Search Based Software Engineering for Variability Management

Roberto E. Lopez-Herrejon

Systems Engineering and Automation Institute

Johannes Kepler University Linz, Austria

Project Context

- FWF Lise Meitner Fellowship
  - Named after distinguished austrian nuclear fission scientist
  - Sponsored by the Austrian Science Fund (FWF)
- Fellowship duration: 2 years
  - Start date: **August 2012**
- Personnel involved
  - Research fellow
  - Academic host: Alexander Egyed

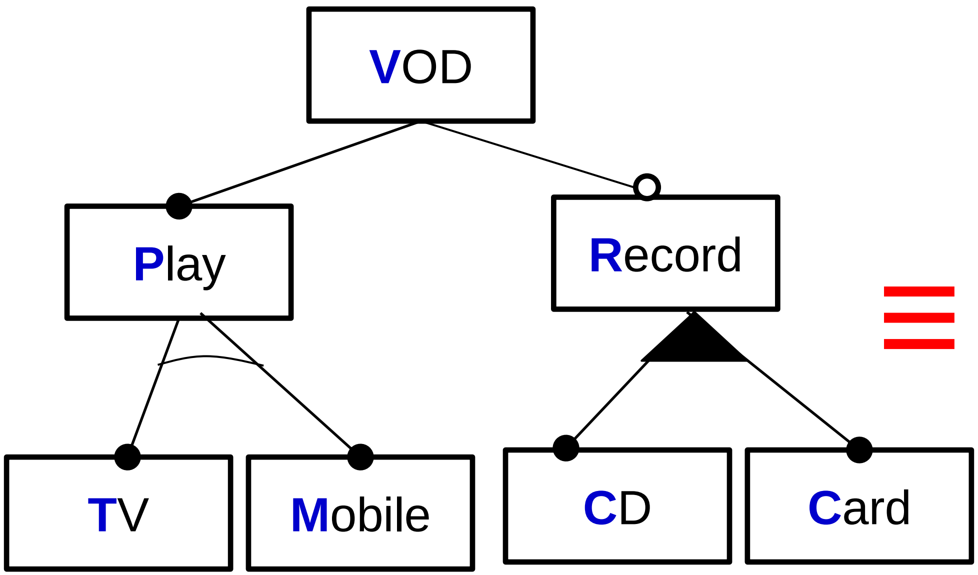
For this talk …

- Give an overview of the project
  - Goals
  - Novel contributions
- Brief description of …
  - Early results
  - Ongoing and upcoming work

Background – Feature Models

JOHANNES KEPLER UNIVERSITY LINZ | JKU

- Feature models
  - de facto standard to model variability
  - denote sets of „valid“ feature combinations



≡

	V	P	T	M	R	CD	C
P1	✓	✓	✓				
P2	✓	✓		✓			
P3	✓	✓	✓		✓	✓	
P4	✓	✓		✓	✓	✓	
P5	✓	✓	✓		✓		✓
P6	✓	✓		✓	✓		✓
P7	✓	✓	✓		✓	✓	✓
P8	✓	✓		✓	✓	✓	✓

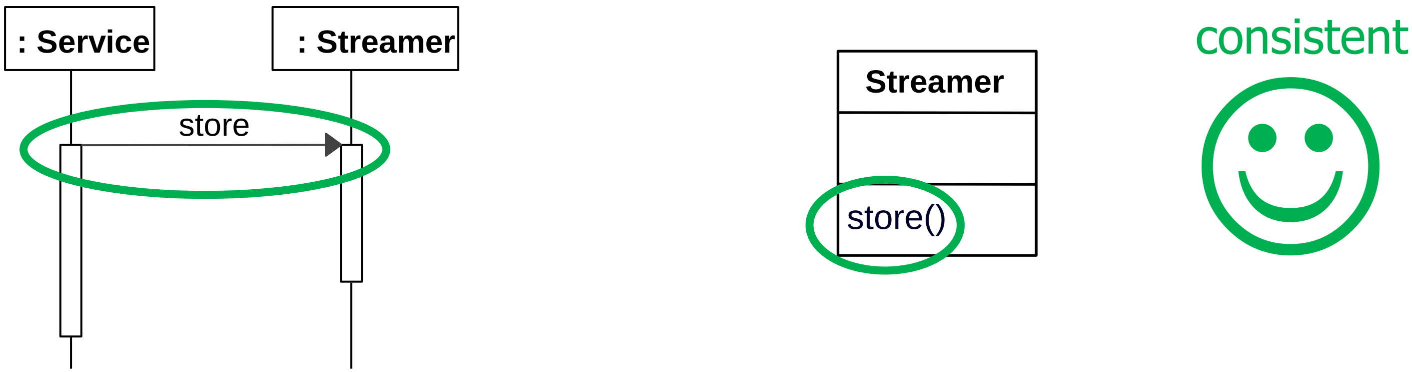
Background – Software Evolution

- Reverse Engineering
  - Process of analyzing a software system to identify its components and their relationships with the goal of creating a higher level abstraction of them

- Software Evolution
  - Process of progressive changes to the software artifacts or their properties

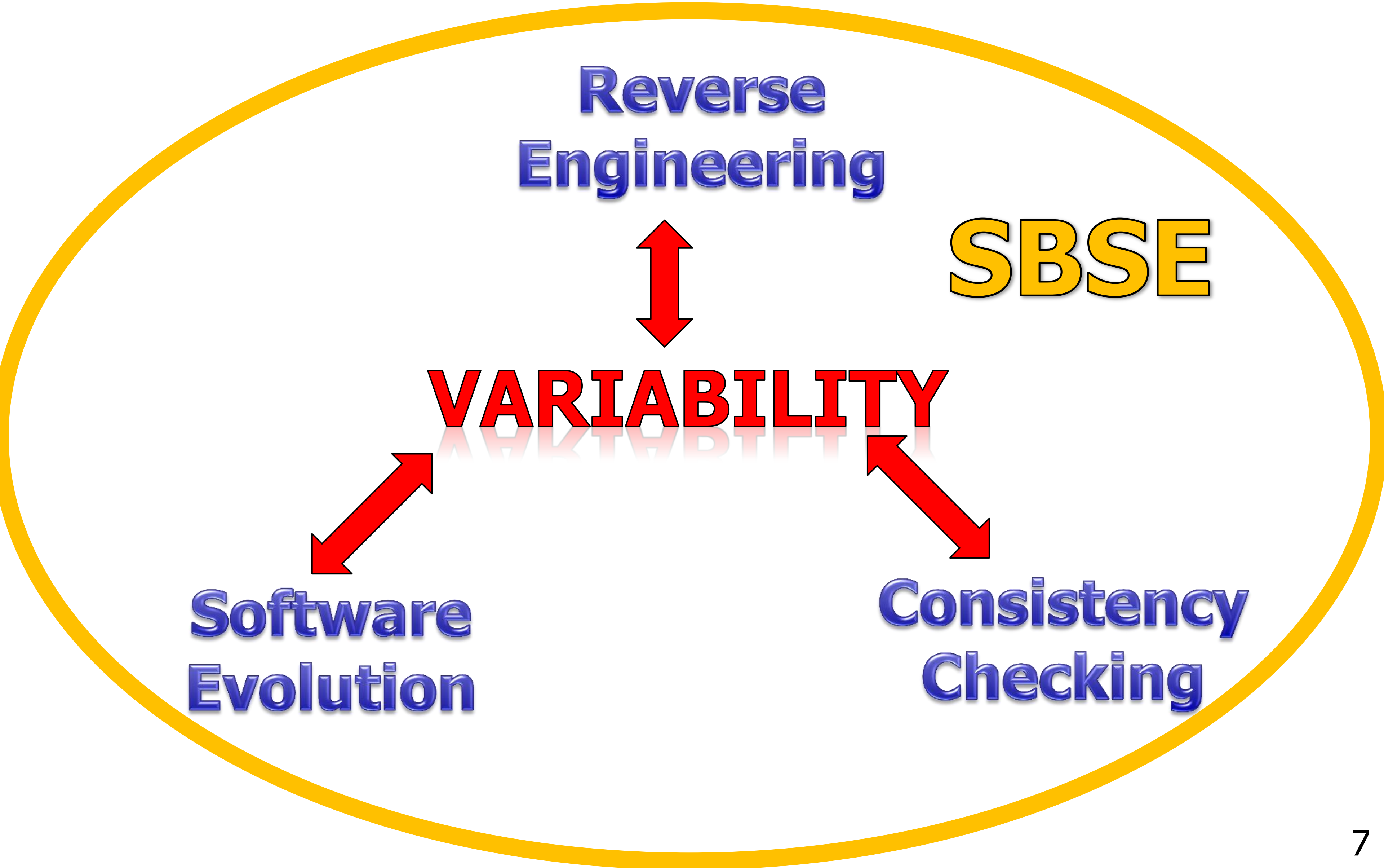
Background – Consistency Checking

- Consistency checking
  - Verifies that artifacts adhere to ***consistency rules*** that describe the semantic relationships among elements
- Example. UML consistency rule
  - Message action must be defined as an operation in receiver's class.



Project Big Picture

JOHANNES KEPLER UNIVERSITY LINZ | JKU



Relation with workshop …?

JOHANNES KEPLER
UNIVERSITY LINZ | JKU

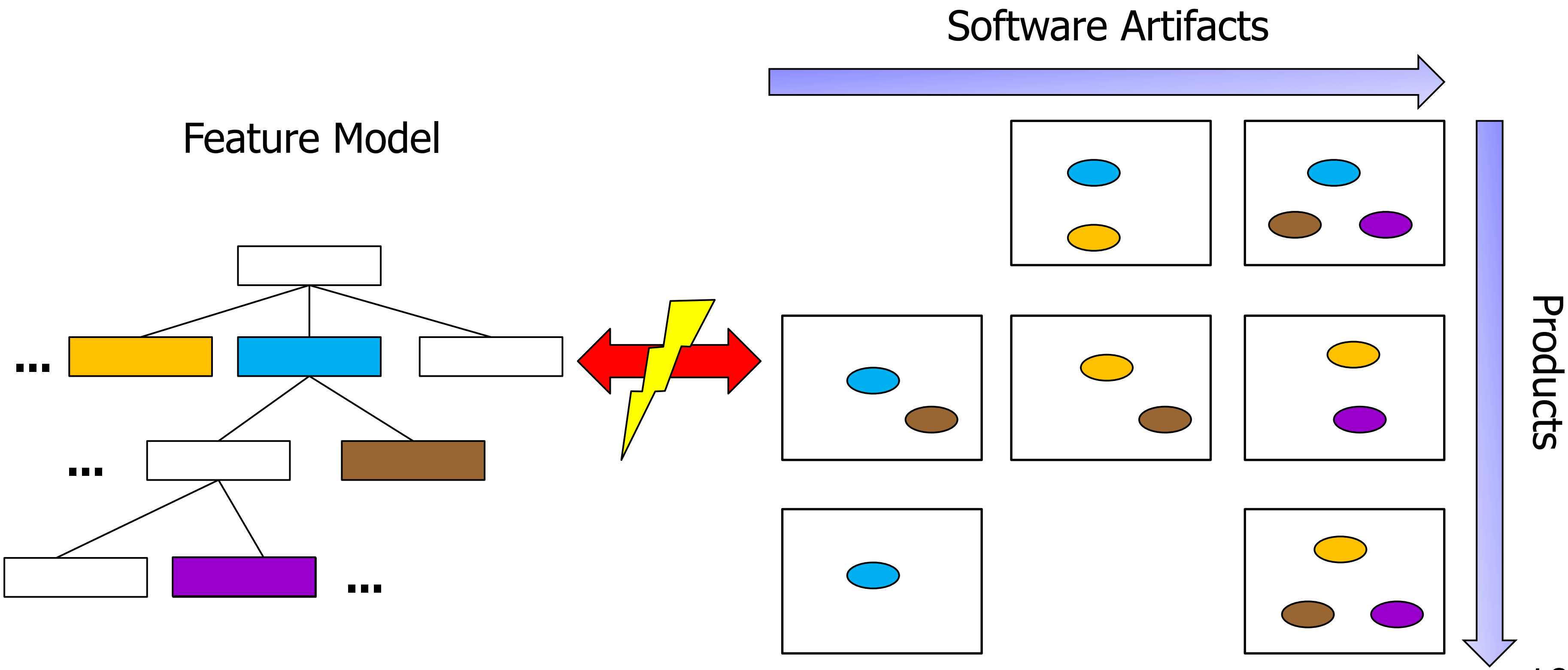
Change Impact Analysis ⇒ Reverse Engineering

⇒ Software Evolution

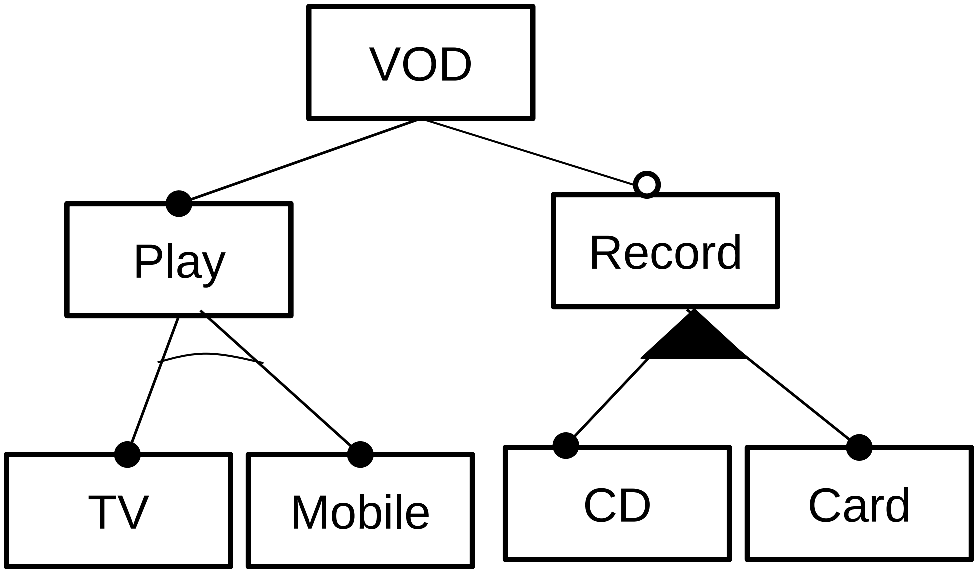
Testing ⇒ Consistency Checking

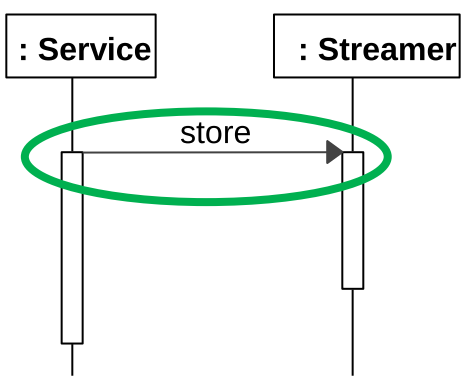
Problems Addressed

Fixing inconsistencies in the presence of variability

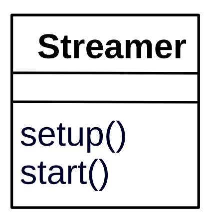


Problem 1 – Example





Feature CD



Feature Record

How can this instance be fixed ?

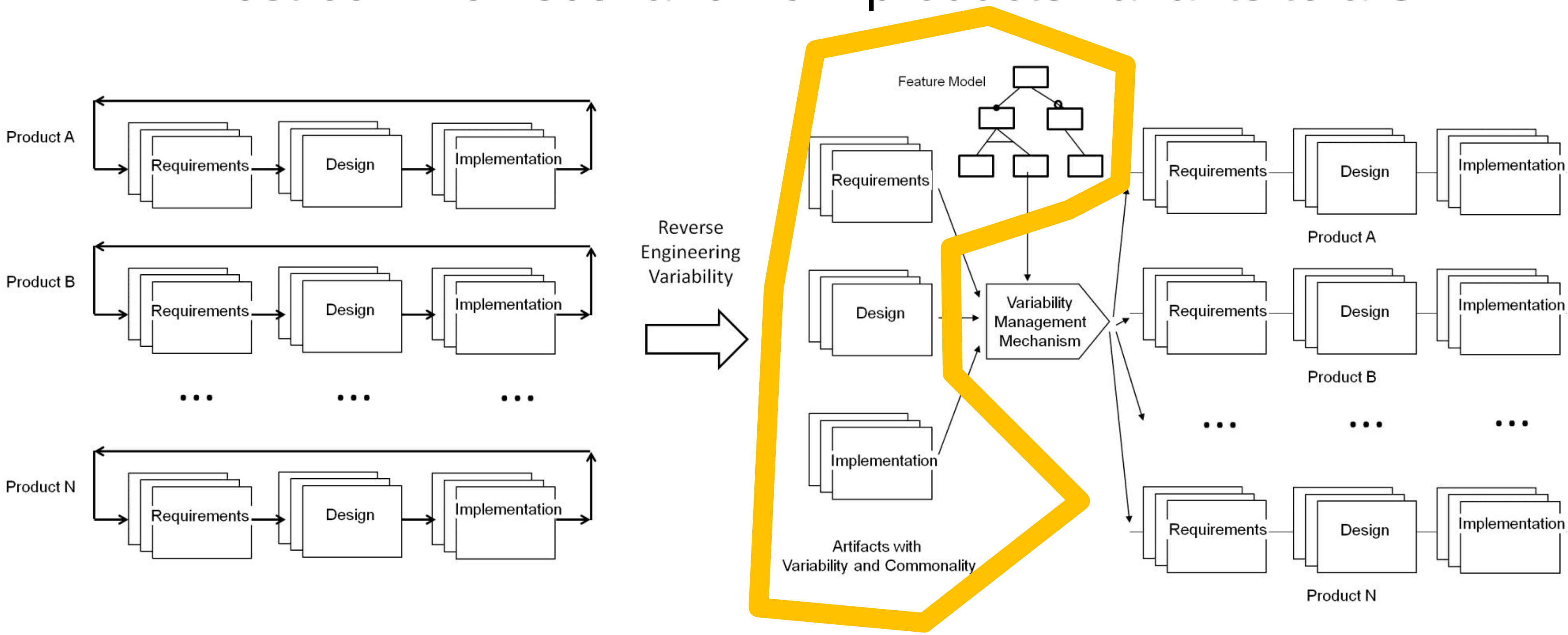
- Define store in CD
- Define store in R
- Define store in
- Define store i
- Define stor

What if …

- Incorr ame – start ✓
- Inc target
- Incor e ascription
- Instances overlap
- …

Problem 2

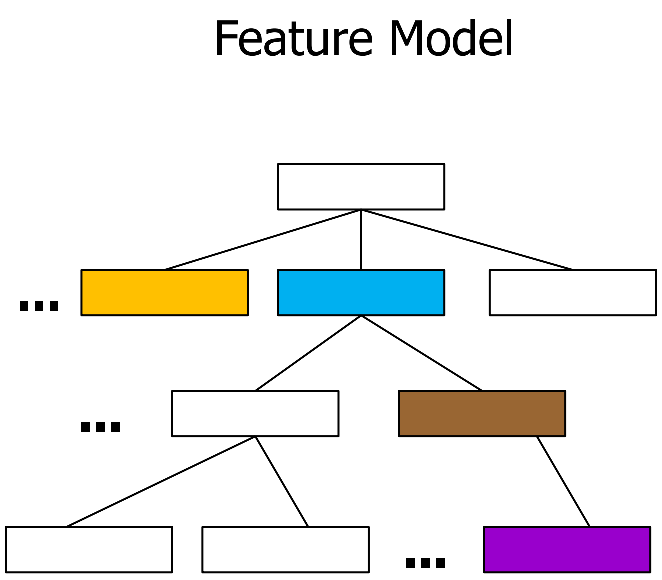
- Reverse Engineering of Variability
  - Most common scenario from products variants to a SPL

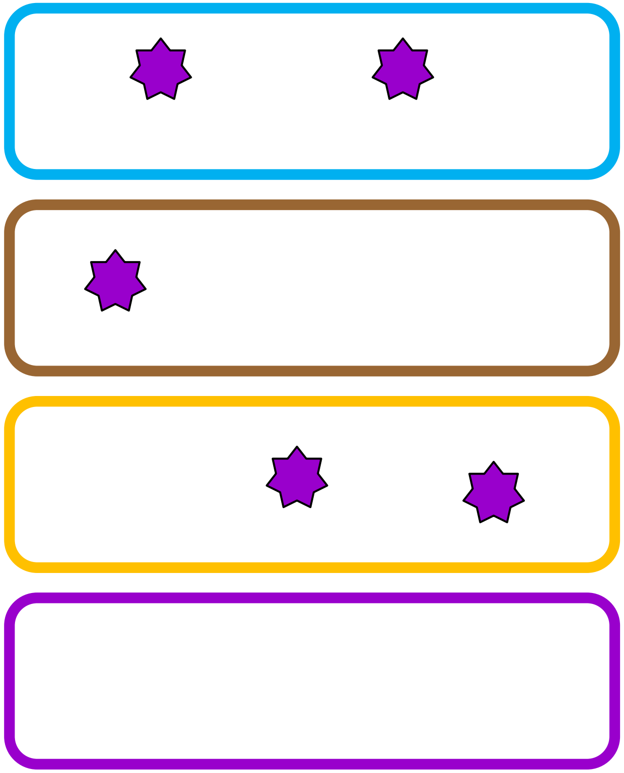


Search Based Variability Mining

Problem 3

- Variability Evolution
  - Adding new features, new members of the product family, modifications





Search Based Feature Oriented Refactoring

Early Results

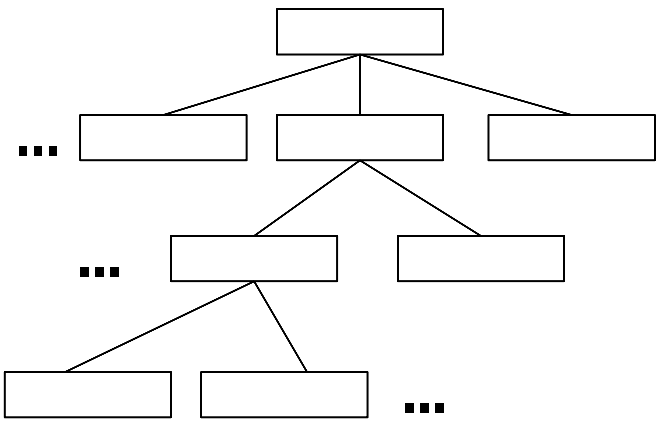
Reverse Engineering Feature Models from Product Configurations (SBSE 2012)

Collaboration with University of Seville

David Benavides, Jose Galindo, Sergio Segura, Jose Parejo

Problem Pictorial View

Feature Model



Reverse Engineering

non trivial

error prone

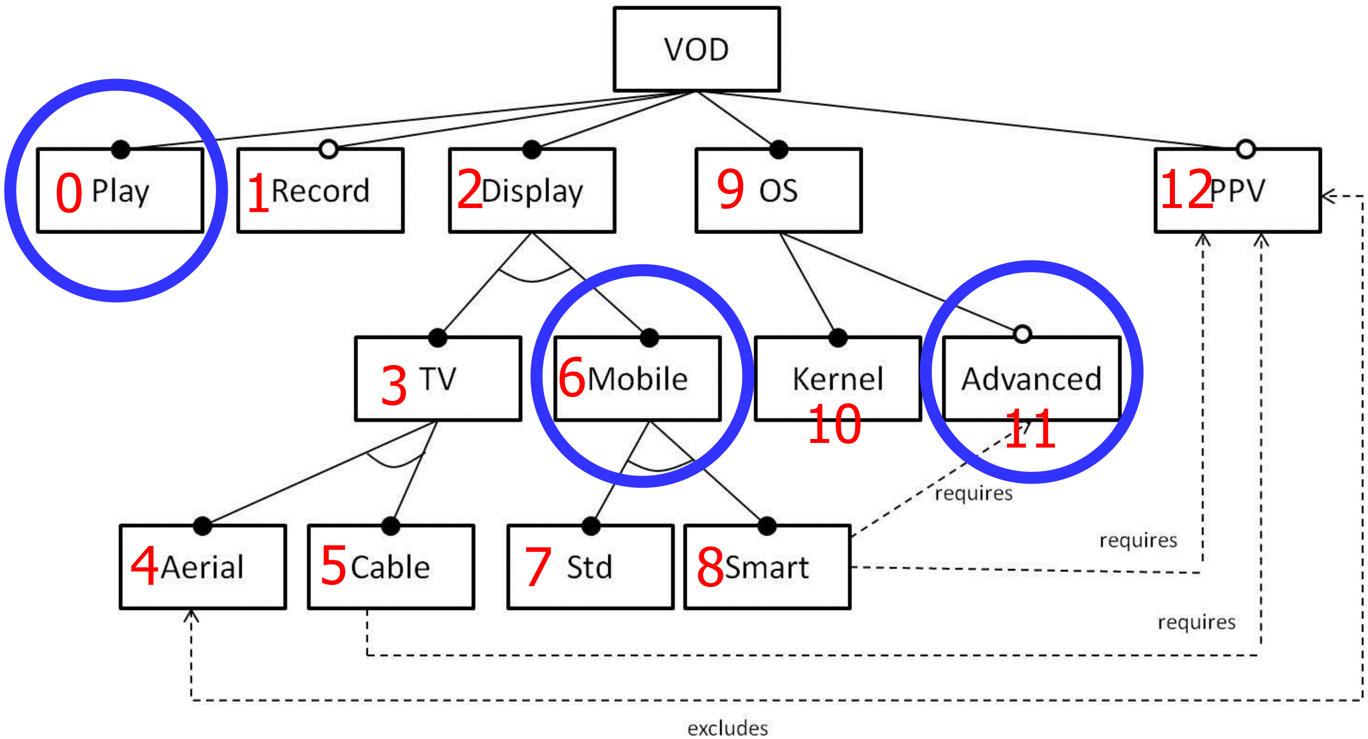
non unique

non optimal

Feature Sets

A	B	C	...	N
✓		✓	...	
	✓		...	✓
✓	✓	✓	...	
		✓	...	✓
	✓	✓	...	✓
✓	✓	✓	...	
...	...	...	...	...
✓	✓	✓	...	✓

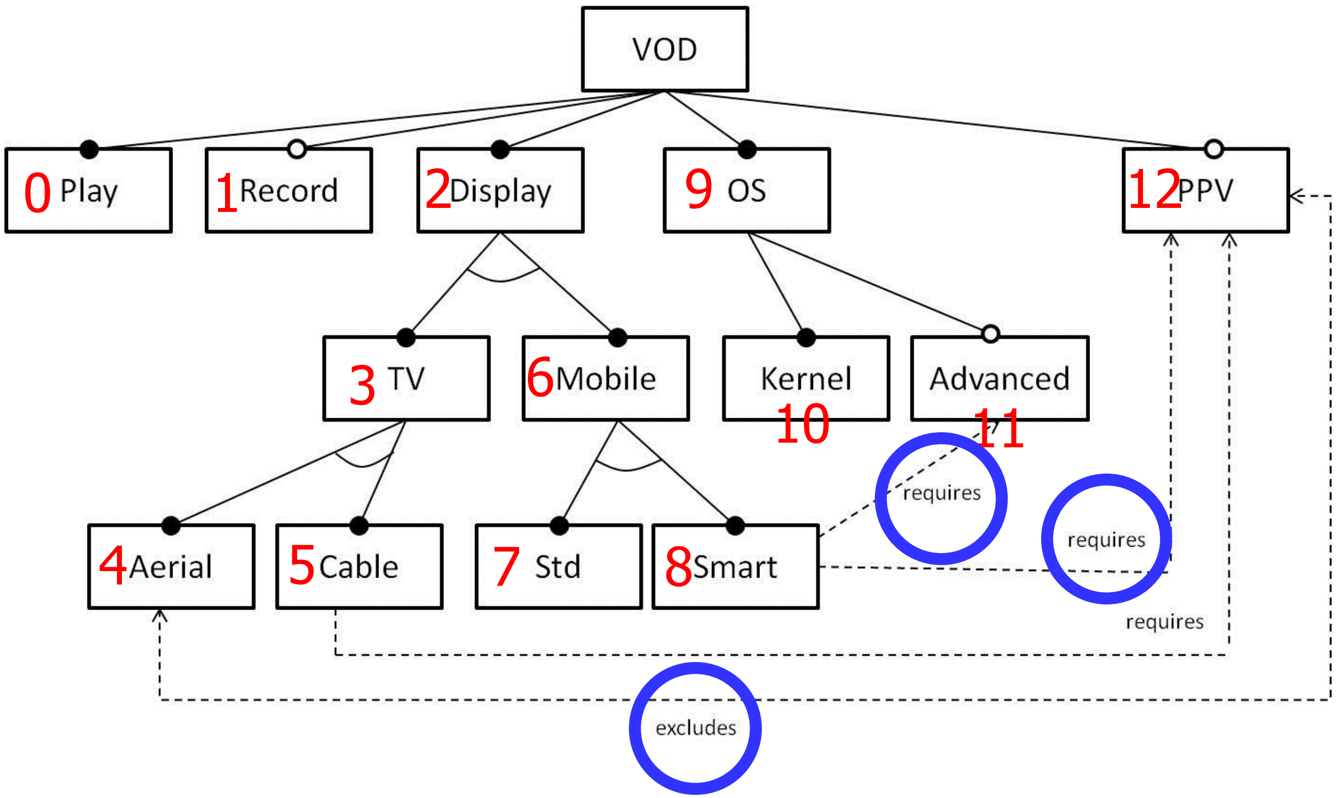
ETHOM
Structural Encoding Example



0	1	2	3	4	5	6	7	8	9	10	11	12
M,0	Op,0	M,2	Alt,2	Alt,0	Alt,0	Alt,2	Alt,0	Alt,0	M,2	M,0	Op,0	Op,0

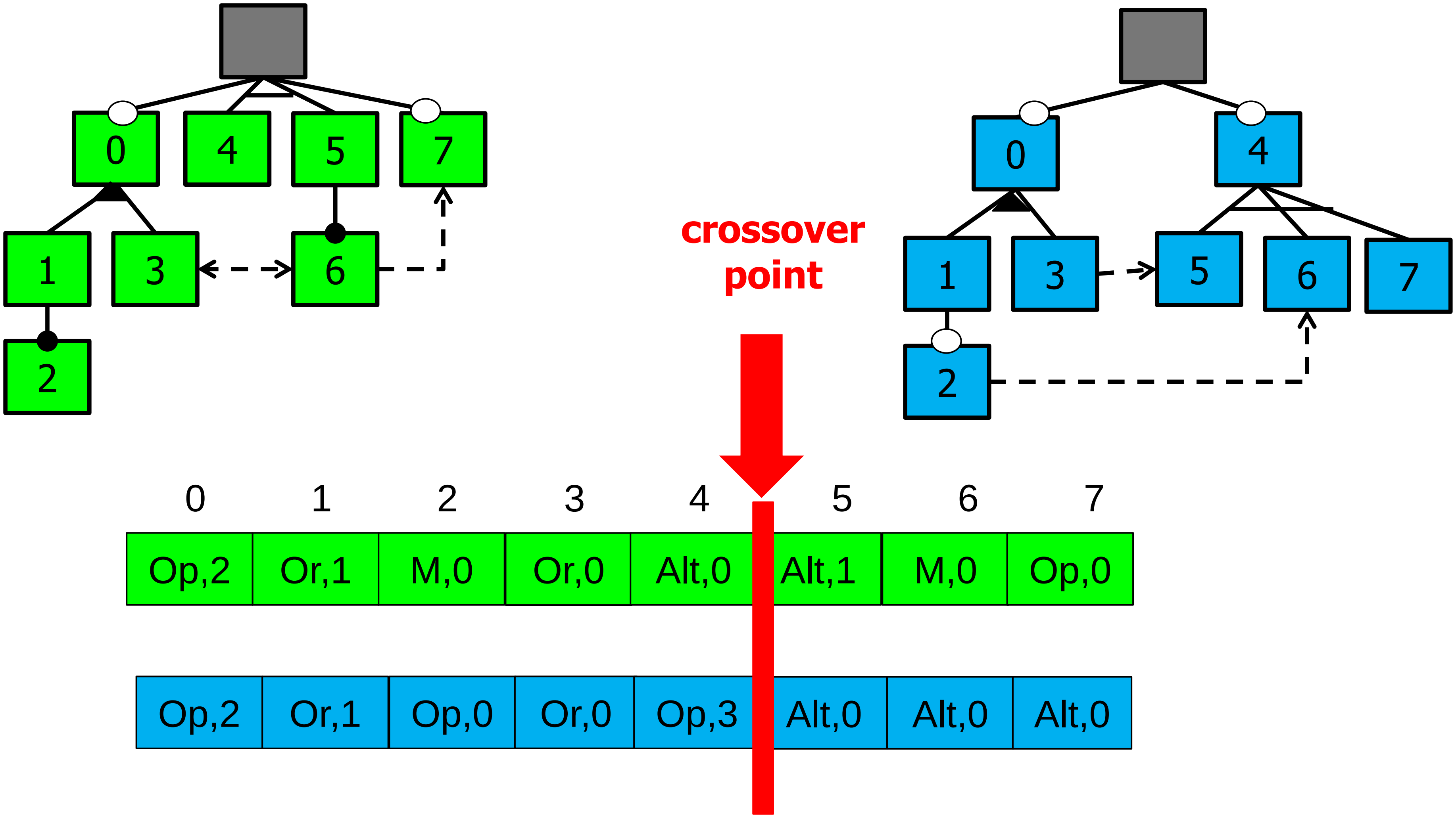
ETHOM
CTC Encoding Example

JOHANNES KEPLER UNIVERSITY LINZ | JKU

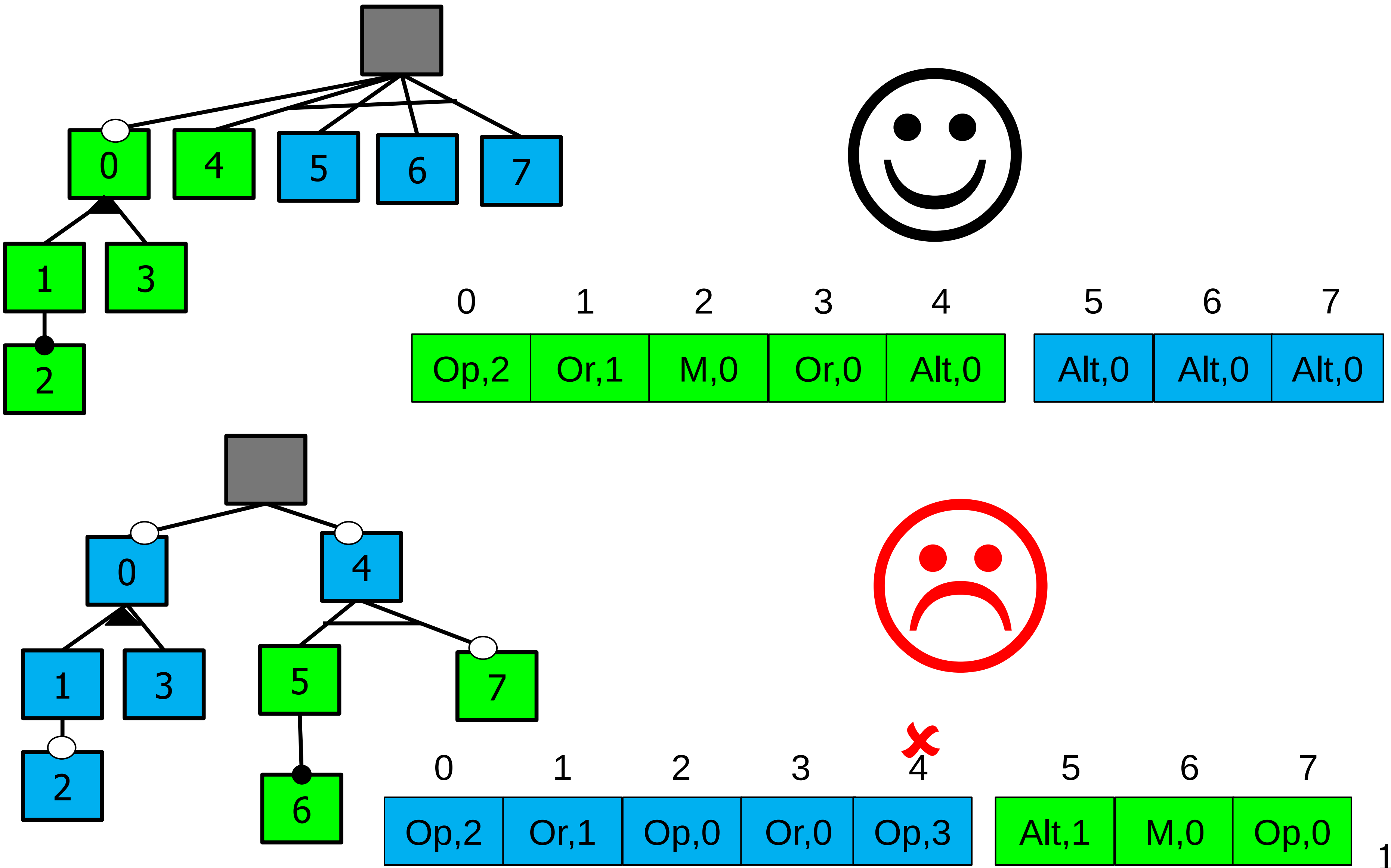


E,4,12	R,5,12	R,8,12	R,8,11

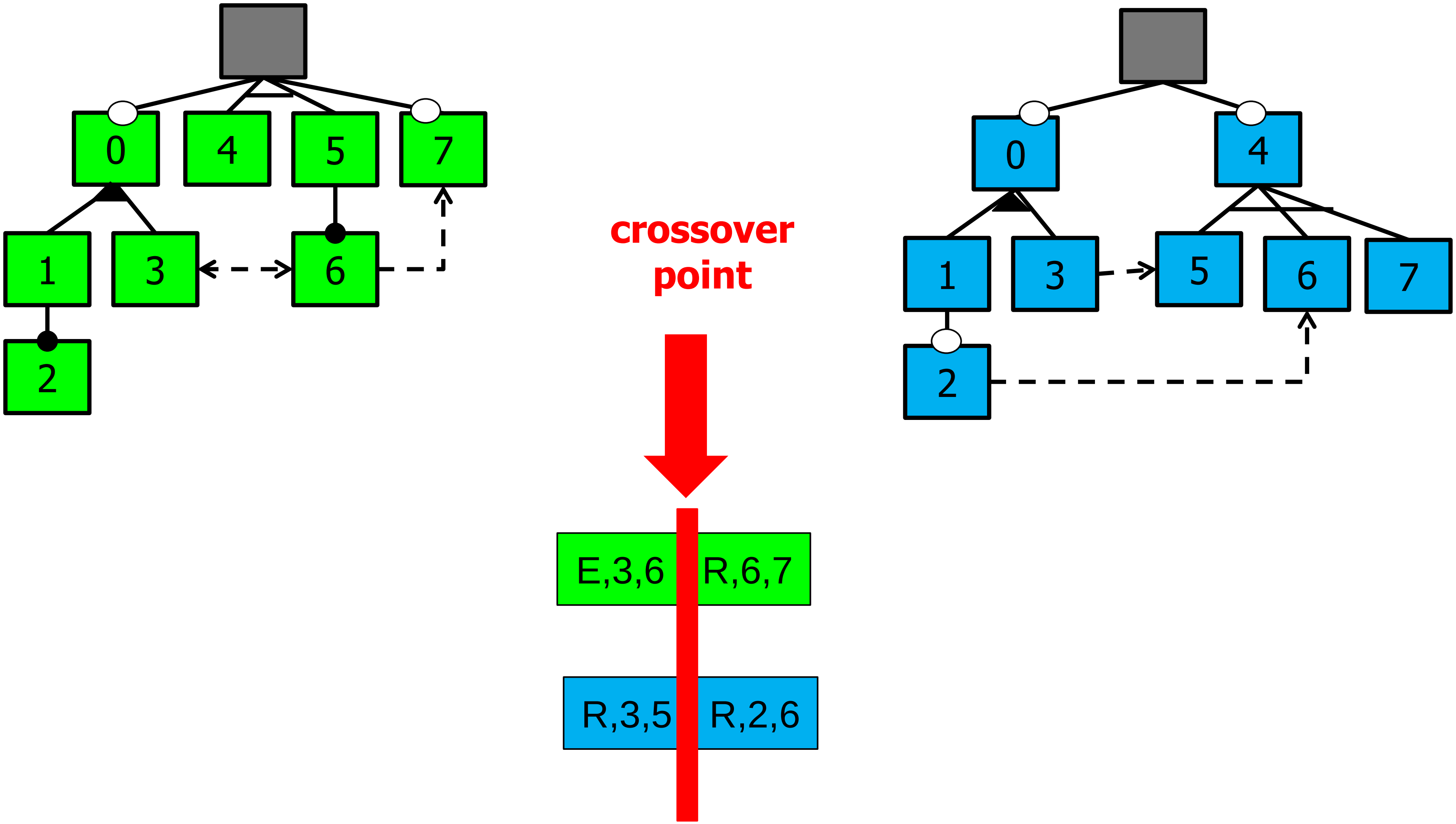
Crossover — One point
(1) Feature Diagram



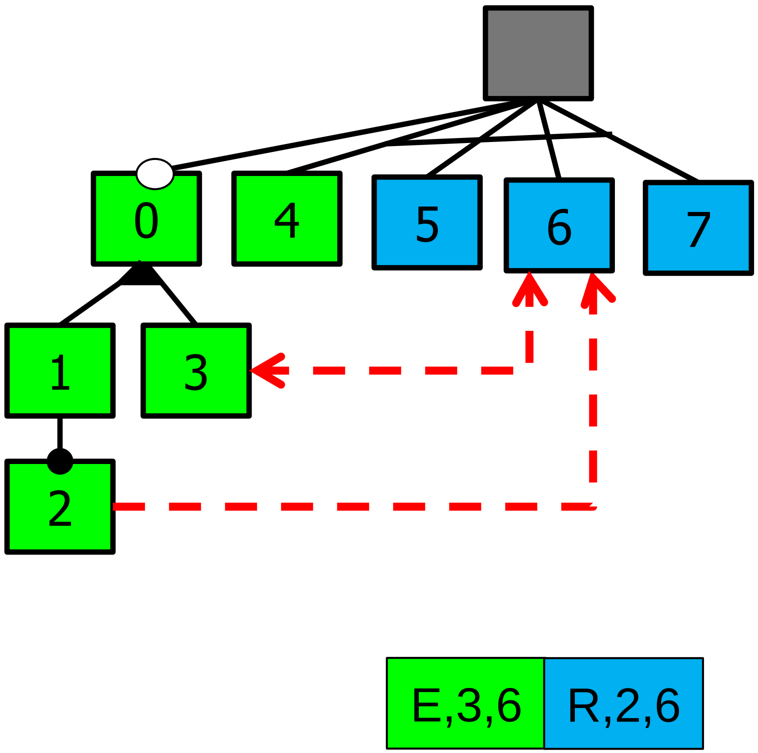
Crossover — One point
(1) Result



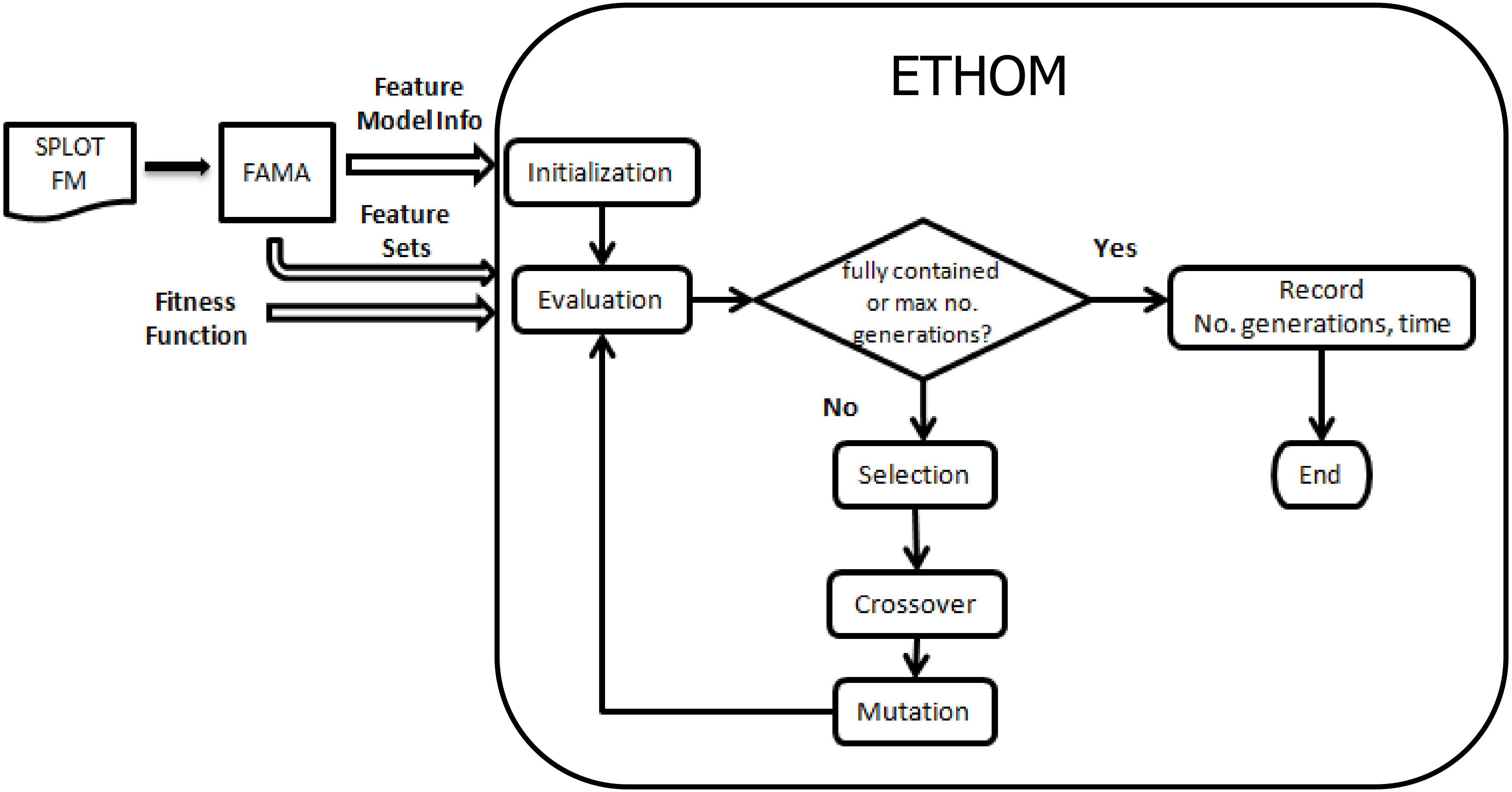
Crossover — One point (2) Cross-Tree Constraints



Crossover — One point
(2) Result



Experimental Setting

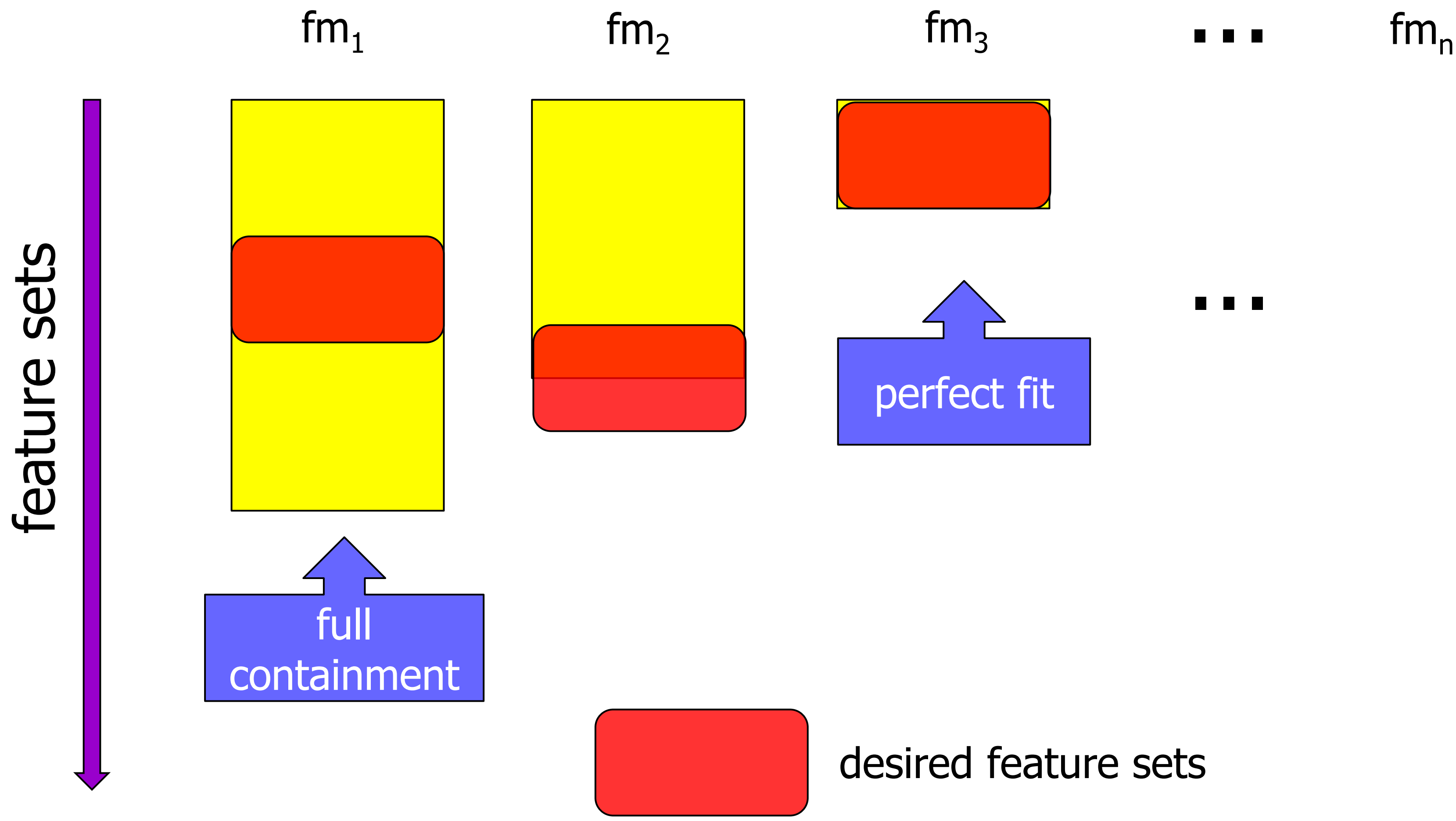


Evaluation

- Case studies
  - 59 feature models from SPLOT repository
  - No. products 1...896
  - No. features 9 ... 27
- Executions
  - Initial populations for each feature model were the same for the fitness functions being analysed
  - 10 runs for each feature model for each fitness function
  - 16 cores at 2.40 GHz, 25GB RAM, Cent OS, Java 1.6

Pictorial View of a Generation

individuals = feature models that denote tables of feature sets



Relaxed Fitness Function

- Relaxed Fitness Function – maximized

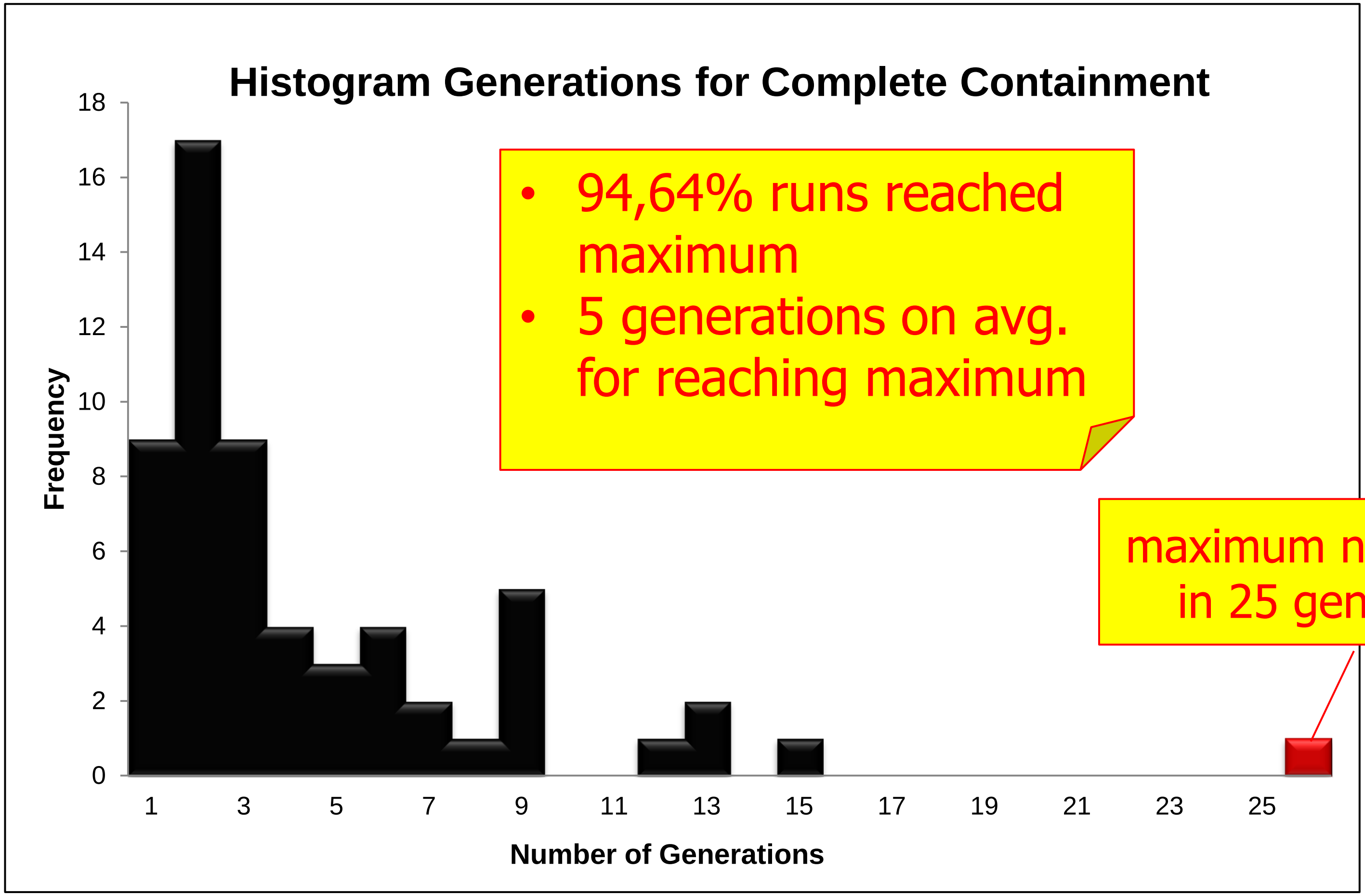
  FFRelaxed(sfs, fm) = | {fs : sfs | validFor(fs,fm) } |

  sfs = set of desired feature sets
  fm = feature model to evaluate
  fs = a feature set

- Auxiliary function validFor
  - checks if a feature set is valid in a FM
  - computed with FAMA using propositional logic

- *Maximizes containment of desired feature sets*

FFRelaxed Results (1)

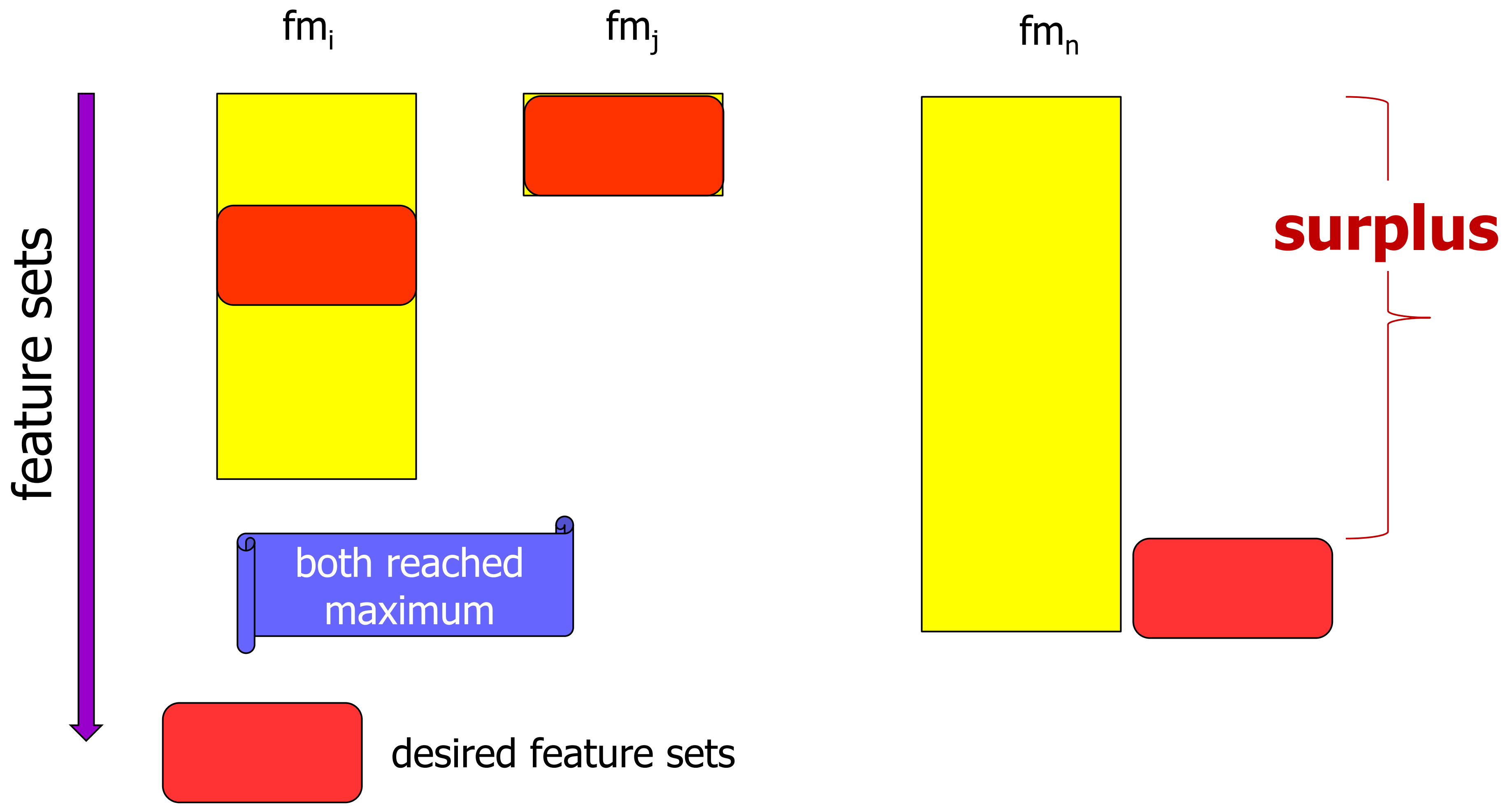


- 94,64% runs reached maximum
- 5 generations on avg. for reaching maximum

maximum not reached in 25 generations

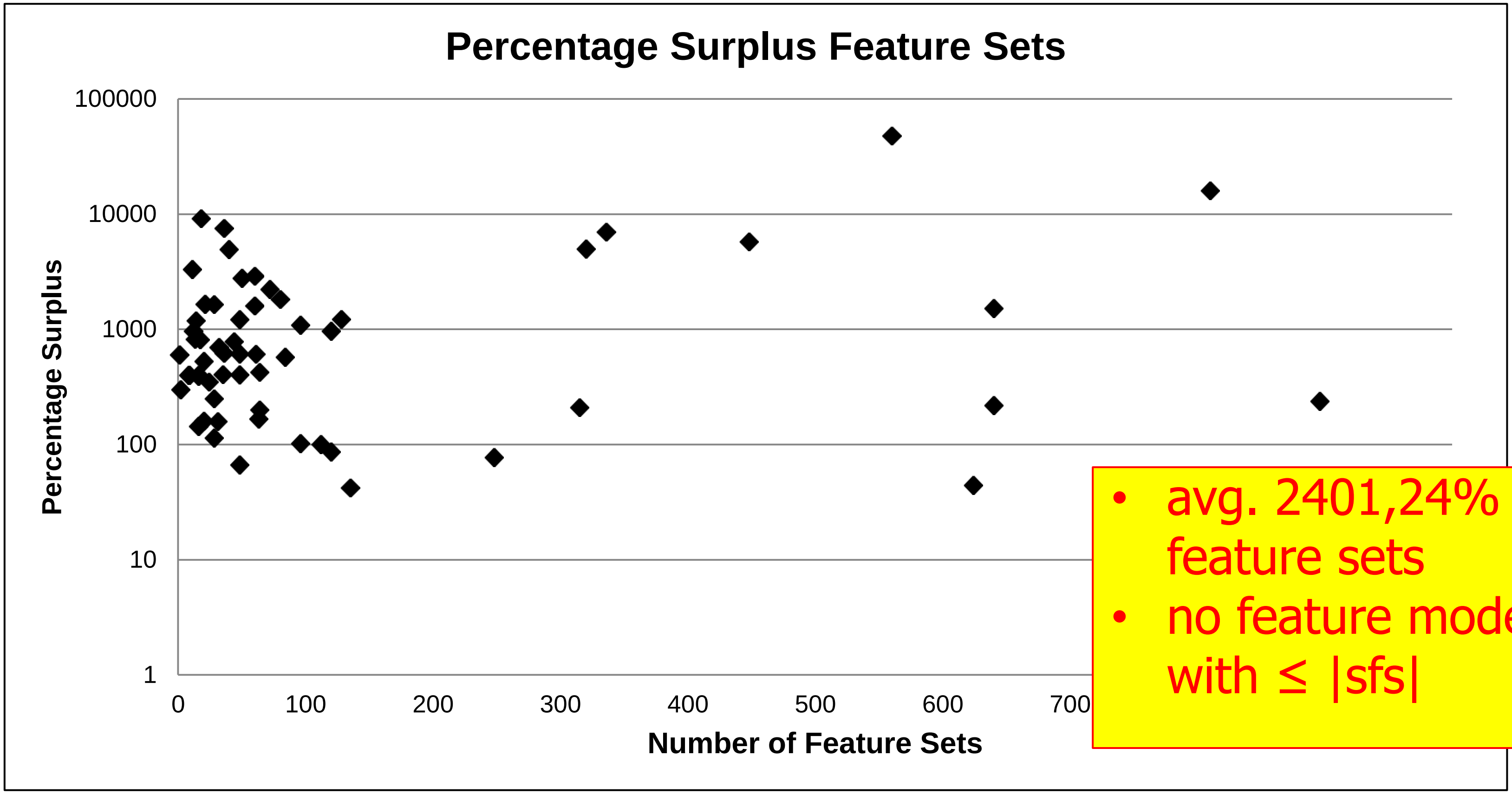
FFRelaxed Results (2)

maximum = cardinality of the desired feature sets



FFRelaxed Results (3)

JOHANNES KEPLER UNIVERSITY LINZ | JKU



- avg. 2401,24% more feature sets
- no feature model with ≤ |sfs|

$$\text{Surplus}(sfs,fm) = \frac{\#products(fm) - |sfs|}{|sfs|} \times 100$$

Ongoing work

- Analyzing other fitness functions
  - Finer comparison granularity
- Comparison with local search approaches
  - Extension to HeuristicLab platform
- Studying variability-aware chromosome operators
  - Crossover and mutation
- Extensions to feature model encodings
  - Based on genetic programming

Stay Tuned!

http://www.sea.uni-linz.ac.at/sbse4vm/

Acknowledgements

C2MV2

Consistency and Composition
for Managing Variability
in Multi-View Systems

MARIE CURIE ACTIONS

FWF Der Wissenschaftsfonds.

Questions?