Making Frameworks using SDL

Birger Møller-Pedersen ERICSSON
Rolv Bræk SINTEF
A framework is:

The Free On-line Dictionary of Computing:

“In object-oriented systems, a set of classes that embodies an abstract design for solutions to a number of related problems.”

From a tutorial:

“Frameworks are reusable designs for an application or a subsystem expressed as a set of classes and the way that instances of these classes collaborate.”
Libraries and Frameworks

- Libraries
- Predefined classes

Frameworks
- Predefined classes
- Predefined application
- Predefined structure

In SDL: System types
with predefined instance structure
Two SDL models

Application specific SDL specification

Complete, implementation specific SDL description
The complete SDL description as a Framework

Benefits:
- Separation for development, maintenance and portability
- Combination for code generation and system handling
A Framework as a System Type

system type FrameWork

virtual AT1
virtual IT1

A1: AT1
I1: IT1

system type Application
inherits FrameWork

redefined AT1
redefined IT1
virtual AT2

A2: AT2

system MyAppl:Application
An Application Block

system type FrameWork

virtual AT1
virtual IT1

A1: AT1
I1: IT1

virtual block type AT1

virtual APT1
virtual IT2

App(5,):
APT1
inf: IT2

SINTEF Telecom and Informatics
Making Frameworks using SDL Foil no 71998-06-25
If an Application Process is common to many parts:

```
system type FrameWork2

virtual AT1  virtual IT1

virtual APT1

A1:AT1
Ap1:APT1

I1: IT1

Ap2:APT1
```
A Framework with no instances

system type FrameWork3
virtual AT3     virtual IT3

system type Application2
inherits FrameWork3
redefined AT3   redefined IT3
A3:AT3          I3:IT3

system SpecialAppl:Application2
A virtual block type without instances

- application known through context parameters
- instances only in the last step

virtual block type AT3

process type infPT

...create cp(...)

redefined block type AT3

process type SpecialApp
inherits applicationPT

process type SpecialInf
inherits infPT<actualapp>

actualapp(0,): SpecialApp
actualInf: specialInf
A Framework with virtual creation procedures

system type FrameWork4

virtual IT4

Infrastructure:
IT4
OP

virtual block type IT4

virtual IT5

Inf:
IT5

• application known through virtual procedures
• instances only in the last step
SetUp as a virtual procedure

virtual block type IT4

virtual IT5

Inf: IT5

virtual process type IT5

virtual procedure SetUp

/* This is redefined to reflect the set up of the actual system*/

/* procedure SetUp is called upon the reception of the signal setUp from the operator*/
Application part created by virtual procedure

system type Actual inherits Framework4

redefined block type IT4

redefined process type IT5

redefined procedure SetUp

process type App1 inherits Application

process type App2 inherits Application

a1(0,):App1

a2(0,):App2

REDEFINED PROCEDURE setUp

AP1?

yes

a1

no

AP2?

yes

a2

no
Using Framework in AXE-10

**Application design**

- **block type** SubServices
  - call(0,): callhandler
  - sub(0,): subhandler

**Framework design**

- **block type** SS inherits AXE10Block
  - redefined blockproc
  - bp: blockproc
  - call(0,): callhandler
  - sub(0,): subhandler

**Implementation**

- PLEX, C

Focus on service behaviour
Simulation, validation of services

Focus on platform and implementation issues
Simulation of start/restart, size alteration, ...
Code generation

---

Making Frameworks using SDL Foil no 141998-06-25

SINTEF Telecom and Informatics
Framework oriented development

• Support SDL development
  • on a platform of pre-defined set of types that capture the implementation specific properties,
  • by specialising these types to the needs of the actual applications.
• Support the separation of application- and implementation specific parts of applications
• Support the definition of architectures of combined application- and implementation specific parts defined in such a way that the application specific parts can be modified without modifying the architecture.