Workflow Engines: Why So Many?

Hook Hua (NASA/JPL)

ESIP Information Technology and Interoperability Rants and Raves Webinar Series

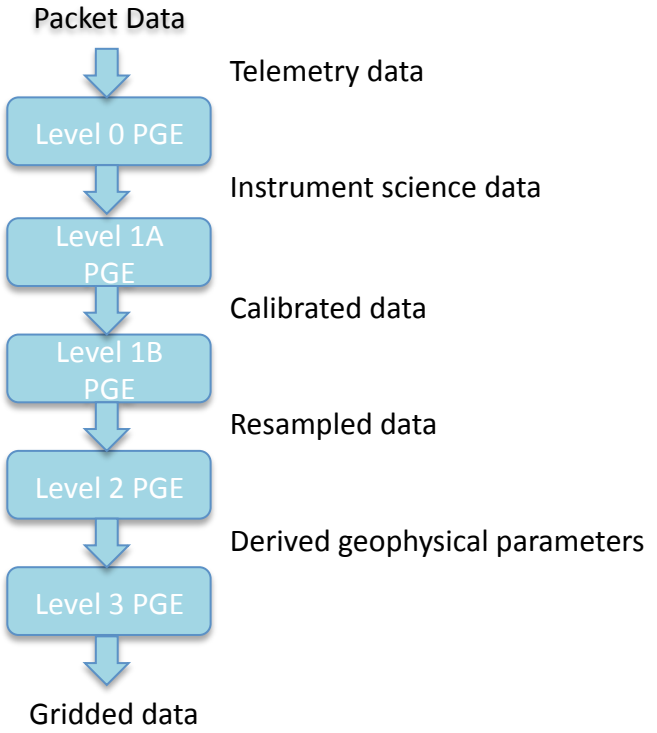
Wednesday, April 7, 2010

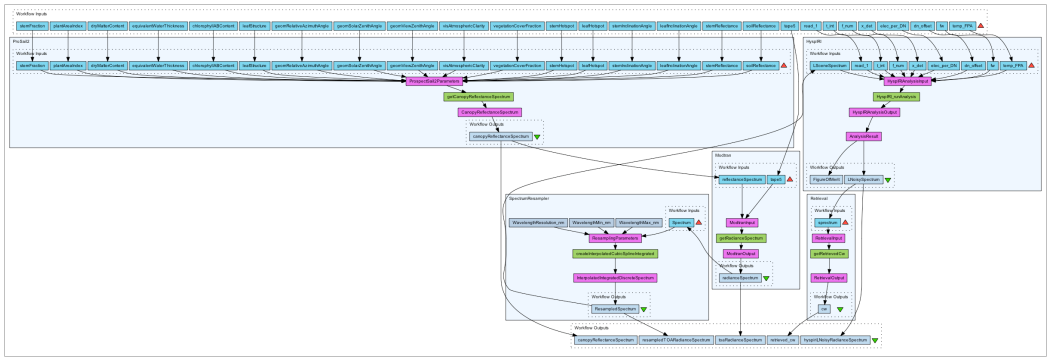
Overview

1. So many workflows
2. Workflow management
3. Example workflow engines
4. Earth Science processing
   - Workflow patterns
   - Useful features
5. So why so many?

Workflow Engines

- Facilitates the flow of information, tasks, and events
- Provides method of orchestrating individual execution units
- Management of control flow and data flow
- Connects distributed models
- Codify production rules / policies





Increasingly being used in Earth science processing

ARE THERE ANY CONSISTENTLY POPULAR WORKFLOW ENGINES IN USE?

Duopolies and Oligopolies?

- *"a market form in which a market or industry is dominated by a small number of sellers."**
- The four-firm concentration ratio
  - Verizon, AT&T, Sprint Nextel, and T-Mobile
  - Sony Music Entertainment, Universal Music Group, Warner Music Group, and EMI
  - JDeveloper, Eclipse, NetBeans, and IntelliJ IDEA
- Duopolies
  - Visa and Mastercard
  - Airbus and Boeing
  - ATI and Nvidia
  - Intel and AMD
  - Oracle and MySQL
  - Java and C#
  - Python and Ruby
  - Matlab and IDL
  - HDF and NetCDF

* http://en.wikipedia.org/wiki/Oligopoly

What About Workflow Engines?

- ActiveBPEL
- Antflow
- Apache Agila
- Apache ODE
- Beexee
- Bonita
- Bossa
- BpmScript
- Carnot
- con:cern
- Dalma
- Eclipse Java W
  Tooling
- Modeling Workfl
  Engine (MWE)
- Enhydra Shark
- FlowMind
- Flux
- Freefluo
- Galaxia
- Imixs IX Workflow
- jawflow
- JBoss jBPM
- JFlower
- JFolder
- kbee.w
- Windows Workflow
  dation
- ObjectWeb Bonita
- Open Business Engine
- OpenSymphony OSWorkflow
- OpenWFE
- Pegasus
- Phoenix Integration PHX ModelCenter
- PXE
- ruote (ruby)
- UNA WFE
- asvati
- ish
- Syrup
- Taverna
- Triana
- Tobflow
- Web and Flo / Kontinuum
- Werkflow
- WfMOpen
- Wilos
- Workpoint
- XFlow
- YAWL
- Zebra

and many more...!

Why So Many?

- No general dominating workflow engine
- Most can exec processes
- Many support invoking web services
- Many written in Java
- Many target business processes
- Others target scientific processes
- Many support control logic
- Many are derivatives of other implementations

Workflow Management

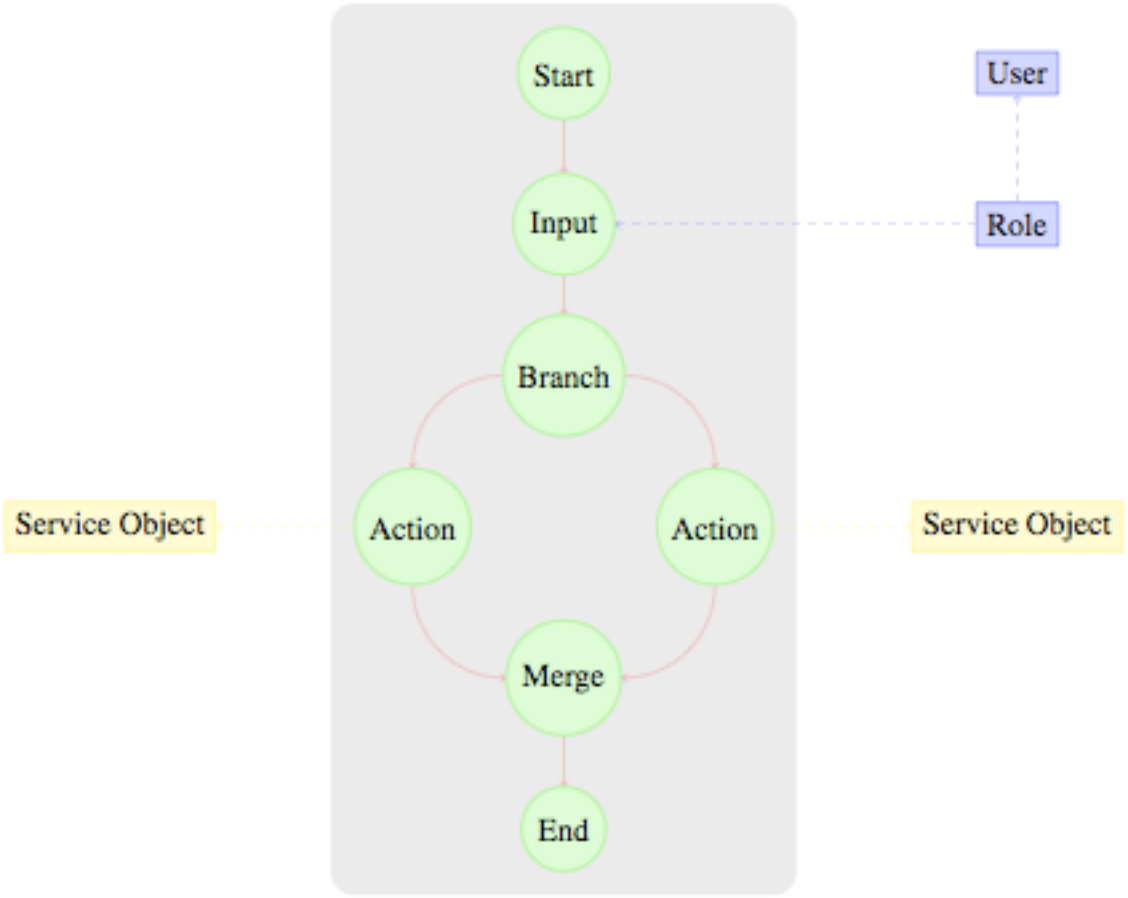


Figure 1.1: Who must do what when and how?

Design and Implementation of a Workflow Engine, Sebastian Bergmann, IAI-TR-2007-5, September 2007.

Workflow Definition Language Models

- **Dataflow model / Entity-based**
  - The workflow is constructed from data processing and data transport (processors and data links).
  - Directed graphs
  - *Natural for scientific workflows*
  - E.g. Simple Conceptual Unified Flow Language (Scufl)
- **Process-centric model / Activity-based**
  - The nodes in the workflows are activities and the "data" passed between them form a control system rather than being a genuine flow of messages.
  - "State transitions"
  - *Natural for business processes*
  - E.g. Business Process Execution Language (BPEL)

Workflow Engines

SOME EXAMPLES

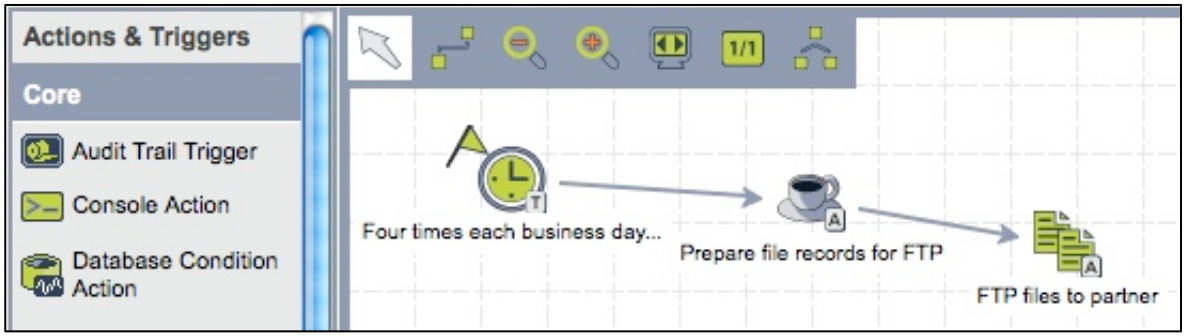
Example BPEL-based Workflow Engines

- Apache ODE (Orchestration Director Engine)
- OASIS WS-BPEL 2.0 standard /compatibility for BPEL4WS 1.1

- Enterprise business process orchestration and BPM.
- Coordinate people, application and services

- Automate and streamline the intricate processes in the enterprise.
- Torque open source resource manager

- Job scheduling, File Transfer, Workflow and business process management (BPM) engine



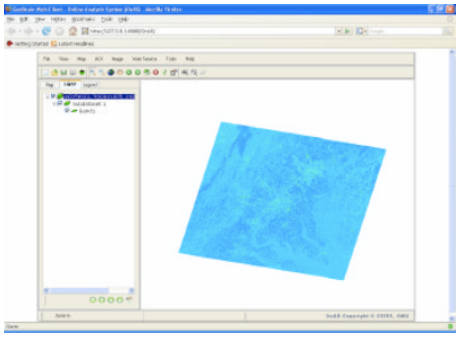
GMU's GeoBrain

BPELPower

- BPEL-based web service chaining from web application servers

GeoBrain Online Analysis System (GeOnAS)

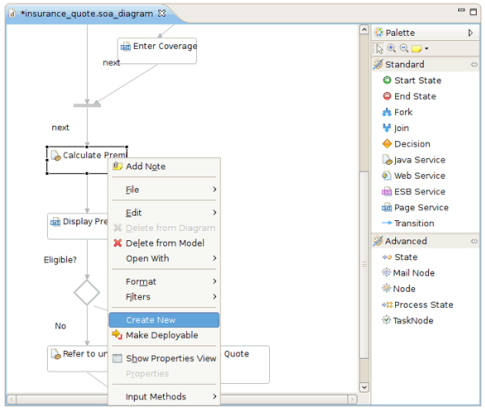
- Automated data access, management, visualization, analysis, and workflow composition
- *Demoed automated service workflow composition*



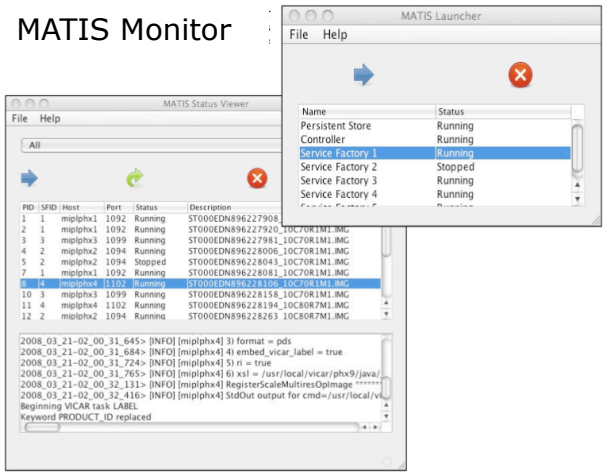
Multi-mission Automated Task Invocation System (MATIS)

- A distributed workflow manager used for automated product generation.
- Built from jBPM (jBoss Business Process Management)
  - Based on BPEL
- Used in JPL production missions
  - Phoenix and Diviner
  - Future: MCS and MSL
- Consists of
  - a multi-mission core workflow component (JBoss jBPM)
  - a project-specific adaptation

BPEL Editor

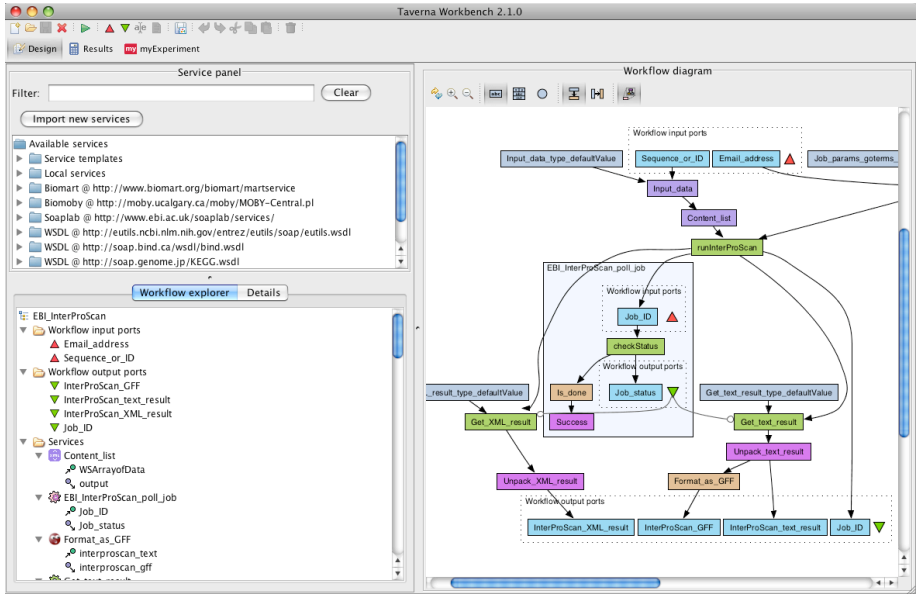


MATIS Monitor



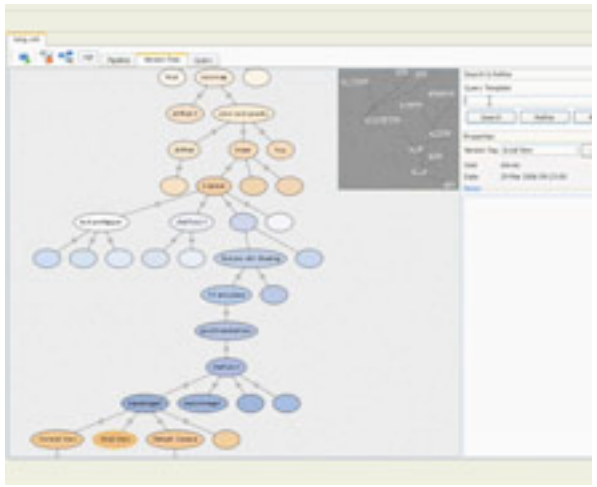
Taverna Workbench

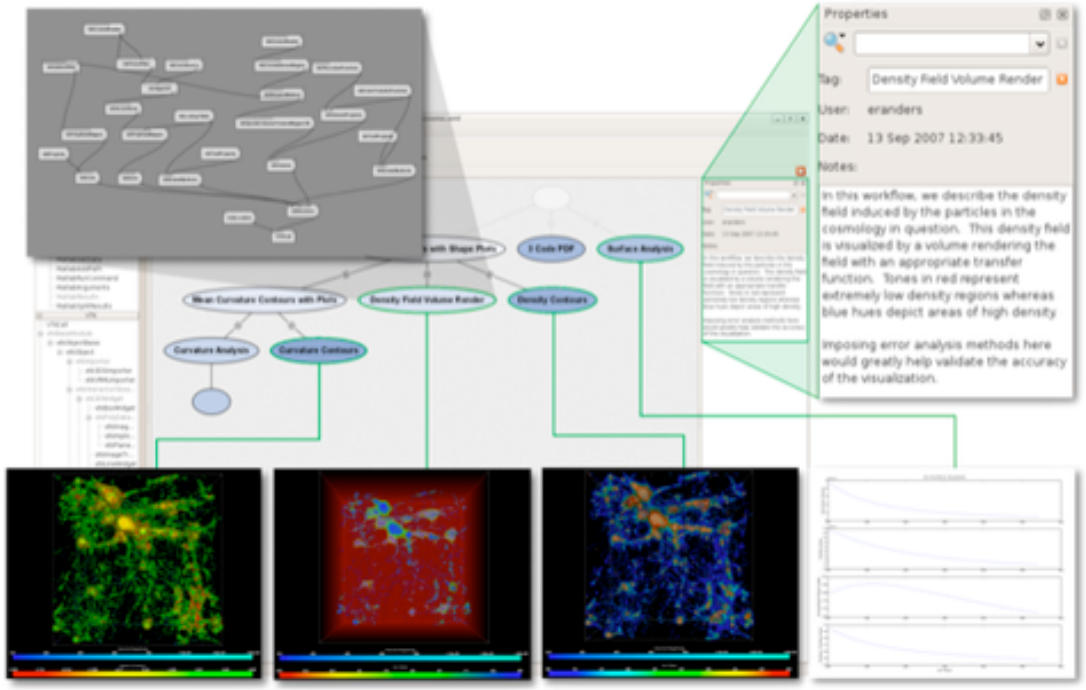
- An open source tool for designing and executing workflows created by the myGrid project and funded through the OMII-UK.
- Supports nesting of workflows and parallel execution
- Vectorization/iteration
  - Dot product and cross product enumerations
- SCUFL2
- Mature
- “fault tolerant”
- myExperiment Collaboration
- GUI workflow editor and visualization
- API built with software design patterns
  - E.g. enables easy adding of **provenance** observers/listeners



VisTrails

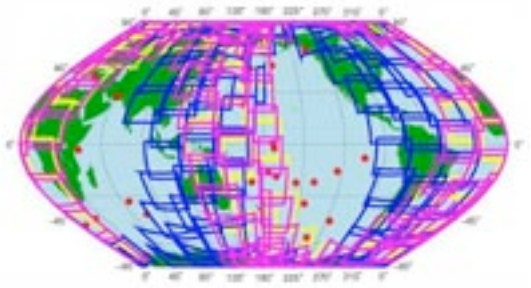
- An open-source scientific workflow and provenance management system developed at the University of Utah that provides support for data exploration and visualization.
- Emphasis on visualization and **provenance**
- Workflow nesting
- Workflow **versioning**
- Python-centric
- Academia adaptations





SciFlo

- Scientific Dataflow
- Python
- Web-based
  - AJAX editor
- Employs a Peer-to-Peer (P2P) Network of Grid workflow nodes
- Data & operator movement
  - Sometimes better to migrate processing, not data



Phoenix Integration PHX ModelCenter

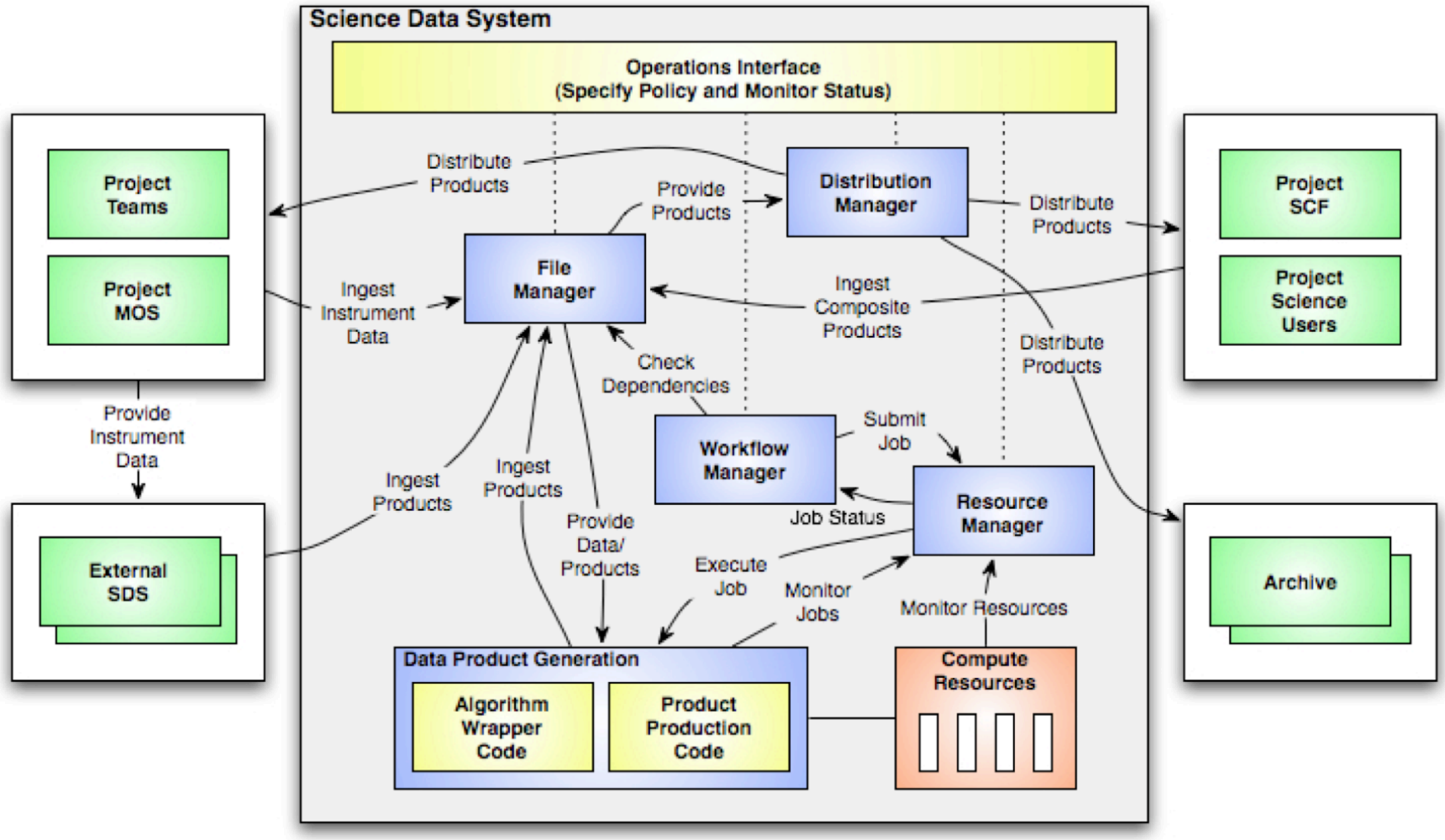
- Commercial (~$30K?)
- Windows-centric
- Design-Of-Experiments
- Trade studies
- Plugins connect to Excel, Matlab, Mathematica, JMP, Pspice, etc.

Workflow Design Patterns

SOME PATTERNS FOR EARTH SCIENCE PROCESSING

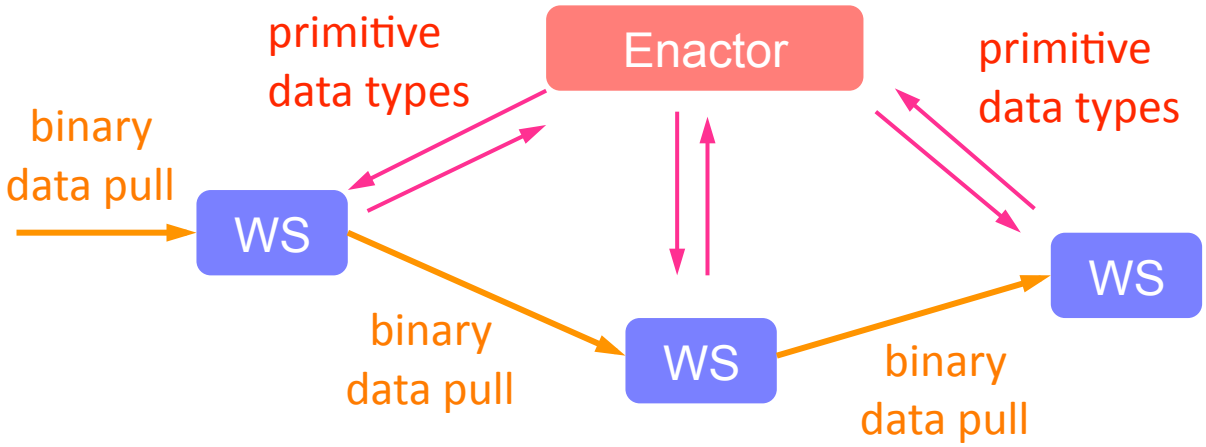
Usage in Science Data Systems

Generic Software Architecture View



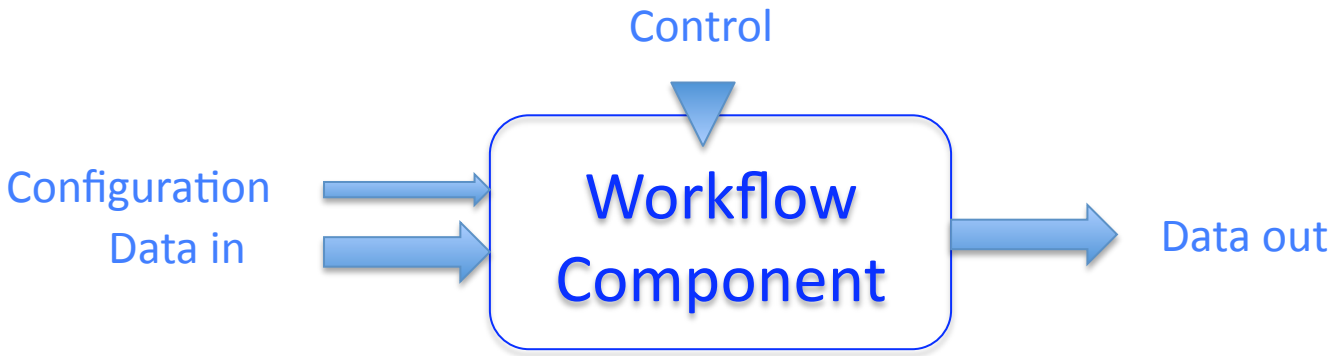
Handling Large Data Transfers

- Keep interface of workflow connections light
  - Orchestration engine passes data location, and not the data itself
- Each service endpoint pulls in its own large input data



Configuration Not in Flow

- Configuration for each workflow component should not be in workflow pipes
- "lazy loading" of configuration
  - Each workflow component reads configuration settings from file
- Enables modifications to configuration for long running workflow instances

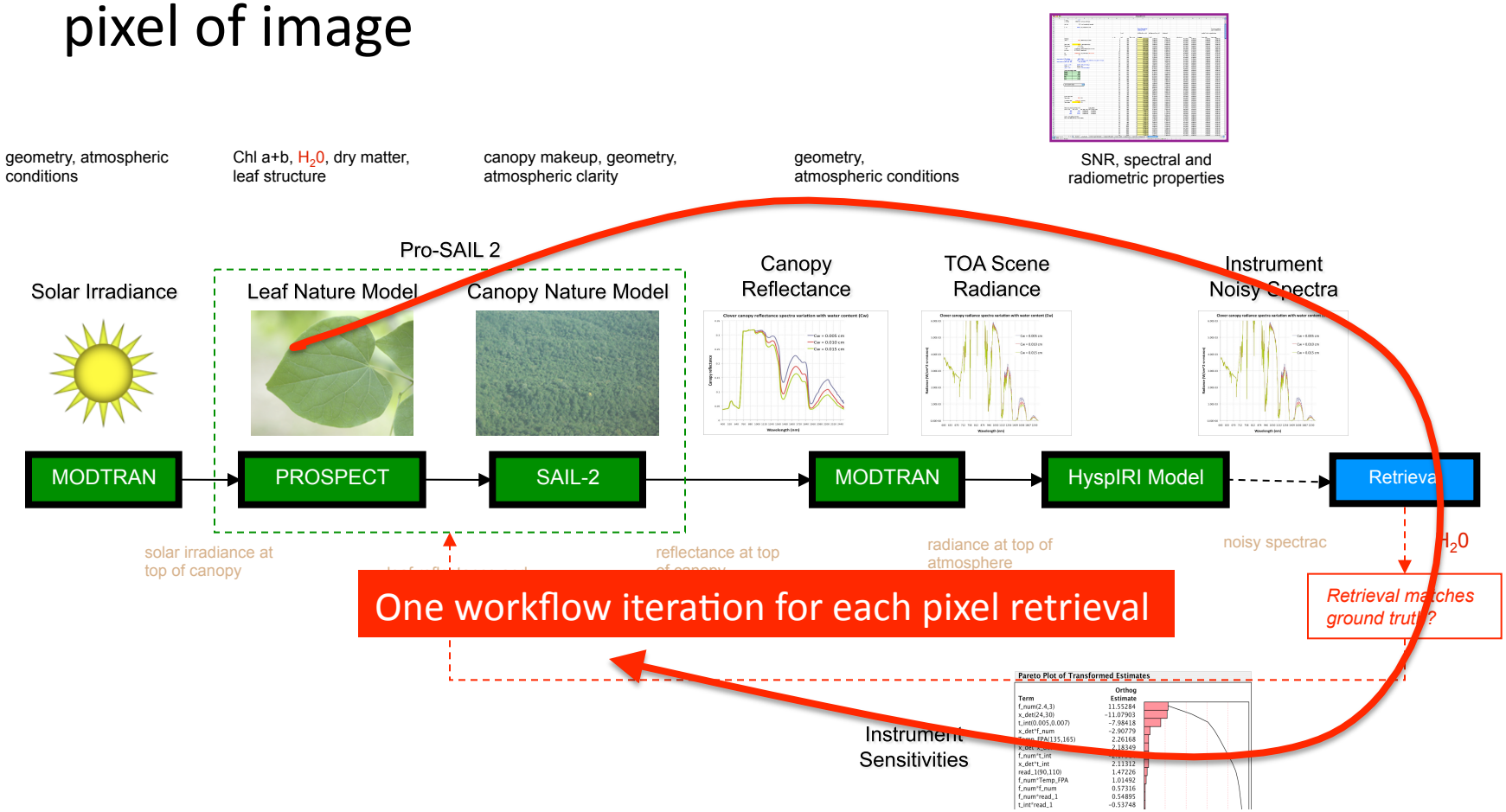


Outdated Input Settings

- Long runtimes of PGE
- Need to check configuration inputs once PGE completed in case of change.
- Rerun PGE workflow component if input configuration has changed

Vectorizing Runs

- Apply workflow on a sequence of data
- Example: Hyperspectral retrieval iterating through each pixel of image



Scientific Workflows

USEFUL FEATURES FOR SCIENTIFIC WORKFLOWS

Desirements for Scientific Workflows

- Hierarchical (nested) workflows
  - Layered abstractions, modular
- Vectorization / iterations
  - Processing sequence of data flow
  - Analogous to vector operations in Matlab and IDL
- Orchestrating distributed services
  - SOAP, REST, OGC services, etc.
- Runtime WSDL and WADL introspection
- Integrated service registries discovery
  - UDDI, ServiceCasting, etc.

Desirements for Scientific Workflows

- Bean shell components
  - "Shim" services
- Collaboration
  - e-Science
- Semantics
- Provenance
  - Traceability
- Reproducibility of results
  - "Climate-gate"
- Workflow instance callable as API

CONCLUSION

So Why So Many?

- Domain-specific workflow features
  - Data flow for Bioinformatics and Earth science
  - Activity flow for business process management
- Fragmented "market"
  - Many derivatives of BPEL engines
  - Many custom adaptations
- Popular workflow engines in each domain-specific field. Examples:
  - Kepler (ecology, Ptolemy II)
  - Taverna (biology)
  - VisTrails (visualization)
  - ModelCenter (DOE)

Where We Are At / Heading To?

- Mixed results with workflow-based visual programming
- Asynchronous services
  - WS-Eventing and WS-Messaging
  - "Jobification" of SOAP/REST service interfaces
- Integrating with other services
  - ServiceCasting, DataCasting, Federated OpenSearch, etc.
- Collaborative workflows
  - myExperiment (Taverna)
  - Drupal-based Talkoot collaboration workflow (Rahul and Chris)
- Semantic service and datatype ontology
  - ESIP and ESDSWG activity
- Automated workflow discovery, execution, composition and interoperation
  - OWL-Services, WS-BPEL (legacy OASIS BPEL4WS)
- Provenance, semantic web services, and Proof Markup Language (PML)