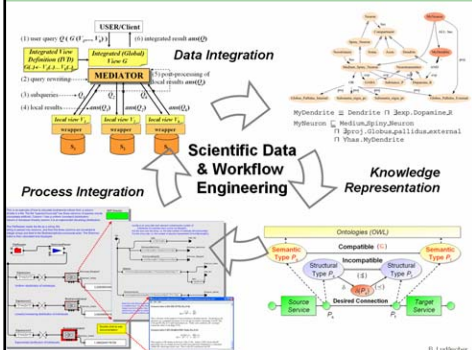
Scientific Data Integration: From the Big Picture to some Gory Details



Bertram Ludäscher
ludaesch@ucdavis.edu

Associate Professor
Dept. of Computer Science & Genome Center
University of California, Davis

UC DAVIS
Department of Computer Science

UC DAVIS genome center

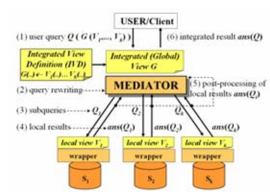
Fellow
San Diego Supercomputer Center
University of California, San Diego

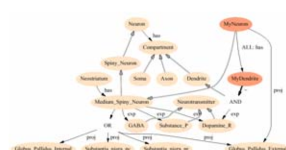
San Diego Supercomputer Center SDSC

Outline

Kepler

- **Data Integration & Mediation (DI)**
- **Challenges with Scientific Data**
- **Knowledge-based Extensions & Ontologies (DI+KR)**
- **Scientific Workflows (SWF+DI+KR)**
- **Scientific Workflow Design**





Kepler

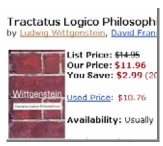
An Online Shopper's Information Integration Problem

El Cheapo: *"Where can I get the cheapest copy (including shipping cost) of Wittgenstein's Tractatus Logicus-Philosophicus within a week?"*

addall.com

Mediator (virtual DB)
(vs. Datawarehouse)
NOTE: non-trivial data engineering challenges!

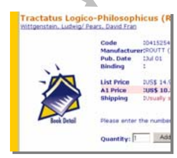
"One-World" Mediation



amazon.com

barnes&noble.com

half.com



A1books.com

Bertram Ludäscher, UC **DAVIS**

A Home Buyer's Information Integration Problem

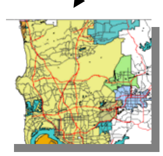
What houses for sale under $500k have at least 2 bathrooms, 2 bedrooms, a nearby school ranking in the upper third, in a neighborhood with below-average crime rate and diverse population?

?

Information Integration

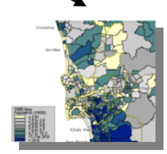
"Multiple-Worlds" Mediation

Realtor



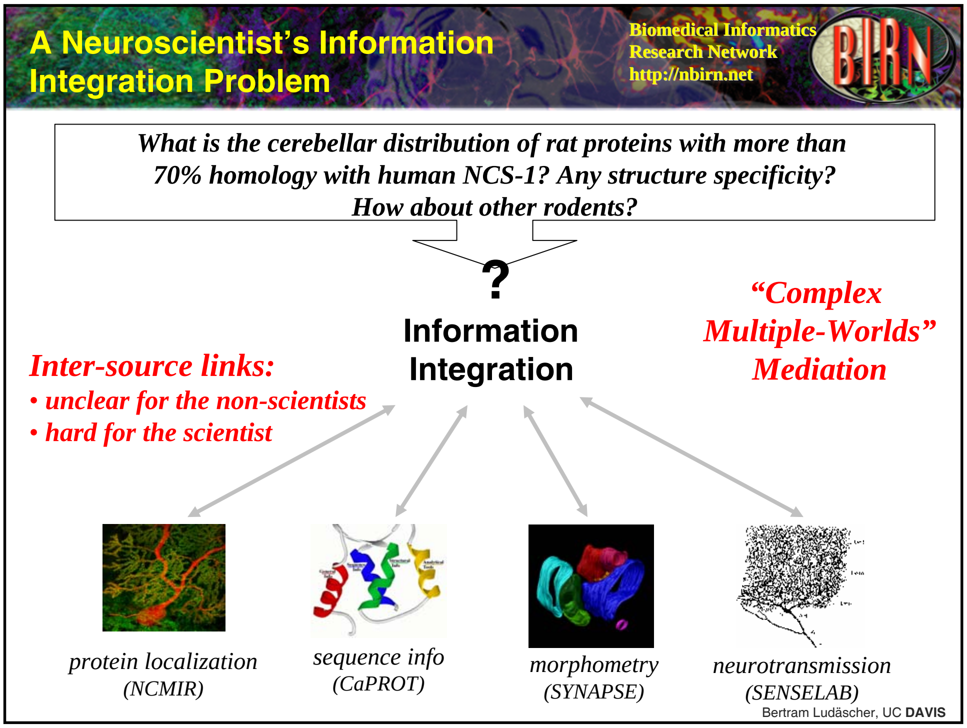
Crime Stats

School Rankings

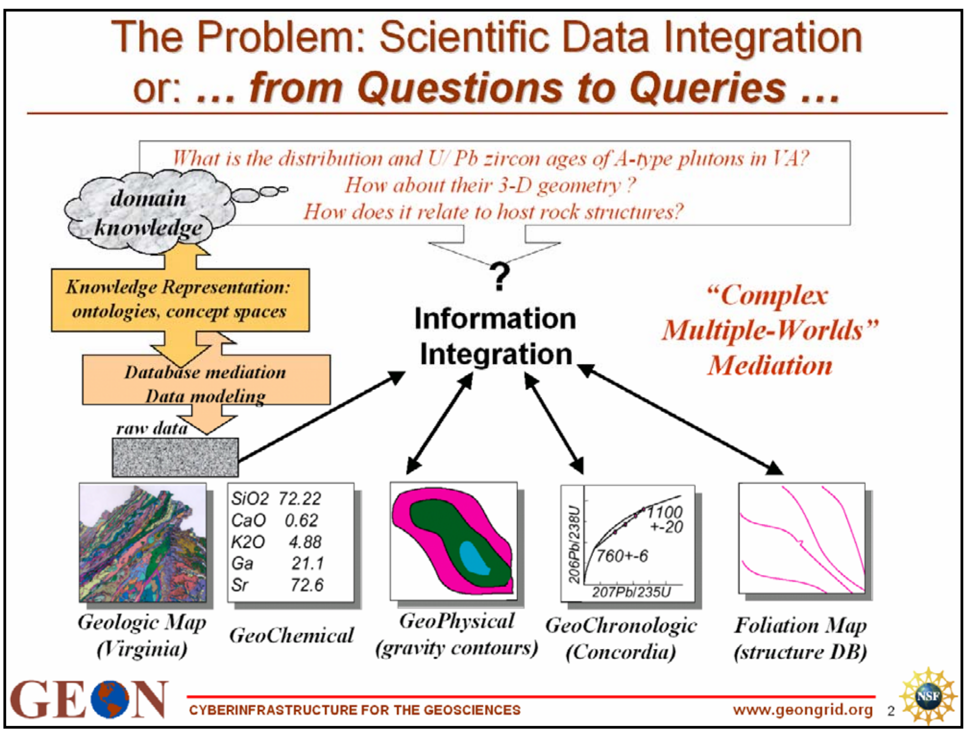


Demographics

Bertram Ludäscher, UC **DAVIS**

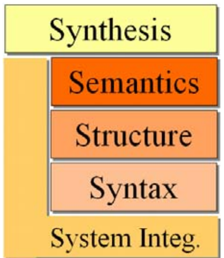


A Neuroscientist's Information Integration Problem
Biomedical Informatics Research Network http://nbirn.net
BIRN
What is the cerebellar distribution of rat proteins with more than 70% homology with human NCS-1? Any structure specificity? How about other rodents?
?
Information Integration
"Complex Multiple-Worlds" Mediation
Inter-source links:
• unclear for the non-scientists
• hard for the scientist
protein localization (NCMIR)
sequence info (CaPROT)
morphometry (SYNAPSE)
neurotransmission (SENSELAB)
Bertram Ludäscher, UC DAVIS



The Problem: Scientific Data Integration
or: … from Questions to Queries …
What is the distribution and U/Pb zircon ages of A-type plutons in VA?
How about their 3-D geometry ?
How does it relate to host rock structures?
domain knowledge
Knowledge Representation: ontologies, concept spaces
Database mediation Data modeling
raw data
?
Information Integration
"Complex Multiple-Worlds" Mediation
SiO2 72.22
CaO 0.62
K2O 4.88
Ga 21.1
Sr 72.6
1100 +-20
760+-6
206Pb/238U
207Pb/235U
Geologic Map (Virginia)
GeoChemical
GeoPhysical (gravity contours)
GeoChronologic (Concordia)
Foliation Map (structure DB)
GEON
CYBERINFRASTRUCTURE FOR THE GEOSCIENCES
www.geongrid.org 2

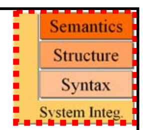
Interoperability & Integration Challenges



- **reconciling S^5 heterogeneities**
- **"gluing" together resources**
- **bridging information and knowledge gaps computationally**

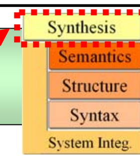
- ***System aspects: "Grid" Middleware***
  - distributed data & computing, SOA
  - resource discovery, authentication, authorization
  - web services, WSDL/SOAP, WSRF, OGSA, …
  - *(re-)sources = services, files, data sets, nodes …*
- ***Syntax & Structure: (XML-Based) Data Mediators***
  - wrapping, restructuring
  - (XML) queries and views
  - *sources = (XML) databases*
- ***Semantics: Model-Based/Semantic Mediators***
  - conceptual models and declarative views
  - Knowledge Representation: ontologies, description logics (RDF(S),OWL ...)
  - *sources = knowledge bases (DB+CMs+ICs)*
- ***Synthesis: Scientific Workflow Design & Execution***
  - Composition of declarative and procedural components into larger workflows
  - *(re)sources = services, processes, actors, …*

Information Integration Challenges: ***S^4 Heterogeneities***



- **System aspects**
  - platforms, devices, data & service distribution, APIs, protocols, …
  - ➔ **Grid middleware technologies**
  - \+ e.g. single sign-on, platform independence, transparent use of remote resources, …
- **Syntax & Structure**
  - heterogeneous data formats (*one for each tool* ...)
  - heterogeneous data models (*RDBs, ORDBs, OODBs, XMLDBs, flat files, …*)
  - heterogeneous schemas (*one for each DB* ...)
  - ➔ **Database mediation technologies**
  - \+ XML-based data exchange, integrated views, transparent query rewriting, …
- **Semantics**
  - descriptive metadata, different terminologies, "hidden" semantics (context), implicit assumptions, …
  - ➔ **Knowledge representation & semantic mediation technologies**
  - \+ "smart" data discovery & integration
  - \+ e.g. ask about ***X*** ('**mafic**'); find data about ***Y*** ('**diorite**'); be happy anyways!

Information Integration Challenges: *S^5 Heterogeneities*



- ***S**ynthesis* of applications, analysis tools, data & query components, … into "**scientific workflows**"
  - How to put together components to solve a scientist's problem?

➔ **Scientific Problem Solving Environments (PSEs)**

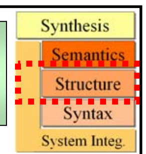
➔ **Portals, Workbench ("scientist's view")**

+ ontology-enhanced data registration, discovery, manipulation
+ creation and registration of new data products from existing ones, …

➔ **Scientific Workflow System ("engineer's view")**

+ for designing, re-engineering, deploying analysis pipelines and scientific workflows; ***a tool to make new tools …***
+ e.g., creation of new datasets from existing ones, dataset registration, …

Information Integration from a Database Perspective



- **Information Integration Problem**
  - **Given:** data sources S_1, ..., S_k (databases, web sites, ...) and user questions Q_1,..., Q_n that can –in principle– be answered using the information in the S_i
  - **Find:** the answers to Q_1, ..., Q_n

- **The Database Perspective:** source = "database"

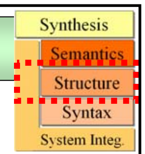
  ⇒ S_i has a **schema** (relational, XML, OO, ...)

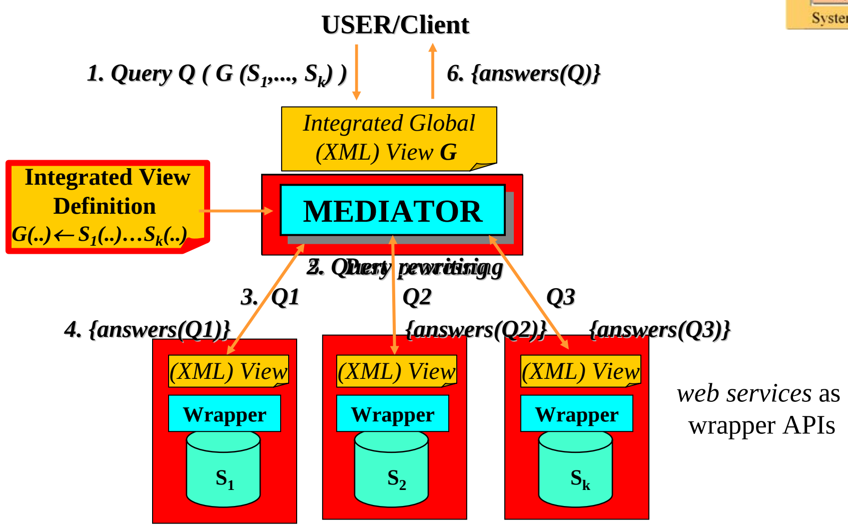
  ⇒ S_i **can be queried**

  ⇒ define virtual (or materialized) **integrated (or global) view G** over local sources S_1 ,..., S_k using **database query languages** (SQL, XQuery,...)

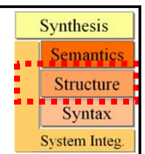
  ⇒ ***questions become queries*** Q_i against $G(S_1,..., S_k)$

Standard (XML-Based) Mediator Architecture





Query Planning in Data Integration



- **Given:**
  - **Declarative user query Q**: ***answer(…) :– …G …***
  - … & { G **:–** … L … } *global-as-view* (GAV)
  - … & { L **:–** … G … } *local-as-view* (LAV)
  - … & { ic(…) **:–** … L … G… } *integrity constraints* (ICs)
- **Find:**
  - equivalent (or *minimal containing*, *maximal contained*) **query plan Q'**: ***answer(…) :– … L …***

  ➔ query rewriting
- **Results:**
  - A variety of results/algorithms; depending on classes of queries, views, and ICs: **P**, **NP**, … , **undecidable**
  - still a hot research area in the database community

Data Integration: Limited Access Patterns+Views+ICs

We study the problem of rewriting a query Q in terms of a given set of views $\mathcal{V}$ with limited access patterns $\mathcal{P}$, under a set Σ of integrity constraints. More precisely, we are interested in determining whether there exists a query plan Q', expressed in terms of the views $\mathcal{V}$ only, that is executable (i.e., observes $\mathcal{P}$) and equivalent to Q for all databases satisfying Σ. We say that Q is *feasible* if such Q' exists. For infeasible Q we seek the minimal containing and maximal contained executable queries, which provide the "best possible" executable query plans for approximating the answer to Q from above and below.

ICDT'05

Rewriting Queries Using Views with Access Patterns Under Integrity Constraints*

Alin Deutsch[1], Bertram Ludäscher[2], and Alan Nash[3]

University of California, San Diego
deutsch@cs.ucsd.edu, ludaesch@sdsc.edu, anash@math.ucsd.edu

Abstract. We study the problem of rewriting queries using views in the presence of access patterns, integrity constraints, disjunction, and negation. We provide asymptotically optimal algorithms for finding minimal containing and maximal contained rewritings and for deciding whether an exact rewriting exists. We show that rewriting queries using views in this case reduces (a) to rewriting queries with access patterns and constraints without views and also (b) to rewriting queries using views under constraints without access patterns. We show how to solve (a) directly and how to reduce (b) to rewriting queries under constraints only (semantic optimization). These reductions provide two separate routes to a unified solution for all three problems, based on an extension of the relational chase theory to queries and constraints with disju… negation. We also handle equality and arithmetic compariso…

the set of listed publishers $L^{i}(p)$, repository $R^{oo}(a,t)$, ACM anthology $A^{iii}(a,t,o)$, and DBLP conference article $D^{ooo}(a,t,c)$. The relation symbols are annotated with access patterns, indicating which arguments must be given as inputs (marked 'i') and which ones can be retrieved as outputs (marked 'o') when accessing the relation. For example $C^{io}(a,t)$ means that an author a has to be given as input before one can retrieve the titles t of a's conference publications from $C(a,t)$.

Let Q be the query which asks for pairs of authors and titles of conference publications, journal publications, and magazines which are not PC-magazines:

Query Q

$$Q(a,t) \leftarrow \underline{C(a,t)} \quad (1)$$
$$Q(a,t) \leftarrow \underline{J(a,t)} \quad (2)$$
$$Q(a,t) \leftarrow M(a,t), \underline{\neg P(a,t,p)}, \underline{L(p)} \quad (3)$$

Q cannot be executed since no underlined literal is answerable: e.g., the access patterns require a to be bound before invoking $C(a,t)$ but no such binding is available. Worse yet, Q is not even feasible, i.e., there is no executable query Q' equivalent to Q. However, if the following set Σ of integrity constraints is given, an executable Q' can be found that is equivalent under Σ:

ICs Σ

$$\forall a \forall t\ C(a,t) \rightarrow \exists c\ D(a,t,c) \quad (4)$$
$$\forall a \forall t\ J(a,t) \rightarrow \exists p\ R(a,t) \wedge \neg P(a,t,p) \wedge L(p) \vee\ \exists o \exists c\ A(a,t,o), D(a,t,c) \quad (5)$$
$$\forall a \forall t\ M(a,t) \rightarrow \exists p\ \neg P(a,t,p), L(p) \quad (6)$$

Constraint (4) states that every conference publication is a DBLP conference publication; (5) states that every journal publication is available from a repository, comes from a listed publisher and is not a PC magazine, or is available from the ACM anthology and from DBLP; and (6) states that magazine articles are not PC-magazine articles. We are only interested in databases which satisfy these constraints Σ. On those databases, Q is equivalent to Q^{Σ}, obtained by "chasing" Q with Σ:

Chase result Q^{Σ}

$$Q^{\Sigma}(a,t) \leftarrow C(a,t), D(a,t,c)$$
$$Q^{\Sigma}(a,t) \leftarrow J(a,t), R(a,t), \underline{\neg P(a,t,p)}, \underline{L(p)}$$
$$Q^{\Sigma}(a,t) \leftarrow J(a,t), \underline{A(a,t,o)}, D(a,t,c)$$
$$Q^{\Sigma}(a,t) \leftarrow M(a,t), \underline{\neg P(a,t,p)}, \underline{L(p)}$$

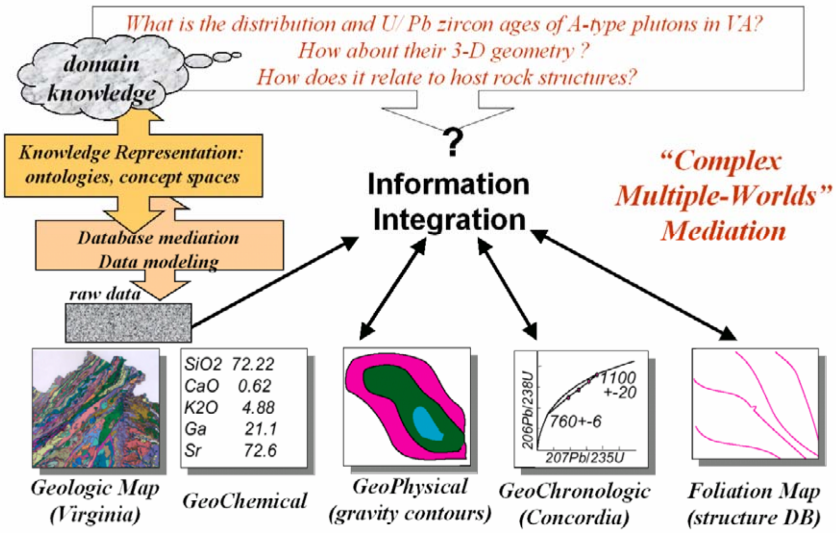
Again, unanswerable literals are underlined. The *answerable part* $\text{ans}(Q^{\Sigma})$ is obtained (roughly) by removing unanswerable parts:

Answerable part $\text{ans}(Q^{\Sigma})$

$$\text{ans}(Q^{\Sigma})(a,t) \leftarrow C(a,t), D(a,t,c) \quad (7)$$
$$\text{ans}(Q^{\Sigma})(a,t) \leftarrow J(a,t), R(a,t) \quad (8)$$
$$\text{ans}(Q^{\Sigma})(a,t) \leftarrow J(a,t), D(a,t,c) \quad (9)$$
$$\text{ans}(Q^{\Sigma})(a,t) \leftarrow M(a,t) \quad (10)$$

Scientific Data Integration using Semantic Extensions

The Problem: Scientific Data Integration or: *… from Questions to Queries …*



GEON CYBERINFRASTRUCTURE FOR THE GEOSCIENCES www.geongrid.org 2

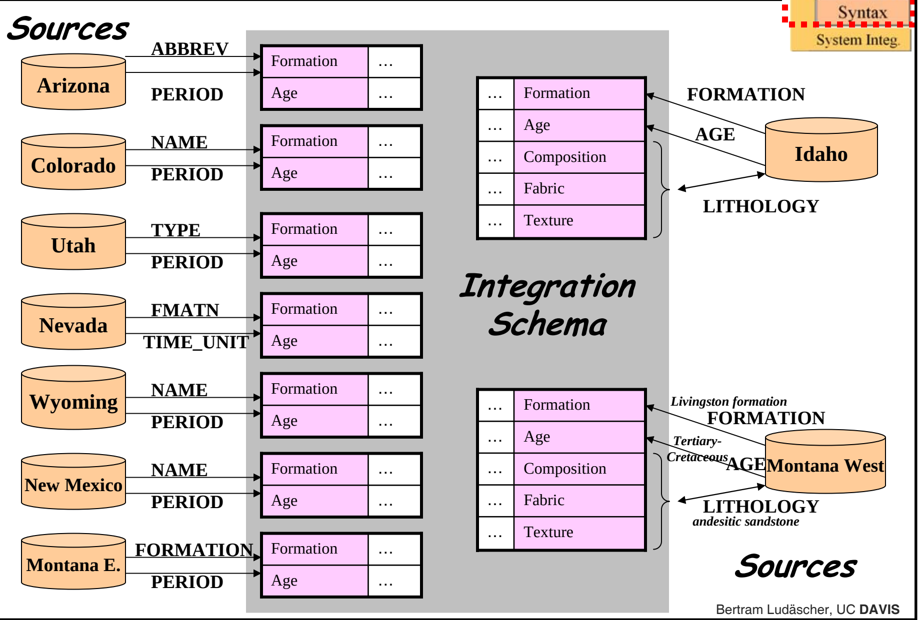
GEON The Geosciences Network

Example: Geologic Map Integration

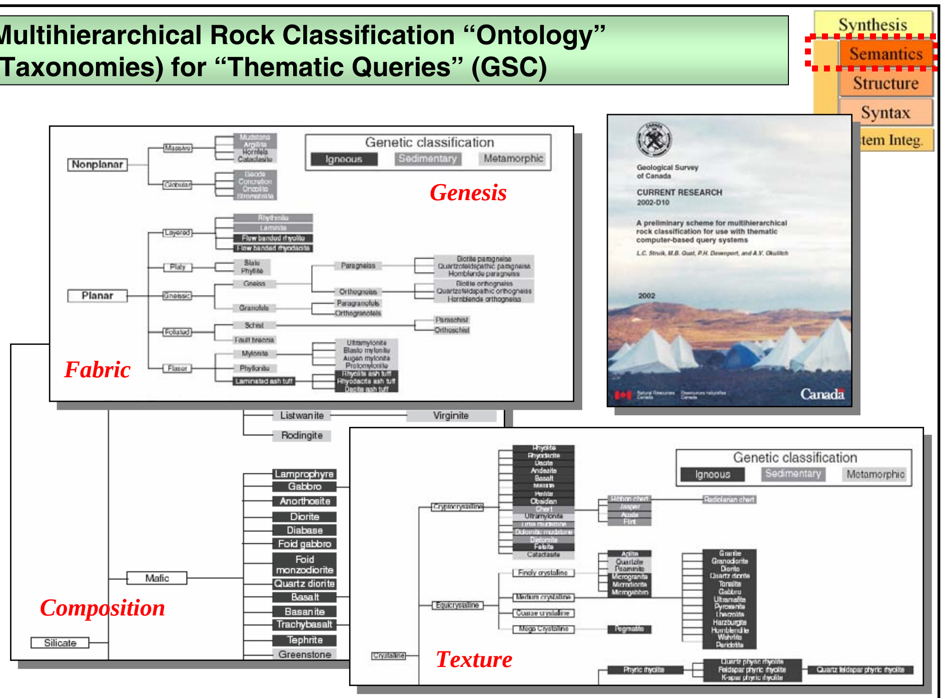
- **Given**:
  - Geologic maps from different state geological surveys (shapefiles w/ different data schemas)
  - Different ontologies:
    - Geologic age ontology (e.g. USGS)
    - Rock classification ontologies:
      - Multiple hierarchies (chemical, fabric, texture, genesis) from Geological Survey of Canada (GSC)
      - Single hierarchy from British Geological Survey (BGS)
- **Problem**:
  - Support **uniform queries** across all maps
  - … using **different** ontologies
  - Support **registration** w/ ontology A, **querying** w/ ontology B

Bertram Ludäscher, UC **DAVIS**

Schema Integration
(“registering” local schemas to the global schema)

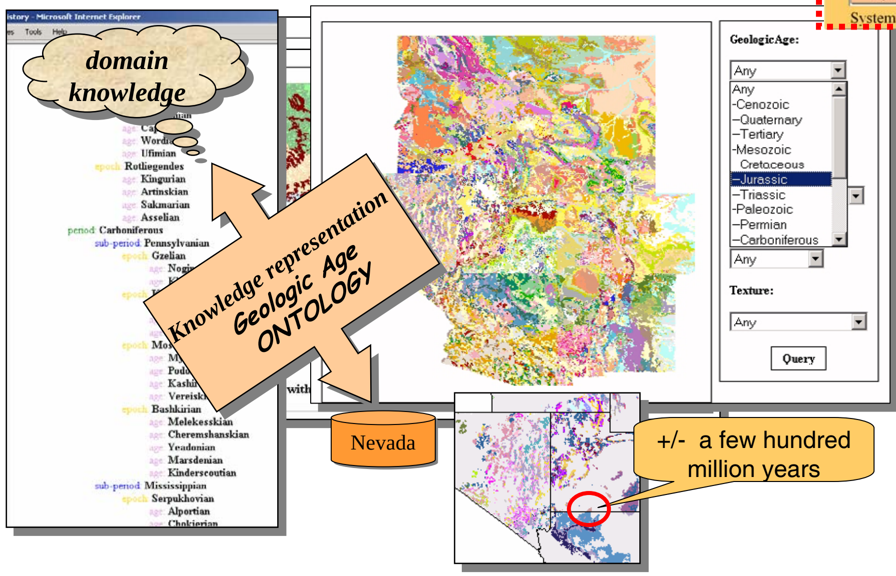


Multihierarchical Rock Classification “Ontology”
(Taxonomies) for “Thematic Queries” (GSC)



Ontology-Enabled Application Example: **Geologic Map Integration**

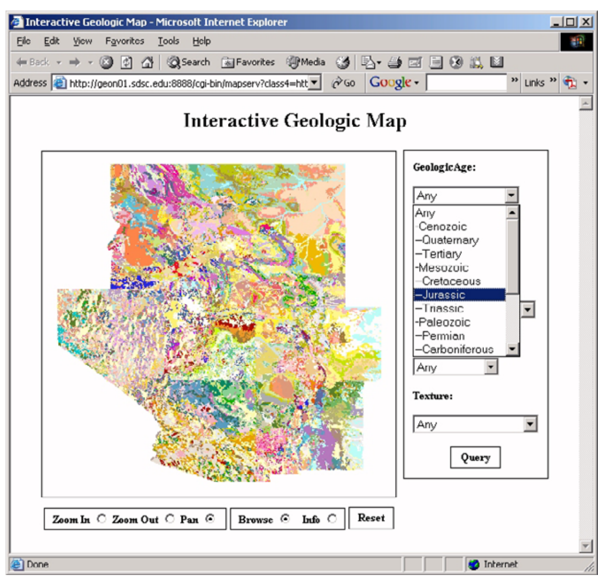
Synthesis
Semantics
Structure
Syntax
System Integ.



Bertram Ludäscher, UC **DAVIS**

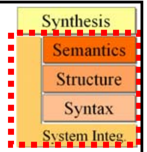
Querying by Geologic Age ...

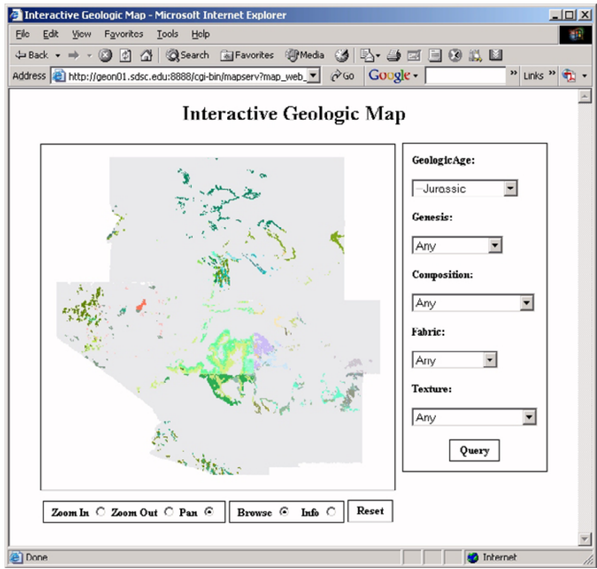
Synthesis
Semantics
Structure
Syntax
System Integ.



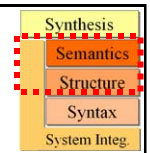
Bertram Ludäscher, UC **DAVIS**

Querying by Geologic Age: Results

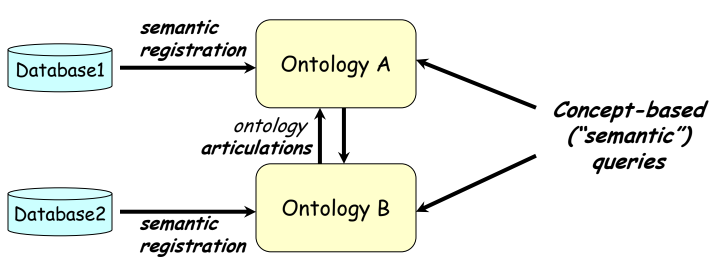


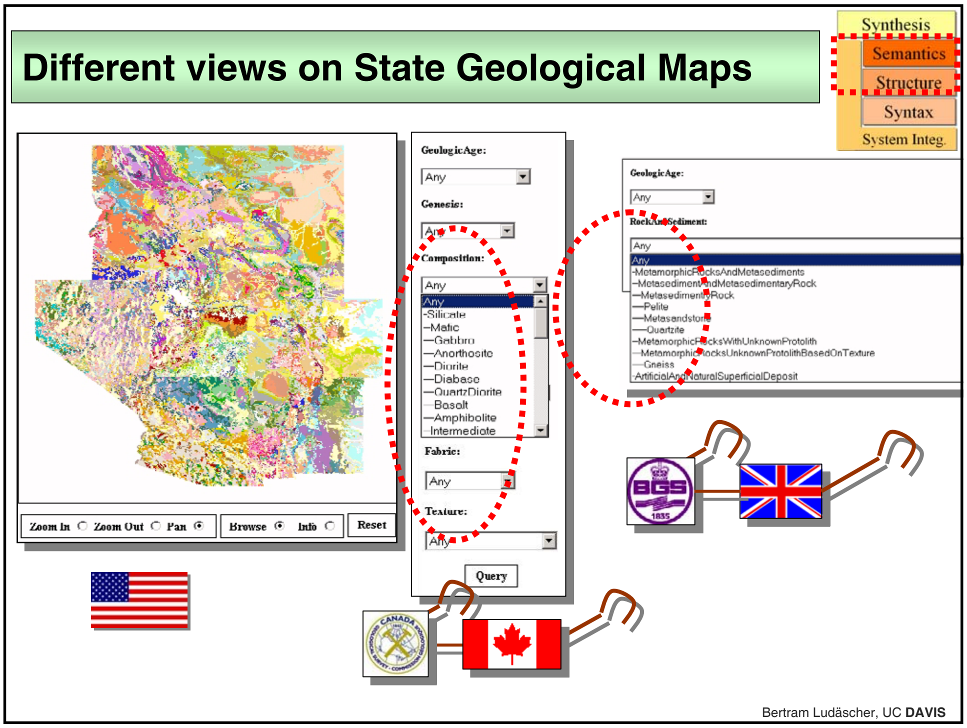


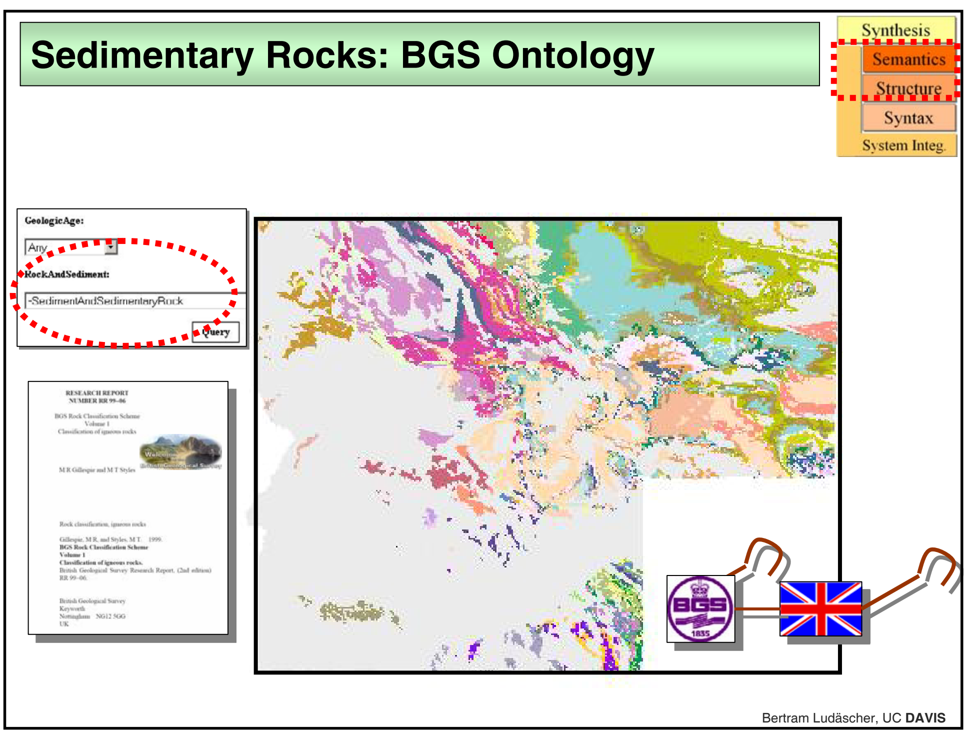
Semantic Mediation (via "semantic registration" of schemas and ontology articulations)



- **Schema elements** and/or data values are **associated** with concept expressions from the target **ontology**
  - ➔ conceptual queries "through" the ontology
- **Articulation ontology**
  - ➔ source registration to A, querying through B
- **Semantic mediation: query rewriting w/ ontologies**

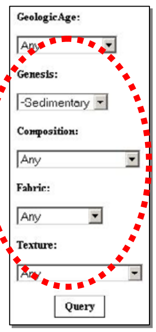


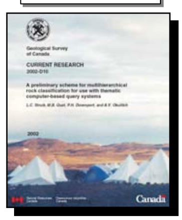

Different views on State Geological Maps
Synthesis
Semantics
Structure
Syntax
System Integ.
Zoom In
Zoom Out
Pan
Browse
Info
Reset
GeologicAge:
Any
Genesis:
Composition:
Any
-Silicate
—Mafic
—Gabbro
—Anorthosite
—Diorite
—Diabase
—QuartzDiorite
—Basalt
—Amphibolite
Intermediate
Fabric:
Texture:
Query
RockAndSediment:
-MetamorphicRocksAndMetasediments
-MetasedimentAndMetasedimentaryRock
—MetasedimentaryRock
—Pelite
—Metasandstone
—Quartzite
-MetamorphicRocksWithUnknownProtolith
—MetamorphicRocksUnknownProtolithBasedOnTexture
—Gneiss
-ArtificialAndNaturalSuperficialDeposit
BGS
1835
Bertram Ludäscher, UC DAVIS

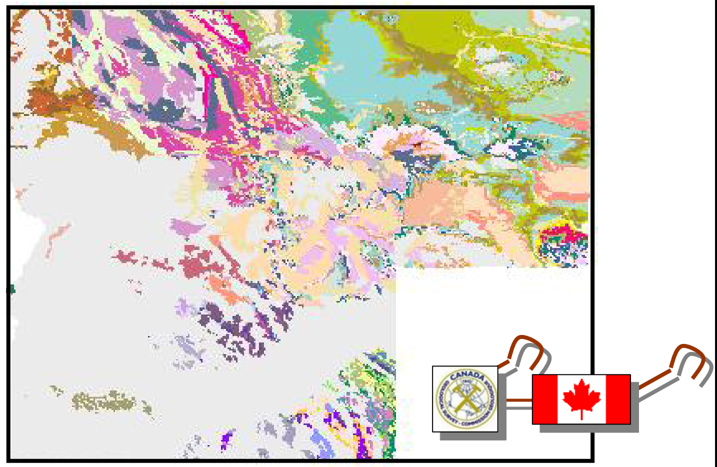

Sedimentary Rocks: BGS Ontology
Synthesis
Semantics
Structure
Syntax
System Integ.
GeologicAge:
Any
RockAndSediment:
-SedimentAndSedimentaryRock
Query
RESEARCH REPORT
NUMBER RR 99-06
BGS Rock Classification Scheme
Volume 1
Classification of igneous rocks
BGS
1835
Bertram Ludäscher, UC DAVIS

Sedimentary Rocks: GSC Ontology

Synthesis
Semantics
Structure
Syntax
System Integ.



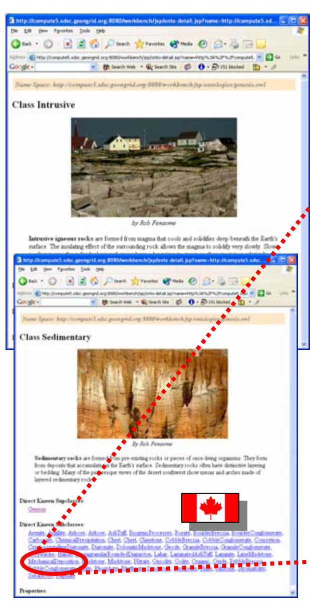


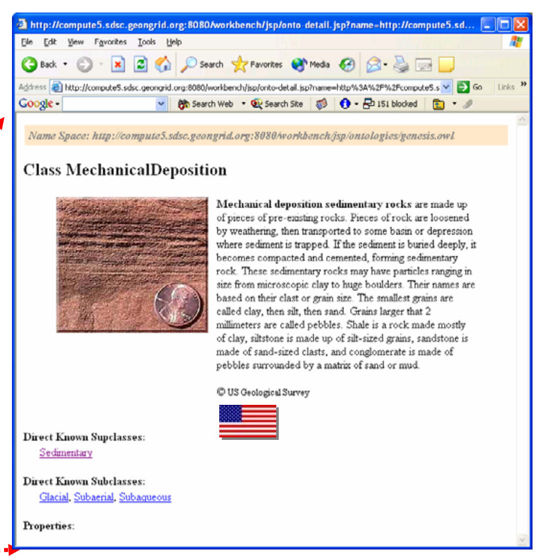


Bertram Ludäscher, UC **DAVIS**

Implementation in OWL: Not only "for the machine" …

Kepler



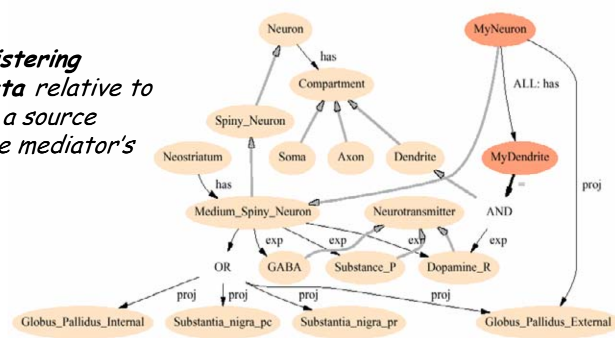


Bertram Ludäscher, UC **DAVIS**

Source **Data Contextualization** through **Ontology Refinement**

Biomedical Informatics Research Network

*In addition to **registering** ("hanging off") **data** relative to existing concepts, a source may also **refine** the mediator's domain map...*



$\texttt{MyDendrite} \equiv \texttt{Dendrite} \sqcap \exists \texttt{exp.Dopamine\_R}$

$\texttt{MyNeuron} \sqsubseteq \texttt{Medium\_Spiny\_Neuron}$
$\sqcap\ \exists \texttt{proj.Globus\_pallidus\_external}$
$\sqcap\ \forall \texttt{has.MyDendrite}$

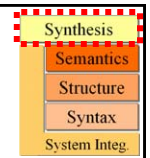
$\Rightarrow$ *sources can register **new concepts** at the mediator ...*

Bertram Ludäscher, UC **DAVIS**

Scientific Workflows

Bertram Ludäscher, UC **DAVIS**

Motivation:
Scientific Workflows, Pre-Cyberinfrastructure



- **Data Federation & Grid "Plumbing":**
  - access, move, replicate, query … data (***Data-Grid***)
    - authenticate … SRB Sget/Sput … OPeNDAP, … Antelope/ORBs
  - schedule, launch, monitor jobs (***Compute-Grid***)
    - Globus, Condor, Nimrod, APST, …
- **Data Integration:**
  - Conceptual querying & integration, structure & semantics, e.g. mediation w/ SQL, XQuery + OWL (*Semantics-enabled Mediator*)
- **Data Analysis, Mining, Knowledge Discovery:**
  - manual/textbook (e.g. ternary diagrams), Excel, R, simulations, …
- **Visualization:**
  - 3-D (volume), 4-D (spatio-temporal), n-D (conceptual views) …

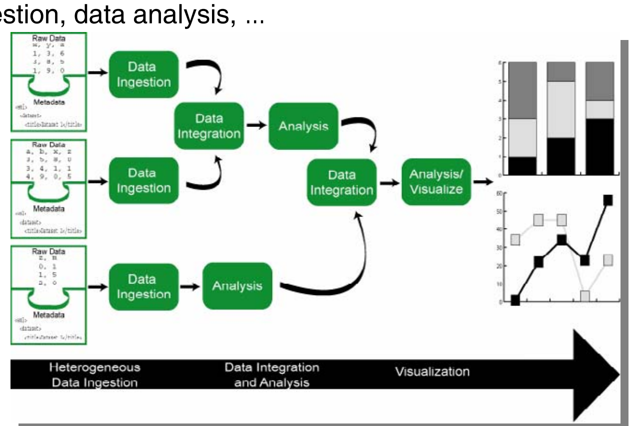
➔ **one-of-a-kind custom apps., detached (island) solutions**
➔ workflows are **hard to reproduce, maintain**
➔ **no/little** workflow design, automation, reuse, documentation

➔ need for an **integrated scientific workflow environment**

Bertram Ludäscher, UC **DAVIS**

What is a Scientific Workflow (SWF)?

- **Model the way scientists work with their data and tools**
  - Mentally coordinate data export, import, analysis via software systems
- **Scientific workflows emphasize** data flow **(≠ business workflows)**
- **Metadata** (incl. provenance info, semantic types etc.) is crucial for automated data ingestion, data analysis, ...
- **Goals:**
  - SWF **automation**,
  - SWF & component **reuse**,
  - SWF **design** & **documentation**
  - **making scientists' data analysis and management easier**!



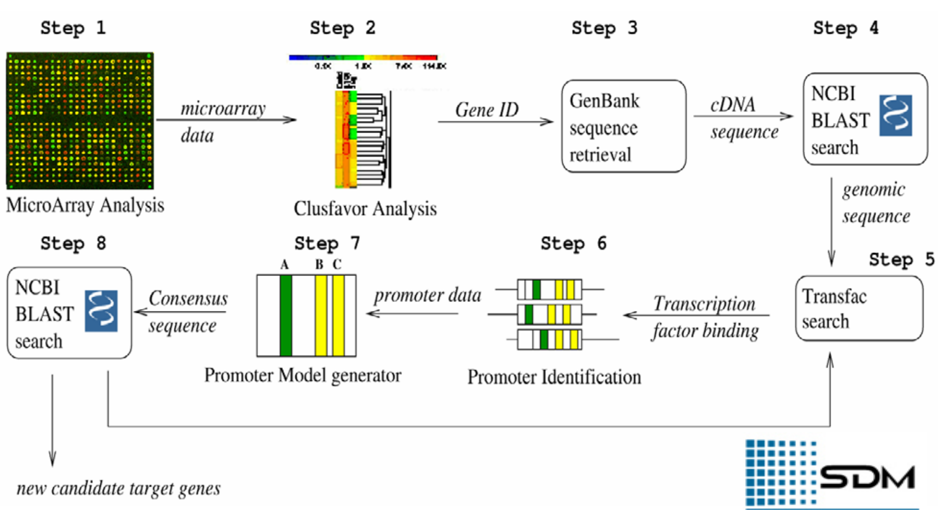
Bertram Ludäscher, UC **DAVIS**

Some Scientific Workflow Features

- Typical requirements/characteristics:
  - data-intensive and/or compute-intensive
  - plumbing-intensive
  - dataflow-oriented
  - distribution (data, processing)
  - user-interaction “in the middle”, …
  - … vs. (C-z; bg; fg)-ing (“detach” and reconnect)
  - advanced programming constructs (map(f), zip, takewhile, …)
  - logging, provenance, “registering back” (intermediate) products
  - …

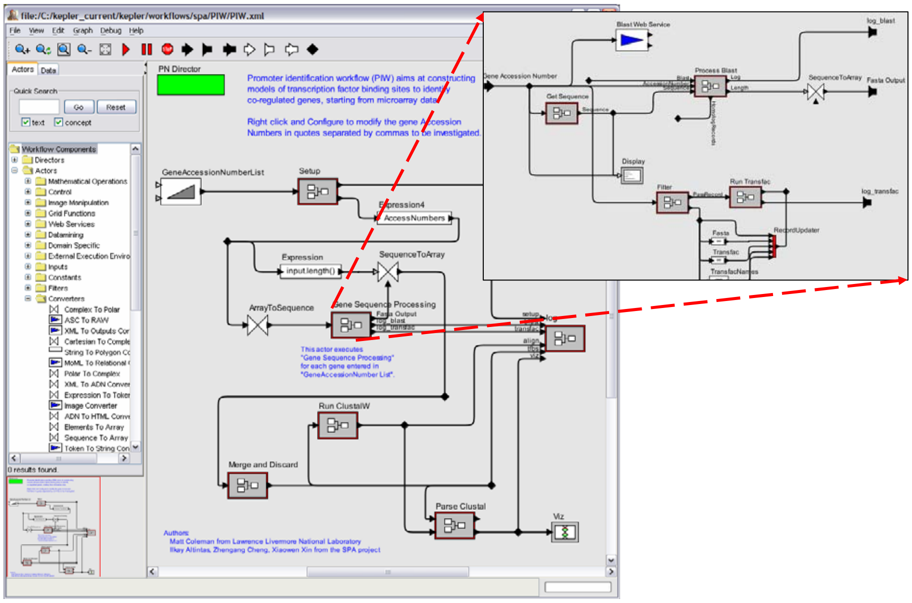
- … easy to recognize a SWF when you see one!

Promoter Identification Workflow (Napkin Drawing)



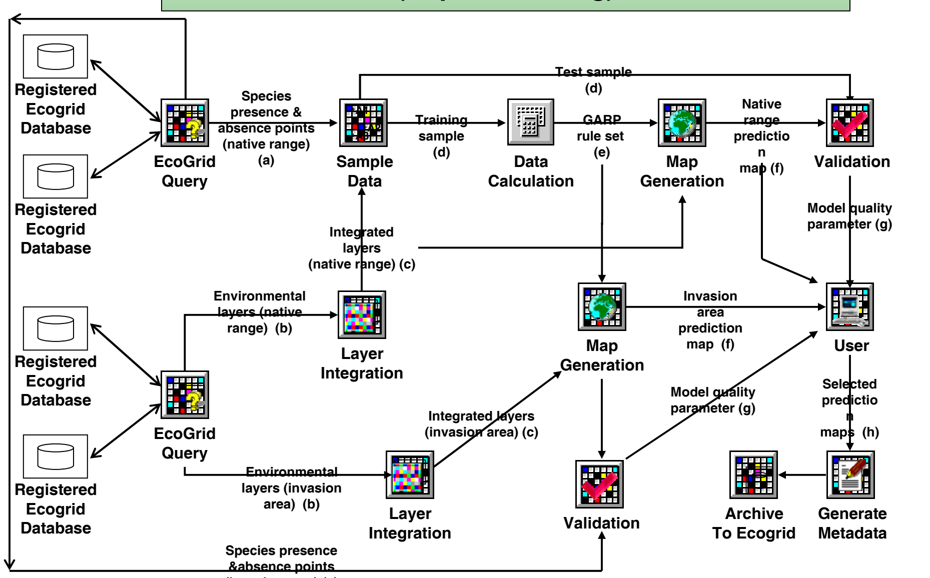
Source: Matt Coleman (LLNL)

Promoter Identification Workflow in Kepler



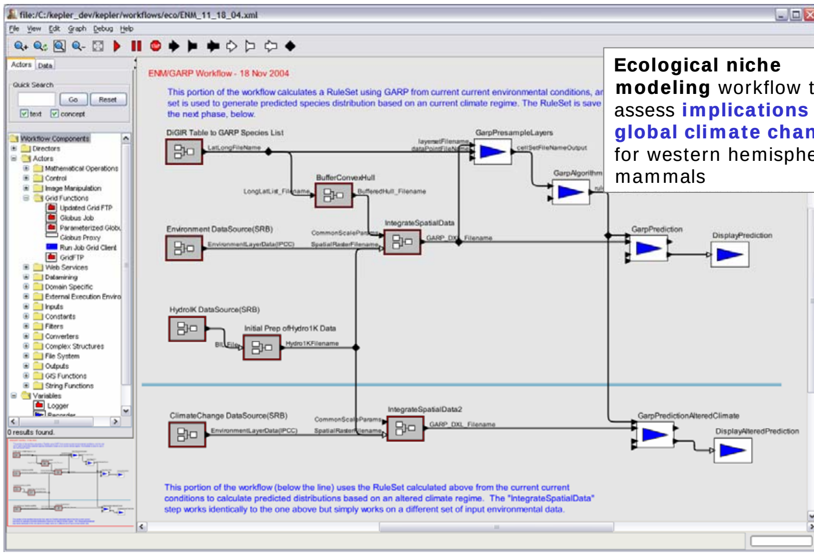
Bertram Ludäscher, UC **DAVIS**

Ecology: Invasive Species Prediction
(Napkin Drawing)



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

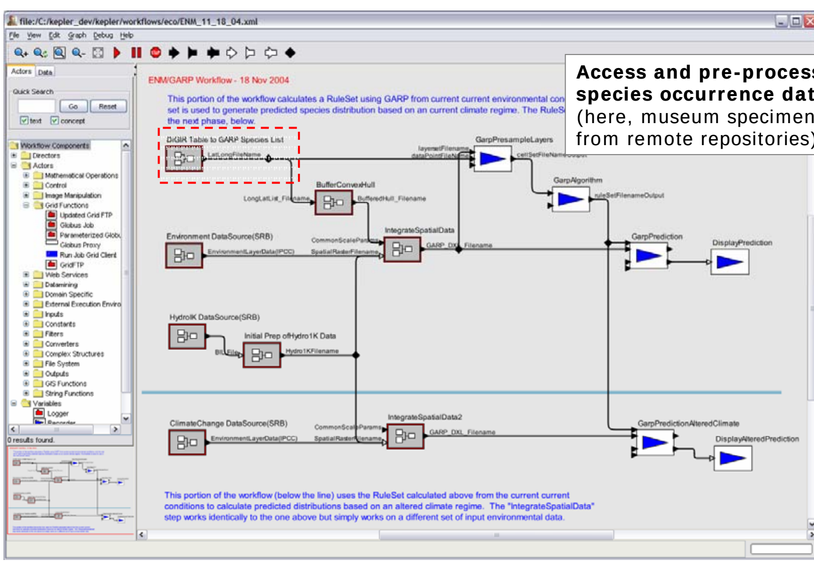
Ecological Niche Modeling in Kepler



Ecological niche modeling workflow to assess **implications of global climate change** for western hemisphere mammals

Bertram Ludäscher, UC **DAVIS**

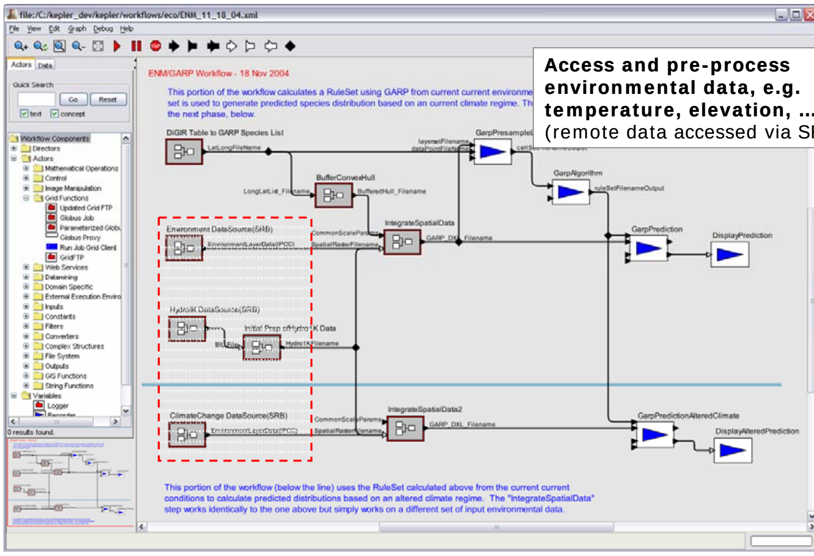
Ecological Niche Modeling in Kepler



Access and pre-process species occurrence data (here, museum specimen data from remote repositories)

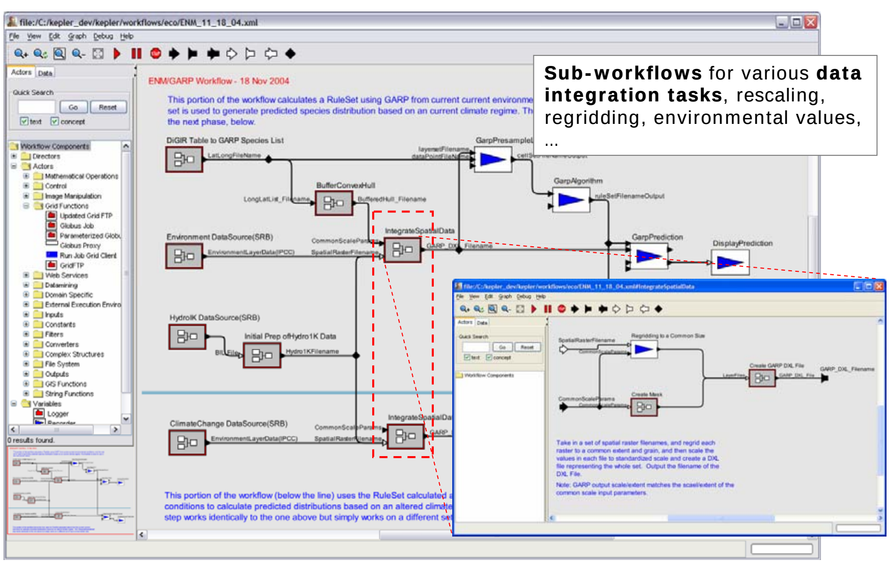
Bertram Ludäscher, UC **DAVIS**

Ecological Niche Modeling in Kepler



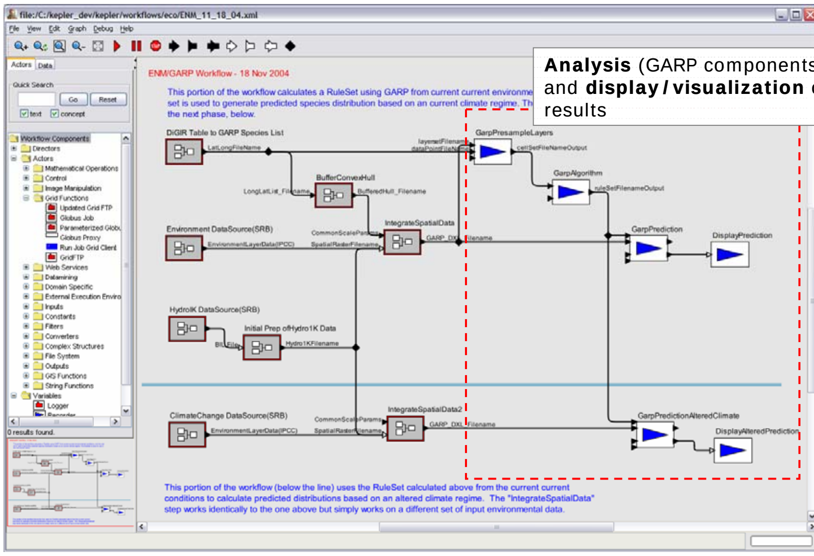
Access and pre-process environmental data, e.g. temperature, elevation, …
(remote data accessed via SRB)

Ecological Niche Modeling in Kepler



Sub-workflows for various **data integration tasks**, rescaling, regridding, environmental values, …

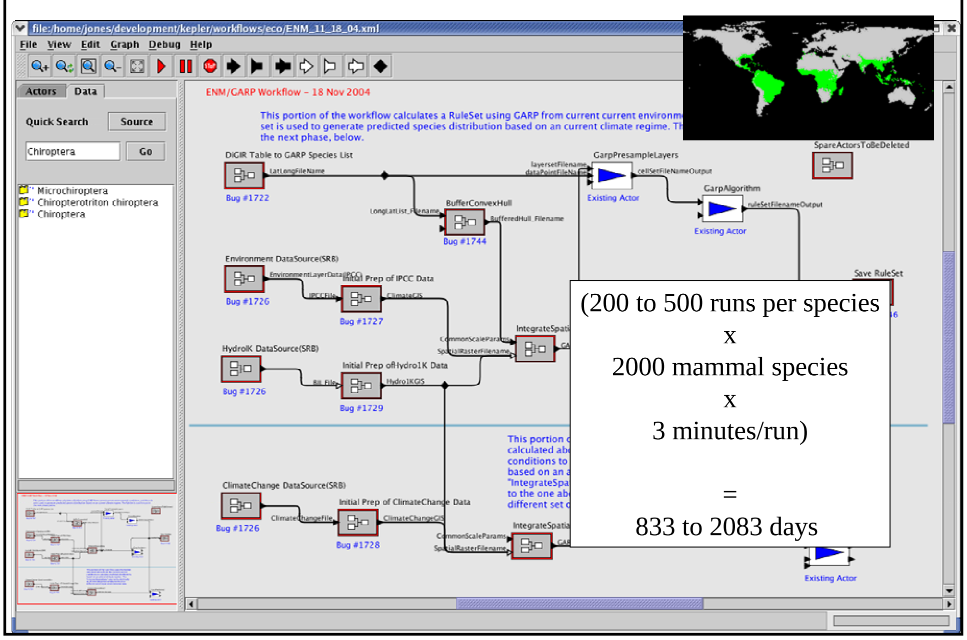
Ecological Niche Modeling in Kepler



Analysis (GARP components) and **display / visualization** of results

Bertram Ludäscher, UC **DAVIS**

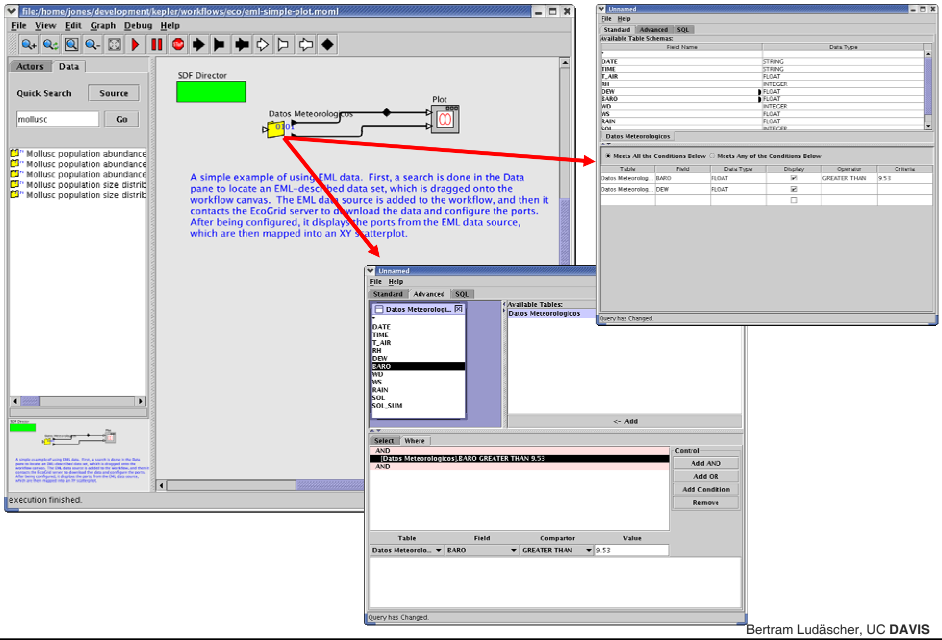
Ecological Niche Modeling in Kepler



(200 to 500 runs per species
x
2000 mammal species
x
3 minutes/run)

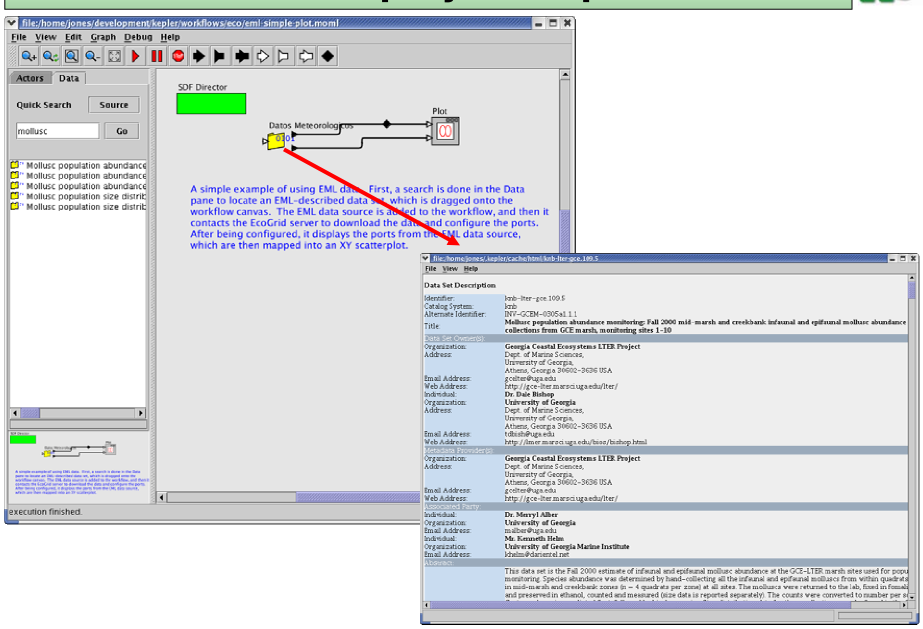
=
833 to 2083 days

Query Builder



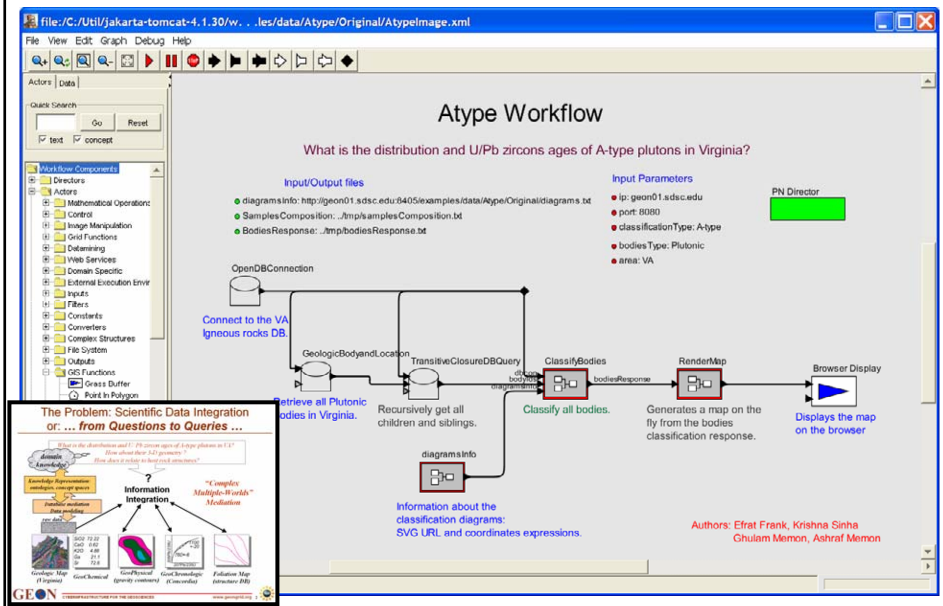
Bertram Ludäscher, UC **DAVIS**

EML Metadata Display in Kepler

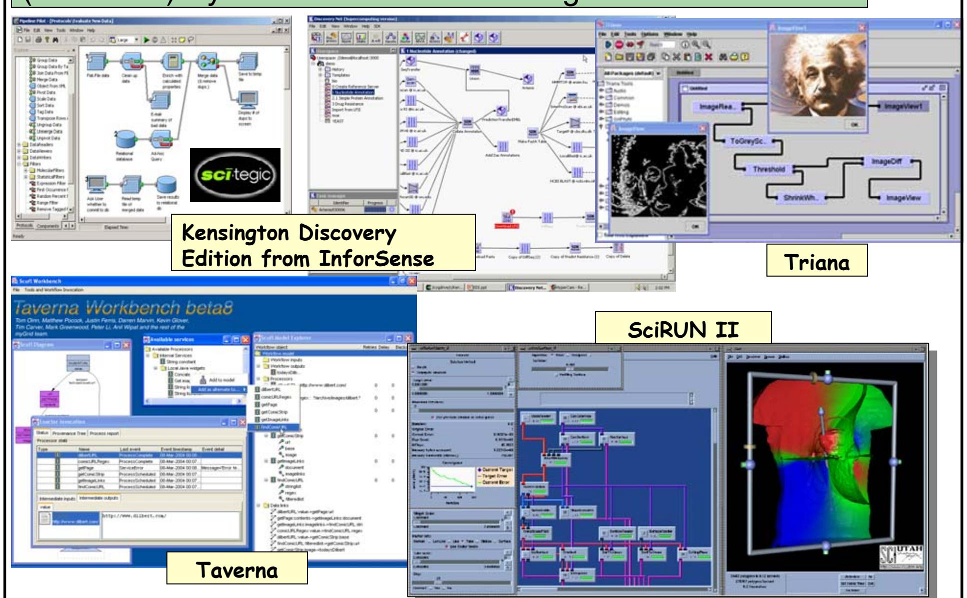


Bertram Ludäscher, UC **DAVIS**

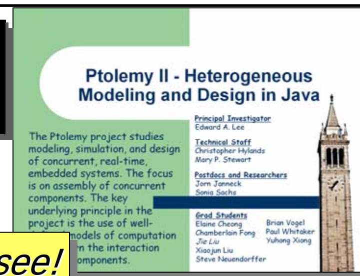
GEON Analysis Workflow in KEPLER



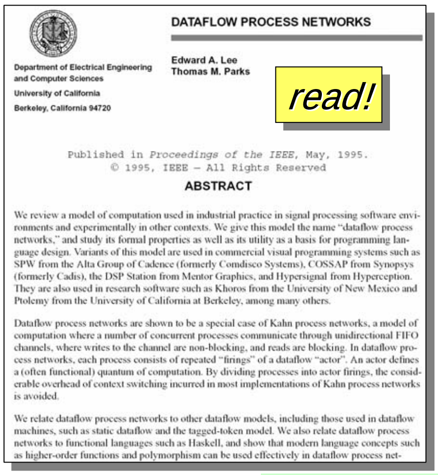
Commercial & Open Source Scientific Workflow and (**Dataflow**) Systems & Problem Solving Environments



Our Starting Point: Ptolemy II



see!



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

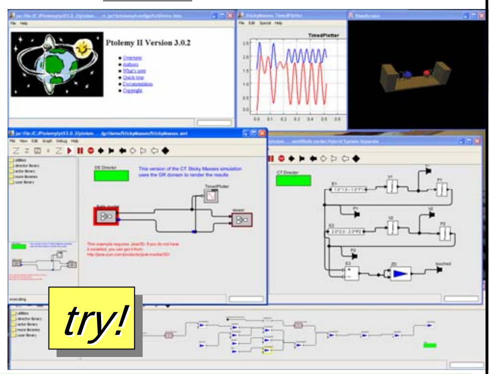
ABSTRACT

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!



try!

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

Why Ptolemy II ?

- **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**."
- **Dataflow Process Networks w/ natural support for** abstraction**,** pipelining **(streaming)** actor-orientation, **actor** reuse
- **User-Orientation**
  - Workflow design & exec console (Vergil GUI)
  - **"Application/Glue-Ware"**
    - excellent modeling and design support
    - run-time support, monitoring, …
    - **not** a middle-/underware (we use someone else's, e.g. Globus, SRB, …)
    - but middle-/underware is conveniently accessible through actors!
- **PRAGMATICS**
  - Ptolemy II is mature, continuously extended & improved, well-documented (500+pp)
  - open source system
  - many research results
  - Ptolemy II participation in Kepler

KEPLER/CSP: Contributors, Sponsors, Projects

Ilkay Altintas *SDM, NLADR, Resurgence, EOL, …*
Kim Baldridge *Resurgence, NMI*
Chad Berkley *SEEK*
Shawn Bowers *SEEK*
Terence Critchlow *SDM*
Tobin Fricke *ROADNet*
Jeffrey Grethe *BIRN*
Christopher H. Brooks *Ptolemy II*
Zhengang Cheng *SDM*
Dan Higgins *SEEK*
Efrat Jaeger *GEON*
Matt Jones *SEEK*
Werner Krebs, *EOL*
Edward A. Lee *Ptolemy II*
Kai Lin *GEON*
Bertram Ludaescher *SDM, SEEK, GEON, BIRN, ROADNet*
Mark Miller *EOL*
Steve Mock *NMI*
Steve Neuendorffer *Ptolemy II*
Jing Tao *SEEK*
Mladen Vouk *SDM*
Xiaowen Xin *SDM*
Yang Zhao *Ptolemy II*
Bing Zhu *SEEK*
•••

Ptolemy II

www.kepler-project.org

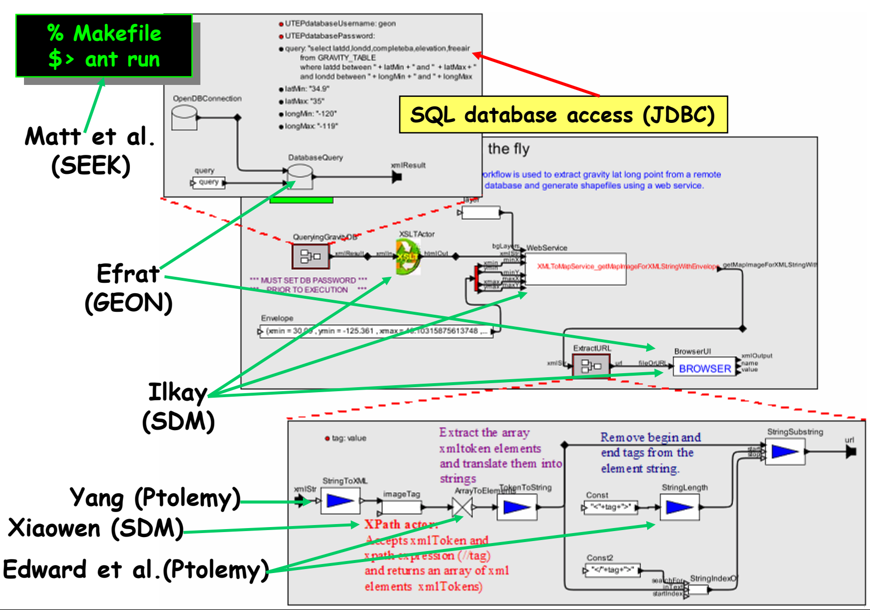
LLNL, NCSU, SDSC, UCB, UCD, UCSB, UCSD, U Man… Utah,…, UTEP, …, Zurich

ROADNet · SEEK · GEON The Geosciences Network · EOL · BIRN Biomedical Informatics Research Network · SciDAC Scientific Discovery through Advanced Computing · SPA SDM · DARPA · NSF · National Institutes of Health

Collab. tools: IRC, cvs, skype, Wiki: hotTopics, FAQs, ..

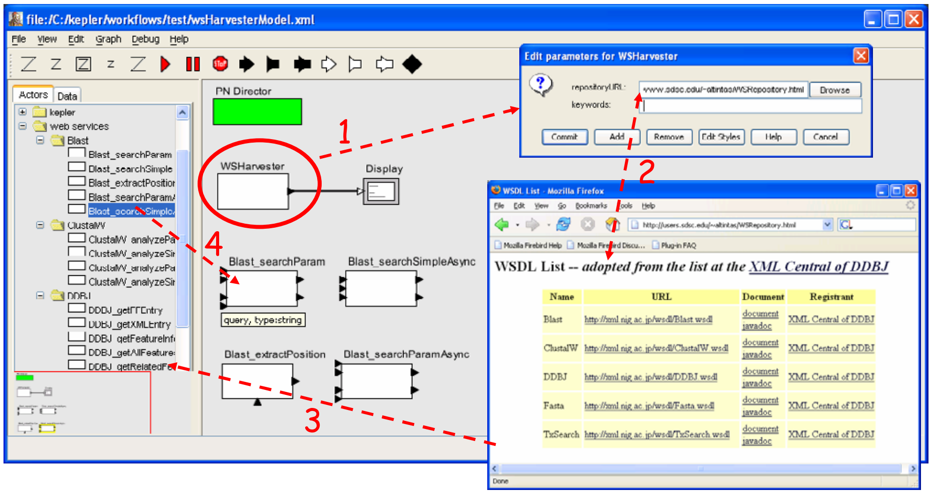
Bertram Ludäscher, UC **DAVIS**

GEON Dataset Generation & Registration
(and co-development in KEPLER)



UC **DAVIS**

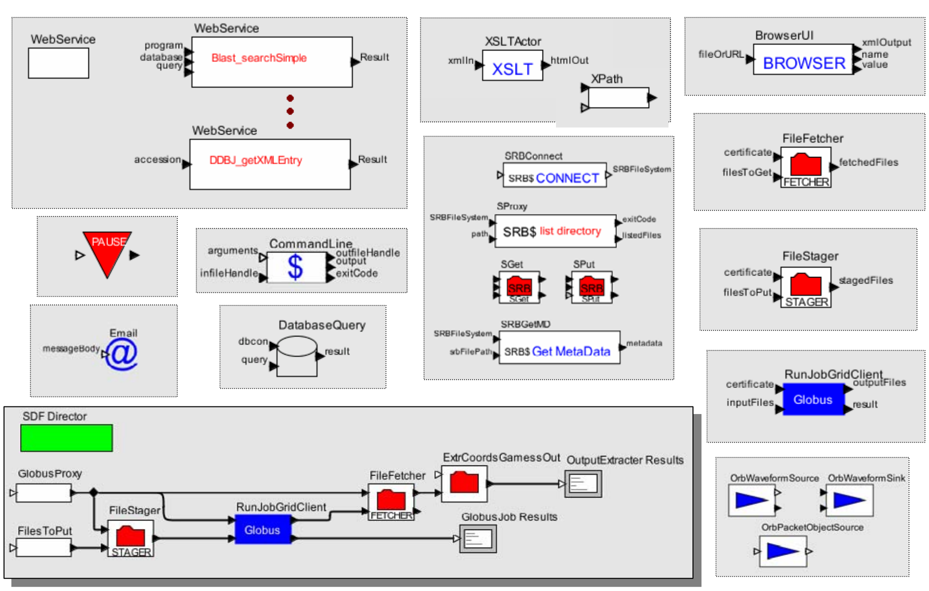
Web Services ➔ Actors (WS Harvester)



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

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

Some KEPLER Actors (out of 160+ … and counting…)



Kepler Today: Some Numbers

- **#Actors:**
  - Kepler: ~160 new + ~120 inherited (PTII)
  - soon there can be thousands (harvested from web services, R packages, etc.)
- **#Developers:**
  - ~ 24+, ~10 very active; more coming… (we think :-)
- **#CVS Repositories: ~2**
  - hopefully not increasing… :-{
- **# "Production-level" WFs:**
  - currently ~8, expected to increase quite a bit …

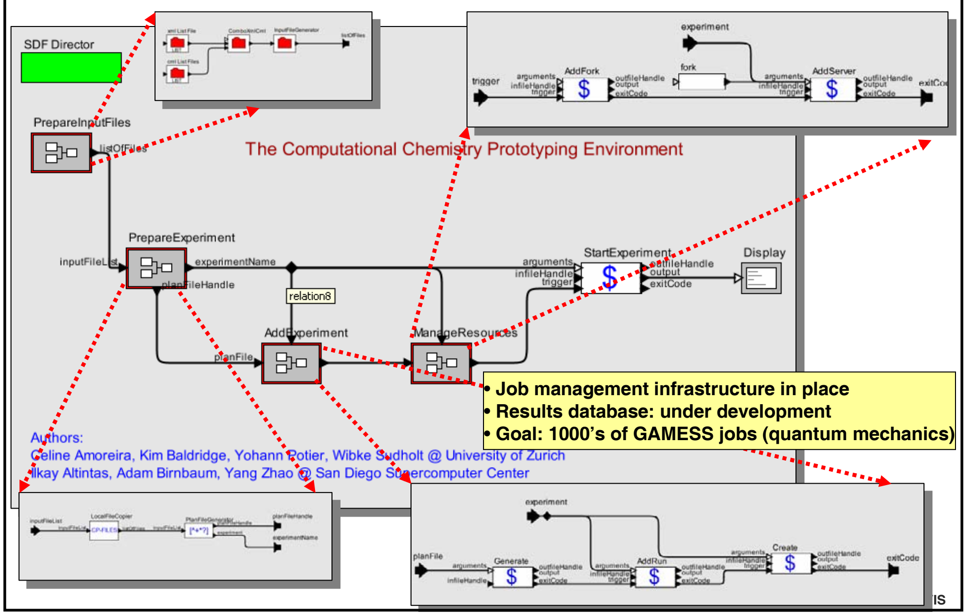
Bertram Ludäscher, UC **DAVIS**

A User's Wish List

- Usability
- Closing the "lid" (cf. vnc)
- Dynamic plug-in of actors (cf. actor & data registries/repositories)
- Distributed WF execution
- Collection-based programming
- Grid awareness
- Semantics awareness
- WF Deployment (as a web site, as a web service, …)
- "Power apps" (➔ SCIRun)
- …

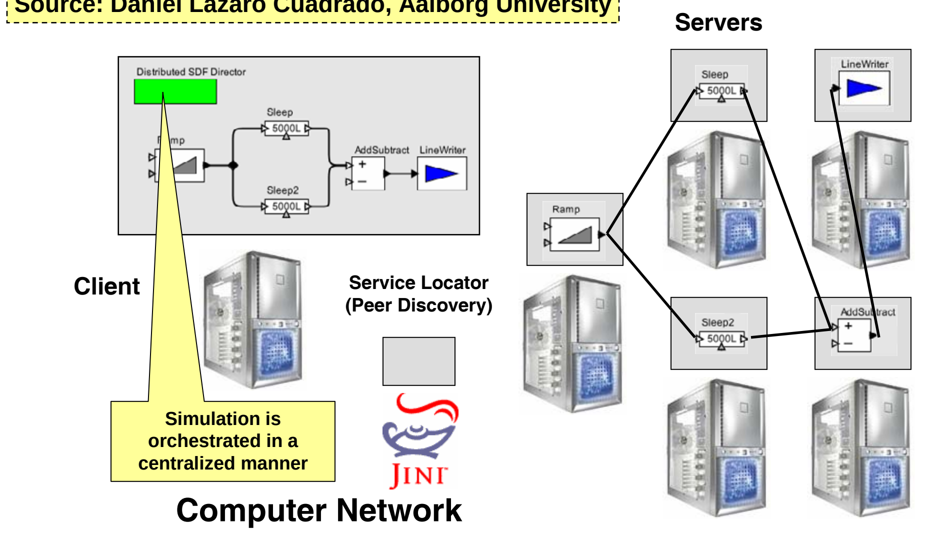
Bertram Ludäscher, UC **DAVIS**

Distributed Execution I:
External Job Manager (here: NIMROD)



Distributed Execution II:
"Distribution-Aware" Director

Source: Daniel Lázaro Cuadrado, Aalborg University



Separation of Concerns

- **A shining example:**
  - Ptolemy Directors – "factoring out" the concern of workflow "orchestration" (MoC)
  - common aspects of overall execution **not** left to the actors
- **Similarly:**
  - The "Black Box" ("flight recorder")
    - a kind of "recording central" to avoid wiring 100's of components to recording-actor(s)
  - The "Red Box" (error handling, fault tolerance)
    - ………
  - The "Yellow Box" (type checking)
    - ………
  - The "Blue Box" (shipping-and-handling)
    - central handling of data transport (by value, by reference, by scp, SRB, GridFTP, …)

SDF/PN/DE/…

Recorder

On Error

Static Analysis

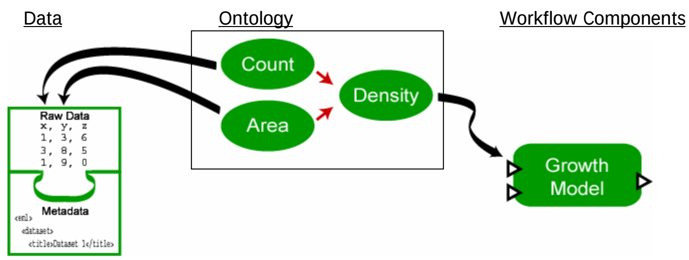
SHA @

Separation of Concerns: Port Types

- **Token consumption (& production) "type"**
  - a director's concern
- **Token "transport type"**
  - by value, reference (which one), protocol (SOAP, scp, GridFTP, scp, SRB, …)
  - a SHA concern
- **Structural and semantic types**
  - SAT (static analysis & typing) concern
  - built after static unit type system…
    - static unit type system as a special case!?

Need for Semantic Annotations of data & actors

- Label **data** with **semantic types** (concept expressions from an ontology)
- Label **inputs and outputs of analytical components** with **semantic types**



Example: Data has COUNT and AREA; workflow wants DENSITY

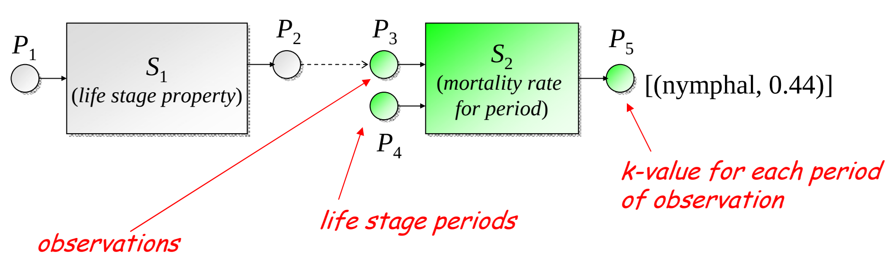
- ➔ via ontology, system "knows" that data can still be used

(because DENSITY := COUNT/AREA)

- Use **reasoning engines** to generate transformation steps
- Use **reasoning engine** to discover relevant components

Bertram Ludäscher, UC **DAVIS**

A Scientist's "Semantic" View of Actors



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]

Structural Type (XML DTD) Annotations

structType(P_2)

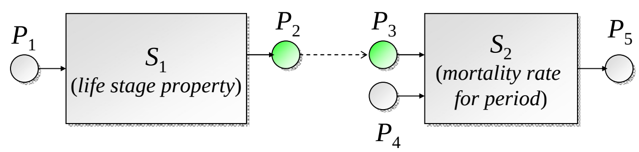
```
root population = (sample)*
elem sample     = (meas, lsp)
elem meas       = (cnt, acc)
elem cnt        = xsd:integer
elem acc        = xsd:double
elem lsp        = xsd:string
```

```
<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
```

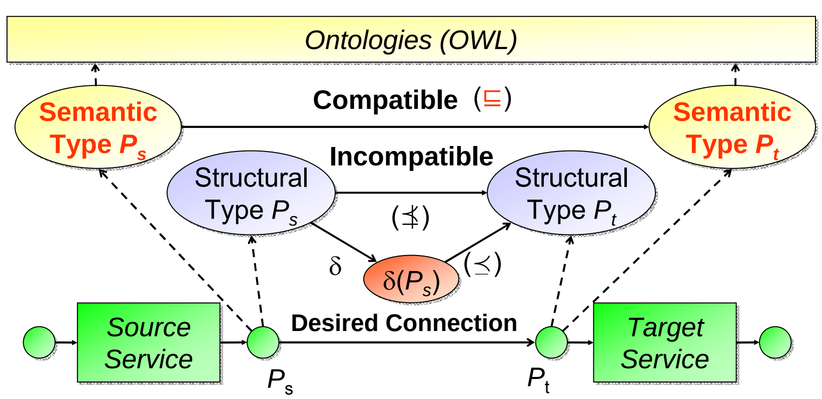
```
<cohortTable>
   <measurement>
      <phase>Eggs</cnt>
      <obs>44,000</acc>
   </measurement>
…
<cohortTable>
```



Source: [Bowers-Ludaescher, DILS'04]

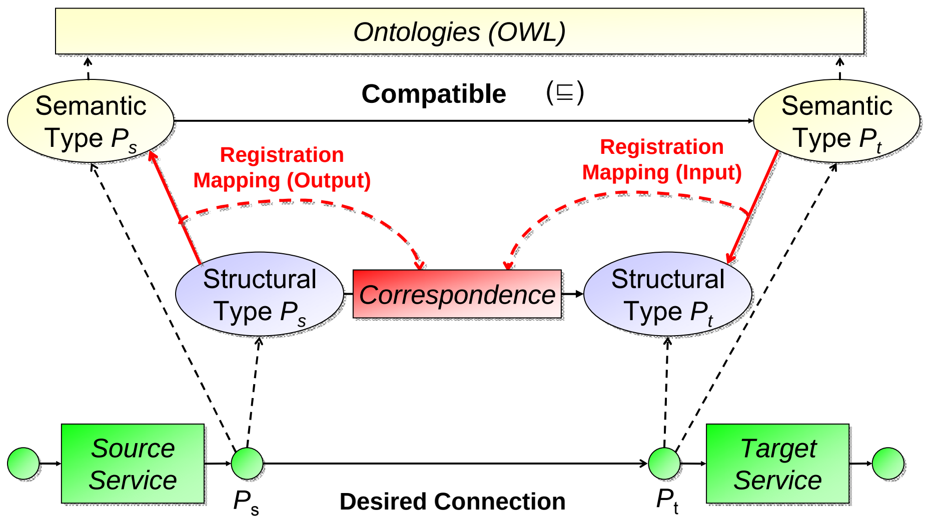
A KR+DI+Scientific Workflow Problem

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



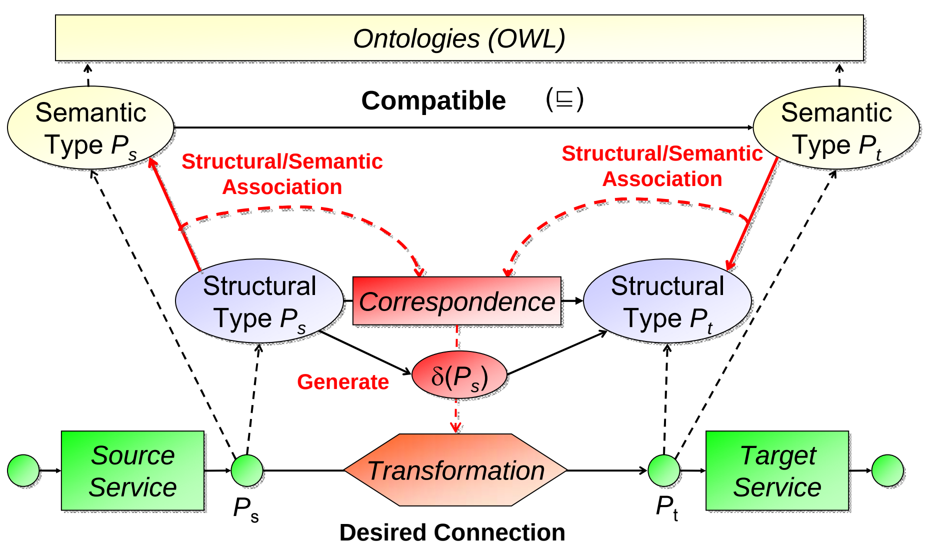
Source: [Bowers-Ludaescher, DILS'04]

The Ontology-Driven Framework



Bertram Ludäscher, UC **DAVIS**

Ontology-Guided Data Transformation



Source: [Bowers-Ludaescher, DILS'04]

Use of Semantics in SWF (DI+KR+SWF)

"Smart" Search

- Concept-based, e.g., "find all datasets containing biomass measurements"

Improved Linking, Merging, Integration

- Establishing links between data ***through*** semantic annotations & ontologies
- Combining heterogeneous sources based on annotations
- Concatenate, Union (merge), Join, etc.

Transforming

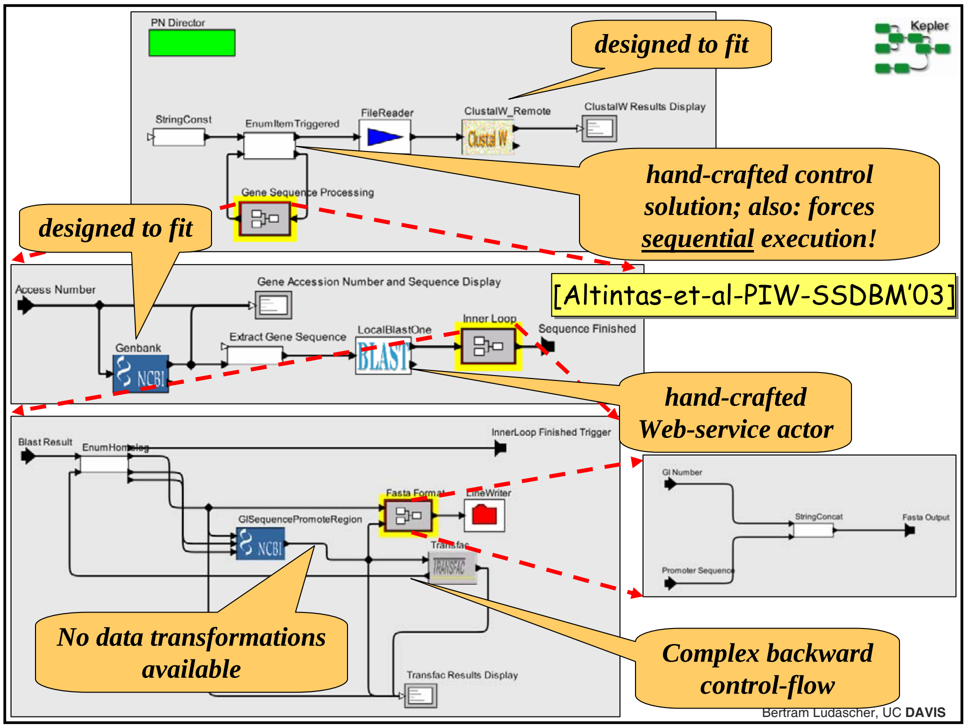
- Construct mappings from schema S1 to S2 based on annotations

Semantic Propagation

- "Pushing" semantic annotations through transformations/queries

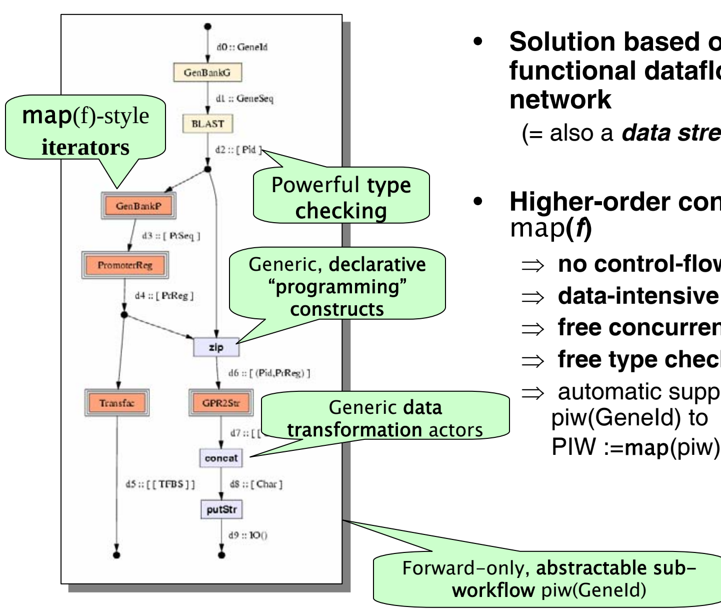
Bertram Ludäscher, UC **DAVIS**

Scientific Workflow Design



Bertram Ludäscher, UC **DAVIS**

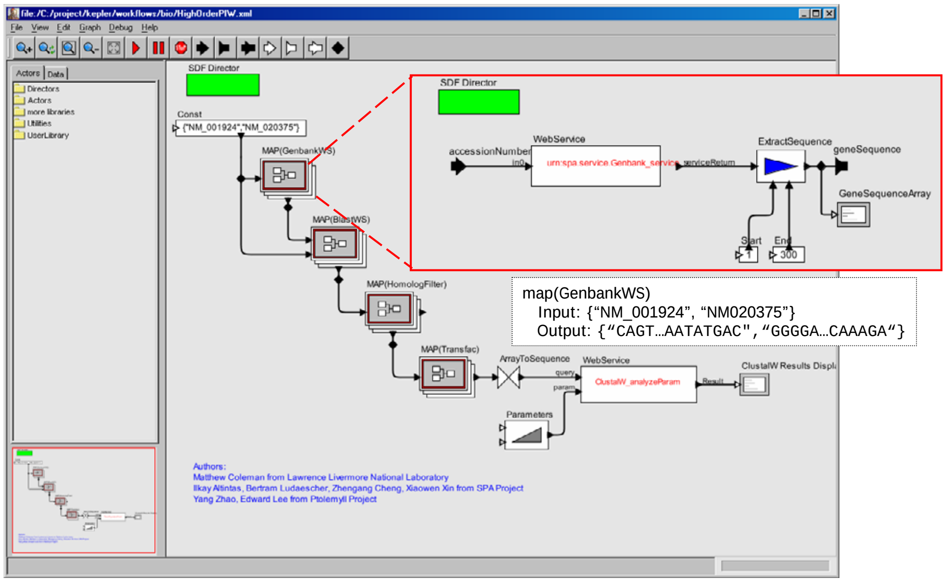
Scientific Workflow Design: The Functional Programming Model



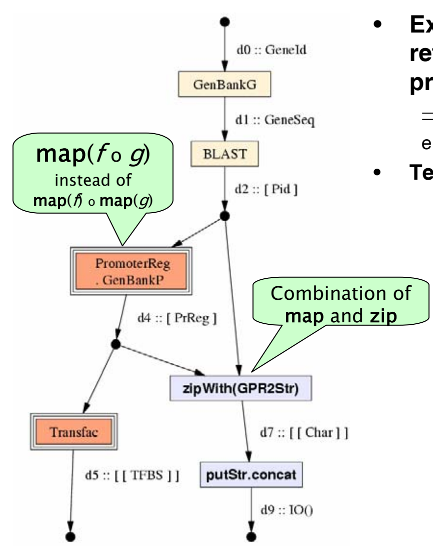
- **Solution based on declarative, functional dataflow process network**
  (= also a ***data streaming model***!)
- **Higher-order constructs:** map(***f***)
  - ⇒ **no control-flow spaghetti**
  - ⇒ **data-intensive apps**
  - ⇒ **free concurrent execution**
  - ⇒ **free type checking**
  - ⇒ automatic support to go from piw(GeneId) to PIW :=map(piw) over [GeneId]

Bertram Ludäscher, UC **DAVIS**

Scientific Workflow Design: The Functional Programming Model



Research Problem: Optimization by Rewriting



- **Example: 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**

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

Scientific Workflow Design: Challenges

While many systems (including Kepler) support execution … support for SWF conceptual modeling and design is lacking

- Formal models for scientific workflows
- Mechanisms for discovery, reuse, and adaptation of existing workflows and components
- End-to-end workflow development (methods and frameworks), especially for early stages

Bertram Ludäscher, UC **DAVIS**

Scientific Workflow Design: Contributions

While many systems (including Kepler) support execution … support for SWF conceptual modeling and design is lacking

- Formal models for scientific workflows

  Based on **Actor-Oriented Modeling**

- Mechanisms for discovery, reuse, and adaptation of existing workflows and components

  plus a rich **Type System** ("hybrid" types)

- End-to-end workflow development (methods and frameworks), especially for early stages

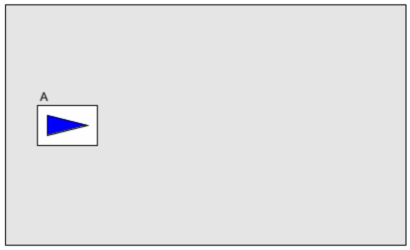
  Modeling Primitives

  (Adapters & Replacement; strategies)

Bowers-Ludaescher-ER-2005

UC **DAVIS**

Actor-Oriented Modeling

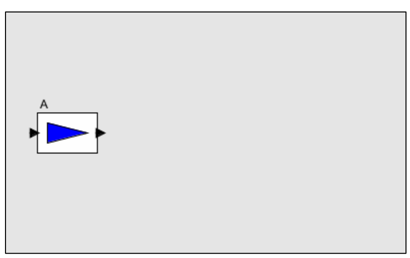


Actors

- single component or task
- well-defined interface (signature)
- dataflow view: given input data, produce output data

Bowers-Ludaescher-ER-2005 UC DAVIS

Actor-Oriented Modeling

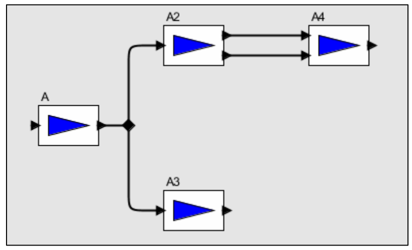


Ports

- each actor has a set of input and output ports
- denote the actor's signature
- produce/consume data (a.k.a. ***tokens***)
- ***parameters*** are special "static" ports

Bowers-Ludaescher-ER-2005 UC DAVIS

Actor-Oriented Modeling



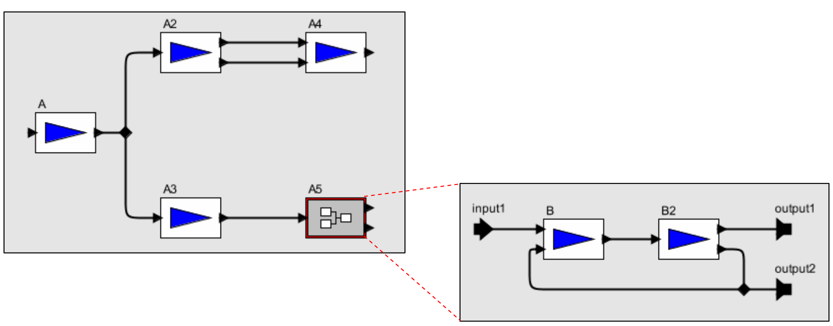
Dataflow Connections

- actor “communication” channels
- directed (hyper) edges
- connect output ports with input ports
- merge step + distribute step

Bowers-Ludaescher-ER-2005

UC DAVIS

Actor-Oriented Modeling



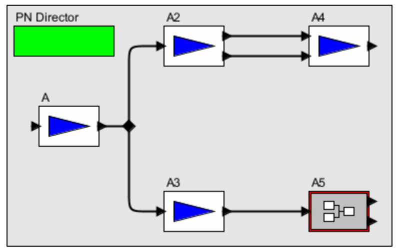
Sub-workflows / Composite Actors

- composite actors “wrap” sub-workflows
- like actors, have signatures (i/o ports of sub-workflow)
- hierarchical workflows (arbitrary nesting levels)

Bowers-Ludaescher-ER-2005

UC DAVIS

Actor-Oriented Modeling



Directors

- define the **Model of Computation (MoC)** of workflow graphs
- executes workflow
- sub-workflows may have different directors
- enables reusability

Bowers-Ludaescher-ER-2005

UC DAVIS

Models of Computation

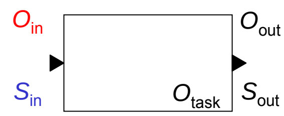
Directors separate the concerns of WF orchestration from Actor execution

- **Process Networks (PN)**
  - Actors execute as independent processes; blocking reads; non-blocking writes; FIFO buffers (queues) of unbounded size
- **Synchronous Dataflow (SDF)**
  - Fixed buffer size, since schedules are statically precomputed (based on token production/consumption rates/**firing**). SDF MoC is highly analyzable and used often in SWFs.
- Continuous Time (CT)
  - Connections represent the value of a continuous time signal at some point in time ... Often used to model physical processes.
- Discrete Event (DE)
  - Actors communicate through a queue of events in time. Used for instantaneous reactions in physical systems.
- …

Bowers-Ludaescher-ER-2005

UC DAVIS

“Hybrid” Types



O_{in} : Measurement ⊓ ∀ItemMeasured.SpeciesOccurrence

S_{in} : r(site, day, spp, occ)

Structural Types: Given a type language* $\mathcal{S}$

- Any port can be associated with a type $S \in \boldsymbol{S}$
- Kepler types include atomic (int, double, string) and complex types (record, list, etc.)

Semantic Types: Given an ontology language $\mathcal{O}$

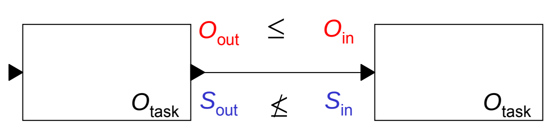
- Any port can be associated a type $O \in \boldsymbol{O}$
- We consider description logic ontologies (e.g., OWL-DL)

* **e.g., XML Schema, DTDs, relational,** Kepler data types

Bowers-Ludaescher-ER-2005

UC DAVIS

Advantages of Hybrid Types



Semantically compatible but structurally incompatible

Separates concerns of structural data typing and conceptual data typing

- Motivated by interoperating **legacy** and **independently** created components
- Allows one to be specified first (e.g., semantic)
- Structural **validation** independent from semantic validation
- Enables **discovery**
- …

Bowers-Ludaescher-ER-2005

UC DAVIS

Semantic Annotations

Semantic and structural types can be "glued" together

… via logical constraints

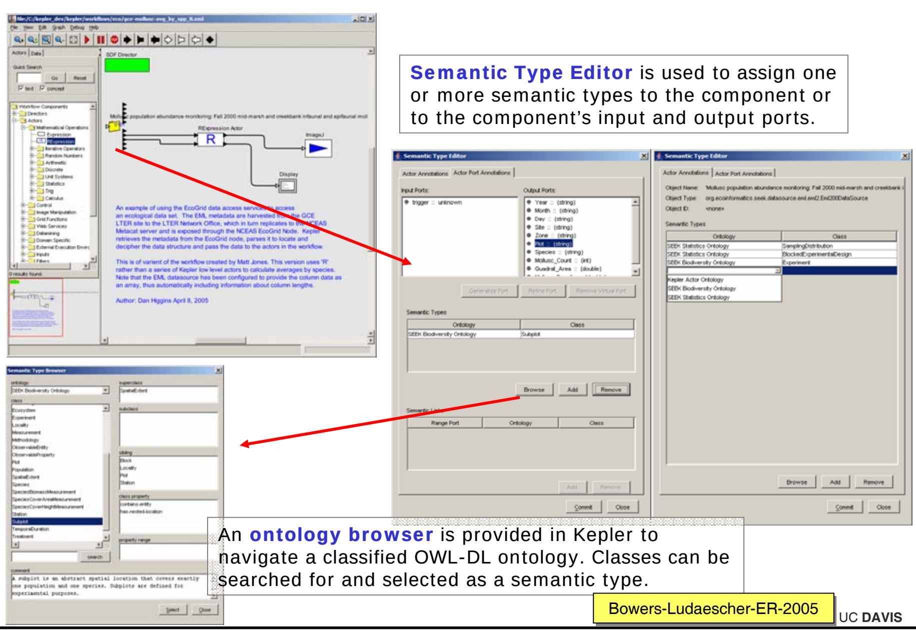
∀site,day,spp,occ R(site, day, spp, occ) →
∃y Measurement(y), ItemMeasured(y, occ), SpeciesOccurrence(occ)

- Can provide **structural correspondences** between semantically compatible, structurally incompatible ports
- When something is known about the input/output dependencies of an actor, semantic types can be **propagated**

Bowers-Ludaescher-ER-2005

UC DAVIS

Semantic Annotations and Hybrid Types



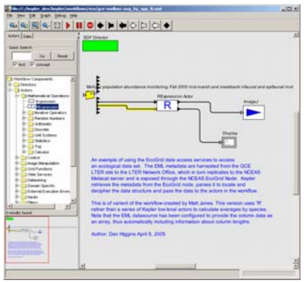
Semantic Type Editor is used to assign one or more semantic types to the component or to the component's input and output ports.

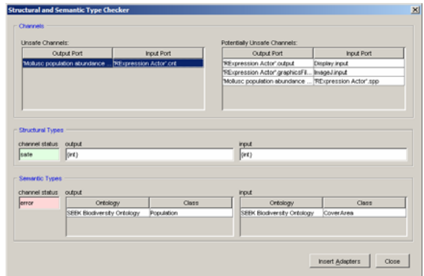
An **ontology browser** is provided in Kepler to navigate a classified OWL-DL ontology. Classes can be searched for and selected as a semantic type.

Bowers-Ludaescher-ER-2005

UC DAVIS

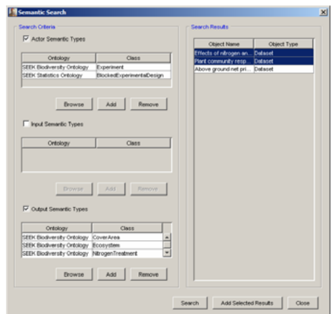
Semantic Annotations and Hybrid Types

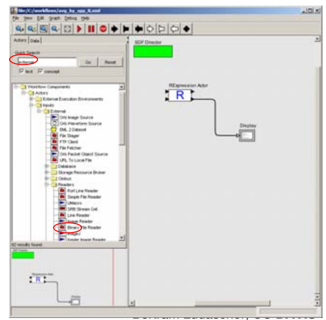




Verifying structural and **semantic compatibility** of workflow connections in Kepler

Searching based on actor-level and input/output port **Semantic Types** in Kepler

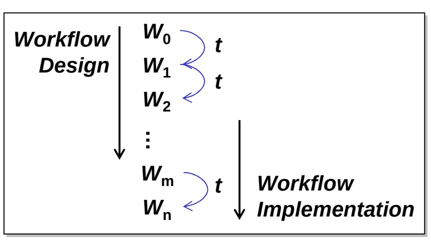


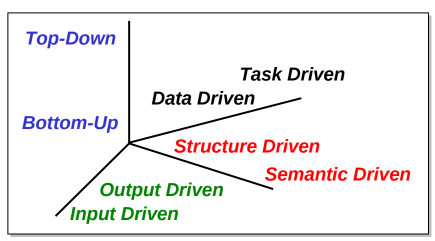


Workflow Design Primitives

End-to-End Workflow Design and Implementation

- Viewed as a series of primitive "transformations"
- Each takes a SWF and produces a new SWF
- Can be combined to form design "strategies"





E.g., re-engineering SWFs often is top-down, structure driven, whereas new SWFs are often a mix of semantic, input, and output driven

Bowers-Ludaescher-ER-2005

UC DAVIS

Basic Actor-Oriented Primitives

Kepler

Basic Transformations	*Starting Workflow*	*Resulting Workflow*	*Resulting Workflow*
t_1: Entity Introduction (actor or data connection)			
t_2: Port Introduction			
t_3: Datatype Refinement (s′ ≼ s, t′ ≼ t)	s t	s′ t	s t′
t_4: Hierarchical Abstraction			
t_5: Hierarchical Refinement			
t_6: Data Connection			
t_7: Director Introduction			

Bowers-Ludaescher-ER-2005

UC DAVIS

Additional Primitives

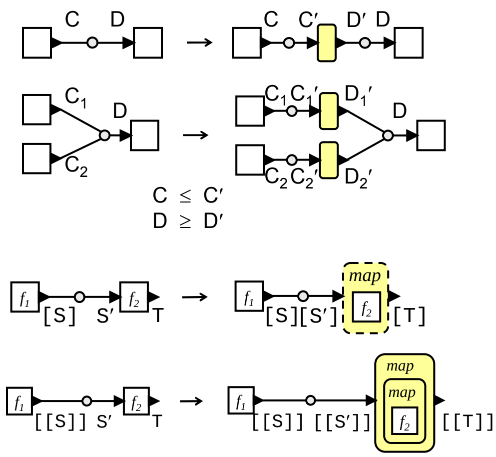
Kepler

Transformations	*Starting Workflow*	*Resulting Workflow*	*Resulting Workflow*
t_9: Actor Semantic Type Refinement ($T' \leq T$)	T	T'	
t_{10}: Port Semantic Type Refinement ($C' \leq C$, $D' \leq D$)	C D	C′ D	C D′
t_{11}: Annotation Constraint Refinement ($\alpha' \to \alpha$)	α_1 C s D α_2 t	α'_1 C s D α_2 t	α_1 C s D α'_2 t
t_{12}: I/O Constraint Strengthening ($\psi \to \varphi$)	φ	ψ	
t_{13}: Data Connection Refinement			
t_{14}: Adapter Insertion			
t_{15}: Actor Replacement	f	f'	
t_{16}: Workflow Combination (Map)	f_1 f_2	f_1 f_2	

Bowers-Ludaescher-ER-2005

UC DAVIS

Adapters (Semantic and Structural Incompatibility)



Adapters:

– Can be abstract (no implementation) or concrete

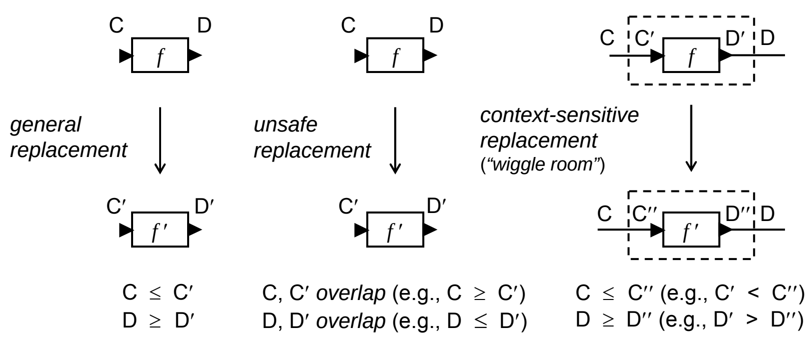
– Can bridge “semantic gaps” or fix structural mismatches

– Can be generated automatically (e.g., Taverna’s “list mismatch”)

– Can be reused (based on signatures)

Bowers-Ludaescher-ER-2005 UC DAVIS

The Replacement Primitive



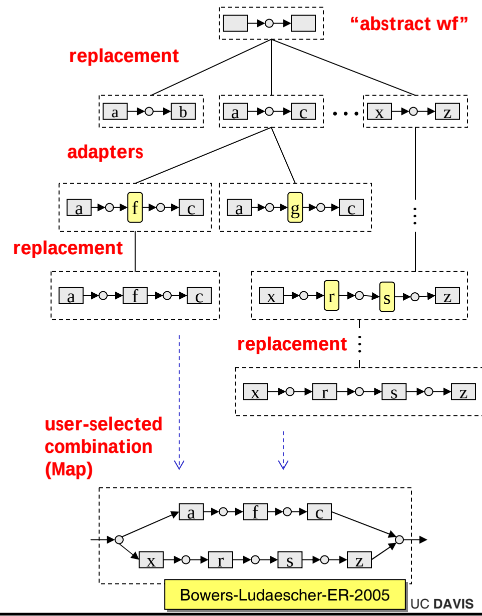
– General replacement doesn’t consider surrounding connections

– Context-sensitive replacement gives more “wiggle room” by “tuning” the actors semantic types based on connections

Bowers-Ludaescher-ER-2005 UC DAVIS

Applying adapter insertion and replacement

- **Adapter insertion and replacement can enable simple SWF elaboration**
  - Given an initial set of connected abstract actors
  - Repeatedly search for replacement concrete actors (atomic/composite)
  - At each step, insert adapters when necessary
  - Allow user to combine desired "variations"



UC DAVIS

The KEPLER Project

The Kepler Scientific Workflow System

- Built on Ptolemy II (UCBerkeley), developed by the Electrical Engineering community (circuit design and simulation)
- Open-source, Java
- Computation Models, Nested WFs, Loops
- Graphical Workflow Interface
- Workflow Execution
- Extensible Architecture
- Component Libraries
- Metadata, Discovery, Archival

The Kepler "Vision"

- *End-to-end scientific workflow design and execution environment*
- *Data and compute intensive workflows*
- *Comprehensive component libraries for a wide range of scientific domains*
- *Enable collaboration, sharing across disciplines ("synergy")*

Natural Diversity Discovery Project

Science Environment for Ecological Knowledge

Ptolemy II

Real-time Observatories Applications and Data Management Network

Encyclopedia of Life

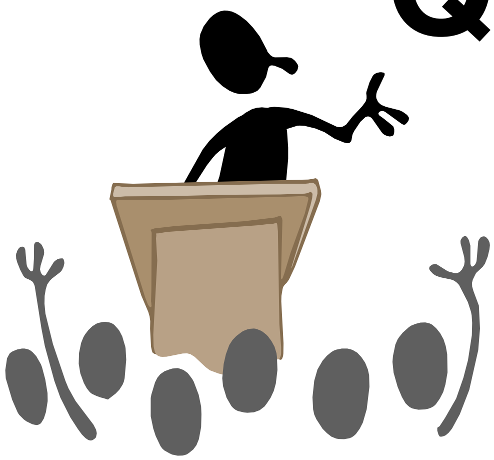
The Geosciences Network

KEPLER ***kepler-project.org***

NDDP	www.nddp.org
EOL	eol.sdsc.edu
GEON	www.geongrid.org
Ptolemy II	ptolemy.eecs.berkeley.edu/ptolemyII
ROADNet	roadnet.ucsd.edu
SEEK	seek.ecoinformatics.org
SciDAC	www-casc.llnl.gov/sdm

Bertram Ludäscher, UC DAVIS

Q & A



Bertram Ludäscher, UC **DAVIS**