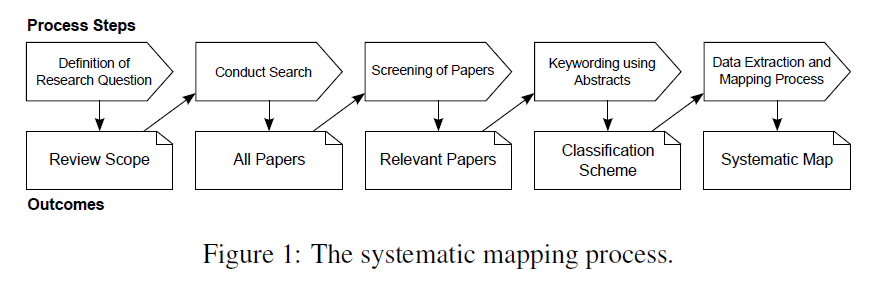
Classification systématique : utilisabilité des APIs

**Auteurs :** Selsabil Dbouba, Constantin Masson, Khaled Fayala, Maxime Gallais-Jimenez



# Définition des hypothèses de recherche

Cette classification systématique nous a été demandé dans le cours de maîtrise « Méthode empirique en génie logiciel » de Houari Sahraoui. Le sujet de notre revue est l’utilisabilité des APIs. L’utilisabilité est un sujet complet que nous allons étudier sous deux aspects. D’une première part qu’est ce qui définit l’utilisabilité et dans un second temps comment l’améliorer. Nous tenterons de répondre aux deux problématiques suivantes.­­­

RQ1 : Quels sont les **critères** d’une utilisabilité efficace d’API ?

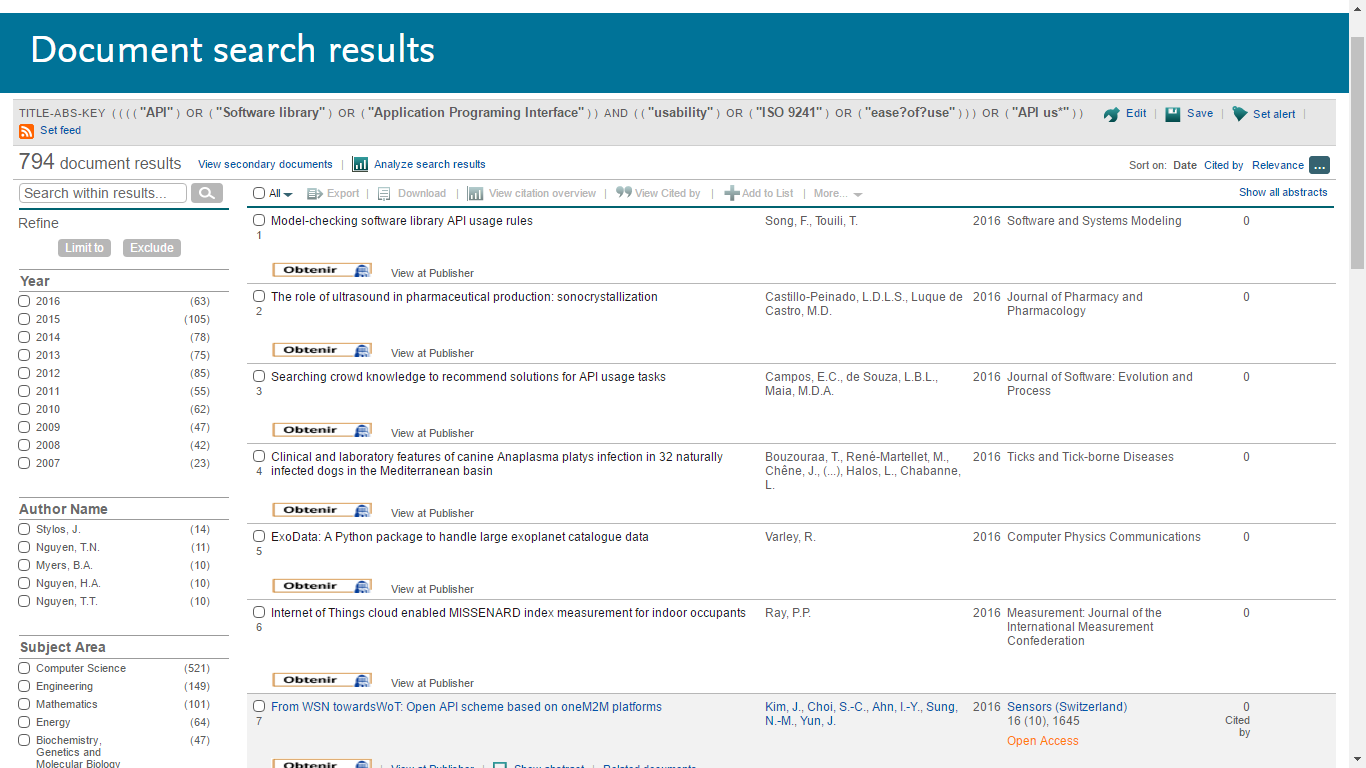
RQ2 : Comment **améliorer** l’utilisabilité d’une API ?

# Choix des mots clés

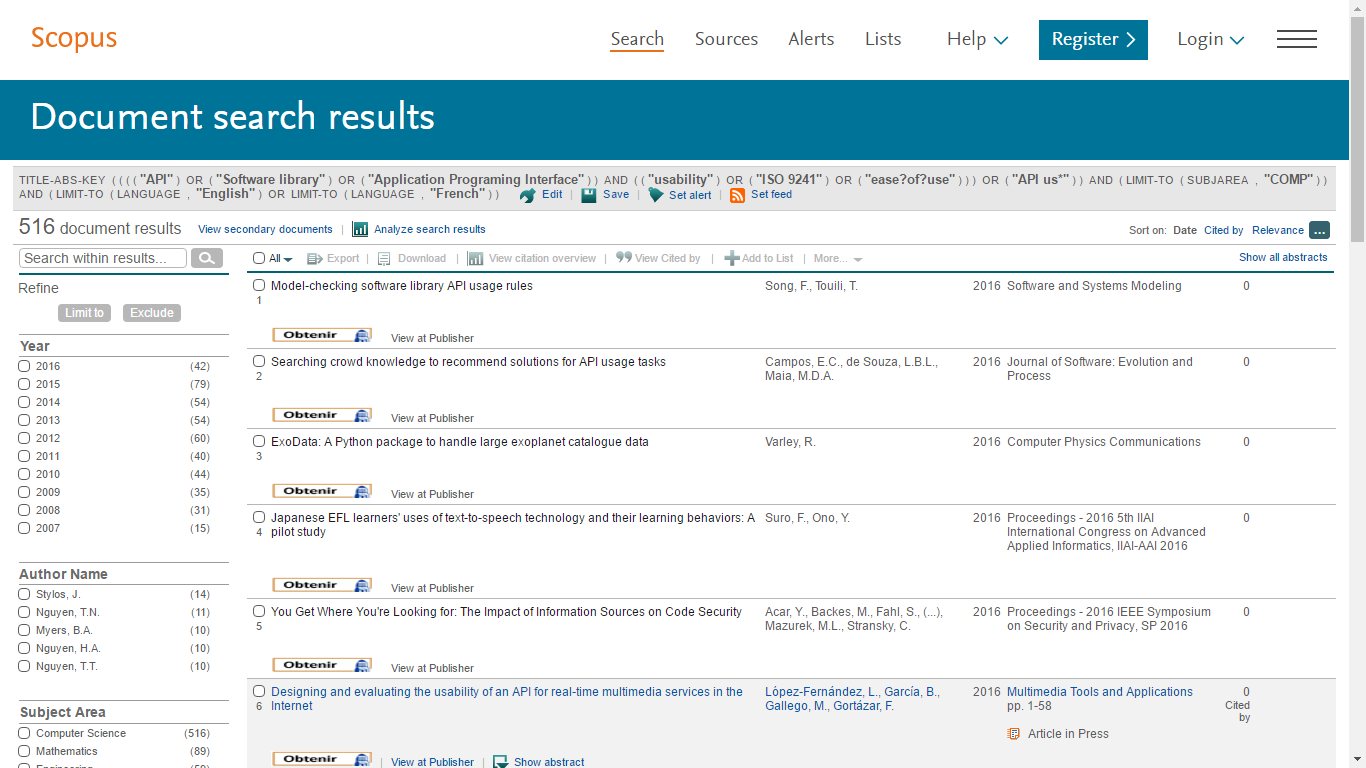
Pour les mots clés nous avons donc choisi les mots *API* et *usability* qui sont les termes de notre sujet. Puis nous avons décidé de les dériver afin de rechercher les termes : *software library* et *Application Programing Interface* en dérivé d’API et *ISO 9241* qui est la norme pour l’utilisabilité et *ease-of-use* qui est un concept qui définit bien l’utilisabilité. Nous avons ajouté à la requête *API us\** afin de pouvoir avoir les articles en fonction de de l’action d’utiliser une API. Cette recherche a été effectuée seulement dans le titre, le résumé et les mots clés de l’article.

**Requête** : *TITLE-ABS-KEY((((‘’API’’) OR (‘’Software library’’) OR (‘’Application Programming Interface’’)) AND ((‘’usability’’) OR (‘‘ISO 9241’’) OR (‘’ease ?of ?use’’)) OR (‘’API us\*’’)*

Cette requête nous a permis d’avoir une sélection de 794 articles.



Puis nous avons appliqué des filtres à cette requête. Le premier filtre fût sur le domaine d’application que nous avons réduit à l’informatique (‘’computer sciences’’). Le second filtre a été mis sur la langue que nous avons réduite au Français et l’Anglais. Après filtrage il nous restait 516 documents pour l’étape suivante, le screening.



**Requête :** *TITLE-ABS-KEY((((‘’API’’) OR (‘’Software library’’) OR (‘’Application Programming Interface’’)) AND ((‘’usability’’) OR (‘‘ISO 9241’’) OR (‘’ease ?of ?use’’)) OR (‘’API us\*’’) AND (LIMIT-TO(SUBJAREA, ‘’COMP’’)) AND (LIMIT-TO(LANGUAGE, ‘’English’’) OR LIMIT-TO (LANGUAGE, ‘’French’’)*

# Screening

Suite au screening nous avons obtenu 56 papiers validés, 196 rejetés et 253 ambigus. Dans les 253 ambigus 91 avait été validés par un membre de l’équipe et refusé par un autre. Après réunion de l’équipe nous avons gardé 17 articles. Nous allons travailler sur 73 articles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Acceptés | Ambigus | Rejetés | A discuter | Après conflit |
| 56 | 253 | 196 | 91 | 73 |

# Liste des articles

* A Case Study of Web API Evolution
* A case study of API redesign for improved usability
* A cooperative approach for combining client-based and library-based API usage pattern mining
* A dataset for API usage
* A framework for extending usability engineering: API usability essentials: Extending usability via component-based platform
* A study of modern Linux API usage and compatibility: What to support when you're supporting
* ACUA: API change and usage auditor
* An API design process in terms of usability: A case study on building more usable apis for smart TV platform
* An exploratory study of API usage examples on the web
* An observational study on API usage constraints and their documentation
* API peer reviews: A method for evaluating usability of application programming interfaces
* API usability peer reviews: A method for evaluating the usability of application programming interfaces
* API usability: CHI'2009 special interest group meeting
* Applying the cognitive dimensions of API usability to improve API documentation planning
* Approximate structural context matching: An approach to recommend relevant examples
* An empirical comparison of static and dynamic type systems on API usage in the presence of an IDE: Java vs. Groovy with eclipse
* An empirical study of API usability
* An exploratory study of api changes and usages based on apache and eclipse ecosystems
* Automated measurement of API usability: The API Concepts Framework
* Automatic evaluation of API usability using complexity metrics and visualizations
* Automatic parameter recommendation for practical API usage
* Approximate structural context matching: An approach to recommend relevant examples
* Building more usable APIs
* Constructing usage scenarios for API redocumentation
* Context-sensitive code completion tool for better API usability
* CriticAL: A critic for APIs and libraries
* Designing and evaluating the usability of an API for real-time multimedia services in the Internet
* Detecting inefficient API usage
* Evaluating the reference and representation of domain concepts in APIs
* Generating API-usage example for project developers
* Improving API documentation usability with knowledge pushing
* Improving API documentation using API usage information
* Improving API usability
* Influence of code completion methods on the usability of APIs
* Influencing factors on the usability of API classes and methods
* Informing API design through usability studies of API design choices: A research abstract
* Linking usage tutorials into API client code
* MAPO: Mining API usages from open source repositories
* Mapping the space of API design decisions
* Measurable concepts for the usability of software components
* Measuring API Usability
* Methods towards API usability: A structural analysis of usability problem categories
* Mining API patterns as partial orders from source code: From usage scenarios to specifications
* Mining API usage examples from test code
* Mining API usage patterns by applying method categorization to improve code completion
* Mining multi-level API usage patterns
* Neural networks for sentiment analysis on Twitter
* Obstacles in using frameworks and APIs: An exploratory study of programmers' newsgroup discussions
* Prose + test cases = specifications
* Recommending API usages for mobile apps with hidden Markov model
* Recommending posts concerning API issues in developer Q&A sites
* Recommending proper API code examples for documentation purpose
* Recommending reference API documentation
* Searching the State Space: A Qualitative Study of API Protocol Usability
* Some structural measures of API usability
* Statistical learning approach for mining API usage mappings for code migration
* Synthesizing API usage examples
* Temporal analysis of API usage concepts
* The API walkthrough method: A lightweight method for getting early feedback about an API
* The concept maps method as a tool to evaluate the usability of APIs
* The factory pattern in API design: A usability evaluation
* The implications of method placement on API learnability
* The role of conceptual knowledge in API usability
* Type names without static type checking already improve the usability of APIs (As Long as the Type Names are Correct): An Empirical Study
* Unit tests as API usage examples
* Usability evaluation of configuration-based API design concepts
* Usability implications of requiring parameters in objects' constructors
* Usability of educational technology APIs: Findings and guidelines
* Usable results from the field of API usability: A systematic mapping and further analysis
* Useful, but usable? Factors affecting the usability of APIs
* Using concept maps to evaluate the usability of APIs
* Visualization based API usage patterns refining
* What makes APIs hard to learn? answers from developers

# Schéma de classification