Agenda

I. The ConIPF Project

II. CPS: The Product Family used for the Demonstration

III. Configuration Models Used in The Demonstration

IV. Demonstration Platform Elements

V. Demonstration
I. The ConIPF Project

Mission:

The ConIPF Project has defined a METHODOLOGY for

FEATURE BASED PRODUCT DERIVATION

where:

- The features specify the Capabilities of the product

and

- The corresponding product artefacts are determined through Structure-Based Configuration

John MacGregor, 31.08.2004
Industrial Complexity

Staggering Complexity:

- **Variability**
  - Thousands of Products
  - (Tens of) Thousands of Components
  - (Tens/Hundreds of ) Parameters per Component

- **Combinability**
  - Cardinality (Mandatory, Optional, Alternative, Limited Number)
  - Interdependence (Excludes, Requires,…)

John MacGregor, 31.08.2004
I. The ConIPF Project

Organisational Situation

- Too much knowledge in heads
- Too little (people with) overview
- Insufficient knowledge Exchange
- No overview of ‘who-knows-what’
- No overview of required vs. available knowledge
- Dependency on experts
- Slow, error-prone configuration process
- Long training times
- Vulnerability to Knowledge gaps
I. The ConIPF Project

The ConIPF Approach (1/2)

- The systems engineer maps product requirements, which are formulated from the customer perspective, to features in the domain feature tree.

- That is, the engineer traverses the domain feature tree, selecting generic features and assigning them values.

- This ultimately results in the feature configuration for a particular product.
I. The ConIPF Project

The ConIPF Approach (2/2)

- An inference machine supports the configuration process to ensure consistency, correctness and completeness.

- Feature configurations do not usually map directly to the artefact configurations that realise the features.

- Intermediate knowledge bases are therefore be needed to transform the features into the solution.
I. The ConIPF Project

Configuration

Meaning of the word:

- An **Action** – the process of combining or arranging
- As well as a **Result** – a list of necessary components
- Relationship to Configuration Management
  - Combination – "make" or IDE project description
  - Lists – describe a deliverable that is stored and managed in a configuration management system
I. The ConIPF Project

Configuration Knowledge

Configuring (as AI discipline):

Assembly of a technical system from individual parameterisable objects to a configuration that fulfils a certain task (or purpose)

Based on:
- Descriptions of objects and their attributes (concept hierarchy)
- Relations/restrictions between the objects
- Knowledge over the solution procedure (control knowledge)
- A description of the goal to be fulfilled

* A. Günter, University of Hamburg
I. The ConIPF Project

Paradox: Feature Tree or Structure Description?

Legend:
- Mandatory
- Optional
- Alternative
- Requires
- Excludes

Automobile
- Air Conditioning
  - Climate Control
  - Manual
- Transmission
  - Automatic
  - Manual

Motor
- 4 Cylinder
- 6 Cylinder
- 8 Cylinder
I. The ConIPF Project

The Product Configurer’s Problem

- Both the target environment and the desired features constrain the choice of components.

- The combined capabilities of the components must meet or exceed the capabilities specified by the features.

- The components that are selected dictate the capabilities of the system.

---

John MacGregor, 31.08.2004
I. The ConIPF Project

Two Levels of Configuration

- **Problem Level: Selection of Capabilities**
  - The user configures the product capabilities and the configuration system determines which components are needed
  - The level preferred by ConIPF

- **Solution Level: Selection of Components**
  - The user is responsible for ensuring that the capabilities of the selected components meet the product requirements
  - Traditional configuration

- ConIPF supports both levels
  - (Features can be defined either on the problem or solution levels)
I. The ConIPF Project

Features

→ Software Engineering Viewpoint:
   are Aggregates of Requirements

→ ConIPF (structure-based configuration) Viewpoint
   are Concepts with Attributes (not binary)
I. The ConIPF Project

Features Types

- **Capability Features (Problem Level)**
  - Describe product capabilities, their attributes and extents
  - Current Hypothesis:
    - Capability Features are the functionalities in the system
    - The attributes describe the feature’s non-functional aspects
    - Example:
      Feature: Air-conditioning
      Attributes: Time to cool car, Effect on gas consumption

- **Product Aspect Features (Solution Level)**
  - Relate to requirements (customer preferences, perhaps)
  - Example: The number and type of knobs to control the a/c

- Perhaps more types
I. The ConIPF Project

The ConIPF Common Application Model (CAM)
I. The ConIPF Project

The ConIPF Derivation Process

- PreSelect
- HW / SW Artefacts
- Features
- HW / SW Constraints
- Environment

Phases:
- Sales
- Engineering
- Calibration

Configuration Activity
Realisation Activity
I. The ConIPF Project

Process Elements

- Direct Derivation
  - The capabilities of available artefacts match those of the features chosen

- Calibration
  - Where the derived configuration is fine-tuned
    - Overengineered configurations are slimmed down
    - Underengineered configurations are beefed up

- Evolution
  - The capabilities available artefacts do not match those of the features chosen
    - New Development
    - Update the Asset Repository, Configuration Model
    - Repeat Direct Derivation
I. The ConIPF Project

Evolution Considerations

- Fundamentally, direct derivation deals with packages of variability (modules)
- Architectures address non-functional requirements through various arrangements of modules
- These modules are actually configurations of components
- New development means developing new components and configuring new packages or reconfiguring existing packages
II. CPS

Car Periphery Systems

- control unit
- sensor harness
- loudspeaker
- system harness
- attachment spring
- sensor module
- sensor housing

Parking Assistance System
II. CPS

The CPS Product Family

- Parking Assistance
- Pre-Crash Detection Front
- Semi-automatic Go
- ACC Stop & Go
- Parking Spot Detection
- Pre-Crash Detection Front & Side
- Blind Spot Detection
- Semi-automatic Parking
- Pre-Crash Detection Side & Rear
II. CPS

CPS Products Used in the Demo

Parking Support
- Monitors the distance to objects while the vehicle is parking
  - Displays the distance to the object, or
  - Sounds alarms when boundaries are crossed

PreCrash Applications: detect an imminent crash
- Preset: Sensitises the airbag sensor
- PreFire: Fires a (seat)belt tensioner
II. CPS

Configuration Front-End for PreCrash

Activation Zone

Trigger Zone
II. CPS

Configuration Front-End for Parking Assistance

Parking Zones:
Near,
Very Near,
Imminent,
In Proximity

John MacGregor, 31.08.2004
Configuration Front-End for Solution Space
Bosch’s Instantiation of the ConIPF Process

III. Configuration Models

John MacGregor, 31.08.2004
III. Configuration Models

CPS Context

Target Vehicle
- Front Bumper
  - Width
  - Height
- Rear Bumper
  - Width
  - Height

Environment
- To Vehicle Information
  - Speed
  - Current Transmission Gear
- Interfaces
  - To Airbag
  - To Belt Tensioner
  - To Audio System
- Regulatory
  - Europe
  - N. America
  - Rest of the World
- Physical
  - Temperature Range
  - Humidity Range

Arrows:
- ↑ Multiple (One or More)
- ↔ Alternative (Just 1)
- Optional
- Property
III. Configuration Models

CPS Artefacts

Software

Hardware

Multiple (One or More)
Alternative (Just 1)
Optional
Property

Audio

Buzzer
Loudspeaker
Radio Adapter

Display

TFT 1
TFT 2
Navil Adapter

Sensor Configurations

Control Unit

1xUS(a) Supervision Depth=1m
1xUS(b) Supervision Depth=2.5m
1x SRR.a Supervision Depth=3m
2xSRR.a Supervision Depth=10m
2xSRR Supervision Depth=20m

Common

Main

Sensor

1 Sensor

MMI

PreCrash

COI

CoI 1D
CoI 2D

Application

Parking

Presel

Prefitr

John MacGregor, 31.08.2004
III. Configuration Models
IV. Demonstrator Elements

Demo Target Platform

- Realistic - Compatible with „real“ applications

- **Hardware Platform:**
  - Control Unit running with reduced CPU frequency
  - PC with pre-recorded sensor signals as input
  - PC with Labview Simulation of MMI as output for Parking Support

- **Software Platform**
  - SRR Sensor Platform
  - Parking Assistance, PreSet, Prefire
IV. Demonstrator Elements

Target Control Unit

Ext. CAN
PC with Pre-recorded Sensor Data

Int. CAN
PC with MMI in Labview
IV. Demonstrator Elements

Pre-recorded Sensor Data
IV. Demonstrator Elements

MMI Simulation
IV. Demonstrator Elements

EngCon - Running Configuration

K-Build
Testumgebung für CPS Guiding Example

<table>
<thead>
<tr>
<th>Product 0 [Product]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

- **Relationen**

<table>
<thead>
<tr>
<th>Konzept</th>
<th>aktueller Wert</th>
</tr>
</thead>
<tbody>
<tr>
<td>has Features</td>
<td></td>
</tr>
<tr>
<td>Parking Assistance</td>
<td>0.0</td>
</tr>
<tr>
<td>Preset</td>
<td>1.1</td>
</tr>
<tr>
<td>PreFire</td>
<td>1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>has Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
</tr>
<tr>
<td>Software</td>
</tr>
</tbody>
</table>

Lösung

- **Product 0**
  - has Features:
  - PreFire 0
  - has Zones:
    - Belt Tensioner Activation Zone 11
    - Belt Tensioner Trigger Zone 10
    - Preset 8
      - has Zones:
        - Airbag Activation Zone 8
        - Airbag Trigger Zone 7
    - has Parts:
      - Hardware 1
      - Sensor Configuration 2
      - Software 3
        - has Applications:
          - PreFire Module 15
          - has Requires:
            - PreCrash Module 16
              - has Parts:
                - PreSet Module 12
                - has Requires:
                  - PreCrash Module 13
                    - has Parts:
                      - Common 4
                      - Main 5
                      - PreCrash Module 14
                        - has Parts:
Software Tools

- iSystem WinIdea:
  - Development IDE for the Control Unit

- Texas Instruments Compiler
  - C / C++

- Saxon
  - XSLT Processor

- XML Spy
  - General XML Editor

- LabView
  - Interface to MMI Simulation
  - Platform for Sensor Data Simulation
V. Demonstration

Demonstration Overview

Derive Application with EngCon ➔ Partial Configuration in XML ➔ Give Open Parameters Default Values with XSLT ➔ Full Configuration in XML ➔ Parse for Software Configuration with XSLT ➔ Compiler Script(s) ➔ Compile Application

Parse Parameter Values with XSLT ➔ EEPROM Parameter Value List ➔ Load EEPROM Parameters ➔ Run Demonstration

Parse for MMI Configuration with XSLT ➔ MMI Configuration File ➔ Start MMI
Demonstration Notes:

- Transition from Context Configuration to Feature Configuration is fluid

- Current solution for calibration:
  - Parameter values are derived explicitly or set during initial configuration process
  - The resulting (partial) configuration is stored as the base
  - Calibration continues by further setting or testing the values as needed
  - Backtracking accomplished by going back to the base configuration and reconfiguring

- Better solutions are being investigated.
Thank-you for your Attention!