

# Ingénierie des Systèmes d'Information Documentaires : Test et validation d'architectures d'informations techniques

Michel Lanque et Philippe Larvet

**Résumé:** Cet article présente un processus de test performant d'un système d'information technique modélisé, ainsi qu'une démarche de vérification et validation d'un système documentaire considéré comme un système logiciel. L'approche proposée permet une validation, effective, technique et fonctionnelle, d'un système directement réalisée à partir de sa documentation technique – par exemple, sa spécification ou ses cas d'utilisation considérés comme données d'entrée.

**Mots clés :** documentation technique, systèmes complexes, systèmes d'information, test de système, validation de systèmes

## 1. LA DOCUMENTATION DE SYSTÈMES COMPLEXES

### 1.1 Contexte global

La conception et le développement de produits industriels induisent une complexité de plus en plus grande dans la gestion de l'information technique, particulièrement si les produits sont des applications basées sur les technologies du web. La raison de cette complexité accrue tient principalement aux interactions des composants internes du système ainsi que, bien sûr, aux relations d'échange du système avec d'autres systèmes, mais elle provient aussi du fait que, de plus en plus, les systèmes doivent couvrir et gérer des informations techniques variées, multiformes et ce dans des domaines parfois fort différents.

Par ce fait même, ce volume croissant d'informations exige davantage de contrôle au niveau de la fiabilité, de la sécurisation, de la qualité du traitement et de l'exploitation de systèmes développés avec un engagement de performance, pouvant être facilement accessibles à des opérateurs, experts et utilisateurs finals.

Comme c'est le cas dans le cadre de produits logiciels intégrant des applications basées sur les technologies du web, l'information manipulée est dans le même temps unidirectionnelle et autonome dans son propre contexte, et multidirectionnelle si nous considérons sa capacité d'association avec d'autres informations du même contexte.

Le volume d'information augmente également à cause de la nécessité de prendre en compte l'existant : les données nouvelles, liées par exemple à une évolution fonctionnelle d'un système donné, doivent impérativement être intégrées avec les fonctionnalités déjà opérationnelles du système existant et doivent être implémentées selon la politique de développement de l'industriel créateur et distributeur du produit.

En conséquence, une application logicielle, considérée comme un produit industriel et prenant en compte différents niveaux de fonctionnalités opérationnelles – par exemple des opérations locales conduites par un opérateur, de la gestion et du « monitoring », de la gestion de nœuds de réseau matériels ou des opérations de type « d'un bout à l'autre » – exige, pour être spécifiée de manière complète et cohérente, la mise en œuvre et la gestion maîtrisée de sous-ensembles complexes d'informations techniques.

### 1.2 Performance du produit et information utilisateur

La plus grande part des produits technologiques industriels actuels – du moins ceux qui nous intéressent dans le cadre du présent article – consistent en produits logiciels comprenant une interface utilisateur graphique (GUI, Graphical User Interface), élément clé par lequel passent le lancement et le contrôle de toutes les tâches utilisateur : installation, administration, opérations et maintenance. Ceci veut dire que le produit peut être vu essentiellement comme un système d'informations permettant la gestion de tâches utilisateur via une interface graphique qui expose et embarque des ensembles d'informations techniques.

En conséquence, la performance du produit dépend essentiellement du niveau de qualité de l'information fournie à l'utilisateur final. L'information technique, sa modélisation et son utilisation dans le but d'aider à la validation du système représente le sujet principal sur lequel porte le présent article.

### 1.3 Échange de données

Comme il vient d'être dit, le niveau de qualité de l'information fournie à l'utilisateur participe pour une part importante à la performance du produit. Dans les systèmes logiciels, l'élément-clé de fourniture d'information à l'utilisateur est l'interface graphique, ce qui nous permet de dire que pour la plupart des applications logicielles, le produit *est* l'interface graphique.

Ceci signifie que la qualité de l'information fonctionnelle affichée à l'utilisateur ou référençant des composants d'information (documentation, aide en ligne, etc.) est une part importante de la performance du produit, qu'il s'agisse de la qualité des données elles-mêmes, de la compréhensibilité des actions, de l'interprétation des messages, du temps de traitement des données en accord avec les résultats obtenus, des profils de compétence des utilisateurs relativement à l'information fournie, etc.

Par le biais de l'interface graphique, la priorité de l'utilisateur final concerne principalement les tâches opérationnelles et de maintenance, à travers la nécessité d'un accès immédiat et efficace à la bonne information prête à être utilisée. L'utilisateur doit également être capable d'identifier l'information qui requiert des tâches spécifiques : actions d'urgence, maintenance corrective, monitoring d'actions préventives, etc.

Ces points se réfèrent au contenu technique de la documentation et/ou de l'information embarquée dans l'application logicielle, telles que les aides en ligne et le système d'informations techniques affichées sur l'interface graphique. Or, il se trouve que le contenu de cette documentation peut être défini à partir de la spécification technique du système. La manière dont cette information est gérée et fournie à l'utilisateur final, afin de l'aider à réaliser ses tâches opérationnelles, a un impact direct sur la bonne utilisation et la performance de l'application.

### 1.4 Spécifications techniques et documentation produit

L'information technique est la référence pour n'importe quelle équipe R&D qui doit créer et mettre à jour de l'information dans le cadre du développement de nouvelles fonctionnalités d'un système. Les informations nouvelles sont construites à partir des informations existantes, informations qui sont principalement décrites dans la documentation technique du système. Cette dernière est développée à destination de plusieurs catégories d'utilisateurs : équipes R&D, pour le développement et le test du produit ; rédacteurs techniques, pour la production de la documentation client et de la documentation utilisateur ; support technique, pour la ligne directe d'assistance et les équipes d'installation et de maintenance ; équipes de formation, etc.

Compte tenu d'une masse d'informations techniques de plus en plus abondante et complexe au sein des entreprises créatrices et distributrices de produits industriels, et de l'ouverture du contenu de cette documentation, facilement accessible par tous les acteurs concernés, il se trouve que les exigences utilisateur concernant le fonctionnement du produit dépassent largement le contexte de la seule documentation. La nécessité d'accéder à une information pertinente et la cohérence de cette information induisent une exigence de contrôle et de gestion de la totalité du contenu de la documentation.

Par voie de conséquence, une augmentation du volume d'information technique nécessaire à la spécification, à la réalisation et à la gestion de systèmes complexes requiert davantage d'exigences de contrôle en matière de fiabilité, de sécurité et de qualité de traitement de l'information. Ces exigences sont directement liées à la performance des systèmes livrés aux opérateurs, experts et utilisateurs finals – et donc, concernent la fiabilité et la qualité de l'information technique.

### 1.5 Objectif visé : répondre aux exigences R&D et utilisateurs

Une question pertinente peut donc être posée : Comment vérifier la fiabilité et la qualité de l'information technique et garantir la cohérence de son contenu, exigences légitimes des équipes R&D dans le cadre de leur travail de conception et de réalisation, comme des utilisateurs finals dans le cadre de leurs tâches opérationnelles ?

L'approche que nous proposons pour répondre à cette exigence est de *tester* l'information technique comme s'il s'agissait d'un ensemble de composants logiciels. Cette démarche de test permet de cibler l'information pertinente et d'aider à sa mise à jour tout en conservant sa cohérence technique en accord avec la documentation existante.

### 1.6 Différentes approches de test documentaire [1]

Les moyens de contrôler et de valider l'information technique peuvent reposer sur plusieurs techniques différentes. Les cas d'utilisation provenant des spécifications techniques peuvent être vérifiés par un utilisateur final via un test manuel d'opérations [2]. Le produit, en tout ou partie, peut être testé, manuellement, à l'aide de la documentation opérateur [3]. Clients et utilisateurs finals peuvent également être impliqués dans le cycle de vie produit (phase de «*first off*») [4]. Enfin, il est possible de produire un système capable de cibler l'information opérationnelle nécessitée par l'utilisateur final, ce système traitant cette information comme le ferait un logiciel exécutant un processus de test.

Cette dernière technique représente le type de solution que nous avons retenue, conformément à nos objectifs qui sont de garantir la cohérence de l'information technique en accord avec l'utilisation du produit.

## 2. CONTEXTE DE L'ÉTUDE

### 2.1 Un « module d'information » est un objet

L'étude que nous avons menée, mise en application dans le cadre de projets pilotes, consiste en une nouvelle approche de modélisation de la documentation technique, aboutissant au test technique de système et basée sur un ensemble rationnel d'informations fonctionnelles. Les éléments d'information composant les documents sont définis par un type d'information (fonctionnelle, opérationnelle, descriptive, etc.) et un ensemble de méta-données (sommaire, pré-conditions, opération, résultat, exceptions, etc.) Ceci permet de les considérer et de les gérer en tant que « modules d'information », ayant un fonctionnement identique aux « objets » tels qu'ils sont manipulés dans un logiciel conçu avec une technique « orientée objet ».

Dans le cadre de notre étude, nous mettons l'accent sur *l'information opérationnelle*. Tout d'abord, cette catégorie d'information est la part la plus importante des données requises par l'utilisateur final (opérateur) pour assurer ses propres tâches : tâches quotidiennes, tâches spécifiques, tâches et procédures d'urgence. Ensuite, toute l'information technique délivrée aux clients s'appuie en priorité sur les opérations utilisateur (cas d'utilisation) et les tâches de maintenance [5]. Enfin, les produits logiciels sont principalement utilisés à travers leur interface graphique qui affiche de l'information opérationnelle produite à partir de systèmes d'informations embarqués.

Avec l'approche proposée, notre objectif est d'améliorer la qualité de l'information opérationnelle produite à partir de la documentation technique, et principalement à partir de la documentation technique spécifique client, de la documentation produit ou utilisateur, de la documentation de formation au produit, du contenu des extranets pour les clients, des aides en ligne et de la documentation embarquée dans le produit ainsi que de la documentation destinée au support technique. L'information technique contenue dans ces documents, que nous appelons « sources de données produit », représente le point de départ d'un processus automatisable permettant de construire un système d'information/produit opérationnel – autrement dit, un ensemble de modules d'information, considérés comme des composants logiciels, constituant un système « qui tourne » et apte à être testé, vérifié et validé. Ceci signifie que l'information opérationnelle est l'élément-clé à tester intégralement afin d'assurer une bonne cohérence fonctionnelle d'utilisation du produit

## 2.2 Les cas d'utilisation comme source principale d'information

Notre principale source d'information est constituée par les cas d'utilisation, qui décrivent les tâches utilisateur représentatives de l'emploi opérationnel du produit/système [6]. Les cas d'utilisation sont exprimés formellement dans la documentation opérateur en tant que « procédures » [7], une procédure étant la description d'une tâche utilisateur, décrite étape par étape pour les actions opérateur.

Afin de répondre aux exigences des utilisateurs du produit concernant la performance opérationnelle du futur système et le haut niveau de complétude de test pour l'ensemble de ses fonctionnalités, avant livraison finale, il est intéressant de disposer d'une solution garantissant le test de fonctionnement des opérations utilisateur sans coûts additionnels pendant toute la phase de développement, et si possible au-delà.

## 3. L'« INGÉNIERIE DE LA DOCUMENTATION »

Le procédé que nous avons développé est basé sur trois idées principales. Tout d'abord, un procédé à source unique (*single sourcing process*) de développement d'information technique permet la réutilisation de l'information et l'application automatisable d'une approche d'analyse sémantique des textes documentaires [8]. Ensuite, chaque information élémentaire est vue comme un *module d'information*, similaire à un objet dans le contexte du développement orienté objet. Nous avons retenu le format XML pour la gestion et l'indexation de ces informations élémentaires [9]. Enfin, les modules d'information sont *gérés comme du code logiciel* afin de construire des combinaisons opératoires, testables et exécutables, telles que par exemple une architecture de l'information pour l'ensemble du produit, un modèle d'information orienté tâche, des scénarios de test système, etc.

Ce procédé nous permet d'assurer la conformité aux standards de développement de l'information [10] et d'utiliser les outils usuels pour la rédaction, l'édition, la vérification et l'accès aux contenus documentaires. De même que nous utilisons XML comme source pour les fichiers de nos modules d'information, l'approche permet d'utiliser d'autres standards de documentation structurée (DITA [11], S1000D [12], etc.), des bases de données pour le stockage des modules (CMS [13]) ainsi que les environnements de rédaction et d'édition classiquement utilisés dans le domaine documentaire (XML Editor, Wiki, MS-Word®, Framemaker, etc.)

### 3.1 Détail du processus

Notre démarche d'« ingénierie de la documentation » (voir Figure 1) s'appuie sur les étapes suivantes :

1. Une **analyse sémantique** des spécifications du produit [14] (ou de la documentation technique équivalente) permet d'identifier les structures de cas d'utilisation, même dans le contenu de documents informels.
2. Un processus dédié d'**extraction des cas d'utilisation** à partir du contenu des documents techniques permet d'exprimer ces cas dans un format structuré.
3. Un module spécifique permet ensuite la production de **procédures formelles** [7] construites à partir du contenu structuré des cas d'utilisation. Ces procédures, contenant les modules d'information utilisateur, sont sauvegardées au format XML (voir Figure 2).
4. **Le stockage des modules** en base de données permet de les indexer à partir de leurs méta-données.
5. Enfin, la construction de l'**architecture fonctionnelle du produit** par assemblage des procédures formelles donne une représentation virtuelle du fonctionnement du futur système ainsi que de la cohérence fonctionnelle des données, et permet de produire un rapport indiquant les modules d'information opérationnels à revoir : données manquantes, redondantes ou dupliquées, traitement de tâches incohérentes, conformité ou non-conformité, etc.

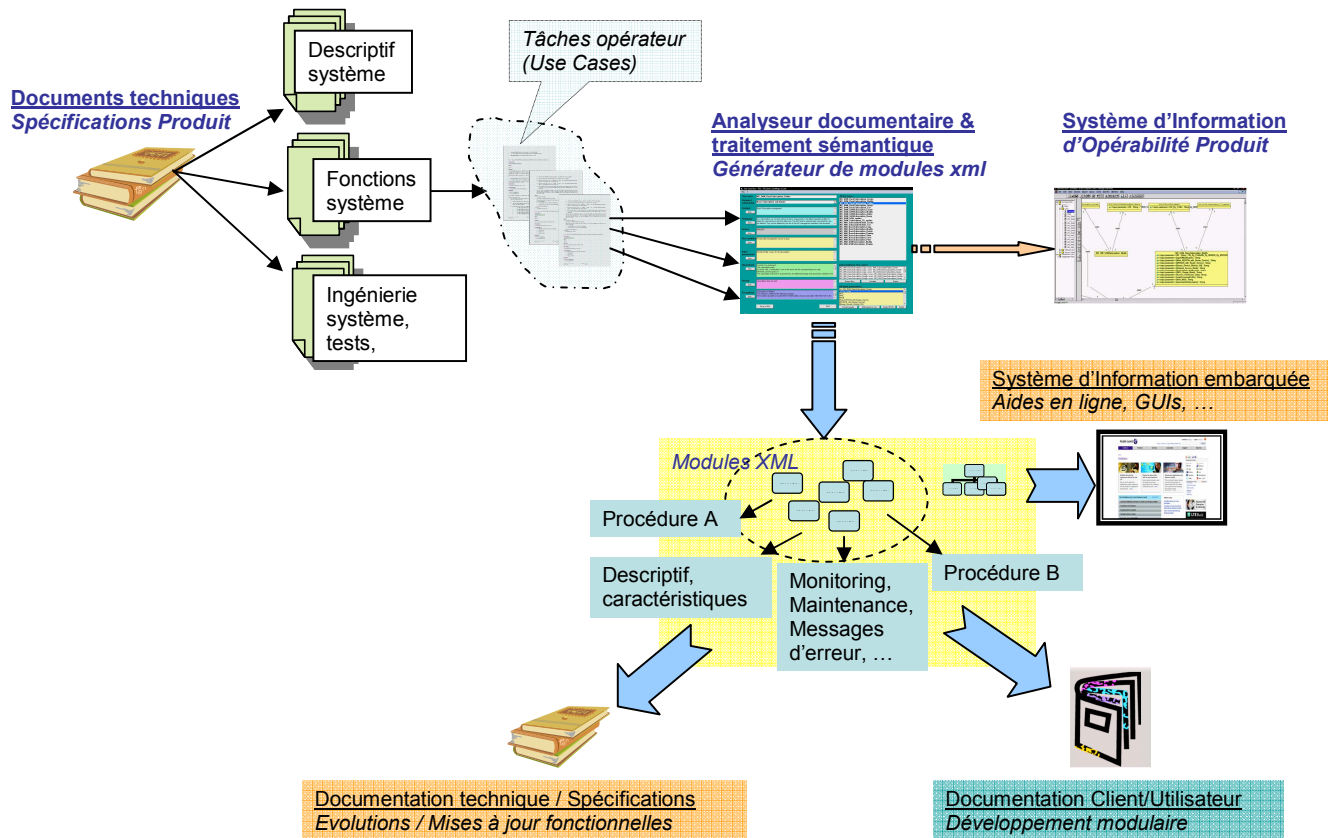


Figure 1 : Ingénierie de la documentation

L'« architecture fonctionnelle » évoquée dans la dernière étape ci-dessus peut être exprimée sous la forme d'un modèle UML, ce qui est un moyen permettant, via les outils appropriés (par exemple Objectteering® [15], Rhapsody [16] ou Modeleurs UML open source [17]), de tester la cohérence des fonctions et des données internes au modèle d'architecture d'informations du système.

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <!-- Generated by ProcedureModeler - P.Larvet 11-28-2009 -->
3  <paramodule lang="EN">
4  <docstatus>
5  <refdoc codif="old">
6  <oldrefdoc>3BL00000000199ACBAPA</oldrefdoc><sheettype></sheettype>
7  </refdoc>
8  <titledoc>
9  <title1>7570_MS_HMI_SubscriptionFolder_Create_PROCEDURE</title1>
10 </titledoc>
11 <ednum>02</ednum>
12 <status statuscode="RL"/>
13 <lastmodifdate><year>2007</year><month>04</month><day>24</day></lastmodifdate>
14
15 <description>
16 <descriptitem>Use type: operation.</descriptitem>
17 <descriptitem>Customer: generic.</descriptitem>
18 <descriptitem>Comment: created for HMI SubscriptionFolder sub-domain.</descriptitem>
19 <descriptitem>Product/Platform release: from G3.1.</descriptitem>
20 <descriptitem>Reason for update: &mdash;.</descriptitem>
21 <descriptitem>Network solution(s): CORE.</descriptitem>
22 <descriptitem>Method(s): GUI.</descriptitem>
23 <descriptitem>External link(s): &mdash;.</descriptitem>
24 </description>
25 <nature naturelibelle="procedure"/>
26 </docstatus>
27
28 <doccontent>
29 <docbody>
30 <hfree>
31 <ie level1="HMI_SubscriptionFolder sub-domain" level2="procedure" level3="MS_HMI_SubscriptionFolder_Create"/>
32 <ie level1="Create" level2="MS_HMI_SubscriptionFolder"/>
33 <ie level1="MS_HMI_SubscriptionFolder" level2="Create"/>

```

Figure 2 : Exemple de module d'information XML

### 3.2 En résumé : concepts principaux de la démarche

L'idée principale de la démarche réside dans le fait que la documentation (c'est-à-dire un ensemble de documents) peut être vue comme du logiciel. Une conséquence intéressante en est que les techniques de développement modulaires ou orientées objet, valables pour l'ingénierie du logiciel, peuvent donc s'appliquer également à l'ingénierie de la documentation.

Comme le logiciel, les documents techniques contiennent des modules d'information réutilisables qui échangent entre eux des données (voir Figure 3).

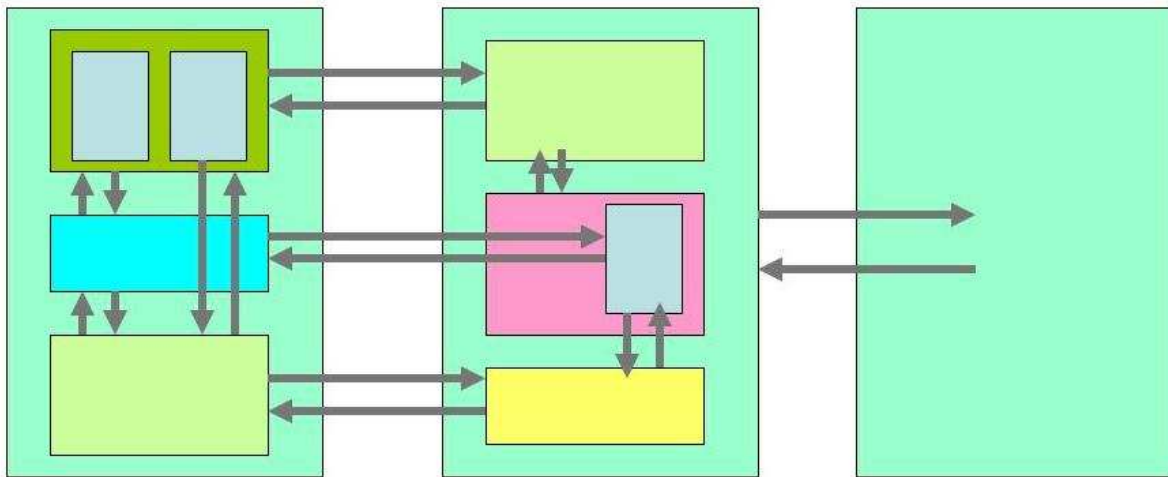


Figure 3 : Les modules d'information, comme les objets logiciels, échangent des données

## 4. MODÉLISATION DE L'INFORMATION

À partir de la description de la démarche proposée, nous pouvons énoncer les principes de la modélisation de l'information, appliquée au test fonctionnel de produits industriels :

1. La documentation technique d'un produit, faite de parcelles d'information, est considérée comme un logiciel.
2. Les modules de ce logiciel sont des « modules d'information ».
3. Chacun de ces modules est considéré comme un composant testable et exécutable.
4. Ces composants peuvent être assemblés au sein d'une « architecture fonctionnelle ».
5. Cette architecture est la représentation virtuelle du futur système/produit et peut être utilisée pour des tests fonctionnels, ainsi que pour de la vérification, validation, et simulation fonctionnelles.

### 4.1 Module d'information fonctionnelle

Dans le contexte de notre approche, un module d'information fonctionnelle est un objet actif muni de caractéristiques précises : pré-conditions, paramètres d'entrée, description du fonctionnement (procédure), résultats attendus, exceptions (dues aux résultats de traitement ou aux valeurs des données d'entrée). À partir de ces éléments, le module agit comme un composant fonctionnel, c'est-à-dire prenant en compte ses données d'entrée, exécutant étape par étape, action par action, sa procédure interne et produisant pour résultat, par exemple, des opérateurs logiques ou des valeurs de données de sortie.

Étant donné que le module d'information a été produit à partir du contenu textuel de la documentation, cela signifie que le fonctionnement du module permet de tester la procédure d'information, ses données et ses opérations. L'architecture fonctionnelle étant un assemblage de modules d'information, simuler cette architecture et vérifier la validité des résultats de simulation apporte une aide importante pour la vérification et la validation du comportement du système virtuel, donc du futur produit.

## 4.2 Gestion des modules

La gestion des modules suppose la mise en application de critères précis pris en compte par le système documentaire interne, avant livraison des modules aux clients et aux utilisateurs. L'accès à cette information peut être réalisé au moyen d'un extranet par exemple.

Dans le contexte de notre étude, nous avons retenu trois critères principaux :

1. les modules d'information sont stockés dans un système de gestion de contenu (CMS [13]) ;
2. après un traitement sémantique, chaque module devient un composant typé, selon la nature des données qu'il contient et son mode d'utilisation ;
3. chaque module a une structure interne bien définie, permettant de faciliter sa gestion, son utilisation et sa réutilisation (voir Figure 4).

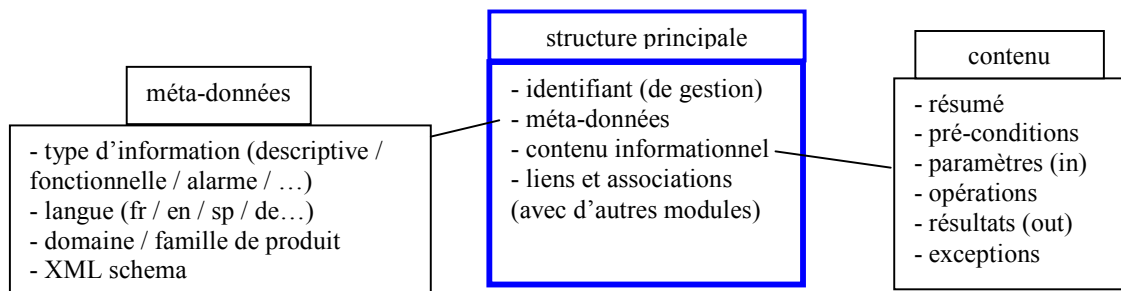


Figure 4 : Patron de description d'un module d'information

Nous pensons que DITA [11], par exemple, est un bon choix pour structurer les modules d'information [18] et faciliter leur gestion dans un CMS.

## 5. TEST DE L'INFORMATION = VÉRIFICATION & VALIDATION D'ARCHITECTURES

Les avantages du « test de l'information » vont bien au-delà de la seule vérification de conformité de la documentation. Comme nous venons de le voir, une architecture fonctionnelle peut être construite par assemblage d'un ensemble de modules d'information directement extraits du contenu de la documentation technique (voir Figure 5), cette documentation consistant par exemple dans la spécification du futur produit.

Comme suite logique de notre étude, nous nous proposons de mettre en place un processus automatique afin de simuler cette architecture fonctionnelle, réalisant de cette manière une sorte de mise en marche fonctionnelle complète du produit virtuel. Ce processus de simulation de « test de l'information » ouvre de larges possibilités pour une véritable vérification et validation d'architecture.

Les possibilités de ce processus de simulation sont nombreuses :

- validation de procédures opérationnelles ;
- vérification de la cohérence fonctionnelle et de la non-régression, notamment grâce au test des évolutions du produit (nouvelles fonctionnalités) ;
- validation technique du contenu des documents avant leur diffusion aux équipes R&D (pour les phases suivantes de conception et développement) et aux clients ;
- validation opérationnelle des plans de test et d'intégration ;
- tests d'interopérabilité (cohérence des liens entre sous-systèmes), autrement dit validation automatique d'architectures modulaires d'informations fonctionnelles ;
- tests de fiabilité automatisés : validation des documents à destination des utilisateurs finals (installation, administration, opération, maintenance) en accord avec le cycle de vie du produit ;
- simulation fonctionnelle, notamment dans le contexte d'une nouvelle architecture modulaire, dans le but de préparer et aider la conception d'un nouveau produit, conception facilitée grâce à la réutilisabilité des modules d'information.



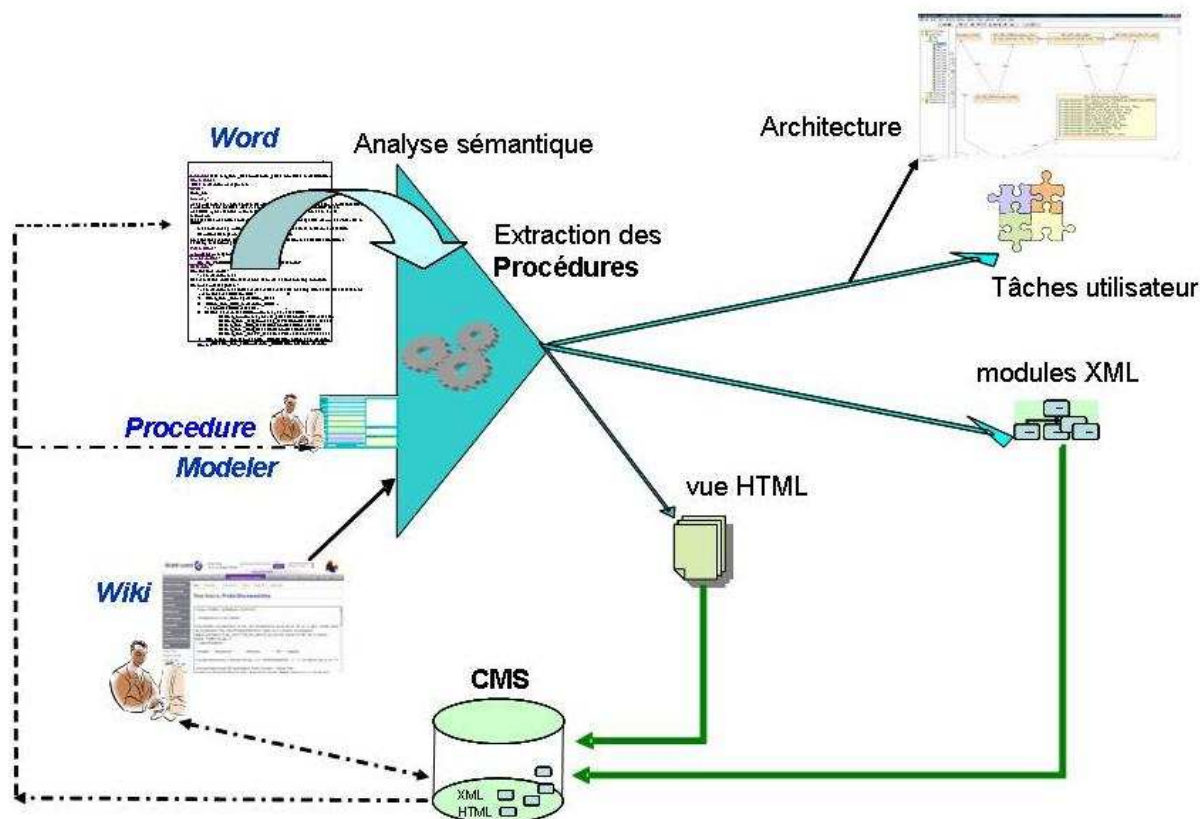


Figure 5 : Construction d'une architecture fonctionnelle à partir de modules d'information extraits du contenu de documents techniques

## 6. CONCLUSION ET PERSPECTIVES

Dans le contexte de la documentation technique, plus précisément de « systèmes d'informations utilisateur extraits de cas d'utilisation », définis à partir des spécifications d'un produit industriel, nous avons décrit un processus automatisable permettant de transformer des éléments d'information technique en un ensemble efficace de modules d'information aptes à être assemblés et exécutés comme une application logicielle.

Ce processus peut facilement être implémenté dans le cadre d'une démarche R&D avec d'intéressantes possibilités : tout d'abord, l'information technique et fonctionnelle devient un ensemble de modules testables et exécutables ; et ensuite, ce point de vue permet une optimisation immédiate de l'opérabilité des modules, grâce à leur réutilisabilité et en accord avec les contextes fonctionnels : anticipation des évolutions de produits, test de systèmes, V&V, tests de fiabilité et de cohérence, simulation de comportement de système.

Nous sommes actuellement en train d'implémenter ce processus dans un outil, afin de le rendre plus concret et de proposer une aide efficace pour la V&V de systèmes, directement dérivée de la documentation technique du futur produit.

## Notes et références bibliographiques

[1] IEEE 829-1998, également appelé Standard 829 pour la documentation de test logiciel (829 Standard for Software Test Documentation), est un standard de l'IEEE qui spécifie la forme que doivent prendre les documents dans le contexte des huit étapes définies par l'IEEE pour le test logiciel, chaque étape étant à même de produire son propre type séparé de document. Le standard spécifie le format de ces documents mais ne stipule pas si les documents doivent tous être produits, et ne donne pas non plus de critères concernant le contenu adéquat de ces documents. Ces éléments sont considérés comme étant en dehors du périmètre couvert par le standard.

[2] Alessandro Fantechi, Stefania Gnesi, Giuseppe Lami et Alessandro Maccari : *Application of Linguistic Techniques for Use Case Analysis* ; in Proceedings of RE'2002. <http://fmt.isti.cnr.it/WEBPAPER/RE02-revfin.PDF>

[3] Technique parfois utilisée dans le cadre de recettes utilisateur.

[4] John Sheil : *Towards economically optimal first-off inspection* ; European Journal of Operational Research, Volume 94, n° 1, 11 octobre 1996, pages 179-185).

La phase de « first off » est une livraison préliminaire ou partielle de la documentation pour certains clients. Il s'agit en général de clients (majeurs ou historiques) souhaitant avoir des fonctionnalités spécifiques du produit avant finalisation. Cette livraison partielle leur permet de spécifier concrètement leurs besoins (ajouts/changements en fonction des tests) plus précisément avant la phase de production, par rapport aux spécifications techniques « théoriques » réalisées avant développement et test. La documentation technique disponible en l'état (contexte des résultats attendus) peut à la fois servir « d'avant-vente » (périmètre des fonctionnalités et conformité par rapport à la version antérieure) et de documentation de ciblage des tests des fonctions/tâches majeures à engager ou nécessitant des ajustements.

[5] Jaroslav Drazan et Vladimir Mencl : *Improved processing of textual use cases: Deriving behavior specifications*. Organization: SOFSEM (2007)

<http://d3s.mff.cuni.cz/publications/download/DrazanMencl-ImprovedUC-SOFSEM2007.pdf>

[6] Kurt Bittner et Ian Spence : *Use case modelling* ; Addison-Wesley (2003), p. xvi. ISBN 9780201709131

[7] Michel Lanque et Philippe Larvet : *Automatic procedure building from use case detection within informal technical documents* ; CIDM (2010) <http://www.infomanagementcenter.com/enewsletter/2010/201001/fourth.htm>

[8] Samhaa R. El-Beltagy : *Ontology-based annotation of text segments* ; Computer Science Department (2007), Cairo University. Gizeh, Égypte, [http://www.claes.sci.eg/coe\\_wm/Sac07.pdf](http://www.claes.sci.eg/coe_wm/Sac07.pdf)

[9] Jean-Pierre Chanod, Boris Chidlovskii, Hervé Dejean, Olivier Fambon, Jérôme Fuselier, Thierry Jacquin et Jean-Luc Meunier : *From legacy documents to XML: A conversion framework* ; in Proceedings of European Conference on Digital Libraries (2005), <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.73.1502>

[10] OASIS specification tools and guidelines. Providers of tools and guidelines (juillet 2002), Eve Maler et Norman Walsh (Sun Microsystems).

[11] DITA Version 1.1 Language Specification, OASIS Standard, (août 2007)  
Chair: Don Day, Editors: Michael Priestley, Robert D. Anderson, JoAnn Hackos

[12] S1000D est une spécification internationale pour les publications techniques, utilisant une base de données sources commune. Voir <http://www.s1000d.net/>

[13] Un CMS (Content Management System) est une famille d'outils fournissant une collection de procédures de gestion de documents et modules documentaires, mises en œuvre dans un environnement collaboratif.

[14] Philippe Larvet : *Semantic application design* ; Bell Labs Technical Journal, août.2008, Volume 13, n° 2, pages 75 – 91, <http://portal.acm.org/citation.cfm?id=1405607> et <http://www3.interscience.wiley.com/journal/121376597/abstract?CRETRY=1&SRETRY=0>

[15] Objecteering, UML modeler, de Objecteering Software, <http://www.objecteering.com/>

[16] Rhapsody, UML modeler et simulateur, outil créé par i-Logix et désormais diffusé par IBM Rational  
<http://www-01.ibm.com/software/awdtools/rhapsody/>

[17] Par exemple Visual Paradigm for UML, <http://www.visual-paradigm.com> ; voir également  
[http://en.wikipedia.org/wiki/List\\_of\\_Unified\\_Modeling\\_Language\\_tools](http://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools)

[18] Philippe Larvet et Michel Lanque : *Generation of XML-DITA Procedures from use case detection in informal technical documents* ; DITA Europe 2009 Conference, novembre 2009, Munich, Allemagne.  
<http://www.infomanagementcenter.com/DITAEurope/2009/abstracts.htm>