actor libraries.

Running a Genomics WF (Ilkay Altintas, SDM)



file:/C:/Program Files/SDM/SWE/Demo/PIW-full.xml

Model parameters:

PromoteRegionExtra:	"300"
BlastHomologNumber:	"1"

Director parameters:

initialQueueCapacity:	1
maximumQueueCapacity:	65536

Transfac Results Display

```
        </tf_matrix>
        <tf_matrix>
            <matrixName>V$BRN2_01</matrixName>
            <link>/cgi-bin/qt/getEntry.pl?M00145</link>
            <positionOfMatrix>140 (+)</positionOfMatrix>
            <coreSimilarity>1.000</coreSimilarity>
            <matrixSimilarity>0.956</matrixSimilarity>
            <sequence>gccattatTAATcggc</sequence>
        </tf_matrix>
        <tf_matrix>
            <matrixName>V$GATA_C</matrixName>
            <link>/cgi-bin/qt/getEntry.pl?M00203</link>
            <positionOfMatrix>76 (-)</positionOfMatrix>
            <coreSimilarity>1.000</coreSimilarity>
            <matrixSimilarity>0.956</matrixSimilarity>
```

Gene Accession Number and Sequence Display

```
AA045112
cacctggagaaacttctgcactggcactgtgttccnagagctccttctatgcgtccctcccaagtgatttaatttcagctgattggactacgaattcaca
AA039967
aaattttgaaagattgcatgatttcttgacagaaatcgatcttgatgctgtggaagtagtttgaggaacatcctatgagttttcttagaatgtataaagg
```

Transfac Result Display

Context: AA045112

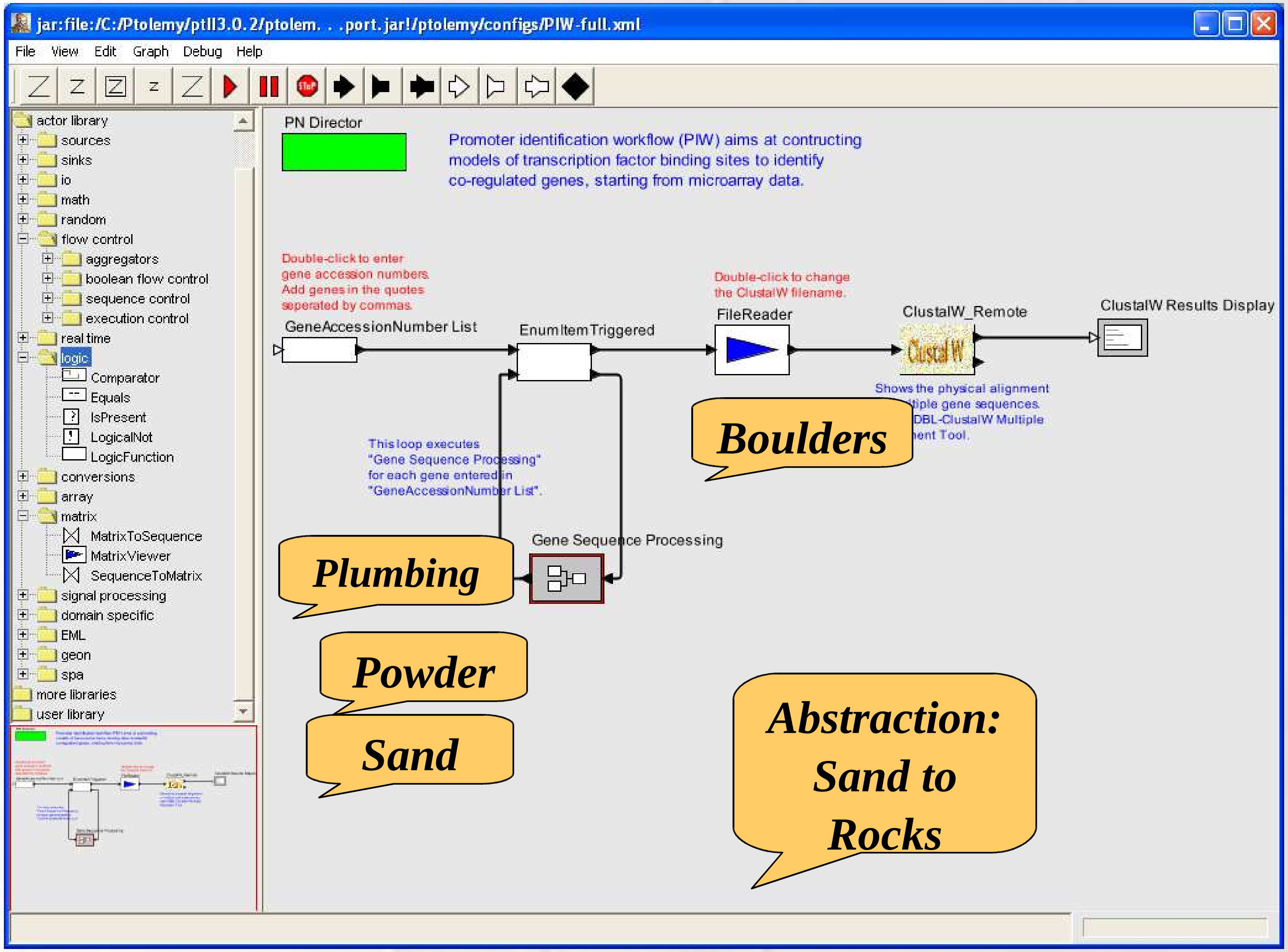
matrixName	link	positionOfMatrix	coreSimilarity	matrixSimilarity	sequence	frequencies
V$DELTAEF1_01	http://transfac.gbf.de/cgi-bin/qt/getEntry.pl?M00073	590 (-)	1.000	0.988	aatcACCTgag	25.0%
V$IK2_01	http://transfac.gbf.de/cgi-bin/qt/getEntry.pl?M00087	276 (-)	1.000	0.960	ggatGGGAaaaa	25.0%
V$MZF1_01	http://transfac.gbf.de/cgi-bin/qt/getEntry.pl?M00083	396 (-)	1.000	0.956	ctgGGGGa	25.0%
V$NFAT_Q6	http://transfac.gbf.de/cgi-bin/qt/getEntry.pl?M00302	275 (-)	1.000	0.955	gatggGAAAaag	25.0%

```
                                                                   G-G-----C-C-CA--G-
                                                                   GTGAAAAATATACAATGA
                                                                   * *        **  *

533325_AA045112      -------GT-CG--CTGTCCCT-CCA-GCCC--C----AGC--T---CACG----C----
984508_AA039967      AAATTTTGTTCATACTGTATTTATCAAGTATGATGAAAAGCAATAGATATATATTCTTTT
                            ** *   ****   *  ** *          ***  *    *       *

533325_AA045112      A--GCGCT---CCG-GGCTGC-A--GT-A---GG-AA---GTGGACA-G-AGCGCCCCTC
984508_AA039967      ATTATGTTAAATTATGATTGCCATTATTAATCGGCAAAATGTGGA-GTGTA-TGTTCTTT
                     *    * *       *  *** *   * *   ** **   *****   * *  *  * *
```

execution finished.

B. Ludäscher, SDSC

Support for Multiple Workflow **Granularities**



jar:file:/C:/Ptolemy/ptII3.0.2/ptolem. . .port.jar!/ptolemy/configs/PIW-full.xml

File View Edit Graph Debug Help

actor library
- sources
- sinks
- io
- math
- random
- flow control
  - aggregators
  - boolean flow control
  - sequence control
  - execution control
- real time
- logic
  - Comparator
  - Equals
  - IsPresent
  - LogicalNot
  - LogicFunction
- conversions
- array
- matrix
  - MatrixToSequence
  - MatrixViewer
  - SequenceToMatrix
- signal processing
- domain specific
- EML
- geon
- spa

more libraries
user library

PN Director

Promoter identification workflow (PIW) aims at contructing models of transcription factor binding sites to identify co-regulated genes, starting from microarray data.

Double-click to enter gene accession numbers. Add genes in the quotes seperated by commas.

Double-click to change the ClustalW filename.

This loop executes "Gene Sequence Processing" for each gene entered in "GeneAccessionNumber List".

Shows the physical alignment ltiple gene sequences. DBL-ClustalW Multiple ent Tool.

Boulders

Plumbing

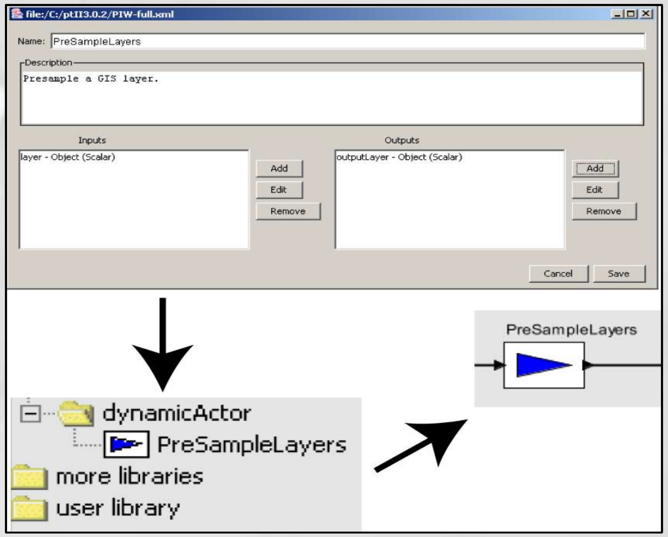
Powder

Sand

Abstraction: Sand to Rocks

Some KEPLER Core Capabilities

- *Designing scientific workflows*
  - *Composition of actors (tasks) to perform a scientific WF*
- *Actor prototyping*
- *Accessing heterogeneous data*
  - *Data access wizard to search and retrieve Grid-based resources*
  - *Relational DB access and query*
  - *Ability to link to EML data sources*



Name: PreSampleLayers

Description: Presample a GIS layer.

Inputs: layer - Object (Scalar)

Outputs: outputLayer - Object (Scalar)

Some KEPLER Core Capabilities

- *Data transformation actors to link heterogeneous data*
- *Executing scientific workflows*
  - *Distributed and/or local computation*
  - *Various models for computational semantics and scheduling*
  - **SDF** *and* **PN***: Most common for scientific workflows*
- *External computing environments:*
  - *C++, Python, C, … through Command-Line or WS: anything!*
- *Deploying scientific tasks and workflows as web services themselves(… planned …)*

Distributed Workflows in KEPLER

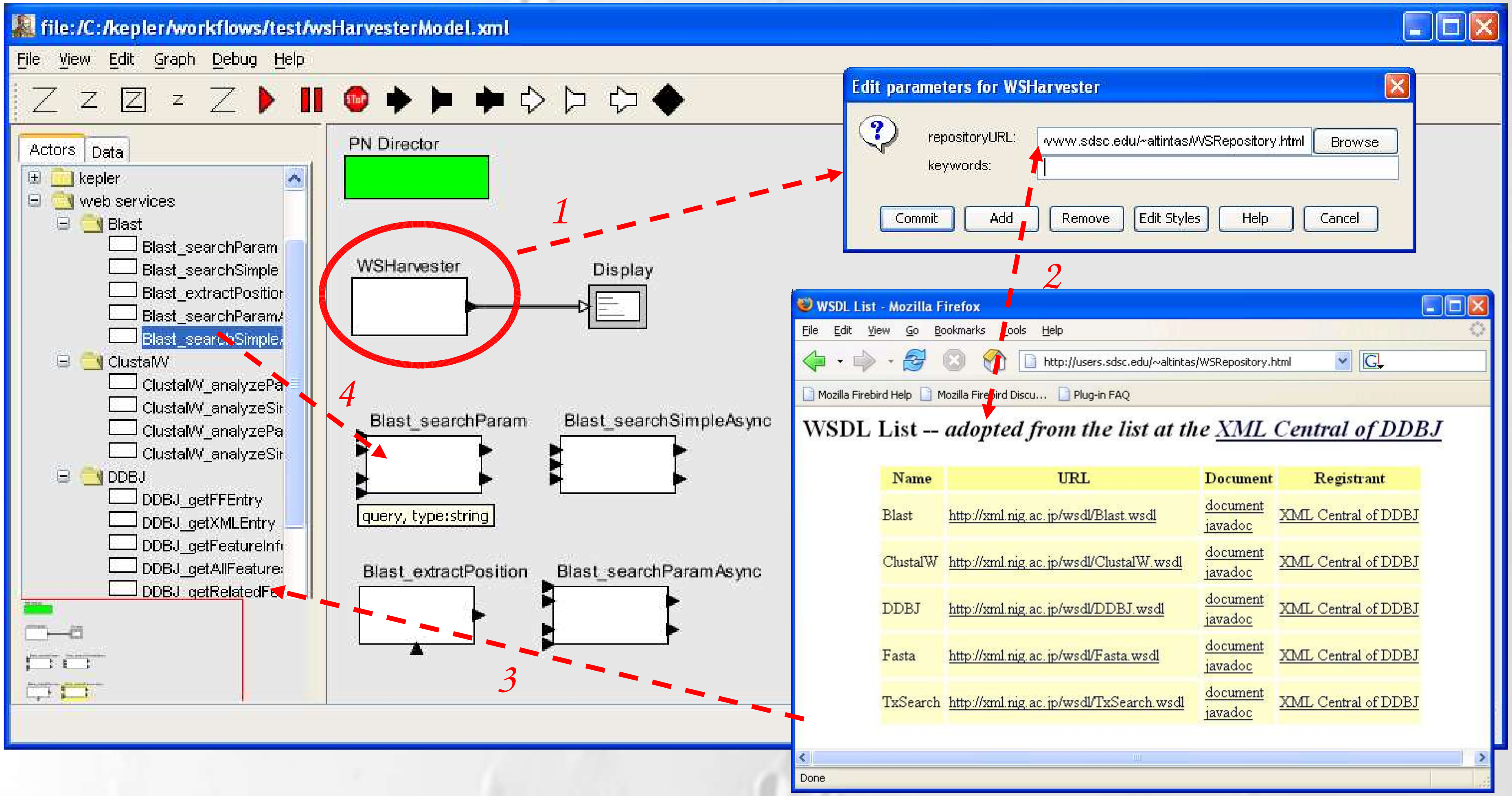
- *Web and Grid Service plug-ins*
  - *WSDL (now) and Grid services (stay tuned …)*
  - *ProxyInit, GlobusGridJob, GridFTP, DataAccessWizard*
  - *SSH, SCP, SDSC SRB, OGS?-???… coming*

- *WS Harvester*
  - *Import query-defined WS operations as Kepler actors*

- *XSLT and XQuery Data Transformers*
  - *to link* **not** *"designed-to-fit" web services*

- *WS-deployment interface (coming)*

Web Services → Actors (WS Harvester)



repositoryURL: www.sdsc.edu/~altintas/WSRepository.html

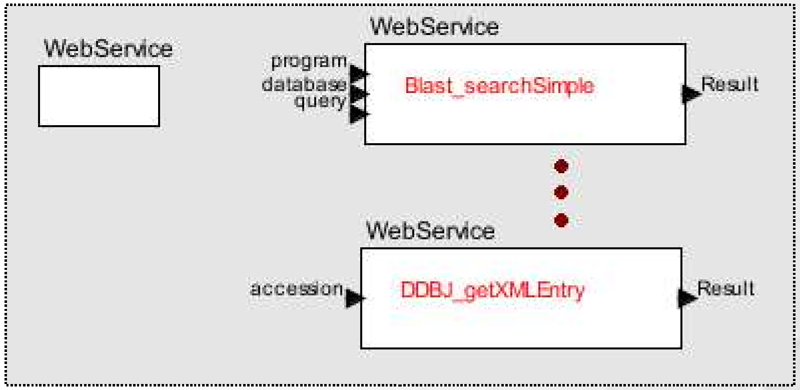
keywords:

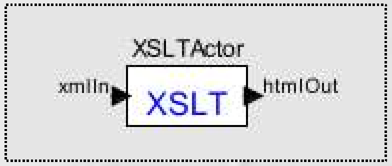
http://users.sdsc.edu/~altintas/WSRepository.html

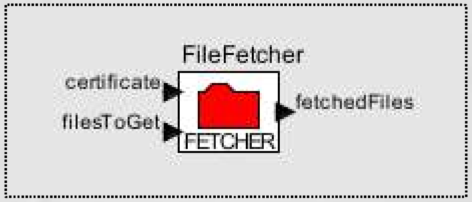
WSDL List -- *adopted from the list at the XML Central of DDBJ*

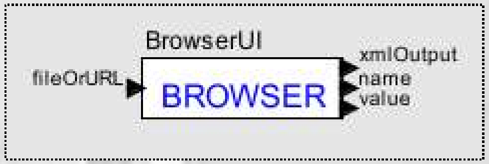
Name	URL	Document	Registrant
Blast	http://xml.nig.ac.jp/wsdl/Blast.wsdl	document javadoc	XML Central of DDBJ
ClustalW	http://xml.nig.ac.jp/wsdl/ClustalW.wsdl	document javadoc	XML Central of DDBJ
DDBJ	http://xml.nig.ac.jp/wsdl/DDBJ.wsdl	document javadoc	XML Central of DDBJ
Fasta	http://xml.nig.ac.jp/wsdl/Fasta.wsdl	document javadoc	XML Central of DDBJ
TxSearch	http://xml.nig.ac.jp/wsdl/TxSearch.wsdl	document javadoc	XML Central of DDBJ

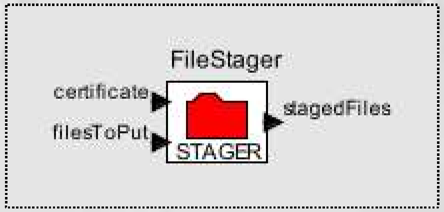
Some special KEPLER actors …

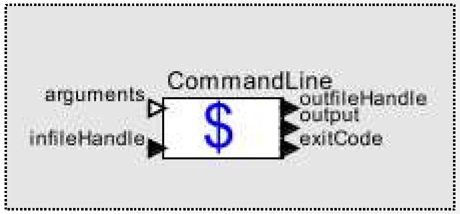




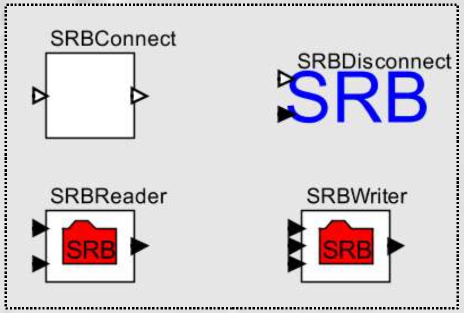


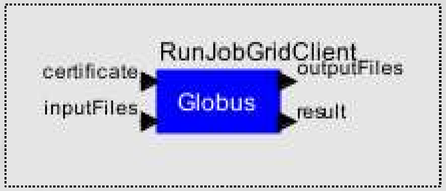


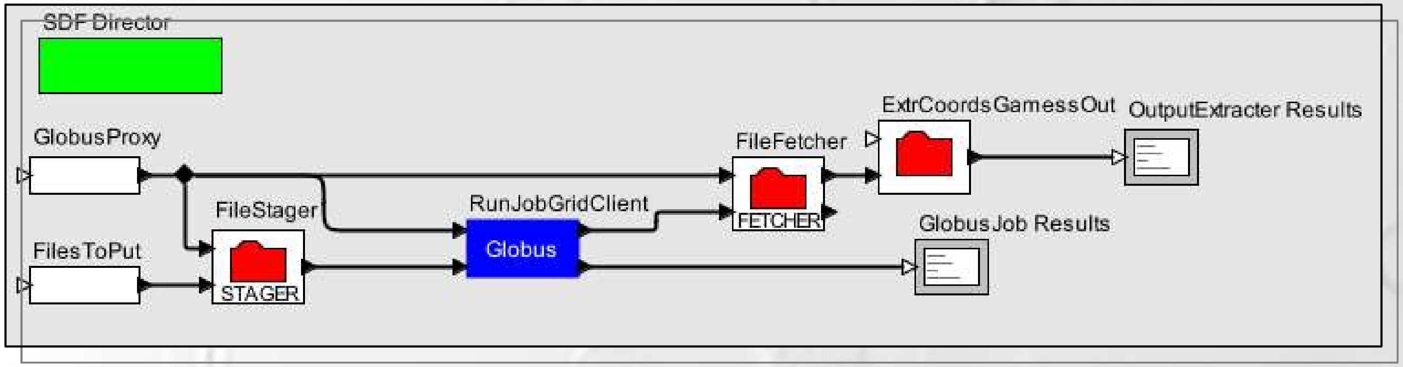


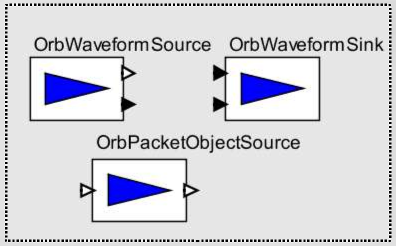


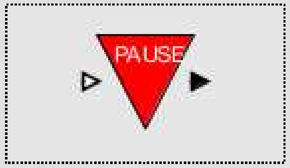
Email
messageBody
@



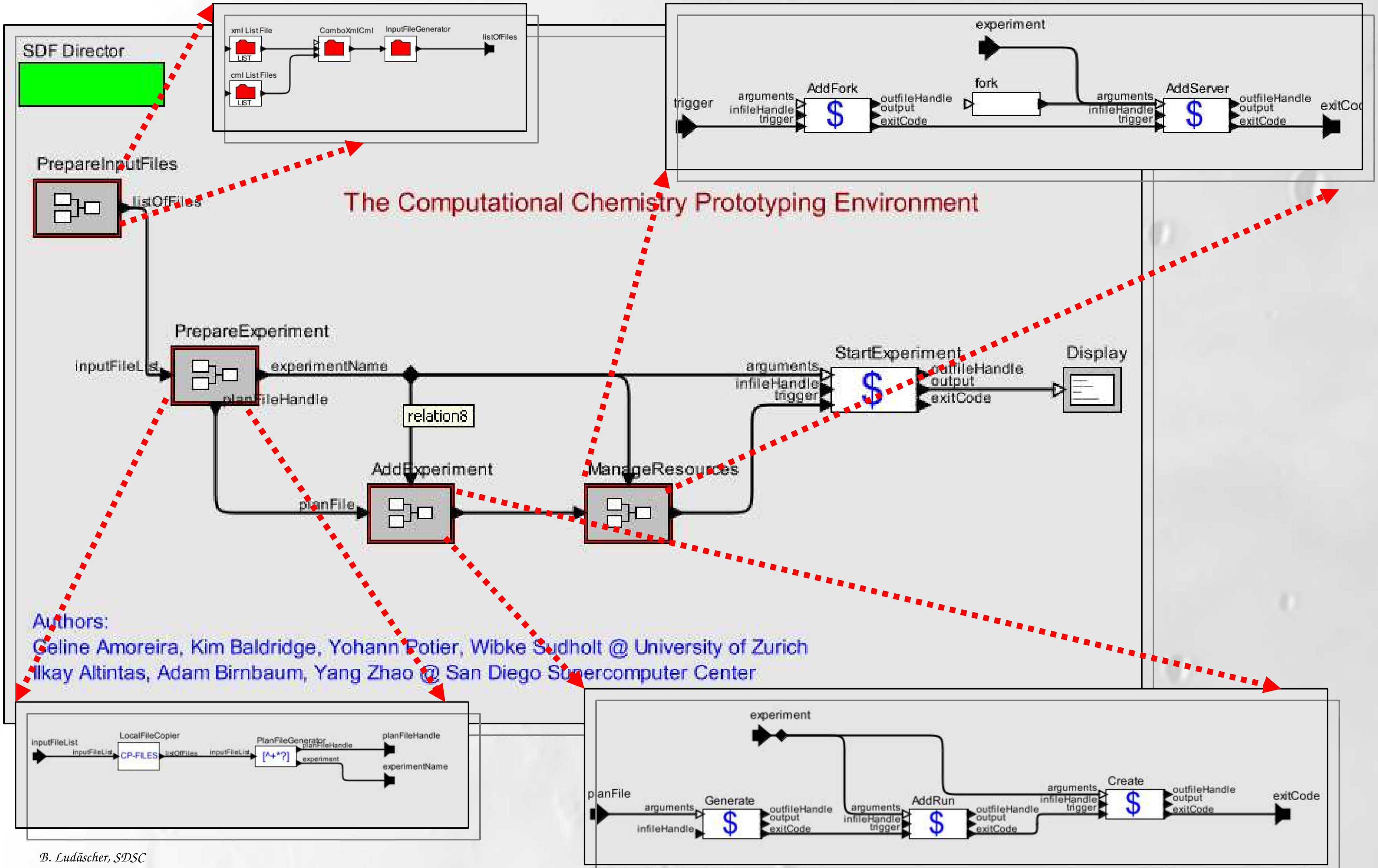








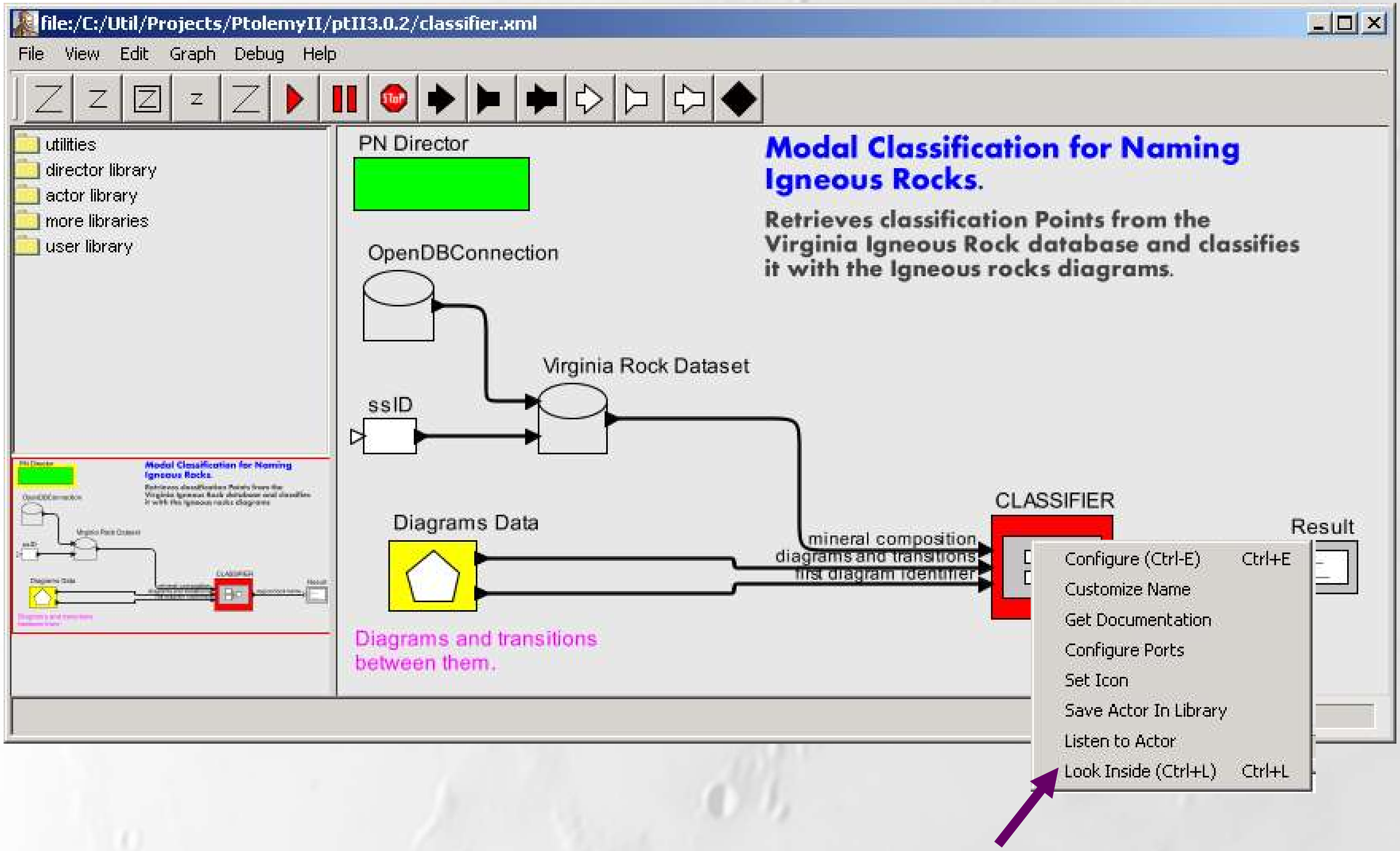
Job Management w/ NIMROD



The Computational Chemistry Prototyping Environment

Authors:
Celine Amoreira, Kim Baldridge, Yohann Potier, Wibke Sudholt @ University of Zurich
Ilkay Altintas, Adam Birnbaum, Yang Zhao @ San Diego Supercomputer Center

Application Examples: Mineral Classification with KEPLER … *(Efrat Jaeger, GEON)*

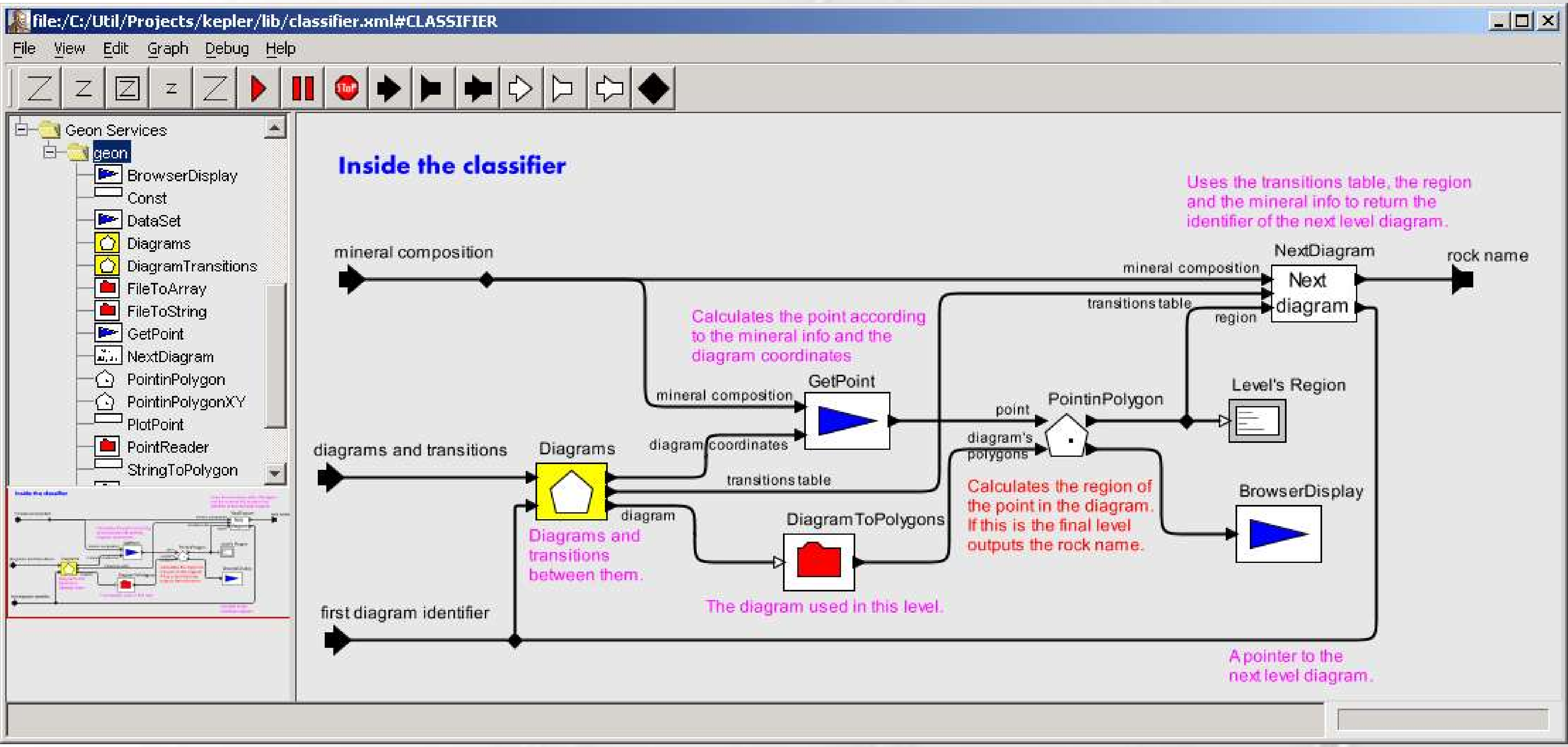


Modal Classification for Naming Igneous Rocks.

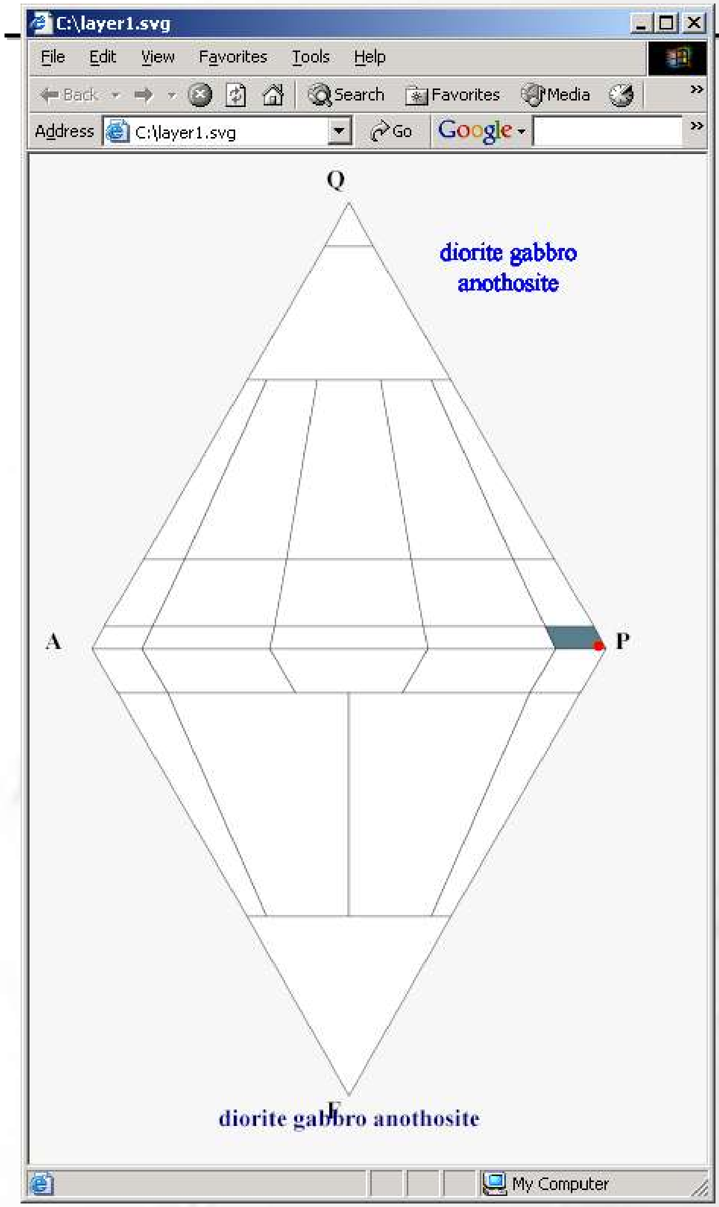
Retrieves classification Points from the Virginia Igneous Rock database and classifies it with the Igneous rocks diagrams.

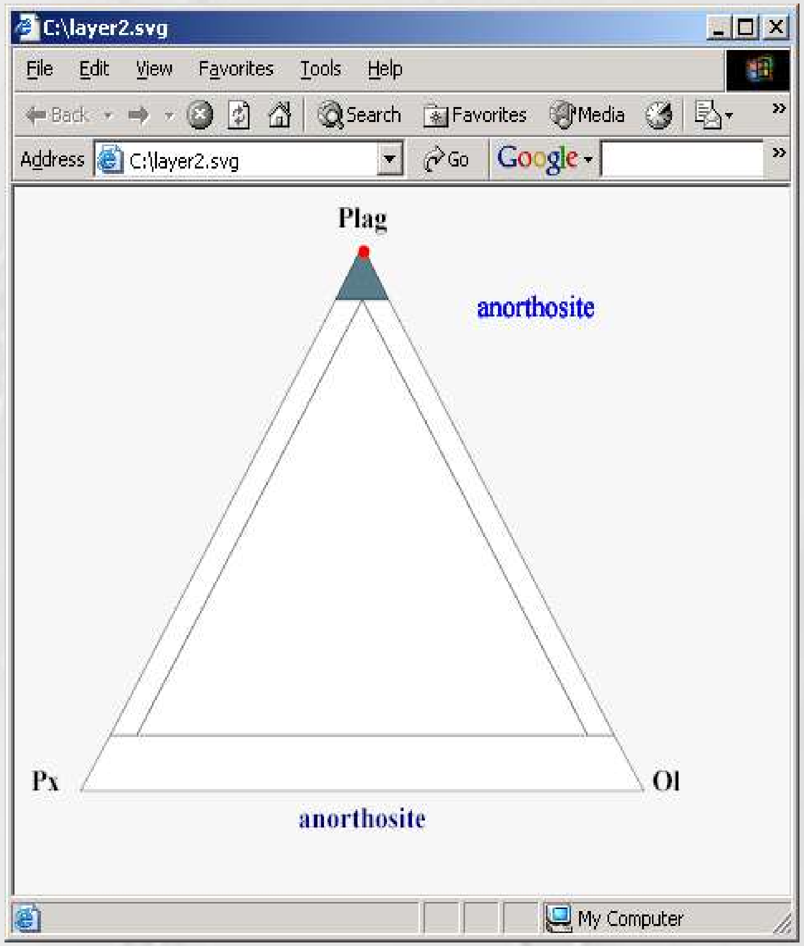
Diagrams and transitions between them.

… inside the Classifier

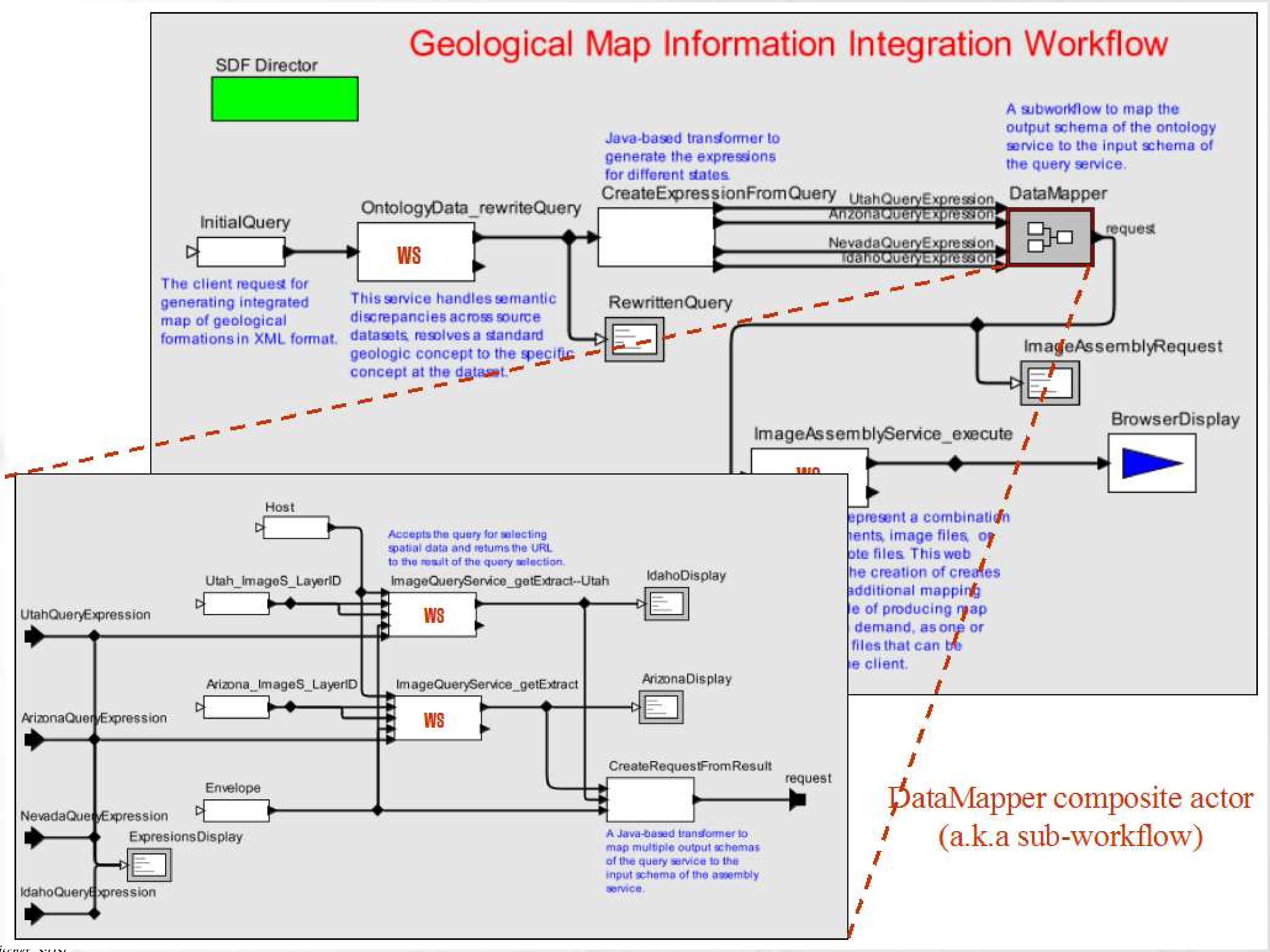


Standard BrowserUI: Client-Side SVG


C:\layer1.svg
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\layer1.svg Go Google
Q
A
P
F
diorite gabbro anothosite
diorite gabbro anothosite
My Computer


C:\layer2.svg
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\layer2.svg Go Google
Plag
anorthosite
Px
Ol
anorthosite
My Computer

SWF Reengineering *(GEON)*



The client request for generating integrated map of geological formations in XML format.

This service handles semantic discrepancies across source datasets, resolves a standard geologic concept to the specific concept at the dataset.

Java-based transformer to generate the expressions for different states.

A subworkflow to map the output schema of the ontology service to the input schema of the query service.

epresent a combination nents, image files, or ote files. This web he creation of creates additional mapping le of producing map demand, as one or files that can be e client.

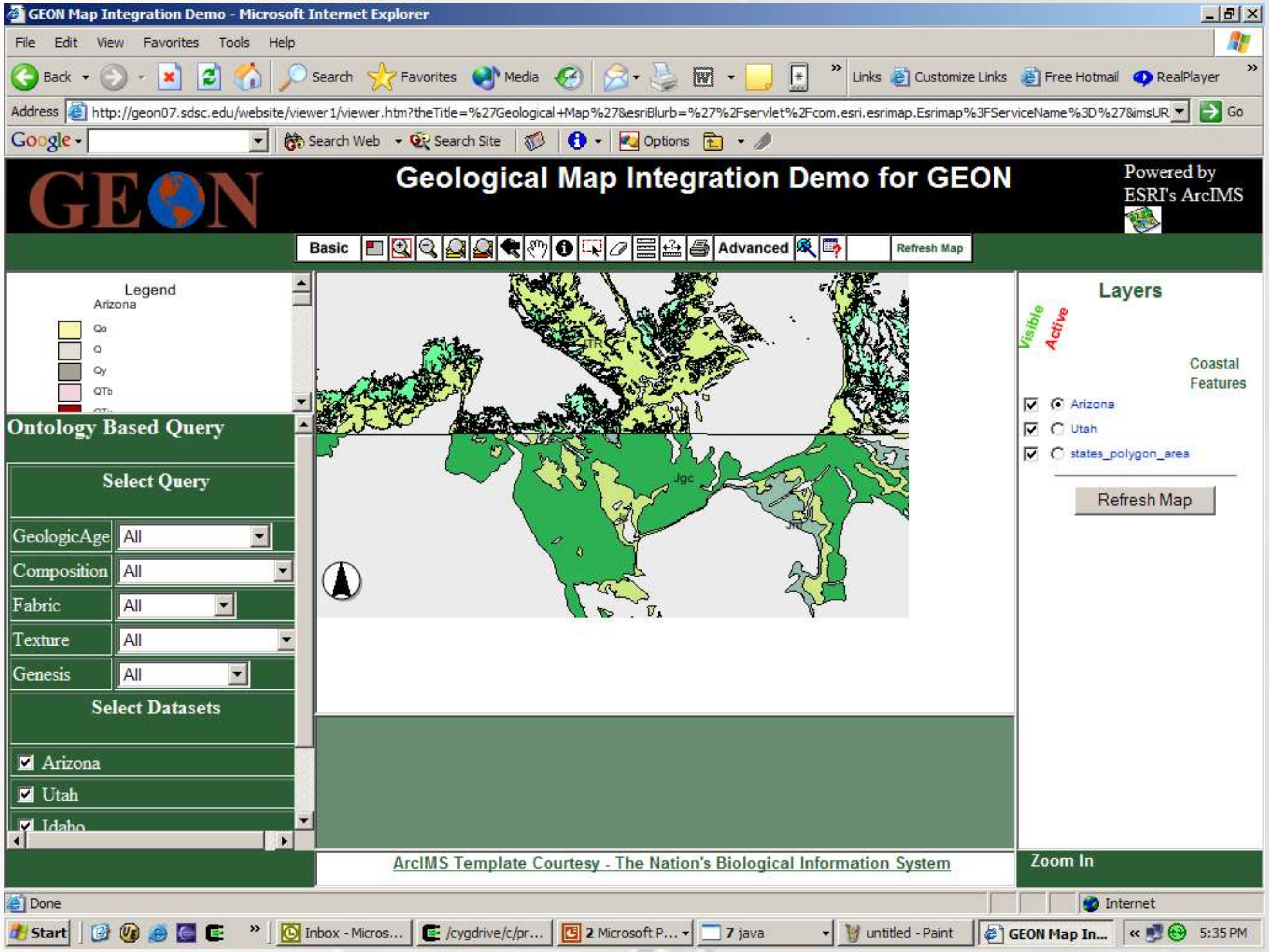
Accepts the query for selecting spatial data and returns the URL to the result of the query selection.

A Java-based transformer to map multiple output schemas of the query service to the input schema of the assembly service.

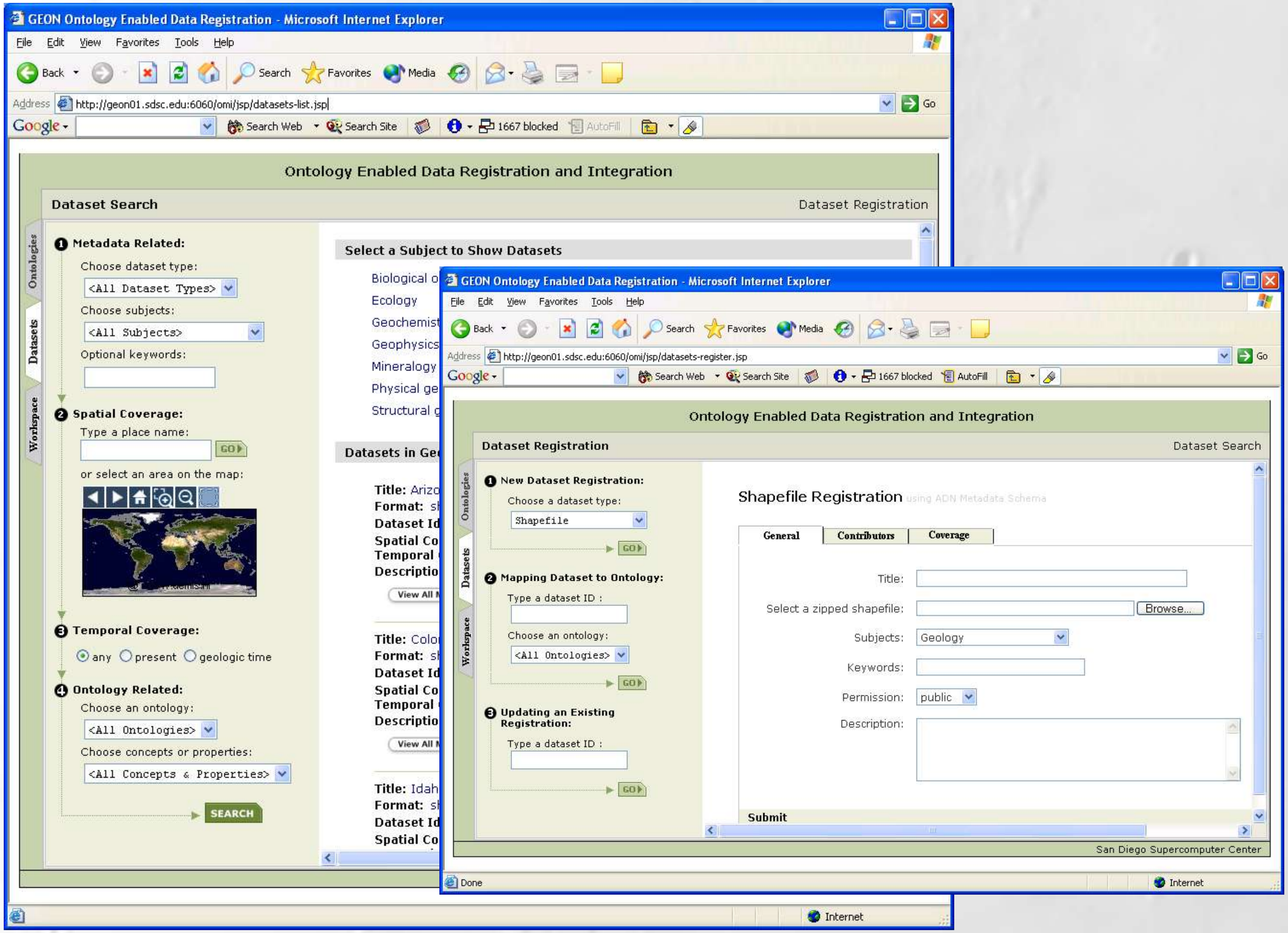
DataMapper composite actor (a.k.a sub-workflow)

Result launched via BrowserUI actor

(coupling with ESRI's ArcIMS)


GEON Map Integration Demo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Customize Links Free Hotmail RealPlayer
Address http://geon07.sdsc.edu/website/viewer1/viewer.htm?theTitle=%27Geological+Map%27&esriBlurb=%27%2Fservlet%2Fcom.esri.esrimap.Esrimap%3FServiceName%3D%27&imsUR
Go
Google Search Web Search Site Options
GEON
Geological Map Integration Demo for GEON
Powered by ESRI's ArcIMS
Basic Advanced Refresh Map
Legend
Arizona
Qa
Q
Qy
QTb
Ontology Based Query
Select Query
GeologicAge All
Composition All
Fabric All
Texture All
Genesis All
Select Datasets
Arizona
Utah
Idaho
Layers
Visible
Active
Coastal Features
Arizona
Utah
states_polygon_area
Refresh Map
ArcIMS Template Courtesy - The Nation's Biological Information System
Zoom In
Done
Internet
Start
Inbox - Micros...
/cygdrive/c/pr...
2 Microsoft P...
7 java
untitled - Paint
GEON Map In...
5:35 PM

Data Registration UI



GEON Ontology Enabled Data Registration - Microsoft Internet Explorer

Address http://geon01.sdsc.edu:6060/omi/jsp/datasets-list.jsp

Ontology Enabled Data Registration and Integration

Dataset Search | Dataset Registration

1 Metadata Related:
Choose dataset type: <All Dataset Types>
Choose subjects: <All Subjects>
Optional keywords:

2 Spatial Coverage:
Type a place name:
or select an area on the map:

3 Temporal Coverage:
[✓ any] present geologic time

4 Ontology Related:
Choose an ontology: <All Ontologies>
Choose concepts or properties: <All Concepts & Properties>

SEARCH

Select a Subject to Show Datasets

Biological o
Ecology
Geochemist
Geophysics
Mineralogy
Physical ge
Structural g

Datasets in Ge

Title: Arizo
Format: sl
Dataset Id
Spatial Co
Temporal
Descriptio

Title: Colo
Format: sl
Dataset Id
Spatial Co
Temporal
Descriptio

Title: Idah
Format: sl
Dataset Id
Spatial Co

GEON Ontology Enabled Data Registration - Microsoft Internet Explorer

Address http://geon01.sdsc.edu:6060/omi/jsp/datasets-register.jsp

Ontology Enabled Data Registration and Integration

Dataset Registration | Dataset Search

1 New Dataset Registration:
Choose a dataset type: Shapefile

2 Mapping Dataset to Ontology:
Type a dataset ID :
Choose an ontology: <All Ontologies>

3 Updating an Existing Registration:
Type a dataset ID :

Shapefile Registration using ADN Metadata Schema

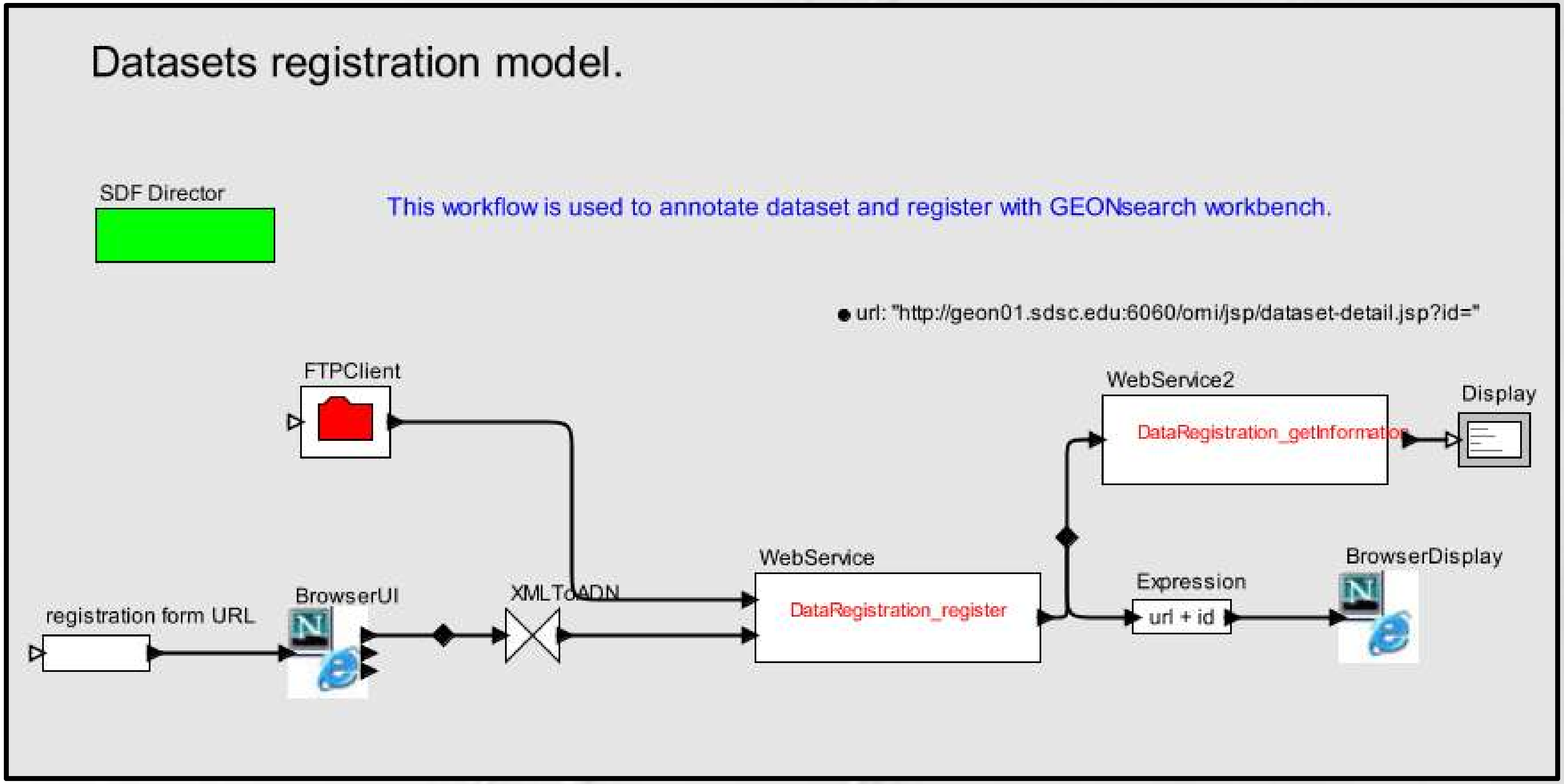
General | Contributors | Coverage

Title:	
Select a zipped shapefile:	Browse...
Subjects:	Geology
Keywords:	
Permission:	public
Description:	

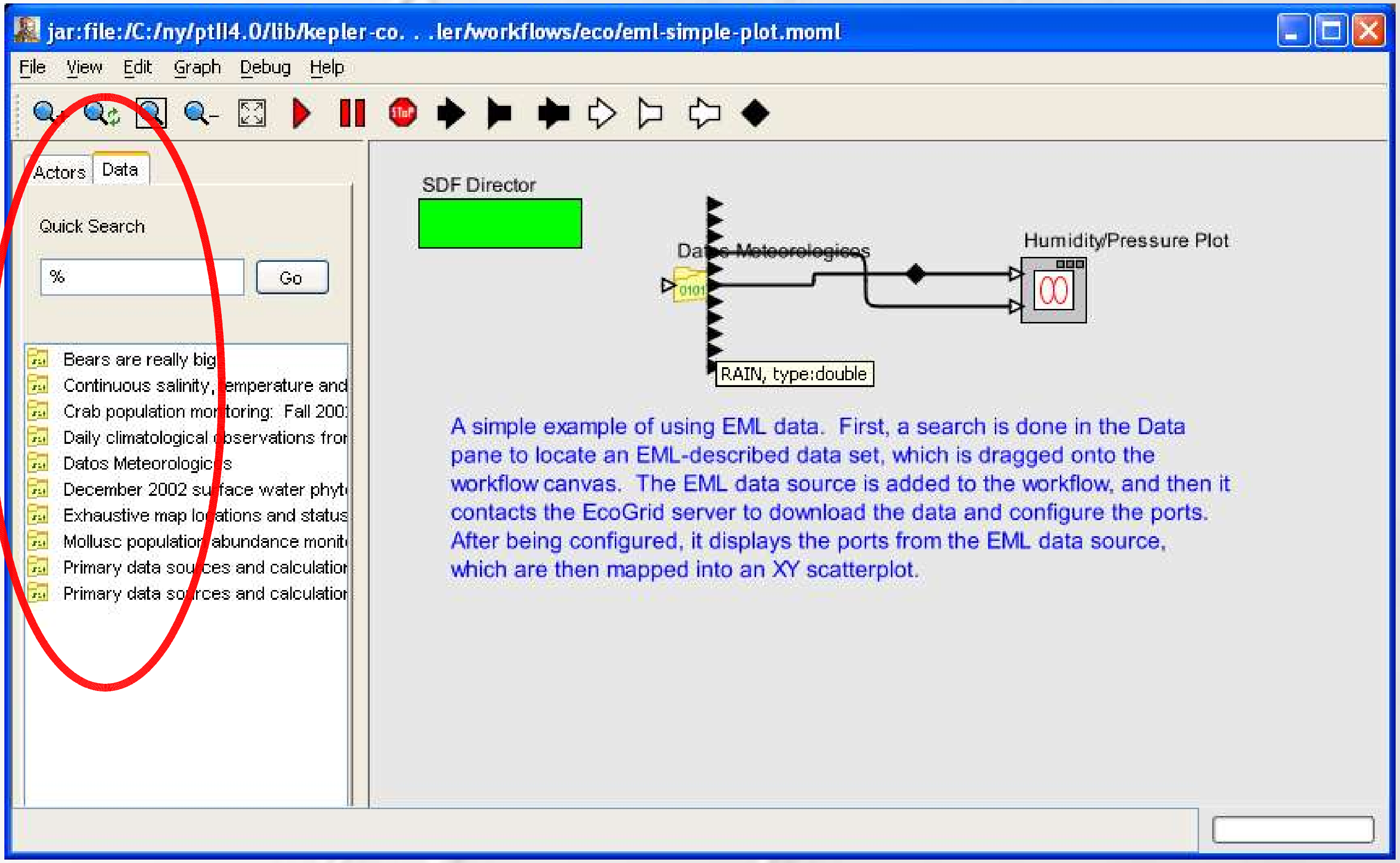
Submit

San Diego Supercomputer Center

Data Registration: as a KEPLER WF


Datasets registration model.
SDF Director
This workflow is used to annotate dataset and register with GEONsearch workbench.
● url: "http://geon01.sdsc.edu:6060/omi/jsp/dataset-detail.jsp?id="
FTPClient
WebService2
DataRegistration_getInformation
Display
WebService
DataRegistration_register
BrowserDisplay
Expression
url + id
registration form URL
BrowserUI
XMLToADN

Registered Data shows up in KEPLER

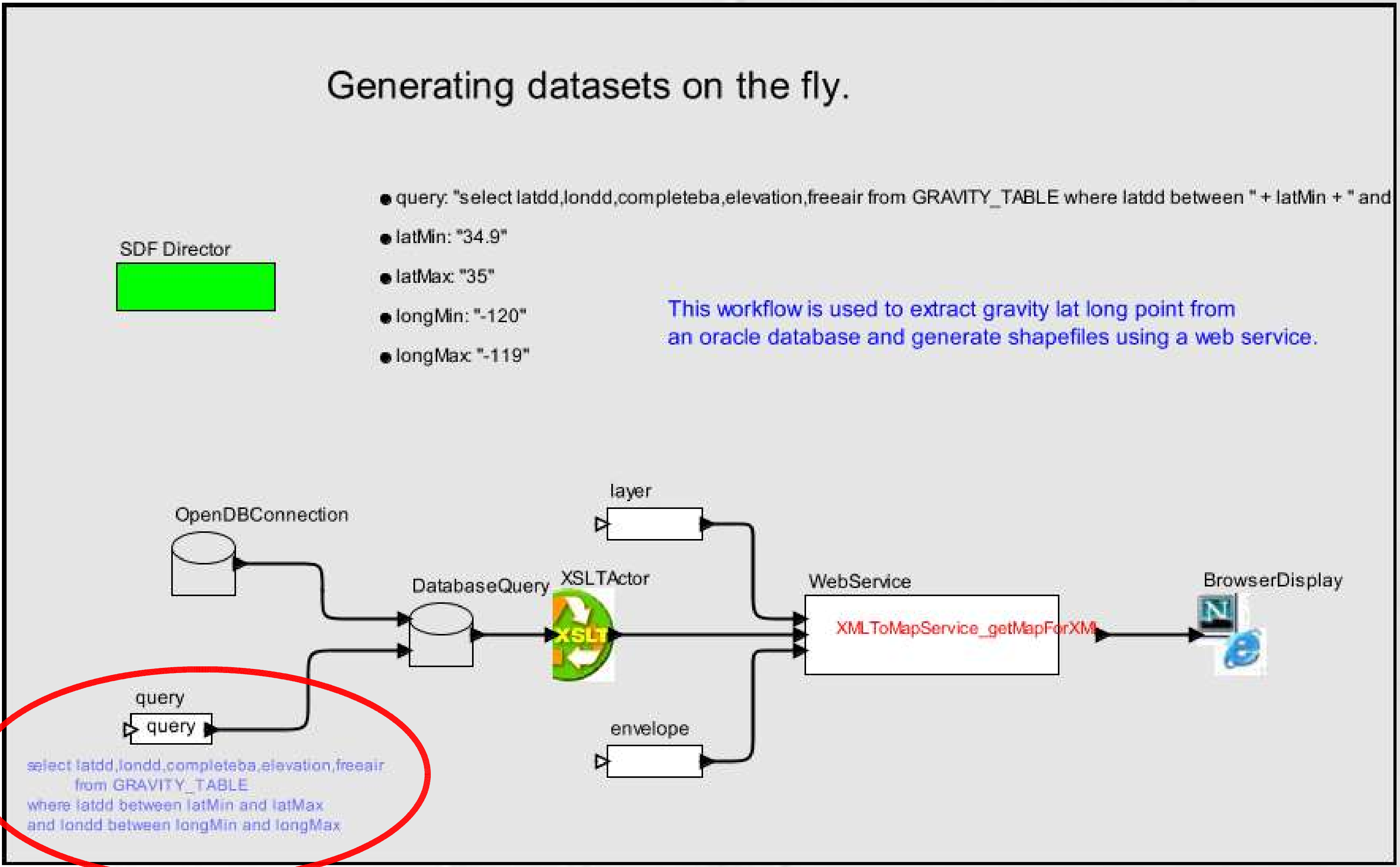

jar:file:/C:/ny/ptII4.0/lib/kepler-co. . .ler/workflows/eco/eml-simple-plot.moml
File View Edit Graph Debug Help
Actors Data
Quick Search
%
Go
SDF Director
Datos Meteorologicos
Humidity/Pressure Plot
RAIN, type:double

Bears are really big
Continuous salinity, temperature and
Crab population monitoring: Fall 200
Daily climatological observations fro
Datos Meteorologicos
December 2002 surface water phyt
Exhaustive map locations and status
Mollusc population abundance monit
Primary data sources and calculatio
Primary data sources and calculatio

A simple example of using EML data. First, a search is done in the Data pane to locate an EML-described data set, which is dragged onto the workflow canvas. The EML data source is added to the workflow, and then it contacts the EcoGrid server to download the data and configure the ports. After being configured, it displays the ports from the EML data source, which are then mapped into an XY scatterplot.

Kepler

More WF Plumbing



● query: "select latdd,londd,completeba,elevation,freeair from GRAVITY_TABLE where latdd between " + latMin + " and

● latMin: "34.9"

● latMax: "35"

● longMin: "-120"

● longMax: "-119"

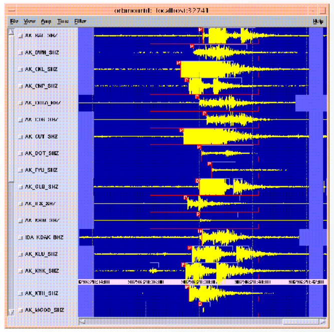
This workflow is used to extract gravity lat long point from an oracle database and generate shapefiles using a web service.

```
select latdd,londd,completeba,elevation,freeair
      from GRAVITY_TABLE
where latdd between latMin and latMax
and londd between longMin and longMax
```

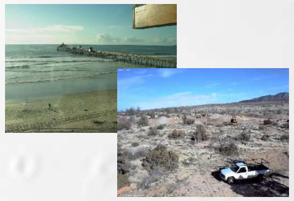
KEPLER & ROADNet Real-Time Scientific Workflows

Architecture:

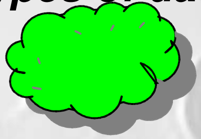
Seismic Waveforms



Images



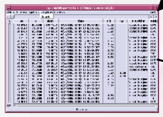
other types of data

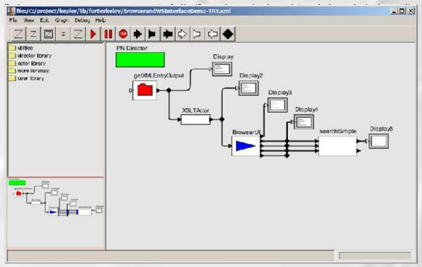


Real-time Packet Buffer

ORBserver

Near-real-time database

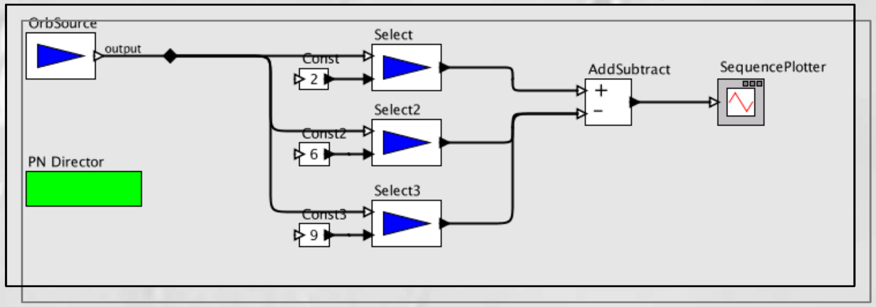




Scientific Workflow

Straightforward Example:

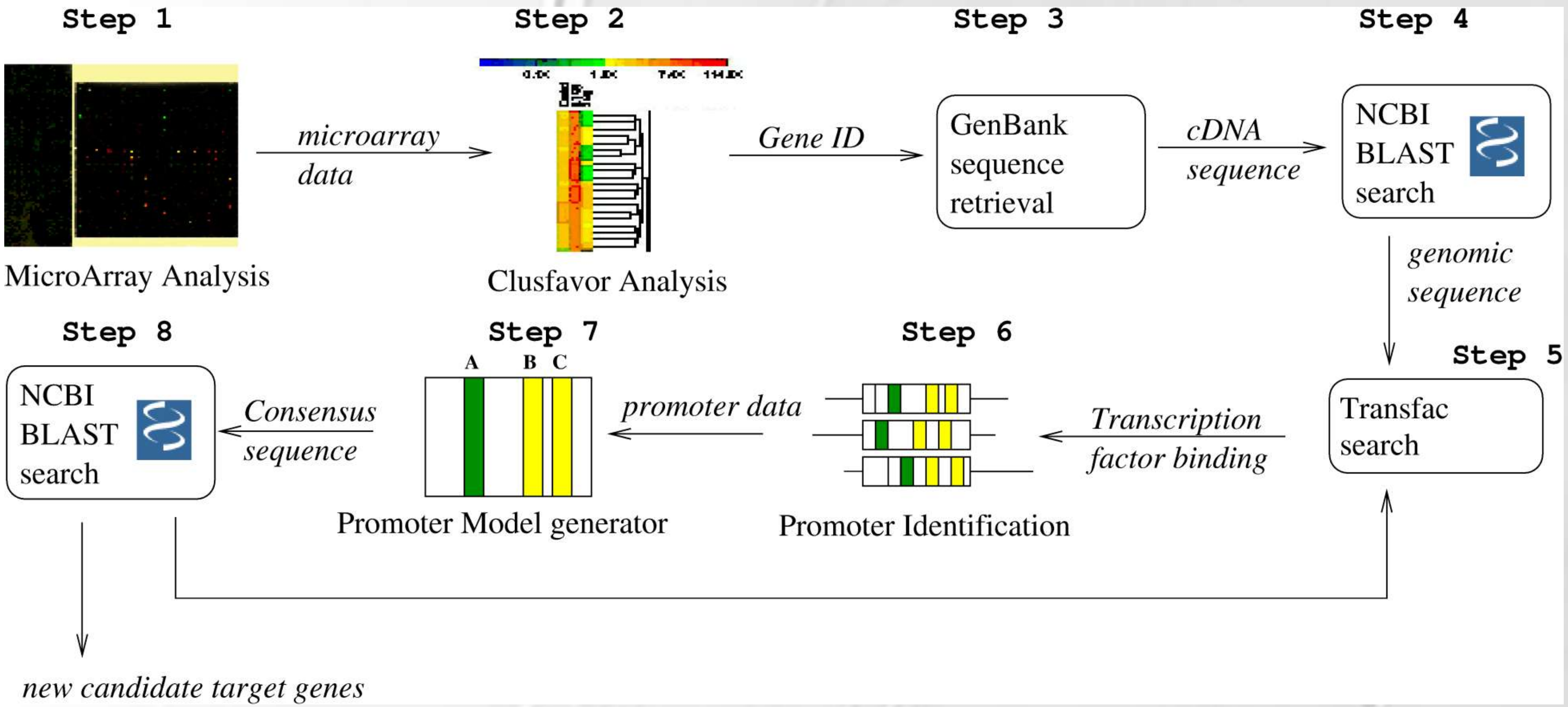
Laser Strainmeter Channels in;
Scientific Workflow;
Earth-tide signal out



Target Directions:

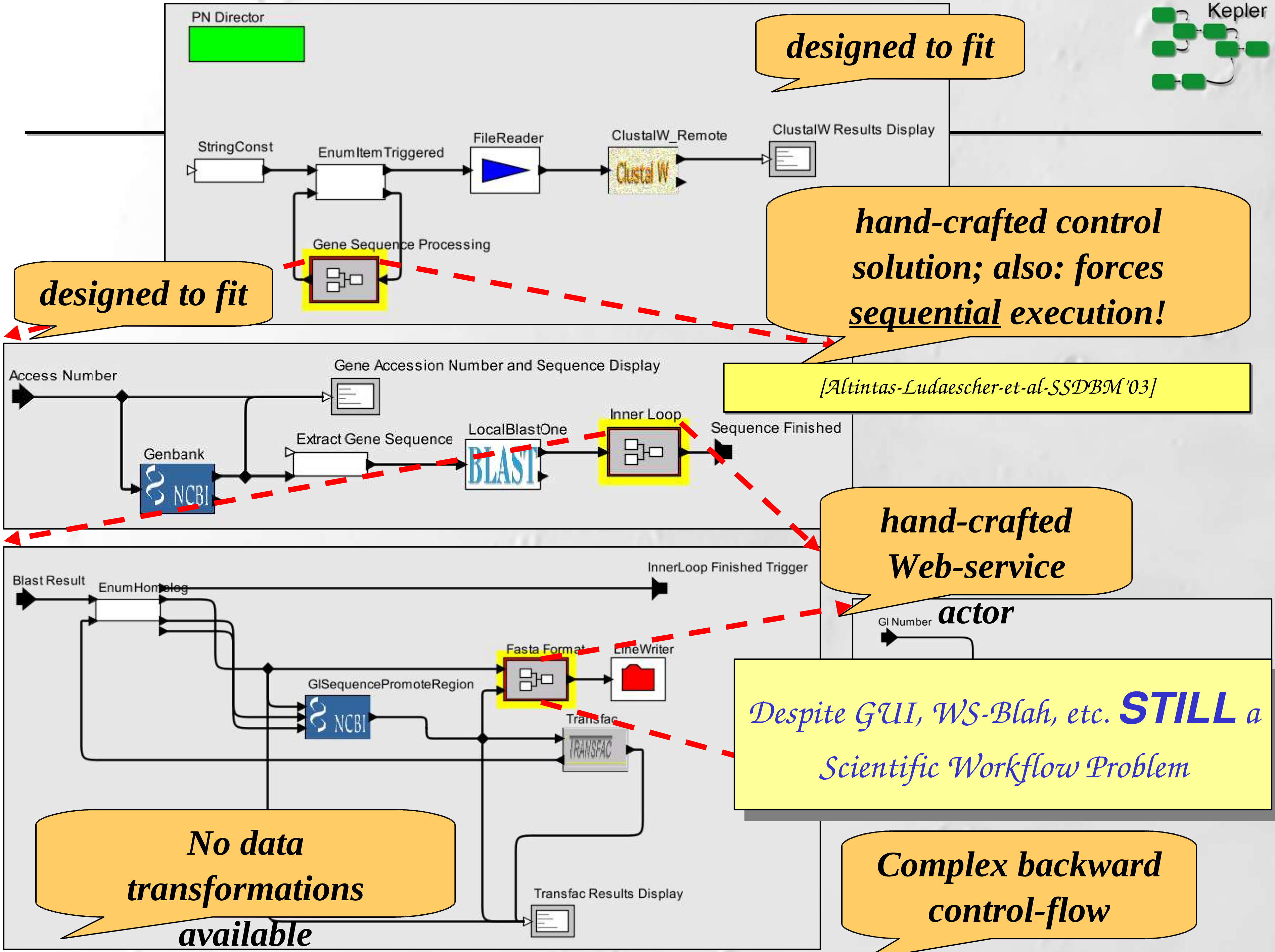
- Complex Processing Results
- Cross-disciplinary signals analysis
- Geophysical Stream Algebras

A Scientific Workflow Problem



Promoter Identification Workflow (PIW)

Source: Matt Coleman (LLNL)



designed to fit

hand-crafted control solution; also: forces <u>sequential</u> execution!

[Altintas-Ludaescher-et-al-SSDBM'03]

designed to fit

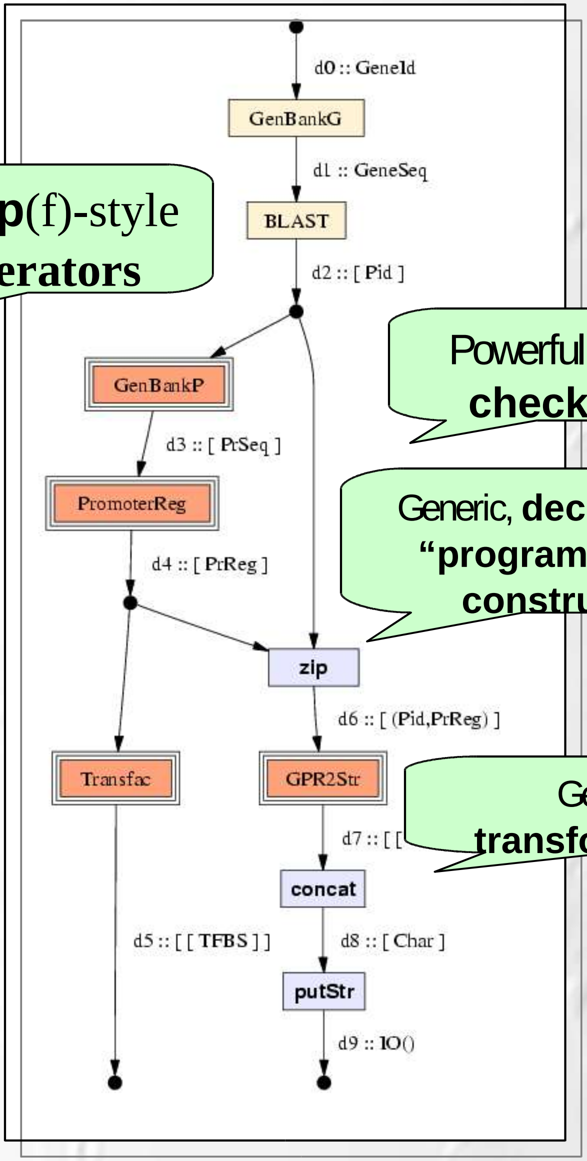
hand-crafted Web-service actor

Despite GUI, WS-Blah, etc. **STILL** *a Scientific Workflow Problem*

No data transformations available

Complex backward control-flow

A Scientific Workflow Problem: Solved



- *Solution based on declarative, functional dataflow process network*

  (= also a **data streaming model***!)*

- *Higher-order constructs:* **map** *(f)*

  ⇒ ***no control-flow spaghetti***

  ⇒ ***data-intensive apps***

  ⇒ ***free concurrent execution***

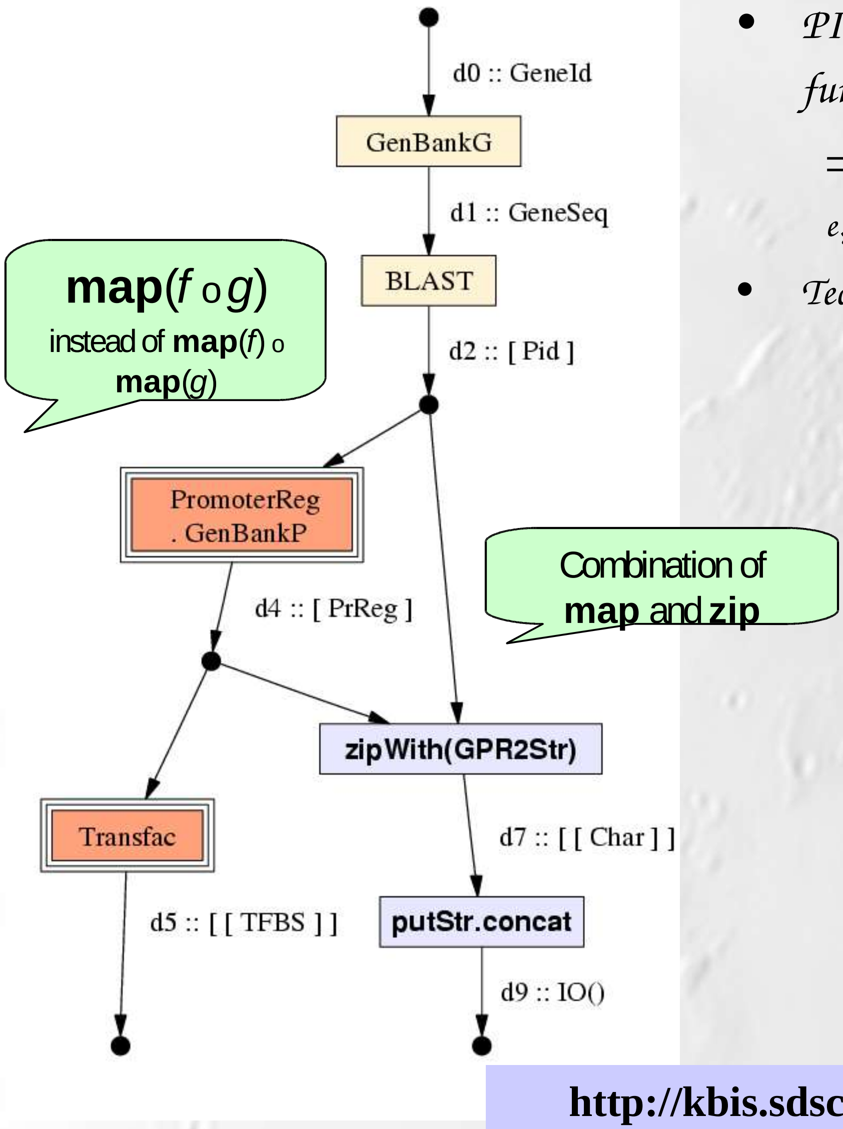
  ⇒ ***free type checking***

  ⇒ *automatic support to go from piw(GeneId) to*

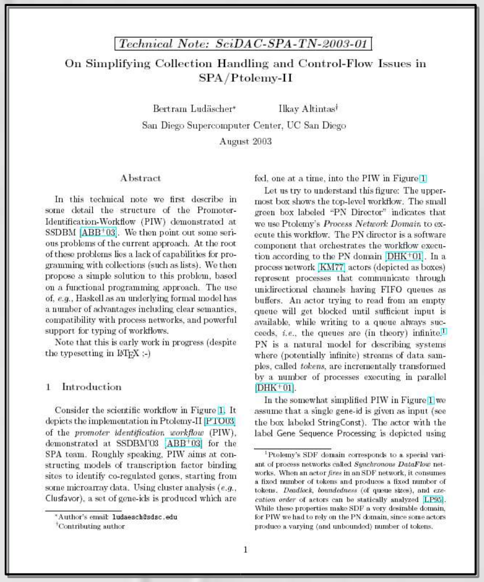
  *PIW :=***map***(piw) over [GeneId]*

B. Ludäscher, SDSC

Optimization by Declarative Rewriting I



- *PIW as a declarative, referentially transparent functional process*
  - *⇒ optimization via functional rewriting possible*
  - *e.g. map(f o g) = map(f) o map(g)*
- *Technical report &PIW specification in Haskell*



Technical Note: SciDAC-SPA-TN-2003-01

On Simplifying Collection Handling and Control-Flow Issues in SPA/Ptolemy-II

Bertram Ludäscher* Ilkay Altintas†

San Diego Supercomputer Center, UC San Diego

August 2003

Abstract

In this technical note we first describe in some detail the structure of the Promoter-Identification-Workflow (PIW) demonstrated at SSDBM [ABB+03]. We then point out some serious problems of the current approach. At the root of these problems lies a lack of capabilities for programming with collections (such as lists). We then propose a simple solution to this problem, based on a functional programming approach. The use of, *e.g.*, Haskell as an underlying formal model has a number of advantages including clear semantics, compatibility with process networks, and powerful support for typing of workflows.

Note that this is early work in progress (despite the typesetting in LaTeX ;-)

1 Introduction

Consider the scientific workflow in Figure 1. It depicts the implementation in Ptolemy-II [PTO03] of the *promoter identification workflow* (PIW), demonstrated at SSDBM'03 [ABB+03] for the SPA team. Roughly speaking, PIW aims at constructing models of transcription factor binding sites to identify co-regulated genes, starting from some microarray data. Using cluster analysis (*e.g.*, Clusfavor), a set of gene-ids is produced which are fed, one at a time, into the PIW in Figure 1.

Let us try to understand this figure: The uppermost box shows the top-level workflow. The small green box labeled "PN Director" indicates that we use Ptolemy's *Process Network Domain* to execute this workflow. The PN director is a software component that orchestrates the workflow execution according to the PN domain [DHK+01]. In a process network [KM77] actors (depicted as boxes) represent processes that communicate through unidirectional channels having FIFO queues as buffers. An actor trying to read from an empty queue will get blocked until sufficient input is available, while writing to a queue always succeeds, *i.e.*, the queues are (in theory) infinite.[1] PN is a natural model for describing systems where (potentially infinite) streams of data samples, called *tokens*, are incrementally transformed by a number of processes executing in parallel [DHK+01].

In the somewhat simplified PIW in Figure 1 we assume that a single gene-id is given as input (see the box labeled StringConst). The actor with the label Gene Sequence Processing is depicted using

*Author's email: ludaesch@sdsc.edu
†Contributing author

[1] Ptolemy's SDF domain corresponds to a special variant of process networks called *Synchronous DataFlow* networks. When an actor *fires* in an SDF network, it consumes a fixed number of tokens and produces a fixed number of tokens. *Deadlock*, *boundedness* (of queue sizes), and *execution order* of actors can be statically analyzed [LP95]. While these properties make SDF a very desirable domain, for PIW we had to rely on the PN domain, since some actors produce a varying (and unbounded) number of tokens.

1

http://kbis.sdsc.edu/SciDAC-SDM/scidac-tn-map-constructs.pdf

Optimizing II: Streams & Pipelines

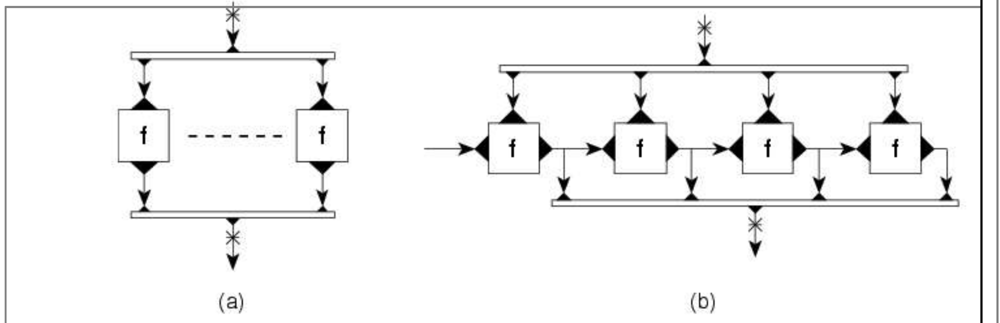
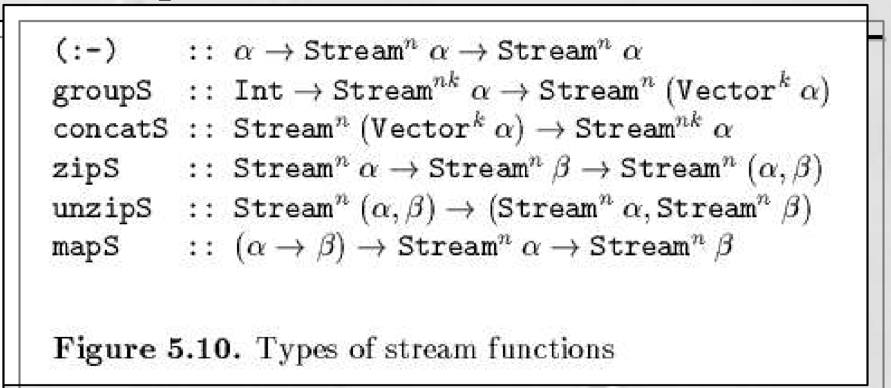


Figure 4.24. Unfolded higher-order functions: a) *map*; b) *scanl*



```
(:-)     :: α → Stream^n α → Stream^n α
groupS   :: Int → Stream^nk α → Stream^n (Vector^k α)
concatS  :: Stream^n (Vector^k α) → Stream^nk α
zipS     :: Stream^n α → Stream^n β → Stream^n (α, β)
unzipS   :: Stream^n (α, β) → (Stream^n α, Stream^n β)
mapS     :: (α → β) → Stream^n α → Stream^n β
```

Figure 5.10. Types of stream functions

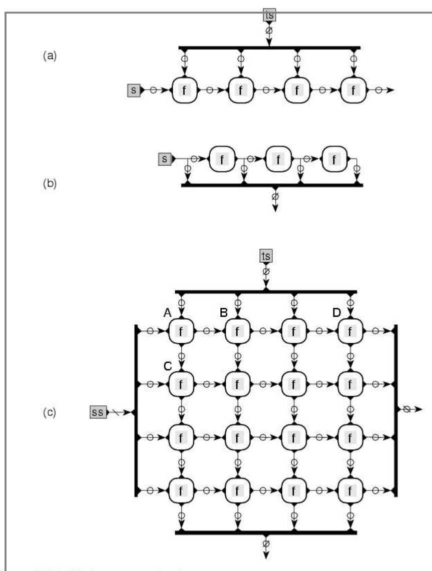
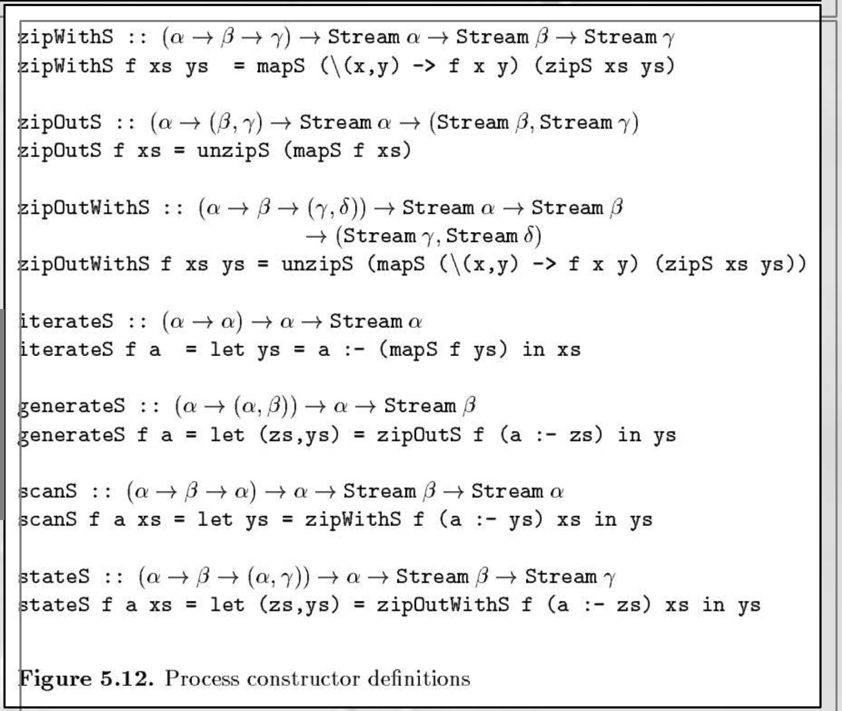


Figure 5.19. Mesh process networks

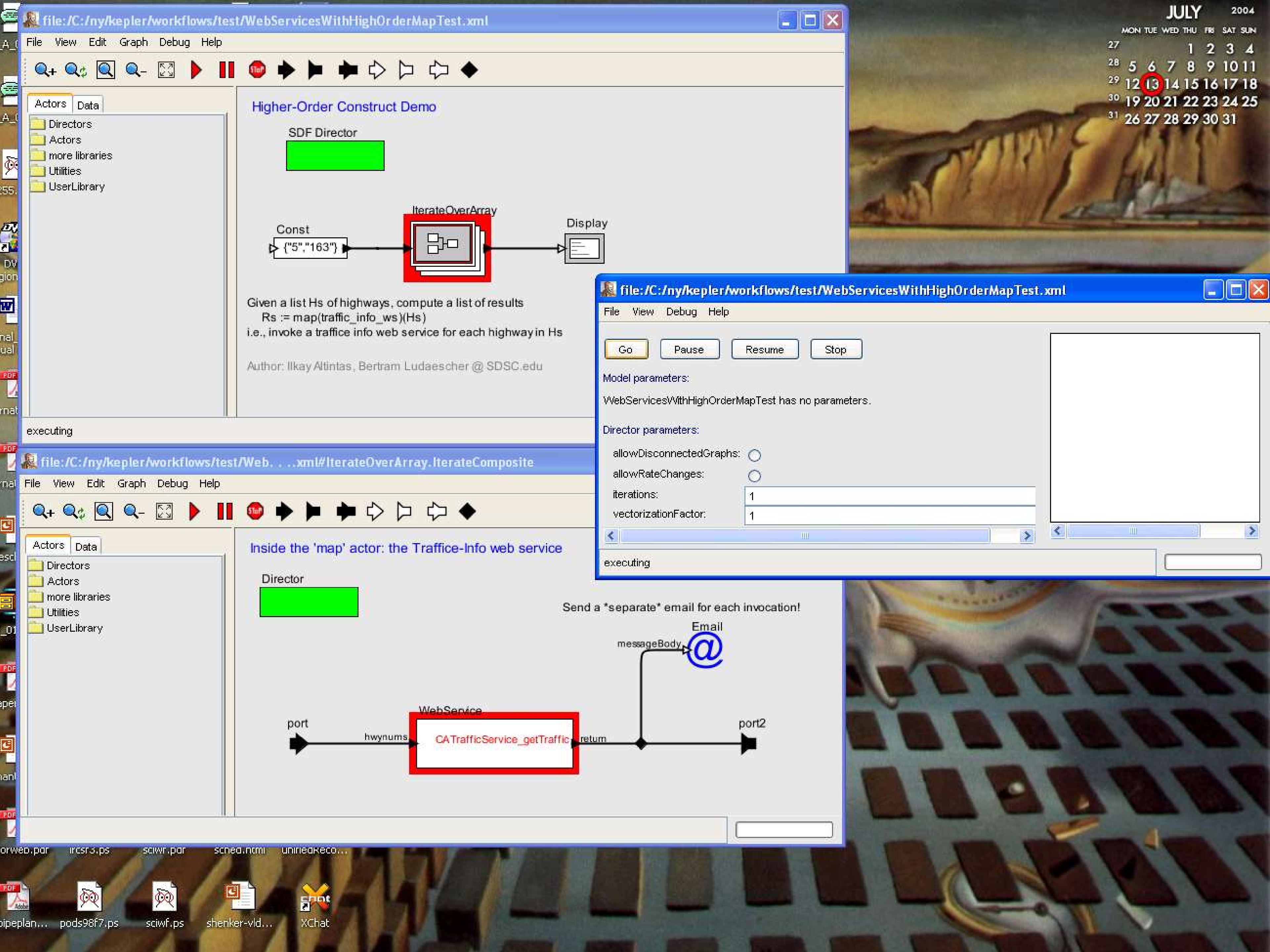
Source: Real-Time Signal Processing: Dataflow, Visual, and Functional Programming, Hideki John Reekie, University of Technology, Sydney



```
zipWithS :: (α → β → γ) → Stream α → Stream β → Stream γ
zipWithS f xs ys  = mapS (\(x,y) -> f x y) (zipS xs ys)

zipOutS :: (α → (β, γ) → Stream α → (Stream β, Stream γ)
zipOutS f xs = unzipS (mapS f xs)

zipOutWithS :: (α → β → (γ, δ)) → Stream α → Stream β
                          → (Stream γ, Stream δ)
zipOutWithS f xs ys = unzipS (mapS (\(x,y) -> f x y) (zipS xs ys))

iterateS :: (α → α) → α → Stream α
iterateS f a  = let ys = a :- (mapS f ys) in xs

generateS :: (α → (α, β)) → α → Stream β
generateS f a = let (zs,ys) = zipOutS f (a :- zs) in ys

scanS :: (α → β → α) → α → Stream β → Stream α
scanS f a xs = let ys = zipWithS f (a :- ys) xs in ys

stateS :: (α → β → (α, γ)) → α → Stream β → Stream γ
stateS f a xs = let (zs,ys) = zipOutWithS f (a :- zs) xs in ys
```

Figure 5.12. Process constructor definitions

- *Clean functional semantics facilitates algebraic workflow (program) transformations (Bird-Meertens); e.g. mapS f • mapS g ➔ mapS (f • g)*



Higher-Order Construct Demo

Given a list Hs of highways, compute a list of results
Rs := map(traffic_info_ws)(Hs)
i.e., invoke a traffice info web service for each highway in Hs

Author: Ilkay Altintas, Bertram Ludaescher @ SDSC.edu

Inside the 'map' actor: the Traffice-Info web service

Send a *separate* email for each invocation!

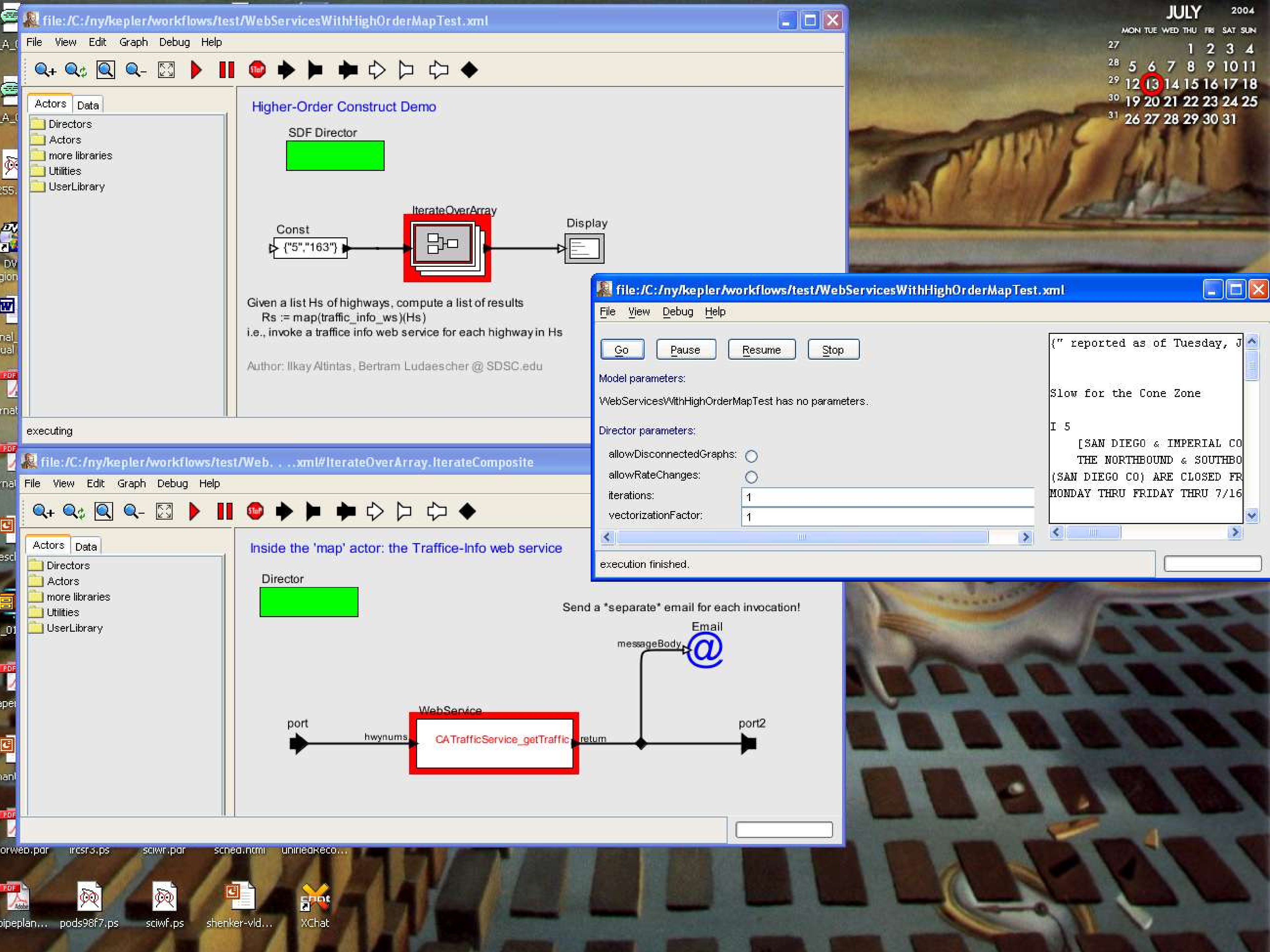
Model parameters:

WebServicesWithHighOrderMapTest has no parameters.

Director parameters:

allowDisconnectedGraphs:	○
allowRateChanges:	○
iterations:	1
vectorizationFactor:	1

executing



file:/C:/ny/kepler/workflows/test/WebServicesWithHighOrderMapTest.xml

Higher-Order Construct Demo

Given a list Hs of highways, compute a list of results
Rs := map(traffic_info_ws)(Hs)
i.e., invoke a traffice info web service for each highway in Hs

Author: Ilkay Altintas, Bertram Ludaescher @ SDSC.edu

executing

file:/C:/ny/kepler/workflows/test/Web. . . .xml#IterateOverArray.IterateComposite

Inside the 'map' actor: the Traffice-Info web service

Send a *separate* email for each invocation!

file:/C:/ny/kepler/workflows/test/WebServicesWithHighOrderMapTest.xml

Model parameters:

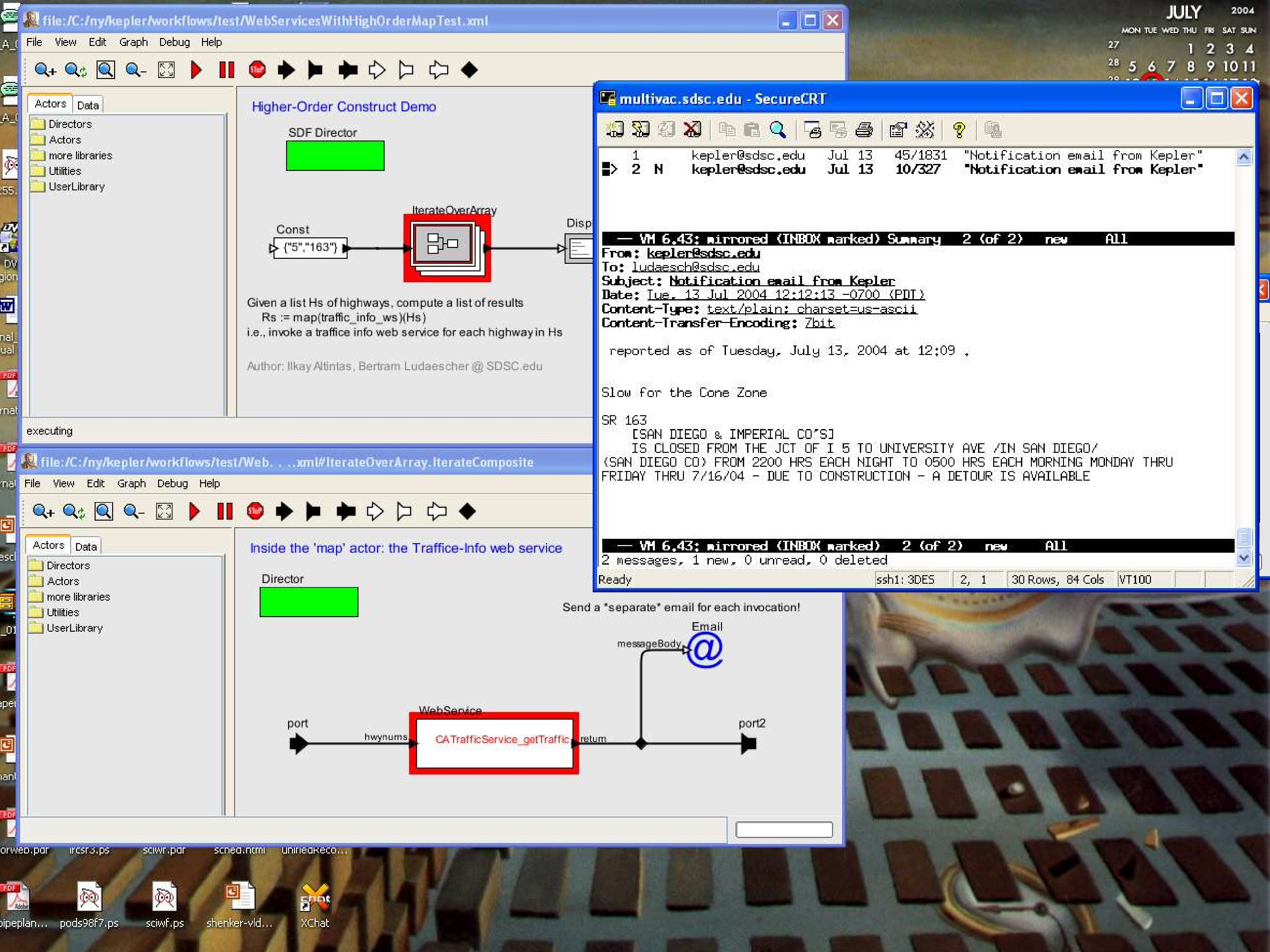
WebServicesWithHighOrderMapTest has no parameters.

Director parameters:

allowDisconnectedGraphs: ○
allowRateChanges: ○
iterations: 1
vectorizationFactor: 1

```
{" reported as of Tuesday, J


Slow for the Cone Zone


I 5
    [SAN DIEGO & IMPERIAL CO
    THE NORTHBOUND & SOUTHBO
(SAN DIEGO CO) ARE CLOSED FR
MONDAY THRU FRIDAY THRU 7/16
```

execution finished.


file:/C:/ny/kepler/workflows/test/WebServicesWithHighOrderMapTest.xml
File View Edit Graph Debug Help
Actors Data
Directors
Actors
more libraries
Utilities
UserLibrary
SDF Director
IterateOverArray
Const
{"5","163"}
Disp
executing
file:/C:/ny/kepler/workflows/test/Web. . . .xml#IterateOverArray.IterateComposite
Director
Email
messageBody
WebService
port
hwynums
CATrafficService_getTraffic
return
port2
multivac.sdsc.edu - SecureCRT
JULY 2004
MON TUE WED THU FRI SAT SUN

Higher-Order Construct Demo

Given a list Hs of highways, compute a list of results
Rs := map(traffic_info_ws)(Hs)
i.e., invoke a traffice info web service for each highway in Hs

Author: Ilkay Altintas, Bertram Ludaescher @ SDSC.edu

Inside the 'map' actor: the Traffice-Info web service

Send a *separate* email for each invocation!

```
   1      kepler@sdsc.edu   Jul 13   45/1831  "Notification email from Kepler"
=> 2  N   kepler@sdsc.edu   Jul 13   10/327   "Notification email from Kepler"

 -- VM 6.43: mirrored (INBOX marked) Summary   2 (of 2)   new     All
From: kepler@sdsc.edu
To: ludaesch@sdsc.edu
Subject: Notification email from Kepler
Date: Tue, 13 Jul 2004 12:12:13 -0700 (PDT)
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

 reported as of Tuesday, July 13, 2004 at 12:09 .


Slow for the Cone Zone

SR 163
    [SAN DIEGO & IMPERIAL CO'S]
    IS CLOSED FROM THE JCT OF I 5 TO UNIVERSITY AVE /IN SAN DIEGO/
(SAN DIEGO CO) FROM 2200 HRS EACH NIGHT TO 0500 HRS EACH MORNING MONDAY THRU
FRIDAY THRU 7/16/04 - DUE TO CONSTRUCTION - A DETOUR IS AVAILABLE



 -- VM 6.43: mirrored (INBOX marked)   2 (of 2)   new     All
2 messages, 1 new, 0 unread, 0 deleted
Ready                              ssh1: 3DES   2, 1   30 Rows, 84 Cols   VT100
```

KEPLER Today

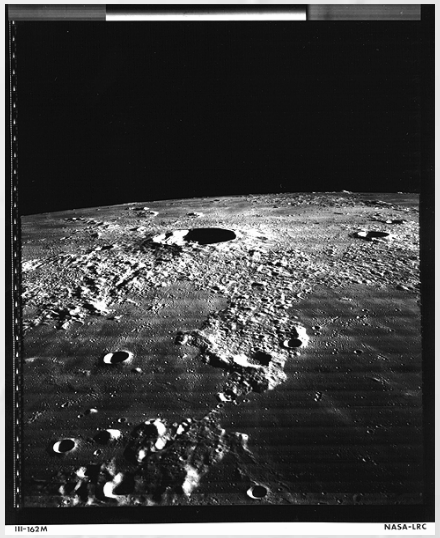
- **Lots of Ptolemy II goodies!**

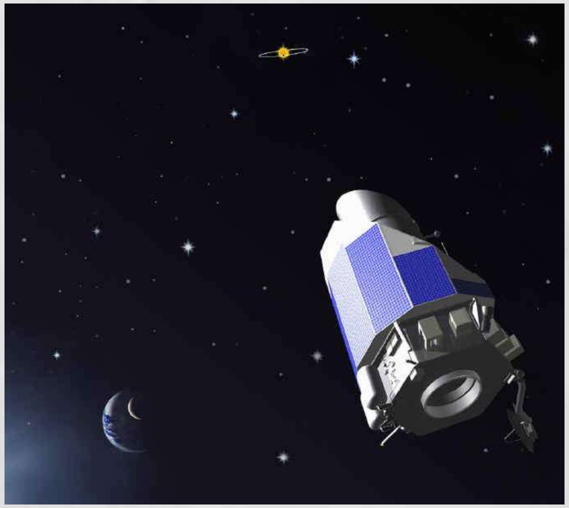
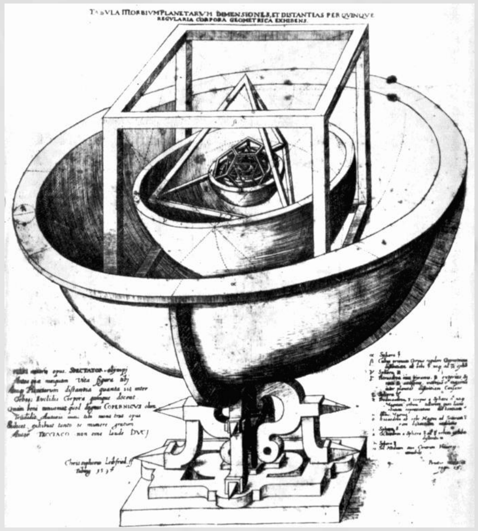
- **Coarse-grained** *scientific workflows, e.g.,*
  - *web service actors, grid actors, command-line actors*
  - *…*
- **Fine grained** *workflows and simulations, e.g.,*
  - *CT predator/prey model (already in Ptolemy)*
  - *Database access, XSLT transformations, …*
- *Special extensions*
  - **Real-time data streaming** *(ROADNet)*
  - *Special end-user extensions (e.g. GEON, SEEK)*

KEPLER Tomorrow

- *More* **generic support** *for*
  - **data-intensive** *and*
  - **compute-intensive** *workflows*
- *Special workflow* **deployment modes**
  - *Pack maximal non-interactive components into exportable web services*
  - *Take into account cost models, load balancing, …*
- *Extended type system with* **semantic types**
- *… and much more!*

Semantics: What's in a name?

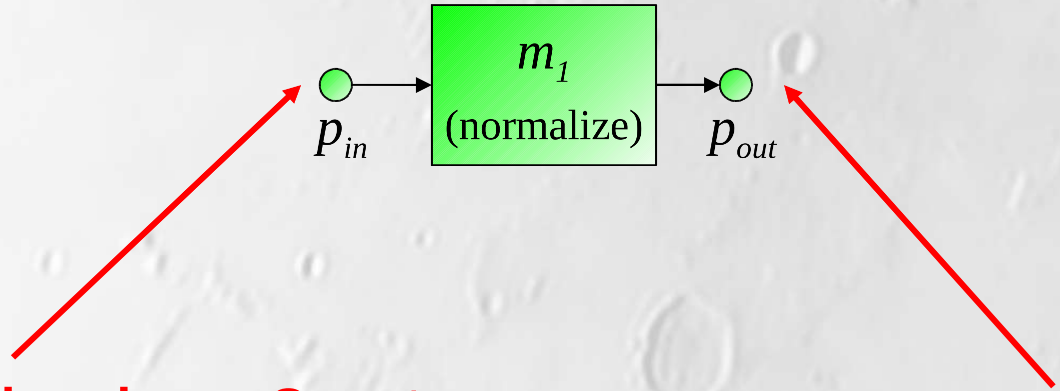


- *XML is the silver bullet, right?*
  - *<tag>Kepler</tag>*
- *What 'Kepler' are we talking about here??*
  - *Historic person, crater, space craft, workflow system, …*

KEPLER adds (will add) **Semantics Types**

- *Take concepts and relationships from an ontology to "semantically type" the data-in/out ports*
- *Application: e.g., design support:*
  - *smart/semi-automatic wiring, generation of "massaging actors"*

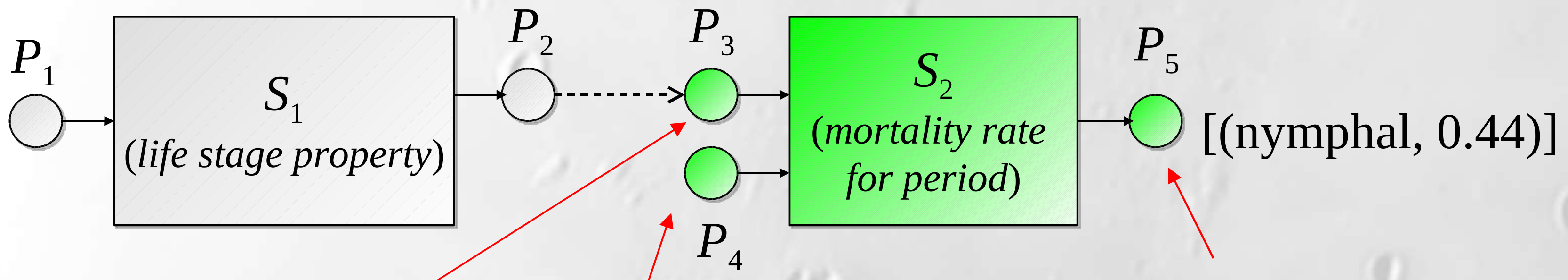


Takes **Abundance Count Measurements** ***for*** **`Life Stages`**

Returns **Mortality Rate** ***Derived Measurements for*** **Life Stages**

Source: [Bowers-Ludaescher, DILS'04]

A Simple SEEK Workflow Example



observations

life stage periods

k-value for each period of observation

Phase	Observed
Eggs	44,000
Instar I	3,513
Instar II	2,529
Instar III	1,922
Instar IV	1,461
Adults	1,300

Period	Phases
Nymphal	{Instar I, Instar II, Instar III, Instar IV}

Periods of development in terms of phases

Population samples for life stages of the common field grasshopper [Begon et al, 1996]

Source: [Bowers-Ludaescher, DILS'04]

Example Structural Types (XML)

structType(P_2)

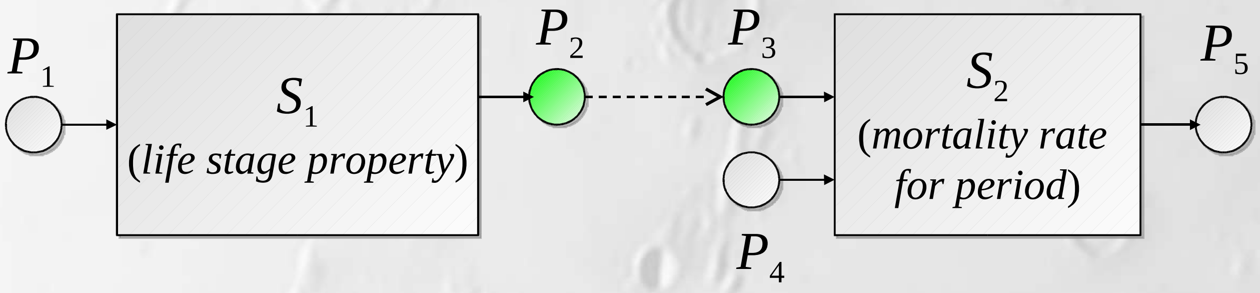
```
root population = (sample)*
elem sample     = (meas, lsp)
elem meas       = (cnt, acc)
elem cnt        = xsd:integer
elem acc        = xsd:double
elem lsp        = xsd:string
```

```
<population>
   <sample>
      <meas>
         <cnt>44,000</cnt>
         <acc>0.95</acc>
      </meas>
      <lsp>Eggs</lsp>
   </sample>
   …
<population>
```

structType(P_3)

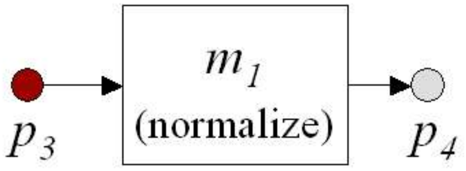
```
root cohortTable= (measurement)*
elem measuremnt = (phase, obs)
elem phase      = xsd:string
elem obs        = xsd:integer
```

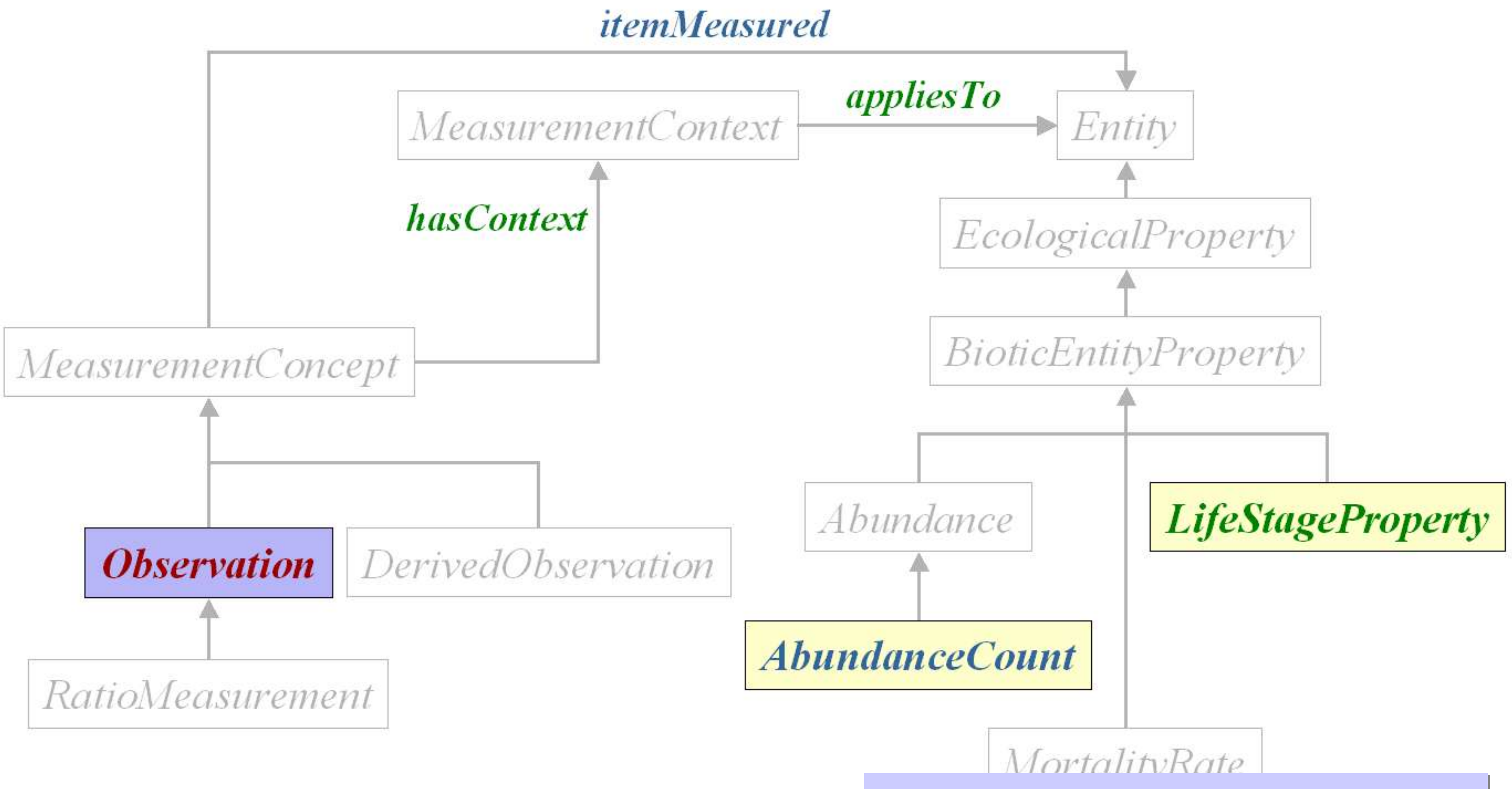
```
<cohortTable>
   <measurement>
      <phase>Eggs</cnt>
      <obs>44,000</acc>
   </measurement>
…
<cohortTable>
```



Source: [Bowers-Ludaescher, DILS'04]

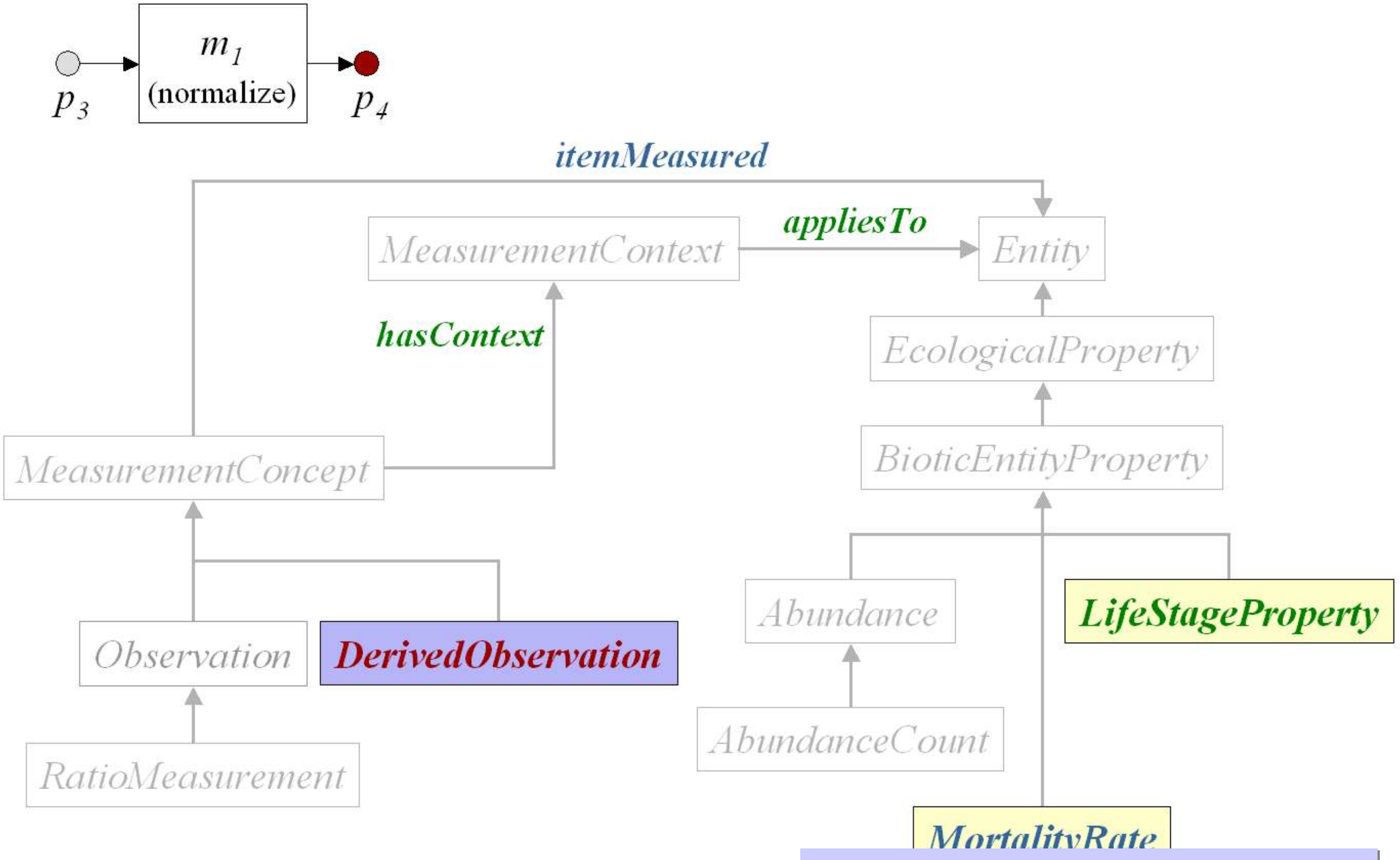
Selecting Concepts and Relationships





Source: [Bowers-Ludaescher, DILS'04]

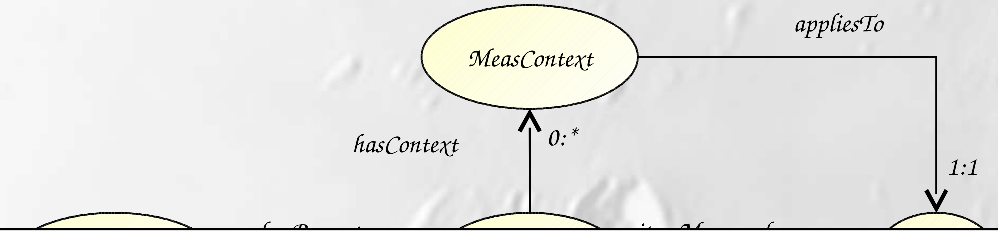
Selecting Concepts and Relationships



Source: [Bowers-Ludaescher, DILS'04]

Example Semantic Types

Portion of SEEK measurement ontology



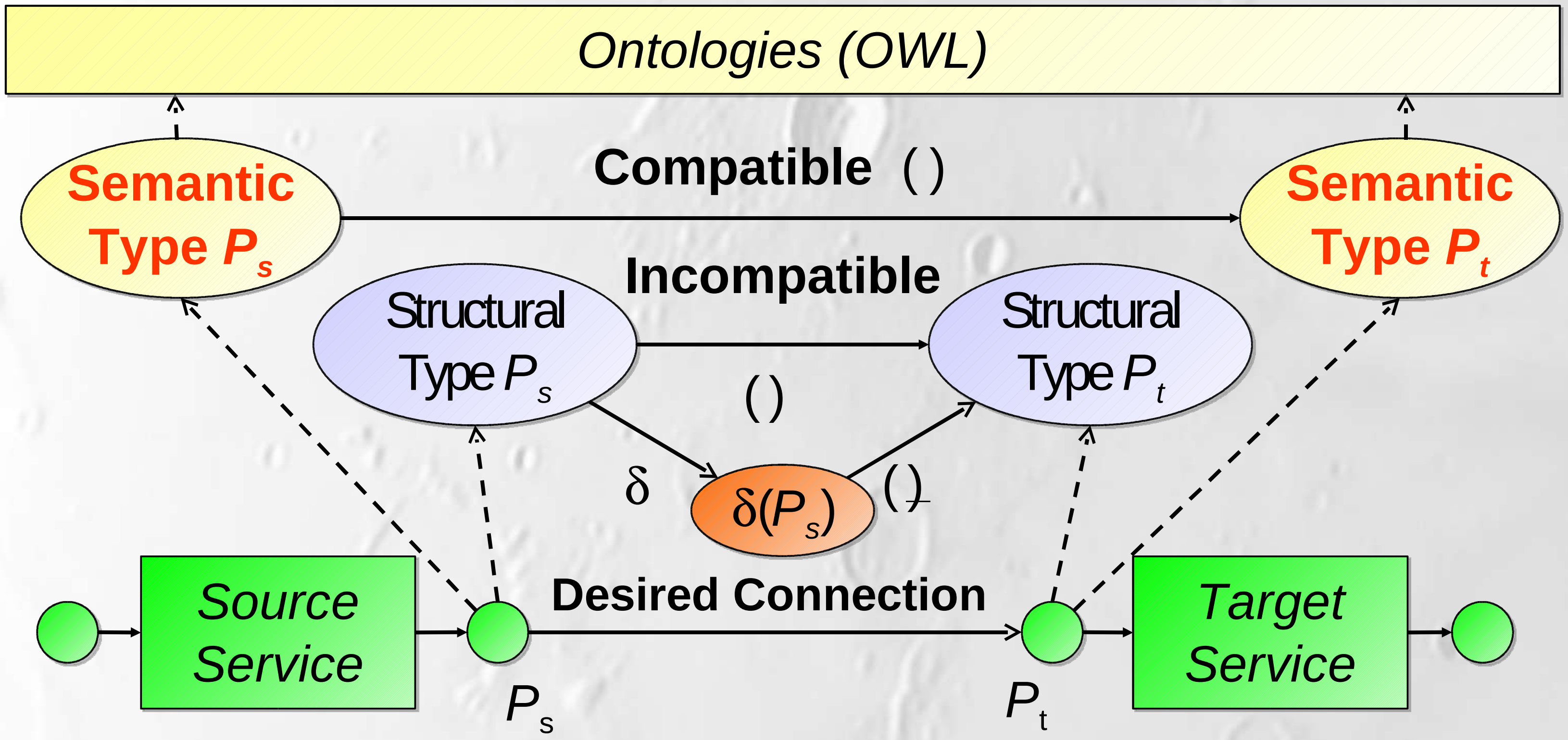
Same in OWL, a description logic standard (here, Sparrow syntax):

```
      Observation subClassOf forall hasContext/MeasContext and
                             forall hasProperty/MeasProperty and
                             exists itemMeasured/Entity.

      MeasContext subClassOf exists appliesTo/Entity and
                             atmost 1/appliesTo.

EcologicalProperty subClassOf Entity.

 LifeStageProperty subClassOf EcologicalProperty.

    AbundanceCount subClassOf EcologicalProperty and
                             exists hasLocation/SpatialLocation and
                             atMost 1/hasLocation and
                             exists hasCount/NumericValue and
                             atMost 1/hasCount.
```

Source: [Bowers-Ludaescher, DILS'04]

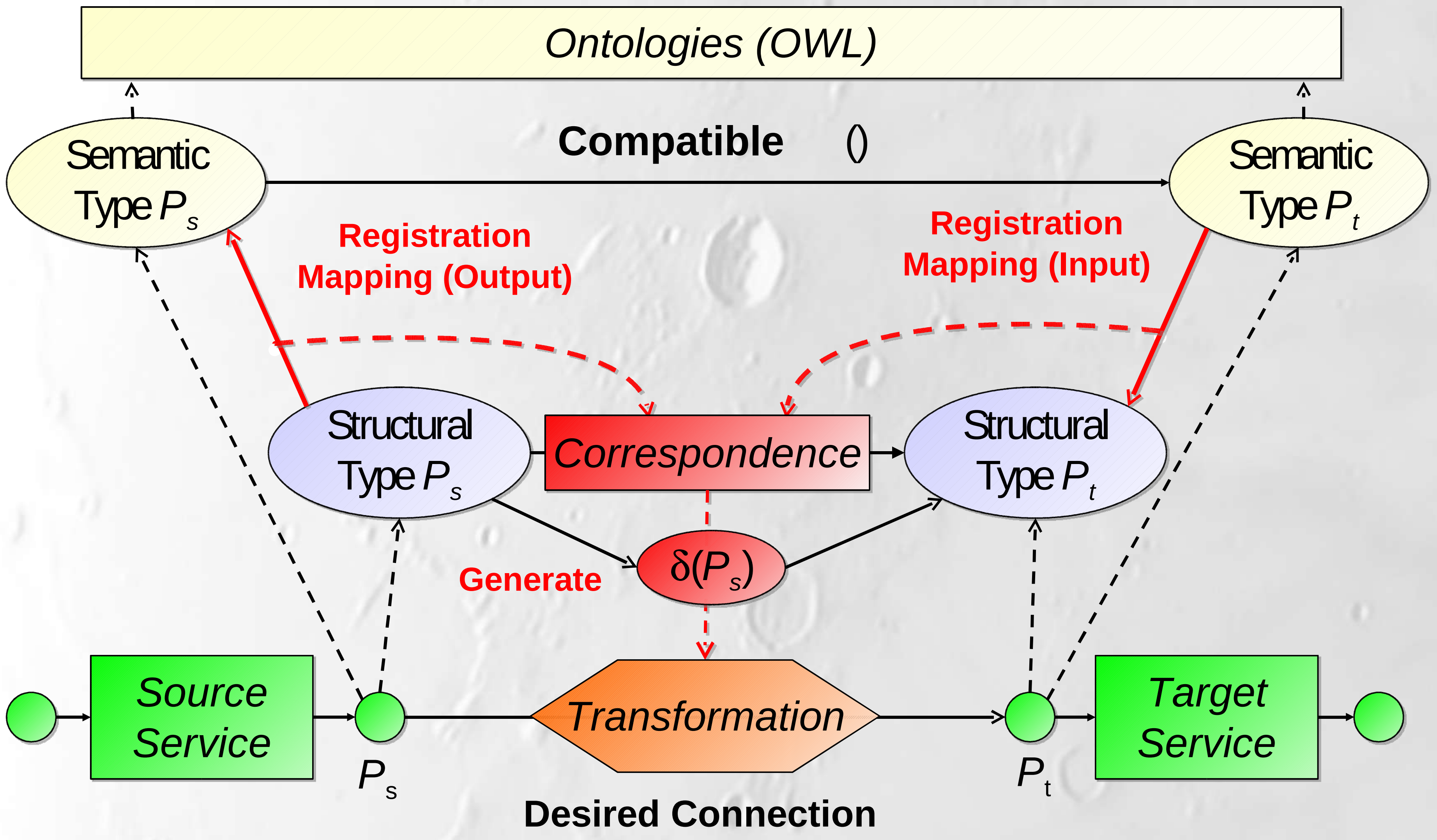
A KR+DI+Scientific Workflow Problem

- ***Services can be*** *semantically compatible*, ***but*** *structurally incompatible*



Source: [Bowers-Ludaescher, DILS'04]

Ontology-Informed Data Transformation



Source: [Bowers-Ludaescher, DILS'04]

Kepler

Some KEPLER Grid Plans …

An (oversimplified) **Model of the Grid**

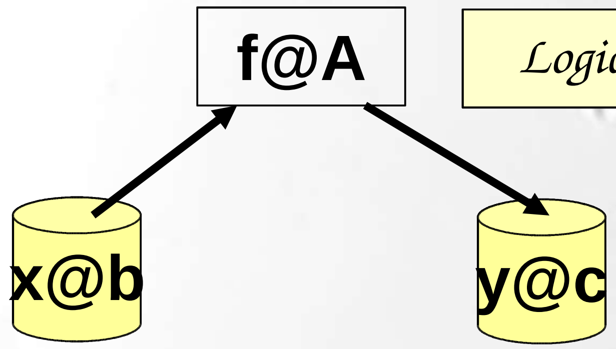
- *Hosts: {h1, h2, h3, …}*
- *Data@Hosts: d1@{h_i}, d2@{h_j}, …*
- *Functions@Hosts: f1@{h_i}, f2@{h_j}, …*

$$\begin{tikzcd} X \arrow[r, "f"] & Y \arrow[r, "g"] & Z \end{tikzcd}$$

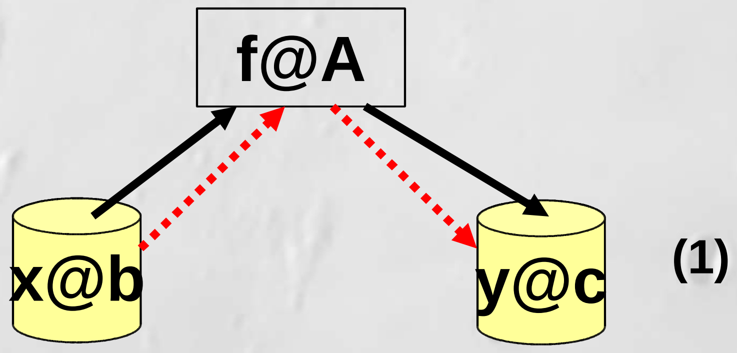
- *Given*: **data/workflow:**
- ***… as a functional plan: [...; Y := f(X); Z := g(Y); …]***
- ***… as a logic plan: [...; f(X,Y)$\wedge$g(Y,Z); …]***

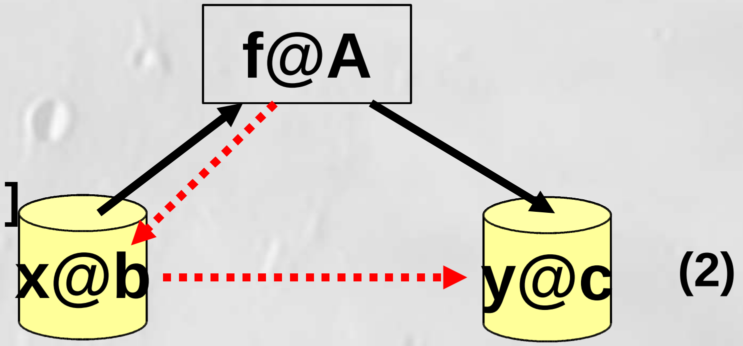
- *Find Host Assignment:* $\mathbf{d_i \rightarrow h_i\ ,\ f_j \rightarrow h_j}$
  - **for all** $\mathbf{d_i, f_j}$ *… s.t. […; d3@h3 := f@h2(d1@h1), …] is a* **valid** *plan*

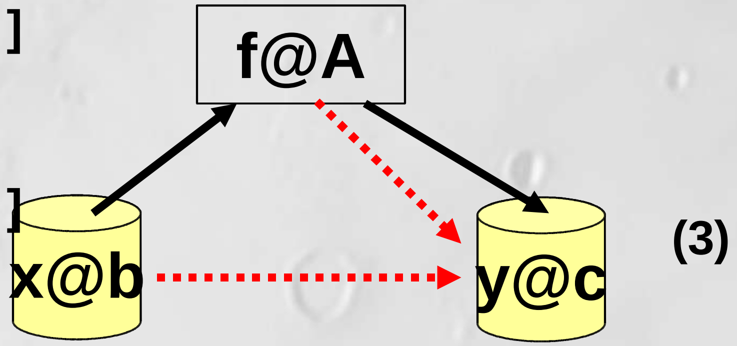
Shipping and Handling Algebra (SHA)



Logical view







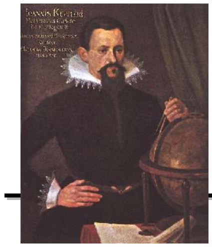
plan Y@C = F@A of X@B =

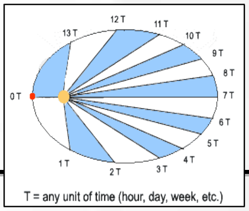
2. [X@B to A, Y@A := F@A(X@A), Y@A to C]

4. [F@A => B, Y@B := F@B(X@B), Y@B to C]

6. [X@B to C, F@A => C, Y@C := F@C(X@C)]

Physical view: SHA Plans

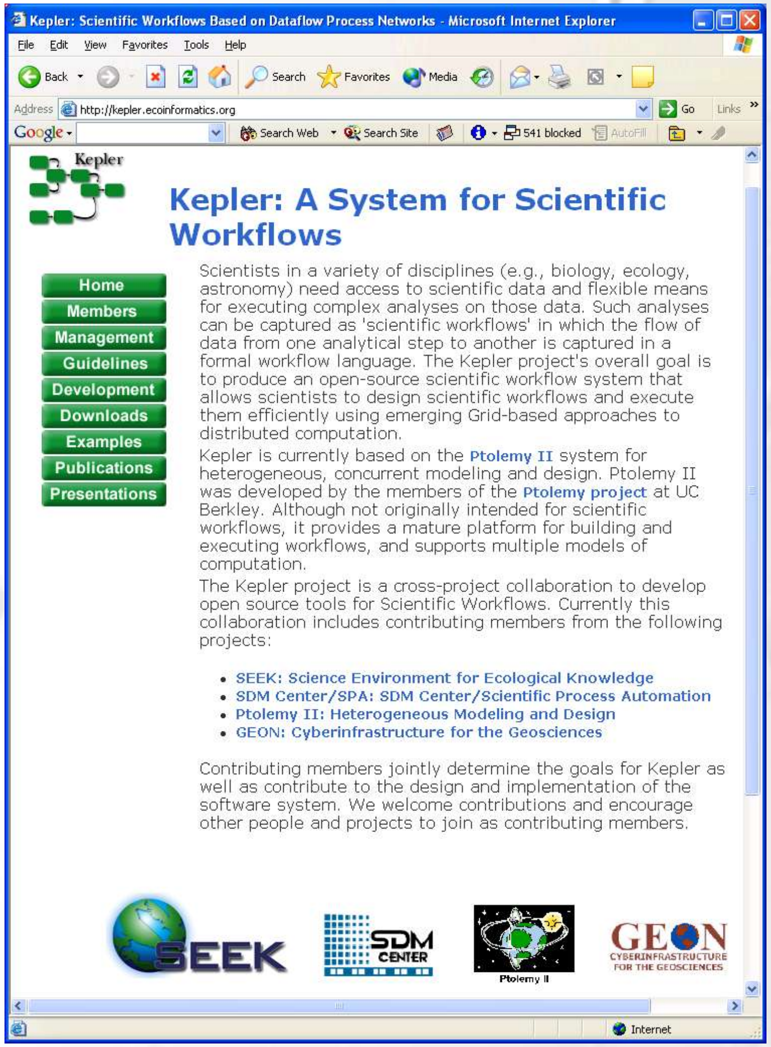




KEPLER and YOU

http://kepler.ecoinformatics.org

- *KEPLER …*
  - *is a community-based, cross-project, open source collaboration*
  - *can use web services as basic building blocks*
  - *has a joint CVS repository, mailing lists, web site, …*
  - *is gaining momentum thanks to contributors and contributions*
    - *BSD-style license allows commercial spin-offs*
- *An Invitation:*
  - *Provide some time (student?) and a scientific workflow to be built, and then let's just do it…*
  - *(we provide KEPLER expertise)*



Kepler: Scientific Workflows Based on Dataflow Process Networks - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://kepler.ecoinformatics.org Go Links
Google Search Web Search Site 541 blocked AutoFill
Kepler
Home
Members
Management
Guidelines
Development
Downloads
Examples
Publications
Presentations
SEEK
SDM CENTER
Ptolemy II
GEON CYBERINFRASTRUCTURE FOR THE GEOSCIENCES
Internet

Kepler: A System for Scientific Workflows

Scientists in a variety of disciplines (e.g., biology, ecology, astronomy) need access to scientific data and flexible means for executing complex analyses on those data. Such analyses can be captured as 'scientific workflows' in which the flow of data from one analytical step to another is captured in a formal workflow language. The Kepler project's overall goal is to produce an open-source scientific workflow system that allows scientists to design scientific workflows and execute them efficiently using emerging Grid-based approaches to distributed computation.

Kepler is currently based on the Ptolemy II system for heterogeneous, concurrent modeling and design. Ptolemy II was developed by the members of the Ptolemy project at UC Berkley. Although not originally intended for scientific workflows, it provides a mature platform for building and executing workflows, and supports multiple models of computation.

The Kepler project is a cross-project collaboration to develop open source tools for Scientific Workflows. Currently this collaboration includes contributing members from the following projects:

- SEEK: Science Environment for Ecological Knowledge
- SDM Center/SPA: SDM Center/Scientific Process Automation
- Ptolemy II: Heterogeneous Modeling and Design
- GEON: Cyberinfrastructure for the Geosciences

Contributing members jointly determine the goals for Kepler as well as contribute to the design and implementation of the software system. We welcome contributions and encourage other people and projects to join as contributing members.