



TP-RCA

Etudiants

BENGOUFA Ryan

ROUQUAIROL Lucas

Enoncé

Etudier des matrices de comparaison de produits (PCMs) afin d'en extraire de possibles relations de variabilité entre caractéristiques de la même PCM ou de différentes PCMs.

On choisit le contexte des web serveurs pour notre étude ([source](#)).

Selection et Nettoyage des Données

Avant de travailler sur les données, nous allons les traiter et les nettoyer suivant certains critères, voici des types de pré-traitement utilisés avec un exemple sur nos données.

Normalisation des Données

Des concepts similaires sont représentés de manière cohérente dans toutes les matrices. Par exemple, si une matrice utilise "Yes" et "No" et une autre utilise "Supported" et "Not Supported", ces termes doivent être normalisés en un format commun.

Server	Windows
boa	UNKNOWN
Caucho Resin...	Yes
Caudium	No
Cherokee HTT...	No
HFS	Yes
Hiawatha HTT...	with Cygwin
IBM HTTP Serv...	Yes
Internet Inform...	Yes
Jetty (Java)	Yes
Jexus	No
lighttpd	Yes

Voici le fichier qui fait correspondre les *serveurs* et les *systèmes d'exploitation compatible*. Nous devons traduire les “Yes” en 1 et les “No” en 0 pour un traitement uniforme et une compatibilité avec les outils que nous allons utiliser.

Traitement des Valeurs Manquantes ou Inconnues

Dans certains cas, les valeurs inconnus peuvent être considéré comme des absences (0), ou si elles sont trop nombreuses, de supprimer la caractéristique.

Entrepreneur	1	0	1	0	1	0	1	0	1	0
Mongoose	1	1	1	1	1	0	1	0	1	0
Monkey HTTP...	0	1	0	0	0	0	0	0	0	0
NaviServer	1	1	Unknown	Unknown	1	Unknown	Unknown	Unknown	Unknown	Unknown
NCSA HTTPd	Unknown	1	0	1	1	1	1	0	0	Unknown
nginx	1	1	1	1	1	1	0	0	0	0
Openlink	1	1	1	1	1	1	0	0	0	Unknown

Le serveur **NaviServer** possède beaucoup de champs “Unknown”, nous avons donc choisis de ne pas le conserver dans notre étude.

Utilisation de **ChatGPT** : Grace au modèle de langage chatGPT 4, nous avons pu avoir plus de précisions sur les valeurs “Unknown” ou sur les valeurs ambiguës en lui donnant directement le fichier.csv et avec ce prompt : “Dans le fichier que je viens de te donner, à chaque qui n'est pas "0" ou "1", donne moi plus d'informations sur le sujet afin de standardiser mes données.”

Voici une autre situation dans laquelle la valeurs est ambiguë.

Server	Windows
Boa	UNKNOWN
Caucho Resin...	Yes
Caudium	No
Cherokee HTT...	No
HFS	Yes
Hiawatha HTT...	with Cygwin
IBM HTTP Serv...	Yes
Internet Inform...	Yes
Jetty (Java)	Yes
Jexus	No
lighttpd	Yes

Dans notre cas, nous avons un serveur **Hiawatha HTTP** qui est compatible sur Windows en passant par l'environnement **Cygwin**. Nous allons donc faire le choix de considérer **Hiawatha** comme compatible avec Windows.

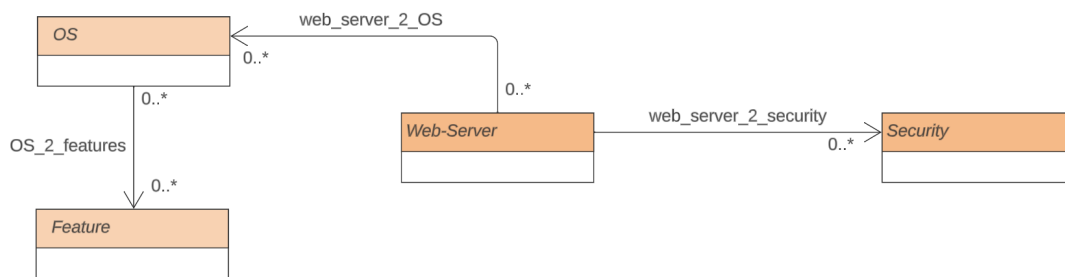
Un autre exemple avec les systèmes de sécurités quand le serveur est exécuté directement depuis le terminal, le niveau de sécurité dépend des droits de l'utilisateur qui exécute le programme. Cette donnée n'est pas très pertinente et ne coïncide pas avec nos autres données, nous choisissons donc de supprimer cette colonne.

Runs in user space or kernel space
user
user
user
user
user
user
user
user
user
user
kernel and user
user
user
user
user
user
user
user
user
user
Unknown
user
user
user
user
kernel

Pour conclure sur les données :

- **OS.csv** : Listes des OS
- **security.csv** : Sécurité supporté par les serveurs web
- **feature.csv** : Fonctionnalité disponible
- **web_server.csv** : Liste des serveurs web
- **web_server_2_security.csv** : Sécurité supporté par les serveurs web
- **web_server_2_OS.csv** : Compatibilité entre les serveurs web et les OS
- **OS_2_feature.csv** : Relation entre les OS et certaines features

Modèle UML



Extraction d'informations

Ajout des éléments dans la family :

- `java -jar fca4j.jar family web-serverRCFT.rcft -a IMPORT web_server.csv -n web_server -x CSV -s SEMICOLON`
- `java -jar fca4j.jar family web-serverRCFT.rcft -a IMPORT OS.csv -n OS -x CSV -s SEMICOLON`
- `java -jar fca4j.jar family web-serverRCFT.rcft -a IMPORT security.csv -n security -x CSV -s SEMICOLON`

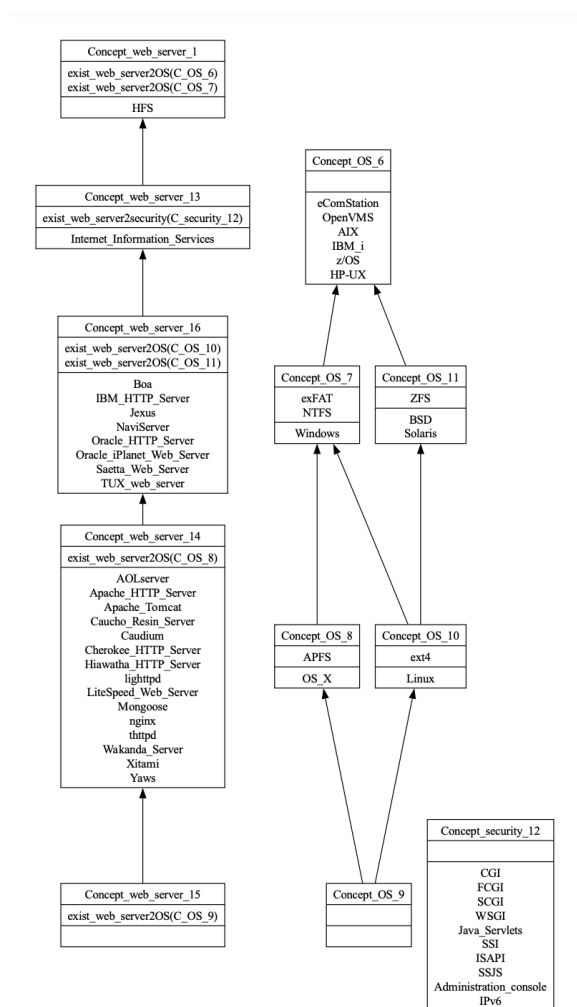
Ajout des relations :

- `java -jar fca4j.jar family web-serverRCFT.rcft -a IMPORT web_server_2_security.csv -n web_server2security -x CSV -v -s SEMICOLON -op exist -source web_server -target security`
- `java -jar fca4j.jar family web-serverRCFT.rcft -a IMPORT web_server_2_OS.csv -n web_server2OS -x CSV -v -s SEMICOLON -op exist -source web_server -target OS`

Analyse RCA :

- `java -jar fca4j.jar RCA ./web-serverRCFT.rcft ./results -v -a ADD_EXTENT`
- `dot -Tpdf results/step3.dot -o results/resultat.pdf`

Graphe de la Famille de Structures Conceptuelles



Voici le graphe extrait de l'analyse RCA de nos données qui illustre les relations entre différents concepts de serveurs web, de systèmes d'exploitation (OS), et de sécurité.

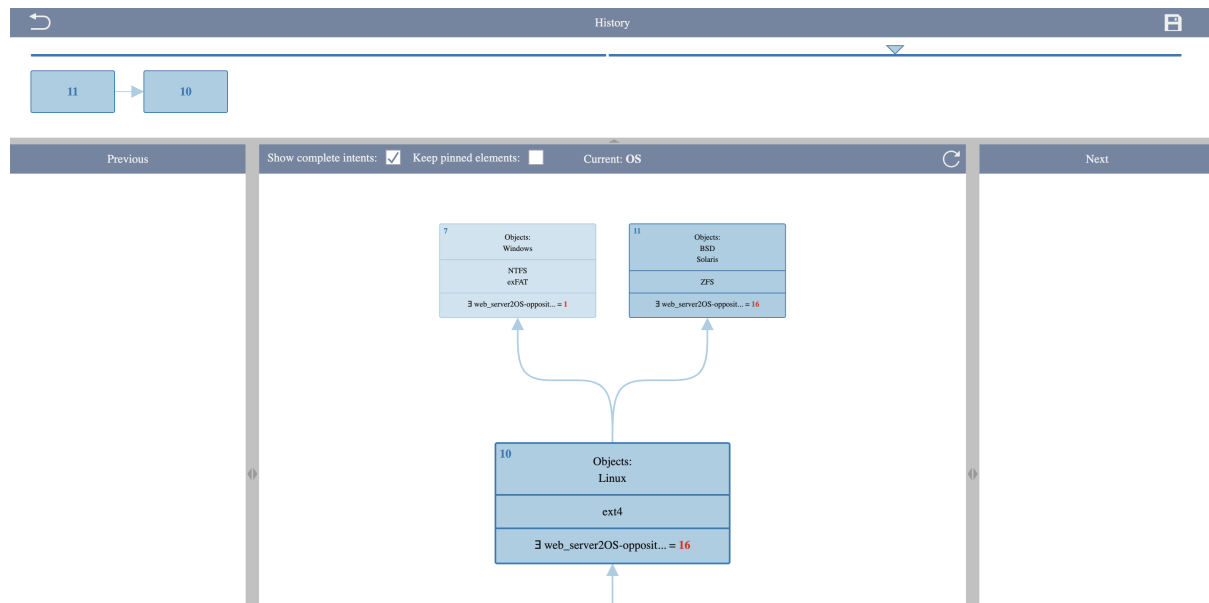
Voici quelques informations que l'on peut déduire :

- Les systèmes d'exploitation sont classés en différents concepts, comme **Concept_OS_7** pour **Windows**, ou **Concept_OS_10** pour **Linux**. Ces regroupements permettent de conclure que les OS **Windows** et **Linux** partagent des caractéristiques en commun.
- Le fait que **Concept_web_server_1** soit lié à **c_OS_6** et **c_OS_7** indique qu'il existe une variabilité dans les systèmes d'exploitation qui peuvent

être utilisés avec ce type de serveur web.

Bonus

Nous pouvons utiliser l'outil **rcaviz** pour naviguer dans le fichier **json** généré.



- Nous sommes partis du système de gestion de fichier **ZFS** supporté par les OS **BSD** et **Solaris**
- Pour arriver à **ext4** car ces **ext4** et **ZFS** sont supporté par l'OS **Linux**
- De plus, on s'aperçoit que **Linux** supporte aussi le **NTFS** et le **exFAT** qui sont eux même supporté par **Windows**

Discussion

Même si le jeu de donnée n'est pas totalement représentatif et un peu léger, ce modèle pourrait être utilisé pour la prise de décisions en matière d'architecture logicielle, pour identifier les configurations possibles des serveurs web et pour comprendre les contraintes entre les différents composants d'un système informatique.

Source

Comparison of web server software

Web server software allows computers to act as web servers. The first web servers supported only static files, such as HTML (and images), but now they commonly allow embedding of server side applications.

↗ https://en.wikipedia.org/wiki/Comparison_of_web_server_software

Comparison of Operating Systems

https://eylenburg.github.io/os_comparison.htm