

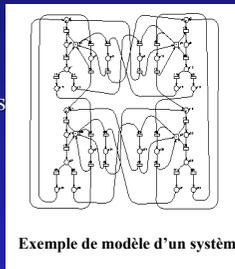
# Des logiciels fiables par la vérification

## FAQ (Question Fréquemment posées)

1. Où faut-il des logiciels fiables?
2. Qu'est-ce que la conception fiable de logiciels (et de matériel)?
3. Existe-t-il des outils de vérification?
4. Quels sont les problèmes rencontrés quand on effectue de la vérification?
5. Existe-t-il des « Success stories » en vérification?

### 1. Où faut-il des logiciels fiables?

- Contrôleurs de centrales électriques, d'usines
- Aide à la navigation d'avions, de trains, de voitures (ABS, airbag, ...), de navettes spatiales
- Contrôleurs d'équipements pour les diagnostics et soins médicaux
- Protocoles de réseaux, protocoles de sécurité informatique, etc.



Le contrôle d'équipements prend une part de plus en plus importante des logiciels conçus

### 2. Qu'est-ce que la conception fiable de logiciels (et de matériel)?

- Utilisation lors de la conception des **méthodes formelles** (c'est-à-dire basées sur les mathématiques)
- Les deux ingrédients principaux de ces méthodes
  - Un **modèle S** du système c'est-à-dire une description (abstraite) des comportements possibles du système
  - Une **spécification P** des propriétés que doit satisfaire le système

Le but est de prouver automatiquement (à l'aide d'un ordinateur) que le modèle **S** du système satisfait la spécification **P**

### Le Groupe de Vérification à l'ULB

#### Responsables:

- Thierry Massart
- Jean-François Raskin

#### URL:

<http://www.ulb.ac.be/di/ssd/grouperif.html>

#### Thèmes de Recherche principaux:

- Systèmes temps réels et hybrides
- Vérification de systèmes infinis
- Vérification de protocoles de sécurité
- Génération d'environnement de contrôle fiable d'équipements industriels

Verification  
Group

ULB

### 3. Existe-t-il des outils de vérification?

- **Oui!**
  - Mais... principalement académiques
  - Néanmoins des outils industriels commercialisés commencent à intégrer des possibilités de vérification
    - Exemple: Telelogic (environnement UML - SDL avec analyse, un peu de vérification, génération de code)
  - Certaines sociétés importantes ont leurs propres outils: exemple: Intel, IBM, Motorola, Siemens, etc.

### 4. Quels sont les problèmes rencontrés quand on effectue de la vérification?

- En pratique, la vérification « brutale » du logiciel est généralement trop difficile:
  - Est indécidable en général
  - Prend trop de temps et de place mémoire
- Des recherches sont en cours pour pallier ce problème.
  - Abstraction
  - Représentation compacte de grands ensembles d'états
  - Algorithmes efficaces, etc.

### 4. Existe-t-il des « Success stories » en vérification?

- Voir site [www.fmeurope.org](http://www.fmeurope.org):
  - Centaines d'applications vérifiées (et le plus souvent corrigées)
- **INTEL:**
  - En 1994 perd 475 M \$ suite au bug du processeur Pentium (FDIV)
  - Maintenant: la vérification est complètement intégrée à la conception de certaines parties critiques des processeurs INTEL

### Le Groupe de Vérification à l'ULB

#### Collaborations belges:

- Université de Namur (Prof. Pierre-Yves Schobbens)
- Université de Liège (Prof P. Wolper et B. Boigelot)
- Université de Mons Hainaut (Prof. Véronique Bruyère)
- Université Catholique de Louvain (B. Le Charlier)

#### Collaborations étrangères:

- University of California, Berkeley
- University of Genova, Italy
- Ecole Centrale de Nantes, France
- Ecole Normale Supérieure de Cachan, France
- University of Southampton, UK
- Verimag, Grenoble, France
- Stanford Research Institute, California, USA
- INRIA Rhône-Alpes Grenoble, France