



HAL
open science

La méthode de développement en X, un autre point de vue sur le cycle de vie.

M. Tahan, Amara Touil, Jean Vareille, Philippe Le Parc

► To cite this version:

M. Tahan, Amara Touil, Jean Vareille, Philippe Le Parc. La méthode de développement en X, un autre point de vue sur le cycle de vie.. 20ème Congrès Français de Mécanique, Aug 2011, Besançon., France. hal-00662752

HAL Id: hal-00662752

<https://hal.univ-brest.fr/hal-00662752>

Submitted on 28 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

La méthode de développement en X, un autre point de vue sur le cycle de vie.

M. Tahan, A. Touil, J. Vareille, P. Le Parc

Lisyc Laboratoire Informatique des Systèmes Complexes (EA 3883)
UEB, UBO, 20 avenue Le Gorgeu 29200 BREST
mehdi.tahan@gmail.com, amara.touil@univ-brest.fr,
jean.vareille@univ-brest.fr, philippe.le-parc@univ-brest.fr

Résumé :

Le développement en X est une méthodologie de conception récente qui jette un regard différent sur l'élaboration des systèmes techniques, tout en intégrant des méthodes déjà éprouvées. Elle s'appuie sur deux idées fondatrices : l'omniprésence de l'environnement et le fait que le comportement du système résulte de sa définition. Elle vise à diminuer le temps de développement, à prendre en compte l'environnement dès le début de la conception jusqu'après la valorisation en fin de vie, à réduire l'impact environnemental, et à détecter les problèmes de conception plus tôt grâce à des simulations en environnement virtuel réaliste.

Abstract:

The X development method is a newly developed one, which provides a different perspective on the construction of technical systems. It relies on two founding ideas: a ubiquitous environment and a simulation of the behaviour induced by the definition of the system itself. The main advantages targeted are a shortened development time, a better integration into the environment, a consideration of the whole lifecycle from the design until the end-of-life recovery, and a detection of design problems earlier.

Mots clefs : méthode de développement, gestion du cycle de vie, CAO-PLM, éco-conception, IDM.

1 Introduction

Le développement des systèmes techniques complexes fait intervenir différentes compétences dont la coopération est délicate à maîtriser. Les systèmes mécatroniques en sont un bon exemple. Le découpage de leur développement par métier permet d'en diminuer la difficulté, toutefois chaque branche dispose de méthodes de travail différentes, et les échanges entre elles sont rapidement difficiles.

Pour réussir il est important de connaître les contextes d'utilisation du système développé, mais il est souvent difficile d'être exhaustif en la matière. Une fois réalisé, le système est confronté à un environnement sans certitude qu'il corresponde à ce qui a été envisagé durant sa conception.

Nombre d'individus, parmi lesquels des industriels, agissent encore comme si l'environnement était infini. Cette attitude ne peut perdurer car les ressources en matières premières sont finies, par exemple le coltan, la rhytine. Il est nécessaire de tenir compte de telles contraintes dès le début de la conception d'un système.

Dans un premier temps nous ferons un rapide rappel de méthodologies existantes dont la méthode de développement en X s'inspire, puis nous la présenterons.

2 Quelques méthodes de développement de produit éprouvées et disponibles

Cette partie est moins un état de l'art qu'une succession de courtes présentations de méthodes éprouvées, avec pour dessein d'en retirer les notions utiles à la synthèse d'une démarche applicable à la totalité du cycle de vie d'un système. Les méthodes évoquées ci-après sont, pour les unes, des enchaînements de tâches allant de l'expression du besoin jusqu'à la livraison du système (qui, où, quand, quoi), pour les autres, des démarches de conception, de la modélisation du problème à la définition du système.

- Le développement en cascade [1] est utilisé depuis fort longtemps. La figure 1 en montre une représentation schématique, qui part de l'idée et se termine par la réalisation concrète. L'axe horizontal symbolise le temps, l'axe vertical représente le degré d'avancement du projet. Deux notions fortes ressortent : une progression séquentielle depuis l'idée initiale jusqu'à la réalisation concrète, en passant par une conception abstraite, la production du système est entreprise à partir d'une définition complète.
- Le cycle en V [2] est utilisé depuis de nombreuses années dans la conception logicielle et très largement diffusé, voir figure 2. La première moitié du V ressemble à la première partie de la cascade, une analyse de conception abstraite précède l'intégration. Les flèches bidirectionnelles symbolisent des vérifications. Une amélioration pour le management de projets, « Dual Vee » [2], permet le parallélisme.

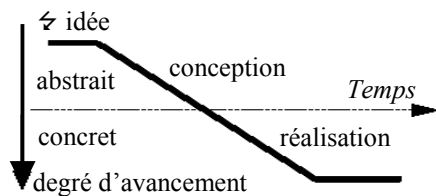


FIG. 1– Le développement en cascade

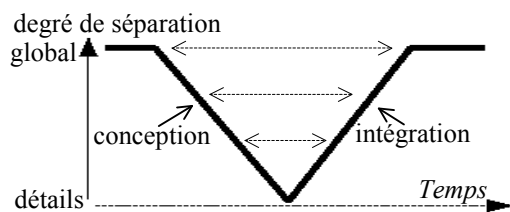


FIG. 2– Le cycle en V

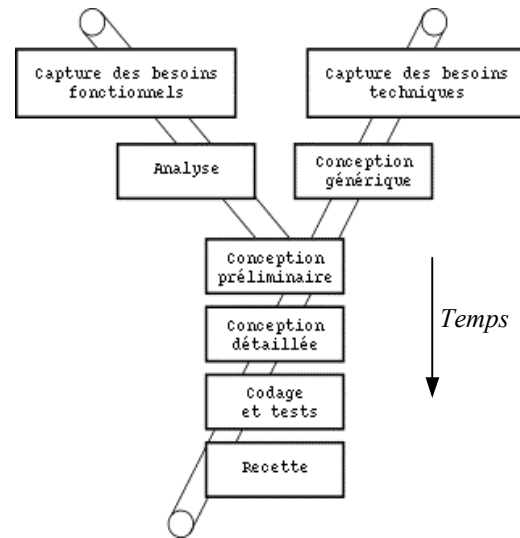


FIG. 3 – La méthode 2TUP (image due à VT Virtual)

- L'Idef0 [3] est une méthode de modélisation par schéma basée sur la méthode SADT. Elle permet de modéliser et analyser des systèmes matériels et des logiciels (immatériels).
- La méthode 2TUP [4] est une méthode de développement en Y, dans laquelle le début de la conception est séparé en deux processus menés parallèlement, voir figure 3. Elle corrobore l'importance des choix matériels « au plus tôt ».
- La méthode en spirale [5] et les méthodes agiles [6] s'appuient sur des prototypages rapides et des itérations à cycle court. Elles proposent aussi un dialogue continu avec le client et l'utilisateur final afin de produire un résultat adapté aux conditions d'utilisation, le plus proche possible des attentes.
- La méthode TRIZ [7] pour la conception innovante met en œuvre 40 principes parmi lesquels la segmentation (« diviser un objet en parties indépendantes ») et l'extraction (« séparer de l'objet une partie <une propriété> "perturbatrice" ou extraire une partie <une propriété> nécessaire »). Elle poursuit la recherche de la solution innovante idéale en résolvant des contradictions. L'analyse des 40 principes permet de les classer en plusieurs catégories selon qu'ils concernent plutôt la géométrie, les matériaux, le comportement attendu, l'environnement, ou des interactions entre ces notions. La méthode TRIZ a été transcrite sous la forme de l'algorithme séquentiel ARIZ.
- L'ontologie de conception « Function Behavior Structure » (FBS) [8] propose de comparer le comportement attendu {Be} défini dans le cahier des charges et le comportement de la structure {Bs} de la solution envisagée. Un processus itératif permet de converger vers un comportement {Bs} acceptable car voisin de {Be}. Le « comportement obtenu » du système technique est évalué avant même d'en avoir une définition complète ou un prototype virtuel, ce qui représente une difficulté majeure. L'analyse se termine par la production d'une documentation permettant la conception détaillée puis la réalisation du système. Cette méthode séquentielle se situe clairement lors de la phase de préconception.
- La méthode des indices de performance d'Ashby [9] est dédiée au choix des matériaux. Elle est basée sur l'hypothèse de la séparabilité des problèmes de conception. Soient : les contraintes de conception {F}, les propriétés géométriques {G} et les propriétés des matériaux {M(p)}, les performances {P} du

système sont évaluées grâce au système d'expressions (1) dans lequel le symbole « := » traduit le fait qu'elles résultent des choix de conception.

$$\{P\} := f \{F, G, M(p)\} \quad (1)$$

Si (1) est séparable, elle conduit à l'expression (2), où les f_i peuvent être optimisées indépendamment.

$$\{P\} := f_1 \{F\} f_2 \{G\} f_3 \{M(p)\} \quad (2)$$

Les performances $\{Ps\}$ de la structure de la solution peuvent être déduites de la formulation (3).

$$\{Ps\} := f_{1a} \{Env, Bs\} f_{1b} \{F'\} f_2 \{G\} f_3 \{M(p)\} \quad (3)$$

L'environnement $\{Env\}$ y apparaît en tant que contraintes, le comportement de la structure y est noté $\{Bs\}$. La fonction $f_{1b}\{F'\}$ n'étant pas nécessaire, a été écrite en italique, car les contraintes $\{F'\}$ qui sont en argument, sont celles qui ne sont dues à l'environnement, ni au comportement de la structure.

- La méthode de conception basée sur l'environnement (Environment Based Design) [10] est axiomatique. Elle part de l'idée que le système a pour but de modifier une interaction à l'intérieur de l'environnement, que ce dernier est présent avant la conception et subsistera après le démantèlement du système, dont tous les éléments sont extraits. La méthode a l'ambition de traiter les problèmes de conception en imitant la logique des concepteurs. Un modèle formel de celle-ci est développé, en suivant une démarche itérative dans laquelle l'environnement est lui-même modifié au fur et à mesure que progresse le développement du système.

La méthode de développement en X [11] reprend l'omniprésence de l'environnement. Elle considère la fin du cycle de vie comme une contrainte de conception : lorsque le système est devenu inutile, il doit être stoppé, ses éléments rejetés dans l'environnement ou recyclés. Elle reprend la dualité abstrait/concret et le parallélisme des tâches dès la conception. Elle intègre aussi une simulation réaliste du comportement d'un système dans l'environnement afin d'en évaluer l'impact sur celui-ci et d'aider au dialogue avec le client.

3 La méthode de développement en X

Dans cette section nous allons présenter cette méthode et préciser desquelles elle hérite.

L'environnement existe avant la conception du système, continue d'exister pendant sa création, son utilisation, lors de son démantèlement, et finalement après sa valorisation ou destruction. L'environnement et tout ce qui est matériel au sens des éléments physiques, chimiques et biologiques est placé au fondement de la représentation schématique de la méthode, voir figure 4. L'espace du schéma est séparé en deux zones : l'une correspond à tout ce qui est accompli au contact de l'environnement matériel avec effet tangible, l'autre correspond à tout ce qui est immatériel, les idées, les informations, les données, les savoirs, les savoir-faire, etc. dont l'usage ou la transformation n'ont pas d'effet direct sur l'environnement matériel. La séparation entre les deux zones est symbolisée par une flèche horizontale qui porte l'axe du temps, car le temps est un milieu immatériel mais ayant une dimension physique, auquel est associé un temps logique, immatériel et sans dimension.

Le développement démarre par une idée, qui précède l'expression du cahier des charges. Une des difficultés majeures du développement se situe au départ car le cahier des charges est pauvre en données, ce qui laisse une très grande liberté de choix. L'analyse initiale enclenche deux processus parallèles, l'un immatériel de production d'informations quant à l'architecture du système, à ses performances et comportements attendus, l'autre au contact du matériel est un processus de sélection de ses éléments matériels. Ce second processus commence par la délimitation d'un « supersystème » composé du système inconnu à définir, et de tout ce avec quoi il interagit dans l'environnement, du début de sa réalisation jusqu'à la fin de son démantèlement. Le « supersystème » a comme particularités d'être généralement non isolé, hors équilibre, variable, voire ouvert. Son identification est du ressort de l'ingénieur aidé par des experts. La méthode du professeur Ashby permet de choisir dès la phase d'analyse d'un système, les familles de matériaux utilisables pour sa construction, alors que son architecture et les géométries de ses composants ne sont pas définies. Pour étudier les contradictions la méthode Triz emploie une analyse substance-field. Lors de l'analyse préalable à la conception, les données et contraintes connues $\{F\}$ peuvent être classées selon leurs dimensions physiques, puis faire l'objet d'une analyse dimensionnelle [12] au sein du « supersystème » menant à une formulation abstraite sans dimension du problème qui amorce la conception. Cette dernière a pour but de définir le système technique solution du problème, c'est à dire énumérer la totalité des éléments standard à

intégrer, modéliser précisément les assemblages de pièces, désigner les matériaux constitutifs des pièces $\{M(p)\}$, décrire leurs formes et dimensions géométriques $\{G\}$, et modéliser les logiciels de contrôle commande à employer. Des échanges verticaux d'informations entre les branches du X ne sont pas représentés sur la figure 4, ni les itérations effectuées au cours de certaines tâches.

Après la pré-étude commencent simultanément l'étude dans le demi-espace immatériel et la collecte dans le domaine matériel des éléments, des matériaux et des outillages nécessaires à l'industrialisation (hormis les outils de forme). Le bureau des méthodes de fabrication est ainsi impliqué dès le début du développement. Les deux processus convergent vers le point central du schéma en X, qui correspond à l'instant de synchronisation des tâches. La définition détaillée du système y est alors achevée ; si les éléments, les outillages et les matériaux sont approvisionnés, la production peut commencer.

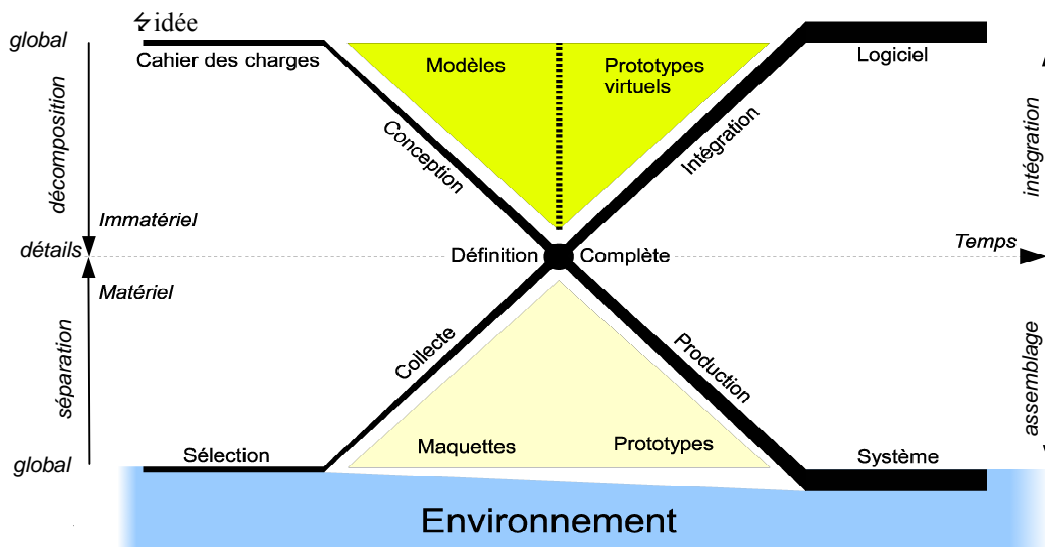


FIG. 4 – Le développement en X

Parallèlement à celle-ci se déroule l'intégration des logiciels qui piloteront le système, la formation des personnes qui l'utiliseront, et l'assemblage progressif de sa partie matérielle et les tests.

À l'issue de ces processus de production et d'intégration le système forme un tout plongé dans l'environnement. Il associe une partie matérielle, à une partie immatérielle composée de données et de logiciels, qui sont mis en œuvre par une partie électronique et informatique embarquée. Après réalisation et intégration, la partie matérielle peut difficilement être modifiée, par contre les données de paramétrage et les logiciels peuvent être mis à jour à tout moment pendant l'utilisation, à condition que la conception ait tenu compte d'en offrir la possibilité.

Le schéma en X peut être compris comme un PERT dont les échéances temporelles sont gouvernées par l'inertie des tâches matérielles. Plus les branches du X s'approchent de la ligne médiane, plus le système est graduellement décomposé ou séparé à gauche du point central, plus elles s'en éloignent, plus il est progressivement intégré et assemblé à droite, jusqu'à constituer un tout.

Le schéma en X permet de retrouver la trace du développement en cascade en suivant la diagonale allant du cahier des charges à gauche, jusqu'au système matériel fini, à droite, en action dans l'environnement. Le cycle en V est situé quant à lui dans la partie immatérielle du schéma, il va du cahier des charges au logiciel en passant par les étapes de conception et d'intégration. Le cycle en Y de 2TUP est également présent dans le X, mais il ne suit qu'une des deux branches situées à droite du point central.

En outre le développement en X introduit deux nouveaux points de vue : le point de vue productique et le point de vue scientifique.

Le point de vue productique prend la forme d'un V inversé (Λ), il est situé entièrement dans la partie matérielle (voir figure 5). Décrivons ce Λ dans l'ordre chronologique. Des ressources disponibles dans l'environnement, qui peuvent être des matières premières mais aussi des composants, des ressources

humaines, de l'énergie, etc. sont sélectionnées très tôt. Ensuite vient la phase ascendante de collecte des ressources matérielles (matériaux bruts ou composants) rassemblées en totalité au point culminant du Λ , qui correspond au moment où la définition du système à réaliser est complète, du moins suffisamment avancée pour lancer la phase de production. La dernière phase descendante est la réalisation matérielle du système. L'intérêt de ce point de vue est d'offrir un panorama d'ensemble sur la matérialité. Il permet de percevoir la totalité des interactions conception du système / environnement et produit / environnement. L'environnement est omniprésent il intervient durant toutes les phases, de la conception, à l'exploitation. Sans un tel point de vue il serait difficile de mesurer l'impact environnemental du système sur l'intégralité de son cycle de vie. Un point de vue analogue est présent dans l'analyse du cycle de vie d'un produit proposée par M. Ashby [13], dans laquelle sont pris en compte le recyclage et la réutilisation en fin de vie.

La représentation du point de vue scientifique, au sens des « Naturwissenschaften » (ce mot allemand n'a pas de traduction en français coïncidant avec ce qu'il signifie) est un schéma en cascade inversé présenté sur la figure 6 : la courbe part du matériel pour aboutir à l'immatériel, en l'occurrence à des modèles et des données. En effet, un scientifique expérimente dans le monde réel différentes théories afin d'en comprendre les mécanismes. Une fois les informations expérimentales collectées il les organise en modèle.



FIG. 5 – Le point de vue productique

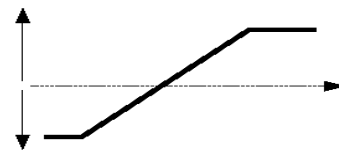


FIG. 6 – Le point de vue scientifique

Chacun de ces modèles contribue à une meilleure compréhension de notre environnement. Selon le système d'expressions (2), il est pensable qu'avec un niveau de connaissance suffisant de l'environnement réel et de la définition matérielle et géométrique du système, son comportement futur au sein de son milieu d'emploi pourrait être simulé de façon réaliste, en utilisant des techniques de réalité virtuelle. Des simulations de prototypes virtuels du système dans des environnements virtuels précèderaient les phases de réalisation.

Le développement de logiciels est un problème granulaire, donc discontinu. Il est résolu dans le domaine immatériel, en utilisant les méthodes et outils de génie logiciel [4], en particulier l'ingénierie dirigée par les modèles. En premier lieu une architecture est choisie, ensuite les composants standard, système d'exploitation, machine, langage, bibliothèques, et enfin les parties non standard sont développées.

Dans le domaine matériel, les dimensions géométriques et les formes des pièces $\{G\}$ ne font intervenir que des longueurs et des angles qui ne peuvent en soi permettre de faire face aux efforts, aux pressions, et aux températures, etc., ce sont les matériaux et leurs caractéristiques $\{M(p)\}$ qui y répondent (directement ou via des champs). Mais celles-ci ne sont pas continues, par exemple si nous reportons sur un graphe le module de Young des matériaux en fonction de leur masse volumique nous voyons immédiatement que les nuances forment un chapelet de petites zones qui ne couvrent pas le plan. Il est évident alors qu'il vaut mieux commencer par choisir les matériaux capables, ensuite déterminer les formes des pièces et ajuster leurs dimensions, plutôt que de procéder dans l'ordre inverse, car les dimensions et les angles sont des variables continues qui couvrent complètement leurs domaines de définition. Les derniers principes ainsi que les tous premiers de la méthode TRIZ, l'analyse substance-fied, et la méthode des indices de performance peuvent donc être utilisés en premier, au moment du choix des matériaux constitutifs [11]. L'architecture des liaisons mécaniques entre les pièces est un problème non linéaire et totalement discontinu. Il est à traiter immédiatement après le choix des matériaux en employant les principes ad hoc de la méthode TRIZ parmi les 30 premiers. Le problème des formes et des dimensions des pièces, souvent linéarisables, peut être résolu en dernier par les méthodes habituelles de dimensionnement, car il s'agit de problèmes continus sans restriction quant aux espaces de définition des dernières variables restantes, toutes de nature géométrique.

Afin de modéliser l'ensemble nous utilisons l'ingénierie dirigée par les modèles (IDM).

4 Utilisation de l'ingénierie dirigée par les modèles

L'IDM [14][15] et des langages du type SysML [16] ou des DSML [17] permettent de modéliser à la fois les systèmes développés ainsi que les différentes tâches de la méthode. Au préalable un métamodèle de domaine [18] est établi, duquel des modèles d'instances sont dérivés pour chaque cahier des charges. Les conversions automatiques entre métamodèle et modèles devraient permettre d'augmenter la part de temps consacré par

les concepteurs à l'innovation et à la création, en réduisant les saisies séparées de modèles du même système et leur vérification. Le challenge futur sera de conserver la cohérence d'ensemble du développement dans un monde où chaque branche de métier emploie des ateliers logiciels différents pour la CAO-PLM et les simulations. Nous ne pouvons malheureusement pas développer ici cette partie, publiée par ailleurs [18].

5 Conclusion

Les techniques de conception n'ont cessé de progresser avec les technologies. Chaque domaine possède ses propres méthodes de développement : méthode en cascade, cycle en V, etc. Le développement en X tente d'en réunir les points de vue et de les intégrer dans un schéma permettant d'en paralléliser des tâches. L'omniprésence de l'environnement dans la méthode de développement en X offre la possibilité d'évaluer l'impact environnemental d'un système tout au long de son cycle de vie. Elle permet d'interpréter le transfert progressif de la valeur vers les technologies « immatérielles » de l'information et de la communication.

Pour l'appliquer concrètement nous proposons d'utiliser l'ingénierie dirigée par les modèles. Celle-ci devrait permettre la coopération des modèles élaborés par des équipes de développement de branches de métier variées, qui suivent des démarches différentes.

L'utilisation du développement en X aurait pour conséquence que les systèmes futurs offriraient des services similaires ou supérieurs aux actuels, en étant composés d'une partie matérielle plus réduite supportant une partie immatérielle pouvant croître sans limite autre que la ressource intellectuelle humaine.

References

- [1] Royce W.W., Managing the Development of Large Software Systems, Proceedings of IEEE WESCON 26 (August), 1970, p.1-9.
- [2] Forsberg, Kevin; Harold Mooz, Howard Cotterman, Visualizing Project Management, Third Edition, New York, NY: J. Wiley & Sons, Inc, 2005.
- [3] Idef0, Integration definition for function modeling (Idef0), Draft Federal Information Processing Standards Publication 183, 1993.
- [4] Roques P. et Vallée F., UML2 en action, de l'analyse des besoins à la conception, Eyrolles, 2007.
- [5] Boehm B, A Spiral Model of Software Development and Enhancement, Computer, IEEE, 21(5), 1988, p.61-72.
- [6] Abrahamsson P. et al., Agile Software Development Methods: Review and Analysis, VTT Publications 478, 2002.
- [7] Altshuller G., Triz l'algorithme de résolution de problèmes innovants ARIZ-85-V, 1985, Altshuller Foundation.
- [8] Gero J.S., Design prototypes: a knowledge representation schema for design, AI Magazine 11(4), 1990, p.26-36.
- [9] Ashby M.F., Materials selection in conceptual design, Mater Sci Tech 5 (6), 1989, p. 517-525.
- [10] Zeng, Y., Environment-Based Formulation of Design Problem, Journal of Integrated Design and Process Science 8(4), 2004, pp. 45-63.
- [11] Tahan M. et al., The X-development method, Crecos seminary, Aalto University, Helsinki, 2010.
- [12] Coatanéa E., Vareille J., et Yannou B., Analysis of the dimensional analysis and measurement theory - Their influence on the comparison and evaluation process during early design phases, CPI'2007, Rabat, 2007.
- [13] Ashby M., Sustainable materials: Inspiring students to ask the right questions, Web Seminar, 2010, p.7.
- [14] Schmidt D.C., Model-Driven Engineering, IEEE Computer 39 (2), 2006.
- [15] AFIS, Pourquoi l'Ingénierie Système ?, Association française d'Ingénierie Système, 2005.
- [16] SysML, Systems Modeling Language (OMG SysML™), 2010.
- [17] Modelica, Modelica® - A Unified Object-Oriented Language for Physical Systems Modeling Language Specification, 2010.
- [18] Touil A, Vareille J., et Le Parc P., Modeling and Analysing Ubiquitous Systems Using MDE Approach, UBICOMM 2010, Florence, 2010.