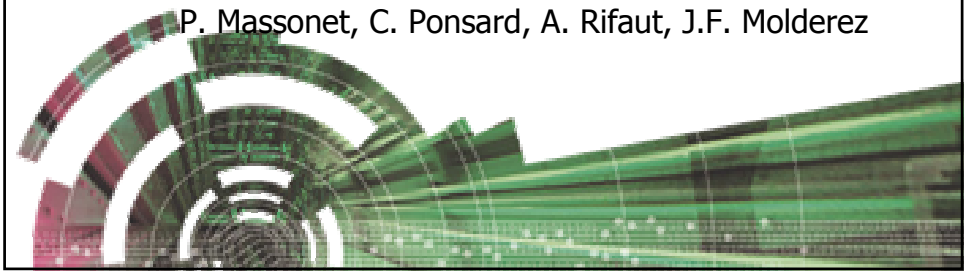
La Modélisation, Vérification et la Validation de Systèmes Critiques

Groupe de discussion CETIC
17 Juin 2004

P. Massonet, C. Ponsard, A. Rifaut, J.F. Molderez

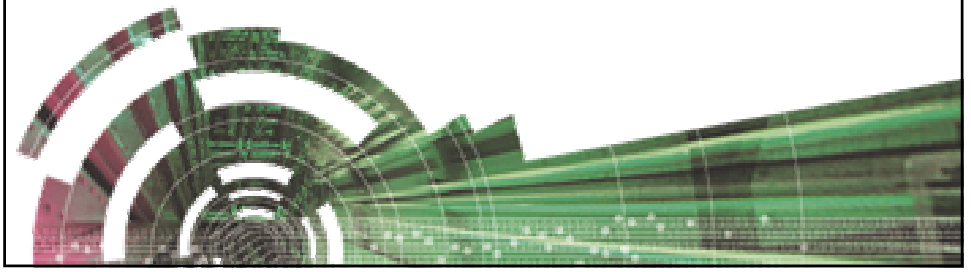


Plan

- Introduction
  - CETIC
  - L'ingénierie des exigences
  - Les systèmes critiques
- Quelques éléments d'analyse orientée « but »
  - Quelques définitions
  - 4 modèles
  - Formalisation et analyse d'obstacles
- Support logiciel: l'atelier FAUST
  - Pour le modèle: animateur, analyseur
  - En aval du modèle: monitoring, tests
- Etudes: quelques cas réels
- Conclusions et discussion

cetic
Your connection to ICT research

Introduction



cetic
Your connection to ICT research

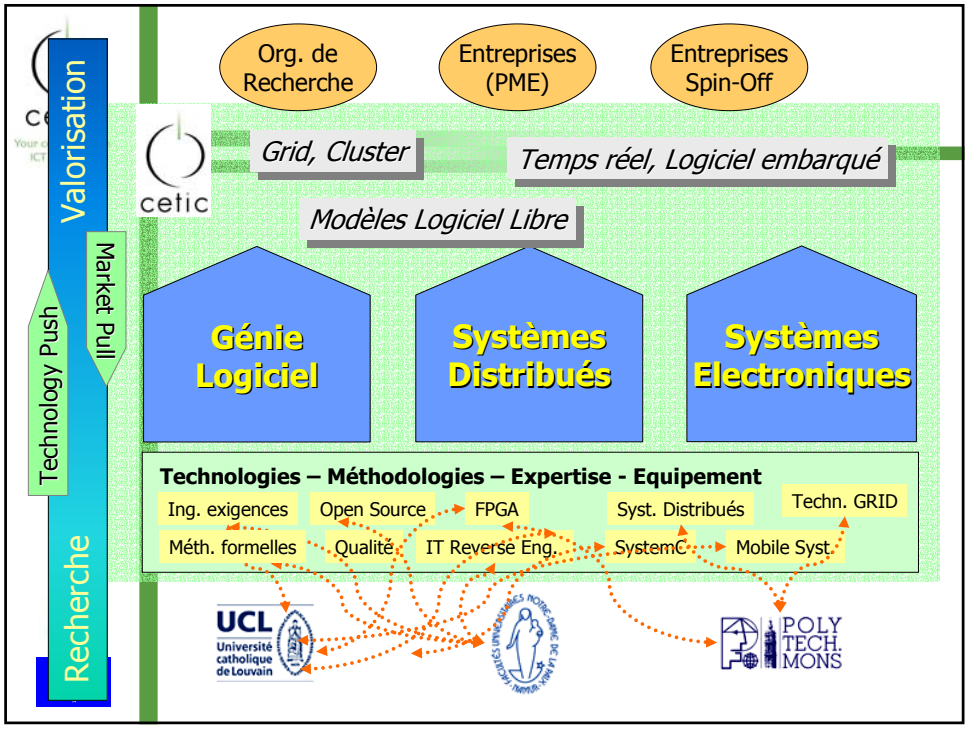
Positionnement du CETIC

Your connection to ICT research

UNIVERSITES

ENTREPRISES

UCL cetic CEDITI

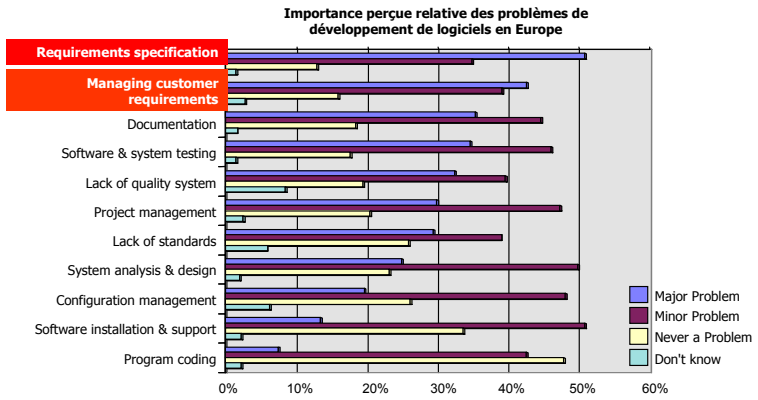


cetic
Your connection to ICT research

Le Problème des Exigences

Enjeux économiques - Etude ESPITI 1996

Enquête auprès de 3800 professionnels (Europe)



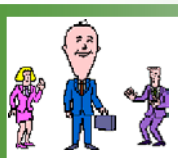
Les Systèmes Critiques

- **Systèmes critiques**: défaillance => conséquences graves (coûts, dégâts pour l'environnement, pertes humaines, …)
- **Haut niveau de qualité requis: « safety », « security », disponibilité, …**
- **Domaines**:
  - aérospatiale, transport, énergie, finance, santé, contrôle des processus industriels,…
  - mais aussi votre société (composantes business critical, mission critical)

Approche pour les systèmes critiques

- Approche rigoureuse dès le départ
  - Identifier le problème avant la solution
  - Modéliser le système composite
  - Dérivation d'artefacts (**cahier des charges**, architecture initiale, …)
- En particulier pour systèmes critiques
  - Identification des parties critiques d'un système
  - Modélisation dans un langage basé sur la logique mathématique
  - Points forts: permet l'analyse rigoureuse
    - validation
    - vérification
  - Points négatifs: difficile à écrire, à communiquer
  - Dérivation (semi) automatique d'artefacts (cahier de charges, jeux de tests, architecture initiale, …)

La place central de l'ingénierie des exigences dans le génie logiciel



Validation

Elicitation

Quoi, Qui et Pourquoi ?

Building the right system

Ingénierie des exigences
(Requirements engineering)

Vérification "early"

Implementation

Tests/(Vérification)

Comment ?

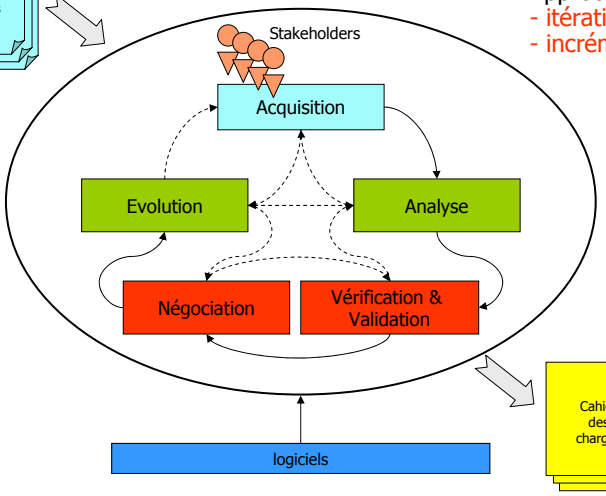
Building the system right

Développement de logiciels
(Software development)

L'IE: un processus en soi

Idées
Contraintes
Normes
Exigences
client
...

Approche
- itérative
- incrémentale



Cahier des charges

Deux problèmes essentiels

- « Are we building the right system ? »
  - Le système ne réponds pas convenablement au besoin des commanditaires
  - => validation
- « Are we building the system right ? »
  - Le système n'assure pas les propriétés désirées avec des conséquences (surtout si le système critique)
  - => vérification

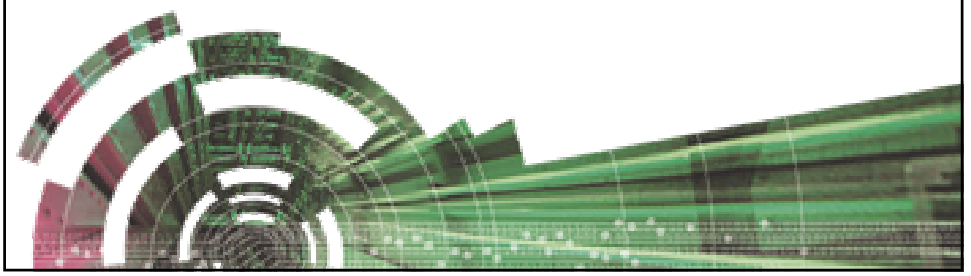
Démarche de validation

- Lecture du cahier de charges/spécification
  - Est-ce assez parlant ?
  - Est-ce dans le vocabulaire de l'utilisateur ?
- Présentation d'une maquette
  - Nécessite des ressources pour la développer
  - A jeter ou première itération de produit
- Génération d'animation
  - Parlant, dans le vocabulaire du domaine
  - **Nécessite un modèle !**

Démarche de vérification au niveau des exigences

- Différents niveaux
  - Langage naturelle:
    - définition du vocabulaire utilisé
    - traçabilité
  - Langage graphique:
    - contraintes sur les notations graphiques utilisées (ex. UML)
    - éventuellement avec langage d'expression plus poussé (ex. OCL)
  - Langage mathématique:
    - sémantique précise du langage (contre exemple: UML)
    - garantie sur le comportement du design
    - **Nécessite une modélisation plus poussée**
- A coût croissant mais à ROI proportionnel !
- Niveaux non mutuellement exclusifs:
  - parties non-critiques => modèle + léger
  - parties critiques => modèle + poussé

Quelques notions d'ingénierie des exigences orientées « but »



Exemple de support: mini-système « ferroviaire »

On considère un système ferroviaire (très simplifié):

- reliant différentes stations disposées selon un circuit fermé
- a seule voie (pas d'aiguillage) et a sens unique
- découpé en blocs consécutifs

Le système doit contrôler:

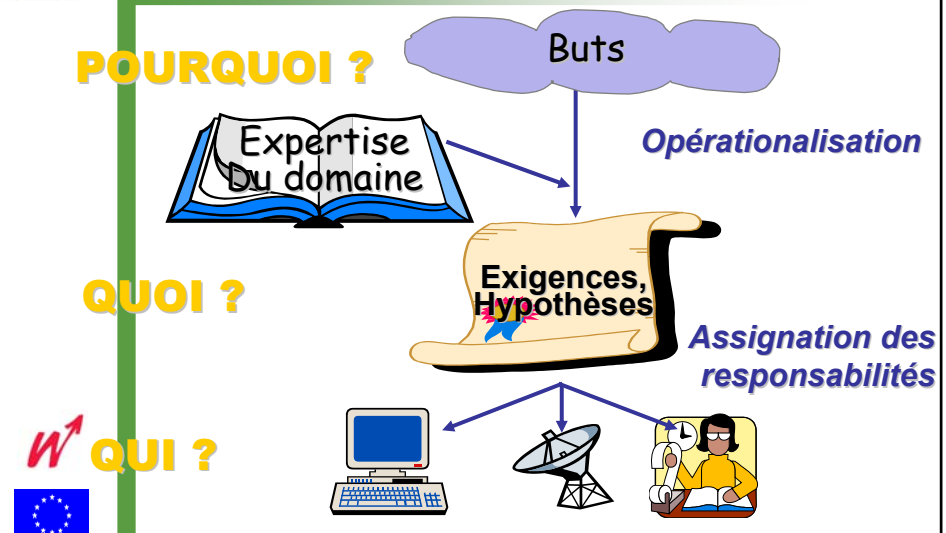
- le démarrage et l'arrêt de chaque train
- l'ouverture et la fermeture des portes de chaque wagon
- L'ouverture et la fermeture des passages a niveau

Le système doit garantir:

- la sécurité des usagers du train et des personnes présentes dans son environnement (voiture, piétons)
- La progression: les trains et véhicules ne peuvent rester indéfiniment bloqués dans leur parcours

L'environnement du système comporte un certain nombre de senseurs pouvant communiquer des informations nécessaires au système de contrôle. Chaque bloc est équipé de senseurs permettant de déterminer la présence ou non du train, l'ouverture ou non de ses portes.

Ingénierie des exigences (IE)



Pourquoi l'orientation but ?

- dirigent l'élaboration des exigences et permettent de les motiver (critère de pertinence)
  - approche « top-down » (HOW?): nouveau système
  - approche « bottom-up » (WHY?): évolution de l'existant
- fournissent de riches mécanismes de structuration facilitant la compréhension du CdC
- permettent l'examen d'alternatives dans les raffinements et d'assignations de responsabilités avec diverses frontières entre le système et son environnement

Buts : définition

« Objectifs à satisfaire par le système »

- Propriétés désirées par des parties prenantes (! conflits, processus de négociations,…)
- Propriétés que le système doit remplir (critères à préciser) pour être acceptable
- >< propriétés propres au domaine (établies et inviolables)

Système:

- logiciel + environnement
- système existant (as-is) vs système en conception (to-be)

Les buts: différents niveaux d'abstraction

- Buts de haut niveau
  - stratégiques, globaux, couvrant l'organisation
  - "assurer le transport rapide des passagers"
  - "assurer le transport sûr des passagers"
- Buts de bas niveau
  - opérationnels, locaux, spécifiques au design
  - "les portes du train doivent être fermée lorsque celui-ci est en mouvement"
  - "un conducteur du train ne peut pas franchir un signal rouge"

Buts fonctionnels (FR) et non-fonctionnels (NFR)

- Buts fonctionnels: **ce que le système doit faire**
  "Transport des passagers"
- Buts non-fonctionnels:
  qualités supplémentaires souhaitées
  - Sécurité : "transport sûr des passagers"
  - Performance : "transport rapide"
  - Coût : "transport à coût raisonnable"
  - Adaptabilité : "transport par toute condition météo"

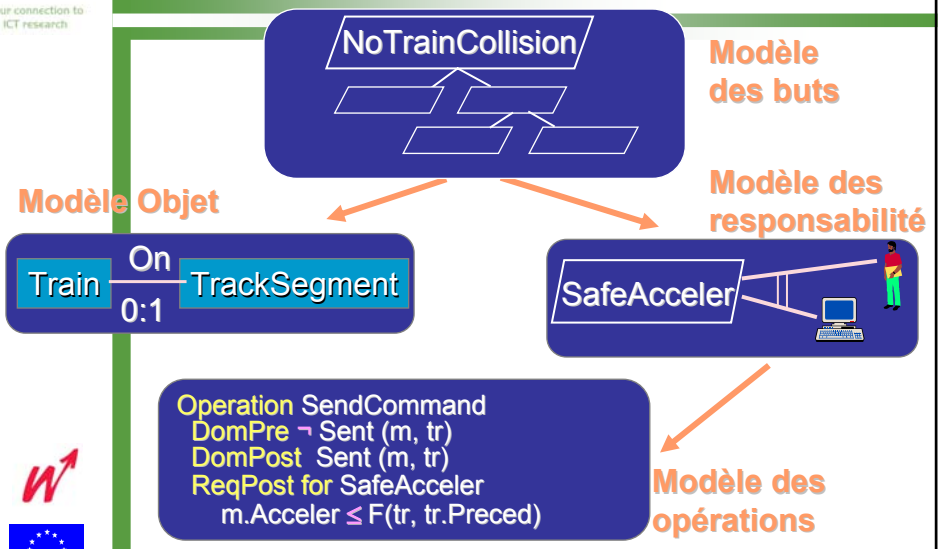
Réalisation coopérative des buts

- coopération d'agents: logiciels, matériels et/ou humains:
  ex. fermeture des portes: passager, conducteur, syst. de contrôle
- agent responsable d'un but:
  ex. démarrage du moteur: syst. de contrôle
- assignation d'un but
  - dans le système = exigence
    - ex. Un train en approche d'un passage a niveau déclenche la fermeture de ses barrières
  - dans l'environnement = hypothèse, norme, politique
    - ex. Arrêt du conducteur a un feu rouge

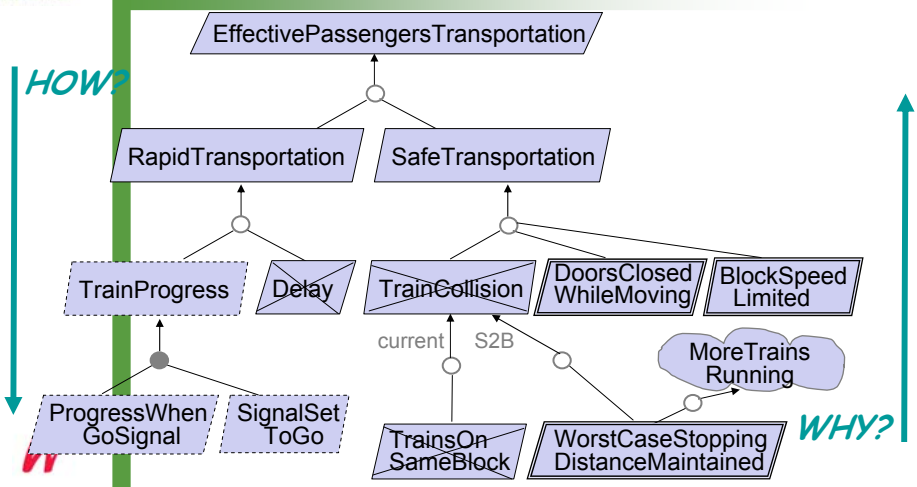
Différents modes de représentation

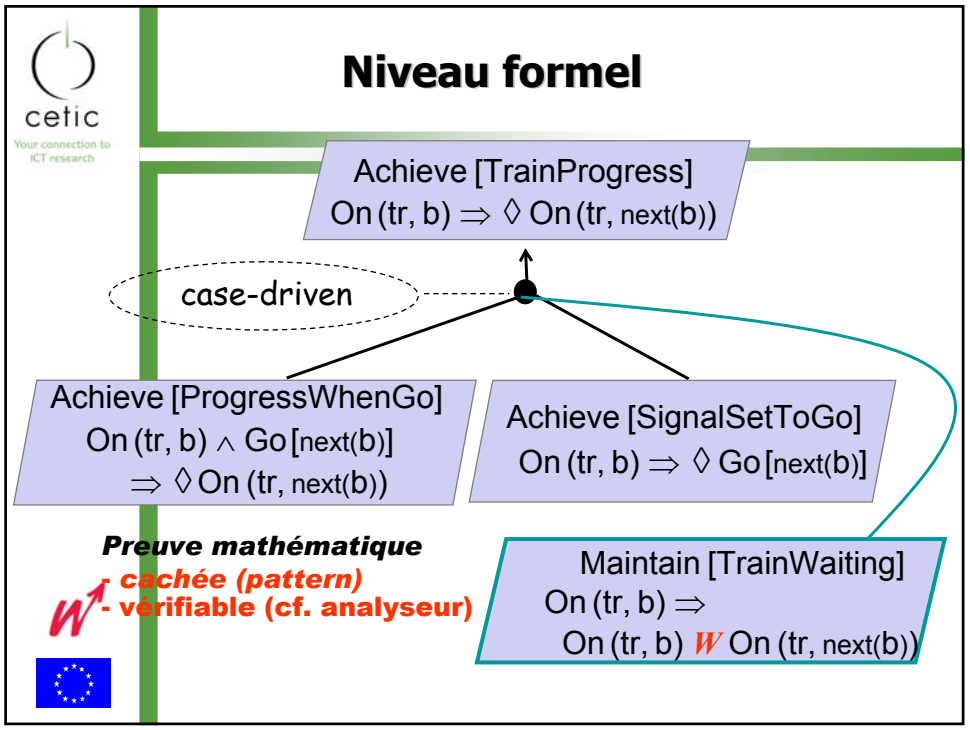
- Informelle: texte (structuré)
  - Canevas de CdC (eg. IEEE-Std-830)
- Semi-formelle: construction d'un modèle
  - i* [Yu, Mylopoulos et al]
  - **KAOS [van Lamsweerde et al]**
  - CSD [Feather]
  - Inquiry Cycle [Potts, Anton]
  - EKD [Bubenko, Rolland, Loucopoulos]
  - Win-Win [Boehm]
  - NFR [Chung, Mylopoulos]
- Formelle: utilisation de notations mathématiques.
  Apport supplémentaire: raisonnement
  - **supporté par KAOS**

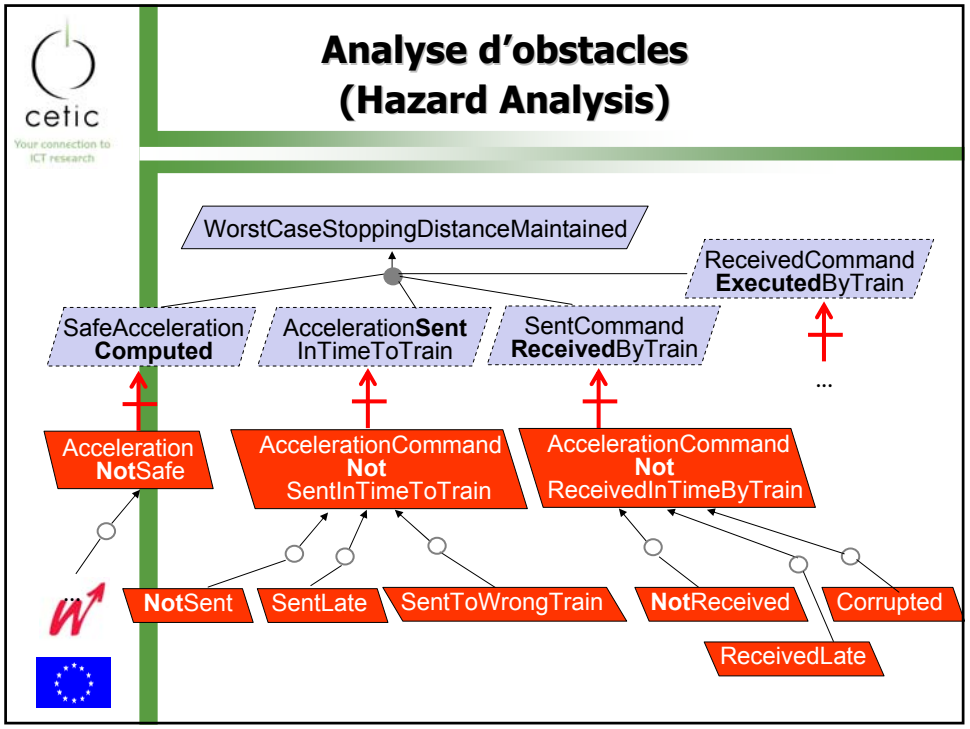
L'IE orientée-but avec KAOS



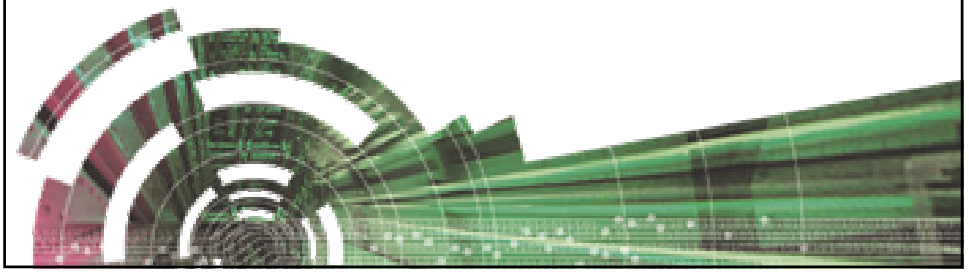
Modélisation des buts



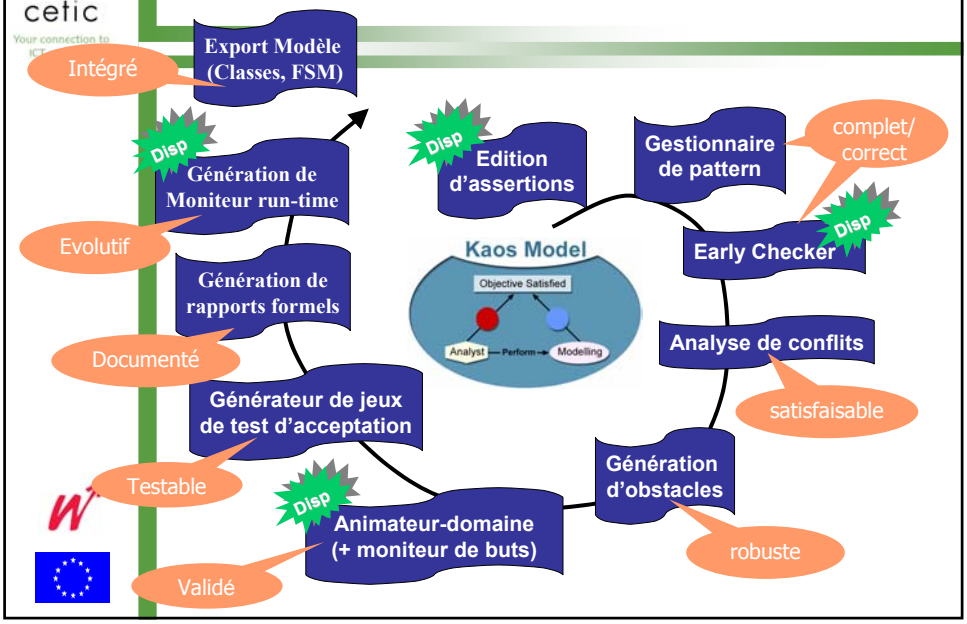

Niveau formel
Achieve [TrainProgress]
On (tr, b) ⇒ ◊ On (tr, next(b))
case-driven
Achieve [ProgressWhenGo]
On (tr, b) ∧ Go [next(b)]
⇒ ◊ On (tr, next(b))
Achieve [SignalSetToGo]
On (tr, b) ⇒ ◊ Go [next(b)]
Preuve mathématique
- cachée (pattern)
- vérifiable (cf. analyseur)
Maintain [TrainWaiting]
On (tr, b) ⇒
On (tr, b) W On (tr, next(b))


Analyse d'obstacles
(Hazard Analysis)
WorstCaseStoppingDistanceMaintained
ReceivedCommand ExecutedByTrain
SafeAcceleration Computed
AccelerationSent InTimeToTrain
SentCommand ReceivedByTrain
...
Acceleration NotSafe
AccelerationCommand Not SentInTimeToTrain
AccelerationCommand Not ReceivedInTimeByTrain
...
NotSent
SentLate
SentToWrongTrain
NotReceived
Corrupted
ReceivedLate

Support logiciel



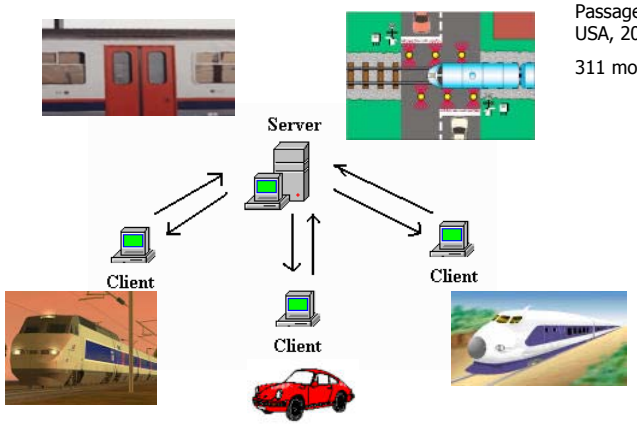
Atelier Intégré FAUST



Validation par animation orientée « but »

- Animations partielles orientée objectifs (scope)
  - Scope
  - Génération de machines à état parallèles
- Scénarios contraints par les exigences
- Détection de violations
- Validation par des experts du domaine (visualisations graph.)
- Validation itérative, incrémentale (générée à partir du modèle)
- Mise au point du modèle par scénarios permis/interdits

Animation multi utilisateurs



Passages à niveaux USA, 2002:

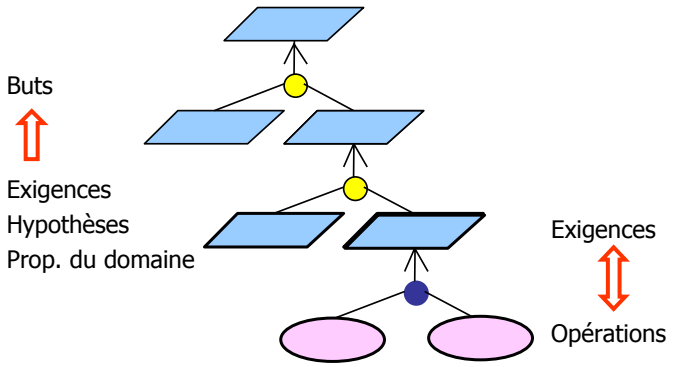
311 morts / 859 blessés

Vérification: le système construit est-il correct ?

- **Objectifs:**
  - Vérifier la correction de modèles KAOS pendant leur élaboration
  - Générer des contre-exemples en cas d'erreur
  - Générer des exemples de comportement possibles
  - Intégration avec l'animateur pour la visualisation
- **Vérifications réalisées: de nature locale**
  - raffinement de but
  - opérationnalisation
  - condition de conflit
  - **...**
- **Techniques**:
  - Techniques de "model-checking": exploration sur des domaines finis
  - Utilisation d'outils formels existants: nuSMV, Alloy, Oz
  - Mapping de KAOS sur langage formel des outils cibles

Vérification formelle

Vérification de nature locale et incrémentale !

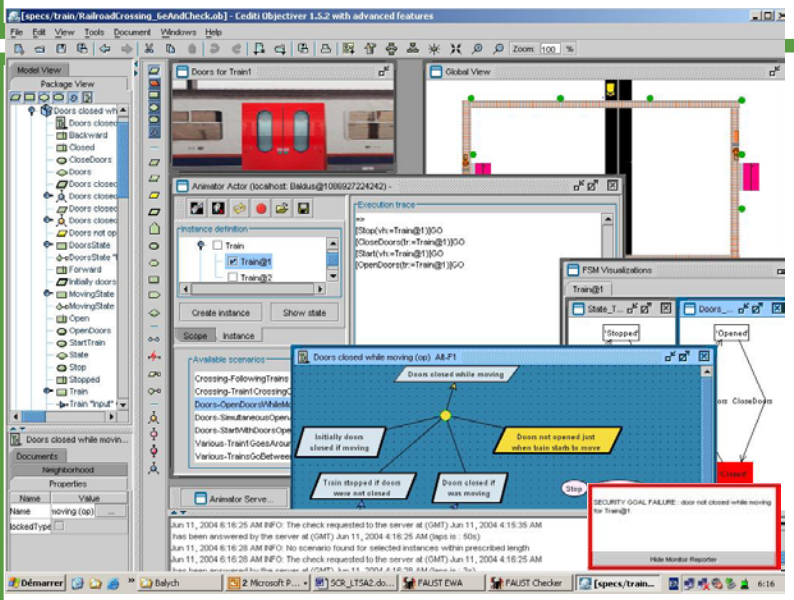


Le moniteur d'exigences

- **Objectif**: vérifier qu'un déploiement opérationnel est conforme à la spécification

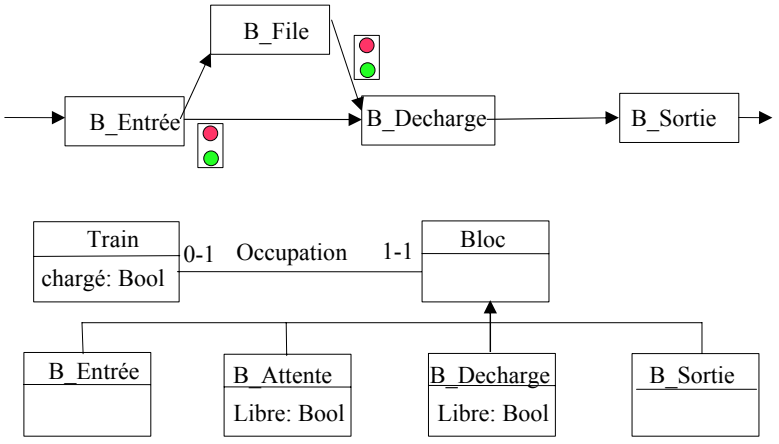
- **Applications:**
  - *Animation*: surveillance des buts, hypothèses (voir démo de l'animateur)
  - *Test*: génération d'oracle pour les tests
  - *Monitoring temps réel*: détection de bugs résiduels, de comportements rares ou déviants, statistiques sur la marge de manœuvre par rapport à des incidents/accidents

Démo de l'animateur

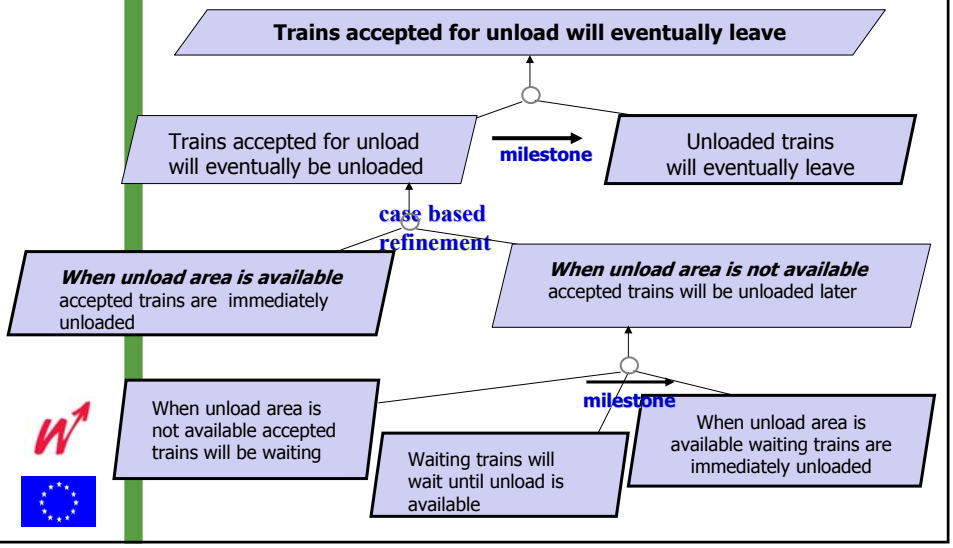


Génération de jeux de tests d'acceptation

Exemple: station de déchargement



Génération de jeux de tests d'acceptation



Stratégie de génération : assurer la couverture des buts

When unload area is available accepted trains are immediately unloaded

Unloaded trains will eventually leave

When unload area is not available accepted trains will be waiting

Waiting trains will wait until unload is available

When unload area is available waiting trains are immediately unloaded

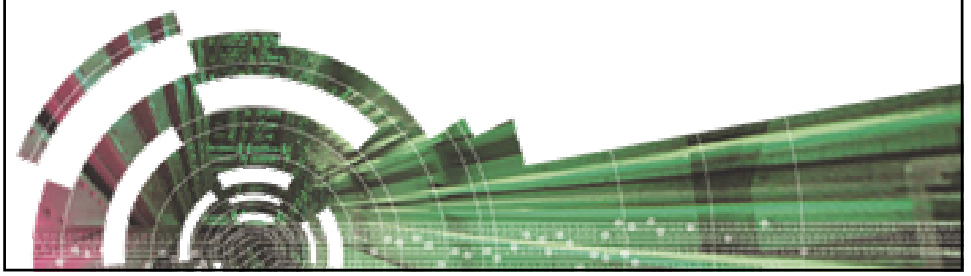
Unloaded trains will eventually leave

Ajout des propriétés du domaine
Satisfaction de contraintes
Génération de diagrammes de séquences

"Arrivée - déchargement disponible"

"Arrivée - déchargement Non disponible"

Etudes



Aperçu des études réalisées

- Ministère des finances: AGORA
  - Modélisation des données critiques
- Ministère des affaires sociales
  - Mesure des obstacles critiques pour les PMR
- Eurocontrol
  - Analyse d'interaction conflictuelles de système pour la résolution d'un danger de collision entre avions
- PROTON [UCL-Milos]
  - Etude a posteriori du design de PROTON

Domaine de la sécurité des personnes Eurocontrol : le problème

- Modélisation (partielle) du système
  - STCA et contrôleur
  - TCAS et pilote

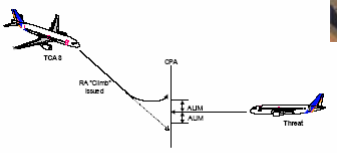


Figure 9: " Non-crossing " RA

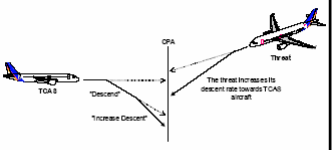
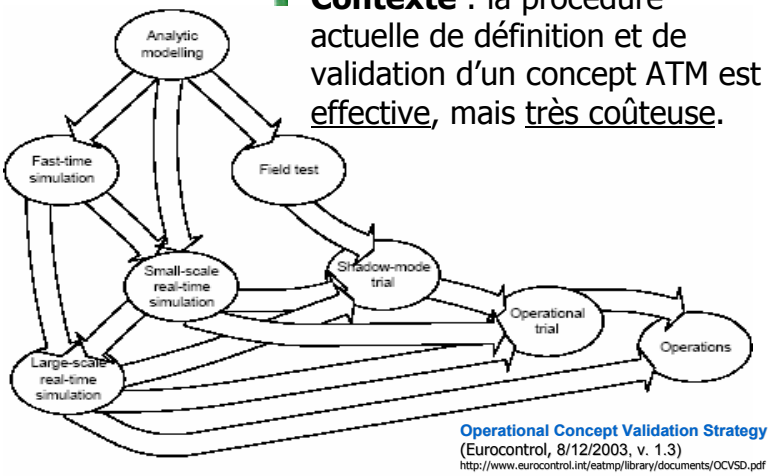


Figure 10: " Increase-vertical-rate " RA

ACAS PROGRAMME, ACASA PROJECT
ACAS/ACASA/02-017 March 2002
http://www.eurocontrol.fr/ba_saf/acas/WP6.pdf

Eurocontrol : le processus actuel

- **Contexte** : la procédure actuelle de définition et de validation d'un concept ATM est effective, mais très coûteuse.



Operational Concept Validation Strategy
(Eurocontrol, 8/12/2003, v. 1.3)
http://www.eurocontrol.int/eatmp/library/documents/OCVSD.pdf

Figure 4: Validation Route Map

Eurocontrol
Etude préliminaire de faisabilité

- Guides constructifs et incrémentaux pour la création, l'analyse, et les modifications du modèle.
  - exemples et contre-exemples sont une aide constructive à la définition des modèles, à l'analyse d'obstruction/«fault», l'analyse de sensibilité
  - des parties de modèles sont facilement réutilisables grâce à la création de « patrons » ATC, et on peut vérifier la bonne utilisation de ces « patrons » ATC.
- Traçabilité riche entre buts, exigences, comportements
  - ces guides constructifs créent automatiquement cette traçabilité et détectent automatiquement les liens incorrects
  - la traçabilité allège fortement les tâches de maintenance du modèle

Eurocontrol
Etude de faisabilité

- Le modèle obtenu est formellement correct
  - Détection automatique des erreurs
  - Petites formules, reformulées en langue naturelle

  =>préservation des aspects de compréhension, expression et de communication
- La formalisation est très utile pour la validation, grâce:
  - aux exemples de comportements permis/interdit, générés automatiquement par le système
  - à l'outil d'animation pour les visualiser dans le domaine
  - à la modélisation des comportements humains et des hypothèses faites sur l'environnement du système.
- Adaptation et intégration possible dans le processus de développement des concepts ATC mis actuellement au point par Eurocontrol.

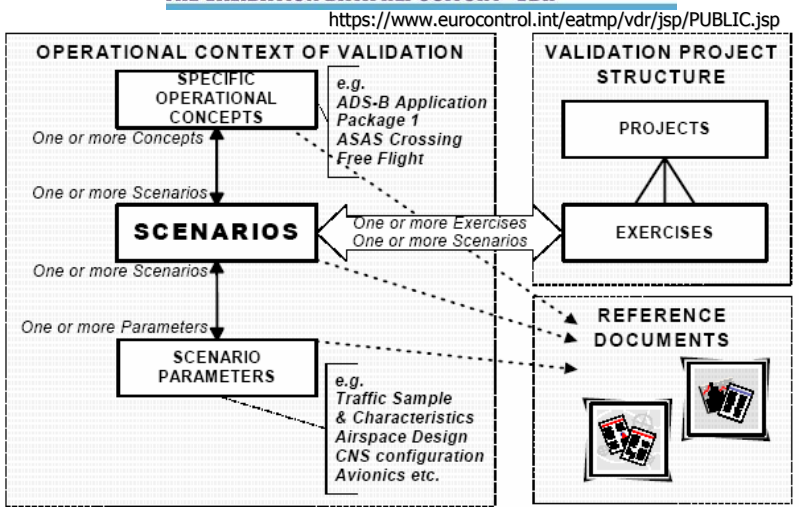
Eurocontrol
Utilisation potentielle des outils

THE VALIDATION DATA REPOSITORY VDR

https://www.eurocontrol.int/eatmp/vdr/jsp/PUBLIC.jsp

- VDR Repository
  - Data Analysis - Qualitative
  - Data Analysis - Quantitative Descriptive
  - Data Analysis - Quantitative Economic
  - Data Analysis - Quantitative Statistical Modelling
  - Data Collection - Automated System Logging
  - Data Collection - Data Obtained from Participants After Exercise
  - Data Collection - Data Obtained from Participants During Exercise
  - Data Collection - Physical Data Measurements from Participants
- Une aide peut être apportée à certaines de ces activités.

Eurocontrol
Utilisation potentielle des outils

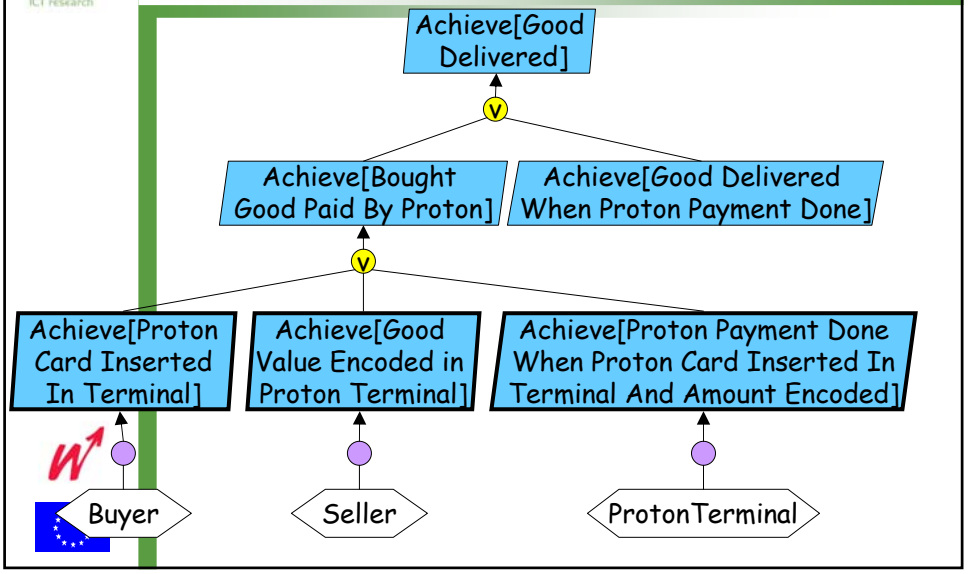


Domaine de la sécurité des informations : PROTON

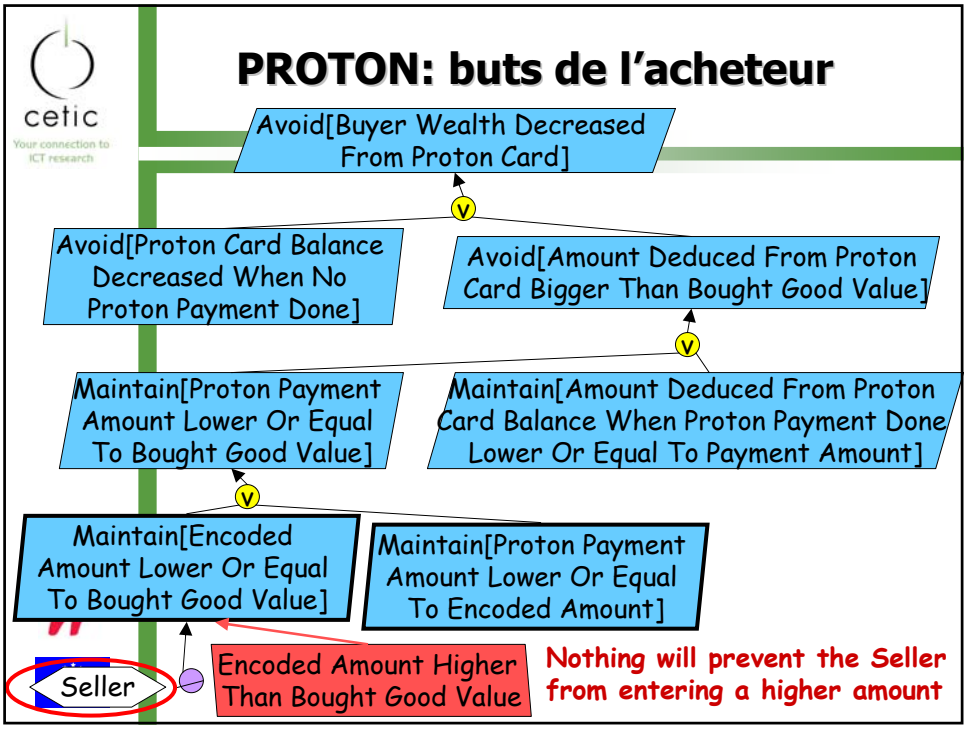
- Porte-monnaie électronique
- Rechargeables
  - aux ATM
  - téléphones publics

  (PIN requis)
- Paiement hors ligne au terminal du vendeur: le terminal collecte la monnaie "électronique"
- Mise en ligne du terminal pour alimenter le compte du vendeur

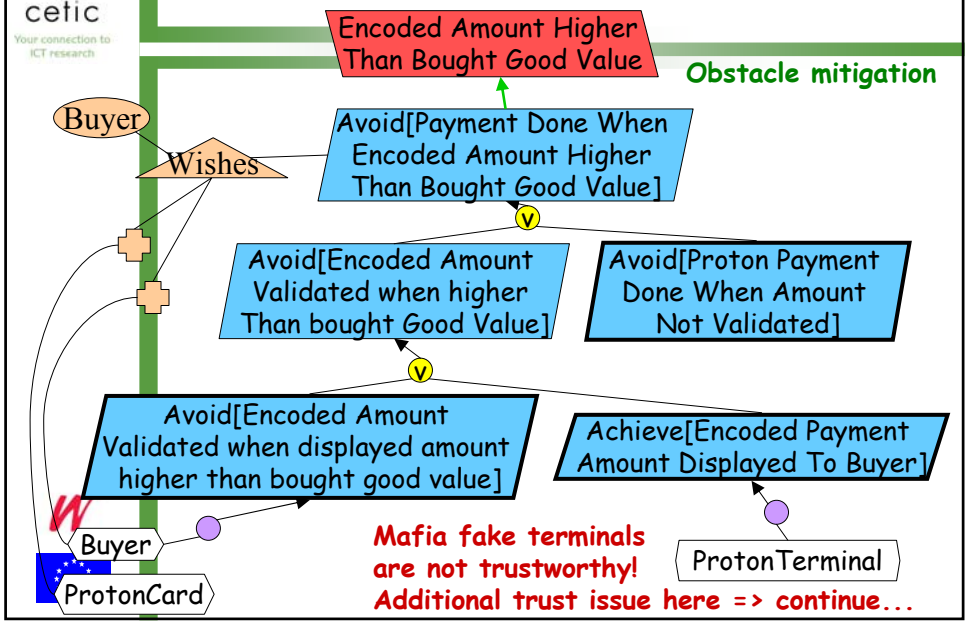
PROTON – aspects fonctionnels



PROTON: buts de l'acheteur



PROTON – Résolution du problème de confiance



Conclusions

- Modélisation, vérification et validation de systèmes ayant des parties critiques
- Démarche de modélisation sur plusieurs niveau de formalisme adapté au degré de criticité => large spectre d'applications
- Valorisation énorme du modèle en aval de l'IE !
- Sujets de discussion:
  - Quels sont les aspects critiques de vos activités (ou de vos clients) ?
  - Comment les gérer vous ? Avec quels techniques/outils ?
  - Faites vous la vérification et validation des exigences ?
  - Avez-vous recours à des jeux de tests d'acceptation ?
  - Avez-vous une idée du degré de votre marge de manœuvre ?
  - ...