Synthèse

Lien du fichier original :
https://docs.google.com/document/d/19yPeCSmRtny1Vnwyh6tGIc8flzw_K5oXS4dh6mMAgoE/edit?usp=sharing

Définition

La variabilité est la capacité qu'a un système à être modulable. C'est-à-dire que celui-ci peut facilement proposer certaines fonctionnalités plutôt que d'autres en fonction de certains paramètres. Ces paramètres peuvent être par exemple : le contexte dans lequel il est exécuté, une configuration, une interopérabilité avec des systèmes externes.

Exemples

- Le noyau Linux propose de la variabilité grâce au système de macro *#ifdef* qui va indiquer au préprocesseur le contexte dans lequel on va compiler le noyau.
- Shopify est un site internet proposant la conception d'une application marchande. L'utilisateur choisit les fonctionnalités qu'il veut retrouver dans son application. Celle-ci propose un code variable qui s'adapte en fonction des besoins de l'utilisateur.
- Un système de livraison est variable car il doit pouvoir s'adapter aux interfaces des différents transporteurs.
- jfreechart est une bibliothèque d'affichage de diagramme. Ce système est variable car il propose l'affichage de graphiques de différentes formes : comme histogramme, diagramme circulaire…

Contexte

La détection de la variabilité dans le code permet de savoir quelle partie du code gère la modularité du système. En mettant en évidence la variabilité d'un système on peut ainsi comprendre plus facilement le code pour le maintenir ou ajouter facilement de nouvelles fonctionnalités.

Problématique

Comment détecte-t-on la variabilité d'un système ?

Caractéristique d'un code variable

Les systèmes variables comportent des caractéristiques communes.

- Des points communs : le point de variation

- Les fonctionnalités : les variant

L'allégorie de la symétrie montre comment la variabilité de certains systèmes peut être complexe. Dans une figure symétrique (=système variable) nous avons un axe/point de symétrie (=point de variation) et chaque partie de la structure (=variants). Une figure symétrique peut contenir plusieurs points de symétrie qui peuvent être des axes ou des points. Dans un système variable certains modules peuvent impliquer l'utilisation d'autres modules ou ils peuvent dépendre de certains modules.

Détection de la variabilité dans un code orienté objet

Dans une base de code orienté objet la variabilité est appliquée grâce à des mécanismes du langage et des patterns de conception comme l'héritage, la surcharge, Stratégie, Factory...
Ces mécanismes peuvent être utilisés pour autre chose. C'est en analysant la relation entre les classes concernées que l'on peut savoir s' ils sont utilisés pour mettre en œuvre de la variabilité. Ainsi en trouvant ces mécanismes en les analysant on peut détecter la variabilité

Symfinder

Symfinder permet de parser du code écrit en java pour détecter les patterns. En analysant le code et les relations entre les classes, il peuple un graphe mettant en évidence les points de variation.

Sujet

L'objectif de ce TER est de proposer une approche de détection de la variabilité pour des bases de code en javascript/typescript. Tout d'abord il faut comprendre comment celle-ci est implémentée dans le langage. Quelles sont les bonnes pratiques du code pour créer cette variabilité. Ensuite il faut vérifier si ces pratiques sont appliquées par les développeurs. Puis il faut proposer un outil permettant de détecter la variabilité à l'aide d'un parser, pour remplir le graphe de symfinder. Enfin, la méthode et le prototype doivent pouvoir être validés sur des bases de code variées.

Planning

semaine 39 : détermination des issue + synthèse
semaine 40 : état de l'art
Semaine 41 : étude de symfinder
Semaine 42 : étude des patterns en typescript / validation des pattern sur Github
Semaine 43 : étude des parsers en javascript
Semaine 44 - 45 : prototypage programme utilisant le pattern + programme qui le parse
Semaine 46 - 47 : réalisation du détecteur de variabilité
Semaine 48 - 49 : validation
Semaine 50 : Réalisation des livrables
Semaine 51 + : Finition des livrables