



**SORBONNE  
UNIVERSITÉ**

## **M1 Informatique – PSTL**

### **Carnet de bord : les coulisses de la recherche documentaire**

**Noms, prénoms et spécialité :**

BRAHIMI Lounes	M1 STL
ANSARI TABRIZI Edwin	M1 STL


**Sujet :**

Localisation de features dans les applications Web

# 1. Introduction :



Tout logiciel nécessite une maintenance continue pour continuer à fonctionner et à servir le client, au cours de son cycle de vie, les développeurs exécutent des activités en constante évolution, principalement en raison du besoin continu de corriger des bogues, de mettre à jour, d'adapter le logiciel et d'améliorer sa productivité en ajoutant de nouvelles fonctionnalités, la compréhension de la structure du système logiciel est donc une nécessité avant que des modifications puissent être apportées, dans ce cas, le développeur doit identifier l'emplacement dans le code source qui correspond à une fonctionnalité spécifique, c'est ce qu'on appelle la localisation de fonctionnalités.

Notre objectif est d'implémenter une extension de VSCode (éditeur de code), qui permettra à partir d'une description textuelle d'une fonctionnalité, la localisation de tous les endroits dans le code source qui correspondent à cette dernière, notre algorithme devra être opérationnelle sur n'importe quelle application web, de nombreuses techniques de localisation d'entités ont été introduites dans la littérature, nous allons nous intéresser à l'analyse formelle de concepts (FCA) et l'analyse sémantique latente (LSI). 

## 2. Les mots clés retenus :

1. Feature location
2. Software documentation
3. Software engineering
4. Concept location
5. Program comprehension
6. Software maintenance
7. Software evolution
8. Software product lines
9. Latent Semantic Indexing
10. Term co-occurrence
11. LSI
12. LSA
13. Latent semantic analysis
14. Formal Concept Analysis
15. FCA
16. Concept lattice
17. Feature model
18. Ontologies
19. Description logics
20. Semantic operations
21. Conceptual graphs
22. Data analysis
23. Natural language processing
24. NLP
25. Vector space model
26. Term vector model
27. Static analysis
28. Dynamic analysis
29. Program understanding
30. Code comprehension

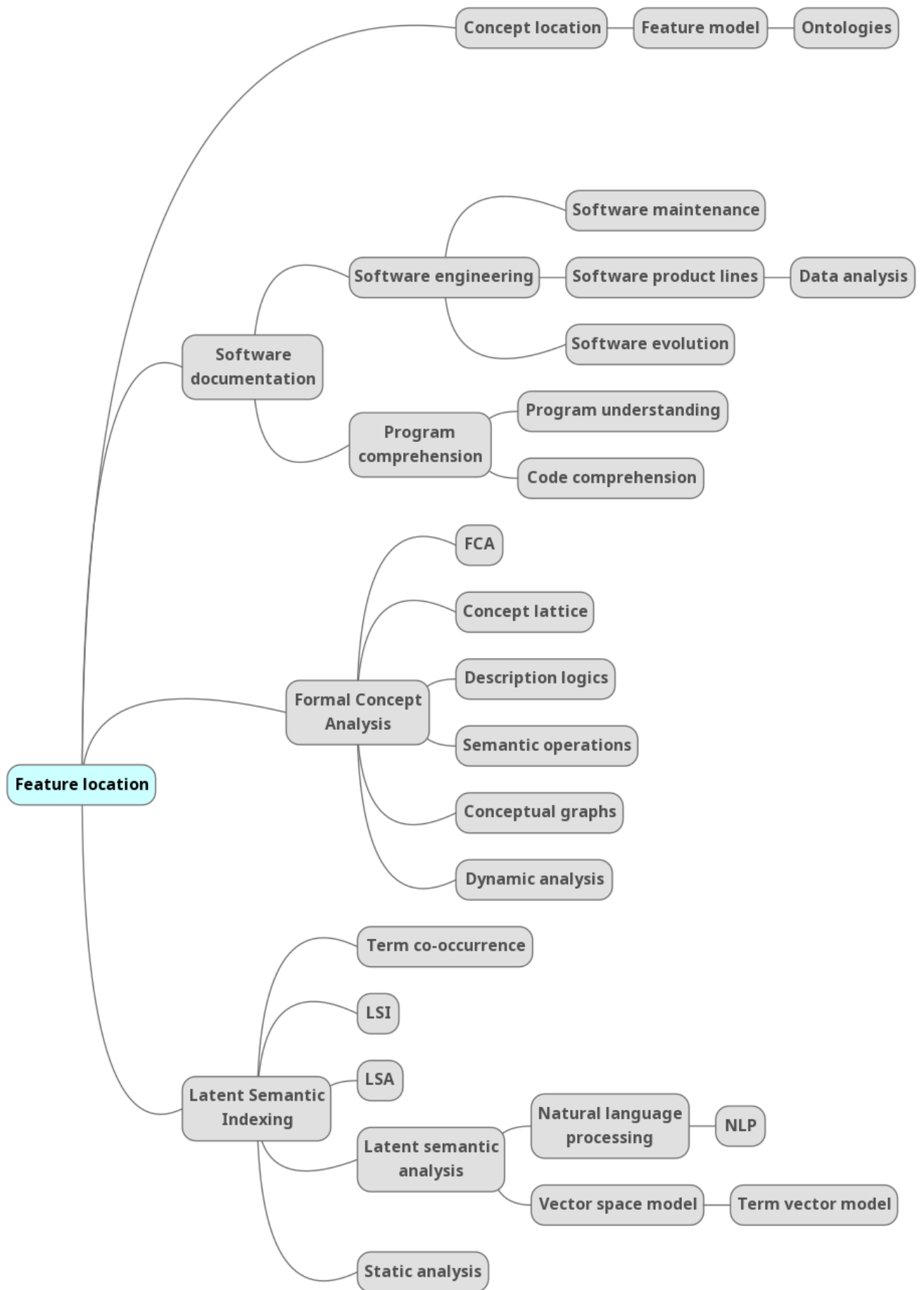


Figure 1 Carte heuristique



### 3. Descriptif de la recherche documentaire :

Nous avons commencé par chercher un article qui nous introduirait et nous présenterait le sujet de la localisation de fonctions de façon progressive en cherchant le mot-clé « feature location » sur les 3 bases de données conseillées par notre bibliothèque, cela nous a permis de plus nous familiariser avec le sujet et de découvrir de nombreux nouveaux mots-clés (exemple « LSI » et « FCA ») qui ont permis à notre recherche documentaire de prendre des directions plus spécifiques.

Nous avons cherché les divers documents scientifiques sur « la base de données de la bibliothèque de la Sorbonne université », sur « Europresse » et sur « Web of Science », on a utilisé « Google scholar » pour avoir des informations vérifiées sur les auteurs, enfin, nous nous sommes servi du logiciel Zotero pour gérer les références aux sources.

Nous avons l'habitude de faire des recherches simplement sur Google et souvent lire la synthèse que propose Wikipédia, les enseignements et astuces (exemple « Pomme de terre ») que nous a apporté les cours de la recherche documentaire nous ont permis de découvrir qu'on pouvait gagner en temps et en qualité de recherche et de travail.

### 4. Citations :

[1]-[6]




#### 4.1 Bibliographie produite dans le cadre du projet :

- [1] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, et G. Dedene, « Formal concept analysis in knowledge processing: A survey on applications », *Expert Syst. Appl.*, vol. 40, n° 16, p. 6538-6560, nov. 2013, doi: 10.1016/j.eswa.2013.05.009.
- [2] D. Poshyvanyk et A. Marcus, « Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code », 2007.
- [3] A. A. Saifan et L. Obeidat, « Feature location enhancement based on source code augmentation with synonyms of terms », *Softw. Pract. Exp.*, vol. 51, n° 2, p. 235-259, 2021, doi: <https://doi.org/10.1002/spe.2900>.
- [4] J. Martinez, T. Ziadi, M. Papadakis, T. F. Bissyandé, J. Klein, et Y. L. Traon, « Feature location benchmark for extractive software product line adoption research using realistic and synthetic Eclipse variants », *Inf. Softw. Technol.*, juill. 2018, doi: 10.1016/j.infsof.2018.07.005.
- [5] D. Poshyvanyk, M. Gethers, et A. Marcus, « Concept location using formal concept analysis and information retrieval », *ACM Trans. Softw. Eng. Methodol.*, vol. 21, n° 4, p. 23:1-23:34, févr. 2013, doi: 10.1145/2377656.2377660.
- [6] A. Kontostathis et W. M. Pottenger, « A framework for understanding Latent Semantic Indexing (LSI) performance », *Inf. Process. Manag.*, vol. 42, n° 1, p. 56-73, janv. 2006, doi: 10.1016/j.ipm.2004.11.007.

## 5. Evaluation des sources :


### 5.1 Evaluation de la source 5 :

*Concept Location Using Formal Concept Analysis and Information Retrieval*

La source est publiée le 7 février 2013, est un article de revue scientifique qu'on a trouvé en faisant notre recherche sur la base de donnée de la bibliothèque de Sorbonne université, l'auteur Andrian Marcus est professeur d'informatique  l'université du Texas à Dallas, doté d'un h-index de 54  a co-écrit l'article avec Denys Poshyvanyk et Malcom Gethers, eux aussi professeurs au Collège de William et Mary des spécialistes dans le domaine, l'article présente et nous introduit de façon élégante dans le cœur du sujet, à savoir la localisation de fonctions, l'article propose une approche intéressante, en combinant l'analyse de concept formelle qui est dynamique et l'indexation sémantique latente qui est statique dans le but d'avoir des résultats plus pertinents. 

### 5.2 Evaluation de la source 6:

*A framework for understanding Latent Semantic Indexing (LSI) performance*

La source est publiée en 2006, est un article de revue scientifique qu'on a trouvé en faisant notre recherche sur la base de donnée « Web of Science », l'auteur W. M. Pottenger est enseignant-chercheur à l'Université Rutgers et PDG d'Intuidex, qui est cités au total 2240 fois, il est co-auteur avec A. Kontostathis qui est professeur d'informatique à l'Académie navale d'Annapolis, donc deux spécialistes dans le domaine de l'informatique, l'article nous présente l'algorithme d'indexation sémantique latente (LSI) en nous expliquant qu'il traite le texte (le code source) de façon statique, ce dernier déjà doté de plusieurs modèles de compréhension, l'article propose une variante qui étudie la valeur produite dans le terme par vecteurs de dimension. 

### 5.3 Evaluation de la source 1 :

*Formal concept analysis in knowledge processing: A survey on applications*

La source est publiée le 15 novembre 2013, est la deuxième partie d'un grand document d'enquête dans lequel le concept formel Analyse (FCA) est analysé, on a trouvé en faisant notre recherche sur la base de donnée de la bibliothèque de Sorbonne université, l'auteur Guido Dedene cité 3861 avec J. Poelmans, D. I. Ignatov et S. O. Kuznetsov sont professeurs et experts dans le domaine, le document nous introduit l'historique de l'analyse formelle de concepts et nous présente différentes technique permettant de résoudre le problème de la localisation de fonctions de façon dynamique. 