On the Applicability of Workflow Management Systems for the Preservation of Business Processes

Rudolf Mayer, Stefan Pröll, Andreas Rauber
sproell@sba-research.org

University of Technology Vienna, Austria
http://www.ifs.tuwien.ac.at/dp

Secure Business Austria
http://www.sba-research.org

October 2012

Overview

- Introduction/Motivation
- Preserving Complex Systems
- Business Processes and eScience Experiments
- Workflow Management Systems: Taverna, Kepler and Activiti
- Example Workflow
- Validation and Verification
- Comparison
- Conclusions / Future Work

Preserving Complex Systems

- **The TIMBUS Approach**
- Preserve business or eScience processes as complete as possible.

1. Identify components and their properties to preserve
2. Describe context of a process
    1. Software dependencies (external and internal)
    2. Hardware dependencies
    3. Legislative requirements and restrictions
3. Gather metadata
4. Store and maintain
5. Rerun

Business and Science

- Workflows
  - Model processes (scientific, business)
  - Describe information flow between entities
  - Formalize intermediate steps
  - Add structure
  - Abstraction

- Goals
  - Repeatability of experiments
  - Re-executability of business processes
  - Validation and Verification

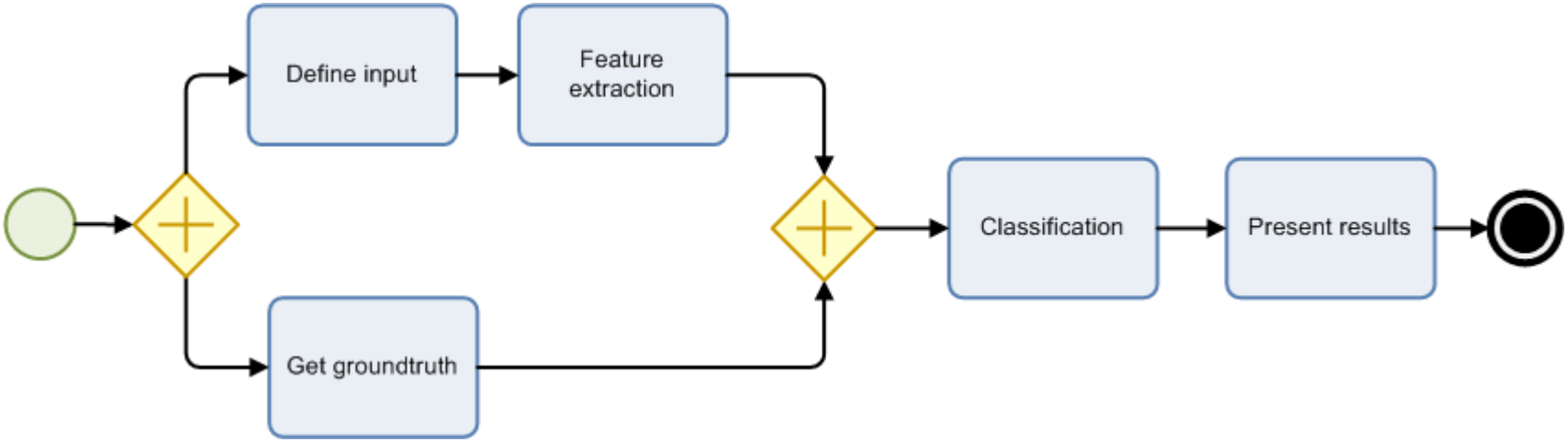
Workflow Management Systems

- Scientific Workflow Management Systems (SWMS) provide an execution environment for workflows
- Orchestrate external services and local tools
- Model the information flow
- Integrate third-party libraries
- Easy to read and get “the big picture”
- Capture metadata about execution

Example – Music Processing

- Case study: Scientific Experiment
- Workflow: classification of music into predefined set of genres
- Learns a machine-learning model from given training data (i.e. data with manually assigned class/genre)
- Predicts genre for previously unseen data
- Useful e.g. for online music store, recommendation services, etc.

Example – Music Processing



- Input: Music (e.g. MP3 format)
- Input: Training data, i.e. music for which the genre is known a-priori
- Output: Classification of Music, e.g. into genres
- Intermediate steps
  - Extract numeric description (features) from music
  - Combine features with ground truth into specific file format

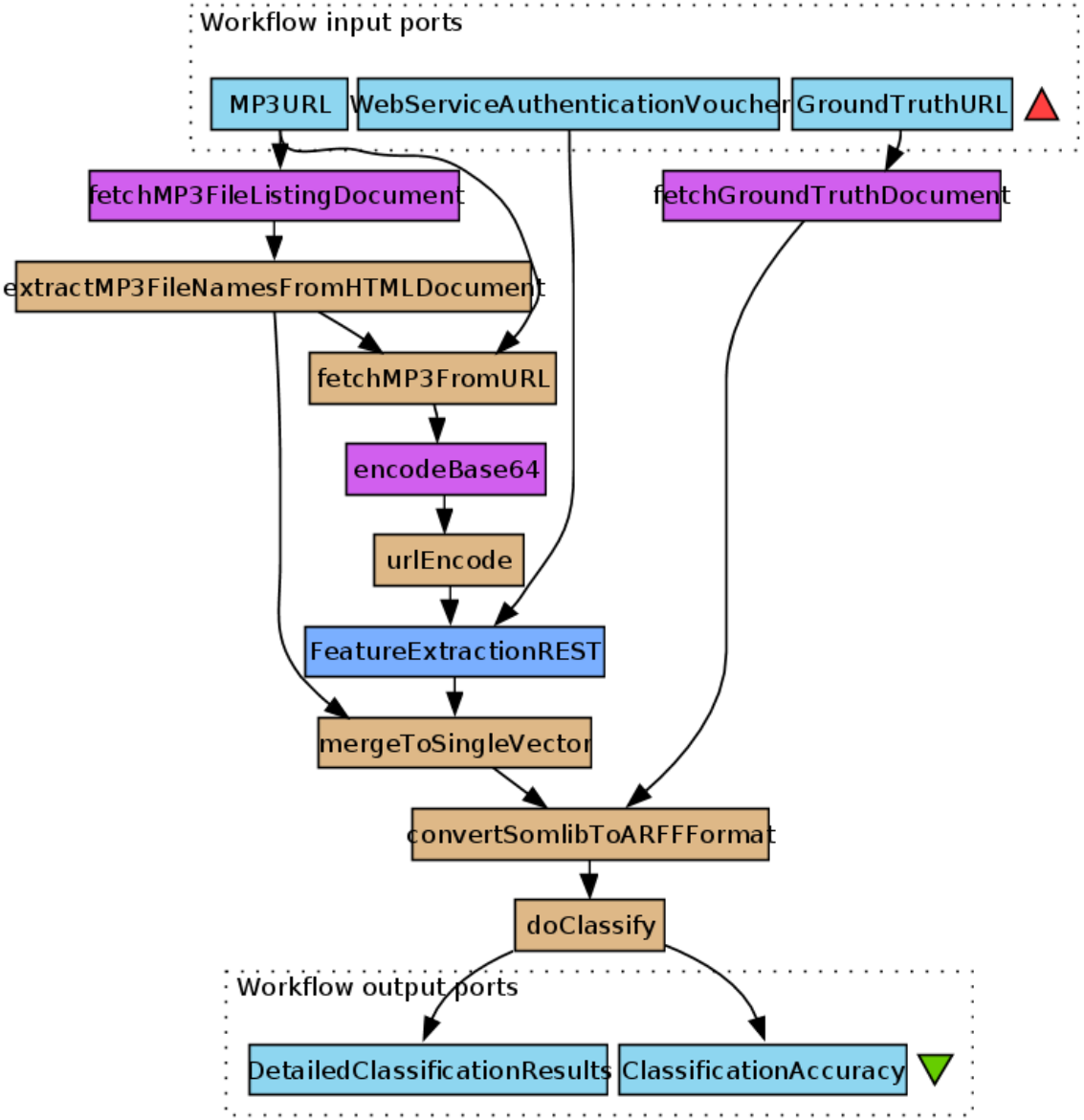
Taverna

- General purpose workflow engine
- Open Source (LGPL), Java
- Data flow oriented
- Local services
  - File operations
  - Format conversions
- Remote services
  - REST

Taverna

- Data flow via ports between services
- Implicit looping and iteration over data structures
- Custom processor implementation
  - Beanshell
  - Java
- Monitoring
  - Events
  - Intermediate results
  - Provenance information

Taverna


Workflow input ports
MP3URL
WebServiceAuthenticationVoucher
GroundTruthURL
fetchMP3FileListingDocument
fetchGroundTruthDocument
extractMP3FileNamesFromHTMLDocument
fetchMP3FromURL
encodeBase64
urlEncode
FeatureExtractionREST
mergeToSingleVector
convertSomlibToARFFFormat
doClassify
Workflow output ports
DetailedClassificationResults
ClassificationAccuracy

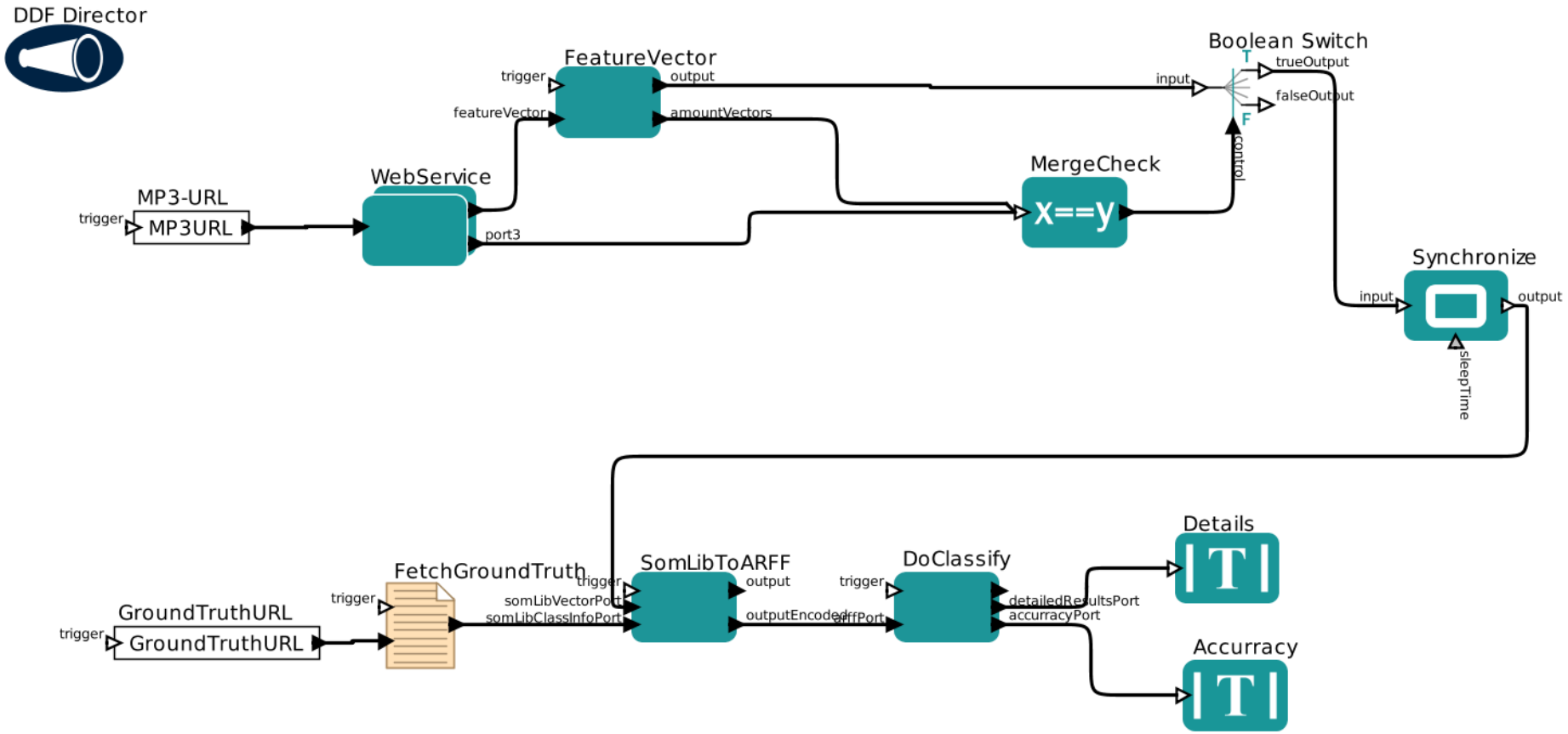
Kepler

- General purpose application suite
- Open source (BSD license), Java
- Based on Ptolemy II
- Data flow oriented
- Actors as processing steps
  - Large library of local actors
  - Fine grained operations
  - Mainly local services
  - Program flow actors

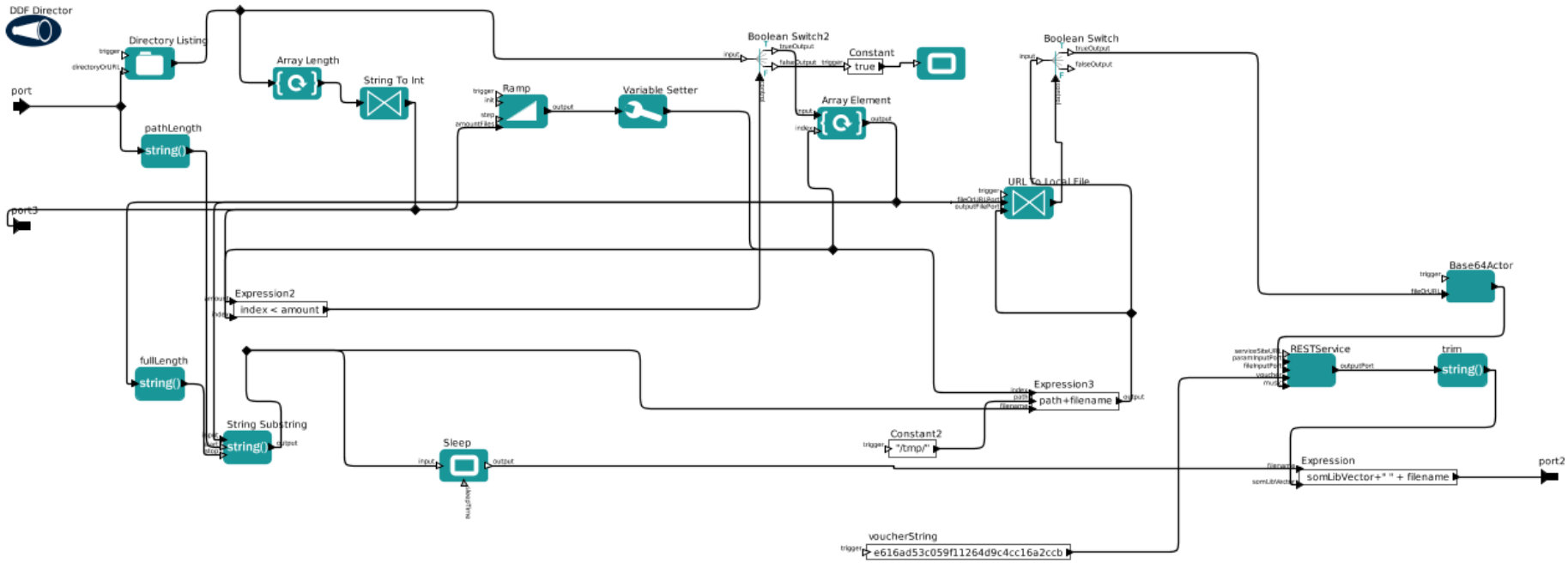
Kepler

- API support
  - Java, custom actors
  - Python
- Workflow orchestration
  - Directors for different purposes
  - Explicit control flow required
- Monitoring
  - Live execution monitor
  - Intermediate results by actors

Kepler


DDF Director
FeatureVector
trigger
output
featureVector
amountVectors
Boolean Switch
input
T
trueOutput
falseOutput
F
control
MergeCheck
x==y
WebService
MP3-URL
trigger
MP3URL
port3
Synchronize
input
output
sleepTime
Details
Accurracy
DoClassify
trigger
detailedResultsPort
accurracyPort
SomLibToARFF
trigger
output
somLibVectorPort
somLibClassInfoPort
outputEncodedArffPort
FetchGroundTruth
trigger
GroundTruthURL
trigger
GroundTruthURL

Kepler


DDF Director
Directory Listing
Array Length
String To Int
Ramp
Variable Setter
Boolean Switch2
Constant
true
Array Element
Boolean Switch
URL To Local File
port
pathLength
string()
port3
Expression2
index < amount
fullLength
string()
String Substring
string()
Sleep
Expression3
path+filename
Constant2
"/tmp/"
Base64Actor
RESTService
trim
string()
Expression
somLibVector+" " + filename
port2
voucherString
e616ad53c059f11264d9c4cc16a2ccb

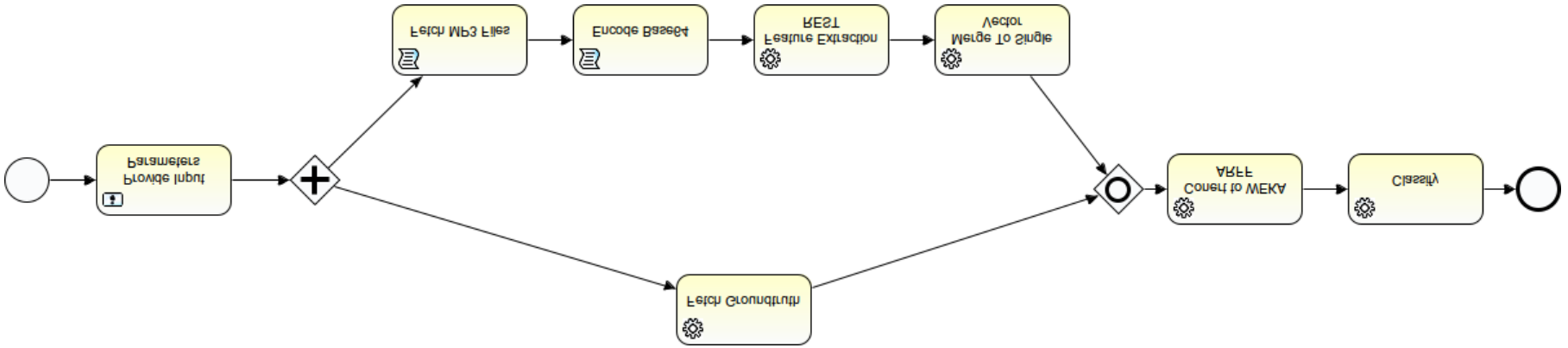
Activiti

- Business Process Management Platform
- Open Source (Apache License), Java
- No direct workflow design via GUI
- BPMN 2.0 editor via Eclipse Plugin
  - All BPMN elements available
  - Core concept: task element
  - Sequence flow connections
- No pre-defined tasks

Activiti

- Tasks implementation
  - Native Java
  - Scripting languages
    - Javacsript
    - Python
    - Ruby
    - Groovy
- User input via forms

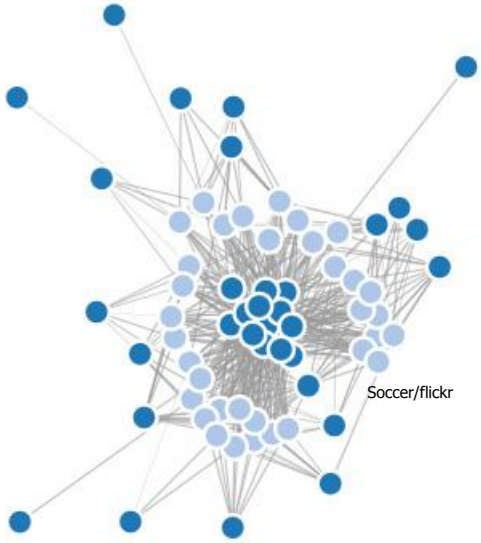
Activiti


Provide Input
Parameters
Fetch MP3 Files
Encode Based4
Feature Extraction
JAST
Merge To Single
Vector
Convert to WEKA
ARFF
Classify
Fetch Groundtruth

Validation and Verification

- Validation is only successful, if and only if all significant properties are equal before and after preservation
- Problem: External services and third party libraries
  - How to detect changes?
- Solution
  - Monitoring necessary
  - Comparison of intermediate results
  - Causal relationships and information flow as significant property
  - Provenance data

Provenance



- Describes data and information flow between computational tasks
- Serves as evidence of execution details
  - Allows to trace the sequence of steps
  - Describes causality between computations
  - Provides formal description of information flow
- Watch service
  - Detect changes or manipulations in computational entities beyond the scope of local processes
  - Verify results

Provenance in Taverna

- Stores all invocations and the sequence of steps in an internal database (Apache Derby)
- Export facilities:
  - Janus: includes data
  - OPM: only invoked steps

Provenance in Kepler

- Dedicated provenance module as seperate actor
- Provenance data stored in HyperSQL database
- Stores details provenance data:
  - Actors
  - Ports
  - Parameters
  - Relations
  - Time stamps
  - Additional informations
- Export facility: OPM

Provenance in Activity

- Process execution history
- H2 database
- No explicit inputs or outputs available, only global state
- No export facility

Comparison

- Activity, Kepler and Taverna can all implement processes by using Java
- Complexity in setup differs vastly
- Activity supports the most programming languages
- Kepler provides the most local services out of the box
- Taverna is the easiest to use
- Only Kepler and Taverna allow provenance export.

Conclusion

- Scientific and business processes need to be preserved
  - Difficult task due to complex service orchestration
- Later verification and re-execution is a crucial task
  - Capturing is not enough, validation is required
- SWMS add a layer of abstraction
  - Monitor execution
  - Evaluate interim and final results
- SWMS formalize processes
  - Documentation of processes
  - Verification possible

The End

Thank you for your attention.

Contact: Stefan Pröll, sproell@sba-research.org