Intitulé du cours : **Conception des systèmes sûrs**. Intervenant : Jean-Louis DUFOUR

Objet : Répondre à la question : « comment faire des systèmes fiables à partir de composants non-fiables ». On donne les méthodes de conception permettant d'atteindre au niveau système un taux d'évènements redoutés de 10^{-9}/h (par exemple) alors que les composants sont largement moins bons voire comportent du logiciel.

Attention, **ce n'est pas un cours de SdF**, c'est un cours de **conception**. Je n'y parle pas d'analyse fonctionnelle ou dysfonctionnelle, d'AMDEC ou d'arbre de faute. Par contre, je commence par des bases de fiabilité, donc le cours est autonome et peut (presque) indifféremment être donné avant ou après un cours de SdF.

Auditoire : ingénieur généraliste ayant besoin de discuter avec des spécialistes pour pouvoir arbitrer entre différentes possibilités, ou juger de la pertinence d'une réponse technique à un cahier des charges : par exemple un donneur d'ordre en face de fournisseurs.

Plan : un système embarqué peut en général être vu comme un réseau de calculateurs, chaque calculateur étant formé d'un couple matériel/logiciel. Le cours se compose donc de 3 parties :

1) Sécurisation des communications
Concepts : canal bruité, taux d'erreur bit (« peb »), codage, probabilité de non-détection (« pnd »), parité, code de Hamming, code BCH, second théorème de Shannon.
Objectifs :
- savoir dimensionner un code détecteur/correcteur en fonction du couple (peb,pnd),
- avoir compris la mécanique du codage et du décodage de Hamming, et pourquoi elle permet la correction d'une erreur,
- avoir compris que pnd = 0 est impossible, et dans quelles conditions pnd < epsilon est possible,
- avoir compris qu'un même code peut être utilisé en détecteur, correcteur ou un mélange des deux, et que plus il sera correcteur, plus Pnd se dégradera.

2) Sécurisation du matériel
Concepts : taux de défaillance, lois exponentielle et normale, redondance chaude et froide, mode commun, courbe de fiabilité en « S », latence d'une panne, autotest, processeur codé, redondance asymétrique « VDA ».
Objectifs :
- savoir choisir une architecture sûre (« safety concept ») et chiffrer le taux de défaillance dangereuse associé,
- avoir compris ce qu'est d'une part la loi exponentielle, d'autre part un taux de défaillance, et le lien entre les deux,
- avoir compris que le temps avant défaillance d'une redondance ne suit pas une loi exponentielle, que donc la notion de taux de défaillance est à utiliser avec précaution et qu'il est en particulier interdit a priori de multiplier les taux de défaillance,
- avoir compris d'où vient la courbe de fiabilité d'une redondance en « S », l'analogie avec les codes correcteurs et pourquoi en général une redondance pure n'est pas adaptée,
- connaître l'état de l'Art des architectures sûres ferroviaires, automobiles et aéronautiques.

3) Sécurisation du logiciel
Concepts : bug, DAL-SIL, allocation de DAL-SIL, exigence dérivée, traçabilité, Validation et Vérification (au sens ARP/DO), modèle formel, raffinement des données et du contrôle, sémantique, interprétation abstraite, model-checking, logique de Hoare, méthode « B ».

Objectifs :

- avoir compris la différence entre bug et défaillance, DAL-SIL et taux de défaillance, et pourquoi d'une part on ne peut pas associer un taux de défaillance à un logiciel, d'autre part on est souvent amené à le faire pour un couple (logiciel, environnement),
- avoir compris les grandes lignes de l'état de l'Art, en particulier la philosophie « obligation de moyen », et l'apport des méthodes formelles qui introduisent une « obligation de résultat »,
- savoir lire un sous-ensemble du langage de spécification de la méthode B (basé sur un exemple de spécification système d'une anticollision ferroviaire),
- avoir compris pourquoi la logique de Hoare « classique » ne passe pas à l'échelle (entre autres pourquoi elle est incapable de formaliser une spécification système), et comment la méthode B répond en partie au problème avec la notion de raffinement,
- avoir compris / connaître le domaine d'application (le type de bug traité) de chacune des 3 technologies formelles existantes (interprétation abstraite, model-checking et logique de Hoare).

La formation a été donnée depuis 2002 dans plusieurs écoles d'ingénieurs :

- en 1/2j (uniquement un sous-ensemble des parties 1 et 3) à l'ISEP,
- en 1j, au mastère « Management de projets et ingénierie des systèmes » de Supélec : http://www.supelec.fr/fc/masteres/ingenierie_systemes/Bienvenue.html
- en 2j, au mastère « SYVAT » des A&M : http://graduateschool.paristech.fr/programme.php?id=958
- en 2j ½ (la dernière ½ journée est un TD sur 2 méthodes formelles), au mastère « COMASIC » : http://www.mpmas.fr

L'intervenant est ancien élève de l'Ecole Polytechnique (X86), a préparé une thèse à l'INRIA Rocquencourt sur le typage du lambda-calcul puis a travaillé

- 8 ans chez Matra Transport (maintenant Siemens ; essentiellement sur la Ligne 14 « Météor » du métro Parisien),
- 4 ans chez Ligeron (société de service spécialisée en SdF ; prestations entre autres pour Sagem Automobile et Alstom Transport),
- 4 ans chez PSA (Division « Organes » : moteur/boîte/liaison au sol)
- et depuis 2008 chez Sagem (groupe Safran ; Division Avionique), où il est expert en systèmes embarqués.

Ses 3 domaines de compétence sont :

- Systèmes temps réel.
  - Publication en 2010 à la conférence ERTS (Toulouse): "Deterministic scheduling reconciles cache with pre-emption for WCET estimation", associée à un brevet sur une mise en œuvre de la mémoire cache certifiable.
  - ERTS 2016 : "Unified MBD: how not to choose between Scade and Simulink",
  - ERTS 2018 : la suite sur les machines à état (où l'unification est partielle).
- Sureté de fonctionnement (« maitre-expert » chez PSA).
  - 1996 : "Available architectures for safe railway digital systems", 26th Fault-Tolerant Computing Symposium (Japon),
  - 1996 : "Safety computations in integrated circuits", 14th VLSI Test Symposium (U.S.A.),
  - 2005 : "Automotive safety concepts : 10^{-9}/h for less than 100€ a piece", 6ième conf. AAET (Allemagne).
- Développement formel de logiciel critique (réalisation d'une appli. en B pour Alstom).
  - ERTS 2012 : "The B method takes up floating-point numbers",
  - 2013 : chap. 10 du livre "Mise en oeuvre de la méthode B" édité par Jean-Louis Boulanger chez Hermès/Lavoisier (trad. en anglais en 2014 chez Wiley),
  - ERTS 2014 : "Compositional certification : the CERCLES2 project".