

Manuel d'utilisateur d'ACCECS

Alexis Giraudet Asma Khelifi Jean-Christophe Guérin
Geoffrey Desbrosses Léo Cassiau
Tantely Randriamaharavomanana Ugo Mahey

4 janvier 2017

Table des matières

1	Préface	2
1.1	Public visé	2
1.2	Présentation du projet ACCECS	2
1.3	Qu'est ce qu'ACCECS ?	2
2	Utilisation de base	3
2.1	Installation	3
2.1.1	Lien du dépôt des sources	3
2.1.2	Lancer le serveur	3
2.2	Génération d'un squelette de machine B	3
2.2.1	Énumération des différents types	3
2.2.2	Déclarations des variables et des propriétés du système	4
2.2.3	Squelette généré	4
3	Sujets avancés	5
3.1	Les outils développement	5
3.1.1	Une interface web	5
3.1.2	Le langage utilisée : Java	5
3.1.3	Le format des données : JSON	6
3.1.4	Le moteur de template Jtwig	6
3.2	La limite entre notre outil et la méthode B	6
3.2.1	Différencier un contrôleur et la future machine B	6
3.2.2	Les évènements	6
3.2.3	Les limites liées à la complexité du B	7

1 Préface

1.1 Public visé

Ce document est destiné aux utilisateurs du logiciel Assistance for the Construction of Critical Embedded Control Systems (ACCECS). ACCECS fonctionne sur la plupart des systèmes de type Unix et Windows ayant java d'installé.

Ce document regroupant un tutoriel d'utilisation d'ACCECS et son manuel utilisateur, est destiné à plusieurs types d'utilisateurs :

Novice Notre logiciel vous intéresse mais vous ne savez pas de quoi il en retourne ? Commencez par la section Présentation du projet ACCECS.

Utilisateur averti Vous souhaitez installer notre logiciel et l'essayer ? Il faut alors vous référer à la section Utilisation de base.

Développeur Ce projet vous intéresse et vous souhaitez modifier son fonctionnement ? Reportez-vous à la section Sujets avancés.

Enfin, ce document étant réalisé dans le cadre d'un projet universitaire, il est libre de droits.

1.2 Présentation du projet ACCECS

Le logiciel ACCECS est issu d'un projet réalisé dans un cadre universitaire, par des élèves de deuxième année du master ALMA, à la ¹faculté des sciences et techniques de Nantes. L'objectif de ce projet est de mettre en œuvre les différentes connaissances acquises durant nos différents cours afin d'atteindre un but défini en équipe. Ce projet est encadré par M. ²Attigbe et M. ³André, tous les deux enseignant-chercheur travaillant au ⁴Laboratoire Informatique de Nantes Atlantique (LINA).

1.3 Qu'est ce qu'ACCECS ?

L'objectif du projet est de réaliser, en équipe, un outil facilitant la rédaction des programmes utilisant la méthode B. La méthode B est une méthode formelle permettant d'assurer le bon fonctionnement des programmes, et notamment dans le cas qui nous intéresse ici, les programmes présents sur les systèmes embarqués. Ces systèmes possèdent des actionneurs et des senseurs, qui permettent de prendre connaissance et d'agir sur l'environnement du système. Ceci nous permet de commencer à décrire le comportement du système, appelé le contrôleur. C'est donc à partir la spécification du système embarqué que l'on commence à rédiger le programme du contrôleur. Notre outil a pour but de faciliter le passage entre la spécification du système et la spécification du contrôleur.

-
1. <http://www.univ-nantes.fr/>
 2. <http://pagesperso.lina.univ-nantes.fr/attigbe-c/>
 3. <http://pagesperso.lina.univ-nantes.fr/andre-p/fr/index.html>
 4. <https://www.lina.univ-nantes.fr/>

2 Utilisation de base

2.1 Installation

Ce projet est avant tout un projet expérimental, de ce fait l'utilisation de l'outil ACCECS n'est pas standardisée.

2.1.1 Lien du dépôt des sources

Le lien suivant contient le code source en libre accès : <https://github.com/Giraudux/ACCECS>

2.1.2 Lancer le serveur

Pour lancer le serveur qui accueille les requêtes de l'interface web, tapez la commande suivante dans le répertoire racine :

```
mvn jetty:run
```

Une fois le serveur lancé, vous pouvez accéder à l'interface à l'adresse suivante : `localhost:8080`

2.2 Génération d'un squelette de machine B

Cette section montre comment utiliser simplement notre outil.

2.2.1 Énumération des différents types

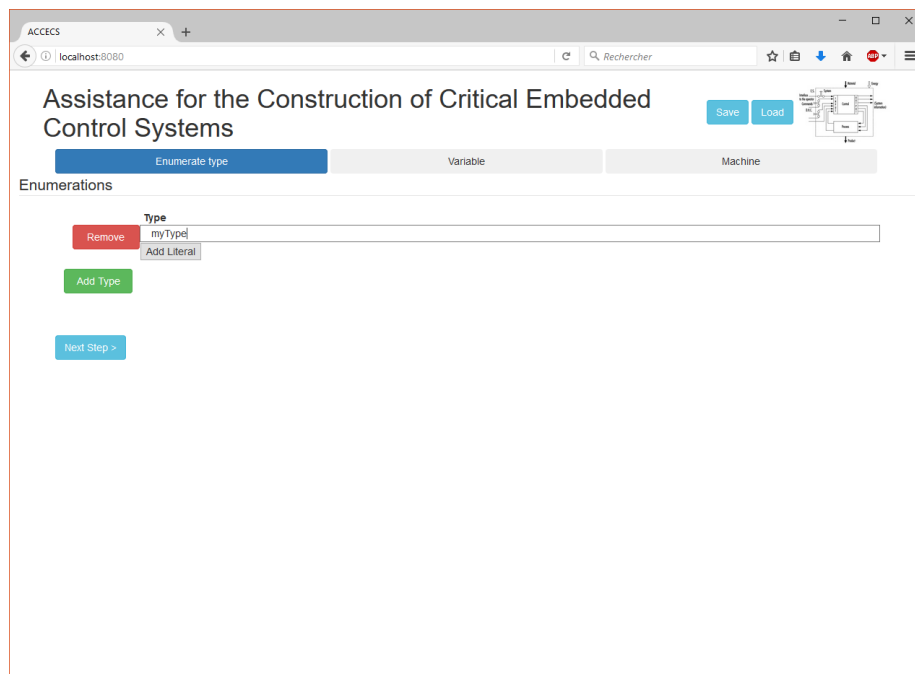


FIGURE 1 – Capture d'écran de la page 1

La première page d'ACCECS permet de spécifier des nouveaux types. Ici, un seul nouveau type est déclaré. Une fois la liste des nouveaux types déclarés, on passe à l'étape suivante.

2.2.2 Déclarations des variables et des propriétés du système

Cette page permet de déclarer la liste des variables de notre système, ainsi que les propriétés du système, que l'on a appelé ici invariant's property pour lever toute ambiguïté. Une variable a un nom, un type, une catégorie, une intervalles de valeurs possibles et une valeur par défaut. La valeur par défaut est par défaut mis à 0 pour un entier. L'image en haut à droite est là pour aider à comprendre la correspondance des variables avec ce que l'utilisateur doit rentrer comme données.

Assistance for the Construction of Critical Embedded Control Systems

Save Load

Enumerate type Variable Machine

System

M0

Variables

Name	Type	Category	Lower bound	Upper bound	Default value
Remove i1	Integer	Input	0	100	0
Remove i2	Integer	Input	-150	150	0
Remove o1	Boolean	Output			True
Remove c1	Natural	Control	0	50	10
Remove r1	Mtype	Internal			

[Add variable](#)

Invariant's properties

[Remove](#) Expression: Category:

[Add property](#)

[Next Step](#)

FIGURE 2 – Capture d'écran de la page 2

Il est possible à tout moment d'enregistrer les données présentes sur cette page avec les boutons save et load en haut à droite.

2.2.3 Squelette généré

Enfin, sur la dernière figure 3 on peut voir le squelette B généré, avec les variables, les invariants, etc. . . Au besoin, on peut régénérer le squelette ou appuyer sur des boutons pour copier/coller plus simplement le code généré.

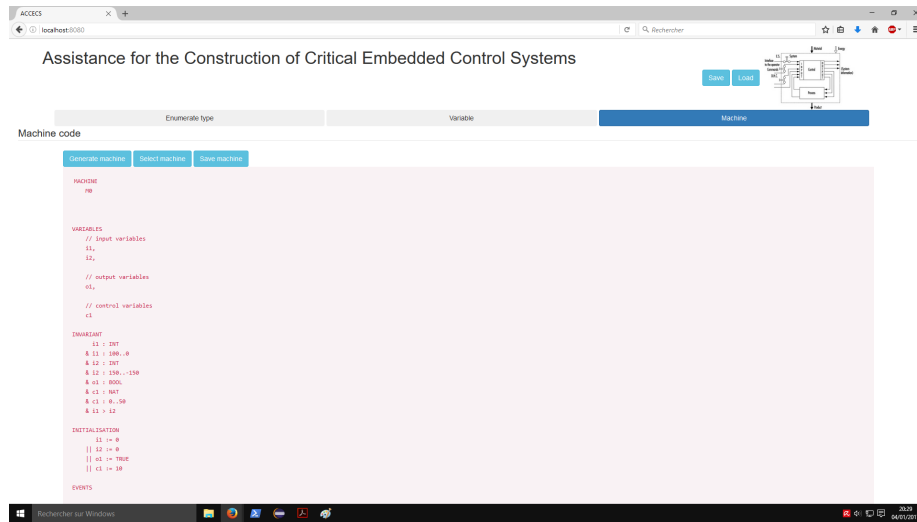


FIGURE 3 – Capture d'écran de la page 3

3 Sujets avancés

Cette partie s'attarde sur les questions soulevées durant la conception de l'outil ACCECS. Cette section abordera de nombreuses notions relatives à la méthode B, il est donc préférable d'être à l'aise avec les notions des méthodes formelles avant de lire la suite.

3.1 Les outils développement

Le sujet du projet nous laissait choisir les différentes technologies à manipuler pour réaliser l'outil ACCECS. Cette partie s'attarde sur ces choix et les explique.

3.1.1 Une interface web

Après plusieurs discussions avec le corps enseignant, une des premières nécessités était de rendre l'outil accessible facilement, si possible depuis internet et utilisable par plusieurs usagers en même temps. En effet, l'outil serait amené à être utilisé par des équipes, et les résultats de l'outil doivent être accessibles facilement aux différents membres de l'équipe. Nous avons donc choisi d'utiliser une interface web.

Nous avons choisi d'utiliser bootstrap pour un meilleur rendu en CSS, et car c'est une bibliothèque populaire. Les scripts sont rédigés en simple javascript car c'est une partie assez minime du projet, la plupart des calculs étant fait à l'aide de Java. Enfin, nous avons utilisé Jetty comme serveur par défaut, car il est facile d'utilisation (sous forme de plugin maven).

3.1.2 Le langage utilisée : Java

L'essentiel du code est écrit en Java, car c'est le langage sur lequel les membres de l'équipe étaient le plus à l'aise. De plus, grâce aux servlets, il est

facile de lier l'interface web et notre code source. Enfin, les de classe de Java permettent de regrouper certaines notions du B. Maven est également utilisé pour gérer facilement les dépendances et faciliter la reprise du projet.

3.1.3 Le format des données : JSON

Nous avons choisi de représenter les données échangées entre l'interface et le code source sous le format JSON, car c'est un format facile à lire et à manipuler, avec des analyseurs syntaxiques disponibles facilement, notamment sur le web. C'est sous ce format que sont enregistrées les données rentrées par l'utilisateur. On peut ainsi enregistrer ces données, créer des jeux de données, les échanger, et les rouvrir sous une autre machine.

3.1.4 Le moteur de template Jtwig

Les résultats fournis par notre outil se doivent de respecter un squelette correspondant à une machine B. Pour modifier facilement le squelette selon nos besoins, nous avons choisi d'utiliser un moteur de template. Jtwig est l'un d'eux, qui s'avère simple d'utilisation, très complet, compatible avec Java et assez populaire pour espérer qu'il soit maintenu dans les années à venir.

3.2 La limite entre notre outil et la méthode B

Notre outil a pour but d'aider à démarrer la conception d'une machine B. L'une des principales difficultés du projet a été de déterminer la fin.

3.2.1 Différencier un contrôleur et la future machine B

L'un des premiers besoins exprimé par le corps enseignant était que notre outil ne soit pas automatiquement associé à la méthode B. Il doit avant tout servir à spécifier un contrôleur qui va agir sur son environnement. Ce contrôleur possède des actuateurs, des senseurs, etc. . . qui vont servir à définir les premières variables de base. Notre interface doit donc permettre de rentrer ces premières variables ainsi que les premières propriétés déduites uniquement des propriétés du contrôleur, le tout de la manière la plus intuitive celui qui manipule le contrôleur, et non pour celui qui sera amené à rédiger le code en B.

3.2.2 Les évènements

Les évènements sont des opérations qui se déclenchent lorsqu'un senseur détecte une valeur critique. On peut déduire des évènements des variables de base indiqués par l'utilisateur, afin de les générer leurs squelettes dans la future machine B. Les variables d'entrées (input) génère ainsi des évènements sensoriels (sense) et les variables de sortie (output) génèrent des évènements de réactions (reaction). Il existe deux dernières catégories d'évènements, ceux de surveillance (monitor) et ceux de contrôles (control). Les évènements de contrôles lisent en entrée les variables de contrôles et agissent sur les variables de sortie, tandis que les évènements de surveillance lisent les variables d'entrées et agissent sur les variables de contrôles. Il n'est pas possible de générer un squelette pour ces deux types d'évènements car les variables de sorties et d'entrées sont impossibles à prévoir.

3.2.3 Les limites liées à la complexité du B

La méthode B elle-même s'est avéré être un frein, le but d'ACCECS étant de faciliter le début de la spécification d'une machine B, mais pas d'essayer de remplacer la méthode elle-même. Nous avons donc choisi de ne pas rendre lisible par notre outil du code en B. Cela demanderait de recoder toutes les notions de B en java pour un résultat sûrement de qualité moindre, et un suivi serait nécessaire en fonction des mises à jour du B. Cela représente un travail imposant pour un résultat moindre, car il est vrai que notre outil facilite la maintenance du code B, mais il se veut rester basique.