

Web-based Execution of Graphical Workflows

**A Modular Platform for Multifunctional
Scientific Process Automation**

erwin.de.ley@isencia.be

Agenda

- iSencia and Soleil
- Vision for process automation
- Passerelle Platform Architecture
- Some examples
- Conclusion

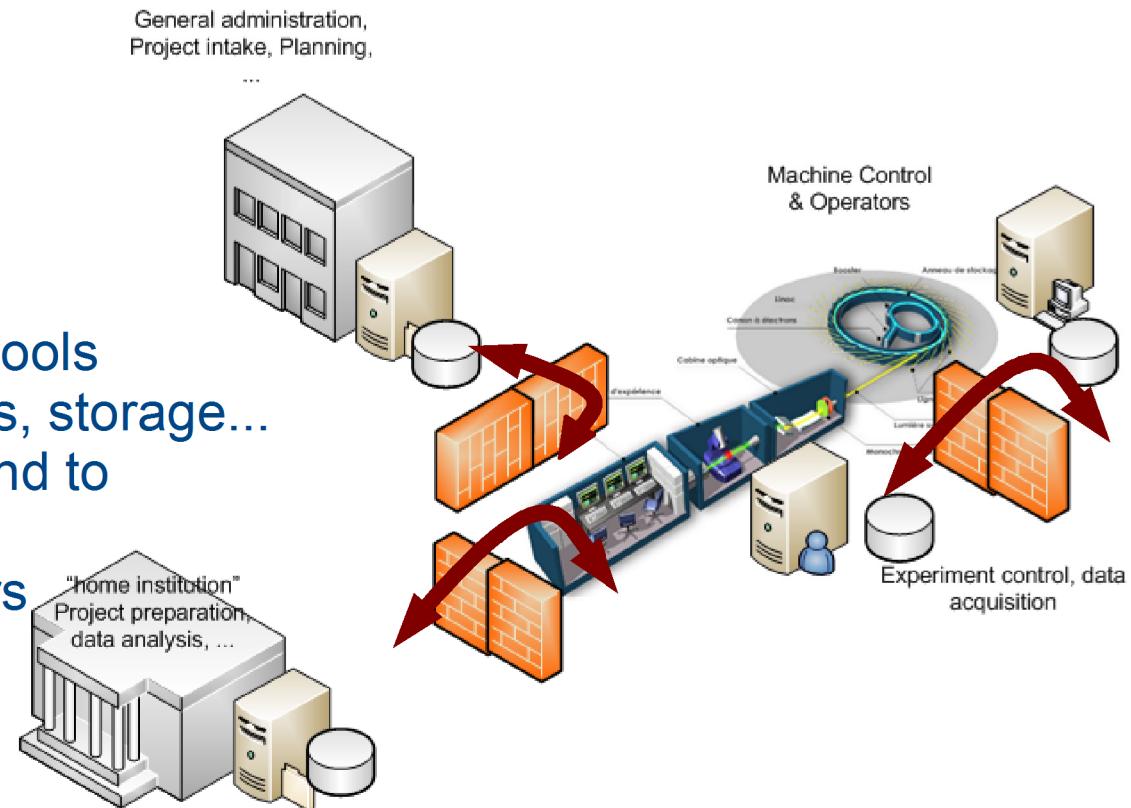
iSencia and Soleil

- iSencia :
 - Software consulting & solutions provider
 - Offices in Ghent, Barcelona, Sweden
 - Specialized in “intelligent process automation” through modular Java Enterprise platform
- at Soleil :
 - 2005 - ...
 - Passerelle - sequencer for acquisition & control
 - 2010 - ...
 - Sunset - improve process support for project intake
 - Passerelle – rules-based monitoring & diagnosis

Vision for process automation

What is (often) :

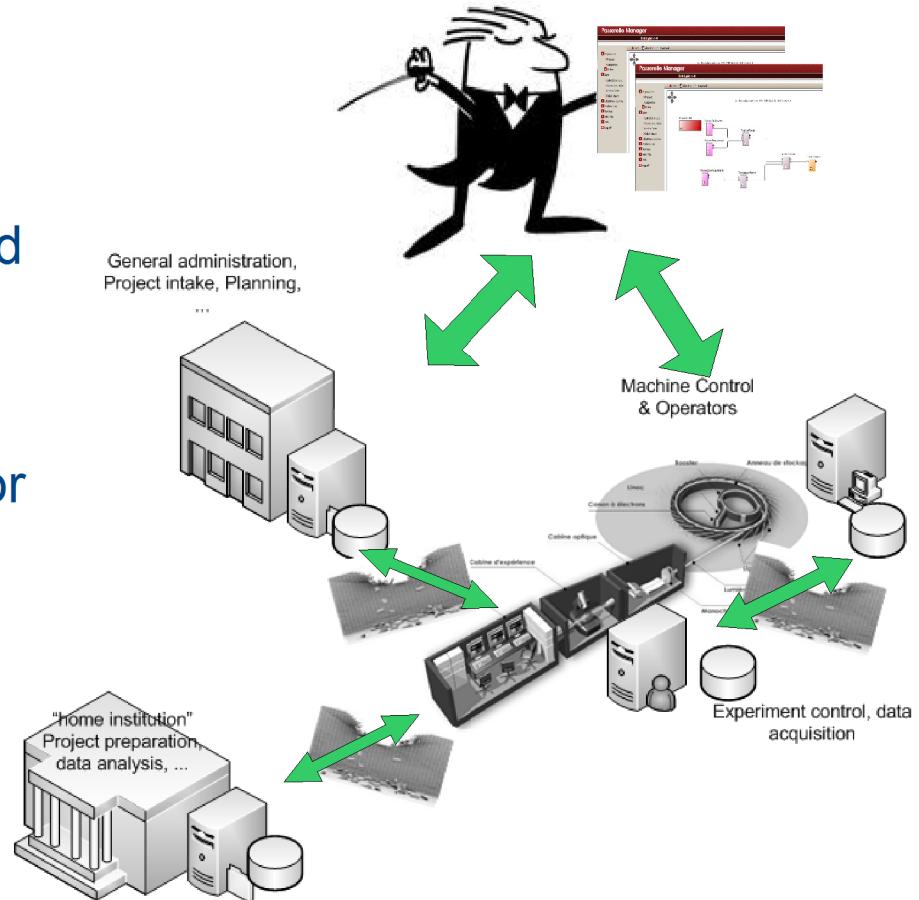
- Small disconnected “islands of automation”
- Disparate organizational structures, procedures & tools
- Non-aligned (data) models, storage...
- Difficult to share, enrich and to protect, guarantee
- Duplicated work and errors
- ...



Vision for process automation

What can be :

- Uniform tool-chain for end-2-end “high-level” automation
- Support for collaboration, transparency, security
- A single consistent repository for models, configurations, traces, datasets, logbooks, ...
- Reusable advanced features



How to get there?

The power of abstraction

In the context of process automation, we talk about :

- **Processes, Tasks**, structured **Results**
- **Actors** execute Tasks & use **Services** when appropriate

Remarkable how many valuable features can be defined, discussed, designed, obtained and freely reused on this limited set of simple abstractions...

The power of an architectural vision

Complex systems need architecture & common vision:

- Ubiquitous domain language used by ALL parties (scientists, IT, operators, management...)
- Structural concepts : modules, layers, components
- Decouple domain concepts from implementation details through clear interfaces (cfr. CDM)
- Consistently discuss/design/document everything matching this vision and language

* Cfr Domain Driven Design by Eric Evans

Modules?

Identifiable subsystems with :

- Defined responsibilities/features/services
- Clear interface(s) matching relevant domain concepts
- Limited and well-managed dependencies

Layers?

Build bridges between low-level techie-stuff and higher levels of domain abstractions

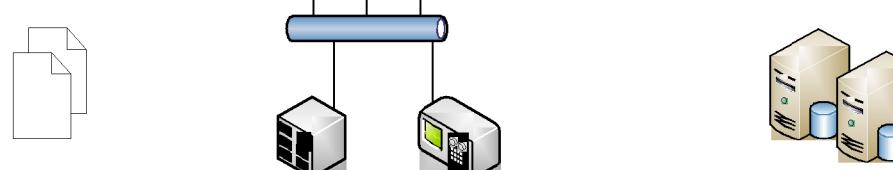
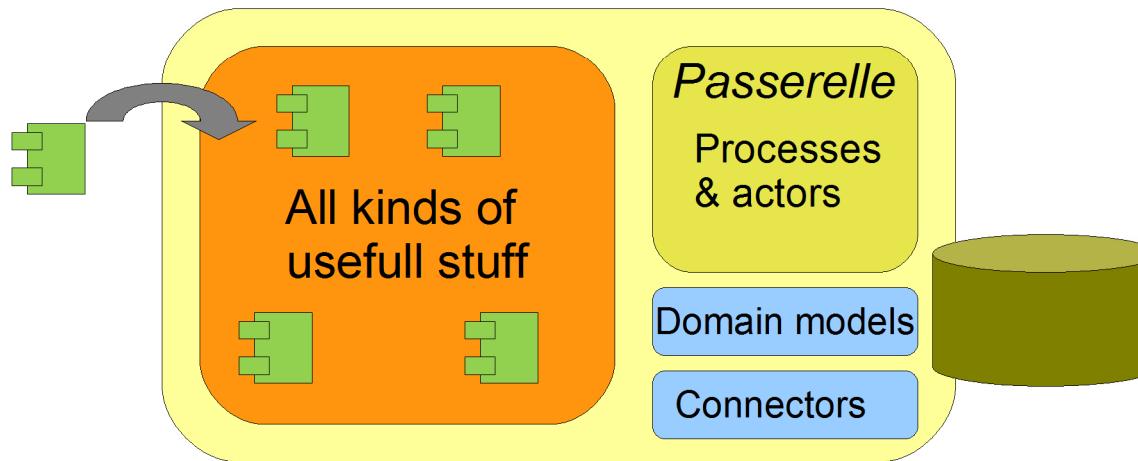
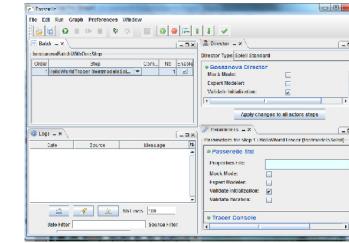
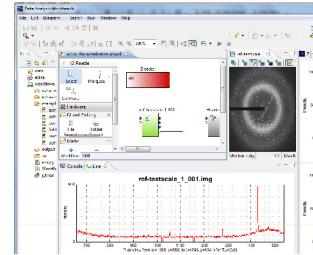
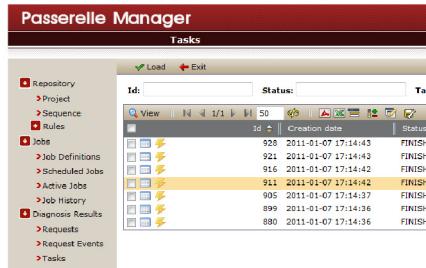
- Device drivers for low-level control, but not everything is a device...
- Higher-level abstractions can take advantage of high-level subsystems like rule systems, web frameworks etc.

The Passerelle platform

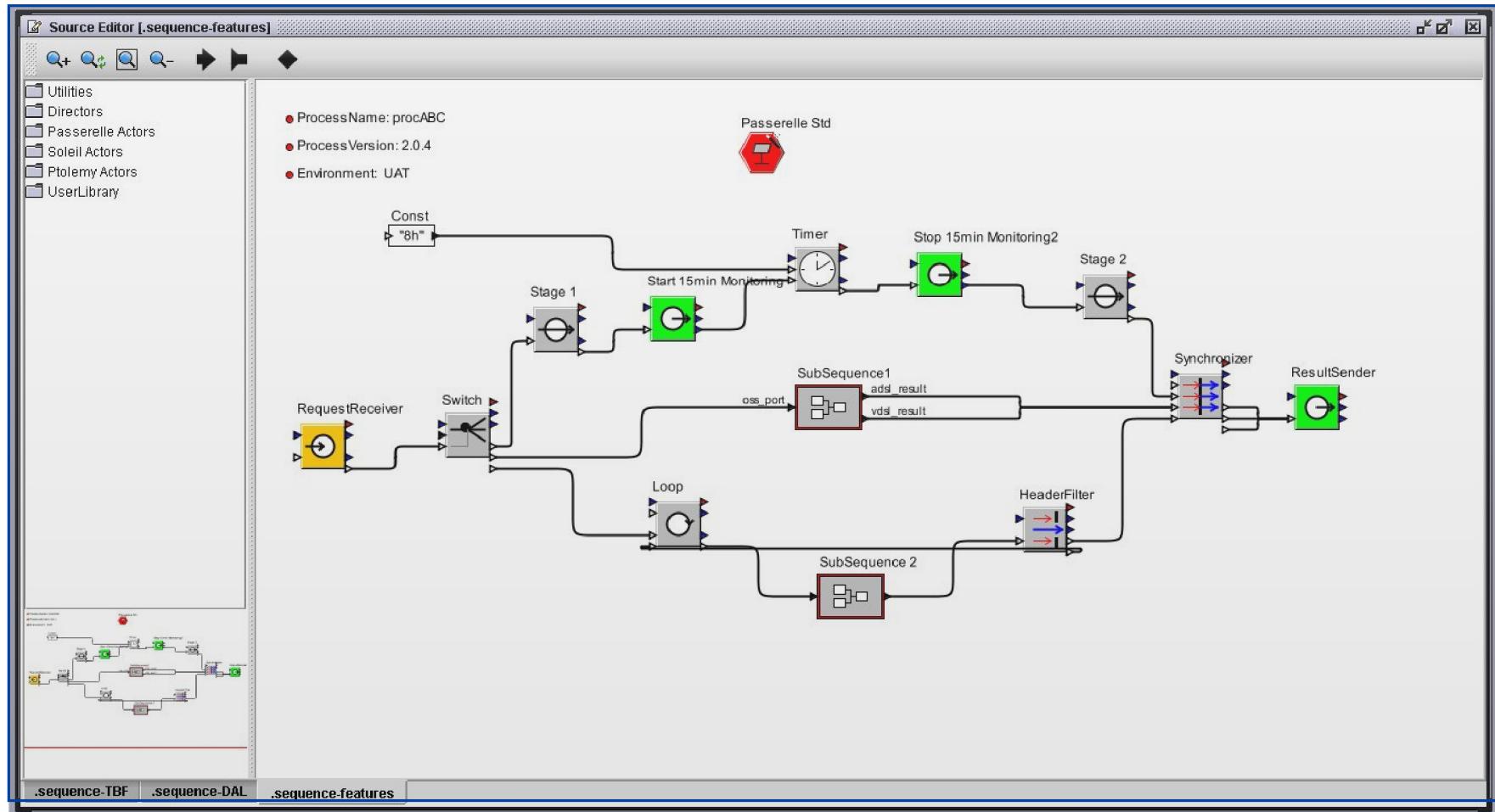
Introducing a modular and dynamic platform for high-level process automation & orchestration.

Obtained by integrating and extending “best-of-breed” Java technologies, matching our architectural vision...

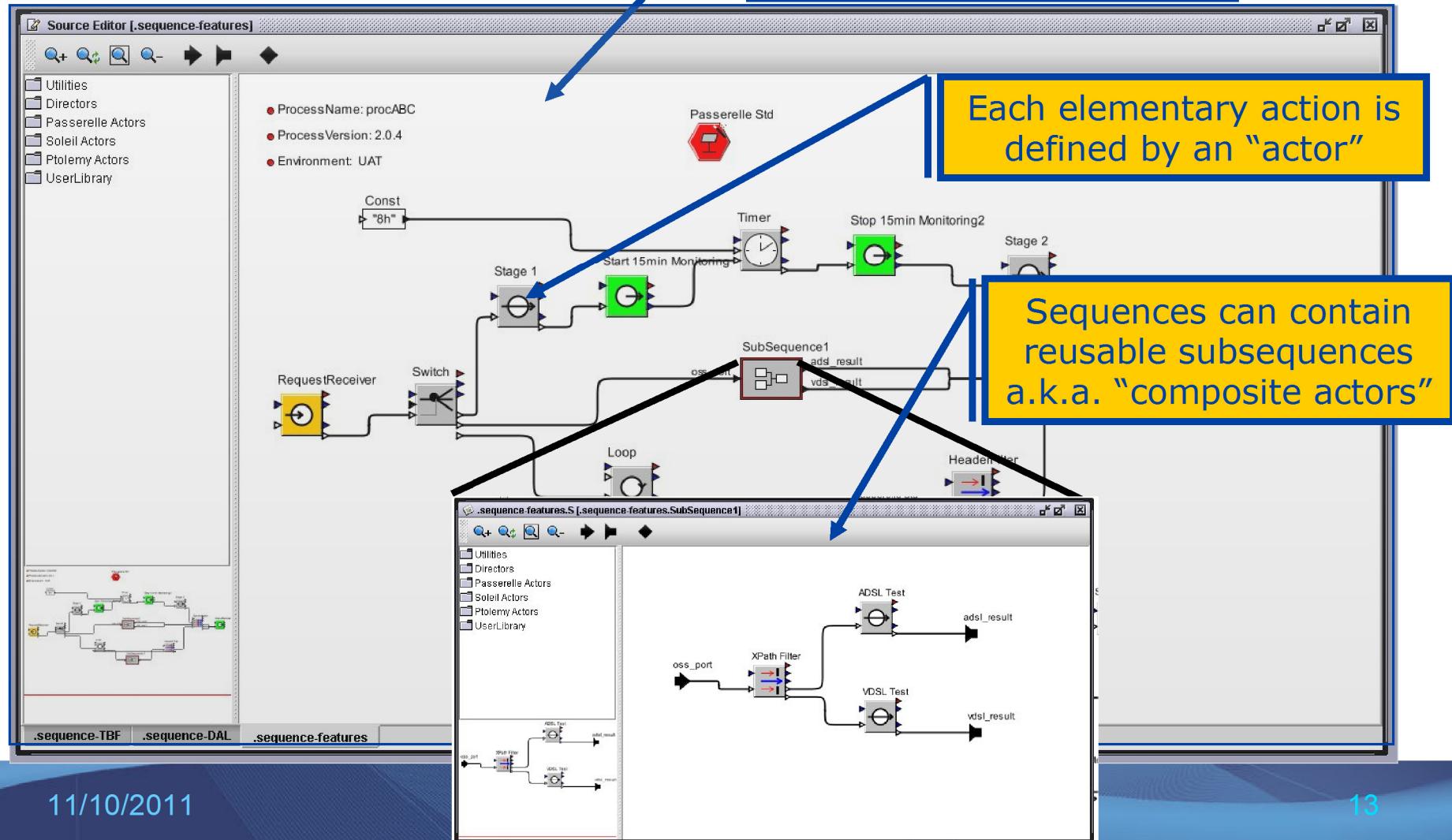
The Passerelle platform



Passerelle



Sequences can be defined
in a graphical editor



Benefits for users & actor developers :

- Reliable high-throughput execution engine
- Transparent concurrency, asynchronous operations, internal buffering
- Graphical executable hierarchical models
- Range of existing reusable actor libraries : maths, flow control, web services, Tango control, rules-based analysis, email notifications, Python scripts, ...

=> *think in terms of reusable high-level tasks i.o. device drivers, scripts, threads, APIs etc.*

Ptolemy II

Ptolemy II is an open-source platform for actor-based modeling and development, from Berkeley University.

Passerelle wraps Ptolemy II, simplifies actor development and adds modeling and operational tools.

See <http://ptolemy.berkeley.edu/ptolemyII>

The reference for modular Java systems :

- Simple packaging : a module is a *bundle* is a .jar
- Strict interface and dependency management with versioning
- Bundles are “active” modules with defined life-cycle
- Support for dynamically updating operational systems without downtime (installing new, updated bundles)
- Clean and high-performance service-based architecture

See <http://www.osgi.org>

Designing actors

Define a domain model for passing data around.

Decouple data representation from physical sources.

Think in reusable actor libraries.

Actors use services, provided by pluggable OSGi bundles.

Actors are “just” the glue between orchestration layer and service layer.

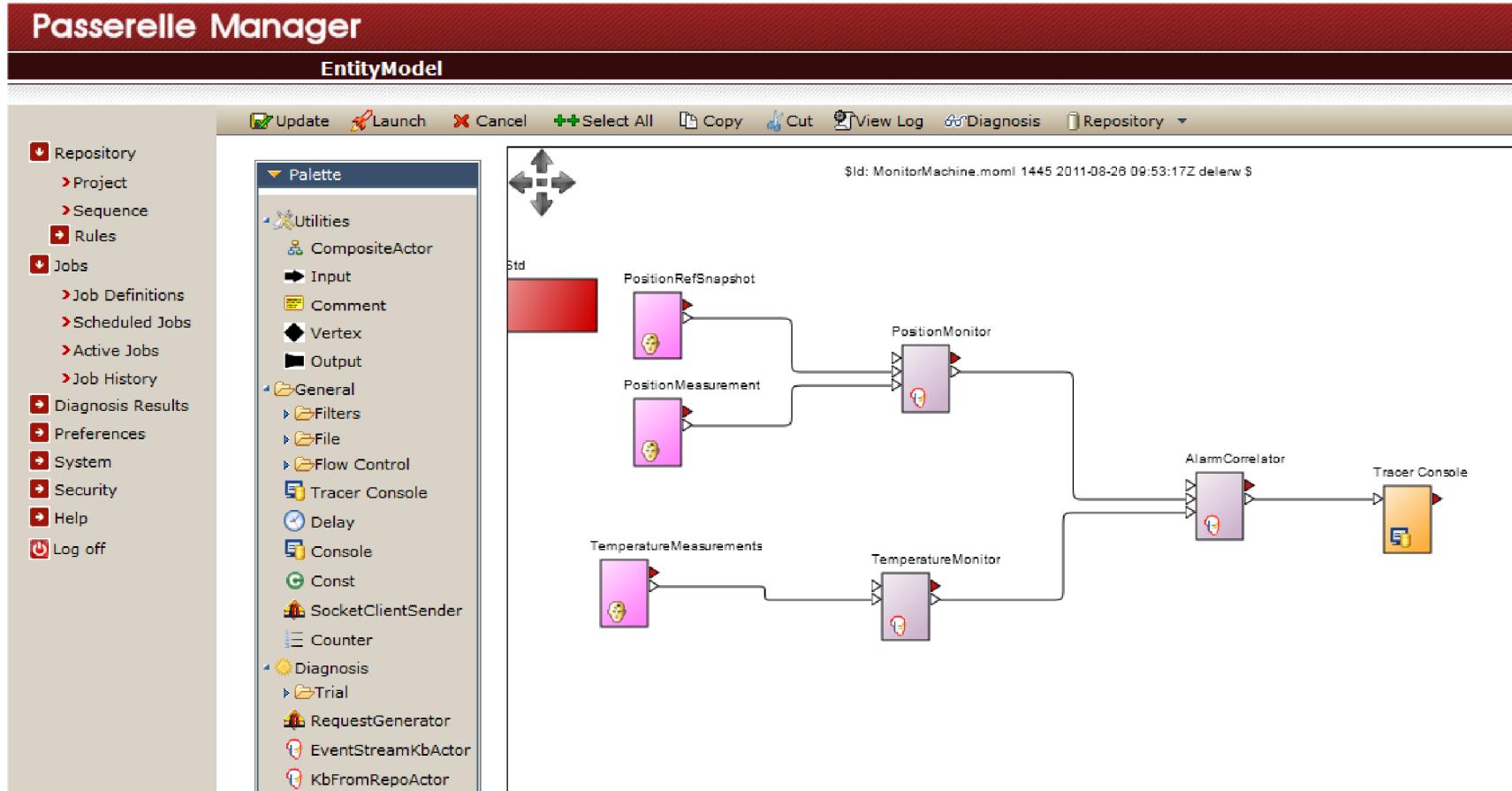
OSGi then allows hot updates without downtime, *which is nice...*

Passerelle Manager

Full-featured remote access via std web technology

- Collaborate on process design, execution, monitoring etc on a secure and robust server-based platform
- Automated scheduling of process executions
- Monitoring and diagnostics within same platform
- Base suite extensible “live” through OSGi bundle upload & integrated management console
- Process definitions, config data, analysis rules, traces & OSGi bundles all in relational DB

Browser-based process editor



Scheduling jobs

Passerelle Manager

Scheduled Jobs

Create **Cancel**

Name: TestJob

Job Definition:

Name	Group	User	Description
full prefs synch	Default	ica	
helloworld	Default	ica	

Select Trigger Type: Every Interval

Every : 1

Type: Days

Start Date: 10/10/2011 22:00:00

End Date:

Repository
Project
Sequence
Rules
Jobs
Job Definitions
Scheduled Jobs
Active Jobs
Job History
Diagnosis Results
Preferences
System
Security
Help
Log off

On-line module management

Passerelle Manager

OsgiBundle

Load Exit

Repository
 > Project
 > Sequence
 + Rules
 Jobs
 > Job Definitions
 > Scheduled Jobs
 > Active Jobs
 > Job History
 Diagnosis Results
 Preferences
 System
 Logs & Trails
 Bundles
 > OSGI Bundle Console
 > OSGI Bundle Repository
 Momo Class
 Security
 Help
 Log off

Bundle Name: State:

	ID	State	Bundle Name
[]	0	Active	org.eclipse.osgi_3.4.3.R34x_v20081215-1030
[]	1	Starting	org.eclipse.birt.chart.device.extension_2.6.1.v20100902
[]	2	Active	org.mortbay.jetty_5.1.14.v200806031611
[]	3	Active	org.eclipse.equinox.app_1.1.0.v20080421-2006
[]	4	Starting	org.eclipse.birt.report.model_2.6.1.201101041006
[]	5	Starting	org.eclipse.datatools.connectivity.oda_3.3.1.v201008030730
[]	6	Active	org.apache.xml.resolver_1.2.0.v201005080400
[]	7	Active	org.apache.commons.math_2.0.0
[]	8	Active	org.w3c.dom.smil_1.0.0.v200806040011
[]	9	Active	org.apache.xml.serializer_2.7.1.v201005080400
[]	10	Active	org.mozilla.rhino_1.6.0.v20070710
[]	11	Starting	org.eclipse.birt.chart.device.svg_2.6.1.v20100902
[]	12	Active	com.isencia.passerelle.engine_5.6.1
[]	13	Active	com.isencia.passerelle.model.rest_1.0.0.qualifier
[]	14	Active	com.isencia.sherpa.flow_5.4.0
[]	15	Starting	org.eclipse.birt.report.engine.dataextraction_2.6.1.v20100826
[]	16	Resolved	com.isencia.passerelle.diagnosis.impl.shb_5.6.0
[]	17	Active	com.isencia.sherpa.security_5.4.0.201012131651
[]	18	Resolved	com.isencia.passerelle.actor.conf_5.6.0
[]	19	Active	com.thoughtworks.xstream_1.3

Other packaging options

Besides the web-based production server,

- Standalone Eclipse RCP
- Eclipse plugins for workflow in other workbenches
(e.g. DAWB – collaboration between ESRF, Soleil, iSencia)
- Swing HMI
- custom packages...

=> all share common core bundles

Some example uses

Soleil :

- Beamline alignment processes defined by scientists (e.g. PX1, PX2, SWING, MARS)
- Data acquisition workflow automation (e.g. ANTARES)
- Machine monitoring :
 - Monitor beamline source positions vs reference and drift
 - Identify possible root causes (temperature drift, vibrations, power supply issues etc)

ESRF : Passerelle as workflow engine for EDNA (DAWB)

Belgacom : Automated diagnosis and repair engine for helpdesk (700 operators) and field technicians (2500).

- +30-50K diagnoses per day, handling 30-100M data items
- access databases, network devices, customer modem &STB etc.

Conclusion

Fundamental modular approach

Rigorously follow good design principles

Use the domain language

Take advantage of the OSGi platform and of actors

Enjoy your extensible platform for process automation!

Thank you