Feature Modeling and Configuration Management

Roche Diagnostics, 16th October 2007

Dr. Ondrej Rohlik
(ETH Zurich and P&P Software GmbH)
rohlik@pnp-software.com

Agenda

- What is a Product Family?
- What is Feature Modelling?
- What is the XFeature Tool?
- XFeature Tool Demonstration
- Pointers to More Informations

Overview of Product Families

Product Families

Product Families Examples

- VW Corrado

Product Families Examples

Product Families Examples

- Computer Hardware Configuration

Wählen Sie Ihr MacBook Pro.

15" : 2,2GHz

2,2 GHz Intel Core 2 Duo
Auflösung von 1440 x 900 Pixeln
2 GB Arbeitsspeicher
120 GB Festplatte[1]
8x SuperDrive Laufwerk (Double Layer)
NVIDIA GeForce 8600M GT Grafikprozessor mit 128 MB SDRAM

Versandfertig in: 3 Tage
Kostenfreie Lieferung

CHF 2.507,08 *
(CHF 2.330,00 exkl. MwSt.)

Auswählen

15" : 2,4GHz

2,4 GHz Intel Core 2 Duo
Auflösung von 1440 x 900 Pixeln
2 GB Arbeitsspeicher
160 GB Festplatte[1]
8x SuperDrive Laufwerk (Double Layer)
NVIDIA GeForce 8600M GT Grafikprozessor mit 256 MB SDRAM

Versandfertig in: 3 Tage
Kostenfreie Lieferung

CHF 3.123,63 *
(CHF 2.903,00 exkl. MwSt.)

Auswählen

17" : 2,4GHz

2,4 GHz Intel Core 2 Duo
Auflösung von 1680 x 1050 Pixeln
2 GB Arbeitsspeicher
160 GB Festplatte[1]
8x double-layer SuperDrive
NVIDIA GeForce 8600M GT Grafikprozessor mit 256 MB SDRAM

Versandfertig in: 7 - 10 Tage
Kostenfreie Lieferung

CHF 3.475,48 *
(CHF 3.230,00 exkl. MwSt.)

Auswählen

15" | 17"

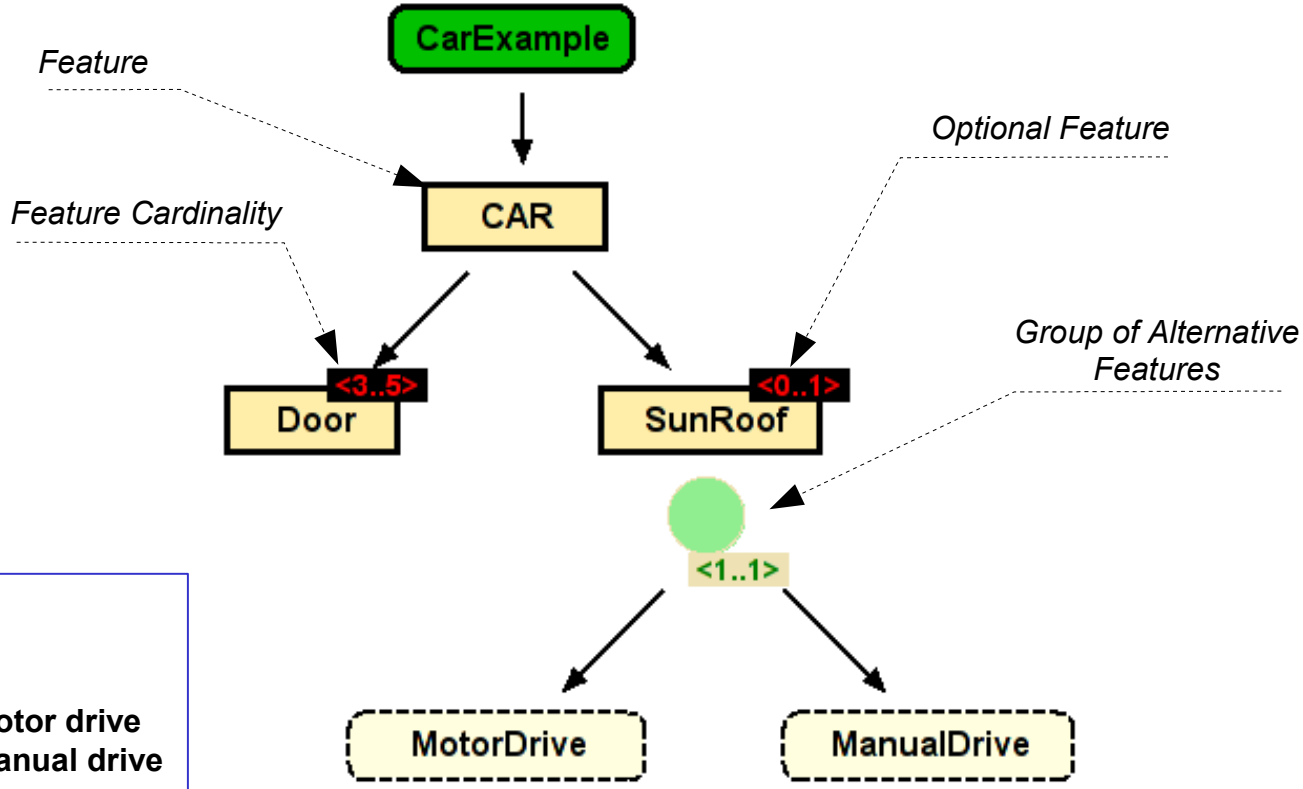
Product Families Examples

SQL Server 2005 Features Comparison

Published: November 7, 2005 | Updated: May 22, 2006

Scalability and Performance					
Feature	**Express**	**Workgroup**	**Standard**	**Enterprise**	**Comments**
Number of CPUs	1	2	4	No Limit	Includes support for multicore processors.
RAM	1 gigabyte (GB)	3 GB	Operating system maximum	Operating system maximum	Memory limited to maximum supported by operating system.
64-bit Support	Windows on Windows (WOW)	WOW	✓	✓	
Database Size	4 GB	No Limit	No Limit	No Limit	
Partitioning				✓	Support for large-scale databases

A Family of Car Types

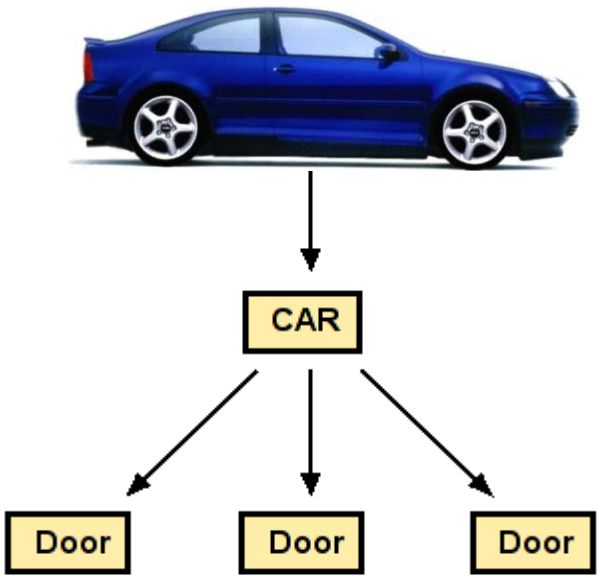


Instances of this family include:

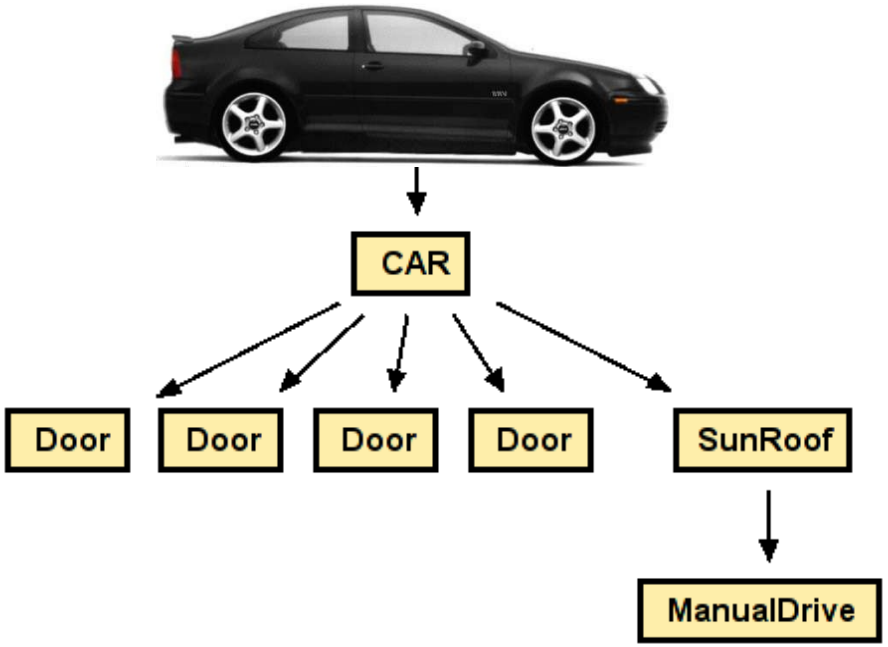
- **A car with 3 doors and no sunroof**
- **A car with 5 doors and a sunroof with motor drive**
- **A car with 4 doors and a sunroof with manual drive**
- **. . .**

Different authors have proposed different graphical notations for representing feature diagrams.

Family Instances



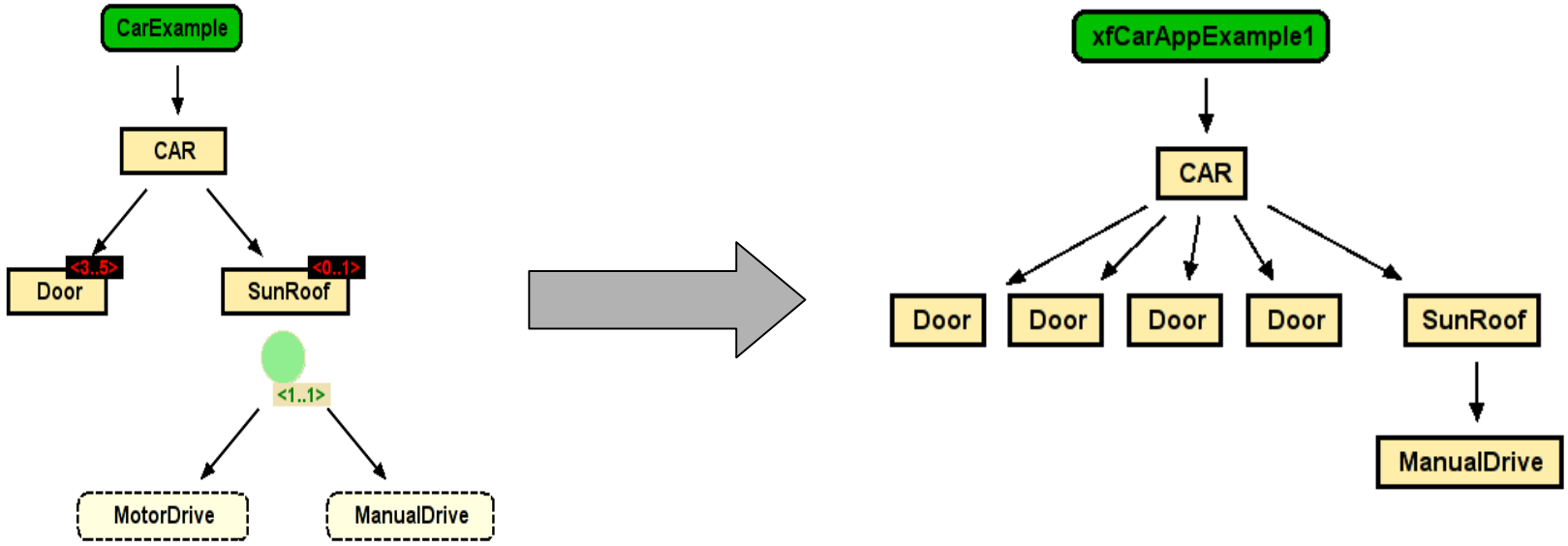
A family instance characterized by three doors and no sun-roof



A family instance characterized by four doors, and sun-roof with manual drive.

Instantiation Process

ETH Zurich



Family specification through a feature model (family domain analysis)

Product specification as an instance of the family feature model (application requirement definition phase)

A good family modelling tool should:
(1) support the definition of family models
(2) support the definition of combination constraints on the features
(3) support the instantiation of the family models to create instance/application models
(4) provide tips to help the user select the instance/application features

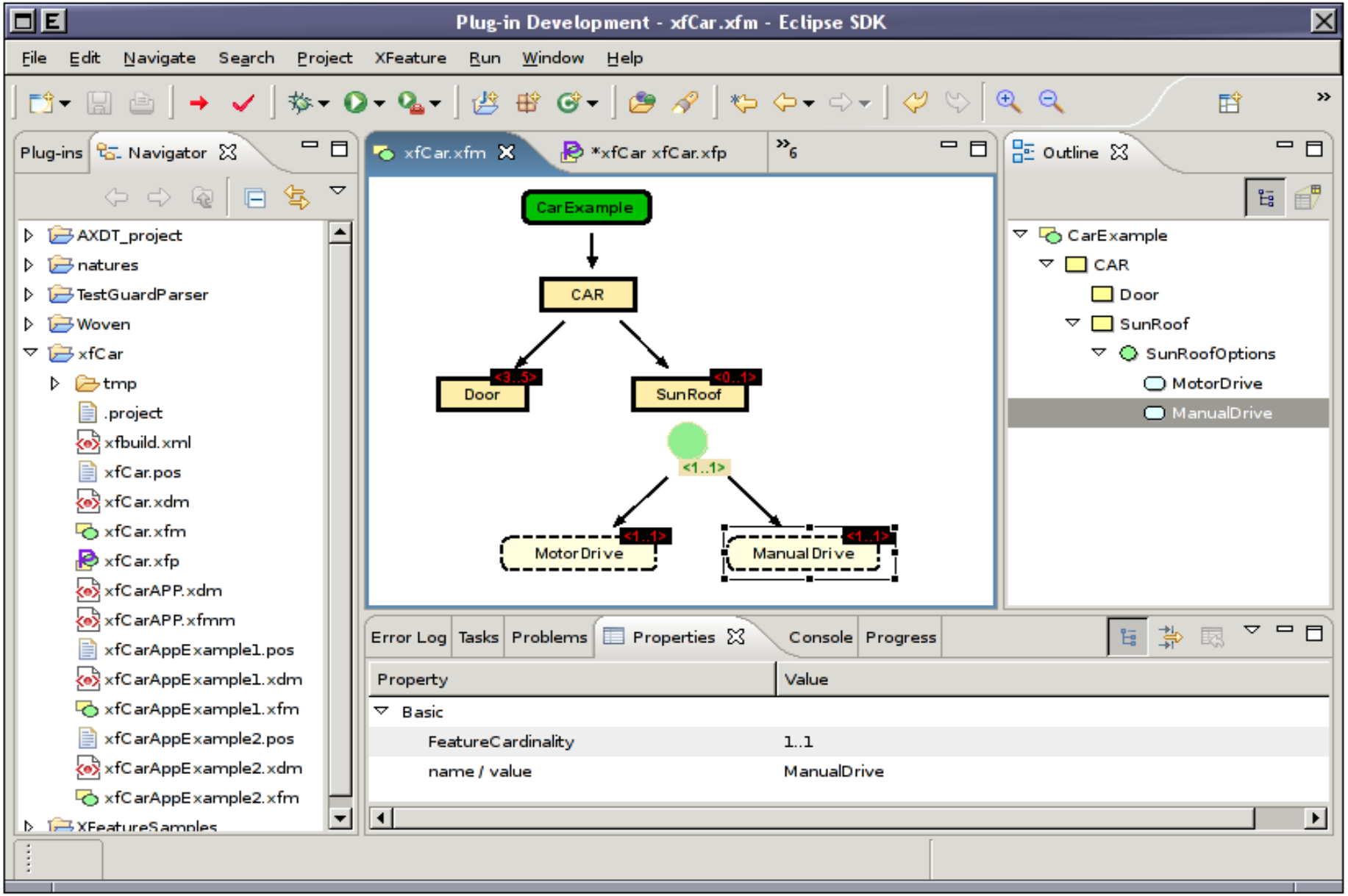
Feature Modelling

- **Feature**: a characteristic of a system that is of relevance to its end-user
- **Feature Model**: a model of all the features of product that can be potentially instantiated from a family together with the constraints on their combinations
  - There may be compatibility constraints between features (i.e. feature A is incompatible with feature B)
- Feature models are expressed using tree-like structures
  - Each node in the tree represents a feature
  - A feature may be broken up into subfeatures represented as children nodes
  - Various conventions are used to describe constraints on the feature combinations
- Feature modelling is used whenever there is a need to model variability within a set of related products

The XFeature Tool

http://www.pnp-software.com/XFeature/

XFeature Screenshot

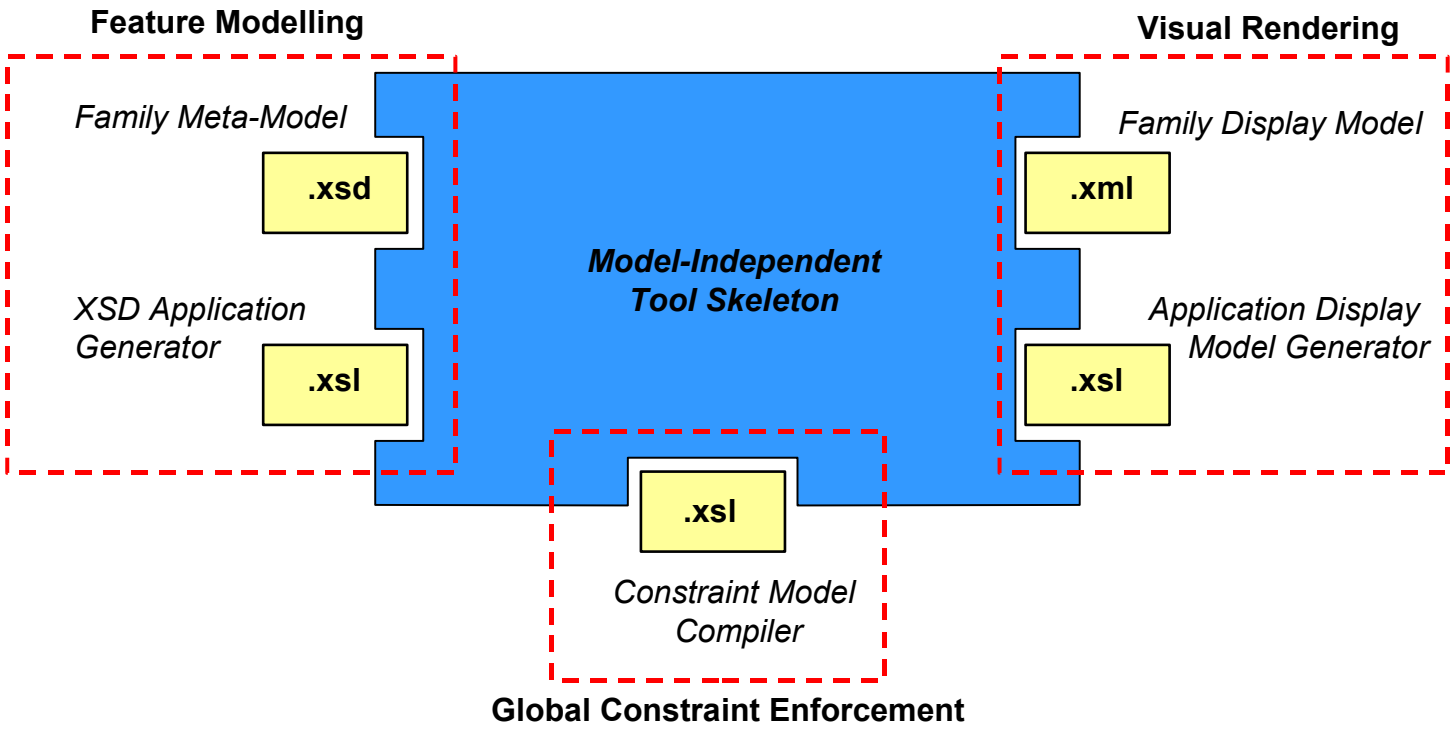


XFeature Principles

- **Feature Model Approach**
  - Both family and product (e.g. an application) models are expressed as feature models
- **Model-Driven Approach**
  - Tool is built on a feature meta-meta-model
  - Users can define their own feature meta-model
- **XML-Based approach**
  - Express a feature meta-model as an XML Schema
  - Express a feature model as an XML document that must comply with the XML Schema
- **Eclipse-Based Approach**
  - Tool is built as a plug-in for the Eclipse platform
- **Open Software Approach**
  - Tool available as free and open software under EPL

XFeature Configuration

ETH Zurich



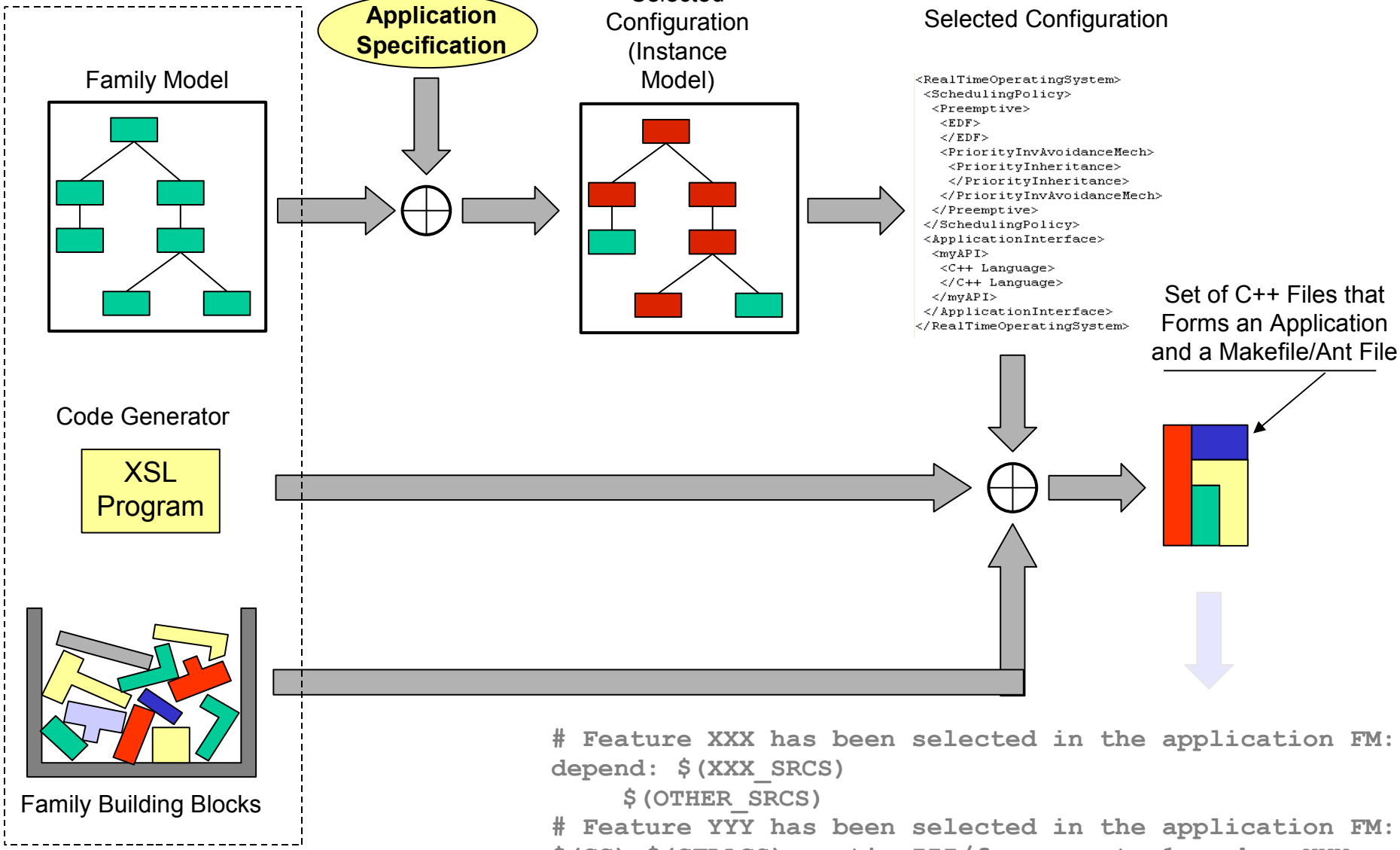
- Users define both the meta-model used by the tool and the tool "look & feel"
- Constraint modelling has been recently implemented

The XFeature Tool
Live Demonstration

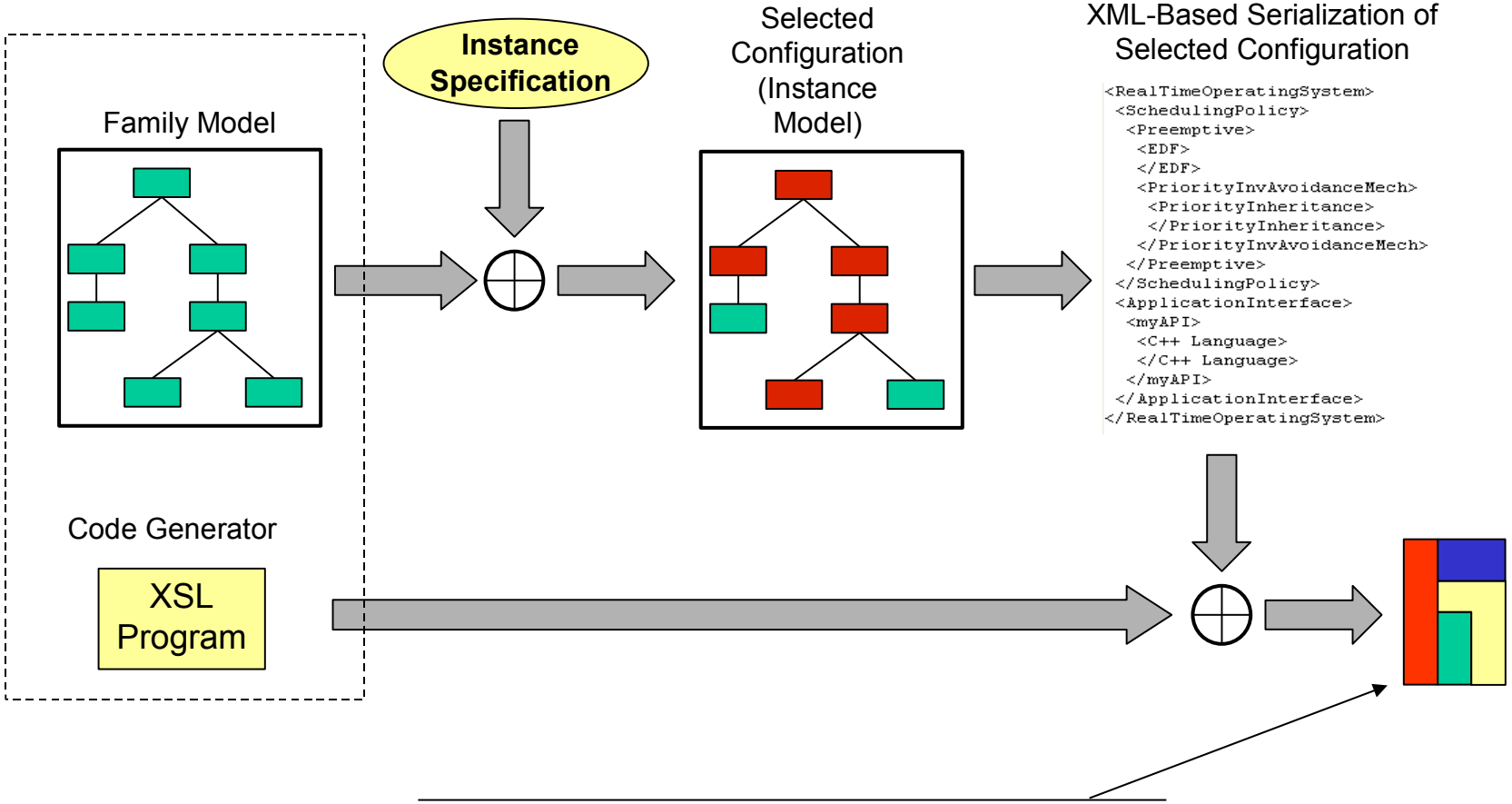
Use of XFeature as Automatic Configuration Tool

Automatic Configuration of an Application from Family BBs

ETH Zurich



Configuration Tool



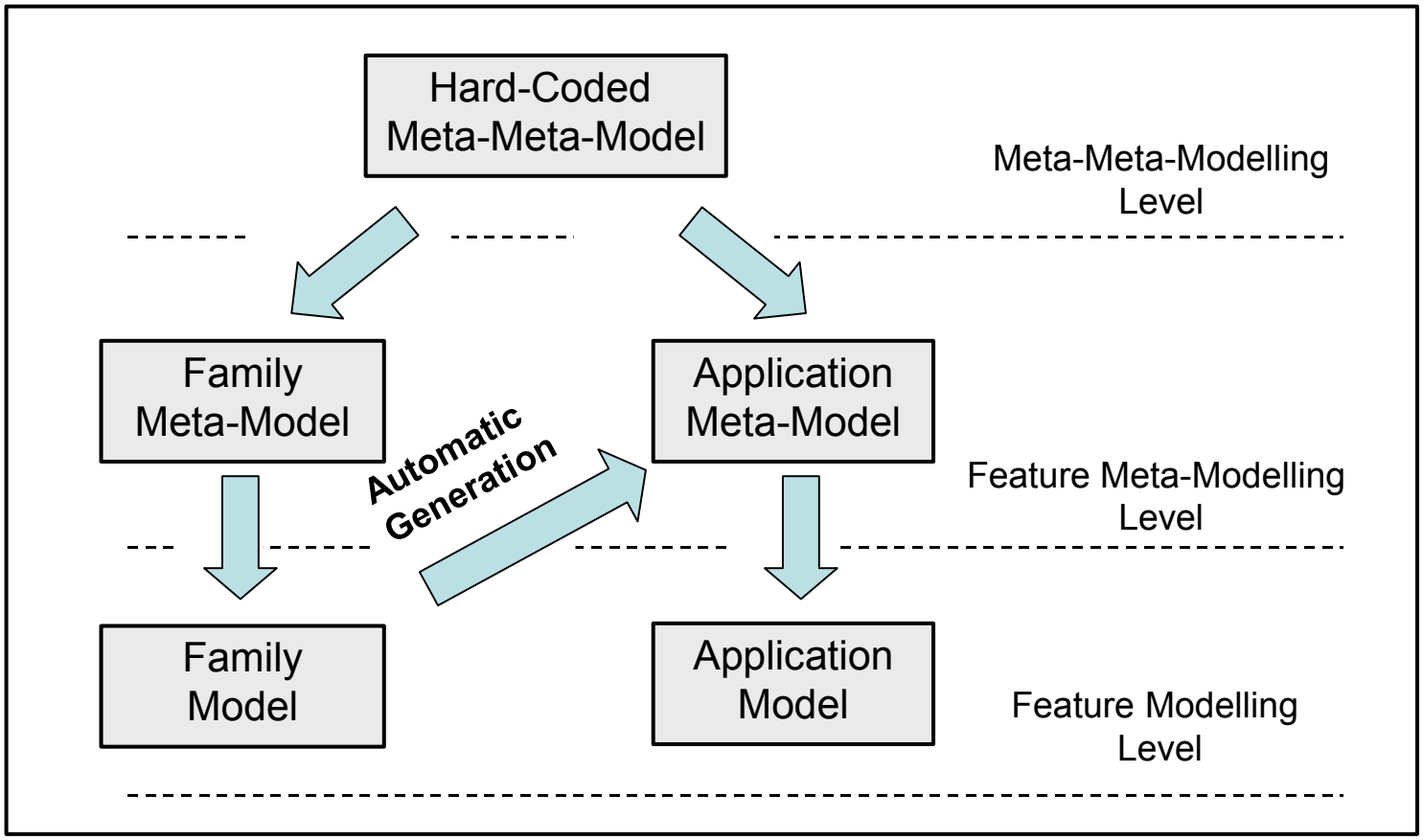
What can be generated from serialized model?

- **Invoice**
- **User Manual**
- **System Specification:** `RAM size in the product 16kB / 64kB`
  `GUI language EN / DE / FR / IT`
  `Version of certain component 1.1 or 1.2`
  `HAV / HBV / HCV`

References

- More Info:
  - XFeature website
    - http://www.pnp-software.com/XFeature
  - Chapter 4 of “Generative Programming”
    - by Krzysztof Czarnecki and Ulrich W. Eisenecker (Addison-Wesley)
  - pure::variants tool
    - Non-graphical Eclipse-based tool
    - http://www.pure-systems.com/3.0.html
  - University of Waterloo
    - Research oriented tool
    - http://gp.uwaterloo.ca/fmp/

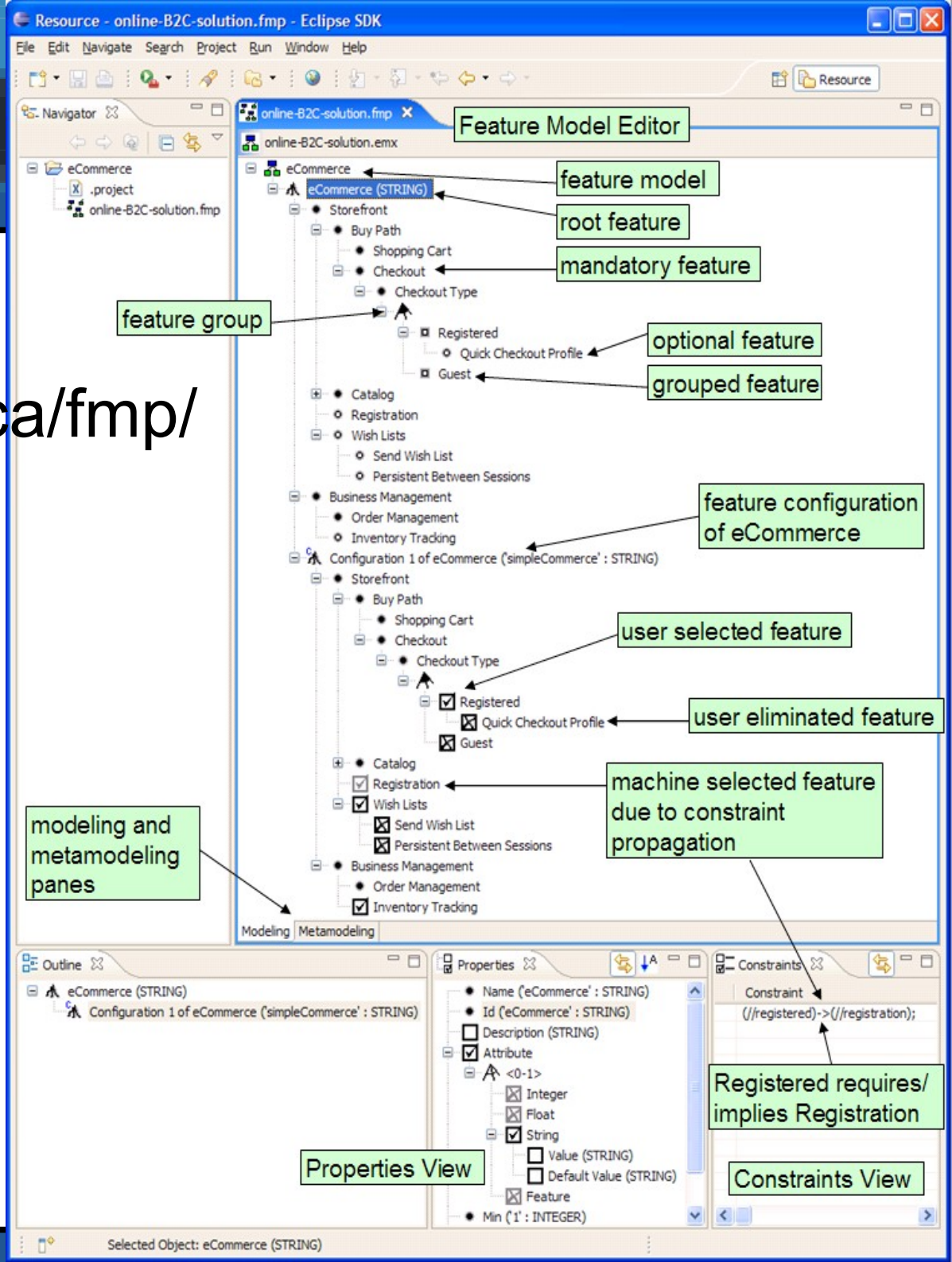
XFeature Architecture



- XFeature users typically operate the tool at Feature modelling Level

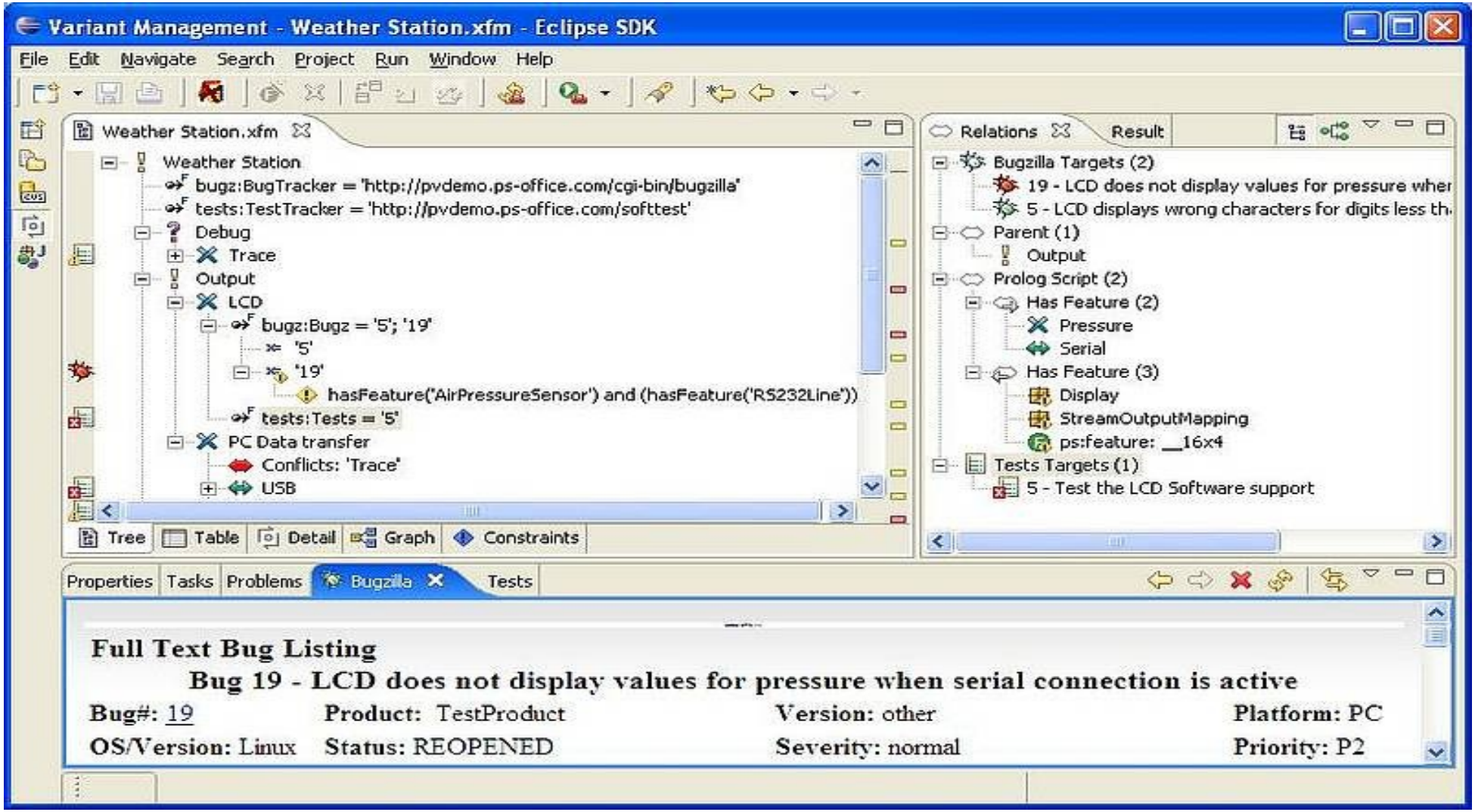
Feature Modelling Plug-in (FMP)

- http://gp.uwaterloo.ca/fmp/



pure::variants

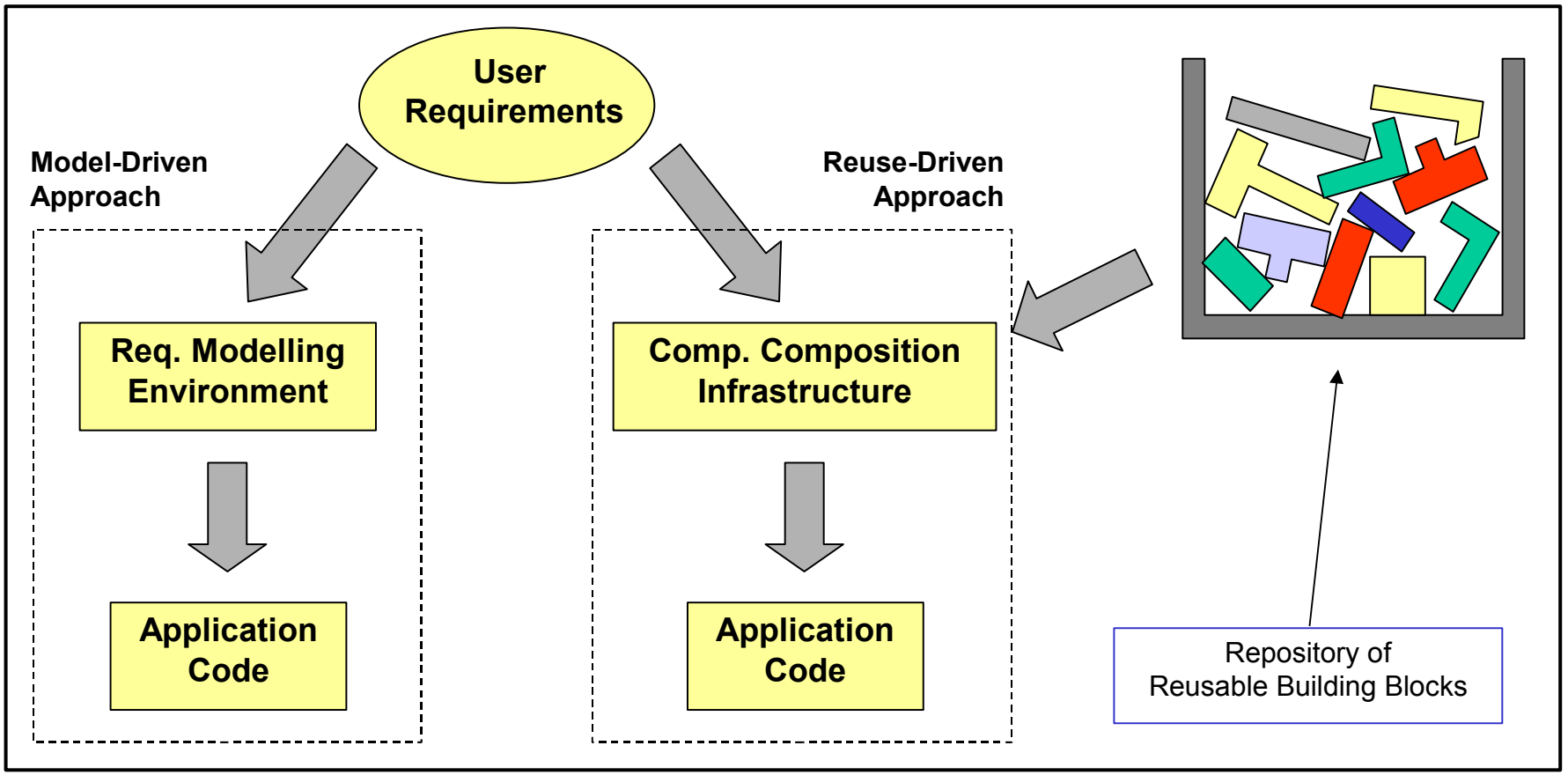
- http://www.pure-systems.com/



Feature Modelling in Software Development

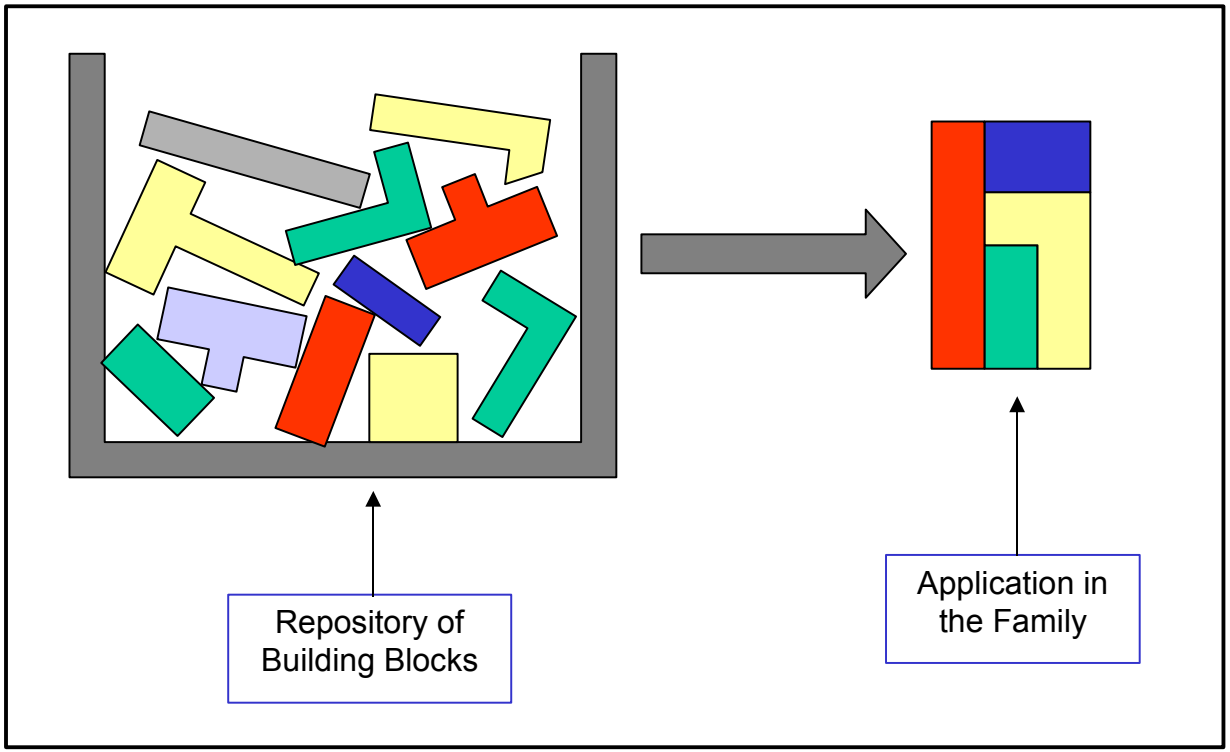
Reuse Driven SW Development

ETH Zurich



Software Product Family

- Software Product Family
  - A set of applications that can be built from a set of shared software assets (the “building blocks”)



Family-Based Process - 1

- Two main stages:

FAMILY CREATION: process whereby the software family and its reusable software **building blocks are created** together with the **language and tools** required to instantiate applications from the family

FAMILY INSTANTIATION: process whereby the assets provided by the software family are tailored to match the needs of a specific application

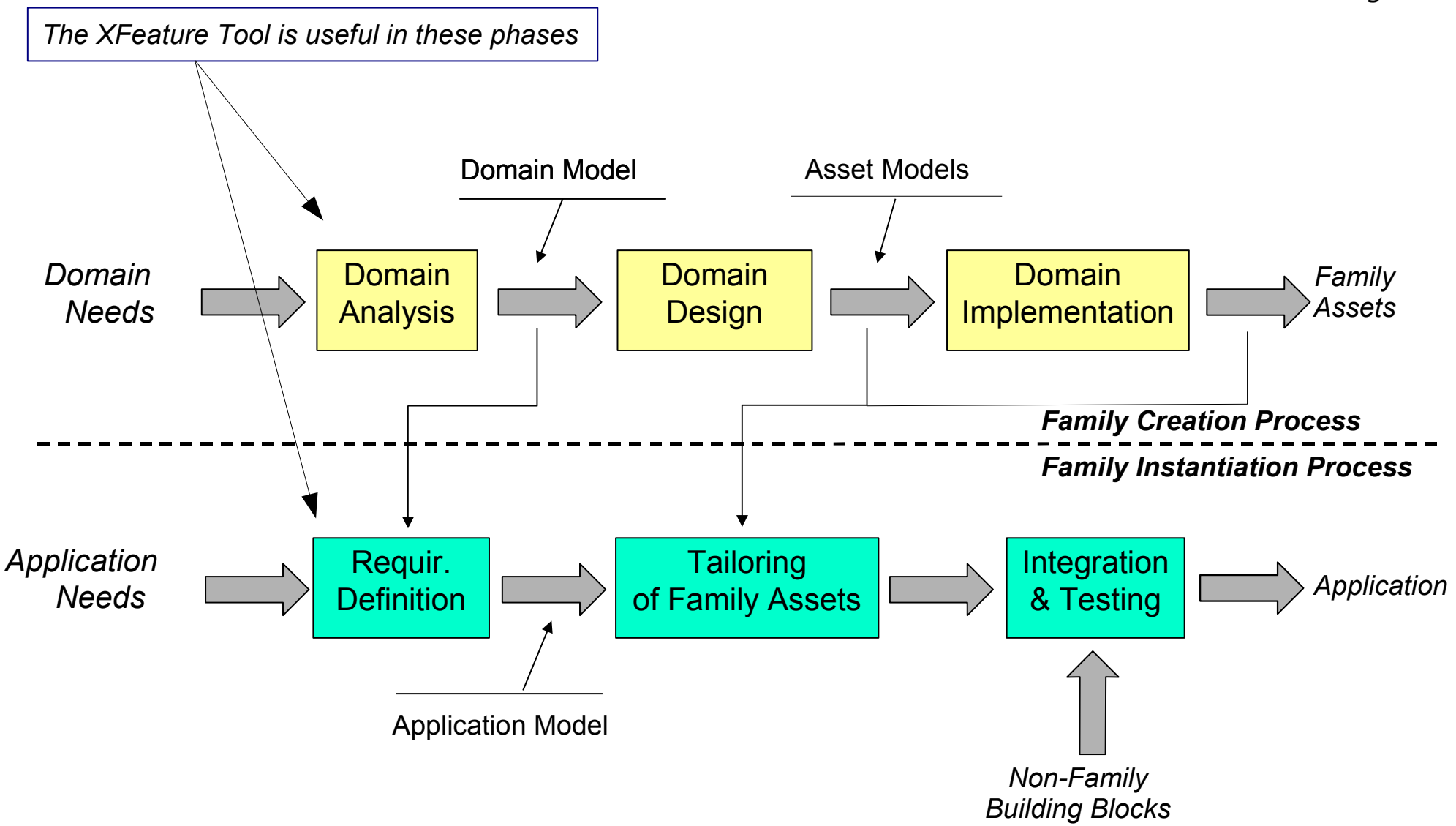
- The family creation process has three main stages:

DOMAIN ANALYSIS: create a model of the family domain

DOMAIN DESIGN: design the shared software assets for the family

DOMAIN IMPLEMENTATION: implement the shared software assets for the family

Family-Based Process - 2



Domain Analysis

- Domain analysis results in the construction of a **domain model**
- The domain model captures the user's point of view on the software framework
  - Similar to user requirements in single application development
- Domain model consists of:
  - Description of the features that the framework supports
  - Definition of constraints on their legal combinations
- Standard Technology to construct Domain Models:
  - **Feature modelling, of course**