Extraction and Representation of Variability in User Stories in a Family of Similar Software

Marianne Huchard

LIRMM, CNRS and Université de Montpellier

HAI916I Intelligence Artificielle pour le Génie Logiciel
Faculté Des Sciences
Université de Montpellier

Outline

Rappel du contexte

Product lines

- Industrialiser la production de produits (logiciels) similaires personnalisés
- construits à partir :
  - d'une architecture commune (architecture de référence)
  - de points de variabilité prévus dans l'architecture de référence
  - d'un dépôt d'artefacts pluggables dans l'architecture de référence
- Bénéfices attendus
  - choix plus grand
  - coût plus bas
  - meilleure qualité (contrôle des variantes)

Product lines

Ingénierie de domaine

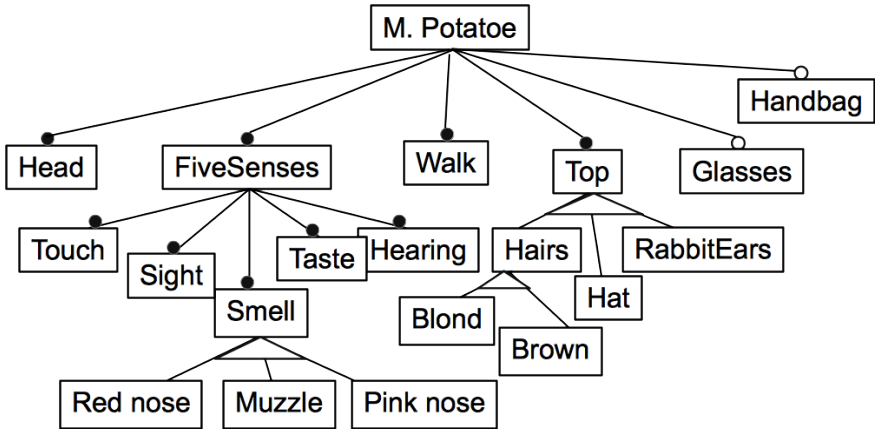
- modélisation de la variabilité (par ex. avec des *Feature Models*)
- développement de l'architecture de référence
- développement des artefacts pluggables

Ingénierie d'application

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

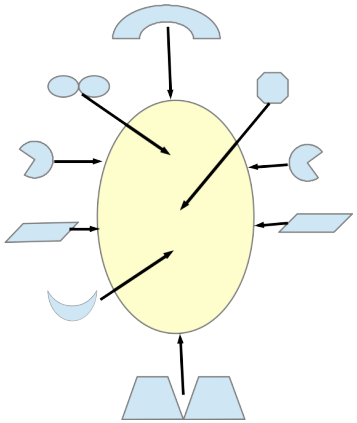
Product Line Engineering (domain engineering)

Feature model



Assets

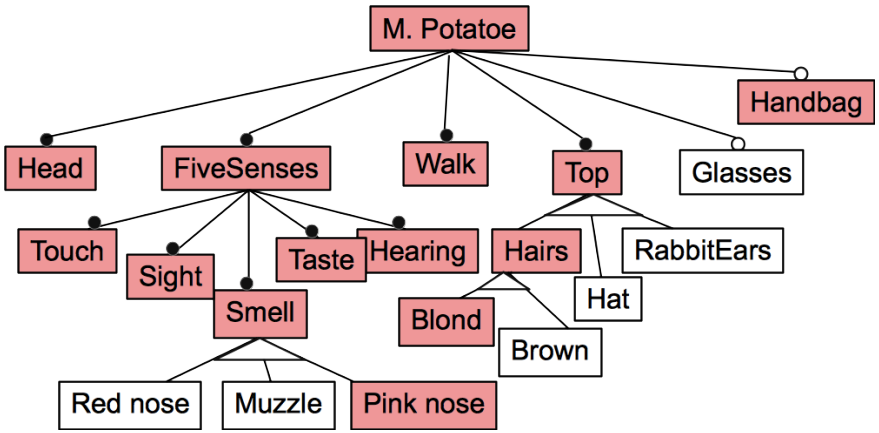
Variable Architecture



Derived Products

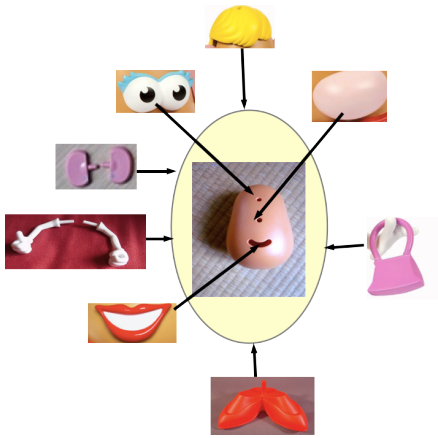
Building a product (application engineering)

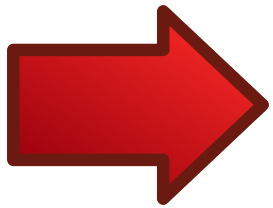
Feature selection



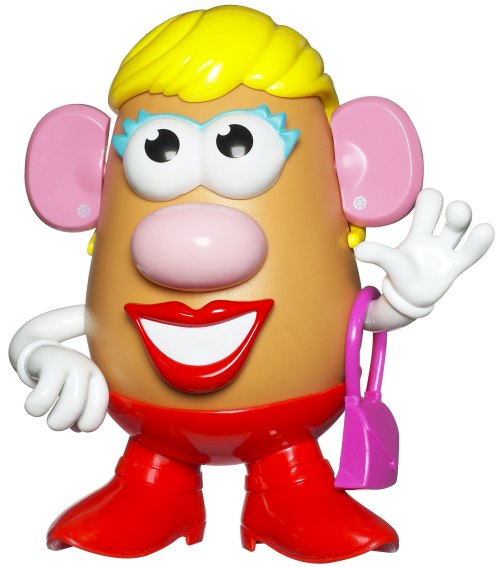
Selected Assets

Implemented Architecture

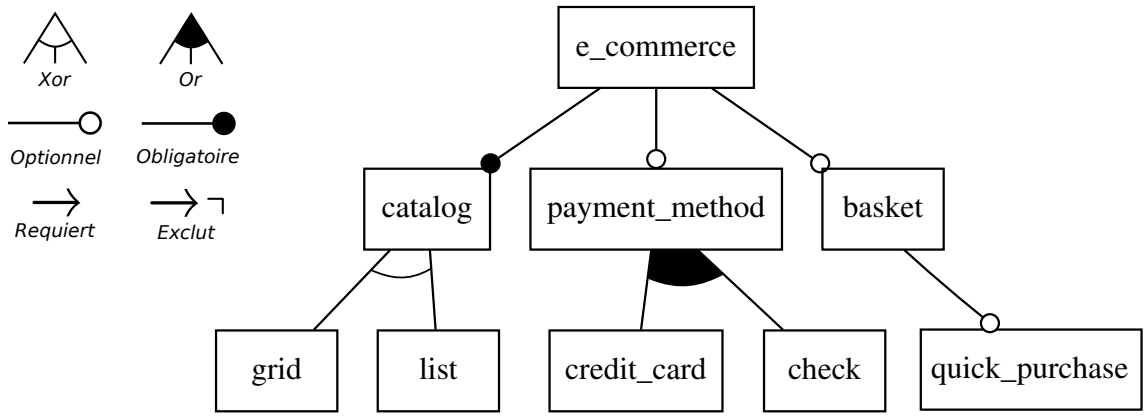




Product

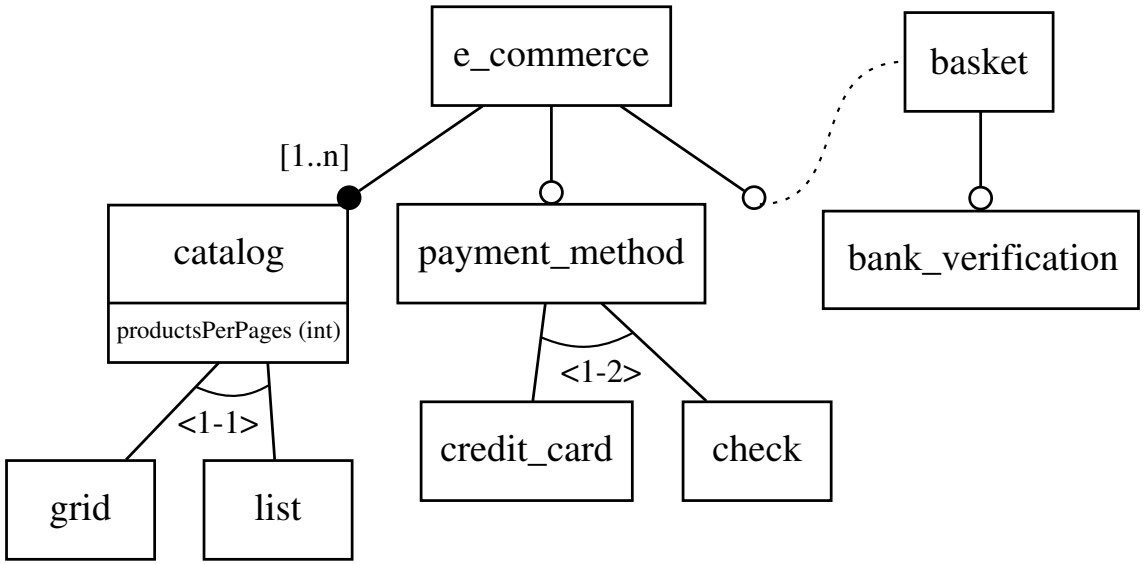


Feature models [Czarcnecki et al.]



$$\text{basket} \longrightarrow \text{payment\_method}$$
$$\text{quick\_purchase} \longrightarrow \neg\ \text{check}$$

Extended Feature models: references, cardinalities, attributes



payment_method -> basket
basket -> payment_method
credit_card -> bank_verification
grid -> (productsPerPages > 15)

Context

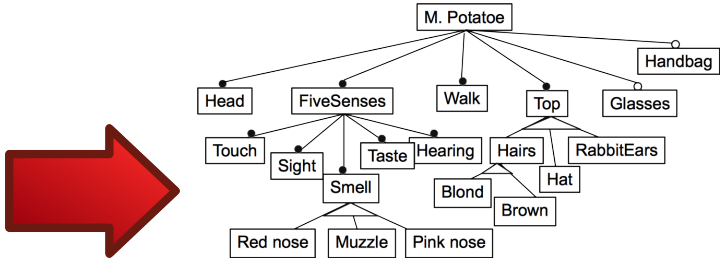
Product lines construction approaches:

- proactive (ex nihilo)
- extractive (by reengineering)
- reactive approaches (spiral, by building a few first products and making the PL evolve along the new developments)

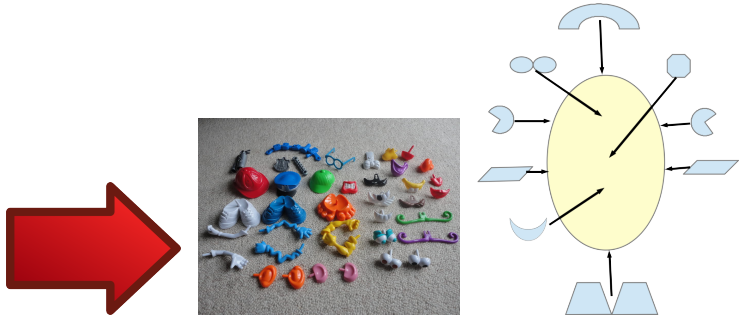
Product Line Reverse Engineering (extractive approach)

Similar Products
developed in undisciplined manner

Which Feature model?



Which Assets?
Which Architecture?



Extractive Approach with FCA

- Variability extraction and representation
  - Understanding
  - Migration
  - Refactoring
  - Adaptation
  - Recommendation

Targeted description elements

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

Extractive Approach with FCA

Targeted variability relationships extraction

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

Method

FCA: 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 [Ganter and Wille, 1999]

- General theory on (Galois) connections between lattices structures [Birkhoff, 1940, Barbut and Monjardet, 1970]
- 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 presentation qualityTS
MsOff libreOff TexM TexLxM ScWrd

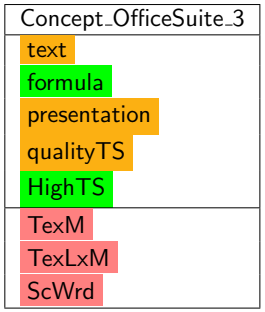
concept

Concept = maximal configuration set sharing maximal feature set

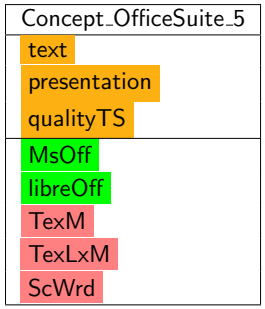
FCA

Principles

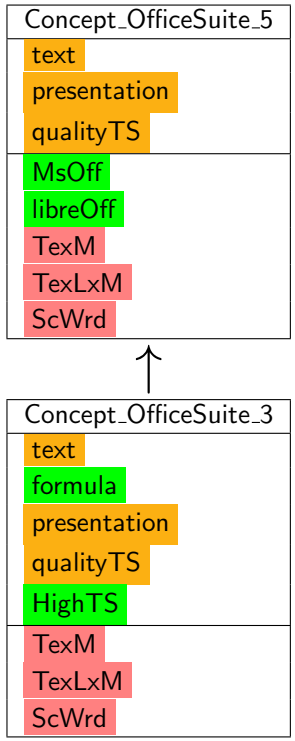
Configurations + Features = Concepts + Hierarchy



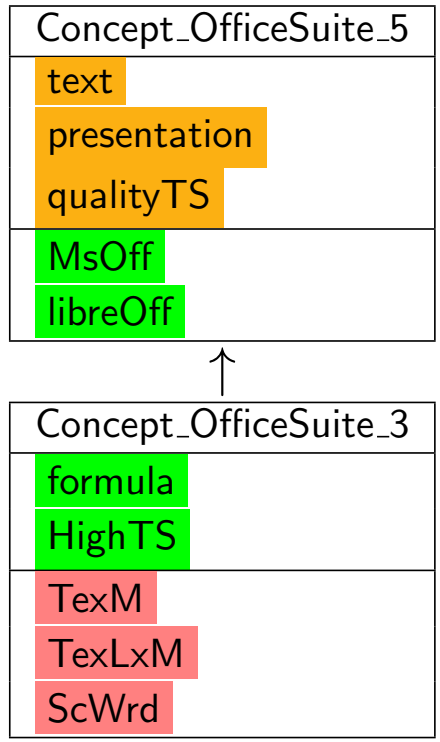
concept



concept



Specialization



Inheritance

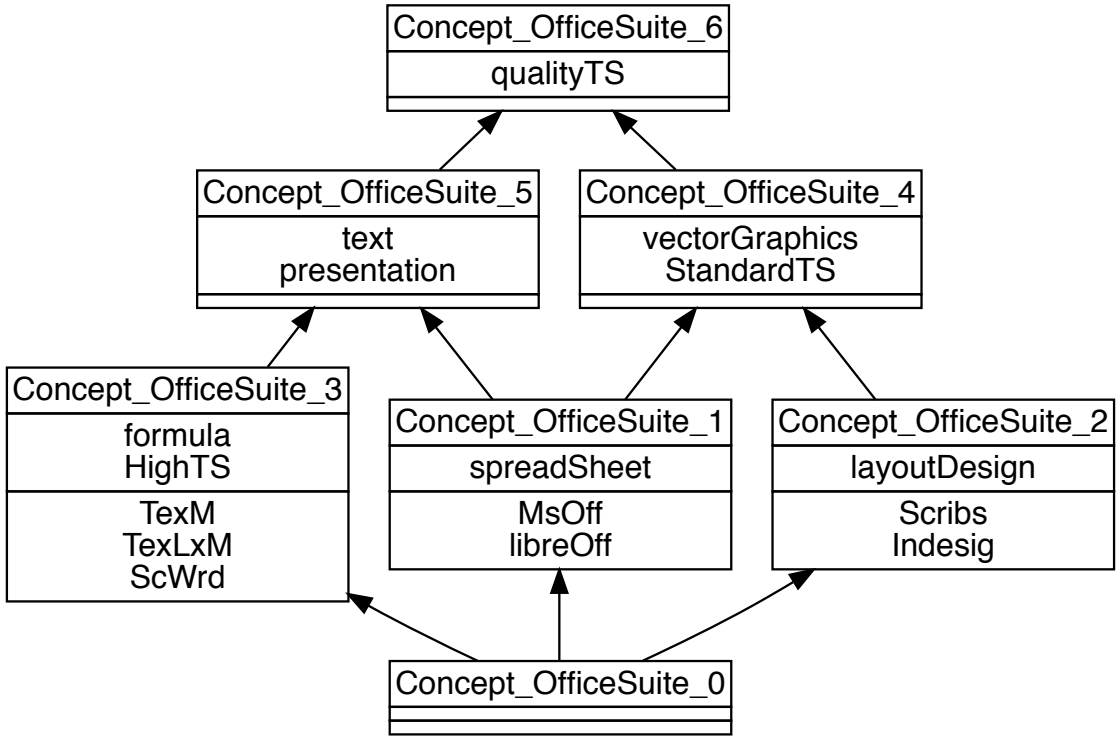
Features: top-down

Configurations: bottom-up

FCA

Principles

Configurations + Features = Concepts + Hierarchy

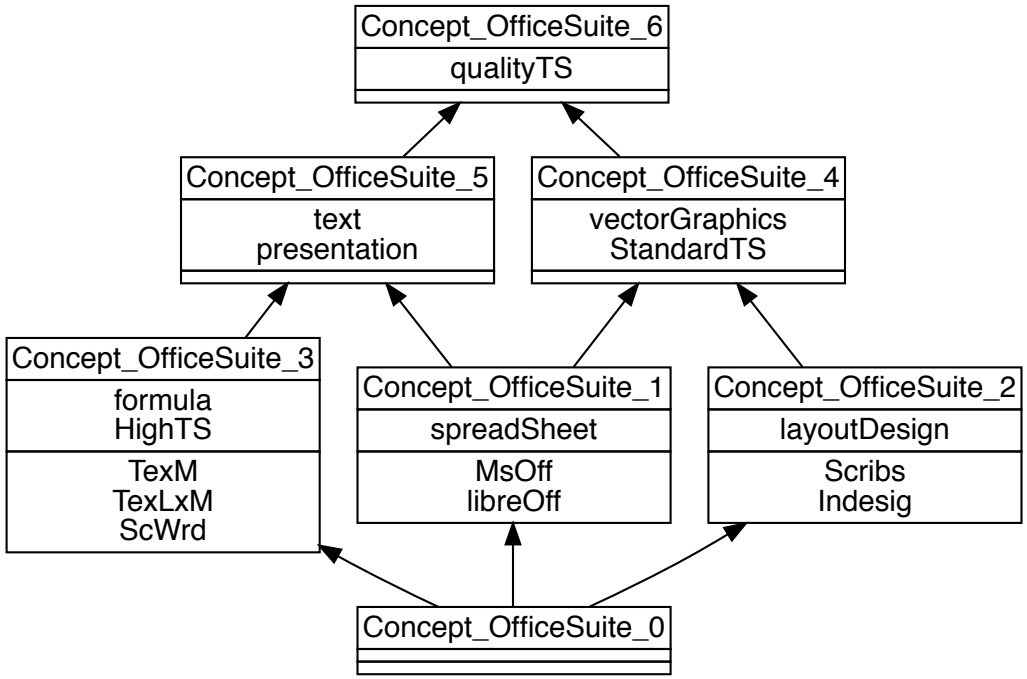


Concept lattice

FCA

Variability extraction [Loesch and Ploedereder, 2007, Ryssel et al., 2011]

Implications and co-occurences



Concept_OfficeSuite_3
subconcept of
Concept_OfficeSuite_5

formula $\Rightarrow$ text

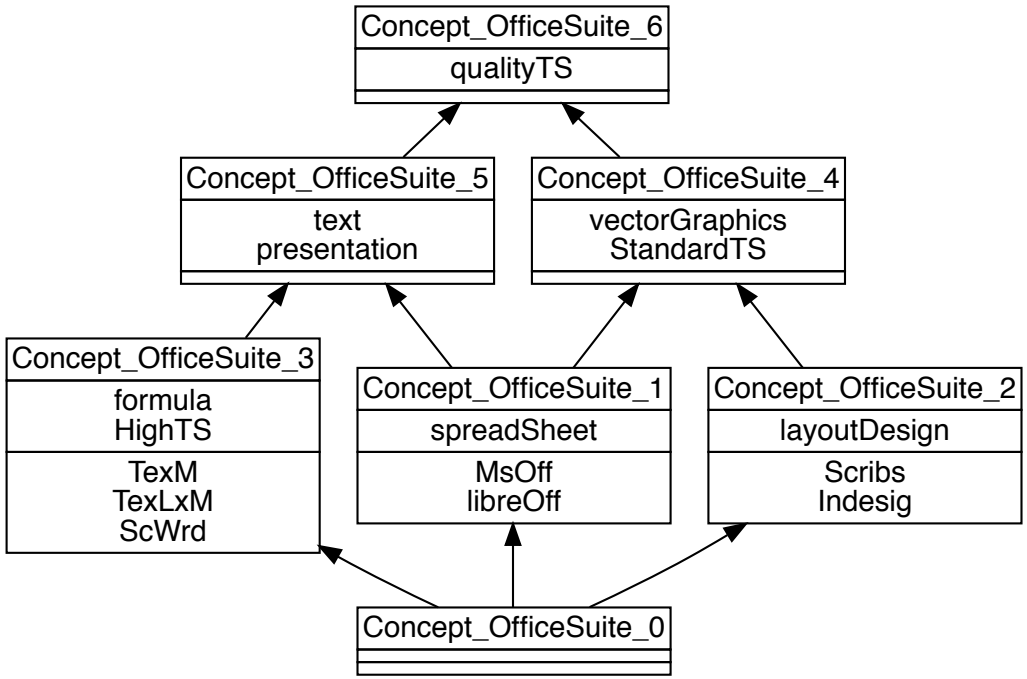
Both introduced in
Concept_OfficeSuite_5

text $\Leftrightarrow$ presentation

FCA

Variability extraction

Mutex



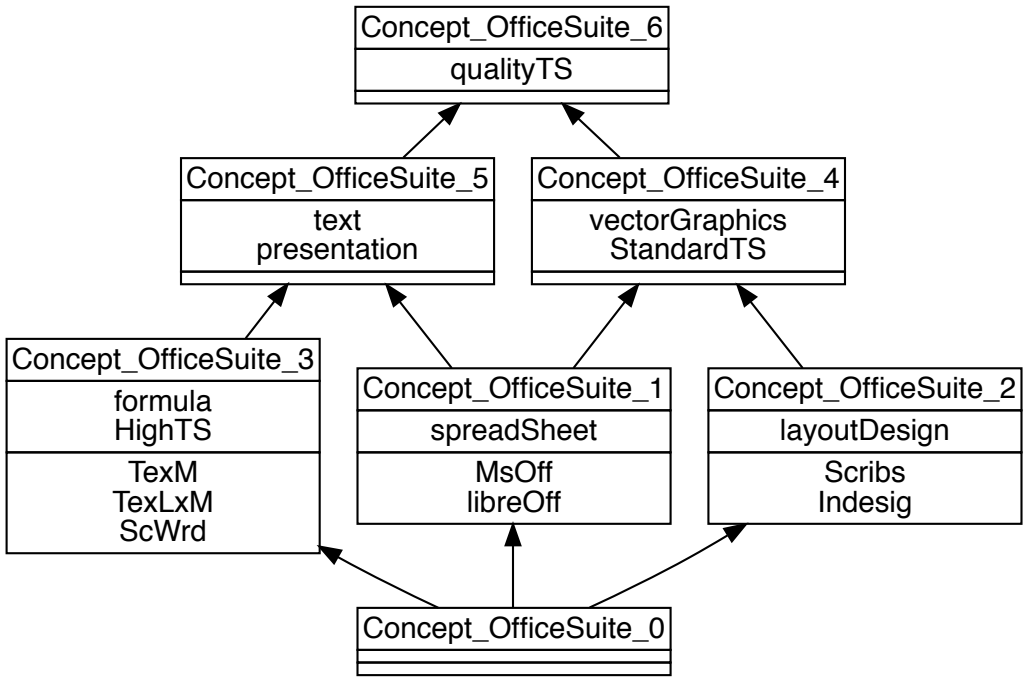
Concept_OfficeSuite_3
and
Concept_OfficeSuite_2
have *no common subconcept introducing a configuration*

formula $\Rightarrow \neg$ layoutDesign

FCA

Variability extraction

Xor Groups



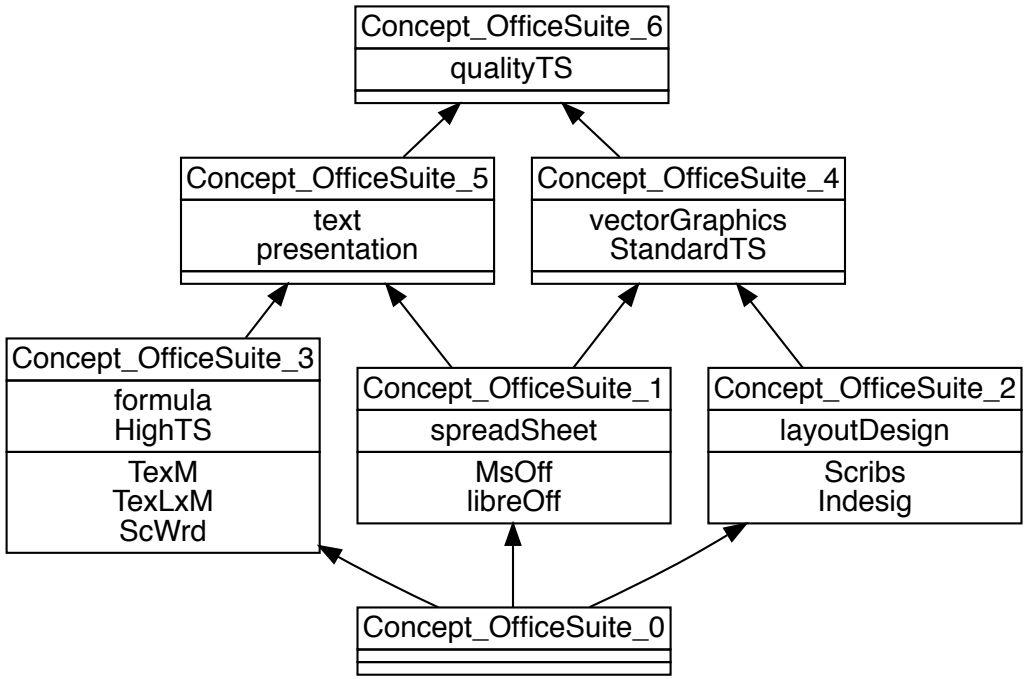
Configurations of
Concept_OfficeSuite_3
and
Concept_OfficeSuite_4
form a partition of
the configurations of
Concept_OfficeSuite_6

qualityTS
Xor
{ HighTS, StandardTS }

FCA

Variability extraction

Candidate Or Groups

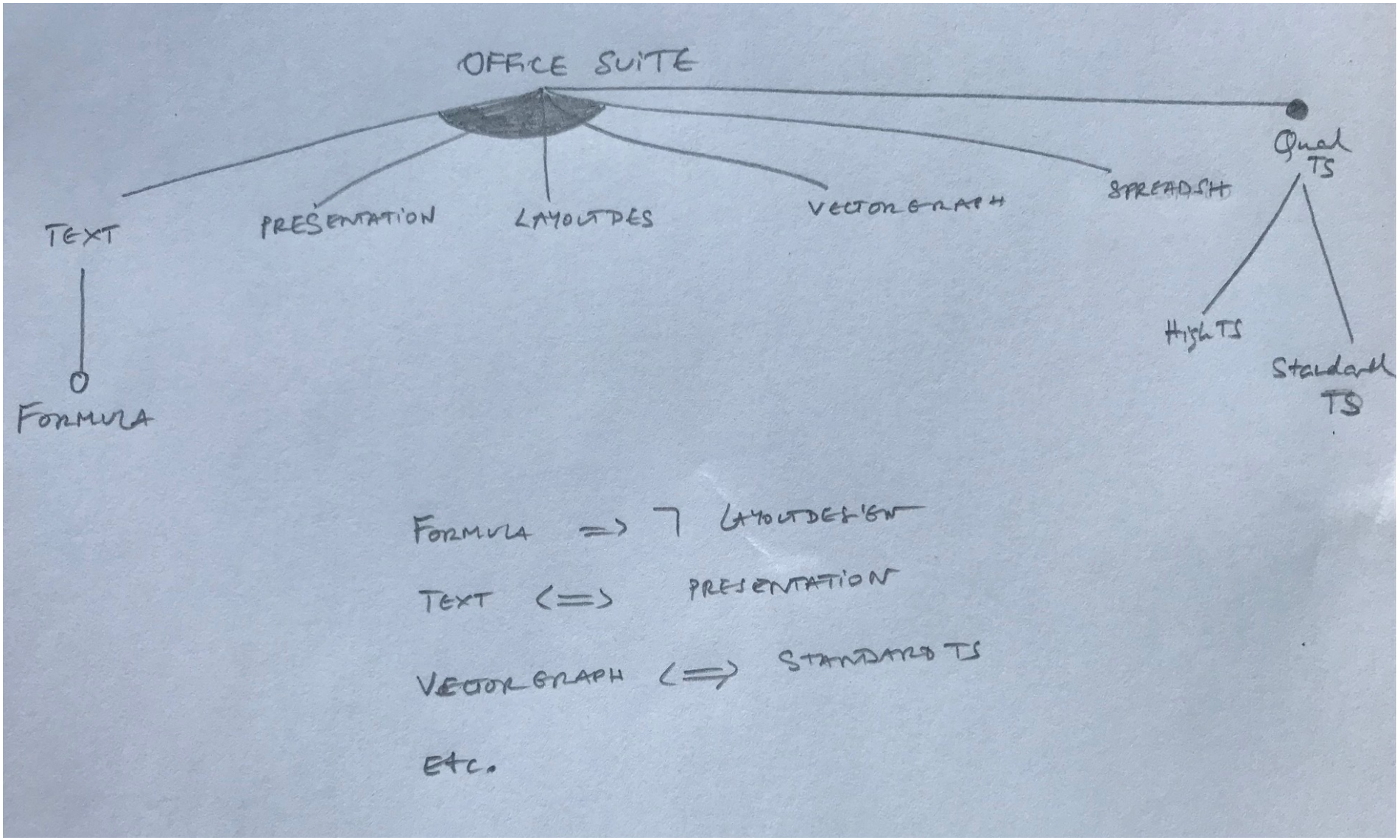


Configurations of
Concept_OfficeSuite_6
are covered by
non disjoint union
of these of
Concept_OfficeSuite_5
and
Concept_OfficeSuite_4

Office suite
Or
{ text, vector graphics }

Candidate Feature Model

Graphical view of variability of descriptions Software × Feature



Outline

Requirement engineering

Principales techniques existantes

- *use cases* introduits en 1987 par I. Jacobson
- scripts et scénarios (J. Hooper)
- langages formels, comme les réseaux de Petri
- langages de modélisation de buts comme KAOS (A. Dardenne) ou I* (J. Mylopoulos)
- diagrammes de cas d'utilisation en UML (1995) et méthodologie décrite dans le Unified Process (1997)
- templates de description de use case par A. Cockburn (2001)
- récits utilisateurs (user stories), cartes de récits (K. Beck 1999, M. Cohn 2004)
- adaptation des use cases au contexte AGILE, SCRUM (Use Case 2.0, I. Jacobson et al. 2011): User stories et Epics
- cartographie des récits (J. Patton, 2014)

Requirement engineering

User stories

- Description simple d'une exigence
  - Titre, niveau de priorité, estimation de sa complexité
  - Format "En tant que" (qui), "je souhaite" (quoi), "afin de" (pourquoi)
  - En pratique souvent réduit à "En tant que" (qui), "je souhaite" (quoi)
  - (qui) est personnage fictif représentant un utilisateur (appelé un persona)

User stories

User stories in one Manga website ([Bazin et al., 2024])

System *MyManga*

- As a *FinalUser*, I can *search*
- As a *Administrator*, I can *search*
- As a *Administrator*, I can *manage cart*
- As a *ProductManager*, I can *manage cart*

Alternative view as pairs

- (*FinalUser*, *search*)
- (*Administrator*, *search*)
- (*Administrator*, *manage cart*)
- (*ProductManager*, *manage cart*)

User story family

User stories in a family of Manga websites ([Bazin et al., 2024])

A ternary relation between:

- systems (*MyManga*, *MangaStore*, *MangaHome*)
- features (search *s*, view comment *vc*, manage cart *mc*)
- roles (*FinalUser*, *Administrator*, *ProductManager*)

Example:

- In *MyManga*, As a *FinalUser*, I can *search*
- In *MyManga*, As a *Administrator*, I can *manage cart*
- ....
- In *MangaStore*, As a *FinalUser*, I can *search*
- ....
- In *MangaHome*, As a *FinalUser*, I can *view comment*
- ....

User story family

User stories in a family of Manga websites ([Bazin et al., 2024])

A ternary relation between:

- systems (*MyManga*, *MangaStore*, *MangaHome*)
- features (search *s*, view comment *vc*, manage cart *mc*)
- roles (*FinalUser*, *Administrator*, *ProductManager*)

A view as 3-tuples:

- (*MyManga*, *FinalUser*, *search*)
- (*MyManga*, *Administrator*, *manage cart*)
- ....
- (*MangaStore*, *FinalUser*, I can *search*)
- ....
- (*MangaHome*, *FinalUser*, *view comment*)
- ....

Requirement engineering

User stories in Manga websites ([Bazin et al., 2024])

A ternary relation between:

- systems (*MyManga*, *MangaStore*, *MangaHome*)
- features (search *s*, view comment *vc*, manage cart *mc*)
- roles (*FinalUser*, *Administrator*, *ProductManager*)

Alternative view:

	s	*vc*	*mc*	*s*	*vc*	*mc*	*s*	*vc*	*mc*
MyManga	×			×		×			×
MangaStore	×			×					
MangaHome	×	×			×			×	×
	FinalUser			*Administrator*			*ProductManager*		

Variability in a User Story family

Questions of SPL domain engineering

- How to extract variability relationships?
  - $(Administrator, search) \implies (FinalUser, search)$
  - All the systems that offer the search feature to administrators also offer it to final users.
- How to represent variability relationships (variability model)?
  - Textually
  - Graphically

Questions of SPL application engineering

- How to leverage variability model to produce a new system ?

Questions explorées dans les Travaux pratiques

Questions of SPL domain engineering

- How to extract variability relationships?
- How to represent variability relationships (variability model)?
  - Textually
  - Graphically

Combinaisons de techniques à explorer

- Pour l'extraction

  LLM LLM (dialogue à inventer suivant cible de représentation)
  RCA Relational Concept Anaysis (sortie : structure conceptuelle à commenter)
  TCA Triadic Concept Analysis (sorties : liste des concepts et perspectives à commenter)

- Pour la représentation

  LLM Texte généré par LLM par des prompts dirigeant la forme du texte
  FM Feature Model annoté par les rôles
  UCV Diagramme de cas d'utilisation UML variable

Jeux de données

Tirés de https://doi.org/10.18167/DVN1/BWCC71

- Petit exemple pour se familiariser avec le problème
- Exemple réaliste créé avec l'aide des étudiants de L2 informatique en 2023
- Fichiers à utiliser déposés sur Moodle
  - UserStoriesInitialSmall.csv
  - UserStoriesInitial.csv

Approche RCA

- Outil FCA4J
- https://www.lirmm.fr/fca4j/ mais : utiliser le jar du dossier Moodle plus à jour sur la réécriture des attributs relationnels
- Fichier rcft pour le petit exemple : websites.rcft
- D'autres modélisations sont possibles, n'hésitez pas à en proposer
- Avec FCA4J, vous pourrez construire le treillis et une base de règles

Approche TCA

- Recherchez quelques éléments pour comprendre l'approche
- Testez avec l'outil: https://fca-tools-bundle.com/
  - Se créer un compte
  - Utiliser les fonctionnalités :
    - Create dyadic or triadic contexts
    - Analyze a triadic context using a novel triadic navigation method that uses perspectives

Approche FM

- Il n'existe pas de modèle, vous devez inventer une représentation graphique appropriée inspirée des Feature Models
- La représentation peut se faire en un ou plusieurs diagrammes

Approche UCV

- Vous trouverez une source d'inspiration dans deux articles déposés sur Moodle
- Vous pouvez rechercher d'autres articles des mêmes auteurs pour avoir d'autres exemples

Organisation du travail

- Vous travaillez par groupes de 2 ou 3
- Vous vous serez inscrits pour une des combinaisons sur Moodle (s'il en manque, prévenez-moi)
- Les documents doivent être déposés sur Moodle, ne mettez pas de liens externes dans vos documents :
  - Un rapport décrivant votre travail
  - L'intégralité des fichiers de travail (données, résultats, dialogues avec le LLM).

Barbut, M. and Monjardet, B. (1970).

Ordre et classification – Algèbre et combinatoire, Tome 2.
Hachette, Paris.

Bazin, A., Georges, T., Huchard, M., Martin, P., and Tibermacine, C. (2024).

Exploring the 3-dimensional variability of websites' user-stories using triadic concept analysis.
Int. J. Approx. Reason., 173:109248.

Birkhoff, G. (1940).

Lattice Theory.
Colloquium publications. American Mathematical Society.

Ganter, B. and Wille, R. (1999).

Formal concept analysis - mathematical foundations.
Springer.

Loesch, F. and Ploedereder, E. (2007).

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.

Ryssel, U., Ploennigs, J., and Kabitzsch, K. (2011).

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.