# Explorer la variabilité des lignes de produits logicielles avec l'Analyse Formelle de Concepts

HAI916I - 2021

Avec de nombreux schémas tirés de la thèse de Jessie Carbonnel

# Outline

# Contexte

## Lignes de produits/Product lines

- Industrialiser la production de produits similaires (variantes) personnalisés
  - Matériels
  - Immatériels
- Ex. : Logiciels, Voitures, Téléphones, Ordinateurs, Parcours pédagogique






- Ligne de produits :
  - un modèle de variabilité : caractéristiques variables et contraintes
  - une architecture commune (*architecture de référence*)

  - des points de variabilité prévus dans l'architecture de référence
  - un dépôt d'artefacts pluggables dans l'architecture de référence

# Contexte

## Lignes de produits/Product lines

- Bénéfices attendus
  - solutions personnalisées
  - choix plus grand
  - rapidité de production
  - coût plus bas
  - meilleure qualité (contrôle des variantes)
  - logiciel allégé
    - généré automatiquement en partie ou en totalité
    - léger d'usage (fonctionnalités resserrées sur le besoin)
    - moins gourmand en ressources (*green*)
- Obstacles
  - Investissement dans le développement de la ligne de produits
  - Suivi des versions ("variabilité dans le temps")
  - Maintenance et évolution complexifiées

## Une maturité académique et industrielle

Software Product Line Conference - Hall of Fame (réussites industrielles)

# Product lines

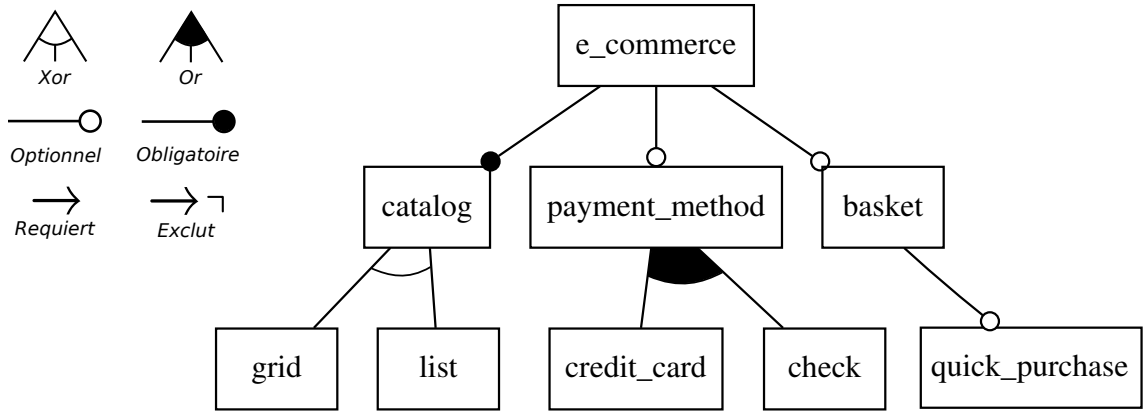
## Ingénierie de domaine

- modélisation de la variabilité (par ex. avec des *Feature Models*)
- développement de l'architecture de référence "à trous"
- développement des artefacts pluggables

## Ingénierie d'application

- sélection de *Features*
- génération d'une application exécutable

# Modèles de variabilité : Feature models [Czarcnecki et al.]




basket $\longrightarrow$ payment_method

quick_purchase $\longrightarrow \neg$ check

- Sémantique ontologique (Expression de la connaissance du domaine)
- Sémantique logique (formule propositionnelle)
  $(e\_commerce \wedge catalog \wedge list) \vee (e\_commerce \wedge catalog \wedge list \wedge basket \wedge ...)...$
- Sémantique de configurations (liste des variantes correctes)

```
e_commerce, catalog, list
e_commerce, catalog, list, payment_method, check
e_commerce, catalog, list, basket, payment_method, ...
(...)
```

# Focus

## Extract a variability model

### Elements

- Features
- Multi-valued attributes
- Cardinalities (for feature or feature group)
- References (for cross-product lines)

### Relationships

- Binary implications, co-occurrences
- Mutual exclusions
- Groups (OR, XOR)
- Constraints, Logic formulas
- ▶▶▶ *Intermediate structures*

# Method

## Using a framework

- In correspondence with propositional/predicate/description logics background
- Sound and complete extraction capabilities
- Canonical and exhaustive constructions
- Graphical representation capabilities
- Reusable, Extensible

## Underlying theory

- Formal Concept Analysis
  - Configurations + Features

# Method

## Approach in a nutshell

- Apply FCA
- Analyze and interpret the conceptual structures
  - Mapping to variability relationships (e.g. for refactoring)
  - Exploring the conceptual structures (e.g. for recommendation)

# Outline

# Formal Concept Analysis

## Principles [GW99]

- General theory on (Galois) connections between lattices structures [Bir40, BM70]
- Restricted in FCA to Configurations + Boolean Features = Concepts + Hierarchy

| OfficeSuite | document type | typesetting quality |
|---|---|---|
| **MsOff** | spread sheet, text vector graphics, presentation | standard |
| **libreOff** | spread sheet, text vector graphics, presentation | standard |
| **TexM** | text, math formula, presentation | high |
| **TexLxM** | text, math formula, presentation | high |
| **ScWrd** | text, math formula, presentation | high |
| **Scribs** | layout design, vector graphics | standard |
| **Indesig** | layout design, vector graphics | standard |

product description table

| OfficeSuite | spreadSheet | text | formula | layoutDesign | vectorGraphics | presentation | qualityTS | StandardTS | HighTS |
|---|---|---|---|---|---|---|---|---|---|
| **MsOff** | × | × | | | × | × | × | × | |
| **libreOff** | × | × | | | × | × | × | × | |
| **TexM** | | × | × | | | × | × | | × |
| **TexLxM** | | × | × | | | × | × | | × |
| **ScWrd** | | × | × | | | × | × | | × |
| **Scribs** | | | | × | × | | × | × | |
| **Indesig** | | | | × | × | | × | × | |

binary table Configurations + Boolean Features (formal context)

*Simplified description of dummy office suites*

# FCA

## Principles

Configurations + Features = Concepts + Hierarchy

| OfficeSuite | spreadSheet | text | formula | layoutDesign | vectorGraphics | presentation | qualityTS | StandardTS | HighTS |
|---|---|---|---|---|---|---|---|---|---|
| MsOff | × | × | | | × | × | × | × | |
| libreOff | × | × | | | × | × | × | × | |
| TexM | | × | × | | | × | × | | × |
| TexLxM | | × | × | | | × | × | | × |
| ScWrd | | × | × | | | × | × | | × |
| Scribs | | | | × | × | | × | × | |
| Indesig | | | | × | × | | × | × | |

formal context

| Concept_OfficeSuite_5 |
|---|
| text<br>presentation<br>qualityTS |
| MsOff<br>libreOff<br>TexM<br>TexLxM<br>ScWrd |

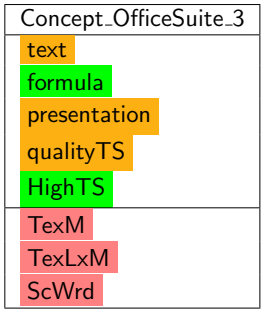
concept

*Concept = maximal configuration set sharing maximal feature set*

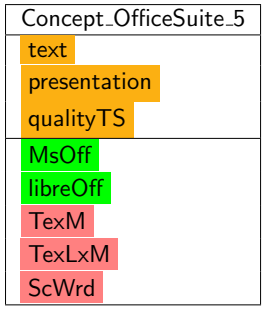
# FCA

## Principles

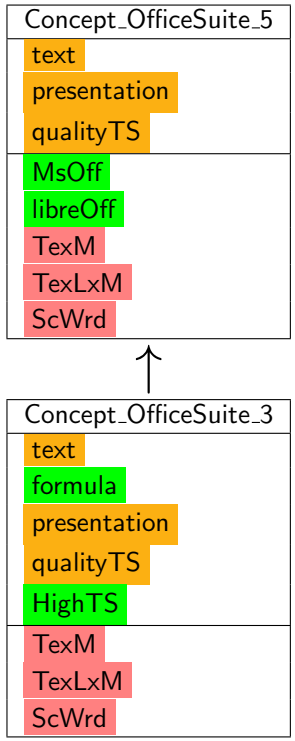
Configurations + Features = Concepts + Hierarchy



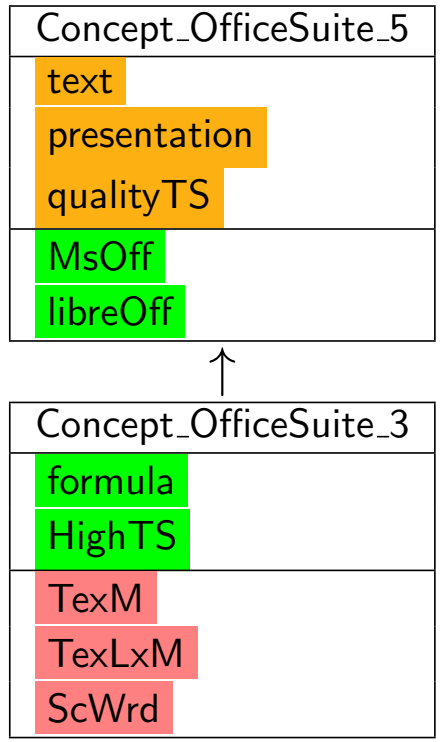

concept




concept




Specialization




*Inheritance*

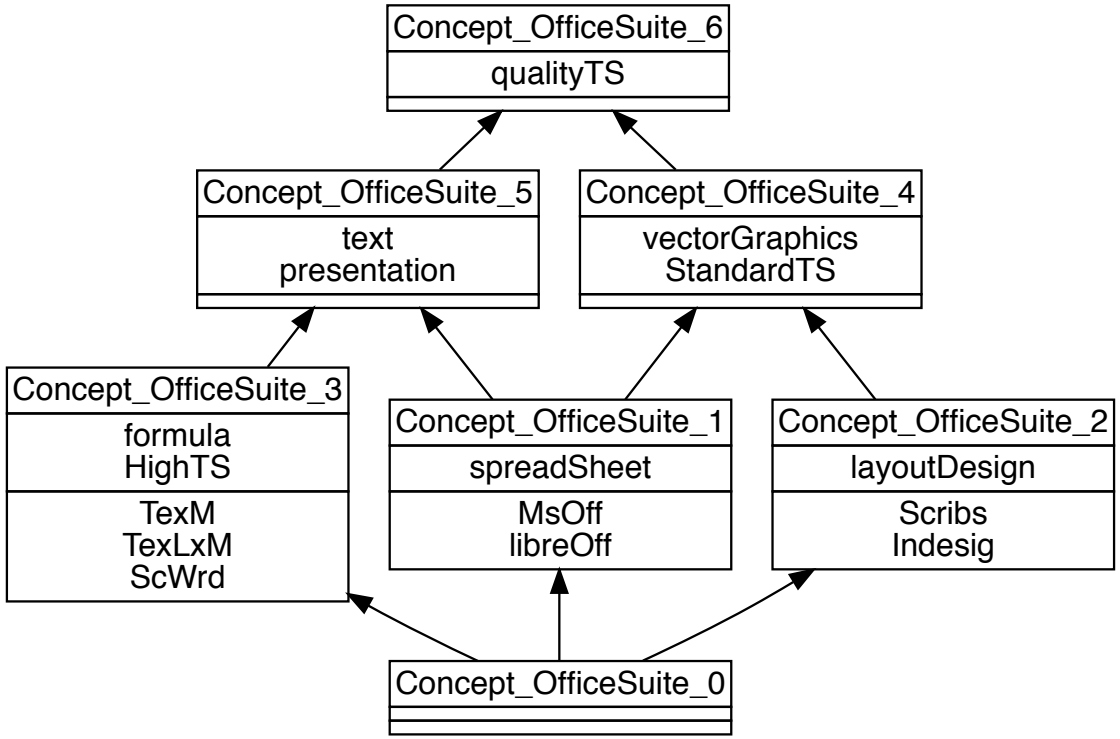
Features: top-down

Configurations: bottom-up

# FCA

## Principles

Configurations + Features = Concepts + Hierarchy

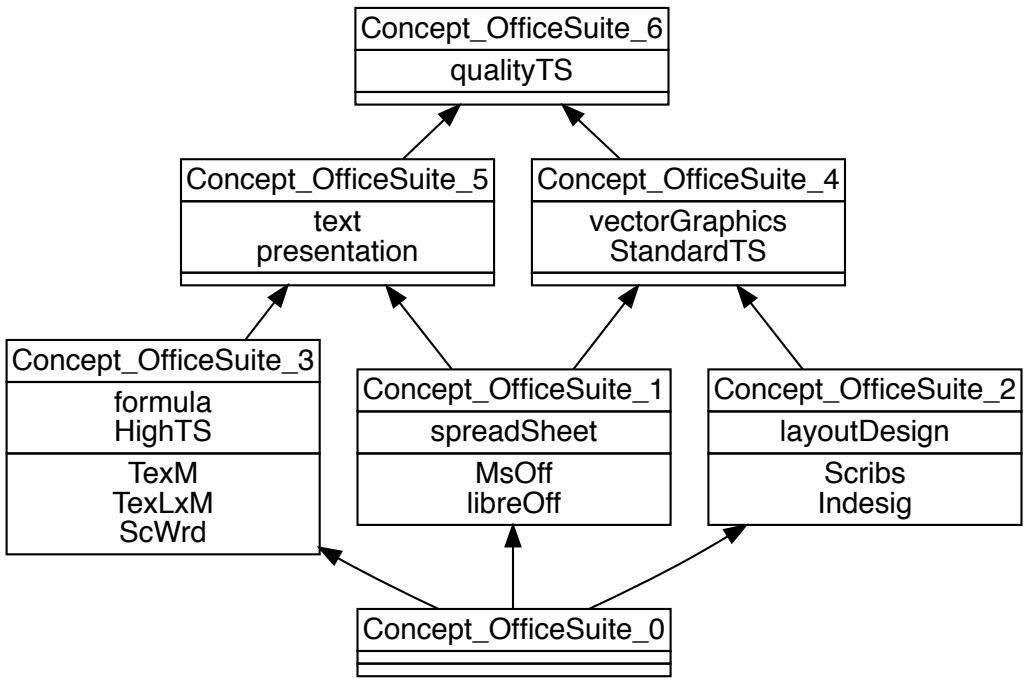



*Concept lattice*

# FCA

## Variability extraction [LP07, RPK11]

Candidates for Implications and co-occurences, or child-parent in a feature model




Concept_OfficeSuite_3
*subconcept of*
Concept_OfficeSuite_5

formula $\Rightarrow$ text

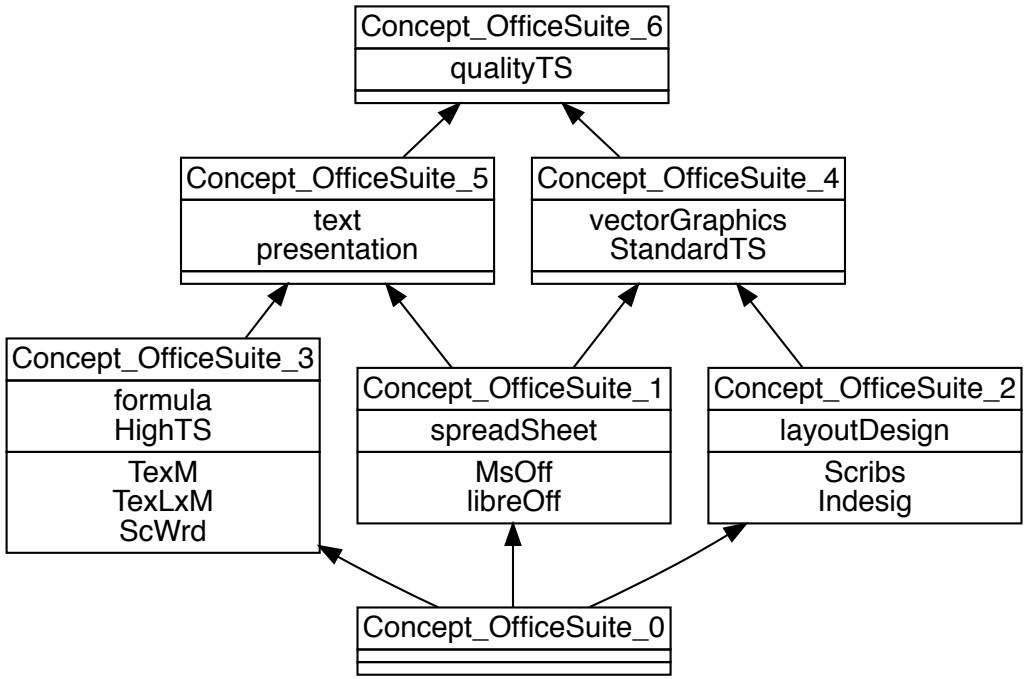
*Both introduced* in
Concept_OfficeSuite_5

text $\Leftrightarrow$ presentation

# FCA

## Variability extraction

### Candidates for Mutex



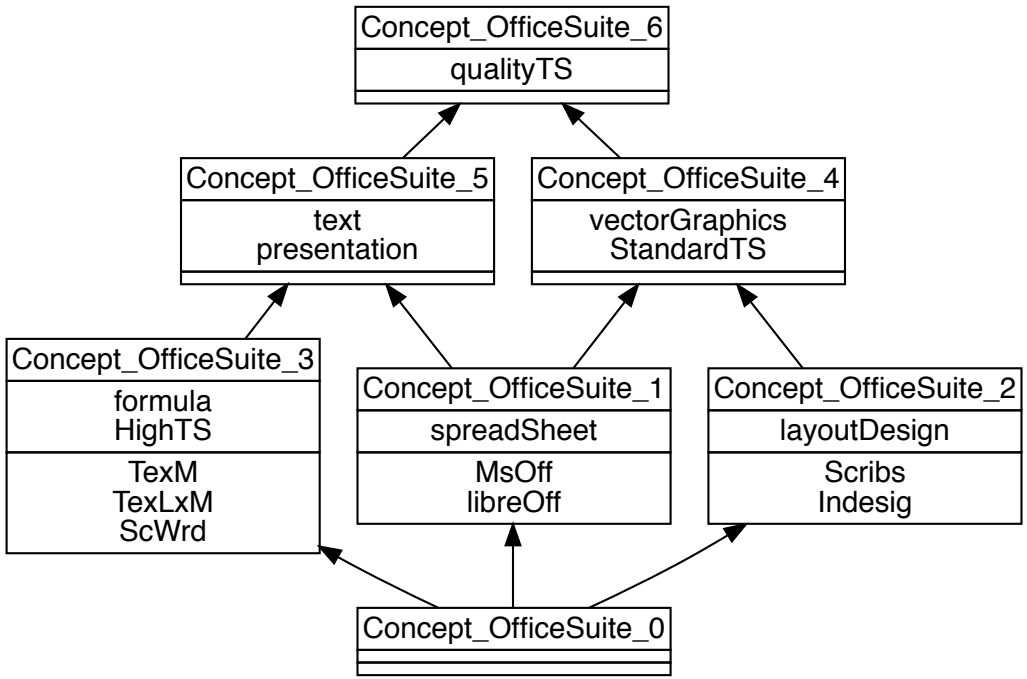

Concept_OfficeSuite_3
and
Concept_OfficeSuite_2
have *no common subconcept introducing a configuration*

formula $\Rightarrow \neg$ layoutDesign

# FCA

## Variability extraction

## Candidate Xor Groups



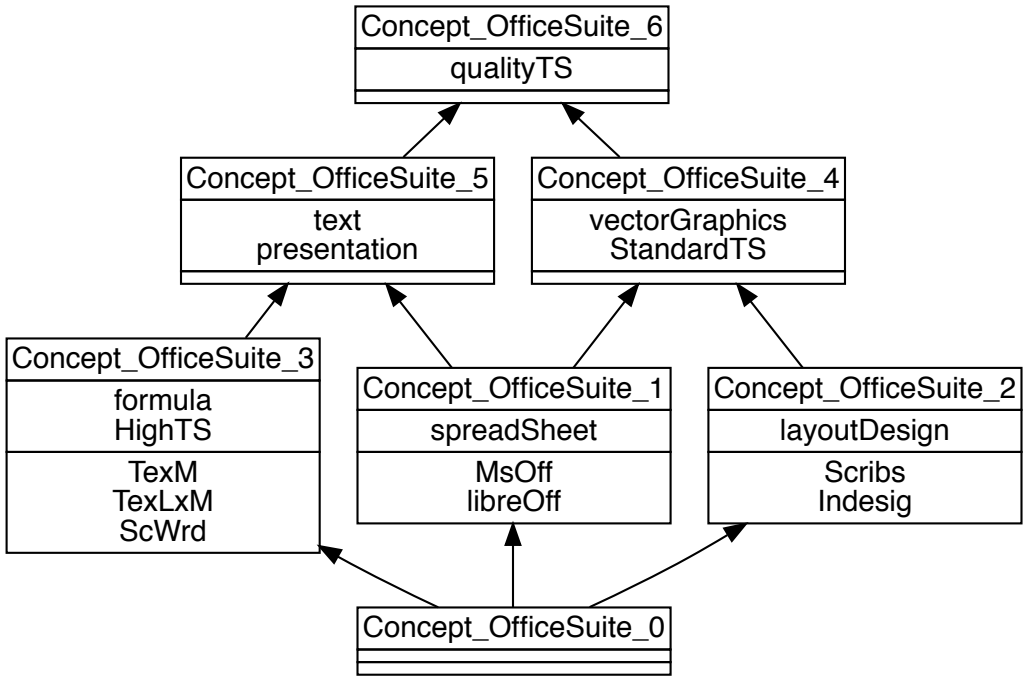

*Configurations of*
Concept_OfficeSuite_3
and
Concept_OfficeSuite_4
*form a partition of*
the configurations of
Concept_OfficeSuite_6

qualityTS
$= Xor$
{ HighTS, StandardTS }

# FCA

## Variability extraction

### Candidate Or Groups



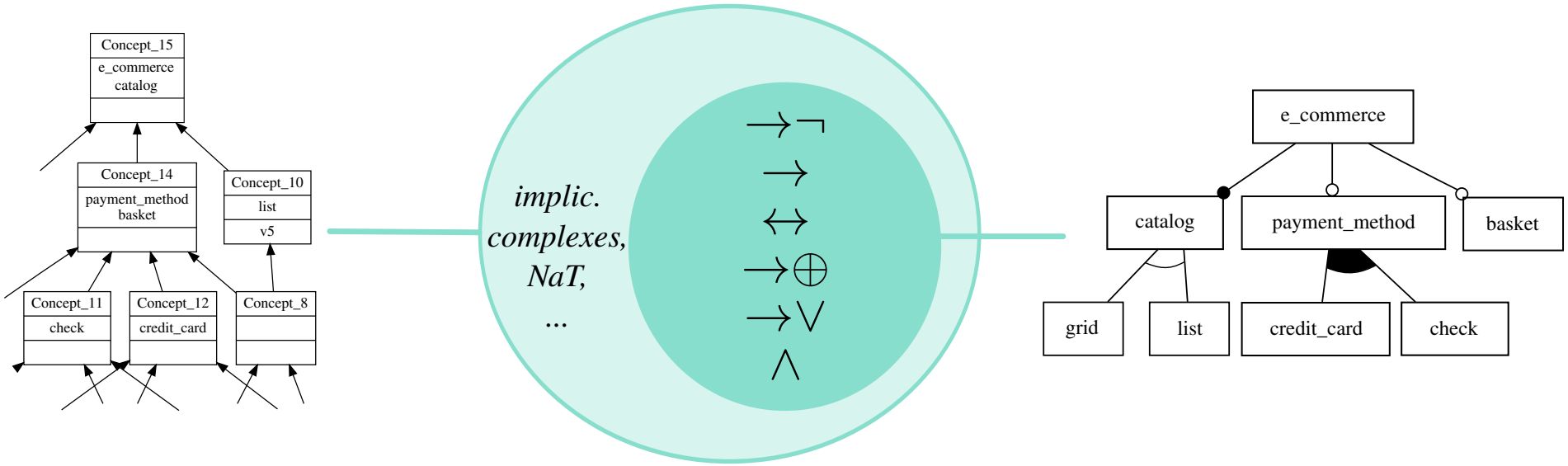

*Configurations of*
Concept_OfficeSuite_6
*are covered by*
non disjoint union
*of these of*
Concept_OfficeSuite_5
and
Concept_OfficeSuite_4

Office suite
$= Or$
{ text, vector graphics }

# Outline

# Cadre structurant

- Equivalence Treillis de concepts / Formule propositionnelle quelconque
- Représentation correcte et complète
- Dans la partie graphique d'un FM, certaines formules ne peuvent pas être représentées




J. Carbonnel et al. [CDHN19]

# Cadre structurant

- Représentation plus riche que les autres formalismes

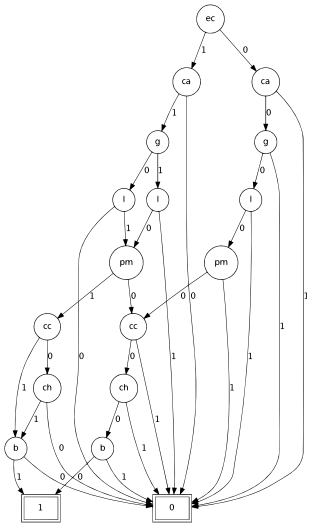
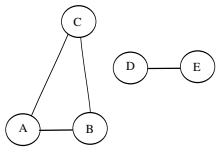
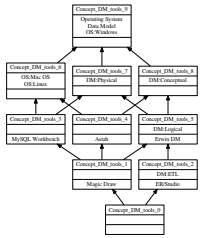


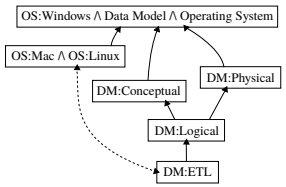

**DIAGRAMME DE DÉCISIONS BINAIRES**
[CW07, ACP+12, SRA+14]




**GRAPHE DE MUTEX**
[SRA+14]

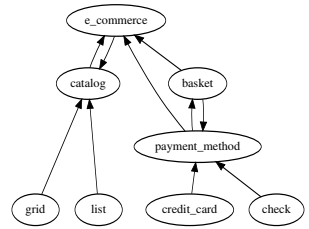


**STRUCTURES CONCEPTUELLES**
**ANALYSE FORMELLE DE CONCEPTS**
[RPK11, AMdSH+14, SSS17]




Go =
{(OS:Mac ∧ OS:Linux), DM:Physical}
{(OS:Mac ∧ OS:Linux), DM:Conceptual}
{DM:Conceptual, DM:Physical}
{(OS:Mac ∧ OS:Linux), DM:Logical}

**MODÈLE DE CARACTÉRISTIQUES GÉNÉRALISÉ**
[CW07]




**GRAPHE D'IMPLICATIONS BINAIRES**
[CW07, ACP+12, DDH+13, SRA+14]

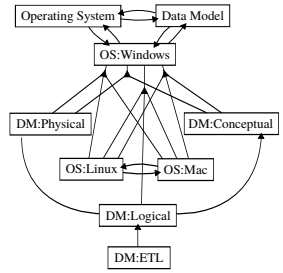
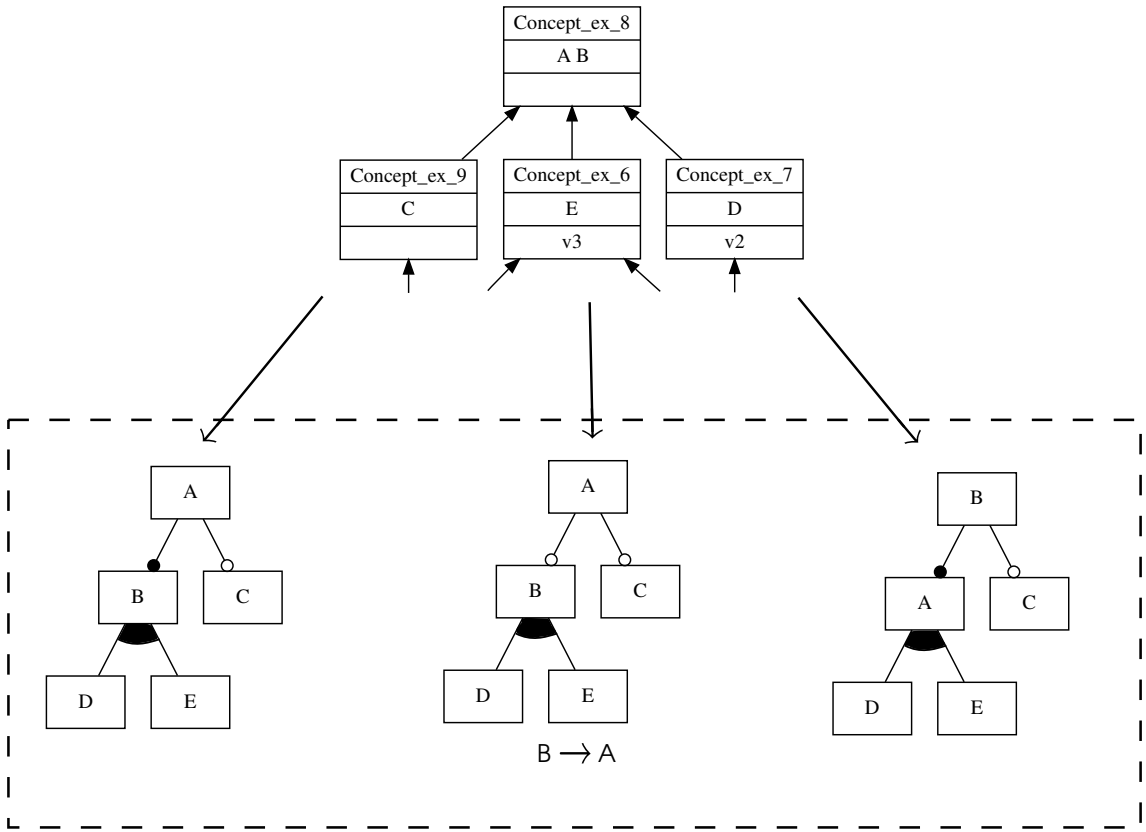



**DIAGRAMME DE CARACTÉRISTIQUES**
[SRA+14]

J. Carbonnel et al. [CDHN19]

# Cadre structurant

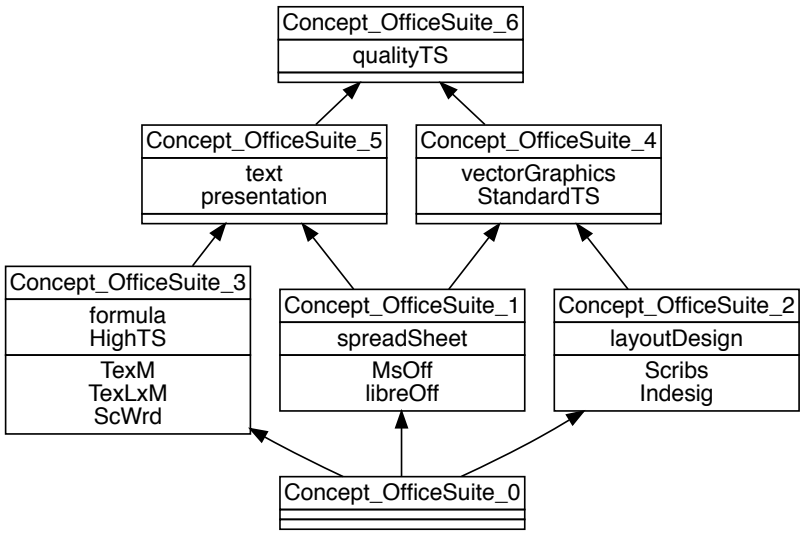
- Structure canonique
- Unicité du treillis de concepts associé à une formule propositionnelle
- Tous les Feature Models de même sémantique logique (formule) se plongent dans le treillis (ou inversement se dérivent du treillis)

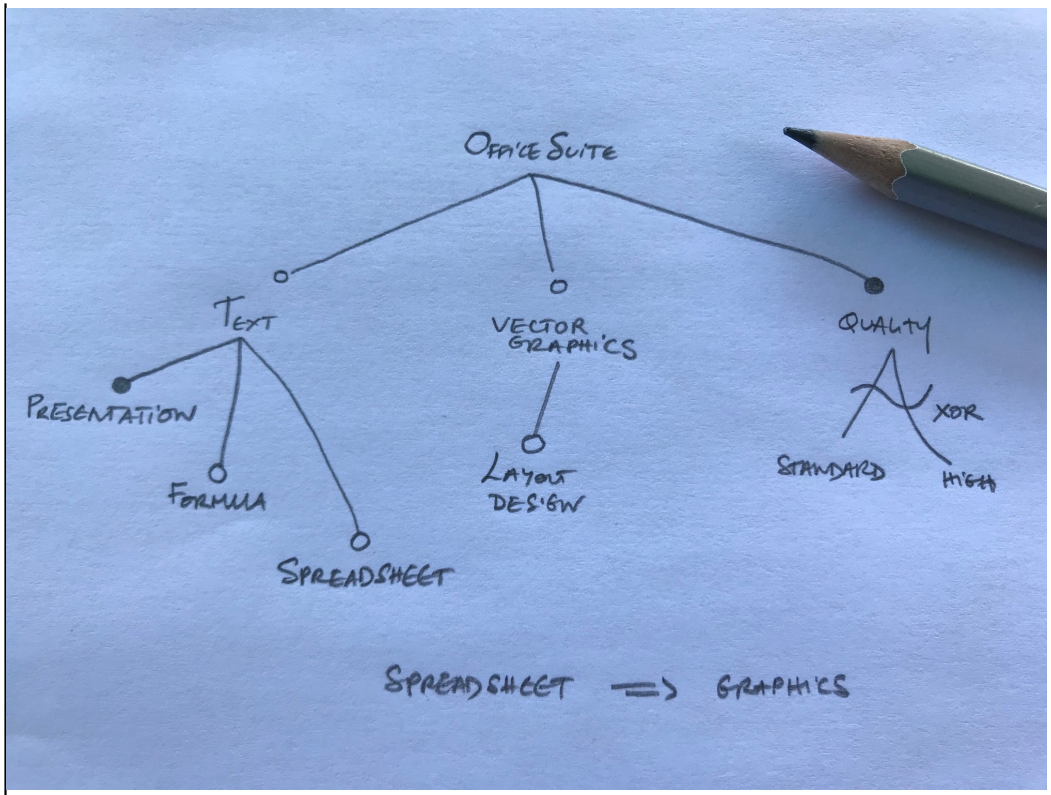



J. Carbonnel et al. [CHN19a]

# Application

## Guide de conception de Feature Model plus approfondi que [AHS+14]

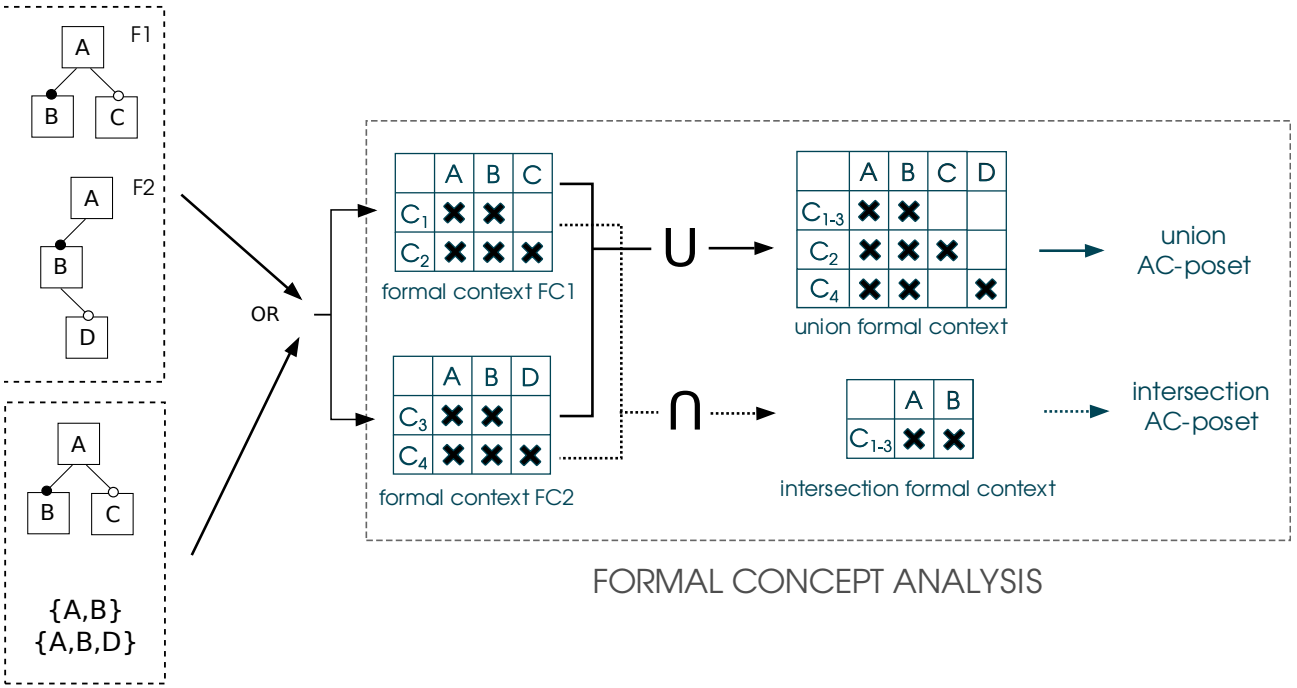


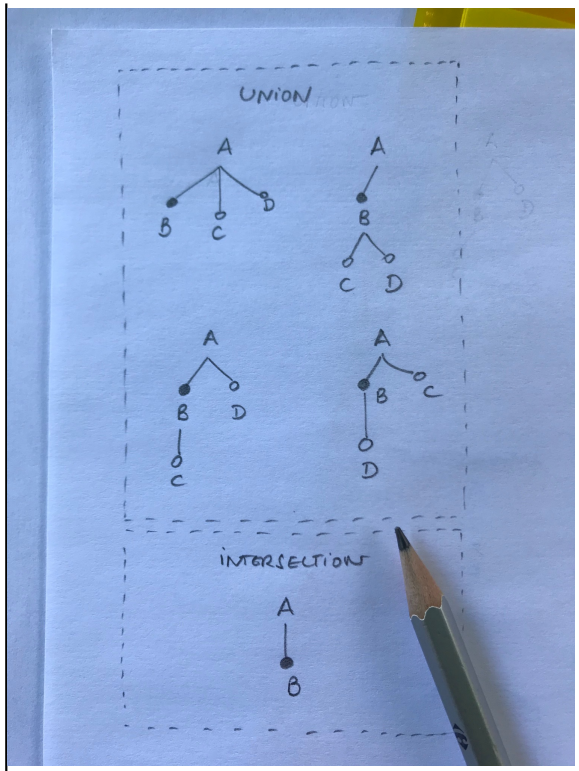




J. Carbonnel et al. [CHN19a]

# Application

## Union et intersection de modèles de variabilité (formats quelconques)




J. Carbonnel et al. [CHMN17]

# Application

## Exploration de la famille de produits (analyse, résumé par des bases d'implications)

### Règles d'implications
### Latviz (https://latviz.loria.fr)

**Implications: 12**

« Prev | 1 | Next »

| | | |
|---|---|---|
| Impl 0: spreadSheet --> text, vectorGraphics, presentation, qualityTS, StandardTS | Lift: 3.5 | Support: 0.2857 |
| Impl 1: vectorGraphics, text --> spreadSheet, presentation, qualityTS, StandardTS | Lift: 3.5 | Support: 0.2857 |
| Impl 2: vectorGraphics, presentation --> spreadSheet, text, qualityTS, StandardTS | Lift: 3.5 | Support: 0.2857 |
| Impl 3: StandardTS, text --> spreadSheet, vectorGraphics, presentation, qualityTS | Lift: 3.5 | Support: 0.2857 |
| Impl 4: StandardTS, presentation --> spreadSheet, text, vectorGraphics, qualityTS | Lift: 3.5 | Support: 0.2857 |
| Impl 5: text --> presentation, qualityTS | Lift: 1.4 | Support: 0.7143 |
| Impl 6: presentation --> text, qualityTS | Lift: 1.4 | Support: 0.7143 |
| Impl 7: formula --> text, presentation, qualityTS, HighTS | Lift: 2.33 | Support: 0.4286 |
| Impl 8: HighTS --> text, formula, presentation, qualityTS | Lift: 2.33 | Support: 0.4286 |
| Impl 9: vectorGraphics --> qualityTS, StandardTS | Lift: 1.75 | Support: 0.5714 |
| Impl 10: StandardTS --> vectorGraphics, qualityTS | Lift: 1.75 | Support: 0.5714 |
| Impl 11: layoutDesign --> vectorGraphics, qualityTS, StandardTS | Lift: 1.75 | Support: 0.2857 |

*Math. Sci. hum.* (24e année, n°95, *1986*, p.5-18)

FAMILLES MINIMALES D'IMPLICATIONS INFORMATIVES
RESULTANT D'UN TABLEAU DE DONNEES BINAIRES

J.L. GUIGUES* et V. DUQUENNE**

Discrete Applied Mathematics 273 (2020) 43–64

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

FCA for software product line representation: Mixing configuration and feature relationships in a unique canonical representation

Jessie Carbonnel [a,*], Karell Bertet [b], Marianne Huchard [a], Clémentine Nebut [a]

[a] LIRMM, CNRS & Université de Montpellier, 161 rue Ada, 34095 Montpellier Cedex 5, France
[b] L3i, Université de La Rochelle, Av. Michel Crépeau, 17042 La Rochelle Cedex 1, France

ARTICLE INFO

*Article history:*
Received 11 October 2017
Received in revised form 10 May 2019
Accepted 5 June 2019
Available online 2 July 2019

*Keywords:*
Software product line
Feature model
Formal concept analysis
Concept lattice

ABSTRACT

Software Product Line Engineering (SPLE) is a set of methods to help build a collection of software systems which are similar enough to enable appropriate artefact reuse. An important task consists in documenting in variability models the common and variable features which may compose the similar software systems along with compatibility constraints between these features. Several models and formalisms have been proposed to model variability: each one of them has specific properties making it pertinent to support certain management operations, which are of primary importance in SPLE. Switching from one kind of variability model to another is thus important to benefit from a wide range of operations and efficiently manage a software product line. In this paper, we review the various approaches proposed to manage and organise features and product configurations (a product configuration being a chosen subset of features). We discuss the originality of concept lattices, canonical structures presenting a dual view on features and configurations, and the advantages to use these structures as variability representations. Switching from existing variability models to an equivalent concept lattice raises scaling issues related to the size of the needed input dataset and thus hinders their exploitation. We propose an alternative relying on implicative systems and define a transformation method which does not suffer from scaling issues.


**1. Introduction**

Software Product Line Engineering (SPLE) [45] is a set of methods to help build a collection of software systems which are similar enough to enable appropriate artefact reuse. In this way, it optimises development and maintenance costs of the whole collection. To achieve efficient artefact reuse, analysing and modelling the software systems' *variability* is a crucial step. This task mainly consists in documenting the common and variable artefacts which may compose the similar software systems, as well as constraints about the compatibility of the different artefacts. A common way to document variability is to describe software systems depending on a set of high level *features* representing the systems' distinguishable characteristics or behaviours [10,39] and which are relevant to both designers and end-users. Each feature then matches a set of low level software artefacts having eventually less meanings for the end-user. These features are then organised in models depicting the corresponding artefact compatibilities: a combination of these features satisfying

* Corresponding author.
*E-mail addresses:* jcarbonnel@lirmm.fr (J. Carbonnel), karell.bertet@univ-lr.fr (K. Bertet), huchard@lirmm.fr (M. Huchard), nebut@lirmm.fr (C. Nebut).

https://doi.org/10.1016/j.dam.2019.06.008
0166-218X/

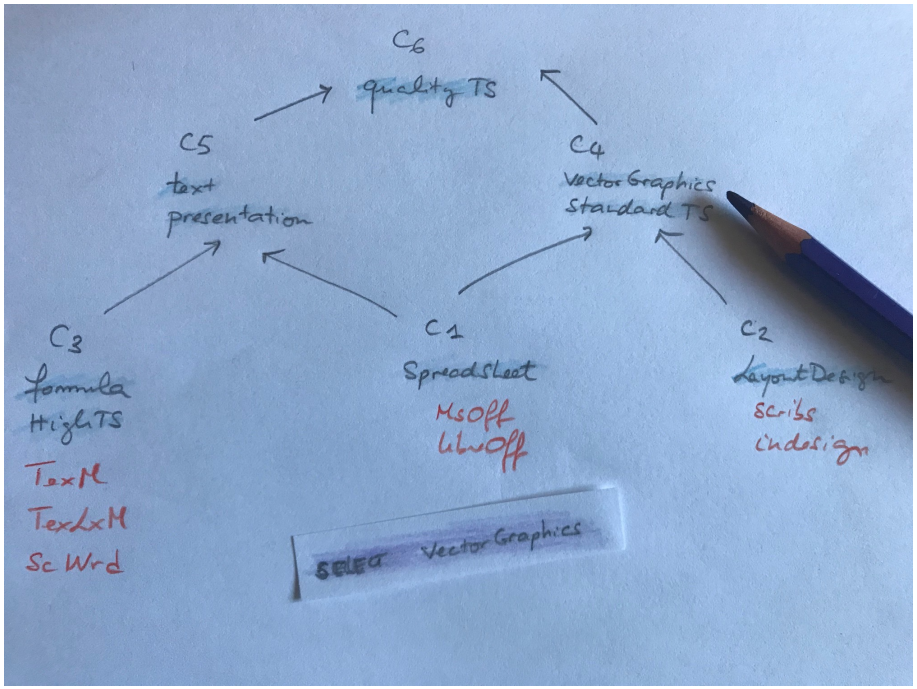
# Application

## Exploration de la famille de produits (navigation)

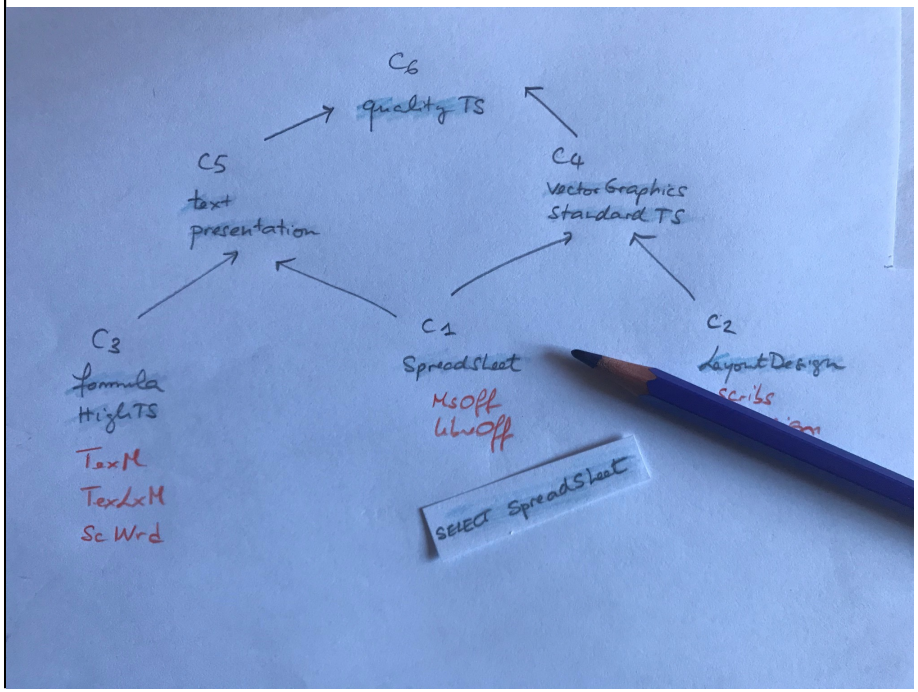
S. Ferré, 2014, *Conceptual Navigation*

A. Bazin, J. Carbonnel, et al. ICFCA 2019 *algorithms on-demand*

Sélection de `vectorGraphics` $\longrightarrow$ C4




Sélection de `spreadSheet` $\downarrow$ C1



# Travaux pratiques sur les options d'accessibilité visuelle de systèmes d'exploitation

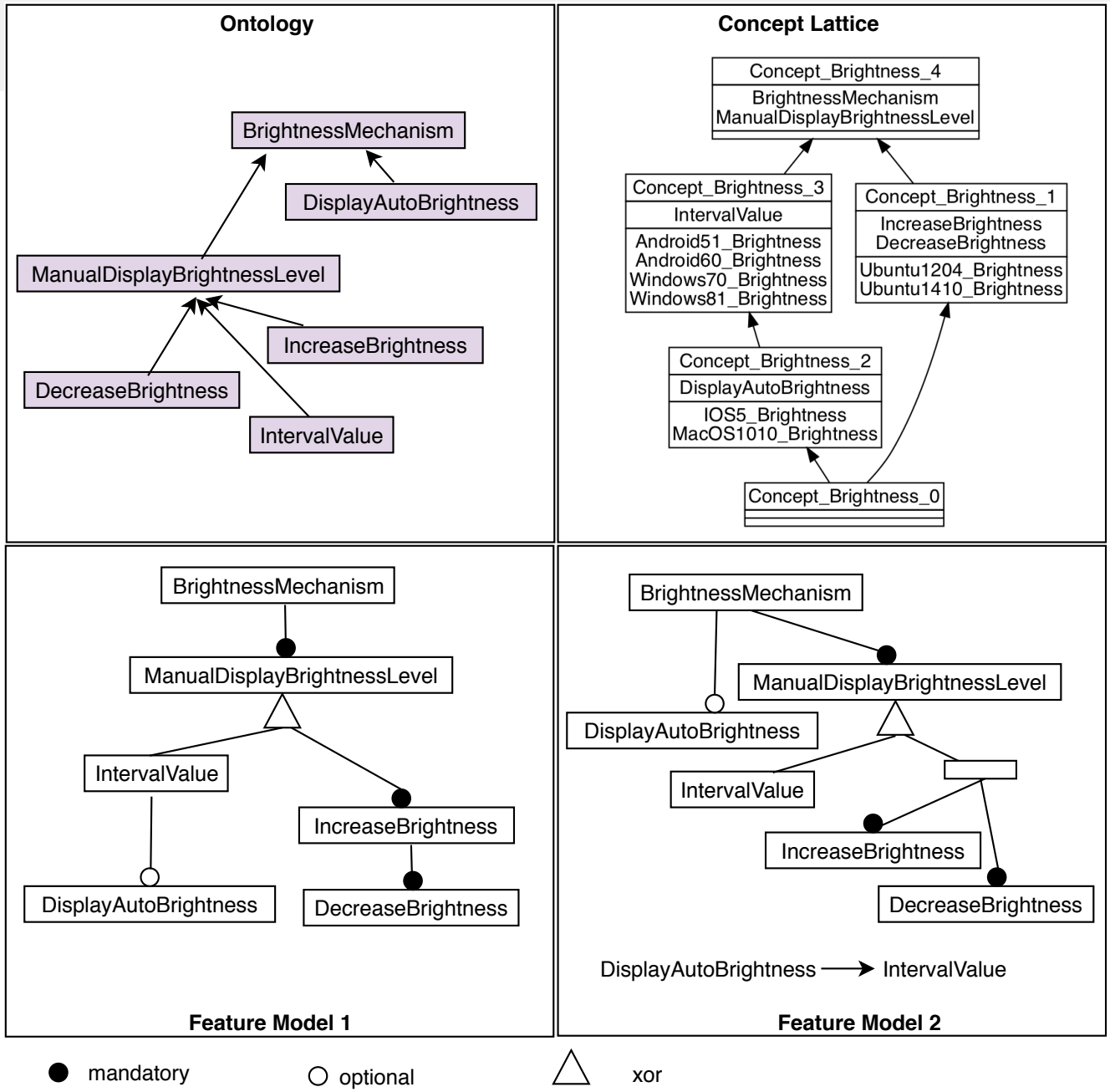
Le but sera de produire des *feature models* pour des options d'accessibilité dans des systèmes d'exploitation. Les données ont été recueillies et présentées dans un article (voir sur Moodle pour les trouver).

Dans ce TP, on s'intéressera à produire un feature model pour les mécanismes *Brightness* et *TextSize* uniquement. Le principe général est d'utiliser les ontologies et les treillis de concepts pour produire un feature model.

Vous travaillerez par groupe de 1 à 4 de l'évaluation en utilisant les ontologies et les treillis de concepts. Vous rendrez sur Moodle :

- un document pdf décrivant tous les choix que vous avez fait et vos réflexions.
- les sources des *features models* que vous créez et une image pdf
- pour faire les feature models, vous utiliserez au choix l'éditeur en lignede SPLOT `http://www.splot-research.org/`, ou de FeatureIDE `https://featureide.github.io/` ou de FAMILIAR `http://familiar-project.github.io/`. Ces deux derniers demandent une installation.

# Principe


Ontology
BrightnessMechanism
DisplayAutoBrightness
ManualDisplayBrightnessLevel
IncreaseBrightness
DecreaseBrightness
IntervalValue
Concept Lattice
Concept_Brightness_4
BrightnessMechanism
ManualDisplayBrightnessLevel
Concept_Brightness_3
IntervalValue
Android51_Brightness
Android60_Brightness
Windows70_Brightness
Windows81_Brightness
Concept_Brightness_1
IncreaseBrightness
DecreaseBrightness
Ubuntu1204_Brightness
Ubuntu1410_Brightness
Concept_Brightness_2
DisplayAutoBrightness
IOS5_Brightness
MacOS1010_Brightness
Concept_Brightness_0
BrightnessMechanism
ManualDisplayBrightnessLevel
IntervalValue
IncreaseBrightness
DisplayAutoBrightness
DecreaseBrightness
Feature Model 1
BrightnessMechanism
ManualDisplayBrightnessLevel
DisplayAutoBrightness
IntervalValue
IncreaseBrightness
DecreaseBrightness
DisplayAutoBrightness → IntervalValue
Feature Model 2
mandatory
optional
xor

# Guide pour structurer votre réponse

Dans votre réponse, vous répondrez en particulier aux questions suivantes :

- Pouvez-vous décrire dans quel ordre vous avez procédé ? Par exemple : partir de l'ontologie et la modifier ; ou partir du treillis et le modifier ; ou mélanger les deux ; partir du haut des structures, partir du bas des structures, un peu dans les deux sens ; etc.
- Avez-vous utilisé principalement l'ontologie ou le treillis, ou autant les deux pour choisir (donner un exemple pour chaque information) :
  - les arcs du feature model
  - les indications 'optionnel', 'obligatoire', groupe OR, groupe XOR
  - les contraintes textuelles (requiert binaire, exclus binaire)
- Y a-t-il des informations de l'ontologie que vous n'avez pas utilisées ? Indiquez lesquelles et pourquoi.
- Y a-t-il des informations du treillis que vous n'avez pas utilisées ? Indiquez lesquelles et pourquoi.
- Avez-vous eu besoin d'ajouter des features (des termes non présents dans l'ontologie) ?
- Avez-vous ignoré des features (des termes présents dans l'ontologie que vous n'avez pas utilisé) ?
- Avez-vous pensé à des informations qui vous ont manqué pour effectuer certains choix ? Indiquez-les et dites comment elles pourraient être procurées.
- Tout autre retour que vous pourriez faire concernant ce procédé.


Ra'Fat Al-Msie'deen, Marianne Huchard, Abdelhak Seriai, Christelle Urtado, and Sylvain Vauttier.
Reverse engineering feature models from software configurations using formal concept analysis.
In Karell Bertet and Sebastian Rudolph, editors, *Proceedings of the Eleventh International Conference on Concept Lattices and Their Applications, Košice, Slovakia, October 7-10, 2014*, volume 1252 of *CEUR Workshop Proceedings*, pages 95–106. CEUR-WS.org, 2014.


G. Birkhoff.
*Lattice Theory*.
Colloquium publications. American Mathematical Society, 1940.


M. Barbut and B. Monjardet.
*Ordre et classification – Algèbre et combinatoire, Tome 2*.
Hachette, Paris, 1970.


Jessie Carbonnel, David Delahaye, Marianne Huchard, and Clémentine Nebut.

Graph-based variability modelling: Towards a classification of existing formalisms.

In *Graph-Based Representation and Reasoning - Proceedings of the 24th International Conference on Conceptual Structures (ICCS'19)*, volume 11530 of *Lecture Notes in Computer Science*, pages 27–41. Springer, 2019.


Jessie Carbonnel, Marianne Huchard, André Miralles, and Clémentine Nebut.
Feature model composition assisted by formal concept analysis.
In Ernesto Damiani, George Spanoudakis, and Leszek A. Maciaszek, editors, *ENASE 2017 - Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering, Porto, Portugal, April 28-29, 2017*, pages 27–37. SciTePress, 2017.


Jessie Carbonnel, Marianne Huchard, and Clémentine Nebut.
Modelling equivalence classes of feature models with concept lattices to assist their extraction from product descriptions.
*Journ. of Syst. and Soft.*, 152:1 – 23, 2019.


Jessie Carbonnel, Marianne Huchard, and Clémentine Nebut.
Towards complex product line variability modelling: Mining relationships from non-boolean descriptions.
*J. Syst. Softw.*, 156:341–360, 2019.


Bernhard Ganter and Rudolf Wille.
*Formal concept analysis - mathematical foundations*.

Springer, 1999.

Felix Loesch and Erhard Ploedereder.
Restructuring Variability in Software Product Lines using Concept Analysis of Product Configurations.
In *Proceedings of the 11th European Conference on Software Maintenance and Reengineering, Software Evolution in Complex Software Intensive Systems (CSMR'07)*, pages 159–170. IEEE Computer Society, 2007.

Uwe Ryssel, Joern Ploennigs, and Klaus Kabitzsch.
Extraction of feature models from formal contexts.
In *Workshop Proceedings (Volume 2) of the 15th International Conference on Software Product Lines (SPLC'11)*, pages 4:1–4:8. IEEE Computer Society, 2011.