

Stage : Extraction d'exigences à partir de code et d'information de git (6 mois, master 2 recherche)

Contexte

Dans le monde de l'ingénierie système dirigée par les modèles, la route qui part des idées premières et se termine par la mise en exploitation du système est parsemée de différents modèles liés par de la traçabilité. La première phase consiste à réaliser un cahier des charges, étude de faisabilité puis la réalisation proprement dite du système.

Le système réalisé est souvent éloigné du cahier des charges réalisé au début. Cela est dû aux aléas des projets et aussi lié au manque de temps des mise à jour du cahier des charges ou même du maintien de la traçabilité entre les modèles.

Axes de travail

Pour adresser ce problème, nous proposons d'automatiser aux maximum ces étapes de traçabilités , non seulement au niveau modèle mais aussi au niveau du code. Nous pensons aussi que cette activité ne peut pas être fait tout au long de la conception du au contrainte projet. Nous pensons donc qu'il est indispensable d'assister les concepteurs en permettant de refaire une traçabilité de manière ponctuelle.

Pour permettre la création de lien de traçabilité et même la création d'exigence, l'idée est d'exploiter tous les artefacts de code ou de modélisation (pattern, nom des classes). D'autres artefacts pourront être utilisés, fichier texte (readme), historique du git. L'utilisation de NLP sera d'un grand apport pour exploiter ces sources d'information.

Objectif du stage

- La première partie du stage consistera d'abord à développer un mécanisme d'annotation dans le code pour faire le lien avec un corpus d'exigences. Les langages visés sont pour l'instant le python et le java.
- La deuxième partie du stage consistera à réaliser un état de l'art sur les moyens de proposer des exigences à partir des artefact existants (code, git, pattern...) lorsque les exigences n'existent pas.
- La troisième partie du stage consistera à réaliser un prototype permettant de proposer des exigences.

Compétence du stage : IA, java, python, conception orienté modèle.

[1] K. Pohl and C. Rupp, *Requirements Engineering Fundamentals - A Study Guide for the Certified Professional for Requirements Engineering Exam: Foundation Level - IREB compliant*. rockynook, 2011.

[2] Huffman Hayes, Jane, Alex Dekhtyar, and Jared Payne. "The REquirements TRacing On Target (RETRO).NET Dataset." In 2018 IEEE 26th International Requirements Engineering Conference (RE), 424–27, 2018.

[3] Sabir, Umair, Farooque Azam, Sami Ul Haq, Muhammad Waseem Anwar, Wasi Haider Butt, and Anam Amjad. "A Model Driven Reverse Engineering Framework for Generating High Level UML Models From Java Source Code." IEEE Access 7 (2019): 158931–50.

Stage : Questionnaire adaptatif pour la conception orienté modèle (6 mois, master 2 recherche)

Contexte

Dans le monde de l'ingénierie système dirigée par les modèles, la route qui part des idées premières et se termine par la mise en exploitation du système est parsemée de différents formalismes de modélisation à chacune des étapes. Chaque ingénieur se doit d'avoir une connaissance approfondie de ces formalismes, car le modèle est l'artefact principal de spécification et de validation, ainsi que la source pour la génération de code [1]. Cependant l'apprentissage de ces formalismes est souvent considéré comme une tâche fastidieuse et parfois difficile, et cela constitue un frein à l'adoption et à la généralisation de l'approche orientée modèle.

Axes de travail

Pour adresser ce problème, nous proposons de soumettre l'ingénieur système à un questionnaire adaptatif qui l'aidera à exprimer ses idées [2] (de conception, i.e. ce qu'il veut ou à l'intention de faire !) et les lui restituera de manière structurée sous la forme de modèles (concept de « Mind Twin », le jumeau numérique de ses idées) dans un paradigme adapté à son domaine métier, et compréhensible par lui et la grande majorité des parties prenantes. Ce questionnaire permettra à l'ingénieur système de définir et/ou préciser sa problématique, tout en construisant automatiquement une représentation qui pourra être exploitée pour identifier des (sous-) structures reconnaissables ou tout du moins similaires dans une base de modèles de système (heuristiques à déterminer). Et récursivement, on utilisera ces éléments de modélisation bien (re-) connus pour pouvoir adapter (orienter) les (futurs) questions afin de renforcer la confiance dans la bonne compréhension de ses intentions.

Objectif du stage

- La première partie du stage consistera à réaliser un état de l'art : lister les catégories de problèmes dont on connaît les solutions liées à une famille de modèles connus et dérouler les heuristiques permettant de poser les bonnes questions. Etudier les décompositions du problème et inversement la composition des solutions (décompositions de système en boîte ou suite de processus).
- La deuxième partie du stage consistera à réaliser un prototype en ayant choisie une catégorie et une heuristique.

.Selon les résultats du stage, Le stage pourra éventuellement constituer une seule et même thèse.

Références

[1] Heijstek, W., and M. R. V. Chaudron. "The Impact of Model Driven Development on the Software Architecture Process." In 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 333–41, 2010.

[2] Floyd, Robert W. "TOWARD INTERACTIVE DESIGN OF CORRECT PROGRAMS." In Readings in Artificial Intelligence and Software Engineering, edited by Charles Rich and Richard C. Waters, 331–34. Morgan Kaufmann, 1986.

