Experiences with the use of a configuration language

Jacqueline Floch  SINTEF
The PROTEUS project

- ESPRIT project 1992 - 1995

- Objective: to provide methodological and tool support for system evolution

- Main result: Proteus Configuration Language and associated PCL tools

- In Norway two companies Garex and Stentofon participated as “application” providers.
Similar products, similar contexts

• *customized* voice communication control systems vs. *customized* internal intercom systems

• Design and supply both HW and SW

• Few employees; numerous installations world-wide!
Different focus in the development process

- Stentofon uses an SDL based methodology. Focus is set on SDL design. C code is produced automatically.

- Garex has developed library of highly configurable components. Focus is set on reusability.
There are so many components...

...and all of them evolve...

Experiences with the use of a configuration language - 6th IWSCM
Requirements

- Visibility of the overall system structure
- Support for system building
  
  (Stentofon)
- Support for composition and configuration of SDL components
  
  (Garex)
- Support for system configuration on TST
PROTEUS Configuration Language

- PCL is a language for modelling systems and system building
- The PCL model is based on the family notion

Experiences with the use of a configuration language - 6th IWSCM
Experiences with the use of a configuration language - 6th IWSCM

PCL main concepts

Family entity
- attributes
  - classification
  - interface
  - parts
- physical
  - relationships

Part-of

Defined by

Attribute type def.

Classified

Version descriptor
- attributes
  - parts

Inherits from

Tool entity
- attributes
  - inputs
  - outputs
  - scripts

Selects

Classified

Relation def.
Experiences with the use of a configuration language - 6th IWSCM

PROTEUS Tools

PCL editor browser

PCL family model

PCL Compiler

Bind

Select

MakeGen

Repository browser

Repository

Unix file hierarchy

PCL Reverse

PCL version description

bound model

selected model

makefile(s)

checkin checkout
Achievements

- PCL model provides overall system visibility
- SDL system structure variability can be described (Stentofon)
- Makefiles for building particular system instance can be generated automatically
- TST configuration files can be generated (Garex)
Evaluation results (+)

- Tasks can be done at a higher abstraction level
- All system knowledge is represented in one place
- Using PCL reverse, it is easy to produce an initial system model
- The models for system building are concise
- The repository is built on top of an existing version management system (RCS)
- Repository browser is a user-friendly tool which provides advanced functions for version management
Evaluation results (-)

• SDL and PCL are loosely integrated

• The models for TST configuration generation are complex: limited support for instance description.

• The implementation of interactive binding is not satisfactory: no mechanism for updating version descriptors.

• The PCL toolset is not a fully commercialised product
Conclusions

• Support is needed in order to deal with the large number of source code elements and the complex building process

• PCL is a promising approach for supporting system evolution

• We need more experience in order to:
  - learn about maintenance of PCL models
  - develop guidelines for using PCL