

# CHAPITRE 3

## INGENIERIE DES EXIGENCES

L'analyse des besoins est vue comme l'une des étapes cruciales dans la conception et le développement des systèmes logiciels car elle aborde le problème critique de concevoir le bon logiciel pour le client. Cette activité est plus difficile, voire complexe, pour le développement des systèmes ontogénétiques car l'analyse d'une situation courante est moins problématique que celle du futur. Néanmoins, il existe des approches prometteuses qui ont participé à réduire l'écart entre l'ingénierie des besoins des systèmes logiciels et la prise en compte des évolutions anticipées dans le cycle de développement.

Nous nous intéressons dans ce chapitre à la présentation de l'ingénierie des besoins et de quelques approches représentatives de cette phase.

### 3.1 L'ingénierie des Besoins : Une Etape Clé

---

Actuellement les systèmes logiciels sont devenus plus complexes, et leur développement est plus difficile et par conséquent leur évolution devient aussi fastidieuse. La mesure primaire du succès d'un système logiciel est le degré auquel il réalise le but pour lequel on l'a construit. Le processus d'ingénierie des Besoins est un ensemble de tâches bien plus complexe que la simple production en début de projet d'un document spécifiant le système à construire. Définir les besoins pour un logiciel est un processus qui fait appel à des compétences humaines, techniques et méthodologiques très variées, alliant rigueur et créativité. Une étude efficace des besoins réduit très sensiblement le coût du développement et de la maintenance d'une application et accroît sa qualité [Aur 05].

Dans les méthodes d'ingénierie des exigences, deux types d'exigences peuvent être identifiés. Les exigences fonctionnelles qui spécifient les fonctions que le système à concevoir doit être en mesure d'effectuer. Les exigences non-fonctionnelles qui capturent les propriétés ou les contraintes sous lesquelles le système à concevoir doit fonctionner, telles que les aspects de performance, de qualité ou de sécurité. Les pratiques industrielles actuelles consistent à spécifier les exigences fonctionnelles dès les premières phases de développement logiciel et à laisser la prise en compte des exigences non-fonctionnelles au

niveau de l'implémentation. Ce choix est souvent justifié par l'énorme pression (en termes de temps) pour le déploiement rapide d'un prototype du logiciel [Cys 03].

## 3.1.1 Définitions

---

L'expression "Requirements Engineering" semble avoir été utilisée pour la première fois par [Hag 88]. Une traduction exacte devrait être "ingénierie des exigences" mais nous suivons l'usage en utilisant "ingénierie des besoins".

Tous les projets commencent par l'étape des besoins. Les besoins décrivent ce qu'un produit logiciel doit exécuter. Un besoin se rapporte typiquement à un certain aspect d'un nouveau produit ou un service ajouté [Aur 05].

Dans les sciences informatiques, le terme "ingénierie des besoins" est à la fois utilisé pour désigner un thème de recherche, pour référencer une activité du processus de développement de systèmes logiciels ainsi que pour référencer un "processus" de construction d'une spécification des besoins. Plusieurs définitions du terme "ingénierie des besoins", que nous énumérons ci-dessous, ont été introduites dans la littérature :

Le standard IEEE 610.12-1990 [ANS 90] définit un besoin comme étant :

« Une condition ou capacité nécessaire à un utilisateur pour résoudre un problème ou atteindre un objectif », ou comme étant « une condition ou capacité qui doit être reconnue ou possédée par un système pour satisfaire un contrat, un standard, une spécification ou d'autres types de documents formellement imposés ».

On trouve dans [Zav 93] une des définitions les plus précises de l'ingénierie des besoins :

«L'ingénierie des besoins est la branche du génie logiciel concernée par des buts du monde réel pour des fonctions des systèmes logiciels et des contraintes sur ces derniers. Elle est également concernée par la relation de ces facteurs pour préciser des spécifications du comportement du logiciel et leur évolution avec le temps et à travers des familles de logiciels».

Cette définition est attrayante pour certaines raisons. D'abord, elle accentue l'importance sur « les buts du monde réel » qui motivent le développement d'un système logiciel. Puisqu'elle représente aussi bien le "Pourquoi" que le "Quoi" d'un système (Figure 3.1). Par ailleurs, elle se rapporte « à des spécifications précises ». Celles-ci constituent la base d'analyse des besoins, validant ce que les parties prenantes veulent, définissant ce que les concepteurs doivent construire, et vérifiant qu'ils ont fait le travail correctement à la livraison.

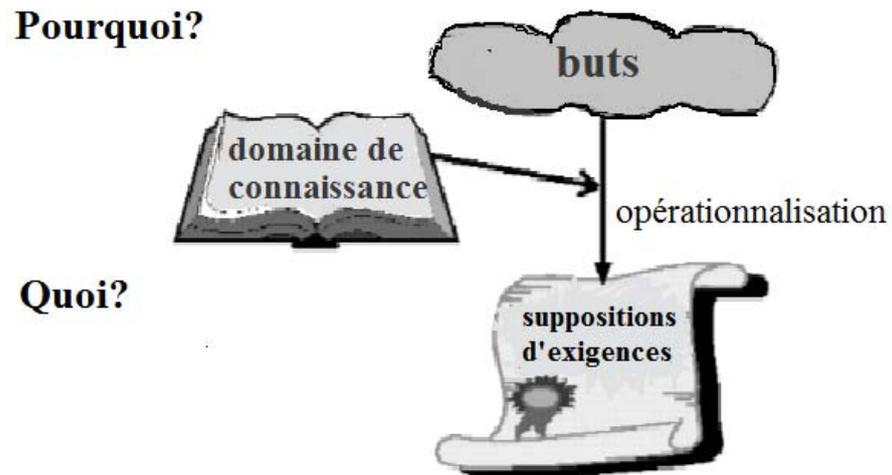


Figure 3.1 Le fondement de l'ingénierie des besoins [Rol 03]

## 3.1.2 Classification des Exigences

La littérature établit une distinction entre différents types de besoins, dans la pratique il n'est pas toujours facile d'identifier de telles différences [Ber 98]. Idéalement les besoins sont indépendants de la conception, montrant le "Quoi" que le système logiciel doit faire, plutôt que le "Comment" il doit le faire. Les besoins peuvent être classés dans plusieurs directions, comme illustré dans la table 3.1 [Aur 05].

Classification	Classe des besoins	Eclaircissement
1	Besoins fonctionnels	Ce que le système doit faire comme fonctions
	Besoins non-fonctionnels	Contraintes sur les types de solutions qui mènent les besoins fonctionnelles. Par exemple : précision, performance, sécurité et fiabilité
2	Besoins niveau but	Relié aux buts métier
	Besoins niveau domaine	Relié à l'espace du problème
	Besoins niveau produit	Relié au produit
	Besoins niveau conception	Quoi construire
3	Besoins primaires	Elicités à partir des parties prenantes
	Besoins dérivés des besoins primaires	Elicités à partir des besoins primaires
4	Besoins métiers	Besoins liés à l'activité de l'entreprise (son métier)
	Besoins processus	Comment les gents interagissent avec le système
	Besoins basés rôle	Par exemple: besoins consommateur, besoins utilisateurs, besoins IT, besoins système, et besoins de sécurité

Table 3.1 Classification des besoins

## 3.1.3 Activités de l'Ingénierie des Besoins

On trouve dans [Lam 09] et [Nus 00] une description des activités qui caractérisent une démarche d'ingénierie des exigences. Nous les présentons dans ce qui suit.

### ✦ La compréhension du domaine

Cette activité consiste en l'identification des parties prenantes à l'aide d'interviews en étudiant aussi l'environnement du système. Les problèmes du système actuel sont ainsi découverts et les possibilités d'amélioration sont analysées.

### ✦ L'élicitation des exigences

Cette activité consiste à découvrir les exigences et les hypothèses candidates pour le système en cours en se basant sur les faiblesses du système actuel, extraites dans l'activité de la compréhension du domaine. Plusieurs techniques peuvent être employées pour avoir les informations appropriées telles que les interviews avec les parties prenantes, les scénarios, les questionnaires, etc.

Nous avons sélectionné un noyau de huit techniques et approches qui offrent une couverture appropriée à travers la gamme de techniques et approches disponibles (par exemple **Ethnographie** comprend l'observation, et **JAD** est un exemple de travail actuel en groupe), et qui représentent à la fois l'état de l'art et l'état de la pratique.

La table suivante, tirée de [Aur 05], montre dans quelle phase les techniques d'élicitation peuvent être utilisées.

	Interview	Domaine	Groupwork	Ethnographie	Prototypage	Buts	Scenarios	Point de vue
<b>Compréhension du domaine</b>	X	X	X	X		X	X	X
<b>Identification des sources des besoins</b>	X	X	X			X	X	X
<b>Analyse des parties prenantes</b>	X	X	X	X	X	X	X	X
<b>Sélection des techniques et approches</b>	X	X	X					
<b>Extraction des besoins</b>	X	X	X	X	X	X	X	X

**Table 3.2 Techniques et approches pour les activités d'extraction [Aur 05]**

#### ✦ **L'évaluation des exigences**

Le but de cette activité est d'évaluer les exigences afin de choisir les meilleures alternatives, en se basant par exemple sur l'analyse des risques et des conflits entre les exigences.

#### ✦ **La spécification et la documentation des exigences**

Cette activité consiste à détailler, structurer et documenter les caractéristiques acceptées du système futur telles qu'elles ressortent de l'activité d'évaluation. Le produit obtenu est une première version du cahier des charges qui est écrit en langage naturel, semi-formel (des diagrammes) ou formel.

#### ✦ **L'assurance de la qualité des exigences**

Le cahier des charges obtenu à l'issue de l'activité précédente doit être soigneusement analysé et validé avec les parties prenantes afin de repérer les insuffisances (incohérences, omissions, contradictions ou ambiguïtés) par rapport aux besoins réels. Diverses techniques peuvent être utilisées telles que les reviews, l'animation, etc. Le résultat final de cette activité est un cahier des charges consolidées.

#### ✦ **La gestion de l'évolution des exigences**

Elle consiste à se préparer aux changements que le système futur peut subir. Cette anticipation d'évolution doit prendre en compte tous les aspects (les exigences, les hypothèses sur l'environnement, etc.) en utilisant diverses techniques telles que la gestion de la traçabilité, les changements à la volée, etc.

## 3.1.4 Importance de l'Ingénierie des Besoins

---

Le processus d'ingénierie des besoins tient une place importante à l'intérieur du processus de développement des systèmes logiciels. Le développement des spécifications des besoins de logiciel est largement identifié comme la base de la fonctionnalité du système. Les besoins d'un logiciel sont les causes déterminantes critiques de la qualité de ce logiciel. Des études empiriques prouvent que les erreurs dans les besoins sont les plus nombreuses pendant le cycle de vie de logiciel, elles sont aussi les plus chères et les plus longues à corriger. Le processus d'ingénierie des besoins tient une place importante à l'intérieur du processus de développement de systèmes logiciels.

Par exemple la plupart des problèmes de développement de système d'information se produisent lorsque le processus d'ingénierie de besoins n'est pas respecté (Figure 3.2). Fransiskus Adikara et al. ont catégorisé de nombreux risques des utilisateurs liés au processus d'exigences, à savoir: le risque de la connaissance, le risque de volatilité des exigences et les risques de la documentation [Adi 13].

- Le risque de connaissance se compose des risques et des problèmes qui se produisent dans le développement du système. C'est parce que les parties prenantes, utilisateurs ou les développeurs ont un manque de connaissances dans le processus du génie logiciel, en particulier dans la méthodologie d'ingénierie des exigences, modèles et techniques, ainsi que le support d'information nécessaire sur les exigences non fonctionnelles. Les parties prenantes, les utilisateurs et les équipes de développeurs doivent tirer partie de leurs connaissances et s'appuyer sur l'ingénierie du logiciel et l'ingénierie des exigences pour développer un nouveau système. Le manque de connaissances des utilisateurs dans le génie logiciel constitue un risque lié à la connaissance, etc. [Arm 05]. Aussi les auteurs ajoutent dans [Adi 13] que les risques dans les exigences non-fonctionnelles peuvent entraîner des risques de sécurité et des risques légaux.
- La volatilité des exigences se réfère à des ajouts, des suppressions et des modifications des besoins durant le processus de développement des systèmes [Kas 10]. Les risques de la volatilité des exigences est un problème qui se produit lorsque les conditions changent dans la relation entre les parties prenantes, les utilisateurs et les développeurs.
- Les risque de la documentation se produit quand les développeurs et les utilisateurs ne se servent plus des moyens et des outils de stockage appropriées pour documenter la progression du processus des exigences [Mea 05]. Il en résulte une perte d'attributs, de la difficulté à tracer, planifier et hiérarchiser des tâches dans les activités du processus d'analyse des besoins.

Des études ont effectivement mis en évidence l'échec du développement d'applications de taille importante dû à une définition des besoins « inadéquate » [Boe 81].

Les erreurs commises lors du processus d'ingénierie des besoins peuvent avoir de très lourdes conséquences pour les activités se trouvant en aval et donc pour le système logiciel résultant. La Figure 3.3 montre l'impact qu'une erreur faite lors du processus d'ingénierie des besoins peut avoir sur les activités de conception, d'implémentation et de test [Dav 93].

L'expérience a montré qu'une erreur de spécification aussi infime soit-elle détectée en cours d'exploitation, peut non seulement entraîner des surcoûts de correction considérables, mais aussi provoquer des dégâts dont les conséquences sont parfois irréversibles sinon dramatiques comme la perte de vie humaines dans les systèmes critiques.

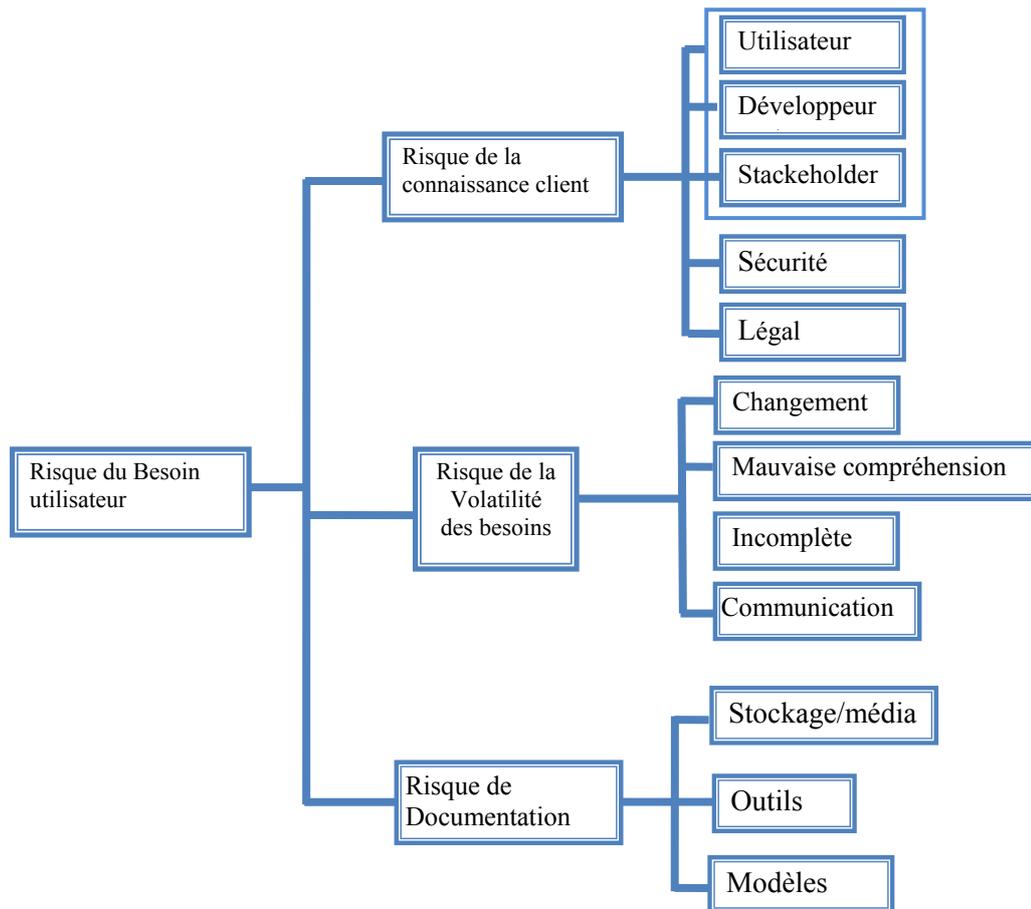


Figure 3.2 Classification des risques des besoins utilisateurs [Adi 13]

## 3.2 Les Approches pour l'Ingénierie des Besoins

La littérature propose plusieurs approches d'ingénierie des exigences. Chacune d'entre elles se concentre sur des activités spécifiques du processus d'ingénierie des exigences. Dans ce qui suit, nous présentons brièvement quelques approches :

- L'approche à base de scénarios [Sut 03], [Rol 98]

Dans ce type d'approche, les exigences sont décrites à l'aide de scénarios. Ces derniers sont des descriptions du monde réel qui sont exprimées à l'aide du langage naturel, de diagrammes, etc.

- L'approche GORE (Goal Oriented Requirements Engineering)

L'approche GORE se base sur la définition des exigences comme étant des buts qui peuvent être divisés et raffinés [Lam 09].

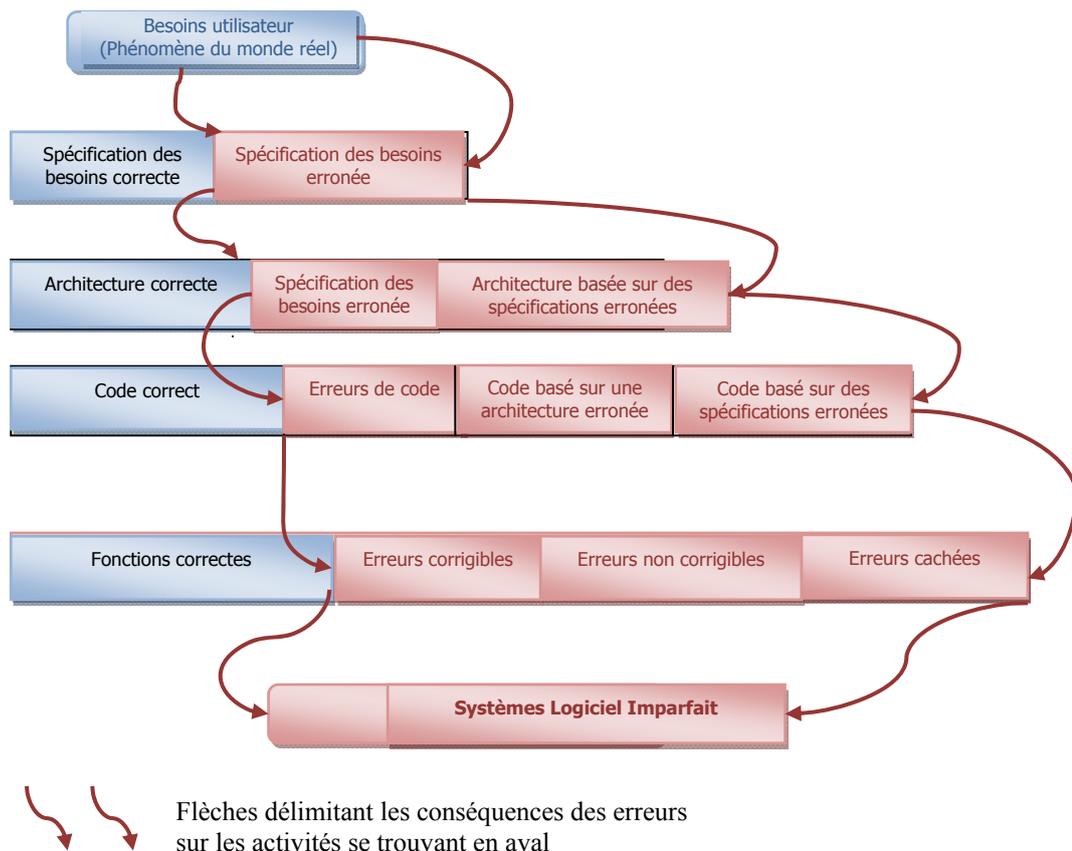


Figure 3.3 Effets des erreurs faites lors de l'ingénierie des besoins sur le reste du cycle de vie [Dav 93]

- L'approche orientée-aspects [Ras 08]

Cette approche reconnaît explicitement l'importance de bien identifier, de traiter et de séparer des préoccupations transversales qui ont été par ailleurs réparties dans des artefacts d'autres exigences (des cas d'utilisation, des modèles de buts, etc.).

- L'approche « Problem Frames » (schéma de problème) [Jac 01]

C'est une approche qui permet de structurer l'analyse des exigences d'un logiciel et de concevoir une solution logicielle. Elle aide à comprendre d'une part le contexte dans lequel réside le problème et d'autre part les aspects pertinents pour la conception d'une solution. Cette approche s'intéresse surtout aux exigences fonctionnelles.

Dans la dernière décennie, la popularité de l'approche GORE a augmenté et cela est dû au fait que son processus d'élaboration se termine là où la plupart des autres approches d'ingénierie des exigences débutent [Lam 02]. Les approches GORE se concentrent beaucoup sur les activités qui précèdent la formulation des exigences [Nus 00]. En effet, la plupart des autres approches d'ingénierie des exigences sont concentrées sur ce que le

logiciel doit faire et comment il doit le faire en traitant les exigences seulement en terme de processus et de données et en ne s'intéressant donc pas à la justification de ces exigences.

Par conséquent, il sera difficile par la suite de comprendre les exigences et de juger si elles capturent vraiment tous les besoins des parties prenantes.

Aussi, dans [Cas 02], les auteurs affirment que la tendance de la plupart des autres approches est d'abstraire les constructions de programmation (de bas niveau) au niveau des exigences plutôt que de pousser les abstractions des exigences jusqu'au niveau de la conception.

## 3.2.1 Approche dirigée par les buts : GORE

---

La naissance de l'approche GORE remonte à plus de deux décennies avec précisément le travail de Yue [Yue 87] qui était le premier à affirmer que la modélisation explicite des buts dans les modèles d'exigences peut fournir un critère pour la complétude des exigences. Dans [Dar 93, Lam 09], on trouve une définition du but comme un objectif que le système et son environnement doivent réaliser grâce à la coopération de différents agents (matériel, logiciel ou humain).

Un but placé sous la responsabilité d'un agent du système est appelé une exigence (*requirement*), tandis qu'un but placé sous la responsabilité d'un agent de l'environnement du système est appelé une attente (*expectation*). Les principales activités suivantes sont normalement présentes dans la plupart des méthodes basées sur l'approche GORE :

- l'élicitation des buts,
- le raffinement des buts,

Divers types d'analyse des buts et l'attribution de la responsabilité d'un but à un agent.

Nous présentons dans ce qui suit, une panoplie de méthodes adoptant le paradigme GORE.

## 3.2.2 L'Approche i\*

---

Le framework iSTAR ou i\*, acronyme de « Intentional STRatégic Actor Relationships » [Yu 95, Yu 97], part du principe que les acteurs (les parties prenantes ou les entités actives) d'un système sont non seulement reliés entre eux par leurs actions ou l'information qu'ils échangent, mais aussi par les attributs intentionnels (les buts, les capacités, les croyances et les engagements) qui les caractérisent.

Le framework i\* comporte deux modèles principaux : le modèle de dépendance stratégique et le modèle de raisonnement stratégique. Dans ces modèles i\*, les acteurs sont décrits

dans leur contexte organisationnel et dépendent les uns des autres pour réaliser leurs buts, exécuter leurs tâches, ou disposer de ressources.

i\* peut être utilisé tout au long des phases du processus d'ingénierie des exigences. Pendant les premières phases, le framework i\* est utilisé pour faciliter l'analyse du domaine par des diagrammes qui permettent de représenter les acteurs du système, leurs buts et leurs relations. Les modèles i\* développés à ce stade aident à comprendre pourquoi le futur système est nécessaire. Durant les dernières phases de l'ingénierie des exigences, les modèles i\* sont utilisés pour proposer de nouvelles configurations du système qui seront évaluées en fonction de la façon dont elles répondent aux besoins fonctionnels et non-fonctionnels des utilisateurs. Pour cela, les softgoals (comme dans le NFR framework) sont utilisés comme critères de sélection pour choisir l'alternative du processus de configuration qui répond le mieux aux exigences non-fonctionnelles du système. En i\*, il y a les liens de type « moyen-finalité » (means-ends en anglais) entre les buts pour spécifier les différentes alternatives pour réaliser un but. Les liens de décomposition quant à eux connectent un but/tâche avec ses composants (sous-tâches, softgoals, etc.). Un softgoal, un but ou une tâche peuvent être également liés à d'autres softgoals avec des liens de contribution, comme dans le NFR framework. Ces liens de contribution spécifient deux niveaux de contributions positifs (+ et ++) et négatifs (- et --) pour la satisfaction d'un softgoal, la réalisation d'un but, ou l'exécution d'une tâche. La figure 3.4 illustre un exemple.

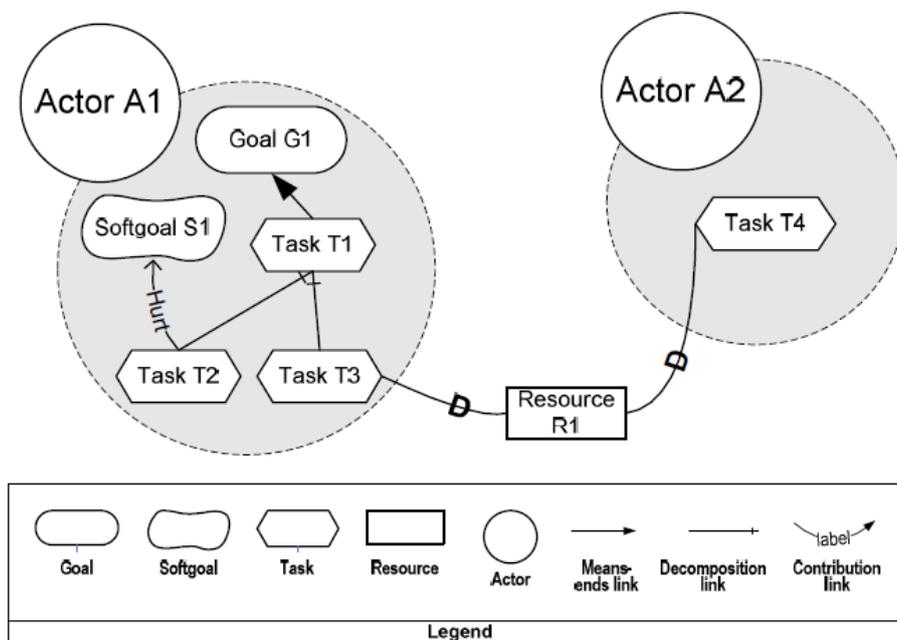


Figure 3.4 Exemple d'un modèle de buts illustrant ses concepts de base

Un langage, basé sur i\* et nommé Goal-oriented Requirements Language (GRL), définit tous les concepts de i\* (but, tâche, ressource, liens de contribution, etc.). GRL [Liu 01] est une partie de la notation des exigences utilisateur (URN) qui est une nouvelle recommandation de l'union internationale des télécommunications (ITU). URN fournit le premier standard des langages orientés-buts et il intègre aussi la notation UCM (Use Case Maps). TROPOS 3 [Cas 01] est une méthode d'ingénierie logicielle orientée-agent qui couvre tout le processus de développement logiciel. TROPOS se base sur i\* afin de couvrir les toutes premières phases de l'analyse des exigences, permettant ainsi une meilleure compréhension :

1. de l'environnement dans lequel le logiciel doit fonctionner ;
2. des types d'interactions qui devraient se produire entre les logiciels et les agents humains.

TROPOS adopte le framework i\* afin d'exprimer les concepts d'acteur, de buts et de leurs interdépendances.

### 3.2.3 Approche KOAS

---

KAOS, acronyme de « Keep All Objectives Satisfied » [Dar 93, Lam 09], est une méthode d'ingénierie des exigences qui résulte des travaux de recherche menés à l'Université de Louvain, en collaboration avec l'Université d'Oregon. Cette méthode permet aux analystes de construire des modèles d'exigences et de produire des documents à partir de ces modèles. KAOS a distingué les buts des propriétés du domaine qui sont des déclarations descriptives sur l'environnement telles que des lois physiques ou des normes organisationnelles.

Le modèle des exigences KAOS se compose de cinq sous-modèles fortement liés par des règles de cohérence :

**Le modèle central** : c'est le modèle de buts qui décrit les buts du système et de son environnement. Ce modèle est organisé dans une hiérarchie obtenue grâce au raffinement de buts de plus haut niveau (les buts stratégiques) vers des buts de bas niveau (les exigences).

**Le modèle objet** : Il permet de décrire le vocabulaire du domaine. Il est représenté par un diagramme de classes UML.

**Le modèle des responsabilités** : Il permet d'assigner les exigences (les buts feuilles) aux différents agents. Ces agents appartiennent au système à construire (agents internes) ou à son environnement (agents externes).

**Le modèle des opérations** : Il représente les opérations du système en termes de caractéristiques individuelles et leurs liaisons avec les modèles de buts (liens

d'opérationnalisation des exigences), objet (liens entrée-sortie) et responsabilités (liens d'exécution).

**Le modèle des comportements :** Il résume tous les comportements que les agents doivent accomplir pour satisfaire les exigences.

La Figure 3.5 donne un aperçu des quatre premiers sous-modèles KAOS ainsi que des principaux éléments qui les constituent. Le modèle des exigences KAOS peut être vu donc comme une sorte de puzzle visant à assembler les différentes pièces identifiées au fur et à mesure que les sources d'information sont exploitées. Le modèle de buts joue un rôle central dans la méthode KAOS. Par exemple, il est possible de dériver certains modèles KAOS (le modèle des opérations, le modèle objet, etc.) à partir du modèle de buts d'une façon systématique [Lam 09]. De plus, le modèle de buts permet de supporter très tôt différentes formes d'analyse des exigences telles que l'analyse de risques, l'analyse de conflits, ou l'évaluation des alternatives.

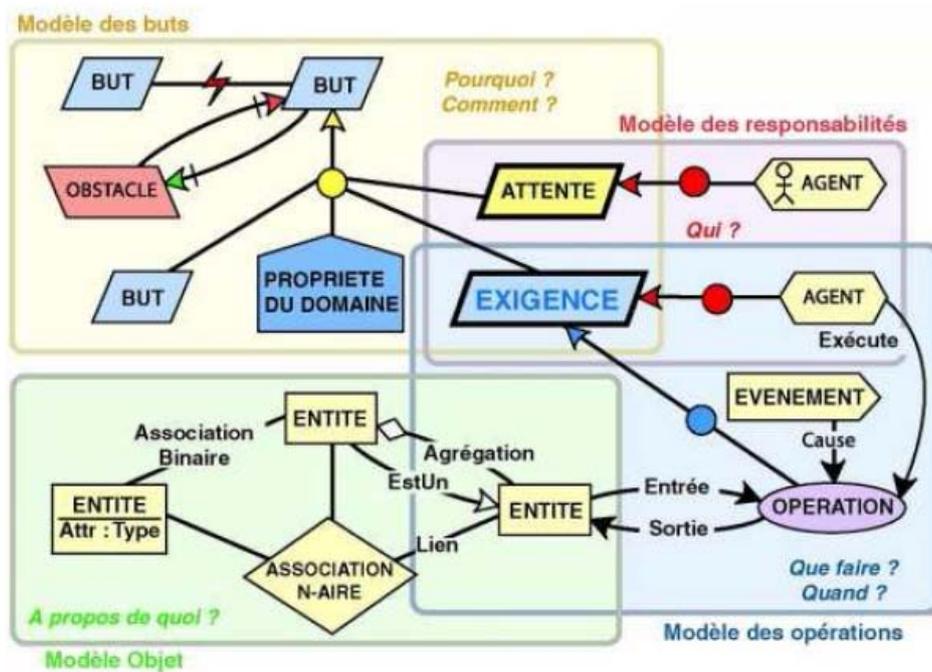


Figure 3.5 Un Aperçu du Modèle KAOS

Comme dans la plupart des méthodes GORE, les buts dans KAOS peuvent être raffinés selon des raffinements ET ou OU. Un raffinement ET implique que la conjonction des sous-buts est une condition suffisante pour réaliser le but parent.

Un raffinement OU associe un but à des sous-buts alternatifs pour lesquels l'accomplissement du but de plus haut niveau exige l'accomplissement d'au moins un de ses sous-buts. KAOS offre un ensemble de patrons de raffinement [Lam 09] qui décomposent les buts.