17-708 SOFTWARE PRODUCT LINES: CONCEPTS AND IMPLEMENTATION

FEATURE AND DECISION MODELING

CHRISTIAN KAESTNER
CARNEGIE MELLON UNIVERSITY
INSTITUTE OF SOFTWARE RESEARCH

LEARNING GOALS

define the relevant terms: product line, feature, concern, option, feature selection, feature dependency, product, domain, variant

understand why a product line targets a specific domain,

model features and feature dependencies by means of feature models,

tradeoffs among representations

WHAT IS A FEATURE?

Feature

Concern

Configuration Option

Configuration

Configuration Space

Constraint

Variant

Product

IN-CLASS EXERCISE

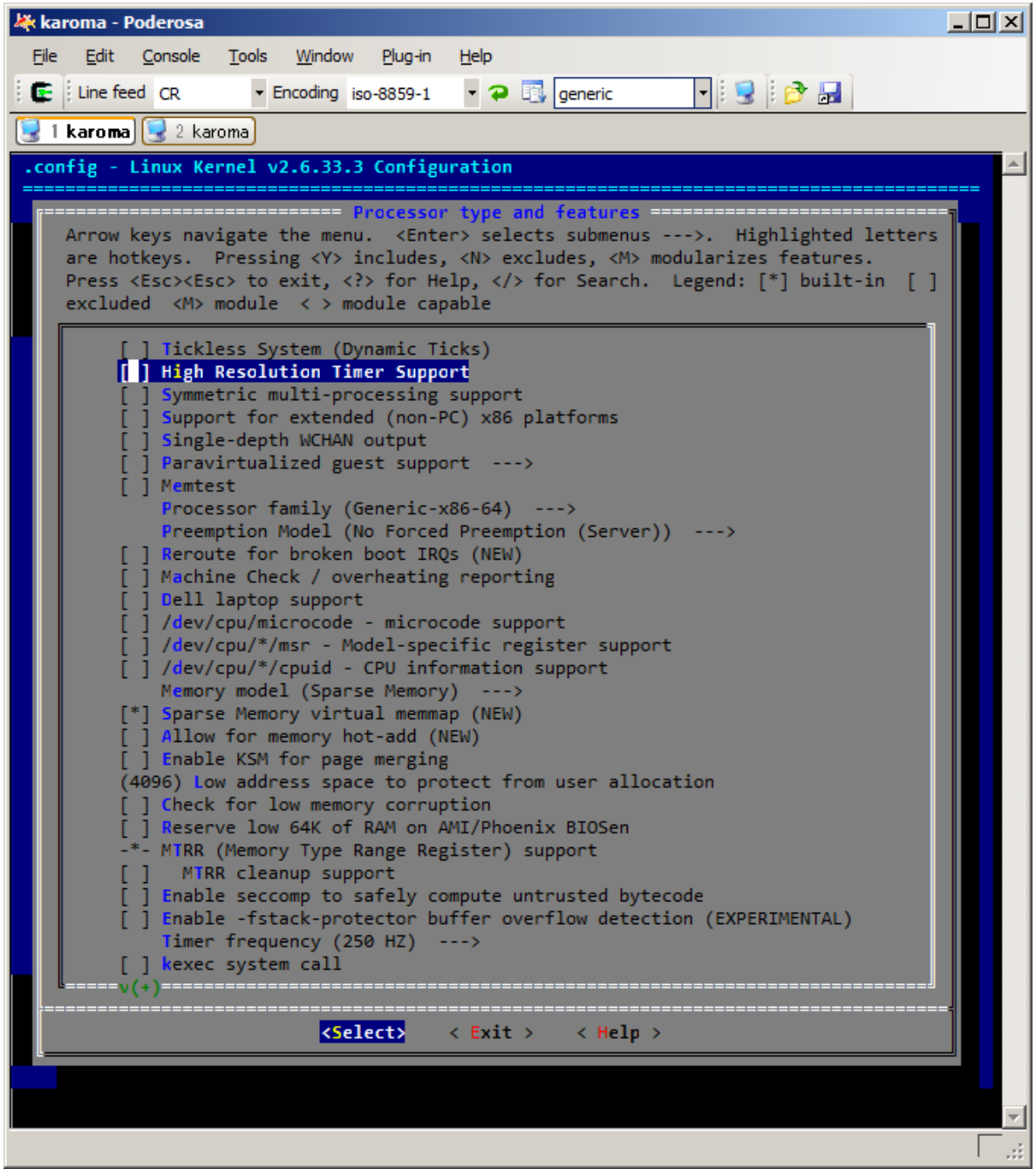
List 10 features in domain X

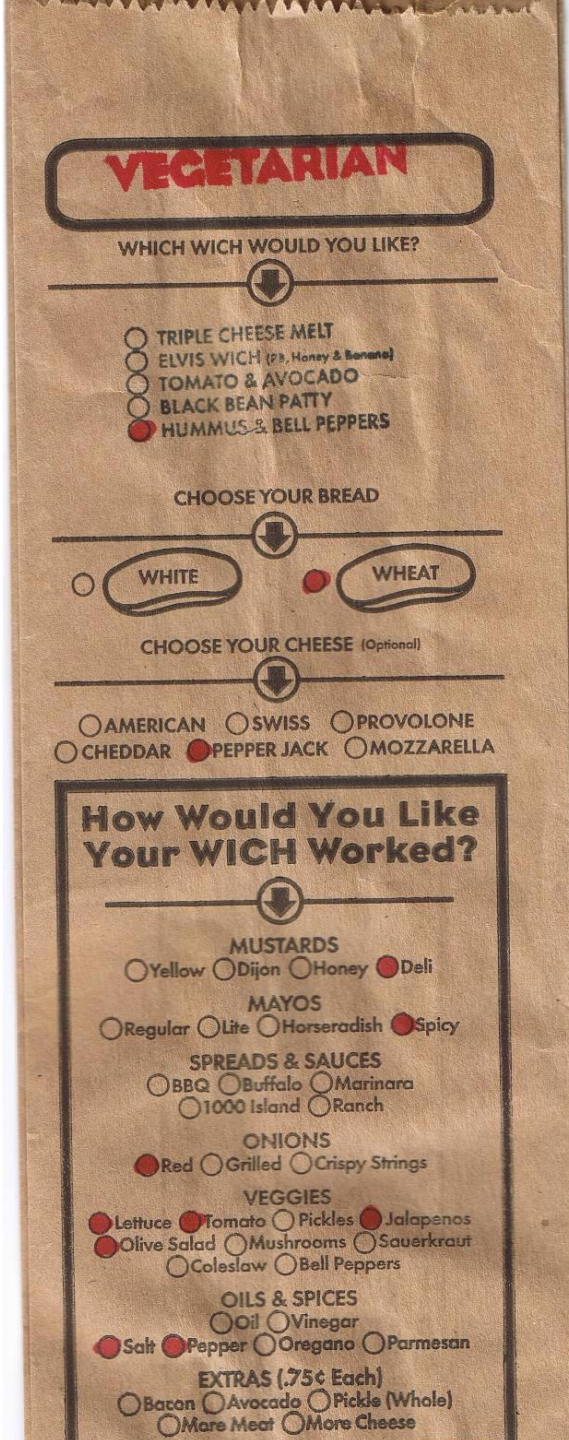
SOFTWARE PRODUCT LINES

A software product line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.

Software Engineering Institute
Carnegie Mellon University

WHAT IS A DOMAIN?


karoma - Poderosa
File Edit Console Tools Window Plug-in Help
Line feed CR Encoding iso-8859-1 generic
1 karoma 2 karoma
.config - Linux Kernel v2.6.33.3 Configuration
Processor type and features
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
[ ] Tickless System (Dynamic Ticks)
[ ] High Resolution Timer Support
[ ] Symmetric multi-processing support
[ ] Support for extended (non-PC) x86 platforms
[ ] Single-depth WCHAN output
[ ] Paravirtualized guest support --->
[ ] Memtest
Processor family (Generic-x86-64) --->
Preemption Model (No Forced Preemption (Server)) --->
[ ] Reroute for broken boot IRQs (NEW)
[ ] Machine Check / overheating reporting
[ ] Dell laptop support
[ ] /dev/cpu/microcode - microcode support
[ ] /dev/cpu/*/msr - Model-specific register support
[ ] /dev/cpu/*/cpuid - CPU information support
Memory model (Sparse Memory) --->
[*] Sparse Memory virtual memmap (NEW)
[ ] Allow for memory hot-add (NEW)
[ ] Enable KSM for page merging
(4096) Low address space to protect from user allocation
[ ] Check for low memory corruption
[ ] Reserve low 64K of RAM on AMI/Phoenix BIOSen
-*- MTRR (Memory Type Range Register) support
[ ] MTRR cleanup support
[ ] Enable seccomp to safely compute untrusted bytecode
[ ] Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)
Timer frequency (250 HZ) --->
[ ] kexec system call
v(+)
<Select> < Exit > < Help >


VEGETARIAN
WHICH WICH WOULD YOU LIKE?
TRIPLE CHEESE MELT
ELVIS WICH (PB, Honey & Banana)
TOMATO & AVOCADO
BLACK BEAN PATTY
HUMMUS & BELL PEPPERS
CHOOSE YOUR BREAD
WHITE
WHEAT
CHOOSE YOUR CHEESE (Optional)
AMERICAN SWISS PROVOLONE
CHEDDAR PEPPER JACK MOZZARELLA
How Would You Like Your WICH Worked?
MUSTARDS
Yellow Dijon Honey Deli
MAYOS
Regular Lite Horseradish Spicy
SPREADS & SAUCES
BBQ Buffalo Marinara
1000 Island Ranch
ONIONS
Red Grilled Crispy Strings
VEGGIES
Lettuce Tomato Pickles Jalapenos
Olive Salad Mushrooms Sauerkraut
Coleslaw Bell Peppers
OILS & SPICES
Oil Vinegar
Salt Pepper Oregano Parmesan
EXTRAS (.75¢ Each)
Bacon Avocado Pickle (Whole)
More Meat More Cheese

DESCRIBING CONFIGURATION SPACES

List of configurations

List of options and textual constraints

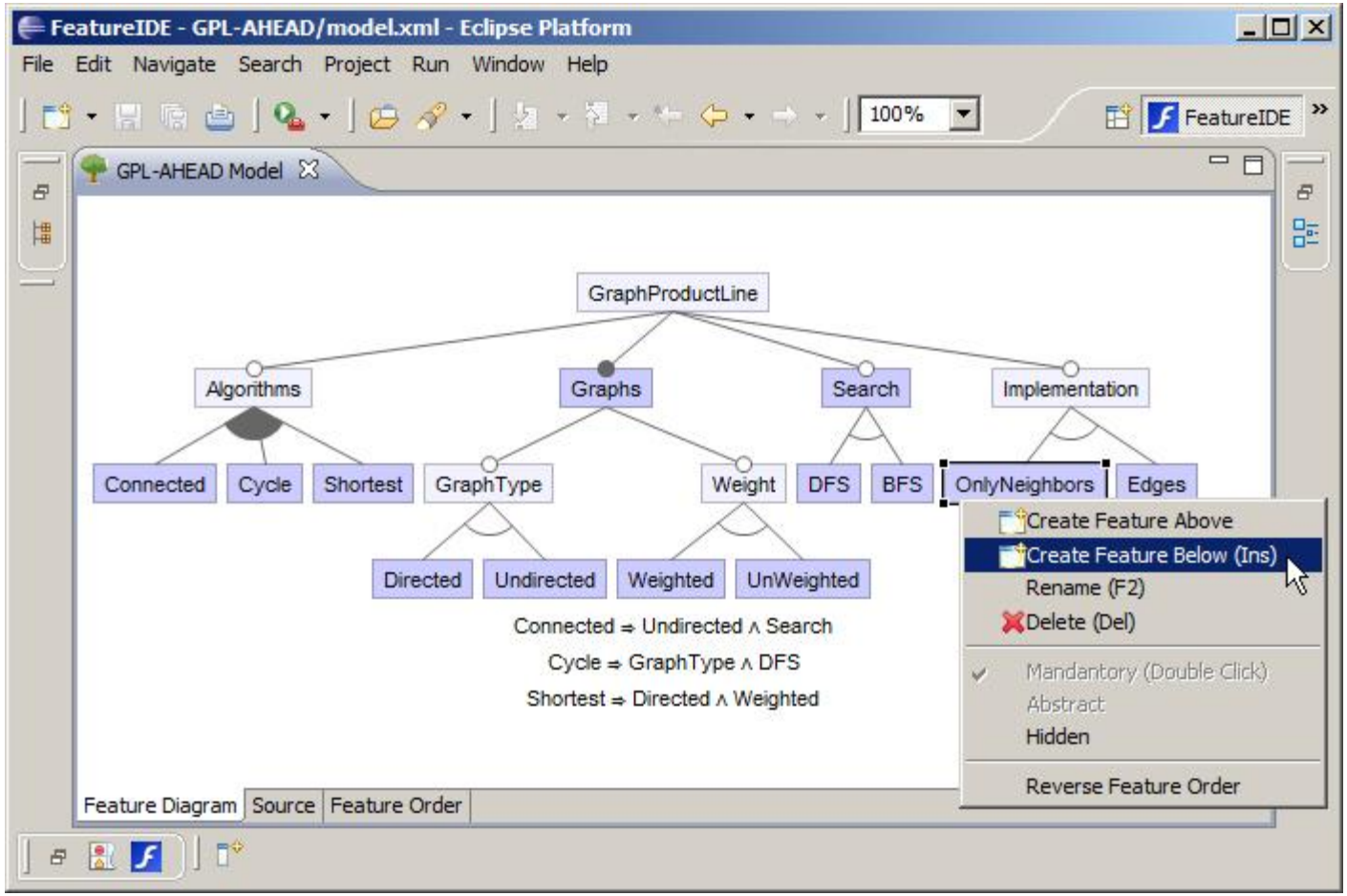
Formula

Feature models

Decision models

Tradeoffs

TOOL DEMO: FEATUREIDE



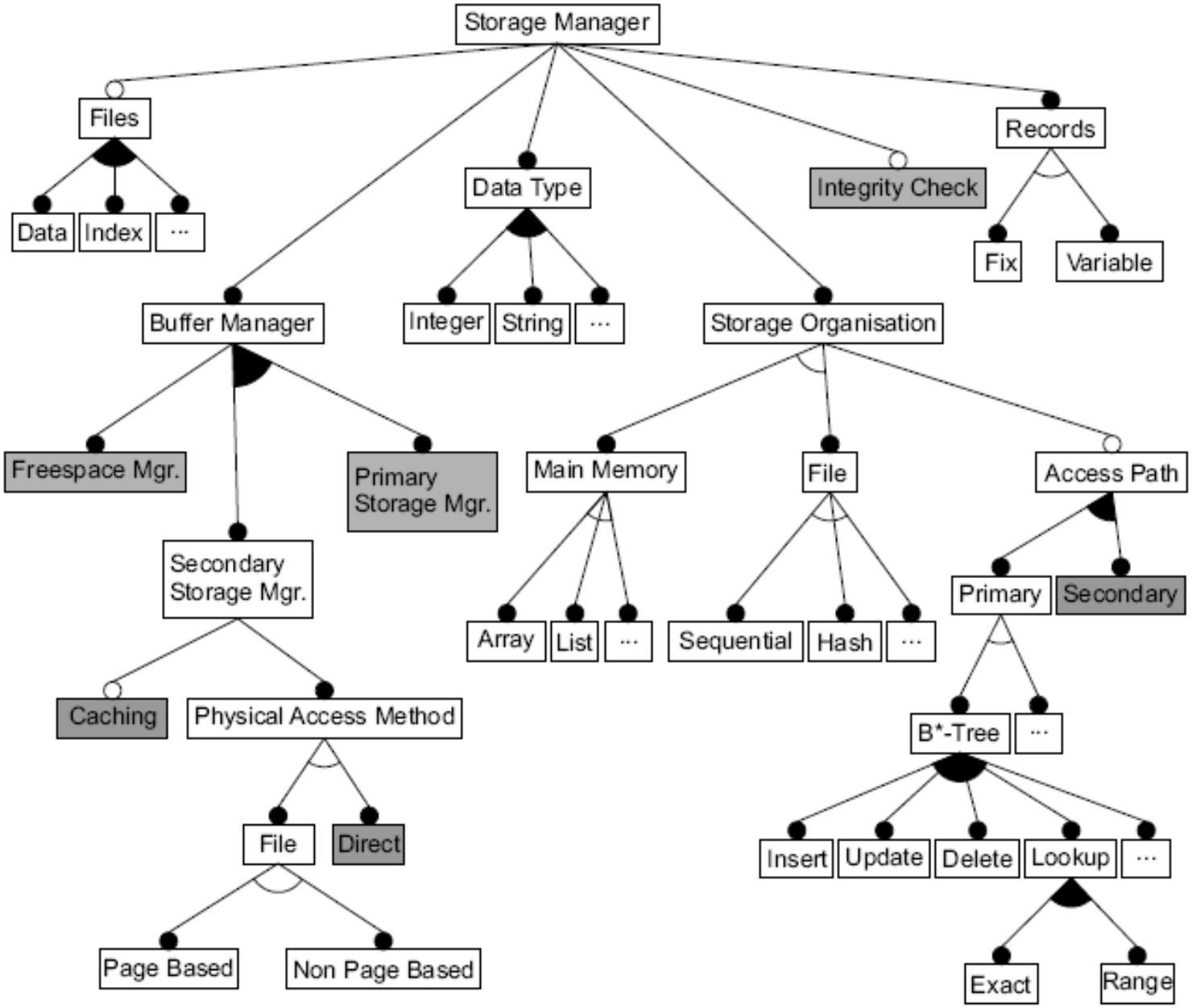
IN-CLASS EXERCISE

Organize your features and identify constraints

Create a feature model

Create a corresponding formula

Estimate the number of configurations


Storage Manager
Files
Data
Index
···
Data Type
Integer
String
···
Integrity Check
Records
Fix
Variable
Buffer Manager
Freespace Mgr.
Secondary Storage Mgr.
Primary Storage Mgr.
Caching
Physical Access Method
File
Direct
Page Based
Non Page Based
Storage Organisation
Main Memory
Array
List
···
File
Sequential
Hash
···
Access Path
Primary
Secondary
B*-Tree
···
Insert
Update
Delete
Lookup
···
Exact
Range

DOCUMENTING FEATURES

Description of a feature and its corresponding (set of) requirements

Relationship to other features, especially hierarchy, order, and grouping

External dependencies, such as required hardware resources

Interested stakeholders

Estimated or measured cost of realizing a feature

Potentially interested customers and estimated revenue

Configuration knowledge, such as ‘activated by default’

Configuration questions asked during the requirements analysis step

Constraints, such as “requires feature X and excludes feature Y”

All kinds of behavioral specifications, including invariants and pre- and post-conditions

Known effects on non-functional properties, such as “improves performance and increases energy consumption”

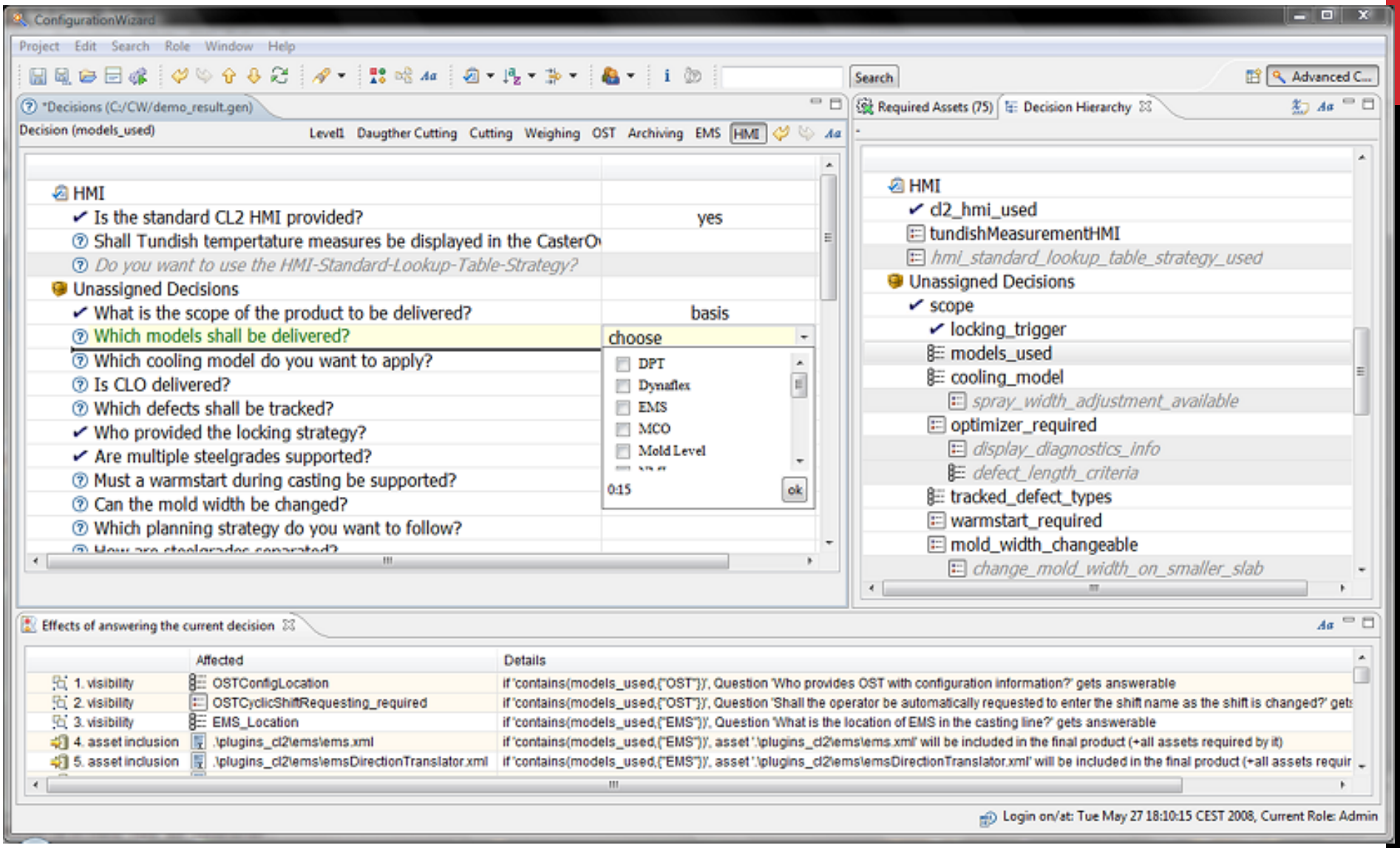
Rationale for including a feature in the scope of the product line

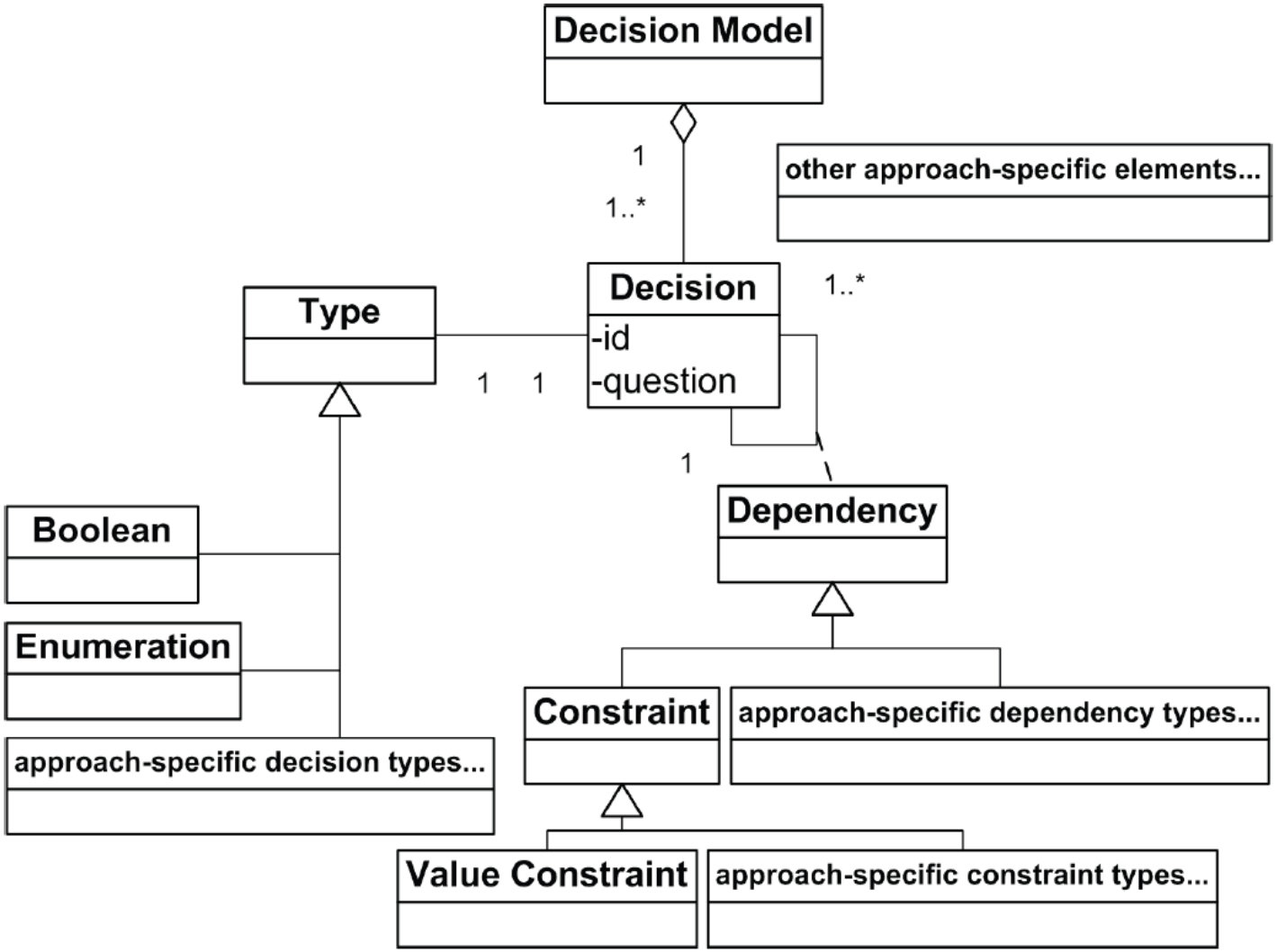
Additional attributes, such as numbers and textual parameters, used for further customization during product generation

Potential feature interactions

CASE STUDIES: KCONFIG, PURE::VARIANTS

DECISION MODELING


ConfigurationWizard
Project Edit Search Role Window Help
Search
Advanced C...
*Decisions (C:/CW/demo_result.gen)
Decision (models_used)
Level1 Daugther Cutting Cutting Weighing OST Archiving EMS HMI
HMI
Is the standard CL2 HMI provided?
yes
Shall Tundish tempertature measures be displayed in the CasterO
Do you want to use the HMI-Standard-Lookup-Table-Strategy?
Unassigned Decisions
What is the scope of the product to be delivered?
basis
Which models shall be delivered?
choose
Which cooling model do you want to apply?
Is CLO delivered?
Which defects shall be tracked?
Who provided the locking strategy?
Are multiple steelgrades supported?
Must a warmstart during casting be supported?
Can the mold width be changed?
Which planning strategy do you want to follow?
DPT
Dynaflex
EMS
MCO
Mold Level
0:15
ok
Required Assets (75)
Decision Hierarchy
HMI
cl2_hmi_used
tundishMeasurementHMI
hmi_standard_lookup_table_strategy_used
Unassigned Decisions
scope
locking_trigger
models_used
cooling_model
spray_width_adjustment_available
optimizer_required
display_diagnostics_info
defect_length_criteria
tracked_defect_types
warmstart_required
mold_width_changeable
change_mold_width_on_smaller_slab
Effects of answering the current decision
Affected
Details
1. visibility
OSTConfigLocation
if 'contains(models_used,{"OST"})', Question 'Who provides OST with configuration information?' gets answerable
2. visibility
OSTCyclicShiftRequesting_required
if 'contains(models_used,{"OST"})', Question 'Shall the operator be automatically requested to enter the shift name as the shift is changed?' gets
3. visibility
EMS_Location
if 'contains(models_used,{"EMS"})', Question 'What is the location of EMS in the casting line?' gets answerable
4. asset inclusion
.\plugins_cl2\ems\ems.xml
if 'contains(models_used,{"EMS"})', asset '.\plugins_cl2\ems\ems.xml' will be included in the final product (+all assets required by it)
5. asset inclusion
.\plugins_cl2\ems\emsDirectionTranslator.xml
if 'contains(models_used,{"EMS"})', asset '.\plugins_cl2\ems\emsDirectionTranslator.xml' will be included in the final product (+all assets requir
Login on/at: Tue May 27 18:10:15 CEST 2008, Current Role: Admin



source: Schmid, K., Rabiser, R., & Grünbacher, P. (2011, January). A comparison of decision modeling approaches in product lines. In *Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems* (pp. 119-126). ACM.

dimension	decision modeling	feature modeling	Kconfig	CDL	CVL initial
applications	*variability modeling; derivation support*	**diverse applications: concept modeling (e.g., domain modeling)**, *variability* **and commonality** *modeling; derivation support*	modeling variability in the kernel; derivation support	modeling variability in eCos; derivation support	variability modeling; derivation support
unit of variability	**decisions to be made in derivation**	**features are properties of concepts, e.g., systems**	drivers, subsystems, kernel options, build option	drivers, subsystems, kernel options, build option	VSpecs: essentially decisions in derivation; pre-made decisions (mandatory features)
orthogonality	orthogonal	mostly used in orthogonal fashion	orthogonal (added configuration UI concepts, e.g., menus)	orthogonal (added architectural concepts, e.g., packages, components)	orthogonal (but admitting non-orthogonal uses is discussed)
data types	*comprehensive set of basic types*; *composite*: **sets, records, arrays**	*comprehensive set of basic types*; **references**; *composite*: **via hierarchy, group and feature cardinalities**	Boolean, tristate, numbers and strings; choices	none, bool, data (dynamically typed values incl. int, string, real), booldata	choices; parameters with comprehensive set of basic types; classifiers
hierarchy	**secondary concept; diverse approaches**, e.g., **visibility or relevance** hierarchy (no decomposition)	**essential concept; single approach**: tree hierarchy modeling, parent-child configuration constraints and **decomposition**	characteristics of FM&DM: essential organization means (FM), visibility induced, driven by UI concepts (DM)	like in FM (essential organization means; decomposition hierarchy)	essential concept; vspec tree, like in FM
dependencies and constraints	*no standard constraint language but similar range of approaches (Boolean, numeric, sets)*	*no standard constraint language but similar range of approaches (Boolean, numeric, sets, quantifiers)*	propositional three-valued logics with comparisons	propositional Boolean logics with expressions on data	propositional and predicate logic with expressions on data
mapping to artifacts	**essential concept**; *no standard mechanism*	**optional concept**; *no standard mechanism*	mapping to C preprocessor via a custom build system (no explicit mapping model)	explicit mapping in the variability model; variability symbols available to C preprocessor	essential concept; mapping model, base-model independent
binding time and mode	*not standardized, occasionally supported*	*not standardized, occasionally supported*	static or dynamic binding decided at compile time	static binding	not included in CVL (dependent on application)
modularity	*no standard mechanism;* **decision groups play partly this role**	*no standard mechanism;* **feature hierarchy plays partly this role**	model is split into files; no modularization beyond hierarchy in the language	loadable packages, reparenting	explicit support (packages, configurable units)
tool aspects	*representation of models as lists, tables, trees, and graphs*; **configuration UI: an (ordered) list of questions** *diverse solutions for configuration workflows* **(essential)**	*representation of models as lists, tables, trees, and graphs*; **configuration UI: usually a tree (unordered)** *diverse solutions for supporting configuration workflows* **(secondary concept)**	modeling in textual syntax; configuration UI: a t… vis… no… co… rec…	modeling in textual syntax; configuration UI:	user interfaces are the domain of vendors; basic

source: Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., & Wąsowski, A. (2012, January). Cool features and tough decisions: a comparison of variability modeling approaches. In *Proceedings of the sixth international workshop on variability modeling of software-intensive systems* (pp. 173-182). ACM.

ADOOPTION PATHS

FURTHER READING

K. Kang, S. Cohen, J. Hess, W. Novak, and A. Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, SEI,1990.

K. Czarnecki and U. Eisenecker. Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000.

Apel, S., Batory, D., Kaestner, C., & Saake, G. (2013). Feature-Oriented Software Product Lines. Berlin: Springer. Chapter 2.3

Schmid, K., Rabiser, R., & Grünbacher, P. (2011, January). A comparison of decision modeling approaches in product lines. In *Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems* (pp. 119-126). ACM.

Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., & Wąsowski, A. (2012, January). Cool features and tough decisions: a comparison of variability modeling approaches. In *Proceedings of the sixth international workshop on variability modeling of software-intensive systems* (pp. 173-182). ACM.

Krueger, C. (2002). Easing the transition to software mass customization. In *Software Product-Family Engineering* (pp. 282-293). Springer Berlin Heidelberg.