

On Development of Workflow Management Service for Distributed Computations

Igor Lazarev, Oleg Sukhoroslov

Centre for Grid Technologies and Distributed Computing of ISA RAS

lazarev.igor@gmail.com

Workflow

A formal representation (model) of a process:

- Elementary tasks forming the process
- Entities which perform the tasks
- Dependencies between tasks (control and data flows)
- External events

Outline

- Workflow terminology and background
- Specifics of distributed computing workflows
- Workflow description approaches
- Experience of building IARnet WfMS
- Applications
- Further development

Workflow Management System

A system allowing the user to create a workflow, execute it and control the execution. A WfMS is a set of program components for storing and interpreting workflow representations, creating and controlling running workflow instances, as well as organizing their interaction with process participants and external applications.

Distributed Computational Workflows

- Today more and more computational resources become available as services via the Internet.
- Users need high-level tools to access these services and combine them for solving their problems
- Service-oriented architecture: people compose services into new services
- Visual workflow composition
- Web 2.0, mash-ups, Yahoo! Pipes

Specifics Of Distributed Computational Workflows

- Involve complex computational resources
- The amount of resources not known a priori
- Rapidly changing distributed environment
- Huge amounts of data
- Large number of identical tasks with different parameters
- The need for monitoring process execution and steering it interactively
- The subprocess hierarchy, being created on demand

IARnet Toolkit

- Software toolkit for integration of information-algorithmic resources in the global network for solving applied problems
- Oriented on solving problems which can be decomposed into several well-known subproblems with existing solvers
- Implements a high-level programming model, suitable for a wide range of application programmers

Service-oriented approach

- Representing resources via remotely accessible services, supplied with metadata and discovered by dedicated search engines
- IARnet
 - Means for rapid service creation
 - API for creating applications using services
 - Service registration and search
 - Service composition via the WfMS

General Workflow Representation Approaches

- Script languages
- Graphs
 - Directed Acyclic Graphs (DAG)
 - Petri Nets
- Hybrid approaches
- Data flow oriented models

Script Languages

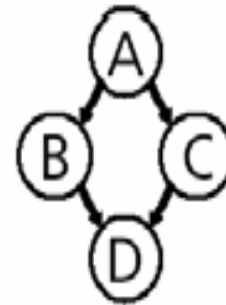
- Convenient for users familiar with programming languages
- Not enough clear and effective for inexperienced users
- Scanty “vocabulary”, we need to add new language elements to describe computational workflows
- Grid Ant, Karajan

Graph Representation

- Visual workflow composition - convenient for inexperienced users
- Hard to describe control elements
- Hard to describe and view large complex graphs - the need for hierarchical structure.
- Two types of graphs:
 - Directed Acyclic Graphs (DAG)
 - Petri Nets

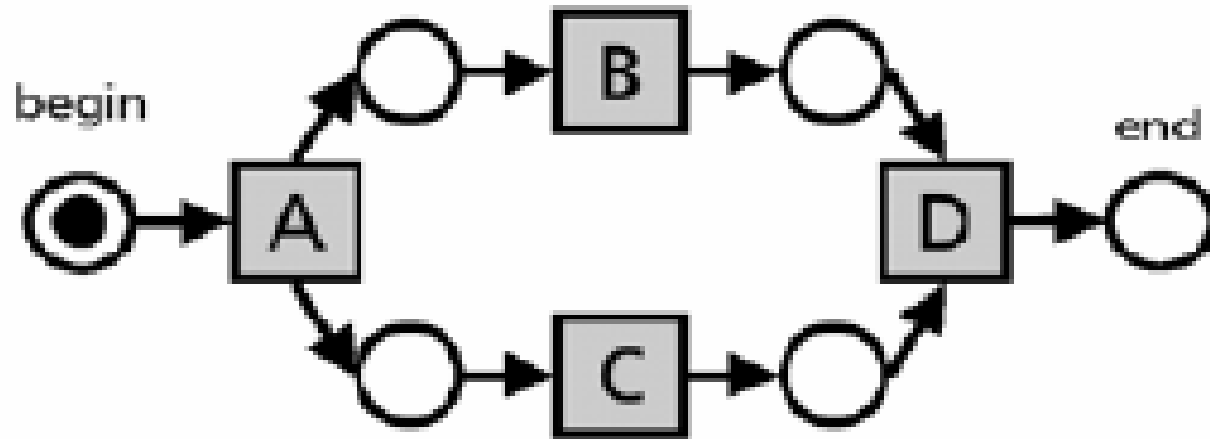
Directed Acyclic Graphs

PARENT A CHILD B C
PARENT B C CHILD D



- Easy to implement, non-complex structure
- Not all types of workflows can be described
- Only the behavior of the process can be described, but not its state
- Condor DAG, Symphony, Cactus, UNICORE, jOpera

Petri Nets



- Graphical language
- Formal description
- Expressiveness
- Thoroughly examined properties
- Convenience of analysis
- Renew
- Petri Net Kernel
- jFern
- YAWL

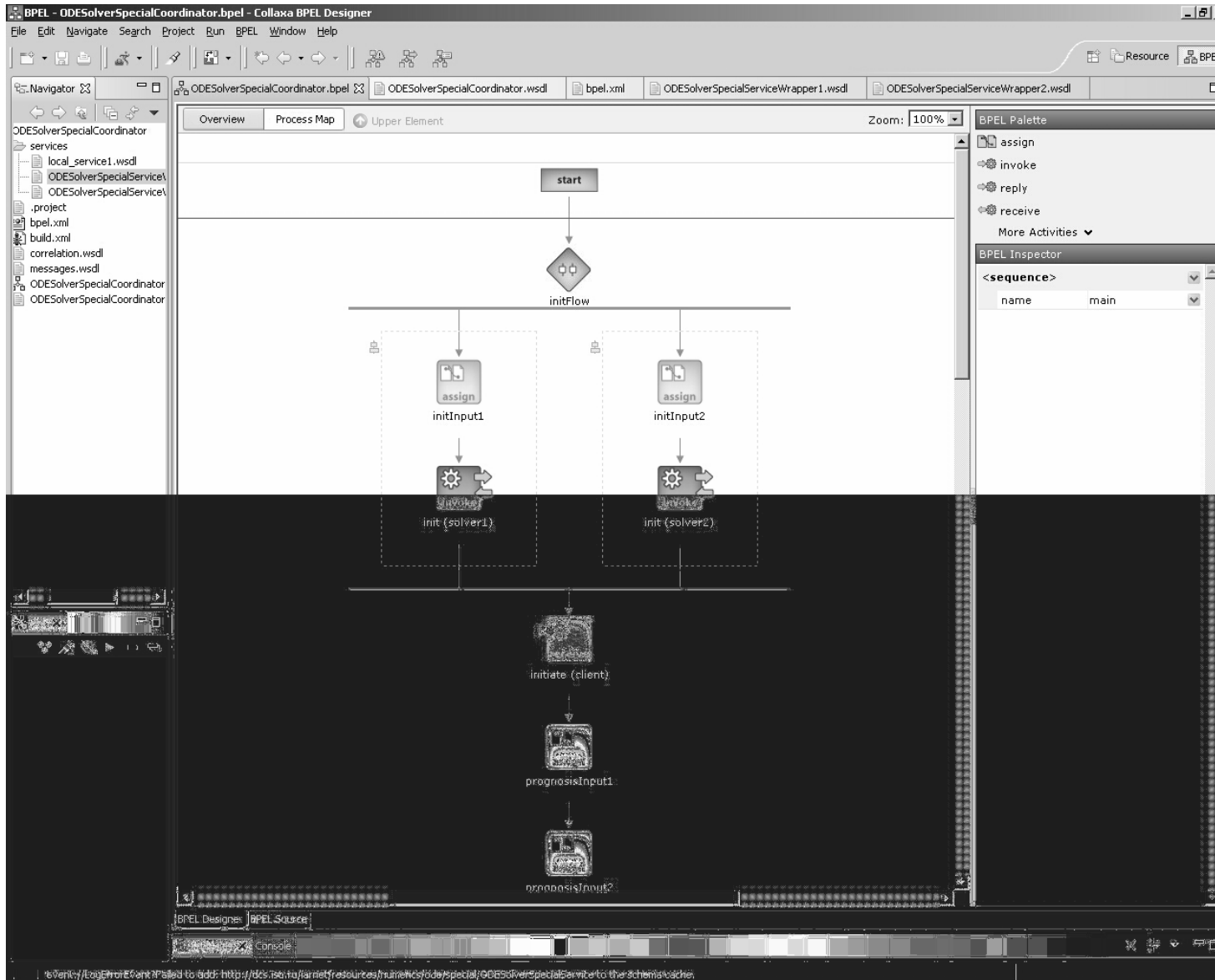
Hybrid Approaches

- Approaches combining graph and script workflow representations
- Examples
 - BPEL
 - JBoss jBPM

Data Flow Oriented Models

- In most scientific applications control flows and data flows are coincident
- Workflow composition is reduced to connecting elementary data processing blocks into a net
- Kepler, Triana, Taverna

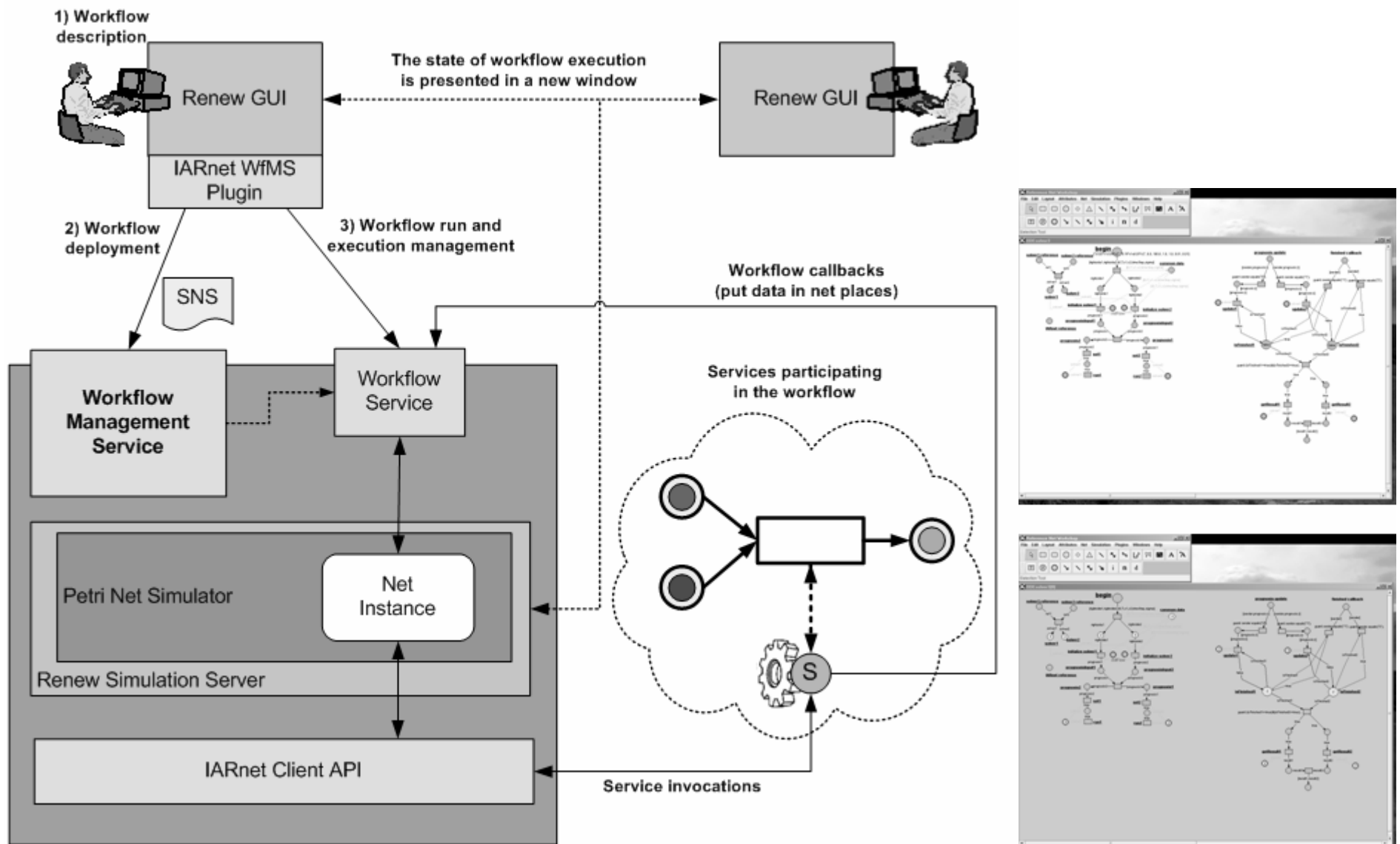
First IARnet WfMS Prototype (Collaxa BPEL)



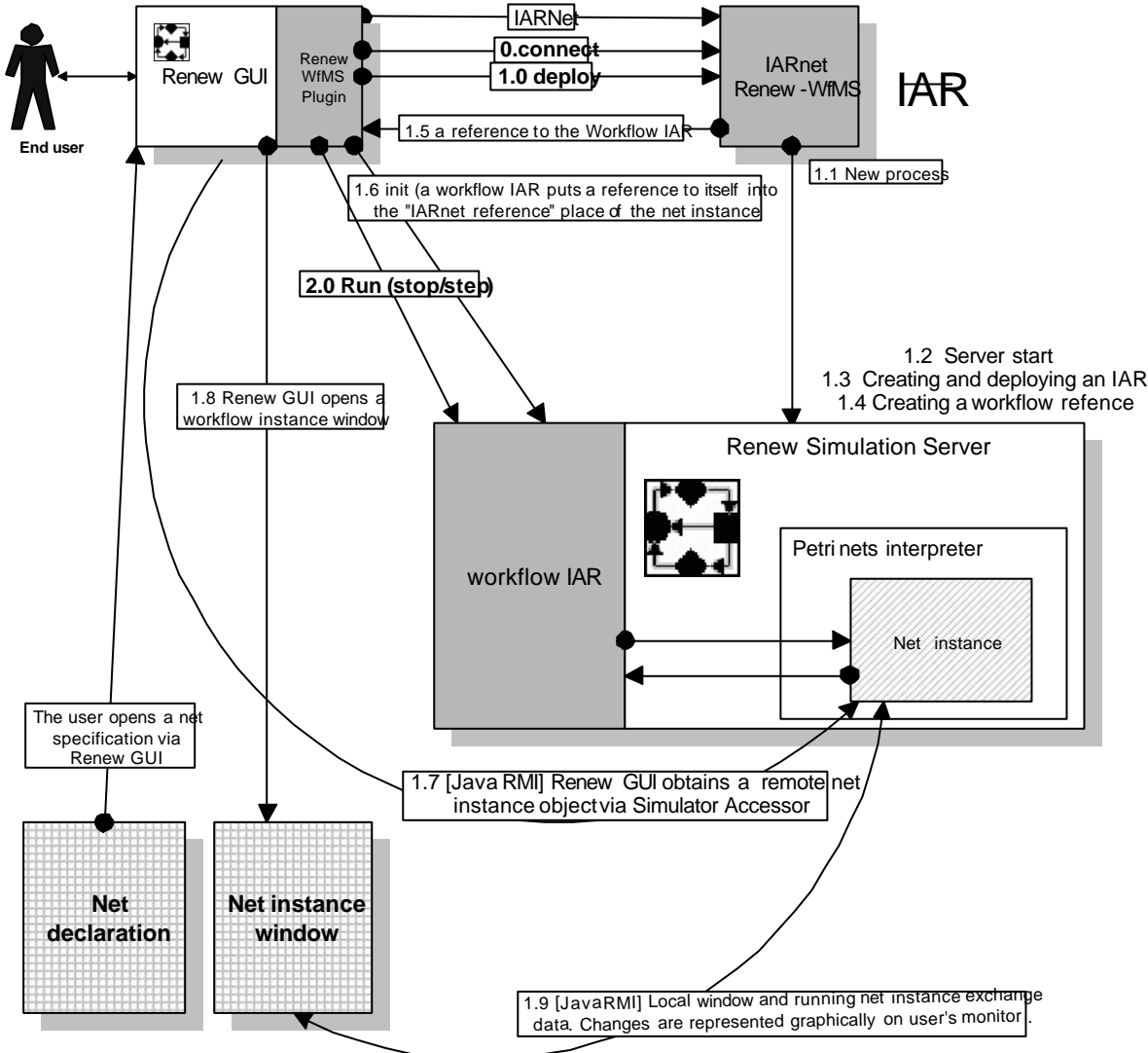
Second IARnet WfMS Prototype

- Based on the Renew toolkit
- Uses Coloured Petri Nets for workflow representation (tokens are Java objects)
- Showed the ability of the approach to solve such problems in principle
- Convenient analysis, expressiveness
- Complex and unclear workflow descriptions

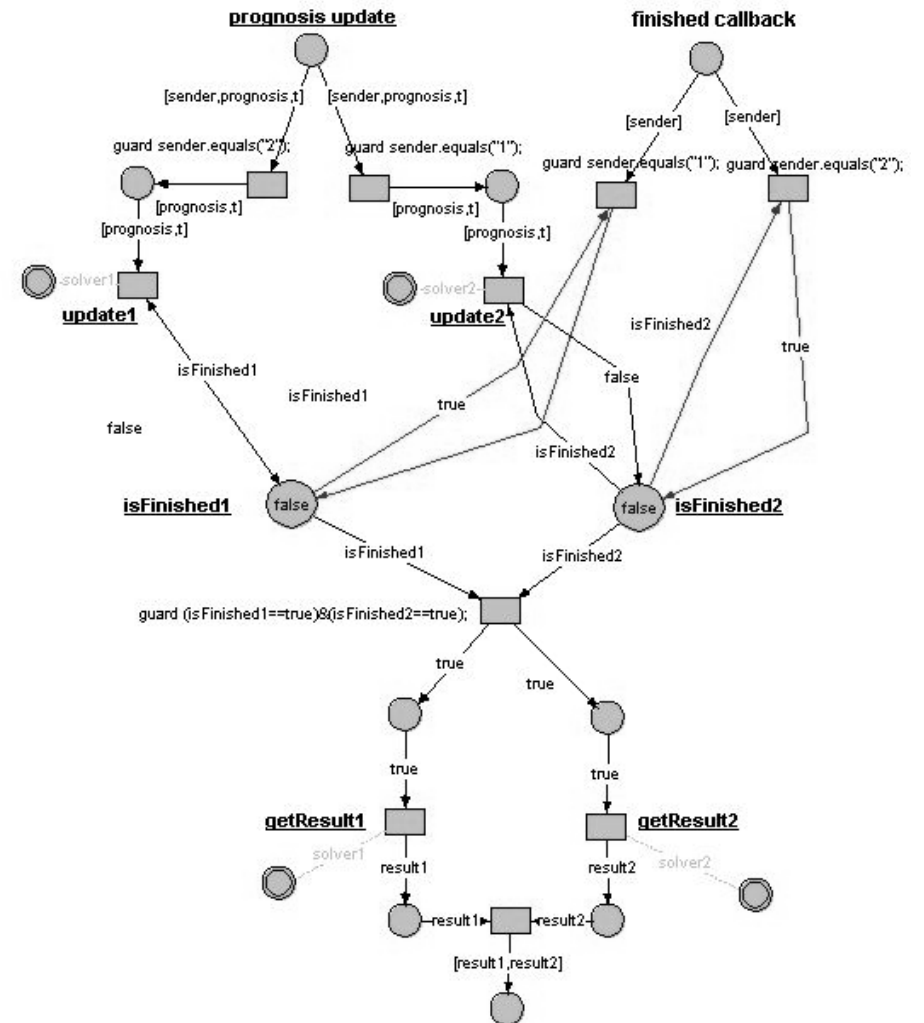
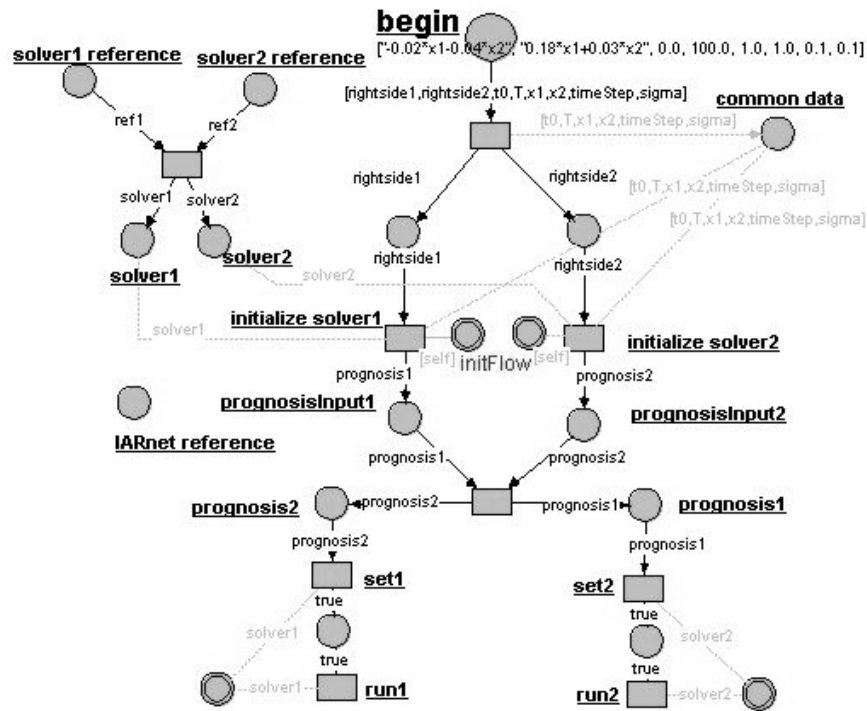
Second WfMS Prototype Architecture



Second WfMS Prototype Architecture



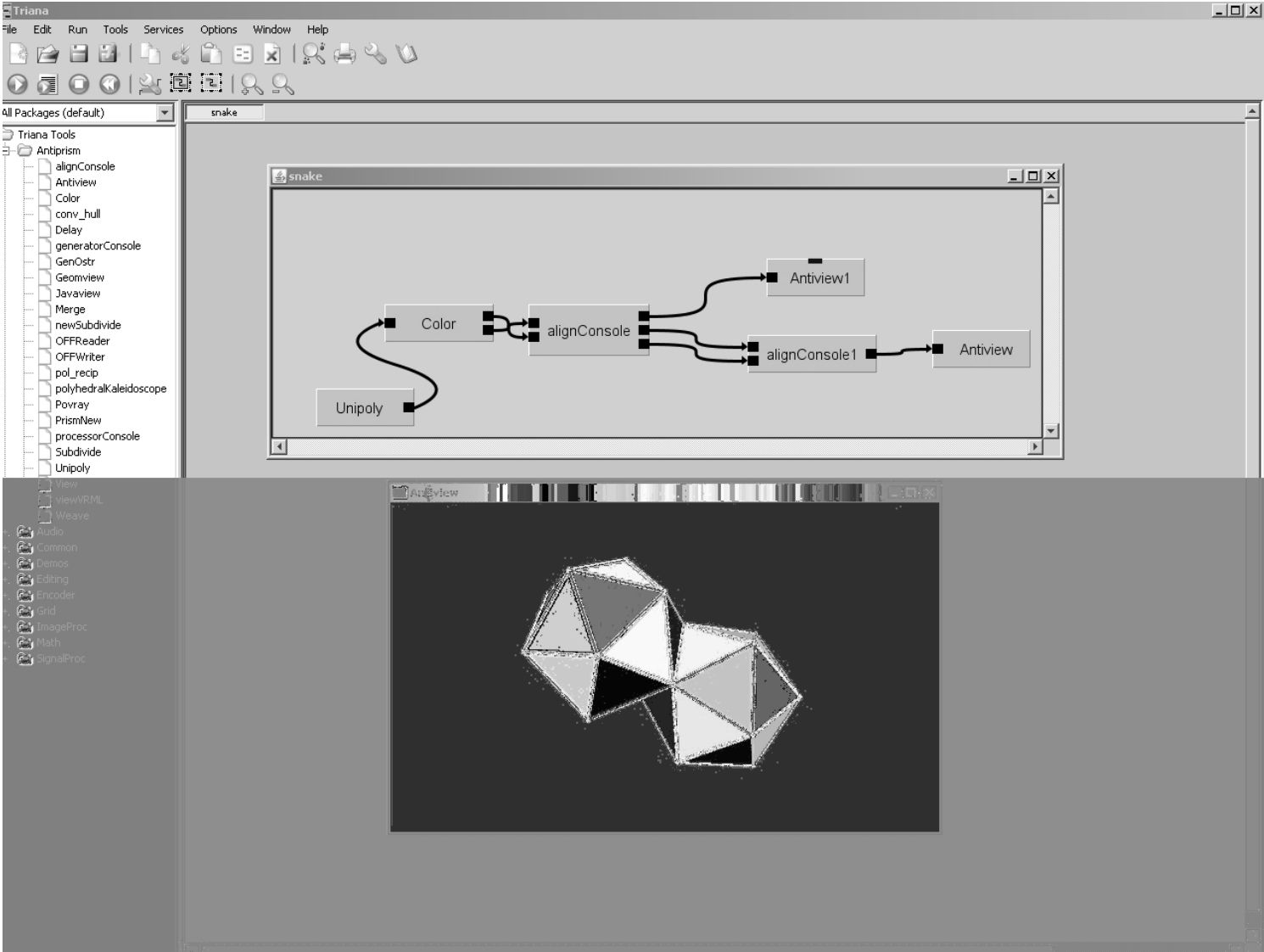
Test Workflow



Current WfMS Prototype

- Data flow oriented
- Based on Triana
- Easy and intuitive workflow composition
- Was used to combine programs from Antiprism toolkit and our own resources represented as REST-services (MathCloud) into workflows

Current WfMS Prototype



Further Development

- Web integration, composing workflows in browser
- Creation of an own workflow runtime
- Visual workflow composition is not a universal programming paradigm



Thank you!