Automating Design and Verification of Embedded Systems Using Metamodeling and Code Generation Techniques

Well known Metamodels in EDA and Design: UML/SysML

Wolfgang Ecker, Infineon; Rainer Findenig, Intel
The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.

en.wikipedia.org
Unified Modeling Language

The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.

en.wikipedia.org
The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.
The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.
Unified Modeling Language

- Structural modeling:
  - Class diagram
  - Component diagram
  - Deployment diagram
  - ...

![Class Diagram Example](image-url)
Unified Modeling Language

■ Behavioral modeling:
  □ Activity diagram
  □ Sequence diagram
  □ State diagram
  □ ...

- State 1
  - entry / ...

- State 2

- State 3
  - exit / ...

- Activity 1
- Activity 2
- Activity 3

- Entry point
An Example: UML State Diagrams
UML?

- **Graphical Language**
  - Easy to read
  - Easy to write?

- **Semantics**
  - Not formally defined; software oriented
  - Given to your model as part of the code generation
    - Tool support is critical!

```
  Standard -> Semantics
  |
  v
  Model   Generator -> Code
```
UML: The Spirit of Metamodelling

M3

- UML Infrastructure

(meta-meta model)

M2

- UML Superstructure

M1

- User Diagrams

M0

Number
0 200 400 600 800 1000 1200

Test Chip Card Wireless
Extending UML: Profiles

- Extension mechanism for customizing UML
- Light-weight, easy
- Strictly additive, no fundamental changes

Metaclasses (M2)

User-specified extensions

Stereotype: “new” Transition with additional properties
Extending UML: MOF – Meta-Object Facility

- UML itself is defined in the MOF
- Allows defining completely new Metamodels
SysML

- Extended subset of UML
- Defined using profiles
SysML: Block Definition Diagram
SysML: Internal Block Diagram
UML State Diagrams for Different Abstraction Levels

UML Profile

- Event-driven transitions:
  - Derived from time, transactions, or other internal/external events

- Clock-driven transitions:
  - Derived from an internal clock
  - Can use guards for specifying timeouts
UML State Diagrams for Different Abstraction Levels

- UML Profile
  - Initial states to conform with hardware reset semantics
  ![](image)
  - Global and local variables
  ![](image)
UML State Diagrams for Different Abstraction Levels

- **UML Profile**
  - Link to external interface definition
    - Including selection of desired abstraction level
  - Refinement between states and transitions
Example: SIF

```
<initialState>
  idle
entry
  busy = 0;
exit
  busy = 1;
data = data_in;

if (mode == 1 & & addressed(addr))

Sending
  do
    .send(START);
    .wait(BIT_TIME);
    .for (uint8_t i=0; i<8; ++i)
      { 
        .send(data & 1);
        .data >>= 1;
        .wait(BIT_TIME);
      }
    .send(STOP);
    .wait(BIT_TIME);
```

```
ClockedSending
entry
  bitcount = 0;

<initialState>
  StartBit
entry
  send(STOP);
  after(100, clk);
```

```
DataBit
entry
  send(data & 1);
  after(100, clk) [bitcount > 0]
  after(100, clk) [bitcount == 0]
```

```
StopBit
entry
  send(STOP);
  after(100, clk);
```
Thank you!