Model-based Development with Giotto@Simulink

Wolfgang Pree
University of Salzburg, Austria
www.SoftwareResearch.net

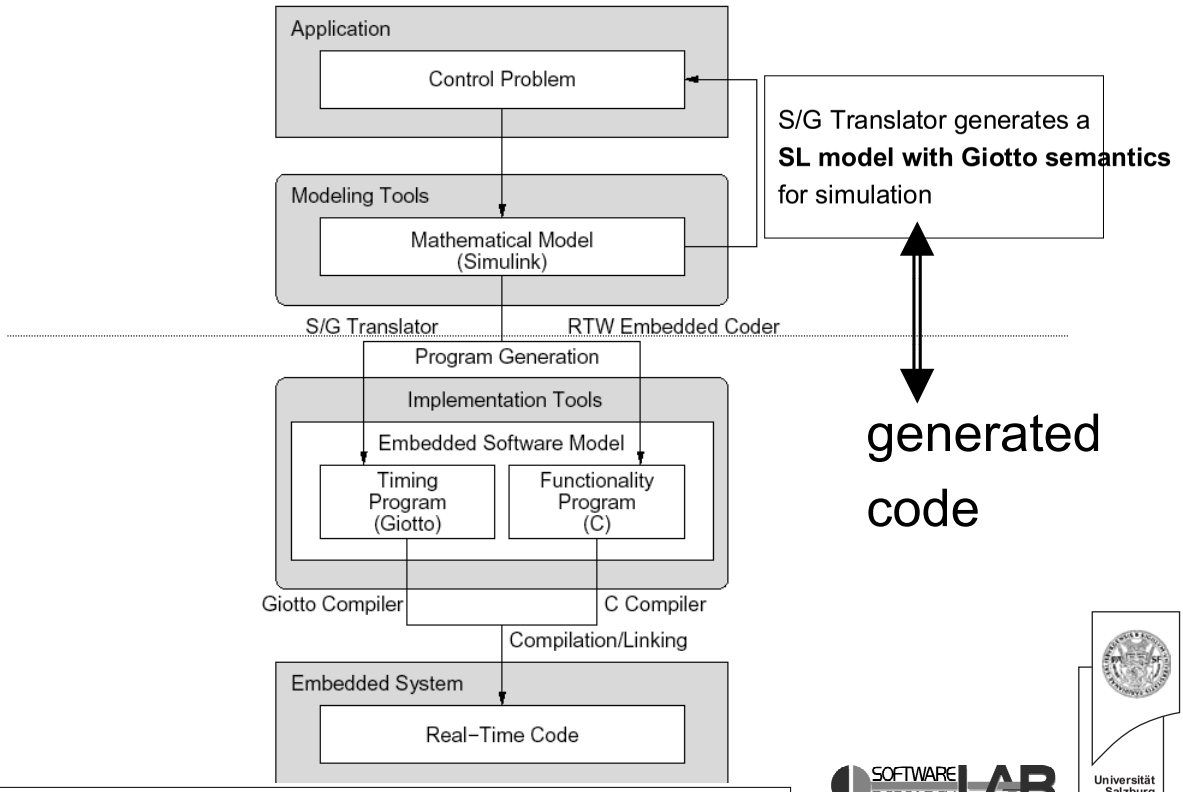
A joint project of
W. Pree, G. Stieglbauer and C. Kirsch

Contents

Giotto@Simulink tool chain

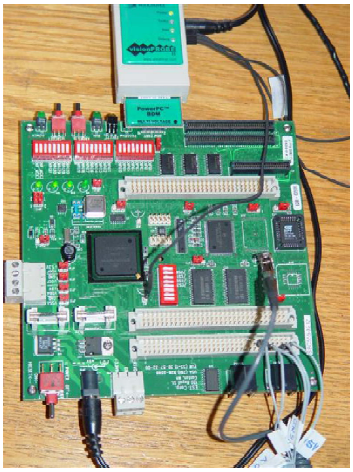
- S/G Translator:
  model transformation,
  Giotto code generation
- illustrated by the development of a throttle control system

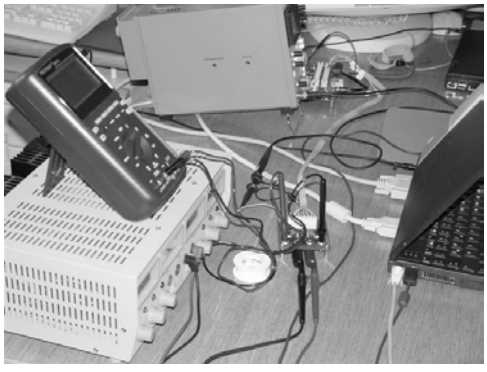
Giotto-based development process



(from the Giotto paper published in the IEEE Control Systems Magazine, Feb. 2003)

Case study: code generation from a Giotto@Simulink model of a throttle control system





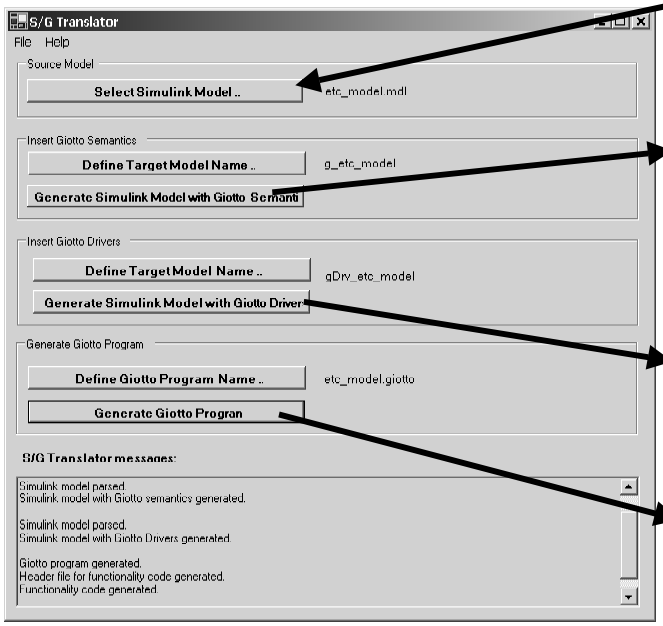
S/G Translator

- model transformation for simulation
- model transformation for functionality code generation
- generation of Giotto program

SOFTWARE RESEARCH LAB

Universität Salzburg

S/G Tranlsator tool



SL model

SL model with Giotto semantics

SL model with drivers for integration with E-machine

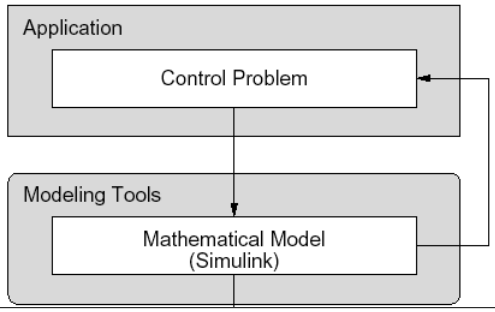
Giotto program

SOFTWARE RESEARCH LAB

Universität Salzburg

Step 1: S/G model for simulation

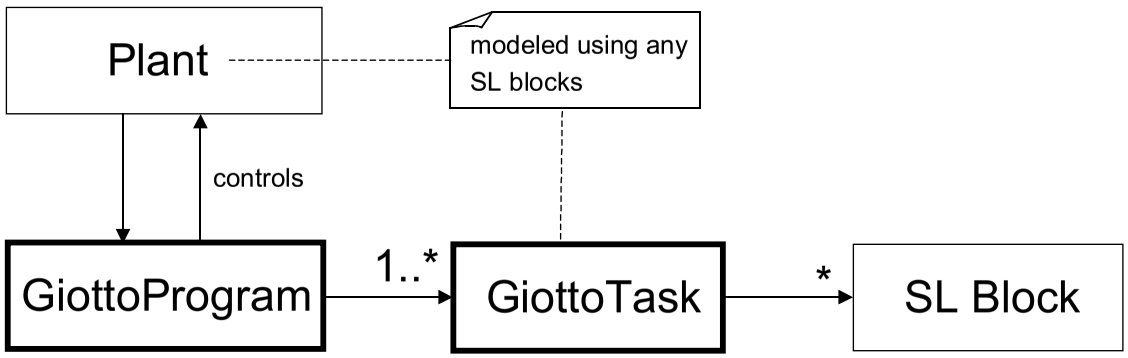
Step 1

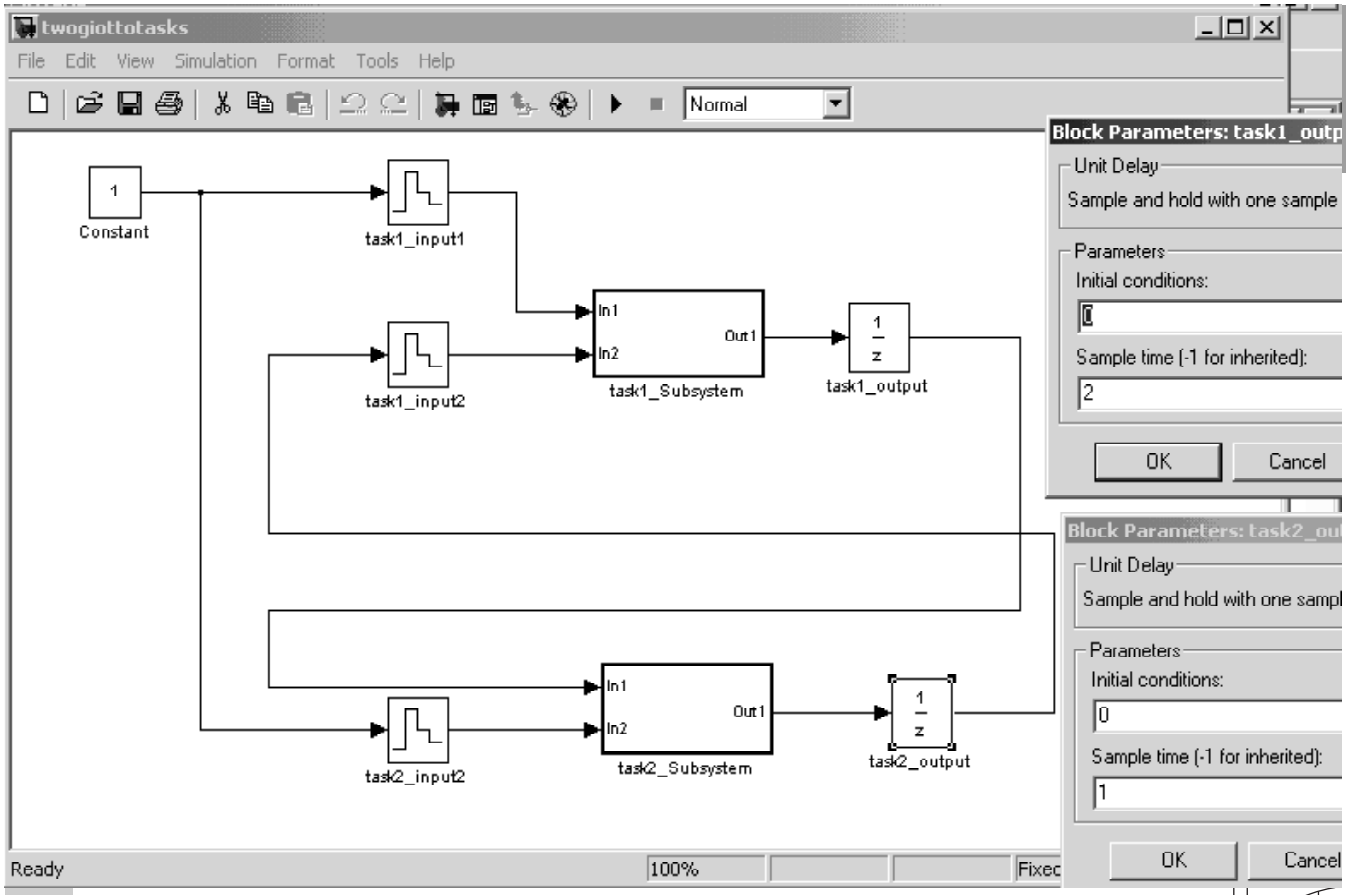


S/G Translator generates a **SL model with Giotto semantics** for simulation

(from the Giotto paper published in the IEEE Control Systems Magazine, Feb. 2003)

required input for the S/G translator

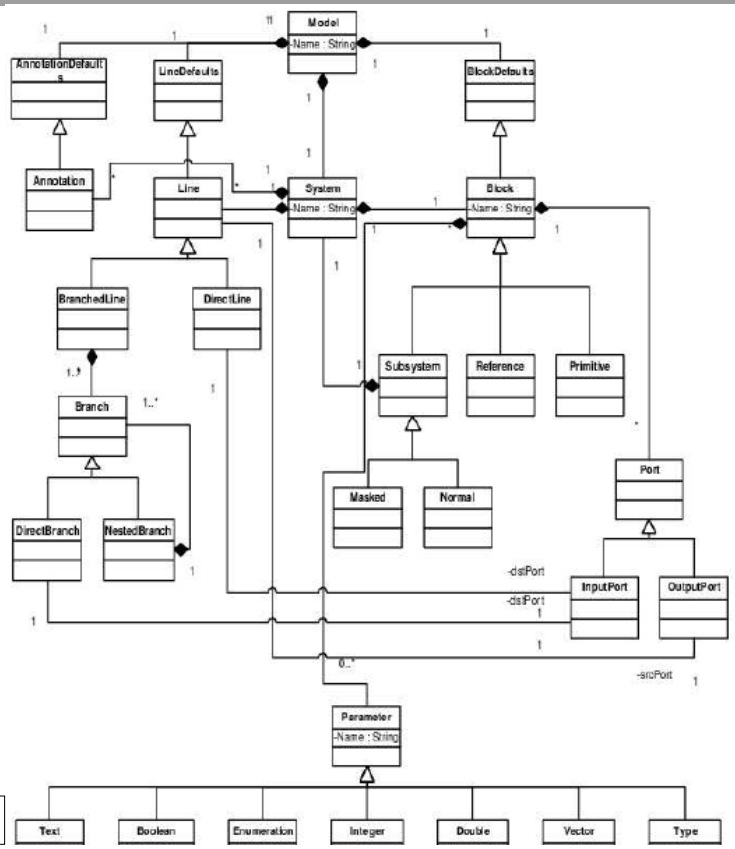




SOFTWARE RESEARCH LAB

Universität Salzburg

S/G translator is fully compliant with the current SL syntax

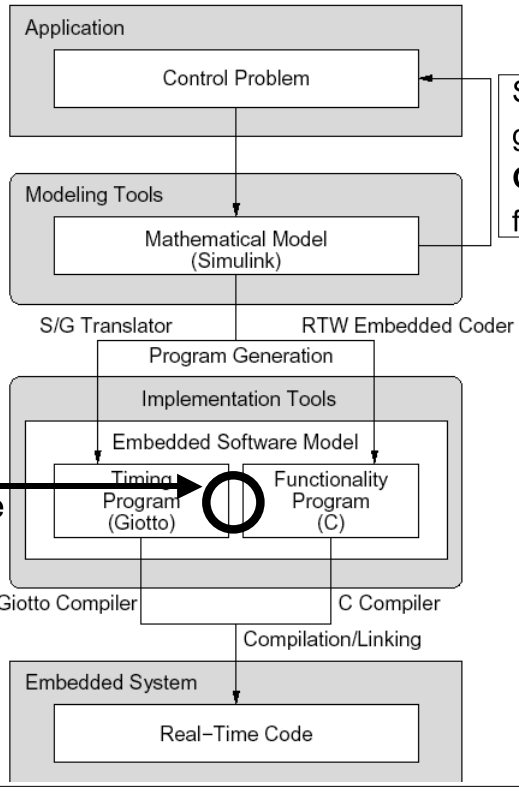


SOFTWARE RESEARCH LAB

Universität Salzburg

Step 2a: S/G model for the generation of functionality code that seamlessly integrates with the E-machine

Step 1 √



S/G Translator: generates a **SL model with Giotto semantics** for simulation

Step 2a:
SL model for generating glue code

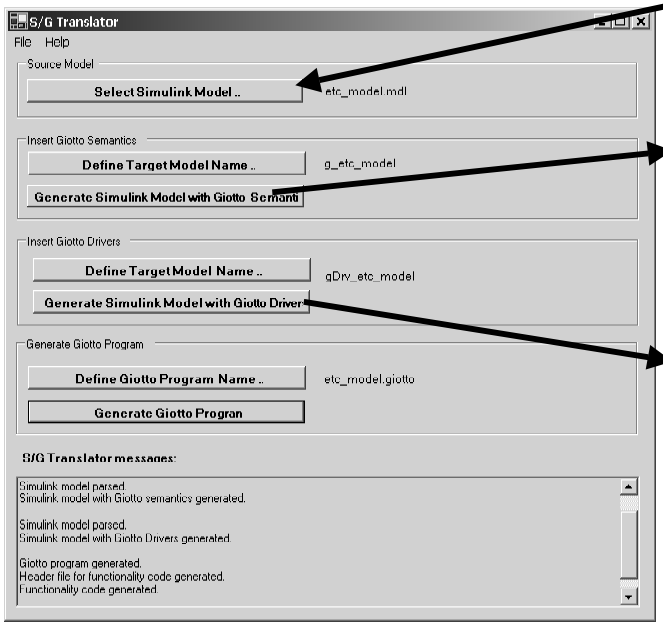
(from the Giotto paper published in the IEEE Control Systems Magazine, Feb. 2003)

SOFTWARE RESEARCH LAB

Universität Salzburg

S/G Tranlsator tool

SL model √



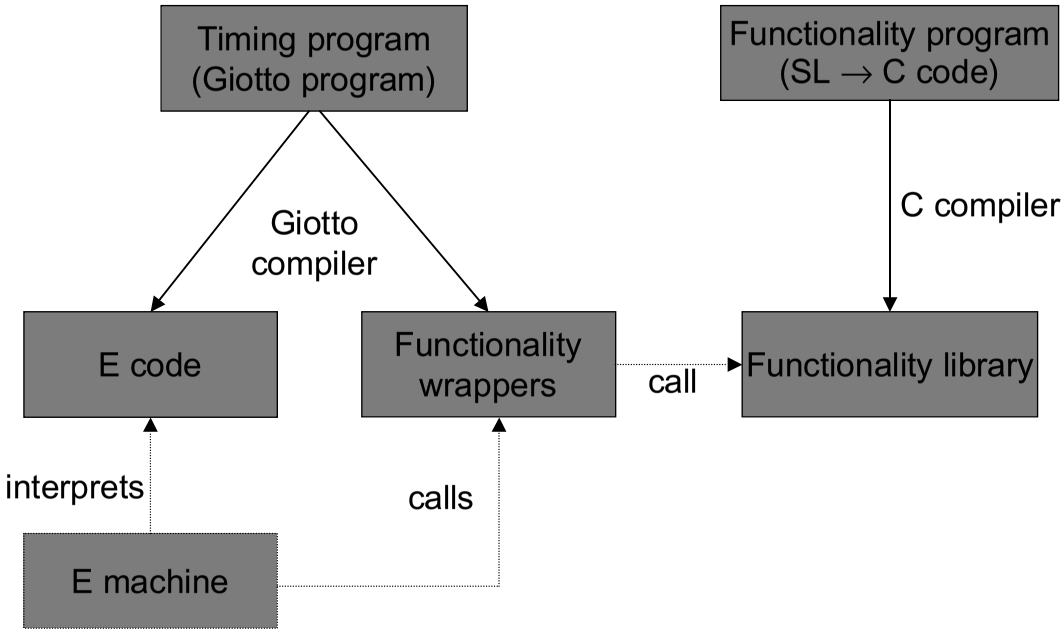
SL model with Giotto semantics √

SL model with drivers for integration with E-machine

SOFTWARE RESEARCH LAB

Universität Salzburg

preparation for linking timing code and functionality code (I)



SOFTWARE RESEARCH LAB

Universität Salzburg

preparation for linking timing code and functionality code (II)

- Giotto program segment

```
task GiottoTask1( … )output ( … ) state ( …) {
    schedule GiottoTask1();
}
```

- Functionality wrapper

```
void task_GiottoTask1() {
    GiottoTask1();
}
```

- Functionality code

```
void GiottoTask1(void) {
    local_GiottoTask1_output_1=GiottoTask1_input1+GiottoTask_input_2;
}
```

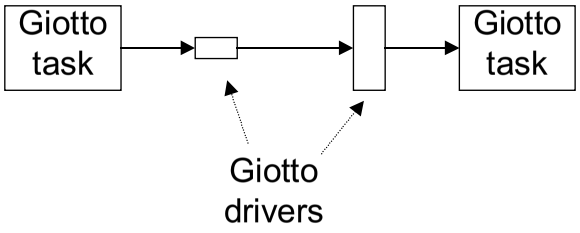

SOFTWARE RESEARCH LAB

Universität Salzburg

preparation for linking timing code and functionality code (III)

transport and convert values between task ports:

- via global variables (Simulink/RTW)
- via the Giotto driver concept



Giotto drivers are called by the E-machine

SOFTWARE RESEARCH LAB

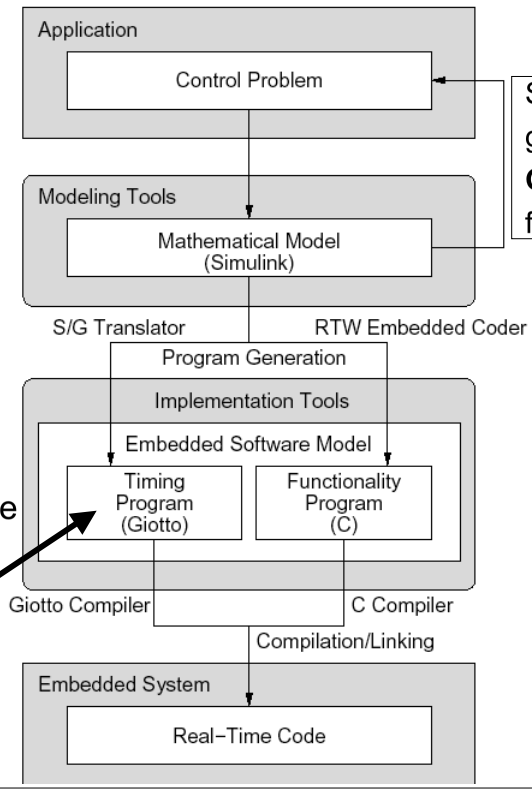
Universität Salzburg

Step 2b: generation of the Giotto program

Step 1 √

Step 2a: √
SL model for generating glue code

Step 2b:
Giotto program



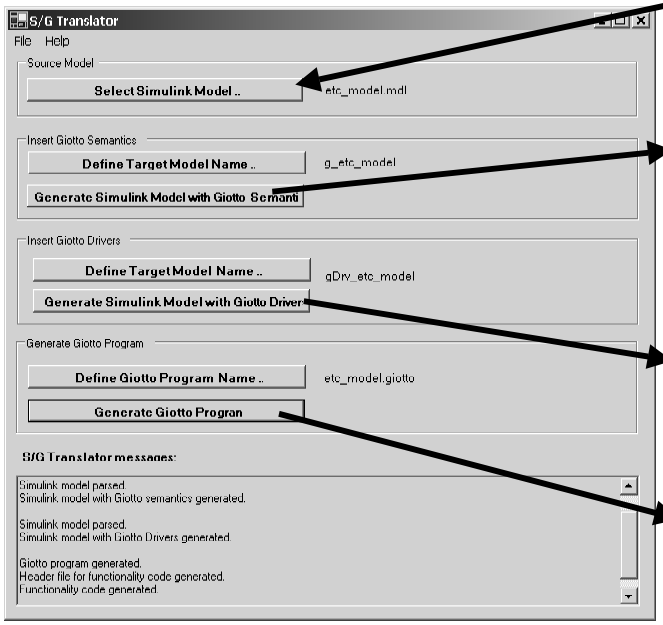
S/G Translator: generates a **SL model with Giotto semantics** for simulation

(from the Giotto paper published in the IEEE Control Systems Magazine, Feb. 2003)

SOFTWARE RESEARCH LAB

Universität Salzburg

S/G Tranlsator tool



SL model ✓

SL model with Giotto semantics ✓

SL model with drivers ✓
for integration with E-machine

Giotto program

SOFTWARE RESEARCH LAB

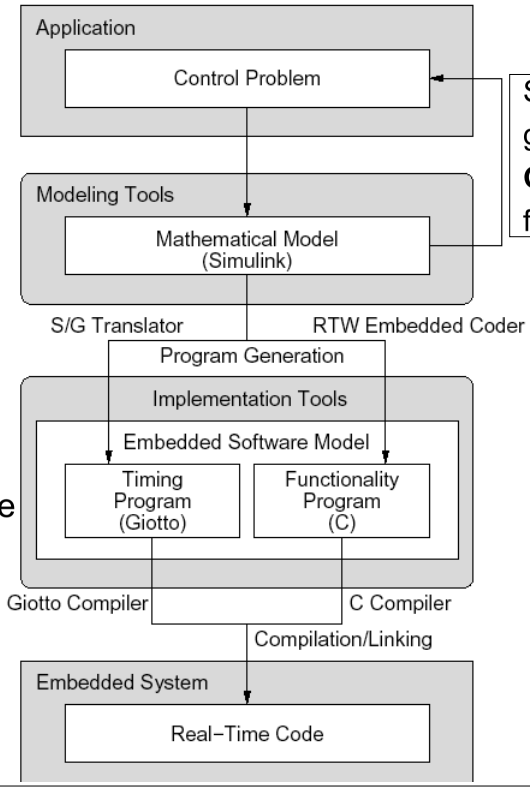
Universität Salzburg

Step 2c: generation of the functionality program with the RTW Embedded Coder

Step 1 ✓

Step 2a: ✓
SL model for generating glue code

Step 2b: ✓
Giotto program



S/G Translator: generates a **SL model with Giotto semantics** for simulation

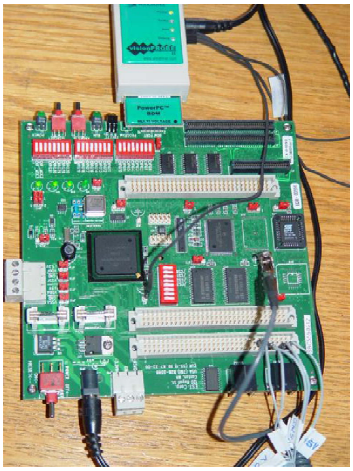
Step 2c:
C program

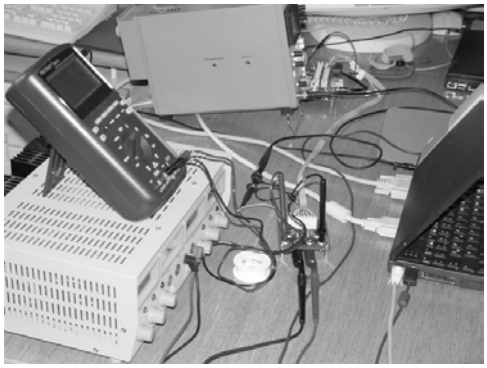
(from the Giotto paper published in the IEEE Control Systems Magazine, Feb. 2003)

SOFTWARE RESEARCH LAB

Universität Salzburg

throttle control system @ work





SOFTWARE RESEARCH LAB

Universität Salzburg

how long it took to ...

- upgrade the S/G Translator: **4 p. months**
  - a redesign that streamlines the architecture and makes the tool fully compliant with SL syntax: **2.5 m**
  - generation of SL model for glue code generation: **1m**
  - reimplementation of the C# version in Java: **0.5 m**
- implement the ETC case study: **0.7 p. months**

SOFTWARE RESEARCH LAB

Universität Salzburg

Future plans

SOFTWARE RESEARCH LAB

Universität Salzburg

Next steps

short term:

- illustrate composition and time safety checks in the realm of the ETC case study
- integration of Giotto modes into Simulink

mid-term:

- S/G-based prototype implementations of more complex control system components
- concepts for control system product families

SOFTWARE RESEARCH LAB

Universität Salzburg

The end

Thank you for your attention!