Final Project Assignments

- Schema Matching
  - Ji-Yeong Chong
- Biological Pathways & Ontologies
  - Russell D’Sa
- FCA – Theory and Practice
  - Bill Man, Betty Chan
- Practice of Data Integration (GAV)
  - Jenny Wang
- Kepler/Data Analysis (Biodiversity)
  - Mike Kofi (Biodiversity)
  - Carlos Rueda (visualization focus)
- Kepler/Data-intensive & SRB
  - Tim Wong
- Kepler/ROADnet/Geostreams
  - Haiyan Yang

Promoter Identification Workflow (PIW)

Source: Matt Coleman (LLNL)

KEPLER/CSP: Contributors, Sponsors, Projects
(or loosely coupled Communicating Sequential Persons :-)

www.kepler-project.org
GEON Dataset Generation & Registration
(a co-development in KEPLER)

SQL database access (JDBC)

Matt, Chad, Dan et al. (SEEK)

Edward et al. (Ptolemy)

Efrat (GEON)

Ilkay (SDM)

Edward et al. (Ptolemy)

Ptolemy II/KEPLER GUI (Vergil)

"Directors" define the component interaction & execution semantics

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

Web Services ➔ Actors
(WS Harvester)

"Minute-made" (MM) WS-based application integration
• Similarly: MM workflow design & sharing w/o implemented components

Rapid Web Service-based Prototyping
(Here: ROADNet Command & Control Services for LOOKING Kick-Off Mtg)

Source: Ilkay Altintas, SDM, NLADR
ROADNet: Vernon, Orcutt et al
Web services: Tony Fountain et al
An "early" example: Promoter Identification
SSDBM, AD 2003

- Scientist models application as a "workflow" of connected components ("actors").
- If all components exist, the workflow can be automated/executed.
- Different directors can be used to pick appropriate execution model (often "pipelined" execution PN director).

"Run Window"

Enter initial inputs, Run and Display results

PIW Workflow Today

Custom Visualization
Job Management (here: NIMROD)

- Job management infrastructure in place
- Results database: under development
- Goal: 1000’s of GAMESS jobs (quantum mechanics)

Some Recent Actor Additions

in KEPLER (w/ editable script)

Source: Dan Higgins, Kepler/SEEK

in KEPLER (interactive session)

Source: Dan Higgins, Kepler/SEEK
### Scientific “Workflows”: Some Findings

- More **dataflow** than (business control-) workflow
  - DiscoveryNet, Kepler, SCIRun, Scilaeic, 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 cycles:
  - “(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 often **divorced**!
Focus on Actor-Oriented Design

- **Object orientation:**
  - class name
data
  - methods
  - call return

  What flows through an object is sequential control

- **Actor orientation:**
  - actor name
data (state)
  - ports
  - Input data
  - parameters

  What flows through an object is streams of data

Object-Oriented vs. Actor-Oriented Interface Definitions

- **Object Oriented**
  - TextToSpeech
    - speak(string text); void speak(double text);

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

- **Actor Oriented**
  - “Give me text and I’ll give you speech”

  AO interface definition says “Give me text and I’ll give you speech”

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

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 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

*Note: Ptolemy has been out for long!
Directors and Combining Different Component Interaction Semantics

KEPLER & ROADNet: Real-Time Scientific Workflows (Tobin Fricke et al.)

A Scientific Workflow Problem: Solved

Optimization by Declarative Rewriting I
Optimizing II: Streams & Pipelines

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

Traffic info for a list of highways: Uses iterate (higher-order "map") actor to access highway info web service repeatedly, sending out one email per highway.

An (oversimplified) Model of the Grid

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

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

Find Host Assignment: di ⇒ hi, fi ⇒ hi
  - for all di, fi s.t. [...; d3@h3 := f1@d1@h1, ...] is a valid plan

Shipping and Handling Algebra (SHA)

1. plan Y@C = F@A of X@B =
   1. [ X@B to A, Y@A := F@A(X@A), Y@A to C ]
   2. [ F@A ⇒ B, Y@B := F@B(X@B), Y@B to C ]
   3. [ X@B to C, F@A ⇒ C, Y@C := F@C(X@C) ]

Logical view

Physical view: SHA Plans