

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

Android applications are inspired by Multi-Agent Systems

This is the author's manuscript

Original Citation:

Availability:

This version is available <http://hdl.handle.net/2318/151902> since 2016-06-27T17:28:31Z

Terms of use:

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

Android applications are inspired by Multi-Agent Systems

a.k.a. What does teaching technologies at university mean? How
to teach Android?

Matteo Baldoni
Università degli Studi di Torino
Dipartimento di Informatica

DroidCon Torino, February 6-7, 2014

1 Abstract

Any technology, sooner or later, passes away. A certain technology will eventually be replaced by newer technologies and computer science based technologies has a quicker turn over than other kinds of technologies. CS students expect to learn technologies because they are fashionable. Companies expect to hire students who are aware of the ultimate trend technologies because they need those technologies to be competitive in the market. So, everything seems simple: I, as a teacher, should teach those technologies to students that will be hired by the companies. Every time I face a new course, especially when it has a strong technological content, I wonder which is the best way to teach it because while I am convinced that teachers should be up to date about technologies, I am not convinced that teachers should teach technologies. The key aspect, however, is not only on the solution of a problem but also in the problem-posing. Technologies as part of the problem-posing and solving pedagogical methodology. A fundamental part of the learning process is the discussion of the problem itself and the generalization of the discovered process towards a solution. In other words, a key aspect of teaching is to create “connections”, “links” among previous technologies and methodologies, even better if they are from unrelated problems. Event-driven programming, the success of OO technologies for realizing frameworks, the achievement of software reuse are just few examples of topics, which are involved by teaching Android. So, it should not surprise that I think it is important to teach how strong the relation is between the organization of a Multi-Agent Systems, and the way in which an Android application works. Especially in order to achieve a better software reuse. In my opinion, this is a good way of teaching technologies and of overcoming their premature departure.

2 Bio

Dr. Matteo Baldoni has a Ph.D. in Computer Science. He is an associate professor of Computer Science at the Computer Science Department of the University of Turin (Italy). He is a member of both the Agent and Service-Oriented Computing group and the Knowledge Representation and Reasoning group. He has been working for over 15 years on multi-agent systems, computational logics, declarative programming languages, knowledge representation and reasoning, interaction protocols. He has also a long-term research experience on Semantic Web and E-learning.

Matteo Baldoni is co-author of over 100 papers (see <http://di.unito.it/baldonibibcat>) on the above topics, and served the research community by taking part to the organization or to the program committee of several conferences (including IJCAI, AAI, AAMAS, ICWS), and by working as a reviewer for major journals. From 2005, he is a member of the Steering Committee of the workshop “From Objects to Agents” (WOA) and of the workshop “Declarative Agent Languages and technologies” (DALT) Steering Committee since 2009. From 2006 to 2010 he chaired the working group “Sistemi ad Agenti e Multi-agente” (Agent and Multi-agent Systems) of the Italian Association for Artificial Intelligence (AI*IA). In 2011 and 2013, he was elected also as a member of the Board of AI*IA.



Droidcon Torino 2014

Android applications are inspired by Multi-Agent Systems

a.k.a. What does teaching technologies at university mean?
How to teach Android?

Matteo Baldoni¹

¹UNIVERSITÀ DEGLI STUDI DI TORINO, DIPARTIMENTO DI INFORMATICA

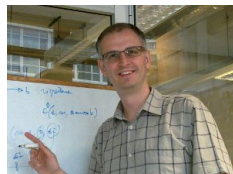
Torino, February 6-7, 2014

Overview

- 1 Who I am and Where I am from
- 2 Teaching technologies
- 3 Learning is Making Links
- 4 An example: Agents inspire Android
- 5 Discussion and conclusions

Who am I?

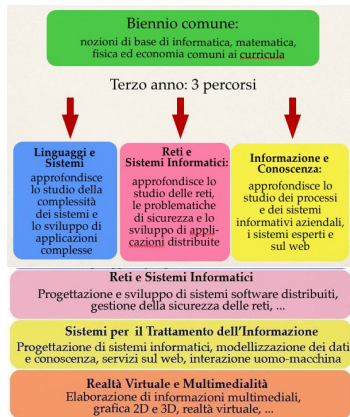
- I am an Associate Professor at Dipartimento di Informatica, Università degli Studi di Torino
- I am teaching Programmazione II, Interazione Uomo Macchina (Android), both at I level, and Agenti Intelligenti at II level
- I am a member of both the Agent and Service-Oriented Computing group and the Knowledge Representation and Reasoning group



Corso di Laurea in Informatica - I, II, and III level

Three curricula at the first level, three curricula at the II level and one at the third level.

The contents of curricula have been established in collaboration with Comitato di Indirizzo whose members are from University itself, CCIAA of Torino, and Unione Industriale of Torino



<http://www.educ.di.unito.it/orientamento/>

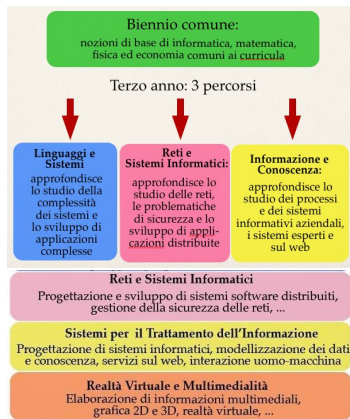
Android in our curricula

Where?

Interazione Uomo Macchina (IUM), third year of the I level (mandatory for one curriculum, optional for the remains), 3 CFU, 30 hours (theory and lab)

What?

An introduction to Android framework (theory), development of an app for reading and presenting RSS from the Dip. di Informatica (lab)



What is the problem of teaching technologies?

Expectations:

- CS students expect to learn technologies because they are fashionable
- Companies expect to hire students who are aware of the ultimate trend technologies because they need those technologies to be competitive in the market.

So, everything seems simple:

I, as a teacher, should teach those technologies to students that will be hired by the companies!

What is the problem of teaching technologies?

Expectations:

- CS students expect to learn technologies because they are fashionable
- Companies expect to hire students who are aware of the ultimate trend technologies because they need those technologies to be competitive in the market.

So, everything seems simple:

I, as a teacher, should teach those technologies to students that will be hired by the companies!

What is the problem of teaching technologies?

- Any technology, sooner or later, passes away
- A certain technology will eventually be replaced by newer technologies and computer science based technologies have a quicker turn over than other kinds of technologies.

which is the best way to teach technologies?

While I am convinced that teachers should be up to date about technologies, I am not convinced that teachers should teach technologies

What is the problem of teaching technologies?

- Any technology, sooner or later, passes away
- A certain technology will eventually be replaced by newer technologies and computer science based technologies have a quicker turn over than other kinds of technologies.

which is the best way to teach technologies?

While I am convinced that teachers should be up to date about technologies, I am not convinced that teachers should teach technologies

Learning is Making Links!

- The key aspect is not only on the solution of a problem but also in the problem-posing
- Technologies as part of the problem-posing and solving pedagogical methodology
- A fundamental part of the learning process is the discussion of the problem itself and the generalization of the discovered process towards a solution

In other words:

A key aspect of teaching is to create "connections", "links" among previous technologies and methodologies, even better if they are from unrelated problems

Learning is Making Links!

- The key aspect is not only on the solution of a problem but also in the problem-posing
- Technologies as part of the problem-posing and solving pedagogical methodology
- A fundamental part of the learning process is the discussion of the problem itself and the generalization of the discovered process towards a solution

In other words:

A key aspect of teaching is to create "connections", "links" among previous technologies and methodologies, even better if they are from unrelated problems

Learning is Making Links!

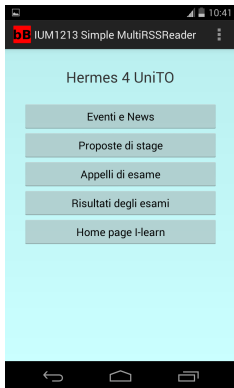
- Topics involved by teaching Android: Event-driven programming, the success of OO technologies for realizing frameworks, the achievement of software reuse
- How strong the relation is between the organization of a Multi-Agent System, and the way in which an Android application works

In my opinion,

this is a good way of teaching technologies and of overcoming their premature departure

Let us consider an example

- An app with two activities, the first one offers a choice to the user and the second one visualizes the information associated with the choice of the user in the previous activity

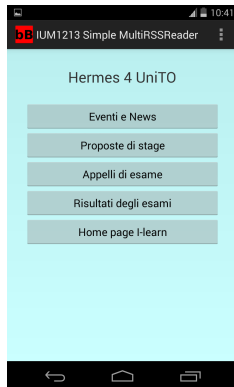


Let us consider an example

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="it.unito.di.iium1213.rssreader"
3   android:versionCode="1"
4   android:versionName="1.0" >
5   <uses-sdk
6     android:minSdkVersion="8"
7     android:targetSdkVersion="16" />
8   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
9   <uses-permission android:name="android.permission.INTERNET" />
10  <application
11    android:allowBackup="true"
12    android:icon="@drawable/ic_launcher"
13    android:label="@string/app_name"
14    android:theme="@style/AppTheme" >
15    <activity
16      android:name="it.unito.di.iium1213.rssreader.MainActivity"
17      android:label="@string/app_name" >
18      <intent-filter>
19        <action android:name="android.intent.action.MAIN" />
20        <category android:name="android.intent.category.LAUNCHER" />
21      </intent-filter>
22    </activity>
23    <activity
24      android:name="it.unito.di.iium1213.rssreader.DisplayRSSActivity"
25      android:label="@string/title_activity_display_rss"
26      android:parentActivityName="it.unito.di.iium1213.rssreader.MainActivity" >
27      <meta-data
28        android:name="android.support.PARENT_ACTIVITY"
29        android:value="it.unito.di.iium1213.rssreader.MainActivity" />
30    </activity>
31  </application>
32 </manifest>
```

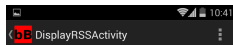
Let us consider an example

```
1 // ...
2 public class MainActivity extends Activity {
3     // ...
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(R.layout.activity_main);
7     }
8     public void showSelectedRSS(View view) {
9         Intent intent = new Intent(this,
10             DisplayRSSActivity.class);
11         switch(view.getId()) {
12             case R.id.eventi_news:
13                 intent.putExtra(RSS_CODE, 0);
14                 break;
15             case R.id.proposte_stage:
16                 intent.putExtra(RSS_CODE, 1);
17                 break;
18             case R.id.appelli_esame:
19                 intent.putExtra(RSS_CODE, 2);
20                 break;
21             case R.id.risultati_esame:
22                 intent.putExtra(RSS_CODE, 3);
23                 break;
24             case R.id.ilearn:
25                 intent.putExtra(RSS_CODE, 4);
26                 break;
27         }
28         startActivity(intent);
29     }
30     // ...
31 }
```



Let us consider an example

```
1 // ...
2 public class DisplayRSSActivity extends Activity {
3     // ...
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(
7             R.layout.activity_display_rss );
8         Intent intent = getIntent();
9         int rssID = intent.
10            getIntExtra( MainActivity.RSS_CODE, -1);
11        TextView =
12            (WebView) findViewById(R.id.myRSS);
13        ConnectivityManager connMgr =
14            (ConnectivityManager) getSystemService(
15                Context.CONNECTIVITY_SERVICE);
16        NetworkInfo networkInfo =
17            connMgr.getActiveNetworkInfo();
18        if (networkInfo != null &&
19            networkInfo.isConnected()) {
20            new DownloadFeed().execute(
21                MainActivity.RSS_LINKS[rssID]);
22        } else {
23            currentFeed =
24                "No network connection available.";
25            textView.loadData(
26                currentFeed, "text/html", null);
27        }
28    }
29    // ...
30 }
```



Feed I-learn

Updated: Dec 12 10:41AM

[TARM-RECUPERO per gli studenti che non l'hanno sostenuto a settembre](#)

Questa notizia scade il 08 Gennaio 2014.

Editore: gatti.
Codice: 1987

In allegato il file * di tipo: wiki, avete dimensioni
566 Byte

[Borsa di studio Cloud Computing \(INFN\)](#)

Questa notizia scade il 31 Dicembre 2013.

Editore: gatti.
Codice: 1985

In allegato il file * di tipo: wiki, avete dimensioni
737 Byte



What did I learn?

- How to specify an AndroidManifest
- How to define an activity
- How to call an activity from another activity

Something more is there?

What did I learn?

- How to specify an AndroidManifest
- How to define an activity
- How to call an activity from another activity

Something more is there?

What is an agent?

An intelligent agent is:

A computer system capable of flexible autonomous action in some environment in order to achieve its goals

Object

- Has identity
- Encapsulates some state and behaviour
- Has interfaces by/through which they communicate with each other

Agent

- Has identity
- Encapsulates some state and behaviour
- Has interfaces by/through which they communicate with each other
- Is autonomous
- Uses an agent-communication language
- Is pro-active

What is an agent?

An intelligent agent is:

A computer system capable of flexible autonomous action in some environment in order to achieve its goals

Object

- Has identity
- Encapsulates some state and behaviour
- Has interfaces by/through which they communicate with each other

Agent

- Has identity
- Encapsulates some state and behaviour
- Has interfaces by/through which they communicate with each other
- *Is autonomous*
- *Uses an agent-communication language*
- *Is pro-active*

What is an agent?

An intelligent agent is:

A computer system capable of flexible autonomous action in some environment in order to achieve its goals

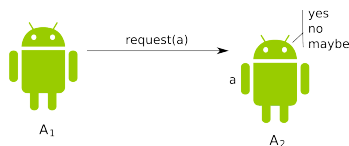
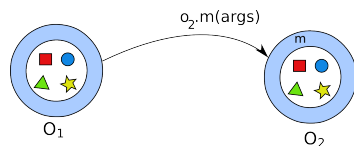
Object

- Has identity
- Encapsulates some state and behaviour
- Has interfaces by/through which they communicate with each other

Agent

- Has identity
- Encapsulates some state and behaviour
- Has interfaces by/through which they communicate with each other
- Is *autonomous*
- Uses an *agent-communication language*
- Is *pro-active*

What is an agent?

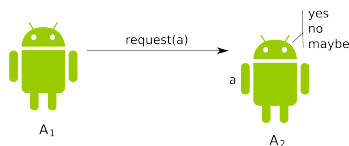
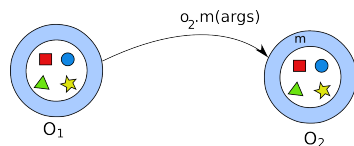


- Which object makes the decision about the execution of method m ?
- Object O_2 has no control over the execution of m : the decision about whether to execute m lies entirely with O_1 .

- A_2 is an autonomous agent: it has control over *both* its state and its behaviour
- A_1 cannot take for granted that agent A_2 will execute action a just because A_1 wants it to

Objects do it for free. Agents do it because they want to

What is an agent?

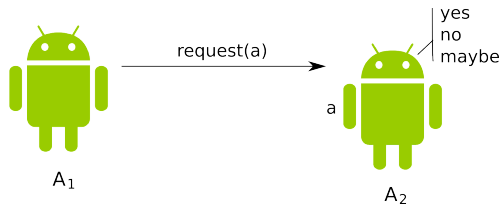


- Which object makes the decision about the execution of method m ?
- Object O_2 has no control over the execution of m : the decision about whether to execute m lies entirely with O_1 .

- A_2 is an autonomous agent: it has control over *both* its state and its behaviour
- A_1 cannot take for granted that agent A_2 will execute action a just because A_1 wants it to

Objects do it for free. Agents do it because they want to

Why agents?

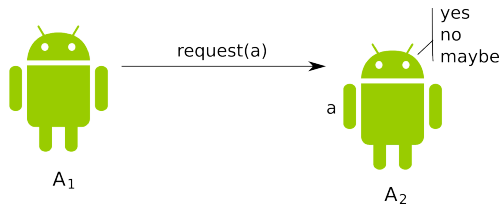


Advantages:

- Break the control flow
- Decouple components
- Achieve a better modularization
- Achieve an easier software re-use

JADE/WADE have been developed by Telecom Italia

Why agents?



Advantages:

- Break the control flow
- Decouple components
- Achieve a better modularization
- Achieve an easier software re-use

JADE/WADE have been developed by Telecom Italia

Our example: revisited

```
1 // ...
2 public class MainActivity extends Activity {
3     // ...
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(R.layout.activity_main);
7     }
8     public void showSelectedRSS(View view) {
9         Intent intent = new Intent(this, DisplayRSSActivity.class);
10        switch(view.getId()) {
11            case R.id.eventi_news:
12                intent.putExtra(RSS_CODE, 0);
13                break;
14            case R.id.proposte_stage:
15                intent.putExtra(RSS_CODE, 1);
16                break;
17            case R.id.appelli_esame:
18                intent.putExtra(RSS_CODE, 2);
19                break;
20            case R.id.risultati_esame:
21                intent.putExtra(RSS_CODE, 3);
22                break;
23            case R.id.ilearn:
24                intent.putExtra(RSS_CODE, 4);
25                break;
26        }
27        startActivity(intent);
28    }
29    // ...
30 }
```

Our example: revisited

```
1 // ...
2 public class MainActivity extends Activity {
3 // ...
4 protected void onCreate(Bundle savedInstanceState) {
5     super.onCreate(savedInstanceState);
6     setContentView(R.layout.activity_main);
7 }
8 public void showSelectedRSS(View view) {
9     Intent intent = new Intent(this, DisplayRSSActivity.class);
10    switch(view.getId()) {
11        case R.id.eventi_news:
12            intent.putExtra(RSS_CODE, 0);
13            break;
14        case R.id.proposte_stage:
15            intent.putExtra(RSS_CODE, 1);
16            break;
17        case R.id.appelli_esame:
18            intent.putExtra(RSS_CODE, 2);
19            break;
20        case R.id.risultati_esame:
21            intent.putExtra(RSS_CODE, 3);
22            break;
23        case R.id.ilearn:
24            intent.putExtra(RSS_CODE, 4);
25            break;
26    }
27    startActivity(intent);
28 }
29 // ...
30 }
```

This is a communication not a method invocation!
There is no reference to the other activity, there is no an instruction of the form *o.m(args)*

Our example: revisited

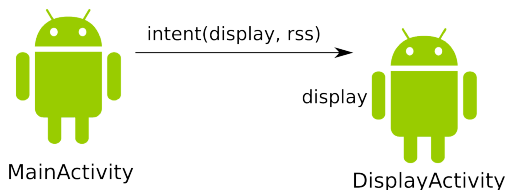
```
1 // ...
2 public class DisplayRSSActivity extends Activity {
3     // ...
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(
7             R.layout.activity_display_rss);
8         Intent intent = getIntent();
9         int rssID = intent.getIntExtra(MainActivity.RSS_CODE, -1);
10        textView =
11            (WebView) findViewById(R.id.myRSS);
12        ConnectivityManager connMgr =
13            (ConnectivityManager) getSystemService(
14                Context.CONNECTIVITY_SERVICE);
15        NetworkInfo networkInfo =
16            connMgr.getActiveNetworkInfo();
17        if (networkInfo != null &&
18            networkInfo.isConnected()) {
19            new DownloadFeed().execute(
20                MainActivity.RSS_LINKS[rssID]);
21        } else {
22            currentFeed =
23                "No network connection available.";
24            textView.loadData(
25                currentFeed, "text/html", null);
26        }
27    }
28    // ...
29 }
```

Our example: revisited

```
1 // ...
2 public class DisplayRSSActivity extends Activity {
3     // ...
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(
7             R.layout.activity_display_rss );
8         Intent intent = getIntent();
9         int rssID = intent.getIntExtra( MainActivity.RSS_CODE, -1);
10        textView =
11            (WebView) findViewById(R.id.myRSS);
12        ConnectivityManager connMgr =
13            (ConnectivityManager) getSystemService(
14                Context.CONNECTIVITY_SERVICE);
15        NetworkInfo networkInfo =
16            connMgr.getActiveNetworkInfo();
17        if (networkInfo != null &&
18            networkInfo.isConnected()) {
19            new DownloadFeed().execute(
20                MainActivity.RSS_LINKS[rssID]);
21        } else {
22            currentFeed =
23                "No network connection available.";
24            textView.loadData(
25                currentFeed, "text/html", null);
26        }
27    }
28    // ...
29 }
```

The activity autonomously decide to read the message!

Activities as Agents

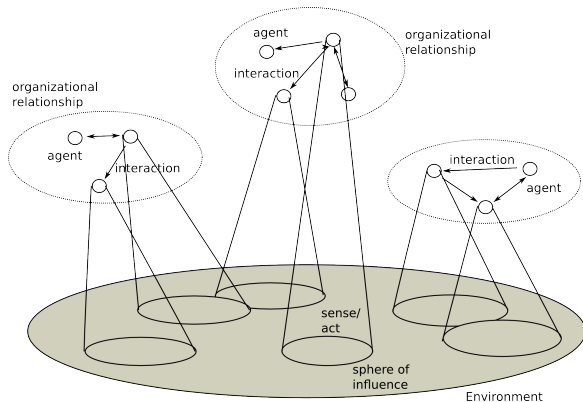


Advantages:

- Break the control flow: each activity has own life cycle
- Decoupling components: an activity does not refer directly to another activity
- Achieve a better modularization: an activity is more self-defined
- Achieve an easier software re-use: more than one activity may be used to answer to a intent request

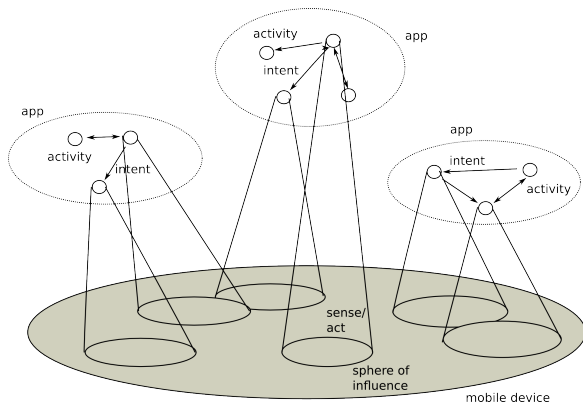
A Multi-Agent System (MAS)

A multiagent system contains a number of agents which interact through communication, are able to act in an environment, have different "spheres of influence" (which may coincide), will be linked by other (organizational) relationships



An Android Device

An Android device contains a number of activities which interact through communication (intent), are able to act in an mobile device, have different "spheres of influence" (which may coincide), will be linked by other apps



Our example: revisited

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="it.unito.di.iium1213.rssreader"
3   android:versionCode="1"
4   android:versionName="1.0" >
5   <uses-sdk
6     android:minSdkVersion="8"
7     android:targetSdkVersion="16" />
8   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
9   <uses-permission android:name="android.permission.INTERNET"/>
10  <application
11    android:allowBackup="true"
12    android:icon="@drawable/ic_launcher"
13    android:label="@string/app_name"
14    android:theme="@style/AppTheme" >
15    <activity
16      android:name="it.unito.di.iium1213.rssreader.MainActivity"
17      android:label="@string/app_name" >
18      <intent-filter>
19        <action android:name="android.intent.action.MAIN" />
20        <category android:name="android.intent.category.LAUNCHER" />
21      </intent-filter>
22    </activity>
23    <activity
24      android:name="it.unito.di.iium1213.rssreader.DisplayRSSActivity"
25      android:label="@string/title_activity_display_rss"
26      android:parentActivityName="it.unito.di.iium1213.rssreader.MainActivity" >
27      <meta-data
28        android:name="android.support.PARENT_ACTIVITY"
29        android:value="it.unito.di.iium1213.rssreader.MainActivity" />
30    </activity>
31  </application>
32 </manifest>
```

Our example: revisited

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="it.unito.di.iium1213.rssreader"
3   android:versionCode="1"
4   android:versionName="1.0" >
5   <uses-sdk
6     android:minSdkVersion="8"
7     android:targetSdkVersion="16" />
8   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
9   <uses-permission android:name="android.permission.INTERNET" />
10  <application
11    android:allowBackup="true"
12    android:icon="@drawable/ic_launcher"
13    android:label="@string/app_name"
14    android:theme="@style/AppTheme" >
15    <activity
16      android:name="it.unito.di.iium1213.rssreader.MainActivity"
17      android:label="@string/app_name" >
18      <intent-filter>
19        <action android:name="android.intent.action.MAIN" />
20        <category android:name="android.intent.category.LAUNCHER" />
21      </intent-filter>
22    </activity>
23    <activity
24      android:name="it.unito.di.iium1213.rssreader.DisplayRSSActivity"
25      android:label="@string/title_activity_display_rss"
26      android:parentActivityName="it.unito.di.iium1213.rssreader.MainActivity" >
27      <meta-data
28        android:name="android.support.PARENT_ACTIVITY"
29        android:value="it.unito.di.iium1213.rssreader.MainActivity" />
30    </activity>
31  </application>
32 </manifest>
```

The diagram consists of an orange rounded rectangle on the right side of the slide containing the text "Organization and agents!". Three black arrows originate from this box and point to specific XML elements in the code: one points to the opening <application> tag on line 10, one points to the opening <activity> tag on line 15, and one points to the opening <activity> tag on line 23.

What did I learn?

- How to define an activity: how to define components that act and sense an environment
- How to specify an AndroidManifest: how to specify an organization of autonomous components that can commit to do something
- How to call an activity from another activity: how to obtain an easier software reuse and increase the decoupling of components by means of communication

Indeed, something more is there!

What did I learn?

- How to define an activity: how to define components that act and sense an environment
- How to specify an AndroidManifest: how to specify an organization of autonomous components that can commit to do something
- How to call an activity from another activity: how to obtain an easier software reuse and increase the decoupling of components by means of communication

Indeed, something more is there!

Conclusions

- We are aware of the importance of teaching technologies, both for students and companies
- We are aware that technologies, sooner or later, pass away
- Technologies as part of the problem-posing and solving pedagogical methodology
- A key aspect of teaching is to create "connections", "links" among technologies and methodologies
- An example: App as MAS
- In my opinion, this is a good way of teaching technologies and of overcoming their premature departure

Thanks for your attention!

Conclusions

- We are aware of the importance of teaching technologies, both for students and companies
- We are aware that technologies, sooner or later, pass away
- Technologies as part of the problem-posing and solving pedagogical methodology
- A key aspect of teaching is to create "connections", "links" among technologies and methodologies
- An example: App as MAS
- In my opinion, this is a good way of teaching technologies and of overcoming their premature departure

Thanks for your attention!