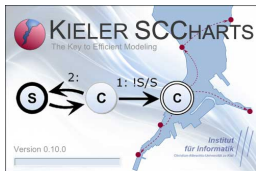


Compiling & Simulating SCCharts /w KIELER

<http://www.sccharts.com>

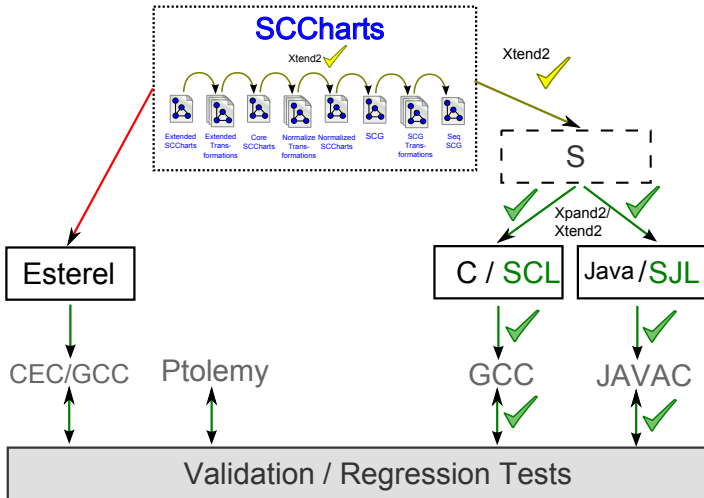
Christian Motika

Real-Time Systems and Embedded Systems Group
Department of Computer Science
Christian-Albrechts-Universität zu Kiel, Germany

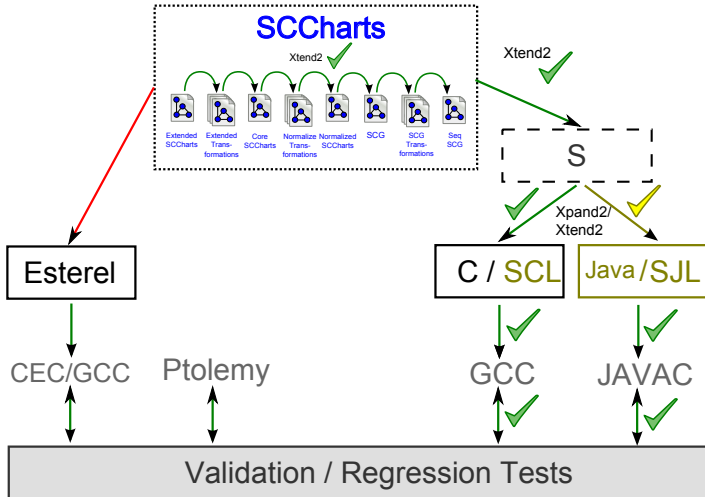


Oberseminar, SS 14
27.08.2014

Context WS 13/14



Now



Overview

- ▶ KIELER Compiler (KiCo)

Overview

- ▶ KIELER Compiler (KiCo)
- ▶ Online Compiler

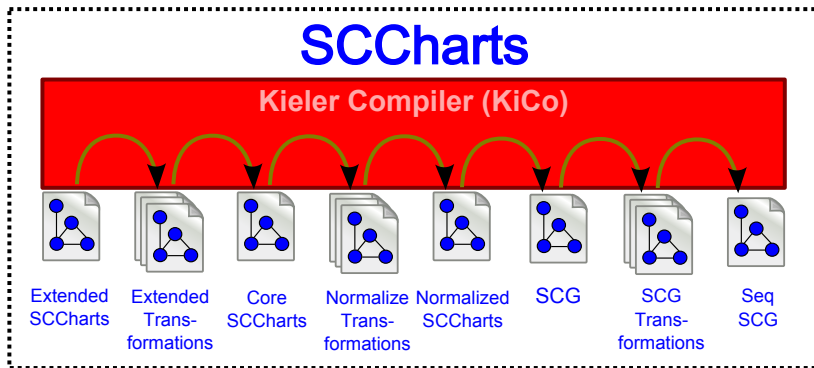
Overview

- ▶ KIELER Compiler (KiCo)
- ▶ Online Compiler
- ▶ Compiling & Simulating SCCharts
 - ▶ SCCharts Transformations
 - ▶ Simulation
 - ▶ Validation

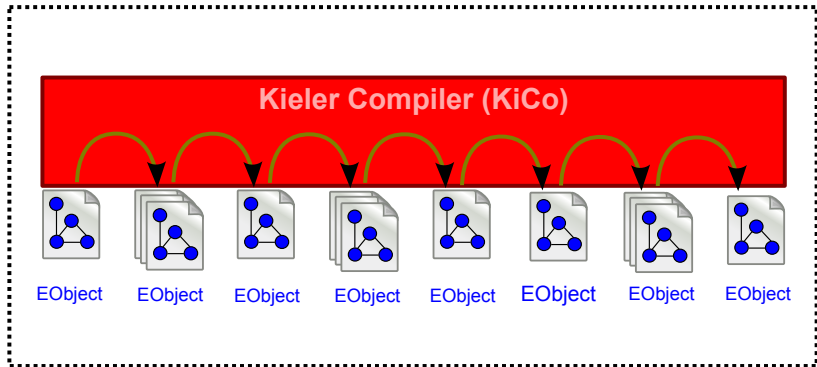
Overview

- ▶ KIELER Compiler (KiCo)
- ▶ Online Compiler
- ▶ Compiling & Simulating SCCharts
 - ▶ SCCharts Transformations
 - ▶ Simulation
 - ▶ Validation
- ▶ Summary & Future Work

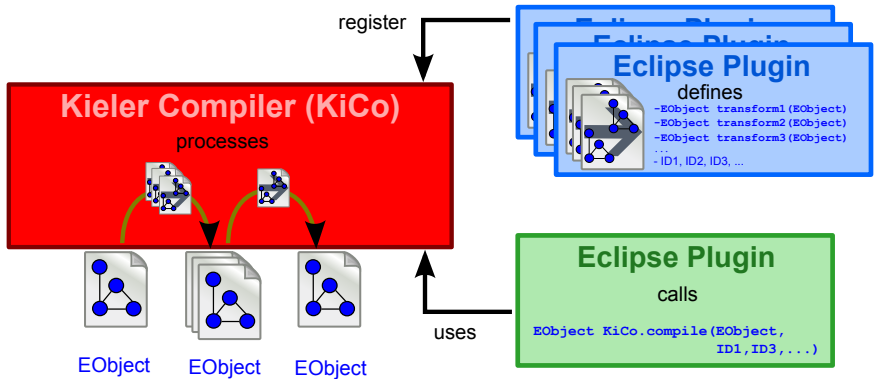
KIELER Compiler (KiCo) – Context (Recall)



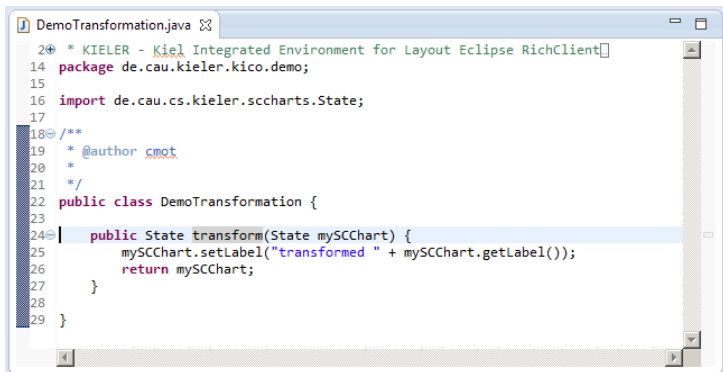
KIELER Compiler (KiCo) – Context (Recall)



KIELER Compiler (KiCo) – Basic Idea (Recall)



KIELER Compiler – Register (1/3)



```
DemoTransformation.java
20 * KIELER - Kiel Integrated Environment for Layout Eclipse RichClient
14 package de.cau.kieler.kico.demo;
15
16 import de.cau.cs.kieler.sccharts.State;
17
18 /**
19  * @author cmot
20  *
21  */
22 public class DemoTransformation {
23
24     public State transform(State mySCChart) {
25         mySCChart.setLabel("transformed " + mySCChart.getLabel());
26         return mySCChart;
27     }
28
29 }
```

- ▶ 1. Create a transformation method
 - ▶ (a) Takes, (b) modifies, and (c) returns an EObject

KIELER Compiler – Register (2/3)

The screenshot shows the 'Extensions' configuration window in the KIELER IDE. The main area is titled 'All Extensions' and contains a tree view showing the extension 'DEMO (transformationMethod)' under the package 'de.cau.cs.kieler.kico.transformation'. Below the tree are 'Add...' and 'Remove' buttons. To the right, the 'Extension Element Details' panel is active, showing the following configuration:

- class*: de.cau.kieler.kico.demo.DemoTransformation (with a 'Browse...' button)
- method*: transform
- id*: DEMO
- name: DEMO
- dependencies: (empty)

At the bottom of the window, there are several tabs: Overview, Dependencies, Runtime, Extensions (selected), Extension Points, Build, MANIFEST.MF, build.properties, and plugin.xml.

- ▶ 2. Register the transformation in KiCo
 - ▶ (a) Choose class, (b) choose method, and (c) choose ID
 - ▶ Optionally: (d) Choose different display name, (e) choose dependencies

KIELER Compiler – Use (1/1)

```
State result =  
(State) KielerCompiler.compile(''DEMO'',mySCChart).getEObject();
```

KIELER Compiler – Register (3/3)

The screenshot shows the Eclipse IDE's 'Extensions' dialog for the plugin `de.cau.cs.kieler.scharts.text.ui`. The dialog is divided into two main sections:

- All Extensions:** This section allows defining extensions for the plug-in. It features a search filter 'type filter text' and a list of extension points. The selected extension point is `SCCHARTS_SIMULATION, SCCHAR` under the package `de.cau.cs.kieler.kico.ui.transformation`. Buttons for 'Add...' and 'Remove' are visible.
- Extension Element Details:** This section is used to set the properties of the extension element. It specifies the 'editor' as `de.cau.cs.kieler.scharts.text.sct.Sct` and the 'transformations' as `DEMO, SCCHARTS_SIMULATION, SCCHARTS_SIMI`.

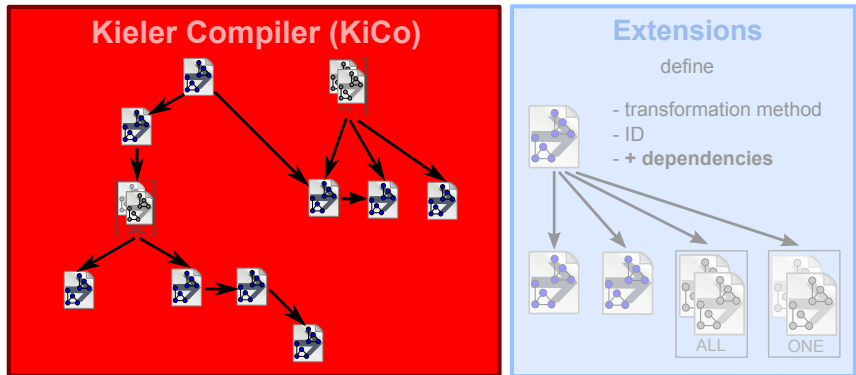
At the bottom of the dialog, a tabbed interface is visible with the following tabs: Overview, Dependencies, Runtime, **Extensions**, Extension Points, Build, MANIFEST.MF, plugin.xml, and build.properties.

- ▶ 3. Tell KiCo to show transformation for the editor
 - ▶ Use the extension point and transformation ID

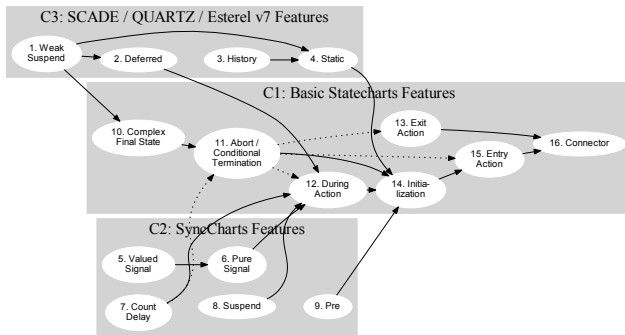
KiCo Transformation Registration Demo

LIVE DEMO

KIELER Compiler – Internals (Recall)



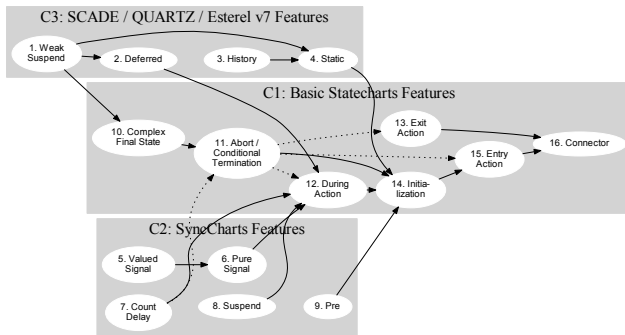
SCCharts – Extended Feature Dependencies



[ISoLA14]

- Types: **Produced by** (solid) & **not handled by** (dashed)

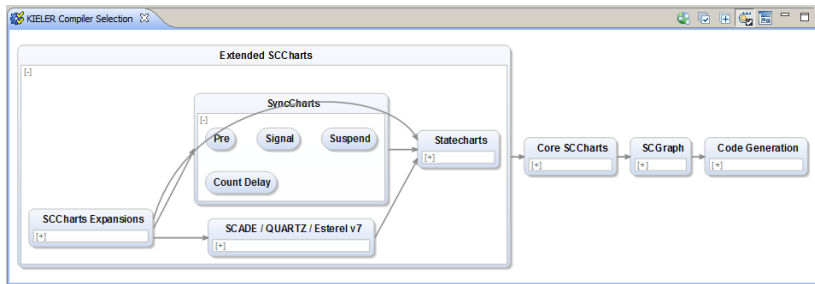
SCCharts – Extended Feature Dependencies



[ISoLA14]

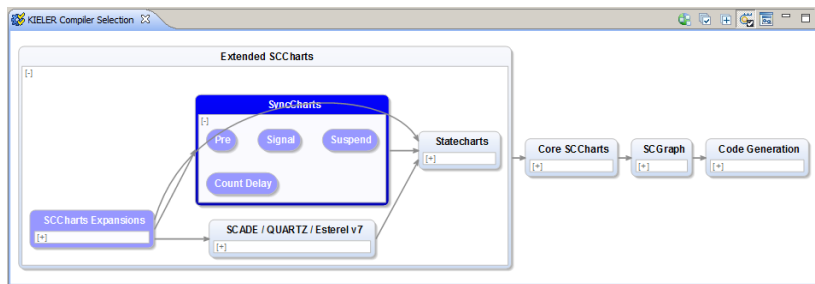
- ▶ Types: **Produced by** (solid) & **not handled by** (dashed)
- ▶ Currently: Both considered as *dependency*

SCCharts – KiCo Selection View



- ▶ Static dependencies

SCCharts – KiCo Selection View

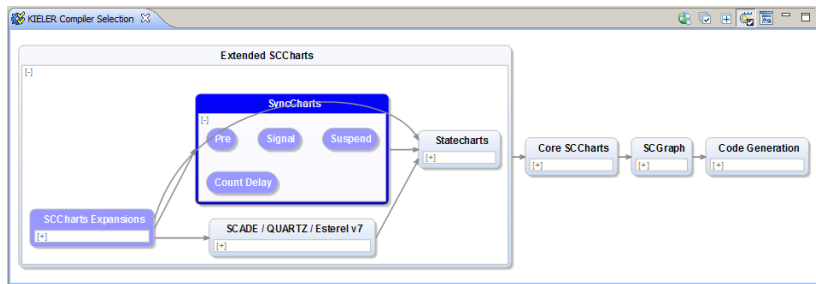


- ▶ Static dependencies
- ▶ Automatic selection of **required** transformations
 - ▶ Groups
 - ▶ Dependency relations (possibly *not handled by*)

KiCo Selection View Demo

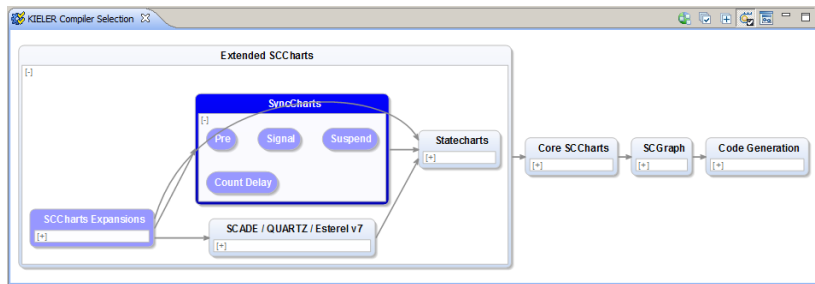
LIVE DEMO

SCCharts – KiCo Selection View Flaws



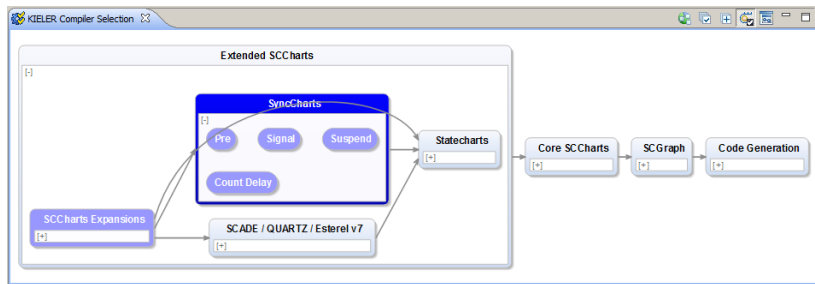
- ▶ Automatic selection of **required** transformations
 - ▶ ☹ Possibly **not handled by** relations

SCCharts – KiCo Selection View Flaws



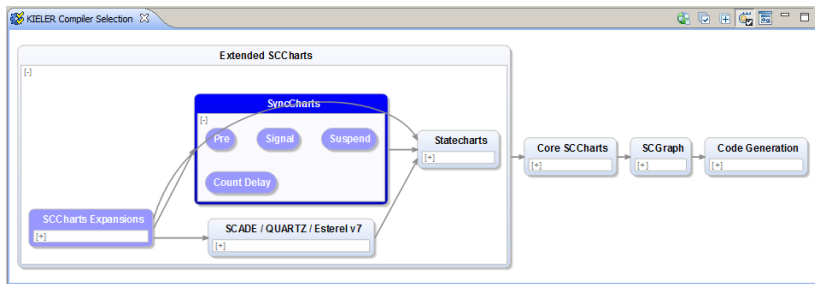
- ▶ Automatic selection of **required** transformations
 - ▶ 😞 Possibly **not handled by** relations
 - ⇒ 😊 Distinguish different types in implementation

SCCharts – KiCo Selection View Flaws



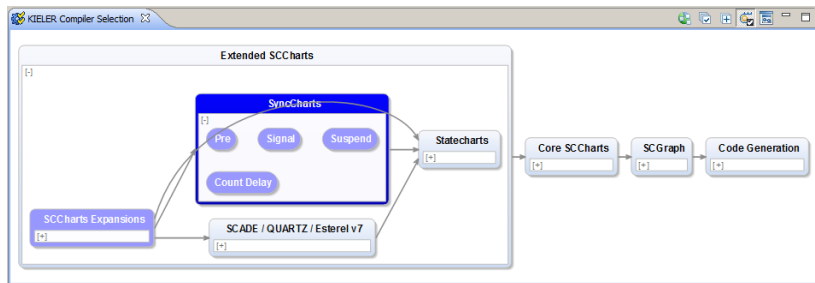
- ▶ Automatic selection of **required** transformations
 - ▶ 😞 Possibly **not handled by** relations
⇒ 😊 Distinguish different types in implementation
- ▶ Static dependencies
 - ▶ 😞 Not considering features **really** in the model

SCCharts – KiCo Selection View Flaws



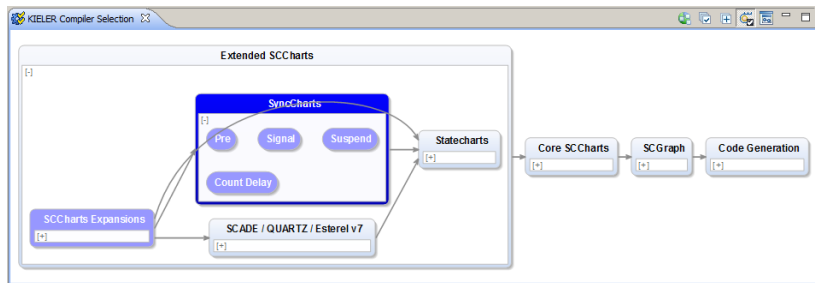
- ▶ Automatic selection of **required** transformations
 - ▶ 😞 Possibly **not handled by** relations
⇒ 😊 Distinguish different types in implementation
- ▶ Static dependencies
 - ▶ 😞 Not considering features **really** in the model ⇒ 😊 Adaptive

SCCharts – KiCo Selection View Flaws



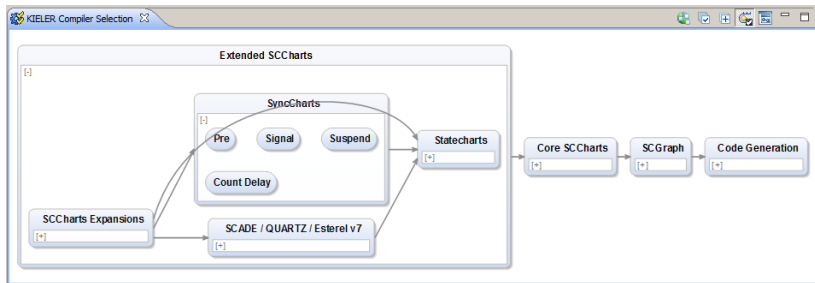
- ▶ Automatic selection of **required** transformations
 - ▶ 😞 Possibly **not handled by** relations
⇒ 😊 Distinguish different types in implementation
- ▶ Static dependencies
 - ▶ 😞 Not considering features **really** in the model ⇒ 😊 Adaptive
 - ▶ 😞 Not considering features **really** produced

SCCharts – KiCo Selection View Flaws



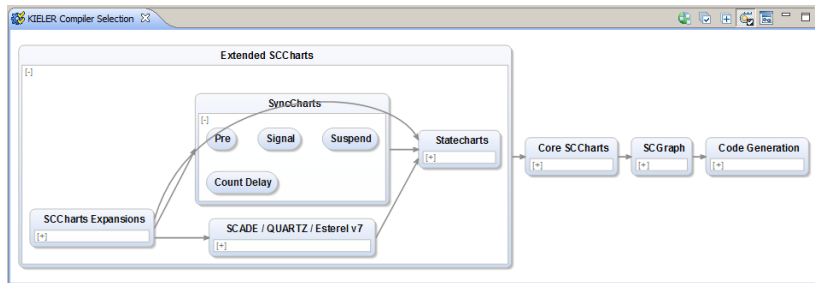
- ▶ Automatic selection of **required** transformations
 - ▶ ☹ Possibly **not handled by** relations
⇒ 😊 Distinguish different types in implementation
- ▶ Static dependencies
 - ▶ ☹ Not considering features **really** in the model ⇒ 😊 Adaptive
 - ▶ ☹ Not considering features **really** produced ⇒ 😊 Dynamic

KiCo Selection View Additional Features



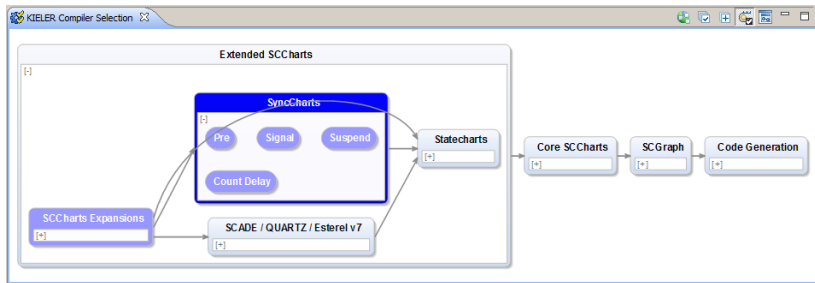
- ▶ Select all/none button

KiCo Selection View Additional Features



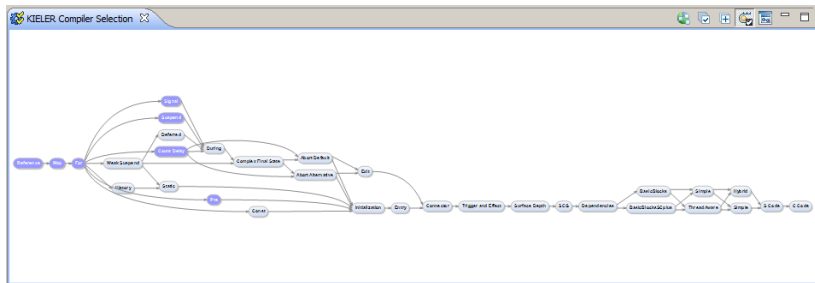
- ▶ Select all/none button
- ▶ Expand or collapse all button

KiCo Selection View Additional Features



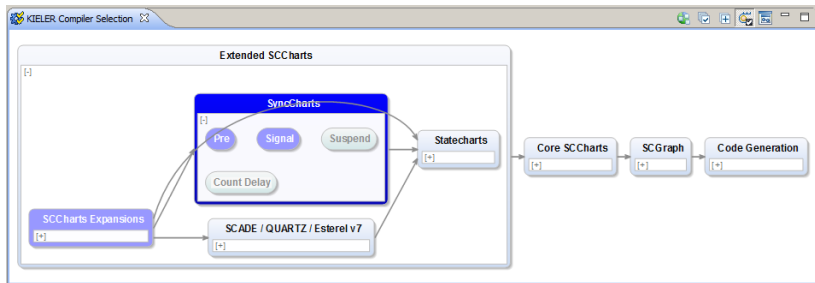
- ▶ Select all/none button
- ▶ Expand or collapse all button
- ▶ Autoselection button

KiCo Selection View Additional Features



- ▶ Select all/none button
- ▶ Expand or collapse all button
- ▶ Autoselection button
- ▶ Hierarchical and flat view

KiCo Selection View Additional Features



- ▶ Select all/none button
- ▶ Expand or collapse all button
- ▶ Autoselection button
- ▶ Hierarchical and flat view
- ▶ Disable transformation (double click)

Overview

- ▶ KIELER Compiler (KiCo)
- ▶ Online Compiler
- ▶ Compiling & Simulating SCCharts
 - ▶ SCCharts Transformations
 - ▶ Simulation
 - ▶ Validation
- ▶ Summary & Future Work

Online Compiler - KIELER Wiki - RTSYS Confluence - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Online Compiler - KIELER Wiki - RTSYS Conflu... +

Back Forward Reload Home Zoom Out Zoom In Print Downloads Stop

https://www.sccharts.com/online-compiler/

Spaces Browse

Log in Sign Up

Panic! Get me back to the home page!

Navigation

Search

- Overview
- KIELER Layout
- KIELER Pragmatics
- KIELER Semantics / SCCharts
 - Environment Visualization (KEV)
 - Execution Manager (KEM)
 - Kieler Compiler
 - Lego On-Line Testing System (KLOTS)
 - Leveraging Ptolemy Semantics (KiePto)
- SCCharts
 - Command Line Compiler
 - Command Line Rendering
 - Examples
 - Online Compiler
 - PLD/14 Artifact on SCCharts
 - Quick Start Guide
 - Textual SCCharts Language SCT
- The SC Language (SCL)
- Transformation Mapping (KTM)
- UML State Machine Simulation/Model Checking with Maude
- KIELER Demonstrators
- Development
- Administration
- Meeting notes

```
graph LR; S((S)) -- "2:" --> S; C1((C)) -- "1: !S/S" --> C1; S --> C1; C1 --> C2((C));
```

Sequentially Constructive Charts – SCCharts

ONLINE COMPILER

If after a while there is still no compilation result, please select another server or local host below.

1. Textual SCChart (SCT) [\(see here\)](#)

```
%%Layout
schart: ABO {
  input bool A;
  input bool B;
  input bool S;
  output bool O = false;
  region Main:
    initial state ABO {
      initial state WaitAB {
        region HandleA:
          initial state WA
          --> DA WITH A;

```

2. Compile Transformations (choose separated)

Select Transformations: Strict (only apply the listed transformations)

Textual output (sample only)

Graphical output (SVG) (compile and render)

Graphical output (PNG) (compile and render)

Image Quality: Size:

Advanced Options

3. Other Options

Main Server (fastest)

Backup Server 1 (slowest)

Backup Server 2 (slowest)

Local Host (fastest)

(KIELER HTTP servers must run locally, see here and here)

4. Server (compile/render)

```
%%Layout
schart: ABO {
  input bool A;
  input bool B;
  input bool S;
  output bool O = false;
  bool _trig;
  region Main:
    initial state ABO {
      entry / _trig = false;
      initial state WaitAB {
        region HandleA:
          initial state WA
          --> DA WITH _trig
          --> DA WITH A;
        final state DA:
          region HandleB:
            initial state WB
            --> DB WITH _trig
            --> DB WITH B;
          final state DB:
            --> _C;
          state Done
          --> _Aborted WITH _trig;
        final state _Aborted:
          connector state _C
          --> _Aborted WITH _trig
          --> Done / O = true;
      region _Ctrl:

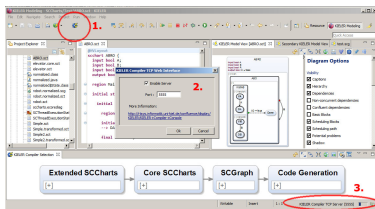
```

Powered by a free Atlassian Confluence Open Source Project License granted to Kiel University Comp. Sci. Evaluate Confluence today.
Powered by Atlassian Confluence 5.4.3, Team Collaboration Software - Report a bug - Atlassian News

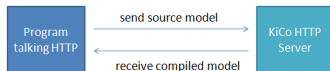
KiCo Http Server

The screenshot displays the KIELER Modeling IDE interface. The toolbar at the top contains a red circle around the 'Run' button, labeled '1.'. A dialog box titled 'KIELER Compiler TCP Web Interface' is open in the center, with 'Enable Server' checked and 'Port: 5555' entered, labeled '2.'. Below the IDE, a workflow diagram shows four steps: 'Extended SCCharts', 'Core SCCharts', 'SCGraph', and 'Code Generation', each with a '+' button, labeled '3.'. The status bar at the bottom right shows 'KIELER Compiler TCP Server (5555)' running.

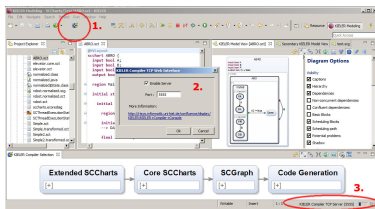
KiCo Http Server (cont'd)



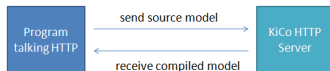
- ▶ Can be used Using HTTP protocol:



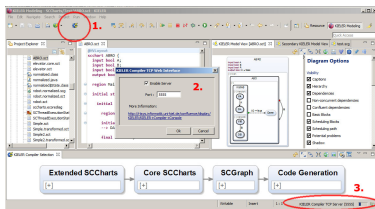
KiCo Http Server (cont'd)



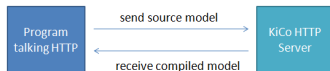
- ▶ Can be used Using HTTP protocol:
- ▶ Support GET and POST requests



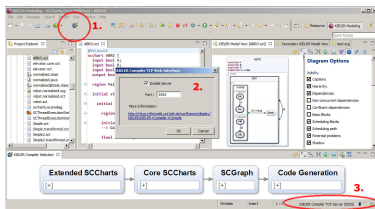
KiCo Http Server (cont'd)



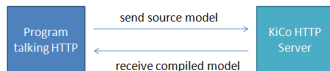
- ▶ Can be used Using HTTP protocol:
- ▶ Support GET and POST requests
- ▶ Request: application/x-www-form-urlencoded
 - ▶ Form data: model, transformations



KiCo Http Server (cont'd)



- ▶ Can be used Using HTTP protocol:
- ▶ Support GET and POST requests
- ▶ Request: application/x-www-form-urlencoded
 - ▶ Form data: model, transformations
- ▶ Response: text/plain

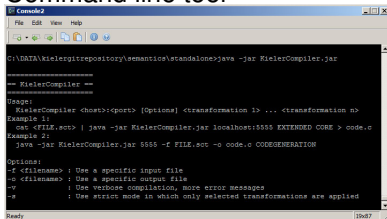


KiCo Http Server Usage

- ▶ Web site/application

KiCo Http Server Usage

- ▶ Web site/application
- ▶ Command line tool



```

C:\DATA\kielergitrepository\semantics\standalone>java -jar KielerCompiler.jar

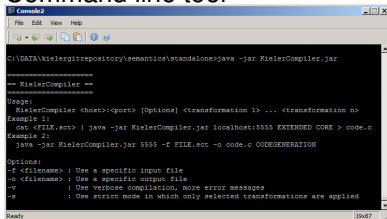
=====
-- KielerCompiler --
=====
Usage:
  KielerCompiler <host>:<port> [Options] <transformation 1> ... <transformation n>
Example 1:
  cat <FILE.sc> | java -jar KielerCompiler.jar localhost:5555 EXTENDED CORE > code.c
Example 2:
  java -jar KielerCompiler.jar 5555 -f FILE.sc -o code.c CODEGENERATION

Options:
  -f <filename> : Use a specific input file
  -o <filename> : Use a specific output file
  -v            : Use verbose compilation, more error messages
  -s            : Use strict mode in which only selected transformations are applied

```

KiCo Http Server Usage

- ▶ Web site/application
- ▶ Command line tool



```
C:\DATA\kielergitrepository\semantics\standalone>java -jar KielerCompiler.jar

=====
-- KielerCompiler --
=====
Usage:
KielerCompiler <host>[:<port>] [Options] <transformation 1> ... <transformation n>
Example 1:
cat <FILE.sc> | java -jar KielerCompiler.jar localhost:5555 EXTENDED CORE > code.c
Example 2:
java -jar KielerCompiler.jar 5555 -f FILE.sc -o code.c CODEGENERATION

Options:
-f <filename> : Use a specific input file
-o <filename> : Use a specific output file
-v           : Use verbose compilation, more error messages
-s           : Use strict mode in which only selected transformations are applied

Ready
```

- ▶ Use any other programming language understanding HTTP
 - ▶ Javascript / WWW
 - ▶ Java
 - ▶ C++
 - ▶ Android
 - ▶ ...

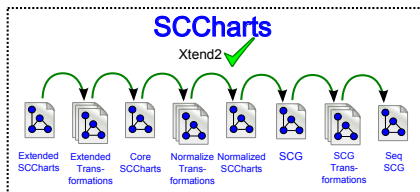
SCCharts Online Compiler Demo

LIVE DEMO

Overview

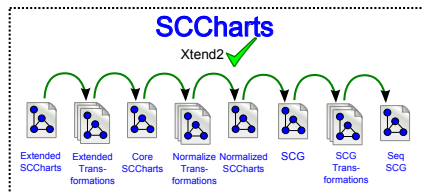
- ▶ KIELER Compiler (KiCo)
- ▶ Online Compiler
- ▶ Compiling & Simulating SCCharts
 - ▶ SCCharts Transformations
 - ▶ Simulation
 - ▶ Validation
- ▶ Summary & Future Work

SCCharts Compilation - What happened?



- ▶ Revised many transformations
 - ▶ Abort
 - ▶ Count Delay
 - ▶ During
 - ▶ ...

SCCharts Compilation - What happened?



- ▶ Revised many transformations
 - ▶ Abort
 - ▶ Count Delay
 - ▶ During
 - ▶ ...
- ▶ Added new transformations
 - ▶ Const
 - ▶ Reference
 - ▶ Map
 - ▶ For

Simulation C Code Wrapper

```
CSimulation.xtend
---
131 // Generate the main function.
132 def mainFunction(State scchart) {
133   ..
134 int main(int argc, const char* argv[]) {
135   reset();
136   output = cJSON_CreateObject();
137   readInputs();
138   tick();
139   while(1) {
140     writeOutputs();
141     char* outString = cJSON_Print(output);
142     strip_white_spaces(outString);
143     printf("%s\n", outString);
144     fflush(stdout);
145     output = cJSON_CreateObject();
146     readInputs();
147     tick();
148   }
149 }
```

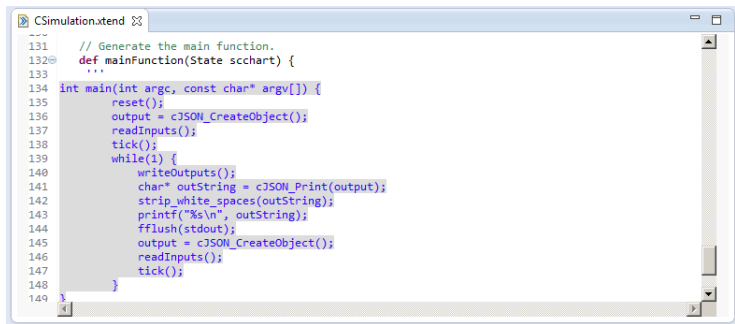
- ▶ 150 lines of code

Simulation C Code Wrapper

```
CSimulation.xtend
---
131 // Generate the main function.
132 def mainFunction(State scchart) {
133   ...
134 int main(int argc, const char* argv[]) {
135   reset();
136   output = cJSON_CreateObject();
137   readInputs();
138   tick();
139   while(1) {
140     writeOutputs();
141     char* outString = cJSON_Print(output);
142     strip_white_spaces(outString);
143     printf("%s\n", outString);
144     fflush(stdout);
145     output = cJSON_CreateObject();
146     readInputs();
147     tick();
148   }
149 }
```

- ▶ 150 lines of code
- ▶ Generic main function

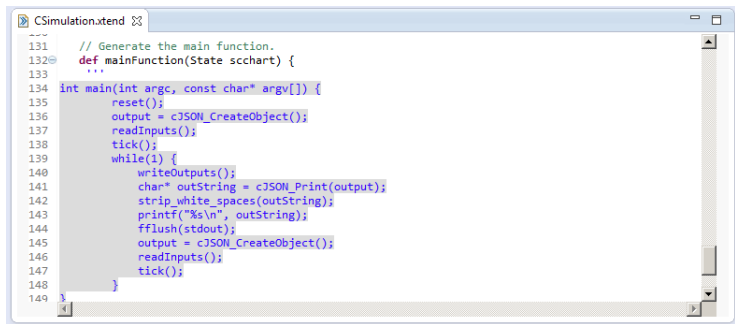
Simulation C Code Wrapper



```
CSimulation.xtend
---
131 // Generate the main function.
132 def mainFunction(State scchart) {
133     ...
134 int main(int argc, const char* argv[]) {
135     reset();
136     output = cJSON_CreateObject();
137     readInputs();
138     tick();
139     while(1) {
140         writeOutputs();
141         char* outString = cJSON_Print(output);
142         strip_white_spaces(outString);
143         printf("%s\n", outString);
144         fflush(stdout);
145         output = cJSON_CreateObject();
146         readInputs();
147         tick();
148     }
149 }
```

- ▶ 150 lines of code
- ▶ Generic main function
- ▶ Header includes `scchart.c`

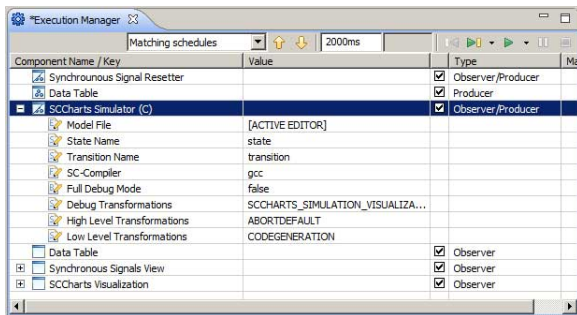
Simulation C Code Wrapper



```
CSimulation.xtend
---
131 // Generate the main function.
132 def mainFunction(State scchart) {
133   ...
134 int main(int argc, const char* argv[]) {
135   reset();
136   output = cJSON_CreateObject();
137   readInputs();
138   tick();
139   while(1) {
140     writeOutputs();
141     char* outString = cJSON_Print(output);
142     strip_white_spaces(outString);
143     printf("%s\n", outString);
144     fflush(stdout);
145     output = cJSON_CreateObject();
146     readInputs();
147     tick();
148   }
149 }
```

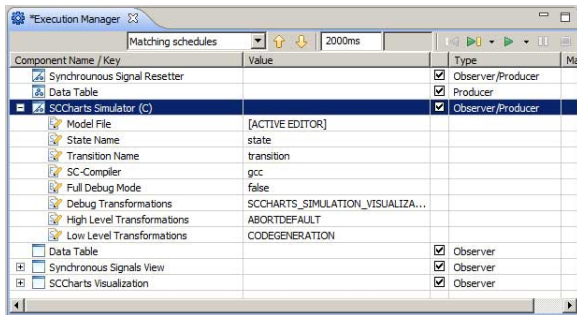
- ▶ 150 lines of code
- ▶ Generic main function
- ▶ Header includes `scchart.c`
- ▶ `readInputs()` and `writeOutputs` generated from SCChart

SCCharts Simulation KIEM DataComponent



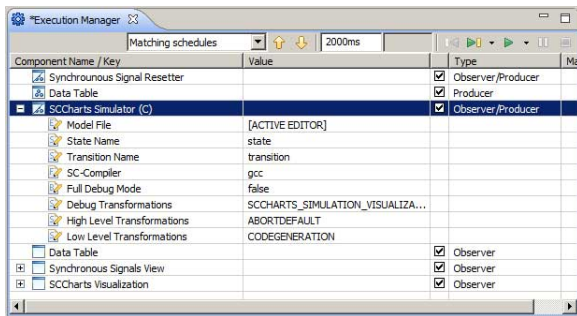
- ▶ Define high-level and low-level transformations

SCCharts Simulation KIEM DataComponent



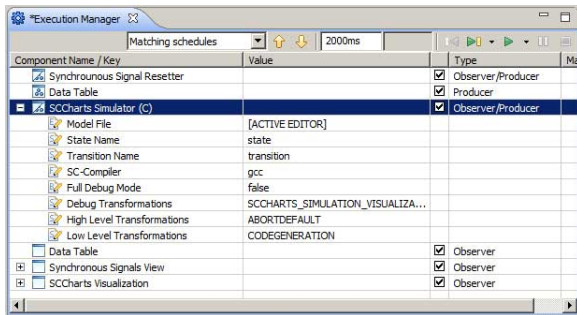
- ▶ Define high-level and low-level transformations
- ▶ Wrapper generated after high-level transformations

SCCharts Simulation KIEM DataComponent



- ▶ Define high-level and low-level transformations
- ▶ Wrapper generated after high-level transformations
- ▶ Debug mode (turn on/off visualisation)

SCCharts Simulation KIEM DataComponent



- ▶ Define high-level and low-level transformations
- ▶ Wrapper generated after high-level transformations
- ▶ Debug mode (turn on/off visualisation)
- ▶ Define debug transformations

SCCharts Simulation Demo

LIVE DEMO

Broken Build 😊

The screenshot shows the Bamboo web interface for the project 'KIELER Semantics - Continuous Plugins (Kepler 4.3)'. The top navigation bar includes 'Build', 'Deploy', and 'Reports'. The main content area shows the 'Plan summary' for the 'master' branch, which is continuously building all semantics plug-ins upon repository changes. The 'Recent history' table lists 8 builds, all of which failed (1 of 1 failed). The 'Plan statistics' section shows 25 builds, 40% successful, and a 6m average duration. The 'Branches' section lists four branches: ssm-master (35 builds, 100% success), als-master (9 builds, 100% success), release-0.10 (17 builds, 100% success), and cmot-master (202 builds, 0% success).

Build ID	Changes by	Time	Status
#988	Changes by Christian Schneider	4 days ago	1 of 1 failed
#987	Changes by Steven Smyth	4 days ago	1 of 1 failed
#986	Changes by Steven Smyth	5 days ago	1 of 1 failed
#985	Changes by Christian Motika	1 week ago	1 of 1 failed
#984	Changes by Christian Motika	1 week ago	1 of 1 failed
#983	Changes by Christian Motika	1 week ago	1 of 1 failed
#982	Manual run by Christian Motika	1 week ago	1 of 1 failed
#981	Changes by Christian Motika	1 week ago	1 of 1 failed

Broken Build 😊

The screenshot shows a web browser window displaying the Atlassian Bamboo build interface. The page title is "KIELER Semantics - Continuous Plugins (Kepler 4.3) 985: Test results - KIELER Bamboo - Mozilla Firefox". The browser address bar shows the URL "rtsys.informatik.uni-koel.de/bamboo/browse/KISEMA-PLUGINS-985/test". The page header includes the Bamboo logo and navigation links for "Build", "Deploy", and "Reports". Below the header, the build details for "Build #985" are shown, including the project path "KIELER Semantics / Continuous Plugins (Kepler 4.3)" and the build type "master". A red banner indicates that the build failed. The "Test results" section shows "1 test failed" and "5 seconds taken in total". A table lists the failing test: "SCChartsSimSAutomatedJUnitTest /test-sccharts-c/07-abo.eso" which failed since build #980. The table also provides links to "View job" and "Compile and Package". At the bottom of the page, there is a footer with the Atlassian logo and a message: "Level up your DevOps kung fu with Bamboo, the Continuous Delivery tool for JIRA teams. (Free open source license for Kiel University Comp. Sci.)".

KIELER Semantics - Continuous Plugins (Kepler 4.3) 985: Test results - KIELER Bamboo - Mozilla Firefox

File Edit View History Bookmarks Tools Help

KIELER Semantics - Continuous Plugins (Kepl... +

Back Forward Reload Home Zoom Out Zoom In Print Downloads Stop

rtsys.informatik.uni-koel.de/bamboo/browse/KISEMA-PLUGINS-985/test

Bamboo Build - Deploy - Reports - Log in Sign up

Build projects / KIELER Semantics / Continuous Plugins (Kepler 4.3)

Build #985 master -

Continuously builds all semantics plug-ins upon repository changes.

#985 failed - Changes by Christian Motika

Build summary Tests Commits Artifacts Logs Metadata

Test results

1 test in total 1 test failed 5 seconds taken in total.

Existing test failures 1

Test	Failing since	View job	Duration
SCChartsSimSAutomatedJUnitTest /test-sccharts-c/07-abo.eso	#980 (Manual run by Christian Motika)	Compile and Package	5 secs

Continuous integration powered by Atlassian Bamboo version 5.4.2 build 4208 - 03 Mar 14

Report a problem Request a feature Contact Atlassian Contact Administrators

Atlassian

Level up your DevOps kung fu with Bamboo, the Continuous Delivery tool for JIRA teams. (Free open source license for Kiel University Comp. Sci.)

Broken Build 😊

The screenshot shows a web browser window displaying a Bamboo build failure page. The browser's address bar shows the URL: `rtays.informatik.uni-koel.de/bamboo/browse/KISEMA-PLUGINS-JOB1-985/test/case/27273536`. The page title is "KISEMA-PLUGINS-JOB1-985 /test-schcharts-c/07-abo.eso: Test case result - KIELER Bamboo - Mozilla Firefox".

The Bamboo interface shows the build details for "Build #985" in the "master" branch. The build status is "Failed" with a message: "Job. Compile and Package failed". The specific failure is under the "Compile and Package" stage, titled "/test-schcharts-c/07-abo.eso: Test case result".

The failure summary states: "The below summarizes the result of the test '/test-schcharts-c/07-abo.eso' in build 985 of KIELER Semantics - Continuous Plugins (Kepler 4.3) - Compile and Package." The description is "/test-schcharts-c/07-abo.eso", the duration is "5 secs", the test class is "de.cau.cs.kielcr.schcharts.sim.c.test.SCChartsSimSaut", and the status is "Failed (Existing Failure)".

The error log shows the following message, which is highlighted with a red box in the image: **Error (0) in tick 5 of trace 0 of ESO file '/test-schcharts-c/07-abo.eso'**. Below this, the full stack trace is visible, starting with `java.lang.AssertionError: Error (0) in tick 5 of trace 0 of ESO file '/test-schcharts-c/07-abo.eso' during execution 'schcharts_validation_headless...`

At the bottom of the page, there is a yellow banner with the text: "Level up your DevOps kung fu with Bamboo, the Continuous Delivery tool for JIRA teams. (Free open source license for Kiel University Comp. Sci.)"

Overview

- ▶ KIELER Compiler (KiCo)
- ▶ Online Compiler
- ▶ Compiling & Simulating SCCharts
 - ▶ SCCharts Transformations
 - ▶ Simulation
 - ▶ Validation
- ▶ Summary & Future Work

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM

- ▶ KiCo: Separate Production and Handling dependencies (orders)

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM

- ▶ KiCo: Separate Production and Handling dependencies (orders)
- ▶ SCCharts
 - ▶ Simulation Visualization (Scheduling Difficulties)

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM

- ▶ KiCo: Separate Production and Handling dependencies (orders)
- ▶ SCCharts
 - ▶ Simulation Visualization (Scheduling Difficulties)
 - ▶ Fix/Rewrite Validation Component

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM

- ▶ KiCo: Separate Production and Handling dependencies (orders)
- ▶ SCCharts
 - ▶ Simulation Visualization (Scheduling Difficulties)
 - ▶ Fix/Rewrite Validation Component
- ▶ Explore different compilation variants / optimization phases

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM

- ▶ KiCo: Separate Production and Handling dependencies (orders)
- ▶ SCCharts
 - ▶ Simulation Visualization (Scheduling Difficulties)
 - ▶ Fix/Rewrite Validation Component
- ▶ Explore different compilation variants / optimization phases
- ▶ Evaluate SCCharts surveys (TR) with Steven

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM




- ▶ KiCo: Separate Production and Handling dependencies (orders)
- ▶ SCCharts
 - ▶ Simulation Visualization (Scheduling Difficulties)
 - ▶ Fix/Rewrite Validation Component
- ▶ Explore different compilation variants / optimization phases
- ▶ Evaluate SCCharts surveys (TR) with Steven
- ▶ Integrate KTM in KiCo (Alex)

Summary & Future Work

- ▶ KIELER Compiler
- ▶ Online Compiler
- ▶ SCCharts Simulation & Validation with KiCo & KIEM

- ▶ KiCo: Separate Production and Handling dependencies (orders)
- ▶ SCCharts
 - ▶ Simulation Visualization (Scheduling Difficulties)
 - ▶ Fix/Rewrite Validation Component
- ▶ Explore different compilation variants / optimization phases
- ▶ Evaluate SCCharts surveys (TR) with Steven
- ▶ Integrate KTM in KiCo (Alex)
- ▶ Hunt Bugs :-)

To Go Further

-  <http://www.sccharts.com>
-  C. Motika, S. Smyth, and R. von Hanxleden. *Compiling SCCharts – A Case-Study on Interactive Model-Based Compilation*. 6th International Symposium On Leveraging Applications of Formal Methods, Verification (ISOLA'14), Corfu, Oct 2014.
-  R. von Hanxleden, B. Duderstadt, C. Motika, S. Smyth, M. Mendler, J. Aguado, S. Mercer, and O. O'Brien. *SCCharts: Sequentially Constructive Statecharts for Safety-Critical Applications*. Proc. ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI'14), Edinburgh, Jun 2014.

That's all folks!

Any questions or suggestions?