

---

# SPAS : un profil SysML pour les systèmes auto-adaptatifs

**Jean-Michel Bruel\*, Nicolas Belloir\*\* & Manzoor Ahmad\***

*\* IUT de Blagnac – Université Toulouse 2 – Le Mirail  
Département Informatique  
B.P. 60073, 31703 Blagnac Cedex, France*

**contact : [bruel@irit.fr](mailto:bruel@irit.fr)**

*\* Université de Pau et des Pays de l'Adour  
Département Informatique  
B.P. 1155, 64013 Pau Université Cedex, France*

**Sections de rattachement : 27 (informatique)  
Secteur : Tertiaire**

*RÉSUMÉ. Nous souhaitons présenter dans cet article un exemple de recherche appliquée, transversale et fédérative, comme le milieu des IUT peut en provoquer. Nous présentons un travail qui démarre seulement et dont les résultats ne pourront être évalués que dans quelques années, mais qui nous le pensons est tout à fait illustratif des intérêts à mener des activités de recherche en milieu IUT et donc tout à fait dans l'esprit de l'appel à communication du congrès. En effet ce projet consiste à développer un langage dédié à l'expression des contraintes des systèmes informatiques adaptatifs (c'est à dire qui modifient leur comportement en fonction de l'évolution de leur environnement extérieur). Le domaine applicatif que nous privilégions est celui des systèmes d'aide au maintien et au monitoring des personnes à domicile.*

*MOTS-CLÉS : Ingénierie des Modèles, Langages Spécifiques aux Domaines, SysML.*

## **1. Introduction**

L'aide à la surveillance et au maintien à domicile est un problème complexe qui requiert des compétences humaines et techniques variées induites par la multiplicité des expertises mises en jeu (sociales, médicales, techniques etc.), la dimension collaborative des analyses et des prises de décisions et enfin l'obligation de résultat lors des étapes de

gestion de crise. Les 3 départements de l'IUT de Blagnac (Informatique, Réseaux & Télécommunications, et Génie Industriel & Maintenance) ont été à l'origine de la création d'un quatrième département, fédérateur : Aide et Assistance pour le Monitoring et le Maintien à Domicile (2A2M). Cette formation ouverte depuis la rentrée 2008, propose un enseignement en sciences humaines (principalement en psychologie, sociologie et management) associé à une formation plus technique qui combine les compétences de ces 3 départements (transmission, réseaux, informatique). C'est une nouvelle formation innovante dans le domaine du service à la personne et dans sa transversalité entre technologies et sciences humaines.

Afin d'initier concrètement les collaborations sur ce thème, un projet de bien moindre envergure, mais aussi à plus court terme, PLEXUDOM (PLate-forme EXpérimentale dédiée à la SURveillance et au maintien à DOMicile) (Bruei *et al*, 2009), a été initié avec les deux laboratoires présents sur le site : les laboratoires LATTIS et IRIT, en particulier l'équipe Systèmes Communicants Sans Fil (SCSF) du LATTIS, l'équipe Modèles, Aspects, Composants pour les Architectures à Objets (MACAO) de l'IRIT, l'équipe SIG/DDSS (Systèmes d'Information Généralisés/Documents et Données Semi-Structurées) de l'IRIT, et l'équipe Ingénierie des Connaissances, de la Cognition et de la Coopération (IC3) de l'IRIT. Ces équipes travaillent sur des problématiques qui se trouvent complémentaires dans un certain nombre de domaines tels que les réseaux sans fil, l'électronique sans fil, l'informatique appliquée. L'objectif de ce projet est de proposer une infrastructure d'aide à la surveillance et au maintien à domicile qui prend en compte tous les aspects évoqués précédemment. L'architecture technique doit proposer des solutions fiables, performantes, sécurisées et validées permettant, dans un environnement distribué multiutilisateurs, de collecter des informations d'analyse, de piloter certains équipements, d'alerter certains acteurs ou institutions et enfin de produire une trace des événements rencontrés en vue d'analyses rétrospectives sur la qualité des processus. La validation des processus nécessite de définir, en concertation avec tous les types d'acteurs, des scénarios types reflétant toutes les circonstances d'exploitation de la plate-forme et de proposer des solutions respectant les contraintes (médicales, sociales, techniques, etc.) parfois critiques à traiter. La diversité des acteurs et de leurs fonctions, nécessite de définir et d'exploiter un système d'information adapté permettant tout à la fois de superviser (fonction de type tableau de bord) ou de prendre des décisions concertées (collaboratives) en apportant à chaque acteur l'information exacte dont il a besoin pour prendre ses décisions. Enfin, la mise en œuvre d'une telle plate-forme requiert l'usage d'outils de modélisation et de développement adaptés afin de satisfaire aux contraintes énoncées d'obligation de résultat en temps réel. Il existe à notre connaissance assez peu de travaux portant spécifiquement sur les langages d'expression des besoins dans le cadre des systèmes auto-adaptatifs. Dans le domaine de l'automobile des profils SysML ont été proposés pour la gestion des exigences de sûreté de fonctionnement (Dubois, 2008). Ces travaux visent surtout à permettre la réutilisation des outils de vérification et de validation existants dans un cadre IDM.

La Fig. 1 résume notre démarche qui consiste (1) à partir des besoins, de séparer les besoins fonctionnels (indispensables au fonctionnement) de ceux que l'on qualifie d'« extra-fonctionnels » (du type sécurité, persistance, etc.) ; puis (2) à appliquer un processus spécifique aux besoins qui vont être directement concernés par le caractère adaptatif du système ; (3) à partir d'un profil dédié, ces besoins particuliers seront modélisés puis (4) intégrés aux modèles classiques de l'application. Pour ce dernier point nous utiliserons des travaux en cours sur les connecteurs dédiés sur étagères. La suite de cet article est la suivante : nous détaillons le langage d'expression de contraintes que nous préconisons (RELAX, cf. § 2) et l'utilisation d'une notation normalisée pour supporter ce langage (SysML, cf. §3). Nous montrerons la nécessité d'adapter cette notation au langage d'expression des besoins et justifierons ainsi le développement adapté d'un profil dédié (SPAS, cf. §4). Nous détaillerons dans la §5 les résultats attendus de ces travaux qui débutent. Nous explorerons dans la §6 l'état de l'art en matière de modélisation des besoins dans les systèmes adaptatifs et nous concluons en §7 par les perspectives attendues de ce travail.

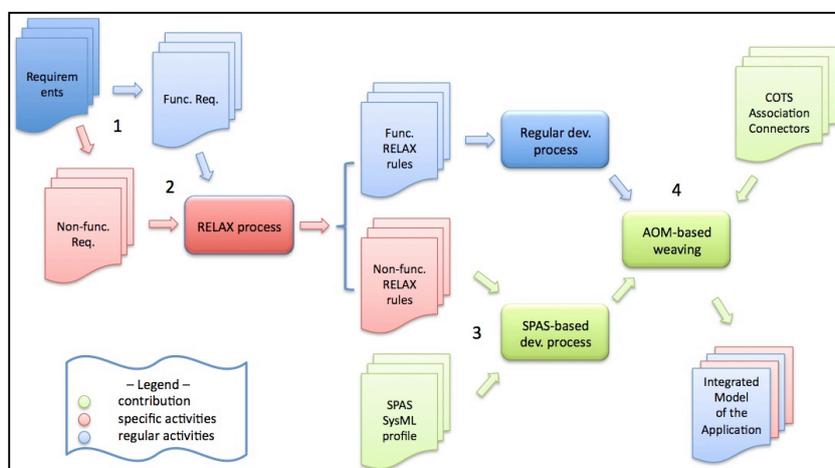


Figure 1. Vue d'ensemble de la démarche

## 2. Un langage d'expression de contraintes : RELAX

Nous travaillons activement à la définition d'un langage d'expression des besoins qui soit dédié aux systèmes auto-adaptatifs : RELAX (Whittle *et al*, 2009). Les systèmes auto-adaptatifs ont la capacité de modifier de manière autonome leur propre comportement en fonction des changements constatés dans leur environnement d'exécution. Les nouveaux mobiles qui basculent automatiquement en communication Wi-Fi (et donc gratuite) dès qu'ils sont proches de bornes d'accès en sont un exemple parmi bien d'autres (automobile, télécommunications, systèmes de monitoring, etc.). De nombreux travaux de recherche portent sur le développement et les architectures

dédiées à de tels systèmes, mais paradoxalement assez peu portent sur l'expression des besoins de tels systèmes. Le caractère auto-adaptatif de ces systèmes n'est donc considéré que comme une caractéristique comme une autre. Or, notre point de vue est que les contraintes et les besoins exprimés sur lesquels vont s'appliquer le caractère adaptatif du système doivent être traités de manière particulière. Ceci permettra : (i) de mettre l'accent et de justifier le caractère requis d'auto-adaptativité du système à développer ; (ii) de faciliter le traitement, l'évolution et la traçabilité de ces besoins spécifiques ; (iii) éventuellement de détecter des impossibilités ou des inconsistances au niveau de l'expression des besoins. Ce dernier point est d'autant plus critique que dans de tels systèmes (comme ceux évoqués dans l'introduction de système d'aide au maintien à domicile), les besoins sont exprimés par des intervenants complètement différents (par exemple : médecins, architectes, spécialiste des réseaux, informaticiens). Nous proposons ainsi un langage dédié à l'expression de ces contraintes. Ce langage, appelé RELAX, permettra d'identifier les besoins sur lequel le système a un certaine marge de manœuvre en terme d'adaptation (les besoins qui peuvent être « relâchés », d'où le nom du langage) par rapport à ceux sur lesquels il n'a aucune marge (invariants). L'intérêt d'un tel langage est de pouvoir bénéficier d'opérateurs particuliers et d'environnement de prise en charge spécifique (analyse, statistiques, etc.).

Le but (et la limite en longueur) de cet article ne nous permettent pas de détailler le langage RELAX, mais nous renvoyons le lecteur à (Whittle *et al*, 2009) pour plus de détails. L'étude de cas choisie pour y illustrer le langage est justement celui d'un système de surveillance de personnes à domicile (le système AAL). Voici à titre d'exemple un extrait des besoins exprimés en RELAX (cf. Fig. 2). Les besoins sont exprimés sous forme de règles. Les opérateurs en majuscule ont une sémantique définie formellement (utilisation d'une forme de logique floue, FBTL cf. Moon *et al*, 2004). Des attributs de la règle permettent d'exprimer les variables d'environnement sur lesquels portent la règle (ENV), les valeurs sur lesquels le système doit pouvoir s'adapter en les contrôlant (MON), les relations qui peuvent exister entre ces deux ensembles (REL), les liens avec les autres règles, le besoin précis du cahier des charges que la règle relâche, etc. Cette règle décrit que le frigo intelligent détecte la présence (et obtient en conséquence les informations importantes) d'autant d'éléments de nourriture que possible (via les puces RFID de ces derniers).

Dans ce travail, né d'une collaboration entre plusieurs pays, nous nous sommes fixé comme objectif de maintenant faire le lien entre le langage RELAX que nous avons défini et les notations utilisées plus généralement dans le domaine du développement de systèmes. Dans la section suivante nous décrivons la notation qui fait autorité en la matière, à savoir le langage SysML.

**R1.1'**: The fridge *SHALL* detect and communicate information with *AS MANY* food packages *AS POSSIBLE*.  
**ENV**: Food locations & food information.  
**MON**: RFID tags; Cameras; Weight sensors.  
**REL**: RFID tags provide food locations/food information; Cameras provide food locations; Weight sensors provide food information (whether eaten or not).  
**DEP**: R1.1' negatively impacts R1.2'; R1.1' positively impacts R1.4 and R1.5.

Figure 2. Exemple de règle RELAX

### 3. Pourquoi l'utilisation de SysML ?

Nous travaillons depuis quelques années (Belloir *et al*, 2008) sur le langage SysML (OMG, 2008). SysML est un langage de modélisation développé par l'OMG, INCOSE et AP233. SysML est un profil d'UML 2.0 fournissant aux ingénieurs les éléments de modélisation de systèmes qui leur manquait dans la notation (qui se veut universelle) UML. Son champ d'application est en ce sens plus large que celui d'UML mais sa filiation le rend tout particulièrement intéressant pour la modélisation de systèmes embarqués majoritairement composés de logiciel, comme c'est le cas des systèmes auto-adaptatifs qui nous intéressent ici. Nos travaux sur SysML ont porté sur l'utilisation de cette notation pour le développement des logiciels déployés sur les réseaux de capteurs sans fil. Dans ce type d'application la prise en compte de l'interaction forte entre le matériel et le logiciel inhérente à ce type de système est une condition importante pour une modélisation efficace. La raison pour laquelle nous souhaitons utiliser SysML est que cette notation supporte, de notre point de vue, assez bien l'expression des besoins (diagramme spécifique, cf. exemple en Fig. 3), le lien entre ces besoins et les éléments de modélisations du système (plusieurs liens de dépendances du type « *allocated* », « *satisfies* », etc.). De plus, une des difficultés de RELAX est l'expression des relations (REL) entre les paramètres (ENV et MON). Or, SysML dispose de la notion de diagramme paramétrique qui permet de faire le lien formel (mathématique même si la relation est connue comme dans le cas des lois physiques) entre paramètres.

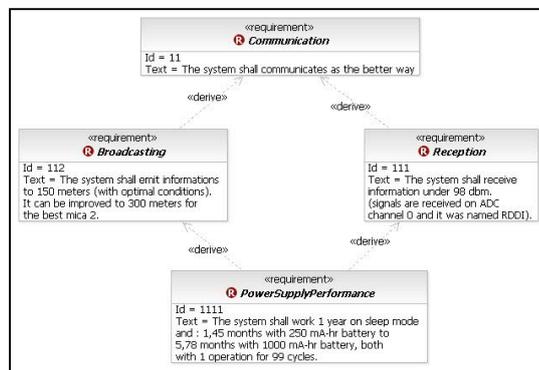


Figure 3. Exemple de diagramme SysML d'expression des besoins

De plus, l'avantage d'utiliser un langage comme SysML est la possibilité de disposer des outils qui commencent à atteindre une relative maturité pour cette notation. Plusieurs autres points plaident pour l'utilisation de SysML dans notre approche. D'une part, SysML a été conçu initialement pour permettre la modélisation de systèmes de systèmes, ce qui est le cas de notre domaine d'application dans lequel de nombreux systèmes peuvent être amenés à collaborer pour rendre les services attendus. D'autre part, SysML hérite d'UML des capacités d'extension permettant de spécialiser le langage pour un domaine cible donné, à travers les concepts de profil, de stéréotypes, de valeurs marquées et de contraintes. Ainsi, il sera aisé de spécialiser SysML à nos besoins en lui donnant un aspect moins informatique et plus adapté aux différents acteurs interagissant dans la conception.

Cependant, l'utilisation de SysML manque encore de maturité sur un certain nombre de points. En premier lieu, à notre connaissance il n'existe aucune méthode associée à l'utilisation de ce langage. En effet, les méthodes existantes en ingénierie système sont basées sur des approches documentaires. Les approches diagrammatiques telles que celle prônée par SysML ne sont pas compatibles avec. Les méthodes orientées objet qu'on utilise avec UML, même si elles sont diagrammatiques ne sont pas non plus satisfaisantes puisque bâties sur le paradigme objet qui n'est pas central à SysML comme il l'était en UML. Cela induit à ce jour un manque important dans l'utilisation d'une démarche centrée sur SysML et pensée pour ce type de modélisation. D'autre part, notre expérimentation a montré que SysML était un langage surtout utile dans les phases d'analyse et de conception préliminaire. En effet, SysML étant basé sur une utilisation par raffinements successifs, il arrive un moment à partir duquel la modélisation multi-domaines (mélange sur les mêmes diagrammes de concepts hardwares, softwares, hydrauliques...) de SysML n'est plus satisfaisante. Il convient alors d'utiliser des langages spécifiques au domaine visé, *i.e.* UML pour les parties logicielles du système, ce qui pose un problème de transmission des informations modélisées en SysML et qu'il faudrait réutiliser ou affiner dans un autre langage. De plus, un des atouts majeurs de SysML tient dans sa capacité à tracer les éléments du modèle entre eux ainsi qu'à maintenir une traçabilité tout au long du cycle de modélisation. Le passage à un autre langage entraîne donc une perte de cette traçabilité.

#### **4. Pourquoi la définition d'un profil spécifique ?**

Comme nous l'avons vu dans la section précédente, SysML est un langage idéal pour servir de langage de développement à partir du langage d'expression RELAX présenté au § 2. Il nous reste maintenant à faire le lien entre ces deux langages. Nous nous sommes fixés comme objectif de démontrer l'intérêt d'une telle combinaison en développant dans un premier temps un profil SysML pour RELAX. L'idéal serait bien sûr de développer un DSL (*Domain Specific Language*), complètement dédié aux systèmes auto-adaptatif. Cet objectif nous intéressera dans un deuxième temps et nos

avons choisi dans un premier temps de commencer de façon pragmatique par la définition d'un profil UML, en nous basant sur nos premières expériences dans ce domaine (Belloir 2004).

Le profil SPAS (SysML Profile for Adaptive Systems) solutionnera un certain nombre de limites du langage RELAX :

- il permettra de fournir un environnement de développement et un support graphique à l'expression des contraintes RELAX ;
- il permettra de disposer des outils SysML qui commencent à maintenant être utilisés en milieu industriels ;
- il permettra de vérifier la consistance des opérateurs définis par leur implémentation en contraintes OCL ;
- il permettra de faire la transition entre l'expression des besoins et l'analyse de manière efficace et avec un minimum de perte d'information (*seamless*).

## 5. Déroulement prévu du projet et résultats attendus

Le développement de ce profil n'est pas un aboutissement en soi. Il s'inscrit dans le cadre d'une thèse qui vise à l'élaboration d'un ensemble (de la spécification des besoins à l'implémentation de connecteurs dédiés) d'outils logiciels pour le développement de systèmes auto-adaptatifs. Le schéma de la Fig. 1 résume l'approche. L'étape 1 sera matérialisée par un ensemble d'heuristiques et de bonnes pratiques issues des experts en ingénierie des besoins (*requirements engineering*). Notre collaboration aux activités de recherche sur les « *Early Aspects* » (Moreira *et al*, 2005) est en relation avec cette étape. L'étape 2, qui consiste à appliquer le processus RELAX défini dans (Whittle *et al*, 2009) est nécessairement manuelle car elle demande une bonne connaissance des besoins et implique la participation des clients. L'étape 3 nécessite le développement d'un profil SysML qui supporte l'expression et la prise en compte des éléments de RELAX directement dans les outils SysML. Nous explorons actuellement l'utilisation d'outils comme xText<sup>1</sup> afin de générer le métamodèle de RELAX pour tenter de permettre la génération automatique de modèles SPAS/SysML à partir des spécifications RELAX. L'étape 4 fait le lien avec les travaux que nous menons sur la définition d'« associations sur étagères » (Bruel *et al*, 2009) où la sémantique enrichie des associations de compositions proposées permettra de modéliser les propriétés issues des phases précédentes (dépendances de vies, partageabilités des composants, etc.) pour bénéficier de la génération des connecteurs associés au niveau de l'architecture. L'idée est de faire (en commençant par UML) la taxonomie des types d'association. D'abord les intra modèle (relations, compositions, etc. dans un diagramme de classe par exemple), puis inter modèle (comme les dépendances), avec dans ce dernier cas les mono niveaux de méta modélisation (M1) ou les multi niveaux (M1/M2, du type *conforms to* par exemple). Nous souhaitons pouvoir fournir un guide de type arbre de décision, pour

---

<sup>1</sup> <http://wiki.eclipse.org/Xtext>

sélectionner sur étagère l'association la plus adaptée. Cette dernière possédant des propriétés génériques, permettra de générer, en fonction des éléments auxquels elle est reliée, le code des connecteurs en fonction de l'architecture cible de développement.

## 6. Conclusion et perspectives

Les travaux présentés dans cet article sont des travaux « en cours ». Nous avons souhaité néanmoins communiquer sur le sujet pour montrer le caractère pluridisciplinaire et appliqué des recherches qui peuvent s'effectuer dans un cadre IUT. Le développement concret d'une plate-forme expérimentale permettra de démontrer l'intérêt de l'approche décrite et de soulever encore d'autres questions à la fois pratiques mais aussi de fond sur le développement hardware/software à base d'ingénierie des modèles.

## Bibliographie

Belloir N., « Composition logicielle basée sur la relation Tout-Partie ». Thèse de doctorat, Université de Pau et des Pays de l'Adour, Décembre 2004.

Belloir N., Bruel J.-M., Hoang N., and Pham C., « Utilisation de SysML pour la modélisation des réseaux de capteurs sans fil ». Actes de la conférence Langages et Modèles à Objets (LMO'08), Montréal, Canada, 2-7 March 2008. Pages 171-186. RNTI.

Bruel J.-M., Campo E. et Wei A., « PLEXUDOM : Plate-forme Expérimentale dédiée à la Surveillance et au Maintien à Domicile ». Projet en cours de soumission dans le cadre du Programme Ponctuel de Recherche PPR'2009, Université de Toulouse 2 – Le Mirail, 2009.

Bruel J.-M., Ahmad M., Beugnard A., « From Composition to Connectors », soumis à la session Work in Progress de l'école d'été MDD4DRES, 2009.

Dubois, H., « Gestion des exigences de sûreté de fonctionnement dans une approche IDM ». Revue Génie Logiciel, n°85, juin 2008.

Moon S., Lee K., and Lee D., « *Fuzzy branching temporal logic. Systems, Man, and Cybernetics, Part B* », *IEEE Transactions*, 34(2):1045–1055, April 2004.

Moreira A., Araujo J., and Rashid A., "A Concern-Oriented Requirements Engineering Model", Proceedings of International Conference on Advanced Information Systems Engineering (CAiSE), 2005, Springer-Verlag Lecture Notes in Computer Science 3520, pp. 293-308.

OMG. SysML™ v1.1, published by the OMG as an *Available Specification* formal/2008-11-01, december, 2008. Cf. <http://www.omg.sysml.org/>

Whittle J., Sawyer P., Bencomo N., Cheng B.H.C., and Bruel J.-M., « *RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive Systems* ». Soumis à la conférence *Requirements Engineering*, 2009.