A Platform Independent Framework for Statecharts Code Generation

L. Andolfato, G. Chiozzi, ESO
N. Migliorini, Universita’ di Ferrara, Italy
C. Morales, Universidad Tecnica Federico Santa Maria, Chile
Outline

• Motivations
  • Application Frameworks
  • Model Driven SW Development

• Architectural Concepts
  • Statecharts Semantic and the SCXML Standard
  • UML-to-SCXML Mapping
  • COMODO Profile for UML
  • Cross Platform Model2Text Transformations

• Future Plans
Statecharts for the Very Large Telescope (VLT)

- VLT SW Platform: Linux, GNU C/C++, TCL/TK, CCS
- Workstation Software Framework: framework for the development of control/monitor applications from Statecharts models (more than 30 applications)
Statecharts for Alma Common Software (ACS)

- ACS SW Platform: Linux, C/C++, Java, Python, CORBA
- Code Generator tool to generate the MasterComponent application from a Statecharts model
Statecharts for Adaptive Optics

- SPA SW Platform: Linux, C/C++, CORBA, DDS
- Application framework to build Statecharts based application (+ behavioural inheritance feature).
Application Frameworks

- Concentrate on Domain Specific concepts
- Reuse and document of Domain Specific know-how
- Independence of platforms and languages
- Reuse the models on different platforms:
  - Easier to migrate applications
  - Prototyping applications for future platforms
Crane & Dingel paper on differences between Statecharts syntax & semantics: “Not all models are created equal”

State Chart XML
Supported by IBM, HP, Microsoft, Nokia

Interpreted Statecharts

```xml
<state id="A" initial="B">
  <state id="B">
    <transition event="e1" target="C"/>
  </state>
  <state id="C">
    <transition event="e1" target="D"/>
  </state>
</state>
```
UML2SCXML Mapping

```xml
<state> id="" initial="" </state>

<parallel> </parallel>

<transition> event="" guard="" target="" </transition>

<initial> </initial>

<final> </final>

<history> type="deep|shallow" </history>

<onentry> </onentry>

<onexit> </onexit>

<invoke> </invoke>
```
The Data Flow

Platform Independent

M2T Transformation

Platform Specific

COMODO Profile and MetaModel

Cross Platform M2T Transformation (EMF + Xpand)

Clear separation between generated and manually developed code

Generated Application uses SCXML Model and SCXML Engine library
COMODO Profile for UML

![COMODO Profile for UML diagram](image-url)
StopWatch Example
StopWatch Example

```
public class ResetDisplay extends Action {
    public void execute() {
        // ...
    }
}
```

```
public class ActivityCount extends Activity {
    public void run() {
        // ...
    }
}
```
1) Input:
- Model
- TargetPlatform

2) One workflow per platform + SCXML

3) Verify Model is complete and unambiguous

4) Xpand Templates generates the artifacts using Xtend functions

5) Xtend functions to help navigating the model
Future plans

• Development of the C++ SCXML Engine
• Additional Model2Text transformations
  • ESO supported platforms/languages
  • ModelChecker language (Promela/SPIN)

Acknowledgments
Nicolas Beneš, Nicola Migliorini, Alexis Tajeda, Arturo Hoffstadt, Cristian Morales, Joao Lopez