Module « problématique des systèmes embarqués »

Documents sur : http://jeanlouis.boizard.free.fr/m1_eset/som_m1_eset.htm

Avant-propos

Le module vise à présenter quelques aspects des problématiques qui se posent lors de la conception de systèmes embarqués complexes et a pour but de montrer comment les disciplines enseignées par ailleurs peuvent interagir et faciliter la conception et la fabrication sûre de tels systèmes.

Mais tout d'abord qu'est-ce qu'un système embarqué ?

Un **système embarqué** peut être défini comme un système électronique et informatique autonome, qui est dédié à une (des) tâche(s) bien précise(s). Ses ressources disponibles sont généralement limitées. Cette limitation est généralement d'ordre spatial (taille limitée), énergétique (consommation restreinte) et à temps contraint. A ces systèmes sont souvent associées des notions de fiabilité et sureté de fonctionnement (que se passe-t-il lors d'une dégradation du fonctionnement ?).

(http://www.techno-science.net/?onglet=glossaire&definition=799)

Quelques exemples de systèmes embarqués :
Grand public : micro-ondes, machine à laver, téléphone portable, appareil photo, …
Automobile : calculateur ABS, calculateur d'injection, GPS, …
Avionique : calculateur commandes de vol, …

Dans le cadre de ce module, seront abordés :

- Quelques aspects de l'ingénierie système (Ingénierie dirigée par les modèles, flot de conception, modélisation UML/SysMl, ..)
- L'exploration architecturale
- Des notions de sureté de fonctionnement
- Les moniteurs multi-tâches
- La gestion de l'énergie
- La compatibilité électromagnétique (CEM)

Bien entendu beaucoup d'autres disciplines peuvent être amenées à interagir dans la conception de ces systèmes (thermique, résistance des matériaux, …) mais ne seront pas abordées dans le cadre de ce module.

Le flot de conception (de l'idée à la réalisation concrète) :
Il s'appuie sur les recommandations de la norme EIA 632, laquelle définit un ensemble d'activités permettant d'aboutir à la réalisation d'un système. Cette norme regroupe 5 thèmes, déclinés en 13 processus, eux-mêmes déclinés en 33 exigences. Nous nous limiterons au thème qui traite de la conception système, lequel regroupe deux processus : la définition des exigences et la définition de la solution. Chacun de ces processus se découpe également en trois exigences.

Historique des flots de conception :

l'approche Top Down et ses limitations

Issue du Génie logiciel, elle est représentée par un cycle en V partant, dans sa partie descendante, de l'analyse du besoin jusqu'au codage et se terminant, dans sa partie montante, par la phase de recette. Cette méthode a vite trouvé ses limites dans la conception des systèmes embarqués complexes matériels car un certain nombre de contraintes ne peuvent être remplies qu'a posteriori. C'est le cas par exemple lorsqu'il y a des contraintes sur le système fini tels que le poids, l'encombrement, la consommation, la compatibilité électromagnétique, le temps de réponse, …

l'approche Bottom Up et ses limitations

Méthode prisée par les concepteurs de systèmes matériels, elle s'appuie sur la mise en œuvre de composants existants (sur étagère), lesquels permettent par leur association, de construire un système. Cependant l'énorme quantité de composants disponibles (particulièrement en électronique) rend fastidieux les choix à opérer.

L'approche itérative "Meet in the Middle"

Elle s'appuie sur les deux démarches précédentes en exploitant les avantages respectifs de chacune : une recherche de solution technologique par analyse de l'existant (Bottom Up) guidée par les contraintes mises en évidence par la voie Top Down. Cette méthode évite une exploration inutile de solutions technologiques qui seraient de toute façon ultérieurement rejetées.

L'approche « Meet in the Middle » fait largement appel à l'ingénierie dirigée par les modèles. En effet de plus en plus et aux divers stades de la conception, il est fait appel à la vérification et validation par la simulation. Celle-ci s'appuie au départ, sur des modèles comportementaux simplifiés puis de plus en plus précis au fur et à mesure de l'avancement de la conception.

Les étapes dans le flot de conception et les outils associés

L'objectif n'est pas de dresser ici une liste exhaustive des outils disponibles mais d'en citer quelques uns. Nous les aborderons dans le même ordre que celui du processus de conception.

La capture des exigences : elle a pour but d'identifier aussi précisément que possible, le besoin, les services qui sont attendus du système, le contexte dans lequel il est utilisé, le degré de qualification des utilisateurs, etc …

Des outils tels que Rational DOORS, Rational Rhapsody IBM, permettent la capture et la gestion des exigences, la génération de matrices de traçabilité, le travail collaboratif, …

L'analyse fonctionnelle : MODELIO est un outil de modélisation UML. Il permet de générer des diagrammes de contexte, de cas d'utilisation, de séquences, … lesquels peuvent être des points d'entrée à la simulation (et ainsi simuler, d'un point de vue logique, le comportement attendu du système).

L'exploration architecturale : il s'agit de rechercher des solutions technologiques répondant au problème posé et d'en choisir une en s'appuyant sur une métrique prédéfinie (coût, consommation, ..). L'exploration architecturale peut faire appel à plusieurs niveaux de raffinement et est basée sur l'Ingénierie Dirigée par les Modèles (IDE ou MDE en anglo-saxon) :

- Décomposition en fonctions logiques nominales à temps non contraint : les fonctions sont modélisées par des modèles comportementaux simples dans lesquels les temps d'exécution, les aléas ou incertitudes (de mesure, de calcul, ..) sont exclus. Il s'agit ici de vérifier que la fonctionnalité au sens flot de données (ou Cas d'utilisation) est assurée.

- Dans un deuxième temps on affine, pour chaque fonction, les degrés de liberté tolérés : ceux-ci résultent de la simulation globale du système et des performances globales observées.

- Enfin on introduit dans les contraintes, pour chaque fonction, les tolérances acceptables (fourchettes de temps d'exécution par exemple). (outils : Matlab Simulink, VHDL-AMS, VHDL, SystemC, réseaux de Petri, …)

- Partitionnement/choix technologiques : à ce niveau on définit une première architecture avec les fonctions qui seront portées par du matériel, du logiciel ou une contribution des deux.

- Pour chaque technologie retenue, on extrait les paramètres obtenus en termes de performances (temps d'exécution, consommation, précision des mesures, encombrement, …) et on les reporte dans la simulation comportementale pour une vérification à haut niveau. Ce processus doit être réitéré jusqu'à la convergence vers une solution acceptable.

Exemple d'exploration architecturale: le pilote de barre franche

(On ne traitera pas ici la partie amont : spécifications, capture des exigences)

Dans un premier temps on considère l'objet à concevoir comme une boîte noire et on s'attache à définir de manière exhaustive ses finalités, son environnement au sens le plus large et les contraintes normatives qui lui sont applicables :

- Services attendus

- Enumération de tous les signaux d'entrées/sorties et leurs caractéristiques physiques

- Contraintes d'environnement : gamme de température de fonctionnement, étanchéité, …

- Certification CE par exemple, …

A ce stade, il peut déjà être fait appel à la simulation : on s'attachera par exemple à simuler, entre autres, les cas d'utilisation (USE CASE). Cette phase présente une importance capitale car elle permet aux différentes parties prenantes de s'assurer que les exigences, notamment en termes de fonctionnalités, ont bien été comprises. Les modèles utilisés sont dits « logiques », parfois « comportementaux » car ils n'ont aucun lien avec une quelconque solution technologique et ont l'avantage de réduire considérablement les temps de simulation.

Dans un deuxième temps, on s'intéresse à un contenu possible de la boîte : on identifie un certains nombres de fonctions considérées comme incontournables dans la réalisation de l'objet sans préjuger pour autant des choix technologiques qui seront retenus. A ce stade la simulation logico-temporelle peut jouer un élément clé : elle permet de vérifier la logique d'enchaînement des tâches qui contrôle le flot de données et de fournir des intervalles de temps d'exécution acceptables pour chacune d'entre elles. Plus largement, on peut associer à chaque fonction des contraintes non fonctionnelles (température, consommation, …). Les modèles utilisés ici sont également « comportementaux ».

Enfin, pour chaque fonction identifiée, on recherche des solutions technologiques qui peuvent répondre au problème posé. De là découlent plusieurs possibilités :

- Une solution technologique « sur étagère » couvre complètement le besoin exprimé par une fonction, auquel cas elle peut être retenue (mais pas nécessairement).

- Une solution technologique « sur étagère » couvre partiellement le besoin : il peut être intéressant de re-décomposer la fonction en sous fonctions en prenant en compte la solution existante.

- Aucune solution « sur étagère » ne répond au problème : on développe alors des solutions spécifiques. Dans ce contexte on peut également être amené à affiner les fonctions identifiées (re-décomposition à un niveau inférieur).

Les (principaux) services attendus du pilote de barre franche :

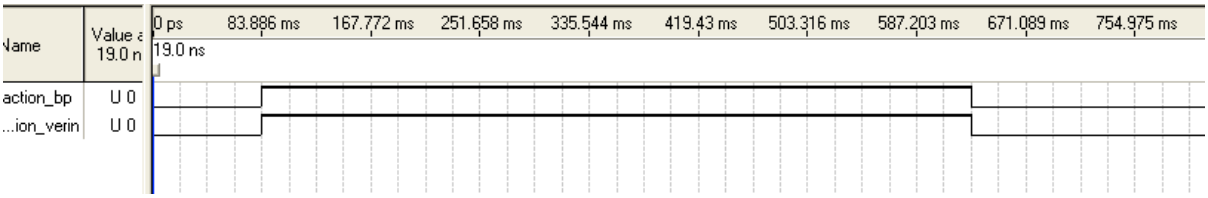
- manœuvrer la barre manuellement (babord ou tribord) et de manière sécurisée
- pilotage automatique par rapport à un cap ou à la direction du vent
- modifier la consigne de cap en + ou – par pas de 1° ou 10°
- s'interfacer avec des instruments complémentaires compatibles NMEA 0183
- supporter les contraintes d'environnement (température, étanchéité, …)
- se conformer aux normes qui lui sont applicables (marquage CE, …)

Le réseau de Petri présenté sur le diaporama illustre une partie des UC (Cas d'Utilisation), attendus par le pilote. La partie en bleu résume le fonctionnement en mode manuel ; en orange, le passage au mode pilotage automatique ; vert et violet correspondent à des changements de consigne de cap en mode pilotage automatique/cap. On peut parler ici de modélisation logique voire logico-temporelle car aucun modèle comportemental n'est associé à cette description de fonctionnement.

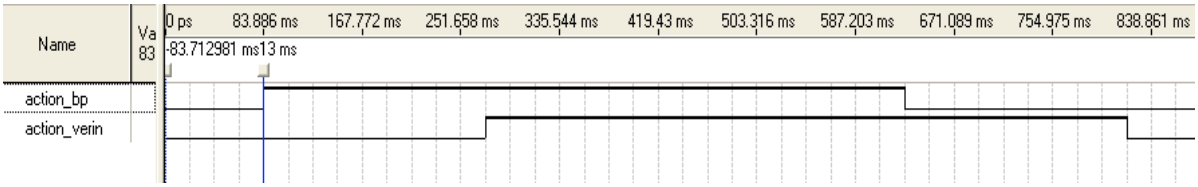
Autre Exemple : fonction F7 :« Gestion commandes et Indications barreur »

Mode manuel : Un appui continu sur le Bouton Poussoir « Babord » ou « Tribord » entraine le déplacement de la barre à gauche, respectivement à droite.

La simulation ci-dessous représente le cas de fonctionnement logique à temps non contraint :



La simulation ci-dessous représente le cas de fonctionnement logique à temps contraint (tolérance de 0,2s entre commande et action effective pour un confort d'utilisation) :



=> La contrainte de temps sera un des facteurs déterminant pour le choix de l'architecture. Ici, il peut être envisagé une unité de traitement partagée entre plusieurs tâches.

Exemple de la fonction F5 « pilotage sécurisé » :

Cette fonction peut être réalisée par un dispositif micro programmé (matériel + logiciel). Les circuits éligibles sont essentiellement : Les ASIC, les microcontrôleurs, les FPGA. Chaque technologie ayant ses avantages et inconvénients, il conviendra d'évaluer chaque solution en regard du contexte (très gros volumes de production favorisent la solution ASIC, la pérennité de l'investissement favorise le FPGA, le moindre coût immédiat favorise le µcontrôleur).

Rétro propagation des paramètres dans la simulation logico-temporelle :

Lorsqu'un certain nombre de choix technologiques sont opérés, on peut en extraire des paramètres pertinents pour la simulation logico temporelle. C'est le cas par exemple pour le temps d'exécution d'une fonction, son encombrement matériel, ses incertitudes de mesure associées, …

Exemple : mesure du cap suivi par le voilier

Cette fonction est assurée par un compas magnétique dont les données constructeur sont: précision de mesure = 1 degré, consommation = 5 mA sous 5V, étanchéité = IP65, temps de réponse = 100ms, …

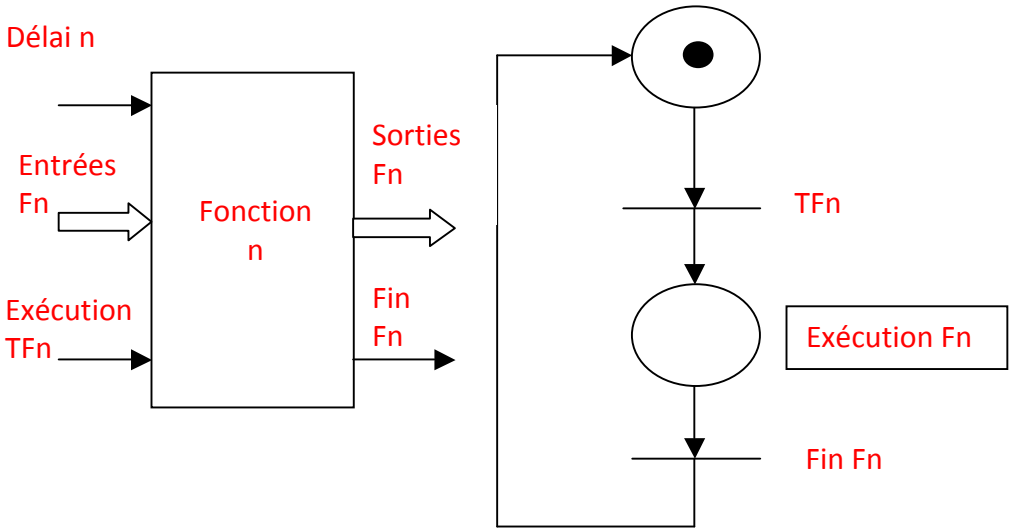
Ces paramètres, extraits pour chaque fonction, et réintroduits dans les différents modèles permettent de vérifier si lors de la simulation du système complet, les exigences globales sont respectées.

Si tel est le cas, on peut affiner la simulation globale en intégrant des modèles physiques représentatifs des solutions technologiques retenues.

La simulation logico-temporelle à haut niveau :

Chaque fonction n peut être représentée :

- par un modèle comportemental plus ou moins élaboré
- une demande d'exécution de la fonction
- un délai associé qui représente le temps mis par la fonction pour s'exécuter ou rendre le service attendu
- des entrées si la fonction en a la nécessité
- des sorties élaborées, si nécessaire, par la fonction
- un top de fin d'exécution
- un réseau de Petri associé représentant au minimum la fonction en veille ou en activité



Remarque : les fonctions qui sont activées en permanence et qui rendent le service instantanément (cas d'un amplificateur de signal par exemple) se voient attribuées un délai nul et ne passent pas par le mode « veille ».