

Gestion de projet – Projet Développement Logiciel (PDL)

Projet #1: « OpenCompare et Rétro-ingénierie »

Ressources :

<http://mathieuacher.com/teaching/PDL/>

opencompare.org a pour but d'aider une communauté d'utilisateurs à importer, éditer, visualiser, et exploiter des *matrices de comparaison*, typiquement pour choisir des produits dans un domaine donné.



« rétro-ingénierie du JSON à partir du code JavaScript d'OpenCompare »

Le "format" PCM (Product Comparison Matrix) est caché dans le code.

En sortie de ce projet j'attends

- (1) une petite API écrite en JAVA capable de parser le JSON du nouveau opencompare et de le transformer en un objet Java "PCM"
- (2) une documentation (incl. schema JSON, diag. classe UML, etc.) de sorte d'expliquer le format

L'objectif de ce projet de PDL est d'effectuer une rétro-ingénierie (reverse engineering en anglais) de opencompare, i.e., étudier opencompare pour en déterminer le fonctionnement interne. Plus spécifiquement on s'intéresse au format JSON utilisé en interne par opencompare pour représenter les matrices.

Ce format, matérialisé par du JSON (JavaScript Object Notation), est aussi appelé « product comparison matrix » (PCM), car il sert à représenter des données tabulaires qui permettent de comparer des produits en fonction de certaines caractéristiques (« features »).

Un exemple de PCM est donné ci-dessous. Il est stocké dans une base de données et différents services (configurateur, « chart », tri, etc.) ont été implémenté en JavaScript.

Model	Image processor	Sensor format	Sensor type	Sensor manufacturer	Megapixels
D1	-	APS-C	CCD	Sony	2.66
D100	-	APS-C	CCD	Sony	6.1
D1H	-	APS-C	CCD	Sony	2.7
D1X	-	APS-C	CCD	Sony	5.3
D200	-	APS-C	CCD	Sony	10.2
D2H	-	APS-C	JFET-LBCAST	Nikon	4.1
D2Hs	-	APS-C	JFET-LBCAST	Nikon	4.1
D2X	-	APS-C	CMOS	Sony	12.4
D2Xs	-	APS-C	CMOS	Sony	12.4
D3	EXPEED	Full-frame	CMOS	Nikon	12.1
D300	EXPEED	APS-C	CMOS	Sony	12.3
D3000	EXPEED	APS-C	CCD	Sony	10.2
D300S	EXPEED	APS-C	CMOS	Sony	12.3
D3100	EXPEED 2	APS-C	CMOS	Nikon	14.2
D3200	EXPEED 3	APS-C	CMOS	Nikon	24.2
D3300	EXPEED 4	APS-C	CMOS	Nikon	24.2
D3400	EXPEED 4	APS-C	CMOS	Nikon	24.2
D3S	EXPEED	Full-frame	CMOS	Nikon	12.1
D3X	EXPEED	Full-frame	CMOS	Sony	24.5

En l'état, le format PCM est caché dans le code JavaScript. C'est une limitation, car deux problèmes (au moins) se posent:

- Il est difficile d'inter-opérer opencompare avec d'autres outils, par exemple une API Java à même de traiter ces matrices ;
- Il est difficile pour un nouveau développeur de comprendre le format, car il n'est pas explicitement documenté

Nous proposons de remédier à ces deux problèmes de la manière suivante :

- Analyser le code source d'opencompare et identifier la structure du PCM/JSON utilisé en interne
- Récupérer quelques PCM/JSON existants
- Ecrire une API Java capable de (1) « parser » ces PCM (en JSON) et (2) les représenter sous la forme d'objets Java à même de structurer convenablement l'information d'un PCM. Vous devez donc concevoir un ensemble de classes Java pour représenter une matrice de comparaison de produits. A ce titre, on pourra de nouveau étudier le code JavaScript et en première approximation on essaiera de créer un ensemble de classes/attributs/méthodes similaires (e.g., avec des concepts similaires)
- Produire le diagramme de classes UML de votre API Java
- Produire une documentation en anglais du format PCM et de votre API Java en utilisant ce diagramme de classes et éventuellement un « schema » JSON
- Comparer votre API Java avec l'API Java d'une ancienne version d'OpenCompare (<https://github.com/OpenCompare/OpenCompare>)
- Formuler des recommandations sur le futur du format PCM (basées sur votre expérience)

Comment commencer ?

Le code source de opencompare est ici : <https://github.com/OpenCompare/OpenCompare2>

Installer et lancer le¹. Importer quelques matrices. Etudier le fonctionnement général, y compris sur l'aspect base de données.

Assez rapidement, étudier le code source JavaScript et quelques exemples de PCM. Faire une première itération du diagramme de classes UML. Puis compléter en poursuivant votre analyse. Ecrire l'API Java et itérer. Tester.

En parallèle de vos expérimentations et de votre rétro-ingénierie, écrire le document de spécification.

¹ Il faut installer MongoDB et Node/npm : de fait je recommande vivement de le faire sur un ordinateur personnel, portable