Improvements for Developing Statecharts

Steffen Prochnow

Distributed and Complex Systems Group
VERIMAG Laboratory

DCS Seminar, March, 2009
Simple Statecharts Example

traffic_light

normal

1

Pred

Pgreen

Pgo Pstop

30 sec / Pstop

Cred

Credyel

3 sec / Pgo

Cgreen

Cyellow

3 sec

20 sec

error

1

Poff

Cyellow

2 sec 2 sec

Coff

error

ok

Steffen Prochnow
Statechart Example: Window Wiper

Daimler Chrysler AG, Research REI/SM

Steffen Prochnow

Improvements for Developing Statecharts – 2/31
Motivation and Purpose

Motivation:
- Statecharts possess high complexity (combinations of components, dependencies, system dynamics, concurrency)
- tools for modeling Statecharts provide restricted facilities to enter and understand complex system behavior

Purpose:
- formulation of improvements for easy modeling, analyzing and understanding complex Statecharts
- establishment of these improvements in a highly configurable tool for modeling and simulation
- experimental analysis of automatic Statecharts layout and Statechart editing techniques
Outline

Motivation and Purpose

Difficulties with Statecharts and their Improvements

An Empirical Study

SPEEDS

Summary and Conclusion
Difficulties with Statecharts

1st Observation:
Graphical models are nice to browse, but hard to write.
Creating Graphical Models

State of the Practice

- WYSIWYG editors to create graphical models
- some editors offer alignment tools (ArgoUML)
- in initial phase, often resort to paper and pencil
- creating and maintaining graphical models is time consuming
Editing Action Sequence

1. if needed, create free space.
2. focus on a Statechart element for modification resp. supplementation.
3. apply an editing schema.
4. if needed, rearrange the modified chart to improve readability.
Alternatives to Accelerate Editing

1. add-on to traditional editors
   - create quick-and-dirty graphical model (WYSIWYG) ...
   - ... then apply automated layout

2. macro-based modeling
   - direct manipulation of Statechart structure using keyboard macros
   - employs Statechart production rules
   - ensures syntax-consistency
   - automated layout of graphical elements

3. text-based modeling
   - model synthesis from textual description
   - automated placement of graphical elements
   - separate content from layout
Statechart Layout

Autolayout from Rational Rose

Blobs (Harel and Yashchin, 2002)

Vista (Castelló, Mili, and Tollis, 2002)
Statechart Layout

(a) original layout

(b) after auto-layout

Steffen Prochnow and Reinhard von Hanxleden.
Comfortable modeling of complex reactive systems.
In Proceedings of Design, Automation and Test in Europe (DATE’06), Munich, Germany, March 2006.
Macro-Based Modeling

- identified nine main Statechart editing schemata
- three categories: 1. creation, 2. modification, 3. deletion

(a) insertion of a simple successor state

(b) modification of transition direction

(c) deletion of a Statechart element

(d) insertion of hierarchical successor state

(e) insertion of a parallel region
Macro-Based Modeling

(a) example of applying the “insert simple successor state” schema

(b) navigation with key strokes

Text-Based Modeling

KIel Statechart extension of doT:
- implicit declarations in *dot*,
- hierarchy construction in *Argos*,
- orthogonal construction in *Esterel*, and
- ability to describe different Statechart dialects

(a) KIT description representation

(b) SSM representation.
Language Transformation

(a) Esterel source

Steffen Prochnow, Claus Traulsen, and Reinhard von Hanxleden.
Synthesizing Safe State Machines from Esterel.
Difficulties with Statecharts

2nd Observation:

Graphical languages are appealing, but not effective enough.
Use of Statechart Normal Form

(a) Statecharts with manually placed elements

(b) automatic layout confirming to the Statechart Normal Form
Difficulties with Statecharts

3rd Observation:

Graphical languages are good for understanding structures, but limited for analyzing dynamics.
Dynamic Focus-and-Context Simulation

(a) entering the initial state

(b) cars get a red/yellow light

(c) entering the error state

Statistics for this example:
- 15 states
- 10 state configurations
- 2 views
Difficulties with Statecharts

4th Observation:

Graphical languages are easy to model syntactically correct, but limited for avoiding modeling errors.
Checking Statechart Style

Statechart style guide:
- operational instructions for humans and configuration for automated analysis
- set of 41 wellformedness-, syntactic, and semantic rules
- defines a subset of the language Statechart

Statechart style checking:
- based on defined style guide
- allows to express new rules in OCL or in Java
- theorem prover for more advanced checks

Syntactic rules

Semantic rules

The KIEL Modeling Tool

**Kiel Integrated Environment for Layout**

- employs generic concept of Statecharts
- several layout heuristics
- micro-step simulation and visualization
- has been used in teaching “System Modeling and Synchronous Languages”
- URL: http://rtsys.informatik.uni-kiel.de/~rt-kiel
Outline

Motivation and Purpose

Difficulties with Statecharts and their Improvements

An Empirical Study

SPEEDS

Summary and Conclusion
An Empirical Study

Goals

1. investigation of differences in editing using an WYSIWYG Statechart editor, the KIEL macro editor, and the KIT editor
2. comparison of the readability of Statechart layouts created by the KIEL layouter and other Statechart layouts

Subjects

• graduate-level students attending the lecture “Model-Based Design and Distributed Real-Time Systems”
• conducted two series of experiments, at beginning and end of semester

Support

• ... in experiment design and statistical analysis by Prof. Golz (Department of Psychology, CAU Kiel)
Statechart Editing

Statechart Creation: Hypotheses
Novices will need less time to create a Statechart using the WYSIWYG editor. Advanced will need less time using the KIT editor.

Statechart Modification: Hypothesis
Statechart modification using the KIT editor or the KIEL macro editor is faster than using the WYSIWYG editor.

Results

Distribution of times for creating a new Statechart

Distribution of times for modifying an existing Statechart
Statechart Layout Alternatives

(a) Alternating dot layout (ADL)
(b) ADL backwards (ADBL)
(c) Linear layer layout (LLL)
(d) Alternating linear layout (ALL)
(e) Arbitrary layout (AL)
Statechart Aesthetics: Hypotheses

We expect the best scores for Statecharts laid out according certain layout styles realized by the KIEL Statechart layouter.

Well arranged Statecharts, as laid out by the KIEL Statechart layouter are better (faster) understandable than arbitrary layouts.

Distribution of subjective Statechart layout scores

Distribution of Statechart comprehension times

Steffen Prochnow
Related Work

Statechart layout
▶ David Harel and Gregory Yashchin.
An algorithm for blob hierarchy layout.

Statechart layout framework
▶ Rodolfo Castelló, Rym Mili, and Ioannis G. Tollis.
A framework for the static and interactive visualization for statecharts.

graph Layout
▶ Emden Gansner, Eleftherios Koutsofios, and Stephen North.
Drawing graphs with dot.

semantic focus-and-context
▶ Oliver Köth and Mark Minas.
Structure, Abstraction, and Direct Manipulation in Diagram Editors.

Statechart style checking
▶ Martin Mutz.
Eine durchgängige modellbasierte Entwurfsmethodik für eingebettete Systeme im Automobilbereich.

empirical studies in graphical modeling
When visual programs are harder to read than textual programs.
Outline

Motivation and Purpose

Difficulties with Statecharts and their Improvements

An Empirical Study

SPEEDS

Summary and Conclusion
SPEEDS: SPEculative and EXploratory Design in System Engineering

- EU project
- supports tools for safety-critical embedded system design

**Purpose:**

- component-based construction and analysis of system models

**Approach:**

- **Heterogeneous, Rich Component-based modeling of systems (HRC)**
- rich Components: functional components with static interfaces + non-functional behavior constraints (e.g. the quality of services in terms of timing, resource, etc.)
- constraints: a pair of the form \((Assumption, Promise)\) owned by a component
- a promise is guaranteed only if the corresponding assumption about the environment is fulfilled
SPEEDS Infrastructure

Consortium: potential end-users from aerospace and automotive industries, tool-vendors, and European research institutes:

- Airbus Deutschland GmbH
- Airbus France
- Israel Aerospace Industries Ltd. (formerly Israel Aircraft Industrie Ltd.)
- Carmeq
- Robert Bosch GmbH
- Magna Powertrain Engineering Center
- Knorr-Bremse Fekrendszerek Kft
- SAAB AB
- EADS
- Telelogic Israel Ltd.(formerly I-Logix)
- Esterel Technologies SA
- Extessy
- GEENSYS (formerly TNI)
- Université Joseph Fourier - VERIMAG
- PARADES GEIE
- Kuratorium OFFIS E.V.
- INRIA

SPEEDS Tool Interaction:
VERIMAG Contribution—Dominance Analysis

Purpose:
- verify compositional context dependent properties of systems

Approach:
- verification of dominance between contracts

\[(A_1, P_1) \parallel (A_2, P_2) \text{ dominates } (A, P) \text{ w. r. t. } \text{System}\]

\- A, A_1, A_2: Assumptions
\- P, P_1, P_2: Promises
\- (A, P), (A_1, P_1), (A_2, P_2): Contracts
**Dominance Tools**

**HRC**: component specification
- models provided by project partners

**CSL**: Constraint Specification Language

**Maude**: language for equational and rewriting logic specification of systems
- used for checking deadlock freedom
- deadlock found: no dominance
- checking output: location of the problem
• bug fixing
• improve expressiveness of Maude output
• visual debugging using graphical Statecharts representations
• code generation using BIP for hosted simulation
• syntactical simplification of transformed Statecharts
Summary and Conclusion

Summary

• KIEL: platform for experimenting with the pragmatics of graphical modeling
  • editing alternatives to WYSIWYG
  • paradigm to simulate complex Statecharts: semantic focus-and-context
  • uniform framework for robustness checking
  • empirical study

• SPEEDS: model transformation and dominance checking

Conclusion

• experimental studies show
  – efficiency of Statechart layout
  – good performance in modifying Statechart structure

• Statechart layout (Secondary Notation) can improve development process (comp. SPEEDS)

• improvements combine the best of two worlds

Thanks!
Questions or comments?
## Appendix: Simulation and visualization performance of KIEL

<table>
<thead>
<tr>
<th>MODEL</th>
<th>GRAPHICAL ELEMENTS</th>
<th>STATE SPACE</th>
<th>OCCUPIED AREA</th>
<th>COMPUTATION TIMES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>States</td>
<td>Transitions</td>
<td>Total</td>
<td>Configurations</td>
</tr>
<tr>
<td>traffic-light</td>
<td>15</td>
<td>25</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>window-wiper</td>
<td>36</td>
<td>82</td>
<td>118</td>
<td>80</td>
</tr>
<tr>
<td>bintree-1</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>bintree-2</td>
<td>9</td>
<td>15</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>bintree-3</td>
<td>21</td>
<td>35</td>
<td>56</td>
<td>8</td>
</tr>
<tr>
<td>bintree-4</td>
<td>45</td>
<td>75</td>
<td>120</td>
<td>16</td>
</tr>
<tr>
<td>bintree-5</td>
<td>93</td>
<td>155</td>
<td>248</td>
<td>32</td>
</tr>
<tr>
<td>bintree-6</td>
<td>189</td>
<td>315</td>
<td>504</td>
<td>64</td>
</tr>
<tr>
<td>bintree-7</td>
<td>381</td>
<td>635</td>
<td>1016</td>
<td>128</td>
</tr>
<tr>
<td>bintree-8</td>
<td>765</td>
<td>1275</td>
<td>2040</td>
<td>256</td>
</tr>
<tr>
<td>bintree-9</td>
<td>1533</td>
<td>2555</td>
<td>4088</td>
<td>512</td>
</tr>
<tr>
<td>quadtree-1</td>
<td>10</td>
<td>11</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td>quadtree-2</td>
<td>50</td>
<td>55</td>
<td>105</td>
<td>64</td>
</tr>
<tr>
<td>quadtree-3</td>
<td>210</td>
<td>231</td>
<td>441</td>
<td>16384</td>
</tr>
<tr>
<td>quadtree-4</td>
<td>850</td>
<td>935</td>
<td>1785</td>
<td>1073741824</td>
</tr>
</tbody>
</table>

Steffen Prochnow

Improvements for Developing Statecharts – 31/31
## Appendix: Experimental results of SSM synthesis

<table>
<thead>
<tr>
<th>Model</th>
<th>Lines of code (loc)</th>
<th>States</th>
<th>Pseudo-states</th>
<th>Transitions</th>
<th>Total graphical elements (ges)</th>
<th>ges/loc</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABRO</td>
<td>7</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>32</td>
<td>4.57</td>
<td>861</td>
</tr>
<tr>
<td>schizophrenia</td>
<td>10</td>
<td>13</td>
<td>9</td>
<td>14</td>
<td>36</td>
<td>3.60</td>
<td>838</td>
</tr>
<tr>
<td>reincarnation</td>
<td>25</td>
<td>27</td>
<td>17</td>
<td>28</td>
<td>72</td>
<td>2.88</td>
<td>642</td>
</tr>
<tr>
<td>jacky1</td>
<td>27</td>
<td>31</td>
<td>19</td>
<td>33</td>
<td>83</td>
<td>3.07</td>
<td>913</td>
</tr>
<tr>
<td>runner</td>
<td>55</td>
<td>39</td>
<td>24</td>
<td>42</td>
<td>105</td>
<td>1.91</td>
<td>725</td>
</tr>
<tr>
<td>tokenring3</td>
<td>79</td>
<td>77</td>
<td>61</td>
<td>91</td>
<td>229</td>
<td>2.90</td>
<td>733</td>
</tr>
<tr>
<td>greycounter</td>
<td>82</td>
<td>211</td>
<td>148</td>
<td>254</td>
<td>613</td>
<td>7.48</td>
<td>789</td>
</tr>
<tr>
<td>abcd</td>
<td>101</td>
<td>231</td>
<td>130</td>
<td>250</td>
<td>611</td>
<td>6.05</td>
<td>943</td>
</tr>
<tr>
<td>tokenring10</td>
<td>247</td>
<td>245</td>
<td>194</td>
<td>294</td>
<td>733</td>
<td>2.97</td>
<td>1267</td>
</tr>
<tr>
<td>mejia</td>
<td>555</td>
<td>374</td>
<td>246</td>
<td>414</td>
<td>1034</td>
<td>1.86</td>
<td>3085</td>
</tr>
<tr>
<td>tcint</td>
<td>687</td>
<td>475</td>
<td>285</td>
<td>543</td>
<td>1303</td>
<td>1.90</td>
<td>3382</td>
</tr>
<tr>
<td>atds-100</td>
<td>948</td>
<td>961</td>
<td>558</td>
<td>1092</td>
<td>2611</td>
<td>2.75</td>
<td>7046</td>
</tr>
<tr>
<td>ww</td>
<td>1088</td>
<td>342</td>
<td>228</td>
<td>386</td>
<td>965</td>
<td>0.89</td>
<td>4470</td>
</tr>
<tr>
<td>tokenring50</td>
<td>1207</td>
<td>1205</td>
<td>954</td>
<td>1454</td>
<td>3613</td>
<td>2.99</td>
<td>6608</td>
</tr>
<tr>
<td>tokenring100</td>
<td>2407</td>
<td>2405</td>
<td>1904</td>
<td>2904</td>
<td>7213</td>
<td>3.00</td>
<td>20910</td>
</tr>
<tr>
<td>mcalo</td>
<td>7269</td>
<td>5159</td>
<td>3931</td>
<td>5947</td>
<td>15037</td>
<td>2.07</td>
<td>61510</td>
</tr>
</tbody>
</table>

*Improvements for Developing Statecharts – 31/31*
Appendix: Static and Dynamic View Areas

![Graph showing the occupied view area for different Statechart models. The x-axis represents the Statechart model, while the y-axis shows the occupied view area in pixels. The graph includes lines for the area of Statechart static view, area of Statechart dynamic view, and a dotted line for a 1024x768 pixel view.](image-url)
Appendix: Model Sizes before and after Optimization

The diagram shows the total number of graphical elements in various Esterel models before and after transformation and optimization. The models include ABRO, schizophrenia, reincarnation, jacky1, runner, tokenring3, abcd, greycounter, tokenring10, ww, mejia, tcint, atds−100, tokenring50, tokenring100, and mca200. The chart compares the number of elements after transformation and after optimization.
Dominance Example—UTS

Assumption: Whenever [reqS] occurs [3#reqS] does not occur during following [(10, min)]
Promise: Whenever [reqS] occurs [servedS] occurs within [(15, min)]
Assumption: whenever \([\text{reqCS}]\) occurs \([\#2\text{reqCS}]\) does not occur during following \([(2,\text{min})]\)

Promise: whenever \([\text{reqCS}]\) occurs \([\text{dispatchCS}]\) occurs within \([(2,\text{min})]\)
Dominance Example—UTS

Assumption: whenever \([\text{reqCS}]\) occurs \([\text{dispatchCS}]\) does not occur during following \([(2, \text{min})]\)

Promise: whenever \([\text{reqCS}]\) occurs \([\text{dispatchCS}]\) does not occur during following \([(2, \text{min})]\)

Steffen Prochnow

Improvements for Developing Statecharts – 31/31
Dominance Example—UTS

Analysis Idea:

\((A_{CS}, P_{CS}) \parallel (A_{Car}, P_{Car})\) dominates \((A_{UTS}, P_{UTS})\) w. r. t. UTS