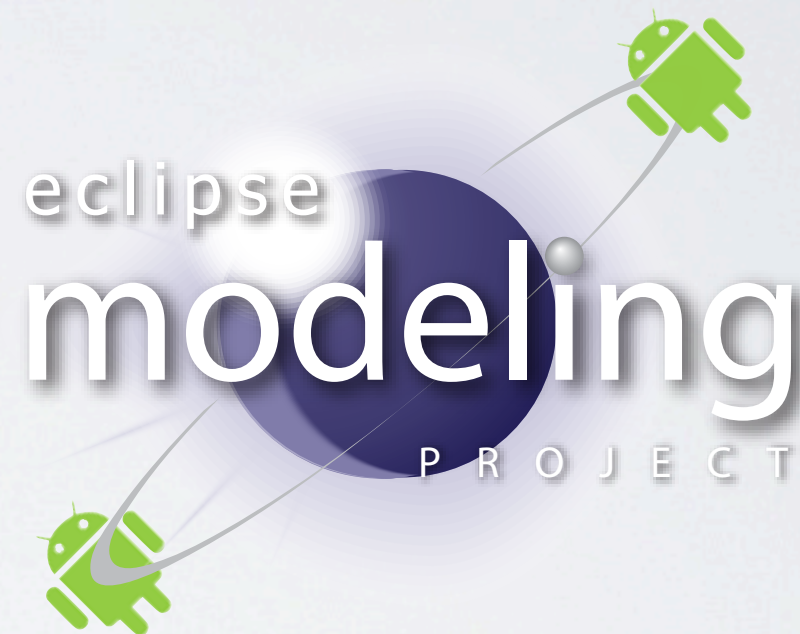


INSTALLATION



- 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 *tutorial.zip* into your workspace using *File > Import > Existing Projects into Workspace > Archive File*

CREATING A LANGUAGE FOR ANDROID APPS WITH ECLIPSE MODELING



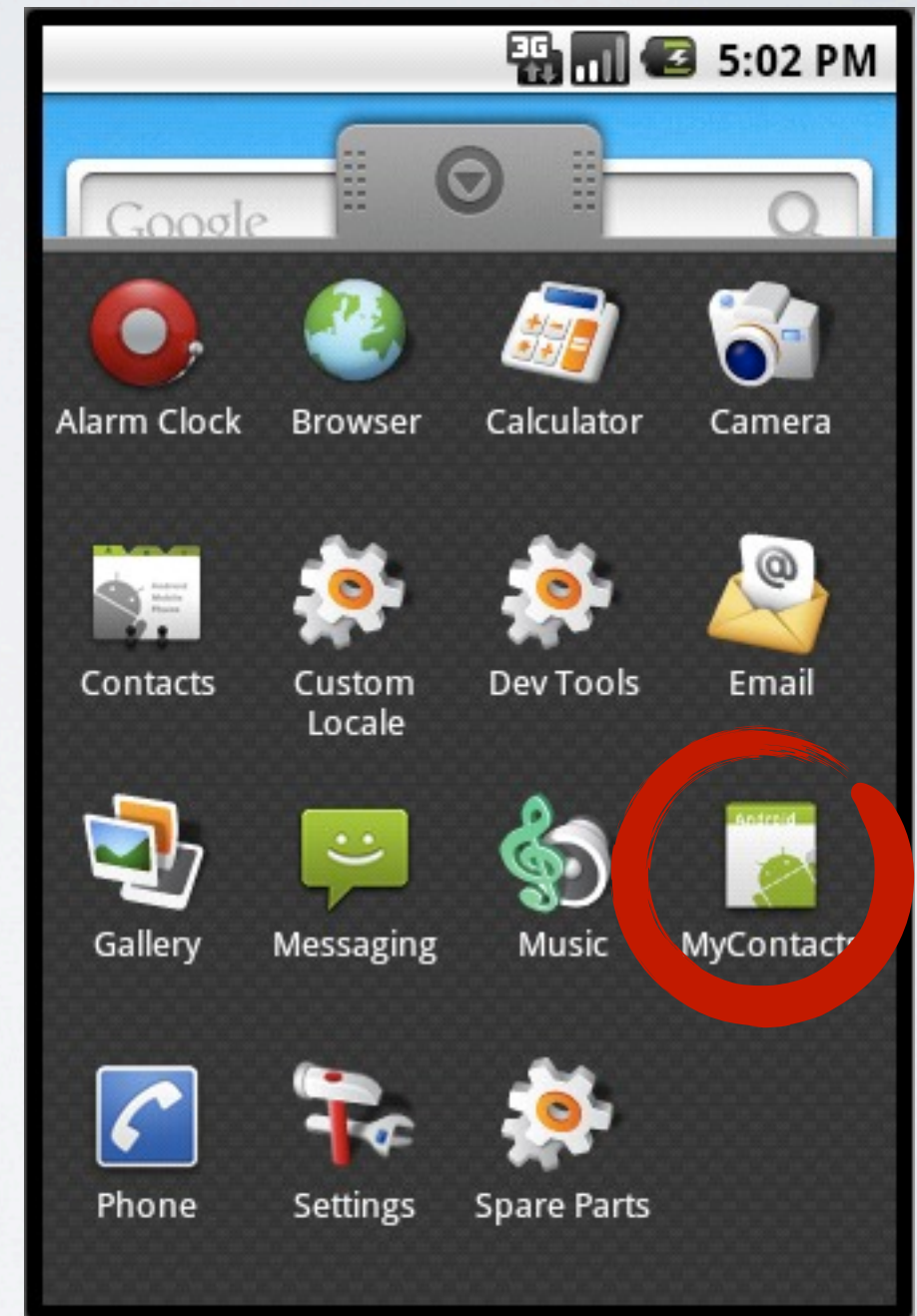
Mikaël Barbero, Stéphane Begaudeau - OBEO
Jan Köhnlein, Holger Schill, Dennis Hübner - itemis



APP CONCEPTS

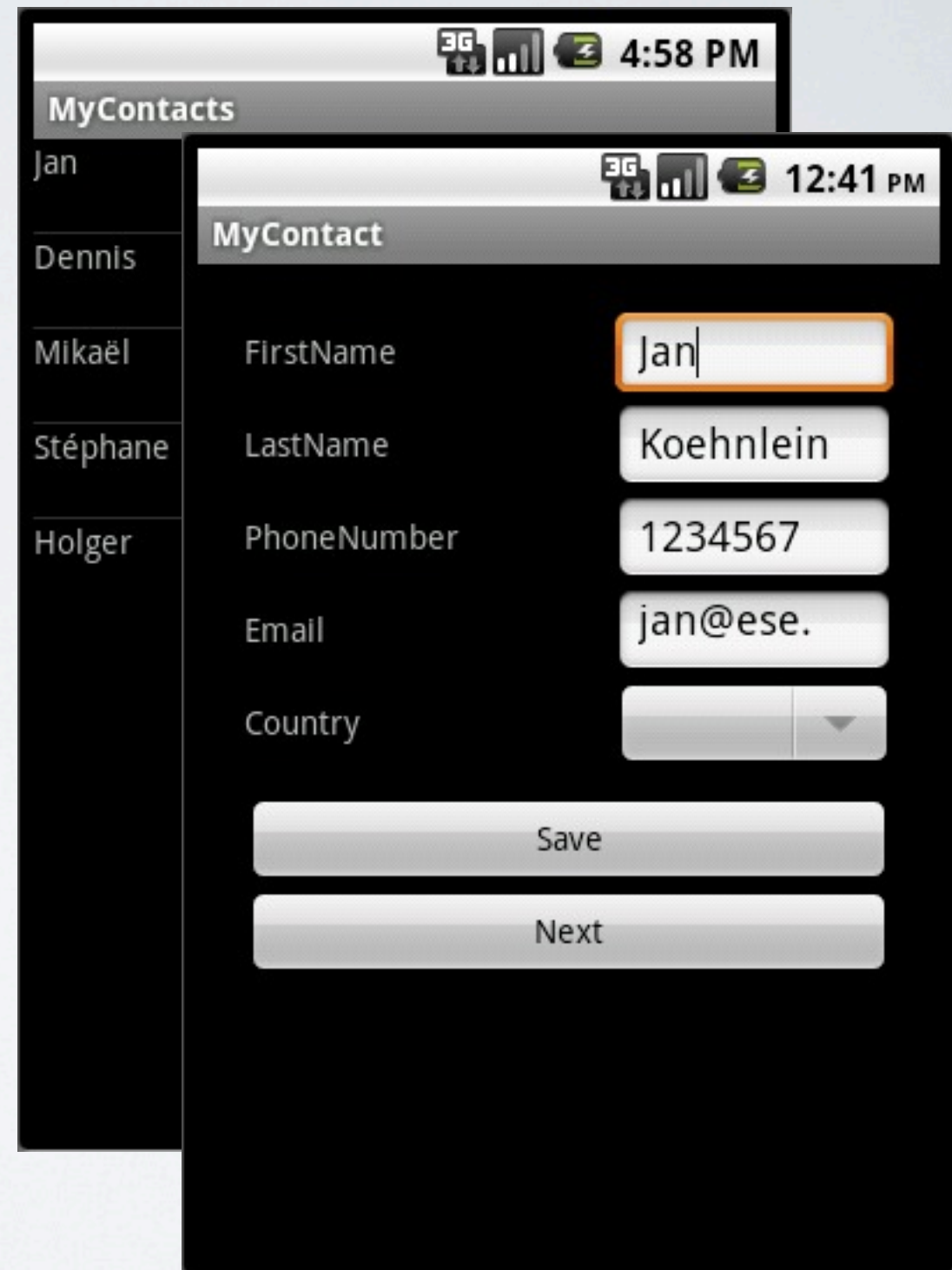
APPLICATION

- 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



WIDGETS

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

The screenshot shows an Android application titled "MyContact". The status bar at the top indicates 3G connectivity, signal strength, battery level, and the time 12:41 PM. The form contains the following fields and values:

Field	Value
FirstName	Jan
LastName	Koennlein
PhoneNumber	1234567
Email	ian@ese.
Country	(Spinner)

At the bottom of the form are two buttons: "Save" and "Next". Red circles are drawn around the "FirstName" field, the "Country" spinner, and the "Next" button.

Text

Spinner

Link

PROBLEM

- Concepts span across XML as well as Java code
- No common abstraction mechanism



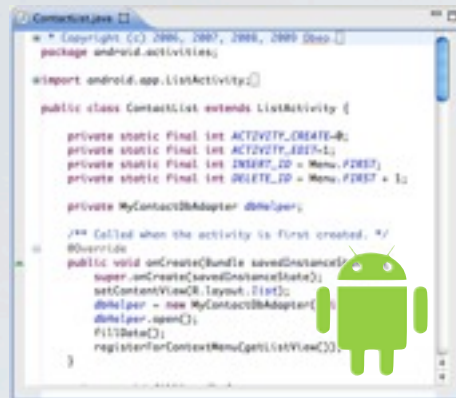
DSL SOLUTION

- Define concepts in an Ecore model
- Use Xtext to create an editor for model instances
- Use Acceleo to generate code from model instances

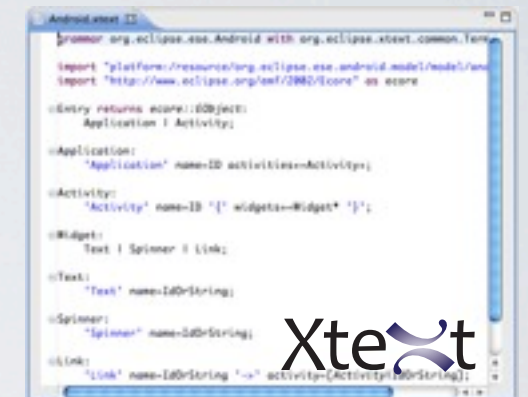


LANGUAGE SETUP

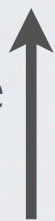
Reference



Xtext Grammar



derive



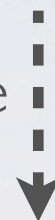
derive



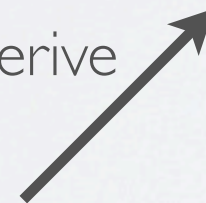
derive



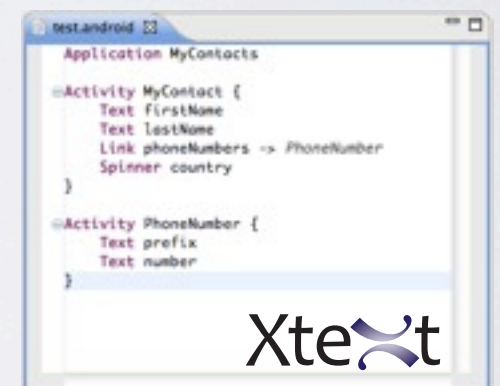
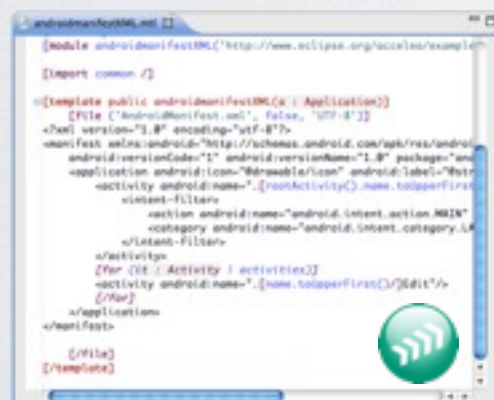
generate



derive



Ecore Model

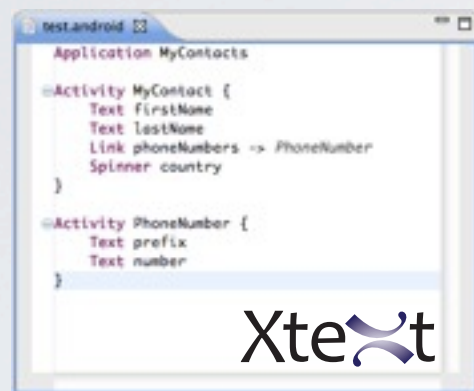


Acceleo Templates

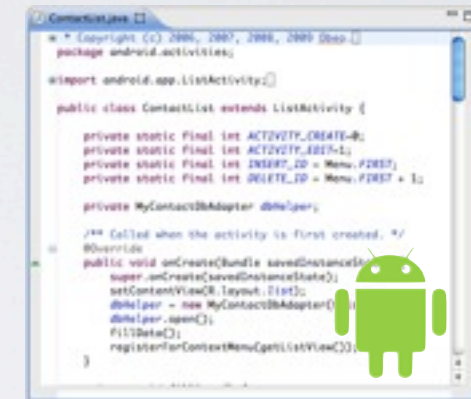
DSL Editor

CODE GENERATION

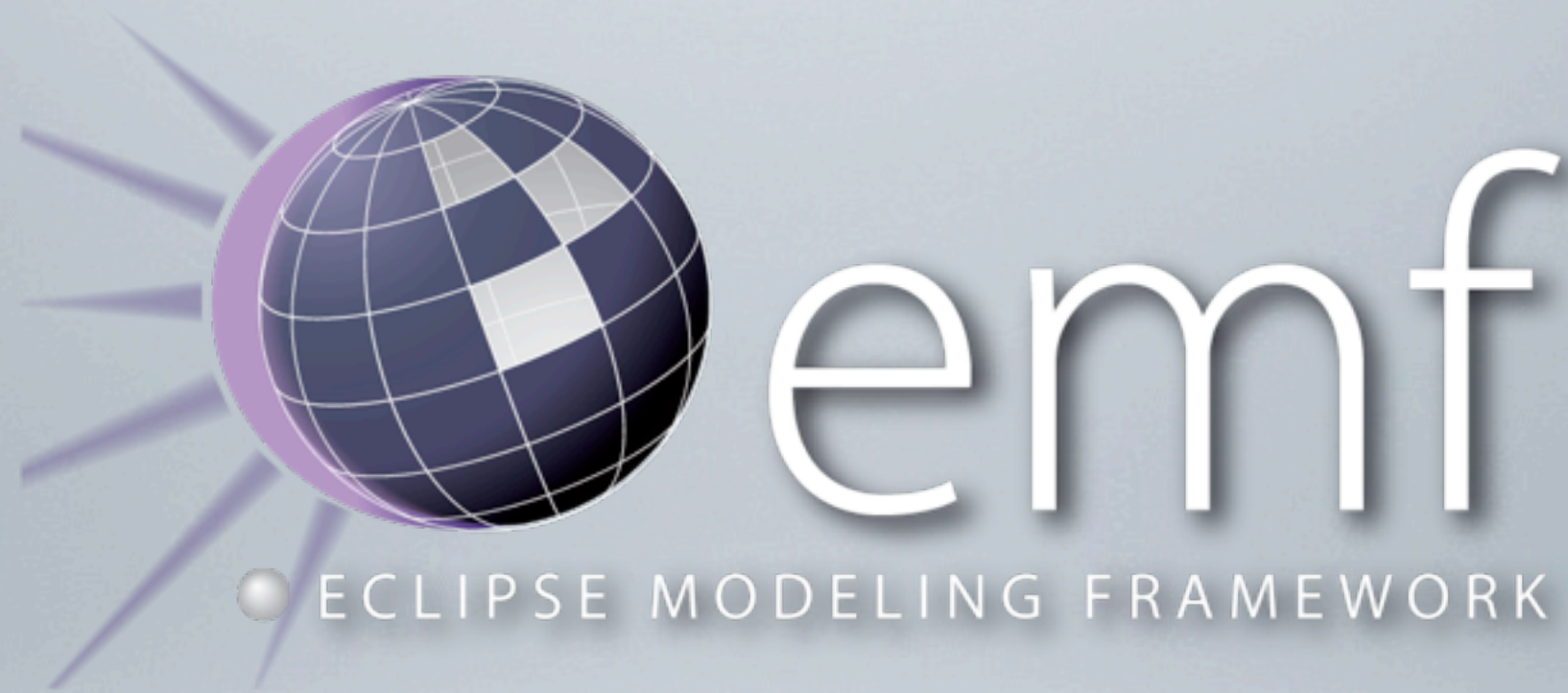
Android Model



Android App

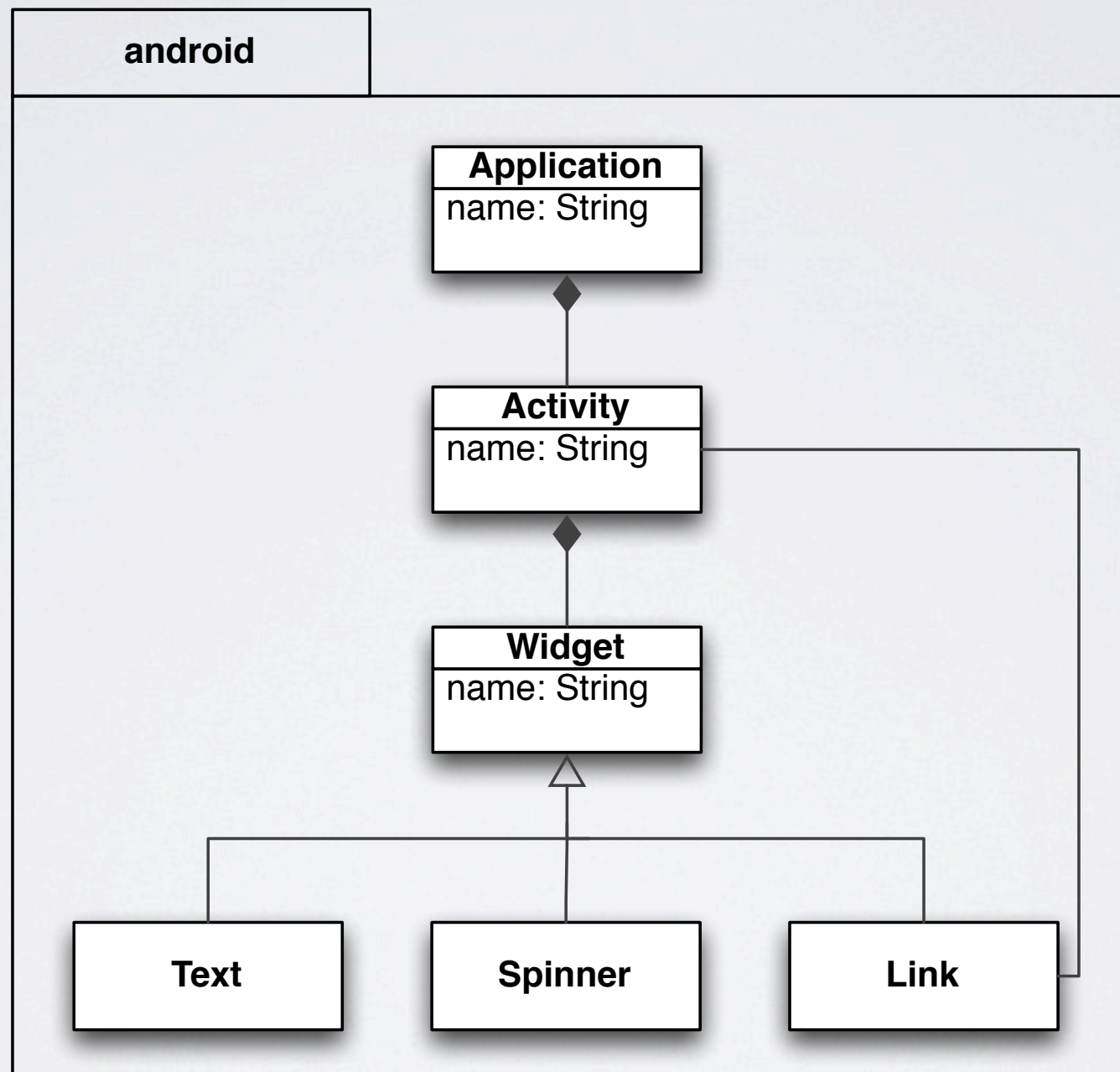


Acceleo Generator



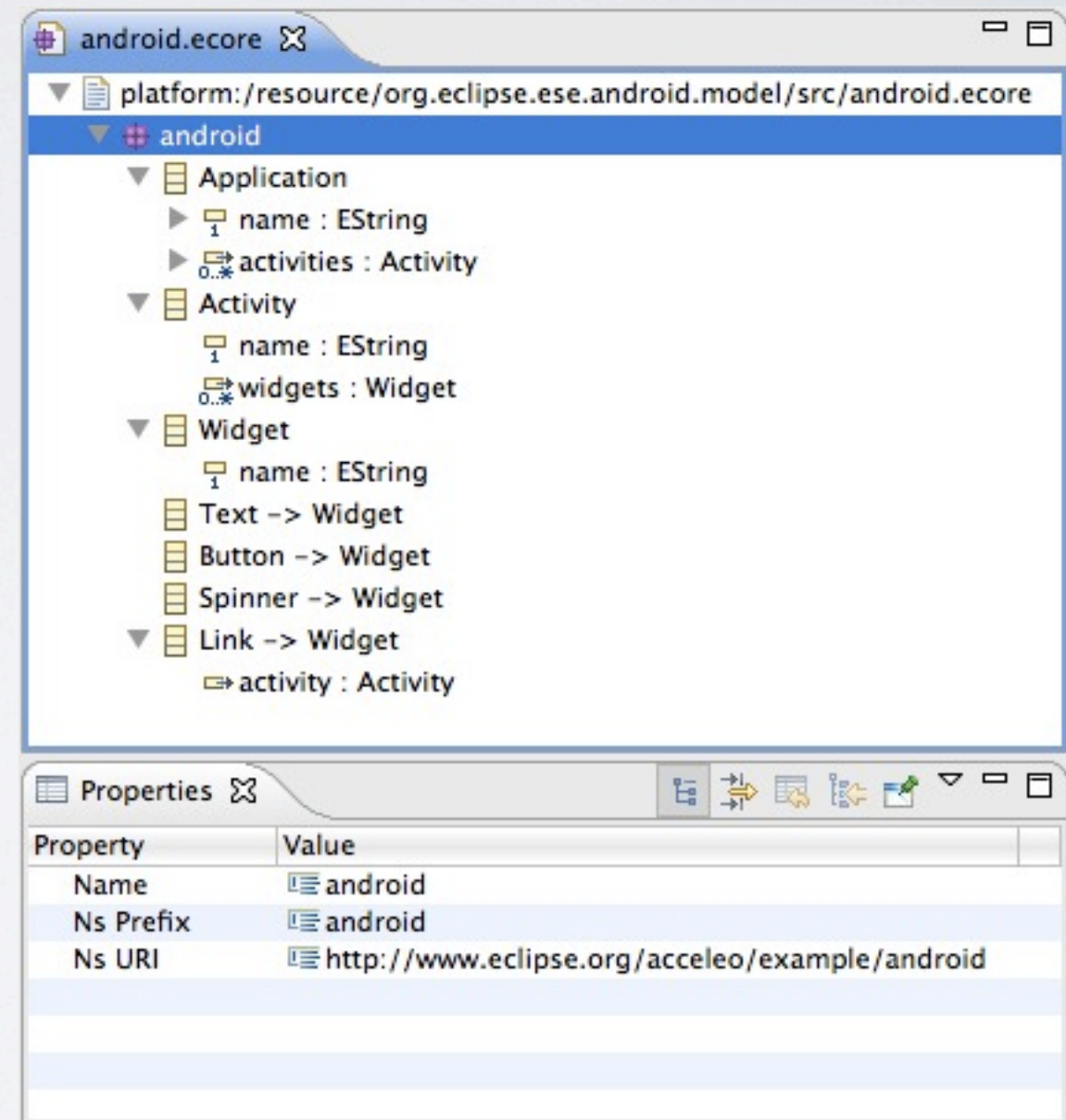
THE MODEL

THE MODEL



ECORE CONCEPTS

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



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"
```

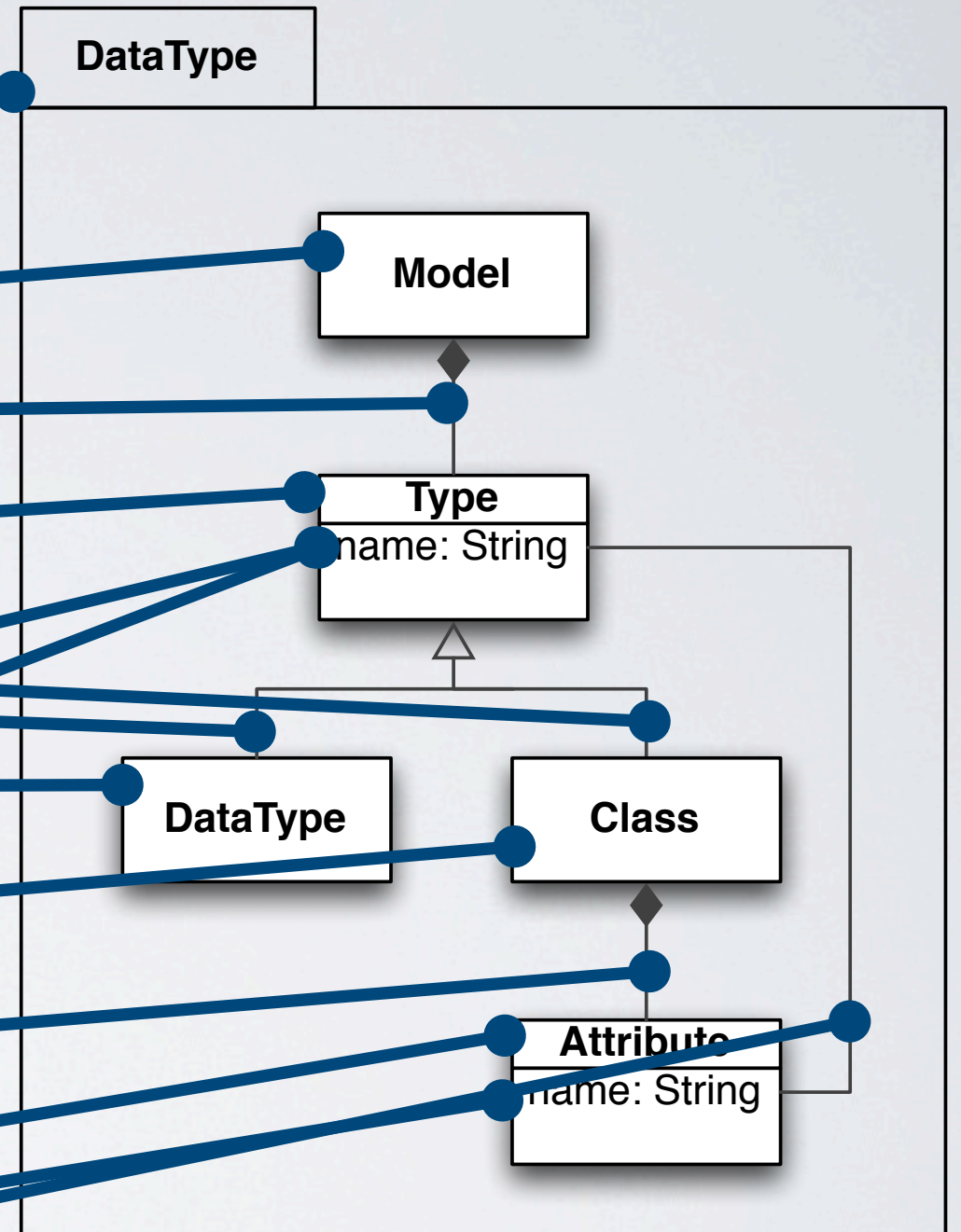
```
Model:
(types+=Type)+;
```

```
Type:
Class | DataType;
```

```
DataType:
'datatype' name=ID;
```

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

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

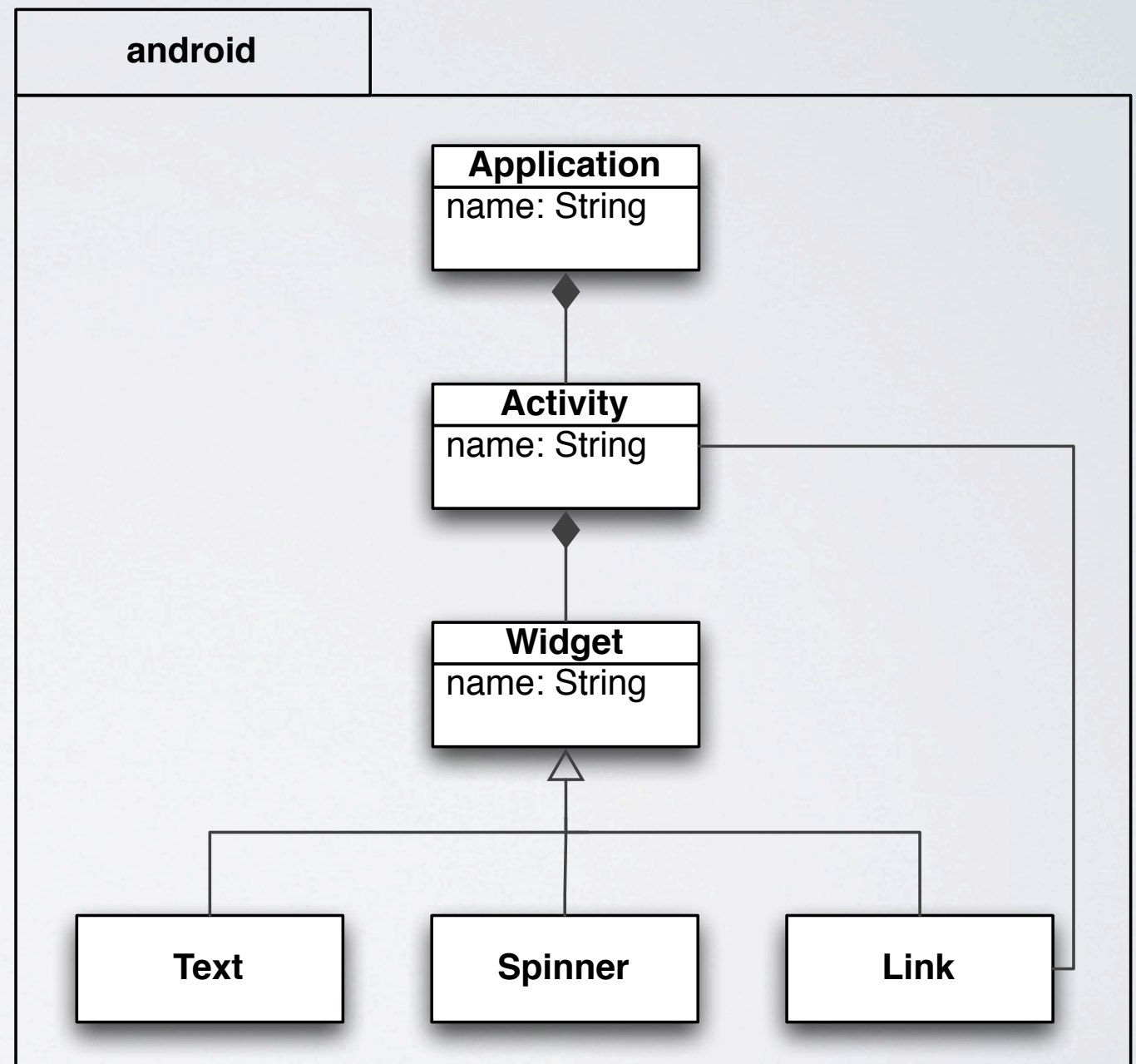


REFERENCE MODEL

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.ee.android.validation.Capitalize";  
  
@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
- ...

