
Gestion des changements et étude d'impact dans un système d'information réglementaire

Ovidiu Vasutiu^{*,**} — David Jouve^{**} — Youssef Amghar^{*}

^{*} *Laboratoire d'InfoRmatique en Images et Systèmes d'information – INSA de Lyon*
20 av. Albert Einstein, F-69621 Villeurbanne cédex
{ovidiu.vasutiu, youssef.amghar}@insa-lyon.fr

^{**} *Centre National d'Études et de Développements Informatiques de Lyon – CNAF*
Tour du Crédit Lyonnais, 129 r. Servient, F-69326 Lyon cédex 03
{ovidiu.vasutiu, david.jouve}@cnedi69.cnafmail.fr

RÉSUMÉ. Pour des organisations gérant le droit la maîtrise de la réglementation et de ses évolutions revêt d'un caractère stratégique très fort et passe par la maîtrise du patrimoine documentaire et du système d'information. Dans cet article nous proposons une approche qui permet de conduire des études d'impact sur les deux volets (documentaire et logiciel) basées sur des représentations génériques du SI sous forme d'un graphe canonique de dépendances. Ce travail de recherche a été conduit dans le cadre de la Caisse Nationale des Allocations Familiales, dont la mission principale est l'attribution des prestations familiales conformément à la législation. C'est ainsi que les principes de notre modèle sont évalués dans un contexte réel lors d'une expérimentation qui laisse déjà entrevoir, comme perspective applicative, la mise en place des outils adéquats pour une réelle maîtrise des connaissances réglementaire et de leur évolution.

ABSTRACT. Managing the law is a very strategically aspect for organization working and applying the legal domain. The management of the law implies controlling the documents representing it and the information systems implementing it. In this paper we propose an approach for identifying and quantifying impacts in the two forms of law (documents and its software implementations) based on a dependency graph representation of information system. This research was conducted at the Caisse Nationale des Allocations Familiales whose mission is to apply the law in attributing family allowances. Therefore our model was tested in a real environment and presents already several opportunities, such as the development of tools managing legal knowledge and its evolution.

MOTS-CLÉS : Gestion des changements, Mesure d'impact, Système d'Information, Graphes, XML, UML, Document Réglementaire.

KEYWORDS: Changes management, Impact quantification, Information System, Graphs, XML, UML, Legal Document

1. Introduction

Les textes réglementaires constituent, à l'image de toute entreprise produisant des biens de consommation, une matière première spécifique pour les organisations qui ont pour mission la mise en application du droit : leur activité est fondée sur l'analyse et la manipulation de cette matière. Au vu de la quantité massive d'information traitée, leur travail dépasse très largement les capacités humaines, les amenant à solliciter l'assistance de la machine et plus précisément d'un système d'information réglementaire. Les textes réglementaires (corpus documentaire) et leur implémentation logicielle (le système d'information) constituent un véritable patrimoine, une ressource stratégique. Une des principales caractéristiques des textes réglementaires est que ceux-ci sont en perpétuelle évolution. Assurer la cohérence et la conformité de ce patrimoine réglementaire, face aux évolutions réglementaires, engendre alors un coût humain et financier important. Ainsi la maîtrise des impacts au sein du patrimoine documentaire et du système d'information est une problématique stratégique.

Dans cet article nous proposons un modèle générique pour la maîtrise du patrimoine réglementaire basé sur des mécanismes d'étude d'impacts au sein des textes réglementaires d'une part et des composants du système d'information d'autre part.

Cette problématique générale est issue du contexte de la Caisse Nationale des Allocations Familiales (Cnaf) en charge d'une des législations les plus complexes du droit français, exprimée au travers de nombreuses ressources critiques pour l'entreprise : documents de référence, supports de formation, documentation métier, etc. Ainsi notre modèle générique sera illustré dans le cadre spécifique de la Cnaf mais son champ d'application peut-être élargi à d'autres domaines.

La section 2 énonce plus précisément cette problématique en identifiant les caractéristiques du patrimoine réglementaire (notamment les liens entre les textes réglementaires et le système d'information). Puis après une présentation et discussion sur l'état de l'art (section 3), la section 4 décrit le modèle pour l'étude des impacts au sein d'un système d'information. Dans la section 5, les principes de notre modèle seront évalués dans un contexte réel et nous analyseront et discuterons leurs résultats. Nous concluons en présentant quelques perspectives applicatives et académiques (section 6).

2. Problématique

La matière réglementaire est normative, hiérarchisée et dynamique (Chabbat, 1997) (Jouve *et al.*, 2001a). Elle est *normative* puisqu'elle décrit un ensemble de contraintes comportementales auquel il faut se conformer. Elle est *hiérarchisée* puisqu'elle est organisée en une structure pyramidale (Figure 1) dont les différents niveaux

correspondent aux niveaux de droit et déterminent ainsi la valeur juridique des textes et leur degré d'applicabilité. Généralement ce degré est inversement proportionnel à la valeur juridique. Les lois fondamentales, les plus importantes, sont souvent très générales, elles énoncent des principes difficiles à appliquer en situation concrète. En France, par exemple cette hiérarchie présente au sommet : la Constitution (la loi fondamentale) suivie de lois, des décrets, des circulaires ministérielles, etc.

Par ailleurs, une organisation qui a pour mission l'application du droit, peut prolonger cette pyramide, en produisant ses propres documents internes venant préciser et enrichir les textes externes. Ainsi la matière réglementaire est travaillée, affinée tout au long d'un processus de création du droit permettant d'atteindre un degré d'applicabilité et d'opérationnalité maximal situé au plus bas niveau de la pyramide : la spécification des règles de gestion, niveau utilisé pour la spécification et la conception des composants du système d'information (Jouve *et al.*, 2001b).

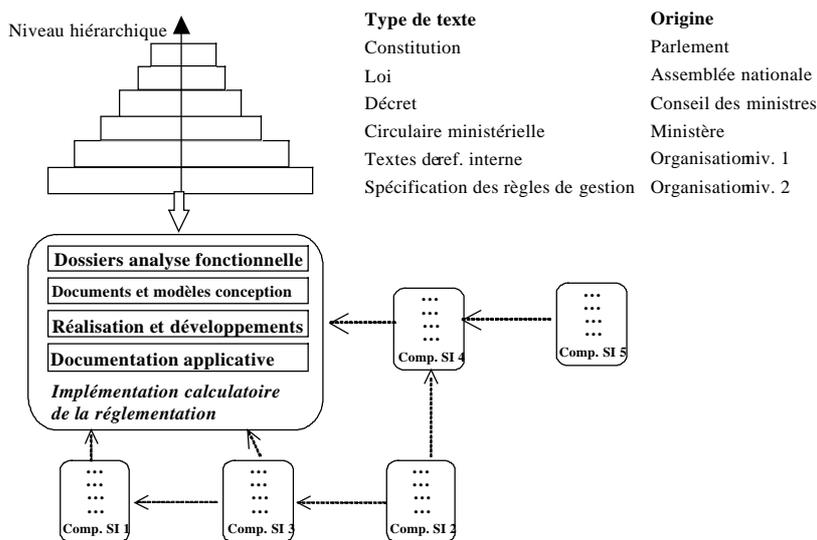


Figure 1. Pyramide réglementaire et système d'information

La matière réglementaire est *dynamique*. Les textes réglementaires doivent être perçus tel un flux exprimant de connaissances en constante évolution. Pour arriver à gérer la cohérence de la matière réglementaire il est nécessaire de gérer cette spécificité. En effet, lorsqu'une modification intervient à un certain niveau de la pyramide, l'ensemble des ressources en aval (documents réglementaires, documents de spécification, les composants du SI liés à la réglementation, etc.) peut être impacté. L'impact à un niveau est répercuté en cascade affectant toute la pyramide jusqu'au

niveau de base qui est représenté par les règles de gestion, internes à l'organisation impactant ainsi directement un nombre de composants du système d'information liés à la réglementation. Mais le chaînage des impacts ne se cantonne généralement pas à ces composants frontières. Eux mêmes peuvent être liés, au moyen de relations de dépendances, à d'autres composants au sein du SI, induisant ainsi une réaction en chaîne qu'il est nécessaire de maîtriser (Figure 1).

Pour des organisations telles que la Cnaf, la maîtrise globale (patrimoine documentaire et système d'information) des impacts liés au changement de la réglementation, revêt un caractère très fort.

Comment les identifier ? Comment arriver à une maîtrise globale de ce patrimoine réglementaire ?

Dans le cadre de cet article nous proposons une approche en trois phases : la maîtrise de la production documentaire à tous les niveaux de la pyramide, la maîtrise de la mise en oeuvre dans le SI et la mise en relation de ces deux mécanismes de gestion.

Dans de précédents travaux, David Jouve (Jouve, 2003) a étudié les problématiques liées à la gestion et à la maîtrise des textes réglementaires. La relation de dépendance entre deux composants documentaires de même niveau ou de niveaux différents de la pyramide est basée sur les connaissances exprimées (deux textes différents peuvent exprimer une même connaissance). Ainsi, il propose un système basé sur la représentation des connaissances pour garantir la cohérence des textes réglementaires entre les différents niveaux de la pyramide réglementaire (voir section 3). Or les relations de dépendances entre les composants du système d'information sont des relations de référencement, de composition, d'utilisation, etc.

Néanmoins le même modèle de dépendance ne peut pas être appliqué de la même façon au monde logiciel et à celui de la documentation. Ainsi dans le cadre de cet article nous concentrons nos explications sur une partie de la problématique, celle spécifique au monde logiciel. Nous explicitons les formes de dépendances entre les composants du SI et proposons un mécanisme d'étude d'impact logiciel.

3. Etat de l'art

La problématique exprimée ci-dessus se situe au carrefour de quatre grands domaines de recherche : l'ingénierie documentaire, la théorie du droit, la représentation de connaissances et le domaine de la conception des SI. Les recherches bibliographiques, ainsi que les travaux de recherche conduits précédemment nous ont permis de couvrir certaines parties de ces domaines.

3.2 Théorie du droit, ingénierie documentaire, représentation de connaissances

De nombreux travaux scientifiques ont permis d'élaborer des modèles de représentation et de manipulation des documents réglementaires : logique déontique (Wright, 1963) (Royackers, 2000), ontologies (Kralingen *et al.*, 1995), argumentation juridique (Bench-Capon, 1997), indexation et recherche d'information (Pietrosanti *et al.*, 1999) (Winkels *et al.*, 2000), représentations non formelles et traitements (Merckel *et al.*, 1999). David Jouve, en se basant sur cet état de l'art et les travaux de B. Chabbat (Chabbat, 1997), s'est attaché à la problématique engendrée par les changements de réglementation sur les corpus documentaires et sur ses implémentations calculatoires. Il décrit un modèle de représentation des connaissances réglementaires et de mise en relation des niveaux de droit, à l'aide d'une syntaxe abstraite hybride dérivant à la fois des langages de frames et des Graphes Conceptuels (Sowa, 1984) (Chein *et al.*, 1992) : *système primitif des G-Frames* (Jouve *et al.*, 2003b). Le système de *G-Frames* ainsi que les mécanismes d'inférence et d'annotation sémantique qui lui sont attachés apportent une réponse aux principales problématiques rencontrées dans la maîtrise d'un patrimoine documentaire réglementaire : études d'impact, détection des incohérences et des conflits.

3.3 Architecture, conception et maîtrise des SI

Les recherches bibliographiques conduites au préalable nous ont permis de dénombrer deux grandes approches dans la conception des systèmes d'information:

— L'approche basée sur des langages formels, décrivant le système à l'aide d'une sémantique forte, parfaitement définie, utilisés souvent à des fins calculatoires (modèles compréhensibles et traitables par l'ordinateur). Nous pouvons citer en exemples:

- le langage Z (PRG, 1980) (Spivey, 1992), appliqué surtout dans le cadre de projets d'informatique industrielle, est un langage mathématique formel, basé sur la théorie des ensembles de Zermelo-Fränkel et sur la logique de premier ordre. Des évolutions existent avec Z++ (Lano, 1991), Object Z (Duke *et al.*, 1991), et la méthode Syntropy (Cook *et al.*, 1994) remplacée par UML.

- le langage B et la méthode B (Abrial, 1996) décrivent le système à l'aide des automates abstraits en utilisant, comme le langage Z, les opérateurs de la théorie des ensembles. Le langage B n'est pas orienté objet et n'a pas un formalisme graphique.

REMARQUE — Des travaux sont en cours pour faire évoluer les deux langages (B et Z) qui ont du mal à s'imposer dans les contextes très opérationnels des entreprises. Les charges associées à la réalisation, la maintenance et l'exploitation de tels modèles sont très élevées ce qui représente un frein à leur appropriation par les entreprises.

— L'approche fondée sur des langages moins formels, plus proches du langage humain, souvent orientés objet, plus adaptés à l'entreprise. Nous pouvons citer ici:

- Merise, qui est avant tout une méthode d'analyse, permet d'aboutir séparément à des modèles conceptuels, semi-formels avec peu de possibilités d'extension, représentés dans deux types de vues, indépendantes de toute implémentation: une vue statique des données (MCD) et une vue dynamique des traitements (MCT).

- UML (OMG, 2003) propose une notation semi-formelle composée de différents diagrammes : diagrammes de classes, d'état-transition, de cas d'utilisation, etc... Le langage est suffisamment souple pour s'adapter à un contexte très opérationnel d'une entreprise (rapidité de conception, clarté architecturale des modèles, supporté par la plupart des outils de conception du marché, facilité de compréhension par les informaticiens ou les non-informaticiens) et en même temps il peut être enrichi du point de vue sémantique avec le langage OCL – Object Constraint Language (OMG, 2003) et syntaxique par des mécanismes tels que les profils, les stéréotypes, etc.

Les contraintes et les exigences du contexte de nos travaux nous conduisent à choisir des langages de modélisation avec les propriétés suivantes :

- capacité du langage à fournir, tout au long de son cycle de vie, une représentation du SI selon le paradigme objet,
- capacité du langage à être mis en œuvre dans un contexte industriel sans un surcoût important,
- possibilité de décrire des études d'impact,
- extensibilité et spécialisation du langage.

Dans ce contexte, notre choix penche en faveur des langages de la deuxième catégorie. Au vu de l'état de l'art, UML¹ nous semble offrir tous les avantages et les fonctions qui nous sont nécessaires pour une modélisation du système d'information.

3.4 Maîtrise des modèles (Etude d'impact au sein d'un modèle UML)

Au delà de la conception des systèmes d'information, plusieurs travaux se sont intéressés à la maîtrise des modèles obtenus : leur rétro-conception, leur optimisation ou bien le maintien de leur cohérence.

Les ateliers de développement (ex: Eclipse, IBM Rational Rose (IBM, 2005)), mettant en œuvre UML, ont, grâce à l'étroite liaison entre le code et le modèle (Enns, 2003), des fonctionnalités de génération de squelettes de code et de rétro-conception.

¹ Pour assurer l'évolutivité de la proposition, notre réflexion a porté premièrement sur la base commune aux dernières versions du langage UML : la version 1.4.2 d'UML (OMG, 2003) et la version 2.0 (OMG, 2004).

D'autres travaux en cours se penchent sur la problématique de refonte des modèles, « *refactoring* » (Sunuyé *et al.*, 2001) (Roberts, 1999), compliquée à automatiser et devenue d'actualité grâce notamment à l'émergence des patrons de conception (« design patterns ») et de l'architecture orientée modèles (MDA). D'autres auteurs se sont intéressés au contrôle de la cohérence des modèles UML, « *consistency* » (Sourrouille *et al.*, 2002) (Krishnan, 2000).

Néanmoins, la problématique de la gestion des changements et des études d'impact au sein des modèles et représentations UML, nous semble être peu abordée dans la littérature.

4. Modèle d'étude d'impact

Nous proposons de nous focaliser sur l'étude des interactions dans un système d'information en se basant sur les diagrammes UML, chaque diagramme représentant une vue différente sur le modèle. Ainsi, les connaissances extraites d'un diagramme peuvent être enrichies par des connaissances obtenues d'autres diagrammes afin d'obtenir une vision plus complète du système.

4.1 Objet de l'étude d'impact : les concepts UML

Définition 1 : Concept UML – Nous appelons concept UML tout élément présent dans le formalisme UML : les diagrammes, les classes, les attributs, les rôles, les méthodes ainsi que leurs propriétés telles que le nom, le type, la visibilité, etc.

Dans l'usage des concepts UML on remarque que parfois le même aspect d'un modèle peut être représenté des manières différentes avec des concepts UML différents. L'exemple suivant présente le concept d'attribut et celui de rôle : d'une part, Figure 2a, une classe *ClasseA* avec un attribut de type *ClasseB*, et d'autre part, Figure 2b, une association entre deux classes, *ClasseA* et *ClasseB*, basée sur un rôle.

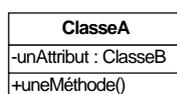


Figure 2a. Relation classe-attribut



Figure 2b. Relation d'association avec rôle

Définition 2 : Concept élémentaire – L'ensemble de concepts élémentaires *CE* est un sous ensemble des concepts UML composé de telle manière qu'aucune relation d'équivalence sémantique ne peut être définie entre deux de ses éléments (c'est à dire

qu'il n'existe aucun moyen de représenter la même connaissance avec deux concepts élémentaires différents).

La stratégie de construction de cet ensemble de concepts élémentaires consiste à ne conserver entre deux éléments sémantiquement équivalents que l'élément le plus expressif. Par exemple : le concept d'attribut de la Figure 2a peut se rapporter au concept de rôle, Figure 2b, plus générale et plus représentatif puisqu'il fait apparaître, combiné avec la relation d'association, la notion de cardinalité (0..1, ou 1..1). C'est ainsi que le concept UML d'attribut n'est pas considéré comme un *concept élémentaire* à l'inverse du concept UML de rôle.

Définition 3 : Relation élémentaire – Nous appelons une relation élémentaire notée $re(x,y)$, une relation transitive définie sur l'ensemble de concepts élémentaires CE induisant une forme de dépendance entre deux *concepts élémentaires* x et y . Le sens de la relation est donné par le sens de cette dépendance.

A partir du sous-ensemble UML manipulé à travers les diagrammes de classe et de séquence² nous identifions 7 types de *relations élémentaires* :

- *association* : association UML entre deux classes
- *généralisation* : relation UML de généralisation/spécialisation entre deux classes
- *agrégation par valeur, par référence* : relation UML de composition de deux classes
- *dépendance* : relation UML de dépendance entre deux classes
- *relation structurelle d'encapsulation*: relation entre une classe et ses membres (méthodes et rôles) et entre un membre et ses propriétés (nom, visibilité, cardinalité)
- *relation d'accès à un membre (relation d'utilisation)*: relation déduite des diagrammes de séquence qui offrent une autre perspective sur le modèle : *une méthode invoque une autre méthode, une méthode utilise un rôle.*

Les notions de *concept élémentaire* et de *relation élémentaire* permettent d'élaborer un graphe de dépendance servant de support pour les études d'impacts.

4.2 Définition d'une étude d'impact

Définition 4 : Élément « Impactable » – Élément d'un modèle susceptible d'être marqué comme « *impacté* »³. Pour un modèle donné, l'ensemble des éléments impactables est constitué par l'ensemble des occurrences des concepts élémentaires du modèle. Exemple : une classe, un rôle, son nom ou sa visibilité.

² Pour des raisons de concision on s'est limité à l'étude des diagrammes de classe et de séquence, mais l'ensemble de *concepts élémentaires* et de *relations élémentaires* peut être élargi aux 7 autres diagrammes du formalisme UML 1.4.2 (OMG, 2003).

³ Un élément marqué comme « *impacté* » est un *élément impacté*, voir *définition 9*.

Définition 5 : Evénement impactant – Evénement ayant pour cible directe un *élément impactable*, et dont la conséquence est de marquer ce dernier de l'étiquette « impacté ».

Les *événements impactants* se propagent sur les *relations élémentaires* pour se répercuter sur d'autres *éléments impactables* (cibles indirectes). Ils peuvent être de différents types : Ajout / Suppression / Modification. Ainsi, par exemple : modifier le nom d'un rôle d'une classe impacte en premier lieu le rôle, mais en deuxième temps, les méthodes qui l'utilisent et plus généralement la classe.

Définition 6 : Impact – désigne la conséquence de la survenue d'un *événement impactant* sur le modèle. Ainsi, l'*impact* d'un *événement impactant* *E* sur un modèle *M* correspond à l'ensemble des *éléments impactables* de *M* marqués de l'étiquette « impacté » suite à la survenue de *E*.

Définition 7 : Degré d'impact élémentaire – représente sur une échelle de 0 à 100 l'étendue estimée de l'impact sur un *élément impactable*.

Le *degré d'impact élémentaire* permet d'établir une mesure estimative de l'effort nécessaire pour mettre à jour l'élément pour palier à cet impact. On peut également prendre en compte le type d'impact et le type de l'élément impacté : supprimer un rôle peut représenter dans certains systèmes un impact plus coûteux que la modification de la visibilité du même rôle.

Définition 8 : Conviction d'impact élémentaire – représente sur une échelle de 0 à 100 la précision avec laquelle nous pouvons affirmer la présence d'un *impact élémentaire*.

UML est un langage semi-formel. La *conviction d'impact élémentaire* permet de prendre en compte les cas où les impacts ne peuvent pas être identifiés avec certitude. La valeur 0 correspond au cas où nous ne pouvons pas déterminer s'il existe un impact. C'est à dire que nous n'avons, au vu des données fournies sur le système, *aucune* conviction qu'il y ait un impact. La valeur 100 correspond quant à elle au cas où un impact a été identifié à l'aide des données fournies. Les valeurs intermédiaires peuvent nuancer cette conviction en fonction de notre connaissance du domaine : 75 plus de conviction que dans le cas de 50. Par analogie avec la logique floue nous essayons ainsi de mesurer avec des heuristiques la certitude qu'un impact ait lieu ou non.

Définition 9 : Elément impacté – est un élément « *impactable* » élémentaire qui subit un impact avec un *degré* et une *conviction* d'impact supérieurs à zéro.

Définition 10 : Etude d'impact – est le calcul du degré d'impact et de la conviction d'impact pour tous les *éléments impactables* du modèle et suite à un *événement impactant*.

Les modalités de propagation des *événements impactants* au sein d'un modèle dépendent étroitement de la nature des relations définies entre ses *éléments impactables*.

Ainsi, si la classe B est une spécialisation de la classe A, la modification du nom de B n'engendre aucun impact sur A : la propagation des événements s'effectuant dépendamment de l'orientation des relations.

Définition 11 : Coefficient de propagation: est un indice de 0 à 100, porté par toute relation élémentaire du modèle, qui permet lors de la propagation de l'impact de A vers B, d'atténuer l'impact.

Ce coefficient dépend d'un certain nombre de paramètres établis en fonction du contexte du domaine, du système représenté, des technologies employées, du type et de la signification associée à chaque relation (par ex. : une relation de composition aura un coefficient de propagation différent d'une relation d'association).

Les concepts définis ci-dessus définissent la statique de notre modèle d'étude d'impacts. Au cours de la section suivante, nous proposons une procédure d'exploitation de cette structure afin d'obtenir une mesure estimative de la portée d'un impact sur le modèle UML du SI.

4.3. Procédure d'étude d'impact

Pour un modèle M donné, l'analyse des concepts et des relations élémentaires permet d'élaborer un graphe de dépendance. Un tel graphe est nommé *graphe canonique*.

4.3.1 Construction du graphe canonique

Le *graphe canonique*, consolide les informations extraites du modèle UML et constitue ainsi un support au calcul des degrés d'impact et des convictions d'impact au sein du modèle, suite à la survenue d'un événement impactant.

Définition 12 : Ensemble des occurrences – Soit un modèle M , nous notons CE_M l'ensemble des occurrences des concepts élémentaires apparaissant dans M .

Définition 13 : Graphe canonique – Nous appelons Graphe Canonique un graphe $GC_M = (CE_M, dep)$ tels que :

- CE_M : ensemble des *concepts élémentaires* représentés par les nœuds du graphe associé au modèle M
- dep : est une application $CE_M \times CE_M \rightarrow [0..100]$ telle que $\forall a, b \in CE_M$ donc:
 - $dep(a,b)=0$ s'il n'y a pas de relation élémentaire de dépendance entre a et b , c'est-à-dire : $\neg re(a,b)$
 - $dep(a,b)=\alpha$ où α est le coefficient de propagation associé au type de la relation élémentaire $re(a,b)$, entre a et b

4.3.2 Exemple

Considérons le diagramme UML suivant :



Figure 4. Exemple UML

Nous en déduisons le graphe canonique associé :

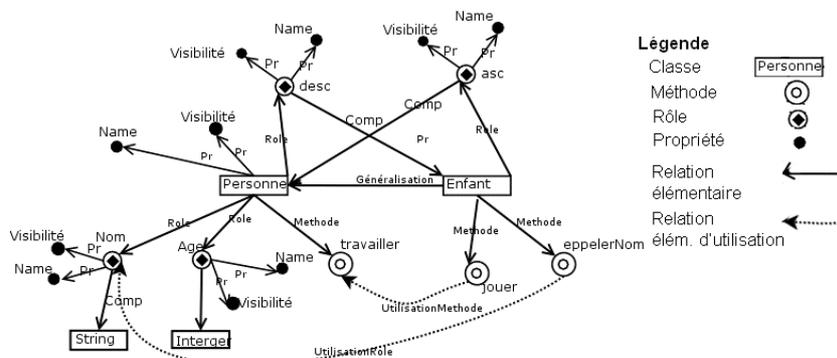


Figure 5. Graphe canonique associé

Dans l'exemple ci-dessus, l'occurrence *Personne* du concept élémentaire Classe est une généralisation de l'occurrence *Enfant* du même concept élémentaire et donc induit que la relation élémentaire : $dep(Enfant, Personne) = \alpha$ où α est le coefficient de propagation associé à la relation de généralisation (un impact de la classe parent risque de se propager à la classe fils). Par contre $dep(Personne, Enfant) = 0$ puisqu'une modification de la classe fils n'affecte pas le parent.

4.3.3 Calcul des impacts

Traduire un modèle UML dans un *graphe canonique* nous permet d'appliquer les notions et les principes liés à la théorie des graphes (notamment des mécanismes de parcours de graphes). Chaque arc a un sens donné par la *relation élémentaire* entre les deux *concepts élémentaires* représentés par ses extrémités, la propagation des impacts se fait en sens inverse de l'arc. En partant d'un *élément impacté* et après identification de tous les éléments qui en dépendent, nous construisons tous les chemins de propagation possibles.

Définition 14 : Chemin de propagation : succession d'éléments impactés suite à une modification du modèle.

Chaque concept élémentaire $a \in CE_M$ du chemin de propagation est caractérisé par le *degré élémentaire* et la *conviction élémentaire* de l'impact, calculés en fonction de son prédécesseur b , dans le chemin de propagation.

Le degré d'impact élémentaire de b sur a noté $degElem(a,b)$ se définit :

$$degElem(a,b) = \frac{degElem(b,c)}{card(C_a)}$$

c étant le prédécesseur de b dans le chemin de propagation et C_a est le C_a le sous-ensemble de CE_M contenant les éléments dont a dépend :

$$C_a = \{\forall b \in CE_M \mid re(a,b) \neq 0\}$$

NOTE — Si le prédécesseur c de b n'existe pas, cela signifie que b est la source initiale de l'impact, donc, par défaut $degImpact(b,c)=100$.

Dans le même contexte, la *conviction d'impact élémentaire* de b sur a notée $convictionElem(a,b)$ se définit de la forme suivante :

$$convictionElem(a,b) = \frac{convictionElem(b,c) \times dep(a,b)}{100 \times card(C_a)}$$

Par ailleurs, au niveau algorithmique, le graphe étant susceptible de contenir des cycles et afin d'éviter des chemins de propagation infinis, à chaque étape nous vérifions si le nœud a été déjà visité dans le chemin de propagation en cours. Nous utiliseront un algorithme de parcours de graphe cyclique en calculant, pour chaque élément, le degré et la conviction d'impact en fonction des prédécesseurs et en déduisant tous les chemins de propagation. En effet lors qu'un élément a plusieurs successeurs, l'impact va prendre plusieurs de chemins de propagation. Le parcours du graphe produit ainsi une liste de chemins de propagation qui seront consolidés par des algorithmes de consolidation.

4.3.4 Consolidation des impacts

Dès que le modèle contient de multiples éléments liés par diverses relations, la propagation des impacts devient complexe et induit la création de nombreux chemins de propagation. Un même élément peut être membre de plusieurs chemins de propagation et donc être touché par divers impacts élémentaires. Les utilisateurs préfèrent avoir une vision consolidée des impacts.

Définition 15 : Degré d'impact consolidé: Le degré d'impact consolidé d'un élément a appartenant à un modèle M , noté $degre(a)$, représente sur une échelle de 0 à 100 la somme de ses degrés d'impact élémentaires.

Définition 16 : Conviction d'impact consolidée: La conviction d'impact d'un élément a appartenant à un modèle M , notée $conviction(a)$, représente sur une échelle de 0 à 100 la somme de ses convictions d'impact élémentaires.

$$\text{degre}(a) = \sum_{b \in C_a} \text{degreElem}(a,b) \quad \text{conviction}(a) = \sum_{b \in C_a} \text{convictionElem}(a,b)$$

5. Expérimentation et discussions

Afin de valider par l'expérimentation les diverses propositions émises dans le cadre de nos travaux, nous avons développé, en utilisant le langage Java, une application web et mis ainsi en place les bases d'une plate-forme de gestion des composants du système d'information intégrant un module d'étude d'impacts. L'outil construit en mémoire le graphe canonique des impacts en analysant un document XMI (forme sérielle UML) (OMG, 2003). Le graphe canonique ainsi obtenu est alors utilisé afin de réaliser les études d'impact et d'en déduire, à partir d'un événement impactant sélectionné, les impacts consolidés ainsi que tous les chemins de propagation. Les résultats sont reportés sur les mêmes diagrammes UML fournis en entrée grâce à un mécanisme d'annotation visuelle (jeu de couleurs, info-bulles) (Figure 6). Pour cela l'outil met en œuvre un ensemble de transformations XSLT (W3C, 2004) afin d'enrichir le document XMI et d'en obtenir une représentation graphique : notation UML encodée en SVG⁴ ce qui nous permet d'avoir un procédé transparent pour l'utilisateur qui ne verra que les diagrammes UML enrichis. Nous souhaitons faciliter ainsi l'utilisation de l'outil et éviter l'appropriation d'un nouveau formalisme tel que le graphe canonique. Les diagrammes et les impacts peuvent être visualisés dans n'importe quel navigateur web.

Notre expérimentation a été réalisée sur un modèle UML correspondant au modèle de gestion de prestations familiales de la Cnaf. Pour cette expérimentation nous avons défini les coefficients de propagation suivants :

<i>Relation élém.</i>	<i>Coeff.</i>	<i>Relation élém.</i>	<i>Coeff.</i>
Association	100	Dépendance	50
Généralisation	75	Relation structurelle	100
Agrégation par valeur , par réf.	100	Relation d'utilisation	100

Tableau 1. Table de paramétrage des coefficients de propagation employés à la Cnaf

Dans la pratique, pour un modèle d'application précis, ces paramètres peuvent être ajustés après des expérimentations afin de mieux s'approcher de la réalité. Donc pour améliorer l'acuité de l'étude nous travaillons à spécifier un paramétrage plus fin en fonction du domaine de la Cnaf et à affiner les formules heuristiques de calcul.

⁴ SVG : Scalar Vectorial Graphics : langage de dessin vectoriel en XML (W3C, 2003)

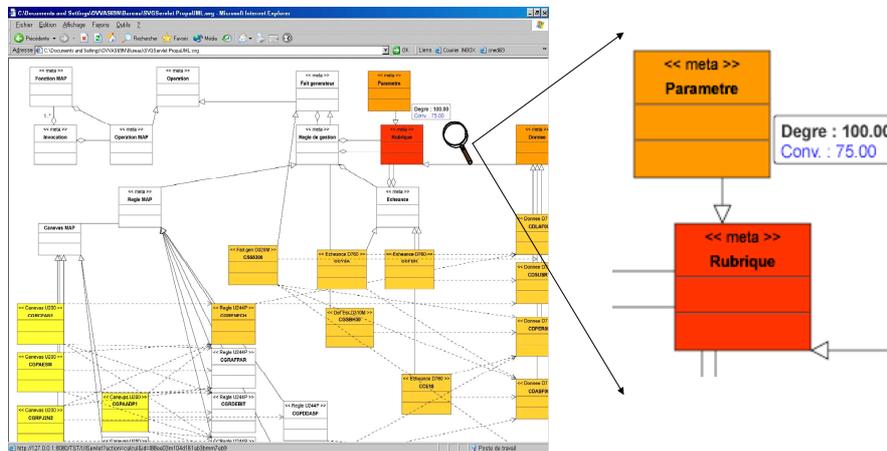


Figure 6. Etude d'impact sur le système de gestion des prestations familiales.

Par ailleurs les résultats estimés ont été proposés à un groupe de développeurs en charge de l'application afin qu'ils soient confrontés à leur propre perception. Les résultats ont été jugés assez proches de la réalité, de ceux obtenus par une estimation manuelle. Néanmoins une limitation a été identifiée : le système dans sa version actuelle ne permet pas de moduler l'estimation du nombre ou l'étendu de l'impact en fonction du respect des bonnes règles de développement orienté objet (bonne utilisation de l'encapsulation). Ces règles permettent, en effet, de réduire les impacts réels par rapport aux impacts estimés seulement en analysant le modèle UML. Par conséquent le modèle d'estimation est qualifié comme pessimiste, dans la situation actuelle il détecte plus d'impacts que dans la réalité.

6. Conclusion

Dans cet article nous avons proposé un mécanisme générique pour l'étude d'impact au sein d'un modèle d'un système d'information nous permettant d'identifier l'ensemble des composants impactés à partir d'une modification survenue. Notre contribution a permis d'établir un support essentiel et nécessaire pour l'instauration d'un contrôle de la cohérence et de la conformité de la composante logicielle du patrimoine réglementaire. Notre problématique générale est par contre la maîtrise globale ce patrimoine réglementaire face aux évolutions de la réglementation. Ces évolutions sont propagées d'abord dans l'ensemble du corpus documentaire à l'aide d'une étude d'impact documentaire basée sur les principes énoncés par David Jouve (Jouve, 2003), jusqu'aux documents de spécification des composants logiciels directement liés à la réglementation. C'est ainsi que le maillon manquant pour arriver à une maîtrise globale,

que nous retenons comme une perspective de recherche, est l'unification des procédures d'études impacts menées sur le SI avec celles opérées sur le plan documentaire en se basant notamment sur la notion de traçabilité de l'usage fait des connaissances réglementaires du domaine.

Par ailleurs les perspectives applicatives les plus importantes sont liées à l'intégration de ces principes dans des outils d'analyse et d'aide à la conception permettant ainsi d'estimer les charges associées avec les évolutions du système d'information.

Notre travail de recherche s'est déroulé dans le cadre de la Caisse Nationale des Allocations Familiales, c'est ainsi qu'un autre axe de recherche concerne de prendre en compte le passage à l'échelle pour arriver à une modélisation de systèmes d'information très complexes et de grande envergure.

Remerciements

Les auteurs tiennent à remercier le laboratoire LIRIS et la Cnaf (notamment M. Jacques Faveeuw, M. Cyrille Broillard et M. Bertrand Chabbat) pour avoir mis à leur disposition les structures et les ressources nécessaires au bon déroulement des travaux.

7. Bibliographie

- Abrial J. R., *The B Book –Assigning Programs to Meanings*, Cambridge University Press, ISBN 0-521-49619-5, 1996
- Bench-Capon T.J.M, *Arguing With Cases*, *Legal Knowledge Based Systems JURIX 97 : The Tenth Conference*, The Vrije Universiteit, 1997, pp. 85-99.
- Chabbat B., *Modélisation Multiparadigme de textes réglementaires*, Thèse de doctorat, LISI, Décembre 1997, 392p. Convention CIFRE 1994.
- Chein M., Mugnier M-L., *Conceptual Graphs : Fundamental Notions*. *Revue d'Intelligence Artificielle*, 1992, vol 6, n°4, p 365–406.
- Cook S., Daniels J. *Designing Object Systems: Object-oriented Modelling with Syntropy*. Prentice Hall 1994.
- Duke R., King P., Rose G.A., Smith G., *The Object-Z specification language*, *Technology of Object-Oriented Languages and Systems: TOOLS 5*, Prentice Hall, 1991, pp. 465–483
- Enss R., *Refactoring in Eclipse*, www.cs.umanitoba.ca/~eclipse
- Jouve D., Chabbat B., Amghar Y., Pinon J-M. *L'archivage des documents réglementaires : maîtrise d'une matière première spécifique*. *Document Numérique*, 2001, vol 4, p 315-329.
- Jouve D., Chabbat B., Amghar Y., Pinon J-M. *Modélisation sémantique de la réglementation*. *Ingénierie des Systèmes d'Information*, 2001, vol 6, n°2, p. 95-119.
- Jouve D., *Modélisation sémantique de la réglementation*, Thèse de doctorat, LIRIS, Novembre 2003, 258 p., Convention CIFRE 277/2000.

- Jouve D., Amghar Y., Chabbat B., Pinon J.-M.. Conceptual Framework for Document Semantic Modelling : an Application to Document and Knowledge Management in the Legal Domain. *Data & Knowledge Engineering*, 2003, vol 46, n_3, p 345 – 375.
- Kralingen (van) R. W., Frame-based Conceptual Models of Statute Law, *Computer/Law serie 16*, Kluwer Law International, The Hague , 1995
- Krishnan P., Consistency Checks for UML, *Proceedings of the 7th Asia-Pacific Software Engineering Conference*, 162-169, IEEE, December 2000.
- Lano K.C., Z++ - an object-orientated extension to Z, *Z User Workshop*, Oxford 1990, in: J.E. Nicholls Ed. *Workshops in Computing*, Springer-Verlag, 1991, pp. 151–172
- Merkel D., E. Schweighofer et W. Winiwarter, Exploratory Analysis of Concept and Document Spaces with Connectionist Networks, *Artificial Intelligence and Law*, Vol. 7, Nos. 2-3, Kluwer Academic Publishers, 1999, pp. 185-209.
- OMG *Unified Modeling Language Specification* version 1.5, formal/2003-03-01 The Object Management Group : <http://www.omg.org/>
- OMG *Unified Modeling Language Specification* version 2.0, formal/05-07-04 The Object Management Group : <http://www.omg.org/>
- Pietrosanti E. et Graziadio B., Advanced techniques for legal documents processing and retrieval, *Artificial Intelligence and Law*, Vol. 7, No. 4, Kluwer Academic Publishers, 1999, p. 341-361
- PRG *Programming Research Group*, Oxford University, 1980
- Roberts D. B.. Practical Analysis for Refactoring. PhD thesis, University of Illinois at Urbana-Champaign, 1999.
- Royackers L., Combining deontic and action logics for collective agency, *Legal Knowledge and Information Systems. Jurix 2000*, Amsterdam : IOS Press, 2000, pp 135-146.
- Spivey J.M., *The Z Notation: A Reference Manual*. Prentice-Hall, 1992.
- Sourrouille J.L., Caplat G., Constraint Checking in UML Modeling, Int. Conf. *SEKE'02, ACM-SIGSOFT*, 2002, pp217-224
- Sowa J. F. Conceptual Structures : Information Processing in Mind and Machine. *The system programming series*. Addison-Wesley, 1984. 481 p.
- Sunuyé G., Pollet D., Le Traon Y., Jézéquel J.-M., Refactoring UML Models, in: M.Gogolla and C.Kobryn (eds), *UML 2001*, LNCS 2185, 134-148, Springer 2001.
- W3C World Wide Web Consortium *Scalable Vector Graphics (SVG) 1.1 Specification*, 14 January 2003. Disponible sur: <http://www.w3.org/TR/SVG/> (consulté le 15/01/2006)
- W3C World Wide Web Consortium. *Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 04 February 2004*. Disponible sur <http://www.w3.org/TR/REC-xml/>
- Winkels R.G.F, Bosscher D.J.B., Boer A.W.F et Hoekstra R., Extended conceptual retrieval, *Legal Knowledge and Information Systems. Jurix 2000*, Amsterdam : IOS Press, 2000, pp. 85-97.
- Wright (von)G.H., Norm and Action : A Logical Enquiry. *International Library of Philosophy and Scientific Method*, Routledge and Kegan Paul, London, 1963. 214 p.