UNIVERSITÉ DE NANTES 2025

systerel
Solutions temps réel sécurisées

Utilisations de la Méthode B à Systerel

Frédéric Badeau (Systerel)

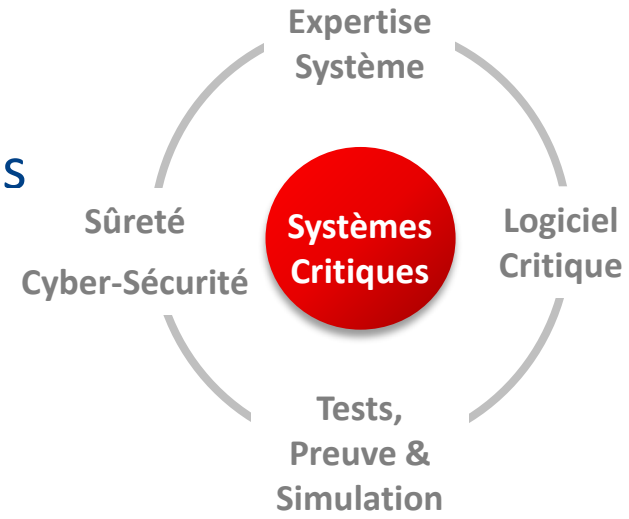
27/05/2025

Systerel

Bureau d'Etudes Spécialisé

Ingénieriste des systèmes critiques

Briques technologiques



systerel
Solutions temps réel sécurisées

Chiffres clés

11

M€ CA

+ 10% dédiés à la R&D

120+

INGENIEURS

Grandes écoles ou formations supérieures

10

ANS

Expérience moyenne

3

SITES

Aix-en-Provence
Paris
Toulouse

systerel
Solutions temps réel sécurisées

Quelles utilisations de la Méthode B ?

Introduction

B logiciel

B système

Données

Conclusion

systerel
Solutions temps réel sécurisées

Pourquoi utiliser la méthode B ?

Qu'apporte la méthode B ?

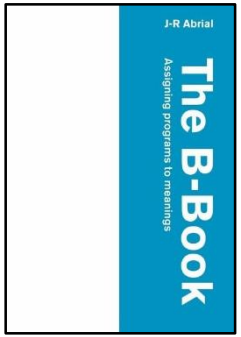
- Formalisation mathématique (non ambiguë)
- Preuve effectuée / vérifiée par la machine

Maitrise des risques

- Détecter les problèmes au plus tôt
- En spécification plutôt que pendant le codage ou les tests
- Pendant la preuve plutôt que les tests ou une fois déployé

Avantages de la démarche

- Démarche scientifique de l'ingénieur au lieu d'empirique
- En cherchant le meilleur modèle on maîtrise mieux le sujet étudié



Méthode B pour le logiciel

Méthode B

Inventée par Jean-Raymond Abrial à la fin des années 80

- nombreuses lignes de métro (L14, CBTC)
- contrôle de vitesse par balise (KVB SNCF), …

Objectif : Développer du logiciel sûr

- conforme à sa spécification
- sans erreur à l'exécution
- qui termine

Notation mathématique basée sur

- logique des prédicats du premier ordre avec égalité
- théorie des ensembles
- arithmétique

Démarche B

Modélisation abstraite au départ

Utilisation d'invariants, pre et post-conditions

Raffinement pour passer de l'abstrait au concret (code)

Décomposition en un arbre de modules (machine + implantation)

Preuve de cohérence interne et raffinement correct

Traduction des implantations dans un langage informatique (1pour1)

Pas de tests unitaires / intégration

Mais toujours des tests de validation

Logiciels sécuritaires pour des systèmes ferroviaires

Logiciel de contrôle/commande d'équipement sécuritaire

Logiciel sécuritaire d'un système ferroviaire (ex. bord/sol de CBTC)

Développement pour des industriels ferroviaires (tout, partie, preuve)

Modèle B pour le haut niveau

- Logiciel modulaire, procédural, modélisant 1 cycle du logiciel
- Structures abstraites : type, ensemble, relation, fonction, suite

Traits particuliers

- Approche Hautement Recommandée par la norme logiciel ferro
- Structure de code : tableaux, pas de pointeur/allocation dynamique
- Code B traduit en C/Ada et intégré à du code classique
- Entrée/Sortie de messages par lecture/écriture dans des tableaux

Safe and Secure OPC UA

Protocole OPC UA (norme ISO/IEC 62541) pour Industry 4.0

Protocole client / serveur sécurisé

Développement open-source

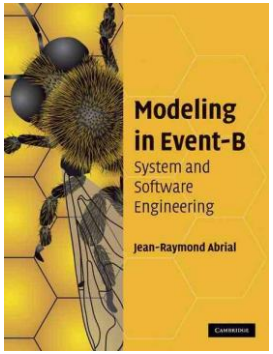
Modèle B pour le haut niveau

- Gestion des canaux sécurisés et des sessions
- Traitement des requêtes / réponses

Traits particuliers

- Modélisation de pointeurs / allocation dynamique
- Parcours d'un graphe compliqué (address space)

systerel
Solutions temps réel sécurisées



B événementiel pour la conception système

Conception formelle de systèmes

Système = contrôleur + son environnement

Objectif : Modéliser un système pour montrer qu'il est sûr

- Hypothèses sur l'environnement
- Propriétés de sécurité (invariant)
- Preuve formelle que l'invariant est préservé

Modélisation réactive sous la forme d'événements

- Événement = paramètres + garde + action

Raffinement :

- Ajouts de détails de conception
- Temps plus fin

Applications à Systerel

Applications

- DIR41 : Spécification des principes de signalisation RATP
- Sol d'un système CBTC
- Systèmes de signalisation NG (RATP, SNCF)
- Treuil d'hélicoptère

À l'exception du premier, utilisation très en amont, bien avant de développer le système

Apports

- Raisonner de manière très abstraite
- Découvrir les problèmes au plus tôt
- Expliciter des hypothèses suffisantes pour tenir la sécurité

Animation pour la validation du modèle

Validation et génération formelles de données

Validation formelle de données

Système ferroviaire = logiciel générique + données de paramétrage

Beaucoup de données (~100 000 valeurs)

Propriétés de sécurité (~1000 propriétés)

Ne peut être vérifié manuellement, besoin d'un outil (Excel ?)

Formalisation des propriétés en B

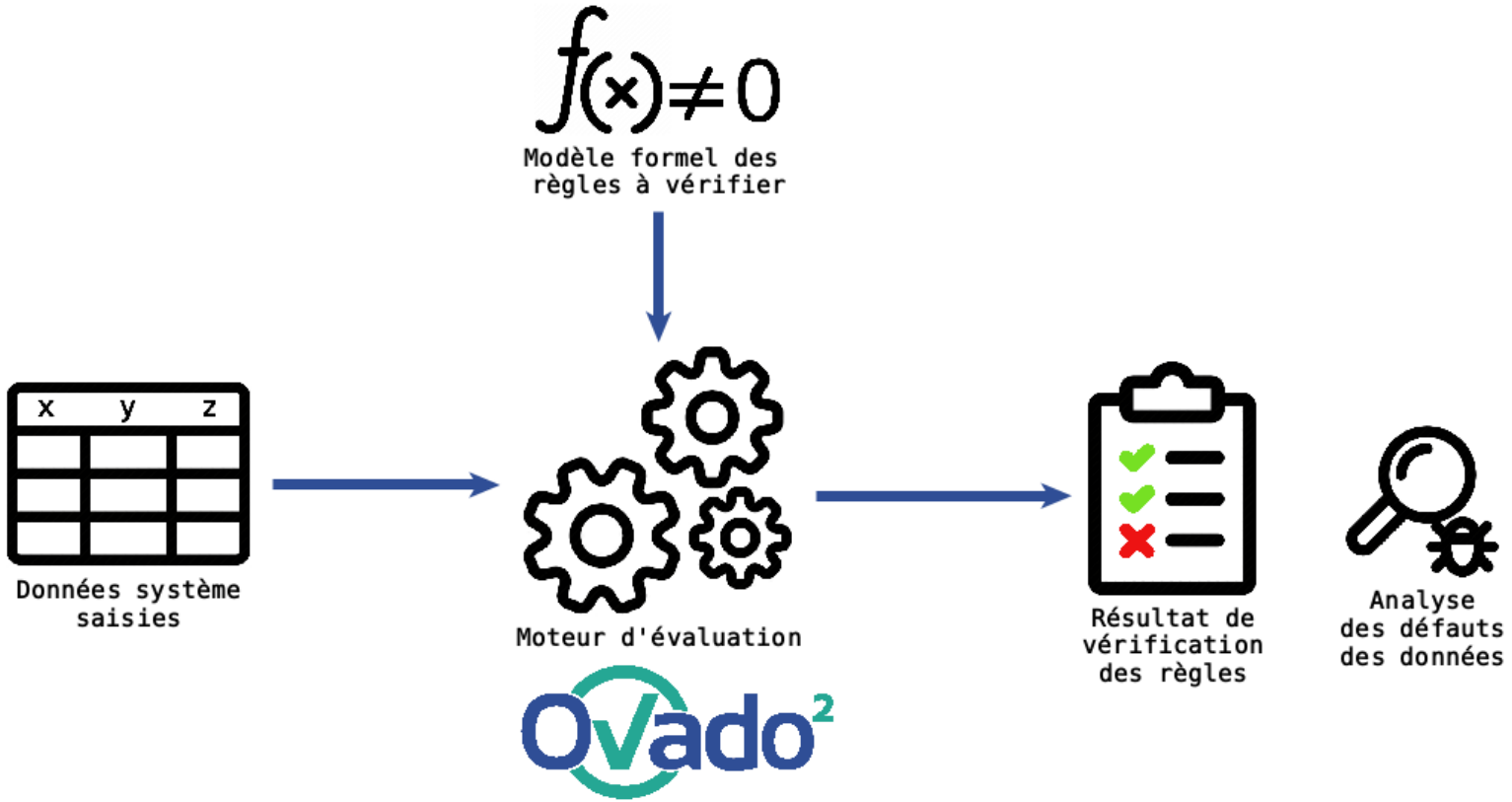
- permet d'introduire des abstractions (concepts intermédiaires)
- plus lisible qu'un programme
- plus facile à faire évoluer qu'un programme

Ovado$^{2®}$: Vérification automatique en sécurité

Contre-exemple => données ou modèle à corriger

Utilisé sur de nombreuses lignes de métro

L'outil RATP OVADO²®



systerel
Solutions temps réel sécurisées

Génération formelle de données

Provenance des données :

- Mesures brutes (géométrie), découpage en segments, zones
- Données dérivées par calcul
- Données mises en forme (e.g., triées)

Besoin d'un outil de génération de données

Problème similaire à la validation (programme difficile à valider et faire évoluer)

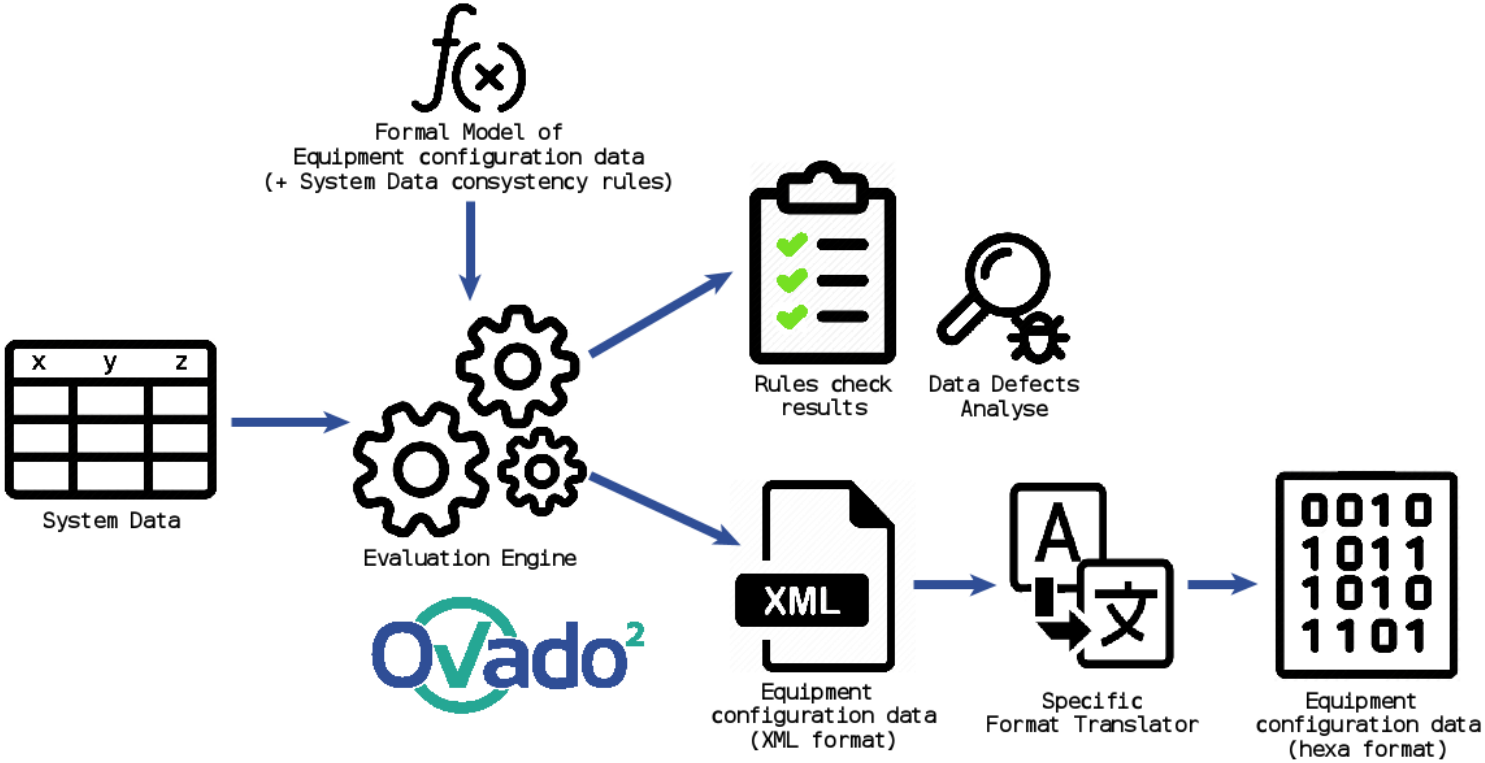
Formalisation des transformations à appliquer en B

Calcul automatique par Ovado$^{2®}$

Validation formelle du résultat comme avant

Utilisé sur les lignes les plus récentes (L6, L10, etc.)

Génération de données avec OVADO²®



systerel
Solutions temps réel sécurisées

Conclusion

Conclusion

Des cas très variés d'utilisation

- Système, logiciel, données

Outil puissant de vérification, force à simplifier

Un investissement rentable dans le temps

Approche pragmatique de maitrise des risques

Risques résiduels :

- Passer trop de temps sur des problèmes non critiques
- Plate-forme d'exécution
- Validité du modèle / des hypothèses

systerel
Solutions temps réel sécurisées

Merci