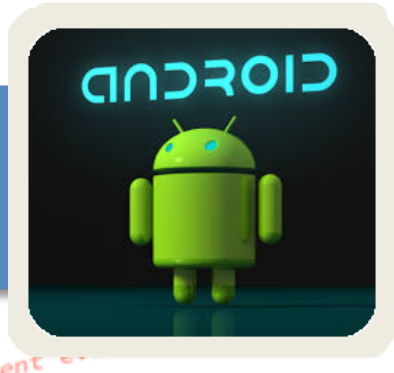
Aspect-oriented user interface design for Android applications

Jiří Šebek

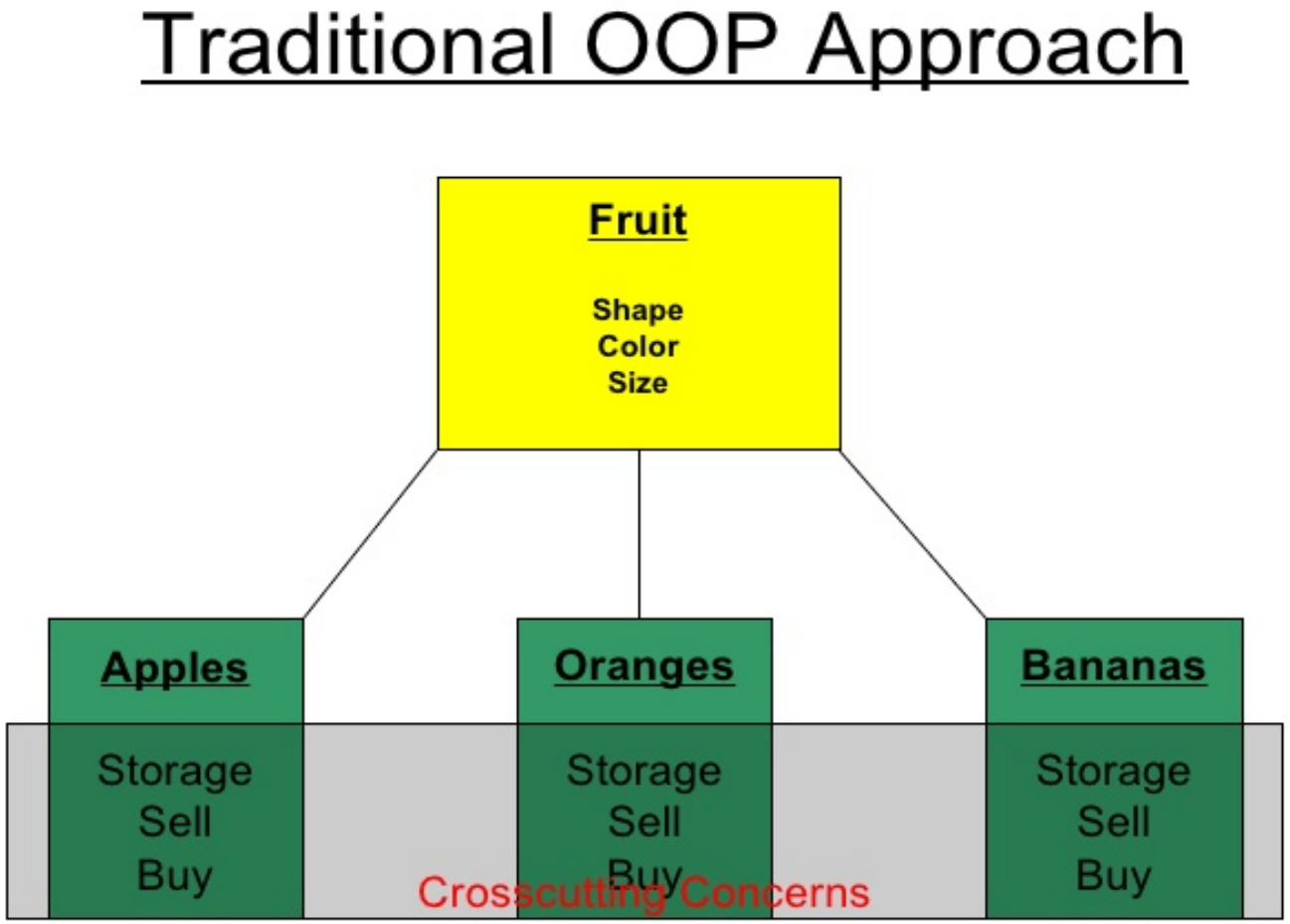
Department of Computer Science and Engineering



Aspect oriented programming

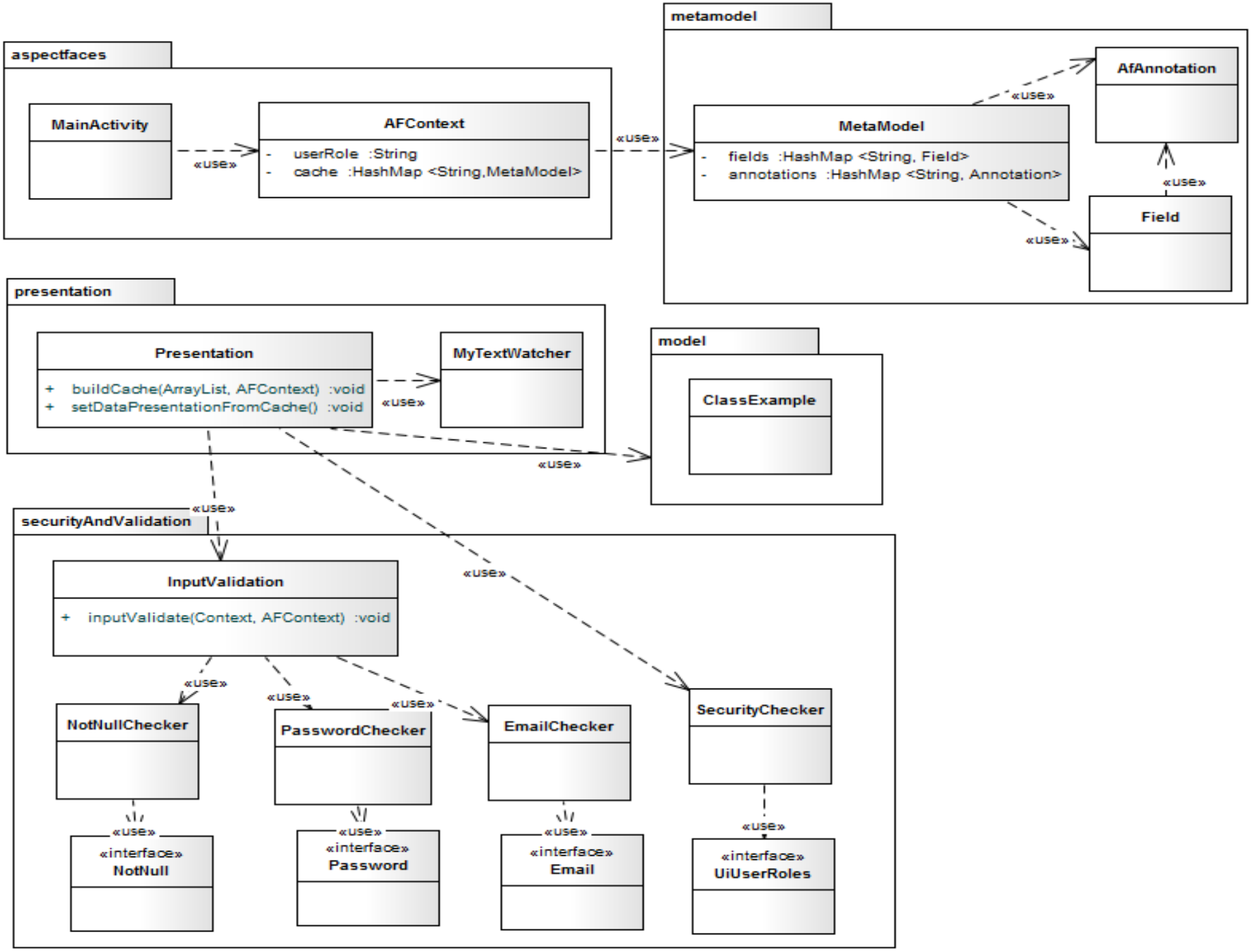
- Divide program into moduls (aspects)
- crosscutting concerns
- Aspect weaving
- Static vs dynamic
- Front-end vs back-end

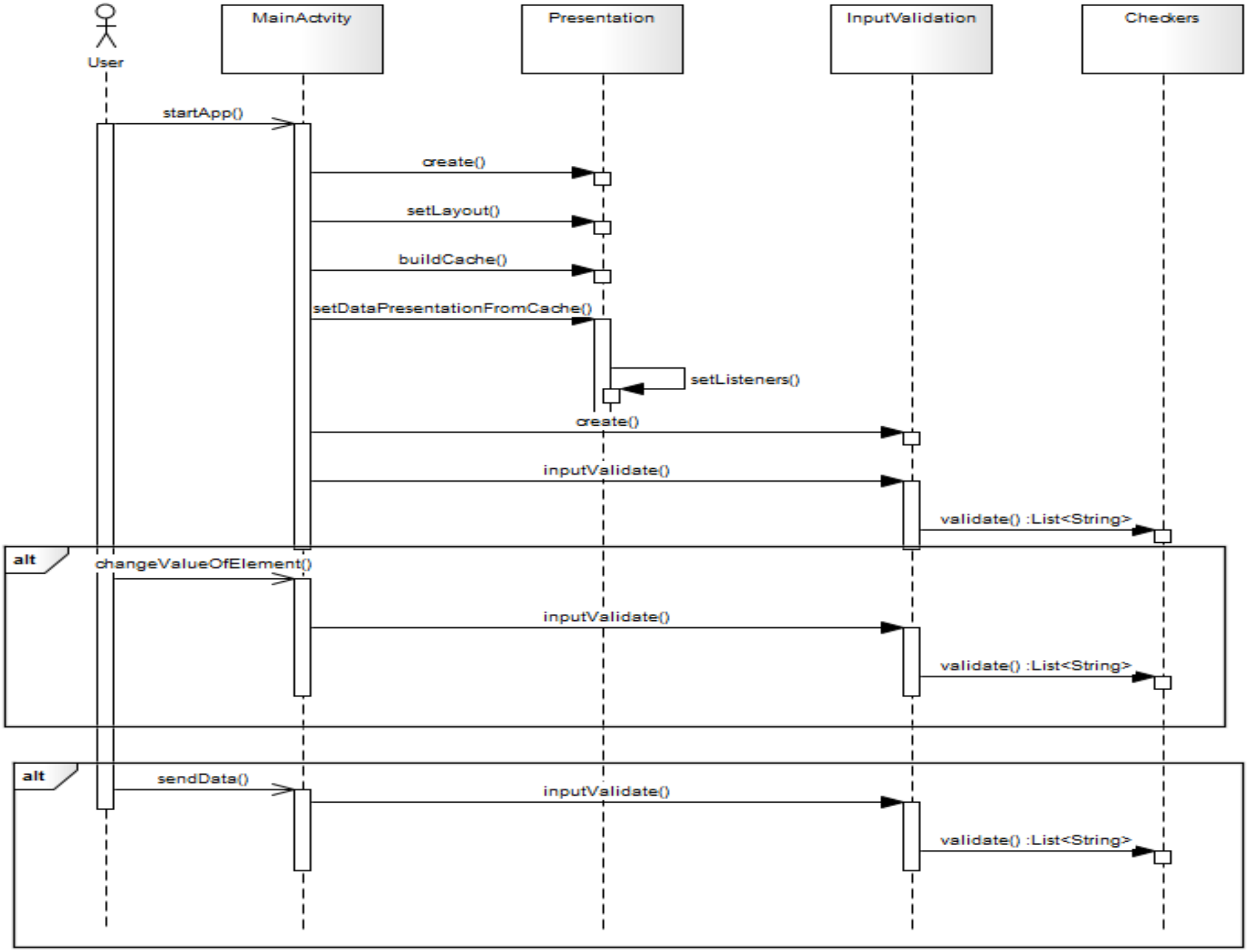
Aspect oriented programming



Background

- Existing techniques of app development
  - Traditional development
  - Aspect oriented development
  - Model driven development (MDD)
  - Generative programming (GP)
  - Meta programming (MP)
  - Domain-specific language (DSL)
- Existing tools
  - Metawidget, AspectFaces...


aspectfaces
MainActivity
«use»
AFContext
- userRole :String
- cache :HashMap <String,MetaModel>
«use»
metamodel
AfAnnotation
«use»
MetaModel
- fields :HashMap <String, Field>
- annotations :HashMap <String, Annotation>
«use»
Field
«use»
presentation
Presentation
+ buildCache(ArrayList, AFContext) :void
+ setDataPresentationFromCache() :void
«use»
MyTextWatcher
«use»
model
ClassExample
«use»
securityAndValidation
InputValidation
+ inputValidate(Context, AFContext) :void
«use»
NotNullChecker
«use»
PasswordChecker
«use»
EmailChecker
«use»
SecurityChecker
«use»
«interface»
NotNull
«use»
«interface»
Password
«use»
«interface»
Email
«use»
«interface»
UiUserRoles


User
MainActvity
Presentation
InputValidation
Checkers
startApp()
create()
setLayout()
buildCache()
setDataPresentationFromCache()
setListeners()
create()
inputValidate()
validate() :List<String>
alt
changeValueOfElement()
inputValidate()
validate() :List<String>
alt
sendData()
inputValidate()
validate() :List<String>

Usage

```
Presentation p = new Presentation(this, arraylistOfInstances);
p.buildCache(arraylistOfInstances, afContext);
View v = p.setDataPresentationFromCache();
setContentView(v);
```

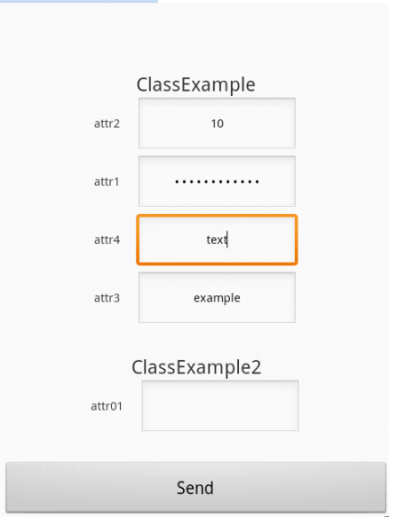
```
@UiUserRoles(role={"ROLE_ADMIN"})
public class ClassExample{
 @NotNull(message="attr1 can not be null.")
 @Password
 String attr1;
 @UiUserRoles(role={"ROLE_ADMIN","ROLE_USER"})
 @NotNull(message="attr2 can not be null.")
 int attr2;
 @UiUserRoles(role={"ROLE_ADMIN"})
 @Email(message="attr3 is not in email format.")
 String attr3;
```

Implementation

- Aspects implemented:
  - Layout
  - Prezentation
  - Data binding
  - Validation
  - Security
- Techniques used:
  - intern cache, Rich Entity Aspect/Audit Design (READ)

```
Presentation p = new Presentation(this, arraylistOfInstances);
p.buildCache(arraylistOfInstances, afContext);
View v = p.setDataPresentationFromCache();
setContentView(v);
```

```
@UiUserRoles(role={"ROLE_ADMIN"})
public class ClassExample{
 @NotNull(message="attr1 can not be null.")
 @Password
 String attr1;
 @UiUserRoles(role={"ROLE_ADMIN","ROLE_USER"})
 @NotNull(message="attr2 can not be null.")
 int attr2;
 @UiUserRoles(role={"ROLE_ADMIN"})
 @Email(message="attr3 is not in email format.")
 String attr3;
```



Comparison AOP on platform Android and Java EE

- Android


```
Presentation p = new Presentation(this, arraylistOfInstances);

p.buildCache(arraylistOfInstances, afContext);

View v = p.setDataPresentationFromCache();

setContentView(v);
```

- Java EE

```
<af:ui instance="#{bean.entity1}" edit="true"/>

<af:ui instance="#{bean.entity2}" edit="true"/>
```

Comparison of AOP and traditional development on Android

Features	AOP	Conventional approach
Reuse	yes	no
Runtime approach	yes	no
Reduce code	yes	no
Better to maintain	yes	no
Separated each aspects	yes	no
Readable code	yes	no (depends on developper)
Time to launch the form (average)	119,5ms	193,1ms
Standard declination (std)	5,35ms	14,7ms
Lines of code (LOC)	29	495

Table 7.1. Comparison of AOP and conventional approach

Conclusion

- Aspect oriented framework for Adroid was created
- Code created with framework is:
  - readable
  - reuseable
  - better to maintain (separated each aspect)
  - reduce size of code
- Framework uses runtime approach

Future work

- New aspect friendly ui (learning phase, deployed phase)
- Adding aspects for REST api, database layer...
- Create new language for security
- Use of devices hw (sensors..)
  - Change UI based on this information

Thank you for your attention

Jiří Šebek