
JOURNEES NEPTUNE

Les journées NEPTUNE ont eu lieu le 6 et 7 Juin 2012 à Télécom ParisTech. Cette conférence est un lieu de rencontre des acteurs du génie logiciel et de l'ingénierie des systèmes. Une particularité de cette conférence est qu'elle rassemble une majorité d'industriels. Ces grands acteurs de l'industrie ont présenté leurs idées et expériences concernant l'ingénierie dirigée par les modèles dans différents contextes.

JOUR 1

La première journée, une cinquantaine de personnes ont assisté à des conférences centrées sur la thématique de l'IDM appliquée aux systèmes d'information.

L'ACTIF IMMATERIEL DU SI

Pierre Bonnet – www.pierre-bonnet.com

L'intervenant a présenté une manière de déterminer la valeur des systèmes d'information à travers une méthodologie pragmatique. D'abord il sépare les propriétés tangibles (machines, outils, argent) des intangibles (marque, propriété intellectuelle, réseaux de partenaires, connaissance acquises et logiciels) et montre que, pour les marques, il existe des initiatives de normalisation, comme la dernière norme ISO qui définit des règles comptables pour définir leur valeur. Cela n'est pas le cas pour les systèmes d'information. Il faut une métrique et c'est cela qu'il propose.

L'intervenant définit la valeur du SI de trois manières.

- a. IS BUSINESS value : satisfaction des actionnaires, agilité de réponse au marché,
- b. IS USE value (valeur d'usage) : organisation de l'entreprise,
- c. IS INTRINSIC value : richesses des données stockées dans les bases de données.

Il existe sur le marché des frameworks pour déterminer la valeur des deux premiers, mais pas pour la troisième.

Pour évaluer cela il définit trois axes.

- a. Gestion de la connaissance, Documentation.
- b. Caractéristiques de la gouvernance, Disponibilité
- c. Gestion du SI, Implémentation des règles de processus.

Il génère donc un formulaire pour ces trois axes, et la valeur du système d'information est déterminée à partir des réponses données au formulaire.

AUTOMATIC BUSINESS LOGIC

Patrick Albert, Leader IBM CAS France

L'intervenant a montré une manière intéressante de remplacer le codage machine par une syntaxe plus proche du langage humain. Le but est d'avoir des langages plus naturels qui puissent être interprétés par les machines. Cela pourrait faciliter la vie d'un manager qui, au lieu de s'évertuer à utiliser un langage de programmation, pourrait se concentrer sur la logique économique de son marché en utilisant un langage plus adapté.

Voici un exemple :

Langage orienté objet	Langage Naturel
If (customer.status==GOLD) then customer.setDiscount(0.2)	If customer status is GOLD, apply a discount of 20%

Avec un modèle linguistique il est possible de décrire un modèle de business qui peut être traduit dans un modèle exécutable, c'est-à-dire un modèle logiciel.

REMICS: REUSE AND MIGRATION OF LEGACY APPLICATION TO INTEROPERABLE CLOUD SERVICES.

A. Henry – Netfective

Dans cette présentation, A. Henri a montré une méthodologie pour faire migrer du code COBOL vers les nuages globaux de manière automatique et simple. Cette migration se fait par la conversion des algorithmes (récupérés directement du code) vers un modèle UML, et une fois que ce modèle est complet, les algorithmes sont traduits dans un langage cible pour une plateforme cloud ou une architecture spécifique. Le but est de restaurer le savoir qui a été perdu.

Le budget utilisé pour ce projet qui a commencé en 2010 a été de 5,7 millions € avec un travail de 468 homme.mois.

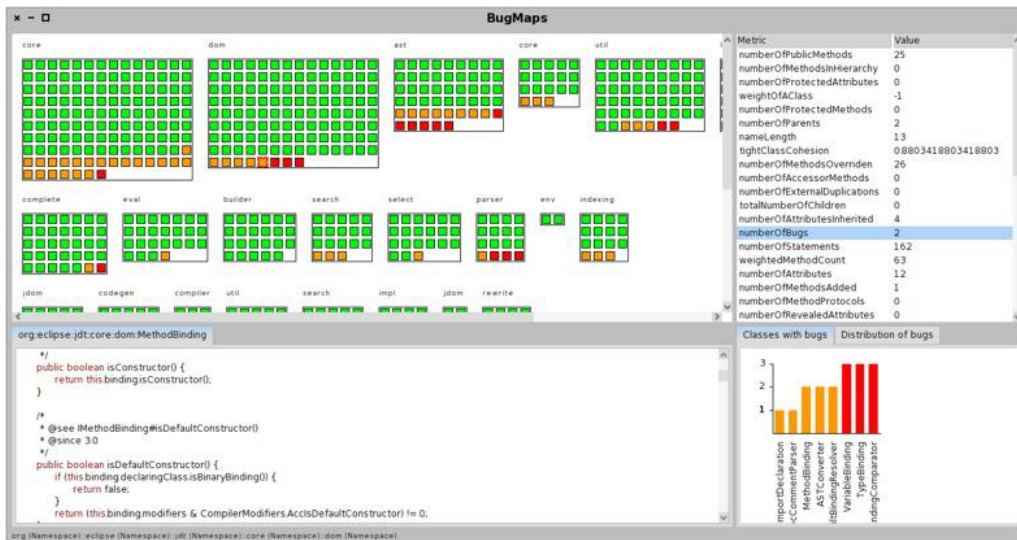
OUTILS DEDIES POUR L'ANALYSE DES SYSTEMES D'INFORMATIONS

S. Ducasse (INRIA)

L'idée est de rendre les énormes bases de données et de code visibles pour les gestionnaires de projet, les gens qui prennent les décisions. Pour faire cela, S. Ducasse a présenté un outil graphique pour la visualisation de l'évolution du code dans un grand projet.

Il est donc possible de lire une grande base de données et d'afficher dans une carte les parties de code qui sont en développement ou bien qui ont encore des bugs.

Cela permet aux gestionnaires du projet de bien localiser les problèmes et d'évaluer si leurs actions ont donné de bons résultats. L'objectif principal est de faire des analyses des systèmes d'information.



IDM ET METHODOLOGIE DE GESTION DE PROJETS

P. Laurent (Sopra Group)

L'intervenant met l'accent tout au long de sa présentation sur l'intérêt de l'IDM et son impact majeur sur le cadre global des projets ainsi que sur les métiers de l'ingénierie.

Les rôles et responsabilités qu'apporte l'IDM dans un projet sont assez diversifiés :

- le management intégrateur (l'assistance à la maîtrise d'ouvrage AMOA),
- les référents intégrateurs (modélisation, spécifications, qualification, implémentation, optimisation...),
- le management client (pilotage du projet, manipulation des applications),
- MOA / MOE Client (architecture, normes, validation, conduite du changement...).

Avec cet impact important sur les projets, l'IDM a créé un nouveau métier que l'intervenant nomme 'Modeleur'. Alors que l'analyste définit des modèles pour la compréhension et la documentation du projet, le modeleur définit des modèles pour l'exécution et l'automatisation.

Pour conclure, l'intervenant distingue deux aspects concernant le développement de l'IDM. L'aspect métier où l'IDM a des impacts profonds sur les personnes et leurs métiers, et l'aspect modèle, où les modèles de l'analyste et ceux du modeleur doivent toujours rester en cohérence.

JOUR 2

Au cours de cette deuxième journée, une audience de plus de soixante personnes a assisté à des présentations sur des expériences de l'industrie en rapport avec la modélisation des systèmes.

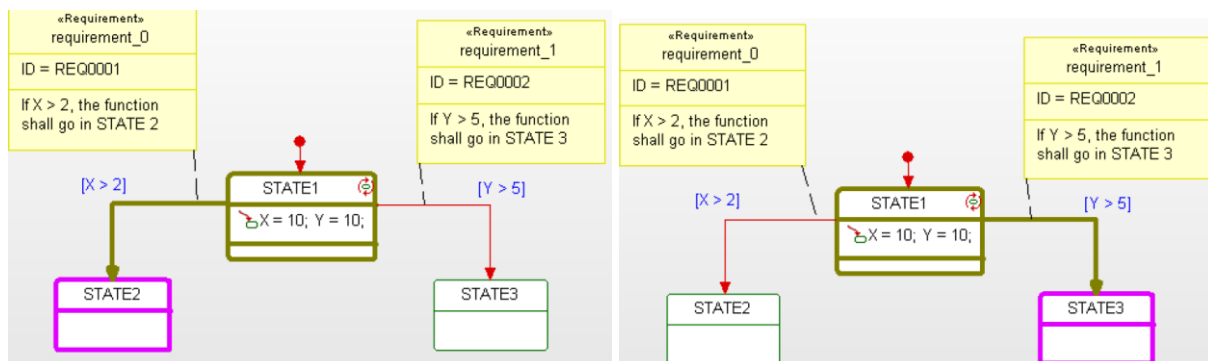
MODELISATION DANS UN CONTEXTE INDUSTRIEL.

D. Lesens (Astium)

Dans le contexte de la gestion d'un système complexe telle qu'une station spatiale, la communication entre différentes équipes est d'importance cruciale. Sachant que dans ces types de projet, les différentes équipes ont des nationalités différentes, il faut un langage commun pour que tout le monde parle de la même chose. L'objectif de ce projet est donc l'amélioration de la communication entre équipes pour permettre un développement du système en harmonie.

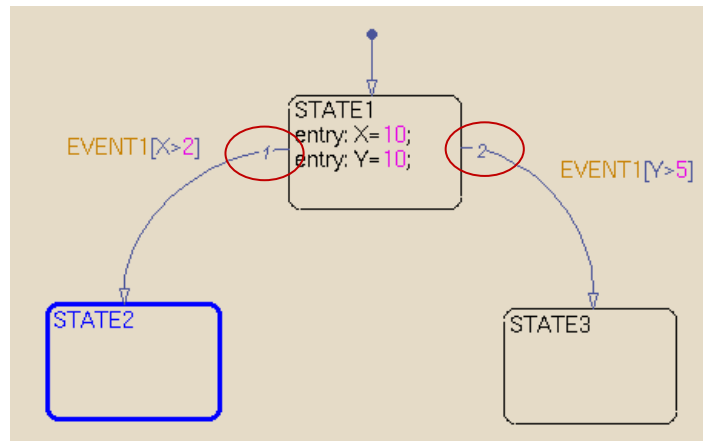
L'intervenant a commencé son discours en disant que l'IDM n'est pas de l'UML ou bien du SysML. Il défend le fait que ces langages sont de bons candidats pour l'approche IDM mais ne constituent pas une méthodologie. Ensuite, il a montré une comparaison entre certains langages de modélisation utilisés par les architectes systèmes.

Il a montré que SysML est un des langages les plus répandus dans l'industrie pour sa généralité mais malheureusement il n'est pas assez formel. Pour illustrer ce cas il a donné l'exemple suivant :



Il s'agit d'une machine d'état où les conditions de passage d'état peuvent arriver en même temps. SysML ne permet pas de décrire une préférence de choix, créant ainsi une ambiguïté dans le modèle.

Si nous essayons de faire la même chose avec Simulink, il est possible de déterminer quelle transition est prioritaire, créant ainsi un modèle avec un fonctionnement déterministe.



Cependant, ce langage n'est pas intuitif, même s'il est bien documenté. Finalement, l'intervenant a montré SCADE, un outil de modélisation qui vérifie les ambiguïtés dans les modèles. Cet outil alerte l'utilisateur s'il y a un problème d'ambiguïté dans l'étape de compilation.

Il a fait la même comparaison pour les langages textuels. Il a montré qu'ADA est un langage plus sécurisé que C++ ou Matlab car toutes les variables sont typées et les conversions implicites ne sont pas permises.

Finalement, il a présenté son expérience d'utilisation de SysML dans un vrai projet qui a été réalisé et lancé dans l'espace. Il a valorisé le fait que les modèles SysML servent aussi pour générer toute la documentation du projet de manière automatique et homogène.

MODELISATION : OUTILLAGE ET INTEGRATION.

E. Gaudin (Pragmadev)

Partant de la motivation de tester le plus tôt possible, l'intervenant a présenté une approche de test de modèles au premier niveau du cycle de projet en V : la phase de spécification. Pour cela il utilise un SDL (Specification and Description Language) standardisé par ITU-T. Il s'agit d'un langage formel et orienté objet qui permet la simulation du comportement d'un système. En étant formel, ce langage évite les ambiguïtés discutées dans la dernière présentation. Il a montré un exemple de contrôle d'accès à un bâtiment. Dans cet exemple, il a montré l'utilisation des divers graphes de son langage et l'exécution en temps réel du modèle. Ce langage lui a même permis de modéliser le clavier utilisé pour entrer le code d'accès.

Son outil est très complet. Il est possible de modéliser un système hétérogène et de faire communiquer les différentes parties du système (algorithmes, machine d'état, interfaces d'entrée et sortie). Il est aussi possible de modéliser les scénarios de test et de faire le débogage pas à pas. Tout cela sans générer du code.

MIGRATION VERS L'IDM : UN RETOUR D'EXPERIENCE CONCRET SUR LA MODELISATION ET LES OUTILS.

J.-F. Tilman (Geensyde)

M. Tilman a présenté une expérience qu'il a eue en utilisant une méthodologie IDM pour développer un produit concret, dans ce cas, un calculateur avionique. En connaissant l'architecture du calculateur, des cartes et les interfaces d'entrées et sorties, il est intéressant de modéliser tout ça dans un langage graphique. Le but est de partager l'information sans risque de mauvaise interprétation. C'est un discours similaire à celui des entreprises avec de grands projets. Ce type de travail sert à faciliter la communication entre les équipes et évite les risques d'ambiguïté de spécification. M. Tilman n'a pas voulu entrer en détails par rapport à son projet, mais il a dit qu'avant de choisir les outils et les langages de modélisation, il faut bien savoir ce que l'on doit modéliser.

Le choix du langage est très important. Il faut une syntaxe robuste avec une sémantique claire. Le choix de l'outil se fait par préférence par rapport aux capacités d'édition, simulation, débogage, traçabilité.

Sa conclusion a été qu'une démarche fondée sur les modèles n'est pas triviale, elle exige du travail et la formation de gens. Le succès vient d'un bon compromis entre les enjeux et les coûts d'implémentation de telles méthodologies, et les bénéfices que cela pourrait apporter.

L'IDM POUR LA SECURITE ET LA SURETE DANS LA RADIO LOGICIELLE.

E. Borde

L'intervenant présente le projet PARSEC. La motivation du projet réside dans les exigences de sûreté de fonctionnement, les exigences de certification et les contraintes d'intégration système. L'objectif présenté est de définir un processus unifié, basé sur l'IDM, pour le développement des systèmes critiques dans le domaine de l'avionique et du ferroviaire, en utilisant des outils dédiés.

L'intervenant présente la problématique sous forme de question : comment prouver que le code généré est sémantiquement équivalent à la description d'architecture ?

Au cours de ce projet, l'équipe avec laquelle il travaille a défini le langage de modélisation COAL qui possède une sémantique de flot de données. Ils utilisent B-événementiel pour décrire le formalisme des modèles COAL. À partir d'un modèle COAL, ils génèrent un modèle AADL pour ensuite générer du code C/ARINC/Makefiles :

1. point de départ – modèle COAL
2. AADL logique : data flow vers control flow
3. AADL ARINC : raffinement en éléments sémantiques moins complexes pour l'analyse et génération de code.
4. Génération du code C/ARINC/Makefile

Le raffinement progressif permet de détecter une parité 1 par 1 entre les modèles. L'interlocuteur conclut en présentant les travaux en cours, à savoir l'évaluation du déterminisme du langage AADL.

DSL GRAPHIQUE : LE MOYEN D'ÉQUIPER DES SPECIALISTES METIER D'OUTILS SUR MESURE

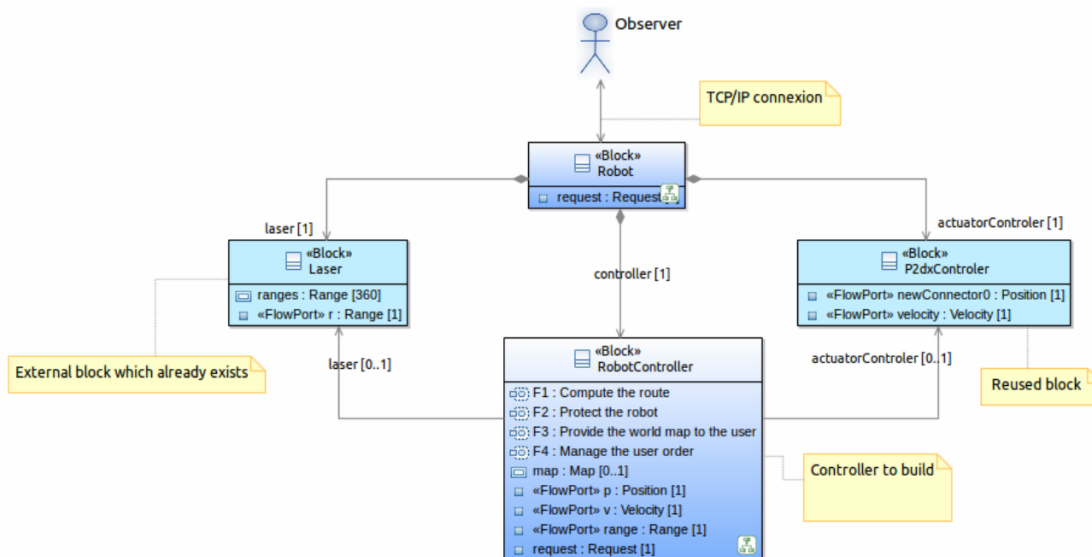
E. Juliot (OBEO)

E. Juliot a présenté une manière de créer des DSLs sur mesure pour des besoins spécifiques des entreprises. Dans tous les grands projets il y a des problèmes de management, des contraintes de « time to market » et de sécurité. La principale motivation d'OBEO avec cet outil est de promouvoir l'intégration des ingénieurs de plusieurs domaines. L'ingénierie des systèmes est un métier multidisciplinaire qui entoure différentes vues d'un même projet.

La vision d'OBEO est qu'il est impossible d'avoir un outil universel. Un ingénieur tout seul ne peut pas tout savoir. L'idée est d'avoir une plateforme unique pour la gestion de projet et de laisser les ingénieurs spécialistes travailler dans leurs domaines.

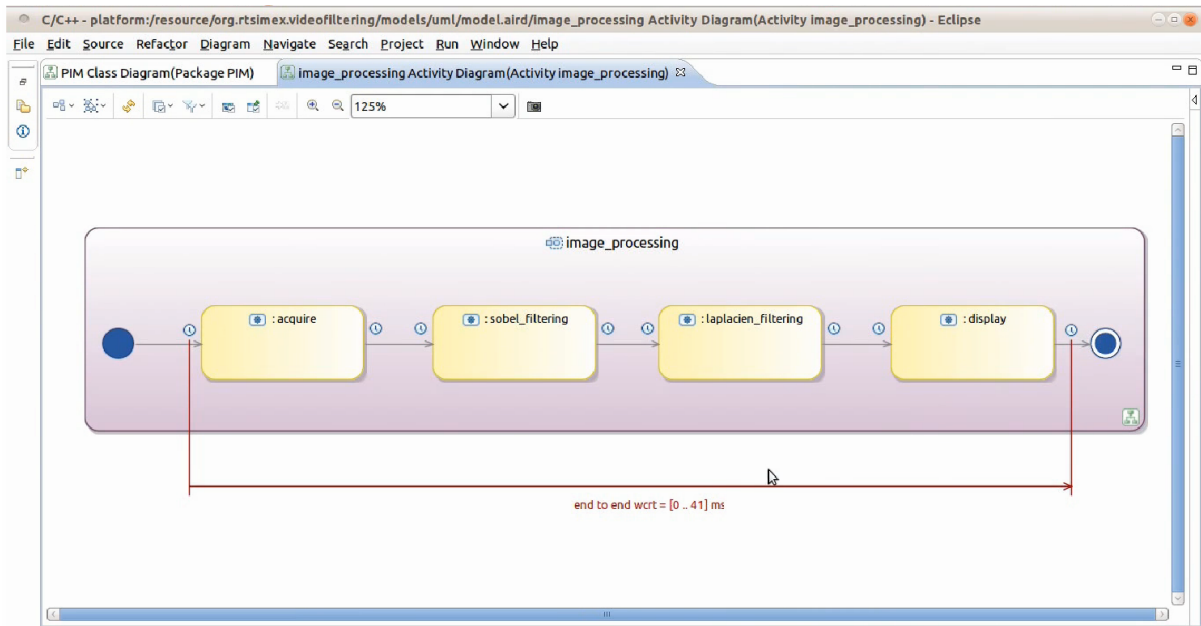
M. Juliot a montré un exemple très intéressant en montrant 4 vues d'un projet d'un robot. Une vue système, une vue de contraintes de temps, une vue sécurité et une vue de spécification.

Vue Système :



La vue système sert à décrire la structure du système. Les dépendances et l'utilisation matérielle. Cette vue n'est qu'un diagramme UML avec les boîtes noires du système.

Vue Temporelle :



La vue temporelle ajoute des annotations dans le modèle pour définir certaines contraintes de temps telles que les deadlines d'exécution d'un code pour les systèmes embarqués par exemple.

Vue de sécurité :

1 - CBTC_HA Primary Hazard Analysis

1.1 - Operational Contexts analysis

1.1.1 - Normal at Change_of_ends_phase in Change_of_ends_zone analysis

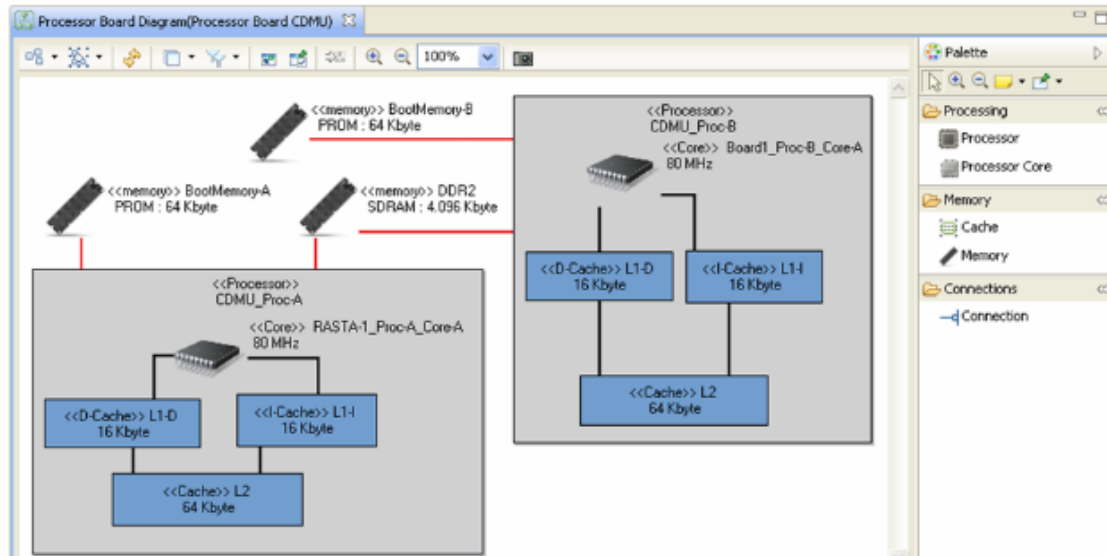
The current operational context is:

- Mode: Normal
- Zone: Change_of_ends_zone
- Phase: Change_of_ends_phase

Accident				Barrier			Qualitative Analysis		
Accident	Severity	Hazard	Consequences	Function	Responsibility	Requirement	TAR	THR	SIL
Collision with another train									
	catastrophic	Wrong travel direction					improbable	occasional	
				F2 To space the 1000 issue	CBTC				4

La vue de sécurité montre avec un code couleur les spécifications par rapport à la sécurité du système. Les couleurs changent dynamiquement avec les changements dans le projet.

Vue de spécification :



Finalement, la vue de spécification sert à déterminer les contraintes du projet, telles que la taille mémoire, la vitesse du processeur...

« On ne va pas réinventer la roue » : Pour créer cette plateforme, ils ont utilisé SysML comme langage de base. L'outil est aussi basé sur Eclipse, cela permet une adaptation plus facile puisqu'Eclipse est un outil très diffusé dans l'industrie et l'académie.

Une caractéristique intéressante est la synchronisation entre les différents points de vue. Si un ingénieur travaille dans une vue et quelqu'un d'autre fait un changement, cela bloque tout autre changement jusqu'à ce que le premier ait fait une sauvegarde.

L'interlocuteur a montré un point de vue de l'industrie par rapport aux DSLs. Il dit que les DSLs ne sont que de l'UML industrialisé. Pour lui, utiliser de l'UML pur dans toutes les applications industrielles est une utopie. Il défend que UML ne doit pas imposer une sémantique, que cela doit être rajouté par un DSL.