Workflow Management

John Watt

http://csperkins.org/teaching/2004-2005/gc5

UNIVERSITY
of
GLASGOW

Overview

- Introduction to Workflows
- Construction and Enactment
- e-Science Workflows
- Critical Issues
- Geodise Project
- Taverna Project
- Summary

Workflow

- Definition:
  - “The set of relationships between all the activities in a project, from start to finish. Activities are related by different types of trigger relation. Activities may be triggered by external events or by other activities.”
    - “The Free Online Dictionary of Computing (FOLDOC) “

Workflow

- Definition:
  - "The automation of a business process, in whole or in part, during which information or tasks are passed from one participant to another for action, according to a set of procedural rules."
    - Workflow Management Coalition (WfMC)

- Workflow is an established methodology for business process management.

e-Science Workflow

- Can adapt this definition for e-Science:
  - ‘business process’ ⇒ ‘scientific process’
  - ‘participants’ ⇒ ‘compute or data oriented resources’
  - ‘information or tasks’ ⇒ ‘data flow or control flow’

- Participants may be geographically distributed.
- Data and control flows may span organisational boundaries.
  - Workflow well suited for describing e-Science applications and activities.

e-Science Workflow

- Workflows allow the e-Scientist to describe and enact their experimental processes in a structured, repeatable and verifiable way.
  - From MyGrid project website
    - Development of a simple workflow language and toolset in collaboration with the European Bioinformatics Institute and the Human Genome Mapping Project

- e-Science Workflow is a very new field

Workflow and Grids

- Workflow is a critical part of the emerging Grid
  - Captures the linkage of constituent services together in a hierarchical fashion to build larger composite services
  - Encompasses
    - "Programming the Grid"
    - "Service Orchestration"
    - "Service or Process Coordination"
    - "Service Conversation"
    - "Web or Grid Scripting"
    - "Application Integration"
      - And many many more….!! (software bus)

Workflow Projects

- Manual composition
  - Triana, BPWS4J, Self-serve
    - Not scalable, user requires low-level knowledge
- Semi-automated
  - Cardoso, Sheth, GeoDISE (myGrid)
    - User still needs to select services required
- Automated (uses AI technology – Semantic Grid)
  - SHOP2, Pegasus-ISI, IRS-II
    - Most systems make simplistic assumptions
    - Difficult to reuse (static environment)

Important Aspects

- Representation and language
- User Environments or Workflow IDE
- Translation or compilation
- Execution and runtime support

Grid Workflow Approaches

- Inherent model
  - Workflow is defined inside the software components
  - e.g. MPI, CORBA, Cactus

- External model
  - Workflow is defined on top of software components
  - Complete view of workflow
  - e.g. scripts or graphs

Workflow Representations

- Graph based
  - Nodes of graph represent services
  - Directed edges represent data flow or control flow
- XML based
  - Conforms to the schema of some workflow definition language

- Workflow is inherently hierarchical
  - Workflow of more than one node may be represented by one workflow within other workflows

Workflow Implementations

- Scripts
  - GridAnt, JPython (XCAT)
- Combined scripts + graphs
  - WSFL, XLANG, BPEL4WS, UNICORE, GSFL
- Graphs
  - DAG: Condor DAGman, Symphony,
  - Petri net: GJobDL

Possible Standards

- Represent workflow by some XML-based workflow definition language
  - Business Process Execution Language for Web Services (BPEL4WS) – IBM and Microsoft
  - XML Process Definition Language (XPDL) – WfMC
    - Open question as to whether we can use e-Business workflow description languages for e-Science (must support programming abstractions such as conditional and loop constructs)
- e-Science
  - Service Workflow Language (SWFL) – Cardiff
  - Grid Service Flow Language (GFSL) - Argonne

Workflow Construction

- Use graphical representation to construct XML workflow description document
  - Visual service composition environment (VSCE)
  - Links services comprising a workflow
  - Provides mechanism for service discovery to populate a virtual service repository
  - Interfaces repository services by visual connection of data and control links on a ‘canvas’
  - ‘plug-and-play’ capability
    - Requires that only services with compatible interfaces may be connected

Workflow Construction

- Service interfaces must be syntactically AND semantically compatible – a challenge!
  - Syntactic: requires data types of data items flowing into a target service to be the same as the data types of the data output by the source service
    - Common data types defined in some XML namespace that everyone should use
  - Semantic: requires data items to have similar meaning when they may have different names
    - Need a mechanism which determines if complex data types defined in different XML namespaces have the same meaning
      - e.g. compatible units for non-dimensionless quantities

Workflow Construction

- General problem of determining and comparing the behaviours of interacting services
  - Use ontologies and agent-based mediation to assess semantic compatibility
- VCSE accesses service descriptions given in WSDL (for example) to determine the syntax of a service interface
  - Additional metadata is needed in the service description to describe the service semantics and provenance

Workflow Construction

- More complications!
- Services in a workflow may not be bound to specific service implementations at runtime
  - Services may only be bound dynamically at runtime
  - May not be compatible
  - Happens when workflows constructed on semantics, not on interfaces
- We could ‘compile and link’ workflows in the VSCE to check the interacting services are compatible and discoverable
  - Doesn’t guarantee future service availability

Workflow Enactment

- Constructed workflow submitted to a workflow engine for execution
  - Converts XML document into an executable form
  - Discovers and schedules services
  - Central tasks of any service-oriented architecture for Grid Computing
  - Should be able to exploit parallelism
    - Grid Runtime Environment

Current state of play

- Require a scientific workflow engine
  - Compatible with different runtime environments
    - Enterprise JavaBeans
    - Scientific JavaBeans
  - Need to integrate ontology support
- Require a scientific workflow language
  - Identify differences between e-Business and e-Science needs – can we use BPEL4WS?
- Need parallel execution and dynamic discovery of services

User Requirements

- Reflect the modelling paradigm of the scientist
  - Varies across disciplines
  - Maintain appropriate levels of abstraction
  - “Work in MY problem solving environment, so I don’t have to change the way I work”
- Different users, different environments
  - Creators, users, auditors, validators…
- Simple to use, with intuitive creation, deployment, execution and debugging environments

e-Science Workflow Lifecycles

- Incrementally exploratory prototypes
  - Got the data, publish ASAP!!
- Large scale production
  - Got the idea, get the data for many experiments, communities, collaborations
- Migration
  - Capture of prototype for non-interactive reply at a later date
- Different parts of lifecycle
  - Interaction of many different users, environments…

User Interactions

- Creation and Discovery
  - Drag ‘n drop, by example, plagiarism
- Collaborative multi-user interaction in creation
  - Reuse workflows with different data
  - Compose workflows from different disciplines
- Single user interaction with workflow execution
  - Choice between paths of execution in certain states
  - Parameter modification mid-run
- Collaborative multi-user interaction during execution???

Scientific Workflow Characteristics

- Very large amounts of data
  - Files, streams, database queries
  - GridFTP, http, ftp, sockets
  - Sometimes the computation needs to be moved to the data
- Data model and types
  - Metadata and provenance
- Driven by
  - Scientific questions, outcomes, bravado
  - More creators than users in science?

Critical Issues

- Managing complex workflows
  - Parameter and constraint management
  - Workflow Tools
- Grid Job IDs
- Security
- Engineering Workflows
  - Geodise

Managing Complex Workflows

- Parameter and constraint Management Problem
  - When workflow nodes contain many attributes or attributes that are related to attributes in other nodes
    - HEP Use Case: Consistent calibration sets, fudge factors, simulation input parameters
  - When workflow nodes' parameters and constraints vary with execution or logical context
    - HEP Use Case: Physics groups, parameters coming from gurus, different kinds of infrastructure across the VO
    - Dynamism in workflow due to execution environment can be modelled as dynamism in the constraints from site to site

Managing Complex Workflows

- Workflow building tools
  - Should address Parameter and Constraint Management Problem
    - Factor workflows away from constraint and parameter specification
    - Constraint satisfaction implies another partial order in addition to control flow and data flow
  - Provide services at runtime to handle dynamism by resolving late constraints
    - e.g. The RunJob project (projects.fnal.gov/runjob)

Workflow Job IDs

- Grid jobs (or workflows) have many different stages, e.g. Input data staging, Authentication, Scheduling, Running and Returning results
  - Each of these stages uses one or more Grid services, which may be servicing other Grid workflows, or parallel branches of this workflow
    - Time correlation of logs may be ambiguous
  - To track the job, we need two things to be a standard part of every Grid Service
    - 1. A “Grid Job ID” metadata element
    - 2. Grid Service lifecycle monitoring: log an event at START of service, END of service and include Grid Job ID in these events

Security

- Requirements for Workflow systems often include security
  - Data access controls, constraints, provenance
- Issues:
  - Need to distinguish between functionality & security guarantees
    - Workflow systems are often interposed between users, data and services without considering the trust responsibilities that this design imposes on planning and enactment systems
  - Workflows are process or data centric
    - They do not always naturally map to user-centric system security policies

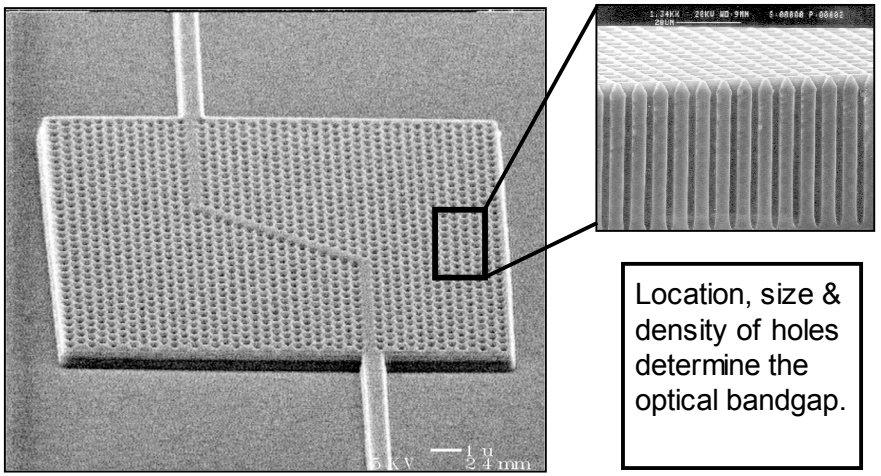
Security

- Issues (cont.)
  - Planning and enactment are complex/rich processes
    - It is poor security design to trust a complex mechanism
- Need a systems design approach that separates enactment and protection by refactoring the protection requirement away from planning and enactment, and into the distributed system. e.g
  - Pass data by reference
    - Users access data via normal system access control, rather than via workflow
  - Protect services at the point they are invoked
    - Rather than trust the correctness of the planning & enactment process

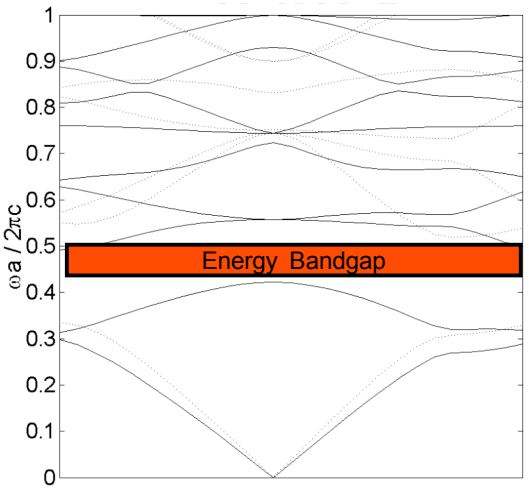
Geodise Project: Engineering Workflows

- Scenario: design optimisation
  - Model device, discretize, solve, postprocess, optimise
- Scripting approach
  - Flexibility & High Level functionality
  - Quick application development
  - Extend user's existing PSE e.g. Matlab, Python
    - Is execution/enactment engine too
- Favourites:
  - Create, retrieve, cut 'n' shut (re-use)
  - Configure, execute, monitor (bring grid to user)
  - Share, steer, dynamically modify (semantic support)

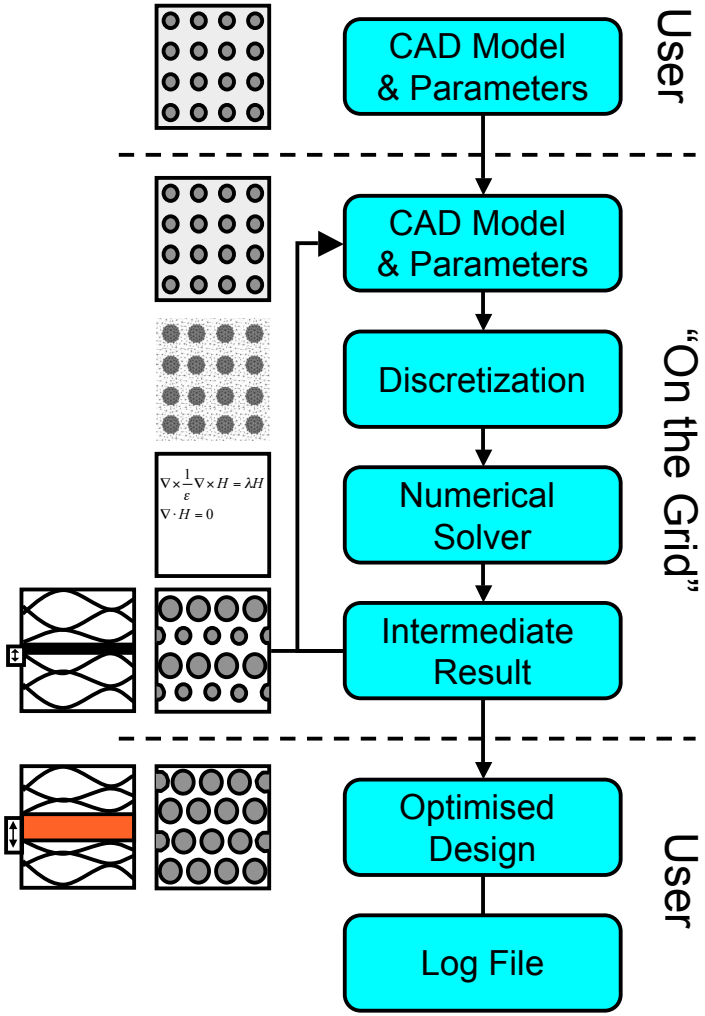
Geodise – Photonic Crystal Optimisation



Location, size & density of holes determine the optical bandgap.



The aim of this design optimisation process is to find a configuration which maximises this bandgap and minimizes energy loss.



Geodise: Grid enabled Matlab scripting

- Motivations
  - Flexible, transparent access to computational res.
    - Easy to use for engineers (and in widespread use)
  - Matlab chosen as hosting environment
    - Extends user's existing PSE, high level func., quick development
  - Computational resources exposed in the form of Matlab functions
    - Job submission to Globus server using Java CoG
    - Job submission to Condor pool via Web Services interface
  - Integration of CAD, Mesh generation via the use of intermediate data format, often package-neutral

e-Science Research Projects

- myGrid (SeSC)
  - Workflow Enactment Engine
- WEGS (NEReSC)
  - Workflow Enactment Grid Service
- SWFL (WeSC)
  - Service Workflow Language
- IT Innovation's Workflow Enactor

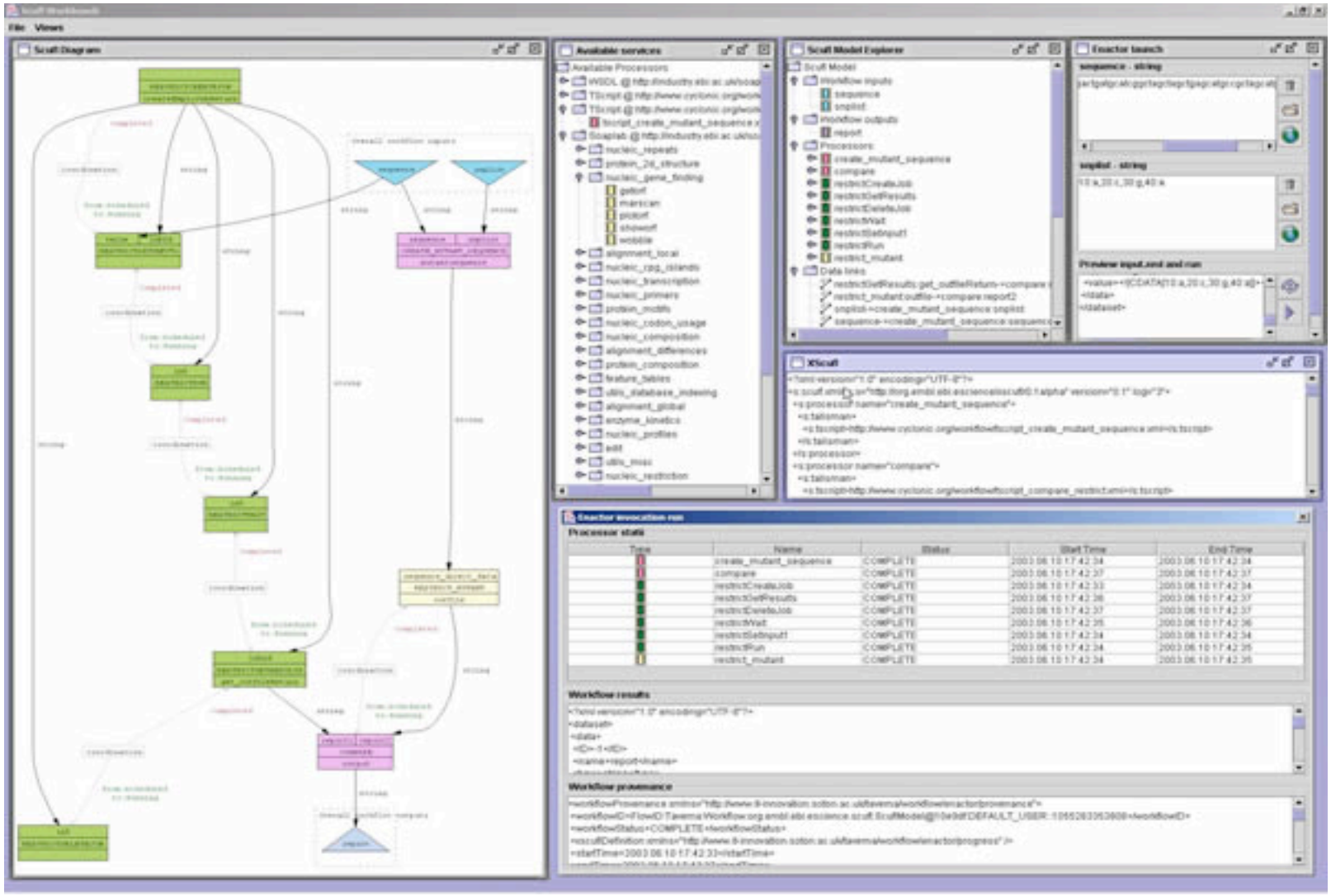
The Taverna Project

- Aims to provide a language and software tools to facilitate easy use of workflow and distributed compute technology within the e-Science community.
  - Component of the myGrid project
  - Available freely under GNU Lesser General Public License
  - Project aims to provide a workflow-based approach to the specification and execution of ad-hoc in-silico experiments using bioinformatics resources.

Taverna

- Consists of a workflow workbench to graphically build, edit and browse workflows.
  - Includes easy import of external web service and workflow definitions.
  - Can submit workflows directly to the workflow enactor (freefluo) for execution
- Freefluo coordinates execution of parallel and sequential activities in the workflow
  - Supports data iteration and nested workflows
  - Can invoke arbitrary web services and specific bioinformatics services (Talisman, Soaplab)

Taverna workbench



Summary

- Workflows are good for describing e-Science activities
  - Geographically distributed, cross-organisational
- Workflows link discrete Grid Services into larger composite services
  - Semantic/syntactic compatibility, parameter constraint
- Workflows ideally constructed in a VSCE
  - Workflow workbench, drag ‘n’ drop, dynamic creation
  - Establish a standard language for workflows
    - All open questions in a new field