

SysML

Quelques remarques suite aux relectures AFIS

Françoise Caron, animatrice du GT Méthodes et Outils

mo@afis.fr

Les enjeux

D'une façon générale, la coordination des activités industrielles tout au long du cycle de vie des produits souffre d'un défaut de standards. Lorsque les produits sont des systèmes, les difficultés dues à la coordination des travaux et la maîtrise de la traçabilité sont particulièrement complexifiées. L'adoption de standards permettant de formaliser, structurer, gérer et partager l'information est indispensable à la maîtrise des architectures comme à celle des programmes de développement. C'est pourquoi, il est de la mission du groupe « Méthode et Outil » de coordonner les travaux de veille de l'AFIS sur l'émergence et les évolutions de ces standards qui incluent SysML.

Le langage SysML

SysML (*Systems Modeling Language*) se présente comme un langage visant à supporter une approche « orientée modèles » dans les processus techniques d'ingénierie système. De même que, dans le domaine des logiciels, UML (*Unified Modeling Language*) a fédéré et standardisé les principes et les techniques de modélisation issus d'initiatives hétérogènes, l'objet de SysML est de fédérer les principales techniques de modélisation de l'ingénierie système. Par voie de conséquence, SysML vise aussi à rationaliser la diversité des outils de modélisation.

SysML résulte d'une collaboration entre l'INCOSE (*International Council for Systems Engineering*), et l'OMG (*Object Management Group*), organisme éditeur d'UML. SysML est basé sur UML2 et en définit un paramétrage dédié à l'ingénierie système. Il traite deux sujets de modélisation bien distincts selon nous : le premier concerne les architectures système et le second, les sorties des activités techniques d'ingénierie système et leurs règles de traçabilité. L'objet de la modélisation dans le premier cas sont les systèmes eux-mêmes et, dans le second, les données gérées par les ateliers d'outils informatiques en support à l'ingénierie.

Nous nous proposons de présenter très brièvement quelques points clés des évolutions qu'introduit SysML par rapport à UML2.

Modélisation d'architectures système

SysML définit les éléments constitutifs de l'architecture par des **Blocks**. Leurs frontières sont caractérisées en termes de services requis/fournis (**Service Ports**) et/ou d'entrées/sorties acceptées (**Flow Ports**). Les entrées/sorties peuvent être soit discrètes, soit des flux continus.

Les diagrammes statiques, permettant de décrire les relations inter et intra-blocks, dérivent du diagramme de classe d'UML. Mais, pour prendre en compte les problématiques de compromis et d'optimisation de l'analyse système, SysML introduit un concept complémentaire, les contraintes paramétrées, et un diagramme dédié, le **parametric diagram**. Les contraintes paramétrées permettent, d'une part, de définir des règles d'impacts entre valeurs de propriétés (masse, surface, force, pression, ...) et, d'autre part, d'affecter ces règles aux **Blocks** pourvus des propriétés.

Les diagrammes de cas d'utilisation et, pour les diagrammes dynamiques, les diagrammes de séquence et les machines à état sont repris directement d'UML2. Toutefois, en ce qui concerne les diagrammes d'activité, SysML introduit des évolutions significatives en vue d'une mise à niveau avec les techniques de modélisation par EFFBD (*Enhanced Functional Flow Block Diagrams*, ou diagrammes d'enchaînements fonctionnels).

Modélisation de données d'ingénierie et traçabilité

SysML définit les exigences par leurs attributs a minima et la sémantique des stéréotypes de leurs relations : « **dérive** » et « **copy** » (exigence/exigences), « **satisfy** » (élément d'architecture/exigences) et « **verify** » (cas de test/exigences). SysML définit aussi un stéréotype de relation entre données de types différents : « **allocate** » mais ici, sans en contraindre la sémantique.

Les relectures de l'AFIS

Une entreprise française seulement, Thales, a participé à l'élaboration du standard. C'est pourquoi le groupe « Méthodes et Outils » a voulu fournir la possibilité aux adhérents AFIS qui le souhaitaient d'exprimer leur point de vue selon un canal plus

officiel que le forum SysML en leur proposant celui de l'AFIS, chapitre de l'INCOSE.

Nous avons organisé deux relectures, l'une en 2004 sur la version V0.8 du standard et l'autre récemment, au cours du quatrième trimestre 2006, sur la version « *Draft Adopted Specification ptc/06/05/03* » du 3 mai 2006. Dans les deux cas, nos remarques ont été accueillies de façon très positive par le comité de standardisation et ont abouti à l'ouverture d'« *issues* ». Nos remarques et les retours du comité sont disponibles dans le système documentaire accessible à partir du site AFIS.

Notre point de vue

Modélisation d'architectures système

En ce qui concerne les systèmes à logiciels prépondérants, UML2 offre déjà beaucoup de possibilités de modélisation. Toutefois, avec les **Flow Ports** permettant de spécifier des flux d'entrées/sorties SysML lève une limitation significative relative aux **Ports** UML2 limités aux services.

Cependant, pour notre part, nous regrettons deux points majeurs.

D'une part, la classe UML n'est plus présente dans SysML qui prédéfinit tous ses stéréotypes dérivés de classes (**Block**, **Constraint Block**, **Requirement**, **Test Case** ...). Un utilisateur du strict profil SysML, s'il a besoin de classes non prévues par le standard, devra stéréotyper des **Blocks** pour redéfinir des classes de base ! Cela nous semble d'autant moins heureux qu'un des objectifs de SysML est de faciliter la communication entre ingénieurs système et développeurs de logiciels. Et ces derniers manipulent des classes et non pas des **Blocks**. Néanmoins, une solution pourrait venir des outils s'ils permettent de charger, conjointement au profil SysML, un sous ensemble approprié des possibilités d'UML, par exemple un profil UML dédié aux modèles de domaine métier ou d'analyse logicielle. Ceci permettrait d'allouer aux **Blocks** système, des classes métier dont la spécification pourrait être réutilisée pour le développement des logiciels.

D'autre part, avec l'absence du concept de classe, vient celle de la distinction entre classes et instances (ou objets) qui, de notre point de vue, a son importance. C'est une chose d'exprimer une spécification portée par une classe et devant être satisfaite par tous les objets de la classe susceptibles de concrétisation. C'est autre chose d'exprimer, sur un diagramme d'instance, une façon de mettre en jeu les spécifications des classes : par exemple, sur un diagramme de séquence, comment interviennent certains objets au cours de l'exécution d'un scénario

donné, ou encore, sur un diagramme de **Blocks**, les connexions point à point pour une configuration donnée.

Par ailleurs, les contraintes paramétrées, les flux continus et les évolutions du diagramme d'activité permettent désormais, pour utiliser un raccourci un brin provocateur, « de faire du fonctionnel » avec SysML. Néanmoins, la preuve par l'expérience reste encore à démontrer et nous y voyons aujourd'hui encore deux difficultés de mise en œuvre auxquelles SysML, en tant que standard de notation, n'a d'ailleurs pas la vocation de répondre.

La première est d'ordre méthodologique : comment harmoniser de façon pertinente une démarche d'architecture fonctionnelle supportée par SysML et les principes de modularité et d'encapsulation qui sont la base de la philosophie « objet » d'UML.

La seconde concerne les outils : les modélisateurs/simulateurs SysML ne pourront pas répondre à tous les besoins utilisateurs de simulation de contraintes. Au-delà de SysML, il sera nécessaire de développer des standards et des mécanismes d'interopérabilité avec des logiciels externes de calcul.

Modélisation de données d'ingénierie et traçabilité

SysML est centré sur les exigences, le domaine le plus exploré. Il nous semble, peut-être avec un peu de partialité que le modèle de données AFIS est plus avancé. Mais celui-ci nous a montré la difficulté à définir un modèle robuste et nous n'avons que trop conscience des questions encore ouvertes, par exemple l'articulation des données d'ingénierie et de tests au travers de la gestion de configuration système, auxquelles il faudra trouver des réponses avant d'aboutir à une réelle solution de traçabilité.

En conclusion

SysML nous apparaît, déjà dans sa version actuelle, comme un travail significatif de synthèse et de formalisation de l'état de l'art des pratiques de modélisation en ingénierie système. Nos échanges interdisciplinaires dans le cadre de l'AFIS et sur nos lieux de travail nous démontrent quotidiennement combien une telle initiative de fédération est loin d'être simple. Et si des points restent en débat, n'est-ce pas un reflet de la maturité de notre communauté professionnelle face aux champs d'exploration auxquels la confrontent l'ingénierie système et ses problématiques méthodologiques ? C'est pourquoi leur résolution ne peut pas appartenir aux seuls éditeurs d'outils et nécessite de la part des entreprises, notamment en France, l'investissement nécessaire à l'implication d'experts de terrain.