INSTALLATION

- Get a grip on one of the USB sticks.
- Install the Eclipse SDK and the Android SDK matching your platform from the memory stick.
- Start Eclipse and set the location of the Android SDK in *Preferences > Android*
- Import the *android_workspace.zip* into your workspace using *File > Import > Existing Projects into Workspace > Archive File*

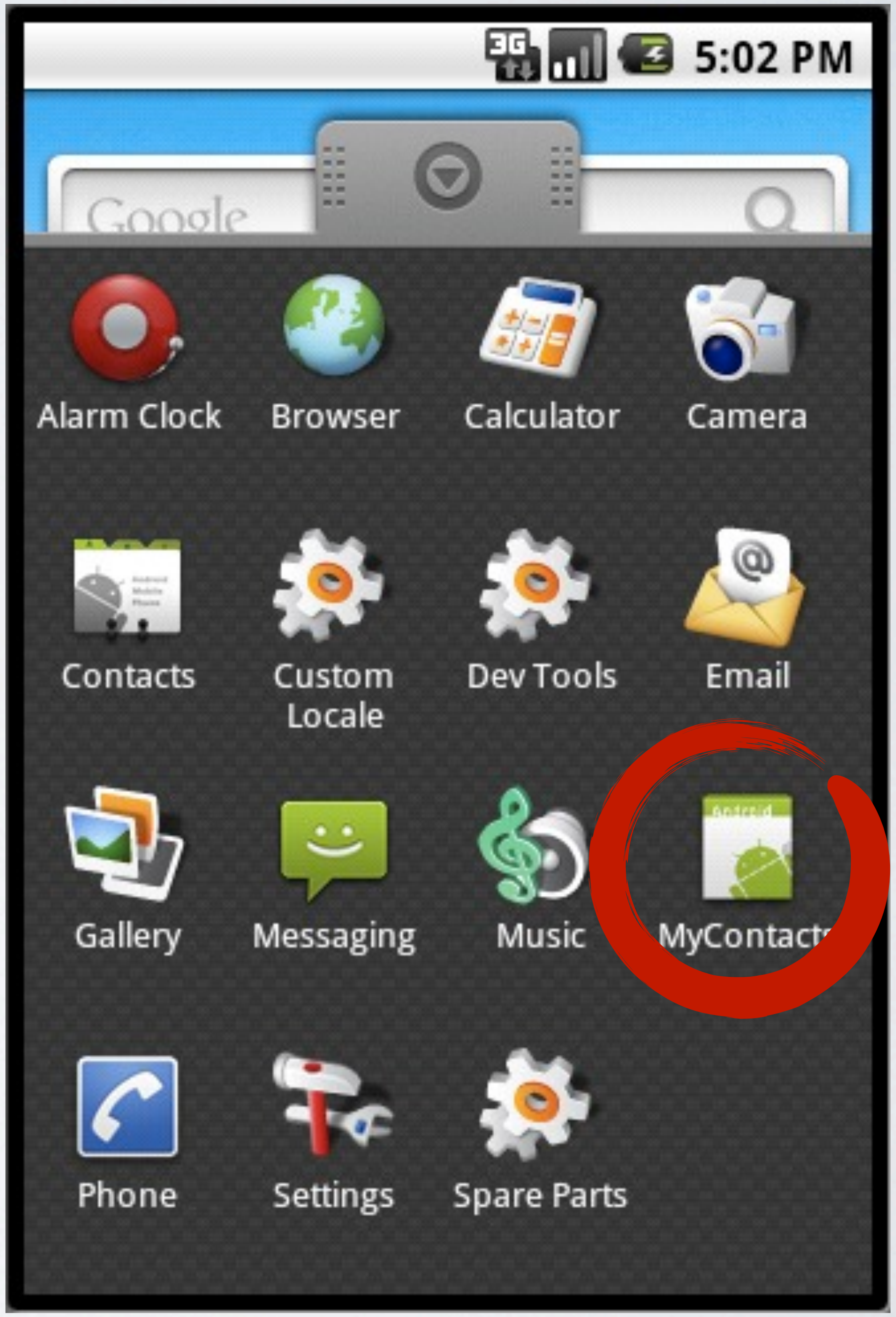
CREATING A LANGUAGE FOR ANDROID APPS WITH ECLIPSE MODELING

Mikaël Barbero, Stéphane Begaudeau - Obeo
Jan Köhnlein, Holger Schill, Dennis Hübner - itemis

APP CONCEPTS

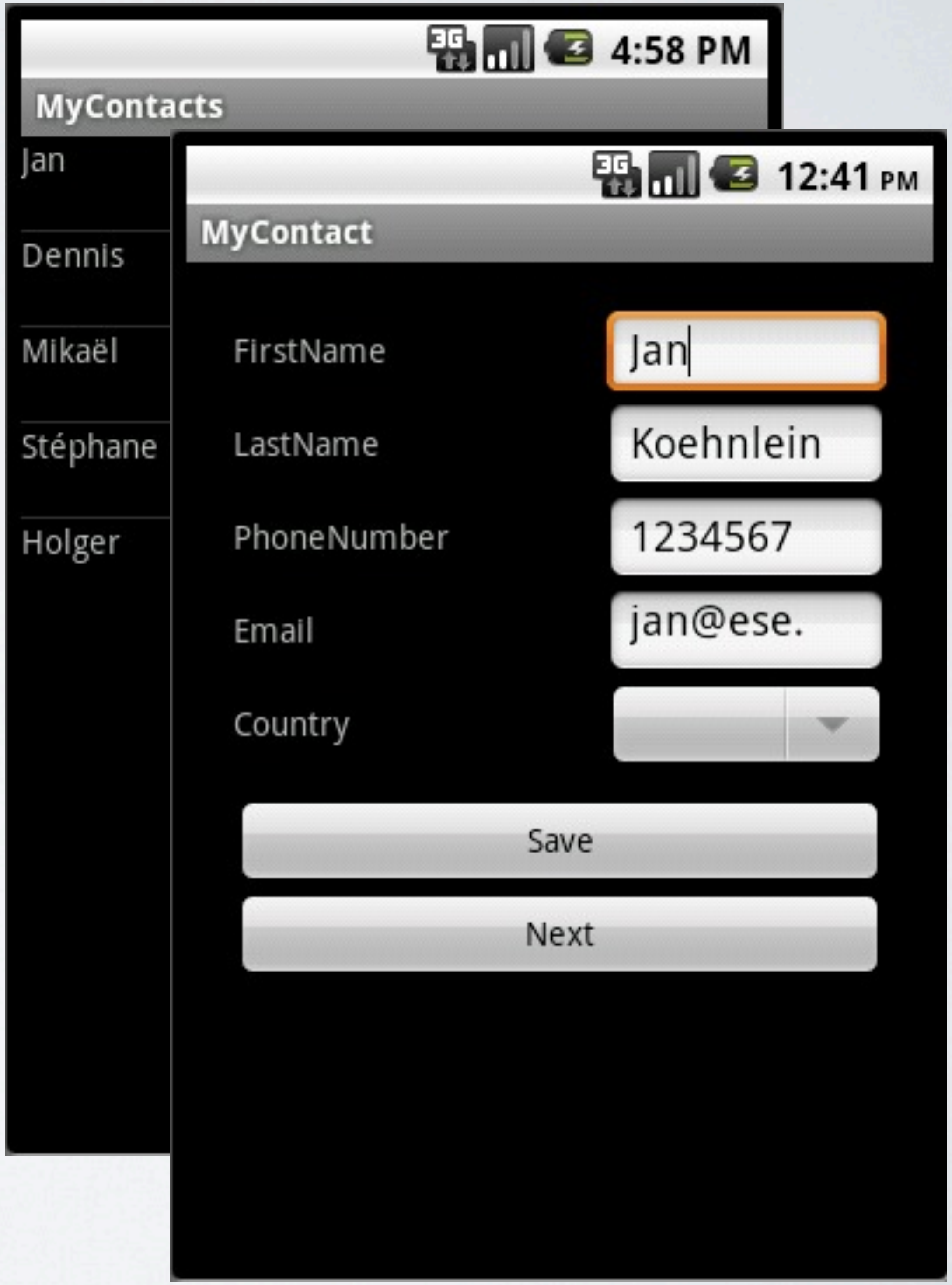
APPLICATION

- Declared in AndroidManifest.xml
- Defines Activities
    - One default Activity



ACTIVITY

- One page of an App
- Java class extends android.app.Activity
- Entry in AndroidManifest.xml
- Layout in separate XML file



4:58 PM

MyContacts

Jan

Dennis

Mikaël

Stéphane

Holger

12:41 PM

MyContact

FirstName: Jan

LastName: Koehnlein

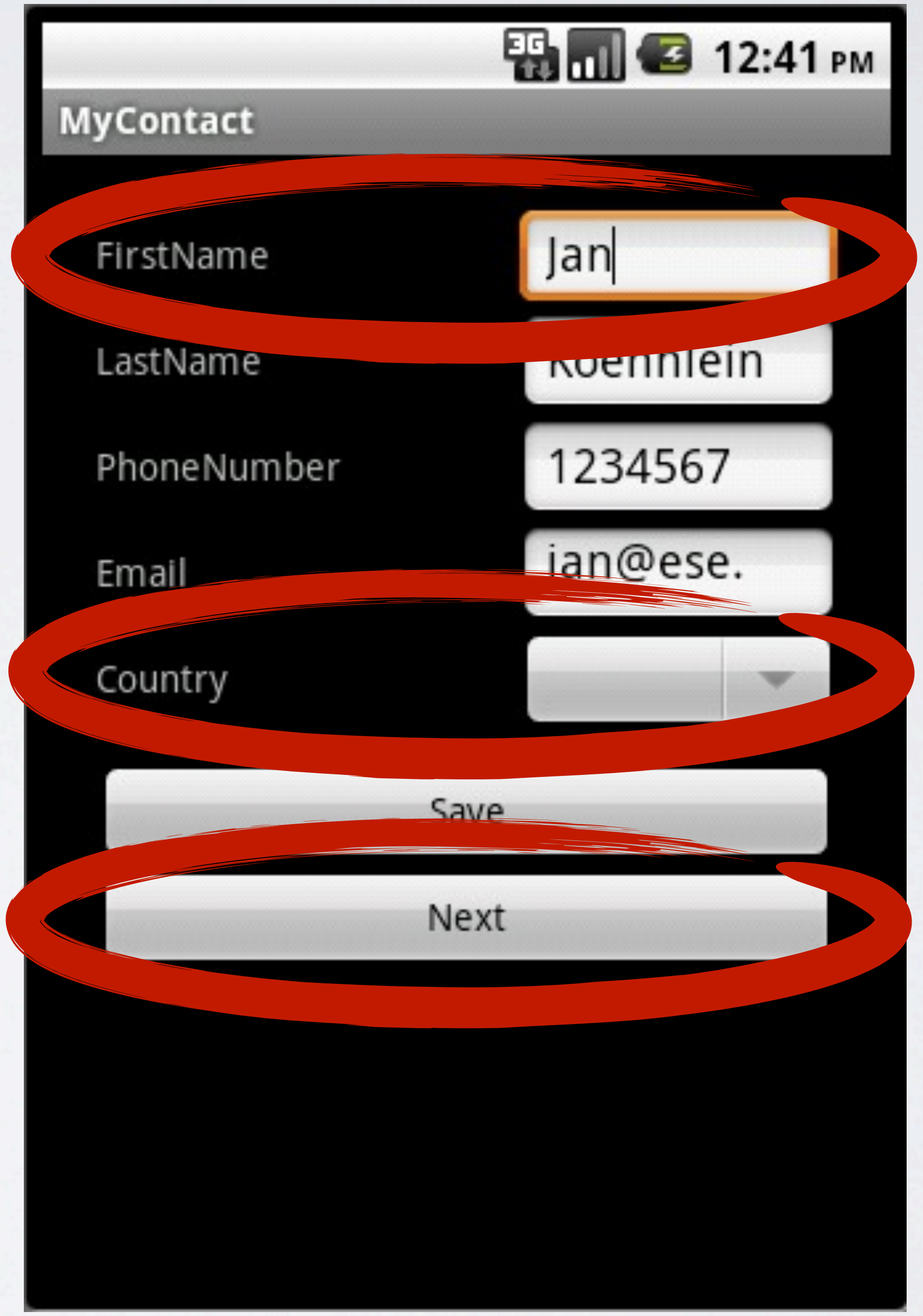
PhoneNumber: 1234567

Email: jan@ese.

Country:

WIDGETS

- Member in activity class
- Layout in XML
- Refer to DB entry



MyContact

FirstName: Jan

LastName: Koehnlein

PhoneNumber: 1234567

Email: jan@ese.

Country

Text

Spinner

Link

PROBLEM

- Concepts span across XML as well as Java code

- No common abstraction mechanism

DSL SOLUTION

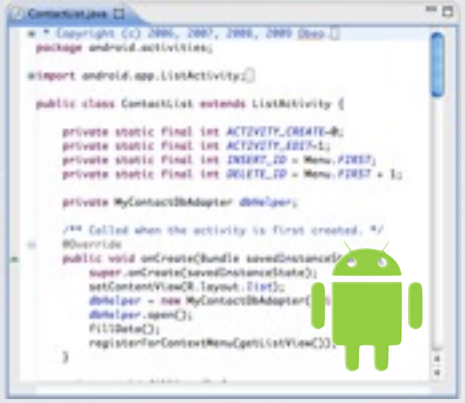
- Define concepts in an Ecore model

- Use Xtext to create an editor for model instances

- Use Acceleo to generate code from model instances

LANGUAGE SETUP

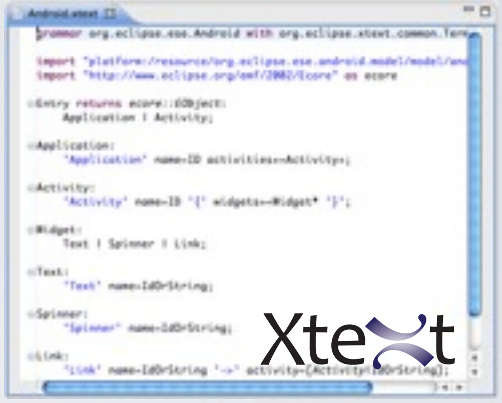
Reference



```
* Copyright (c) 2006, 2007, 2008, 2009 Obeo.
package android.activities;

import android.app.ListActivity;

public class ContactList extends ListActivity {

    private static final int ACTIVITY_CREATE=0;
    private static final int ACTIVITY_EDIT=1;
    private static final int INSERT_ID = Menu.FIRST;
    private static final int DELETE_ID = Menu.FIRST + 1;

    private MyContactDbAdapter dbHelper;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceSt
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list);
        dbHelper = new MyContactDbAdapter(
        dbHelper.open();
        fillData();
        registerForContextMenu(getListView());
    }
```

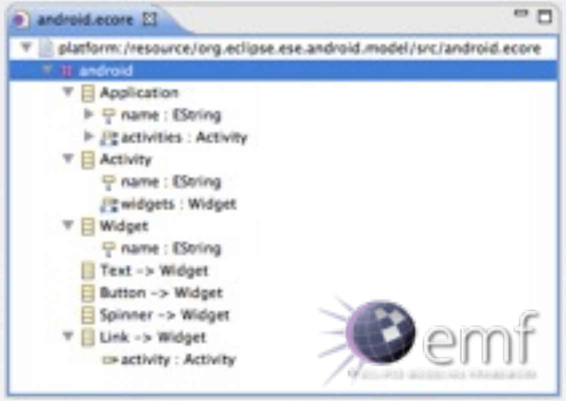
derive

Xtext Grammar



```
grammar org.eclipse.ese.Android with org.eclipse.xtext.common.Term

import "platform:/resource/org.eclipse.ese.android.model/model/and
import "http://www.eclipse.org/emf/2002/Ecore" as ecore

Entry returns ecore::EObject:
    Application | Activity;

Application:
    'Application' name=ID activities+=Activity;

Activity:
    'Activity' name=ID '{' widgets+=Widget* '}';

Widget:
    Text | Spinner | Link;

Text:
    'Text' name=IdOrString;

Spinner:
    'Spinner' name=IdOrString;

Link:
    'Link' name=IdOrString '->' activity=[Activity|IdOrString];
```

derive

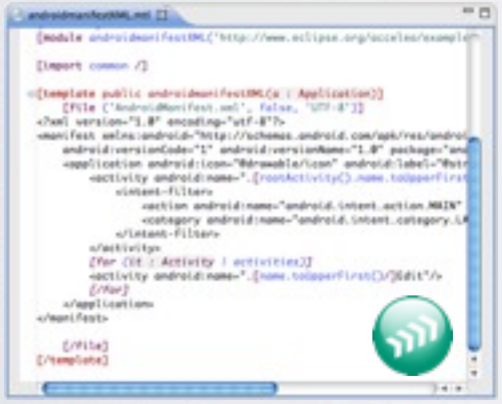


```
platform:/resource/org.eclipse.ese.android.model/src/android.ecore
  android
    Application
      name : EString
      activities : Activity
    Activity
      name : EString
      widgets : Widget
    Widget
      name : EString
    Text -> Widget
    Button -> Widget
    Spinner -> Widget
    Link -> Widget
      activity : Activity
```

Ecore Model

derive

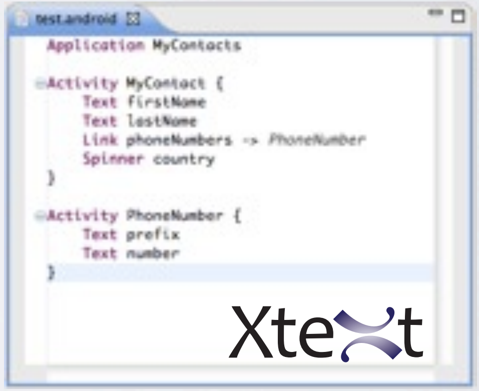
generate



```
[module androidmanifestXML('http://www.eclipse.org/acceleo/example

[import common /]

[template public androidmanifestXML(a : Application)]
    [file ('AndroidManifest.xml', false, 'UTF-8')]
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/andro
    android:versionCode="1" android:versionName="1.0" package="and
    <application android:icon="@drawable/icon" android:label="@str
        <activity android:name=".[rootActivity().name.toUpperFirst
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
                <category android:name="android.intent.category.LA
            </intent-filter>
        </activity>
        [for (it : Activity | activities)]
        <activity android:name=".[name.toUpperFirst()/]Edit"/>
        [/for]
    </application>
</manifest>

    [/file]
[/template]
```

derive

Acceleo Templates

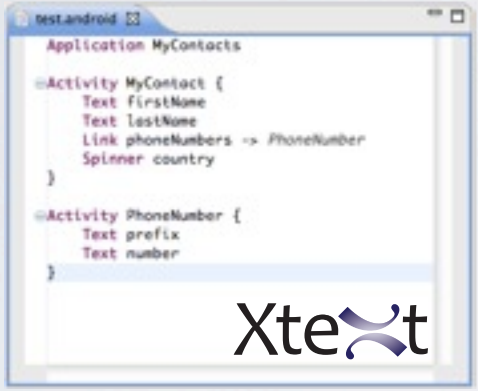


```
Application MyContacts

Activity MyContact {
    Text firstName
    Text lastName
    Link phoneNumbers -> PhoneNumber
    Spinner country
}

Activity PhoneNumber {
    Text prefix
    Text number
}
```

DSL Editor

CODE GENERATION

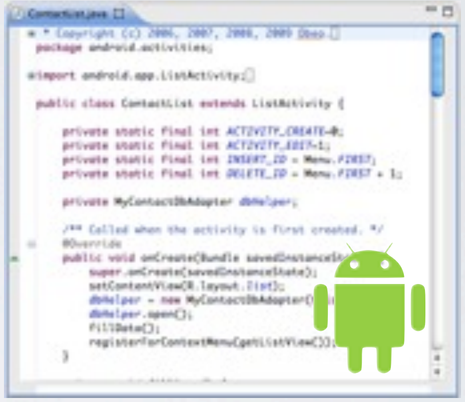
Android Model



```
Application MyContacts

Activity MyContact {
    Text firstName
    Text lastName
    Link phoneNumbers -> PhoneNumber
    Spinner country
}

Activity PhoneNumber {
    Text prefix
    Text number
}
```

Acceleo Generator

Android App



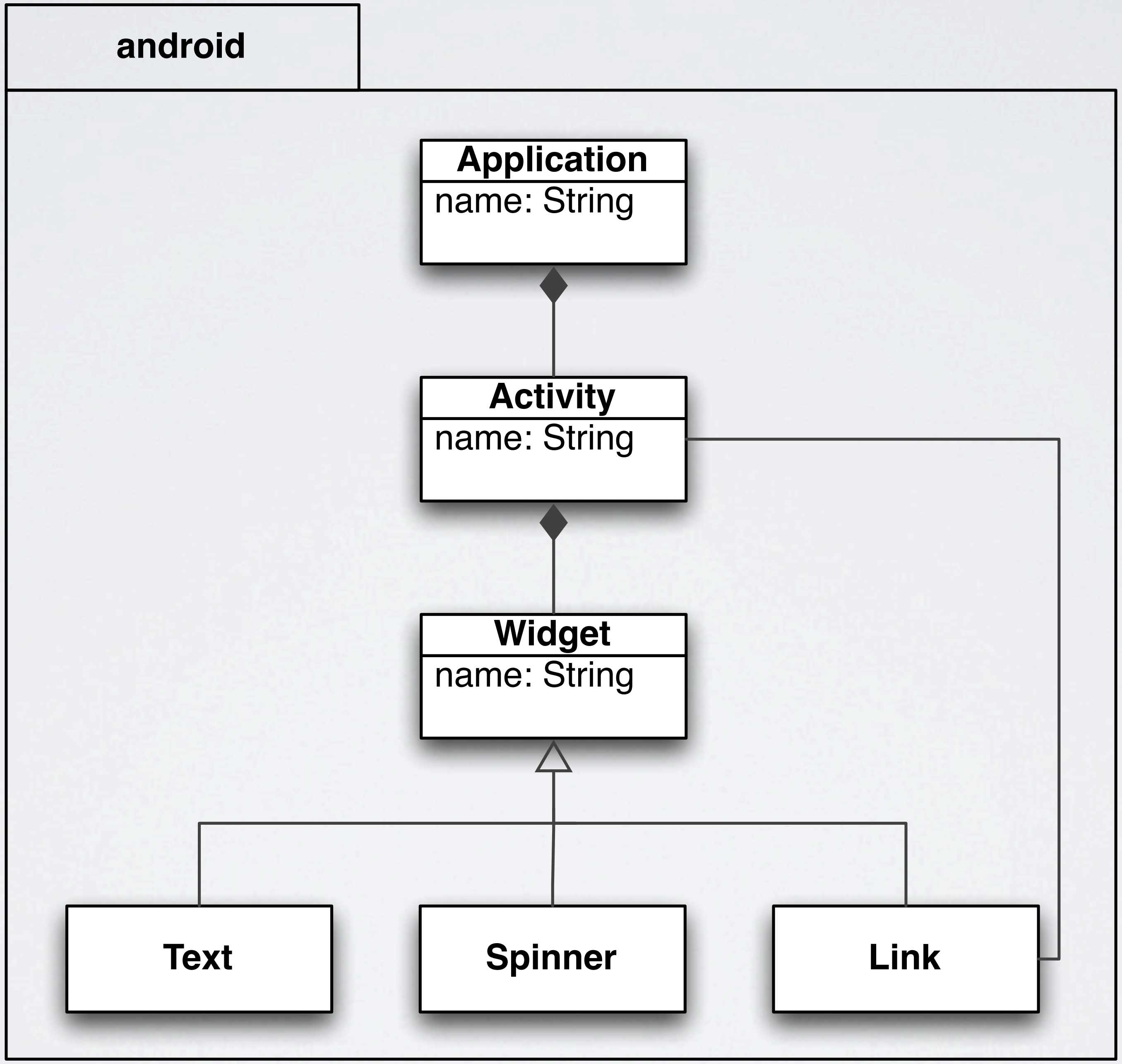
```
 * Copyright (c) 2006, 2007, 2008, 2009 Open
 package android.activities;

import android.app.ListActivity;

 public class ContactList extends ListActivity {

     private static final int ACTIVITY_CREATE=0;
     private static final int ACTIVITY_EDIT=1;
     private static final int INSERT_ID = Menu.FIRST;
     private static final int DELETE_ID = Menu.FIRST + 1;

     private MyContactDbAdapter dbHelper;

     /** Called when the activity is first created. */
     @Override
     public void onCreate(Bundle savedInstanceSt
         super.onCreate(savedInstanceState);
         setContentView(R.layout.list);
         dbHelper = new MyContactDbAdapter(
         dbHelper.open();
         fillData();
         registerForContextMenu(getListView());
     }
```

emf

ECLIPSE MODELING FRAMEWORK

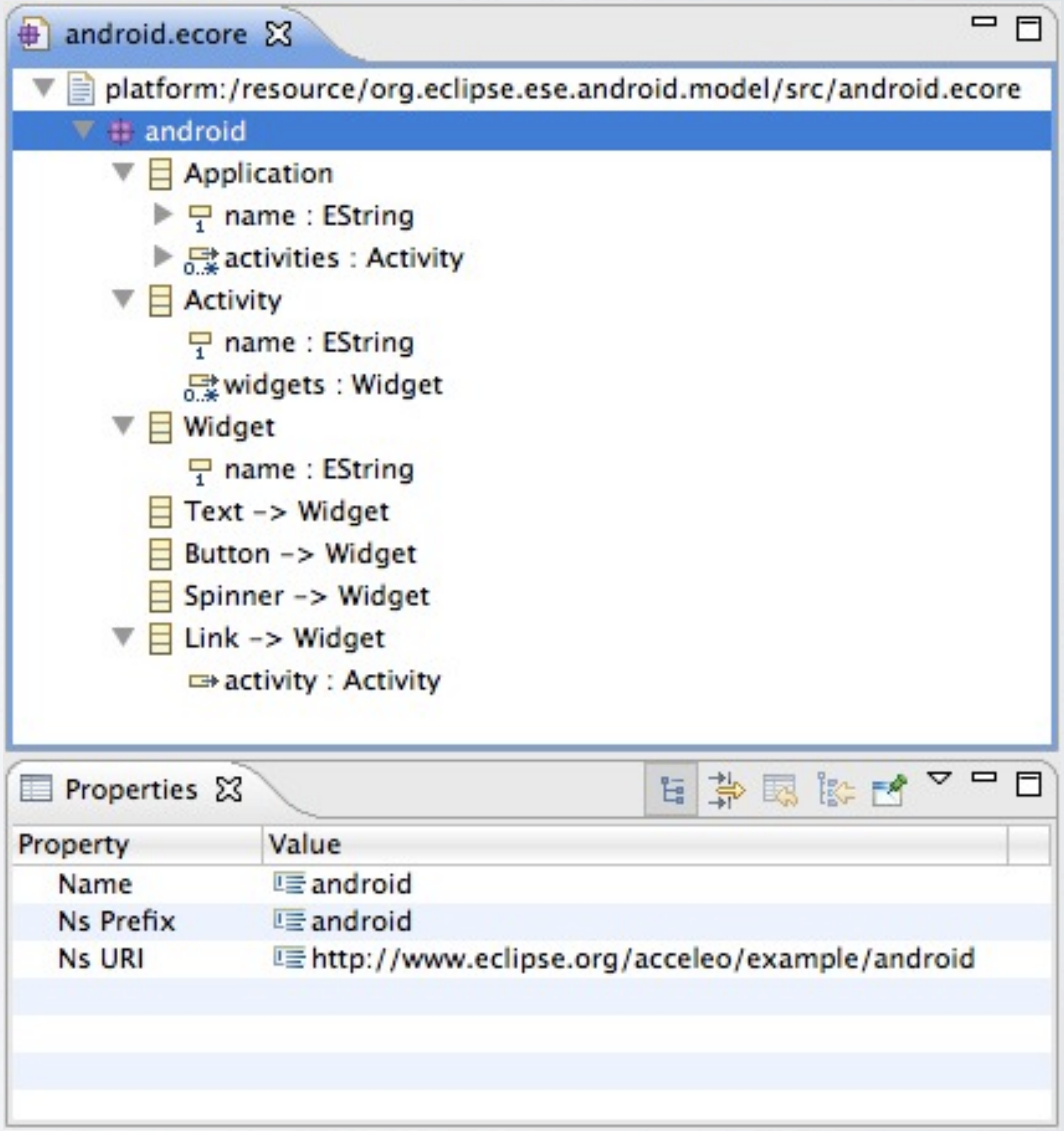
THE MODEL

THE MODEL


android
Application
name: String
Activity
name: String
Widget
name: String
Text
Spinner
Link

ECORE CONCEPTS

- EPackage
- EClass
- EAttribute
- EReference
  - (non-)containment



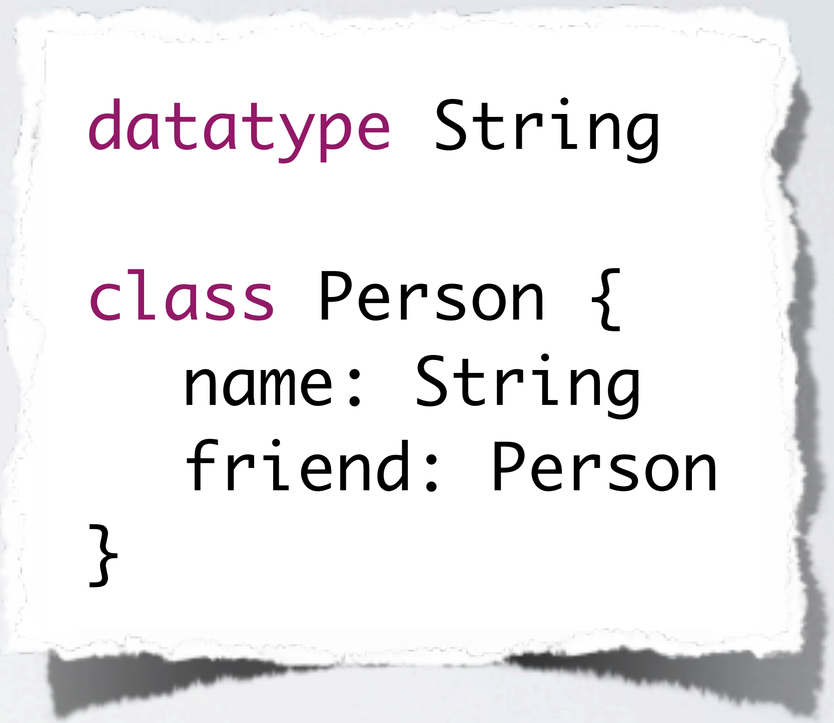
```
▼ platform:/resource/org.eclipse.ese.android.model/src/android.ecore
  ▼ android
    ▼ Application
      ▶ name : EString
      ▶ activities : Activity
    ▼ Activity
        name : EString
        widgets : Widget
    ▼ Widget
        name : EString
      Text -> Widget
      Button -> Widget
      Spinner -> Widget
    ▼ Link -> Widget
        activity : Activity
```

Property	Value
Name	android
Ns Prefix	android
Ns URI	http://www.eclipse.org/acceleo/example/android

Xtext

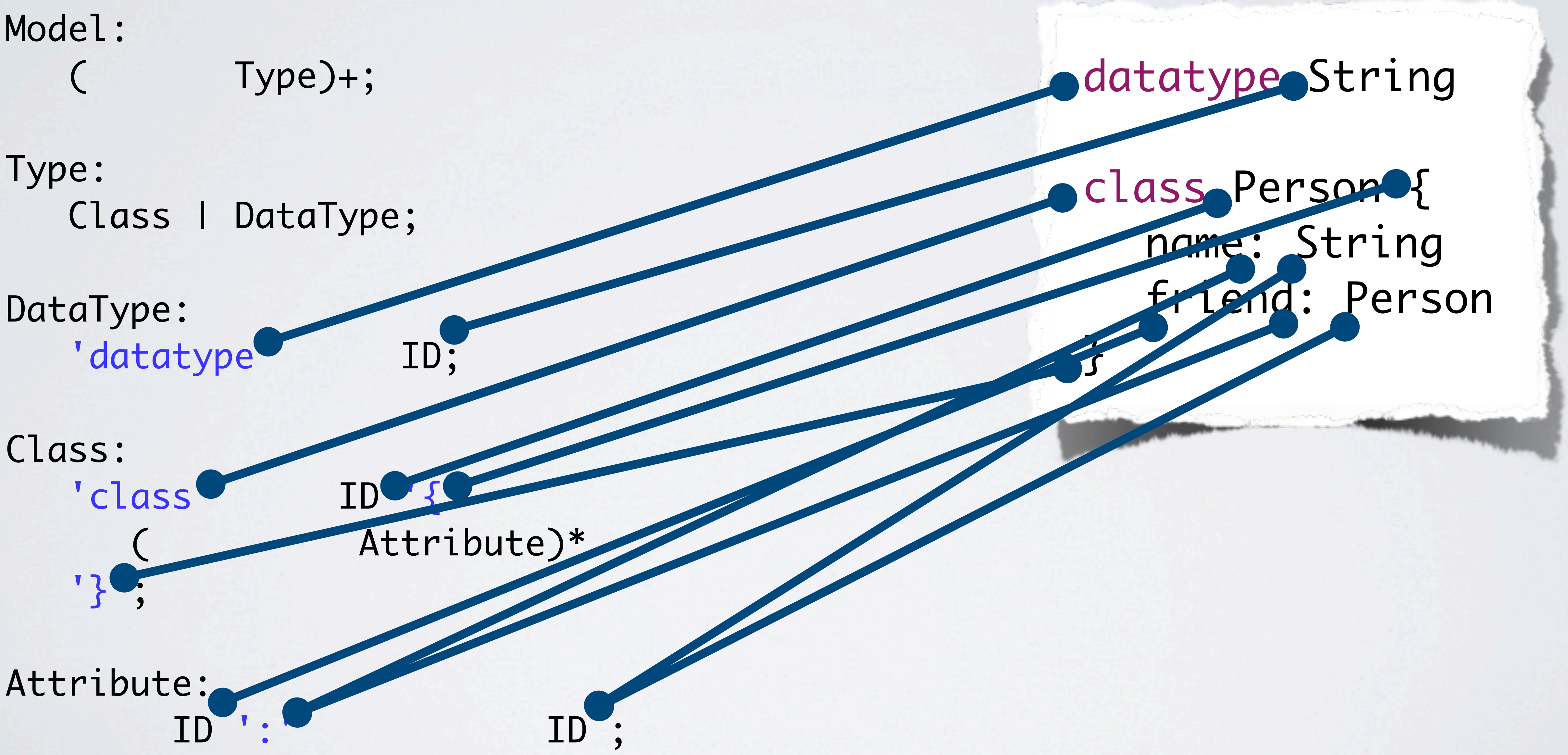
TEXTUAL MODELING

```
grammar org.xtext.workshop.DomainModel
   with org.eclipse.xtext.common.Terminals

import
   "http://www.xtext.org/workshop/DomainModel"

Model:
   (types+=Type)+;

Type:
   Class | DataType;

DataType:
   'datatype' name=ID;

Class:
   'class' name=ID '{'
      (features+=Attribute)*
   '}';

Attribute:
   name=ID ':' type=[Type|ID];
```



```
datatype String

class Person {
   name: String
   friend: Person
}
```

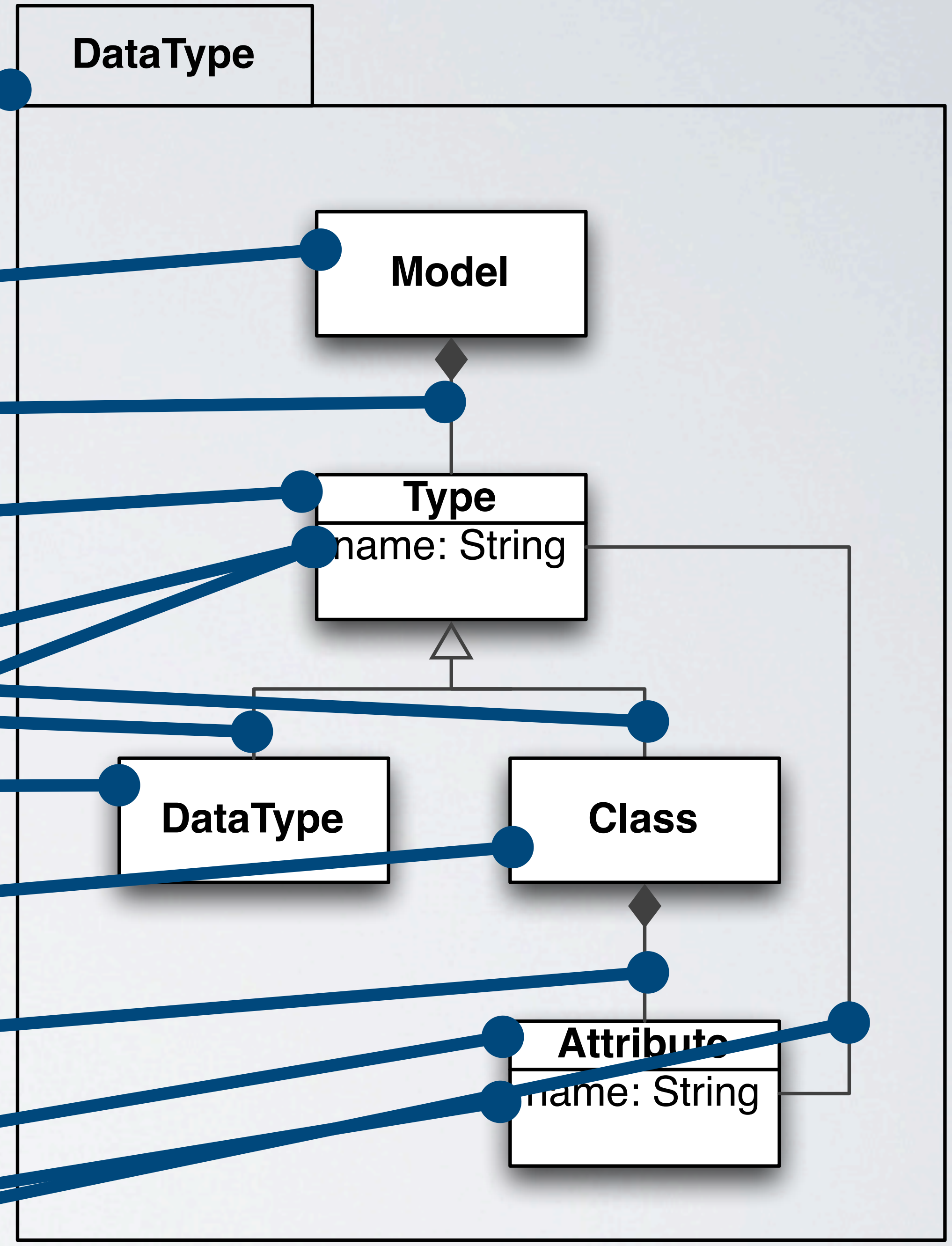
```
grammar org.xtext.workshop.DomainModel
   with org.eclipse.xtext.common.Terminals
```



```
Model:
   (        Type)+;

Type:
   Class | DataType;

DataType:
   'datatype'       ID;

Class:
   'class'       ID '{'
     (         Attribute)*
   '}';

Attribute:
        ID ':'           ID ;
```

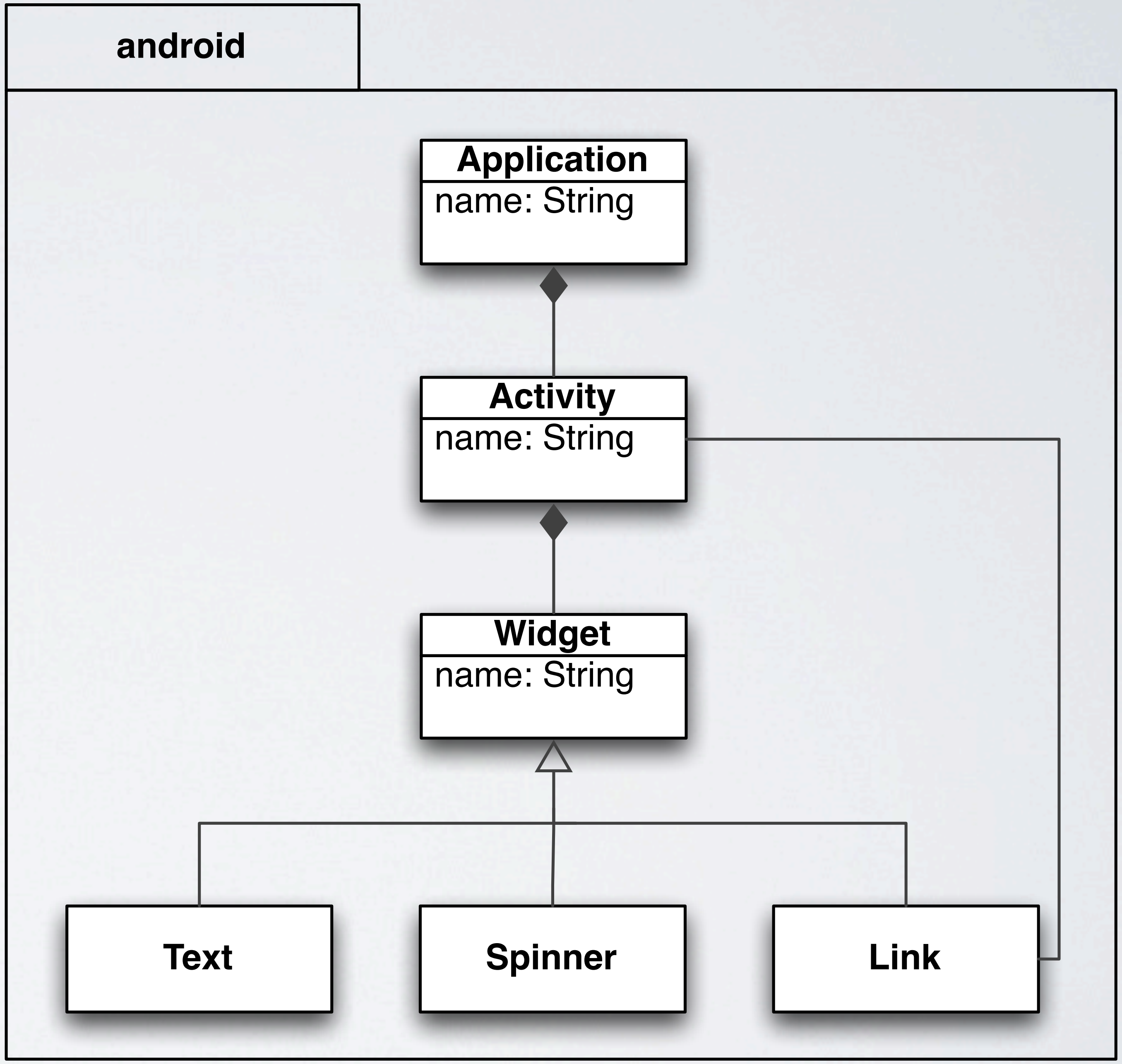
```
datatype String

class Person {
   name: String
   friend: Person
}
```

```
grammar org.xtext.workshop.DomainModel
   with org.eclipse.xtext.common.Terminals

import
   "http://www.xtext.org/workshop/DomainModel"

Model:
   (types+=Type)+;

Type:
   Class | DataType;

DataType:
   'datatype' name=ID;

Class:
   'class' name=ID '{'
      (features+=Attribute)*
   '}';

Attribute:
   name=ID ':' type=[Type|ID];
```



REFERENCE MODEL

```
Application MyContacts

Activity MyContact {
   Text firstName
   Text lastName
   Link phoneNumbers
          -> PhoneNumber
   Spinner country
}

Activity PhoneNumber {
   Text prefix
   Text number
}
```



LABELS

```
public class AndroidLabelProvider
   extends DefaultEObjectLabelProvider {

   public Object image(Application app) {
      return "Application.gif";
   }

   public Object text(Application app) {
      return "<<" + app.getName() + ">>";
   }
}
```

VALIDATION

```
public static final String CAPITALIZE =
   "org.eclipse.ese.android.validation.Captalize";

@Check
public void checkApplicationNameWithCapital(Application app) {
   if (!Character.isUpperCase(app.getName().charAt(0))) {
      warning("Name should start with a capital", app,
              AndroidPackage.APPLICATION__NAME, CAPITALIZE);
   }
}
```

QUICK FIX

```
@Fix(AndroidJavaValidator.CAPITALIZE)
public void capitalizeName(final Issue issue,
                           IssueResolutionAcceptor acceptor) {
   acceptor.accept(issue,
      "Capitalize name", "Capitalize the name.", "upcase.gif",
      new ISemanticModification() {
         public void apply(EObject o, IModificationContext c) {
            Application application = (Application) o;
            String name = application.getName();
            application.setName(name.substring(0,1).toUpperCase()
                  + name.substring(1));
         }
   });
}
```

WHERE TO GO NEXT...

- Add template proposals
- Define scopes
- Implement a formatter
- Highlight cross-references
- ...