

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

La rédaction du Cahier des Charges :

Petit, Michaël; Rousseau, Anne; Schobbens, Pierre-Yves; Lobet-Maris, Claire

Published in:
Urgence aux urgences

Publication date:
2004

Document Version
le PDF de l'éditeur

[Link to publication](#)

Citation for pulished version (HARVARD):

Petit, M, Rousseau, A, Schobbens, P-Y & Lobet-Maris, C 2004, La rédaction du Cahier des Charges : pour une méthodologie pluridisciplinaire et participative d'élicitation des exigences : chapitre 3. Dans V Dumont, LM Cl. & A Rousseau (eds), *Urgence aux urgences: L'informatique, une solution*. Presses universitaires de Namur, Namur, p. 53-74.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Chapitre 3

La rédaction du Cahier des Charges : pour une méthodologie pluridisciplinaire et participative d'élicitation des exigences

Anne ROUSSEAU

Claire LOBET-MARIS

Cellule Interfacultaire de Technology Assessment

(CITA)

Facultés Universitaires Notre-Dame de la Paix

Namur

clo@info.fundp.ac.be

Michaël PETIT

Pierre-Yves SCHOBENS

Laboratoire d'Ingénierie des Exigences du Logiciel

(LIEL)

Facultés Universitaires Notre-Dame de la Paix

Namur

pys@info.fundp.ac.be

1. Introduction

Dans ce chapitre, nous présentons une **méthodologie pluridisciplinaire et participative d'élicitation des exigences** débouchant sur la rédaction d'un cahier des charges nécessaire tant dans le cas de l'acquisition d'un logiciel que pour son développement sur site.

Cette méthodologie combine des éléments d'approche provenant des deux champs disciplinaires que sont l'Ingénierie des Exigences du logiciel (IE) et la Théorie des Organisations (TdO), raison pour laquelle nous parlerons d'Ingénierie Organisationnelle des Exigences (IOE).

Elle a été appliquée dans le cadre du projet ARTHUR⁴, pour les phases initiales du processus de définition des exigences d'un système informatique de support à l'activité du SUH. Cette application servira d'illustration pour les différentes étapes proposées.

Dans un premier temps, nous présentons brièvement les approches qui fondent cette

⁴ Projet ARTHUR (Architecture de Télécommunications Hospitalière pour les services d'Urgence) financé par la Région Wallonne, partenariat UCL, FUNDP, Ulg, FTPMs et 3 services d'urgences

méthodologie (IE et TdO) pour ensuite développer les raisons pour lesquelles cette analyse des exigences organisationnelles est un préalable important à l'informatisation des services d'urgences. Nous identifions ensuite les grandes étapes de la méthodologie proposée et, dans la foulée, les critères de qualité recommandés. Nous illustrons ensuite ces divers éléments par des exemples concrets issus de la démarche suivie dans le projet ARTHUR. Et enfin, nous précisons quel doit être le contenu général d'un cahier des charges pour un système informatique de support à l'activité d'un service d'urgences.

2 . En quoi consiste l'analyse des exigences organisationnelles ?

Comme annoncé ci-dessus, l'analyse des exigences organisationnelles combine des éléments issus des deux champs disciplinaires différents que sont l'Ingénierie des exigences du logiciel (IE) et la Théorie des Organisations (TdO) [30].

Depuis plusieurs années maintenant, l'**Ingénierie des Exigences du logiciel** (IE) est reconnue comme une discipline clef pour assurer le succès d'un projet informatique (voir par exemple [1, 2]) et a fait l'objet de nombreux travaux. L'IE a pour objectif d'élaborer un **cahier des charges** contenant une description, la plus précise et complète possible, des exigences que doit remplir un logiciel à développer.

Une exigence est définie dans le standard IEEE Std-610 [29] comme étant :

- Une condition ou capacité dont un utilisateur a besoin pour résoudre un problème ou atteindre un objectif.
- Une condition ou capacité qui doit être satisfaite ou possédée par un système ou composant de système afin de satisfaire un contrat, un standard, une spécification, ou tout autre document imposé formellement.
- Une représentation documentée d'une condition ou capacité telle que définie en 1 ou 2.

Le cahier des charges est élaboré à partir des besoins souvent exprimés de manière

floue ou imprécise par un certain nombre de parties prenantes à un projet d'informatisation. Il doit être compris et validé par l'ensemble de ces parties prenantes.

L'IE est un « processus systématique de développement d'un ensemble d'exigences grâce à un processus coopératif et itératif d'analyse du problème, de documentation des observations résultantes dans divers formats et de vérification de l'exactitude de la compréhension des exigences » [3].

Il débute lorsque la décision est prise de recourir à un système informatique pour rencontrer tout ou partie des objectifs de l'organisation. Cette décision résulte généralement d'une analyse informelle de l'organisation et de ses objectifs, au terme de laquelle la conviction est acquise qu'un système informatique est susceptible d'améliorer le fonctionnement de l'organisation ou de permettre de saisir des opportunités nouvelles. La phase d'IE consistera donc en un processus systématique visant à préciser les opportunités pressenties d'utilisation de logiciels pour les transformer en un ou plusieurs cahiers des charges contenant des exigences précises pour un ou plusieurs logiciels à développer. C'est précisément pour l'identification de ces opportunités que le recours à la TdO s'avère opportun : il s'agit pour l'essentiel, grâce aux outils de l'analyse organisationnelle, d'identifier les « points de rupture » du fonctionnement existant de l'organisation à informatiser, en vue de repérer les potentiels d'amélioration de celui-ci pour, in fine, cerner ceux qui sont susceptibles d'être satisfaits par le recours à l'informatique.

La phase d'IE se termine lorsque les exigences contenues dans le cahier des charges sont définies de manière suffisamment précise et non ambiguë et qu'un accord existe entre toutes les parties prenantes sur ces exigences et leur adéquation aux objectifs de l'organisation et aux motivations de ses membres [3]. Il importe donc particulièrement de s'assurer que l'ensemble des exigences imposées au logiciel à développer rencontre les objectifs et préoccupations des acteurs de l'organisation, faute de quoi le projet informatique s'expose à des risques d'échec élevés. Le recours à l'analyse organisationnelle vise également dans ce cas de figure à apporter des éléments de réponse à cette préoccupation.

Le propre de la méthodologie proposée dans ce chapitre est de combiner différentes grilles de lecture de l'organisation telles que proposées par la TdO avec des techniques d'élicitation fournies par l'IE. L'objectif de cette combinaison tient en l'acquisition d'informations générales et particulières sur le domaine d'application (en l'occurrence les services d'urgences hospitaliers) et sur les exigences (besoins, attentes et contraintes des utilisateurs potentiels et de l'organisation) relatives au système à développer. Par l'application de cette méthodologie, on déduit progressivement, à partir des informations récoltées grâce aux grilles de lecture de l'analyse organisationnelle, des exigences que l'on formalise et consigne dans un cahier des charges.

Comme nous allons le démontrer par la suite, la combinaison de ces deux points de vue complémentaires permet d'obtenir une vision complète de la problématique, avec pour conséquence une **meilleure adéquation du système informatique défini aux préoccupations organisationnelles**. Le point de vue de l'analyse organisationnelle permet de mettre en perspective les exigences imposées au logiciel.

De plus, la méthodologie repose sur une approche participative et active des acteurs de l'organisation, ayant pour avantage une **probabilité accrue d'acceptation du produit logiciel final** par ceux-ci (à la fois en termes d'adaptation de l'outil à l'organisation et d'appropriation de l'outil par l'organisation).

3. Pourquoi l'analyse des exigences organisationnelles est-elle un préalable important à l'informatisation ?

Dans l'ouvrage de Boehm [25], une étude sur les erreurs réalisées dans l'ingénierie du logiciel montre que les erreurs commises pendant l'IE sont les plus persistantes (parmi les erreurs faites pendant la spécification, trois quarts d'entre elles ne sont corrigées qu'après l'implantation) et les plus coûteuses à corriger (les erreurs de spécification représentent 66% du coût total de correction des erreurs). Une autre étude plus récente [1] montre que dans les projets informatiques réussis, 30% des facteurs de succès sont liés à des aspects relatifs aux exigences et à leur bonne compréhension alors que dans les projets problématiques, les questions relatives

aux exigences représentent plus de 40% des facteurs de difficulté et d'échec. Ces études montrent donc que, bien que souvent perçu comme non immédiatement rentable, l'effort porté sur la phase d'IE est effectivement « récupéré » en évitant des corrections coûteuses par la suite. De la même façon, l'implication des utilisateurs dès la phase de conception du système a maintes fois été avancé dans la littérature relative à l'innovation technologique [36] et au changement organisationnel [37] comme un facteur clé de succès [38].

Par ailleurs, les services d'urgences hospitaliers représentent un contexte d'implantation particulier pour les systèmes informatiques. La plupart d'entre eux utilisent peu l'informatique comme support à leurs activités (tout au moins pour la partie médicale). Du fait de la nature de leur activité, les futurs utilisateurs, que sont, pour l'essentiel, les médecins et les infirmiers, ne tolèrent généralement l'informatique que si elle leur permet d'exercer leur art (« *l'art de guérir dans des situations d'urgence* ») sans leur imposer des contraintes artificielles. Ils expriment généralement des exigences fortes quant à la convivialité du logiciel. Il importe donc de prendre en compte la « réceptivité » des personnels soignants en ce qui concerne l'utilisation de l'outil informatique dans le cadre de leur travail quotidien. Faute de ne l'avoir suffisamment fait, quelques projets d'informatisation des urgences ont connu de cuisants échecs qui se traduisent très concrètement par la non utilisation du logiciel coûteusement implanté⁵.

4. En quoi consiste une démarche d'analyse des exigences organisationnelles ?

La méthodologie présentée ici possède deux traits caractéristiques :

- Elle se veut **pluridisciplinaire** et combine des concepts et techniques des deux disciplines que sont la Théorie des Organisations et l'Ingénierie des Exigences du logiciel.
- Elle se veut **participative** et vise à impliquer fortement les acteurs concernés par l'informatisation de l'organisation.

⁵ Projet Assur (Appropriation de Systèmes d'Informations dans les Services d'Urgences) lié au projet ARTHUR ; voir [39] et [40]

La méthodologie préconise l'élaboration d'un cahier des charges en **quatre grandes étapes**.

4.1 Etape 1 : Analyse de l'existant

La première étape consiste en une étude de l'existant. Elle vise à étudier, comprendre et décrire l'organisation telle qu'elle existe, selon un double éclairage : Ingénierie des exigences et Théorie des Organisations. Ce double éclairage induit une analyse du domaine d'application qui se focalise sur des aspects spécifiques et produit différentes descriptions (voir figure 1).

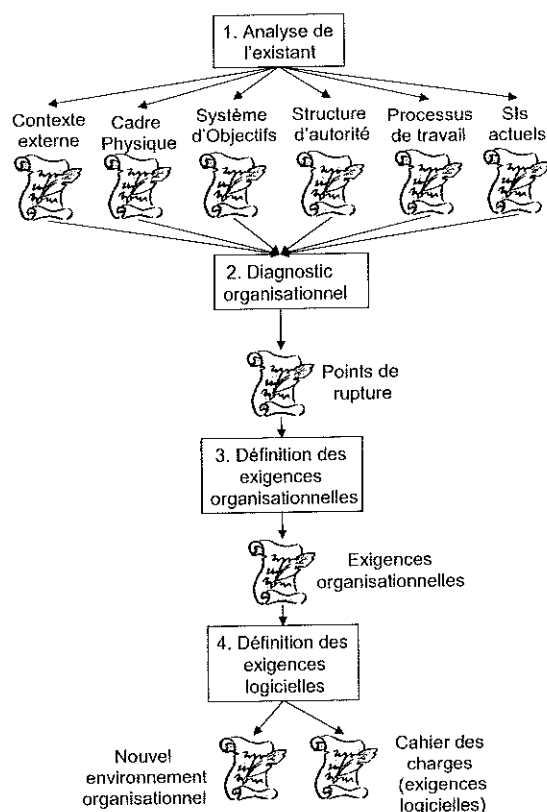


Figure 1. Une démarche d'ingénierie Organisationnelle des Exigences en quatre étapes

Une première description concerne le contexte externe de l'organisation (son environnement, ses relations avec celui-ci et les changements qui sont à l'œuvre dans celui-ci). Ensuite on s'attache à décrire le contexte interne de l'organisation et,

plus particulièrement, la structure d'autorité de l'organisation (qui décide de quoi, qui gère quelle information,...?), le cadre physique du travail (disposition et taille des locaux, localisation des systèmes d'information existants,...), les principaux processus de travail (qui fait quoi, quand , ...?), le système de buts ou d'objectifs de l'organisation (les objectifs qui sous-tendent le système actuel et qui servent à la fois à expliquer la raison de l'existence de l'organisation et son mode de structuration et de fonctionnement), les systèmes d'informations existants (supports, types d'information traitées,...).

4.2 Etape 2 : Diagnostic Organisationnel

Un diagnostic organisationnel est ensuite posé sur cet existant. Il vise à évaluer dans quelle mesure l'existant rencontre de manière satisfaisante ou non les objectifs identifiés. Ce diagnostic se traduit par l'identification de **points de rupture** dans le fonctionnement de l'existant organisationnel, c'est-à-dire d'éléments susceptibles d'être modifiés en vue de mieux répondre aux objectifs poursuivis. Ce diagnostic se base sur une évaluation tant qualitative que quantitative de la réalisation des objectifs.

4.3 Etape 3 : Définition d'Exigences Organisationnelles

A partir de ce diagnostic sont ensuite objectivées des exigences organisationnelles qui seront traitées au cours du projet en vue d'améliorer la rencontre des objectifs de l'organisation. Ces exigences organisationnelles peuvent être définies, de façon similaire à la définition donnée à la section 2, comme des conditions ou capacités qui doivent être satisfaites ou possédées par l'organisation (sa structuration, son fonctionnement, ...). Elles sont exprimées indépendamment du recours ou non à la technologie informatique et expriment des changements envisagés à l'organisation actuelle afin d'améliorer la réalisation des objectifs insuffisamment rencontrés ou d'atteindre de nouveaux objectifs.

4.4 Etape 4 : Définition de la frontière d'informatisation

Cette dernière étape consiste en la définition d'un ensemble des possibles pour l'informatisation d'une partie des activités de l'organisation. Cette étape a deux objectifs complémentaires et indissociables:

- produire une description aussi précise que possible du rôle du logiciel (exprimé sous forme d'exigences fonctionnelles ou non fonctionnelles qui lui sont

imposées);

- définir les hypothèses faites sur la nature et le comportement de l'environnement du logiciel (l'organisation) et les changements imposés à celui-ci par rapport à la situation courante.

Il s'agit ici de définir les rôles respectifs, quant à la rencontre des exigences organisationnelles et des contraintes d'utilisabilité du logiciel à développer d'une part, et de l'organisation dans laquelle ce logiciel opérera, d'autre part. Tout comme Jackson [9], nous arguons que ces types d'exigences sont tous deux importants. Les exigences de ces deux types sont en effet nécessaires si l'on veut s'assurer que les exigences et contraintes organisationnelles définies à l'étape précédente sont bien rencontrées. En effet, ni les exigences imposées au logiciel, ni les changements organisationnels ne permettront, à eux seuls, d'atteindre les objectifs. La combinaison des deux sera nécessaire.

5. Des représentations à divers niveaux de détail

La démarche en quatre étapes présentée ci-dessus peut être appliquée pour obtenir des descriptions plus ou moins détaillées des divers aspects représentés sur la figure 1. Par exemple, dans le projet ARTHUR, elle a été appliquée deux fois, lors de **deux phases principales** :

- Une première phase d'étude globale du contexte d'implantation du système informatique a permis de cerner ce que nous appelons les pistes d'informatisation. Les descriptions produites étaient relativement abstraites et ne tenaient pas compte de tous les détails des SUH étudiés.
- Une seconde phase a étudié de manière plus détaillée chaque opportunité d'informatisation identifiée au terme de la première phase. Les descriptions produites lors de cette phase sont beaucoup plus précises et incluent de nombreuses informations détaillées, par exemple sur les processus de travail étudiés.

La première phase d'étude globale poursuit le double objectif

- d'étudier et de comprendre l'organisation dans laquelle le système informatique va s'insérer (dans notre cas, le service d'urgences hospitalier, sa composition,

son fonctionnement, ses objectifs, le système de valeur de ses membres, ...)

- et d'identifier globalement les opportunités d'amélioration de l'organisation au moyen d'un ou plusieurs systèmes informatiques.

Cette première étape est à nos yeux importante à deux titres :

- tout d'abord afin de s'assurer que les informaticiens participant au projet (développeurs et ingénieurs des exigences) possèdent une bonne connaissance du domaine d'application (l'organisation cible),
- et ensuite, afin de permettre aux diverses parties prenantes de bien percevoir les besoins et opportunités d'amélioration qu'offre un système informatique. Cette démarche est d'autant plus importante que ces besoins ou opportunités sont au départ difficilement identifiables pour les informaticiens (ingénieurs des exigences) participant au projet, mais aussi pour les acteurs du domaine eux-mêmes.

Etant donné les objectifs de cette phase, il n'était pas indispensable d'y obtenir des représentations complètement détaillées des aspects étudiés. L'objectif spécifique de la deuxième phase (d'études détaillées) consiste à approfondir l'analyse de chacune des opportunités identifiées à la première étape, afin de dériver, in fine, un cahier des charges précis pour chacune d'entre-elles. Cet objectif spécifique justifiant, lui, l'utilisation de descriptions précises et détaillées. Dans tous les cas de figure, il y a lieu d'adapter le niveau de détail au contexte du projet, et en particulier à l'objectif poursuivi, en appliquant la démarche aux contraintes de temps et de budget imposées au projet.

6. Un processus participatif et créatif

Nombreux sont les recherches et travaux qui ont mis en exergue la nécessaire implication des utilisateurs dans la conception même des SI comme facteur clé de succès de ces projets informatiques (par exemple, [1,13]). La méthodologie décrite ici satisfait également ces impératifs de participation des professionnels de l'urgence. Il importe que tous les acteurs de l'organisation concernés par le nouveau système informatique soient impliqués dans le processus d'IOE. Il convient donc, avant toute chose, d'identifier les acteurs concernés et de mettre en place des mécanismes qui

leur permettent de participer à la définition des exigences.

D'une façon générale, les participants potentiels à l'activité d'IOE incluent au minimum (voir Figure 2) :

- les gestionnaires-responsables ou décideurs de l'organisation (ou à tout le moins des unités organisationnelles concernées par le projet de système informatique) : ils sont la source d'expression des objectifs de l'organisation et, plus spécifiquement, des objectifs à assigner au projet informatique. Par exemple, dans le cas d'informatisation d'un service d'urgences hospitalier, il peut s'agir des responsables du service (médecin chef de service et infirmier chef) et/ou de l'hôpital (directeur général, directeur médical ou nursing,...) .
- les utilisateurs potentiels : ils sont une des sources principales des exigences car ce sont eux qui connaissent le mieux le système actuel, ses qualités et défauts, et aussi les tâches qui sont susceptibles d'être informatisées. Il est opportun de distinguer plusieurs classes d'utilisateurs dans le cas où des groupes d'utilisateurs potentiels ont des besoins spécifiques. En fonction du logiciel à développer, on pourra, par exemple, avoir des médecins, des infirmiers, du personnel d'accueil, du personnel de secrétariat, du personnel logistique, ... mais aussi des personnes ne relevant pas du service des urgences mais échangeant des informations importantes avec celui-ci, tels que les différents services techniques d'examens et d'analyse (radio, labo,...).
- Divers spécialistes, et notamment :
 - les spécialistes en théorie des organisations : ils étudient l'organisation et ses modes de fonctionnement en détail et posent un diagnostic sur ceux-ci.
 - les ingénieurs des exigences : ils sont responsables de la collecte, de la synthèse, de la documentation et de la gestion des exigences et de leur traduction en cahiers(s) des charges
 - les ingénieurs du logiciel : ils seront responsables du développement du logiciel sur base du cahier des charges. Il importe donc qu'ils participent à la phase d'IOE afin de valider la faisabilité du logiciel défini dans le cahier des charges et de bien comprendre les exigences imposées au logiciel.
 - des juristes spécialisés dans le domaine du droit médical : connaissant les contraintes légales relatives au domaine médical, ils sont en mesure de

fournir des exigences à respecter (par exemple relatives aux droits du patient, au secret médical, aux contraintes imposées par la sécurité sociale,...).

- des spécialistes en diverses technologies dont l'utilisation est pressentie pour la construction du logiciel (par exemple, la cryptographie, les cartes à puces, la reconnaissance vocale, les réseaux sans fil, les PDAs, ...).

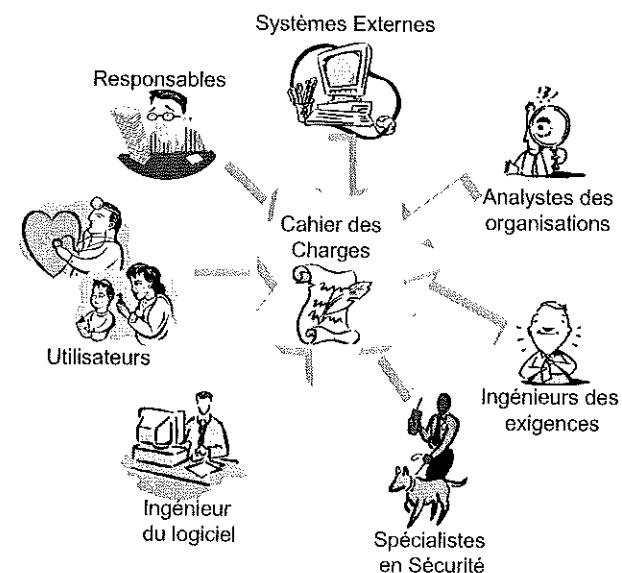


Figure 2. Les participants à l'IOE

En règle générale, il y a lieu d'impliquer dans le processus d'IOE tous les utilisateurs qui ont potentiellement une interaction avec le logiciel, ou dont le travail sera impacté par le logiciel. En particulier, si le logiciel doit avoir des interactions avec d'autres logiciels existants, le point de vue de ces logiciels devra être pris en compte. Cela implique donc la participation d'acteurs responsables de ces systèmes au processus d'IOE. Par exemple, si l'on définit un cahier des charges dans le cadre du développement d'un système logiciel pour informatiser l'envoi d'un formulaire de demandes d'analyses biologiques au laboratoire, il sera nécessaire d'impliquer des représentants du laboratoire dans la démarche d'ingénierie des exigences, à plus forte raison si le logiciel doit être interfacé avec les systèmes logiciels propres au

laboratoire. De même, comme la plupart des développements de systèmes médicaux hospitaliers se font dans le cadre d'une informatique centralisée existante, l'implication des services informatiques responsables de cette infrastructure sera donc nécessaire. Dans certains cas, il peut être intéressant d'impliquer également des représentants des patients.

Le nombre de parties prenantes peut être relativement important, ce qui peut être source de problèmes. Les souhaits, besoins et attentes exprimés par différentes personnes peuvent en effet être contradictoires et il devient alors difficile pour l'ingénieur des exigences de faire des choix parmi ces parfois nombreuses exigences contradictoires. Il est opportun dans ce cas d'élaborer une structure de représentation et de participation de ces parties prenantes, avec des règles de décision claires (par exemple par vote) pour le traitement des exigences contradictoires et la gestion des priorités entre exigences. Nous préconisons le recours aux conférences de consensus pour ce faire.

7. Comment élaborer des modèles de qualité?

La démarche de l'IOE présentée ci-dessus implique, à chacune de ses étapes, la création de plusieurs représentations (modèles) de différents aspects importants apparus durant l'IOE, schématisés à la figure 1, à savoir une représentation de divers aspects du système existant (son contexte externe, son système d'objectif, ...), la description de points de rupture, les exigences organisationnelles, les exigences logicielles (cahier des charges) et le nouvel environnement organisationnel.

Afin d'élaborer chacune de ces descriptions, un micro-processus itératif d'interaction avec les parties prenantes est nécessaire à chaque étape. Les activités de ce micro-processus de l'IOE sont représentées sur la figure 3. Elles sont ensuite décrites plus en détail dans les sections suivantes.

7.1 L'élicitation

L'activité d'élicitation a pour objectif d'acquérir auprès des parties prenantes l'information nécessaire lors de la phase d'IOE. Elle peut avoir recours à diverses

techniques, dont :

- des entretiens : les entretiens [34] avec les parties prenantes peuvent être plus ou moins structurés, en fonction du type d'information désiré. Il peut s'agir d'entretiens individuels ou collectifs qui peuvent être ouverts (discussion libre, brainstorming) ou plus ou moins dirigés (sur base de questionnaires ou check-lists préétablis).
- des observations : l'analyste observe les procédures existantes de travail et de traitement des informations en interférant le moins possible avec elles⁶. L'utilisation d'enregistrements audio ou vidéo constitue également une approche intéressante étant donné la rémanence qu'ils procurent.
- l'analyse documentaire : l'analyste peut également étudier divers types de documents en vue d'acquérir de la connaissance utile à l'IOE. Par exemple : des documents décrivant les procédures de travail actuellement utilisées, des textes légaux, des standards, ...
- l'analyse de logiciels existants : les logiciels actuellement utilisés dans l'organisation sont une source évidente d'information à la fois pour une analyse de l'existant et pour la définition des exigences. Des logiciels existant sur le marché et fournissant des fonctionnalités similaires peuvent également servir de source à la définition des exigences par analogie.

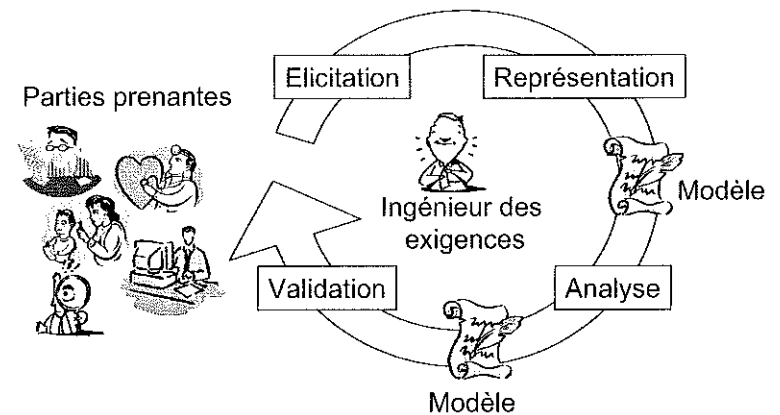


Figure 3. Le cycle d'élaboration de modèles

⁶ "...l'observation consiste à se trouver présent, mêlé à une situation sociale pour l'enregistrer et l'interpréter en s'efforçant de ne pas la modifier », [35 : p.5]

7.2 La représentation

L'activité de représentation consiste à consigner l'information acquise lors de l'élicitation dans des modèles explicites. Les modèles sont exprimés dans un ou plusieurs langages. Un **langage** fournit un ensemble de symboles et une grammaire définissant des règles d'utilisation et de composition valides de ces symboles pour former des modèles.

On distingue plusieurs **catégories de langages** en fonction de leur degré de précision (formalité):

- Les langages naturels (par exemple, le français)
- Les langages graphiques ou semi-formels : ils fournissent un ensemble limité de concepts associés à des symboles graphiques. Par exemple, les schémas Entités/Association ou les diagrammes de cas d'utilisation d'UML (use case diagrams).
- Les langages formels : en plus de fournir un ensemble de concepts limités, la sémantique de ces langages (la signification des concepts qu'ils fournissent) est définie de manière rigoureuse sur base de formalismes mathématiques. Par exemple, la logique des prédicats du premier ordre, les réseaux de Petri ou les diagrammes état-transitions (state charts).

Chacune de ces catégories de langages possède ses avantages et ses inconvénients. Il y a donc lieu de faire un **choix du langage utilisé** pour définir les modèles, en fonction des caractéristiques souhaitées dans une situation particulière.

On peut par exemple comparer les langages en fonction des **critères** suivants :

- L'ambiguïté : c'est la possibilité d'interprétation (de compréhension) de manière différente d'un modèle par des lecteurs différents (ou par un seul lecteur à des moments différents). Le langage naturel est le moins précis et donc le plus ambigu. Les langages semi-formels sont plus précis car ils ne permettent l'expression que d'un nombre plus restreint de modèles et que leur sémantique (la signification des concepts qu'ils fournissent) est généralement définie de manière rigoureuse. Les langages formels eux ne laissent aucune place à des divergences d'interprétation.
- Le caractère naturel : c'est la facilité qu'a le lecteur d'exprimer ou de comprendre un modèle. Les modèles semi-formels graphiques sont généralement les plus

naturels et sont parfois préférés au langage naturel (un petit dessin vaut mieux qu'un long discours). Les langages formels eux sont en général moins accessibles pour des non-spécialistes.

- L'expressivité : c'est le degré de couverture par le langage de l'ensemble des choses que l'auteur du modèle veut exprimer. Le langage naturel est le plus expressif. Il ne possède pratiquement pas de limite à ce qui peut être exprimé. L'expressivité des langages semi-formels et formels dépend des concepts qu'ils renferment mais est toujours plus limitée que celle du langage naturel.
- La possibilité d'analyse et de traitement automatique : plus un langage est formel, plus il permettra un traitement automatisé important des modèles. Par exemple, un modèle graphique pourra être analysé pour vérifier s'il respecte les règles de grammaire qui lui sont imposées. Les langages formels sont ceux qui permettent le plus d'analyses et de traitements. Il est par exemple possible avec certains langages de réaliser une simulation en exécutant le modèle.

Certaines approches tentent de combiner les avantages de plusieurs catégories de langages. Par exemple, des approches basées sur des structures prédéfinies de textes en langage naturel tentent de conserver l'avantage de l'expressivité tout en introduisant plus de structure rendant le traitement automatique plus aisé. C'est le cas, par exemple, des canevas de cahiers des charges [11,18,19] ou encore des structures de description de cas d'utilisation (scénarios) [20].

Certains langages formels sont associés à des symboles graphiques afin de les rendre plus naturels et compréhensibles (par exemple, les réseaux de Petri ou les diagrammes état/transition). En pratique, vu l'effort de formalisation important requis par l'utilisation des langages formels, on en limitera souvent l'utilisation à certains types de systèmes (dits critiques) qui nécessitent un degré de sûreté élevé, et dans lesquels l'ambiguïté doit être évitée. Par exemple, un logiciel de contrôle d'un appareillage médical dont le dysfonctionnement serait potentiellement dangereux pour le patient (comme un pousse-seringue qui administre automatiquement un médicament à un patient) se prêtera bien à l'utilisation d'un langage formel.

7.3 L'analyse de la représentation / modèle

Une fois représentée dans un modèle, l'information élicitée par l'ingénieur des

exigences peut être analysée afin de s'assurer d'un certain nombre de qualités intrinsèques du modèle. Les analyses peuvent être plus ou moins automatisées en fonction du degré de formalité du langage. Un modèle en langage naturel ne permettra que peu d'analyses et sera généralement traité manuellement.

Les **qualités visées dans l'établissement d'un modèle** sont notamment [3]:

- La correction syntaxique : le modèle doit être syntaxiquement bien formé (les « phrases » qui composent le modèle doivent respecter les règles propres au langage);
- La cohérence : le modèle ne contient pas de contradictions;
- La complétude : tous les éléments utilisés dans le modèle sont définis de manière complète;
- La non-ambiguïté : le modèle est précis et ne peut être interprété de manière incorrecte;
- La non-redondance : le modèle devrait éviter de répéter plusieurs fois les mêmes éléments ou de fournir explicitement des éléments qui peuvent être dérivés à partir d'autres.

7.4 La validation de la représentation/modèle

La validation est une étape importante de l'IOE. Elle consiste à s'assurer que le modèle réalisé représente de manière juste et complète l'information exprimée par les parties prenantes. Elle vise à s'assurer de **l'adéquation entre le modèle et la réalité**.

Pour ce faire, **plusieurs techniques** peuvent être employées, comme par exemple :

- la présentation des modèles aux parties prenantes : les modèles sont exposés aux parties prenantes. En fonction du type de langage utilisé pour écrire le modèle et de la connaissance des parties prenantes, un paraphrasage (ré-expression du modèle dans un langage plus naturel) peut être nécessaire;
- le prototypage : une maquette du système informatique est élaborée sur base du modèle défini jusqu'ici et est présentée à l'utilisateur pour obtenir un retour d'information sur son adéquation par rapport à ses besoins;
- l'animation ou la simulation : certains modèles écrits dans des langages formels peuvent être interprétés ou exécutés de telle manière qu'il est possible de

montrer aux parties prenantes des exemples de comportements du système. Il peut alors être demandé aux parties prenantes de confirmer ou d'infirmer la possibilité de tels comportements dans le système réel.

8. Quel doit être le contenu général d'un cahier des charges pour un système informatique de support à l'activité d'un service d'urgences ?

Un cahier des charges typique pour un système d'information destiné aux SUH doit contenir un certain nombre d'informations. Nous en donnons ici une liste non exhaustive et une description sommaire :

- les exigences fonctionnelles : elles concernent les fonctionnalités du logiciel à développer, ce qu'il doit faire ou permettre de faire, les services rendus aux utilisateurs. Par exemple, permettre la gestion des informations relatives au « passage » d'un patient au service d'urgences, imprimer des étiquettes, envoyer une demande d'analyses au laboratoire,...;
- les contraintes légales et de certification : les lois et règlements à respecter lors de la mise en place de logiciels de soins de santé. Ceux-ci incluent des recommandations comme celles élaborées par la Commission Normes en matière de Télématique au service du secteur des Soins de Santé [21], des contraintes légales (voir le chapitre 2), des normes de certification applicables aux logiciels médicaux (par exemple [27,28]), ...;
- les mécanismes de sécurité à mettre en œuvre : le cahier des charges doit décrire les différents niveaux de sécurité nécessaires et les mécanismes de contrôle d'accès à mettre en place (identification par mot de passe, authentification par carte à puces, biométrie,...) pour permettre et restreindre l'accès aux fonctionnalités et données aux différentes catégories de personnes adéquates;
- les contraintes socio-économiques ayant une influence sur les fonctionnalités du logiciel. Par exemple, les contraintes imposées par des organismes publics tels que l'INAMI pour autoriser le remboursement de prestations;
- les contraintes d'ergonomie du logiciel : la nature du travail médical peut imposer des contraintes fortes sur les interfaces homme-machine du logiciel (possibilité de travail en position debout, écrans tactiles, nombre d'interactions réduit, utilisation de PDAs, ...);

- les contraintes de traçabilité des actes et modifications : ces exigences portent sur la nature des actes et des utilisations du logiciel (actes médicaux, infirmiers, gestion du dossier patient, ...) qui doivent pouvoir être retracés (qui, quand, où, pourquoi, ...);
- les contraintes de disponibilité et de robustesse (24h/24, permanence et disponibilité de l'information, procédures de récupération en cas d'incident, résistance à la charge de travail, ...)
- les contraintes d'efficacité (temps de réponse, ...);
- les exigences en terme d'ouverture du logiciel et d'interfaçage avec d'autres logiciels;
- les éventuels standards à respecter (standards de structuration de dossier patient, de communication, ...).

9. Conclusions

Ce chapitre décrit l'activité d'élaboration de cahier(s) des charges pour un système d'information informatisé relatif aux SUH. Cette activité importante doit être menée avec soin si l'on veut garantir le succès du projet d'informatisation. Nous avons décrit une démarche à suivre, basée sur de bonnes pratiques, ainsi que des techniques d'acquisition, de représentation, de traitement et de validation des exigences utiles pour cette démarche. La démarche proposée présente l'originalité de combiner les techniques usuelles d'Ingénierie des Exigences avec les apports de l'Analyse Organisationnelle.

L'intérêt majeur de cette façon de procéder réside dans le caractère plus global de l'approche, ne se limitant pas aux aspects fonctionnels du (des) logiciel(s) à concevoir. Ce dépassement est, entre autres, permis par l'implication forte des utilisateurs dans l'identification des exigences organisationnelles mais aussi par la prise en compte de leurs motivations et habitudes de travail. Cette façon de faire permet donc de prendre en considération les besoins et attentes des utilisateurs, tout en les aidant à les formuler explicitement. Par ailleurs, cette méthodologie permet également de ne pas « se contenter » d'informatiser l'existant mais bien de l'améliorer, en accord avec les principaux intéressés, et de le compléter par un « regard externe » en termes de diagnostic organisationnel. De la sorte, il est possible de poursuivre conjointement un

objectif dit « technologique » de performance de l'outil informatique et un objectif plus « social » d'appropriation par les professionnels de l'urgence concernés.

Bien que celle démarche soit le plus souvent réalisée par des experts/consultants externes à l'organisation, il est tout à fait possible qu'une partie de la démarche soit entreprise par les acteurs de l'organisation elle-même. L'objectif de la présentation faite ici est de permettre aux décideurs d'un projet d'informatisation de maîtriser les enjeux de cette démarche. Il nous semble en effet que, même si la démarche est dirigée par des experts/consultants extérieurs à l'organisation, les décideurs doivent rester attentifs à un certain nombre de préoccupations identifiées dans cet ouvrage, dont notamment :

- l'implication de l'ensemble des parties prenantes au processus (décideurs, utilisateurs, services externes concernés, patients, ...);
- le sérieux de la démarche et des outils utilisés pour l'élaboration du cahier des charges (techniques et effort d'élicitation, de représentation, d'analyse et de validation des descriptions produites);
- la qualité du cahier des charges final qui doit faire l'objet d'un accord entre le « client » (le commanditaire du logiciel à développer) et le « fournisseur » (le développeur et fournisseur du logiciel); le niveau de qualité atteint est souvent le résultat d'un compromis entre la qualité désirée et l'effort consacré à son élaboration;
- la complétude du cahier des charges en prenant en compte non seulement des aspects fonctionnels mais également de nombreuses contraintes notamment légales et d'utilisabilité (efficacité, ergonomie, ...);
- l'utilité et l'importance de la représentation explicite (dans des modèles) de différents aspects de l'organisation et de leur utilisation dans une démarche de qualité visant non seulement à l'introduction d'un logiciel mais également à l'amélioration continue de l'organisation. Il importe par conséquent également que l'organisation elle-même reste maître de ces modèles c'est-à-dire qu'elle y ait un accès aisé et puisse les utiliser dans le futur, indépendamment de l'intervention ponctuelle des experts/consultants et de l'identité des experts/consultant .

Références

- [1] Standish Group Int'l, "The CHAOS report",
http://www.pm2go.com/sample_research/chaos_1994_1.asp

- [2] **Richard H. Thayer and Merlin Dorfman (eds)**, 1990, *System and Software Requirements Engineering*, IEEE Computer Society Press
- [3] **Pericles Loucopoulos and Vassilios Karacostas**, *System Requirements Engineering*, 1995, Mc Graw-Hill Book Company, ISBN 0-07-707843-8.
- [4] **A. De Vos, Cl. Lobet-Maris, A. Rousseau**, 2002, *From coordination to loyalty. Developing a constructivist perspective on organisational and technological changes*, 18th EGOS colloquium, Barcelona, 6-8 July
- [5] **M.J. Hatch**, 2000, *Théorie des organisations*, De Boeck Université
- [6] **W.J. Orlikowski, S.R. Barley**, *Technology and Institutions : What Can Research on Information Technology and Research on Organizations Learn from Each Other ?*, MIS Quarterly, vol. 25, n°2, June, pp. 145-165
- [7] **Eric S. K. Yu**, 1994, *Modelling Strategic Relationships for Process Reengineering*, PhD thesis, Computer Science Department, University of Toronto (Canada).
- [8] i* web site, <http://www.cs.toronto.edu/km/istar/>, January 2003.
- [9] **Michael Jackson**, 1995, *Software Requirements & Specifications, a lexicon of practice, principles and prejudices*, Addison-Wesley, ISBN 0-201-87712-0.
- [10] **Kevin Forsberg and Harold Mooz**, *System Engineering Overview*, In [THA90a], pp. 44-72.
- [11] European Space Agency, *ESA software engineering standards - Issue 2*, ESA PSS-05-0 Issue 2, February 1991, ISSN 0379-4059.
- [12] IFIP-IFAC TF on Architectures for Enterprise Integration, GERAM: Generalised enterprise reference architecture and methodology. Version 1.6.3, Available at <http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/>, March 1999.
- [13] **Karen L. McGraw and Karan Harbison**, 1997, *User-Centered Requirements*, Laurence Erlbaum Associates.
- [14] **H. Mintzberg**, 1982, *Structure et dynamique des organisations*, Ed. d'Organisation, Paris.
- [15] **S. Ghernaoui-Hélie**, 1998, *Stratégie et ingénierie de la sécurité des réseaux*, InterEditions
- [16] **L. Dusserre, H. Ducrot et F-A. Allaert**, 1999, *L'information médicale. L'ordinateur et la loi*, 2^{ième} éd., Editions Médicales Internationales
- [17] **J-F. Chanlat**, *L'analyse des organisations, un regard sur la production de langue française contemporaine (1950-1990)*, Cahier de recherche HEC Montréal, n°91-09, Mar. 1991.
- [18] IEEE Recommended Practice for Software Requirements Specification, IEEE Std 830-1998, IEEE, New York, 1998, also appears in [2]
- [19] **James Robertson and Suzanne Robertson**, *Volere Requirements Specification Template*, Edition 8, Atlantic Systems Guild Limited, <http://www.systemsguild.com/GuildSite/Robb/Template.html>
- [20] **Alistair Cockburn**, 2001, *Writing Effective Use Cases*, The Agile Software Development Series, Addison Wesley.
- [21] Service Public Fédéral Santé publique, Sécurité de la Chaîne alimentaire et Environnement, Commission Normes en matière de Télématique, au service du secteur des Soins de Santé, site web : <http://www.health.fgov.be/telematics/>

- [22] **Martin Fowler with Kendall Scott**, 1999, *UML Distilled, A brief guide to the standard object modeling language*, Second edition, Object Technology series, Addison-Wesley.
- [23] **Grady Booch, James Rumbaugh and Ivar Jacobson**, 1999, *The unified modeling language user guide*, Object Technology series, Addison-Wesley.
- [24] Object Management Group, UML website, <http://www.omg.org/technology/uml>
- [25] **Boehm, B.**, 1981, *Software Engineering Economics*, Prentice-Hall Inc..
- [26] **Ivar Jacobson, Grady Booch, James Rumbaugh**, 1999, *The Unified Software Development Process*, Addison-Wesley Longman, ISBN 0-20-157-169-2
- [27] Législation communautaire, Directive 93/42/CEE du Conseil, du 14 juin 1993, relative aux dispositifs médicaux, Journal officiel n° L 169 du 12/07/1993, Document 393L0042, pp. 0001 - 0043
- [28] IEC SC 62A (COMMON ASPECTS OF ELECTRICAL EQUIPMENT USED IN MEDICAL PRACTICE), IEC 60601-1 - Ed. 2.0: Medical electrical equipment - Part 1: General requirements for safety, December 1988.
- [29] IEEE Standard Glossary of Software Engineering Terminology (IEEE Std610.12-1990), IEEE Computer Soc., Dec. 10, 1990.
- [30] **A. Rousseau and M. Petit**, 2003, *Enriching a Requirements Elicitation Methodology with an Organisational Sciences perspective: An Application in the Health Care domain*, technical report.
- [31] **J. Nizet, F. Pichault**, 1995, *Comprendre les organisations*, De Boeck.
- [32] **S. Ghernaoui-Hélie**, 1998, *Stratégie et ingénierie de la sécurité des réseaux*, InterEditions
- [33] **L. Dusserre, H. Ducrot et F-A. Allaert**, 1999, *L'information médicale. L'ordinateur et la loi*, 2^{ième} éd., Editions Médicales Internationales
- [34] **Albarelo, L., Digneffe, F., Herniaux, J.-P., Maroy, C., Ruquoy, D., de Saint-George, P.**, 1995, *Pratiques et méthodes de recherche en sciences sociales*, Paris, Armand Colin
- [35] **H. Peretz**, 1998, *Les méthodes en sociologie : l'observation*, La Découverte
- [36] **M. AKRICH, M. CALLON et B. LATOUR**, *A quoi tient le succès des innovations ? Premier épisode : l'art de l'intéressement*, in Annales des Mines, Gérer & Comprendre, juin 1988, pp. 4-17
- [37] **H. BARKI, S. RIVARD et J. TALBOT**, 1992, *Risque, mode de gestion et succès d'un projet d'informatisation*, in Technologies de l'Information et Société, vol. 5, n°2, pp. 121-146
- [38] **E. MUMFORD**, 1983, *Participative System Design : Practice and Theory*, in journal of Occupational Behavior, vol. 4, pp. 47-57
- [39] **Dumont V., Rousseau A.**, *How a new information system becomes a practical convention? Medical Information System in Hospital Emergency Departments*, communication at the 19th EGOS Conference (European Group for Organizational Studies), Copenhagen, July 2003, unpublished
- [40] **Dumont V., Rousseau A.**, *Integrating Patient Information Systems in Hospital Emergency Departments: a recipe to success?* Communication at the 9th iSHIMR (international Symposium on Health Information Management Research) , Sheffield, June 2004, proceedings to be published