

PLE Tool Support



KV Product Line Engineering (343.354)

Dr. Roberto Lopez-Herrejon

Dr. Rick Rabiser



Why and when to use a Tool?

Fixed project size (10 features)–Varying process (regular process and advanced process)

Requirements planning process activities	Simple process: <i>regular process</i>						Complex process: <i>advanced process</i>					
	without tool			with tool			without tool			with tool		
	unit effort	execu- tions	total effort	unit effort	execu- tions	total effort	unit effort	execu- tions	total effort	unit effort	execu- tions	total effort
Capture raw requirement	46	12	552	9	12	108	66	12	792	15	12	180
Resolve raw requirements	0.3	89	26.7	10.4	89	925.6	6.8	89	605.2	13.2	89	1174.8
Resolve problem statement	7	32	224	2.9	32	92.8	10.9	32	348.8	3.2	32	102.4
Resolve features	8.3	10	83	3.2	10	32	97.4	10	974	37.7	10	377
Develop release plan	0*	2	0	0	2	0	180	2	360	68	2	136
Total effort			855			1158.4			3080			1970.2
Productivity	28.04			20.71			7.79			12.18		
Productivity Impact	- 26.1%						+ 56.4%					

* this activity does not involve subactivities affected by tool insertion.

[Bruckhaus et al. 1996]

→ SPLE comprises many complex processes!

Motivation

- ▶ Many activities/processes of SPLE can (and have to) be automated/supported by tools, e.g.,
 - Variability modeling
 - tools hide the concrete syntax behind a graphical user interface and help users to create models in an efficient way
 - Product derivation
 - tools help to make configuration decisions and support constraint propagation, automated conflict resolution, configuration generation
 - Product line planning/scoping
 - tools can facilitate effective collaboration, e.g., for the development of a feature plan
 - ...

Requirements for SPL tools

- ▶ Interoperability and Traceability
 - How to integrate SPL tools with (development) processes, teams, and tools already existing in an organization?
 - How to integrate SPL processes with each other (from PL planning to variability modeling to product derivation)?
- ▶ Adaptability
 - How to work with the assets and dependencies specific to a particular domain?
 - How to react to changes e.g., evolving technology or changing customer needs?
- ▶ Scalability
 - How to support multiple teams/large organizations?
 - How to work with large systems?

Requirements for SPL tools

- ▶ Extensibility
 - How to add additionally required functionality (e.g., 3rd party capabilities)?
- ▶ Usability
 - How „easy“ can different user groups work with the tools?
 - Especially important for „non-software-technicians“
 - Learnability, Efficiency, Memorability, Accuracy, Subjective Satisfaction [Nielsen 1994]
- ▶ Automation
 - Which tasks can be automated?
 - Which tasks require user intervention?
- ▶ Error handling
 - How to react to problems?

Custom-developed „PL“ Tools

- ▶ Typical scenario:
 - Company recognized need to manage variability
 - e.g., too many projects with customers having too many specific requirements, project-specific development no longer accomplishable
 - Company implements their own solution to manage variability
 - Often very close to the code
 - E.g., parameterization, config files, IFDefs
 - Variability is usually not explicitly documented (e.g., in a model)
 - Thus: management of variability becomes a maintenance problem in itself
- ▶ However, there are many commercial and academic PL tools existing
 - They are just not well-known and many of them unfortunately are hard to adopt for practical scenarios

SPL Tools – An Overview

- **The DOPLER Tool Suite**
- **Kumbang Tools**
- **pure::variants**
- **Gears**
- **Feature IDE**
- **Other tools**

The DOPLER Tool Suite



CHRISTIAN DOPPLER LABORATORY FOR
AUTOMATED SOFTWARE ENGINEERING



JOHANNES KEPLER
UNIVERSITY LINZ | **JKU**

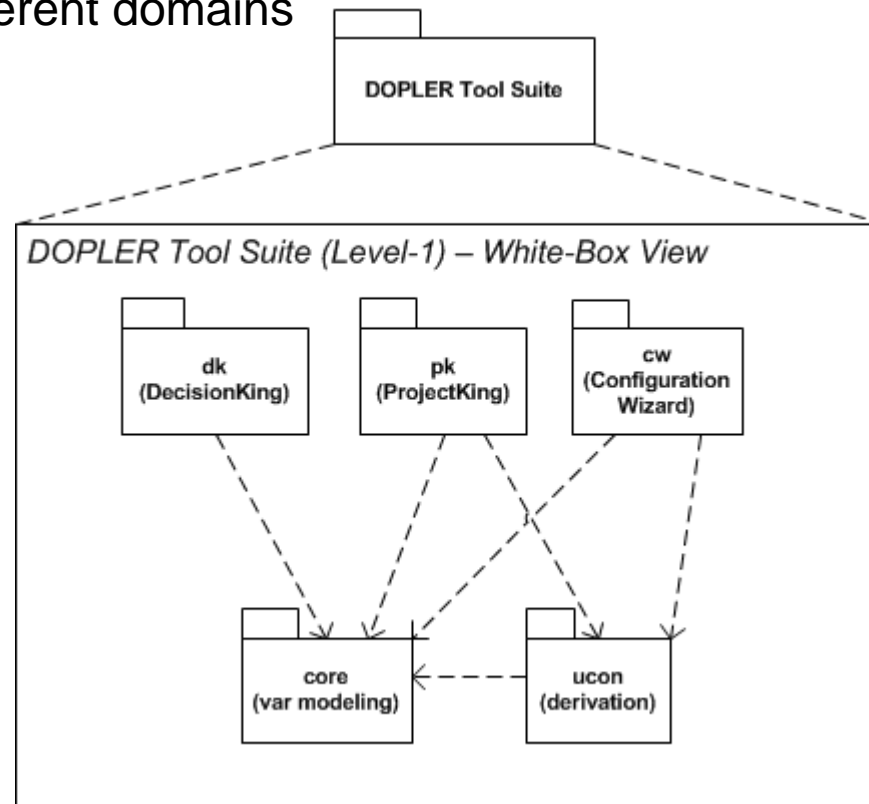
[Dhungana et al. 2006-2009] [Rabiser et al. 2006-2013]

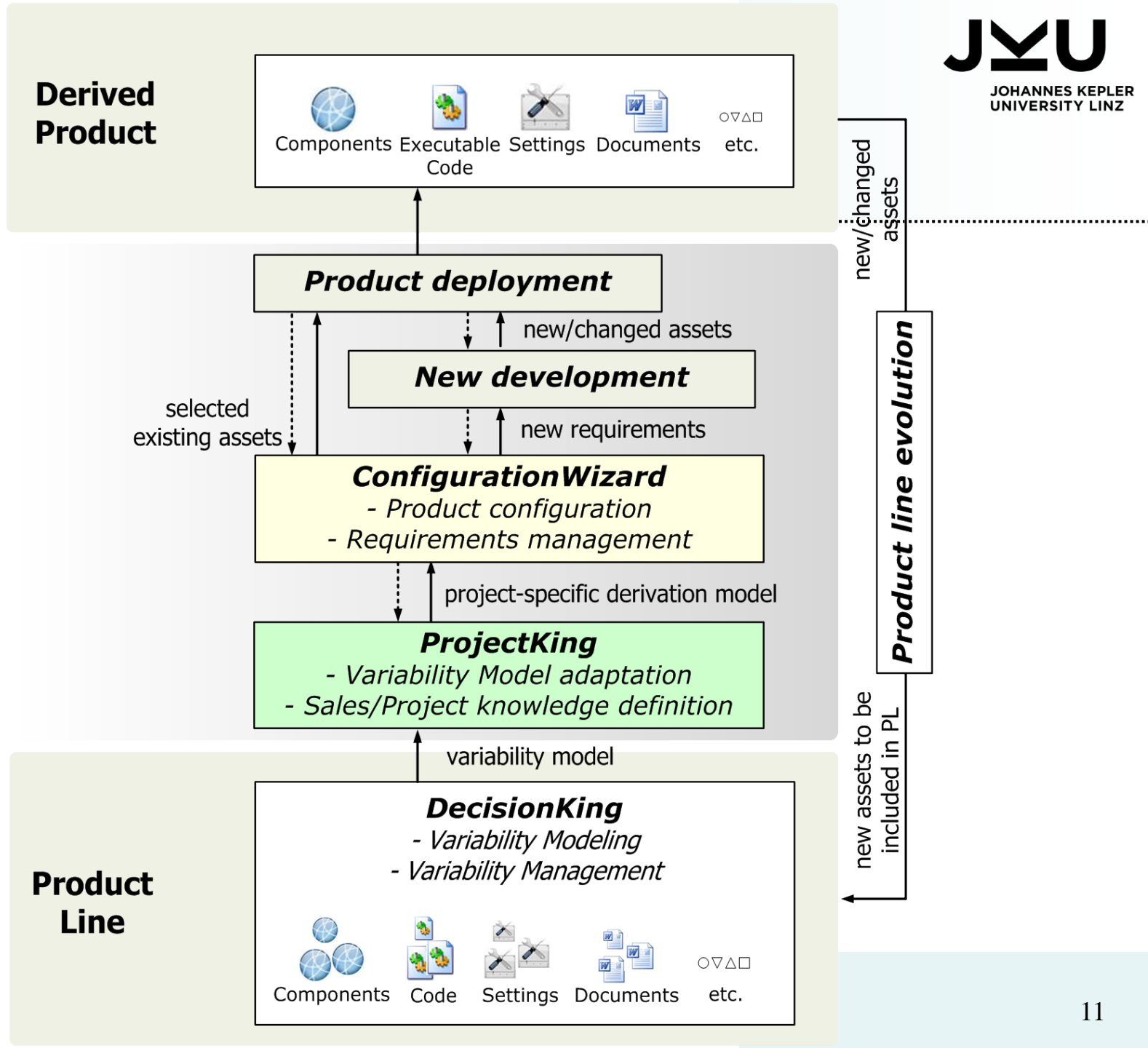
DOPLER Tool Suite

- ▶ Supports Decision-oriented modeling and Wizard-oriented usage of variability
- ▶ Developed at CD Lab for ASE, JKU in collaboration with Siemens VAI Metals Technologies GmbH and Siemens Corporate Technology
- ▶ Flexible and Extensible tool architecture
 - Meta-modeling capabilities for domain-specific adaptations
 - Plug-in Architecture to allow for arbitrary extensions
- ▶ Focuses on the needs of end-users for customizing complex products in an intuitive manner

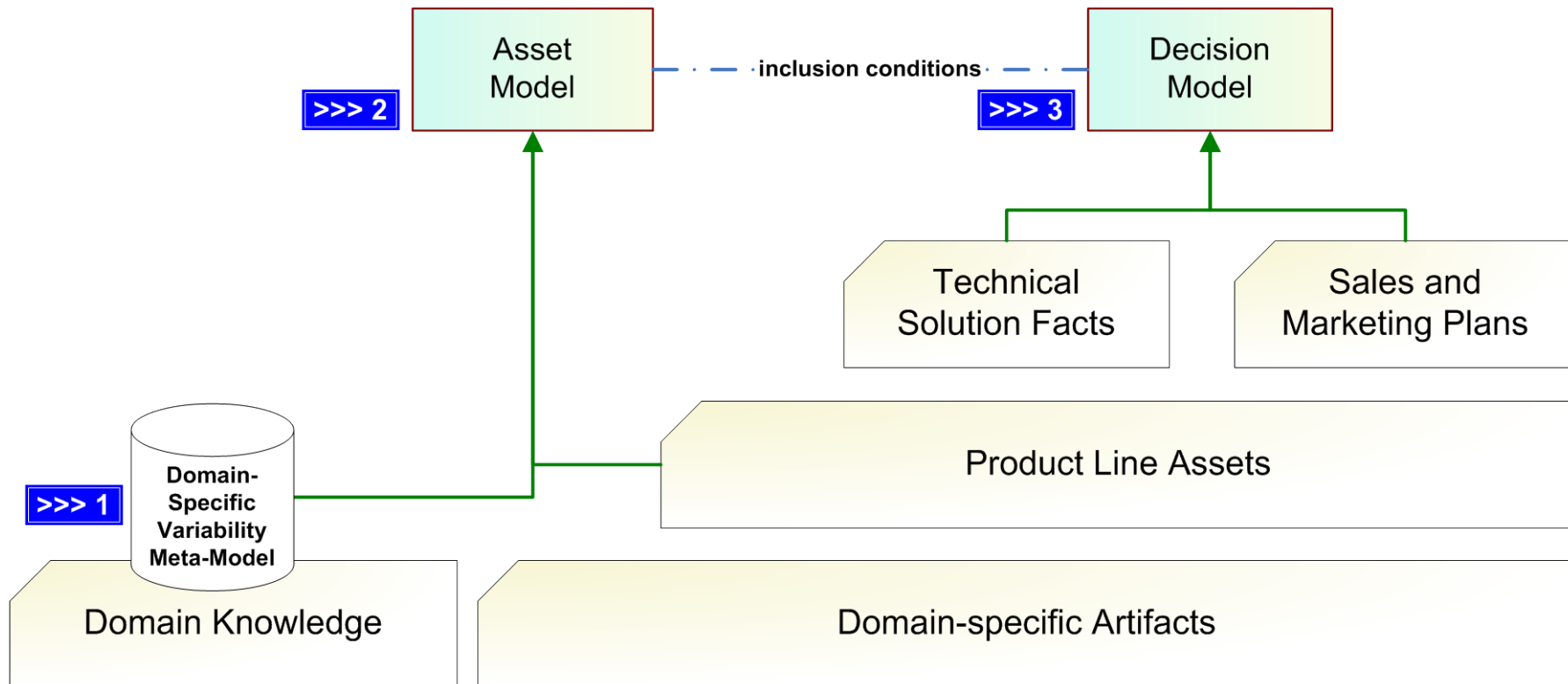
Overview

- ▶ 3 tools (composed of multiple plug-ins) and several (optional) extensions
- ▶ Customizable- but in general independent of languages, notations and practices in different domains





Modeling Approach



DOPLER Modeling Perspective

DOPLER Modeling - models/DOPLER/DOPLER_v26_test.gen - Eclipse Platform

File Edit Navigate Search Project Initialize CheckKing ConfigurationWizard DOPLER Run Server Window Help

Navigator

- models [models]
- ._outdated
- DOPLER
 - .project 10443 10.11.10 16:44 wh
 - artifacts.evoking 10444 10.11.10 16:46 wh
 - DOPLER_v26_export_var.xml 10450 11.11.10 08:45 c
 - DOPLER_v26_test_export_gen.xml 10458 11.11.10 08:45 c
 - DOPLER_v26_test.gen 10482 16.11.10 12:11 cm
 - DOPLER_v26.meta 10446 10.11.10 16:49 cm
 - DOPLER_v26.var 10482 16.11.10 12:11 cm

Review Decisions

List of Decisions

Decision-Question/-Group/Task	Name of Decision	(Possible) Answer(s)	Default Value (From
Unassigned Decisions			
a1?	dec_a1		N/A
CW_resolution_height	CW_resolution_height		0.0
CW_resolution_width	CW_resolution_width		0.0
Do you have old DOPLER models?	DK_modelconverter	false	false
Do you need authentication for product derivation?	CW_authentication	false	false

Pre-conditions for the currently selected Decision

Decision	must fulfill condition
Do you want variability modelling (DecisionKing)?	DK

Project Overview Review Decisions Adapt Model Define Tasks Define Roles/Us

Problems 0 items

Description	Resource	Path	Location
-------------	----------	------	----------

Calculated Values

DOPLER Mo...

- .meta-File
- Meta-Model
- Created with DecisionKing

- .var-File
- Variability Model
- Created with DecisionKing

- .gen-File
- Derivation Model
- Created with ProjectKing

DOPLER Perspective

- Navigator
- Editors
- Problems
- Extensions (additional Views)

DecisionKing: Meta-model Editor

DOPLER_v26.meta

Specify variability modeling language for your domain



- Asset
 - Name
 - Description
 - IncludedIF
- Feature
 - License
 - Copyright
 - Version
 - Location
 - ID
- Plugin
 - ID
 - Version
 - Location
- Setting
 - Key
 - Value
 - File
- Deployment Parameter
 - Unpack
 - Size
 - required eclipse plug-ins
- Extension Point
 - Point ID
- Test
- Model attributes
- Asset relationship
 - requires
 - contributesTo
 - implements
- Decision attributes

Details of Plugin

Name Plural

Properties ☐ Invisible type Icon Order

Description

Parent type

Attributes

Create new Type

ID	String	
Version	String	
Location	Url	

Inherited attributes

Name	
Description	
IncludedIF	

Relationships

Add new link Cardinality To target

requires	*	Plugin	
implements	*	Extension Point	

Variability Model Editor - Decisions

DOPLER_v26.var

enter searchtext

Tasks

- unassigned
 - ALL
 - dec_a1
 - DK
 - DK_builder
 - DK_merger
 - DK_rulelang_editor
 - DK_metamodelling
 - DK_modelconverter
 - DK_sharepoint
 - VAI_specifica
 - PK
 - DK_modelconverter
 - DK_sharepoint
 - VAI_generator
 - CW
 - CW_resolution**
 - CW_autogenerate
 - CW_documentation
 - CW_wordgenerator
 - CW_guidance
 - CW_flatfilegenerator
 - DK_modelconverter
 - CW_legacyViews
 - CW_requirementsView
 - CW_authentication
 - VAI_specifica
 - VAI_generator

List decision 'CW_resolution'

Name: CW_resolution

Flags: ☐ Public

Question: Which default-resolution shall the ConfigurationWizard application h

Default value:

Description:

Visibility condition: CW

Validity condition: true

Rule:

```
if (containsOnly(CW_resolution, Options.CW_re
if (containsOnly(CW_resolution, Options.CW_re
if (containsOnly(CW_resolution, Options.CW_re
if (containsOnly(CW_resolution, Options.CW_re
if (containsOnly(CW_resolution, Options.CW_re
if (containsOnly(CW_resolution, Options.CW_re
```

Possible Values

Cardinality 1 : 1

+ Add - Delete

Display label	Description	Constant	Order
1024x768	<!--description-->	RES1024	1
1280x1024	Default	RES1280	2
1400x1050	<!--description-->	RES1400	3
1600x1200	<!--description-->	RES1600	4
1680x1050	<!--description-->	RES1680	5
800x600	<!--description-->	RES800	6

Decisions Features Plugins Settings Deployment Parameters Extension Points Tests

Variability Model Editor - Assets

The screenshot shows the Variability Model Editor (VME) interface. The title bar indicates the current model is 'DOPLER_v26.var'. The left sidebar contains a search bar and a tree view of plugins. The 'at.jku.ase.cw.docgenerator.word' plugin is selected. The main area displays the details for this plugin.

Details of at.jku.ase.cw.docgenerator.word

- Name: at.jku.ase.cw.docgenerator.word
- Public: ☐
- Description: DOPLER CW Word Document Generator
- IncludedIF: CW_wordgenerator
- ID: at.jku.ase.cw.docgenerator.word
- Version: 2.5.0.v20100818
- Location: <http://cu041.cubit-demo.myproject.collab.net/svn/repos/test/releases/20100818/cw/at.jk>

requires

- at.jku.ase.cw.ui
- at.jku.ase.cw.ui.views.documents
- at.jku.ase.dopler.core
- at.jku.ase.dopler.images
- at.jku.ase.dopler.lib
- at.jku.ase.ucon.core
- at.jku.ase.ucon.lib
- at.jku.ase.ucon.ui.util

implements

- at.jku.ase.cw.ui.generators
- at.jku.ase.cw.ui.views.documents.data

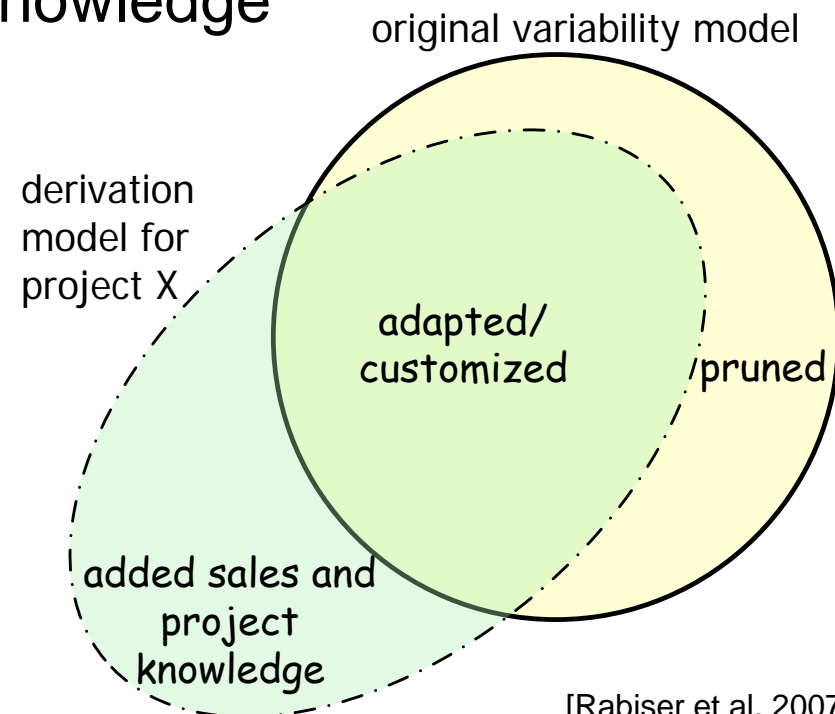
The bottom of the interface features a navigation bar with tabs: Decisions, Features, Plugins (active), Settings, Deployment Parameters, Extension Points, and Tests.

Product Derivation: The Concept of Derivation Models

- ▶ Prepare variability models for concrete projects
 - Identify irrelevant variability
 - Add project-specific/sales knowledge

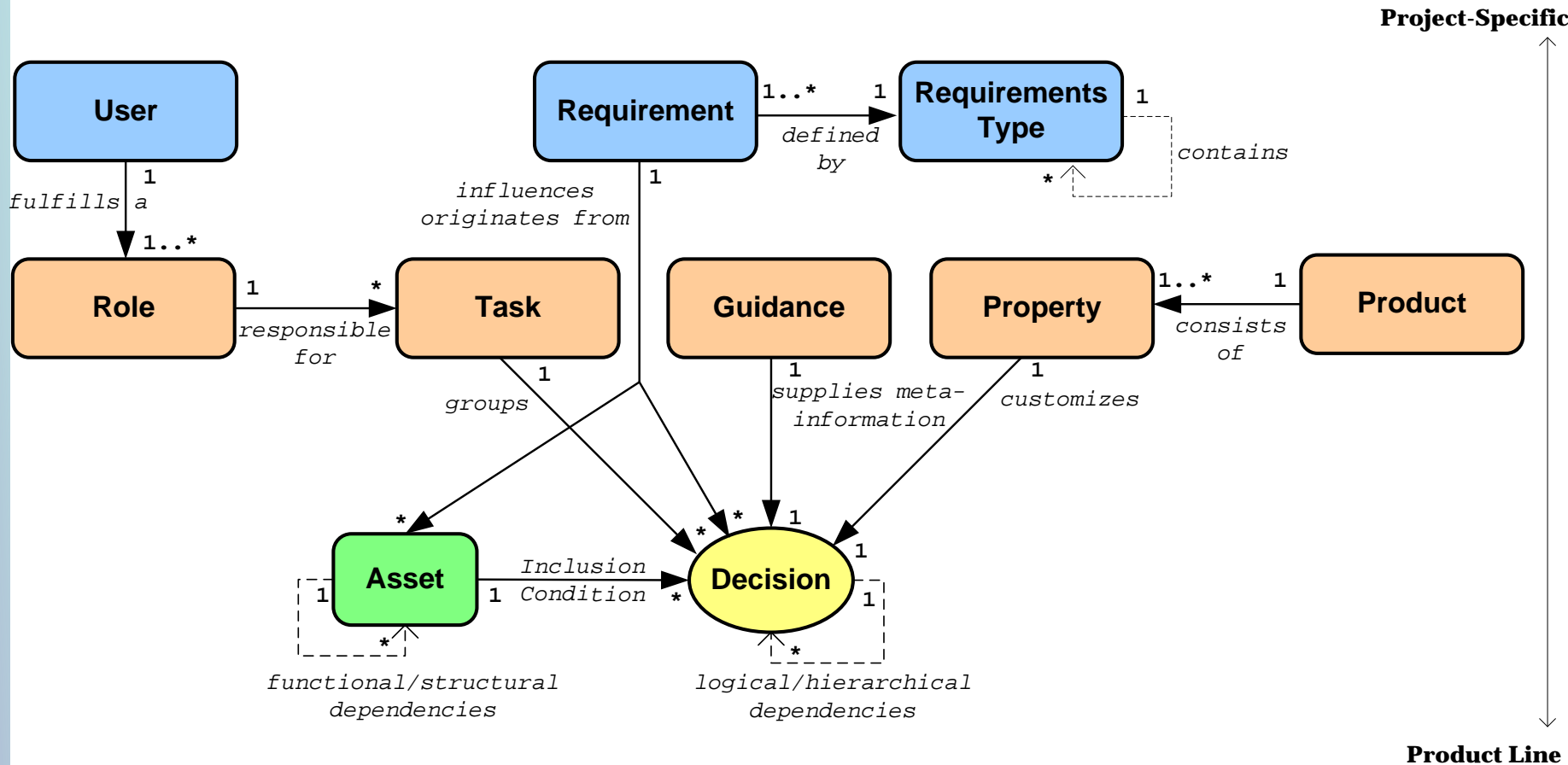
- ▶ **Project-specific Derivation Model**

- Adapted and augmented version of the original variability model
 - Tool-Support: ProjectKing




[Rabiser et al. 2007]

Derivation Meta-Model



ProjectKing: Derivation Model Editor


DOPLER_v26_test.gen

Project Overview

▼ Project details

Project Name

DOPLER_v26



Project Description


This derivation model was automatically generated based on DOPLER_v26.var


Purpose


Variability Model


DOPLER/DOPLER_v26.var






 Refresh VarModel


 Project Overview

 Review Decisions

 Adapt Model

 Define Tasks

 Define Roles/Users

 Review Requirements

Product Derivation with ConfigurationWizard

DOPLER ConfigurationWizard

Project Advanced Help Perspective

1. Start 2. Configure 3. Generate 4. Share Save Filter Decisions Undo Redo

*Decisions (DOPLER_v26_test.gen) enter search text

none

Which default-resolution shall the ConfigurationWizard application have?

CW_resolution_height

CW_resolution_width

Do you want ConfigurationWizard to automatically generate configurations after each save?

Do you want to have documentation integrated in ConfigurationWizard?

Do you want variability modelling (DecisionKing)? yes

Do you want support for derivation and project management (ProjectKing)? yes

Do you want Product Derivation and Configuration? yes

Include MS Word Generator for ConfigurationWizard?

How should guidance on decisions in ConfigurationWizard be displayed? choose

Include consistency checking for variability modelling?

Do you want support for fragmented modelling with model merging?

Do you want a separate editor with synthax support for the rule language?

Do you want to define your own asset types and relationships?

Do you want a generator for simple text output?

Do you want all standard Tools? yes

Do you have old DOPLER models?

Do you want experiemental additional visualisations for product derivation?

Do you want to capture new requirements integrated with the product derivation tool?

Do you want editing support in MS Word to create variable MS Word documents?

2016-04-27

Requirement Required Ass

at.jku.ase.cw.ui

- at.jku.ase.cw.authentication.dumm
- at.jku.ase.cw.docgenerator.word
- at.jku.ase.cw.ui
- at.jku.ase.cw.ui.deploymentparam
- at.jku.ase.cw.ui.intro
- at.jku.ase.cw.ui.nonrcp_feature
- at.jku.ase.cw.ui.rcp_feature
- at.jku.ase.cw.ui.views.documents
- at.jku.ase.cw.ui.views.requirements
- at.jku.ase.decisionking
- at.jku.ase.decisionking.builders.va
- at.jku.ase.decisionking.feature
- at.jku.ase.decisionking.metamode
- at.jku.ase.dopler.core
- at.jku.ase.dopler.images
- at.jku.ase.dopler.lib
- at.jku.ase.dopler.rulelanguage.inte
- at.jku.ase.dopler.sharepoint
- at.jku.ase.pk
- at.jku.ase.pk.feature
- at.jku.ase.ucon.core
- at.jku.ase.ucon.lib
- at.jku.ase.ucon.ui.util
- at.jku.ase.ucon.util
- CW_settings
- PK_settings

Application Reqs. Mgmt in ConfigurationWizard

The screenshot shows the DOPLER ConfigurationWizard application in the '2. Configure' step. The main window displays a list of configuration questions with checkboxes and a 'yes'/'no' column. The 'Requirements' panel on the right shows a 'List of Requirements' with 'GuidanceRqt' selected, and a 'Details' section with fields for Title, Description, Justification, and Origin.

Configuration Questions:

Question	Answer
Which default-resolution shall the ConfigurationWizard application have?	
CW_resolution_height	
CW_resolution_width	
Do you want ConfigurationWizard to automatically generate configurat...	
Do you want to have documentation integrated in ConfigurationWizard?	
Do you want variability modelling (DecisionKing)?	yes
Do you want support for derivation and project management (ProjectKing)?	yes
Do you want Product Derivation and Configuration?	yes
Include MS Word Generator for ConfigurationWizard?	
How should guidance on decisions in ConfigurationWizard be displayed?	
Include consistency checking for variability modelling?	
Do you want support for fragmented modelling with model merging?	
Do you want a seperate editor with synthax support for the rule langua...	
Do you want to define your own asset types and relationships?	
Do you want a generator for simple text output?	
Do you want all standard Tools?	yes
Do you have old DOPLER models?	
Do you want experiemental additional visualisations for product derivat...	
Do you want to capture new requirements integrated with the product ...	
Do you want editing support in MS Word to create variable MS Word docu...	
Do you want a generator for product derivation?	

Requirements Panel:

List of Requirements

- Requirements
 - GuidanceRqt

Details

Title: GuidanceRqt

Description: guidance in the CW should be displayed like... This is currently not an option that can be chosen.

Justification: users want to...

Origin: CW_guidance

OK

2016-04-27

Kumbang Tools

SoberIT

Software Business and Engineering Institute

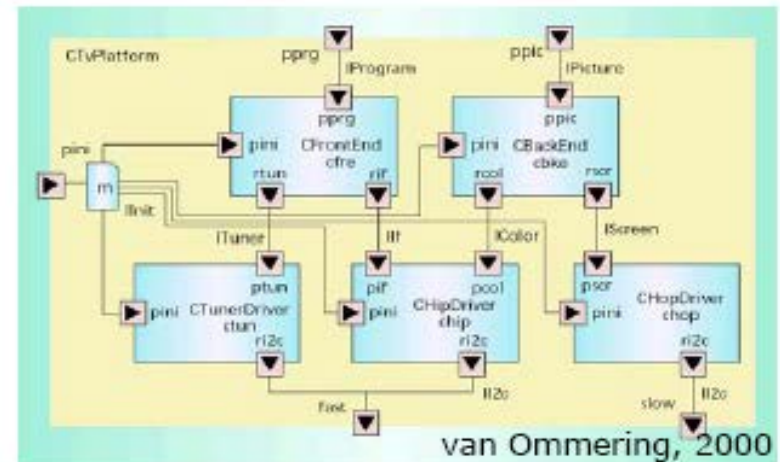
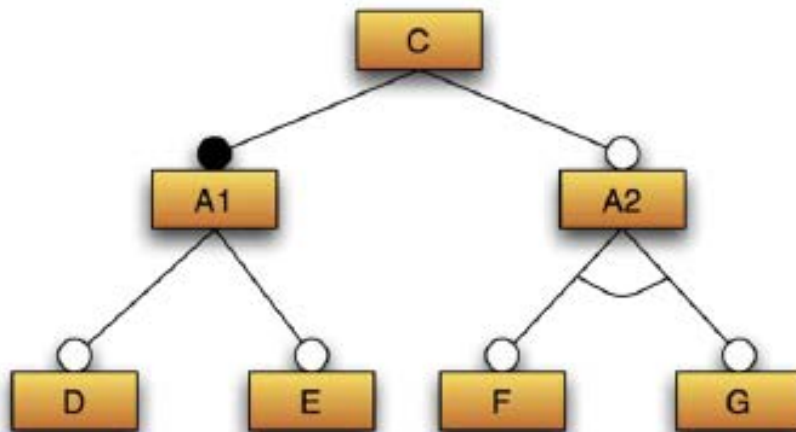


HELSINKI UNIVERSITY OF TECHNOLOGY

[Myllärniemi 2007]

Kumbang Tools

- ▶ Developed at Helsinki University of Technology, SoberIT
- ▶ Eclipse plug-ins, AI inference engine (smodels), GPL
- ▶ Basic Idea:
 - Combining feature modeling and structural modeling
 - Meta-Model called Kumbang

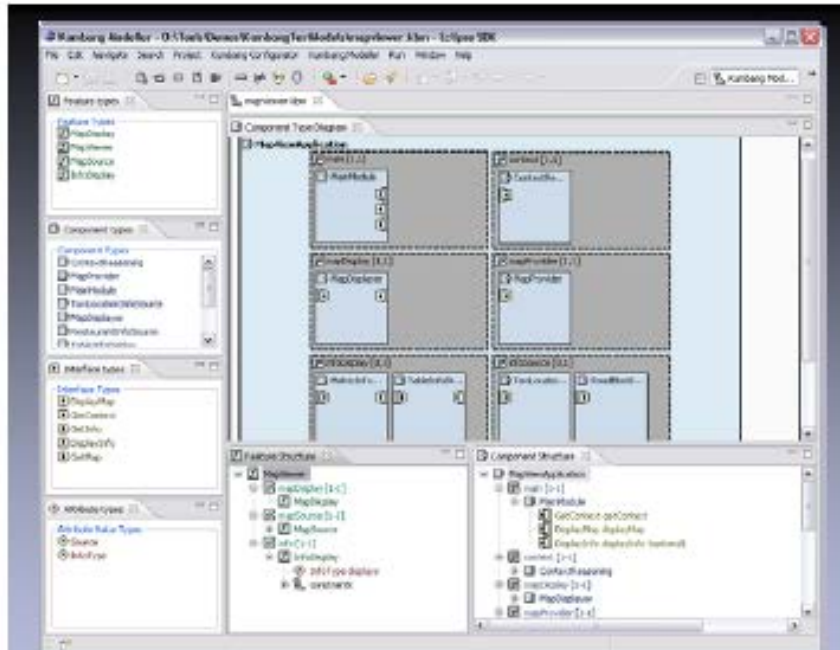


Kumbang Meta-Model

- ▶ Meta-model and language for configurable applications
- ▶ Modeling variability from two viewpoints adhering to IEEE 1471-2000 standard (Recommended Practice for Architecture Description of Software-Intensive Systems)
 - User-visible aspects: Feature Model
 - Structure of products in terms of components, interfaces, ports, etc.: Koala
 - Interrelations between FM/Koala
- ▶ Formal semantics by defining a mapping from the meta-model to Weight Constraint Rule Language

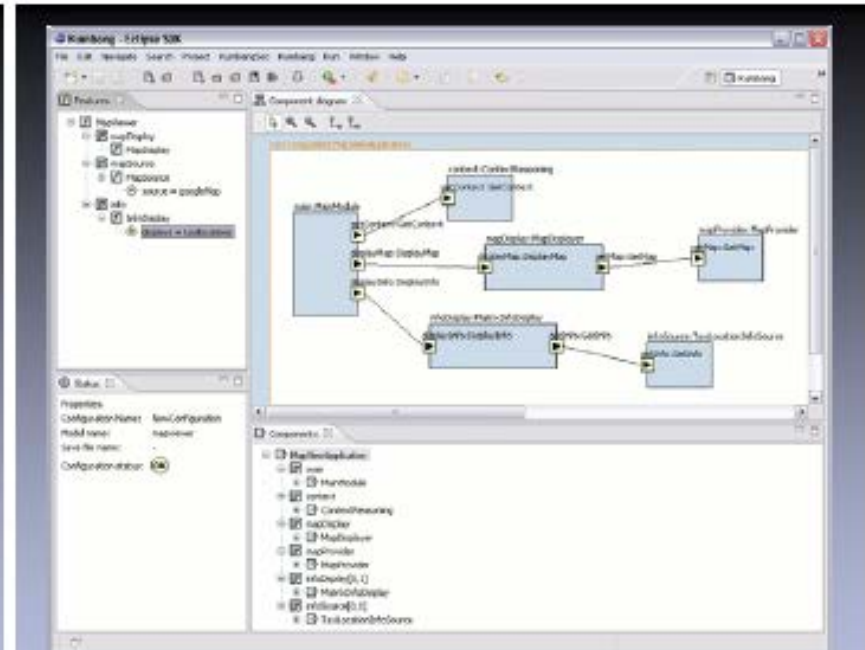
Kumbang Architecture Overview

Types



**Domain engineering by
Kumbang Modeller**

Instances



**Application engineering by
Kumbang Configurator**

**Kumbang Core
uses AI inference engine smodels**

Kumbang Modeller - foo/mapviewer.kbm - Eclipse SDK

File Edit Navigate Search Project Kumbang Configurator Kumbang Modeller Run Window Help

Model name: Change Model type:

Component Type Diagram

Feature Types

Feature Types

- ☒ MapDisplay
- ☒ MapViewer
- ☒ MapSource
- ☒ InfoDisplay

Attributes

Attribute types

Attribute Value Types

- ☒ Source
- ☒ InfoType

Properties

Properties

Property	Value
Name	InfoType
Value	nothing
Value	taxilocations
Value	roadblocks
Value	lunchrestaurants

Component Types

Component Types

- ☒ MapProvider
- ☒ ContextReasoning
- ☒ MapDB
- ☒ TaxiLocationInfoSource
- ☒ MapDisplayer
- ☒ RestaurantInfoSource
- ☒ TableInfoDisplay

Interface Types

Interface Types

- ☒ DisplayMap
- ☒ GetContext
- ☒ GetInfo
- ☒ DisplayInfo

Feature Structure

Feature Structure

- ☒ MapViewer
 - ☒ mapDisplay [1..1]
 - ☒ MapDisplay
 - ☒ mapSource [1..1]
 - ☒ MapSource
 - ☒ info [1..1]
 - ☒ InfoDisplay
 - ☒ InfoType displays
 - value(displays) = nothing <=> not present
 - value(displays) != nothing => present(location)
 - value(displays) = taxilocations <=> instance
 - value(displays) = roadblocks <=> instance
 - value(displays) = lunchrestaurants <=> instance

Component Structure

Component Structure

- ☒ MapViewerApplication
 - ☒ main [1..1]
 - ☒ MainModule
 - ☒ context [1..1]
 - ☒ ContextReasoning
 - ☒ mapDisplay [1..1]
 - ☒ MapDisplayer
 - ☒ mapProvider [1..1]
 - ☒ MapProvider
 - ☒ mapDB [1..1]
 - ☒ MapDB
 - ☒ GetMap getmap (grounded)
 - ☒ GetMap getMap
 - ☒ infoDisplay [0..1]
 - ☒ MapInfoDisplay
 - ☒ MapInfoDisplay
 - ☒ infoSource [0..1]

File Edit Navigate Search Project Kumbang Configurator Kumbang Modeller Run Window Help

Features 23

MapViewer

- MapDisplay
- MapSource
- InfoDisplay
- displays = taxilocations

Components 23

MapViewApplication

- MainModule
- ContextReasoning
- MapDisplayer
 - getMap: GetMap
 - mapDB: MapDB . getma
 - mapDisplay: MapDisplay
- MapDB
 - getmap: GetMap
 - mapProvider: MapP
- MatrixInfoDisplay
 - displayInfo: DisplayInfo
 - main: MainModule . disp
 - getInfo: GetInfo
 - infoSource: TaxiLocatio
- TaxiLocationInfoSource
 - getInfo: GetInfo
 - infoDisplay: MatrixInfoD

Component diagram 23

```
graph LR
    subgraph MapViewApplication
        MainModule
        ContextReasoning
        MapDisplayer
        MapProvider
        MatrixInfoDisplay
        TaxiLocationInfoSource
    end
    MainModule --> ContextReasoning : getContext: GetContext
    MainModule --> MapDisplayer : displayMap: DisplayMap
    MainModule --> MatrixInfoDisplay : displayInfo: DisplayInfo
    ContextReasoning --> MainModule : getContext: GetContext
    MapDisplayer --> MapProvider : getMap: GetMap
    MapProvider --> MapDisplayer : getMap: GetMap
    MatrixInfoDisplay --> TaxiLocationInfoSource : getInfo: GetInfo
    TaxiLocationInfoSource --> MatrixInfoDisplay : getInfo: GetInfo
```

Status 23

Properties:

- Configuration Name: NewConfiguration
- Model name: mapviewer
- Save file name: -
- Configuration complete: OK
- Configuration consistent: Yes

Properties 23

Property	Value
1. Component name	mapProvider
2. Component type	MapProvider
3. Subcomponents	mapDB
4. Interfaces	getMap: GetMap
5. Attributes	

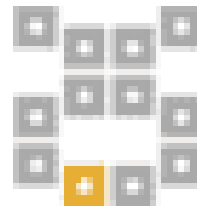
Features

Components

Component Diagram

(c) Mikko Raatikainen, 2007

Pure::Variants



pure-systems

[pure systems GmbH]

pure::variants

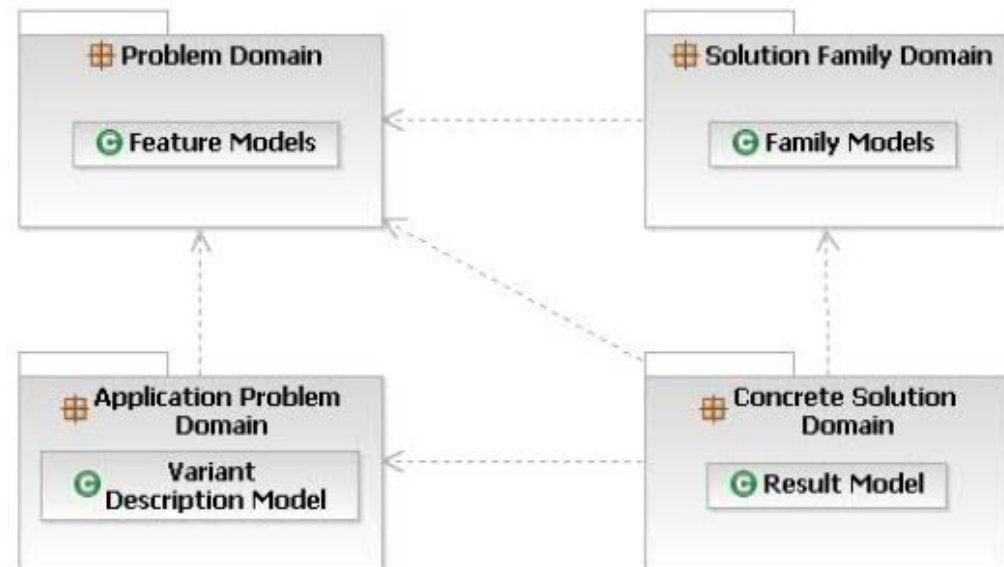
[pure systems GmbH]

- ▶ Developed by pure-systems GmbH/Magdeburg
- ▶ Commercial tool available in an evaluation, community, developer, integration, and server edition
- ▶ Variant management and product configuration based on feature models
- ▶ Uses a Prolog-based constraint-solver
- ▶ Strong focus on integration and extensibility
 - E.g., pv can be integrated in the Eclipse IDE, used with a webbrowser, in a custom application or as a command line client
 - Several extensions to other tools exist, e.g., to the requirements management tool DOORS or to the enterprise business tool SAP

Family-based software development with pure::variants

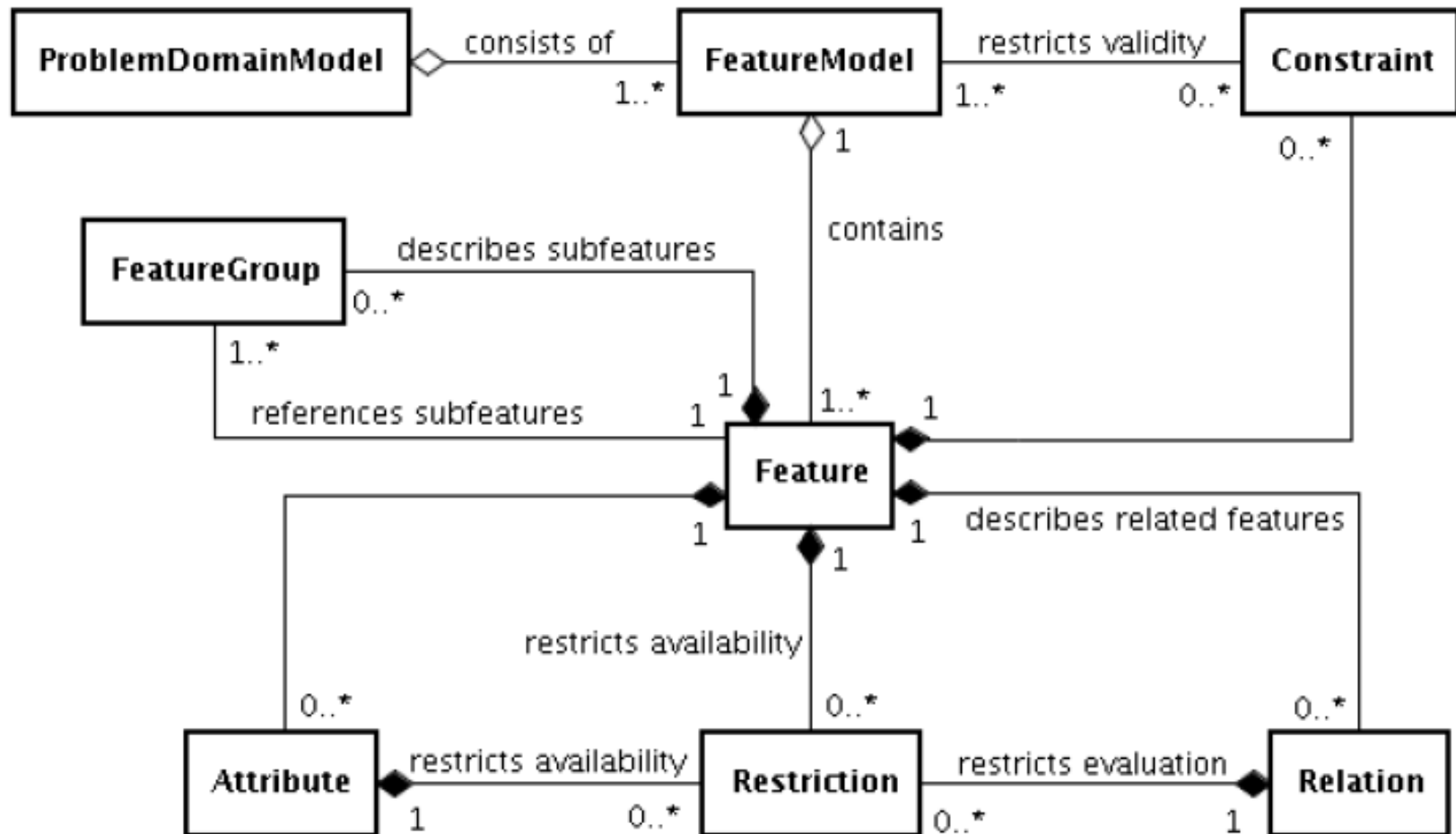
[pure systems GmbH]

- ▶ (Hierarchical) feature models to represent the problem domain
- ▶ Concrete design and implementation of the PL in family models
- ▶ Variant description model to store selected features and values
- ▶ Result model describes a concrete instance from the PL



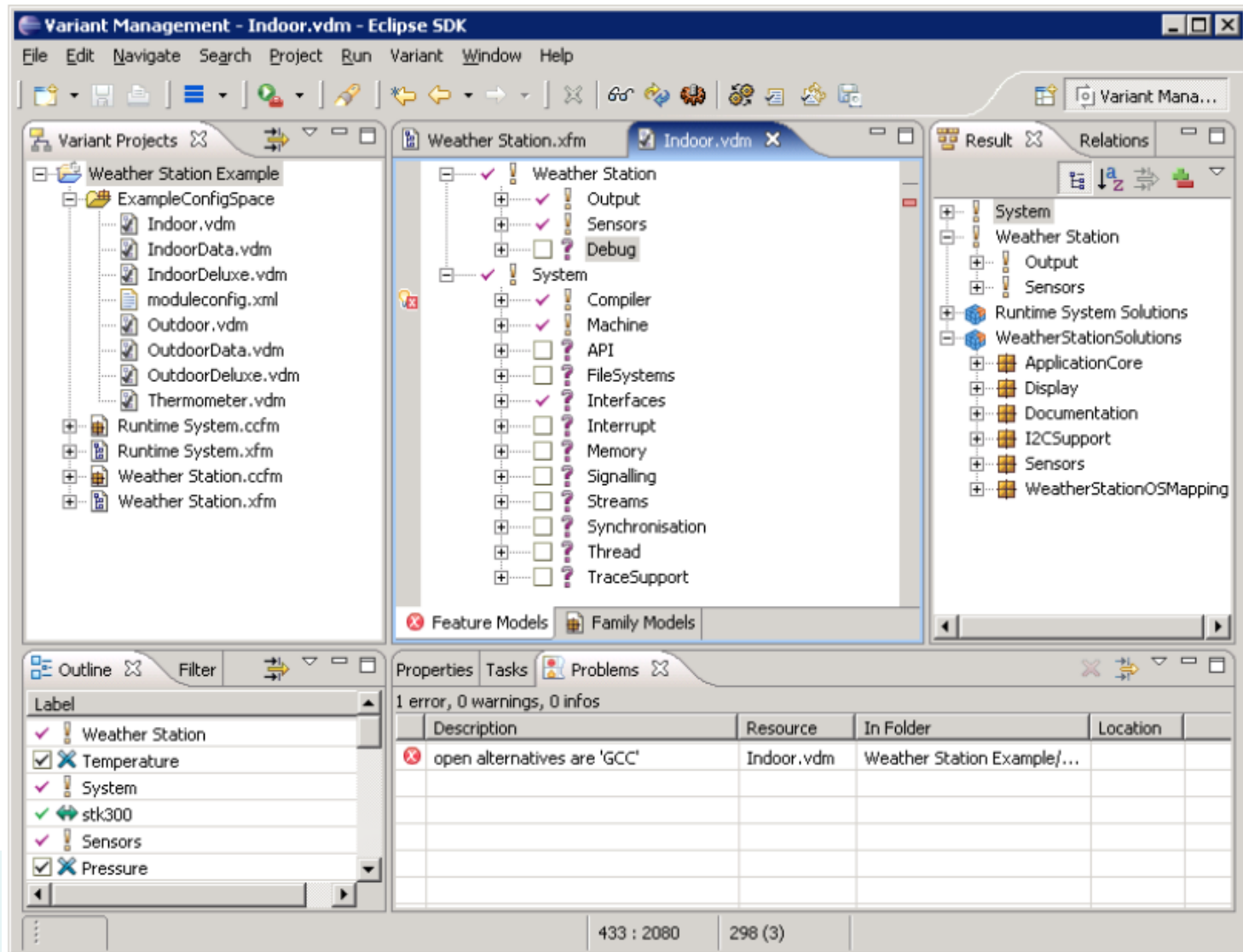
pure::variants feature models

[pure systems GmbH]



pure::variants tool screenshot

[pure systems GmbH]



2016-04-27

pure::variants and constraints

[pure systems GmbH]

► Mo

-
-

► Ele

-
-

► Ele

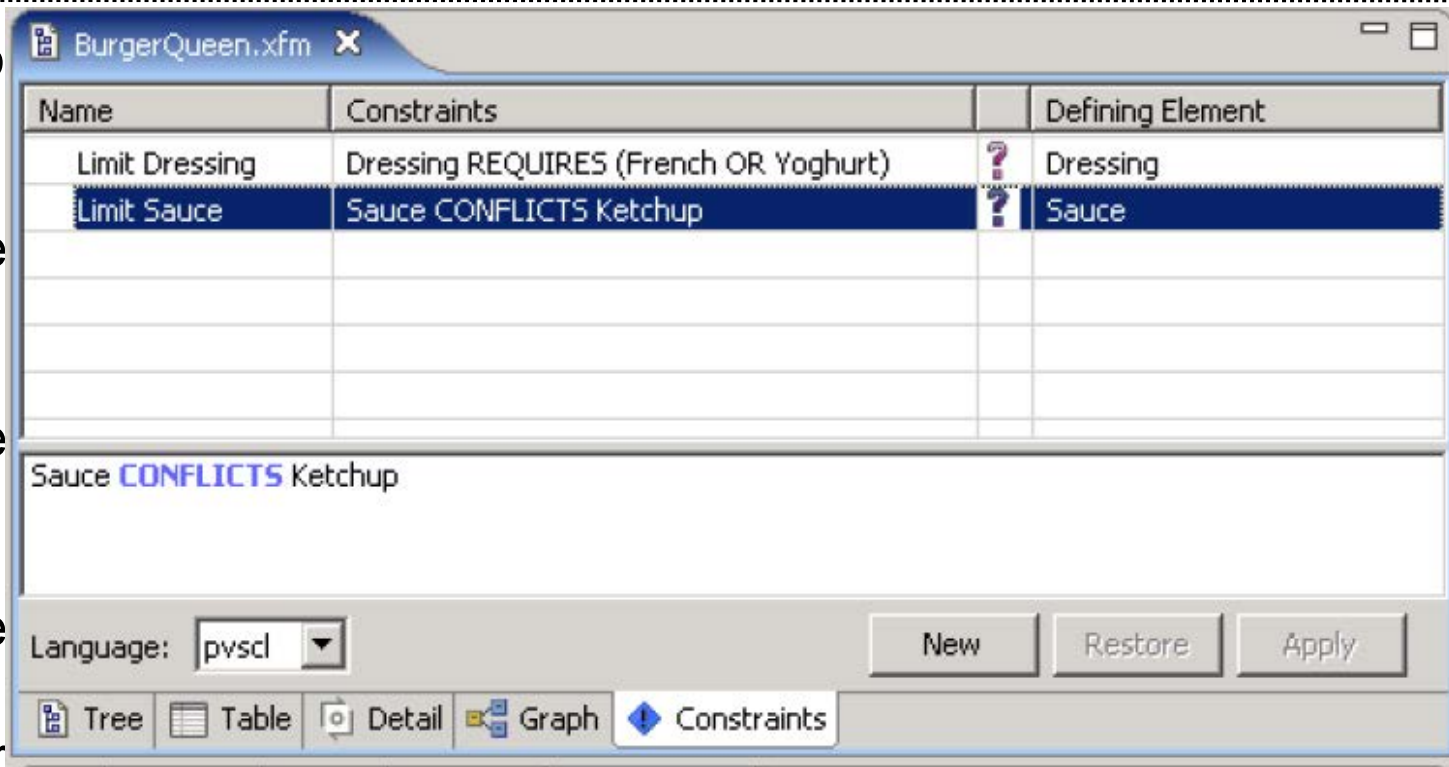
-
-

► Ele

-

► pur

- Dialect of Prolog
- Grammar and available rules, see: <http://www.pure-systems.com/fileadmin/downloads/pure-variants/doc/pv-user-manual.pdf>



s

ons

Pure::Variants Screencast

- ▶ 10min tutorial
 - <http://www.pure-systems.com/flash/pv-spl-10min/flash.html>

Gears

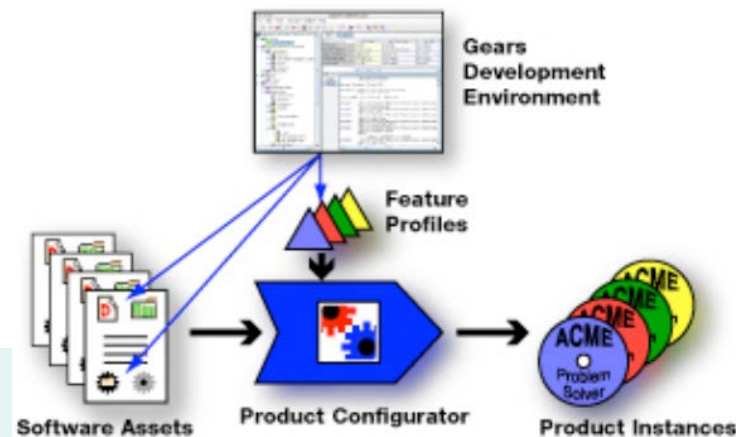


(c) BigLever, Inc.

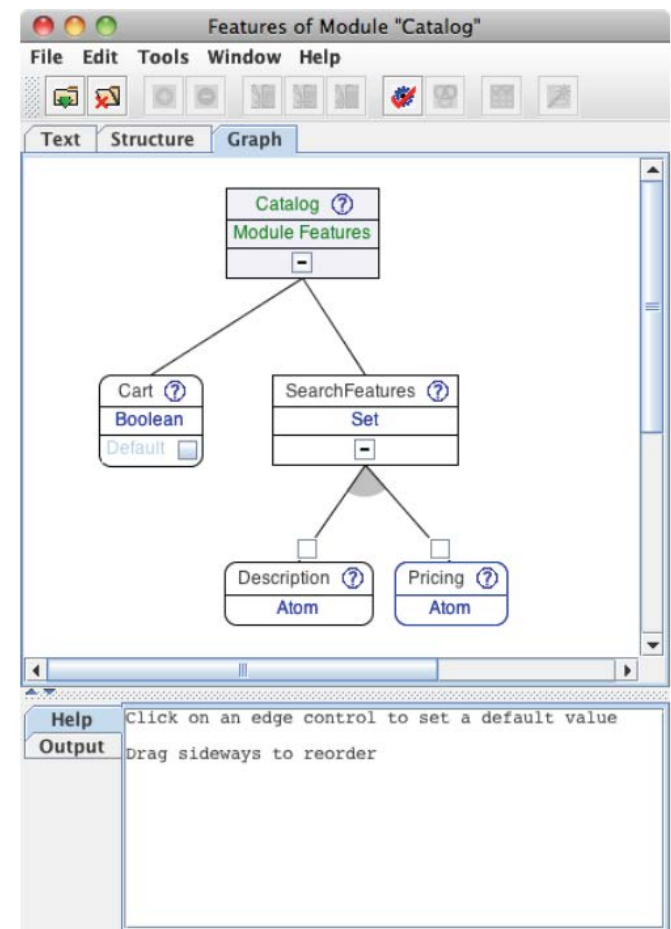
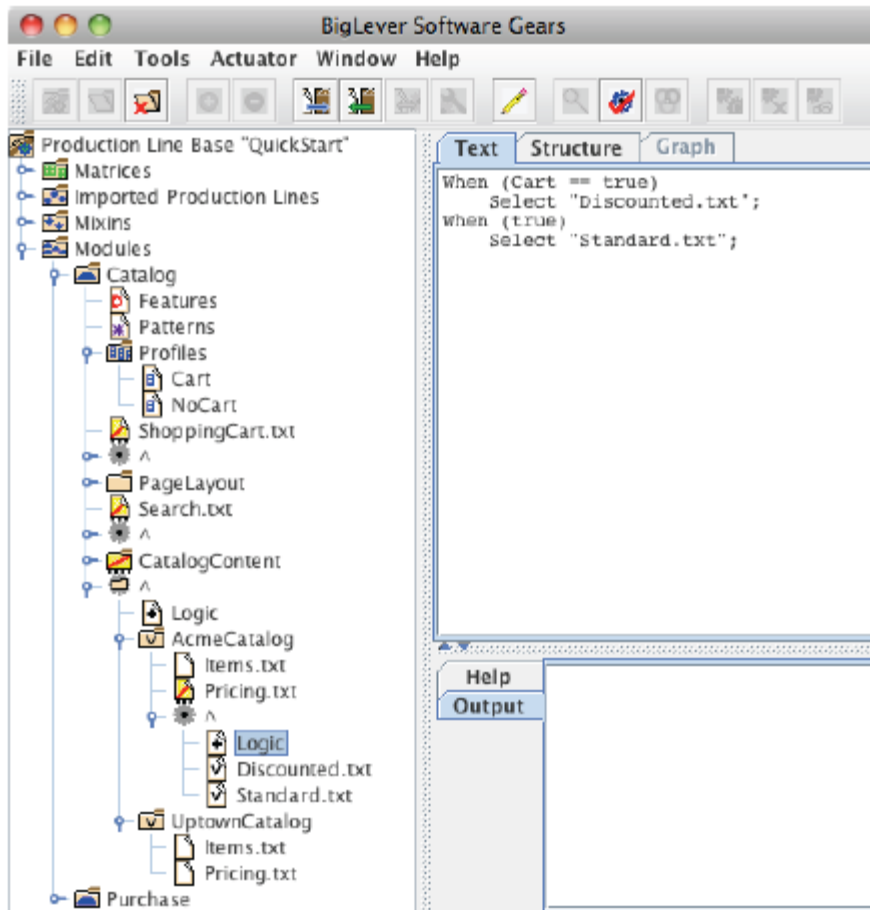
Basic Concepts of Gears

▸ 3 key elements:

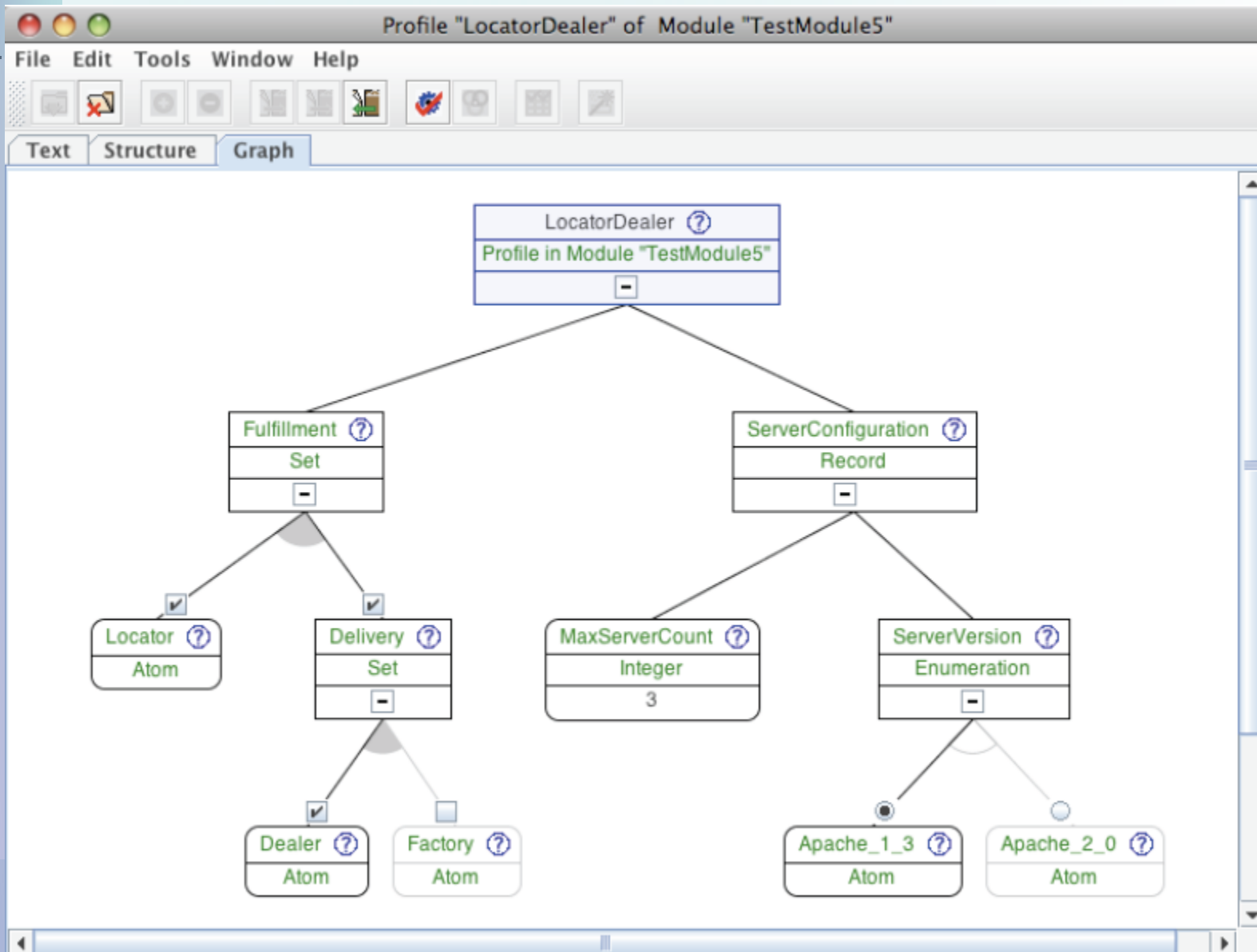
- Software Assets: the configurable software artifacts
- Feature Profiles: model each product in terms of optional and varying feature choices for the PL
- Product Configurator: assembles and configures the software assets, guided by the product feature profiles, to produce each of the product instances



Gears Development Environment



Feature Profile Graph Editor



FeatureIDE

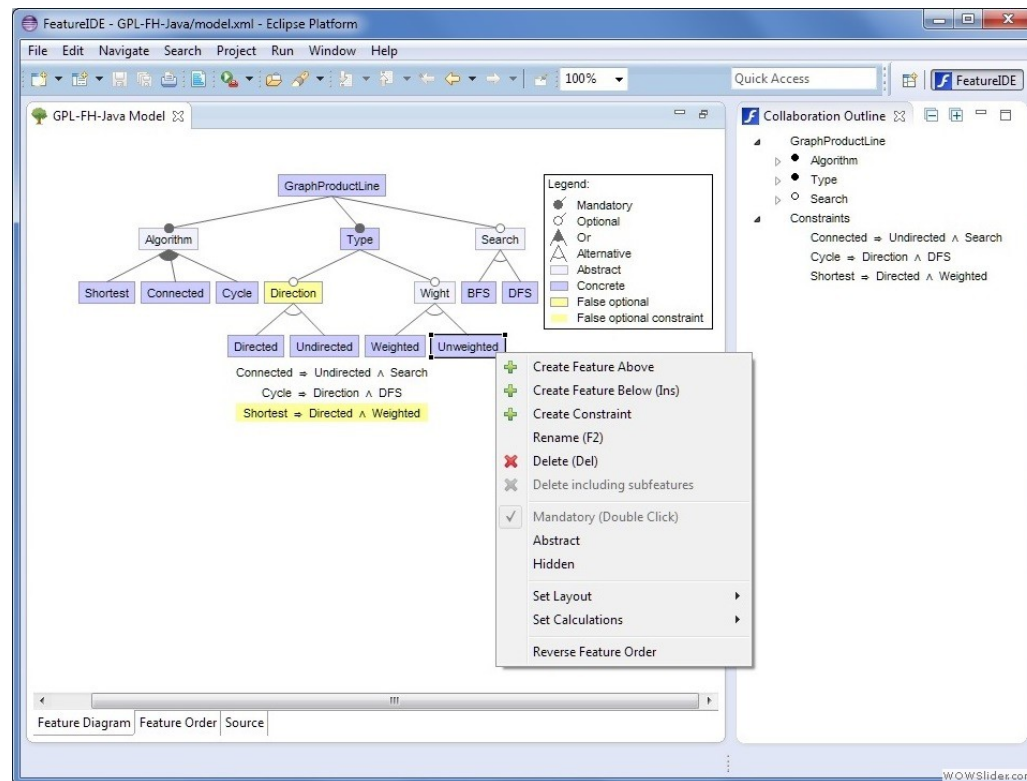


Feature IDE

- ▶ Eclipse plug-in for Feature-Oriented Software Development
- ▶ supports all phases of FOSD
 - domain analysis, domain implementation, requirements analysis, and software generation
- ▶ implementation techniques integrated
 - feature-oriented programming (FOP), aspect-oriented programming (AOP), delta-oriented programming (DOP), and preprocessors
- ▶ provides tool support for AHEAD, FeatureC++, FeatureHouse, AspectJ, DeltaJ, Munge, and Antenna

Tool Demo of the Feature IDE

- ▶ You will get a demo during the invited talk by Dipl.-Ing. Daniela Rabiser



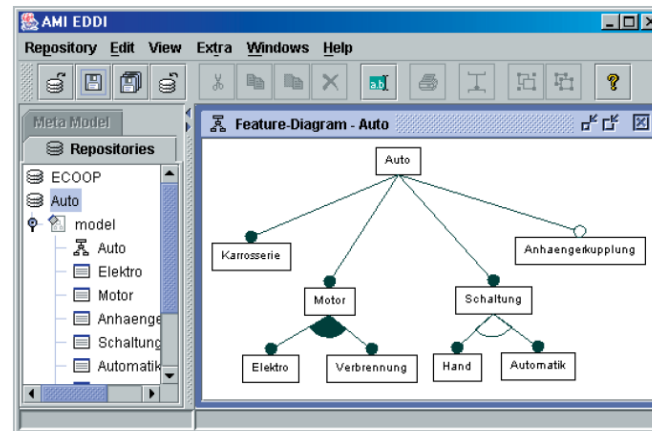
Other SPL Tools

(sorted alphabetically; only a selection, there are more!)

- **AmiEddi/Captain Feature**
- **COVAMOF-VS**
- **Feature Modeling Plug-in (FMP)**
- **FeatureIDE**
- **V-Manage**
- **VARMOD**

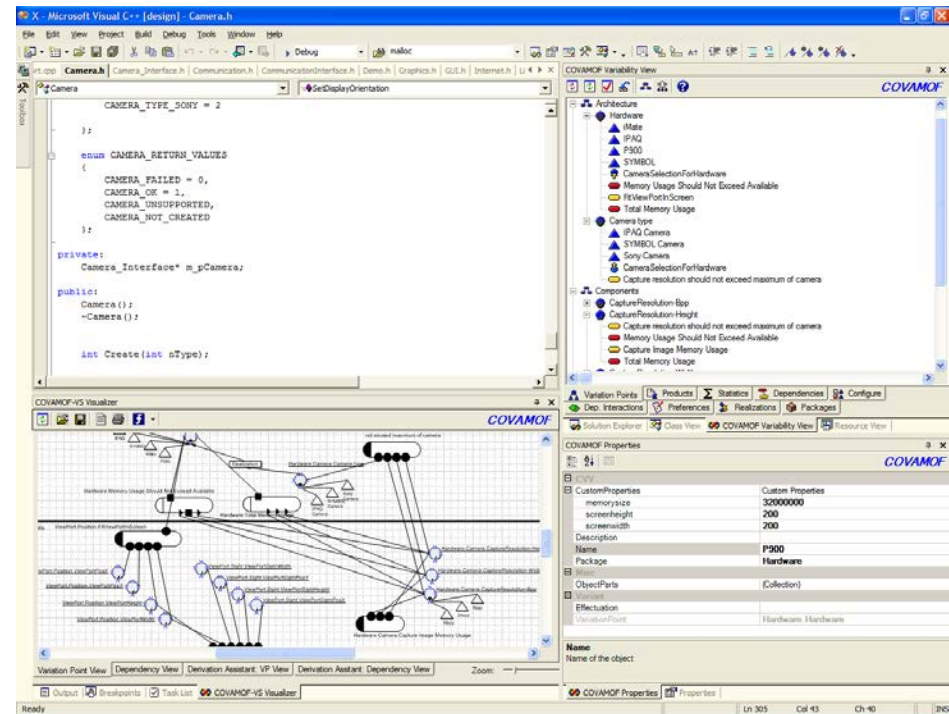
AmiEddi/Captain Feature

- ▶ Developed at Fachhochschule Kaiserslautern
- ▶ Implemented in Java/Swing
- ▶ Supporting the feature modeling notation by Czarnecki and Eisenecker
- ▶ Successor of AmiEddi and implements a cardinality-based notation
- ▶ <https://sourceforge.net/projects/captainfeature/>
- ▶ *Last Update: 2013-03-22*



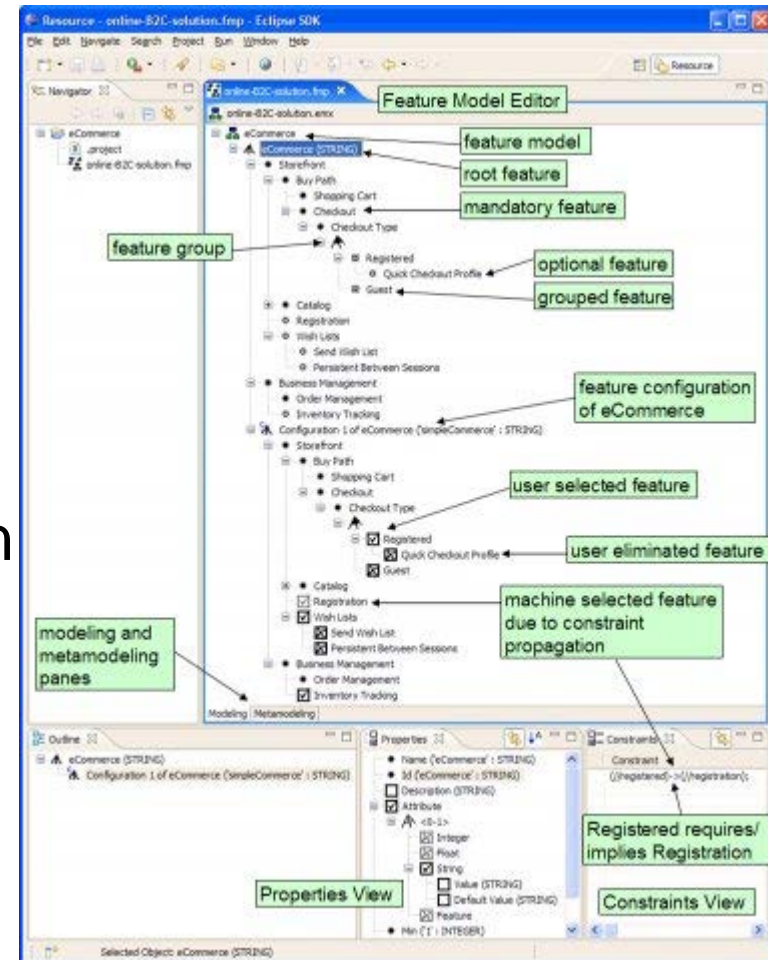
COVAMOF-VS

- ▶ Developed at University of Groningen
- ▶ Implemented in Microsoft Visual Studio .NET 2003/.NET Framework 1.1/C#
- ▶ Supporting the COVAMOF Methodology/Variability Modeling framework
- ▶ Variation points and dependencies are modeled over different abstraction levels
- ▶ e.g., XML-based feature models, parameter settings, C++/C# source files
- ▶ No longer available
- ▶ Part of a book:
<http://dx.doi.org/10.1007/978-3-642-36583-6>



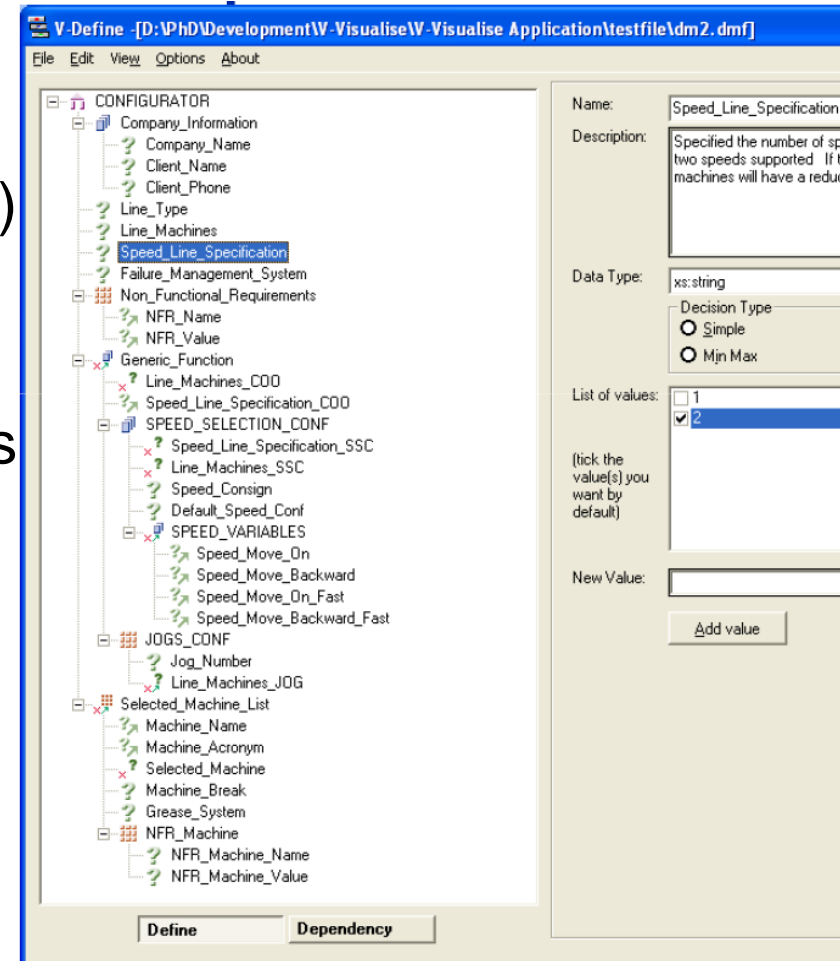
Feature Modeling Plug-in (FMP)

- ▶ Developed at University of Waterloo
- ▶ Implemented in Java/Eclipse/SWT
- ▶ Supports cardinality-based feature modeling as proposed by Czarnecki et al. and configuration based on feature diagrams
- ▶ <https://sourceforge.net/projects/fmp>
- ▶ *Last Update: 2013-04-03*



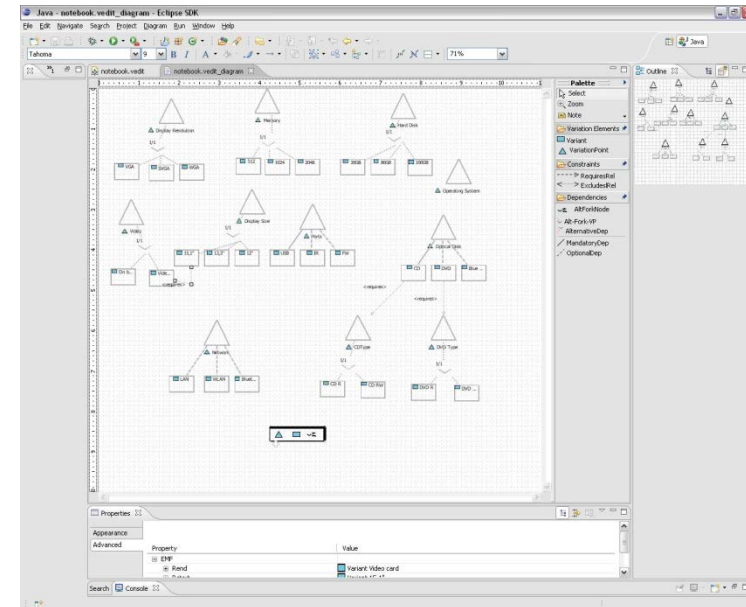
V-Manage Toolset

- ▶ Developed at the European Software Institute (ESI)
- ▶ Implemented in Visual Basic(.NET)
- ▶ Decision-oriented modeling approach
- ▶ V-Define to create decision models
- ▶ V-Implement to define variation points in components/component descriptions
- ▶ V-Resolve to take decisions
- ▶ *No longer active/available*



VARMOD

- ▶ Developed at University of Duisburg Essen (Software Systems Engineering Institute)
- ▶ Implemented in Java/Eclipse
- ▶ Supports Orthogonal Variability Modeling as proposed by Pohl et al.
- ▶ Replaced by <https://sse.uni-due.de/remidemmi> in 2014
- ▶ *Last update: 2014-06-18*



Further tools

- ▶ Also check:
 - <http://www.splot-research.org/> -- live demo now
 - <http://fisd.de/>

Conclusion

- ▶ SPL tool support is an absolute necessity
- ▶ Several SPL tools already exist
 - Few commercial and many research tools
 - All have their benefits and drawbacks
- ▶ Most tools are not stable/adaptable/interoperable/scalable enough to be usable in industry/practice
- ▶ Most tools focus on product line modeling and do not provide (or provide only little) support for the product derivation/application engineering process
- ▶ Empirical data on the usefulness of existing tools is rare
 - You have to try out every tool to find out if it is really useful for your purpose/domain

Next Week (4.5.)

- ▶ No Lecture (Landespatron)
- ▶ Next lecture: 11.5. PL Testing (RL)

Used/Useful References

- [Bruckhaus et al. 1996]
T. Bruckhaus, N. H. Madhavji, I. Janssen, and J. Henshaw, "The Impact of Tools on Software Productivity," *IEEE Software*, vol. 13, pp. 29-38, 1996.
- [Nielsen 1994]
J. Nielsen, *Usability Engineering*: Morgan Kaufmann, 1994.
- [Dhungana et al. 2011]
D. Dhungana, P. Grünbacher, and R. Rabiser, "The DOPLER Meta-Tool for Decision-Oriented Variability Modeling: A Multiple Case Study," *Automated Software Engineering*, vol. 18(1), pp. 77-114, 2011.
- [Rabiser et al. 2007]
Rabiser, R., Grünbacher, P., Dhungana, D.: Supporting Product Derivation by Adapting and Augmenting Variability Models, Proc. of the 11th International Software Product Line Conference (SPLC 2007), Kyoto, Japan, 2007.
- [Myllärniemi 2007]
V. Myllärniemi, M. Raatikainen, and T. Männistö, "Kumbang Tools," 11th International Software Product Line Conference (SPLC 2007), Tool Demonstration, Kyoto, Japan, 2007.
- [pure systems GmbH]
pure systems GmbH, "Variant Management with pure::variants, Technical Whitepaper," <http://www.pure-systems.com/fileadmin/downloads/pv-whitepaper-en-04.pdf>, 2008.
<http://www.pure-systems.com/fileadmin/downloads/pure-variants/doc/pv-user-manual.pdf>
- [Krueger 2007]
C. Krueger, "BigLever software gears and the 3-tiered SPL methodology," Proc. of the Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA'07), Montreal, Quebec, Canada, ACM, 2007, pp. 844-845.