

Re-engineering Software Variants into Software Product Line

Présentation extraite de la soutenance de thèse de
M. Ra'Fat AL-MSIE'DEEN
University of Montpellier

Software product variants

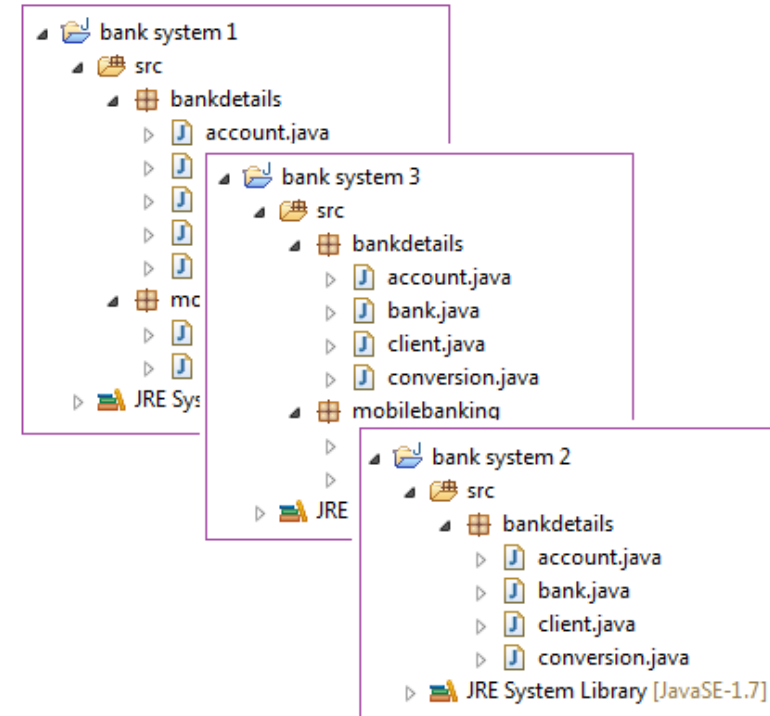
1. Software product variants

■ Are similar software

- Share mandatory features
- Differ in optional features
- Developed via clone-and-own approach

■ Examples

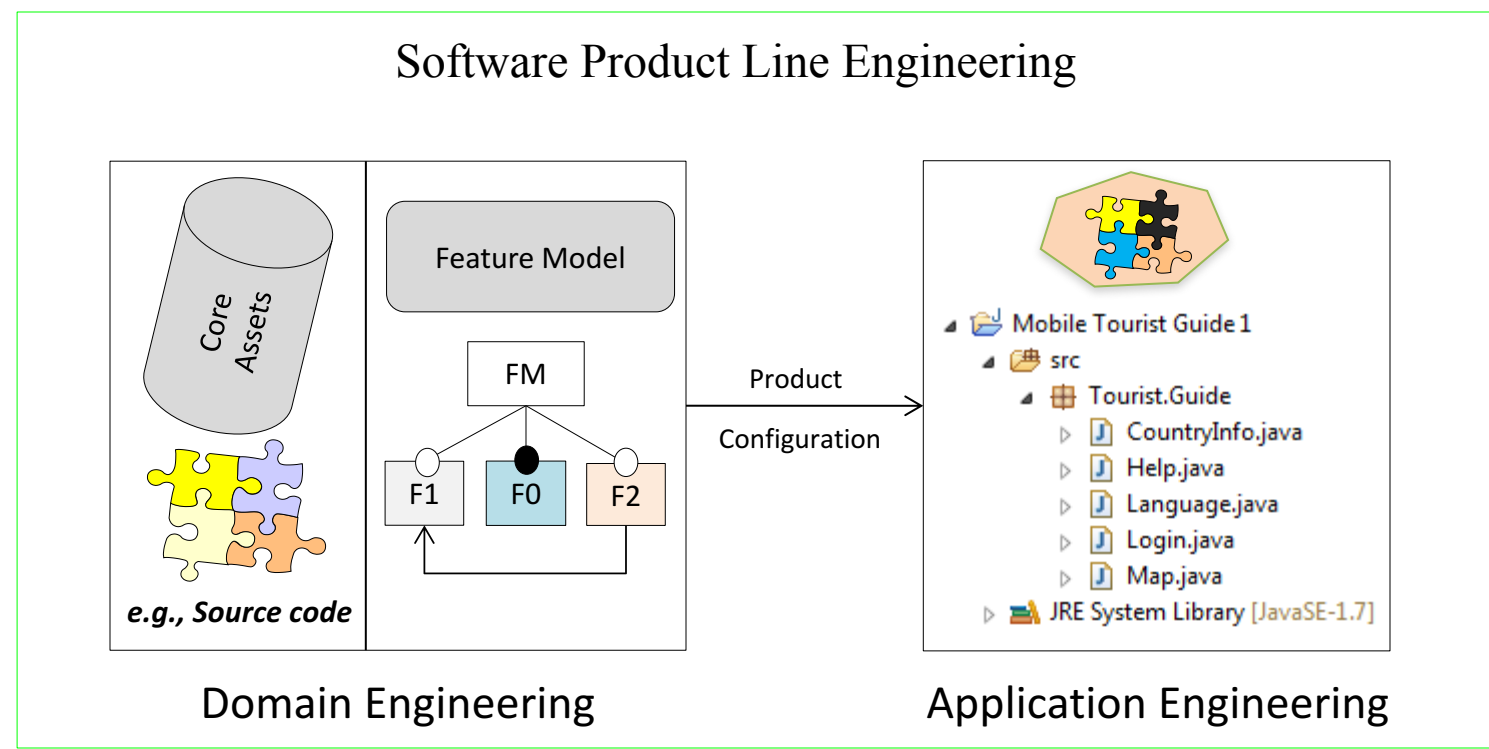
- Linux kernel « <https://www.kernel.org/> »
- Mobile media « <http://www.ic.unicamp.br/~tizzei/mobilemedia/> »
- ArgoUML « <http://argouml-spl.tigris.org/> »



Software product Line

2. Software product Line

- Software-intensive systems come in many variants
- Motivations:
 - Reduce cost and time of software development
 - reuse, etc.



Software product Line

Instance-1

```
class HelloWorld { Hello
    void print() {
        System.out.print("Hello");
    }
    static void main(String[] args) {
        new HelloWorld().print();
    }
}
```

```
class HelloWorld { World
    void print() {
        original();
        System.out.print("_world!");
    }
}
```

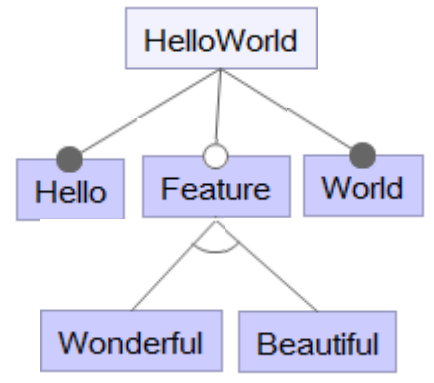
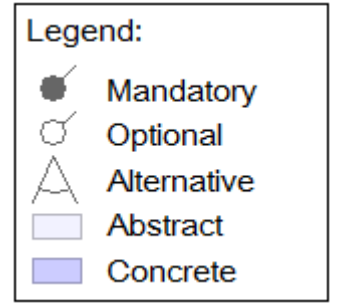
```
class HelloWorld { Wonderful
    void print() {
        original();
        System.out.print("_wonderful");
    }
}
```

Instance-2

```
class HelloWorld { Hello
    void print() {
        System.out.print("Hello");
    }
    static void main(String[] args) {
        new HelloWorld().print();
    }
}
```

```
class HelloWorld { World
    void print() {
        original();
        System.out.print("_world!");
    }
}
```

```
class HelloWorld { Beautiful
    void print() {
        original();
        System.out.print("_beautiful");
    }
}
```



Problem

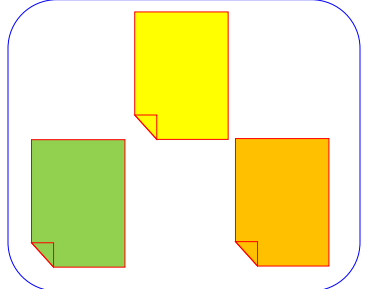
✓ Software product variants

■ Difficulties for:

○ Reuse

○ Maintenance

○ Program understanding (comprehension)



... Feature (mandatory, optional) ?

... Feature Name and description ?

... Feature dependencies (feature model) ?

✓ Software Product Line

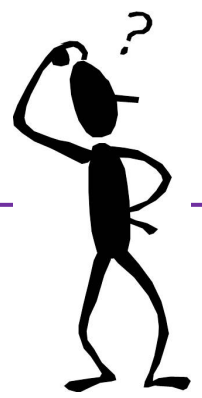
■ Design from scratch is a hard task

Goal

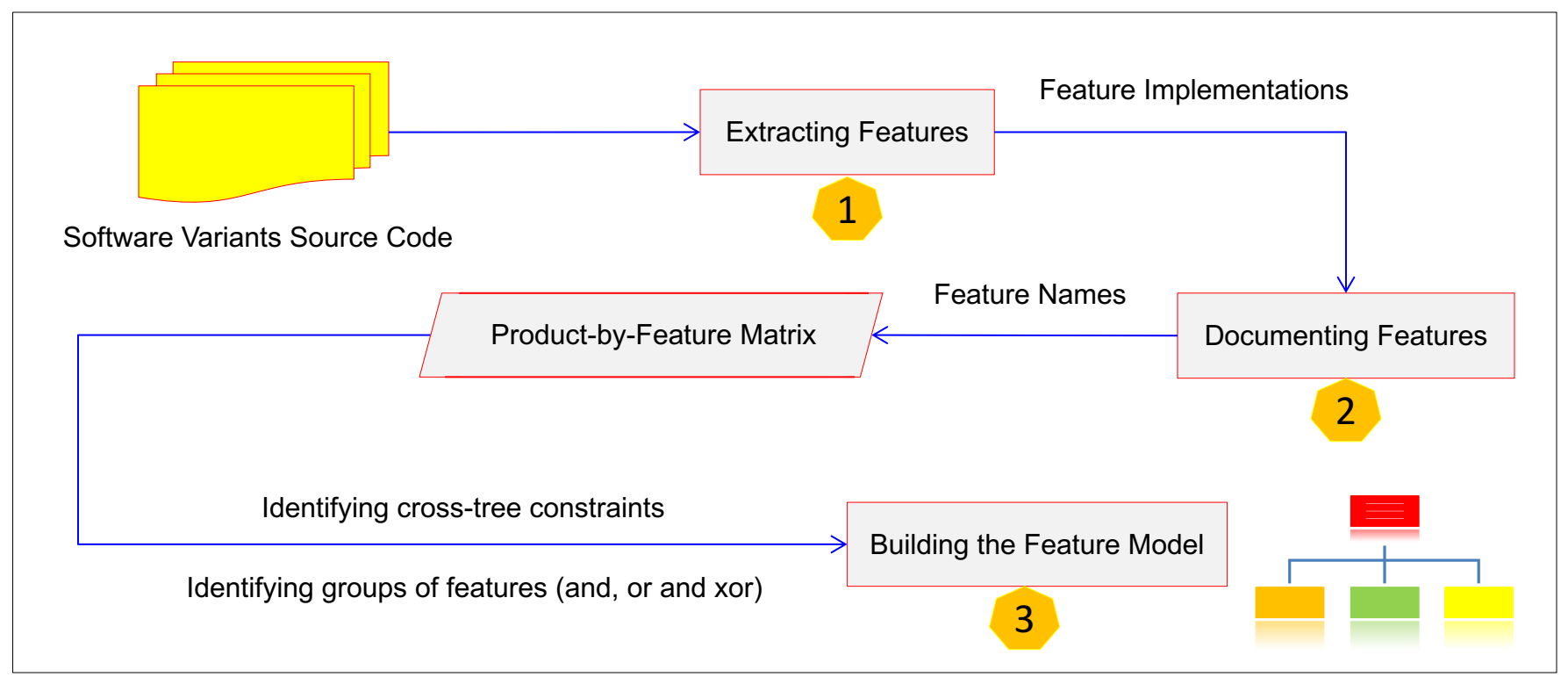
- ✓ Reverse engineering FM from the source code of software product variants

Strategy

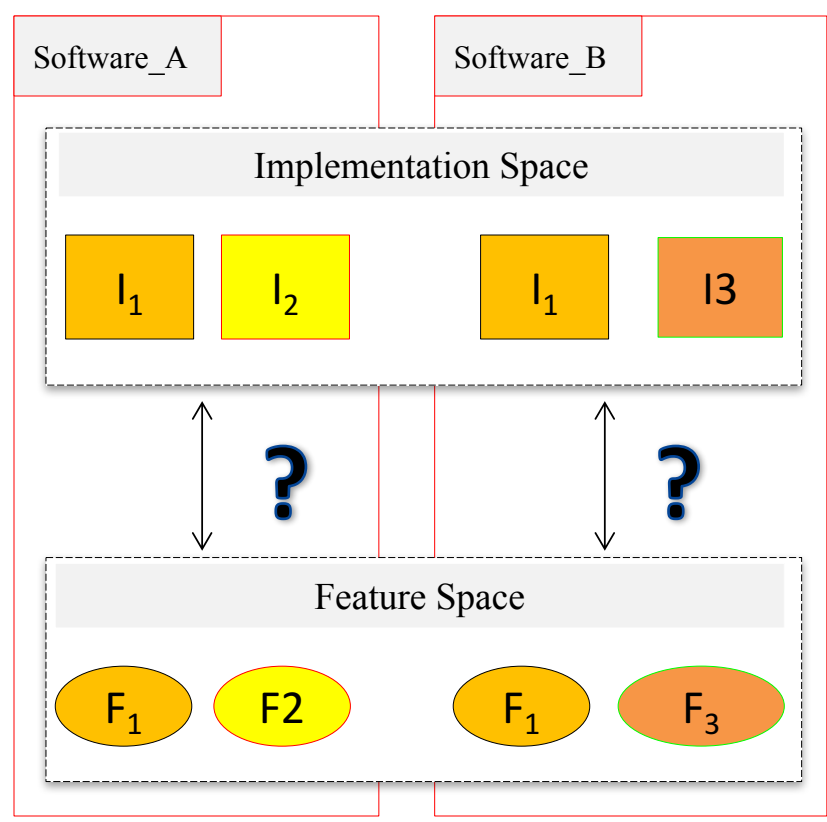
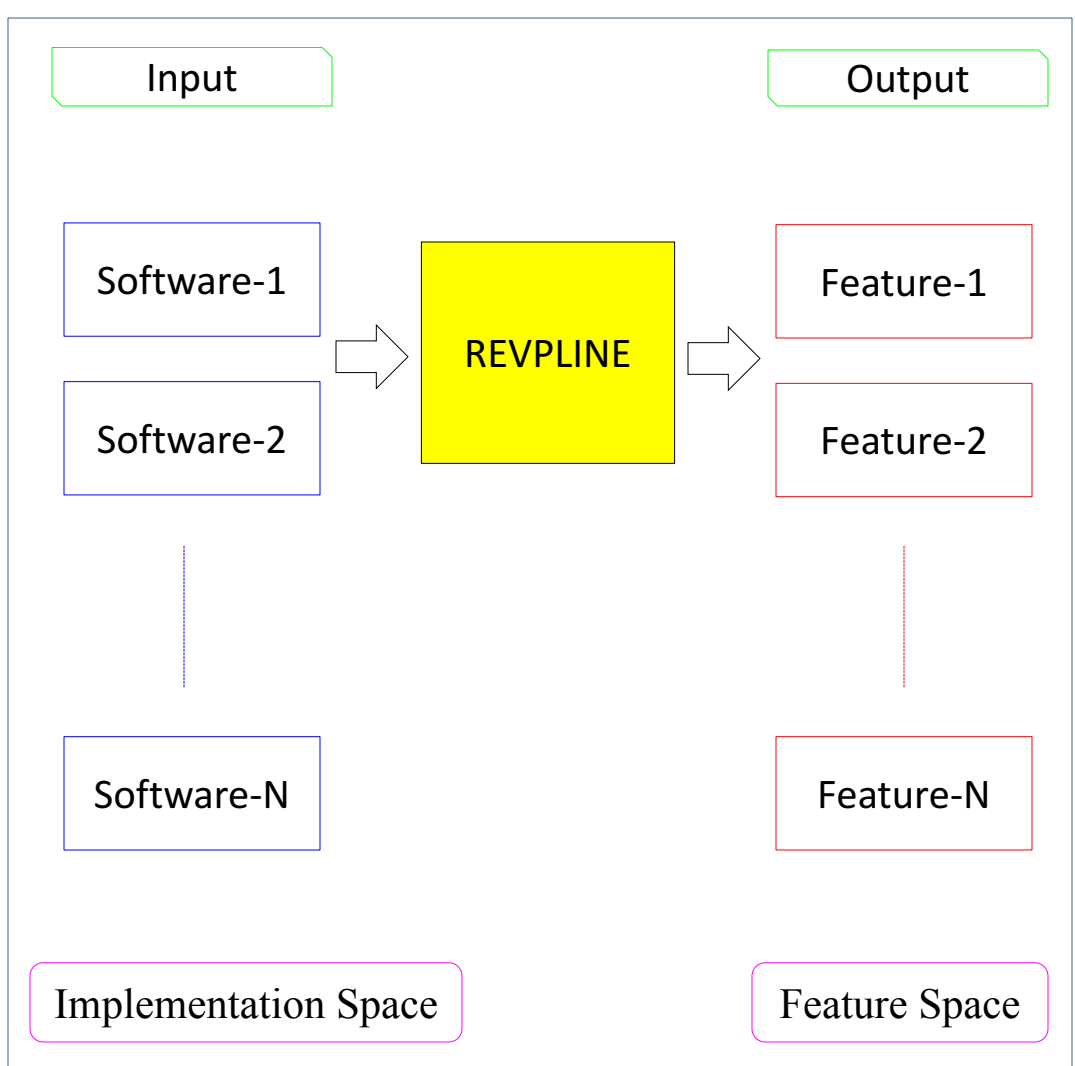
- Feature model mining (reverse engineering step):
 - ✓ Mining functional features
 - ✓ Documenting mined feature implementations
 - ✓ Mining feature dependencies (require, exclude, group of features: xor, or ,and)



Process



Proposal



Contribution

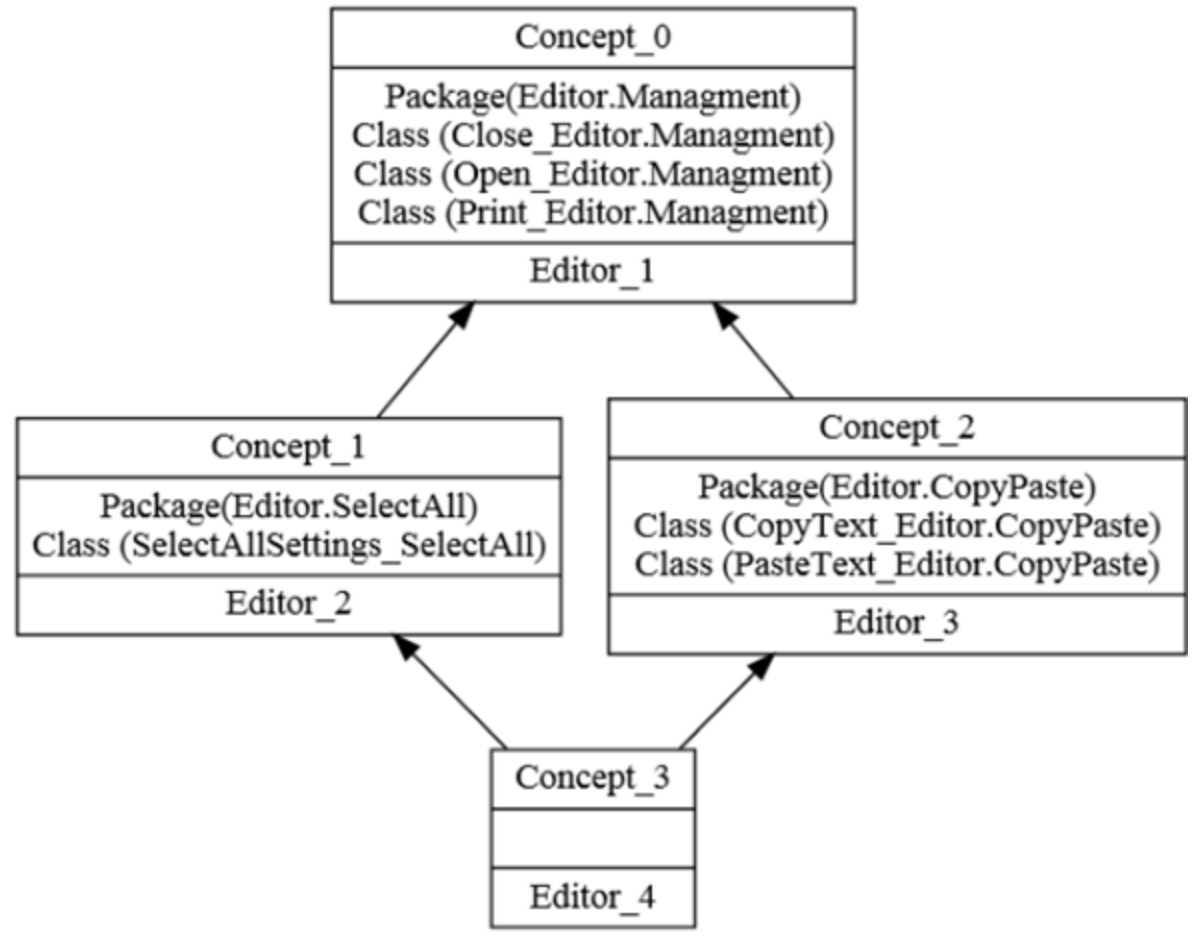
- We **exploit commonality and variability** across the source code of software variants, to apply IR methods in an efficient way
- We rely on **lexical** and **structural** similarity to mine feature implementation
- Variability at **different levels of source code elements**
- The **REVPLINE** feature location approach uses two techniques: **FCA** and **LSI**

Used Technique

Formal Concept Analysis (FCA)

✓ « objects + attributes ⇨ classified concepts »

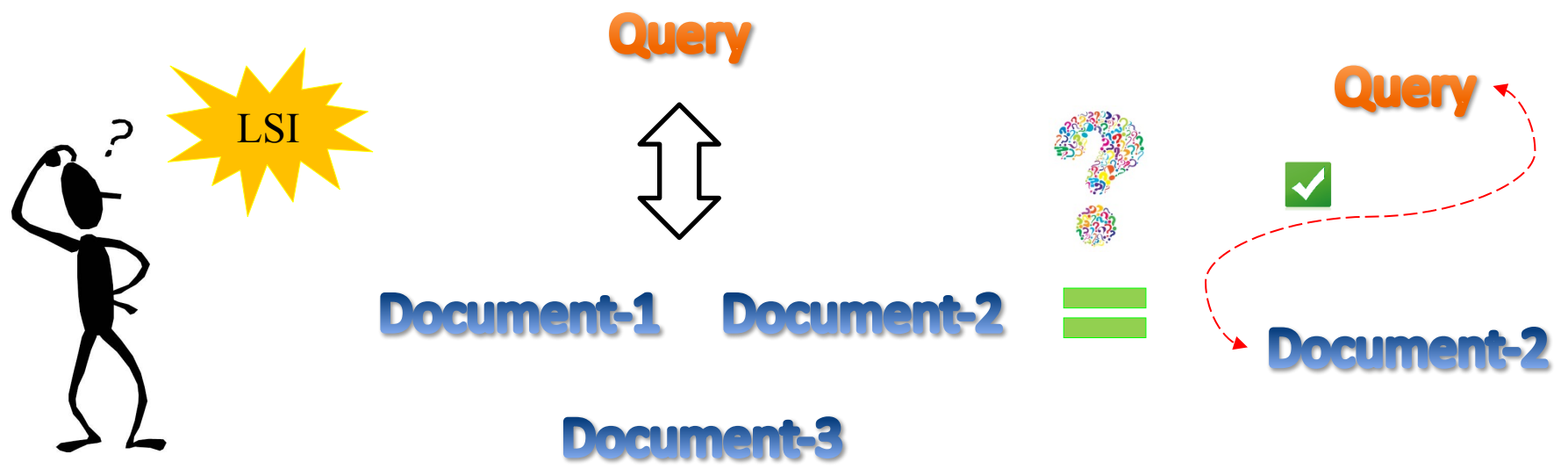
		Package(Editor.Management)	Class (Close_Editor.Management)	Class (Open_Editor.Management)	Class (Print_Editor.Management)	Package(Editor.CopyPaste)	Class (CopyText_Editor.CopyPaste)	Class (PasteText_Editor.CopyPaste)	Package(Editor.SelectAll)	Class (SelectAllSettings_SelectAll)
Editor_1	x	x	x	x						
Editor_2	x	x	x	x				x	x	
Editor_3	x	x	x	x	x	x	x			
Editor_4	x	x	x	x	x	x	x	x	x	x



The Formal Context and AOC-poset for Text Editor software Variants

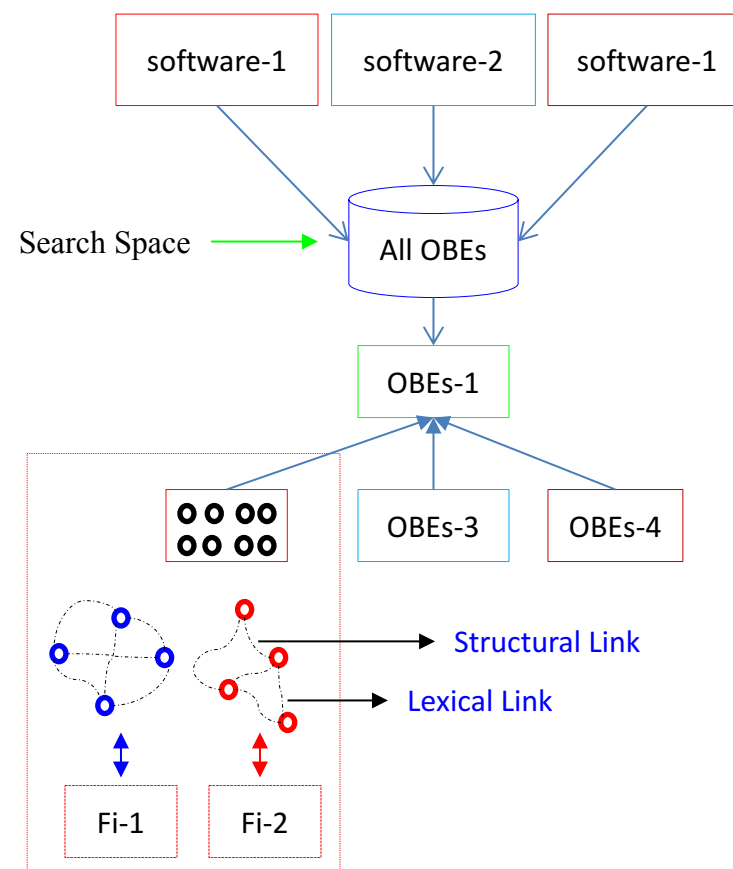
Used Technique

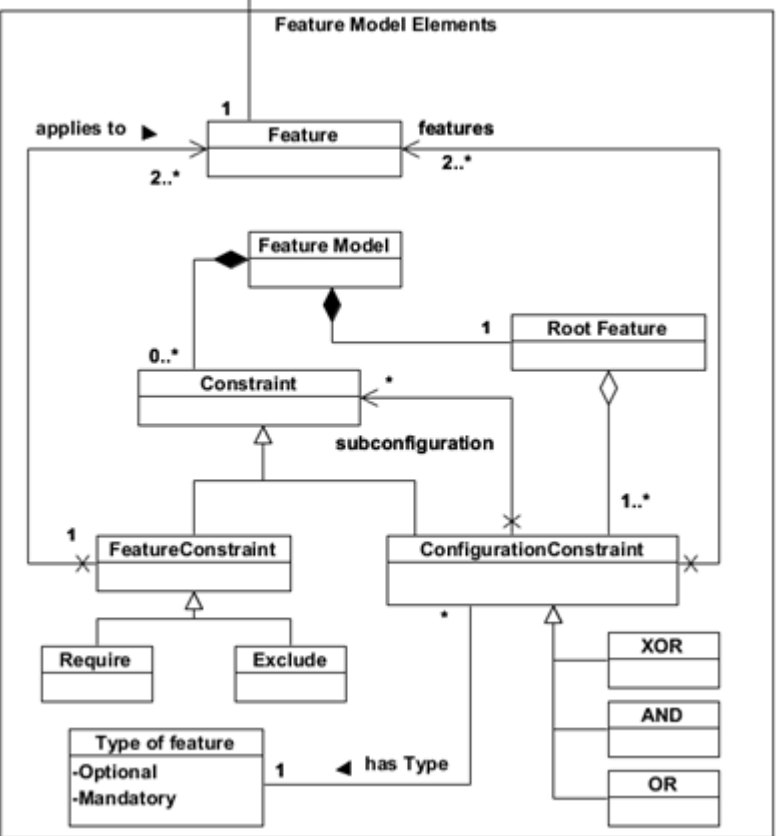
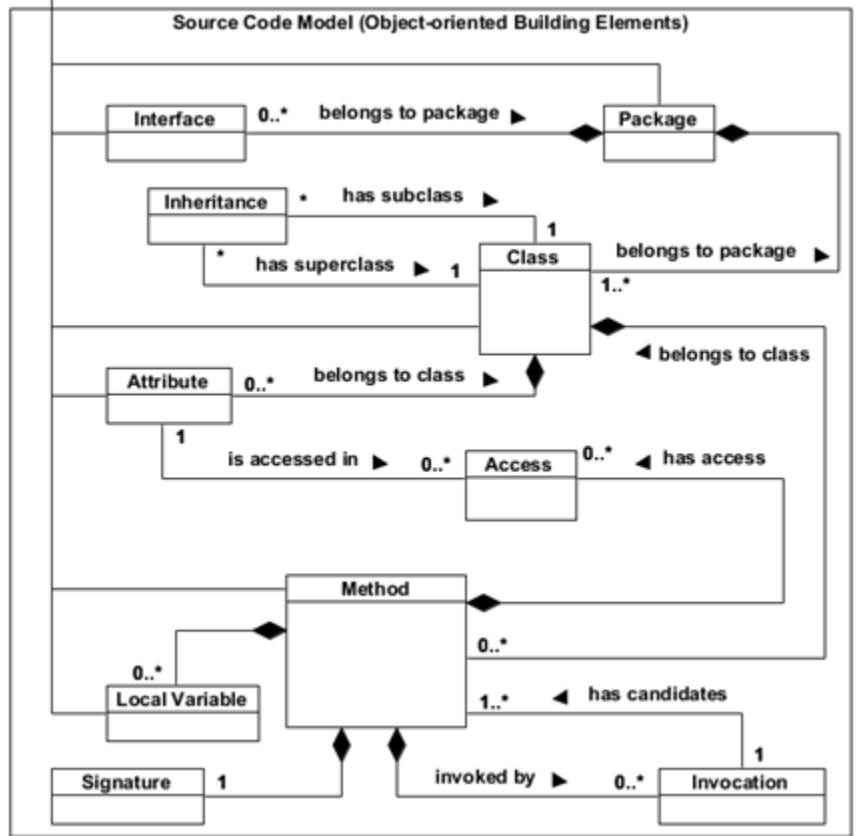
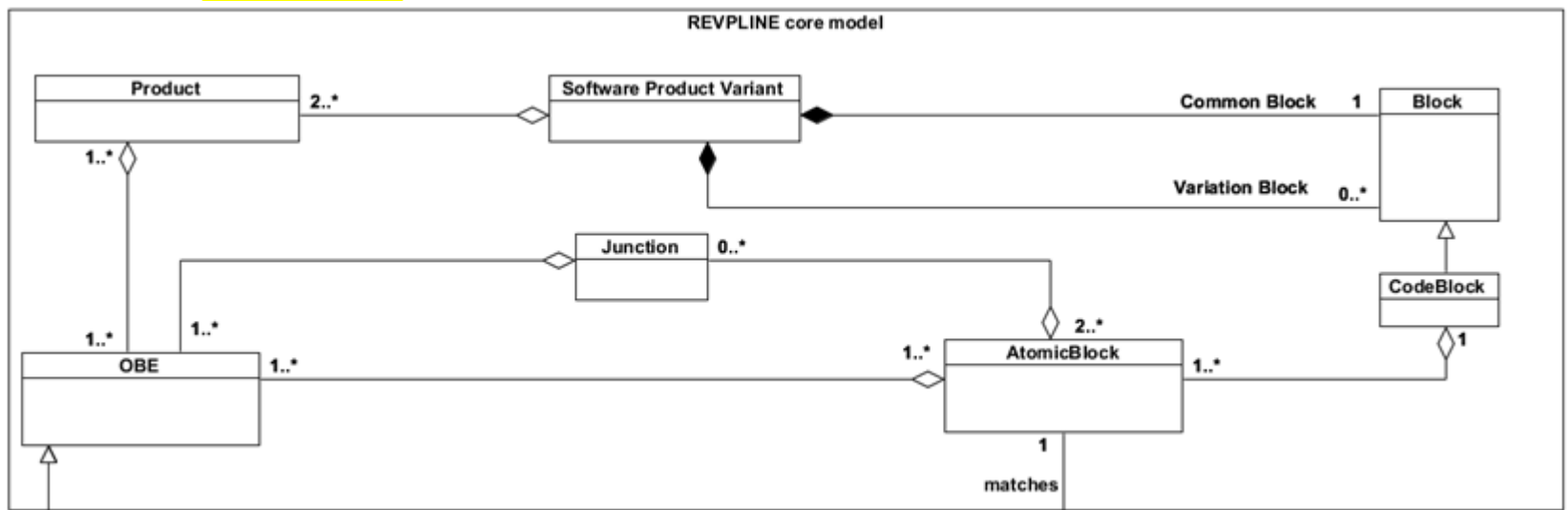
- Latent Semantic Indexing (LSI)
 - ✓ IR technique
 - ✓ Computes textual similarity among different documents
 - ✓ If two documents share a large number of terms, those documents are similar



Key Ideas

- A Feature has the **same implementation** in all product variants where it is present
- Feature are implemented as OBEs: **package**, **class**, **attribute**, **method**, etc.
- **Junction** = overlap of feature implementations

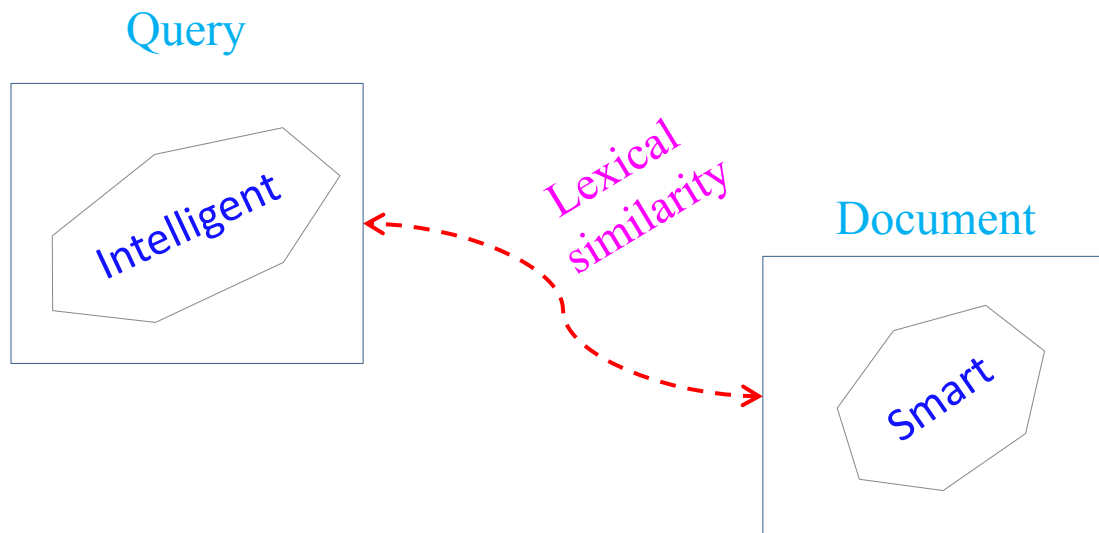




Lexical Versus Structural Similarity Between OBEs

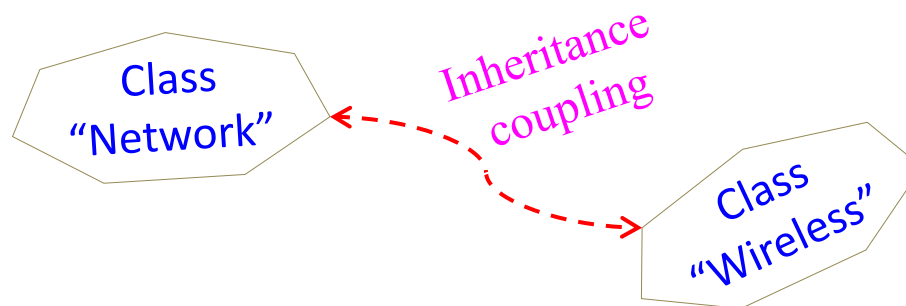
- Lexical Similarity

- LSI method

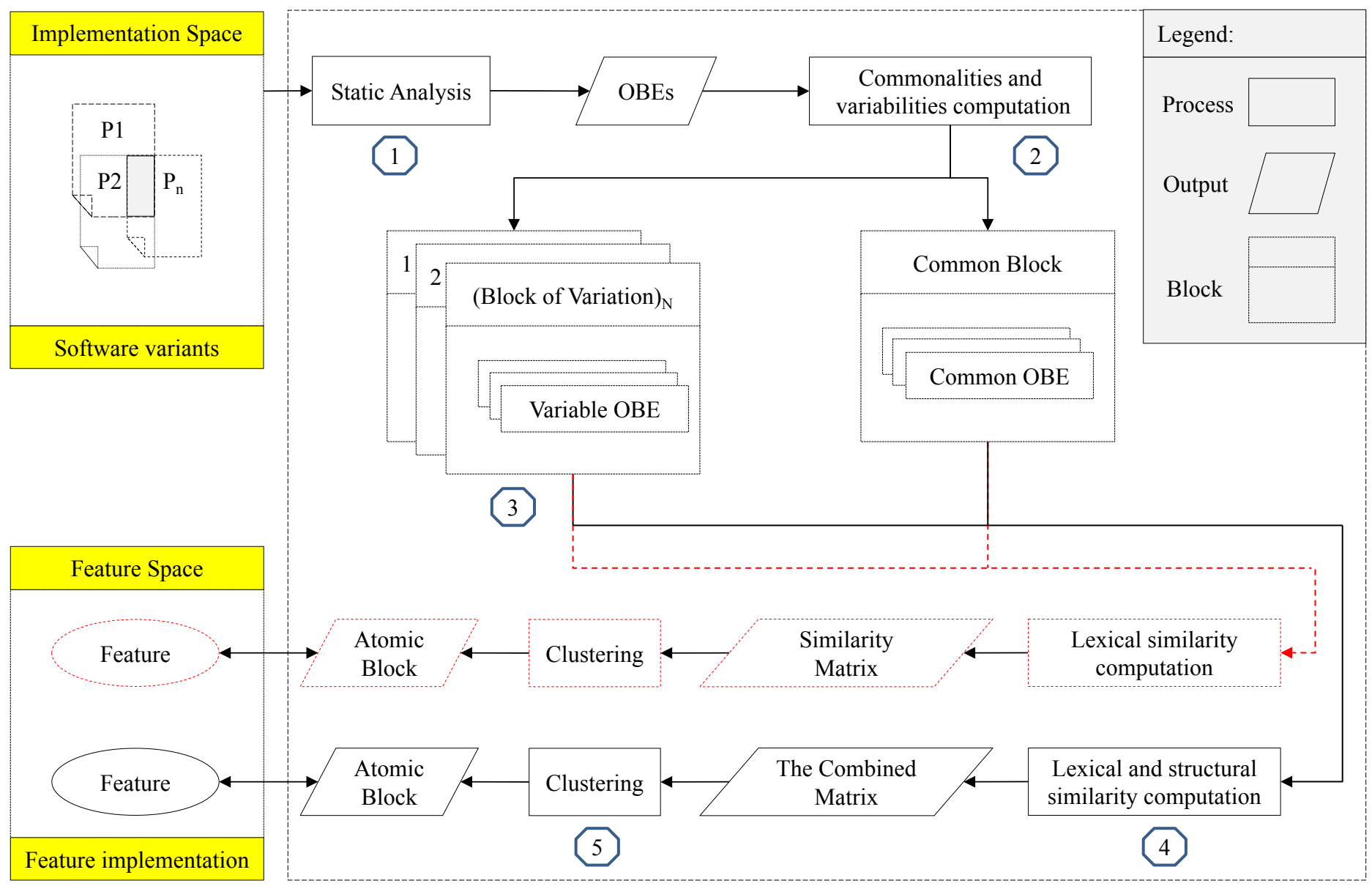


- Structural Similarity

- We consider five dependencies "coupling" between OBEs: inheritance, method invocation, composition, attribute access and combined coupling



Process



Example

- An Illustrative Example: Drawing Shapes Software Variants

<div style="text-align: center;">Features</div> <div style="text-align: center;">Software</div>	Draw_line	Insert_image	Draw_arc	Insert_text	Draw_oval	Draw_rectangle	Draw_ThreeDRectangle	Copy	Paste
Drawing Shapes Software 1	X	X							
Drawing Shapes Software 2	X	X	X	X					
Drawing Shapes Software 3	X	X			X				
Drawing Shapes Software 4	X	X				X			
Drawing Shapes Software 5	X	X	X	X	X	X	X		
Drawing Shapes Software 6	X	X	X	X	X	X	X	X	X

Example

- A formal context describing drawing shapes software variants by their OBEs

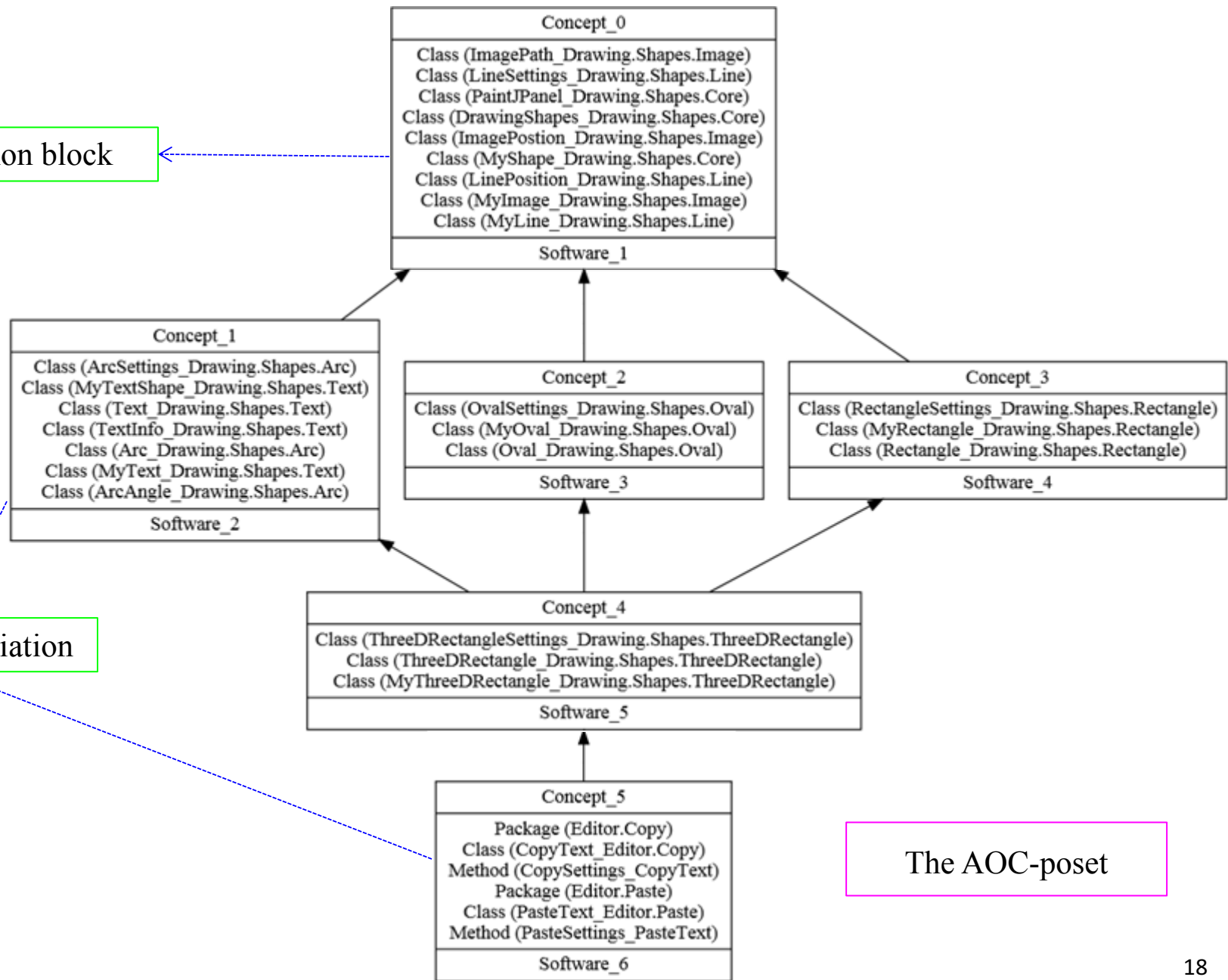
		OBEs															
	Software	Class (PaintJPanel_Drawing.Shapes.Core)	Class (DrawingShapes_Drawing.Shapes.Core)	Class (MyShape_Drawing.Shapes.Core)	Class (LineSettings_Drawing.Shapes.Line)	Class (LinePosition_Drawing.Shapes.Line)	Class (MyLine_Drawing.Shapes.Line)	Class (ImagePath_Drawing.Shapes.Image)	Class (MyImage_Drawing.Shapes.Image)	Class (ImagePosition_Drawing.Shapes.Image)	Class (ArcSettings_Drawing.Shapes.Arc)	Class (Arc_Drawing.Shapes.Arc)	Class (ArcAngle_Drawing.Shapes.Arc)	Class (MyTextShape_Drawing.Shapes.Text)	Class (Text_Drawing.Shapes.Text)	Class (TextInfo_Drawing.Shapes.Text)	Class (MyText_Drawing.Shapes.Text)
Software_1		X	X	X	X	X	X	X	X	X							
Software_2		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Example

Common block

Blocks of variation

The AOC-poset



Measuring OBEs' Similarity Based on Lexical Similarity

1 Concept_5

Package (Editor.Copy)
 Class (CopyText_Editor.Copy)
 Method (CopySettings_CopyText)
 Package (Editor.Paste)
 Class (PasteText_Editor.Paste)
 Method (PasteSettings_PasteText)

Software_6

3

	Class (CopyText_Editor.Copy)	Class (PasteText_Editor.Paste)	Method (CopySettings_CopyText)	Method (PasteSettings_PasteText)	Package (Editor.Copy)	Package (Editor.Paste)
Class (CopyText_Editor.Copy)	1.0	0.035384	1.0	0.031342	0.999467	0.001207
Class (PasteText_Editor.Paste)	0.035384	1.0	0.035384	0.999991	0.002748	0.999415
Method (CopySettings_CopyText)	1.0	0.035384	1.0	0.031342	0.999467	0.001207
Method (PasteSettings_PasteText)	0.031342	0.999991	0.031342	0.999999	-0.001295	0.999545
Package (Editor.Copy)	0.999467	0.002748	0.999467	-0.001295	1.0	-0.031431
Package (Editor.Paste)	0.001207	0.999415	0.001207	0.999545	-0.031431	1.0

2

	Class (CopyText_Editor.Copy)	Class (PasteText_Editor.Paste)	Method (CopySettings_CopyText)	Method (PasteSettings_PasteText)	Package (Copy)	Package (Paste)
copy	2	0	2	0	1	0
paste	0	2	0	2	0	1
settings	0	0	1	1	0	0
text	1	1	1	1	0	0

The term-document matrix
 The term-query matrix

4

	Class (CopyText_Editor.Copy)	Class (PasteText_Editor.Paste)	Method (CopySettings_CopyText)	Method (PasteSettings_PasteText)	Package (Editor.Copy)	Package (Editor.Paste)
Class (CopyText_Editor.Copy)	X		X		X	
Class (PasteText_Editor.Paste)		X		X		X
Method (CopySettings_CopyText)	X		X		X	
Method (PasteSettings_PasteText)		X		X		X
Package (Editor.Copy)	X		X		X	
Package (Editor.Paste)		X		X		X

Formal context

The similarity matrix

5

Concept_0

Class (CopyText_Editor.Copy)
 Method (CopySettings_CopyText)
 Package (Copy)

Package (Copy)
 Method (CopySettings_CopyText)
 Class (CopyText_Editor.Copy)

Concept_1

Class (PasteText_Editor.Paste)
 Method (PasteSettings_PasteText)
 Package (Paste)

Method (PasteSettings_PasteText)
 Package (Paste)
 Class (PasteText_Editor.Paste)

Atomic Blocks

Measuring OBEs' Similarity Based on Lexical & Structural Similarity

Concept_1
Class (ArcSettings_Drawing.Shapes.Arc) Class (MyTextShape_Drawing.Shapes.Text) Class (Text_Drawing.Shapes.Text) Class (TextInfo_Drawing.Shapes.Text) Class (Arc_Drawing.Shapes.Arc) Class (MyText_Drawing.Shapes.Text) Class (ArcAngle_Drawing.Shapes.Arc)
Software_2

1

2

The dependency structure matrix

	Class (ArcSettings_Drawing.Shapes.Arc)						
	Class (Arc_Drawing.Shapes.Arc)						
	Class (ArcAngle_Drawing.Shapes.Arc)						
	Class (MyTextShape_Drawing.Shapes.Text)						
	Class (Text_Drawing.Shapes.Text)						
	Class (TextInfo_Drawing.Shapes.Text)						
	Class (MyText_Drawing.Shapes.Text)						
Class (ArcSettings_Drawing.Shapes.Arc)		X	X				
Class (Arc_Drawing.Shapes.Arc)	X		X				
Class (ArcAngle_Drawing.Shapes.Arc)	X	X					
Class (MyTextShape_Drawing.Shapes.Text)					X		
Class (Text_Drawing.Shapes.Text)				X		X	X
Class (TextInfo_Drawing.Shapes.Text)					X		
Class (MyText_Drawing.Shapes.Text)					X		

Concept_0
Class (Arc_Drawing.Shapes.Arc) Class (ArcAngle_Drawing.Shapes.Arc) Class (ArcSettings_Drawing.Shapes.Arc)
Class (ArcSettings_Drawing.Shapes.Arc) Class (Arc_Drawing.Shapes.Arc) Class (ArcAngle_Drawing.Shapes.Arc)

Concept_1
Class (MyText_Drawing.Shapes.Text) Class (MyTextShape_Drawing.Shapes.Text) Class (Text_Drawing.Shapes.Text) Class (TextInfo_Drawing.Shapes.Text)
Class (MyText_Drawing.Shapes.Text) Class (MyTextShape_Drawing.Shapes.Text) Class (TextInfo_Drawing.Shapes.Text) Class (Text_Drawing.Shapes.Text)

Atomic Blocks

3

The lexical similarity matrix

	Class (Arc_Drawing.Shapes.Arc)	Class (ArcAngle_Drawing.Shapes.Arc)	Class (ArcSettings_Drawing.Shapes.Arc)	Class (MyTextShape_Drawing.Shapes.Text)	Class (MyText_Drawing.Shapes.Text)	Class (Text_Drawing.Shapes.Text)	Class (TextInfo_Drawing.Shapes.Text)
Class (Arc_Drawing.Shapes.Arc)	X	X	X				
Class (ArcAngle_Drawing.Shapes.Arc)	X	X	X				
Class (ArcSettings_Drawing.Shapes.Arc)	X	X	X				
Class (MyTextShape_Drawing.Shapes.Text)				X	X		X
Class (MyText_Drawing.Shapes.Text)				X	X		X
Class (Text_Drawing.Shapes.Text)						X	
Class (TextInfo_Drawing.Shapes.Text)				X	X		X

4

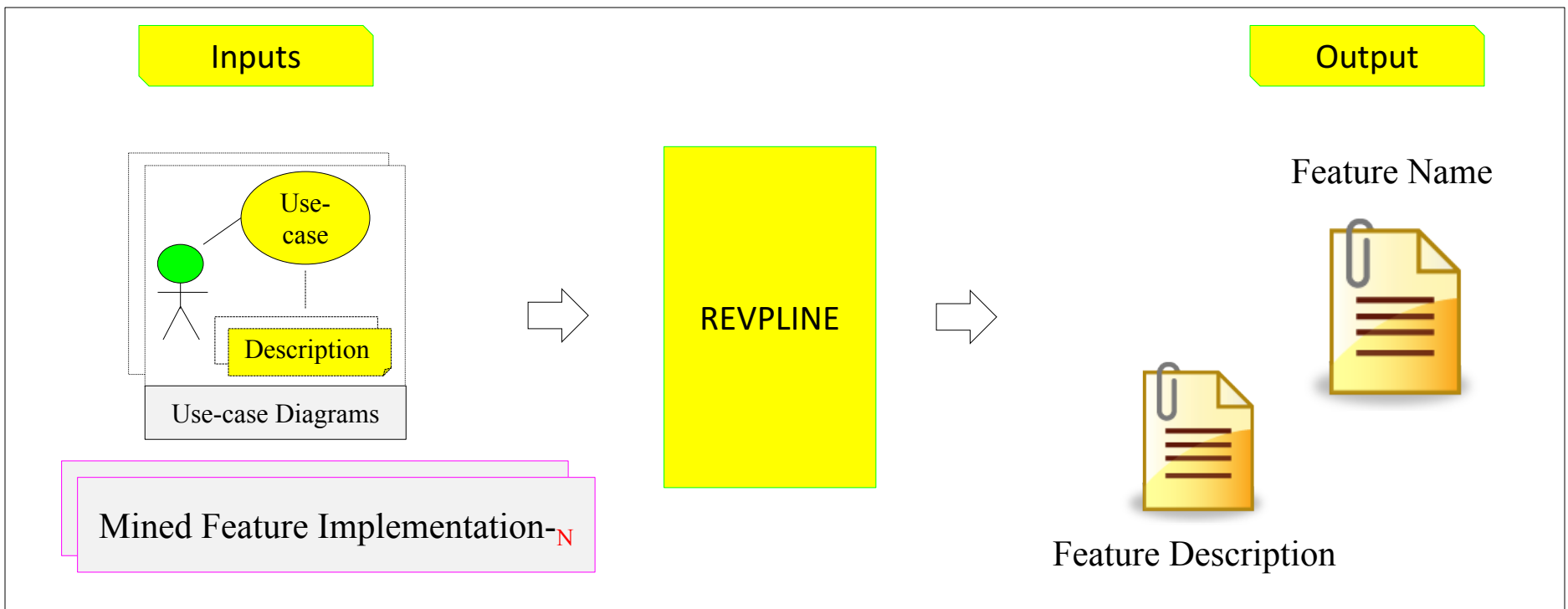
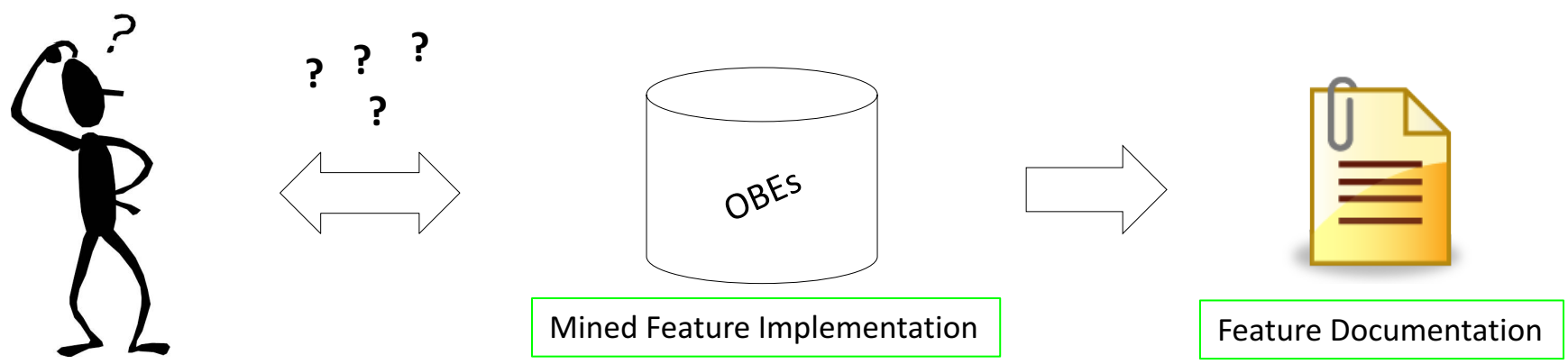
The combined matrix

	Class (Arc_Drawing.Shapes.Arc)	Class (ArcAngle_Drawing.Shapes.Arc)	Class (ArcSettings_Drawing.Shapes.Arc)	Class (MyTextShape_Drawing.Shapes.Text)	Class (MyText_Drawing.Shapes.Text)	Class (Text_Drawing.Shapes.Text)	Class (TextInfo_Drawing.Shapes.Text)
Class (Arc_Drawing.Shapes.Arc)	X	X	X				
Class (ArcAngle_Drawing.Shapes.Arc)	X	X	X				
Class (ArcSettings_Drawing.Shapes.Arc)	X	X	X				
Class (MyTextShape_Drawing.Shapes.Text)				X	X	X	X
Class (MyText_Drawing.Shapes.Text)				X	X	X	X
Class (Text_Drawing.Shapes.Text)				X	X	X	X
Class (TextInfo_Drawing.Shapes.Text)				X	X	X	X

5

Documenting the Mined Feature Implementation

Proposal



State of the Art

1. **Single software system** = labels / names / topics / code summarization
 2. **Software variants** = **manually** assign feature names to mined feature implementations
- ✓ Feature documentation = giving a **name / description** for the mined feature implementation
 - ✓ The mined feature implementation **must be documented**
 - For the purpose of constructing a **FM**

Contribution

- We exploit commonality and variability across software variants, at feature implementation and use-cases levels, to apply IR methods in an efficient way
- Our approach gives each feature implementation a name and description based on the use-case name and description
- Feature documentation = Names of the OBE when use-cases are missing
- The REVPLINE documentation approach uses three techniques: FCA, LSI and RCA

Used Technique

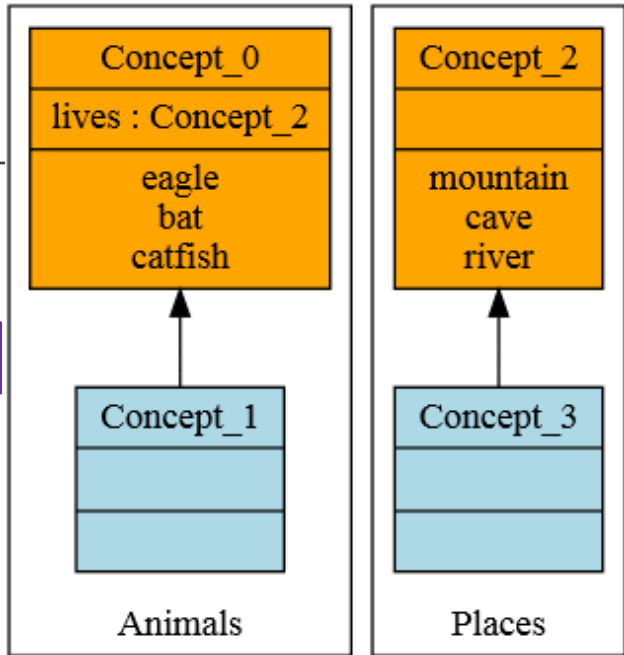
Relational Concept Analysis (RCA)

- ✓ « Objects (in categories) + attributes + relations \Rightarrow classified concepts in several categories »

Relational Context Family

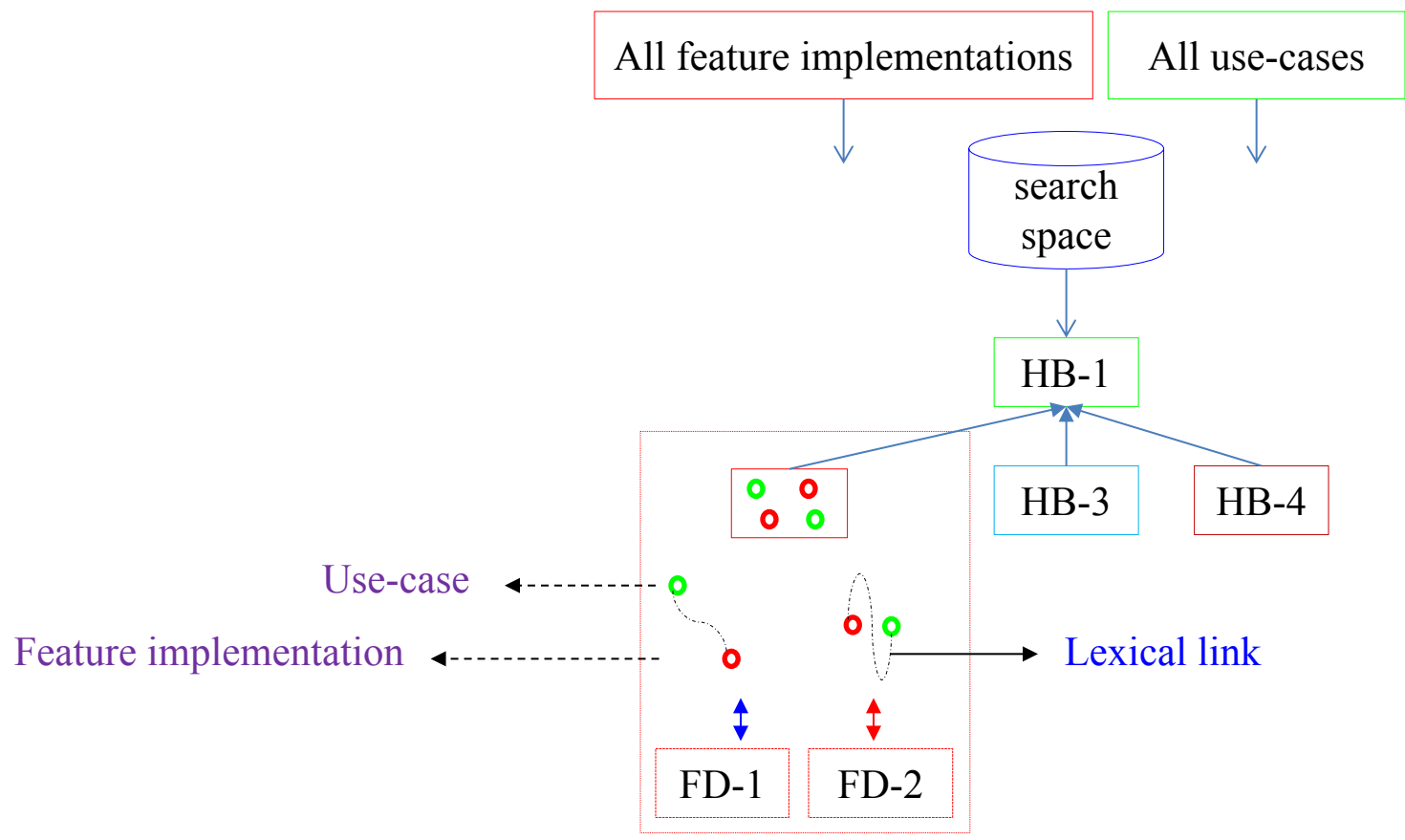
<pre> FormalContext Animals eagle bat catfish FormalContext Places mountain cave river </pre>	<pre> RelationalContext lives source Animals target Places scaling com.googlecode.erca.framework.algo.scaling.Wide mountain cave river eagle x bat x catfish x </pre>
---	---

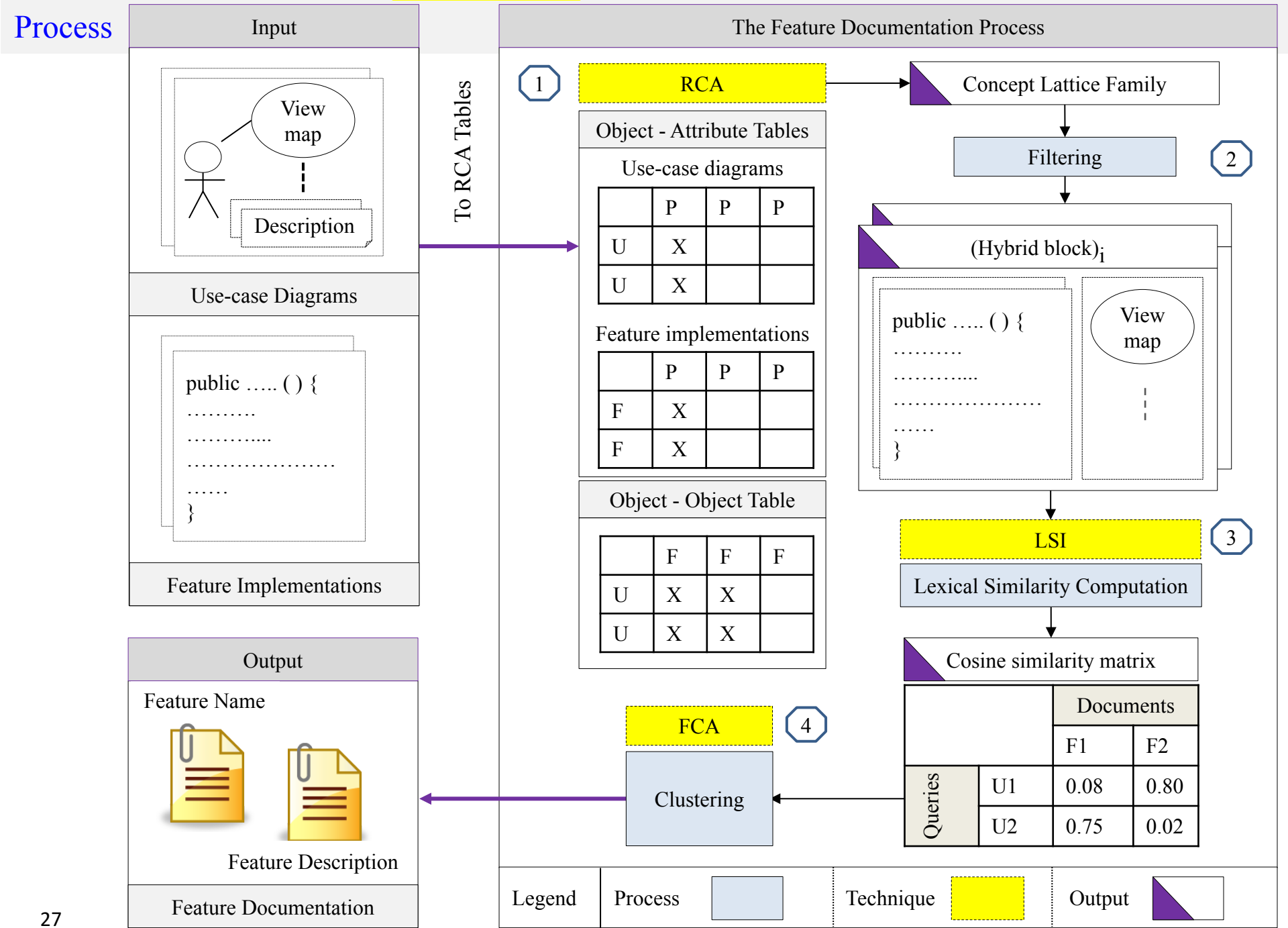
Concept Lattice Family



Key Ideas

- In our work, each use-case represents a feature





Example

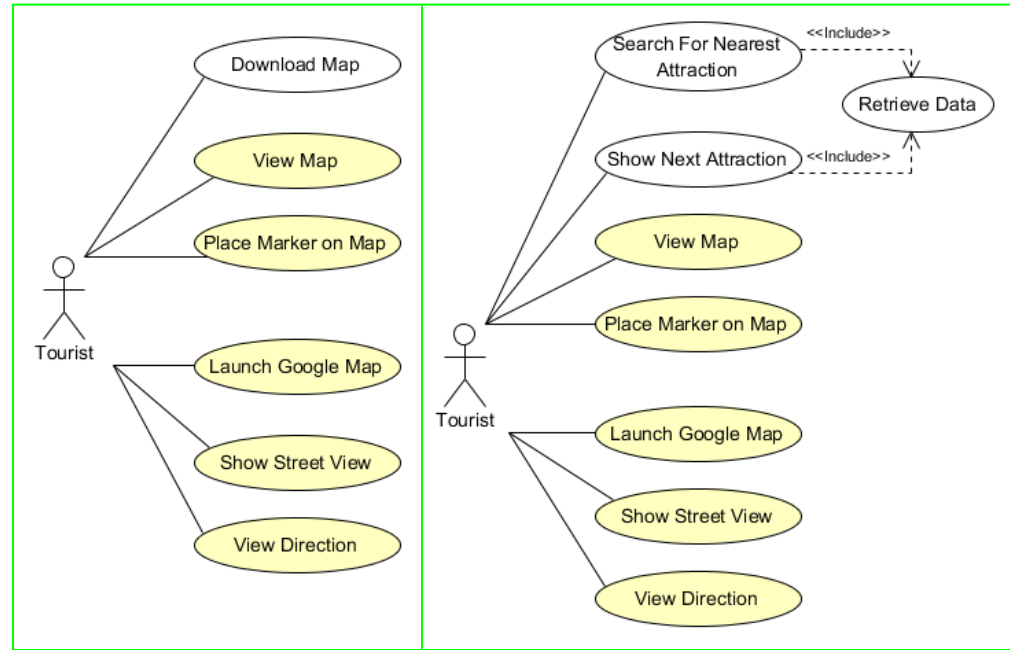
Mobile Tourist Guide (MTG) Software Variants

Mined Feature Implementations	Feature Implementation_1: View map									
	Feature Implementation_2: Place marker on a map									
	Feature Implementation_3: View direction									
	Feature Implementation_4: Launch Google map									
	Feature Implementation_5: Show street view									
	Feature Implementation_6: Download map from Google									
	Feature Implementation_7: Show satellite view									
	Feature Implementation_8: Search for nearest attraction									
	Feature Implementation_9: Show next attraction									
	Feature Implementation_10: Retrieve data									
Mobile Tourist Guide 1	✗	✗	✗	✗	✗					
Mobile Tourist Guide 2	✗	✗	✗	✗	✗	✗				
Mobile Tourist Guide 3	✗	✗	✗	✗	✗		✗			
Mobile Tourist Guide 4	✗	✗	✗	✗	✗			✗	✗	✗

The mined feature implementations from MTG software variants

Use-cases	Software	View map								
		Place marker on a map								
		View direction								
		Launch Google map								
		Show street view								
		Download map from Google								
		Show satellite view								
		Search for nearest attraction								
		Show next attraction								
		Retrieve data								
Mobile Tourist Guide 1	✗	✗	✗	✗	✗					
Mobile Tourist Guide 2	✗	✗	✗	✗	✗	✗				
Mobile Tourist Guide 3	✗	✗	✗	✗	✗		✗			
Mobile Tourist Guide 4	✗	✗	✗	✗	✗			✗	✗	

The use-cases of MTG software variants



The use-case diagrams of the second and fourth MTG software variants

Example



	MTG_1	MTG_2	MTG_3	MTG_4
Use_case_Diagrams				
View Map	X	X	X	X
Launch Google Map	X	X	X	X
View Direction	X	X	X	X
Show Street View	X	X	X	X
Place Marker on Map	X	X	X	X
Download Map		X		
Show Satellite View			X	
Show Next Attraction				X
Search For nearest attraction				X
Retrieve Data				X

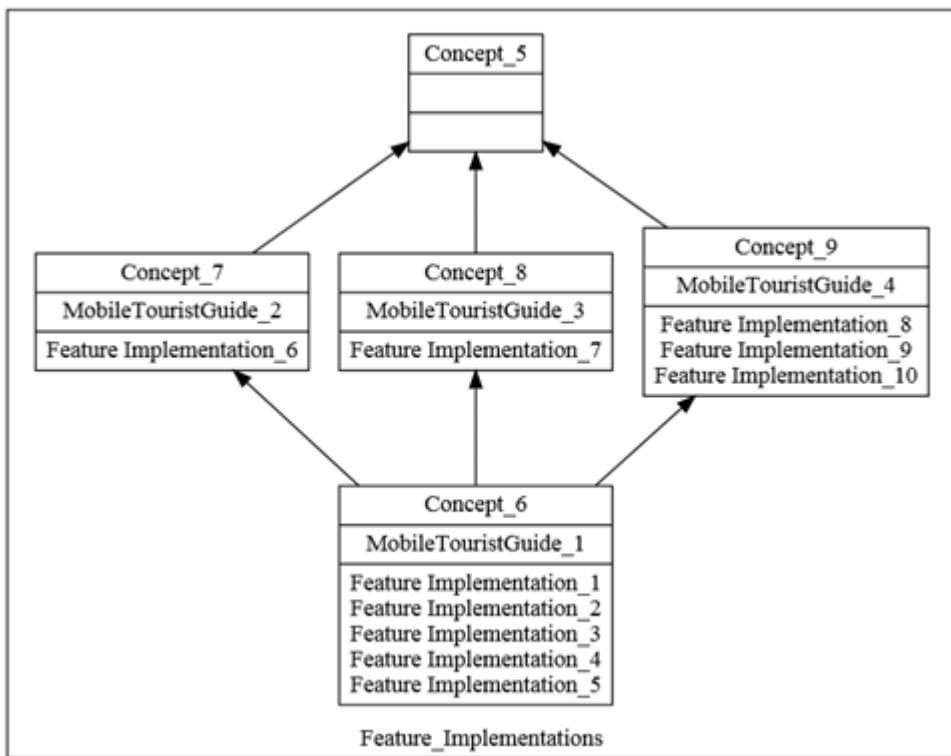
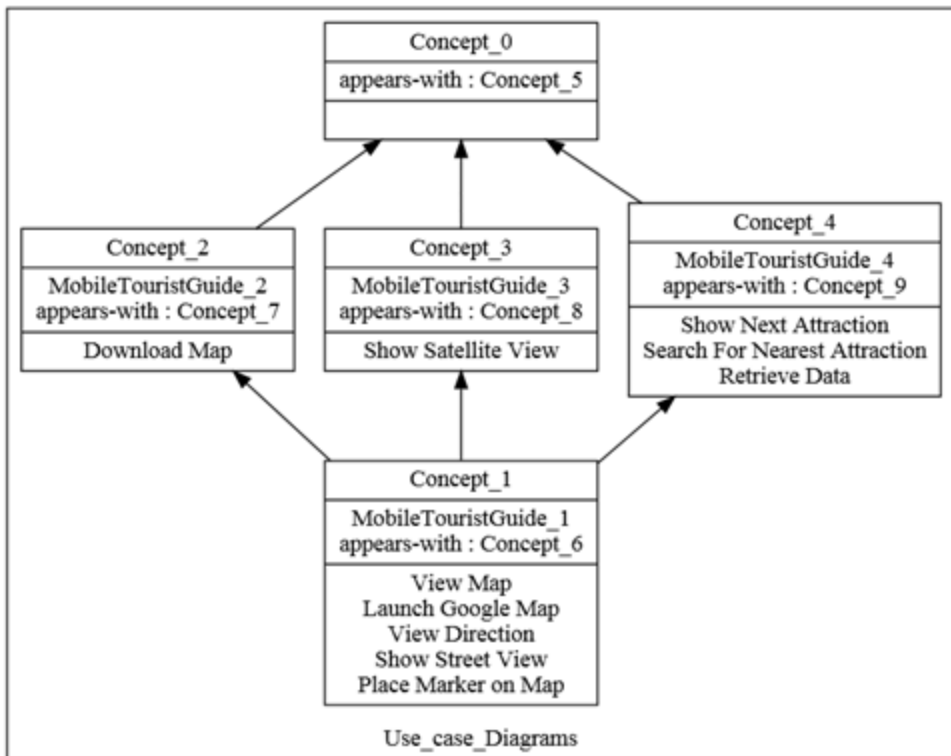


	MTG_1	MTG_2	MTG_3	MTG_4
Feature_Implementations				
Feature Implementation_1	X	X	X	X
Feature Implementation_2	X	X	X	X
Feature Implementation_3	X	X	X	X
Feature Implementation_4	X	X	X	X
Feature Implementation_5	X	X	X	X
Feature Implementation_6		X		
Feature Implementation_7			X	
Feature Implementation_8				X
Feature Implementation_9				X
Feature Implementation_10				X



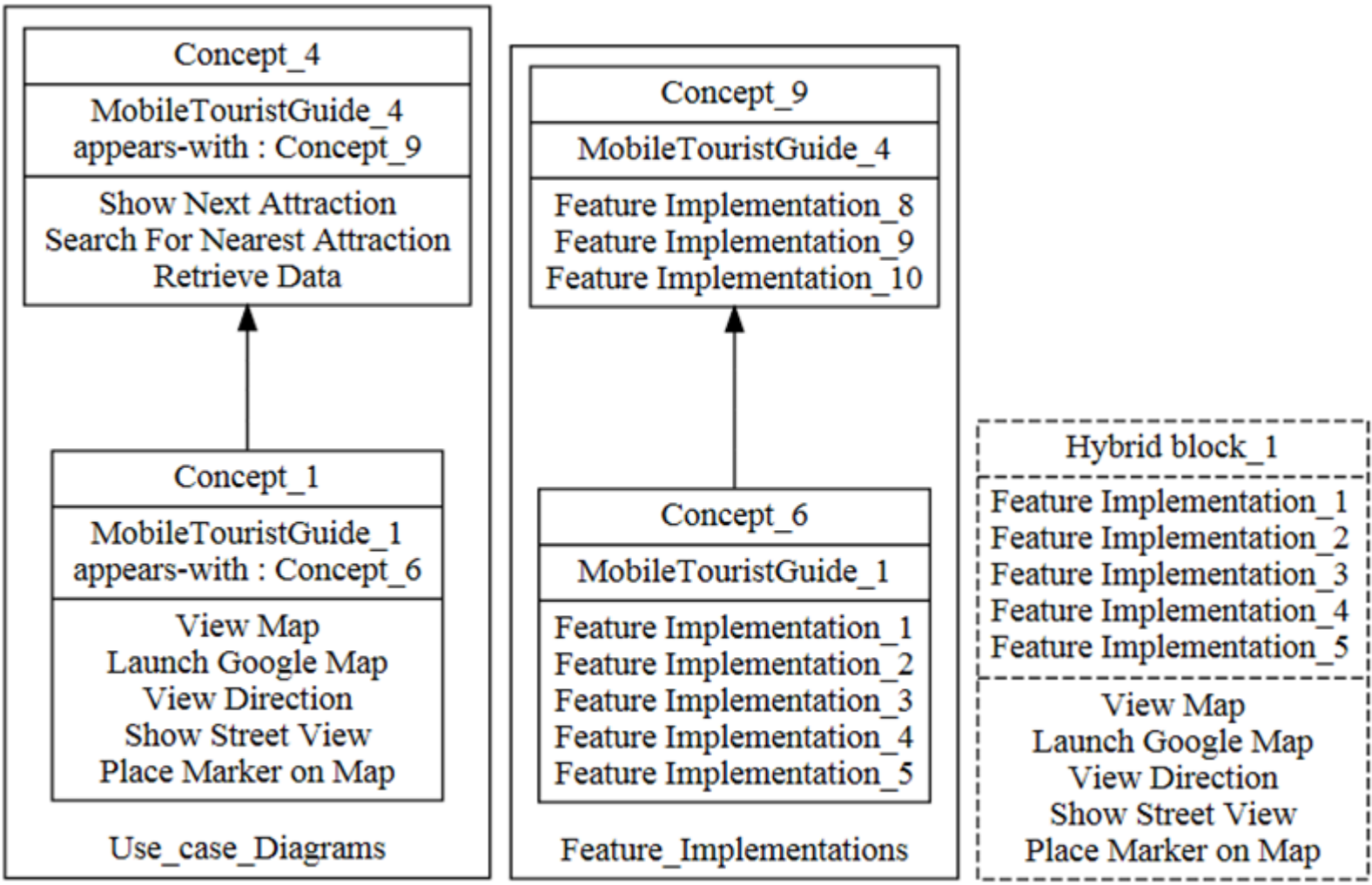
	Feature Implementation_1	Feature Implementation_2	Feature Implementation_3	Feature Implementation_4	Feature Implementation_5	Feature Implementation_6	Feature Implementation_7	Feature Implementation_8	Feature Implementation_9	Feature Implementation_10
Relational context: appears-with										
View Map	X	X	X	X	X					
Launch Google Map	X	X	X	X	X					
View Direction	X	X	X	X	X					
Show Street View	X	X	X	X	X					
Place Marker on Map	X	X	X	X	X					
Download Map						X				
Show Satellite View							X			
Show Next Attraction								X	X	X
Search For Nearest Attraction								X	X	X
Retrieve Data								X	X	X

Example



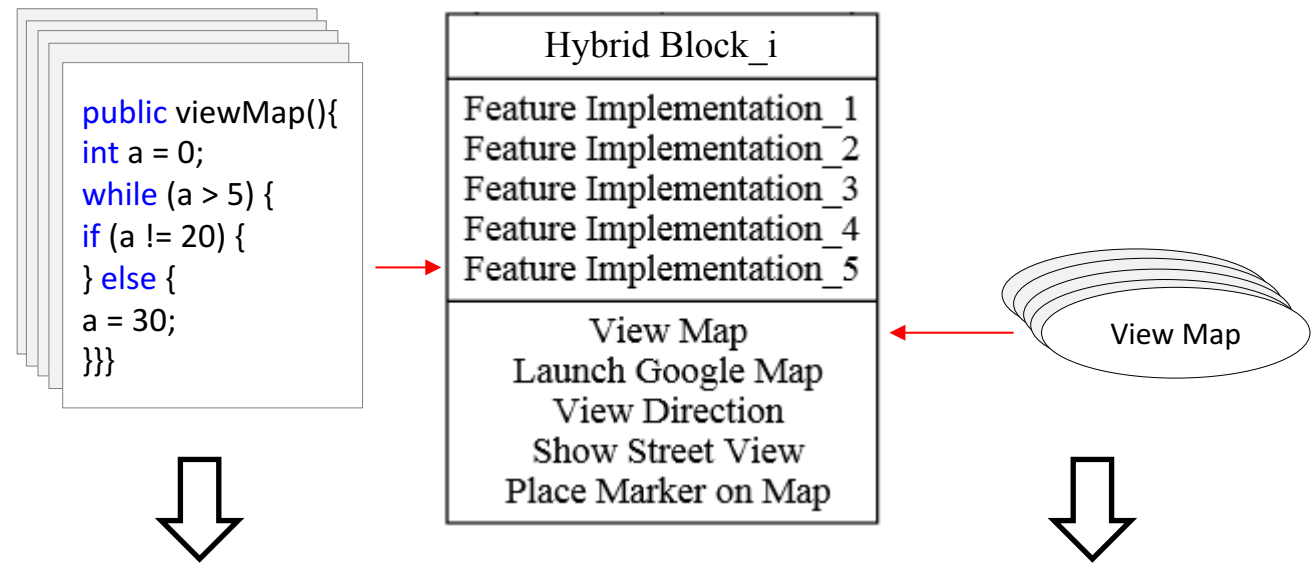
The concept lattice family of relational context family

Example



Exploring and filtering the **hybrid blocks** CLF to identify features documentation

Example



Documents (Feature Implementations)

View Map Implementation

```
viewMap
a
.....
.....
.....
.....
```

Queries (use-cases)

View Map

The tourist can view maps via mobile device

.....

.....

Example

1

	Feature Implement._1	Feature Implement._2	Feature Implement._3	Feature Implement._4	Feature Implement._5
device	1	0	0	0	1
direction	0	0	0	6	0
google	1	0	0	0	0
launch	4	0	0	0	0
map	1	2	0	0	4
marker	0	6	0	0	0
mobile	1	0	0	0	1
place	0	3	0	0	0
show	0	0	2	0	0
street	0	0	5	0	0
tourist	1	1	1	1	1
view	0	0	1	2	5

	Launch Google Map	Place Marker on Map	Show Street View	View Direction	View Map
device	1	0	0	0	1
direction	0	0	0	8	0
google	3	0	0	0	0
launch	3	0	0	0	0
map	2	2	1	1	5
marker	0	3	0	0	0
mobile	1	0	0	0	1
place	0	3	0	0	0
show	0	0	3	0	0
street	0	0	5	0	0
tourist	1	1	1	1	1
view	0	0	1	3	5

2

	Feature Implementation_1	Feature Implementation_2	Feature Implementation_3	Feature Implementation_4	Feature Implementation_5
Launch Google Map	0.861933577	0.0137010	0	0	0.152407
Place Marker on Map	0.01114798	0.9480070	0	0	0.085939
Show Street View	0.004088722	0.0051128	0.98581691	0.00571	0.070920
View Direction	0.00296571	0.0037085	0.0069484	0.999139665	0.108597
View Map	0.114676597	0.0627020	0.039159941	0.070025418	0.993111

➡ The term-document matrix

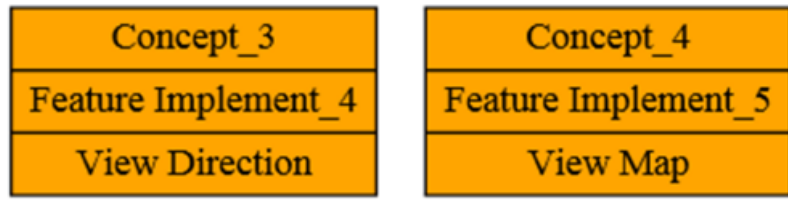
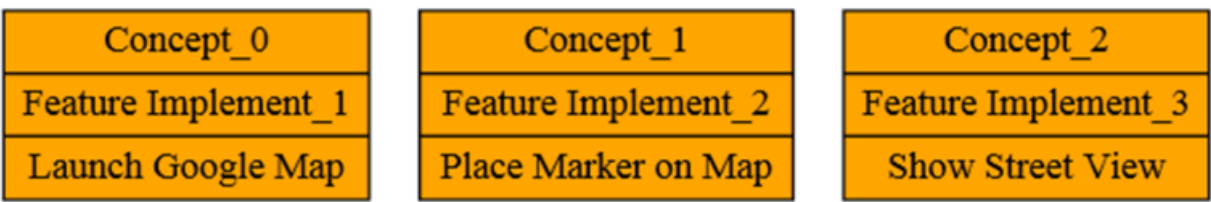
➡ The term-query matrix

The cosine similarity matrix

3

	Feature Implementation_1	Feature Implementation_2	Feature Implementation_3	Feature Implementation_4	Feature Implementation_5
Launch Google Map	✗				
Place Marker on Map		✗			
Show Street View			✗		
View Direction				✗	
View Map					✗

Formal context



The documented features

Example

<div style="text-align: right;">Features</div> <div style="text-align: left;">Software</div>	View map	Place marker on a map	View direction	Launch Google map	Show street view	Download map from Google	Show satellite view	Search for nearest attraction	Show next attraction	Retrieve data
Mobile Tourist Guide 1	X	X	X	X	X					
Mobile Tourist Guide 2	X	X	X	X	X	X				
Mobile Tourist Guide 3	X	X	X	X	X		X			
Mobile Tourist Guide 4	X	X	X	X	X			X	X	X

The *product-by-feature matrix* for MTG software variants

Using Identifier Names

- Naming Feature Implementation Based on OBE Names:
 1. Extracting and tokenizing OBE names from the identified feature implementation
 2. Weighting tokens
 3. Constructing the feature name

OBE names, tokens, weight and strongest weighted tokens for *show street view* feature implementation

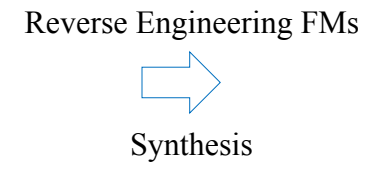
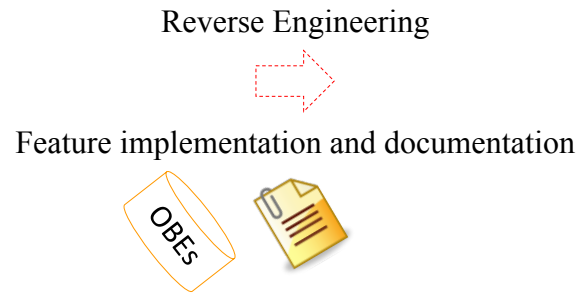
OBE Name	Token/Weight			
	T1/ w=1.0	T2/ w=0.7	T3/ w=0.5	T4/ w=0.5
ShowStreetView	show	Street	View	
StreetPosition	Street	Position		
ChangeStreetSettings	Change	Street	Settings	
getStreetAddress	get	Street	Address	
setStreetAddress	set	Street	Address	
ShowNearestStreet	show	Nearest	Street	
ShowNextStreet	show	Next	Street	
retrieveStreetData	retrieve	Street	Data	
ShowStreet	show	Street		
updateStreetInfo	update	Street	Info	
ViewStreetMap	View	Street	Map	
ViewStreetPositionInfo	View	Street	Position	Info

Token	Total Weight	Top 3	Top 4
Show	4	✗	✗
Street	8	✗	✗
View	2.5	✗	✗
Position	1.2		✗
Change	1		
Settings	1		
get	1		
Address	1		
set	1		
Nearest	0.7		
Next	0.7		
retrieve	1		
Data	0.5		
update	1		
Info	1		
Map	0.5		

✓ The proposed name = [StreetShowView](#)

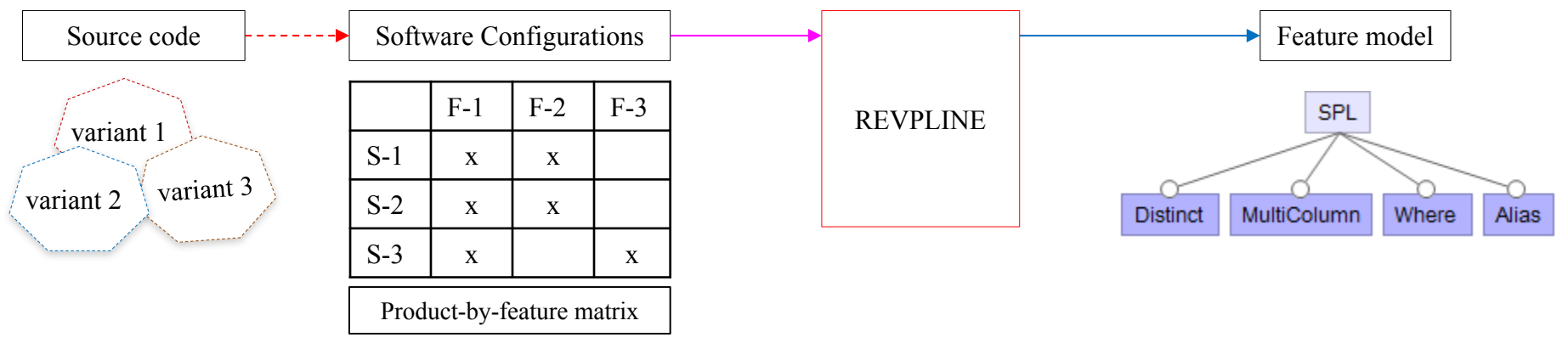
Reverse Engineering Feature Models from Software Configurations

Proposal



Inputs

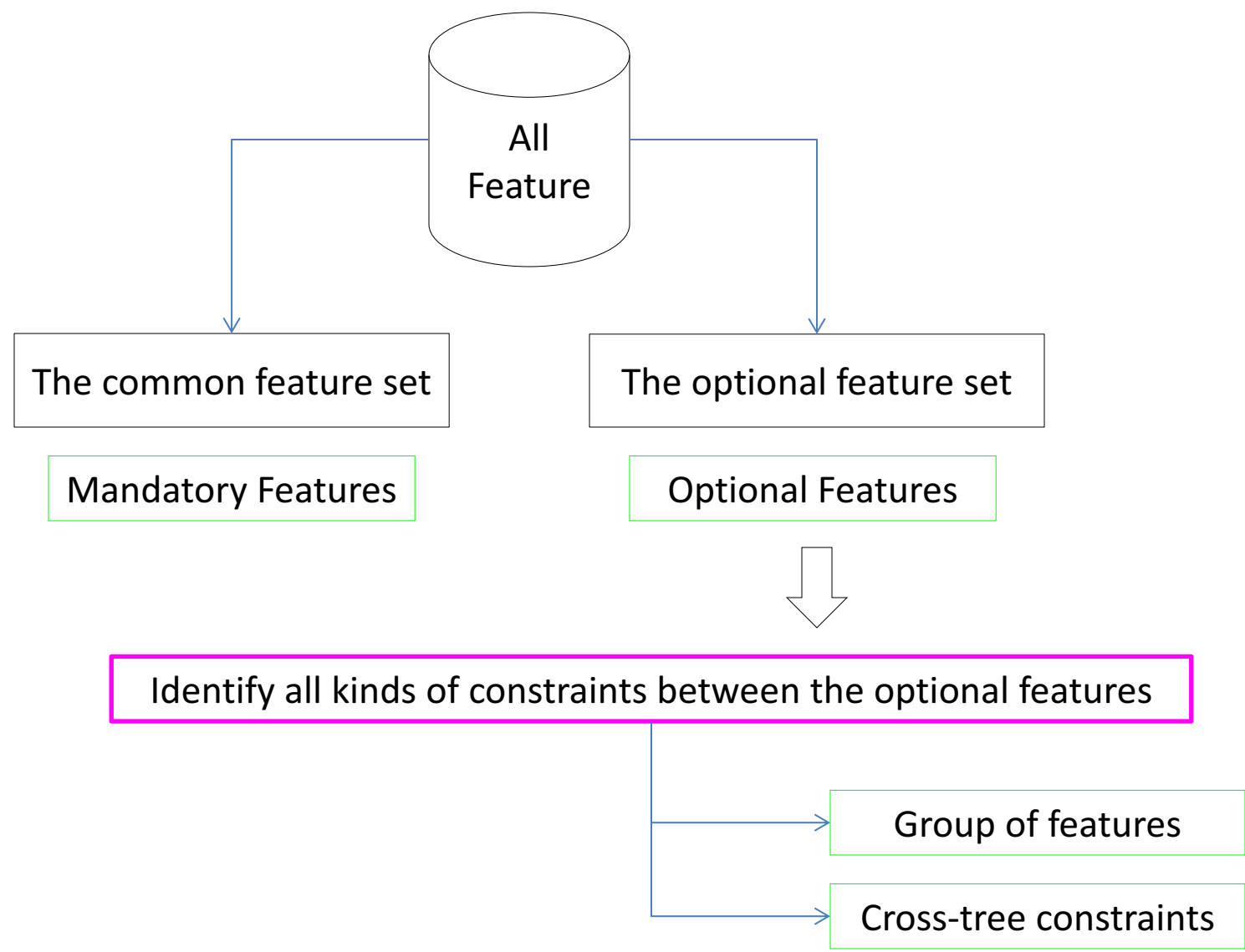
Output



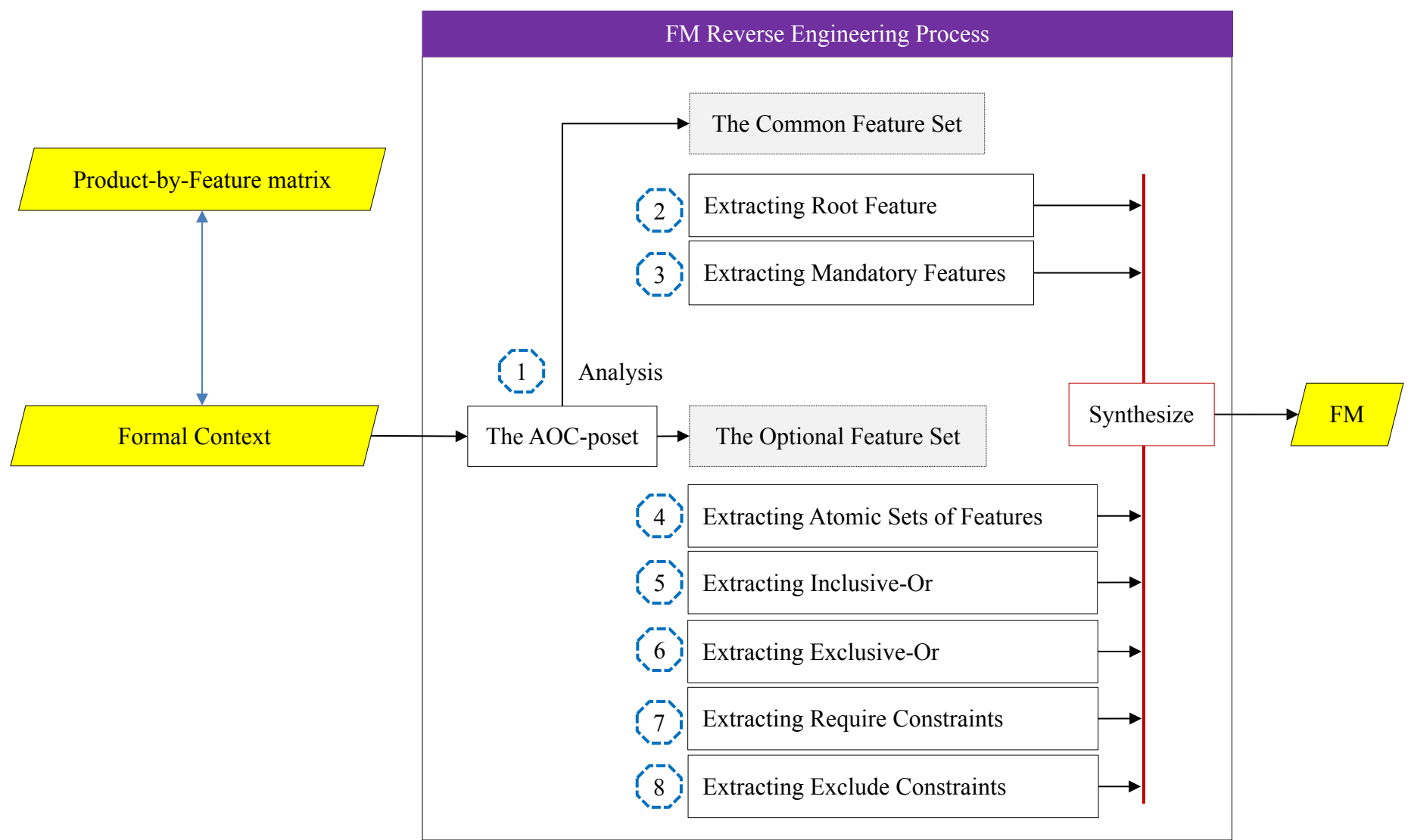
Contribution

- Automatic approach to organize the mined and documented features into a FM
- Features are organized in a tree which highlights
 - ✓ Mandatory features
 - ✓ Optional features
 - ✓ Feature groups (and, or, xor groups)
 - ✓ cross-tree constraints: require and exclude constraints
- We rely on FCA and software configurations
- The FMs are generated in very short time

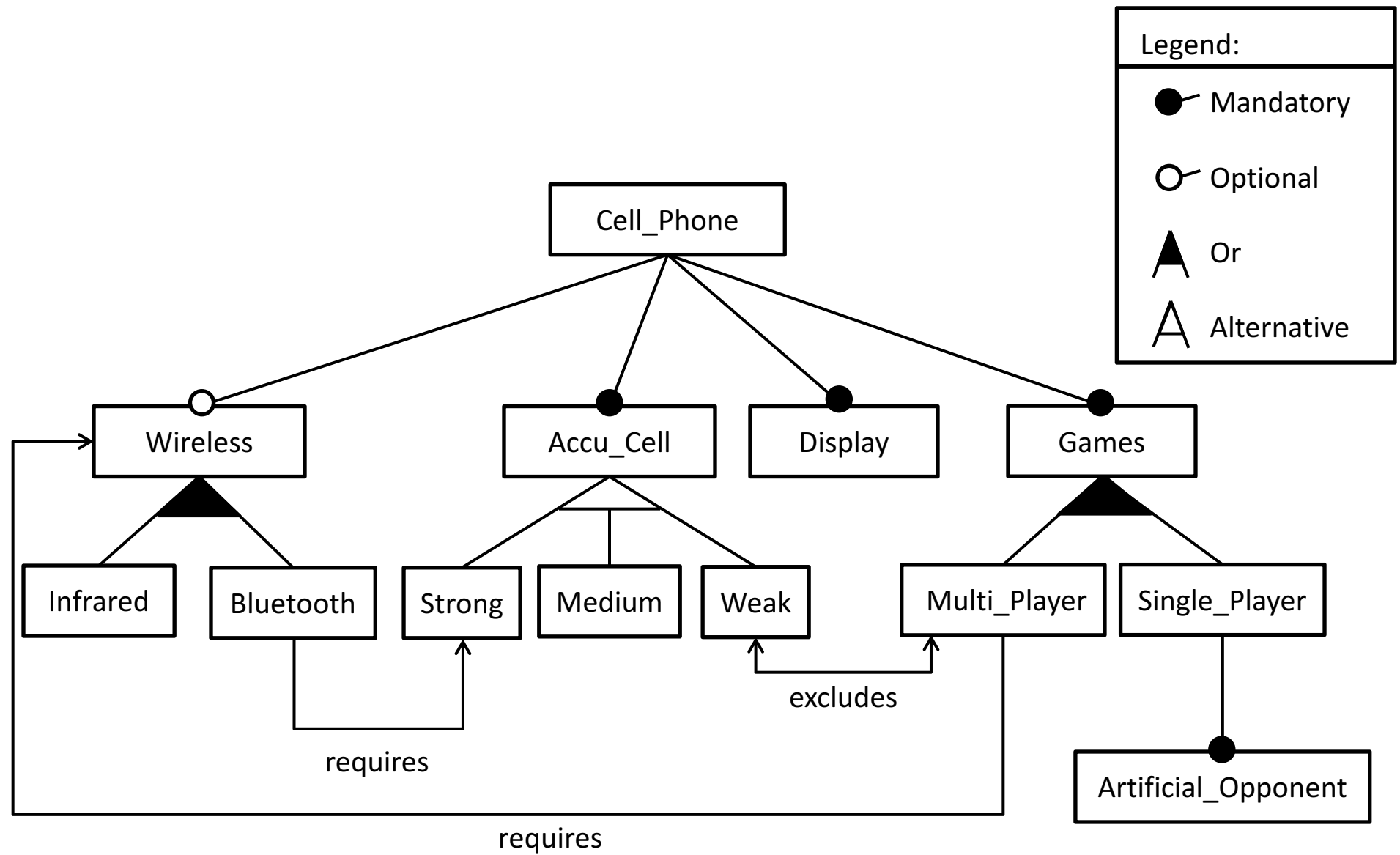
Key Ideas



Process



Example



Legend:

- Mandatory
- Optional
- ▲ Or
- △ Alternative

Existing Cell phone SPL FM

Example

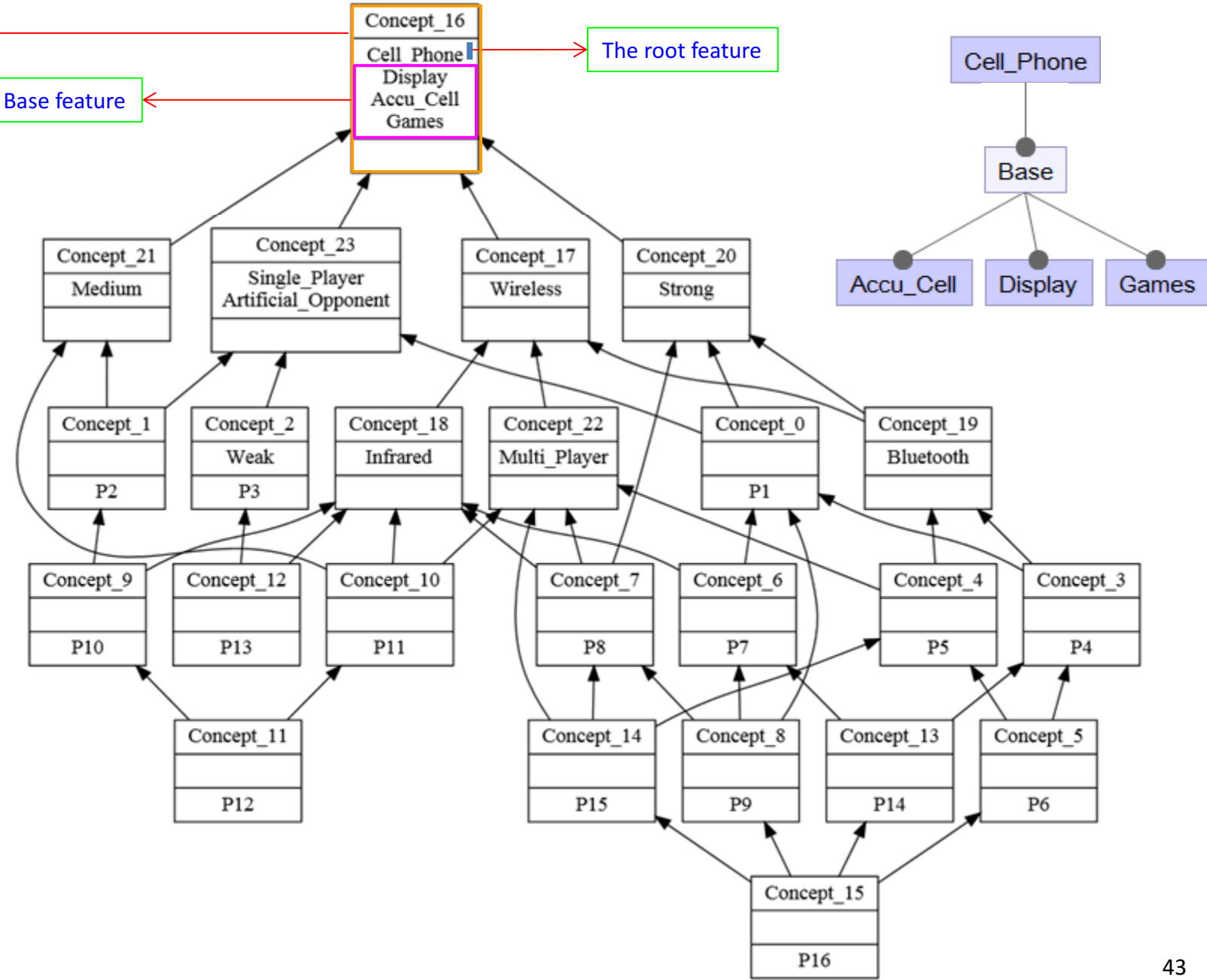
Product configurations	Cell_Phone	Wireless	Infrared	Bluetooth	Accu_Cell	Strong	Medium	Weak	Display	Games	Multi_Player	Single_Player	Artificial_Opponent
Product-1	X	X	X		X	X			X	X	X		
Product-2	X	X		X	X	X			X	X	X		
Product-3	X	X	X	X	X	X			X	X	X		
Product-4	X	X	X		X		X		X	X	X		
Product-5	X				X			X	X	X		X	X
Product-6	X				X	X			X	X		X	X
Product-7	X	X	X		X	X			X	X		X	X
Product-8	X	X		X	X	X			X	X		X	X
Product-9	X	X	X	X	X	X			X	X		X	X
Product-10	X				X		X		X	X		X	X
Product-11	X	X	X		X		X		X	X		X	X
Product-12	X	X	X		X			X	X	X		X	X
Product-13	X	X	X		X		X		X	X	X	X	X
Product-14	X	X	X		X	X			X	X	X	X	X
Product-15	X	X		X	X	X			X	X	X	X	X
Product-16	X	X	X	X	X	X			X	X	X	X	X

Example

The top concept

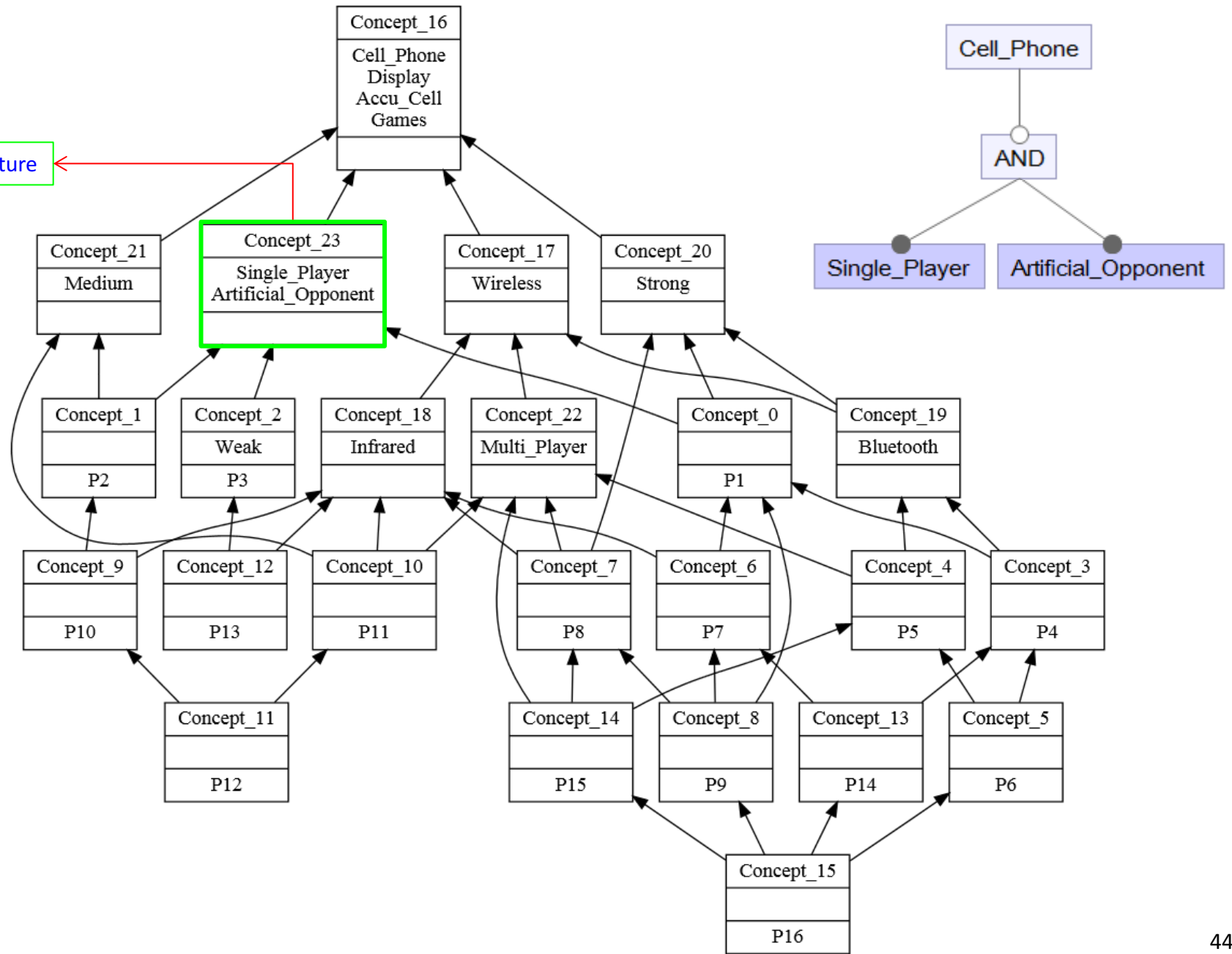
The root feature

The Base feature

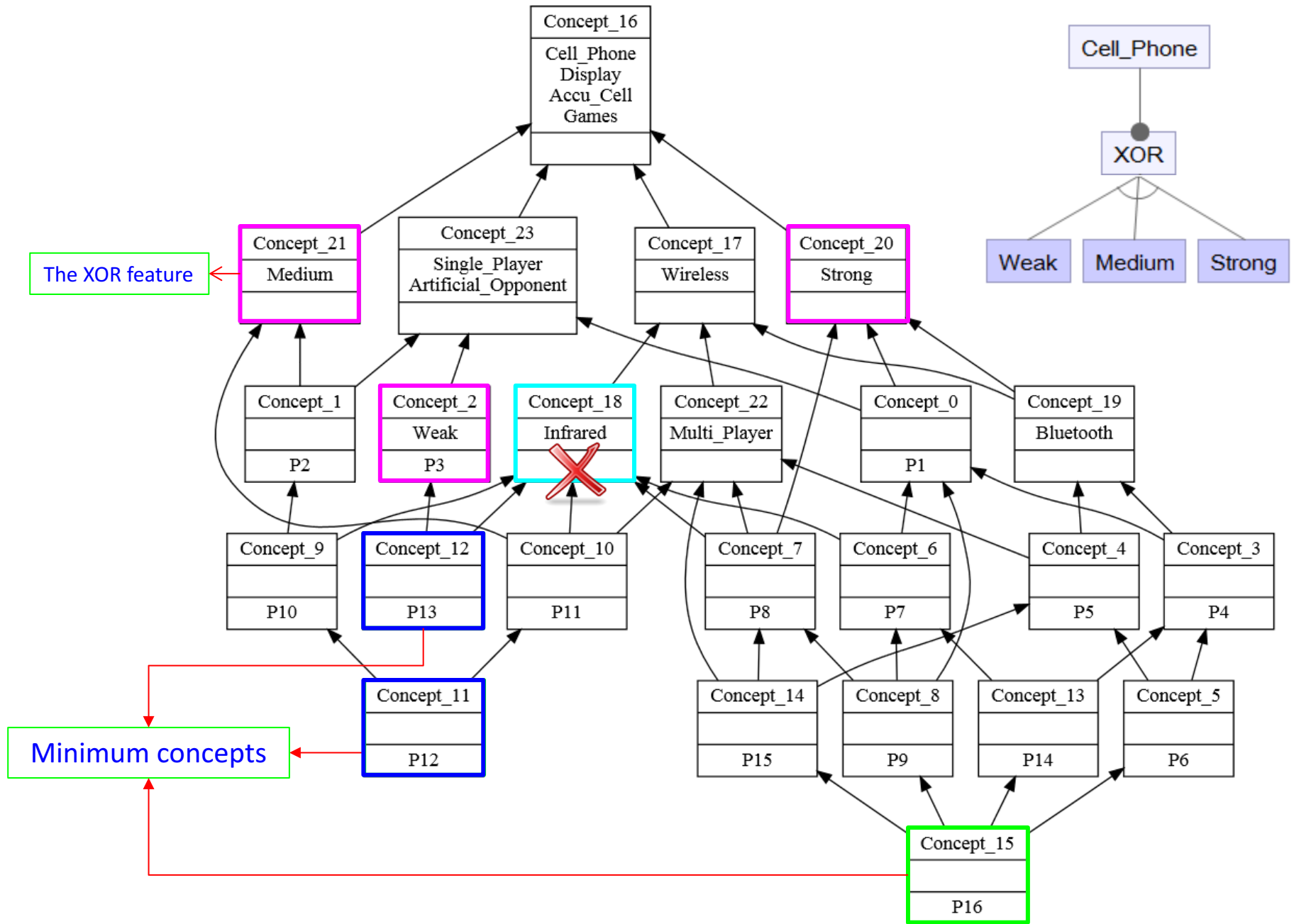


Example

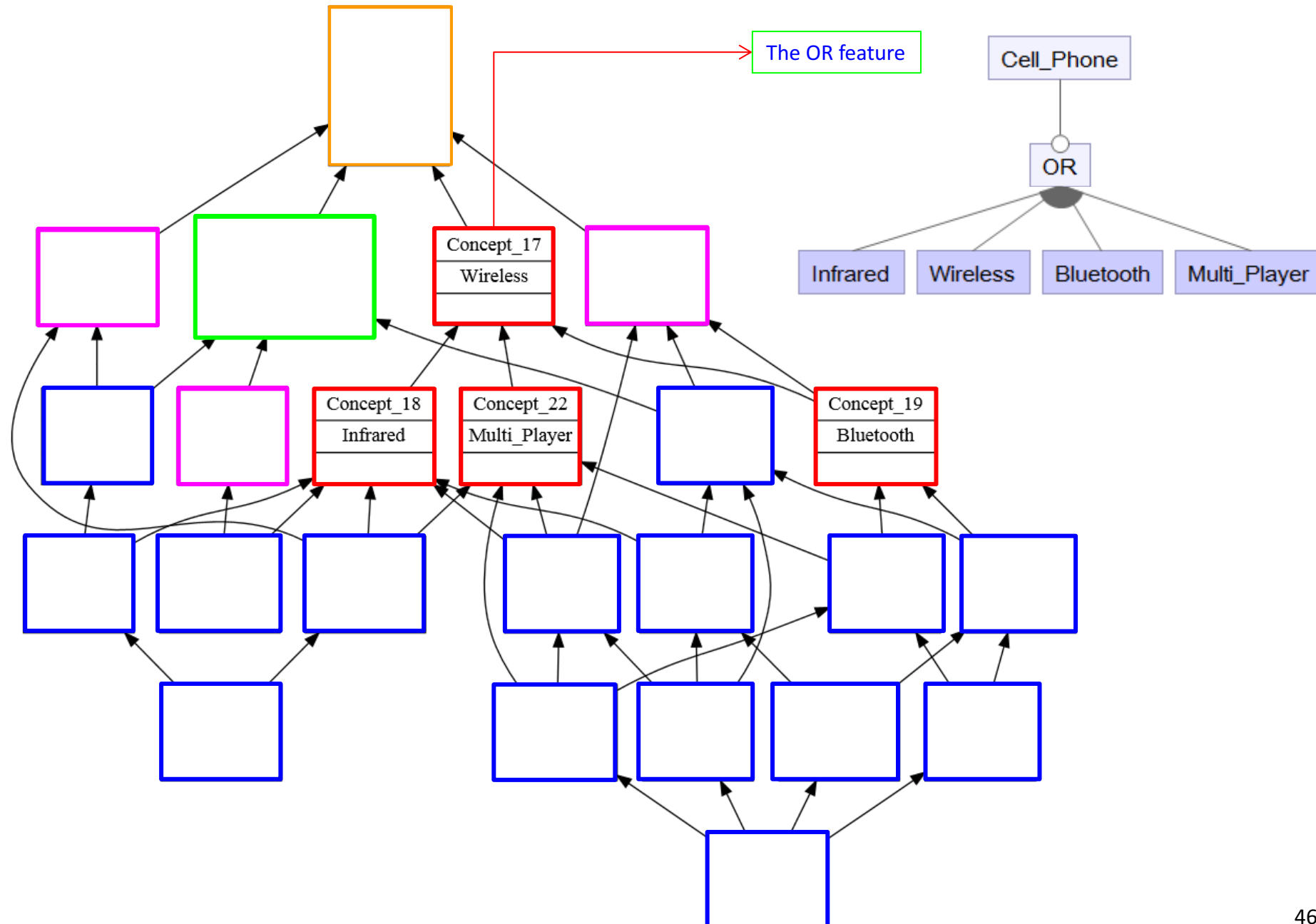
The AND feature



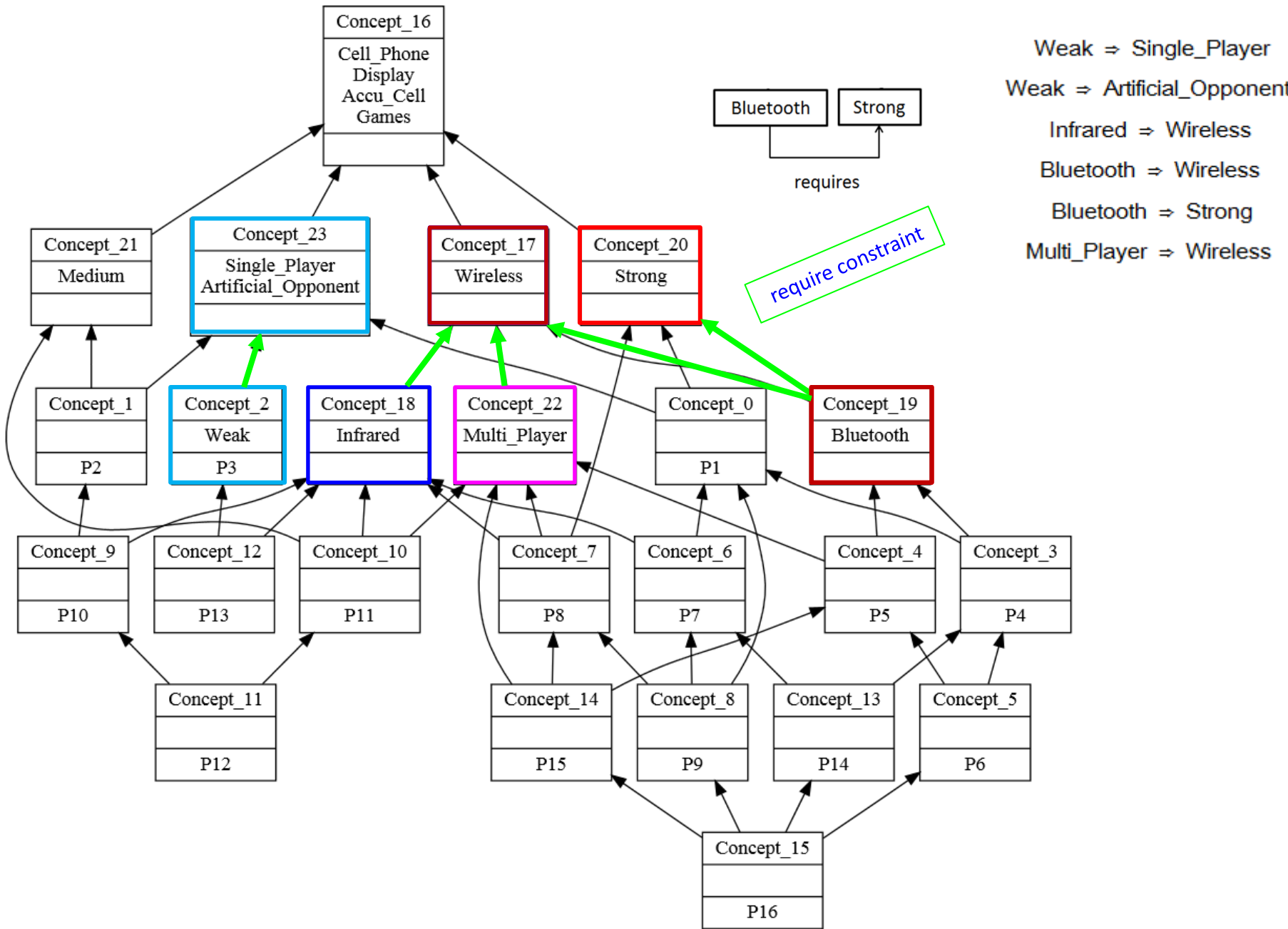
Example



Example

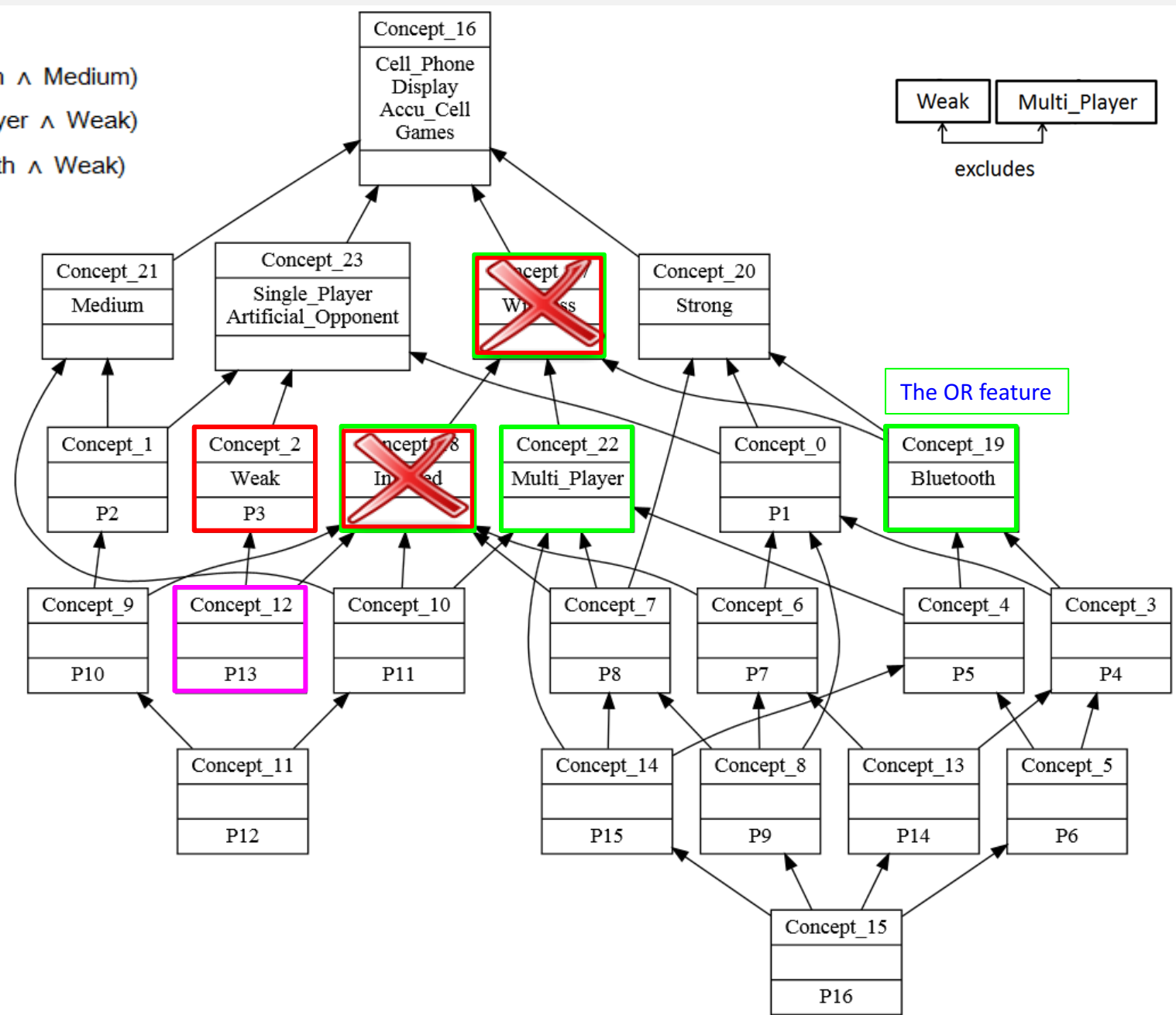


Example

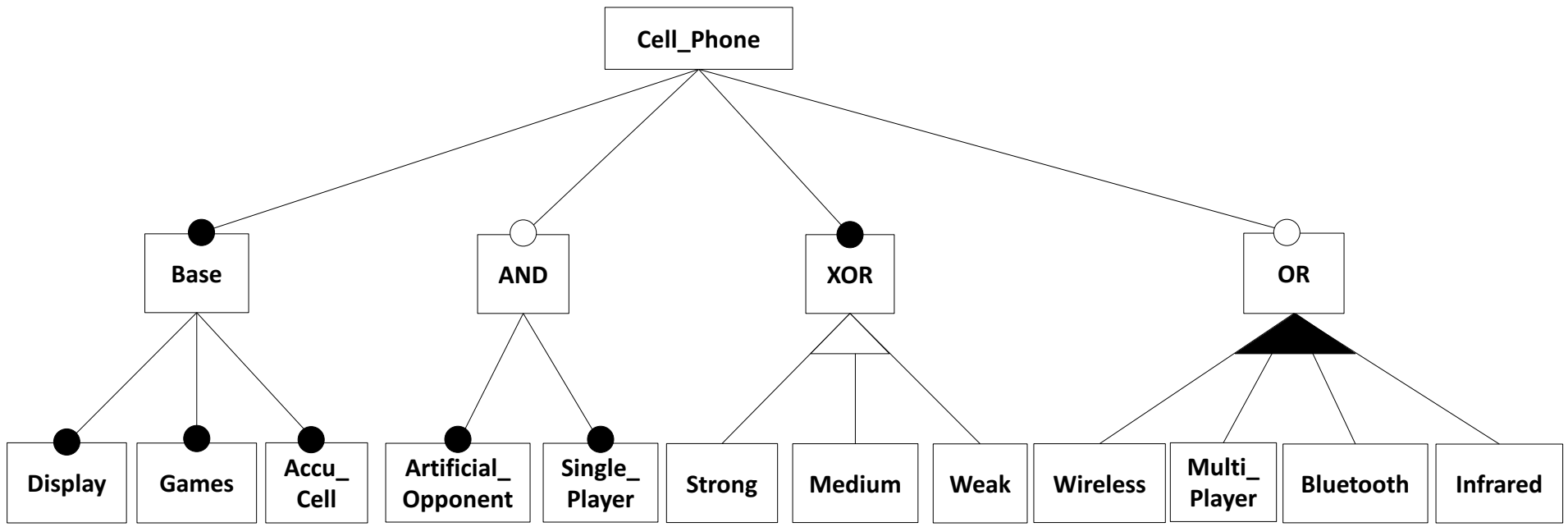


Example

- ¬ (Bluetooth ∧ Medium)
- ¬ (Multi_Player ∧ Weak)
- ¬ (Bluetooth ∧ Weak)



The mined FM



- Weak \Rightarrow Single_Player
- Weak \Rightarrow Artificial_Opponent
- Infrared \Rightarrow Wireless
- Bluetooth \Rightarrow Wireless
- Bluetooth \Rightarrow Strong
- Multi_Player \Rightarrow Wireless
- \neg (Bluetooth \wedge Medium)
- \neg (Multi_Player \wedge Weak)
- \neg (Bluetooth \wedge Weak)

FM Evaluation

Product configurations	Cell_Phone	Wireless	Infrared	Bluetooth	Accu_Cell	Strong	Medium	Weak	Display	Games	Multi_Player	Single_Player	Artificial_Opponent
Product-1	X	X	X		X	X			X	X	X		
Product-2	X	X		X	X	X			X	X	X		
Product-3	X	X	X	X	X	X			X	X	X		
Product-4	X	X	X		X		X		X	X	X		
Product-5	X				X			X	X	X		X	X
Product-6	X				X	X			X	X		X	X
Product-7	X	X	X		X	X			X	X		X	X
Product-8	X	X		X	X	X			X	X		X	X
Product-9	X	X	X	X	X	X			X	X		X	X
Product-10	X				X		X		X	X		X	X
Product-11	X	X	X		X		X		X	X		X	X
Product-12	X	X	X		X			X	X	X		X	X
Product-13	X	X	X		X		X		X	X	X	X	X
Product-14	X	X	X		X	X			X	X	X	X	X
Product-15	X	X		X	X	X			X	X	X	X	X
Product-16	X	X	X	X	X	X			X	X	X	X	X
Product-17	X				X	X			X	X			
Product-18	X	X			X	X			X	X			
Product-19	X	X	X		X	X			X	X			
Product-20	X	X		X	X	X			X	X			
Product-21	X	X	X	X	X	X			X	X			
Product-22	X				X		X		X	X			
Product-23	X	X			X		X		X	X			
Product-24	X	X	X		X		X		X	X			
Product-25	X	X			X	X			X	X	X		
Product-26	X	X			X		X		X	X	X		
Product-27	X	X			X	X			X	X		X	X
Product-28	X	X			X		X		X	X		X	X
Product-29	X	X			X			X	X	X		X	X
Product-30	X	X			X	X			X	X	X	X	X
Product-31	X	X			X		X		X	X	X	X	X



	Evaluation Metrics		
	Precision	Recall	F-Measure
Value	0.51	1	0.68

« In our approach, feature selection constraints are not detected »

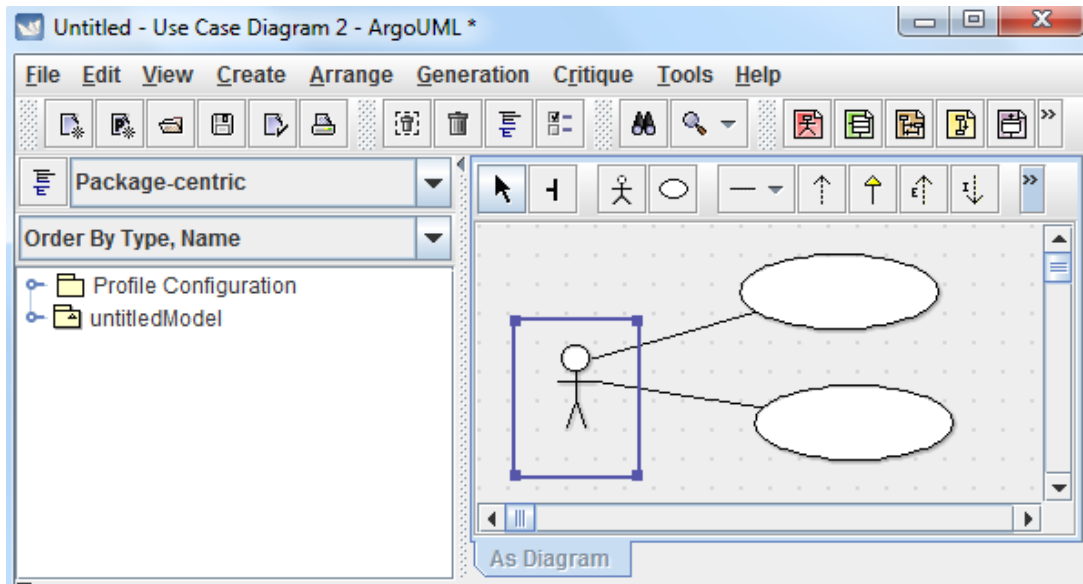
Experimentation and Threats to Validity

Experimentation

- ✓ **ArgoUML-SPL** = real SPL, 10 products, large systems, Java, well documented
 - ✓ **Health complaint-SPL** = real SPL, 10 products, medium systems, Java, well documented
 - ✓ **Mobile Media** = real software variants, 4 products, small systems, Java, well documented
- Evaluation Metrics: precision, recall and F-Measure

ArgoUML SPL

ArgoUML screenshot



Product #	ArgoUML Product Description	LoC	NoP	NoC	Number of OBEs
P1	All optional features disabled	82,924	55	1,243	74,444
P2	All optional features enabled	120,348	81	1,666	100,420
P3	Only logging feature disabled	118,189	81	1,666	98,988
P4	Only cognitive feature disabled	104,029	73	1,451	89,273
P5	Only sequence diagram disabled	114,969	77	1,608	96,492
P6	Only use-case diagram disabled	117,636	78	1,625	98,468
P7	Only deployment diagram disabled	117,201	79	1,633	98,323
P8	Only collaboration diagram disabled	118,769	79	1,647	99,358
P9	Only state diagram disabled	116,431	81	1,631	97,760
P10	Only activity diagram disabled	118,066	79	1,648	98,777

Health Complaint SPL

HealthWatcher - PublicHealthComplaint SPL

Select operation:

[Insert a new complaint](#)[Queries' menu](#)[Employee login](#)[RSS feeds](#)*HealthWatcher - PublicHealthComplaint SPL* by [SED](#) - IC - Unicamp - 2011

Health Complaint screenshot

Product #	Health complaint Product Description	LOC	NOP	NOC	Number of OBEs
P1	Base - no extensions applied	5,288	22	88	6,603
P2	Command pattern applied	5,646	23	92	6,867
P3	State pattern applied	6,112	24	104	7,407
P4	Observer pattern applied	6,222	26	106	7,536
P5	Adapter pattern applied v1	6,379	26	108	7,631
P6	Abstract factory pattern applied v1	6,417	27	112	7,659
P7	Adapter pattern applied v2	6,441	27	116	7,648
P8	Abstract factory pattern applied v2	6,468	28	120	7,669
P9	Evolution - New functionality added	7,709	28	132	9,079
P10	Exception handling applied	7,591	29	135	9,084

Mobile Media software product variants

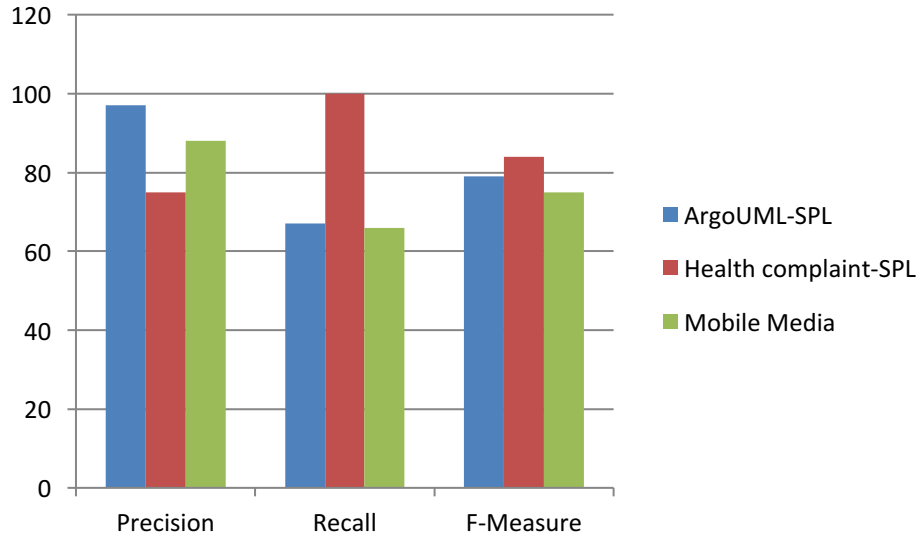


Mobile Media screenshots

Product #	Mobile Media Product Description	LoC	NoP	NoC	Number of OBEs
P1	Mobile photo - Base	936	7	15	822
P2	Exception handling included	1,213	8	24	925
P3	Sorting photos/edit photo label included	1,422	8	26	1,040
P4	New feature added to manage favourites	1,484	8	25	1,066

Feature Location

- Results show that the precision metric appears high
- Results show that the recall metric appears so high
- We cannot use a fixed number of topics for LSI



Average of evaluation metrics for feature location

Feature Location

- ✓ The lexical and structural similarity approach gives better results than the lexical approach alone

Comparing the two ways: lexical versus structural and lexical similarity

ArgoUML-SPL	Precision	Recall	F-Measure	Number of Junctions
Lexical similarity	97%	67%	79%	11
Lexical and structural similarity	97%	100%	98%	1

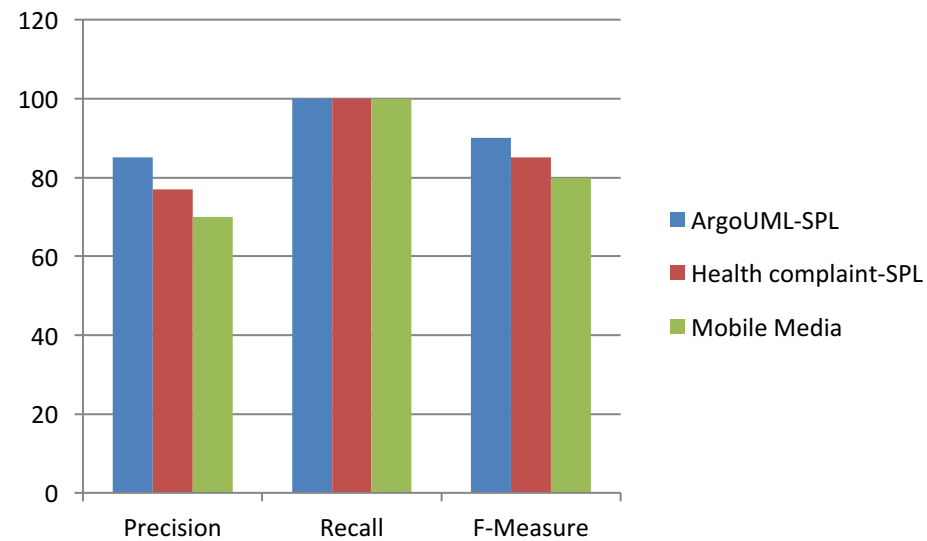
```
Package (argouml.uml.diagram.sequence.ui)
Class (ModeChangeHeight_argouml.uml.diagram.sequence.ui)
Attribute (serialVersionUID_SequenceDiagramGraphModel)
Method (initialize()_PropPanelActionSequence)
Local Variable (lay_UMLSequenceDiagram())
Class (ActionSetOperation_sequence2.diagram)
Method Invocation (info ["SequenceDiagram Module enabled."]._enable())
Method (relocate(base)_UMLSequenceDiagram)
Method Invocation (debug ["Created sequence diagram"]_UMLSequenceDiagram())
```

·
·
·

Part of the sequence diagram feature implementation

Feature Documentation

- Results show that recall value in all cases is 100%
- Results show that precision value either 100% or 50%
- Number of topics for LSI = # of feature implementations



Average of evaluation metrics for Feature Documentation

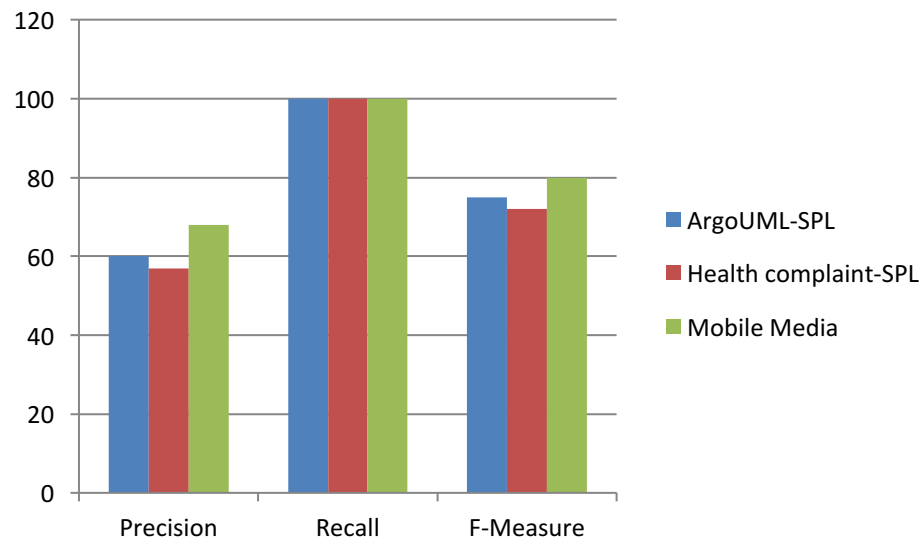
Feature Documentation

Examples (Feature Name and Description)

- ✓ **Use-case diagram:** "a use-case is a set of scenarios that describes an interaction between a user and a system. A use-case diagram displays the relationship among actors and use-cases. The two main components of a use-case diagram are use-cases and actors"
- ✓ **View sorted photos:** "the device sorts the photos based on the number of times photo has been viewed"
- ✓ **Specify food complaint:** "this use case allows a citizen to register a food complaint. The food complaint has the following information: food complaint data, description and observations"

FM Reverse Engineering

- Considering the recall metric, its value is 100% for all case studies
- Results show that precision appears to be not very high for all case studies



Evaluation metrics for FM Reverse Engineering

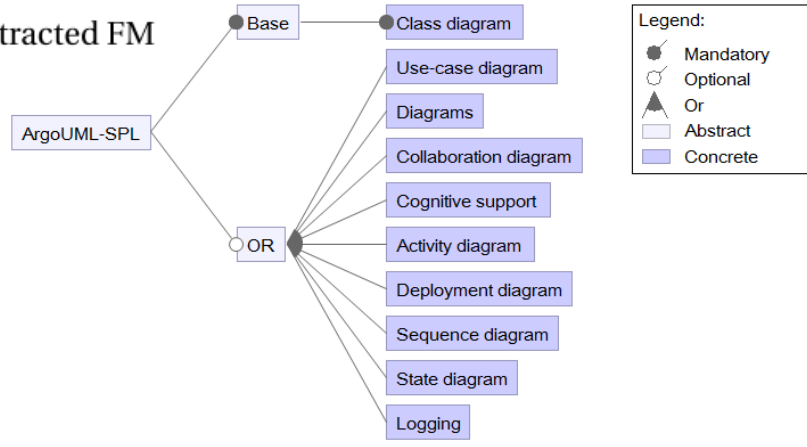
FM Reverse Engineering

The Product-by-feature matrix for ArgoUML software product variants

1	ArgoUML-SPL	Class diagram	Diagrams	Use-case diagram	Collaboration diagram	Cognitive support	Activity diagram	Deployment diagram	Sequence diagram	State diagram	Logging
Product-1	✘	✘									
Product-2	✘	✘	✓	✓	✓	✓	✓	✓	✓	✓	✓
Product-3	✘	✘	✓	✓	✓	✓	✓	✓	✓	✓	
Product-4	✘	✘	✓	✓	✓		✓	✓	✓	✓	✓
Product-5	✘	✘	✓	✓	✓	✓	✓	✓		✓	✓
Product-6	✘	✘	✓		✓	✓	✓	✓	✓	✓	✓
Product-7	✘	✘	✓	✓	✓	✓	✓		✓	✓	✓
Product-8	✘	✘	✓	✓		✓	✓	✓	✓	✓	✓
Product-9	✘	✘	✓	✓	✓	✓	✓	✓	✓		✓
Product-10	✘	✘	✓	✓	✓	✓		✓	✓	✓	✓

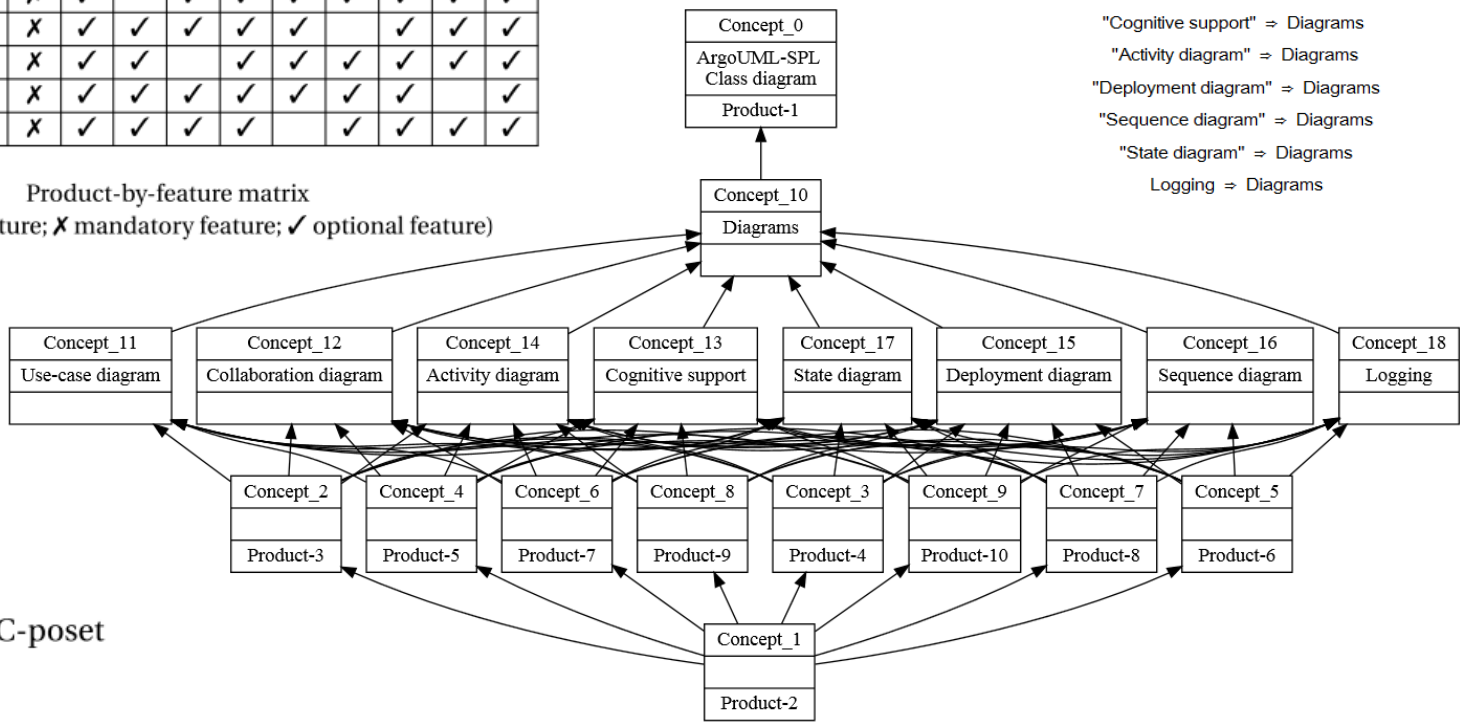
3

The extracted FM



"Use-case diagram" => Diagrams
 "Collaboration diagram" => Diagrams
 "Cognitive support" => Diagrams
 "Activity diagram" => Diagrams
 "Deployment diagram" => Diagrams
 "Sequence diagram" => Diagrams
 "State diagram" => Diagrams
 Logging => Diagrams

Product-by-feature matrix (✘ the root feature; ✘ mandatory feature; ✓ optional feature)



2

The AOC-poset

Reverse Engineering FMs from Samples of Program Configurations

#	case study	Number of Products	Number of Features	Group of Features				CTCs		Algorithm execution times \ (in ms)	Evaluation Metrics		
				Base	Atomic Set of Features	Inclusive-or	Exclusive-or	Requires	Excludes		Precision	Recall	F-Measure
1	Video on demand	16	12	X	X	X		X		572	66%	100%	80%
2	Wiki engines	8	21	X	X	X	X	X	X	555	54%	100%	70%
3	Graph product line	8	18	X		X	X	X	X	551	62%	100%	76%
4	Berkeley DB	10	43	X	X	X	X	X	X	661	50%	100%	66%
5	Mobile phone	5	5	X		X		X		406	70%	100%	82%
6	DC motor	10	15			X		X		444	83%	100%	90%
7	Wikipedia	10	14	X	X	X		X		552	72%	100%	84%
8	Cell phone-SPL	16	13	X	X	X	X	X	X	486	51%	100%	68%
9	Clock	4	6	X		X		X		486	60%	100%	75%
10	1000 × 27 matrix	1000	27	X	X	X	X	X	X	46811	-	-	-
11	1500 × 137 matrix	1500	137	X	X	X	X	X	X	60350	-	-	-

Threats to validity

1. Lexical similarity
2. We consider junction as feature implementations
3. Dynamic analysis techniques
4. There is a limitation of using FCA as clustering technique
5. Each use-case represents a single feature
6. The mined FM defines more configurations than the initial FM