

Travaux Pratiques

Ingénierie Dirigée par les Modèles

Tâche 4:

Génération de Code

Frédéric Fondement

frederic.fondement@uha.fr

1. Description du problème

Vous avez décrit lors de la série de travaux pratique d'Ingénierie des Métamodèles un langage simple à base de composants pour les systèmes embarqués. Ce travail vous a permis de modéliser les concepts de ce langage, concepts à travers lesquels il est aisé de modéliser de tels systèmes. Ces modèles peuvent être édités dans un environnement graphique et vérifiés par rapport à un certain nombre de règles de cohérence. Le but de cette seconde série de travaux pratiques est l'utilisation effective de ces modèles en les "projetant" sur des plate-formes réelles. Vous remarquerez au passage que la spécification abstraite peut être indifféremment projetée sur des plate-formes de types différents. Ceci est le point fort de l'ingénierie dirigée par les modèles, puisqu'il est toujours possible de changer la plate-forme sans changer le comportement du système, et ceci sans perdre le travail des développements en amont.

La mise en place de systèmes décrits de manière abstraite sur des plate-formes matérielles ou logicielles peut être réalisée de plusieurs manières. L'une d'entre elle est la génération de spécifications compréhensibles par la plate-forme cible. Cette spécification dite concrète prend le plus souvent la forme de code source au format textuel. Nous verrons dans cette tâche comment générer du code à partir d'un modèle.

La génération de code est donnée par un ensemble de règles de génération en fonction de la syntaxe abstraite qui est décrite sous la forme d'un métamodèle. La génération de code est encore un exemple de domaine, et la les règles de génération sont encore un exemple de spécification. Vous aurez sans doute deviné qu'une fois de plus un langage et des outils ont été créés dans le but de pouvoir donner cette spécification et de la réaliser.

2. Travail demandé

Vous utiliserez l'outil XPand d'openArchitectureWare pour générer à partir de vos modèles IACA le code C pour la plate-forme PIC 18F458. La Figure 2 vous montre un exemple du code devant être généré à partir du modèle exprimé par la Figure 1. Les parties en noir sont générées indépendamment du modèle. Les parties en vert sont générées en fonction des instances de composants alors que les parties en rouge dépendent des associations entre instances. Les parties en italique dépendent des propriétés de ces éléments.

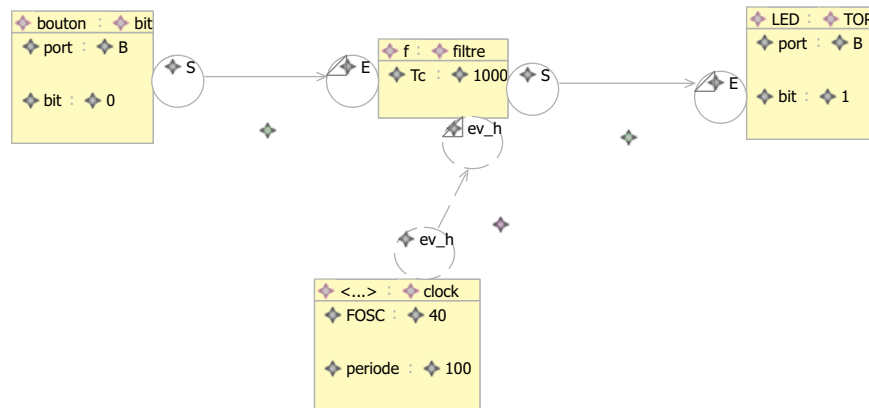


FIGURE 1. Représentation d'un modèle d'utilisation de composants IACA

```
#include "..\IACA\18F458.h"
#include <delay.h>
#include <fuses.h>
#include <classe.h>

//ComponentUsage declarations
struct c_bit bouton;
struct c_filtre f;
struct c_TOR LED;

void main () {

//ComponentUsage initializations
c_TOR_init(&LED, 'B', 1);
c_filtre_init(&f, 1000);
c_bit_init(&bouton, 'B', 0);
clock_init(40, 1000);

while (1) {

//ComponentUsage concurrent run
c_TOR_run(&LED, f.S);
c_filtre_run(&f, bouton.S, clock.ev_h);
c_bit_run(&bouton);
clock_run();

}
}
```

FIGURE 2. Génération de code C sur plate-forme PIC 18F458 pour l'exemple décrit en Figure 1

Vous aurez à disposition en salle de TP une maquette à base de PIC 18F458 pour tester votre génération de code sur une cible matérielle dont les périphériques sont aux adresses données par le Tableau 1. Bien sûr, vous vérifierez que vous êtes capables de générer le code correspondant à la Figure 1, ainsi que d'autres modèles de votre choix.

TABLEAU 1. Adresses des périphériques de la maquette de test

Description	Port	Bit
LED rouge du bas	E	2
LED rouge du haut	B	1
Bouton BP1	B	0
Bouton BP2	A	1

3. Ressources et astuces

Vous trouverez toutes les informations dont vous avez besoin dans le document [openArchitectureWare](#), au chapitre II.6, page 65. Un fichier `workflow` d'exemple vous est fourni sur le site du cours (veillez à bien le recopier à partir de la source, sans le recopier directement du navigateur).

Bien sûr, je vous encourage vivement à vous inspirer de l'exemple des [diagrammes d'états](#).

La maquette est celle que vous avez manipulé en travaux pratiques d'informatique industrielle en 1^{ère} année. Vous n'êtes pas autorisés à déconnecter la sonde du port USB, ni à changer la maquette de place (risques de court-circuits). Vous pourrez par contre déconnecter la maquette de la sonde pour constater que votre programme est bien fonctionnel.

La programmation du PIC se fait par l'intermédiaire de l'environnement MPLab, de Microchip. Un projet est disponible sur `D:\Users\IACA\Projet`. Vous n'aurez qu'à modifier le fichier `main.c` pour le faire correspondre au résultat de votre génération. Une fois l'environnement lancé et le projet (`main.mcp`) chargé, il faudra vous assurer que MPLab a détecté la sonde. Pour connecter la sonde, vous invoquerez la commande du menu `Programmer/Connect`. Pour compiler, vous appellerez la commande `Compile` (🔧). Pour tester, vous effacerez en premier lieu la mémoire du PIC (`Erase target device` - 🗑️), puis vous téléchargerez votre propre programme (`Erase target device` - 📁) pour finalement l'exécuter (`Release from reset` - ⏪). En déconnectant la maquette de la sonde, vous constaterez que votre programme est gardé en mémoire et sera à nouveau exécuté à chaque mise sous tension.

4. Délivrables

- Votre métamodèle (`.ecore`) et le modèle correspondant à la Figure 1 au format XMI y conformant, accompagné du modèle standard IACA.
- Les spécifications XPand (et XTend si vous en avez), et le fichier `workflow` d'exécution (`.oaw`).

- Des modèles de test au format XMI.

Le tout sera empaqueté dans un projet Eclipse au format archive ZIP et accompagné d'un fichier explicatif (par exemple lisez-moi.pdf).