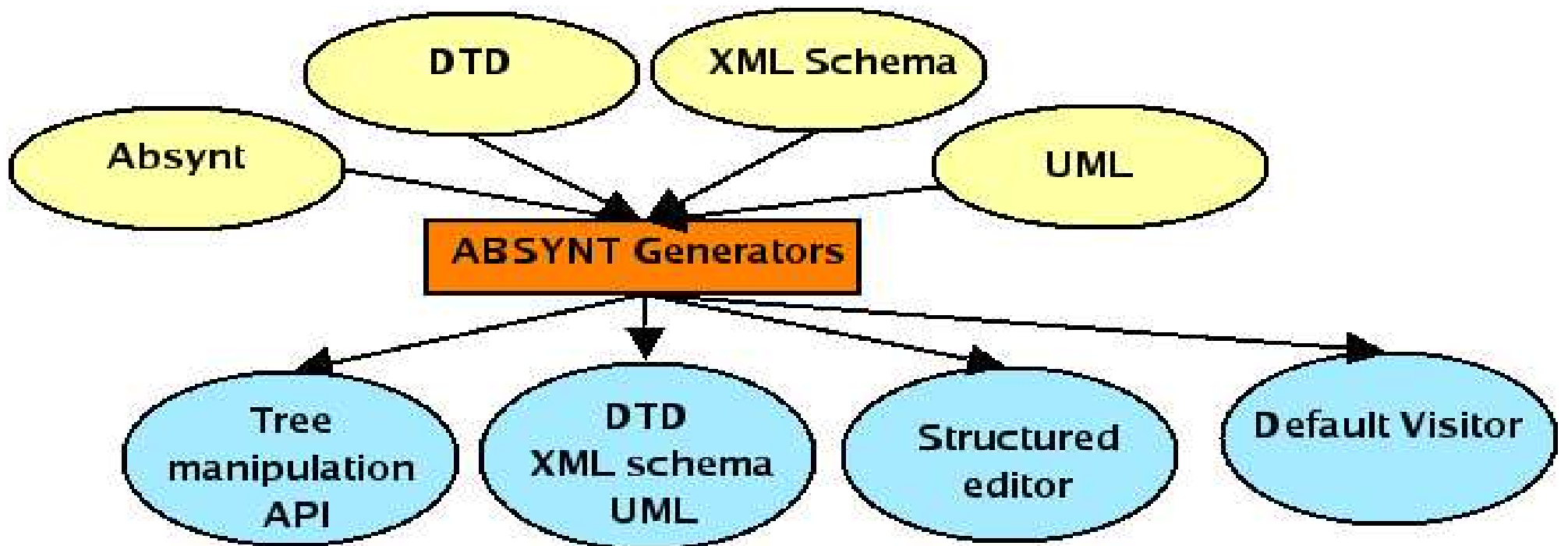


DSL [Absynt] pour définir des DSLs



Absynt

```

    Formalism of lml is

    Root is Layout

    Operator and type definitions {
    Layout = layout (FS [*] fslist);
    FS = Frame, Set;
    Frame = frame (Set [*] setlist);
    Set = set (ViewsGroup vg);
    ViewsGroup = split (ViewsGroup k
    }

    Attribute definitions {
    REQUIRED statusBar as java.lang.St
    REQUIRED width as java.lang.String
  
```

```

    <layout>
    <frame dynTabSwitch="off"
      height="800"
      statusBar="on"
      title="SmartTools: lml"
      width="1000"
    <set title="lml example for
      position="68" >
    <split orientation="1"
      <view behavior=""
        docRef="file:demos/1
        styleSheet="resource
        title="Layout demo.T
        transform="resources
        viewType="fr.smartto
      <view behavior=""
  
```

XML Schema

```

    <group ref="Set" />
  </choice>
</group>
<element name="Frame"
  type="FrameType" />
<element name="Layout"
  type="LayoutType" />
<element name="Set"
  type="SetType" />
<element name="ViewsGroup"
  type="ViewsGroupType" />
<element name="FS"
  type="FSType" />
<complexType name="FrameType">
  <complexContent>
    <extension base="xsd:null">
      <sequence>
        <group maxOccurs="0"
          minOccurs="0"
          ref="Set" />
      </sequence>
      <attribute name="title"
        type="java.lang.String"
        use="required" />
      <attribute name="height"
        type="java.lang.String"
        use="required" />
      <attribute name="width"
        type="java.lang.String"
        use="required" />
      <attribute name="statusBar"
        type="java.lang.String"
        use="required" />
      <attribute name="dynTabSwitch"
  
```

UML

```

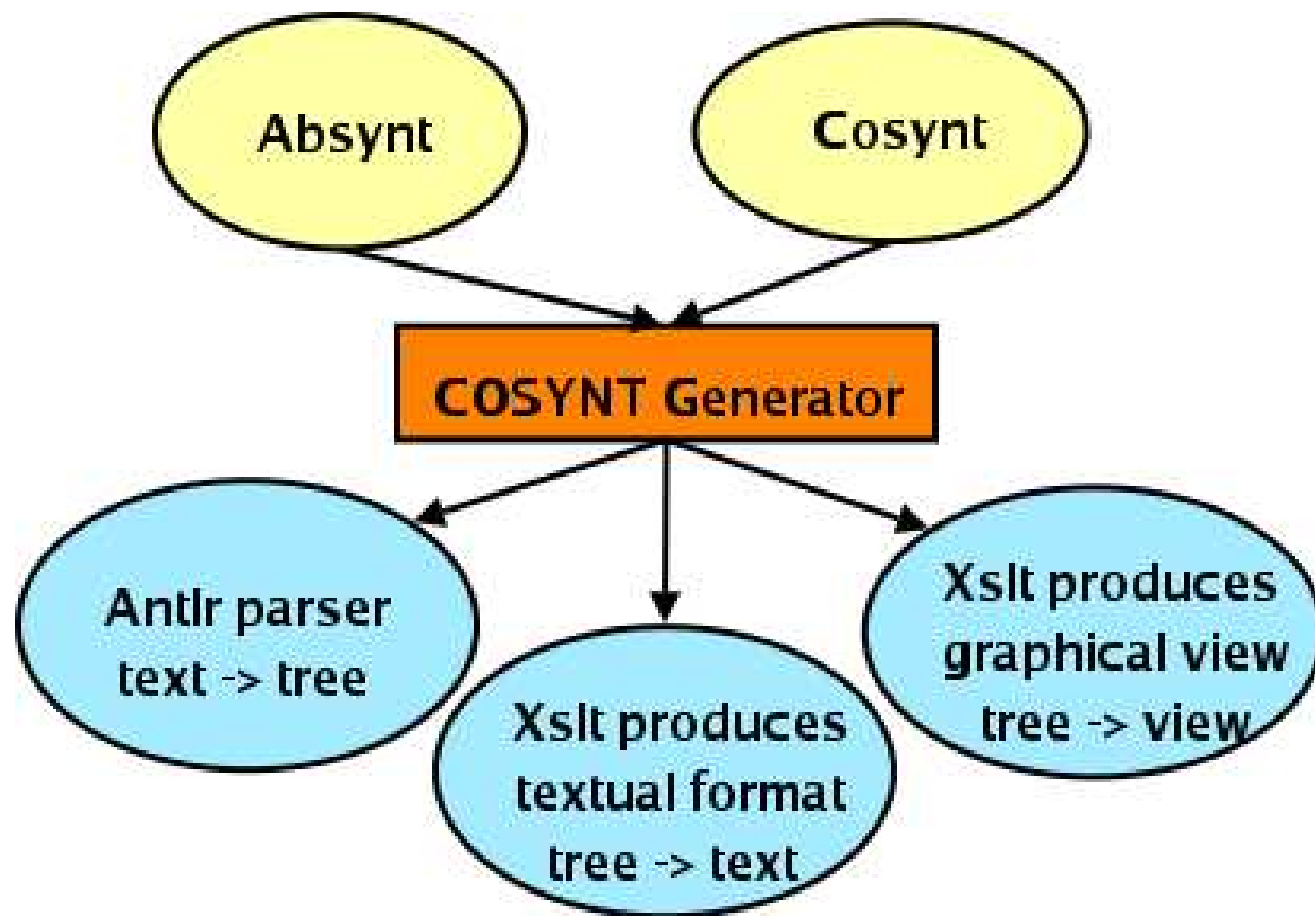
    baseDeRemarques : Entity
    {
    nom : Stringid;
    etats : Etat [*] INVERSE base;
    remarques : Remarque [*] INVERSE base;
    solutions : Solution [*] INVERSE base;
    etapes : Etape [*] INVERSE base;
    origines : Origine [*] INVERSE base;
    priorites : Priorite [*] INVERSE base;
    types : Type [*] INVERSE base;
    themes : Theme [*] INVERSE base;
    }
  
```

DTD

```

    <!-- DTD for 'lml' formalism -->
    <!-- Generate from absynt specification -->
    <!-- begin -->
    <!-- ENTITY definition -->
    <!-- -->
    <ENTITY ViewsGroup (viewsplit)>
    <ENTITY Set (set)>
    <ENTITY Layout (layout)>
    <ENTITY Frame (frame)>
    <ENTITY FS (%Set;%Frame;)>
    <!-- -->
    <!-- ELEMENTS definition -->
    <!-- -->
    <!-- -->
    <!-- -->
    <ELEMENT view EMPTY>
    <ATTLIST view transform CDATA #REQUIREDbehavior CDATA #REQUIREDtitle CDATA
    <ELEMENT frame ( ( %Set;)* )>
    <ATTLIST frame width CDATA #REQUIREDstatusBar CDATA #REQUIREDheight CDATA
    <ELEMENT layout ( ( %FS;)* )>
    <ELEMENT set ( ( %ViewsGroup;)* )>
    <ATTLIST set title CDATA #REQUIRED>
    <ELEMENT split ( ( %ViewsGroup;), ( %ViewsGroup;)* )>
    <ATTLIST split position CDATA #REQUIREDorientation CDATA #REQUIRED>
    <!-- end -->
  
```

DSL [Cosynt] pour définir divers formats

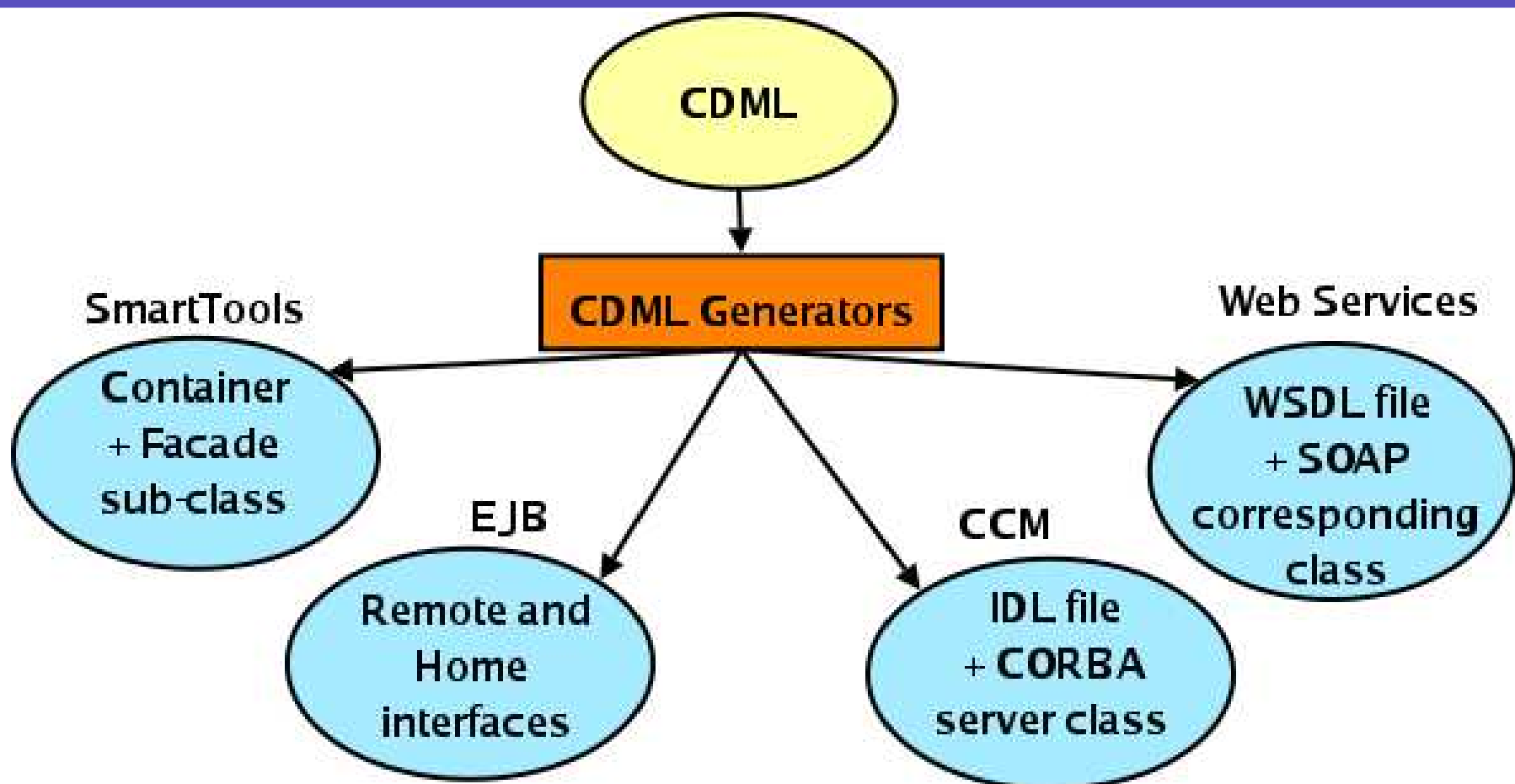


Éditeur

The screenshot shows the SmartTools IDE with two windows. The left window displays the component manager for 'Component:componentsManager of type:manager Extends: abstra'. The right window shows the XSLT editor with the following code:

```
File Other Component help Impl Layout Edition XSLT
Introduction Css Css XSLT XSLT
/home/dparigot/SmartTools1/SmartTools/contrib/comp...
'arg' ->
  Nbox(value-of('name')), {
    AlabelText("null""name"value-of('./@name'));
    Slabel("keyword", text(": "));
    AlabelText("null""javatype"value-of('./@javatype'));
    Slabel("keyword", text("/"));
    AlabelText("null""doc"value-of('./@doc'));
    Slabel("keyword", text("/"));
  });
'component' ->
  Nbox(value-of('name')), {
    Sbox("line", {
      Slabel("keyword", text("Component:"));
      AlabelText("null""na""me"value-of('./@name'));
      Slabel("keyword", text(" of type:"));
      AlabelText("null""type"value-of('./@type'));
    });
    Sbox("line", {
      Slabel("keyword", text(" Extends: "));
      AlabelText("null""extends"value-of('./@extends'));
    });
    Sbox("line", {
      Slabel("keyword", text(" in NameSpace: "));
      AlabelText("null""ns"value-of('./@ns'));
    });
  });
  if('./documentation') do {
    apply template on './documentation'
  };
```

DSL [CDML] pour définir des composants



Composants

The screenshot displays the Eclipse IDE with the SmartTools plugin installed. The Package Explorer on the left shows a list of jars and packages, including 'componentsgenerators'. The main window shows a 'Debug: Components State' graph with a central red node 'ComponentRegistry' and numerous yellow nodes representing different components, all interconnected by red lines.

SmartTools
File Other Component help lml Layout Edition Ls
Absynt Set Css XSLT XSLT XML (default)

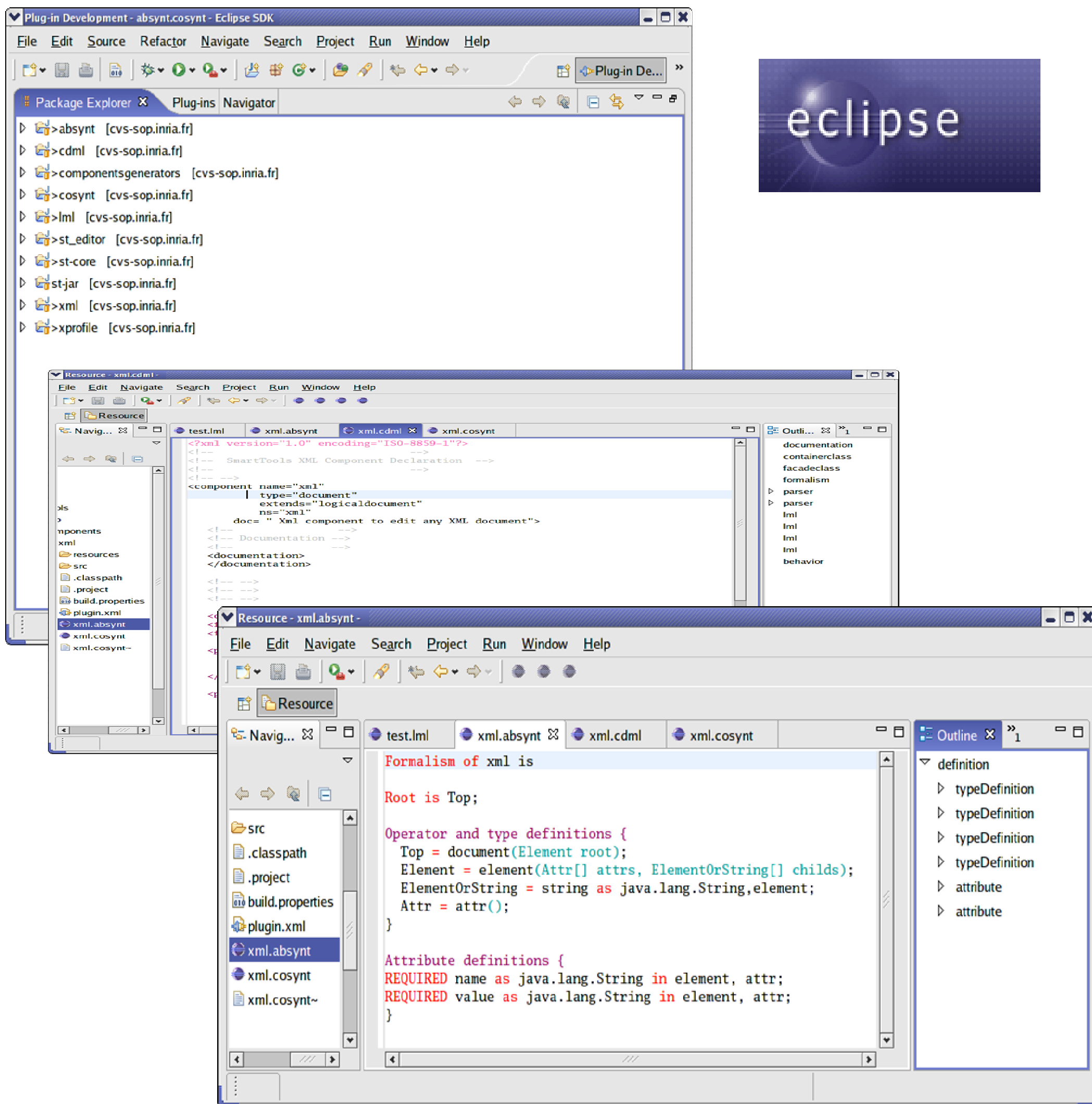
Plug-in Development - absynt.cosynt - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Plug-ins Navigator Debug: Components State

Plug-ins Eclipse

Type de composants

SmartTools dans Eclipse

Résultat d'un mois de travail !



Exemples de syntaxe de surface [Cosynt]

Absynt de CDML

```
SmartTools
File Other Component help lml Layout Edition Absynt
Introduction Cdml (syntax) Cdml (default) Lml Set (default) Absynt Set
/home/dparigot/SmartTools1/SmartTools/contrib/components/cdml/cdml.absynt
Formalism of cdml is
Root is Top;
Operator and type definitions{
  Top= component (Documentation ? doc, ContainerClass ? containerclass, FacadeClass ? facadeclass, Formalism ? formalism, Parser ? parser, Lml ? lml, Behavior ? behavior, Dependence ? dependance);
  Formalism= formalism ();
  ContainerClass= containerclass ();
  FacadeClass= facadeclass ();
  Parser= parser (Extension [] extension);
  Extension= extension ();
  Lml= lml ();
  Behavior= behavior ();
  Dependance= dependance ();
  Attribute= attribute (), arg (), binding (Attribute [] parameter);
  Port= input (Attribute [] parameter), output (Attribute [] parameter), inout (Attribute [] parameter);
  Documentation= documentation as java.lang.String;
}
Attribute definitions{
  REQUIRED name as java.lang.String in formalism, containerclass, facadeclass, extension;
  REQUIRED classname as java.lang.String in parser;
  REQUIRED type as java.lang.String in parser, component;
  REQUIRED extends as java.lang.String in component;
  REQUIRED file as java.lang.String in lml, behavior;
}
Simple Selection current : 1.<height>

'arg' ->
  Nbox(value-of('name()'), {
    AlabelText("null""name"value-of('./@name"));
    Slabel("keyword", text(": "));
    AlabelText("null""javatype"value-of('./@javatype"));
    Slabel("keyword", text("/"));
    AlabelText("null""doc"value-of('./@doc'));
    Slabel("keyword", text("**/"));
  });

'component' ->
  Nbox(value-of('name()'), {
    Sbox("line", {
      Slabel("keyword", text("Component:"));
      AlabelText("null""na""me"value-of('./@name"));
      Slabel("keyword", text(" of type:"));
      AlabelText("null""type"value-of('./@type'));
    });
    Sbox("line", {
      Slabel("keyword", text(" Extends: "));
      AlabelText("null""extends"value-of('./@extends"));
    });
    Sbox("line", {
      Slabel("keyword", text(" in NameSpace: "));
      AlabelText("null""ns"value-of('./@ns"));
    });
    if (./documentation) do {
      apply template on './documentation'
    }
  });
Simple Selection current : 1.5.1.
```

Cosynt de CDML

```
SmartTools
File Other Component help lml Layout Edition Cosynt
Introduction Schema (default) Schema (default) Cdml (default) Cdml (syntax) Cosynt Set (default)
/home/user/dparigot/ST5/SmartTools/contrib/components/cdml/cdml.cosynt
Cosynt for cdml is
Concrete Syntaxe {
  BNF {
    component(Doc, Containerclass, Facadeclass, Formalism, Parser, Lml, Behavior, Dependence) [jar, @name] : " @jar @name;
    documentation : %VAR;
    formalism [ @name ] : " @name;
    parser(Extension) [ @type, @generator, @classname ] : * [ #Extension ] @type @generator;
    extension [ @name ] : " @name;
    facadeclass [ @name, @userclassname ] : " @name @userclassname;
    attribute [ @javatype, @name, @doc ] : @name : " @javatype;
    containerclass [ @name ] : " @name;
    arg [ @javatype, @name ] : @name : toto @javatype;
    behavior [ @file ] : " @file;
    lml [ @file, @name ] : " @file @name;
    input(Parameter) [ @method ] : @method * [ #Parameter separator ' , ' beforeList ' ( ' ;
    inout(Parameter) [ @outputname, @name, @outputmethod, @method ] : * [ #Parameter separator ' , ' beforeList ' ( ' ;
    binding(Parameter) [ @toMethod ] : * [ #Parameter ] @toMethod;
    output(Parameter) [ @method ] : @method * [ #Parameter separator ' , ' beforeList ' ( ' ;
  }
  Parser
  [k=1]{
  }
  Lexer
  [k=1, attributes=VAR]{
    VAR=;
  }
  Layout {
  }
}
Simple Selection current : 1.6.1.<position>
```

XSLT produit

Exemple

```
SmartTools
File Other Component help lml Layout Edition Cdml
Introduction Cdml (syntax) Cdml (syntax)
/home/dparigot/SmartTools1/SmartTools/src/core/fr/smarttools/core/view/resources/layout.cdml
Component:layout of type:view Extends: abstractView in NameSpace: fr.smarttools.core.view
Container class: GLayoutContainer
Facadeclass: GdocLayout Userclassname:
Input:statusInfo (statusText : java.lang.String )
Input:statusWarn (statusText : java.lang.String )
Output:modify (modif : java.lang.String/**/, syntaxe : java.lang.String/**/, attributes : java.util.HashMap/**/ )
Simple Selection current : 1.<height>
```

DSL [Cosynt] pour définir diverses représentations

Concrete syntax ↔ DSL definition → Visual syntax

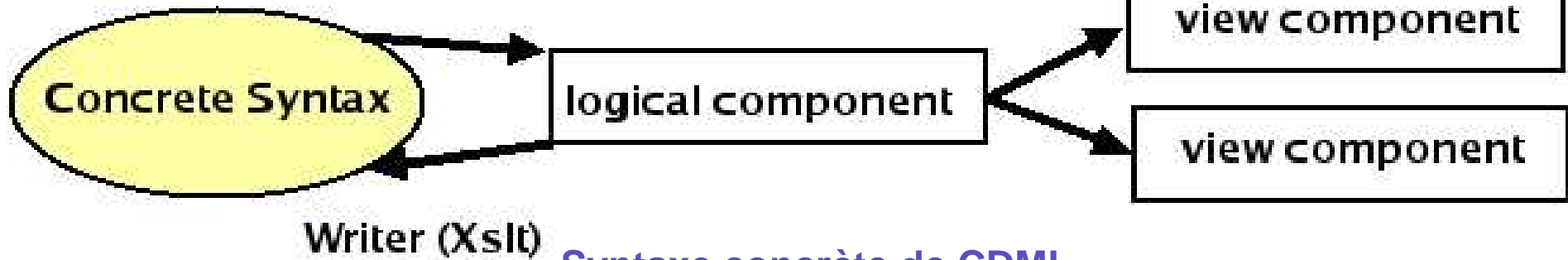
Text format

Logical view

Graphical view

Reader (parser)

Xslt transformations



Syntaxe XML de CDML

Syntaxe concrète de CDML

```

<component doc=Layout Markup Language
  extends=logicaIdocument
  name=lml
  ns=lml
  type=document
  <containerclass name=LmlContainer/>
  <facadeClass name=LmlFacadeFacade
    userclassname=LmlFacade/>
  <formalism dtd=lml.dtd
    file=lml.absynt
    name=lml />
  <parser classname=lml.parsers.LmlXMLParser
    type=xml />
  </parser>
  <lml file=resources:lml/lml-default.lml
    name=DEFAULT />
  <lml file=resources:lml/lml-edition.lml
    name=EDITION />
  <lml file=resources:lml/lml-navig.lml
    name=NAVIG />
  <lml file=resources:lml/lml-syntax.lml
    name=SYNTAX />
  <lml file=resources:lml/lml-absynt.lml
    name=ABSYNT />
  <behavior file=resources:behaviors/lml-behaviors.bhv />
  <input doc=update tree
    method=updateSvc
    name=update />
  <input doc=Process Skeleton
    method=processSkeletonSvc
    name=processLmlSkeleton />
  <input doc=Process Skeleton Direct
    method=processSkeletonDirect
    name=processLmlSkeletonDirect />
  </input>
</component>
  
```

```

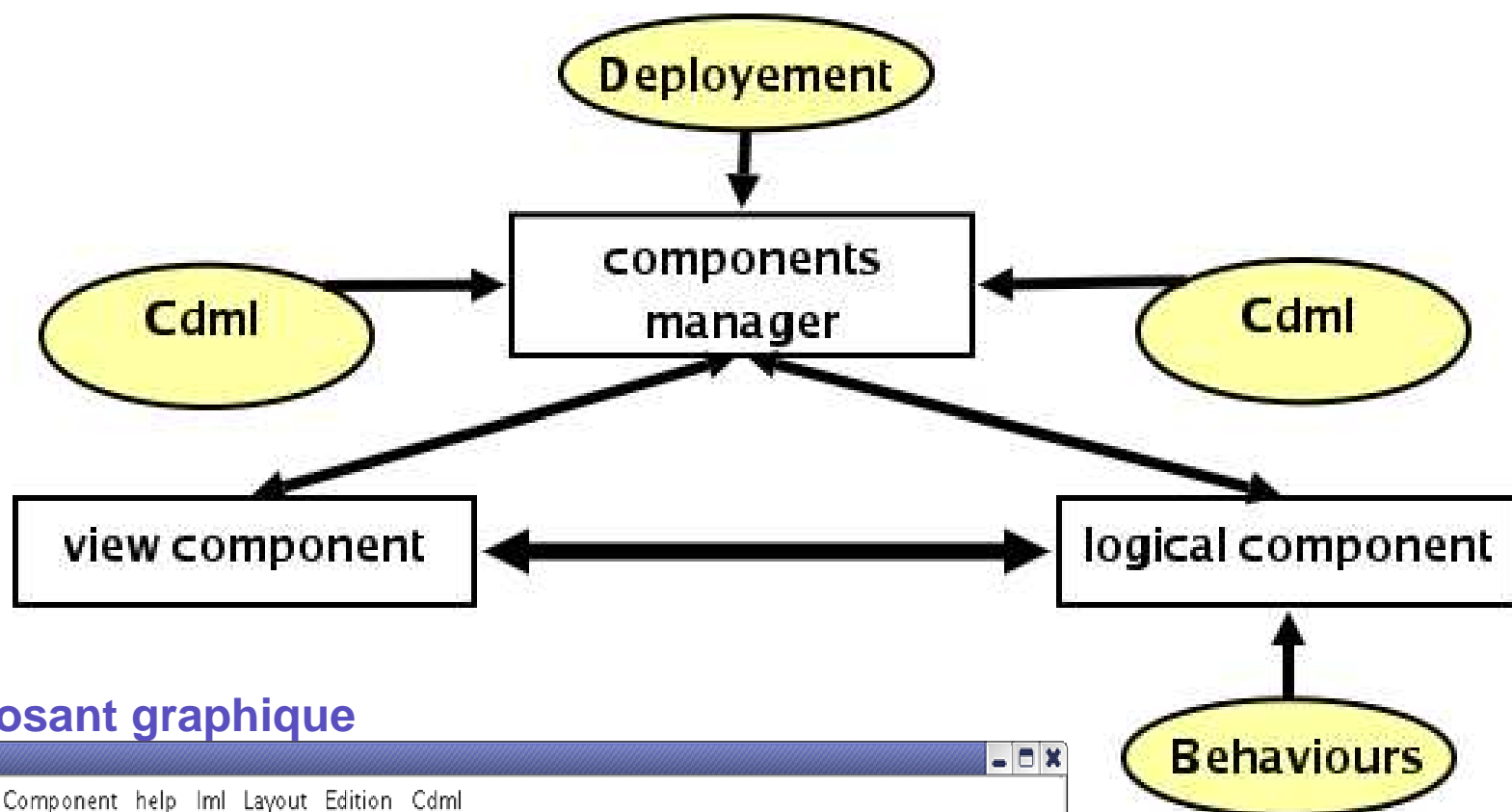
Component:lml of type:document Extends: logicaldocument in NameSpace: lml
Container class: LmlContainer
FacadeClass: LmlFacadeFacade Userclassname: LmlFacade
Formalism: lml
Parser: lml.parsers.LmlXMLParser of Type: xmlExtension:lml
Lml: DEFAULT in File: resources:lml/lml-default.lml
Lml: EDITION in File: resources:lml/lml-edition.lml
Lml: NAVIG in File: resources:lml/lml-navig.lml
Lml: SYNTAX in File: resources:lml/lml-syntax.lml
Lml: ABSYNT in File: resources:lml/lml-absynt.lml
Behavior: resources:behaviors/lml-behaviors.bhv
Input:updateSvc (transformationName : java.lang.String, orientation : java.lang.String
Input:processSkeletonSvc (skeleton : java.lang.String, arguments : java.util.HashMap,
Input:processSkeletonDirect (skeleton : java.lang.String )
  
```

Transformations XSLT

```

<set title=Cdml (default)>
  <view behavior=
    docRef=file:/home/dparigot/SmartTools1/SmartTools/src/components/lml/resources/lml.cdm
    styleSheet=resources:css/xmlPlus-styles.css
    title=/home/dparigot/SmartTools1/SmartTools/src/components/lml/resources/lml.cdm
    transform=resources:xsl/genericXmlPlus.xsl
    viewType=fr.smarttools.core.view.GdocView
  </set>
  <set title=Cdml (syntax)>
    <view behavior=
      docRef=file:/home/dparigot/SmartTools1/SmartTools/src/components/lml/resources/lml.cdm
      styleSheet=resources:css/cdml-styles.css
      title=/home/dparigot/SmartTools1/SmartTools/src/components/lml/resources/lml.cdm
      transform=resources:xsl/cdmlsyntax-bml.xsl
      viewType=fr.smarttools.core.view.GdocView
    </set>
  </set>
  <set title=Lml Set (default)>
  </set>
  
```

Architecture dirigée par les services

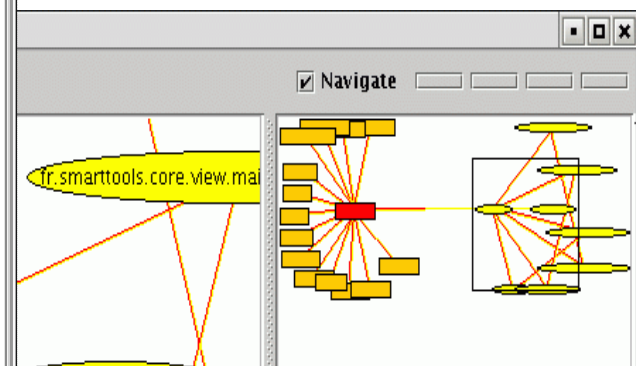


Composant graphique

Component: abstractView of type: view Extends: abstractContainer in Namespace: fr.smarttools.core.view

- Depend of: batik-css Jar file: batik-css.jar
- Depend of: batik-util Jar file: batik-util.jar
- Depend of: sac Jar file: sac.jar
- docRef: java.lang.String
- xslTransform: java.lang.String
- behaviors: java.lang.String
- Input: buildContent
- Input: dolnit (docRef: java.lang.String, xslTransform: java.lang.String, behaviors: java.lang.String)
- Input: modification (modif: fr.smarttools.core.util.modification.ModificationsManager)
- Input: initData (inits: java.util.HashMap)
- Input: selection (selection: fr.smarttools.core.util.selection.Selection)
- Input: addBehaviors (xmlBehaviors: java.lang.String)
- Output: (transformationName: java.lang.String/**/, orientation: java.lang.String/**/)
- Output: (selection: fr.smarttools.core.util.selection.Selection/**/)
- Output: newComponent (component: fr.smarttools.core.component.ContainerProxy/**/, name: java.lang.String/**/)
- Output: (jarName: java.lang.String/**/, urlToJar: java.lang.String/**/, dependenceForComponent: java.lang.String/**/)
- Output: (requestedFilename: java.lang.String/**/)
- Output: requestInitData
- Output: OpenDir (directoryName: java.lang.String/**/)

Graphes des composants



Menus

```

    <?xml Generator="SmartTools" >
    <behaviors >
    <action >
    <action name="Connect To CM">
    <msg name="connectTo"
    chooser="cm.ConnectToDialogBox"
    <msgattr name="id_src"
    value="ComponentsManager"
    </msgattr>
    <msgattr name="type_dest"
    value=""
    </msgattr>
    </msg>
    </action>
    </action>
  
```

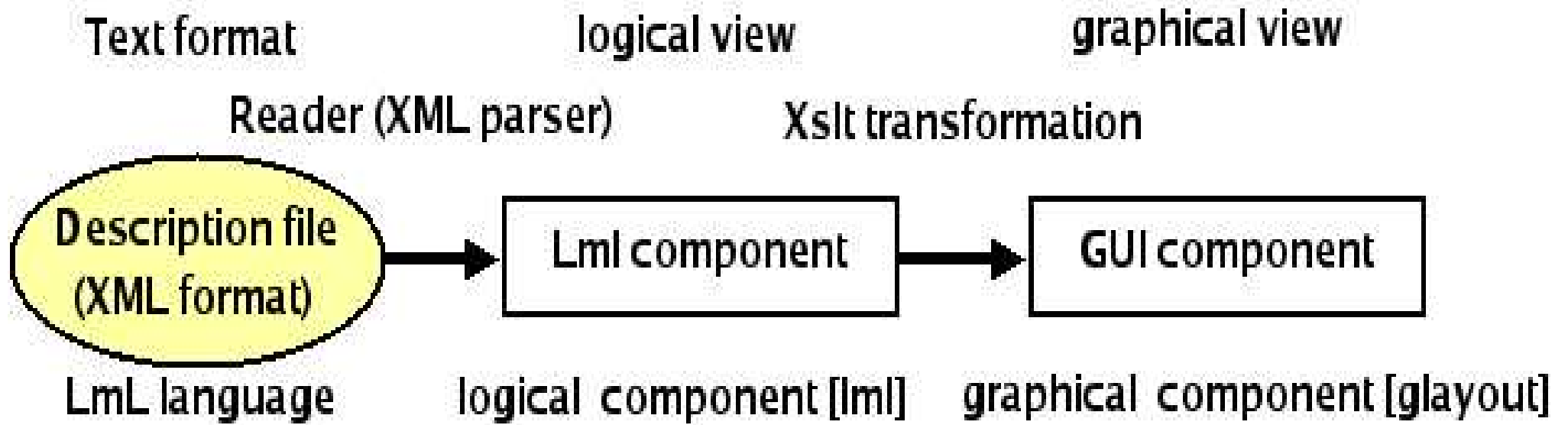
connectTo

Messages

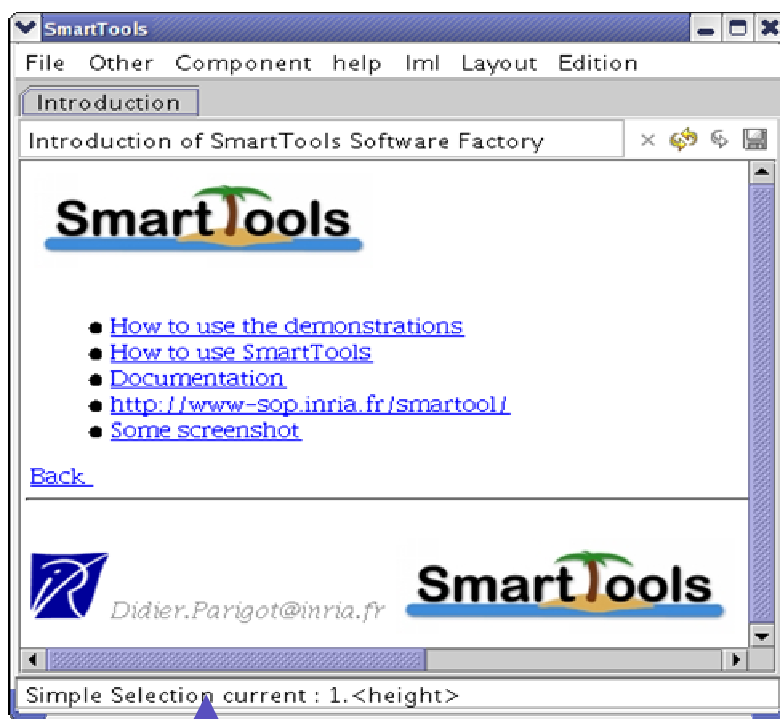
```

    LogView
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <modification> message
    main.GViewContainer:4 send a <select> message
    main.GViewContainer:4 send a <modify> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <selection> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <statusInfo> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <editionStructData> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <modification> message
    main.GViewContainer:4 send a <modify> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <modification> message
    main.GViewContainer:4 send a <select> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <selection> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <statusInfo> message
    SmartTools1/SmartTools/demos/cm/cmdline.xml send a <editionStructData> message
  
```

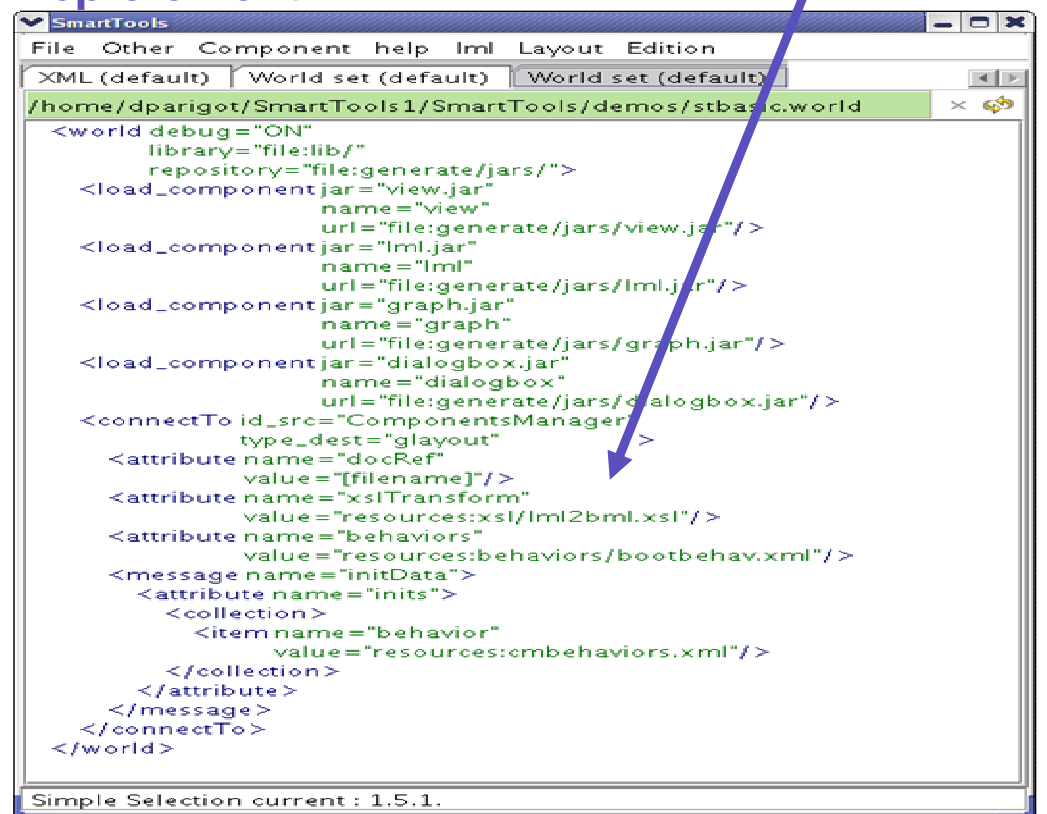
DSL [LML] pour définir une GUI



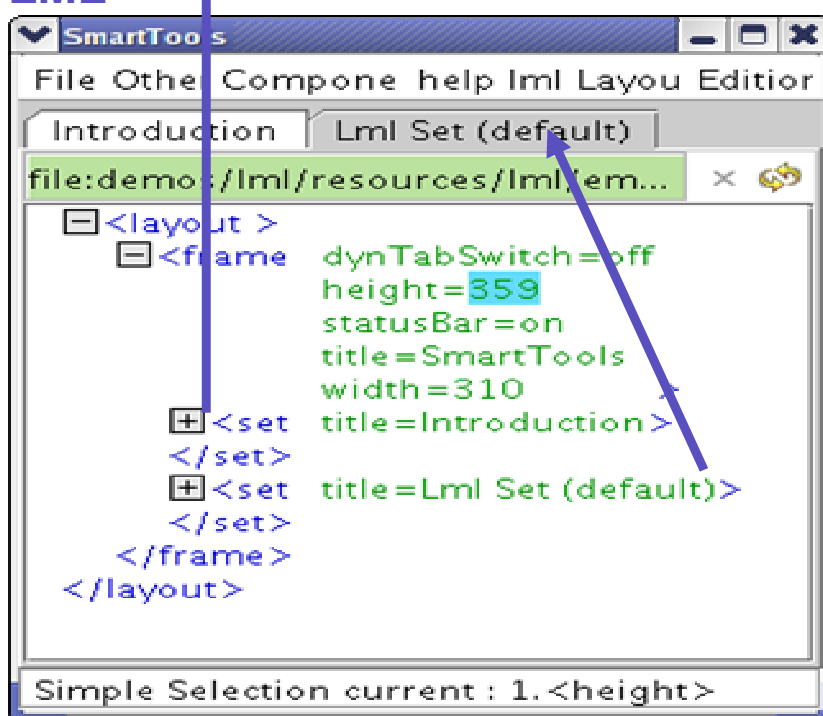
HTML



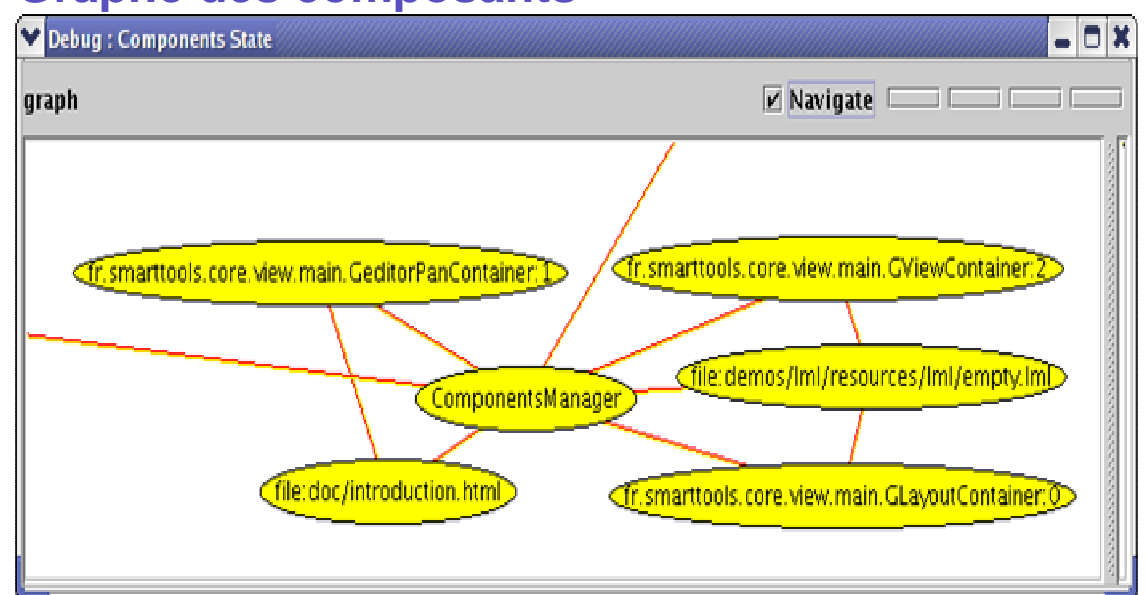
Déploiement



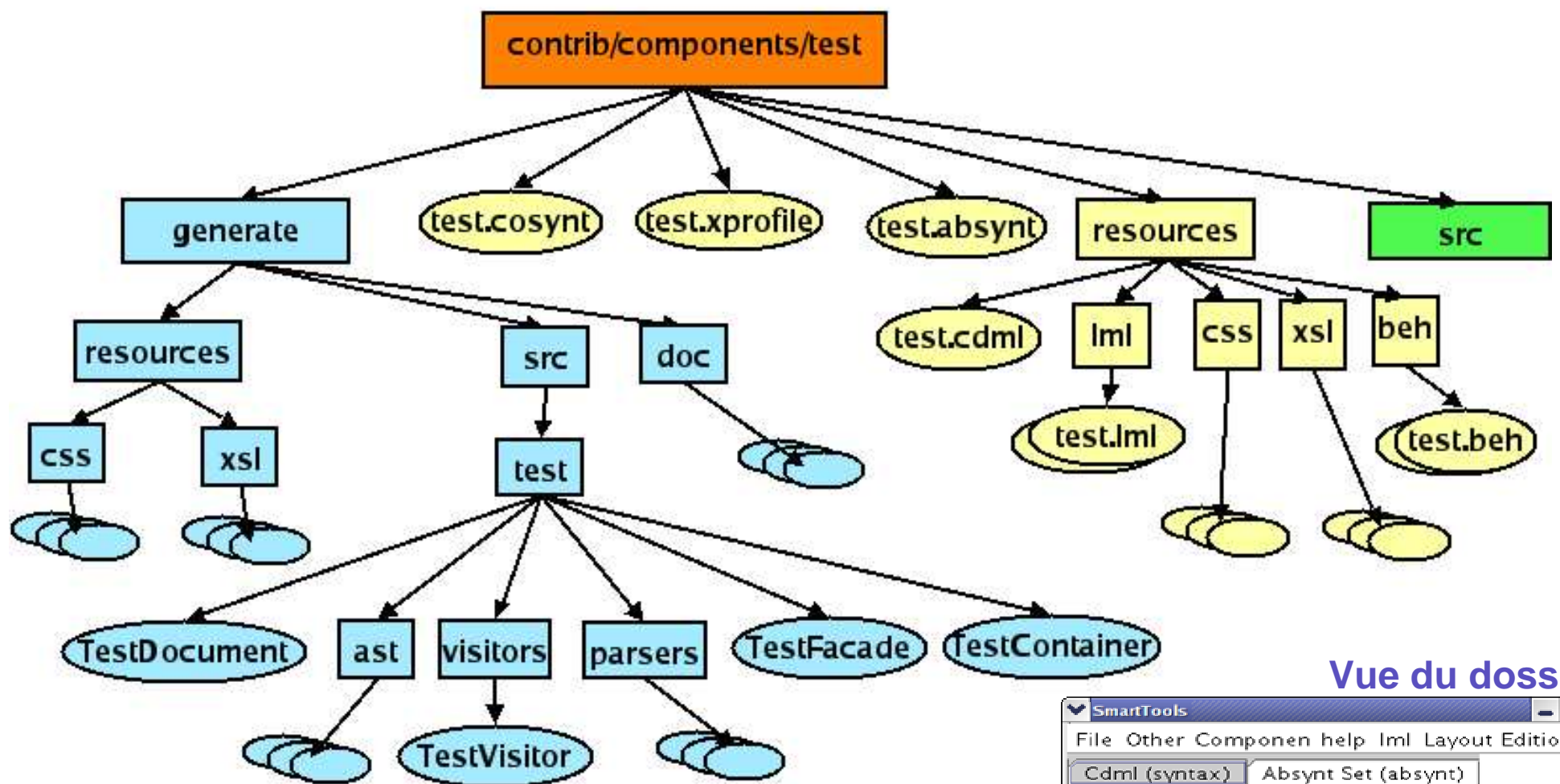
LML



Graphe des composants



Génération de 90% du code des applications



Vue du dossier

CDML du composant document de base

SmartTools

File Other Component help Iml Layout Edition Cdm

Introduction Cdm (syntax)

/home/dparigot/SmartTools1/SmartTools/src/core/fr/smarttools/core/document/resources/logicaldocument.cdm

Component: logicalDocument of type: document Extends: abstractContainer in NameSpace: fr.smarttools.core.doc

urlToSource : java.lang.String

Input: open (filename : java.lang.String)

Input: open

Input: copy

Input: cut

Input: paste

Input: insertSelection (where : java.lang.String)

Input: insert (insertXml : java.lang.String, where : java.lang.String, syntaxe : java.lang.String)

Input: modify (modify : java.lang.String, syntaxe : java.lang.String, attributes : java.util.HashMap)

Input: undo

Input: redo

Input: edition

Input: launchVisitor (visitor : java.lang.String, aspect : java.lang.String, debugViewer : java.lang.String)

Input: launchXSL (XSL : java.lang.String)

Input: save (filename : java.lang.String, xsl : java.lang.String)

Input: toString

Input: select (selection : fr.smarttools.core.util.selection.Selection)

Input: selectXPath (xpath : java.lang.String)

Input: lock (ock : fr.smarttools.core.util.selection.Selection)

Input: startEditionStruct (lml : java.lang.String)

Output: toXML

Output: consistencyError

Output: modification (modif : fr.smarttools.core.util.modification.ModificationsManag

Output: selection (selection : fr.smarttools.core.util.selection.Selection/**/)

Output: editionStructData (selectedNode : org.w3c.dom.Node/**/, selectedType : fr.

Output: (statusText : java.lang.String/**/)

Output: (statusText : java.lang.String/**/)

Output: requestView (requestedView : java.lang.String/**/, arguments : java.lang.Ha

Output: clipboardChange (content : org.w3c.dom.Node/**/)

Simple Selection current : 1.<height>

SmartTools commands

Dialog Box of Components Generator

name : properties

path : /oasis/dparigot/SmartTools1/contrib/components browse

type : document

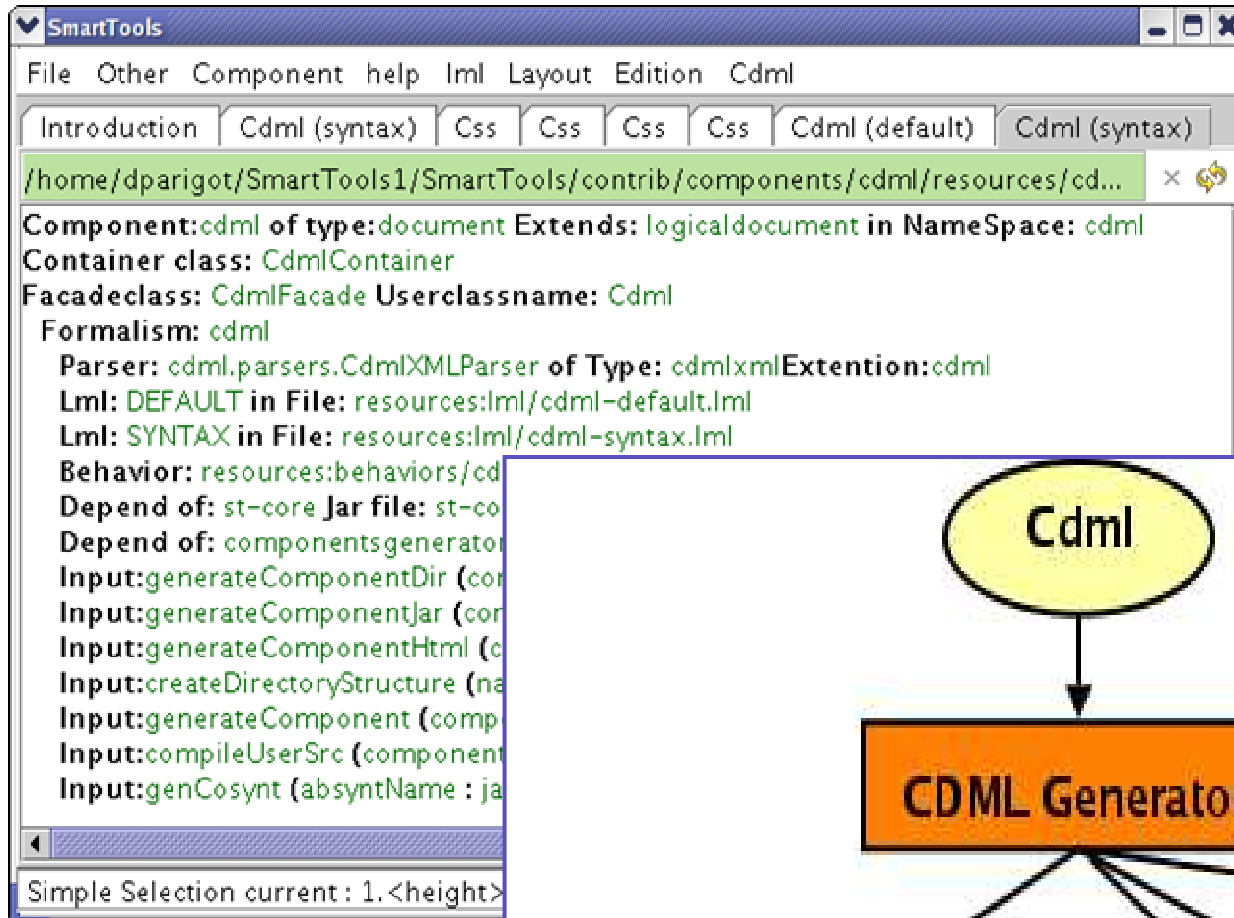
Simple Selection current : 1.2.1.2.

Simple Selection current : 1.<height>

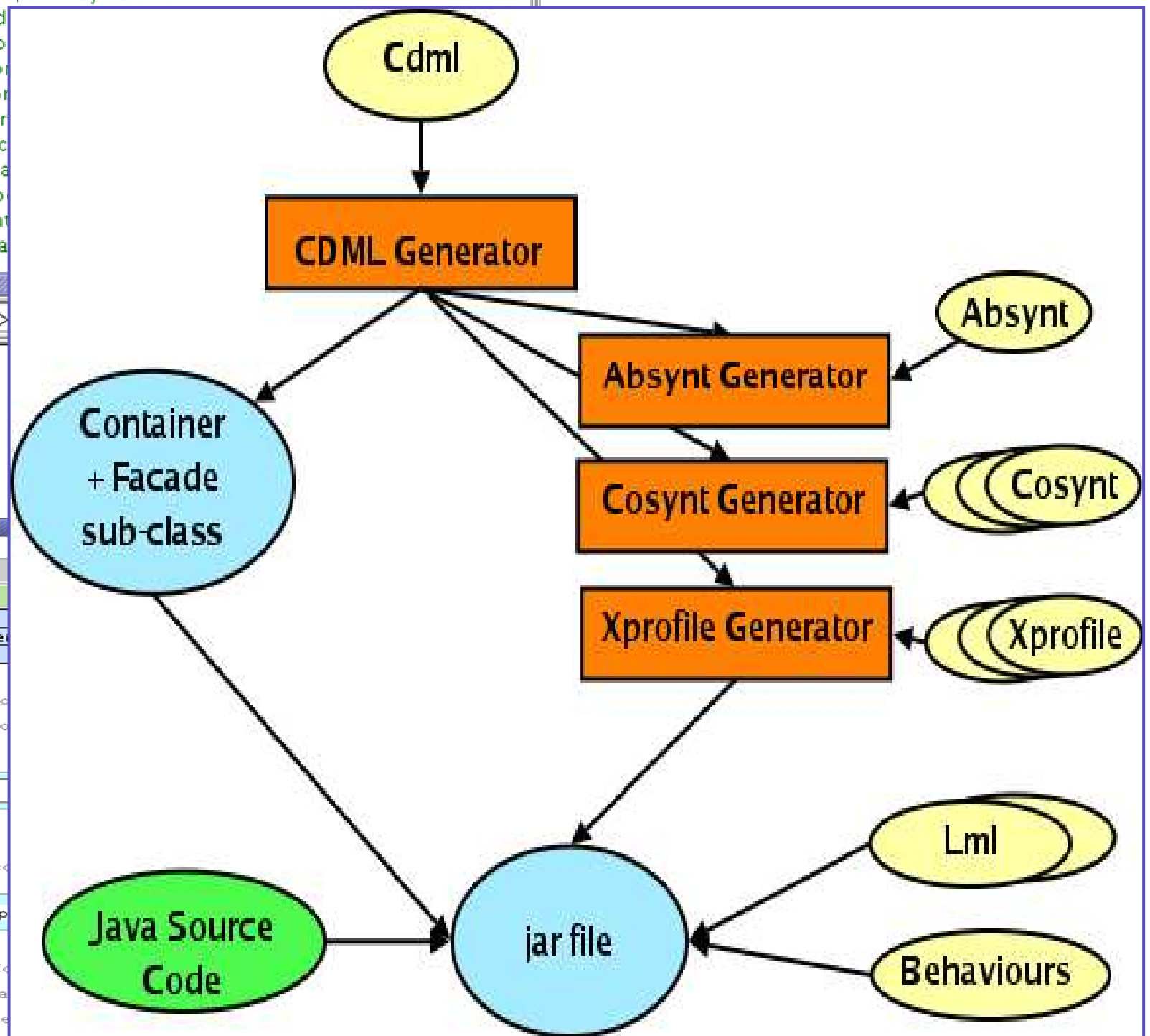
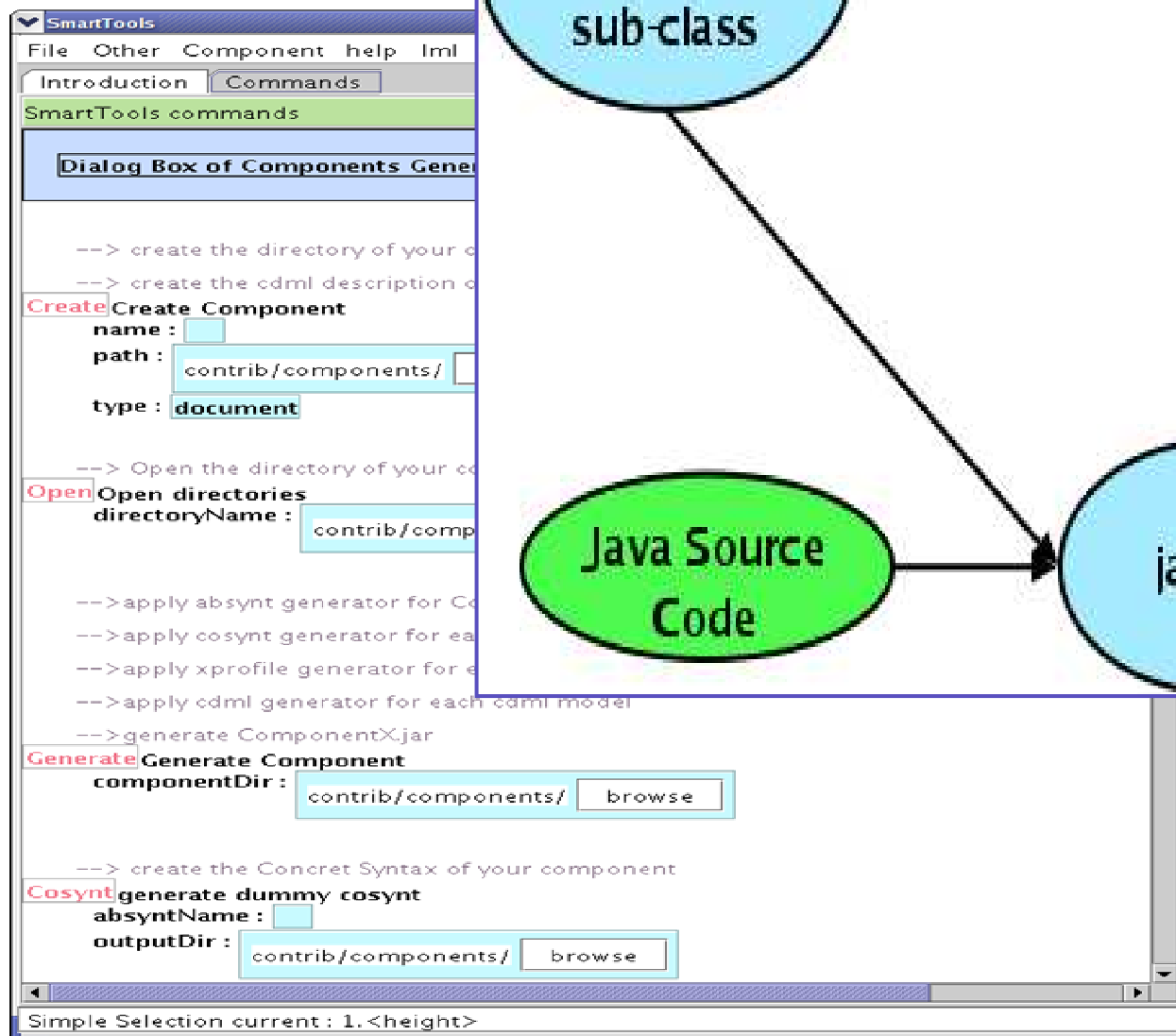
behaviors properties.cdm

Développement dirigé par les DSLs

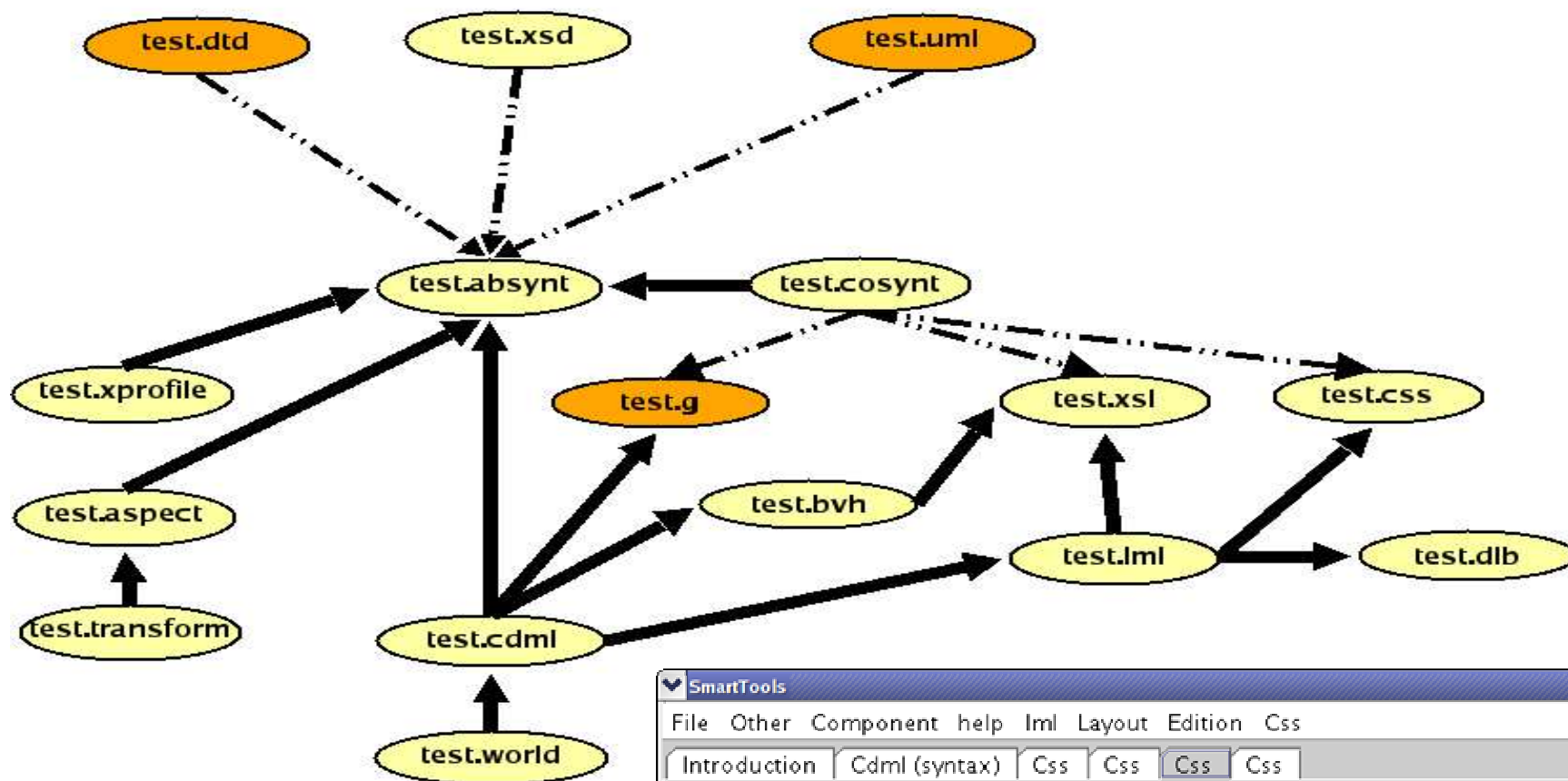
CDML de CDML



Boite de dialogue de CDML



Schémas de dépendance entre les DSLs



XSLT

```

SmartTools
File Other Component help Iml Layout Edition XSLT
Introduction Csx Csx XSLT
/home/dparigot/SmartTools1/SmartTools/src/compone...

'layout' ->
  apply template on"

'frame' ->
  Cframe(id="gframe", {
    value-of('./@title'), value-of('./@statusBar')
    call-method name="setSize"
    castclass="int"
    value-of('./@width')
    cast
    castclass="int"
    value-of('./@height')
    cast
    call-method
    propertyname="dynTabSwitch"
    value-of('./@dynTabSwitch')
    property
    for each in './**do {
      call-method name="addGtab" target="g"
      apply template on:"
      call-method
    };
  });

'set' ->
  Gtab(value-of('./@title'), apply template on"

'split' ->
  choose {
    when ('./@movedFrom and not(./@move

Simple Selection current : 1.<height>
    
```

Menus

```

SmartTools
File Other Component help Iml Layout Edition
Introduction XML (default) XML (default) Behaviors set (d...
/home/dparigot/SmartTools1/SmartTools/contrib/components/

<behaviors >
  <actions >
    <action name=Edition Struct>
    </action>
    <action name=Edition>
    </action>
    <action name=Editor>
    </action>
    <action name=XPath>
    </action>
    <action name=Traversal + DebugAspect>
    </action>
    <action name=Open View>
    </action>
    <action name=Generate Component Dir>
    </action>
    <action name=Generate Component Jar>
    </action>
    <action name=Generate Component Html>
    </action>
    <action name=Create Component>
    </action>
    <action name=Reopen File>
    </action>
    <action name=Save HTML>
      <msg name=save>
        <msgattr chooser=fr.smarttools.core.util.chooser.S
          name=filename
          value=
        <msgattr name=xsl
          value=resources:xsl/cdmlhtml-text.xsl/>
      </msg>
    </action>
  </actions >
</behaviors >
    
```

CSS

```

SmartTools
File Other Component help Iml Layout Edition Csx
Introduction Cdml (syntax) Csx Csx Csx Csx
/home/dparigot/SmartTools1/SmartTools/contrib/components/cdml/resources/css/cdml-st...
component {display : block; color : rgb(0,0,0); font-family : courier; background-color : #FFFFFF; }
component.current {background-color : rgb(100,220,255); }
component.xpath {background-color : rgb(200,0,255); }
childbox, childbox.current, childbox.visitor {margin-left : 10; }
input, output, inout {color : rgb(250,0,110); }
attribute {color : rgb(0,110,0); }
line, line.current, line.visitor {display
keyword {color : rgb(0,0,0); font-weig
null {color : rgb(0,110,0); }
    
```

```

SmartTools
File Other Component help Iml Layout Edition
Cdml (syntax) Cdml (default) Lml Set (default) Absynt Set DialogBox Set (d...
/home/dparigot/SmartTools1/SmartTools/demos/dialogbox/demo.dlb

List of SmartTools Demonstrations

doc Produce a Html or Xml document from text format
requestedFilename : file:demos/doc/example.doc
rawweb Produce a Html file with the DTD definition of Recherche Report INRIA
requestedFilename : file:demos/rawweb/litle.rawweb
cosynt Produce a graphic view and a text format from a logical format
requestedFilename : file:contrib/components/doc/docpre.cosynt
cdml Produce a documentation or a dialog box for a component model
requestedFilename : file:contrib/components/cdml/resources/cdml.cdml
cm Produce a script command to generate a graphic view (component manager mo
requestedFilename : file:demos/world/stworld
lml Produce a Graphic User Interface from an Xml description (Layout Meta Langua
requestedFilename : file:demos/lml/resources/lml/demo.lml
xml Read any Xml file (Generic Xml)
requestedFilename : file:demos/st.xml
Debug Use log component and graph component (component manager)
id_src : ComponentsManager
type_dest : worldStateGraph
user Start user database
componentId : stUsersDb
componentType : stUsersDb
Start Data Base Start database component
componentId : db
componentType : db
db Use database component
id_src : db
id_dest : stUsersDb
behaviors Open behaviors model (menu and tool bar)
requestedFilename : file:demos/world/doc-behaviors.behaviors
ant Open Ant file and execute target
requestedFilename : file:demos/ant/build-user.xmlant
css Open css file
requestedFilename : file:demos/generalresources/videoresources/css/xml-st
xslt Open Xslt file
requestedFilename : file:demos/generalresources/videoresources/xsl/generic
dtd Import Dtd file
requestedFilename : file:demos/rawweb/rawweb.dtd
uml Import Uml file (various meta-models)

Simple Selection current : 1.<height>
    
```

Les fortes évolutions de l'informatique

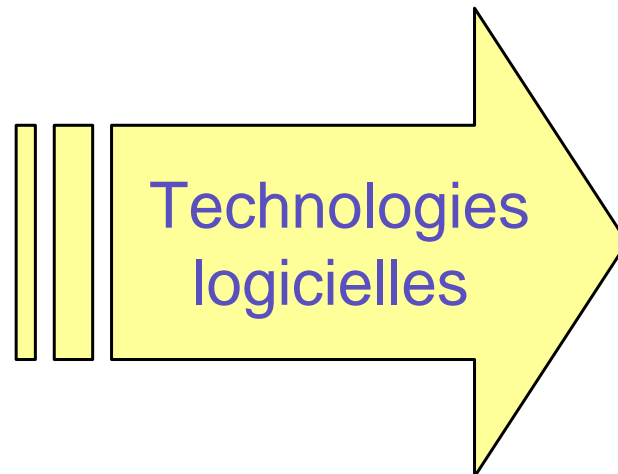
Les applications d'hier



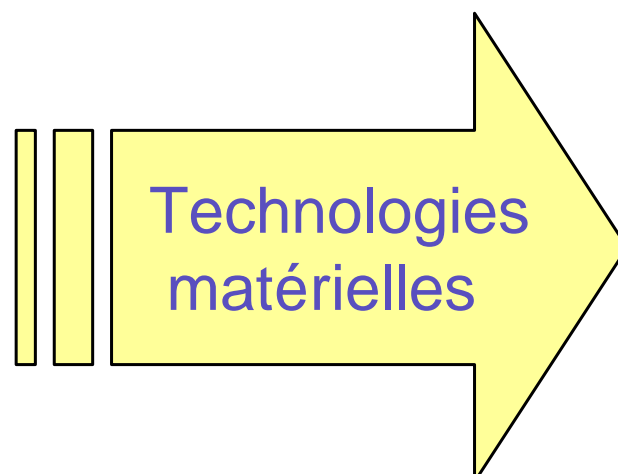
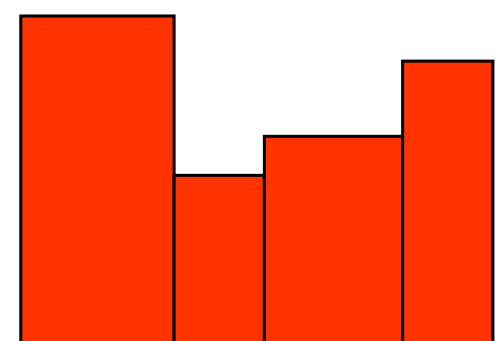
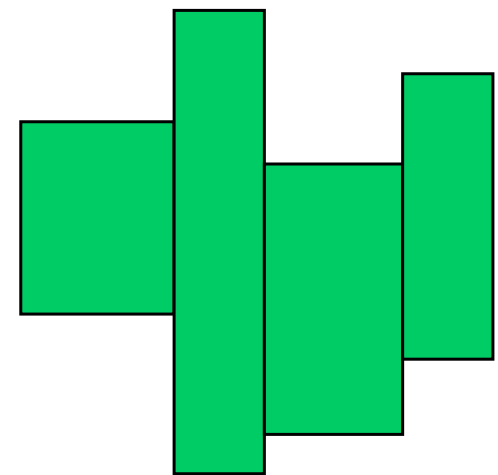
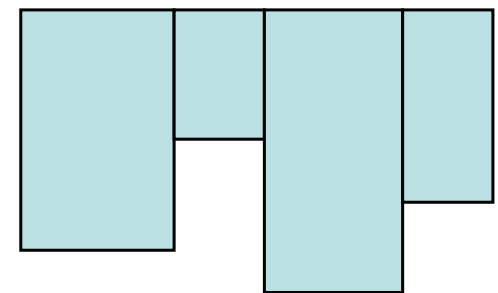
langages



exécution



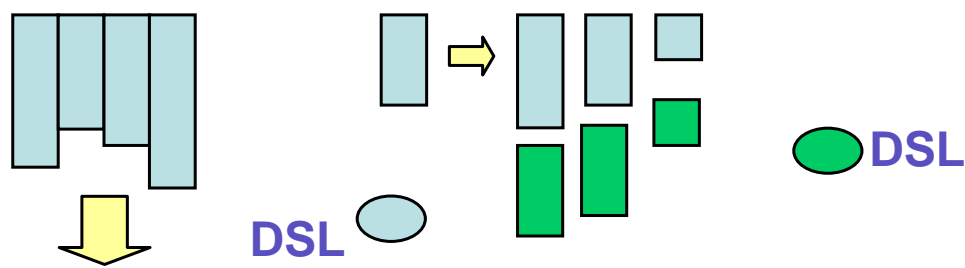
Les applications de demain



Les langages de programmation ont évolué beaucoup moins vite

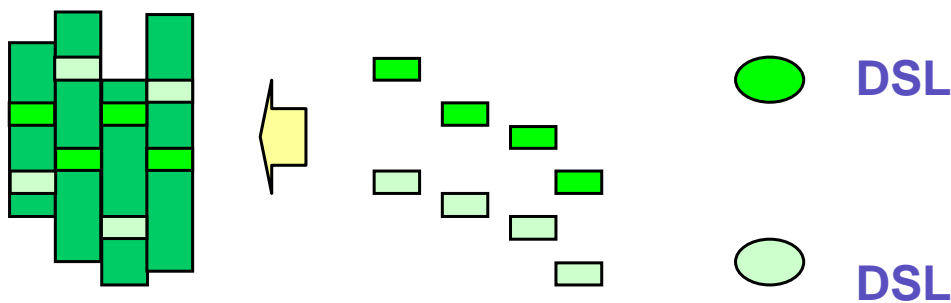
Les quatre concepts

Programmation dirigée par des DSLs



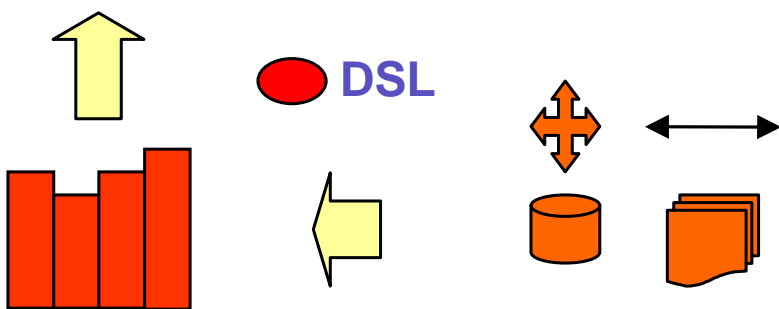
MDA
Ingénierie Dirigée
par les Modèles

Programmation par préoccupations



AOP
Programmation
par aspects

Programmation par composants



SOA
Architecture Orientée
Services

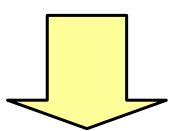
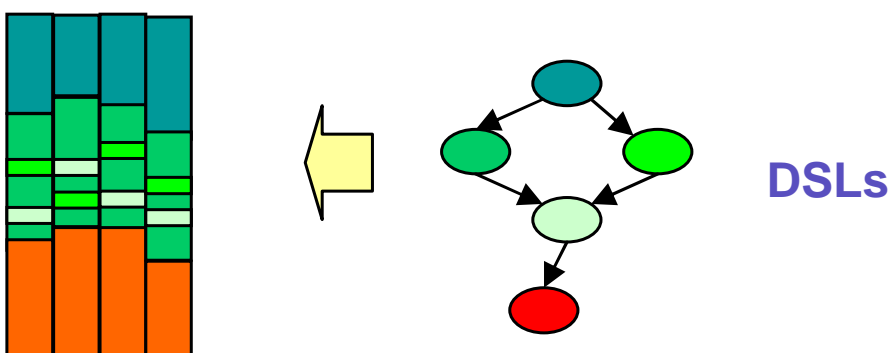


Schéma d'assemblage des DSLs



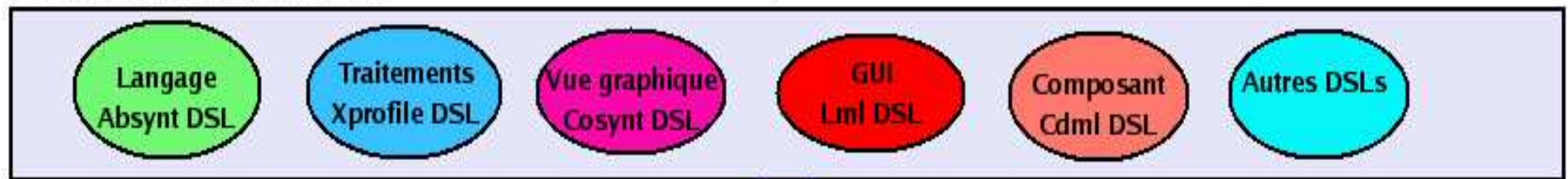
FABRIQUE LOGICIELLE

Développement en quatre phases

Les besoins des applications

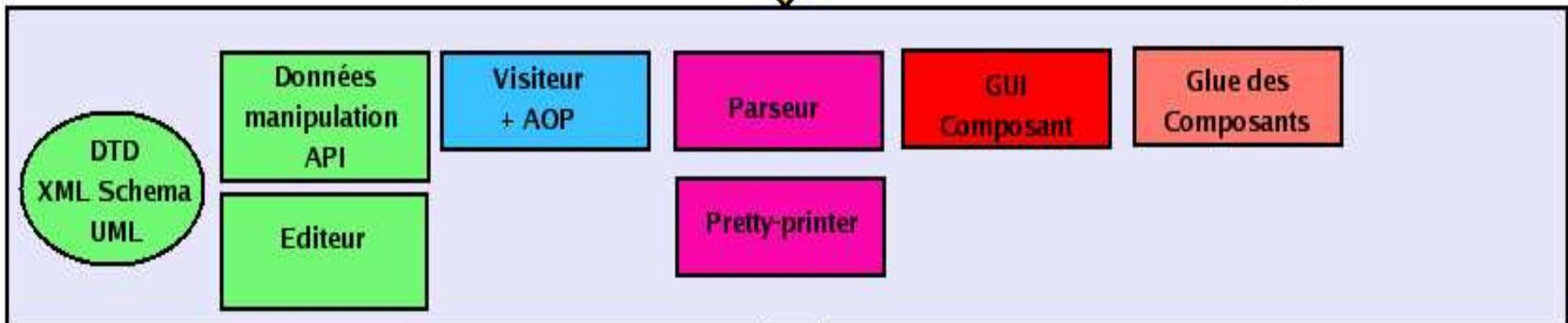
DSL de SmartTools

Phase de spécification



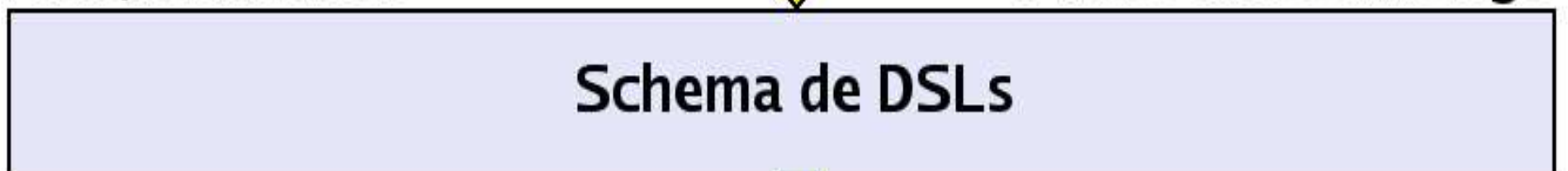
Générateurs de SmartTools

Phase de génération



Fabrique SmartTools

Phase d'assemblage



Génération de 90 %
du code

Composant

Phase d'intégration

Composant

Composant

Intégration dans une plate-forme

Ligne de produits

Smarttools

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA
SOPHIA ANTIPOLIS

La programmation de demain sera :

Tout par composants

Tout par assemblage dynamique

Tout par langages, abstraction

- DSL métiers

Tout par transformations

- Transformation de modèles

Tout par analogies

- Projection des traitements

Tout par séparation des préoccupations

- Une programmation pointilliste

Une programmation incrémentale

- Spécification – Implémentation

Une macro programmation

- Systèmes complexes, évolutifs, multipartenaires

Les programmes ne seront plus manipulés
directement par leurs interfaces (API)

Les facteurs de ce bouleversement

Fortes évolutions :

- Technologies logicielles
- Technologies matérielles
- Communauté d'utilisateurs

Nouveaux besoins et exigences :

- Automatiser le développement
- Réduire le temps du développement
- Améliorer la qualité du logiciel
- Augmenter le retour d'investissement
- Intégrer rapidement les nouvelles technologies

Principaux obstacles :

- Évolution des langages de programmation
- L'existant des systèmes informatiques
- La formation

Nécessité d'agir vite

- Mouvement international grâce aux Consortiums
- Marché en plein expansion (e-commerce)

Les enjeux économiques

- Les grands de l'informatique vont nous imposer leurs standards et outils

Nature des engagements

- De la matière grise (du logiciel)
- Un retour d'investissement rapide possible
- Solution envisageable/raisonnable à très court terme !