

Projet de *Finite State Machine*

Génération automatique de code

Objectif

L'objectif pédagogique de ce projet est de vous faire manipuler des Finite State Machines afin de vous donner les réflexes qui vous permettront de les utiliser dans le reste de votre cursus et bien sûr au delà. L'objectif technique du projet est de vous faire manipuler le DOM des FSM dans le but de générer automatiquement du code fonctionnel. Derrière cet objectif premier qu'est la génération de code, l'objectif est de vous amener à réfléchir sur la mise en œuvre des FSMs dans un domaine d'ingénierie donné. Vous devrez donc replacer votre projet dans un contexte applicatif donné et illustrer son utilisation. À la fin du projet vous devez avoir compris d'une part l'impact de l'expressivité des FSMs sur la facilité/complexité de mise en œuvre, et d'autre part comment l'utilisation de FSMs peut aider à structurer le code de contrôle d'une application (que ce code soit généré automatiquement ou écrit à la main).

Exigences techniques

Peu de contraintes vous sont imposées pour ce projet. La contrainte majeure est que les FSMs à partir desquelles vous générerez le code seront conformes au standard SCXML (<https://www.w3.org/TR/scxml/>). Vous avez libre choix :

- du langage cible dans lequel vous allez générer votre code,
- du domaine applicatif permettant de montrer des exemples de mise en œuvre de votre générateur de code,
- des technologies utilisées pour générer votre code¹

Il est bien sûr important de bien réfléchir aux choix que vous faites et à l'impact qu'ils ont sur le code généré. (par exemple, quel est l'impact si le langage cible n'a pas primitive de gestion du temps ? Est-ce important/bloquant dans votre contexte applicatif ou non).

Aussi vous éviterez les machines à gaz (vu que je dois pouvoir tester votre code sans passer des heures à installer des frameworks obscures) et vous vous assurerez que votre générateur marche sous Linux. Vous donnerez bien sûr les instructions nécessaires à la mise en place de l'environnement de tests et vous expliquerez aussi comment l'utiliser.

Exigences fonctionnelles

Pour avoir la moyenne, votre générateur de code devra, au minimum, permettre de générer le code d'une FSM simple (non hiérarchique, non parallèle, non temporisée, etc), telle que celle utilisée jusqu'au slide 19 du cours http://www.i3s.unice.fr/~deantoni/teaching_resources/SI4/FSM/cours/SCXML_

¹je vous ai donné des pointeurs à la fin du troisième cours

intro.pdf. Vous devrez également gérer l'intégration du code généré dans un processus de développement. Vous devrez donc mettre en place des mécanismes permettant de communiquer en entrées/sorties avec le code de votre FSM. Enfin, vous illustrerez le fonctionnement de votre générateur de code par un ou plusieurs exemples significatifs du domaine que vous avez choisi.

Un efois ce minimum réalisé, vous implémenterez, au choix, autant d'extensions que vous le désirez. Voici une liste non exhaustive d'exemples:

- support des états hiérarchiques,
- support des états parallèles,
- support des transitions temporisées,
- support d'actions élémentaires sur les transitions (*e.g.*, log, cancel),
- support avancé d'appel vers du code métier (en fonction de vos choix technologiques)
- ...

Bien sûr, ce qui est supporté par votre générateur de code doit être clairement stipulé et illustré par un exemple simple.

Rendu

Il n'est pas attendu de "rapport" au sens strict du terme, cependant il sera nécessaire de fournir un manuel utilisateur. Celui-ci peut être succinct mais il doit permettre de comprendre comment utiliser votre générateur de code et quelles en sont les fonctionnalités et limites.

Le reste des modalités de rendu seront précisées prochainement...

Évaluation

Je vous rappelle que vous êtes en contrôle continu et que l'évaluation ne se fera donc pas uniquement sur le rendu final, mais également sur la régularité de votre avancement au cours des trois prochaines semaines.