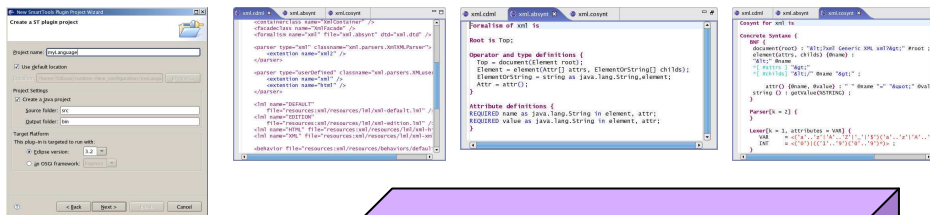


 **Smart Tools** Software Factory

Didier Parigot  
INRIA

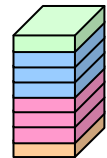
# Overview of SmartTools Software Factory

Wizard      Model      View      Component

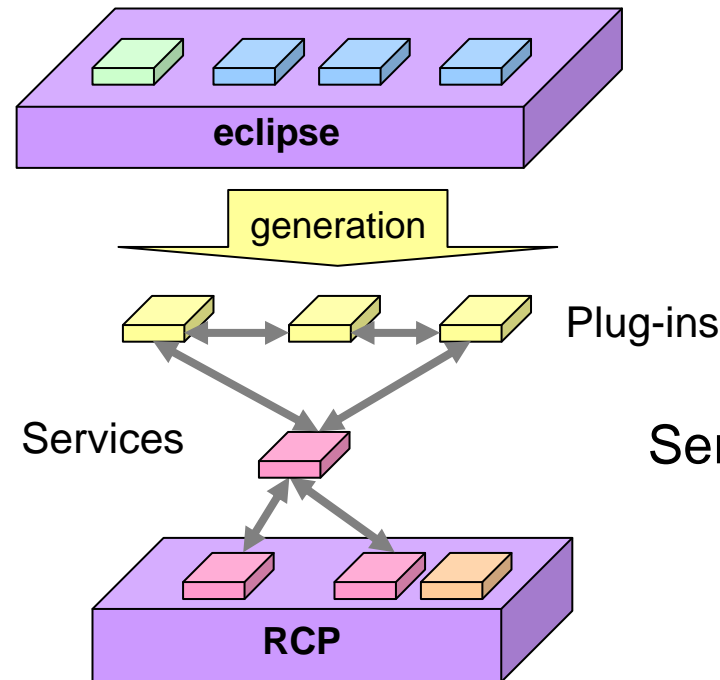


Domain Specific Language

Editor + Generator



SmartTools Features



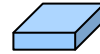
Plug-ins

Services

Service-Oriented Architecture (OSGi)

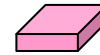
RCP

## DSLs of SmartTools



Editor and Generator

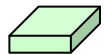
- Absynt DSL describes your language
  - ♦ Produce Java Model, Default visitors, Out-Line view
- Cosynt DSL describes a concrete syntax and a graphical view
  - ♦ Produce the parser and Xslt transformations
- Cdml DSL describes the services of your plug-in
  - ♦ Produce the facade, the container and the activator
- SOA DSL describes the initial topology of your RCP
  - ♦ Used in SmartTools perspective
- Lml DSL describes a GUI configuration



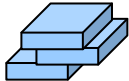
Component Manager

## SmartTools inside Eclipse Framework

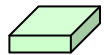
SmartTools Features contains 30 plug-ins



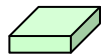
- SmartTools Perspective



- DSL Editors

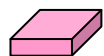


- SmartTools Wizard to create a project

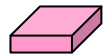


- SmartTools DSL wizards

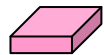
- SmartTools Builder to apply any DSL generators



- SmartTools SOA view



- SmartTools Multi-views



- SmartTools Console (visualization of SOA messages)

- SmartTools Debug mode

# SmartTools in Action

The screenshot displays the Eclipse IDE in the SmartTools Perspective. The interface is divided into several panes:

- SmartTools DSL editors:** The top editor pane shows the source code for a formalism named 'NAME'. The code includes a root definition and operator/type definitions.
- SmartTools Console:** The console pane at the top shows system messages and the start of the formalism definition.
- SmartTools SOA:** The SOA (Software Object Architecture) pane shows a graph of relationships between components, with nodes like 'fr.smarttools.core.view.main.GViewContainer:1' and 'workspace.test/x.absynt' highlighted.
- SmartTools Views:** The bottom pane shows a tree view of the formalism structure, including 'Formalism', 'definition', and 'typeDefinition'.
- SmartTools Perspective:** The overall layout of the IDE, including the menu bar, toolbar, and project explorer on the left.

## Advantages of Software Factory and SOA approach

- First version of SmartTools (without Eclipse) at 2002
  - ◆ Bootstrap approach
- Results of SmartTools integration into Eclipse
  - ◆ Use Eclipse development (feature, GUI support...)
  - ◆ SmartTools is complementary to the Eclipse functionalities
  - ◆ Simplified integration thanks to SOA approach
  - ◆ Build more rapidly and homogeneous Plug-ins and RCPs

## Conclusion

- Demonstrations on each SmartTools functionality
  - ◆ <http://www-sop.inria.fr/smartool/eclipse/presentations/>
- Download SmartTools:
  - ◆ <http://www-sop.inria.fr/smartool/eclipse/update/site.xml>
  - ◆ <http://gforge.inria.fr/projects/smarttools/>

## Futures

- Eclipse Projects: STP, SWORDFISH, etc
- More dynamic SOA approach
  - ◆ Adding dynamic services, treatments, orchestration

Thank you for your attention

- Contact: Didier.Parigot@inria.fr
  - ◆ <http://www-sop.inria.fr/smartool>
- Questions ?



SmartTools