Avant-Propos

Les travaux présentés dans ce mémoire ont été effectués au sein du Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique (LAAS - CNRS). Je tiens à remercier tout d'abord Jean-Claude Laprie, ainsi que Malik Ghalab directeurs successifs du LAAS-CNRS pendant mon séjour, de m'avoir permis de mener mes recherches dans ce laboratoire.

Ma reconnaissance se tourne particulièrement vers David Powell, ancien responsable du groupe Tolérance aux fautes et Sûreté de fonctionnement (TSF) et son successeur Jean Arlat, pour m'avoir accueilli dans ce groupe de recherche.

Mes plus grands remerciements sont naturellement pour Yves Deswarte, qui m'a encadré tout au long de ma thèse. Ma considération est inestimable. Ses remarques et critiques pertinentes m'ont conduit vers la bonne voie. Son soutien m'a permis de ne jamais faiblir et de poursuivre toujours plus loin mes travaux. Je tiens également à souligner que la confiance qu'il a mise en moi a été un moteur à ma réussite.

J'exprime ma gratitude à Yves Dutuit, Professeur à l'Université de Bordeaux I, pour l'honneur qu'il me fait en présidant mon Jury de Thèse, ainsi qu'à :

- Abdelmalek Benzekri, Professeur à l'Université Paul Sabatier ;
- Frédéric Cuppens, Professeur à l'Ecole Nationale Supérieure de Télécommunications (ENST-Bretagne) ;
- Marc Dacier, Professeur à l'Institut Eurecom Sophia Antipolis, Professeur adjoint à l'Université de Liège et professeur visiteur à l'Université catholique de Louvain ;
- Yves Deswarte, Directeur de Recherche CNRS ;
- Gilles Trouessin, Directreur de mission Sénior à Ernst & Young Audit ;

pour l'honneur qu'ils me font en participant à mon jury. Je remercie particulièrement Abdelmalek Benzekri et Marc Dacier qui ont accepté la charge d'être rapporteur.

Je voudrais également remercier l'ensemble des partenaires du projet MP6 avec qui j'ai beaucoup appris, notamment Gilles Trouessin (responsable du projet MP6) Philippe Balbiani, Frédéric Cuppens, Claire Saurel, Salem Benferhat, Alexandre Miège, Rania El-Baida et Didier Vinsonneau. Sans eux, ces travaux n'auraient probablement pas été les mêmes.

Je remercie tout le groupe TSF, les permanents, les doctorants et les stagiaires.

Mes remerciements s'adressent également à l'ensemble des services du LAAS, qui permettent à chacun de travailler dans les meilleures conditions.

Il m'est agréable de remercier chaleureusement tous ceux qui, en dehors du laboratoire, m'ont accompagné et soutenu. Je pense particulièrement à Betty qui m'a beaucoup aidé et qui a partagé avec moi des moments faciles et difficiles durant ces trois années.

Ces avant-propos seraient incomplets sans un remerciement adressé aux membres de ma famille, en particulier mes parents et mon frère Sidi Mohamed. Ce travail leur appartient à tous. Je pense également à Haj Alouane, Haj Belkahia, Florian, mes tantes Aziza, Hagiba, Souad, mes amis Badri, Rachid, Redouane, Noredine, Moustapha ainsi que tous les autres gens aimables et serviables qui m'ont soutenu et qui ont contribué à mon enrichissement personnel.

À tous ces gens-là, je serais éternellement reconnaissant. Merci.

TABLE DES MATIERES

INDEX DES FIGURES

INDEX DES TABLEAUX

Introduction générale

Alors que l'informatisation s'impose dans des domaines complexes, coopératifs et largement distribués comme la télémédecine et les télédéclarations sociales, il est de plus en plus nécessaire d'avoir confiance dans les traitements et la distribution des données et services informatiques.

Les Systèmes d'Information et de Communication en Santé et social (SICSS) permettent de stocker et de gérer des informations médicales, administratives ou sociales relatives à des personnes ou des entreprises. Ils exploitent les technologies de l'informatique pour permettre aux utilisateurs un accès rapide à ces informations, et ainsi faciliter les actes médicaux, les remboursements, les télédéclarations sociales, les télépaiements, etc. Toutefois, les menaces qui pèsent sur de tels systèmes peuvent provoquer la réticence des usagers (patients pour la sphère médicale, entreprises pour la sphère sociale). En effet, l'exploitation abusive par un utilisateur malhonnête d'un SICSS insuffisamment protégé peut rendre possible la divulgation de données personnelles à différents intéressés : employeurs, concurrents, banques, etc. Les erreurs de saisie ou de conception peuvent entraîner des erreurs de diagnostic, de soins ou de paiements. Les défaillances peuvent empêcher le personnel soignant d'accéder à des informations indispensables. Enfin, la peur d'un manque de confidentialité, d'intégrité, de disponibilité ou d'auditabilité de tels systèmes peut inciter des patients et des entreprises à refuser de divulguer des informations pourtant vitales.

Pour atteindre un niveau de protection satisfaisant, il convient de définir une politique de sécurité correspondant aux besoins. En effet, toute démarche de sécurité rigoureuse doit être inscrite dans une politique claire et documentée. Sa conception est donc une étape primordiale, qui consiste à identifier les objectifs de sécurité et à élaborer un ensemble de règles en fonction d'une analyse des risques. Ceci permet de minimiser le risque de dommages indésirables ou de pallier leurs effets et conduit à protéger les informations et les ressources identifiées comme sensibles.

Une politique de sécurité se développe selon trois axes : physique, administratif et logique. Le premier précise l'environnement physique du système à protéger (les éléments critiques, les mesures prises vis-à-vis du vol et des catastrophes). Le deuxième décrit les procédures organisationnelles (répartition des tâches, séparation des pouvoirs). Le troisième a trait aux contrôles d'accès logiques (qui, quoi, quand, pourquoi, comment) et s'intéresse aux fonctions d'identification, d'authentification et d'autorisation mises en œuvre par le système informatique. Dans ce mémoire, nous nous intéressons particulièrement aux politiques d'autorisation (dites aussi politiques de contrôle d'accès).

Les premiers travaux sur l'expression et la mise en œuvre de politiques d'autorisation ont débuté, il y a plus de vingt ans, principalement dans le domaine de la défense. Pour des raisons juridiques (responsabilité), éthiques (respect de la vie privée), déontologiques (secret médical, par exemple), organisationnelles (situations d'urgence, cas particuliers ou non attendus) et techniques (interconnexion de réseaux locaux, régionaux et nationaux), ce type de politiques de sécurité est clairement inadapté au monde de la santé ou du social. La conception de politiques

d'autorisation beaucoup plus dynamiques – et pouvant s'adapter aux contextes des activités médicales ou sociales – est nécessaire. Les modèles et politiques de sécurité, fondés sur le concept de rôle, sont une première étape pour mieux répondre à de tels besoins sectoriels, mais ils ne satisfont pas à toutes les spécificités des SICSS. Des modèles et politiques plus récents, reposant sur les notions d'équipes et de contextes, semblent également intéressants, mais nécessitent des approfondissements théoriques et des études complémentaires.

Ainsi, les politiques et modèles de sécurité actuels étant incapables de couvrir toute la richesse des SICSS, il semble nécessaire de proposer de nouveaux concepts et de présenter un modèle pouvant assurer une meilleure sécurité, sans pour autant gêner le travail des usagers ou porter atteinte aux droits des patients.

La réflexion s'articule autour de six moments :

Le premier chapitre montre la pertinence d'une étude de sécurité dans la sphère santé-social, présente la terminologie utilisée, et situe les politiques de sécurité dans l'éventail des stratégies et outils pour renforcer la sécurité d'un système ou d'une organisation.

Le deuxième chapitre étudie les politiques et modèles de sécurité existants, et montre les limites de leur application aux SICSS. Elle présente également des projets récents dans ce domaine et introduit le projet MP6, *Modèles et Politiques de Sécurité pour les Systèmes d'Information et de Communication en Santé et Social*, projet dans lequel ont été effectués nos travaux.

Le troisième chapitre montre comment bâtir une politique de sécurité pour un système ou une organisation. Cette méthodologie est appliquée en posant les principales briques d'une politique de sécurité pour les sphères sociale et médicale. À ce niveau de l'étude, les SICSS seront décrits en détail à travers des règles de fonctionnement, des objectifs de sécurité ainsi que des règles de sécurité. De par son importance dans les SICSS, le problème d'anonymisation est enfin abordé en détail. Une étude préalable est nécessaire avant toute procédure d'anonymisation. Cette étude doit identifier les besoins, les objectifs ainsi que les exigences en terme d'anonymisation. S'en suivra une présentation des principaux travaux dans ce domaine, une description d'un ensemble de scénarios récapitulatifs, et des propositions de solutions mieux adaptées aux besoins actuels et futurs. Les politiques de sécurité décrites pourraient ainsi être appliquées aussi bien aux données anonymisées qu'aux autres informations sensibles, ressources et services des SICSS.

Le quatrième chapitre rappelle les concepts de base de notre politique de sécurité. Celle-ci tient compte du contexte et réalise un bon compromis entre la flexibilité et l'efficacité du contrôle d'accès. Par ailleurs, une représentation UML (*Unified Modeling Language*) du nouveau méta-modèle Or-BAC "*Organization-Based Access Control*" sera ensuite proposée. Centré sur l'organisation, Or-BAC offre l'expressivité et la flexibilité nécessaires à la représentation de politiques de sécurité pour une large gamme d'organisations et de systèmes, notamment les SICSS.

Le cinquième chapitre présente une vision logique qui servira à formaliser et à raisonner sur une politique de sécurité fondée sur Or-BAC. À cet égard, un langage approprié (fondé sur la logique déontique) sera d'abord proposé, et sera ensuite utilisé pour représenter les règles de fonctionnement, les objectifs de sécurité ainsi que les règles de sécurité des SICSS. Des idées sur l'exploitation de ce formalisme – notamment pour la vérification de la cohérence d'une politique de sécurité ou pour la résolution de conflits – seront également proposées.

Le sixième chapitre commence par présenter une démarche UML pour bâtir une politique de sécurité associée à Or-BAC. Cette démarche tient compte des aspects conceptuels, statiques et dynamiques d'une politique de sécurité. Enfin, il s'agira de décrire une concrétisation de nos idées à travers une implémentation d'un logiciel de contrôle d'accès pour un centre dentaire.

Chapitre 1. Sécurité des systèmes d'information et de communication en santé et social

Préambule

Ce chapitre repose sur deux motivations. D'une part, fournir la base terminologique nécessaire à la compréhension de nos travaux, et d'autre part, situer nos centres d'intérêts dans le vaste domaine de la sécurité.

Aussi, ce chapitre est-il articulé en cinq parties.

C'est par la description de notre champ d'application – les systèmes d'information et de communication en santé et social (SICSS) – que l'étude débutera. Les SICSS couvrent l'ensemble des besoins généralement trouvés dans les autres domaines.

Les concepts de la sûreté de fonctionnement, et plus particulièrement ceux de la sécurité informatique, seront ensuite présentés. Le but est de fournir un support terminologique dans la définition des politiques de sécurité pour les SICSS, puis dans l'élaboration de modèles formels de ces politiques.

Suit une brève introduction à la sécurité des systèmes d'information et, à l'instar des *ITSEC* [ITSEC 1991], critères européens d'évaluation de la sécurité des systèmes d'information, elle décrit la sécurité comme l'association de la *confidentialité*, de *l'intégrité* et de la *disponibilité* vis-à-vis des actions autorisées.

Les ambiguïtés sont ensuite levées sur les notions d'intrusions, d'attaques, de vulnérabilités et de risques, et des exemples spécifiques aux SICSS seront donnés.

La dernière partie du chapitre détaille les techniques de sécurité les plus utilisées pour faire face aux intrusions, et pour renforcer la sécurité d'un système ou d'une organisation. Il s'agira de décrire des mesures comme les politiques de sécurité, les mécanismes de chiffrement, la détection d'intrusion, etc. Ces mécanismes, aussi robustes soient-ils, ne peuvent sécuriser efficacement et rigoureusement un système que s'ils s'intègrent dans une démarche globale fondée sur une politique de sécurité.

1.1. Caractéristiques des systèmes d'information et de communication en santé et social

Cette section a pour but de caractériser, très brièvement, les SICSS, domaine d'application étudié tout au long de ce mémoire. Une analyse plus détaillée de ces systèmes sera donnée dans le troisième chapitre.

1.1.1. Définition et enjeux

Les systèmes d'information et de communication en santé et social (SICSS) permettent de stocker et de gérer des informations médicales, administratives ou sociales concernant des individus ou des entreprises. Ils doivent faciliter les tâches de leurs utilisateurs : médecins, secrétaires médicales, infirmiers, agents d'assurances maladie, ou encore usagers de Net-entreprises[1], tous en charge de traitements tels les diagnostics, les actes médicaux, les soins, les remboursements et les déclarations sociales.

Ces systèmes manipulent, entre autres, des données à caractère personnel et souvent nominatives[2]. Citons, à titre d'exemple, les informations décrivant les situations médicales (historique des pathologies et allergies, diagnostics, actes médicaux, résultats biologiques), administratives (identité et coordonnées, situation familiale, numéro de SIREN[3], salaires) ou sociales (prestations financières et sociales).

Les SICSS exploitent les progrès des technologies de l'informatique et des réseaux pour permettre aux utilisateurs un accès rapide aux informations, et ainsi faciliter et contrôler la prise en charge (médicale, administrative ou sociale) des patients et des ayant-droits. Ils servent aussi à alléger la charge administrative qui pèse sur les petites et moyennes entreprises et notamment la procédure de création des entreprises, le bulletin de paie, le calcul des charges sociales, les mesures fiscales, comptables et statistiques, etc.

Toutefois, la mise en œuvre de ces technologies met en danger les données gérées par les SICSS. En l'occurrence, le marché des informations nominatives intéresse nombre d'organisations : industries pharmaceutiques, compagnies d'assurances, banques, employeurs, presse, stratèges politiques, etc. Les SICSS sont donc des cibles privilégiées pour des individus malintentionnés, susceptibles d'exploiter toute vulnérabilité du système pour violer les exigences de sécurité.

1.1.2. Complexité des SICSS

Les SICSS sont des systèmes riches en fonctionnalités, complexes, sensibles, hétérogènes et exigeant un niveau élevé d'interopérabilité. En effet, les SICSS :

- relient des organisations multiples et des utilisateurs ayant des profils différents ; dans le domaine médical, il s'agit de professionnels de santé, hôpitaux et organismes payeurs ; dans la sphère sociale, il s'agit des organismes de protection sociale, entreprises et banques ;

---

[1] Net-entreprises est le service proposé en France aux entreprises par l'ensemble des organismes de protection sociale pour leur permettre d'effectuer, par Internet, leurs déclarations et leurs paiements.

[2] Nous considérons une information comme nominative si elle contribue à la description d'individus (ou entreprises) bien identifiés ou identifiables.

[3] Le numéro de SIREN est un numéro attribué par l'INSEE à toute personne physique ou morale qui exerce une activité professionnelle lors de l'inscription au répertoire national des entreprises.

- mettent en jeu des technologies complexes comme la communication, le traitement d'information, la télémédecine, le télépaiement et l'archivage ;
- manipulent des informations sensibles et hétérogènes ; il s'agit de données textuelles ou graphiques, d'images et d'enregistrements de cardiogrammes, etc. ; le caractère, souvent personnel, de ces informations, oblige à prendre des précautions particulières, afin d'assurer leur protection, notamment durant toute manipulation (accès, transfert, stockage) ;
- nécessitent une coopération de ses utilisateurs qui échangent les informations, consultent les bases de données, et utilisent les applications médicales, paramédicales, médico-administratives et médico-financières ; en effet, les médecins, hôpitaux, pharmacies et laboratoires doivent pouvoir s'échanger des informations médicales afin de favoriser l'aide au diagnostic ou la recherche épidémiologique ; les services d'étude et de recherche épidémiologique envoient aux médecins des statistiques d'activité et leur fournissent une aide à la décision. Les médecins envoient les feuilles de soins électroniques aux régimes d'assurance maladie et reçoivent, en échange, des accusés de réception, etc.

La diversité des organisations dans lesquelles de telles applications doivent être mises en œuvre ainsi que les interactions entre les applications, nécessitent une grande flexibilité dans la définition des politiques de sécurité. Ces interactions sont exigeantes en matière de sécurité, notamment en intégrant des droits, des interdictions, des obligations et des recommandations, attribués à chacun des acteurs et issus des responsabilités qu'ils doivent exercer.

1.1.3. Diversité des exigences de sécurité

Selon les domaines d'application, les exigences de sécurité peuvent varier pour ce qui concerne l'importance relative de chacune des propriétés de sécurité : disponibilité, intégrité, disponibilité et auditabilité (voir section 3 du présent chapitre).

Ainsi, dans le domaine militaire par exemple, et plus généralement dans le domaine gouvernemental, l'accent est mis principalement sur la confidentialité, l'intégrité étant le plus souvent estimée comme secondaire et la disponibilité étant encore plus négligée : la divulgation d'une information est considérée comme plus grave que son altération ou son indisponibilité.

Dans le domaine financier, qu'il s'agisse d'applications bancaires ou de comptabilité des entreprises, l'intégrité est de loin le souci majeur, bien plus que la disponibilité, la confidentialité étant encore d'un moindre souci. En effet, une altération de l'information, qu'il s'agisse de fraude ou d'une faute accidentelle, peut avoir des conséquences financières démesurées. Le manque de disponibilité est généralement d'une gravité moindre. Quant aux pertes liées au manque de confidentialité, elles sont généralement difficiles à évaluer financièrement et le plus souvent considérées comme négligeables devant les risques liés à l'altération des données.

À l'inverse, les SICSS se caractérisent par leurs fortes exigences, souvent simultanées, de *confidentialité*, d'*intégrité* et de *disponibilité*, mais aussi d'*auditabilité*. En effet, dans le domaine médical, l'intégrité (des diagnostics, par exemple) et la disponibilité (caractère d'urgence) peuvent parfois être vitales, au sens strict du terme, mais la confidentialité est plus qu'une obligation légale : l'accès à des informations médicales peut avoir des implications financières importantes vis-à-vis d'un employeur ou d'un assureur, mais peut aussi être la source d'un chantage. La propriété d'auditabilité est également très importante pour renforcer la responsabilité des personnels soignants. De la même manière, dans le domaine social, la confidentialité des données concernant les personnes et les entreprises, l'intégrité des télédéclarations et des télépaiements, la disponibilité des téléservices de Net-entreprises

(surtout dans les échéances de déclarations et de paiements), ainsi que la responsabilité des actions (paiement à échéance par exemple), sont autant de points cruciaux.

Malheureusement, des conflits potentiels peuvent apparaître entre toutes ces exigences de sécurité des SICSS. En effet, un objectif de confidentialité sur les dossiers médicaux conduit à définir une règle limitant l'accès au dossier médical d'un patient à son seul médecin traitant. Pourtant, un objectif de disponibilité peut amener à définir comme règle qu'en cas d'urgence, n'importe quel médecin puisse y avoir accès. Dans d'autres cas, un professionnel de santé peut avoir besoin de certaines données indirectement nominatives (par exemple, l'âge, l'appartenance sociale, ou la région géographique) pour réaliser une étude épidémiologique. Une telle exigence, liée à la disponibilité, peut entrer en conflit avec des exigences de confidentialité (par exemple, risque d'inférence non autorisée si ces données indirectement nominatives sont divulguées).

1.1.4. Menaces pesant sur les informations manipulées par ces systèmes

Les menaces auxquelles les SICSS sont confrontés peuvent causer différents préjudices, notamment en portant atteinte à la confidentialité des informations concernant les patients et les organisations, à l'intégrité des données et des programmes, à la disponibilité des services et des données nécessaires au bon fonctionnement. Plusieurs études et statistiques récentes montrent l'ampleur de ces menaces. Des enquêtes menées par la commission d'audit britannique [Audit 1998] et par le bureau d'évaluation de la technologie du gouvernement américain ont confirmé que le domaine de la santé est l'une des cibles privilégiées d'attaquants aussi bien internes qu'externes (atteinte à la vie privée, fraudes, etc.) [Woodward 1995]. Plus récemment, en 2001, un pirate a pu s'emparer du serveur de fichiers d'un hôpital à Seattle (aux États-Unis) et diffuser sur le Web (securityfocus.com) les fichiers médicaux de cinq mille patients. Des études plus anciennes [Tufo 1971] révèlent que, dans plus de 30 % des cas, les fichiers médicaux sont indisponibles, et que même quand ils sont disponibles, les délais nécessaires pour extraire les informations sont souvent décourageants. Les 26, 27 octobre et le 4 novembre 1992, suite à une surcharge du système informatique, le service des ambulances de Londres a été bloqué ; causant la mort de vingt personnes. L'enquête a révélé que la transition vers le système de sauvegarde n'avait pas été correctement préparée.

À cet égard, les vulnérabilités des SICSS peuvent être de natures diverses : fautes de conception ou de spécification, comme les portes dérobées permettant des infiltrations malveillantes extérieures (voir 1.4) ; politiques de sécurité ne tenant pas compte de toutes les manipulations illégitimes ; faiblesses dans le système socio-technique, dues par exemple à une procédure d'habilitation trop laxiste des personnels ; protection physique insuffisante du matériel et des ressources ; etc.

En outre, les attaques peuvent aussi bien provenir de l'intérieur (abus de pouvoir, curiosité allant au-delà de l'utilisation des informations et services strictement nécessaires pour l'accomplissement du travail) que de l'extérieur (pirate informatique qui tente de lire ou de modifier une information ou d'usurper l'identité d'un professionnel de santé par exemple).

Une intrusion (attaque au moins particulièrement réussie) peut donner lieu à :

- des divulgations de données personnelles intimes, ou professionnelles secrètes (violation de la confidentialité) ;
- des erreurs de diagnostic, d'actes médicaux, de télédéclarations, ou de télépaiements (violation de l'intégrité et de la disponibilité) ;
- l'indisponibilité d'informations cruciales pour les médecins (respectivement les organismes de protection sociale) qui peuvent en avoir besoin pour leurs patients (respectivement entreprises) ou pour justifier leurs décisions, si nécessaire (violation de la disponibilité et de l'auditabilité).

Enfin, le manque de confiance peut conduire chaque partenaire d'un SICSS à installer sa propre politique de sécurité, au détriment d'une interopérabilité pourtant indispensable à l'échange d'informations entre les usagers des SICSS. Par ailleurs, la peur d'un manque de confidentialité, d'intégrité, de disponibilité ou d'auditabilité de tels systèmes peut inciter des patients (ou, dans le domaine social, les entreprises) à refuser de divulguer des informations pourtant capitales.

[Abou El Kalam *et al.* 2002c] identifie une liste plus exhaustive de risques spécifiques aux SICSS, mais aussi à des risques plus généraux, liés à l'utilisation de l'informatique et de la télématique. Par ailleurs, une caractérisation plus détaillée des besoins des SICSS peut être trouvée dans [Abou el Kalam & Deswarte 2003a].

1.2. Concepts de la sûreté de fonctionnement

Les SICSS, ainsi présentés, touchent un domaine sensible, nécessitant une confiance élevée dans les services délivrés. Cette confiance ne peut être obtenue que si les SICSS sont sûrs de fonctionnement. Cette section présente les concepts et termes classiques usuels de la sûreté de fonctionnement et de la sécurité informatiques, et ceux spécifiques, nécessaires pour appréhender les atypismes et particularités des SICSS.

1.2.1. Définitions de base

La *sûreté de fonctionnement* d'un système informatique est la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qu'il leur délivre [Laprie 1995]. Selon les applications auxquelles le système est destiné, l'accent peut être mis sur différentes facettes de la sûreté de fonctionnement, ce qui revient à dire que la sûreté de fonctionnement peut être vue selon des propriétés différentes mais complémentaires, qui permettent de définir ses attributs :

- le fait d'être prêt à l'utilisation conduit à la *disponibilité* ;
- la continuité du service conduit à la *fiabilité* ;
- la non-occurrence de conséquences catastrophiques pour l'environnement conduit à la *sécurité-innocuité* ;
- la non-occurrence de divulgations non-autorisées de l'information conduit à la *confidentialité* ;
- la non-occurrence d'altérations inappropriées de l'information conduit à l'*intégrité* ;
- l'aptitude aux réparations et aux évolutions conduit à la *maintenabilité*.

Les entraves à la sûreté de fonctionnement sont de trois ordres : défaillances, erreurs et fautes. Il y a défaillance lorsque le service délivré dévie du service attendu. L'erreur est la partie de l'état du système susceptible d'entraîner une défaillance. Et la faute est la cause adjugée ou supposée de l'erreur.

Les moyens de la sûreté de fonctionnement sont de deux ordres : d'une part les méthodes permettant de fournir au système l'aptitude à délivrer un service conforme au service attendu, d'autre part celles permettant de donner une confiance justifiée dans cette aptitude. L'obtention de la sûreté de fonctionnement se fait par *prévention des fautes* ; ce qui consiste à empêcher par construction l'occurrence ou l'introduction de fautes, et par *tolérance aux fautes*, qui permet par redondance de fournir un service conforme en dépit de fautes. Les méthodes de validation de la sûreté de fonctionnement se classent en *élimination des fautes*, correspondant à la réduction, par vérification, du nombre et de la gravité des fautes, et en *prévision des fautes*, c'est-à-dire à l'évaluation de la présence et de la création de fautes et de leurs conséquences futures (figure 1.1).

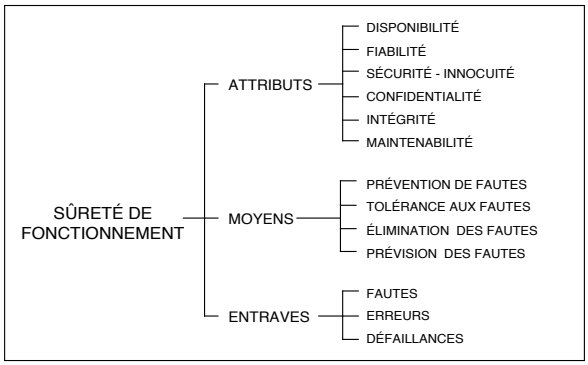


Figure 1.1 : L'arbre de la sûreté de fonctionnement.

1.2.2. Les fautes dues à l'homme

Les fautes, ainsi que leurs sources sont extrêmement diverses. Les principales facettes selon lesquelles on peut les classer sont leur cause, leur nature, leur phase de création ou d'occurrence, leur situation par rapport aux frontières du système, et leur persistance (figure 1.2).

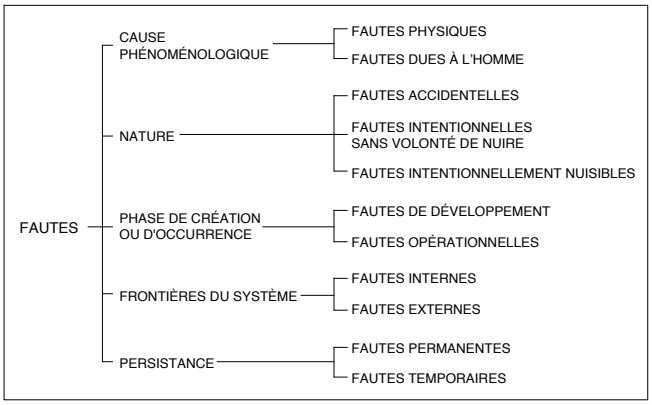


Figure 1.2 : Les classes de fautes élémentaires

Quand on s'intéresse à la sécurité informatique en général et à celle des SICSS en particulier, la principale classe de fautes à prendre en compte est celle des *fautes dues à l'homme*, qu'elles soient *intentionnelles* ou *accidentelles*. Ces fautes donnent lieu à quatre classes de fautes combinées :

- les fautes de conception, qui sont des fautes de développement accidentelles ou intentionnelles sans volonté de nuire ;
- les fautes d'interaction, qui sont des fautes externes, accidentelles ou intentionnelles sans volonté de nuire ;
- les logiques malignes, qui sont des fautes internes intentionnellement nuisibles ;
- les intrusions, qui sont des fautes opérationnelles externes intentionnellement nuisibles.

Les fautes de conception intentionnelles sans volonté de nuire résultent généralement de compromis effectués durant la conception, dans un souci de conserver au système un niveau de performances acceptable, de faciliter son utilisation, ou encore pour des raisons économiques.

Les fautes d'interaction intentionnelles sans volonté de nuire peuvent résulter de l'action d'un opérateur soit destinée à faire face à une situation imprévue, soit violant délibérément des procédures sans avoir réalisé les conséquences malheureuses de son action. Généralement, ces fautes ne sont identifiées qu'après qu'elles aient causé une défaillance.

Les logiques malignes recouvrent aussi bien des fautes de développement comme les chevaux de Troie, les portes dérobées, les bombes logiques, ou des fautes opérationnelles comme les virus et les vers. Ces fautes peuvent être définies comme suit :

- une bombe logique est une partie de programme qui reste dormante dans le système hôte jusqu'à ce qu'un instant ou un événement survienne, ou que certaines conditions soient réunies, pour déclencher des effets dévastateurs en son sein ;
- un cheval de Troie est un programme effectuant une fonction illicite tout en donnant l'apparence d'effectuer une fonction légitime ; la fonction illicite peut être de divulguer ou d'altérer des informations, ou peut être une bombe logique ;
- une porte dérobée est un moyen de contourner les mécanismes de contrôle d'accès ; il s'agit d'une faille du système de sécurité due à une faute de conception accidentelle ou intentionnelle (cheval de Troie en particulier) ;
- un virus est un segment de programme qui, lorsqu'il s'exécute, se reproduit en s'adjoignant à un autre programme (du système ou d'application), et qui devient ainsi un cheval de Troie ; un virus peut être porteur d'une bombe logique ;
- un ver est un programme autonome qui se reproduit et se propage à l'insu des utilisateurs ; un ver peut également être porteur d'une bombe logique.

Les intrusions ne peuvent être couronnées de succès sans l'existence de fautes de conception. Le caractère externe des intrusions n'exclut pas qu'elles soient tentées par des opérateurs ou administrateurs du système qui abusent de leur pouvoir. Les intrusions sont détaillées dans 1.4.

1.3. La sécurité des systèmes d'information

1.3.1. Introduction : définition de la sécurité

Dans le domaine de l'informatique, le mot "*sécurité*" peut couvrir plusieurs acceptions [Deswarte 2003]. La première correspond à la *sécurité-innocuité* (en anglais *safety*) et concerne la prévention de catastrophes : dans ce sens, un système informatique aura une sécurité satisfaisante si aucune de ses défaillances éventuelles ne peut provoquer de dégâts importants, ou si celles de ses défaillances qui peuvent provoquer des dégâts importants sont suffisamment peu probables. Ce type de sécurité est bien évidemment une exigence majeure lorsque le bon fonctionnement du système informatique est nécessaire pour la sauvegarde de vies humaines ou de l'environnement, ou encore d'intérêts financiers importants. C'est en particulier le cas des systèmes tels que les systèmes de transport ou de contrôle des centrales nucléaires.

Une seconde acception du terme de sécurité correspond au mot anglais "*security*" et concerne la capacité du système informatique à résister à des agressions externes physiques (incendie, inondation, bombes, etc.) ou logiques (erreurs de saisie, intrusions, piratages, logique malicieuse, etc.). C'est généralement le sens choisi par les spécialistes de l'audit de sécurité, lorsqu'ils doivent, pour une entreprise donnée, évaluer les risques liés à l'informatique.

Mais plutôt que de définir la sécurité vis-à-vis des conséquences de la non-sécurité (au sens *safety*) ou vis-à-vis des agressions contre la sécurité (au sens "*security*"), il semble préférable, à l'instar des ITSEC [ITSEC 1991], de considérer la sécurité comme la combinaison de trois propriétés : la *confidentialité*, l'*intégrité* et la *disponibilité* de l'information. Notons que ces trois propriétés se rapportent à l'*information*, et le terme d'information doit être pris ici dans son sens le plus large, couvrant non seulement les données et les programmes, mais aussi les

flux d'information, les traitements et la connaissance de l'existence de données, de programmes, de traitements, de communications, etc. Cette notion d'information doit aller jusqu'à couvrir le système informatique lui-même, dont parfois l'existence doit être tenue secrète. Pour être plus précis, on distinguera informations et "*méta-informations*" ; les informations correspondant à des données identifiées, alors que les méta-informations renvoient à des informations indirectes reliées aux informations ou aux services[4]. Voici quelques exemples de méta-informations :

- l'instant de délivrance d'un service, ou de création ou destruction d'une information ;
- l'identité de la personne qui a réalisé une opération : le créateur d'une information, l'auteur d'un document, l'émetteur ou le récepteur d'une information, etc. ;
- l'emplacement d'une information, d'une entité de communication, d'un terminal, etc. ;
- l'existence d'une information ou d'un service ;
- l'existence d'un transfert d'information, d'un canal de communication, ou d'un message ;
- l'occurrence d'une opération ;
- le niveau de sensibilité d'une information ou d'une méta-information ;
- la certitude ou le niveau de crédibilité d'une information ou d'une méta-information ;

La sécurité, telle qu'elle est ici appréhendée, implique d'empêcher la réalisation d'opérations illégitimes contribuant à mettre en défaut les propriétés de confidentialité, d'intégrité et de disponibilité, mais aussi de garantir la possibilité de réaliser les opérations légitimes dans le système. Assurer la sécurité du système, c'est assurer que les propriétés retenues sont vérifiées, autrement dit, garantir la non-occurrence de défaillances vis-à-vis de ces propriétés.

Par ailleurs, même si le présent mémoire étudie la sécurité sous la vision des ITSEC, il paraît évident que la sécurité-innocuité est une propriété importante pour les SICSS, et par conséquent, elle demeure un but global à atteindre. En effet, la sécurité des SICSS traite des problèmes du type : mauvaise identification des patients, modification illégitime d'un diagnostic (manque d'intégrité), retard ou absence de données dans des cas urgents critiques (manque de disponibilité), etc. Puisque ces problèmes peuvent porter atteinte à la vie des patients, ils concernent la sécurité-innocuité, et donc, cette vision de la sécurité (sécurité-innocuité) sera également présente dans ce travail, en tout cas de manière implicite.

1.3.2. Confidentialité

La confidentialité est la propriété d'une information de ne pas être révélée à des utilisateurs non autorisés à la connaître. Ceci signifie que le système informatique doit :

- empêcher les utilisateurs de lire une information confidentielle (sauf s'ils y sont autorisés), et
- empêcher les utilisateurs autorisés à lire une information et de la divulguer à d'autres utilisateurs (sauf autorisation).

Le terme information doit être pris au sens le plus large : il recouvre non seulement les données elles-mêmes, mais aussi les flux d'information et la connaissance de l'existence des données ou des communications. Assurer la confidentialité d'un système est donc une tâche complexe. Il faut analyser tous les chemins qu'une information particulière peut prendre dans le système pour s'assurer qu'ils sont sécurisés. Il importe également de prendre en compte les connaissances qu'un ou plusieurs utilisateurs peuvent déduire à partir des informations qu'ils

[4] Ce qui est "*méta-information*" à un niveau d'abstraction donné (par exemple, une application) peut être une "information" réelle à un niveau plus bas (par exemple, le système d'exploitation).

acquièrent. Il faut donc contrôler non seulement les informations présentes dans le système, mais aussi les liens logiques qui peuvent les relier entre elles ou à des informations publiques.

Les attaques contre la confidentialité consistent à essayer d'obtenir des informations qui doivent être protégées selon la politique de sécurité, en dépit des moyens de protection et des règles de sécurité. Par exemple, les écoutes passives consistent à accéder aux données transmises sur un canal de communication (câble de réseau, par exemple) ou stockée sur un support vulnérable (disques externes, par exemple). Une telle écoute peut, dans certaines circonstances, permettre d'accéder à des informations sensibles, comme le mot de passe d'un utilisateur tapé sur un terminal connecté à un ordinateur central, et qui transite en clair entre ce terminal et la machine. On voit également que cette attaque peut être particulièrement difficile à identifier *a posteriori* étant donné l'absence totale de traces laissées dans le système.

1.3.3. Intégrité

L'intégrité est la propriété d'une information de ne pas être altérée. Cela signifie que le système informatique doit :

- empêcher une modification[5] indue de l'information, c'est-à-dire une modification par des utilisateurs non autorisés ou une modification incorrecte par des utilisateurs autorisés, et
- faire en sorte qu'aucun utilisateur ne puisse empêcher la modification légitime de l'information. Par exemple, empêcher la mise à jour périodique d'un compteur de temps constituerait une atteinte à l'intégrité.

De plus, il faut avoir l'assurance que toute modification de donnée est approuvée et que chaque programme se comporte de manière correcte (c'est-à-dire conformément aux fonctions qu'il est censé remplir, y compris dans ses interactions avec les autres processus). Il faut également s'assurer qu'aucune information ne peut être modifiée par des intermédiaires, que cette altération soit intentionnelle (par exemple, un utilisateur intervient pour modifier une communication entre deux autres utilisateurs) ou accidentelle (une donnée modifiée lorsqu'elle est communiquée *via* un support de communication non-fiable).

Afin de se prémunir contre les fautes affectant l'intégrité des données, il importe d'intégrer dans le système des mécanismes permettant d'une part de détecter les modifications des informations, et d'autre part de contrôler les accès à ces dernières (en gérant les droits d'accès des programmes et utilisateurs). De plus, un travail de validation en amont peut également être réalisé pour prévenir les fautes accidentelles.

1.3.4. Disponibilité

La disponibilité est la propriété d'une information d'être accessible lorsqu'un utilisateur autorisé en a besoin. Cela signifie que le système informatique doit :

- fournir l'accès à l'information pour que les utilisateurs autorisés puissent la lire ou la modifier, et
- faire en sorte qu'aucun utilisateur ne puisse empêcher les utilisateurs autorisés d'accéder à l'information.

Ainsi définie, la disponibilité est une notion qui regroupe plusieurs concepts.

La disponibilité à court terme exige que les données (médicales, par exemple) et services (comme ceux offerts par Net-entreprises) sont des ressources critiques qui peuvent, à un moment donné, être invoquées par plusieurs utilisateurs, et pour différentes raisons

[5] Le terme de modification doit être entendu au sens large, comprenant à la fois la création d'une nouvelle information, la mise à jour d'une information existante, et la destruction d'une information.

(accomplissement des procédures médicales et administratives, étude de l'efficacité, etc.). Il est évident que ces ressources doivent êtres disponibles aux utilisateurs autorisés dans des délais acceptables. La criticité de ces données dépend souvent de l'application. En cas d'urgence par exemple, le médecin du SAMU, doit pouvoir y accéder, en un temps raisonnable et sans être confronté à des difficultés d'accès ou à une rupture du service délivré par le système.

Lorsqu'on s'intéresse à la disponibilité de données persistantes, on parle volontiers de *pérennité* pour insister sur la durée de leur validité plutôt que sur une accessibilité immédiate. En effet, certaines données doivent être conservées pendant des durées très longues voire illimitées. Les exemples sont multiples : données des maladies héréditaires dans le domaine médical et actes authentiques dans le domaine social. Préserver les fichiers est une tâche ardue : au-delà des capacités des supports d'archivage et des choix préalables des formats et logiciels, la majorité des techniques actuelles ne permettent pas de garantir la restitution des informations que pour une durée limitée en raison de leur obsolescence rapide. Il est certes possible de faire passer les informations d'un support à un autre, au fur et à mesure des évolutions, mais la récupération devra être sécurisée et l'information devra rester intègre.

Par ailleurs, l'indisponibilité peut être due à un acte malveillant ou à une faute accidentelle. Une attaque contre un système peut avoir simplement pour but d'empêcher le système de remplir le service pour lequel il a été conçu. Il s'agit alors d'une attaque par *déni de service*. Ces attaques consistent à faire en sorte que les actions du système ne correspondent plus à ce l'on attend de lui, soit parce que le résultat des actions effectuées par le système est erroné (service incorrect), soit parce que ce résultat n'est pas disponible en temps voulu (retard ou arrêt du service). La première catégorie d'attaque est étroitement liée à l'intégrité, étant donné qu'elle consiste à modifier l'information présente dans le système cible, afin qu'il fournisse un résultat erroné. La deuxième catégorie peut également trouver sa source dans une attaque contre l'intégrité des données ou du système, dont l'objectif est d'interrompre le traitement de l'information (ou tout au moins de le retarder), comme dans le cas de la destruction d'un lien de communication. Cependant ce type d'attaque peut également être mis en œuvre en perturbant le fonctionnement temporel du système, en surchargeant certaines des ressources dont il dépend, ou en surchargeant le système lui-même. De telles attaques peuvent, par exemple être mises en œuvre par une machine M_A qui inonde constamment un réseau R, celui-ci étant utilisé par une machine M_B pour remplir un certain service.

Enfin, il est important de noter que l'occurrence d'opérations illégitimes n'est pas forcément le signe d'une action intentionnellement nuisible d'un utilisateur. Des fautes accidentelles mettant en danger la sécurité du système peuvent parvenir du fait qu'un utilisateur, autorisé et bien intentionné, ignore certaines des propriétés attendues du système ou ne maîtrise pas complètement toutes les implications des opérations qu'il effectue. En particulier, le débranchement d'un câble par une manœuvre maladroite peut amener un serveur de fichiers à ne plus répondre, ce qui porte atteinte à la propriété de disponibilité.

1.3.5. Autres facettes de la sécurité

La sécurité peut parfois représenter d'autres caractéristiques, telles que l'intimité, l'authenticité, l'auditabilité, la pérennité, l'exclusivité, la protection contre la copie illicite de logiciels, etc. Nous conjecturons que toutes les propriétés de sécurité peuvent être exprimées en terme de disponibilité, d'intégrité et de confidentialité appliquées à des informations et des méta-informations [Deswarte 2003].

Ainsi l'*intimité*, pour traduire le terme anglo-saxon de "*privacy*", concerne le respect des libertés individuelles et la protection de la vie privée. Elle se rapporte directement à la confidentialité d'informations (données à caractère personnel) et de méta-informations (identité de l'utilisateur qui a effectué une certaine opération, qui a émis ou reçu un certain message, etc.). L'*anonymat* correspond à la confidentialité de l'identité de la personne, par exemple, qui

réalise (ou ne réalise pas) une opération. *L'analyse du trafic* est une attaque contre la confidentialité de méta-informations de communication, en vue d'obtenir connaissance de l'existence d'un canal, l'existence d'un message, des identités, des emplacements ou adresses de l'émetteur et du récepteur d'un message, de la durée de la communication, etc.

L'*authenticité* est la propriété d'être "vrai". Pour un message, l'authenticité est équivalente à l'intégrité à la fois du contenu du message (intégrité des informations) et de son origine (méta-information), ainsi qu'éventuellement d'autres méta-informations telles que l'instant d'émission ou le niveau de classification (intégrité des méta-informations). De la même manière, un document est authentique si son contenu n'a pas été altéré (intégrité des informations) et optionnellement si l'auteur déclaré est vraiment l'auteur et non un plagiaire, si la date de publication est correcte (intégrité des méta-informations), etc. De la même manière, un utilisateur prétendu est authentique si l'identité déclarée est bien la bonne identité de cette personne. *L'authentification* est le processus qui donne confiance dans l'authenticité.

L'*auditabilité* et les propriétés qui en découlent (imputabilité, irréfutabilité, etc.) [Trouessin 2000] correspond à la disponibilité et à l'intégrité d'un ensemble de méta-informations relatives à l'existence d'une opération, à l'identité de la personne qui a réalisé l'opération, à l'instant de l'opération, etc.

La propriété de *non-répudiation* garantit qu'un sujet ayant réalisé une action dans le système ne puisse nier l'avoir réalisée. La *non-répudiation* correspond donc à la disponibilité et l'intégrité de méta-informations telles que l'identité de l'émetteur (et éventuellement l'instant d'émission) d'un message pour la non-répudiation d'origine, ou telles que la réception et l'identité du récepteur d'un message pour la non-répudiation de réception.

1.4. Intrusions, attaques, vulnérabilités

Précédemment, nous avons expliqué que les fautes qui peuvent porter atteinte aux propriétés de sécurité peuvent être accidentelles, comme elles peuvent être intentionnelles, avec ou sans volonté de nuire. Dès lors, il s'agira de détailler cette deuxième catégorie de fautes et de présenter la terminologie nécessaire à son étude dans le cadre des SICSS.

Une *intrusion* est définie comme étant une faute opérationnelle, externe, intentionnellement nuisible, résultant de l'exploitation d'une vulnérabilité dans le système [Powell & Stroud 2003]. L'usage courant du mot intrusion couvre le fait de pénétrer illégalement ou sans y être convié dans un lieu, une société, etc.

En outre, le système peut être attaqué (que ce soit par un intrus interne ou externe) sans succès. Dans ce cas, l'attaque existe, mais la protection est suffisamment efficace pour empêcher l'intrusion. Il existe toujours deux causes sous-jacentes à une intrusion (figure *1*.3) :

- une action malveillante ou *attaque* qui tente d'exploiter une faiblesse dans le système et de violer un ou plusieurs besoins de sécurité ;
- au moins une faiblesse, faille ou *vulnérabilité*, qui est une faute accidentelle ou intentionnelle (avec ou sans volonté de nuire), introduite dans la spécification, la conception ou la configuration du système.

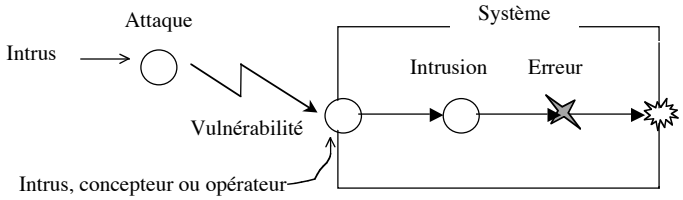


Figure 1.3 : Intrusion interprétée comme une faute composite.

En définissant une *menace* comme une violation potentielle d'une propriété de sécurité, les couples (menace, vulnérabilité) permettent d'identifier les risques auxquels le système étudié peut être soumis. Une attaque contre la sécurité du système peut provenir de l'intérieur ou de l'extérieur. Un intrus interne peut être défini comme étant un utilisateur malveillant appartenant à l'organisation, tandis qu'un intrus externe est une personne n'ayant pas de privilèges. Il est donc un individu non enregistré comme utilisateur, mais qui tente de pénétrer le système en trompant ou en contournant les mécanismes d'authentification et d'autorisation. Voici des exemples d'intrusions, interprétées en termes de vulnérabilités et attaques :

- un intrus externe qui pénètre dans le système en devinant un mot de passe ; la vulnérabilité se trouve dans la configuration du système, qui permet un mauvais choix de mots de passe (trop court, vulnérable aux attaques par dictionnaire) ;
- un intrus interne qui abuse de son pouvoir ; la vulnérabilité réside dans la spécification et la conception ou l'opération du système socio-technique (violation du principe du moindre privilège[6], procédure d'habilitation trop laxiste des personnels, etc.) ;
- un intrus externe qui utilise des moyens d'ingénierie sociale, par exemple en dupant ou corrompant un utilisateur privilégié pour le pousser à exécuter une action malveillante avantageuse pour son propre compte ; la vulnérabilité est la présence d'un utilisateur privilégié corruptible ou trop peu méfiant, ce qui est aussi une faute de conception ou d'opération du système socio-technique (procédure d'habilitation laxiste, par exemple) ;
- un intrus externe qui méne une attaque en déni de service par surcharge de requêtes (comme les attaques massives de sites webs en février 2000). La vulnérabilité réside en partie dans les spécifications mêmes du système puisqu'il est contradictoire d'exiger qu'un système soit totalement ouvert à des utilisateurs bien intentionnés et fermé aux utilisateurs malveillants. Ce type particulier d'attaque exploite aussi des fautes de conception ou de configuration dans les nombreux hôtes connectés à Internet qui ont été piratés pour insérer des processus zombies, nécessaires au montage d'une attaque distribuée et coordonnée. Une troisième vulnérabilité, qui empêche de lancer des contre-mesures efficaces, repose sur une faute de conception de la part des fournisseurs de services Internet qui n'implémentent pas de filtrage (en entrée et sortie) qui permettrait de tracer efficacement l'adresse source de l'attaque.

D'une manière générale, un utilisateur malveillant suit l'une des deux logiques suivantes : soit il contourne les mécanismes qui implémentent la politique[7] de sécurité ; soit il exploite les limites et les failles de cette politique. Cette distinction a un effet direct sur les types d'intrusions qui touchent le plus les SICSS, notamment :

- *les vols de privilèges* ou accroissement non autorisé de privilèges ; il s'agit d'un changement des privilèges d'un utilisateur sans que cela soit autorisé par la politique de sécurité : par exemple, un utilisateur qui essaye de contourner les mécanismes d'autorisation pour lire des informations confidentielles ;
- *les abus de privilèges* ou utilisation abusive des opérations autorisées ; par exemple des utilisateurs privilégiés comme les administrateurs du système, les opérateurs ou les officiers de sécurité, peuvent abuser de leurs privilèges pour effectuer des actions malveillantes.

Par ailleurs, il est intéressant de se pencher également sur les cas où une attaque contre la sécurité du système peut être accidentelle. Par exemple, le statisticien qui, sans volonté

---

[6] Le principe du moindre privilège impose que tout utilisateur ne doit pouvoir accéder à un instant donné qu'aux informations strictement nécessaires pour l'accomplissement du travail qui lui a été confié.

[7] Une politique de sécurité peut être définie par l'ensemble des règles qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées dans le système.

préalable de violer la sécurité du système, tombe "par hasard" sur des données qui, éventuellement par recoupement avec d'autres, dévoilent des informations nominatives sensibles (auxquelles il n'a pas le droit d'accéder). À ce niveau, aucun acte malveillant n'est identifié explicitement. Néanmoins, si l'utilisation de ces informations est malveillante, il s'agit bien d'un abus de pouvoir. Des notions comme les "fautes intentionnelles avec ou sans volonté de nuire" ou les "divulgations accidentelles d'information" sont donc particulièrement pertinentes dans l'étude des SICSS [Abou El Kalam *et al*. 2002b].

1.5. Techniques et mécanismes pour sécuriser un système

Afin d'éliminer les vulnérabilités, contrer les attaques, et garantir un niveau élevé de protection du réseau et du système d'information, on peut utiliser des services, des mécanismes, des outils et des procédures que l'on nomme, de façon générale, des solutions ou des mesures de sécurité. Par exemple, un service d'identification et d'authentification aide à réduire le risque d'intrusion dans un système. Les politiques de sécurité seront présentées comme un dispositif nécessaire pour renforcer la sécurité des systèmes. Puis il conviendra d'aborder succinctement la manière avec laquelle on peut les construire et les implémenter. Nous expliquons également d'autres contre-mesures pour renforcer la sécurité comme les mécanismes cryptographiques, le cloisonnement, l'audit, la détection d'intrusion et la tolérance aux intrusions.

1.5.1. Politiques de sécurité

Dans un système informatique, l'autorisation a pour but de ne permettre que les actions légitimes, c'est-à-dire à empêcher qu'un utilisateur puisse exécuter des opérations qui ne devraient pas lui être permises [Deswarte 2003]. Pour définir quelles sont les opérations autorisées et celles qui sont interdites, il faut établir une *politique de sécurité* ou "*doctrine de sécurité*". Les ITSEC [ITSEC 1991] définissent une politique de sécurité comme étant « l'ensemble des lois, règles et pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système spécifique ». À cet égard, pour construire une politique de sécurité il faut :

- d'une part, définir un ensemble de *propriétés* de sécurité qui doivent être satisfaites par le système ; par exemple "une information classifiée ne doit pas être transmise à un utilisateur non habilité à la connaître", et
- d'autre part, établir un *schéma d'autorisation*, qui présente les règles permettant de modifier l'état de protection du système ; par exemple "le propriétaire d'une information peut accorder un droit d'accès pour cette information à n'importe quel utilisateur".

Si la politique d'autorisation est cohérente, il ne doit pas être possible, partant d'un état initial *sûr* (c'est-à-dire satisfaisant les propriétés de sécurité), d'atteindre un état d'insécurité (c'est-à-dire un état où les propriétés de sécurité ne sont pas satisfaites) en appliquant le schéma d'autorisation. Comme on l'a déjà expliqué (voir 1.3), les propriétés de sécurité peuvent être définies en fonction de la confidentialité, l'intégrité ou la disponibilité d'informations ou de méta-informations.

Une politique de sécurité peut se développer dans trois directions distinctes : les politiques de sécurité physique, administrative et logique.

La politique de sécurité physique précise un ensemble de procédures et de moyens qui protègent les locaux et les biens contre des risques majeurs (incendie, inondation, etc.) et contrôlent les accès physiques aux matériels informatiques et de communication (gardiens, ...).

La politique de sécurité administrative définit un ensemble de procédures et moyens qui traite de tout ce qui ressort de la sécurité d'un point de vue organisationnel au sein de l'entreprise. La

structure de l'organigramme ainsi que la répartition des tâches (séparation des environnements de développement, d'industrialisation et de production des applicatifs) en font partie. Les propriétés de sécurité recherchées visent, par exemple, à limiter les cumuls ou les délégations abusives de pouvoir, ou à garantir une séparation des pouvoirs.

La sécurité logique fait référence à la gestion du contrôle d'accès logique, lequel repose sur un triple service d'identification, d'authentification et autorisation. Elle spécifie qui à le droit d'accéder à quoi, et dans quelles circonstances. Ainsi, tout utilisateur, avant de se servir du système, devra décliner son identité (identification) et prouver qu'il est bien la personne qu'il prétend être (authentification). Une fois la relation établie, les actions légitimes que peut faire cet utilisateur sont déterminées par la politique d'autorisation (ce type de politiques nous intéresse particulièrement, et sera décrit en détail dans le chapitre suivant).

L'autorisation consiste à gérer et à vérifier les droits d'accès, en fonction des règles spécifiées dans la politique de sécurité. On dit qu'un *sujet* (entité qui demande l'accès, dite aussi entité active) possède un droit d'accès sur un *objet* (entité à laquelle le sujet souhaite accéder, dite aussi entité passive) si et seulement s'il est autorisé à effectuer la fonction d'accès correspondante sur cet objet. Les droits d'accès peuvent être symboliquement représentés dans une *matrice de droits d'accès* dont les lignes représentent les sujets et les colonnes représentent les objets. Une cellule de la matrice contient donc les droits d'accès d'un sujet sur un objet. La matrice est gérée conformément aux règles définies dans la politique de sécurité.

L'autorisation est mise en œuvre par des mécanismes de contrôle d'accès. Il est généralement recommandé d'organiser ces mécanismes de façon à implémenter la notion de "*moniteur de référence*", défini dans le livre orange [TCSEC 1985]. Le moniteur de référence est un intermédiaire entre les *sujets* et les objets. Il vérifie que chaque accès d'un sujet vers un objet est garanti par un droit d'accès dans la matrice de droits d'accès ; en l'absence de ce droit d'accès, l'accès est refusé. Le moniteur de référence doit être inviolable (il ne doit pas pouvoir être modifié), incontournable (il ne doit pas être possible d'accéder à un objet sans être contrôlé par le moniteur de référence), et totalement vérifié (il ne doit comporter aucune faute de conception ou de réalisation).

Pour être à la fois incontournables et inviolables, il est souhaitable que les contrôles d'accès soient implémentés par matériel, pour pouvoir contrôler tout accès physique aux informations (mémoire, disques, canaux de communications, etc.). Le co-processeur LOCK d'Honeywell, issu d'un projet du NCSC [Saydjari *et al*. 1989], est un exemple d'implémentation de moniteur de référence par matériel. La plupart des microprocesseurs modernes offrent des mécanismes de contrôle d'accès par matériel, en particulier par la gestion de mémoire avec des registres de segments. Ces registres de segments peuvent être considérés comme des *capacités*, c'est-à-dire une implémentation par lignes de la matrice de droits d'accès : les registres de segments contiennent les références aux objets auxquels le processus en cours (sujet) peut accéder, ainsi que les droits correspondants (au moins, lire et écrire).

Malheureusement, la plupart des systèmes d'exploitation ne tirent pas parti de ces mécanismes. Dans Unix, par exemple, les contrôles d'accès sont principalement basés sur des permissions associées aux fichiers et aux répertoires. Il s'agit donc plutôt d'une implémentation de la matrice de droits d'accès par colonnes, c'est-à-dire par listes de contrôle d'accès : à chaque fichier, on associe la liste des sujets (sous Unix, utilisateur ou groupe d'utilisateur) qui peuvent accéder au fichier et les droits correspondants. Dans ce cas, le contrôle d'accès se fait uniquement à chaque ouverture de fichier. On est donc assez loin de la notion de moniteur de référence.

Cette notion de moniteur de référence est très centralisée, et donc difficile à interpréter dans un système réparti ou sur un réseau. Le livre rouge [TNI 1987] propose un schéma d'autorisation dans lequel chaque machine possède son propre moniteur de référence. Dans ce cadre, les accès des sujets aux objets locaux sont contrôlés par le moniteur de référence local,

alors que les accès aux objets distants donnent lieu à une coopération entre deux moniteurs de références. Le moniteur de référence du site, où se trouve le sujet, garantit l'identité du sujet et éventuellement ses droits. Le moniteur de référence du site de l'objet contrôle l'accès à l'objet en fonction de l'identité du sujet et éventuellement des droits transmis par l'autre moniteur de référence et en fonction des droits gérés localement. Dans ce cas, la matrice de droits d'accès est soit répartie, soit répliquée sur l'ensemble des sites (ce qui rend difficile le maintien de sa cohérence). Mais le principal inconvénient de ce schéma est que chaque moniteur de référence doit faire confiance aux autres. Ainsi, si un intrus prend le contrôle d'un site ou si l'administrateur d'un site est malveillant ou corrompu, il lui est facile, par exemple, de déguiser[8] un sujet local pour obtenir des droits indus sur des objets distants. L'intrusion dans un site donne donc des privilèges sur d'autres sites.

Des schémas d'autorisation à deux niveaux ont été proposés pour pallier ces inconvénients [Nicomette & Deswarte 1997 ; Deswarte *et al*. 2001]. Le premier niveau est constitué d'un serveur d'autorisation, capable de vérifier les droits de l'utilisateur à lancer une transaction ou "opération de haut niveau" et de générer des *preuves d'autorisation* pour l'exécution répartie de chaque élément de la transaction. Le second niveau est constitué du moniteur de référence local à chaque site, et qui vérifie que chaque requête (pour exécuter un élément d'une transaction) est accompagnée d'une preuve d'autorisation valide. Dans ce cas, l'intrusion dans un site ne donne aucun privilège sur les autres sites. Ce schéma d'autorisation est détaillé à l'aide d'une modélisation UML dans le quatrième chapitre.

D'une manière générale, les règles de la politique de sécurité sont souvent spécifiées en terme de *permissions* (par exemple tout médecin a le droit d'accéder aux dossiers médicaux de ses patients) et d'*interdictions* (par exemple, les médecins n'ont pas le droit d'effacer des diagnostics déjà établis), mais aussi en terme d'*obligations* (les médecins sont obligés de garder les dossiers médicaux pendant la durée fixée par la loi). Les politiques de contrôles d'accès classiques sont restreintes aux autorisations, voire aux interdictions. Et même si certaines politiques plus récentes spécifient des obligations [Bettini *et al*. 2002 ; Damianou *et al*. 2001], elles n'expliquent pas comment les implémenter. Nous pensons que les obligations peuvent être implémentées par des traitements automatiques. Un autre exemple concerne l'implémentation de la propriété de disponibilité. Outre l'aspect allocation des ressources [Cuppens & Saurel 1999], cette propriété peut être spécifiée par une obligation de fournir des moyens de tolérance aux fautes comme la redondance ou la diversification logicielle et matérielle. À présent, ces problèmes sont peu étudiés et méritent un approfondissement considérable.

Le plus souvent, une politique de sécurité ne peut malheureusement pas contrer toutes les attaques, et il est parfois possible qu'un utilisateur contourne les mécanismes qui l'implémentent. Dans d'autres cas, certaines des vulnérabilités d'un système sont tout simplement dues à des choix délibérés, résultant de compromis entre facilité d'utilisation, fiabilité, ou coût, d'une part, et sécurité d'autre part.

En outre, la plupart des vulnérabilités sont bien des bogues (*bugs*, en anglais), dus à la maladresse des programmeurs, ajoutée à des vérifications insuffisantes. En effet, il n'est pas toujours facile de prouver que la conception, la configuration, ou la mise en œuvre (par des mécanismes) d'une politique de sécurité sont conformes aux objectifs de sécurité attendus, et qu'ils n'introduisent aucune vulnérabilité pouvant être exploitée par un attaquant.

---

[8] Le *déguisement* (en anglais « *masquerade* ») consiste à tromper les mécanismes d'authentification pour se faire passer pour un utilisateur autorisé de façon à obtenir des droits d'accès illégitimes et ainsi compromettre la confidentialité, l'intégrité ou la disponibilité.

[9] Par exemple, l'obligation d'enregistrement des données d'audit peut être mise en œuvre par une action automatique (enregistrer ces données).

1.5.2. Autres contre-mesures

Dans certains cas, la politique de sécurité peut être incomplète (ne couvre pas tous les accès possibles, conséquence d'une faute de conception ou d'un choix délibéré), contournable ou mal implémentée (sa conception est non-conforme à la vie opérationnelle). Il convient donc de renforcer la sécurité par d'autres contre-mesures, tels les mécanismes cryptographiques, la certification, la détection d'intrusion ou la tolérance aux intrusions.

1.5.2.1 Mécanismes cryptographiques

La *cryptologie* se compose de la *cryptographie*, l'art d'écrire des secrets pour les rendre inintelligibles à des tiers, et de la *cryptanalyse*, l'art de retrouver les secrets cachés dans des informations inintelligibles. Il ne sera ici question que des éléments de cryptographie, qui sont à la base de nombreux mécanismes de sécurité : le chiffrement, le hachage et la signature.

1.5.2.1.1 Chiffrement et déchiffrement

Les fonctions de base de la cryptographie sont le chiffrement et le déchiffrement. Le chiffrement vise à assurer la confidentialité d'informations ; il consiste à transformer un texte en clair en un *cryptogramme*, à l'aide d'un *chiffre* (ou algorithme de chiffrement) et d'une clé de chiffrement. Le déchiffrement consiste à transformer le cryptogramme en un texte en clair identique à celui d'origine, à l'aide d'un algorithme de déchiffrement et d'une clé de déchiffrement (*figure 1*.4).

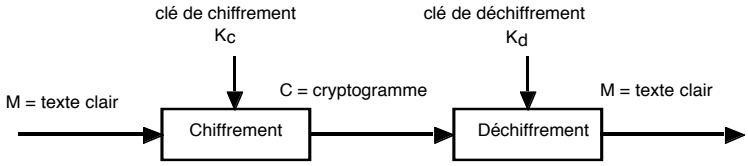


Figure 1.4 : Chiffrement et déchiffrement.

Par convention, on notera l'opération de chiffrement par des accolades :

$$M \rightarrowtail C = \{M\}_{Kc}$$

Où *M* est le message en clair, *C* le cryptogramme et *Kc* la clé de chiffrement. De même, on notera l'opération de déchiffrement par des crochets, avec *Kd* comme clé de déchiffrement :

$$C \rightarrowtail M = [C]_{Kd}$$

On parle de *chiffre symétrique* si Kd=Kc. Tous les chiffres connus jusqu'en 1976 étaient symétriques. Les chiffres symétriques sont encore très utilisés pour la confidentialité, en raison de leurs performances (on peut chiffrer plusieurs centaines de mégabits par seconde avec des matériels spécialisés). Le DES (*Data Encryption Standard*) et plus récemment l'AES (*Advanced Encryption Standard*) sont deux des chiffres symétriques les plus courants [Menezes *et al*. 1996].

Si, à l'inverse, Kc et Kd sont différents, et si, connaissant l'un, il est "impossible[10]" de trouver l'autre, on parle de *chiffre à clé publique* (ou *asymétrique*). Le chiffre à clé publique le

[10] Le terme "impossible" est utilisé dans le sens "impossible avec une puissance de calcul raisonnable et en un temps raisonnable". En particulier, pour peu que le nombre de clés possibles soit suffisamment grand, on peut considérer "impossible" l'attaque par "force brute", qui consiste à essayer

plus courant est le RSA, des noms de ses auteurs (Rivest, Shamir, Adleman). Quand on utilise un chiffre à clé publique pour la confidentialité, la clé de chiffrement *Kc* peut être connue publiquement, mais seul celui qui possède la clé de déchiffrement (secrète) *Kd* peut déchiffrer le cryptogramme.

Un chiffre est dit *hybride* s'il combine à la fois les chiffrements symétrique et asymétrique. Une des manières de faire est la suivante :

- L'émetteur génère aléatoirement une clé symétrique *c* (clé de session), considéré comme clé secrète valable pour la transmission en cours, les clés de sessions sont typiquement de 56 bits ou 128 bits.
- L'émetteur chiffre son message M avec cette clé de session (chiffre symétrique, $\{M\}_c$), et chiffre cette clé avec la clé publique Kc du destinataire (chiffre asymétrique, $\{c\}_{Kc}$), avec RSA par exemple, cette clé est souvent de 1024 bits ou 2048 bits.
- L'émetteur émet à la fois le message chiffré ($\{M\}c$) et la clé de session chiffrée ($\{c\}_{Kc}$).
- Le destinataire retrouve la clé de session en utilisant sa clé privée Kd ($c = [\{c\}_{Kc}]_{Kd}$).
- Le destinataire déchiffre ensuite le message avec cette clé de session($[M = \{M\}_c]_c$).

1.5.2.1.2 Fonctions de hachage

Une fonction de hachage à sens unique (*one-way hash function*, en anglais) permet de générer une *empreinte* de taille fixe *n* (par exemple 128 bits) à partir d'un message de taille quelconque. L'empreinte doit être une caractéristique du texte et il doit y avoir une très faible probabilité (de l'ordre de 2^{-n}) que deux messages différents aient la même empreinte. La fonction de hachage $\mathcal{H}$ doit donc être conçue de telle sorte que :

- connaissant M, il est facile de calculer l'empreinte $\mathcal{H}(M)$;
- connaissant $\mathcal{H}(M)$, il doit être "*impossible*" de trouver *M* ;
- connaissant $\mathcal{H}(M)$, il doit être "*impossible*" de trouver un texte *M*' différent de *M* et ayant la même empreinte : $\mathcal{H}(M') = \mathcal{H}(M)$;

Notons que les fonctions de hachage ne reposent sur aucun secret. Néanmoins, ce sont bien les méthodes de la cryptologie qui permettent de créer de bonnes fonctions de hachage, telles que MD5 et SHA-1, les deux fonctions actuellement les plus utilisées.

1.5.2.1.3 Signature et contrôles d'intégrité

La *signature* sert à garantir l'intégrité d'informations. Elle est obtenue en appliquant à un texte une fonction de génération de signature utilisant une clé de signature *Ks*. La vérification de signature se fait à l'aide d'un algorithme de vérification de signature et d'une clé de vérification *Kv* (figure 1.5).

L'intégrité du texte est garantie par le fait que si le texte ou la signature sont modifiés entre la génération et la vérification, l'algorithme de vérification doit rendre une réponse négative. Cette intégrité repose uniquement sur la fonction de génération de signature.

---

systématiquement toutes les clés de déchiffrement possibles jusqu'à trouver la vraie clé. De même, on considère "impossible" de deviner par hasard la clé.

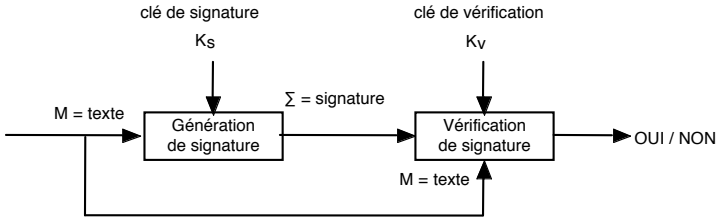


Figure 1.5 : Génération et vérification de signature.

Comme pour le chiffrement, on peut distinguer les signatures symétriques où *Ks=Kv* et les signatures à clé publique où *Kv* est une clé publique (n'importe qui peut vérifier la signature) alors que *Ks* est tenue secrète par le signataire. Dans ce cas, il doit être "impossible" de trouver *Ks* en connaissant *Kv*.

Avec une bonne fonction de hachage, il est possible de créer facilement une signature symétrique : $\Sigma = \mathcal{H}(K \mid M)$, où K est à la fois la clé de génération de signature et la clé de vérification de la signature. Dans ce cas, la fonction de génération de signature est simplement l'application de la fonction de hachage sur la clé concaténée avec le texte, et la fonction de vérification consiste à générer à nouveau une signature de la même façon sur le texte reçu et à comparer les deux signatures.

Il est également possible de générer des signatures en utilisant des algorithmes de chiffrement symétriques (par exemple, le DES en mode CBC), c'est le cas des MAC (pour *codes d'authentification de messages* en anglais). Cependant, les signatures symétriques présentent l'inconvénient suivant : la clé doit être partagée entre le signataire et le vérificateur, et tenue secrète vis-à-vis des tiers. Dans ce cas, signataire et vérificateur doivent se faire mutuellement confiance.

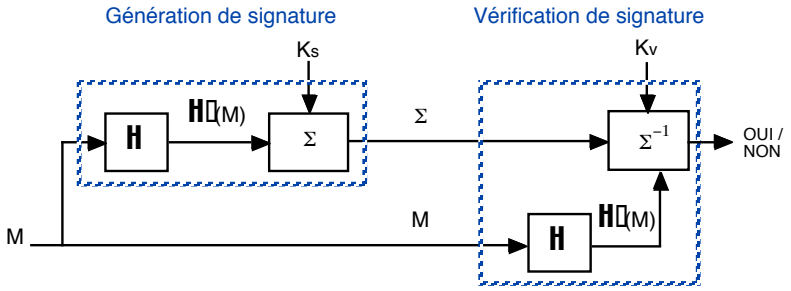


Figure 1.6 : Principe de la signature par DSA.

Cet inconvénient n'existe pas lorsqu'on utilise des signatures à clé publique, puisque seul le signataire connaît la clé de génération de signature. Comme il est souhaitable que les signatures aient une longueur fixe, quelle que soit la longueur du texte à signer, il est préférable de signer une empreinte du texte plutôt que le texte lui-même. Il faut pour cela choisir une fonction de hachage de qualité, puisqu'il serait facile, pour un faussaire de réutiliser une signature générée pour le texte *M* sur un autre texte *M'* ayant la même empreinte que *M*. L'algorithme DSA (pour *Digital Signature Algorithm*), défini dans la norme DSS (pour *Digital Signature Standard*) est un exemple d'algorithme de signature à clé publique utilisant une fonction de hachage (voir figure 1.6). Dans le cas de DSS, la fonction de hachage est SHA.

Il est également possible d'utiliser des algorithmes de chiffrement à clé publique tels que RSA pour générer et vérifier des signatures sur des empreintes de texte. Dans ce cas, la clé

publique *Kp* est la clé de déchiffrement, utilisée comme clé de vérification, et la clé privée *Ks* (maintenue secrète par le signataire) est la clé de chiffrement, utilisée comme clé de génération de signature[11] :

Génération : $\Sigma = \{\mathcal{H}(M)\}_{Ks}$ Vérification : $\mathcal{H}(M) = ?= [\Sigma]_{Kp}$

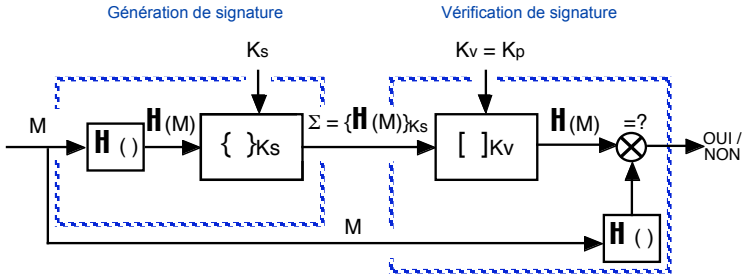


Figure 1.7 : Signature par chiffre à clé publique.

1.5.2.1.4 Certificats

Le raisonnement précédemment appliqué (chiffrement et signatures à clés publiques) suppose l'*authenticité* des clés publiques, disponibles sur un annuaire ou un serveur web par exemple. Néanmoins, cette authenticité n'est pas garantie dans un environnement ouvert tel qu'Internet, et il n'est pas impossible qu'un certain pirate Bob modifie l'annuaire ou le serveur web qui héberge les clés publiques et remplace ainsi la clé publique d'une certaine Alice par la sienne. Une fois ce déguisement commis, Bob peut lire les courriers destinés à Alice et signer des messages en se faisant passer pour Alice. En effet, si un utilisateur envoie un message chiffré à Alice, il va le chiffrer avec la clé publique de Bob (croyant que c'est la clé d'Alice). Bob pourra déchiffrer les messages destinés à Alice avec sa clé privée, et lire ainsi le courrier confidentiel d'Alice. Le raisonnement du scénario de l'usurpation de la signature est le même. Pour contrer ce type d'attaques et afin d'assurer la validité de la clé publique, il a fallu créer un mécanisme supplémentaire, le *certificat électronique*.

Un *certificat* permet d'établir un environnement de confiance entre deux entités distantes ayant besoin de communiquer entre elles et de s'échanger des informations non-répudiables (nécessité de signature) ou confidentielles (application de chiffrement). En effet, un certificat est souvent destiné à remplir trois rôles : authentification de l'émetteur, garantie de l'intégrité des documents, et éventuellement un horodatage.

Selon la norme X509 V3, un certificat électronique doit contenir notamment : le nom de *l'autorité de certification*, le nom et le prénom de la personne, son entreprise, son adresse électronique, sa clé publique, les dates de validité du certificat ainsi qu'une signature électronique (figure 1.8). Cette signature, calculée sur les informations contenues dans le certificat, est l'empreinte de ces informations chiffrée avec la clé privée de l'autorité de certification qui a délivré ce certificat.

Quand Alice et Bob veulent communiquer de manière sûre, par exemple lorsqu'elle veut lui envoyer un message chiffré, le logiciel de messagerie d'Alice a besoin de connaître la clé

[11] Nous avons décrit séparément les mécanismes de chiffrement et de signature. Mais il est possible de cumuler les deux fonctions, par exemple, en envoyant un message chiffré et signé. Les logiciels courants appliquent la signature avant le chiffrement à l'émission, et le déchiffrement puis la vérification de la signature à la réception. Toutefois, il est possible d'inverser l'ordre de réalisation de ces opérations.

publique de Bob. S'il ne la connaît pas, il peut interroger l'annuaire électronique pour récupérer un certificat de Bob. Ce certificat est signé avec une autorité AC_{Bob}. Le logiciel de messagerie peut vérifier la signature de ce certificat pour s'assurer que ce document a bien été crée par l'autorité AC_{Bob} et qu'il n'a pas été modifié. Avec cette assurance, le logiciel de messagerie peut récupérer la clé publique de Bob contenue dans ce certificat. La vérification du certificat et l'extraction de la clé publique sont schématisées dans la figure ci-dessous.

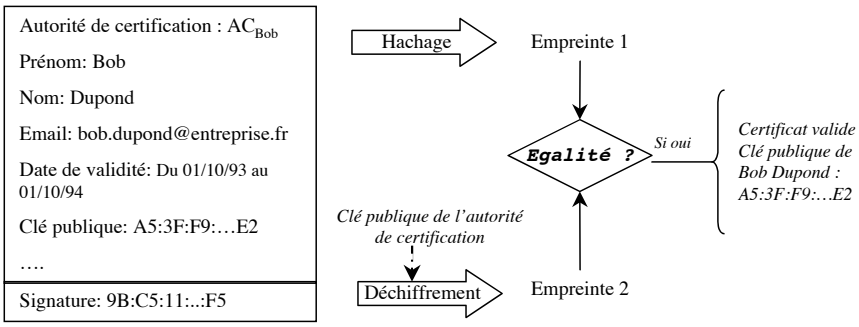


Figure 1.8 : Vérification de certificat et récupération de la clé publique

Le processus ainsi défini considère une autorité de certification. Celle-ci peut être vue comme une structure technique et administrative qui :

- génère un couple de clés publique-privée pour elle-même ;
- diffuse la valeur de sa clé publique auprès des structures qu'elle connaît et des annuaires ; l'un des types d'annuaires reconnus, et implémentés par les principaux outils, est LDAP (pour *Light Directory Access Protocol* en anglais) ;
- crée, délivre et révoque les certificats des utilisateurs qu'elle gère.

Une Infrastructure de Gestion de Clés (IGC ou PKI pour *Public Key Infrastructure* dans la terminologie anglaise) recouvre l'ensemble des services mis en œuvre pour assurer la gestion complète des clés publiques, c'est-à-dire l'enregistrement des utilisateurs et la vérification des attributs, la génération de certificats, la publication des certificats valides et révoqués, l'identification et l'authentification des utilisateurs, l'archivage des certificats, etc.

Plusieurs composants fondamentaux sont nécessaires pour la mise en œuvre d'une IGC, notamment :

- l'autorité de certification ;
- l'autorité d'enregistrement, qui est l'autorité de réception des utilisateurs qui désirent obtenir un certificat ; elle vérifie l'identité du demandeur et ses autres attributs, s'assure que celui-ci possède bien un couple de clés privée-publique, récupère la clé publique du demandeur, et transmet ensuite ces informations ainsi que les autres attributs à l'autorité de certification ;
- un service de publication ou autorité de validation ;
- l'annuaire qui contient les clés publiques, les certificats distribués, ainsi que les listes de certificats révoqués. Il est généralement basé sur un service LDAP.

1.5.2.2 Cloisonnement et pare-feux

Le cloisonnement est un bon principe de sécurité : isoler tout ce qui n'a pas besoin de communiquer. Par exemple, il est bon d'isoler les systèmes de développement des systèmes d'exploitation : cela rend plus difficiles les fraudes par les développeurs de logiciels. Il est bon

également de spécialiser les centres selon la sensibilité des informations traitées : les serveurs d'informations publiques (non-classifiées) ne devraient pas contenir d'informations classifiées. De même, les systèmes d'audit doivent être inaccessibles des systèmes qu'ils surveillent (voir section suivante).

Les "*pare-feux*" (*firewalls* en anglais, [Cheswik & Bellovin 1994]) permettent de surveiller et de restreindre les accès de l'extérieur (par exemple, l'Internet) vers l'intérieur (une machine, un réseau local, les réseaux d'une entreprise) et l'extérieur (par exemple, l'Internet), mais aussi les accès de l'intérieur vers l'extérieur. Un pare-feu est donc l'un des mécanismes de contrôle d'accès qui peut être mis en œuvre pour implémenter les règles de la politique de sécurité.

Un pare-feu comporte essentiellement une fonction de filtrage : il ne laisse passer que les paquets provenant de certaines adresses autorisées (numéro IP + numéro de port) et à destination de certaines adresses autorisées. Mais il peut remplir d'autres fonctions complémentaires, comme la traduction d'adresses (NAT, pour *Network Address Translation*), ou jouer le rôle de mandataire d'application. La traduction d'adresses permet de gérer l'espace d'adressage du réseau interne indépendamment du réseau externe : les adresses internes ne sont pas connues de l'extérieur, elles sont traduites en adresses externes par le pare-feu. Un mandataire (*proxy* en anglais) d'application permet d'interpréter chacune des interactions d'une application (commandes, requêtes, réponses) pour vérifier que les échanges suivent bien un protocole autorisé.

1.5.2.3 Audit

L'audit sert à conserver des traces des opérations susceptibles de mettre en cause la sécurité, de façon à analyser, après coup ou en temps réel, si des malveillances ont lieu ou ont eu lieu et quels sont les moyens et les méthodes utilisés, de façon à punir les coupables et à corriger les vulnérabilités. Il faut donc enregistrer toutes les opérations liées à la sécurité, que ces opérations soient réussies (parce qu'autorisées) ou qu'elles aient échouées (empêchées par les mécanismes de contrôle d'accès). Les principales opérations à surveiller sont :

- la connexion et la déconnexion des utilisateurs ;
- la création, modification, destruction des informations de sécurité (droits d'accès, mots de passe, etc.) ;
- les changements de privilèges.

Les journaux d'audit doivent être indestructibles (sauf par les administrateurs de l'audit). Ils doivent porter sur tous les utilisateurs (y compris les administrateurs et les responsables de la sécurité) et contenir un maximum d'informations utiles (date et heure, identité de l'utilisateur, type d'opération, référence de l'information, etc.). Bien évidemment, l'administrateur de l'audit doit être indépendant des administrateurs du système surveillé, et il est souhaitable que le système surveillé ne puisse pas accéder au système d'audit.

Les journaux d'audit sont en particulier l'une des sources d'informations des systèmes de détection d'intrusion.

1.5.2.4 Détection d'intrusions

Il existe deux types de systèmes de détection d'intrusions (ou IDS pour *Intrusion Detection System*) :

- les IDS sur réseau (*Network Based IDS*), qui observent les paquets circulant sur le réseau ; ce sont des machines indépendantes dédiées à la détection d'intrusions ;
- les IDS sur hôte (*Host Based IDS*), qui observent le comportement du système, en particulier les appels systèmes, ou qui analysent les informations d'audit ; il s'agit alors de fonctions intégrées au système qu'ils observent.

Les techniques de détection d'intrusions se répartissent en deux classes (figure 1.9) : détection d'anomalies, aussi appelée approche comportementale, et détection d'attaques, dite également approche par scénario.

La détection d'anomalies (« *anomaly detection* » en anglais) consiste à comparer le comportement observé d'un utilisateur à une référence de comportement normal de cet utilisateur. Toute déviation entre les deux comportements déclenche une alerte. Différentes méthodes ont été proposées pour définir ce qui est normal : des outils statistiques, des systèmes experts, des méthodes inspirées de l'immunologie, des approches bayésiennes, etc.

La détection d'attaques (« *misuse detection* » en anglais) est fondée sur la comparaison du comportement observé avec une référence correspondant à des scénarios d'attaques connus. Le principe consiste à considérer que tout ce qui est décrit dans la base d'attaques est reconnu comme intrusif ; le reste est considéré comme normal. De nombreux outils du marché utilisent cette approche, on peut citer à titre d'exemple, RealSecure, NFR, Dragon et Snort.

Ces deux types de détection se distinguent par leurs taux – théoriques – de fausses alertes (faux positifs) et de non-détections (faux négatifs). Dans le cas de la détection d'anomalies, on peut en général "régler le gain" du détecteur (par analogie avec les systèmes radar), pour choisir un point de fonctionnement correspondant à un bon compromis entre ces deux taux. Les techniques de détection d'attaques, quant à elles, ont l'avantage d'identifier quel type d'attaque est en cours (avec relativement peu de faux positifs, du moins en théorie), mais ne permettent de détecter que les symptômes d'attaques connues. Dans les deux cas, il ne s'agit que de contrôles de vraisemblance, donc imparfaits. En fait, il faut considérer que les IDS sont surtout une aide à l'administration de la sécurité : sans IDS, l'administrateur ne peut détecter les attaques que par leurs effets.

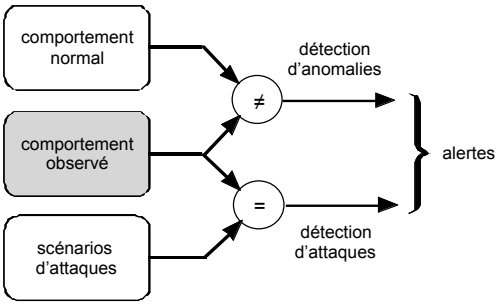


Figure 1.9 : Techniques de détection d'intrusions

Des travaux plus récents tentent de disposer, à terme, d'un système de détection d'intrusions global. Il prend en entrée aussi bien des données réseaux (provenant des IDS sur réseau) que des données systèmes (provenant des IDS sur hôte), en les analysant selon une méthode croisée utilisant à la fois l'approche comportementale et l'approche par scénario [Debar & Wespi 2001]. En outre, ces travaux font coopérer différents outils afin de tirer partie des forces de chacun pour limiter le taux de faux négatifs d'une part, et de corréler les alarmes émises afin de limiter le taux de faux positifs d'autre part :

- Il est très rare qu'une attaque génère une seule alarme. La corrélation permet de grouper les alarmes relatives à une même attaque, d'étudier les différentes attaques en cours, d'évaluer globalement la situation et de préparer une réponse appropriée.
- Étant donné que les IDS génèrent de nombreux faux positifs, la corrélation permet, en utilisant plusieurs sources de données, de vérifier la pertinence des alarmes et d'affiner le

diagnostic par le croisement de plusieurs alarmes ou la recherche d'informations complémentaires.

- Le coût de la collecte et de l'analyse d'informations par un outil de détection d'intrusions est d'autant plus élevé que la source d'informations est précise. La corrélation permet d'adapter la quantité d'informations collectées aux menaces potentielles dont les alarmes indiquent la présence.

1.5.2.5 Tolérance aux intrusions

Il est illusoire de croire qu'on peut éviter les attaques sur les systèmes de grandes tailles. De même il est impossible d'éliminer toutes les vulnérabilités. Il faut donc s'attendre à ce que certaines attaques réussissent, c'est-à-dire produisent des intrusions. D'ailleurs, le terme intrusion doit être pris dans un sens large, puisqu'il ne faut pas considérer que tous les intrus sont externes. Un grand nombre d'entre eux sont des utilisateurs enregistrés qui tentent d'étendre leurs privilèges, voire des utilisateurs privilégiés qui abusent de leurs privilèges.

Puisqu'il est inévitable que des intrusions se produisent, il serait intéressant de tolérer les intrusions, c'est-à-dire de faire en sorte que l'intrusion dans une partie du système n'ait pas de conséquence sur sa sécurité. Pour cela, on pourrait utiliser les techniques développées dans le cadre plus général de la tolérance aux fautes. Mais cela pose deux problèmes principaux :

- si un attaquant a réussi à s'introduire dans une partie du système, il ne doit pas lui être trop facile de réussir la même attaque sur une autre partie ; cela signifie que chaque "partie" soit suffisamment sécurisée et, de préférence, qu'elle soit diversifiée, c'est-à-dire que ses constituants ne présentent pas les mêmes vulnérabilités [Deswarte *et al.*1999] ;
- il ne faut pas qu'une seule intrusion dans une partie du système fournisse à l'attaquant des informations sensibles ; ceci est d'autant plus important que la redondance, nécessaire à la tolérance aux fautes, peut fournir plus d'occasions d'attaques aux pirates éventuels.

Une technique de tolérance aux intrusions a été développée pour préserver la confidentialité tout en permettant de tolérer les fautes accidentelles et les intrusions, y compris par des utilisateurs privilégiés : la *fragmentation-redondance-dissémination*, [Fabre *et al.*1996]. Cette technique est fondée sur le principe d'utilisation de la répartition d'un système sur un réseau local de façon à ce qu'une intrusion ne mette pas en défaut la confidentialité, l'intégrité ou la disponibilité du système. La fragmentation consiste donc à découper les informations sensibles en fragments de telle sorte qu'un fragment isolé ne contienne pas d'information significative (confidentialité). On ajoute de la redondance à ces fragments de façon à ce que la modification ou la destruction de fragments n'empêche pas la reconstruction de l'information (intégrité et disponibilité). Enfin, la dissémination vise à ce qu'une intrusion ne donne accès qu'à des fragments isolés. La dissémination peut être topologique, en utilisant des sites de stockage différents, ou en transmettant les fragments sur des canaux de communications indépendants. Elle peut être temporelle, en transmettant des fragments dans un ordre aléatoire et en y ajoutant éventuellement des faux fragments de bourrage. La dissémination peut aussi porter sur les privilèges, en exigeant la coopération de plusieurs personnes ayant des privilèges différents pour accomplir une opération (séparation des pouvoirs).

Lorsque la confidentialité n'est pas critique, on peut utiliser des méthodes classiques de tolérance aux fautes, comme la détection et la correction d'erreurs, ou le masquage d'erreurs. Dans ce contexte, la *détection d'erreurs* peut reposer sur des techniques de détection d'intrusions, ou sur la comparaison de plusieurs exécutions diversifiées. La *correction d'erreurs* repose alors sur la reprise (ré-exécution à partir de sauvegardes) ou sur la poursuite (on rétablit le système dans une configuration correcte). Le *masquage d'erreurs* consiste à avoir suffisamment d'exemplaires des données et des exécutions pour pouvoir corriger les dégâts causés par les intrusions.

Récemment, le projet européen MAFTIA (pour *Malicious and Accidental-Fault Tolerance for Internet Applications*) visait à faciliter les développements d'applications Internet tolérant les intrusions [Powell & Stroud. 2003]. Des protocoles et des intergiciels ont été développés pour gérer plus facilement les communications de groupe tolérant les fautes (y compris les fautes byzantines). Ces protocoles et intergiciels ont, en particulier, permis le développement de tierces parties de confiance (par exemple, une autorité de certification) qui tolèrent les intrusions (y compris de certains des administrateurs). Des méthodes de détection d'intrusions répartie sur Internet ont été étudiées avec une attention particulière, puisque la détection d'intrusions contribue à la tolérance aux intrusions, mais c'est aussi l'une des cibles privilégiées des attaquants. Il faut donc faire en sorte que ces mécanismes de détection tolèrent eux-mêmes les intrusions.

1.5.2.6 Évaluation de la sécurité

Il est important d'évaluer la sécurité des systèmes d'information et de communication, pour savoir si on a obtenu un niveau de sécurité satisfaisant, pour identifier les points les plus critiques à surveiller ou à corriger, et enfin pour estimer s'il est rentable de mettre en œuvre telle ou telle défense supplémentaire. Trois grandes méthodes d'évaluation de la sécurité peuvent être distinguées : l'utilisation des critères d'évaluation, l'analyse des risques ainsi que les méthodes d'évaluation quantitative de la sécurité.

1.5.2.6.1 Critères d'évaluation

Les premiers critères d'évaluation de la sécurité ont été définis aux Etats-Unis dans ce qui est couramment appelé le Livre Orange ou *TCSEC* (*Trusted Computer System Evaluation Criteria*) [TCSEC 1985], ou dans les différentes interprétations associées à des livres de diverses couleurs qui l'accompagnent, comme le *Livre Rouge* ou *TNI* (*Trusted Network Interpretation of the TCSEC*) [TNI 1987]. Ces critères, fondés à la fois sur des *listes de fonctions* de sécurité à remplir et sur les *techniques employées pour la vérification*, conduisent à classer les systèmes en sept catégories (**D**, **C1**, **C2**, **B1**, **B2**, **B3**, **A1**). Pour chaque niveau, quatre familles de critères sont définies, traitant de la politique d'autorisation, de l'audit, de l'assurance et de la documentation :

- *La politique d'autorisation* stipule une politique précise à suivre (discrétionnaire ou obligatoire) en fonction des différents niveaux de certifications visés. La politique obligatoire imposée est celle définie par Bell-LaPadula [Bell-LaPadula 1976] (voir 2.2).
- *Les critères d'audit* précisent les fonctions requises en matière d'identification, d'authentification et d'audit des actions.
- *Les critères d'assurance* fixent des recommandations concernant des méthodes de conception et de vérification utilisées afin d'augmenter la confiance de l'évaluateur, il s'agit de garantir que le système implémente bien la fonctionnalité qu'il prétend avoir.
- *Les critères de documentation* spécifient les documents qui doivent être fournis avec le produit lors de l'évaluation.

Les caractéristiques principales des différents niveaux définis par le livre orange sont ainsi :

- un système classé au niveau **D** est un système qui *n'a pas été évalué* ;
- jusqu'au niveau **C1** et **C2**, le système peut utiliser une *politique discrétionnaire* ;
- pour les niveaux **B1**, **B2**, et **B3** le système utilise une *politique obligatoire* ;
- un système classé **A1** est fonctionnellement équivalant à un système classé **B3**, sauf qu'il est caractérisé par l'utilisation de *méthodes formelles de vérification* pour prouver que les contrôles utilisés permettent bien d'assurer la protection des informations sensibles.

Les TCSEC visent d'abord à satisfaire les besoins du DoD (*Department of Defense*) des États-Unis, privilégiant ainsi la confidentialité des données militaires. Par ailleurs, le manque de souplesse et la difficulté de leur mise en œuvre, ont conduit au développement de nouvelles générations de critères. À titre d'exemple abordons les critères adoptés par la Communauté Européenne [ITSEC 1991], mais d'autres pays tels que le Canada [CTCPEC 1993] et le Japon [JCSEC 1992] ont également élaboré leurs propres critères d'évaluation.

Les *ITSEC* sont le résultat d'harmonisation de travaux réalisés au sein de quatre pays européens : l'Allemagne, la France, les Pays-Bas et le Royaume-Uni [ITSEC 1991]. La différence essentielle entre les TCSEC et les ITSEC réside dans la distinction entre fonctionnalité et assurance. Une classe de *fonctionnalité* décrit les fonctions que doit mettre en œuvre un système tandis qu'une classe d'*assurance* décrit l'ensemble des preuves qu'un système doit apporter pour montrer qu'il implémente les fonctions qu'il prétend fournir.

Les ITSEC introduisent également la notion de "*cible d'évaluation*" (*TOE* pour *Target Of Evaluation*). Une TOE rassemble les différents éléments du contexte d'évaluation, dont une politique de sécurité, une spécification des fonctions requises dédiées à la sécurité, une définition des mécanismes de sécurité (optionnelle), la cotation annoncée de la résistance minimum des mécanismes, ainsi que le niveau d'évaluation visé.

Les ITSEC proposent dix classes de fonctionnalités de base :

- les classes de fonctionnalité **F-C1**, **F-C2**, **F-B1**, **F-B2**, **F-B3** sont des classes de confidentialité qui correspondent aux exigences de fonctionnalité des classes *C*1 à *A*1 dans les TCSEC ;
- la classe de fonctionnalité **F-IN** concerne les TOE pour lesquelles il y a des exigences d'intégrité (par exemple, pour les bases de données) ;
- la classe de fonctionnalité **F-AV** impose des exigences de disponibilité ;
- la classe de fonctionnalité **F-DI** impose des exigences élevées pour l'intégrité des données au cours de leur transmission ;
- l'exemple de classe de fonctionnalité **F-DC** est destinée aux TOE exigeantes en confidentialité des données au cours de leur transmission ;
- la classe de fonctionnalité **F-DX** est destinée aux réseaux exigeants en matière de confidentialité et d'intégrité de l'information.

Les différents *critères d'assurance* exigés se découpent en deux aspects : les critères d'assurance d'*efficacité* et les critères d'assurance de *conformité*. Ces critères d'assurance se découpent à nouveau en deux catégories vis-à-vis de la construction et de l'exploitation du système. Les critères d'assurance de conformité sont définis vis-à-vis de six niveaux d'exigences, numérotés de E1 à E6, qui correspondent à des contraintes de plus en plus fortes et définissent le niveau de certification atteint par une TOE. De plus amples détails sur les ITSEC et les TCSEC peuvent être trouvés dans [Branstad *et al*. 1990 ; Dacier 1994].

La tentative d'harmonisation des critères canadiens, européens et américains, a donné naissance aux *critères communs* (en anglais « *Common Critéria for Information Security Evaluation* ») [CC 1999a] qui sont maintenant une norme internationale [ISO 15408]. Ces critères contiennent deux parties bien distinctes comme dans les ITSEC : *fonctionnalité* et *assurance*. Les critères communs définissent également une *cible d'évaluation* (TOE) ainsi que les *profils de protection* [CC 1999b], déjà existants dans les critères fédéraux américains [Federal Criteria 1992]. Pour une catégorie de TOE, un profil de protection définit un ensemble d'exigences de sécurité et d'objectifs, indépendant d'une quelconque implémentation. L'intérêt de ces profils est double : un développeur peut inclure dans la définition de la TOE un ou plusieurs profils de protection ; un client désirant utiliser un système ou un produit peut

également demander à ce qu'il corresponde à un profil de protection particulier, évitant ainsi de donner une liste exhaustive des fonctionnalités et des assurances qu'il exige.

1.5.2.6.2 Analyse des risques

Un risque peut être vu comme un événement redouté, auquel on peut attribuer une fréquence d'occurrence et un coût. La littérature définit différentes notions relatives aux risques comme l'évaluation, la gestion et l'analyse des risques [Dacier 1994]. Dans notre travail, nous nous intéressons à ce dernier concept, car plus large, et englobant les deux autres.

L'analyse des risques est une discipline destinée à mettre en œuvre une politique de sécurité en tenant compte des vulnérabilités résiduelles, en évaluant leurs impacts sur la sécurité, et en calculant les rapports coûts/bénéfices apparaissant dans une organisation [ISO 17799].

D'une manière générale, une analyse des risques peut être faite selon les étapes suivantes :

- l'identification des actifs, c'est-à-dire les éléments à protéger dans le système ;
- pour chaque actif, l'identification des menaces correspondantes (contre qui ou quoi on veut protéger les actifs ?) ;
- l'analyse des conséquences de la réalisation d'une menace sur l'actif du système ;
- le calcul des risques : pour chaque menace, calculer le risque qu'elle représente ; ce risque étant évalué comme égal à la fréquence d'occurrences de la menace multipliée par la conséquence financière de sa réalisation ;
- l'évaluation des parades possibles qui identifient les contre-mesures utilisables pour parer à une menace ainsi que leur efficacité et leur coût ;
- la proposition d'un choix de parades optimales du point de vue du rapport coût/efficacité pour l'amélioration du niveau existant de sécurité ;
- la formulation d'un plan de sécurité qui constitue le résultat final de l'étude incluant le compte-rendu des résultats précédents et un plan de mise en œuvre des nouvelles mesures de protection choisies.`

La réalisation de ces différentes étapes peut s'effectuer sous divers modes opératoires. À titre d'exemple, on peut citer les approches basées sur :

- les listes de contrôle qui partent des parades disponibles et étudient la nécessité de leur mise en place ;
- l'identification de scénarios d'attaques, qui prennent en compte non seulement la possibilité de réussir à exploiter une vulnérabilité, mais également la probabilité de réussir à exploiter une succession d'attaques ;
- l'évaluation quantitative qui partant de l'identification exhaustive des actifs, des vulnérabilités et des menaces, propose une quantification des risques ; cette approche sera étudiée en détail dans la section suivante.

Le principal problème de l'analyse des risques reste celui de la collecte des données. Soit la méthode se veut exhaustive et devient alors extrêmement coûteuse en temps et en effort, soit elle utilise une méthode de sélection et s'en remet alors en pratique à la qualité de l'expertise de l'évaluateur. De plus la majorité des organisations ne dispose pas d'une modélisation réelle de leurs fonctionnements. Par conséquent, les évaluations des effets des menaces et des vulnérabilités sur la sécurité demeurent peu efficaces ou tout au moins difficiles.

1.5.2.6.3 Évaluation quantitative

Dans cette section, nous détaillons une méthode d'évaluation de la sécurité qui utilise les indicateurs quantitatifs de la sécurité, de façon à permettre aux administrateurs de surveiller au

jour le jour le niveau de sécurité et de prendre des mesures correctives en cas de baisse de ces indicateurs. Ceci permet d'évaluer la sécurité opérationnelle, c'est-à-dire correspondant à la façon d'utiliser les systèmes et ses mécanismes de protection, plutôt qu'une vision statique comme celle donnée par les critères d'évaluation ou par les méthodes classiques d'analyse de risques. C'est aujourd'hui un domaine de recherche actif, mais qui a jusqu'à présent donné peu de résultats. C'est le cas de la méthode d'évaluation de la sécurité en calculant le temps et l'effort nécessaire à un intrus pour violer les objectifs de protection [Dacier 1994]. Cette méthode utilise un formalisme basé sur les graphes de privilèges (voir 2.2.2.5) et les réseaux de Petri stochastiques, et se base sur les étapes suivantes :

- *détermination* des vulnérabilités à prendre en compte, en s'appuyant notamment sur la politique de sécurité ou sur une liste de vulnérabilités déjà identifiées ;
- *quantification* de ces vulnérabilités ;
- *évaluation* quantitative en utilisant la politique de sécurité et une représentation des vulnérabilités existantes dans le système.

Cette méthode utilise le *graphe de privilège* comme modèle de représentation des vulnérabilités d'un système [Dacier 1994 ; Dacier & Deswarte 1994]. Dans le graphe de privilège, une vulnérabilité correspond à une méthode de transfert de privilèges. Les *nœuds* du graphe représentent les différents privilèges qui existent dans le système. Un *arc* est créé entre deux nœuds si une méthode, possédant les privilèges du nœud origine, permet d'obtenir ceux du nœud destination. L'existence d'un arc entre deux nœuds dépend donc de l'état du système à un instant donné, qui peut ou non permettre l'exploitation d'une certaine vulnérabilité.

Les vulnérabilités, qui doivent tout au début de ce processus être recherchées dans le système, peuvent avoir des origines variées, comme la mauvaise utilisation des mécanismes de protection ou les délégations de pouvoirs. Des valeurs quantitatives sont ensuite associées aux vulnérabilités élémentaires afin de définir des mesures quantitatives de la sécurité globales. Par exemple, on affecte à chaque arc du graphe des privilèges un poids correspondant à l'effort nécessaire à un attaquant potentiel pour exploiter la méthode de transfert de privilèges correspondant à cet arc. Cette notion d'effort regroupe les différentes caractéristiques du processus d'attaque comme la puissance de calcul disponible pour l'attaquant.

Par ailleurs, les objectifs de sécurité définis par la politique de sécurité permettent d'identifier les différents nœuds du graphe correspondant aux attaquants et aux cibles potentiels qui sont pertinents à étudier dans un système donné.

Les travaux effectués par Ortalo ont utilisé cette méthode et ont abouti à l'élaboration du prototype ÉSOPE (pour Évaluation de la Sécurité Opérationnelle) [Ortalo *et al*.1999].

Enfin, il est important de noter que les mesures de sécurité que nous avons décrites (cloisonnement, détection et tolérance aux intrusions, chiffrement, etc.) ne devront pas être mises en place tant que l'on n'aura pas *défini* et *documenté* au préalable une *politique de sécurité*. En effet, une démarche sécuritaire repose sur une politique de sécurité pour recenser les objectifs de sécurité à atteindre, les éléments à protéger ainsi que les risques encourus (*et leurs combinaisons*). Ensuite il convient de définir comment le système évolue (*face aux risques identifiés*) et quelles sont les mesures préventives et les mécanismes à mettre en place. L'évaluation de la sécurité permet, par la suite, d'identifier le niveau de sécurité visé et de savoir (ou prouver) si la politique et les mécanismes de sécurité mis en œuvre permettent bien d'atteindre le niveau visé.

Chapitre 2. Politiques et modèles de sécurité

Préambule

Le précédent chapitre a présenté les définitions relatives à la notion de sûreté de fonctionnement. La sécurité a été présentée comme la combinaison de trois propriétés : la confidentialité, l'intégrité et la disponibilité de l'information. Ensuite, les principales techniques et mesures de sécurité ont été définies, et l'analyse a montré l'intérêt des politiques d'autorisation dans la construction de la sécurité d'un système ou d'une organisation.

Examinons maintenant les grandes familles de politiques de sécurité ainsi que les principaux modèles de sécurité représentés dans la littérature ; en l'occurrence les politiques discrétionnaires, obligatoires, à base de rôles, ainsi que les modèles HRU, Take-Grant, TAM, graphe des privilèges, etc. Nous évalurons les avantages et les limites de ces modèles et politiques de sécurité, puis les confrontons aux spécificités des SICSS, déjà identifiées. Les résultats de projets récents, s'intéressant aux problèmes de la sécurité dans le domaine médical, et plus précisément aux politiques de sécurité, seront également abordés.

Enfin, la discussion des politiques actuellement en vigueur permettra de conclure qu'elles ne couvrent pas toute la richesse des SICSS. Sur le plan réglementaire, les bases et textes légaux existent, même s'ils ne sont pas assez formalisés ou formalisables, de même qu'ils ne sont pas encore appliqués ou applicables dans les pays européens. Sur le plan technique, les notions de rôles et d'équipes sont prometteurs. Néanmoins, elles nécessitent une adaptation considérable et doivent être complétées par de nouveaux concepts.

2.1. Classification des politiques et modèles de sécurité

On considère généralement trois volets de défenses pour assurer la sécurité des systèmes d'information et de communication : physique, administrative et logique. Dans ce chapitre, on s'intéresse à la sécurité logique, et plus particulièrement aux contrôles d'accès, définis à travers les politiques d'autorisations. D'un point de vue conceptuel, il s'agit :

- d'une part, de définir un ensemble *d'objectifs* de sécurité à satisfaire. Ces objectifs portent sur la confidentialité, l'intégrité et la disponibilité, et stipulent un ensemble de contraintes sur ces propriétés ;
- d'autre part, de spécifier un ensemble de *règles* décrivant les actions autorisées (ou non), et ce conformément aux objectifs soulevés. L'ensemble de ces règles définit un *schéma d'autorisation* [Sandhu 1992].

La plupart des politiques de sécurité reposent sur les notions de *sujets*, d'*objets* et de droits d'accès. Un sujet est une entité active, correspondant à un processus qui s'exécute pour le compte d'un utilisateur. Dans ce contexte, un *utilisateur* est une personne physique connue du système informatique et enregistrée comme utilisateur ; ou un serveur, sorte de personne morale représentant des fonctions de service automatiques, tel que le serveur d'impression, serveur de base de données, serveur de messagerie, etc. Un objet[12] est une entité considérée comme "passive" qui contient ou reçoit des informations. À un instant donné, un sujet a un droit d'accès sur un objet si et seulement si le processus correspondant au sujet est autorisé à exécuter l'opération correspondant à ce type d'accès sur cet objet.

Les politiques de sécurité, ou plus précisément leurs schémas d'autorisation, se classent en deux grandes catégories : les politiques discrétionnaires (ou DAC pour *Discretionary Access Control*) et les politiques obligatoires (ou MAC pour *Mandatory Access Control*). Il existe également des variantes de ces politiques qui peuvent mieux s'adapter à des organisations particulières, comme les politiques basées sur la notion de rôles (ou RBAC pour *Role-Based Access Control*) ou encore sur la notion d'équipes (ou TMAC pour *TeaM-based Access Control*). Idéalement, il faut construire une politique de sécurité de telle sorte qu'aucune séquence valide d'applications des règles (du schéma d'autorisation) ne puisse amener le système dans un état tel qu'un objectif de sécurité soit violé. Ceci suppose l'utilisation d'une méthode formelle de construction des règles du schéma d'autorisation, à partir d'une spécification formelle des objectifs de sécurité. Les politiques d'autorisation les plus citées dans la littérature sont généralement associées à un modèle de sécurité.

D'une manière générale, un modèle peut être défini comme un formalisme (souvent mathématique) qui offre une vue subjective mais pertinente de la réalité. On modélise pour mieux comprendre le système qu'on développe, c'est-à-dire pour visualiser ses propriétés, spécifier sa structure ou son comportement, documenter et guider sa construction, etc. À partir de là, un modèle de sécurité peut être défini comme un formalisme permettant de représenter, de façon claire et non-ambiguë, la politique de sécurité. Il aide à l'abstraire (afin de réduire sa complexité) et à faciliter sa compréhension, comme il peut servir à vérifier que cette politique est complète et cohérente, et que la mise en œuvre par le système de protection est conforme aux propriétés attendues du système.

Les modèles de sécurité existants peuvent être classés en deux grandes familles :

- *des modèles généraux*, qui sont plutôt des méthodes de description formelle, pouvant s'appliquer à toute sorte de politiques. C'est par exemple le cas de :

---

[12] Cette notion d'objet est à prendre dans un sens large. Elle peut recouvrir les notions classiques d'objet des systèmes dits orientés-objets et incluant les sujets eux-mêmes.

modèles de machines à états, représentant le système comme un ensemble d'états et de transitions qui, à partir d'un état courant et une valeur d'entrée, détermine le nouvel état du système ; on peut considérer que cette famille englobe les autres (chacune se distingue par la façon de représenter un état, la fonction de transition, ou encore l'ensemble des états qui satisfont la politique de sécurité) ;

modèles basés sur les matrices d'accès, manipulant les trois concepts fondamentaux que sont les sujets, les objets et les actions ; les éléments qui se situent au croisement de la ligne *L* et de la colonne *C* correspondent aux droits que possède le sujet correspondant à *L* sur l'objet de *C* ;

les modèles de Lampson [Lampson 1971], HRU [HRU 1976] et Take-Grant [Jones *et al*. 1976] sont des exemples des premiers modèles représentés par des machines à états ou des matrices d'accès ;

- *des modèles spécifiques*, développés pour représenter une politique d'autorisation particulière ; citons à titre d'exemple :

  les *modèles fondés sur les treillis*, qui affectent à chaque utilisateur et à chaque objet un *niveau* de sécurité précis ; ce type de modèle a été associé aux politiques multi-niveaux de Bell-LaPadula [Bell-LaPadula 1976] et de Biba [Biba 1977] (*voir* 2.3) ;

  d'autres modèles (généralement moins formalisés), comme celui de Clark et Wilson [Clark & Wilson 1987] développé pour les organisations commerciales, celui de la muraille de Chine [Brewer & Nash 1989] visant à représenter les conflits d'intérêts dans les institutions financières, ou ceux à base de rôle [Sandhu *et al*. 2000], adaptés à plusieurs types d'organisations, et qui utilisent des rôles comme entité intermédiaire entre les sujets et les permissions.

Par la suite, il conviendra de détailler chacune de ces politiques et chacun de ces modèles de sécurité. Chaque classe de politiques sera associée aux modèles qui l'ont représentée, ou qui sont susceptibles de la représenter.

2.2. Politiques et modèles d'autorisation discrétionnaires (*DAC*)

2.2.1. Présentation des DAC

Dans le cas d'une politique discrétionnaire, les droits d'accès à chaque information sont manipulés librement par le responsable de l'information (généralement le propriétaire), à sa *discrétion*. Les droits peuvent être accordés par ce responsable à chaque utilisateur, à des groupes d'utilisateurs, ou bien aux deux. Ceci peut parfois amener le système dans un état d'insécurité (c'est-à-dire contraire aux objectifs de sécurité qui ont été choisis).

Prenons un exemple simple, reposant sur les mécanismes de protection d'*Unix*. Dans un tel système, les droits d'accès à un fichier sont définis et modifiables librement par l'utilisateur propriétaire du fichier (et donc par les *processus* qui s'exécutent pour son compte). Supposons que le schéma d'autorisation se définisse (informellement) de la manière suivante : *un utilisateur peut créer des fichiers dont il devient alors propriétaire* ; *le propriétaire d'un fichier peut décider quels utilisateurs sont autorisés à lire ses fichiers*. D'autre part, supposons que la politique exige de respecter l'objectif suivant : *les utilisateurs qui n'ont pas le droit de lire un fichier ne doivent pas pouvoir en connaître le contenu*. Une telle politique n'est pas réalisable par des mécanismes d'autorisation discrétionnaire, parce que[13] :

[13] En raison de sa simplicité, nous avons choisi une notation dérivée du modèle HRU. Celui-ci sera décrit en détail en 2.2.2.2.

- si s_1 est un sujet s'exécutant pour le compte de l'utilisateur u_1 propriétaire du fichier f_1, il peut donner au sujet s_2 (s'exécutant pour le compte d'u_2) le droit de lecture sur f_1 :

$$(s_1, f_1, \textit{propriétaire}) \rightarrow (s_2, f_1, \textit{lire})$$

- s_2 peut créer un fichier f_2 (dans lequel il peut donc écrire) sur lequel il peut donner le droit de lecture à s_3 (s'exécutant pour le compte d'u_3) :

$$(s_2, f_2, \textit{créer}) \rightarrow (s_2, f_2, \textit{écrire}) \wedge (s_3, f_2, \textit{lire})$$

- s_2 peut alors recopier $f_1$dans f_2 pour transmettre les informations de f_1à s_3 à l'insu du propriétaire s_1 :

$(s_2, f_1, \textit{lire}) \wedge (s_2, f_2, \textit{écrire}) \wedge (s_3, f_2, \textit{lire}) \rightarrow (s_3, k(f_1), \textit{lire})$ où ($k(f_1)$ désigne une copie de f_1)

Une politique discrétionnaire n'est donc applicable que dans la mesure où il est possible de faire totalement confiance aux utilisateurs et aux sujets qui s'exécutent pour leur compte. Une telle politique est par là même vulnérable aux abus de pouvoir provoqués par maladresse ou par malveillance. Ainsi, s'il est possible à un utilisateur d'accéder à certains objets ou d'en modifier les droits d'accès, il est possible qu'un cheval de Troie s'exécutant pour le compte de cet utilisateur (à son insu) en fasse de même. De plus, si un utilisateur a le droit de lire une information, il a (en général) le droit de la transmettre à n'importe qui.

2.2.2. Modèles associés aux DAC

Cette section présente les premiers modèles conçus ou utilisés pour représenter les politiques discrétionnaires. Notons tout de même que ces modèles peuvent éventuellement être utilisés pour représenter et formaliser d'autres types de politiques. Néanmoins, pour des raisons principalement chronologiques, il a semblé plus pertinent de les inclure dans cette section.

2.2.2.1 Modèle de Lampson

La notion de *matrice de contrôle d'accès*, dédiée à la représentation des droits d'accès (autorisations), a été introduite par *Lampson* dès 1971 [Lampson 1971]. La structure de ce modèle est celle d'une machine à états où chaque état est un triplet (S,O,M) avec : S désignant un ensemble de sujets, O un ensemble d'objets et M une matrice de contrôle d'accès. Chaque cellule $M(s,o)$ de cette matrice contient les droits d'accès que le sujet s possède sur l'objet o. Les droits correspondent généralement à des actions élémentaires comme *lire* ou *écrire*. La matrice des droits d'accès n'est pas figée. En effet, si de nouveaux objets, de nouveaux sujets ou de nouvelles actions sont ajoutées dans le système, il devient alors nécessaire d'enregistrer toutes les permissions accordées pour ces nouvelles entités. Par conséquent, la mise à jour d'une politique de sécurité exprimée par ce modèle est quelque peu fastidieuse. Enfin, ce modèle ne permet pas d'exprimer directement des interdictions ou des obligations.

Le modèle de Lampson a été progressivement amélioré pour donner naissance à d'autres modèles tels que HRU [HRU 1976] et Take-Grant [Jones *et al*. 1976].

2.2.2.2 Modèle HRU

Le modèle *HRU* s'est intéressé à la complexité de la tâche de vérification des propriétés assurées par une politique d'autorisation. Comme dans le modèle de Lampson, HRU utilise une matrice d'accès classique, la différence réside en ce que HRU précise les *commandes* qui peuvent lui être appliquées. Les seules opérations possibles sont données dans le tableau 2.1. Le format des commandes est présenté dans le tableau 2.2, où x_i est un paramètre de la commande, $a^{(i)} \in A$ est un droit, et op_i est une des six opérations élémentaires possibles.

Enter *a* **into** *M(s, o)*	**delete** *a* **from** *M(s, o)*
Create subject *s*	**destroy subject** *s*
Create object *o*	**destroy object** *o*

Tableau 2.1 : Format d'une commande HRU.

Command $\alpha(x_1, x_2, .., x_k)$
If $a' \in M(s', o')$ and $a'' \in M(s'', o'')$ and … and $a^{(m)} \in M(s^{(m)}, o^{(m)})$
Then $op_1 \boxdot op_2 \boxdot .. ; op_n$
End

Tableau 2.2 : Opérations élémentaires de HRU.

Étant donné un système, une configuration initiale Q_0, et un droit a, on dit que Q_0 est *sûr* pour a, s'il n'existe aucune séquence d'opérations qui, exécutée à partir de l'état Q_0, peut amener le droit a dans une cellule particulière (i.e. pas dans n'importe laquelle) de la matrice de contrôle d'accès dans laquelle a ne se trouve pas déjà. La démonstration de cette propriété constitue le *problème de protection* (*safety problem*). En fait, ce problème revient à vérifier qu'un schéma d'autorisation est correct vis-à-vis d'un ensemble d'objectifs de sécurité. Harrison, Ruzzo et Ullman ont démontré deux théorèmes fondamentaux concernant la complexité du problème de protection :

- le problème de protection est *indécidable dans le cas général*, c'est-à-dire, étant donné une matrice d'accès initiale et un ensemble de commandes, *il est impossible* de savoir si aucune séquence d'applications de ces commandes n'aura pour conséquence de mettre un droit particulier dans un endroit de la matrice où il ne se trouvait pas initialement ;
- le problème de protection est *décidable pour les systèmes à mono-opération*, c'est-à-dire dont les commandes ne contiennent qu'une seule opération élémentaire.

Le modèle *HRU* à mono-opération est très pratique à manipuler néanmoins, il reste trop simple pour couvrir des politiques de sécurité intéressantes. Dans la mesure où il n'y a pas d'opération élémentaire qui permette simultanément de créer un objet et d'y associer des droits, le modèle HRU à mono-opération ne permet pas d'exprimer des politiques dans lesquelles les sujets qui créent des objets se voient attribuer des droits spécifiques sur ces objets.

2.2.2.3 Modèle Take-Grant

Diverses évolutions issues du modèle HRU ont tenté de déterminer un modèle suffisamment expressif pour représenter des politiques d'autorisation sophistiquées, mais pour lequel le problème de protection reste décidable . Le modèle *Take-Grant* [Jones *et al*. 1976] est constitué d'un graphe dont les nœuds sont des sujets ou des objets, et des règles de modification de ce graphe. Il peut être vu comme une variante d'HRU à ceci près qu'il restreint les différentes commandes en les répartissant en quatre catégories (figure 2.1 ; où *P* et *R* sont des sujets, *Q* est un sujet ou un objet) :

- la commande "*create*" qui permet de créer un objet et d'attribuer initialement un droit d'accès à un sujet sur cet objet ;
- la commande "*remove*" qui permet de retirer un droit d'accès d'un sujet sur un objet ;
- la commande "*take*", représentée par un arc étiqueté *t* entre un sujet *P* et un sujet (ou objet) *R*, indique que *P* peut prendre tous les droits que *R* possède ;

- la commande "*grant*" qui permet à un sujet *P* possédant un droit d'accès α sur *Q* ainsi que le droit *g* sur un autre sujet *R*, de céder à *R* le droit α sur Q (que *P* possède sur *Q*) ;

Dans ce modèle, le graphe représentant l'état de protection du système, peut être assimilé à la matrice d'accès, et les quatre règles ci-dessus (dites de réécriture), correspondent au schéma d'autorisation, c'est-à-dire aux commandes.

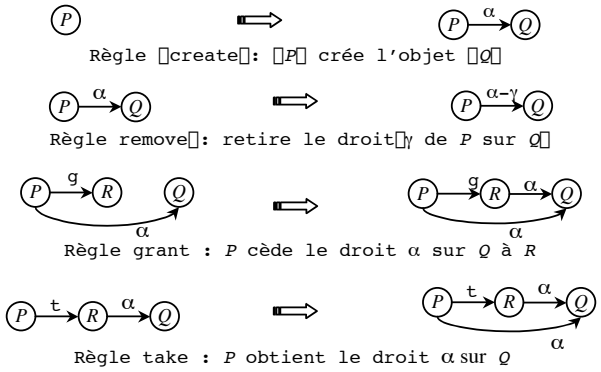


Figure 2.1 : Règles de réécriture du modèle Take-Grant

Même si ces règles peuvent paraître simples, leurs combinaisons peuvent mener le système dans des états d'insécurité. En effet, l'application successive de plusieurs règles (bien choisies) peut donner d'autres droits à des sujets, ce qui risque de compromettre certains objectifs de sécurité. L'exemple de la figure 2.2., extrait de [Dacier 1994], montre un graphe de protection qui contient deux sujets, *P* et *R* et un objet *O*. Dans l'état de protection initial, *R* possède le droit α sur *O* et le droit *t* sur *P*. Considérons un objectif de sécurité stipulant que *le système est déclaré non-sûr si P parvient à acquérir le droit* α *sur O*.

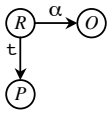
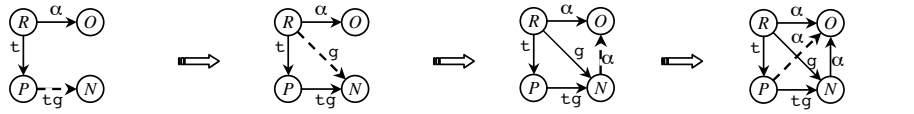


Figure 2.2 : Un exemple simple d'état de protection dans le modèle Take-Grant.

La séquence d'application des règles décrites dans la figure 2.3 indique que le système n'est pas sûr, alors que ce constat n'est pas directement explicite dans la figure 2.2.



1/ *P* crée un arc étiqueté *tg* vers un nouveau sujet *N* **2/** *R* reprend le droit (*g* sur *N*) de *P* **3/** *R* accorde le droit (α vers *O*) à *N* **4/** *P* prend le droit (α vers *O*) de *N*

Figure 2.3 : Exemple d'application des règles de réécriture dans le modèle Take-Grant.

Pour pallier ce type de problème, Jones *et al*. ont étudié le problème de protection dans le cadre du modèle Take-Grant [Jones *et al*. 1976]. Ils définissent le prédicat "*can*" de la façon suivante : « *P can* α *Q* » est vrai si et seulement s'il existe une séquence de graphes "G_1, .., G_n" telle que *P* ait le droit α sur *Q* dans le graphe G_n. Jones *et al*. définissent les conditions nécessaires et suffisantes pour que le prédicat soit satisfait. Ils établissent également l'existence d'une solution algorithmique de complexité linéaire permettant d'établir si le prédicat est

vérifié. Toutefois, les hypothèses sous-jacentes à ce modèle sont assez peu réalistes. En effet, et comme on peut le constater avec l'exemple présenté, s'il est vrai que *P* peut parvenir à acquérir le droit α sur *O*, il ne peut le faire que si *R* collabore avec lui. En réalité, il est difficile d'imaginer que tous les sujets vont collaborer afin de mettre la sécurité en péril. Une telle hypothèse est donc *de pire cas* sur le comportement des utilisateurs du système. Plusieurs raffinements des propriétés démontrables grâce au modèle Take-Grant ont été proposés, notamment afin de lever cette hypothèse et de se concentrer sur les cas où un utilisateur [Synder 1981] ou un ensemble d'utilisateurs [Dacier 1993] tentent de mettre en défaut les objectifs de sécurité. L'étude des propriétés de ce modèle dans des cadres spécifiques a fait l'objet de nombreux travaux, notamment ceux recensés dans [Dacier 1994].

2.2.2.4 Modèle TAM

S'inspirant du modèle HRU, Sandhu a présenté un modèle appelé TAM (*Typed Access Matrix*) [Sandhu 1992]. Dans TAM, chaque objet appartient à un certain *type* qui ne peut changer. Les commandes utilisent cette notion de type. Un modèle de sécurité utilisant *TAM* est composé d'un ensemble fini *A* de droits, d'un ensemble de types d'objets τ et d'un ensemble fini de types de sujets τ_s ($\tau_s \subseteq \tau$). Ces éléments sont utilisés pour définir l'état de protection à l'aide d'une matrice de contrôle d'accès *typée*. Le schéma d'autorisation est constitué de *A*, τ, et d'une collection finie de commandes *TAM*. Les opérations primitives de TAM sont données dans le tableau 2.3 ($t_s \in \tau_s$; $t \in \tau$). Les commandes TAM sont présentées dans le tableau 2.4, où, "$x_i:t_i$" exprime que le paramètre x_i est de type t_i.

Enter *a* **into** *M(s, o)*	**delete** *a* **from** *M(s, o)*
Create subject *s* **of type** *ts*	**destroy subject** *s* **of type** *ts*
Create object *o* **of type** *t*	**destroy object** *o* **of type** *t*

Tableau 2.3 : Opérations élémentaire de TAM.

Command $\alpha(x_1 : t_1, x_2 : t_2, .., x_k : t_k)$
If $a' \in M(s',o')$ and $a'' \in M(s'',o'')$ and … and $a^{(m)} \in M(s^{(m)},o^{(m)})$
Then $op_1 \boxdot op_2 \boxdot \ldots$; op_n
End

Tableau 2.4 : Format d'une commande TAM.

Sandhu montre qu'il est possible de résoudre le problème de protection dans bon nombre de cas pratiques, sans perdre de puissance d'expression. Il décrit un algorithme qui permet d'obtenir un *état maximal de protection* [Sandhu 1992]. Cet état se caractérise par une matrice d'accès sur laquelle on ne peut plus exécuter de règles du schéma d'autorisation.

Une version dite "*augmentée*" de *TAM*, appelée *ATAM*, a été proposée [Ammann et Sandhu 1992] afin de fournir un moyen simple de détecter l'absence de droits dans une matrice d'accès. Pour cela, le modèle *ATAM* offre la possibilité d'utiliser des tests du type « $a_i \notin M(s,o)$ » dans la partie conditionnelle de la commande. La façon de gérer ce type de commande et de résoudre le problème de protection a également été définie. L'intérêt de cette démarche consiste à modéliser facilement la séparation des pouvoirs (celle-ci préconise l'intervention de plusieurs utilisateurs pour mener à bien une certaine tâche).

2.2.2.5 Graphe de privilèges

Dacier et Deswarte ont proposé une extension du modèle ATAM afin d'augmenter son efficacité, notamment dans le cas de schémas d'autorisation particuliers [Dacier & Deswarte 1994]. En effet, ils ont montré que l'introduction de privilèges Ad-hoc dans un modèle TAM peut améliorer sa complexité algorithmique dans certaines situations bien précises.

Puis ils ont proposé un autre modèle, le "graphe de privilèges", dans lequel un privilège est défini comme étant un ensemble de droits Σ qu'un sujet s peut posséder sur un objet o. Les nœuds du graphe sont des ensembles de privilèges que possèdent chaque utilisateur sur des ensembles d'objets, autrement dit, des ensembles de triplets (U, O, Σ_A) où U représente un sujet, O un objet et Σ_A un ensemble de droits. Pour chaque type d'objets θ, Σ_θ désigne l'ensemble des objets de type θ. L'existence d'un arc d'un premier ensemble de privilèges vers un second indique que la possession de ce premier ensemble permet d'acquérir le second, par application d'une ou plusieurs règles du schéma d'autorisation [Dacier & Deswarte 1994 ; Dacier 1994].

Par exemple, l'application de la règle « un sujet b peut accorder tous les droits en lecture qu'il a sur un objet de type obj_3 » créerait un nœud que l'on peut définir de la façon suivante : $N = \{(b, O, lecture) \mid O \in \Sigma_{obj3} \wedge lecture \in [b,O]\}$. Ce nœud représente un sous-ensemble des privilèges que b possède effectivement dans la matrice d'accès lorsque la règle est appliquée, mais ce sous-ensemble n'est pas figé. En effet, une telle définition désigne également les droits insérés dans la matrice après que le nœud a été créé. Ceci est possible puisque le contenu du nœud n'est jamais énuméré mais seulement défini.

À partir d'un état initial, l'application successive des règles du schéma d'autorisation conduirait à l'obtention (pour tout sujet U) de M_U, l'ensemble maximal de privilèges que U pourrait obtenir. Cet ensemble correspond à la ligne de la matrice représentant l'état maximal, au sens de TAM. En pratique, la construction du graphe de privilège (ses nœuds et ses arcs) se fait progressivement par application des règles qui composent le schéma d'autorisation. Pour cela, Dacier et Deswarte ajoutent deux opérations primitives aux six autres définies dans TAM : *make_edge* et *make_node*. La première crée un nœud tandis que la deuxième crée un arc dans le graphe (à conditions que ceux-ci n'existent pas déjà). L'ensemble des opérations est d'abord utilisé pour réécrire les règles du schéma d'autorisation sous forme de commandes.

Il est important de noter qu'à aucun moment, M_U ne devrait être calculé, et que le problème de protection revient tout simplement à trouver un chemin dans le graphe des privilèges selon le processus suivant :

- pour chaque sujet U dans l'état de protection initial, créer un noeud défini par : $N_U = \{(U, O, \Sigma_A) \mid O \in \Sigma_\tau \wedge (\Sigma_a = (A \cap [U,O])) \wedge \Sigma_A \neq \varnothing\}$; (Σ_a désigne l'ensemble des droits de type a), à tout moment, cette définition représentera les privilèges présents dans la matrice pour cet utilisateur ;
- appliquer les commandes jusqu'à atteindre un état maximal, dans ce cas, l'état maximal est caractérisé à la fois par le graphe construit et par la matrice d'accès associée ;
- reformuler le problème de protection en termes de deux ensembles de nœuds conflictuels et chercher si un chemin existe entre ces deux ensembles.

En outre, dans leur modèle "graphe de privilèges", Dacier et Deswarte ont pu établir les conditions nécessaires qui, si elles sont satisfaites, permettent d'avoir un gain considérable en complexité algorithmique par rapport à TAM. Ces conditions ainsi qu'une application de ce modèle dans le cadre du système d'exploitation UNIX sont données dans [Dacier 1994].

2.3. Politiques et modèles d'autorisation obligatoires (*MAC*)

Certes les politiques discrétionnaires présentent plusieurs inconvénients, notamment les fuites d'informations et la vulnérabilité aux chevaux de Troie. Pour pallier ce type de problèmes, les politiques obligatoires décrètent des règles incontournables destinées à forcer le respect des exigences de sécurité. Ainsi, les politiques multi-niveaux affectent aux objets et aux sujets des niveaux non-modifiables par les usagers, et donc qui limitent leur pouvoir de gérer les accès aux données qu'ils possèdent. Les politiques de Bell et LaPadula visant à assurer la confidentialité, et celle de Biba s'intéressant à l'intégrité, en sont les exemples les plus anciens. D'autres politiques ont été développées pour les systèmes commerciaux et pour les institutions financières.

2.3.1. Les politiques multi-niveaux

2.3.1.1 Politique du DoD et modèle de Bell-LaPadula

La politique *multi-niveaux* de Bell-LaPadula est une politique obligatoire, développée pour le *DoD* (*Department of Defense*) des Etats-Unis [Bell-LaPadula 1976]. Elle a été mise au point au moment où les efforts se concentraient pour concevoir un système d'exploitation multi-utilisateurs. Les permissions d'accès sont définies à travers une matrice de contrôle d'accès et un ensemble de niveaux de sécurité. Cette politique considère seulement les flux d'informations qui se produisent quand un sujet observe ou modifie un objet. Destinée au domaine militaire, la propriété à assurer est la confidentialité.

Le modèle associé à la politique de Bell-LaPadula est fondé sur la notion de treillis. Il s'appuie sur l'association de différents niveaux aux sujets (niveaux d'habilitation) et aux objets (niveaux de classification). Chaque niveau $n = (cl, C)$ est caractérisé par ses deux attributs :

- C : un compartiment défini par un ensemble de catégories, par exemple {nucléaire, défense}.
- cl : une classification prise dans un ensemble totalement ordonné, par exemple : {non-classifié, confidentiel, secret}.

Pour un objet o, la classification $c(o)$ est un moyen de représenter le danger que peut constituer la divulgation de l'information contenue dans cet objet ; pour un sujet s, c'est une habilitation $h(s)$ qui désigne la confiance qui lui est accordée. Les niveaux constituent un *treillis* partiellement ordonné par une relation de dominance notée "$\leq$" et définie par :

si $n = (cl, C)$ et $n' = (cl', C')$; $n \leq n'$ (n' domine n) si et seulement si $cl \leq cl'$ et $C \subseteq C'$

Les objectifs de sécurité de cette politique sont les suivants :

- interdire toute fuite d'information d'un objet possédant une certaine classification vers un objet possédant un niveau de classification inférieur ;
- interdire à tout sujet possédant une certaine habilitation d'obtenir des informations d'un objet d'un niveau de classification supérieur à cette habilitation.

Le schéma d'autorisation associé à ces objectifs de sécurité en découle directement. On considère que l'on peut distinguer vis-à-vis de la confidentialité, parmi les différentes opérations qu'un sujet peut effectuer sur un objet, les opérations de lecture et d'écriture, et on introduit les règles suivantes, incontournables même par les propriétaires des informations :

Propriété simple : un sujet s_i ne peut lire un objet o_j que si son habilitation $h(s_i)$ domine la classification $c(o_j)$ de l'objet : $(s_i, o_j, lire) \Rightarrow h(s_i) \geq c(o_j)$

Propriété étoile : un sujet ne peut lire un objet o_j et en écrire un autre o_k que si la classification d'o_k domine celle d'o_j : $(s_i, o_j, lire) \wedge (s_i, o_k, écrire) \Rightarrow c(o_k) \geq c(o_j)$

La propriété simple interdit de lire des informations d'une classification supérieure à l'habilitation, et la propriété étoile empêche les flux d'information d'une classification donnée vers une classification inférieure, ce qui constituerait une fuite d'information : on peut vérifier facilement que si $h(s_n) < c(o_i)$, il n'existe pas de suites $\{i,j,...,k\}$ et $\{l,m,...,n\}$ telles que : $(s_l, o_i,$ *lire*$) \wedge (s_l, o_j,$ *écrire*$) \wedge (s_m, o_j,$ *lire*$) \wedge ... \wedge (s_x, o_k,$ *écrire*$) \wedge (s_n, o_k,$ *lire*$)$.

En effet, ceci conduirait (par les deux propriétés) à : $c(o_i) \leq c(o_j) \leq h(s_m) \leq ... \leq c(o_k) \leq h(s_n) \Rightarrow c(o_i) \leq h(s_n)$, ce qui est contraire à l'hypothèse de départ. Le modèle permet donc de mettre en évidence la cohérence de la politique, c'est-à-dire que le schéma d'autorisation ne peut conduire à un état où la propriété « *une information classifiée ne doit pas être transmise à un utilisateur non habilité à la connaître* » ne soit pas respectée.

Afin d'offrir plus d'expressivité, d'autres variantes de ce modèle ont été introduites, notamment en traduisant les notions d'*observation* et de *modification* de l'information, non seulement par les opérations élémentaires : *lire*, et *écrire*, mais aussi par *exécuter*, *ajouter* :

- *exécuter* : accès sans modification ni observation ;
- *lire* : observer sans modifier ;
- *ajouter* (*append*) : modification sans observation, par exemple dans le cas des écritures dans les fichiers d'audits (*audit logs*), ou en ajoutant un chemin dans le fichier de configuration (autoexec.bat, par exemple) lors d'une installation d'un logiciel ;
- *écrire* : observation et modification.

Le modèle peut être représenté par une machine à états où chaque état est défini par un triplet (S,O,M), où :

- S : un ensemble de sujets ;
- O : un ensemble d'objets ;
- A : un ensemble d'opérations d'accès ; $A = \{$*exécuter*, *lire*, *ajouter*, *écrire*$\}$;
- un ensemble de niveaux de sécurité muni d'une relation d'ordre partiel ;
- B : l'ensemble de tous les états possibles ;
- M : l'ensemble des matrices $(M_{so})s \in S ; o \in O$,
- $F \subset L_s \times L_c \times L_o$: l'ensemble des niveaux de sécurité ; dans un état donné, f_s est le plus haut niveau de sécurité qu'un sujet peut avoir ; f_c est le niveau courant de chaque sujet ; f_o est la classification de l'objet.

Les deux règles de contrôle d'accès sont :

- *La propriété simple* : un état est *sûr* si et seulement si, pour chaque sujet s qui observe un objet o, le niveau de sécurité maximal du sujet domine le niveau de sécurité de l'objet. Formellement, un état satisfait la propriété simple si :
$$\forall (s, o, a) \in b \quad a \in \{lire, écrire\} \Rightarrow f_o(o) \leq f_s(s).$$
- *La propriété étoile* : un état est *sûr* si et seulement si, pour chaque sujet s qui modifie un objet o, le niveau de sécurité de o domine le niveau courant de sécurité de s. Formellement, un état est sûr si :
$$\forall (s, o, a) \in \text{b}, \quad \text{a} \in \{ajouter, écrire\} \Rightarrow f_c(s) \leq f_o(o)$$

D'une manière plus précise :

- Si l'opération est un ajout, elle ne sera possible que si le niveau de sécurité de l'objet domine le niveau courant de sécurité du sujet ;
- si l'opération consiste à créer un objet et d'écrire dedans, le niveau de sécurité de l'objet est égal au niveau courant de sécurité du sujet ;

- si l'opération est une lecture, le niveau de sécurité de l'objet est dominé par le niveau courant du sujet.

Mais la politique de Bell-Lapabula présente plusieurs inconvénients. Le plus important est la dégradation du service provoquée par la *surclassification* des informations. En effet, au cours de sa vie, le niveau d'une information ne peut que croître : si une information non classifiée est utilisée par un sujet habilité au secret, tout objet modifié par ce sujet avec cette information sera classifié secret. Petit à petit, les niveaux de classification des informations croissent de façon automatique, et il faut les "déclassifier", manuellement par un officier de sécurité ou par un processus dit "de confiance" (*trusted process*) n'obéissant pas aux règles du modèle.

De plus, il est possible de construire un système, appelé Système *Z* [McLean 1985], qui vérifie bien les deux propriétés, et qui n'est pourtant pas sûr. Le système *Z* est un système où un utilisateur de niveau minimal met les niveaux de tous les sujets et de tous les objets au niveau minimal, et autorise l'accès de tous les utilisateurs à tous les objets. Ceci est possible car le niveau d'un objet peut, lui-même, être mémorisé dans un objet ; la valeur de ce dernier peut donc être modifiée par un utilisateur de niveau minimal (puisque l'écriture dans un niveau dominant est autorisée).

2.3.1.2 Politique d'intégrité de Biba

D'autres politiques obligatoires ont été développées pour le maintien de l'intégrité. C'est le cas de la politique proposée par *Biba* [Biba 1977]. Celle-ci applique à l'intégrité un modèle analogue à celui de *Bell-LaPadula* pour la confidentialité. À chaque sujet *s* on affecte un niveau d'intégrité *is*(*s*), correspondant à la confiance qu'on a dans ce sujet et chaque objet *o* possède un niveau d'intégrité *io*(*o*), correspondant au niveau d'intégrité du sujet qui l'a créé.

Les objectifs de sécurité de cette politique visent à :

- interdire toute propagation d'information d'un objet situé à un certain niveau d'intégrité vers un objet situé à un niveau d'intégrité supérieur ;
- et interdire à tout sujet situé à un certain niveau d'intégrité de modifier un objet possédant un niveau d'intégrité supérieur.

Le schéma d'autorisation découle de la recherche de ces propriétés, en considérant des labels d'intégrité et le fait que les opérations peuvent être regroupées en trois classes : observation, modification et invocation.

- Un sujet ne peut modifier un objet que si le label d'intégrité du sujet domine le label d'intégrité de l'objet : $(s_i, o_j, \textit{modifier}) \Rightarrow io(o_j) \leq is(s_i)$.
- Un sujet ne peut observer un objet que si le label d'intégrité de l'objet domine le label d'intégrité du sujet : $(s_i, o_j, \textit{observer}) \Rightarrow is(s_i) \leq io(o_j)$.
- Un sujet s_i ne peut invoquer un sujet s_j que si le label d'intégrité de s_i domine le label d'intégrité de s_j : $(s_i, s_j, \textit{invoquer}) \Rightarrow is(s_j) \leq is(s_i)$.

Ces règles garantissent qu'une information ne pourra pas "polluer" celle d'un niveau d'intégrité supérieur. Cela dit, le modèle de Biba présente un inconvénient analogue à celui de Bell-LaPadula : la dégradation des niveaux d'intégrité. Si une information d'un niveau d'intégrité donné est utilisée par un sujet d'un niveau inférieur, tous les objets modifiés ou créés par ce sujet à partir de cette information seront d'un niveau d'intégrité inférieur. Il faut alors remonter artificiellement le niveau d'intégrité de certains objets par des sujets "de confiance", autorisés à violer la politique de sécurité.

Cette politique a été étendue par Totel pour permettre la collaboration de logiciels de différents niveaux de criticité dans un même système [Totel 1998].

2.3.2. Politiques de Clark et Wilson

Dans l'environnement commercial, la prévention contre les modifications non autorisées de l'information (essentiellement contre les fraudes et les erreurs) est un atout primordial. Et même si un utilisateur est autorisé à manipuler certaines données, cela ne doit pas pouvoir entraîner une perte ou une falsification des actifs (capitaux, informations). Afin de répondre à ce type d'objectifs, Clark et Wilson proposent un autre type de politique obligatoire pour l'intégrité [Clark & Wilson 1987]. Celle-ci vise à assurer les besoins suivants :

- Une *cohérence* (uniformité interne) des données faisant partie de l'état interne du système. Ce type de cohérence peut être imposé par le système de calcul.
- Une *cohérence* entre l'état interne du système informatique et le monde réel qu'il représente.

La politique de Clark et Wilson repose sur deux anciens principes bien connus : les *transactions bien formées* et la *séparation des pouvoirs*.

Le concept de *transactions bien formées* stipule que les utilisateurs n'ont pas le droit de manipuler les données d'une manière arbitraire, mais seulement au travers des procédures de transformation spécifiques, préservant l'intégrité des données. Une procédure de transformation peut être par exemple un programme qui applique le principe de l'équilibre de la balance en comptabilité (toute modification dans les grands livres de la comptabilité doit figurer dans les deux parties de la balance).

Le concept de *séparation des pouvoirs* (*separation of duty*) est l'un des mécanismes classiques de contrôle de fraudes et des erreurs. Il est fondé sur la répartition des opérations entre plusieurs parties, et l'attribution des droits différents, mais complémentaires, à différentes catégories de personnes. Dans certaines entreprises par exemple, l'achat d'une marchandise nécessite l'intervention du service commercial qui fait la commande, du service de contrôle qui vérifie l'acquisition de la marchandise et du service financier qui paye les fournisseurs. Notons que toutes ces opérations sont accompagnées, à la fois de traces papiers (bon de commande, bon de livraison, facture), et de traces informatiques. Une exécution convenable de l'opération globale implique une cohérence entre les représentations interne et externe des données, c'est-à-dire la correspondance entre les procédures informatiques et celles du monde réel.

Bien qu'évidentes, ces règles nécessitent des techniques pour leur mise en œuvre au niveau du système informatique. Les procédures (transactions bien formées) manipulant les données doivent être examinées et bien établies ; la capacité à les installer et les modifier doit être contrôlée ; la validité des données doit toujours faire l'objet d'une vérification ; l'affectation des droits aux utilisateurs ainsi que la séparation des pouvoirs doivent êtres menées à bien, etc.

Pour cela, la politique de Clark et Wilson commence par séparer les données manipulées en deux groupes : les données contraintes (notées CDI pour *Constrained Data Items*) et les données non contraintes (notées UDI pour *Unconstrained Data Items*). Le premier groupe désigne les données soumises à des règles de manipulation strictes visant à conserver leur intégrité. En effet, les différentes opérations de transformation doivent permettre de garantir que l'intégrité des données contraintes persiste. Le deuxième groupe concerne les données dont l'intégrité n'est pas garantie et qui peuvent être manipulées arbitrairement. Ces données sont importantes car elles représentent la manière selon laquelle une nouvelle information est introduite dans le système, comme les entrées tapées par l'utilisateur sur le clavier.

Afin de représenter les règles appliquées dans leur politique de sécurité, Clark et Wilson définissent leur modèle de la manière suivante :

- l'ensemble des utilisateurs, U = $\{u_1, u_2, \ldots\}$;
- l'ensemble des données contraintes, CDI = $\{CDI_1, CDI_2, \ldots\}$;
- l'ensemble des données non contraintes, UDI = $\{UDI_1, UDI_2, \ldots\}$;

- un ensemble d'opérations de vérification de l'intégrité des données, IVP = {IVP_1, IVP_2, …}. Ainsi, les CDI sont certifiées par des IVP ;
- l'ensemble des opérations de transformation des données (*Transformation Procedures,* en anglais), TP = {TP_1, TP_2, ...} ; les CDI ne peuvent être manipulées que par des TP ;
- une relation $\mathcal{R}_c$ liant chaque opération de transformation TP_i, à un sous ensemble c de CDI ; $\mathcal{R}_c$ précise les données que l'opération peut manipuler ;
- Une relation $\mathcal{R}_u$ liant chaque utilisateur u et chaque opération de transformation TP_i, à un sous-ensemble de CDI ; $\mathcal{R}_u$ précise les données contraintes qu'un utilisateur est autorisé à manipuler (par le biais de TP_i).

L'objectif de cette politique de sécurité est de garantir l'intégrité des données contraintes et de satisfaire le principe de séparation de pouvoirs. Pour mettre en œuvre la politique d'intégrité, les règles obligatoires du modèle de *Clark et Wilson* imposent que :

- les CDI soient dans un état valide, vérifié par les IVP ;
- toutes les opérations de TP doivent être certifiées ; si une donnée contrainte est valide avant l'exécution d'une opération TP_i, elle reste toujours valide après l'exécution de TP_i ; de plus, les opérations de TP ne peuvent être effectuées que sur des données contraintes spécifiées par $\mathcal{R}_c$;
- chaque utilisateur n'effectue des opérations de transformation sur des données contraintes que si celles-ci lui sont associées par $\mathcal{R}_u$;
- la relation $\mathcal{R}_u$ doit être certifiée afin de refléter les besoins de séparation de pouvoirs ;
- le système doit authentifier l'identité de chaque utilisateur souhaitant exécuter une opération de TP ;
- avant d'être exécutées, les opérations de TP doivent enregistrer leurs identifiants dans les journaux d'audit ; ceux-ci sont considérés comme données contraintes ;
- chaque opération TP_i qui accepte une donnée UDI_j en entrée doit garantir que, quelle que soit la valeur de UDI_j, soit TP_i n'effectue que des transformations conduisant à une donnée contrainte valide CDI_k, soit UDI_j est rejetée.

En dépit de sa présentation relativement informelle, le modèle de Clark et Wilson met en avant clairement un certain nombre de préoccupations qui peuvent apparaître dans une organisation commerciale. Tout d'abord, ce modèle s'intéresse à l'assurance de la propriété d'intégrité par le biais de l'utilisation de procédures de transformation certifiées. Il sensibilise ainsi à la *certification*, activité importante lors de la conception d'un système. En outre, ce modèle met en évidence deux préoccupations importantes dans bon nombre de structures : la *traçabilité*, c'est-à-dire la possibilité de reconstituer les actions importantes du point de vue des objectifs de sécurité, et la *séparation des pouvoirs*.

Enfin, ce modèle de sécurité admet, quoi que de manière un peu implicite, la possibilité que le système dévie de son fonctionnement normal. En effet, si les propriétés d'intégrité ne sont pas satisfaites à un moment donné, l'existence de procédures de validation de l'intégrité, ajoutée au fait que les données non-contraintes sont acceptées par certaines procédures de transformations, offre des moyens de détection d'une infraction aux objectifs de sécurité et de retour vers un état satisfaisant. L'existence d'un historique de l'exécution du système permet alors d'identifier les comportements erronés qui ont pu être à l'origine d'une violation des objectifs de sécurité (par exemple, une faute dans l'implémentation d'une opération, non détectée par la certification). Ce souci de fournir, outre des mécanismes garantissant la sécurité du système, des mécanismes capables de fonctionner en l'absence de propriétés de sécurité attendues pour le système, participe à la volonté de définir des politiques de sécurité

applicables dans un environnement moins rigide que ceux dans lesquels des politiques de sécurité comme les politiques multi-niveaux sont utilisées.

2.3.3. Politique de la muraille de Chine

La politique dite de la Muraille de Chine [Brewer & Nash 1989] est une politique obligatoire pour maintenir la confidentialité des informations. Elle correspond à une réglementation, imposée aux agents de change britanniques, pour résoudre les problèmes de sécurité liés aux conflits d'intérêt dans les institutions financières. Dans ce type d'organisations, un grand intérêt est porté au cloisonnement des différentes informations concernant des clients concurrents. En effet, si un organisme financier est amené à traiter des opérations pour le compte de deux clients en concurrence directe, le personnel de cet organisme ne doit pouvoir accéder qu'aux informations concernant l'un de ces deux clients. Au départ, l'utilisateur a le libre choix, mais une fois que les informations d'un client connues, tout accès aux informations concernant l'autre doit être interdit. Sa décision dresse donc devant lui une barrière qu'il ne peut plus franchir, d'où le nom de muraille.

La formalisation de cette politique nécessite l'identification de trois ensemble : C, l'ensemble des compagnies ; S, l'ensemble des sujets (analystes financiers) ; et O, l'ensemble des objets (fichiers). Le système classe l'information en trois niveaux hiérarchiques :

- le niveau le plus bas contient les données de toutes les compagnies ;
- le niveau intermédiaire, regroupe les objets qui concernent la même compagnie ;
- le niveau supérieur regroupe les données des compagnies en compétition.

À chaque objet, sont associés le nom de la compagnie à laquelle il appartient ainsi que la classe de conflit d'intérêt qui contient ses données. Ce modèle considère deux fonctions :

- la fonction X, tel que $X(o_i)$ désigne la classe de conflit d'intérêt de o_i. Autrement dit, pour un objet donné, X fournit l'ensemble de toutes les compagnies en compétition autour de cet objet ;
- la fonction Y, tel que $Y(o_j)$ désigne l'ensemble des données de la compagnie de o_j.

  Une information est aseptisée si elle a été purgée des détails sensibles et elle n'est pas sujette à des restrictions d'accès. Pour ce type d'information, on a : $X(o) = \varnothing$.

Notons qu'un conflit d'intérêt peut surgir, non seulement à cause des objets consultés à un moment donné, mais aussi en raison des accès antérieurs. Pour enregistrer l'historique des actions, le modèle utilise une matrice N_{so} définie par :

$N(s,o) = 1$ si s a déjà accédé à o ; $N(s,o) = 0$ sinon.

L'objectif de cette politique de sécurité est de garantir qu'aucun utilisateur n'accède simultanément à des données appartenant à des ensembles en conflit d'intérêt. Cet objectif est obtenu en imposant les deux règles suivantes :

(1) *Propriété simple* : l'accès est accordé à un sujet seulement si l'objet demandé appartient au même ensemble de données de compagnie "à l'intérieur de la muraille" (comme un objet déjà lu), ou à une classe de conflit d'intérêt complètement différente.

(2) *Propriété étoile* : un sujet s est autorisé à écrire dans un objet o, si l'accès est autorisé par la propriété simple ; et si s ne peut lire aucun objet o' appartenant à un ensemble de données de compagnies différent de celui de o, et contenant des données non aseptisées.

Étudions cette deuxième propriété à travers un exemple simple. Supposons que le système gère une banque $Banque_A$; deux compagnies en compétition C_A et C_B ; deux sujets (*analystes*) A_1 et A_2 et trois *fichiers* (pour simplifier, on les note $Y(C_A)$, $Y(C_B)$ et $Y(Banque_A)$). La propriété étoile n'interdit pas le fait que A_1 ait accès à $\{Y(C_A), Y(Banque_A)\}$ et que A_2 ait accès à $\{Y(C_B), Y(Banque_A)\}$. Mais si A_1 lit $Y(C_A)$ et l'écrit dans $Y(Banque_A)$, A_2 peut lire cette information

(puisqu'il a accès à Y($Banque_A$) alors que cette information (initialement concernant C_A) est dans la même classe de conflit d'intérêt de C_B (à laquelle il a déjà accédé). La propriété seule, seule, ne permet pas d'empêcher de telles attaques, tandis que l'attaque ne peut pas réussir si le système implémente la propriété étoile.

2.4. Politiques de contrôle de flux

Contrairement à celles qui les ont précédées, les politiques de contrôles de flux proposent des solutions pour l'identification et l'élimination des canaux cachés. Un canal caché est un chemin de communication pouvant être exploité par un processus de transfert d'information de telle sorte qu'il contourne les mécanismes de contrôle d'accès, et qu'ainsi il viole la politique de sécurité. Par exemple, il est parfois possible de transmettre de l'information, non pas au travers d'un tampon, mais grâce au message d'erreur qui est retourné par le système lorsque le tampon est plein. Le piège peut s'appuyer sur la convention que « libre = 1 » et « plein = 0 », et un cheval de Troie peut ainsi transmettre une suite de bits à l'utilisateur malveillant. Dans ce cas, on dit que cet échange d'informations a utilisé un canal caché. D'une manière générale, on distingue deux types de canaux cachés : les canaux de *mémoire* et les canaux *temporels.*

Les canaux de *mémoire* sont constitués par des moyens de mémorisation partagés entre des utilisateurs d'habilitations différentes. Si le système obéit à une politique obligatoire, il interdit bien évidemment l'utilisation de moyens directs tels que la mémoire partagée, l'envoi de messages, le partage de fichiers, la réutilisation de tampon ou de fichier temporaire, etc. Un canal caché de stockage est l'utilisation (détournée) d'un mécanisme qui permet à un processus d'écrire une information qui peut être lue par un autre processus. En fait, il est généralement possible d'identifier et de supprimer tous les canaux cachés de stockage. L'analyse des canaux cachés de stockage (avec estimation de leur bande passante) est d'ailleurs exigée dès le niveau B2 du Livre Orange [TCSEC 1985].

Les canaux *temporels* sont plus difficiles à identifier et à analyser. En fait, dès lors qu'une ressource matérielle (unité centrale, mémoire, réseau, disque, …) ou logicielle (verrou, sémaphore, système de fichiers, …) est partagée entre des utilisateurs d'habilitation différente et qu'il existe une horloge commune suffisamment précise, il est possible de créer un canal temporel en modulant l'usage de la ressource. Un exemple classique est celui de disques partagés : l'utilisateur privilégié peut positionner le bras du disque sur le cylindre extérieur (pour transmettre un « 1 ») ou sur le cylindre intérieur (pour transmettre un « 0 ») ; l'utilisateur non privilégié peut alors positionner le bras du même disque sur le cylindre intérieur : selon le temps de réponse, il saura s'il s'agit d'un « 1 » ou d'un « 0 ». Certains canaux temporels, basés sur des ressources partagées matérielles, peuvent atteindre des vitesses de transmission de plusieurs mégabits par seconde si aucune précaution n'est prise [Hu 1991].

Comme il n'est en pratique pas possible de supprimer tous les canaux cachés (en particulier temporels), on s'attachera à en réduire la bande passante, en réduisant le plus possible le nombre de ressources partagées, en affectant les ressources partagées pendant des durées fixes, en faisant fluctuer les horloges, etc. Mais ces techniques de réduction de bande passante ont une influence négative sur les performances du système. Il faut donc chercher un compromis satisfaisant. Selon le Livre Orange, une bande passante de 100 bits par seconde est inacceptable pour des canaux cachés d'un système qu'on prétend sûr. En revanche, une bande passante d'un dixième de bit par seconde ou même d'un bit par seconde peut être tolérée si le canal correspondant peut être surveillé par le système d'audit

Les *politiques de contrôle de flux* gèrent le problème des canaux cachés en considérant, non seulement des opérations de lecture et d'écriture sur des objets, mais également des flux d'informations entre sujets. Elles s'attachent donc à spécifier les canaux de transmission présents dans le système, à préciser les canaux légitimes et à identifier les canaux cachés.

Une approche originale pour la représentation des flux d'information dans un système consiste à caractériser les *dépendances causales* qui existent, à différents instants, entre les objets du système [Bieber & Cppens 1991 ; d'Ausbourg 1994]. Dans ce modèle, un système est représenté sous forme de points (o, t). Un point désigne, non pas un objet, mais l'état d'un objet o à l'instant t. Certains de ces points sont des entrées, d'autres des sorties, et tous les autres constituent des points internes au système. L'ensemble de ces points évolue avec le temps et cette évolution est due aux transitions élémentaires qui ont eu lieu dans le système. Une transition élémentaire peut, à un instant t, associer une nouvelle valeur à un objet o en ce point. Cet instant et cette nouvelle valeur dépendent donc de certains autres points antérieurs.

La dépendance causale de (o, t) vis-à-vis de (o', t'), avec $t'<t$ est notée « (o', t')→(o, t) ». La fermeture transitive de la relation « → » (notée « →* ») au point (o, t) définit le *cône de causalité* en ce point : $cône(o,t) = \{ (o', t') \quad tel\ que \quad (o', t') \rightarrow^* (o, t) \}$.

Réciproquement, on définit le *cône de dépendance* d'un point (o, t) comme un ensemble des points qui dépendent causalement de (o, t) : $dep(o, t) = \{ (o', t') \ tel\ que \ (o, t) \rightarrow^* (o', t') \}$.

Les dépendances causales représentent la structure des flux d'information dans le système. Si un sujet s possède une certaine connaissance du comportement interne du système, il est en mesure de connaître les dépendances causales. Dans ce cas, en observant une sortie particulière x_0, un sujet s peut être en mesure d'inférer toute information appartenant à $cône(x_0)$ Réciproquement en altérant une entrée x_i du système, s peut éventuellement altérer tous les points appartenant à $dep(x_i)$.

Les objectifs de sécurité de ce modèle peuvent être relatifs à la confidentialité ou à l'intégrité. Soit la notation suivante :

- Obs_s, l'ensemble des points qu'un sujet *s peut* observer, $Obs_s = \bigcup_{xo \in Os} cône(x_0)$;
- R_s, l'ensemble des points que s a le droit d'observer ;
- Alt_s, l'ensemble des points qu'il peut modifier, $Alt_s = \bigcup_{xi \in As} cône(x_i)$;
- W_s, l'ensemble des points que s a le droit de modifier dans le système;

Le système est considéré sûr vis-à-vis de la confidentialité si s ne peut observer que les objets qu'il a le droit d'observer, c'est-à-dire si $Obs_s \subseteq R_s$. De la même manière, le système est sûr vis-à-vis de l'intégrité si s ne peut agir que sur les objets qu'il a le droit de modifier, c'est-à-dire, si $Alt_s \subseteq W_s$.

En considérant un ensemble de niveaux associés aux sujets et aux objets, la propriété $Obs_s \subseteq R_s$ relative à la confidentialité peut être obtenue en imposant deux règles analogues à celles définies dans la politique de Bell-LaPadula :

- un sujet n'est autorisé à observer que les objets dont la classification est dominée par son habilitation ;
- si un objet o' dépend causalement d'un objet o, alors la classification de o' doit dominer la classification de o.

Ce modèle est particulièrement intéressant parce qu'il introduit une nouvelle manière de formaliser les flux d'informations dans un système. L'intérêt principal de cette formalisation réside dans son aspect minimal : la notion de dépendance causale permet de décrire de manière strict un flux d'informations. Toutefois, les implémentations de ce modèle qui ont été réalisées semblent limitées à des applications assez spécifiques [Calas 1995].

2.5. Politiques de contrôle d'interfaces

Les politiques de contrôle d'interfaces spécifient des restrictions sur les *entrées* et les *sorties* du système qui permettent d'obtenir des propriétés de sécurité. Ce type de politiques de

sécurité s'intéresse plus directement au comportement dynamique du système et s'appuie sur des méthodes de modélisation très générales. Elles considèrent une représentation du système incluant les différents sujets et l'ensemble des *traces d'exécutions* associées à ces sujets. Une trace est définie comme l'historique des entrées, c'est-à-dire la suite ordonnée de tous les états successifs du système entre chaque entrée (ou commande) [Jacob 1988]. Certaines commandes particulières déterminent les sorties du système effectuées par un utilisateur, et on s'intéresse principalement aux propriétés que le système possède vis-à-vis de toutes ses sorties.

La principale qualité de cette approche très abstraite est d'avoir permis des avancées significatives dans la compréhension des problèmes de formalisation posés par la sécurité. Un certain nombre de propriétés importantes ont pu être étudiées et comparées en se basant sur ces modélisations très générales des systèmes [McLean 1990 ; Bieber & Cuppens 1992 ; Zakinthinos & Lee 1994].

Les principales propriétés identifiées dans la littérature, et qui constituent les différents objectifs de sécurité, sont relatives à la notion de non-interférence.

La *non-interférence* stipule qu'un groupe d'utilisateurs utilisant un certain nombre de commandes n'interfère pas avec un autre groupe, si ce que fait le premier groupe n'a aucun effet sur ce que le deuxième groupe peut observer. Les premiers modèles assurant la non-interférence ont été établis par Goguen et Meseguer [Goguen & Meseguer 1984]. Leur modèle montre que la sécurité est assurée si les entrées d'un niveau *N* n'interfèrent pas avec les sorties observables à un niveau qui ne domine pas *N*. Autrement dit, les sorties dont le niveau ne domine pas *N* sont les mêmes que l'on considère les entrées de niveau *N* ou non. Cette propriété s'applique aux systèmes *déterministes*.

McCullough [McCullough 1987] a proposé de généraliser ce modèle pour prendre en compte le non-déterminisme. Les sorties ne sont plus considérées comme fonction des entrées, mais on identifie plusieurs sorties possibles pour des entrées données. L'ensemble de ces sorties est appelé "futur possible". La propriété de sécurité est ainsi : les futurs possibles d'un niveau qui ne domine pas *N* sont les mêmes que l'on considère les entrées de niveau *N* ou non.

2.6. Politiques et modèles de sécurité par rôles (*RBAC*)

Certes, les politiques de contrôles de flux ou d'interfaces sont importantes d'un point de vue compréhension et spécification de certains types de systèmes. Néanmoins, leur mise en œuvre est très difficile et d'ailleurs, les implémentations qui en ont été faites demeurent limitées. Par ailleurs, les politiques obligatoires, en particulier les politiques multi-niveaux, imposent des contraintes fortes sur les organisations. Les relations d'ordre partiel qu'elles utilisent sont mal adaptées à la réalité des entreprises : peut-on dire qu'une information du service commercial est plus secrète ou plus critique que celle qui provient du bureau d'étude ? Ce n'est pas sur de tels critères qu'il convient de contrôler les flux d'information. Les politiques discrétionnaires sont elles aussi mal adaptées : elles sont trop laxistes, en général, et permettent difficilement de garantir des propriétés de sécurité. D'autre part, il faut constamment redéfinir les règles, chaque fois qu'un nouvel utilisateur ou un nouvel objet est introduit dans le système. Les politiques discrétionnaires sont donc très difficiles à administrer.

À l'inverse, les politiques basées sur les rôles (RBAC, pour *Role-Based Access Control* [Sandhu 2000 ; Solms & Merwe 1994 ; Lawrence 1993]) visent à faciliter l'administration de la sécurité. Un rôle représente de façon abstraite une fonction identifiée dans l'organisation (par exemple, chef de service, ingénieur d'étude, etc.). À chaque rôle, on associe des permissions (ou privilèges), ensemble de droits correspondant aux tâches qui peuvent être réalisées par chaque rôle. Enfin, et contrairement aux modèles qui ont précédé RBAC (HRU, par exemple), les permissions ne sont plus associées d'une façon directe aux sujets, mais à travers des rôles. Les deux relations de la figure 2.4 « *Détient*(*Rôle*, *Permission*) » et

« *Joue*(*Sujet*, *Rôle*) » définissent précisément les permissions accordées à chaque sujet. Un rôle peut avoir plusieurs permissions et une permission peut être associée à plusieurs rôles. De même qu'un sujet peut être membre de plusieurs rôles et inversement, un rôle peut être exécuté par plusieurs sujets. Ainsi, si le docteur Dupont est à la fois chirurgien et directeur de l'hôpital, en tant que chirurgien, il aura le droit d'accès aux dossiers médicaux, alors qu'en tant que directeur, il pourra accéder aux informations administratives.

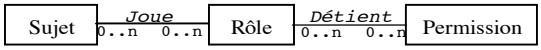


Figure 2.4 : Attribution des permissions aux sujets à travers des rôles.

L'une des manières de définir les rôles au sein d'une organisation consiste à regrouper dans chaque rôle les tâches pouvant exécuter les mêmes opérations, ensuite il s'agit d'identifier les objets que ces tâches utilisent, puis définir les droits d'accès sur ces objets, c'est-à-dire les couples (ensemble de droits, objets) et finalement, d'associer ces droits aux rôles. L'affectation des sujets aux rôles est une tâche à faire séparément, et probablement par d'autres administrateurs.

Des variantes de RBAC ont essayé de le raffiner, notamment en incluant les concepts de *session*, de *hiérarchie* de rôles et de *contraintes* sur les rôles [Sandhu 1996] [Sandhu *et al*. 1996] [Garvila & Barkley 1998] [Ahn & Sandhu 2000] [Ferraiolo *et al*. 2001]. Dans une même session, un utilisateur a la possibilité de ne pas activer tous ses rôles, mais uniquement le sous-ensemble de ses rôles nécessaire à la réalisation de la tâche à accomplir. La hiérarchie de rôles permet de mettre en place un mécanisme d'héritage des permissions entre les rôles et simplifie d'autant l'administration de ce modèle. Par exemple, comme les chirurgiens et les gynécologues sont nécessairement médecins, on assignera des permissions au rôle médecin, et seulement des permissions supplémentaires au rôle chirurgien, d'une part et au rôle gynécologue d'autre part. Des contraintes (par exemple, d'où le rôle peut-il être activé, quand peut-il être activé et quelles sont les données qu'un rôle peut manipuler ?) ont été intégrées dans des versions récentes de RBAC.

Un modèle de contrôle d'accès reposant sur la notion de rôle, est défini dans [Sandhu 1996] de la manière suivante[14] :

- U, R, P, S, respectivement des ensembles d'utilisateurs, de rôles, de permissions et de sessions ;
- $PA \subseteq R \times P$, une relation associant une permission à un rôle ;
- $UA \subseteq U \times R$, une relation associant un ou plusieurs rôles à un utilisateur ;
- $RH \subseteq R \times R$, une hiérarchie de rôles partiellement ordonnés ; $r \geq r'$ signifie que r' est un sous-rôle de r ;
- $User : S \rightarrow U$, une fonction associant chaque session s_i à un seul utilisateur $User(s_i)$, qui reste constante pour la durée de vie de la session ;
- « $Role : S \rightarrow 2^R$ », une fonction associant chaque session s_i à un ensemble de rôles $Role(s_i)$, avec : $Role(s_i) \subseteq \left\{ r \middle| (\exists r' \geq r) \quad et \quad \left(User(s_i), r'\right) \in UA \right\}$;
- une collection de contraintes qui détermine si certains éléments du modèle RBAC sont acceptables (seuls les éléments acceptables sont intégrés dans le modèle).

[14] Une modélisation algébrique plus solide de RBAC enrichie par des notions comme les sessions, la hiérarchie, les contraintes, l'activation des rôles, etc. est donnée dans [Ferraiolo *et al*. 2001].

L'analyse des politiques basées sur les rôles permet de conclure qu'elles sont relativement faciles à administrer et suffisamment souples pour s'adapter à chaque organisation [Sandhu 1996 ; Ferraiolo *et al*. 2001]. En effet, la définition des rôles peut refléter la structure de l'organisation. Les rôles peuvent être structurés de façon hiérarchique, pour simplifier encore l'affectation des permissions. Avec RBAC, il est facile d'ajouter un utilisateur : il suffit de lui assigner les rôles qu'il doit jouer dans l'organisation. De même, il est relativement facile de faire évoluer les tâches suite à la création ou la modification d'un objet : il suffit de mettre à jour les privilèges des rôles concernés.

Le principal inconvénient de RBAC réside dans la difficulté de gérer et d'implémenter des règles du type « *seuls les médecins traitants peuvent lire les informations médicales du dossier d'un patient* ». Pour résoudre ce problème avec RBAC, il faut soit créer autant de rôles « médecin traitant du patient X » que de patients, soit mettre en œuvre des règles supplémentaires dans l'application (par exemple, gestion de la base de données des dossiers médicaux), règles qui ne sont pas exprimables dans le modèle RBAC.

2.7. Politiques et modèles de sécurité par équipes (*TMAC*)

2.7.1. Définition de TMAC

Le modèle *TMAC* (pour *TeaM-based Access Control* en anglais) a été formulé pour la première fois en 1997 par Thomas [Thomas 1997 ; Wang 1999]. Le but était alors de fournir un contrôle d'accès pour les systèmes ayant des activités de collaboration. À cet égard, l'entité de base, l'*équipe*, est une abstraction qui encapsule un ensemble d'utilisateurs, qui ont des rôles différents et qui collaborent dans le but d'accomplir une tâche ou d'atteindre un objectif commun. D'une part, les utilisateurs appartenant à une équipe donnée devront avoir accès à l'ensemble des ressources utilisées par l'équipe et d'autre part, les permissions d'un utilisateur devront être dérivées des permissions accordées aux rôles qu'il joue.

Le concept TMAC distingue l'*attribution* et l'*activation* des permissions. Les permissions attribuées sont, par exemple, celles associées à un utilisateur lorsqu'il choisit un rôle au moment de la connexion ; les permissions activées sont celles qui dépendent des variables environnementales (lieu, temps, etc.), elles peuvent donc changer selon le contexte. Le pouvoir d'intégrer les informations contextuelles, lors de la décision d'accès, permet au modèle TMAC d'être flexible et d'exprimer des politiques d'accès pouvant fournir des permissions plus fines que celles de RBAC.

Selon TMAC, le contexte de collaboration d'une équipe donnée doit tenir compte de deux types de contexte :

- *contexte utilisateur* : les utilisateurs qui forment l'équipe à un moment donné ;
- *contexte objet* : les instances d'objets que l'équipe utilise pour accomplir sa tâche.

Dans TMAC [Thomas 1997], une équipe t est définie par :

- TU, ses membres ;
- TR, l'ensemble des rôles que les membres de l'équipe peuvent jouer ($TR \subseteq R$), où R désigne l'ensemble des rôles présents dans le système d'information ;
- $h \subseteq TR$, un rôle particulier nommé "responsable de l'équipe" ;
- OT, un ensemble d'objets associés à l'équipe ;
- TP, un ensemble de permissions associées à l'équipe ($TP \subseteq TR \times OT$).
- UC, un contexte-utilisateur ($UC \subseteq TR \times TU$) ;
- OC, un contexte-objet ($OC \subseteq OT \times O$), où O désigne l'ensemble des objets.

La figure 2.5 donne une description graphique des concepts de TMAC tels qu'ils ont été définis dans [Thomas 1997].

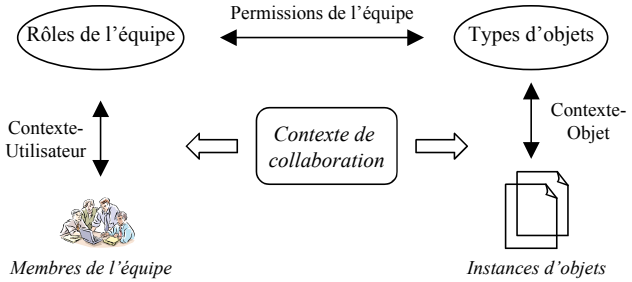


Figure 2.5 : Illustration des concepts de TMAC.

Le travail décrit dans [Thomas 1997] reste une introduction innovante de la notion d'équipe dans la formulation des politiques de sécurité. En effet, dans *TMAC*, l'appartenance d'un utilisateur à une équipe donnée lui confère le droit d'accéder aux ressources associées à cette équipe. Ainsi, les permissions d'un utilisateur dépendent non seulement, du ou des rôles qu'il joue à un moment donné, mais aussi de l'activité courante de l'équipe à laquelle il appartient. Dans le domaine médical, par exemple, un médecin n'a le droit de prescrire des médicaments que pour les patients qu'il traite (le simple fait d'être médecin ne lui donne pas le droit de prescrire des médicaments pour tous les patients). TMAC peut exprimer ce besoin dans la mesure où elle voit un médecin comme un membre d'une ou plusieurs équipes. Néanmoins, Thomas [Thomas 1997] n'a pas spécifié la façon selon laquelle les règles de sécurité seront représentées, ni comment les permissions seront dérivées.

2.7.2. C-TMAC : Context-TMAC

C-TMAC (pour *Context-TeaM Based Access Control* en anglais) [Georgiadis *et al*. 2001] est une amélioration de TMAC qui fournit, de manière explicite, les règles d'activation des permissions selon le contexte. C-TMAC utilise un mélange de RBAC et TMAC, et consiste en cinq entités : utilisateurs, rôles, permissions, équipes et contextes.

Durant une session, les permissions d'un utilisateur représentent l'union des permissions de tous les rôles qu'il a activé. Par ailleurs, dans le contexte sont incluses des informations concernant l'activité de son équipe, ainsi que des informations contextuelles comme le temps.

Très proche de la spécification de RBAC, le modèle C-TMAC définit les éléments suivants :

- U, R, P, S, T, C désignent respectivement l'ensemble des utilisateurs, rôles, permissions, sessions, équipes et contextes ;
- $PRS \subseteq P \times R$ est une relation plusieurs à plusieurs associant les permissions aux rôles ;
- $URS \subseteq U \times R$ est une relation plusieurs à plusieurs associant les utilisateurs aux rôles ;
- $CTS \subseteq C \times T$ est une relation plusieurs à plusieurs associant les contextes aux équipes ;
- $UTS \subseteq U \times T$ est une relation plusieurs à plusieurs associant les utilisateurs aux équipes ;
- *Session-User* : $S \rightarrow U$, une fonction associant chaque session s_i à un utilisateur $User(s_i)$;
- *Session-Role* : $S \rightarrow 2^R$, une fonction associant chaque session s_i à un ensemble de rôles $Roles(s_i)$, avec : $Rôles(s_i) \subseteq \{r \mid (User(s_i), r) \in URS\}$;
- *Session-Team* : $S \rightarrow 2^T$, une fonction associant chaque session s_i à un ensemble d'équipes $Team(s_i)$ avec : $Team(s_i) \subseteq \{t \mid (User(s_i), t) \in UTS\}$;

- *Team-User* : $T \rightarrow 2^U$, une fonction associant chaque équipe t_i à un ensemble d'utilisateurs $User(t_i)$, avec : $User(t_i) \subseteq \{u \mid (u, t_i) \in UTS\} \wedge \exists s_j : user(s_j)=u\}$;
- *Team-Role* : $T \rightarrow 2^R$, une fonction associant chaque équipe t_i à un ensemble de rôles $Role(t_i) \subseteq \{r \mid (user(t_i), r) \in URS\}$; les permissions de l'équipe t_i sont notées *permission-rôle-équipe*. Elles sont déduites à partir de la formule « $\oplus r \in roles(t_i) = \{p \mid (p,r) \in PRS\}$ ». Celle-ci désigne la combinaison (représentée dans la formule par l'opérateur $\oplus$) des permissions de tous les rôles des utilisateurs appartenant à l'équipe.

L'ensemble *permission-rôle-équipe* dépend de la notion de combinaison. Selon C-TMAC, une combinaison correspond à l'un des trois cas suivants :

- si la combinaison est une agrégation, *permission-rôle-équipe* est l'union des permissions des rôles des utilisateurs appartenant à l'équipe ;
- si la combinaison est un maximum (resp. minimum) : *permission-rôle-équipe* est égal à l'ensemble maximal (resp. minimal) des permissions des membres de l'équipe ;
- dans d'autres cas non spécifiés dans [Georgiadis *et al*. 2001], la combinaison peut dépendre de la structure courante de l'équipe.

Selon C-TMAC, la dérivation des permissions (et donc l'accès aux ressources) suit la procédure suivante :

Après son identification et son authentification, l'utilisateur sélectionne un sous-ensemble de rôles et d'équipes auxquels il a droit. L'ensemble des permissions de l'utilisateur (*permission-rôle-session*) est combiné avec l'ensemble des permissions de l'équipe (*permission-rôle-équipe*) comme indiqué dans les deux étapes ci-dessous :

Étape 1 : considérons un utilisateur ayant activé un sous-ensemble de rôles et participant à un sous-ensemble d'équipes. Initialement, les permissions de ses rôles sont dérivées à partir des formules suivantes :

permission-rôle = permission-rôle-session $\oplus$ permission-rôle-équipe

Étape 2 : les permissions finales sont dérivées à partir du contexte de l'équipe et des permissions des rôles. C-TMAC utilise ainsi l'opérateur de filtrage "$\otimes$" pour restreindre les permissions acquises à travers les rôles joués :

permission-contexte = permission-rôle $\otimes$ contexte-équipe

Afin de rendre le contrôle d'accès dynamique, cette deuxième règle déduit l'ensemble final des permissions d'un utilisateur, en utilisant le contexte courant de son équipe comme "filtre". Cette manière de faire permet d'extraire des sous-ensembles de *permission-rôle* en s'appuyant sur les valeurs des variables contextuelles de l'équipe : le lieu, le temps, le patient traité, etc.

Étudions le fonctionnement de C-TMAC dans un contexte médical, environnement où plusieurs équipes peuvent être impliquées dans des processus dynamiques composés de plusieurs tâches. Soit l'exemple d'un patient traité dans le service de médecine générale. Suite à une attaque cardiaque, il est transféré d'urgence à l'unité de soins cardiologiques. La tâche de l'équipe de cardiologie (traiter les patients cardiaques) est exécutée durant un intervalle de temps donné et dans un endroit spécifique (l'unité de cardiologie). Les variables contextuelles sont ainsi, le temps, le lieu et le patient. La procédure est décrite dans la figure ci-dessous.

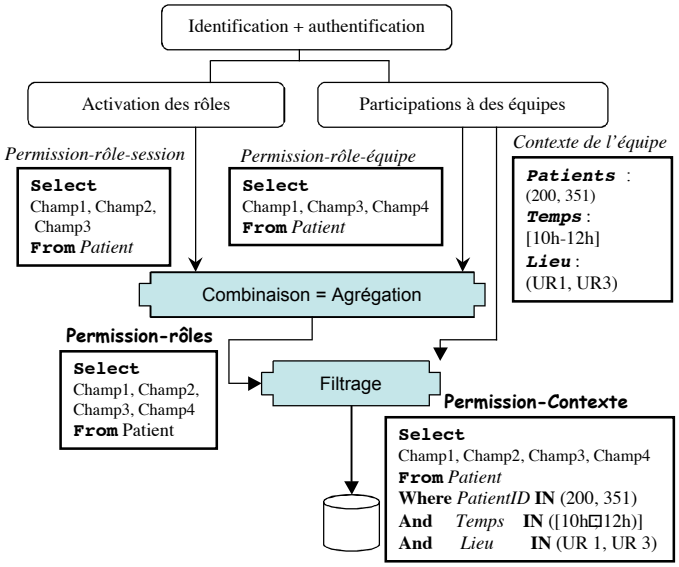


Figure 2.6 : Activation des permissions selon C-TMAC.

Dans cet exemple, on suppose que :

- la base de données contient la table *Patient*(patientID, Champ1, Champ2, Champ3, Champ4) ;
- les paramètres du *contexte*, PatientID, temps et lieu ;
- les *permissions* associées aux rôles médecin, cadre-infirmière et infirmière sont :

  Permissions(*médecin*) : SELECT Champ1, Champ2, Champ3 FROM *Patient*,

  Privilèges(*cadre-infirmière*) : SELECT Champ1, Champ3, Champ4 FROM *Patient*,

  Privilèges(*infirmière*) : SELECT Champ1, Champ4 FROM *Patient*.

Dans l'exemple, on suppose qu'il existe des équipes assurant le service des urgences : *Eq-Ur* localisées dans la salle (*UR*1) ou dans la salle (*UR*3), ainsi qu'une équipe de soins primaires (*GN*2). Le contexte associé à *Eq-Ur* est :

PatientID IN : (200, 351) AND temps IN [10h-12h] AND lieu IN (UR1, UR3).

Supposons maintenant que *Mary* et *Helen* ont commencé leur session s_1 et s_2 et qu'elles activent leurs rôles "cadre-infirmière" et "infirmière" respectivement au sein de *Eq-Ur*. On obtient alors,

Team-User(*Eq-Ur*) = [Mary, Helen]

Team-Role(*Eq-Ur*) = [cadre-infirmière, infirmière].

Les permissions de *Eq-Ur* sont déduites à partir de l'union des permissions des rôles des utilisateurs qui y participent, c'est-à-dire :

Permission-rôle-équipe(*Eq-Ur*) = SELECT Champ1, Champ3, Champ4 FROM *Patient*.

Supposons par ailleurs, que *Chris* commence une session s_3 et qu'il active le rôle médecin au sein de l'équipe *Eq-Ur*. On a donc :

Team-User(*Eq-Ur*) = [Chris, Mary, Helen] ;

Team-Role(*Eq-Ur*) = [médecin, cadre-infirmière, infirmière], et

Permission-Rôle(Chris)=*Permission-rôle-session*(médecin)⊕*Permission-rôles-équipe*(Eq-Ur) = {SELECT Champ1, Champ2, Champ3 FROM *patient*} ∪ {SELECT Champ1, Champ3, Champ4 FROM *patient*}.

En appliquant la règle de filtrage, les permissions finales sont :

Permission-Contexte(Chris) = *Permission-rôle*(Chris) ⊗ *contexte-équipe*(Eq-Ur)

= SELECT Champ1, Champ2, Champ3, Champ4 FROM *patients* ;

Where *PatientID* IN (200, 351) AND *temps* IN [10h-12h] AND *lieu* IN (*UR*1, *UR*3).

À travers cet exemple très simple, on peut constater que C-TMAC présente certaines faiblesses. En effet, si la combinaison est une agrégation, un utilisateur qui rejoint une équipe renforce les permissions de cette équipe en lui ajoutant les siennes : (*Permissions-équipe-après affectation = Permissions-équipe-avant affectation* ∪ *Permissions*(*nouvel utilisateur*)). Ainsi, tous les membres de l'équipe auront les mêmes permissions. Néanmoins, dans le secteur médical, même si les professionnels de santé appartiennent à la même équipe du même hôpital, ils n'ont pas les mêmes droits d'accès aux mêmes parties des fichiers. Il est évident que les permissions finales du médecin doivent être différentes de celles de l'infirmière, même si tous les deux appartiennent à la même équipe de soins. Or, si on reprend l'exemple de la figure 2.6, en considérant, cette fois-ci, que le médecin se connecte avant l'infirmière, on constatera qu'en rejoignant l'équipe, l'infirmière a les mêmes permissions que le médecin.

Le cas où la combinaison est le *maximum* ou le *minimum* est, lui aussi, non réaliste car la déduction des permissions (application des deux règles citées précédemment) se fait de la même manière pour tous les membres d'une équipe donnée. On aura donc le même ensemble de privilèges pour tous ses membres. De plus, dans le secteur de la santé, que veut dire un ensemble minimal ou maximal de permissions ?

Par ailleurs, C-TMAC ne spécifie pas explicitement comment dériver les permissions dans le cas où la combinaison dépendrait de la structure de l'équipe.

2.8. Application de ces approches aux SICSS

2.8.1. Discussion des politiques et modèles existants

L'état de l'art actuel des politiques et modèles de sécurité semble insuffisant pour résoudre les spécificités du contrôle d'accès dans les SICSS. En effet, le contrôle d'accès discrétionnaire présente de graves inconvénients concernant notamment des fuites d'information, au contraire, les politiques obligatoires sont très rigides et très spécifiques aux domaines pour lesquels elles ont été développées. De plus, elles sont trop centralisées et mal adaptées à une analyse et une prise de décision en réseau ou au travers de systèmes réellement répartis. Par ailleurs, ces politiques ne permettent pas de prendre facilement en compte à la fois les exigences de confidentialité, d'intégrité, et *a fortiori* de disponibilité, exigences incontournables des SICSS.

En outre, la nature distribuée et inter-organisationnelle, la diversité des flux et la variété des objets et des sujets des SICSS augmentent la complexité de ces systèmes. Les politiques de contrôle d'interface ainsi que les politiques de contrôle de flux sont, par conséquent, difficiles à implémenter dans le contexte des SICSS. À l'inverse, le concept de rôle fournit une classification des sujets et offre ainsi un début de réflexion prometteur. En effet, l'étude de la fonction de sécurité ne peut être implémentée et mise en œuvre efficacement que si toutes les entités de notre spécification (et non seulement les sujets) sont structurées convenablement.

Le modèle *RBAC*, seul, semble insuffisant pour supporter toute la richesse des SICSS. Tout d'abord, le concept de permission est primitif. En effet, dans le modèle RBAC, rien n'est précisé sur l'usage ou la structure des permissions, considérant qu'elles dépendent de l'application concrète du modèle. Il serait préférable d'ajouter au modèle une structure

générique de permission. Le concept de hiérarchie de rôles est quelque peu ambigu. Il est en général incorrect de considérer que la hiérarchie de rôles correspond à la hiérarchie organisationnelle. Par exemple, le directeur d'un hôpital a un rôle administratif supérieur au rôle de médecin. Pour autant, un directeur d'hôpital n'est pas nécessairement un médecin. Ainsi, il serait incorrect de considérer que le directeur de l'hôpital hérite des permissions du rôle de médecin, comme celle de prescrire par exemple.

De plus, il n'est pas possible dans le modèle *RBAC* d'exprimer des permissions qui dépendent du contexte. Rappelons que si une certaine permission est accordée à un rôle, alors tous les utilisateurs qui jouent ce rôle héritent de cette permission. Par conséquent, il n'existe aucun moyen simple pour spécifier qu'un médecin n'a la permission d'accéder au dossier médical d'un patient que si ce dernier est son patient [Cheng 1999] [Barkley *et al*. 1999].

Enfin, C-TMAC présente certaines limites notamment dans la manière de dériver les permissions. Par ailleurs, dans la quasi-totalité des politiques et modèles étudiés dans ce chapitre, il n'est possible de définir que des *permissions*, alors que pour les SICSS, il faudra exprimer des *interdictions*, des *obligations* et parfois des *recommandations*.

Cette analyse critique constitue le fondement de nos recherches de politiques et modèles mieux adaptés aux SICSS. Le quatrième chapitre clarifie les points évoqués dans cette section et présente le modèle Or-BAC, l'alternative que nous proposons pour spécifier les politiques des SICSS, mais aussi des domaines complexes, interopérables, hétérogènes et fortement distribués [Abou El Kalam *et al*. 2003].

2.8.2. Politiques de sécurité pour les SICSS

Les caractéristiques des SICSS précédemment définies font apparaître un réel besoin d'étude de la sécurité de tels systèmes. À travers plusieurs projets, des associations médicales, des organismes de standardisation ainsi que des équipes de recherche ont montré l'intérêt des politiques de sécurité pour les SICSS. Ils ont mené des investigations et ont proposé des politiques et mécanismes de sécurité pour les systèmes d'information médicales. Néanmoins, les politiques développées sont très générales, très disparates et n'ont généralement pas abouti à un cadre finalisé, pratique, transportable (transférable) et couvrant toute la richesse des SICSS.

Un premier pas a été franchi par l'élaboration de la directive européenne 95/46/EC [Directive 1995] concernant la protection des données personnelles et la libre circulation de ces données. Or, cette directive ne fournit qu'un cadre (légal) harmonisant les législations des pays de la communauté européenne. Combler le fossé qui existe entre la politique générale décrite par la législation et les politiques concrètes, modélisables et applicables dans les organisations reste un grand défi à relever. Intéressons-nous à quelques travaux récents relatifs aux politiques de sécurité pour le domaine médical.

2.8.2.1 Politique de sécurité de SEISMED

SEISMED (*Secure Environment for Information Systems in MEDicine*) est un projet européen s'inscrivant dans le programme AIM (*Advanced Informatics in Medicine*). Il s'est intéressé au développement de directives visant l'amélioration de la sécurité dans les systèmes d'information hospitaliers européens. Le consortium SEISMED a publié une politique de sécurité générique qui repose sur neuf principes [Katsikas 1996]. Chaque principe est détaillé par un ensemble de directives. Pour donner une idée de la forme de ces principes et directives, détaillons le premier : « *un code de bonne conduite concernant la protection des données des patients doit être adopté par chaque établissement de santé* ». Ce principe est décrit par deux directives. La première précise que « *chaque établissement de santé doit publier ce code et exposer les grandes lignes des réglementations concernant la manipulation des données*

personnelles. Le code doit être compatible avec le code de déontologie européen, doit convenir aux caractéristiques de chaque établissement comme il doit réduire les divergences entre la théorie et la pratique ». La deuxième directive précise que « *le code doit reconnaître les droits individuels, doit être conforme aux conventions internationales des droits de l'homme et à la législation nationale au même titre qu'il doit respecter les directives européennes* ».

Notre étude de la politique de sécurité de SEISMED nous a permis de constater que SEISMED manque d'un cadre conceptuel clair et structuré. De plus, cette étude devrait être complétée par un ensemble de mesures spécifiques à chaque environnement.

2.8.2.2 Politique de sécurité de la BMA

LA BMA (*British Medical Association*) a publié en 1996 un livre intitulé "*La sécurité dans les systèmes d'informations cliniques*" [BMA 1996]. Ce document expose une politique de sécurité qui doit être suivie par tous les membres de l'association. Elle fait référence aux déclarations parues dans la brochure « *Good Medical Practice* » du *UK General Medical Council*. S'adressant aux personnels soignants, cette brochure précise que « *les patients ont le droit d'exiger que vous ne transmettiez aucune information que vous déduisez dans le cadre de votre profession, sauf s'ils expriment leur consentement* ». Le document publié par le BMA [BMA 1996] établit un certain nombre de règles assurant le principe de consentement du malade. Une version plus technique a été publiée dans [Anderson 1996a, Anderson 1996b]. Suite aux divergences entre le *National Health Service* et le *British Medical Association*, une conférence a été organisée en juin 1996 à Cambridge et a permis de confirmer neuf principes.

P1 : contrôle d'accès. Chaque dossier médical devrait être marqué avec une liste de contrôle d'accès désignant les personnes ou groupes de personnes qui peuvent y accéder.

P2 : ouverture d'un dossier médical. Un médecin peut ouvrir un dossier avec son nom et celui du patient dans la liste de contrôle d'accès. Lorsqu'un patient a été transféré, il peut ajouter à cette liste le (ou les) médecin(s) en charge du patient.

P3 : contrôle. L'un des médecins de la liste de contrôle d'accès doit être désigné comme responsable, lui seul peut modifier la liste de contrôle d'accès en y ajoutant des personnels soignants, et uniquement des personnels soignants.

P4 : consentement et notification. Le médecin responsable doit informer le patient des noms figurant sur la liste de contrôle d'accès de son dossier. Il est également tenu de l'informer de toute modification ou transfert de responsabilité. Le consentement explicite du malade doit être obtenu, sauf dans les cas d'urgence et dans les cas d'exemption imposés par la loi.

P5 : persistance. Personne ne doit pouvoir effacer un dossier médical avant sa date d'expiration.

P6 audit. Tous les accès aux dossiers médicaux doivent être consignés avec le nom du sujet, la date et l'heure. Une trace de toutes les modifications doit également être conservée.

P7 : flux d'informations. Une information issue d'un dossier *A* peut être ajoutée à un dossier *B* si et seulement si la liste de contrôle d'accès de *B* contient celle de *A*.

P8 : contrôle de l'agrégation. Des mesures efficaces empêchant l'agrégation des informations personnelles de santé doivent être mises en place.

P9 : sous-système de sécurité ou base de confiance (*TCB*). Les systèmes informatiques qui hébergent les informations médicales doivent avoir un sous-système de sécurité qui applique ces principes d'une manière efficace.

Certes, la politique de la BMA a mis en avant des points importants de la sécurité des systèmes médicaux. Néanmoins, elle est loin d'être complète, définissant les objectifs et les règles de sécurité, et couvrant toutes les spécificités des SICSS.

2.8.2.3 Politique de sécurité de la SMA

La SMA (*Swedish Medical Association*) a publié dans la brochure « *Information Technology: The Physician and The Patient* » [SMA 1995] des principes éthiques ainsi qu'un ensemble générique de règles traitant du problème des accès aux fichiers des patients. Parmi ces principes, on retiendra :

- le principe d'*autodétermination* : le patient a le droit de décider pour lui-même ;
- le principe de *non-dommage* : toute action causant des dommages doit être empêchée ;
- le principe d'*intérêt* : toute action prise doit apporter un gain supplémentaire ;
- le principe de *solidarité* : les ressources ne doivent être utilisées que si le besoin le justifie ;
- le principe d'*efficacité* : les ressources doivent être utilisées de façon optimale et efficace.

Les principes de la SMA devraient s'appliquer aux accès aux dossiers médicaux,à la sécurité du réseau et des transmissions des données des patients, à la gestion de l'organisation et à l'amélioration de la qualité, etc.

Néanmoins, même si le document de la SMA soulève des problèmes éthiques assez importants, la notion de politique de sécurité n'apparaît pas de manière claire et explicite. Aucune règle du contrôle d'accès ni mesure à prendre pour la protection de données médicales n'est annoncée. Tout ce que la SMA fournit, c'est sa position, sous forme de déclarations concernant les problèmes et défis de sécurité dans ce domaine.

2.8.2.4 Recommandations de la FMAG

L'association médicale allemande (*Bundesärztekammer*) a publié des recommandations de sécurité intitulées *« recommandations concernant les obligations de confidentialité médicale, protection et traitement des données dans le contexte des pratiques médicales* » [FMAG 1996]. Ce document fournit des conseils aux professionnels de santé, leur montre comment implémenter les lois concernant la protection des données privées en Allemagne et définit les obligations vis-à-vis du maintien et du traitement des données médicales. La politique de sécurité proposée repose sur un ensemble de principes destinés aux praticiens, et qui touchent au droit et à l'intimité des patients, à la confidentialité des données médicales, à l'enregistrement et la documentation des traitements et des résultats, au droit du patient à être informé des résultats et des tests, etc.

Ces recommandations ne doivent pas être considérées comme une politique de sécurité à part entière. Leur but est de spécifier les obligations ainsi que les responsabilités des professionnels de santé en respectant l'utilisation des technologies de l'information.

2.8.2.5 Conclusion et présentation du projet MP6

Outre les travaux énoncés ci-dessus, plusieurs projets européens (ISHTAR, TrustHealth, DIABCARD, MedSec, Harp, EUROMED-ETS, HANSA, PRIDEH et DRIVE) se sont penchés sur les problèmes de sécurité des SICSS. Certaines études ont porté en particulier sur les cartes à puces, les mécanismes biométriques et les infrastructures de gestion de clés.

En matière de politiques et modèles de sécurité, l'analyse présentée montre que les résultats ne sont pas encore complètement satisfaisants. Des mécanismes classiques (authentification [Blobel & Pharow 2001], autorisation, contrôle d'accès, chiffrement) existent tant au niveau des réseaux pour fournir la sécurité nécessaire des communications, qu'au niveau des systèmes informatiques pour apporter la sécurité des systèmes manipulant des informations sensibles. Pour mettre en œuvre efficacement ces mécanismes, il est indispensable de définir précisément

les besoins ainsi que les règles de sécurité à appliquer, ceci implique la définition et la spécification d'une politique de sécurité appropriée aux SICSS.

Le travail jusque là effectué s'attache à prendre en compte les concepts sectoriels particuliers, tels que l'auditabilité juridico-technique [Trouessin, 2002] issue des responsabilités juridiques spécifiques (légales, éthiques, déontologiques) pour les politiques de sécurité à respecter ainsi que des modèles de sécurité adaptés et adaptables aux SICSS. Notre objectif est d'améliorer la pertinence et l'efficacité des systèmes des secteurs ciblés ainsi que la confiance à accorder à de tels systèmes. Plusieurs problématiques de recherche seront étudiées, telles les politiques de sécurité et modèles de politiques de sécurité, politiques d'autorisation et d'anonymisation, et ce dans les rouages des deux secteurs : médical et social.

Nos travaux sont partiellement soutenus par le Réseau National de Recherche en Télécommunication, dans le cadre du projet MP6 (Modèles et Politiques de Sécurité pour les Systèmes d'Information et de Communication en Santé et Social) dont les partenaires sont : Ernst & Young Audit (coordinateur), ENST-Bretagne, ETIAM, France Telecom R&D, LAAS-CNRS, MasterSecuritY, ONERA-DTIM, Supélec-Rennes, et UPS-IRIT.

Le projet MP6 est organisé en 9 sous projets :

- SP1 : Administration, gestion et coordination du projet ;
- SP2 : État des lieux sectoriel/conceptuel/terminologique en sécurité pour les SICSS ;
- SP3 : Politiques de sécurité pour les SICSS ;
- SP4 : Modélisations formelles de politiques de sécurité des SICSS ;
- SP5 : Explorations-A "politiques d'autorisation et gestion des droits" ;
- SP6 : Explorations-B "politiques d'anonymisation et gestion des pseudonymes" ;
- SP7 : Explorations-C "détection de risques d'intrusions, déviances et inférences" ;
- SP8 : Intégrations sectorielles pour la sécurité des SICSS ;
- SP9 : Normes et standards, promotion et valorisation.

Nous contribuons aux sous projets 2, 3, 4, 5, 6, 7 et 9.

Chapitre 3. Bâtir une politique de sécurité pour les SICSS

Préambule

L'ensemble des concepts, de la sûreté de fonctionnement en général et de la sécurité en particulier, nécessaires à la compréhension de ce mémoire ont été définis. Un état des lieux des types et des complexités des modèles et politiques de sécurité qui existent, notamment ceux mis en œuvre réellement dans les SICSS a été présenté. On a donc constaté que ces politiques de sécurité ne couvrent pas toute la richesse des SICSS. Il convient ainsi de développer d'autres politiques plus adaptées.

Il semble évident que la conception et la réalisation de systèmes sûrs nécessite, tout d'abord, le développement de méthodes pour définir précisément des politiques de sécurité. Ce chapitre commence ainsi par proposer une méthodologie de travail dont les principales étapes sont les suivantes : description du système, identification des informations à protéger et la caractérisation des menaces, et définition de la politique de sécurité en exprimant les besoins de sécurité ainsi que les règles qui déterminent comment l'information sensible et les autres ressources sont gérées, protégées, et distribuées dans le système.

Il s'agit également d'appliquer cette démarche pour dériver une politique de sécurité pour un exemple de système d'information médicale, et une autre pour un exemple du domaine social.

Une troisième étude de cas consiste à définir la base terminologique nécessaire à l'étude du problème de l'*anonymisation des données médicales*, où il est question des différents travaux existant, d'une étude des solutions implémentées dans les pays européens, et d'un ensemble de scénarios servant un objectif global de respect de la vie privée. Une démarche progressive d'analyse des besoins d'anonymisation est proposée. Celle-ci commence par poser des questions pertinentes afin de définir les attentes, passe par l'identification des risques encourus, des objectifs et des exigences d'anonymisation, et aboutit à des solutions-types adaptées aux besoins. Par la suite, il s'agira d'une procédure d'anonymisation en cascade à différents niveaux (hôpitaux, centres de traitements et utilisateurs finaux) et de montrer comment cette procédure innovante permet de dégager plusieurs avantages, notamment : la résistance aux attaques par dictionnaire, les anonymisations sectorielles, la possibilité de fusions flexibles et sécurisées des données de plusieurs établissements, etc.

Pour son efficacité, l'anonymisation pose des problèmes particuliers (pour les SICSS) qui méritent d'être étudiés séparément. Néanmoins, une fois anonymisées, les données sont considérées comme objets sensibles auxquels s'applique la politique de sécurité.

Ainsi ce chapitre a une double vocation,

- d'ordre sectoriel, puisque la méthodologie de construction de notre politique de sécurité est illustrée par des exemples issus des secteurs santé et social ;
- d'ordre plus général, car notre démarche et nos solutions ne sont pas restreintes aux seuls secteurs de la santé et du social, mais peuvent être appliquées à différents types de systèmes.

3.1. Méthodologie de notre approche

L'objectif de la sécurité des systèmes d'information et de communication est de garantir qu'aucun préjudice ne puisse violer les objectifs de sécurité. Il s'agit alors de diminuer la probabilité de voir les menaces se concrétiser, à en limiter les conséquences éventuelles, et en cas de problème, à revenir à un fonctionnement normal à des coûts et dans des délais acceptables.

Dans la construction de nos politiques de sécurité, la démarche que nous allons suivre consiste à répondre à un certain nombre de questions : quel système veut-on sécuriser et quelles lois et réglementations régissent ces systèmes ? Que veut-on protéger ? Quelles sont les vulnérabilités de ce système ? Quelles menaces doit-on affronter ? Quels risques encourt-on ? Quels sont les besoins de sécurité ? Quelles sont les règles de sécurité qui permettent de satisfaire ces besoins, de faire face aux menaces, et d'assurer la protection du système ?

3.1.1. Description d'un scénario représentatif

Certaines politiques de sécurité présentées précédemment, comme les politiques de contrôle d'interface (voir 2.6), font le choix de définir les propriétés de sécurité attendues indépendamment de la description du système lui-même. Cette manière de faire considère que la description du fonctionnement interne du système n'est pas nécessaire. Néanmoins, la mise en œuvre d'une politique de ce type conduit généralement à imposer des contraintes fortes sur le fonctionnement de l'organisation, ce qui soulève des difficultés, notamment au niveau du rendement ou de la qualité de service. À l'inverse, notre construction de la politique de sécurité des SICSS intègre une description partielle du système considéré. En effet, il semble que, compte tenu de la complexité des organisations, de la diversité des flux ainsi que des spécificités et des réglementations socio-sanitaires des SICSS, il est nécessaire d'établir un scénario représentatif et de préciser les lois en vigueur, avant d'identifier clairement les informations à protéger, les menaces, les objectifs de sécurité, ainsi que les règles de sécurité.

Le scénario décrit ne sera pertinent vis-à-vis de la sécurité que s'il permet d'exprimer :

- les éléments de base de la politique de sécurité, notamment la classification des utilisateurs (rôles, groupes), les types d'accès possibles, les ressources, etc. ;
- les règles de description du fonctionnement du système, afin de pouvoir en déterminer l'impact sur les objectifs de sécurité ; ll s'agit de spécifier les flux d'informations, les processus, la hiérarchie de rôles ainsi que les contraintes organisationnelles, etc.

3.1.2. Identification des informations à protéger

La deuxième étape consiste à identifier, de manière claire et non ambiguë, les informations et les autres ressources sensibles, leur niveau de criticité en fonction de menaces particulières, ainsi que le risque de perte totale ou partielle de ces ressources.

Les types d'informations à protéger sont caractérisés en fonction de leur contenu informationnel, c'est-à-dire de leur sémantique. On s'intéresse en particulier aux liens logiques susceptibles d'exister entre informations, de manière à anticiper des problèmes d'inférence de données sensibles à partir de données non-protégées.

Ensuite nous étudions contre qui ou contre quoi ces informations doivent être protégées ? C'est-à-dire identifier les menaces auxquelles sont potentiellement soumis de tels systèmes d'information. Il y aura lieu de tenir compte des différents modes de fonctionnement (aspects organisationnels) et des diverses réglementations en vigueur dans les domaines de la santé ou du social, ainsi que dans les domaines adjacents concernés par de tels systèmes d'information.

3.1.3. Expression des objectifs de sécurité

Il s'agit de décrire les objectifs de sécurité, pour faire face aux risques identifiés précédemment et satisfaire ainsi les exigences de sécurité. Ces objectifs s'exprimeront notamment en terme de propriétés de sécurité (confidentialité, intégrité et disponibilité), attendues pour ce système. Ces objectifs seront définis au niveau du système d'information, en tenant compte des caractéristiques multiples de son environnement.

Les objectifs de sécurité doivent déterminer précisément ce qui est permis, interdit, obligatoire et recommandé dans le système. Ceci peut conduire à définir qu'une certaine catégorie d'information doit être inaccessible pour une certaine catégorie d'utilisateurs : par exemple, interdire au pharmacien de créer des ordonnances, interdire aux professionnels de santé de transmettre des dossiers médicaux vers des pays qui ne disposent pas d'une législation de protection des données [Directive 1995], autoriser les agents des services payeurs à accéder aux données financières, etc.

Un état du système qui satisfait l'ensemble des objectifs de sécurité est évidemment un état sûr, autrement dit, considéré comme acceptable du point de vue de la sécurité.

3.1.4. Définition des règles de sécurité

Afin de compléter la spécification de la politique de sécurité, il est important d'exprimer un ensemble de règles visant à définir comment le système peut évoluer sans compromettre les objectifs de sécurité identifiés. L'expression de ces règles doit prendre en compte les différentes structures organisationnelles dans lesquelles sont déployés les systèmes d'information, ainsi que les contextes d'utilisation de nature propre au domaine de la santé (comme la notion d'urgence) ou du social (comme l'authenticité d'une télédéclaration sociale). Imaginons une règle de sécurité : en cas d'urgence, tout professionnel soignant a le droit d'accéder aux données médicales d'un patient, mais en parallèle, le système doit enregistrer les paramètres de l'accès dans le fichier d'audit.

Les tâches décrites jusqu'ici, devront aboutir à une spécification explicite :

- d'un ensemble de règles de fonctionnement des SICSS ;
- d'une liste d'informations et services à protéger et des menaces qui pèsent sur ces informations ou services ;
- d'objectifs de sécurité exprimant les besoins de confidentialité, d'intégrité et de disponibilité, telles qu'ils sont considérés dans le contexte du système étudié ;
- d'un ensemble de règles de sécurité exprimant qui a accès à quoi et dans quelles conditions.

3.1.5. Modélisation formelle

Après sa spécification, la politique de sécurité doit être formellement décrite notamment pour pouvoir :

- lever les ambiguïtés sur la spécification ;
- interroger la politique de sécurité ;
- manipuler la politique de sécurité par des transformations mathématiques et avec l'assistance d'outils de preuve afin d'en vérifier la cohérence et la complétude ;
- étudier les problèmes d'interopérabilité entre plusieurs politiques ;
- avoir la preuve que la spécification assure les propriétés de sécurité ;
- vérifier que l'implémentation du système d'information, et en particulier des mécanismes de contrôle d'accès, permet bien de garantir les propriétés de sécurité souhaitées.

3.2. De la description des SICSS aux besoins de sécurité à satisfaire

Dans cette section, il s'agit d'appliquer aux SICSS les trois première tâches précédemment décrites[15] afin de fournir une asise solide à la définition des politiques de sécurité adaptées aux SICSS.

3.2.1. Étude de cas 1 : Sphère médicale

3.2.1.1 Scénario

3.2.1.1.1 Description et règles de fonctionnement

Un système d'information médicale peut être défini comme un système dédié à la santé. Il posséde des moyens fiables assurant la communication, le traitement, le stockage ainsi que l'archivage des informations médicales, paramédicales, médico-administratives et médico-financières. Un tel système relie un grand nombre d'utilisateurs ayant des responsabilités et des centres d'intérêts différents. Les flux d'informations entre ces différentes catégories d'utilisateurs peuvent se résumer comme suit :

- Entre professionnels de santé et régime d'assurance maladie : envoi des lots de feuilles de soins électroniques et échange des accusés de réception logiques.
- Entre professionnels de santé : médecins, hôpitaux, pharmacies et laboratoires peuvent s'échanger des informations médicales afin de favoriser l'aide au diagnostic et la recherche épidémiologique.
- Entre professionnels de santé et d'autres réseaux : les services d'étude et de recherche épidémiologique envoient aux médecins des statistiques d'activité et leur fournissent une aide à la prescription.
- Accès simplifié au réseau Internet, afin de bénéficier des services offerts par les sites Web médicaux et de disposer des meilleures informations sur la santé.
- Accès aux informations des ministères et aux informations juridiques et fiscales concernant les professionnels de santé.

Par ailleurs, il est possible de spécifier d'autres règles de fonctionnement, notamment :

- *La hiérarchie des rôles*, par exemple : tout médecin (ou infirmière ou aide soignante) est aussi considéré comme personnel soignant ; un utilisateur est considéré comme professionnel de santé seulement s'il est personnel soignant ou personnel paramédical, ou personnel administratif ou hôtelier dans un centre de soin (cabinet médical, hôpital, clinique), autrement dit, titulaire d'une Carte de Professionnel de Santé (CPS) ; les personnes travaillant dans une pharmacie ou dans un laboratoire font partie du personnel paramédical.
- *L'activation des rôles*, par exemple : tout utilisateur qui se connecte avec sa carte de professionnel de santé joue le rôle dans l'organisation mentionnée sur cette carte.
- *Le contexte des rôles*, par exemple un utilisateur ne peut pas être à la fois comptable et médecin dans le même hôpital.

---

[15] Les trois étapes qui seront effectuées dans cette section sont : description d'un scénario représentatif, identification des informations à protéger, et expression des objectifs de sécurité. Pour éviter toute redendonce, la quatrième étape "définition des règles de sécurité" sera partiellement décrite ici, et sera complétée dans les autres chapitres. La cinqième étape "modélisation formelle" sera présentée dans le chapitre suivant.

3.2.1.1.2 Entités de base

Afin de simplifier et afin d'éviter toute redondance inutile, il ne paraît pas nécessaire de décrire toutes les entités de base des systèmes d'information médicale. Celles-ci seront décrites plus en détail lors de la définition du modèle dans le chapitre suivant et dans la représentation de la politique de sécurité dans ce modèle.

Le système de santé peut être vu comme un ensemble d'organisations qui interagissent : hôpitaux, instituts de recherche, organismes payeurs, etc. (figure 3.1). L'hôpital est une structure associant plusieurs types d'unités : unités de soins, pharmacie, services administratifs, médico-techniques et logistiques ([Degoulet 1992 - Ch 11][Degoulet 1989]). En l'occurrence, l'unité de soins, principal site d'accueil des patients, est définie comme étant une entité physique dont la fonction est de produire des soins médicaux (diagnostic, thérapie, évaluation). Il peut s'agir d'un service, d'un regroupement de services dans le cadre d'un département ou au contraire d'une unité fonctionnelle à l'intérieur d'un service. Une unité de soins suppose la présence d'une équipe soignante placée sous une responsabilité bien déterminée et prenant en charge des patients.

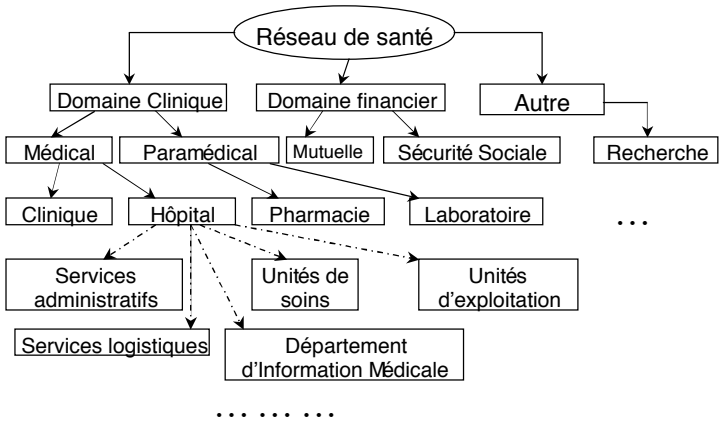


Figure 3.1 : Organisation et domaines d'un réseau de santé.

Le concept de *rôle* est indispensable dans les systèmes d'information médicale. En effet, on trouve les rôles médecin, infirmière, etc.

Les *actions* des SICSS peuvent être *élémentaires* (lire, écrire, etc.) ou *composite*. Par exemple, l'action composite "*prescrire*" correspond à l'exécution des actions élémentaires : lire les données de séjour hospitalier, consulter l'historique des prescriptions, lire le rapport infirmier, lire les résultats d'examens, créer l'ordonnance et y écrire des données.

Plus général que la notion d'action composite, un *processus de soins* fournit le cadre dans lequel les unités de soins interagissent pour traiter les patients. C'est donc une activité, enregistrée dans le serveur, identifiée par un patient, un motif de consultation ou d'hospitalisation, et une ou plusieurs équipes soignantes qui collaborent pour traiter le patient [Clercq 1998] et [Deliège 2001]. Par exemple, un *patient* souffrant de douleurs abdominales (*problème)* peut se présenter chez son médecin traitant qui fait une première évaluation et initialise le processus de soins, avant d'envoyer le patient chez un spécialiste pour compléter et finaliser le diagnostic. Le patient reviendra enfin chez son généraliste pour assurer un suivi du traitement initié par le spécialiste. Le partage des données de santé de ce patient entre ces

différents professionnels de santé s'effectue dans le cadre du processus de soins déclaré par le médecin traitant et constitué des trois plans[16] de soins.

Toute l'activité des professionnels de santé est organisée autour du patient. Son dossier n'apparaît à un acteur de l'unité de soins que sous l'angle des besoins de sa tâche au sein de l'organisation. Chaque acteur ne sera donc concerné que par certaines parties du dossier. Dans la littérature, plusieurs types de dossiers ont été cités : le dossier hospitalier, le dossier de spécialité, le dossier partageable, le dossier biologique, le dossier clinique, le dossier de transmission, le dossier minimum européen, le dossier d'archives, etc. [Degoulet 1989]. Voici les types de dossiers les plus importants :

- Le dossier de spécialité : il est très spécifique à l'unité de soins. Sa constitution tient compte du plan de travail et des contraintes de l'unité à laquelle il appartient. Il existe une grande variabilité dans son contenu et dans la façon dont il est utilisé.
- Le dossier partageable : ce dossier est dynamique (en cours d'élaboration). Il comprend l'histoire médicale actuelle du patient (les problèmes actifs) ainsi que les résultats provisoires et les avis temporaires. Il est le support permettant aux professionnels de santé participant à un processus de soins, de communiquer et d'échanger des informations.
- Le dossier archive : après la fermeture de chaque processus de soins, le dossier archive est alimenté par les résumés des dossiers partageables (seules les données objectives concernant un patient font partie de son dossier et peuvent être conservées dans une base de données médicales nominatives). Chaque résumé comprend, outre l'identification du patient, des informations cliniques de synthèse, les pathologies diagnostiquées, les traitements, la modalité de sortie et la façon dont le suivi de ce patient sera effectué. Ces informations peuvent être structurées dans les résumés : clinique, infirmier, psychiatrique, gériatrique, financier et social. Ainsi, contrairement au dossier partageable qui est dynamique, le dossier archive est stable.

3.2.1.2 Informations à protéger

Plusieurs organisations nationales et internationales s'intéressent de plus en plus à la sécurité des SICSS, problème complexe aux dimensions légales, éthiques, sociales, organisationnelles et techniques. Par exemple, l'Assemblée générale des Nations-Unies a adopté des directives pour la réglementation des fichiers informatisés contenant des données personnelles [Résolution 1990]. Le Conseil de l'Europe a émis des recommandations concernant les banques de données médicales automatisées [Recommandation 1992] et les échanges des données de santé dans les hôpitaux [Recommandation 1997]. La Commission Européenne a développé la directive [Directive 1995] relative à la protection des données personnelles et à la libre circulation de ces données. En France, la récente loi du 4 mars 2002 [Loi 2002] définit les droits des malades et décrit la qualité du système de santé. Le décret [Décret 2002] restreint les accès aux informations personnelles détenues par les professionnels de santé.

Un des points importants qui peut être extrait de cette réglementation concerne la protection de la vie privée, et plus particulièrement les données nominatives. On entend par donnée nominative toute donnée personnelle permettant une identification, *directe* ou *indirecte*, d'un individu [loi 1978]. Voici quelques types de données nominatives :

- générales (situation en matière de vaccinations, histoire médicale, traitements chroniques) ;
- contacts (données relatives à une consultation particulière) ;

[16] Un plan de soin peut être défini comme le résultat d'une consultation (ou d'une hospitalisation). Il contient les commentaires, les traitements à suivre, etc.

- archives (protocoles d'examens, rapports de spécialistes, …) ;
- socio-administratives (données relatives à l'assurance maladie, au régime d'invalidité) ;
- indications personnelles du médecin traitant (hypothèses, réflexions).

Une donnée nominative peut être de nature diverse :

- objective (résultat d'une analyse ou d'un test) ;
- affirmative (consécutive à une interprétation : par exemple, diagnostic) ;
- provisoire (hypothèse de travail non validée mais toutefois consignée dans un document tel le dossier partageable).

Par ailleurs, d'autres données non-nominatives peuvent être présentes dans un SICSS : des valeurs de norme pour les formules sanguines, des règles d'attribution de droits sociaux, etc. Ces données présentent un intérêt moindre pour cette étude du point de vue de la confidentialité. Néanmoins la violation de leur intégrité peut induire en erreurs des traitements des données personnelles, des diagnostics, des calculs de remboursement, etc.

3.2.1.3 Risques identifiés

Les couples (menace, vulnérabilité) permettent d'identifier les risques auxquels peuvent être soumis les SICSS. Les intrusions dans les SICSS revêtent une importance primordiale. En effet, si les propriétés de sécurité sont violées :

- le médecin risque de prendre des décisions portant préjudice aux patients,
- la valeur de l'information comme base de diagnostic est amoindrie,
- dans un cadre médico-légal, un professionnel de santé appelé à justifier ses actions, pourrait se trouver dans l'incapacité d'utiliser les dossiers informatiques comme preuve.

Diverses réglementations se sont intéressées à l'identification des risques auxquels sont confrontés les SICSS et à y proposer des solutions. Les directives européennes [Directive 1995] spécifient que le fournisseur d'un service web doit prendre toutes les mesures techniques et organisationnelles (dont la définition de la politique de sécurité) pour garantir la sécurité de ses services. La résolution [Résolution 1990] met en garde contre les pertes accidentelles d'informations, les accès non autorisés et les utilisations illégitimes. En France, le décret [Décret 2002] identifie des menaces relatives aux accès illégitimes et aux pertes de données. Le code de déontologie médicale [Code 1995a] et le code de santé publique [Code 1995b] sensibilisent les professionnels de santé aux risques liés au secret professionnel.

En se basant sur ces textes, une liste de menaces et de vulnérabilités a été établie. Celle-ci est donnée en annexe A, selon une classification en fonction des propriétés de sécurité auxquelles l'occurrence des menaces peut porter atteinte. Le type, l'origine ainsi que les conséquences de l'intrusion sont également mentionnés. Une liste plus exhaustive incluant des menaces plus génériques est présentée dans [Abou El Kalam *et al*. 2002c]

3.2.1.4 Besoins de sécurité

Les risques identifiés justifient un besoin de confidentialité, d'intégrité et de disponibilité.

3.2.1.4.1 Confidentialité

La confidentialité est à la fois liée au respect du secret professionnel des organismes de santé et à la vie privée des patients :

- Le respect des données personnelles des patients (intimité) : il n'y a pas de traitement médical sans confiance, de confiance sans confidence et de confidence sans secret. Ces secrets ne doivent être confiés qu'aux utilisateurs autorisés. Ainsi, un utilisateur habilité à

comptabiliser les traitements des médecins ou à faire des statistiques ne devrait pas avoir le droit d'accéder aux données médicales nominatives des patients. Les utilisations des données médicales à des fins non épidémiologiques (comme les publications scientifiques) ne devraient pas permettre d'établir le lien entre les données publiées et la personne physique concernée, etc.

- La confidentialité des intérêts professionnels : la confidentialité des informations est d'abord pour les professionnels de santé une obligation personnelle de discrétion envers les organisations auxquelles ils appartiennent (hôpitaux, organismes payeurs, etc.). Le nouveau code de déontologie et les directives européennes précisent que ce secret est absolu, sauf exception clairement définie par la loi.

3.2.1.4.2 Intégrité

L'intégrité peut être mise en cause par des manipulations erronées mais également par la perte de données, accidentelle ou délictueuse. Elle touche également à la validité des données saisies, en particulier, à l'évitement des collisions[17] et des doublons[18] lors de la génération de pseudonymes.

L'obligation d'intégrité est d'abord définie par l'article 29 de la loi du 6 janvier 1978 [Loi 1978] qui enjoint au responsable du fichier de veiller à ce que les informations qui lui sont confiées ne soient ni déformées ni endommagées. Cette obligation d'intégrité reconnaît toutefois un droit de rectification. L'article 36 de la loi « informatique et libertés » affirme que « le titulaire du droit d'accès peut exiger que soient... rectifiées ou effacées les informations le concernant qui sont inexactes... ou dont la collecte... est interdite ». La charte du patient hospitalisé ajoute : « le patient hospitalisé exprime ses observations sur les soins et l'accueil et dispose du droit de demander réparation des préjudices qu'il estimerait avoir subi ». La loi [Loi 2002], relative aux droits des malades, confirme ces droits.

3.2.1.4.3 Disponibilité

L'indisponibilité des données (des patients) et des services est intolérable dans ce type de systèmes où la vie des patients est en jeu. Le système d'information médicale peut être jugé en danger, dès lors qu'une information existante peut être non disponible, suite à une défaillance matérielle ou logicielle, à une suppression intentionnelle ou malveillante ou à une attaque en déni de service. Dans ce domaine, la disponibilité concerne à la fois le caractère d'urgence et la pérennité des données :

- La disponibilité à court terme (*caractère d'urgence*) : les fichiers des patients, les programmes et les applications du système sont des ressources critiques qui peuvent, à un moment donné, être invoquées, par plusieurs utilisateurs, et pour différentes raisons (accomplissement des procédures médicales et administratives, étude de l'efficacité des processus, etc.). Il est évident que ces ressources doivent être disponibles aux utilisateurs autorisés. En cas d'urgence par exemple, le médecin du SAMU doit pouvoir y accéder, en un temps raisonnable et sans rencontrer de problème d'accès ou de rupture du service délivré par le système. De même, dans le cas de la télémédecine, la gestion de l'état de santé du patient doit être immédiate alors qu'il s'agit d'échanger en toute protection des données complexes comme des images, des enregistrements de cardiogrammes, etc.
- La disponibilité à long terme (*pérennité*) : le maintien à long terme des dossiers médicaux est un problème primordial pour les systèmes d'information médicale. La CNIL

[17] Il y a collision lorsqu'à partir de données nominatives différentes, on génère un même pseudonyme (qui risque ainsi d'être alloué à deux personnes différentes).

[18] Il y a doublon lorsque deux pseudonymes différents sont générés pour une même personne.

(Commission Nationale de l'Informatique et des Libertés) comme le Conseil de l'Europe ont toujours affirmé que les informations nominatives ne devaient pas être conservées dans un système informatique plus longtemps que ne le nécessite la finalité déclarée des traitements. Ceci ne soulève pas de grandes difficultés pour les informations destinées à des recherches temporaires. Mais ce n'est pas le cas des dossiers hospitaliers, puisque le règlement des archives hospitalières impose des délais de conservation très longs : soixante-dix ans pour les dossiers de pédiatrie, de neurologie, de stomatologie et de maladies « chroniques », illimités lorsqu'il s'agit de maladies héréditaires. Préserver les fichiers est une tâche qui n'est pas simple. D'une part, au-delà du problème des capacités des supports d'enregistrement, nous n'avons pas besoin d'information identifiée inutile (simple erreur, diagnostic révisé). D'autre part, il ne faut pas faciliter l'écrasement des données, notamment les fautes professionnelles et les traces juridiques. La politique de la *British Medical Association* (BMA) présente deux solutions à ce problème [BMA 1996] :

> la première est la mise à jour des fichiers par ajout plutôt que par effacement. Les versions les plus récentes doivent être présentées en priorité. La destruction doit être réservée aux fichiers dont la durée de vie, déterminée par la loi, est expirée ;
>
> la deuxième solution peut consister à autoriser les destructions à condition qu'un des mécanismes, tels que l'audit, puisse reconstituer les fichiers dans n'importe lequel de ses états antérieurs.

3.2.1.5 Règlement de sécurité

La description du scénario (voir 3.2.1.1) a montré que lorsque le patient se rend chez son médecin de famille, ce dernier crée un processus de soins. Ce processus est matérialisé par un dossier partageable contenant les informations temporaires échangées entre les professionnels de santé en charge du patient. À la clôture du processus de soins, un résumé du dossier partageable met à jour le dossier archive. La figure 3.2 explique que les membres des différentes équipes qui collaborent pour traiter le patient ont accès aux dossiers archives et partageables, tandis que chacune de ces équipes, possède un dossier de spécialité non accessible aux autres.

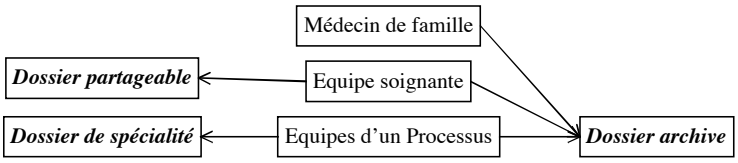


Figure 3.2 : Accès des catégories d'utilisateurs aux différents types de dossiers médicaux.

Par ailleurs, même si les utilisateurs d'un système d'information médicale doivent communiquer, coopérer, échanger les informations, utiliser les applications et consulter les bases de données, ils n'ont pas forcément les mêmes vues. D'une manière générale :

- Les médecins généralistes ont besoin de l'ensemble des données qui leur permettent de gérer les aspects médicaux des cas qu'ils traitent. Ils peuvent visualiser les diagnostics antérieurs ainsi que les rapports infirmiers. Ils ont aussi le droit d'éditer le rapport de consultation.
- Les spécialistes doivent avoir l'accès aux détails relevant de leurs spécialités.
- Le personnel infirmier ne doit pas avoir le droit de prescrire de médicaments, ni le droit d'accéder à certaines données dépassant les limites de leur rôle dans le système. Néanmoins, ils ont le droit de rédiger le rapport infirmier.

- Les pharmaciens doivent pouvoir accéder à la liste des médicaments prescrits, ainsi qu'à l'historique des prescriptions. Ils n'ont le droit de modifier les ordonnances que pour substituer un médicament par le générique correspondant. Tout accès s'inscrivant dans cette exception doit être audité.

Aussi, pouvons-nous déduire que le rôle est certainement l'un des paramètres qui intervient pour décider quelle partie du dossier est accessible par quel utilisateur. À titre d'exemple, le médecin traitant (psychologue, gynécologue) est le seul à posséder une partie privée (commentaire) où il met les données intimes que le patient ne veut pas partager avec une tierce personne. Cette partie n'est donc accessible que par le médecin et son patient. La figure 3.3 montre l'exemple d'une équipe composée d'un médecin, d'une infirmière et d'une secrétaire médicale. Le sens de la flèche indique le sens du flux d'informations : « Médecin → Commentaires » indique un accès autorisé en écriture du médecin à la partie « commentaires » ; « Médecin ← Commentaires » indique un accès autorisé en lecture du médecin à « commentaires ». L'accès au dossier archive n'est autorisé qu'en lecture dans la mesure où l'alimentation du dossier archive se fait automatiquement après la fermeture de chaque processus de soins. Les parties composant le dossier de spécialité diffèrent d'une équipe à une autre.

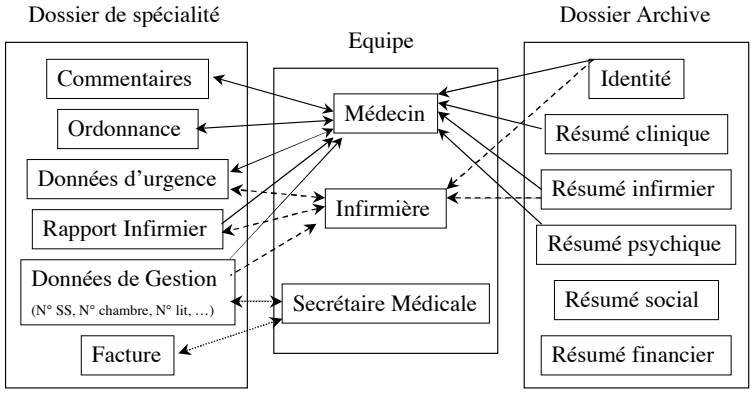


Figure 3.3 : Accès aux parties des dossiers selon le rôle**.**

D'autres règles d'accès peuvent être déduites directement des textes et loi relatifs aux SICSS. Par exemple, le texte [BCN 1999] indique que « *les opérations de soins ne sont permises qu'aux utilisateurs qui sont personnels soignants* ». La loi [Loi 2002] et son décret d'application [Décret 2002] donnent au patient le droit de lire son dossier médical : *« toute personne a accès à l'ensemble des informations concernant sa santé … elle peut accéder à ces informations directement ou indirectement par l'intermédiaire d'un médecin et en obtenir communication … La présence d'une tierce personne lors de la consultation de certaines informations peut être recommandée par le médecin …, pour des motifs tenant au risques que leur connaissance sans accompagnement ferait courir à la personne concernée* ». À partir de cette loi, il est possible de définir des règles de sécurité utilisant une modalité de recommandation, par exemple :

- il est recommandé que le patient accède à son dossier médical à travers son médecin traitant ;
- si le patient est mineur ou souffrant de troubles mentaux ou psychologiques, la présence du tuteur est recommandée.

En effet, le médecin est la personne la mieux placée pour comprendre le codage et pour expliquer le contenu du dossier médical soit à son patient (si son état le permet), soit au tuteur (dans des conditions bien précises).

Ces règles seront complétées et formalisées dans le cinquième chapitre.

3.2.2. Étude de cas 2 : Sphère sociale

La figure 3.4 illustre le schéma global des échanges dans le cadre de Net-entreprises [GIP 2002a ; GIP 2002b].

Dans le premier chapitre, nous avons expliqué que Net-entreprises est le service proposé aux entreprises par l'ensemble des organismes de protection sociale pour leur permettre d'effectuer, par Internet, leurs déclarations et leurs paiements. Le site Net-entreprises est donc un site permettant l'accès à l'ensemble des déclarations (net-xxx) destinées à différents Organismes de Protection Sociale (OPS). La figure 3.4 est divisée en deux parties. Une partie montre les liens entre les entreprises et le site portail de Net-entreprises et une autre partie décrit les flux entre Net-entreprises et les autres organismes tels que les organismes de protection sociale".

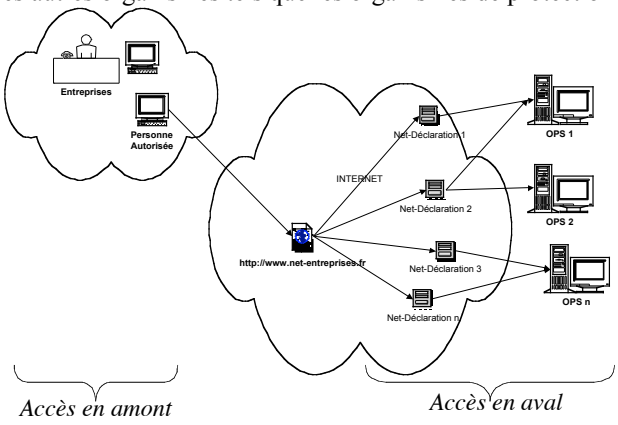


Figure 3.4 : Scénario social.

3.2.2.1 Scénario d'accès en amont

3.2.2.1.1 Typologie des entreprises

Dans la partie *accès en amont* de la figure ci-dessus, plusieurs types de structures organisationnelles peuvent être distingués, selon que l'entreprise est composée d'un ou de plusieurs établissements, selon qu'elle effectue elle-même ses déclarations ou qu'elle les délègue à un tiers, etc. :

- entreprises mono-établissement (un seul N° de SIRET, numéro qui identifie l'établissement de l'entreprise) ;
- entreprises multi-établissements : plusieurs établissements dont l'un est le siège social ;
- tiers déclarant (centres de gestion agréés ou cabinets d'expertise comptable) : établissements habilités par l'entreprise à saisir et transmettre ses déclarations sociales ;
- unités déclarées : des sous-ensembles d'un établissement ou d'une entreprise ; elles peuvent correspondre à des découpages fonctionnels géographiques ou hiérarchiques (l'établissement souhaite que les déclarations concernant une certaine partie du personnel soient effectuées séparément des autres) ;

- établissement adhérent : c'est l'établissement qui va utiliser Net-entreprises pour saisir ses propres déclarations ou le tiers déclarant qui saisit les déclarations d'autres entreprises.

3.2.2.1.2 Rôles fonctionnels dans l'entreprise

3.2.2.1.2.1 Mandataire social

C'est le dirigeant du siège de l'entreprise. Du fait qu'il est le représentant légal, il est informé par courrier des inscriptions (aux services de Net-entreprises) effectuées pour son entreprise. Lorsque ces personnes sont autorisées à effectuer des déclarations pour des entreprises extérieures, le mandataire de l'entreprise à laquelle appartiennent ces personnes sera lui aussi tenu informé.

3.2.2.1.2.2 Dirigeant d'établissement

Le dirigeant d'établissement est le responsable d'un établissement. Pour une entreprise mono-établissement, ce rôle se confond avec celui de mandataire social. Au regard de Net-entreprises, les dirigeants acteurs dans les processus fonctionnels sont les dirigeants des établissements adhérents. Le rôle du dirigeant est, pour ce qui concerne l'inscription, un rôle de supervision des informations connues par Net-entreprises.

3.2.2.1.2.3 Administrateur

L'administrateur gère les inscriptions pour l'établissement adhérent dont il dépend, sous la responsabilité du dirigeant de l'établissement. Il s'occupe de la définition :

- des personnes autorisées à utiliser les sites et les services ;
- des services qui seront utilisés ;
- des droits de chaque personne, autorisée pour utiliser des services Net-entreprises.

3.2.2.1.2.4 Personne autorisée (PA)

La personne autorisée est celle désignée par l'administrateur pour :

- effectuer les déclarations via Net-entreprises ou
- consulter des déclarations ancienne, ou
- effectuer des télépaiements.

3.2.2.1.3 Exemple de processus

Globalement, on distingue trois types de processus :

- le processus d'inscription proprement dit ;
- les processus d'accès aux services sécurisés de Net-entreprises (sites déclaratifs), avec bien évidemment, une authentification des utilisateurs et une vérification des habilitations ;
- les processus d'information et d'alertes : envoi de courriers électroniques, gestion des alertes, dont certaines sont personnalisées et envoyées ou non aux déclarants.

Détaillons quelques exemples de processus.

3.2.2.1.3.1 Le processus d'inscription de l'administrateur

Net-entreprises est un service accessible aux entreprises pour déclarer et éventuellement payer des cotisations sociales. La première étape est l'inscription d'un administrateur qui est un préalable obligatoire.

Ce processus se déroule en deux étapes :

- L'inscription de l'administrateur lui-même.
- L'inscription, par ce nouvel administrateur, de personnes autorisées ainsi que la définition de leurs droits d'accès aux services offerts par Net-entreprises.

L'administrateur peut interrompre son inscription en fin d'étape 1 et reprendre ultérieurement l'étape 2. Selon que l'organisation (déclarée) est mono ou multi-établissements, cette deuxième étape peut prendre deux formes relativement différentes : processus d'inscription mono-établissement, ou processus d'inscription multi-établissements.

3.2.2.1.3.2 Le processus d'inscription mono-établissement

Le processus est mené par un administrateur déjà inscrit. La définition des personnes autorisées et de leurs droits peut se faire selon l'une des deux filières (méthodes) suivantes :

- *flash* : inscription standard et rapide à choix multiples ;
- guidée : filière complète, permettant de faire des inscriptions sur mesure.

3.2.2.1.3.3 Le processus d'inscription multi-établissements

Il s'agit de l'inscription d'utilisateurs par un administrateur afin de gérer les droits d'accès aux déclarations de plusieurs établissements de son entreprise. L'administrateur :

- désigne une ou plusieurs personnes autorisées ;
- crée un ou plusieurs portefeuilles (ensemble d'établissements de la même entreprise) ;
- affecte à chaque personne autorisée la gestion d'un portefeuille.

Comme dans le cas d'un mono-établissement, l'administrateur a le choix entre une filière guidée ou *flash*.

3.2.2.1.3.4 Le processus d'inscription tiers-déclarant

Il s'agit de l'inscription d'utilisateurs par un administrateur (appartenant à un centre agrée ou un cabinet d'expertise comptable) qui souhaite autoriser des personnes à déclarer pour d'autres entreprises que la sienne. Ce type d'inscription affecte des entreprises à des portefeuilles d'entreprise. Ces portefeuilles sont constitués de la liste des entreprises que va devoir gérer une personne autorisée. Les deux filières possibles sont :

- la filière "de masse" : cette filière est destinée aux tiers-déclarants gérant un ensemble d'entreprises clientes assez conséquent ; ce processus de définition de droits d'accès aux déclarations n'est pas effectué en ligne, il s'appuie sur l'envoi, par courrier électronique, de fichiers à partir desquels des opérateurs saisissent les droits d'accès aux déclarations ;
- la filière "guidée" : cette filière donne la possibilité à l'utilisateur d'affecter les droits d'accès aux personnes autorisées par étapes, enchaînées une à une.

En réalité, chacun de ces processus nécessite des traitements plus complexes que ceux décrits dans cette section, les détails peuvent être trouvés dans [Abou El Kalam *et al*. 2003c].

3.2.2.2 Scénario d'accès en aval

Le scénario en back-office est résumé dans la figure 3.5

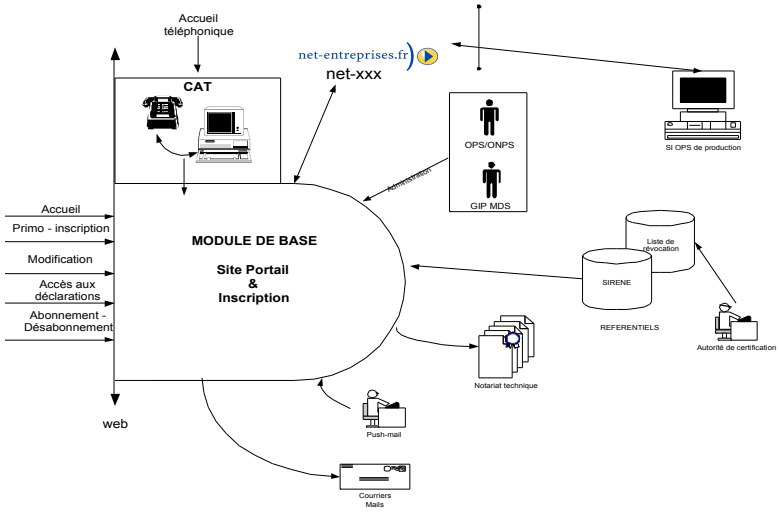


Figure 3.5 : Scénario d'accès en aval.

3.2.2.2.1 Les acteurs

Dans ce scénario, plusieurs organismes entrent en scène, notamment :

- Les OPS ou ONPS (Organismes Nationaux de Protection Sociale) sont les organismes récepteurs d'une ou plusieurs déclarations sociales, quelle qu'en soit la nature. Ils ont comme tâches principales de : fournir, *via* le site, des informations et alertes aux entreprises inscrites ; d'accéder à des informations telles que les statistiques ou la base des inscrits, *via* des outils de requête et de gérer la partie du référentiel OPS qui les concerne. Ce référentiel contient la liste des OPS, leur groupe d'appartenance, leur ONPS de rattachement, leur adresse et éventuellement une adresse de page web (URL). Le référentiel sera utilisé pour l'inscription lors du choix des organismes destinataires.
- Le GIP-MDS (Groupement d'Intérêt Public, Modernisation des Déclarations Sociales) est la structure de pilotage dont se sont dotés les organismes de protection sociale pour permettre aux entreprises d'effectuer leurs déclarations sociales au moyen de téléservices. Outre la gestion du site "www.net-entreprises.fr", le GIP a pour vocation de contribuer à la diffusion des bonnes pratiques, de coordonner les travaux des organismes pour les différentes déclarations, de fournir des informations et alertes aux entreprises et d'utiliser des services transversaux, comme l'accès à des statistiques, des bases de données ou des outils de requête sur les informations disponibles sur le site.
- Les fournisseurs de référentiels. Il existe trois grands types de référentiels :

  Le gestionnaire du référentiel des entreprises (liste des OPS, leurs groupes d'appartenance, leurs ONPS de rattachement) : il est géré par l'INSEE à travers le fichier SIRENE. Ce référentiel est dupliqué à la CNAVTS (Caisse Nationale d'Assurance Vieillesse des Travailleurs Salariés). Aujourd'hui, c'est cette copie qui est utilisée (par le site d'inscription hébergé par cet ONPS), car elle est enrichie d'informations issues de l'exploitation annuelle des déclarations, des informations

relevées par le service d'inspection des accidents du travail et d'autres informations issues du dialogue avec les autres OPS.

Référentiels bancaires : ils sont gérés par la banque de France. Ce référentiel est transmis de façon régulière à Net-entreprises pour permettre des contrôles sur les numéros de codes de guichet et de banque.

Autorités de certification : elles émettent et gèrent les certificats électroniques utilisés pour la sécurisation des accès et échanges dans le cadre de Net-entreprises.

3.2.2.2.2 Les services

Les utilisateurs se connectant à Net-entreprises *via* Internet arrivent d'abord sur une page d'accueil générique. Cette page d'accueil est un point de passage obligé pour accéder à l'ensemble des services du site, notamment le service d'inscription et les services déclaratifs.

3.2.2.2.2.1 Accueil et informations en ligne

Ce service est destiné aussi bien aux inscrits qu'aux personnes souhaitant s'inscrire. Il sert à effectuer des opérations de différentes natures : démonstration, simulation de cotisation ou de déclaration, actualités, consultation d'informations sur les partenaires et l'organisation de Net-entreprises, gestion des demandes de renseignements sur Net-entreprises, dépôt de commentaires et consultation des conditions générales d'inscription.

3.2.2.2.2.2 La primo-inscription

C'est l'inscription du premier administrateur rattaché à un établissement donné. Elle induit l'adhésion de son établissement à Net-entreprises en fournissant le numéro SIRET de l'établissement de l'administrateur. L'administrateur spécifie s'il souhaite être déclarant pour le compte d'un autre établissement en tant qu'établissement de la même entreprise ou en tant que tiers-déclarant.

3.2.2.2.2.3 Définition des habilitations par l'administrateur

Une fois inscrit, l'administrateur peut inscrire et gérer un ensemble de personnes et leur affecter des droits pour effectuer des déclarations ou payer des cotisations pour leurs établissements, d'autres établissements de la même entreprise, ou d'autres entreprises.

3.2.2.2.2.4 L'attribution d'un mot de passe pour les personnes inscrites

Ce service s'adresse aux personnes déjà inscrites ne possédant plus le moyen d'accéder à leur Accueil Personnalisé. Différentes causes sont possibles :

- ils ne se souviennent plus du mot de passe qui leur sertt de clé d'authentification ;
- le mot de passe n'est plus valide (lorsqu'il atteint la fin de sa période de validité) ;
- ils ont égaré le certificat enregistré par Net-entreprises qui leur sert de clé d'authentification ;
- le certificat n'est plus valide (expiré ou révoqué).

3.2.2.2.2.5 Abonnement ou désabonnement à un service d'information

Ce service s'adresse aux utilisateurs n'étant pas forcément inscrits à Net-entreprises (les journaliste, par exemple) mais souhaitant tout de même recevoir de l'information par courrier électronique concernant Net-entreprises.

3.2.2.2.2.6 Des fonctionnalités d'administration du site

Il s'agit des statistiques sur les inscriptions, statistiques techniques, la gestion de référentiels, la gestion de l'information diffusée, les éléments de supervision, etc.

Le tableau 3.1 résume les fonctionnalités de ces services et présente les accès.

Service	*Description*	*Accès*
Démonstration	Permet de se procurer le logiciel de démonstration décrivant les fonctionnalités offertes par Net-entreprises	À partir de l'Accueil Générique ; accessible à tous.
Accès à la simulation de déclaration ou de cotisation	Donne l'accès à une fonctionnalité de simulation de déclaration (ou de cotisation) parmi celles accessibles sur le site	À partir de l'Accueil Générique ; accessible à tous.
Qualification des logiciels	Permet de tester les développements effectués sur les logiciels destinés à élaborer les fichiers transmissibles	À partir de l'Accueil Générique ; accessible à tous les éditeurs de logiciel.
Abonnement ou désabonnement aux services d'alertes	Pour des utilisateurs non inscrits à Net-entreprises qui souhaitent être informés par courrier électronique des nouvelles concernant le site ou les déclarations.	À partir de l'Accueil Générique ; accessible à tous.
Consultation d'informations sur Net-entreprises	Permet à l'utilisateur de disposer, à partir du même service, de l'ensemble des informations concernant les partenaires et l'organisation de Net-entreprises.	À partir de l'Accueil Générique ; accessible à tous.

Tableau 3.1 : Accès en aval aux services de Net-entreprises.

3.2.2.3 Ressources à protéger, menaces, exigences et règles de sécurité

Afin de ne pas alourdir ce mémoire, il a semblé préférable de classer directement les ressources à protéger, ainsi que les menaces auxquelles elles sont confrontées, en fonction des propriétés de sécurité (disponibilité, confidentialité, l'intégrité et responsabilité).

3.2.2.3.1.1 Disponibilit

Composant	*Menace*	*Quelques solutions*
Poste de travail de l'utilisateur.	Incompatibilité avec le site ou le navigateur du téléservice ; Défaillance ponctuelle : panne, virus, etc.	Compatibilités des services selon les systèmes, les navigateurs et les versions.
Accès de l'utilisateur à Internet	Panne modem, impossibilité de se connecter ; indisponibilité du fournisseur.	

Accès au serveur d'administration par Internet	Indisponibilité physique : absence de connexion des serveurs au réseau ; insuffisance de la bande passante (critique si elle se répète dans le temps ou si elle intervient dans les heures précédant l'échéance des télédéclarations ou télépaiements).	Contraintes auprès de l'hébergeur d'accès multiples au réseau, de bande passante suffisante, sites de secours ; solutions organisationnelles comme le report des dates des échéances.
Application d'administration (téléservices, inscription, déclarations).	Surcharge (encombrement), indisponibilité suite à un problème technique dûe à une faute de conception ou à une attaque.	Tolérance aux fautes (traitement de fautes, et des erreurs) ; tests de charges ; contrer les attaques de DoS par une inscription préalable au téléservice, de façon à filtrer les accès ultérieurs;
Informations temporaires (avant validation) saisies par l'utilisateur.	Perte des informations avant que le processus relatif à ce service ne soit entièrement validé ; interruption inattendue d'un processus.	Sauvegardes page à page des informations utiles.
Télédéclarations (ou télérèglements ou accusés de réception) pour consultation ultérieure ou pour preuve.	Indisponibilité des données suite à une suppression illégitime ou à un problème technique matériel ou logiciel ; non-conservation des données relatives aux télédéclarations ou aux télérèglements.	Conservation (par Net-entreprises) des données conformément aux dispositions légales relatives à chaque déclaration ; conservation des accusés de réception par le déclarant.

Tableau 3.2 : Menaces pouvant porter atteinte à la disponibilité dans le social.

Quelques règles, relatives à la disponibilité, extraites des conditions d'adhésion aux services de Net-entreprises peuvent être ajoutées :

- (*Article* 7) : les données relatives aux télédéclarations sont conservées conformément aux dispositions légales et conformément aux règles spécifiques à chaque déclaration. Sauf stipulation contraire et conformément aux règles spécifiques à chaque déclaration, le « déclarant » pourra consulter par l'intermédiaire de Net-entreprises les données concernant les télédéclarations préalablement effectuées et pour lesquelles il est inscrit.
- (*Article* 9) : le service est accessible sept jours sur sept et 24 heures sur 24 pour l'inscription et les déclarations événementielles, et dans le respect des calendriers déclaratifs spécifiques à chaque déclaration à échéance. Toute défaillance relevant du site-portail ou du site déclaratif se traduit par l'émission d'un message indiquant à l'utilisateur l'indisponibilité du service ou le non-enregistrement des informations saisies. En pareil cas, celui-ci doit effectuer une nouvelle tentative ou accomplir ses obligations pour la date limite d'exigibilité par les moyens traditionnels.

3.2.2.3.1.2 Confidentialité

Information sensible	*Menace*	*Solutions*
Accès en aval : Informations nominatives de nature industrielle ou commerciale comme le numéro SIRET d'un travailleur indépendant, les données médicales ; *Accès en aval :* les salaires des cadres, les données soumises au secret professionnel.	Divulgation, à travers le téléservice, des informations sensibles à des personnes externes non habilitées (externes au système).	Contrôle d'accès ; échanges sécurisés entre le déclarant et le serveur, par exemple avec des mécanismes comme SSL ou TSL.
	Divulgation par le téléservice des informations sensibles à des personnes internes non habilitées ; vol de privilèges ; abus de privilèges.	Gestion des habilitations avec une séparation entre l'identité de la personne et le contrôle des droits. Possibilité d'utiliser des certificats d'attribut accolés à un certificat de signature.
Confidentialité des mots de passe de l'administrateur et des déclarants	Perte, divulgation ou vol des mots de passe.	Utilisation de moyens matériels (cartes à puce, par exemple) et/ou biométriques

Tableau 3.3 : Menaces pouvant porter atteinte à la confidentialité dans le social.

Quelques règles, relatives à la confidentialité, extraites des conditions d'adhésion:

- (*Article* 1) : l'inscription confère au représentant de l'employeur ou du tiers déclarant mandaté par l'employeur le statut d'administrateur, et lui permet de désigner les personnes de son choix (personnes autorisées) pour effectuer une ou plusieurs déclarations (ou télérèglements). Ces personnes sont ci-après dénommées le *déclarant*.
- (*Article* 2) : seule (s) la ou les personne(s) autorisée(s) désignées par l'administrateur Net-entreprises peuvent effectuer la ou les déclaration(s) ou accéder aux services sécurisés pour lesquels elles sont inscrites.

3.2.2.3.1.3 Intégrité

Information à protéger	*Risque*	*Solution*
Information communiquée à l'application (c'est-à-dire, saisie par l'utilisateur)	Non-conformité.	Contrôle syntaxique ; tout rejet dû à un contrôle doit entraîner une alerte de l'utilisateur avec une demande de correction.
Information qui transite par le réseau	Altération non autorisée	Contrôle d'accès ; chiffrement ; etc.
Services et programmes offerts par Net-entreprises.	Contournement ou modification non légitime de la part de l'utilisateur	Contrôle d'accès ; signatures ; tests d'intégrité et certification.

Tableau 3.4 : Menaces pouvant porter atteinte à l'intégrité dans le social.

3.2.2.3.1.4 Responsabilité

La question de responsabilité dans les téléprocédures est liée à de multiples facteurs : les textes juridiques en vigueur, les besoins fonctionnels des partenaires au téléservice, les

solutions techniques et organisationnelles mises en œuvres ainsi que le niveau de risque considéré comme acceptable. L'identification des responsabilités s'intéresse, entre autres, aux différents motifs de litiges concernant l'information ou la méta-information :

- existence de la déclaration (méta-information) ;
- contenu de la déclaration (information) : ce qui est saisi et envoyé d'une part, et ce qui est reçu d'autre part ;
- moment de la déclaration (méta-information) sachant que tout retard entraîne des pénalités.

Risque	*Description*	*Solution*
L'administration considère n'avoir pas reçu une déclaration que l'usager affirme avoir effectué.	Le déclarant est *a priori* responsable, et il ne peut dégager la responsabilité qu'en apportant la preuve de la défaillance de l'administration.	La délivrance d'un accusé de réception soit en mode EFI (envoi immédiat) ou EDI (asynchrone, par exemple, par envoi de courrier électronique après vérification de la conformité du format).
L'usager n'a rien envoyé alors que l'administration a reçu une déclaration	Deux cas possibles : mauvaise foi du déclarant ou usurpation malveillante de l'identité de l'usager (fausse déclaration ; demande de mutation)	Mécanismes d'authentification (certificat, mécanismes biométriques, etc.).

Tableau 3.5 : Menaces pouvant porter atteinte à la responsabilité dans le social.

La disponibilité à long terme de certaines méta-informations peut contribuer à assurer la propriété de non-répudiation. En l'occurrence, le stockage des accusés de réception permet de se prémunir contre les litiges. Ainsi, lors d'un désaccord sur le contenu de l'échange, le déclarant pourra présenter l'accusé de réception (envoyé par le téléservice) mentionnant les caractéristiques principales du fichier. De la même manière, la sauvegarde des modèles de courriers à faire parvenir aux centres des impôts, permet de se prémunir contre certaines défaillances.

L'intégrité joue également un rôle important dans la non-répudiation. En effet, la signature électronique de l'accusé de réception aura une valeur probante plus forte (non signé, il pourrait être falsifié). De même, l'intégrité de l'identité de l'émetteur est très importante car, si l'identification est très simple (par exemple en utilisant le nom patronymique, la raison sociale ou le numéro SIRET), le système est confronté à des *risques* de doublons, de confusions ou même d'usurpations des identités.

Par ailleurs, dans les applications du domaine social, il est nécessaire de tracer les modifications des informations des bases de données (des OPS, référentiels, etc.). Cette fonction notariale peut être étendue au traçage des actions effectuées par l'utilisateur, en particulier à chaque changement de page ou d'écran, en prenant en compte les impacts sur les performances et les volumes de stockage. Outre le fait qu'elles permettent le suivi des opérations effectuées par les utilisateurs sur le site, les informations notariales servent également à effectuer des statistiques. Elles devront être accessibles aux OPS, au GIP-MDS (Groupement d'Intérêt Public, Modernisation des Déclarations Sociales), au CAT (Centre d'Accueil Téléphonique) ainsi qu'à certains internautes au moyen du service « *Consultation de l'historique des modifications* ». Bien évidemment, il est nécessaire de pouvoir isoler ces types d'informations : les modifications sur les noms, prénoms, et courriers électroniques des

personnes autorisées n'intéressent pas les dirigeants alors qu'elles peuvent intéresser les administrateurs (protection de la vie privée). Dans ce cas, un dirigeant verra l'historique des modifications survenues dans son entreprise à l'exception de ces informations.

3.2.3. Étude de cas 3 : Analyse de différents scénarios d'anonymisation d'informations médicales

3.2.3.1 Problématique de l'anonymisation

Bien que l'instauration du réseau de soins facilite la communication des données entre différentes structures, elle pose des problèmes concrets de sécurité. D'une part, l'usage d'informations médicales nominatives pour les soins impose d'avoir l'assurance de l'identité des personnes auxquelles se rapportent ces informations. D'autre part, en facilitant l'échange, le partage et le traitement des données de santé entre acteurs, la reconnaissance physique[19] du patient peut contribuer à briser le secret médical ou à permettre d'inférer des informations confidentielles, par exemple en constituant des listes de personnes atteintes de certaines maladies (informations dont des sociétés d'assurance, entre autres, pourraient tirer profit).

Les législations internationale [Résolution 1990] et européenne [Directive 2002 ; Directive 1995 ; Recommandation 1997] visent à protéger les données personnelles et interdisent tout croisement de fichiers. De même, en France, la loi informatique, fichiers et libertés [Loi 1978] accorde une protection particulière aux données nominatives. Il existe toutefois une certaine nuance entre donnée nominative et donnée à caractère personnel. Nous pensons qu'une donnée peut être "non-nominative" tout en restant "personnelle", alors que certains travaux ne font pas la différence entre *donnée nominative* et *donnée à caractère personnel* lorsqu'ils considèrent que, après anonymisation, une donnée nominative (devenue anonyme) ne peut plus être vue comme étant à caractère personnel. Ceci n'est pas toujours exact dans la mesure où, il est parfois possible, de ré-identifier des données anonymisées. Ainsi des données anonymisées peuvent perdre ou non leur caractère anonyme, de même que des données nominatives peuvent perdre leur caractère nominatif. Mais dans tous les cas, nominatives, anonymes ou anonymisées, ces données restent à caractère personnel. Les anglo-saxons parlent d'ailleurs de "de-identification" au lieu d'anonymisation.

Malheureusement, il est souvent possible d'identifier un individu par un simple rapprochement de données personnelles de nature médicale ou sociale. Par exemple, l'âge, le sexe et le mois de sortie de l'hôpital, permettent d'isoler le patient dans une population restreinte ; la donnée de deux dates (voire de deux semaines) d'accouchement pour une femme permet de l'isoler dans une population plus grande (typiquement, la population d'un pays comme la France). D'ailleurs, le domaine de l'inférence d'information dans les bases de données a été étudié depuis de nombreuses années, et il a fait l'objet d'une abondante littérature [Cuppens 2003].

Selon les besoins, le choix en terme de protection de données peut faire appel à des solutions techniques comme le hachage et le chiffrement, ou organisationnelles comme les politiques de contrôle d'accès et les anonymisations thématiques.

Un premier niveau de confidentialité peut être assuré en chiffrant les données transmises, de façon à ce qu'elles ne soient déchiffrées que par le ou les destinataires légitimes. Par exemple, ce peut être le cas d'un échange de données médicales entre le laboratoire d'analyses médicales et le médecin traitant. Dans d'autres cas, on souhaite garder l'anonymat de certaines données

[19] La reconnaissance physique est utilisée dans le sens de rétablissement de la correspondance entre les identifiants et les personnes physiques.

(en l'occurrence, les identifiants) même si le destinataire est légitime. Par exemple, garder l'anonymat total à l'égard des assurés ou des ayants-droit lors de publications scientifiques, ou lors de la transmission des informations relatives à l'activité des médecins libéraux vers les *Unions Professionnelles* (assemblées de médecins élus par région ou regroupés en sections). Si le but est de pouvoir effectuer des trajectoires de soins, c'est-à-dire, avoir un suivi permanent des évolutions des maladies, on utilise un code anonyme, mais toujours le même pour un patient donné. Enfin, il est parfois souhaitable qu'une autorité puisse croiser les données anonymisées avec d'autres données anonymes concernant le même individu, ou même lever l'anonymat dans des cas bien particuliers. Par exemple, dans le cas des études épidémiologiques, les corrélations entre plusieurs pathologies peuvent nécessiter de remonter aux identités réelles pour compléter a posteriori les données recueillies antérieurement, et ainsi affiner ces études.

Notons que, compte tenu des caractéristiques des différents scénarios d'anonymisation, la méthodologie utilisée dans cette section est légèrement différente de celle appliquée dans les deux premières études de cas. Ainsi, nous procédons selon les étapes suivantes :

- d'abord, définition de la base terminologique associée au thème "anonymisation" ;
- puis discussion des différentes solutions d'anonymisation utilisées dans les pays européens ;
- ensuite, définition d'un ensemble de scénarios relatifs au thème d'anonymisation ;
- enfin, pour chacun de ces scénarios, construction d'une démarche d'analyse des besoins : identification des besoins, objectifs et exigences de sécurité et, dans la mesure du possible, choix de solutions adaptées.

Un *besoin* d'anonymisation représente les attentes de l'utilisateur (pas toujours très bien explicitées). Un *objectif* d'anonymisation spécifie le niveau de sécurité à atteindre ou les menaces à éviter (comment satisfaire les exigences ?). Une *exigence* d'anonymisation représente la façon d'exprimer le besoin (dans la mesure du possible, très proche d'un formalisme clair et d'une sémantique non-ambiguë).

Même si tous les scénarios que nous évoquons dans cette section sont extraits du domaine médical, les besoins d'anonymisation de certaines données sensibles concernant des personnes ou des entreprises est également un point crucial. La méthodologie que nous proposons pour l'analyse de la procédure d'anonymisation peut être appliquée aux systèmes de santé, mais aussi à d'autres secteurs, notamment les secteurs social, bancaire, militaire, etc.

3.2.3.2 Notion d'objectifs d'anonymisation

L'anonymisation peut être définie comme une mesure visant à empêcher de déterminer l'identité réelle d'une personne ou, du moins, à ne rendre cette détermination possible qu'au prix d'efforts démesurés. La terminologie des critères communs [CC 1999a] définit la pseudonymisation comme une anonymisation telle que la personne concernée puisse être tenue comme responsable de ses actes. En pratique, cela signifie que la pseudonymisation est une anonymisation réversible, c'est-à-dire permettant à des personnes dûment autorisées à cet effet de retrouver l'identité réelle de la personne concernée. La chaînabilité peut être définie comme la propriété de pouvoir déterminer que différentes informations correspondent à une même personne, sans nécessairement connaître l'identité réelle de cette personne.

Un objectif d'anonymisation est défini en fonction de l'une des trois propriétés suivantes applicables à la fonction d'anonymisation [Trouessin 2001 ; Abou El Kalam *et al*. 2003c] :

- *réversibilité* : cacher les données par un simple chiffrement des données. Dans ce cas, il y a possibilité de remonter depuis les données chiffrées jusqu'aux données nominatives originelles.
- *irréversibilité* : c'est le cas réel de l'anonymisation ; une fois remplacés par des identifiants anonymes, les identifiants nominatifs originels ne sont plus recouvrables ; cependant, avec les techniques d'attaques par inférence[20], les identifiants anonymes, s'ils sont trop universellement utilisés, risquent de permettre la découverte d'identités mal cachées, comme on l'explique ci-après ; pour ce type d'anonymisation, la technique communément utilisée est une fonction de hachage ;
- *inversibilité* : c'est un cas mixte entre réversibilité et irréversibilité, c'est-à-dire entre le "techniquement ou cryptographiquement irréversible" et le "organisationnellement et juridiquement réversible" ; autrement dit, il est impossible en pratique de remonter aux données nominatives, sauf en appliquant une procédure exceptionnelle sous surveillance d'une instance légitime (médecin-conseil, médecin inspecteur) garante du respect de la vie privée des individus concernés ; il s'agit cette fois-ci d'une pseudonymisation au sens des critères communs.

3.2.3.3 Notion d'exigences d'anonymisation

Des informations sur l'environnement[21] informatique du système étudié permettent de compléter l'analyse du besoin. En l'occurrence, même si les informations sont anonymes, un utilisateur malveillant peut construire divers types de raisonnement pour déduire des informations confidentielles. Les exigences d'anonymisation sont exprimées en terme de *chaînage* (continuité de l'anonymisation) et de *robustesse* (sûreté de l'anonymisation).

Le *chaînage* permet d'associer un ou plusieurs identifiants anonymes à une même personne physique. Comme indiqué sur la figure 3.6, un chaînage peut être temporel (toujours, parfois, jamais) ; géographique (international, national, régional, local) ; ou spatio-temporel (par exemple, "toujours et partout", "parfois et partout", "local et jamais") [AFNOR 1997].

La *robustesse* d'un système d'anonymisation est constituée de l'ensemble des caractéristiques à satisfaire vis-à-vis d'attaques ayant pour but de lever l'anonymat de façon non-autorisée. Il peut s'agir d'une *robustesse à la réversion* concernant la possibilité ou l'impossibilité d'inverser la fonction d'anonymisation, mais il peut aussi s'agir d'une robustesse à l'inférence qui consiste à déterminer des informations nominatives à partir d'éléments d'informations purement anonymes. En général, une inférence peut être :

- *déductive* : elle utilise la logique du premier ordre (valeurs : oui, non ; opérateurs : et, ou) pour déduire des informations confidentielles non accessibles ; par exemple, si un certain patient fait un test de dépistage *puis* dans les quelques jours qui suivent, il fait un test de dosage, *alors* le résultat du dépistage était positif ;
- *inductive* : s'apparente souvent à des raisonnements de type loi des grands nombres sans forcément l'appliquer sur de grandes échelles ; cela consiste par exemple, à induire qu'un tel patient est très certainement atteint de telle pathologie compte tenu du fait qu'il lui est prescrit tels médicaments comme il est d'usage pour cette pathologie ;
- *abductive* : lorsqu'un raisonnement classique utilisant les informations explicitement stockées dans le système d'informations ne permet pas d'inférer d'informations, mais ce

[20] Une inférence est la découverte de données confidentielles non directement accessibles, rendue possible par la mise en correspondance de plusieurs données légitimement accessibles, révélant tout ou partie des informations relatives à une personne.

[21] Par environnement informatique du système, on entend l'ensemble des utilisateurs, la nature des attaquants et les types des attaques.

raisonnement pourrait êtte complété en faisant des hypothèses sur certaines informations, par exemple, “et s’il avait un cancer, cela expliquerait pourquoi il s’absente du Conseil des Ministres pour se rendre à l’hôpital Paul Brousse de Villejuif …”.

- *probabiliste* (*ou adductive*) : elle parvient à estimer la vraisemblance d’une information sensible en utilisant les informations accessibles, par exemple, “puisque P est traité à l’hôpital H, et puisque H est spécialisé dans les maladies M_1 et M_2, et puisque à son âge, la probabilité d’avoir M_1 est très faible (10%), alors on peut déduire qu’à 90%, P est atteint de M_2”.

Cette liste n’est pas exhaustive et on peut naturellement imaginer d’autres types de canaux d’inférence fondés sur d’autres types de raisonnement, tel que le raisonnement par évidence ou par analogie.

Pour faire face à ce type de menaces et protéger les données personnelles, divers pays se sont intéressés aux fonctions d’anonymisation des données médicales utilisées à des fins non directement épidémiologiques.

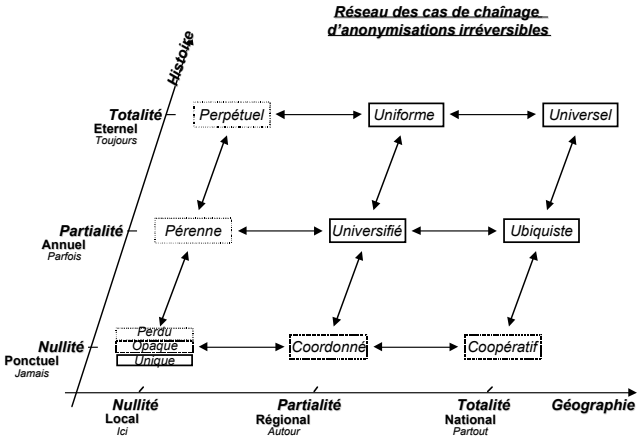


Figure 3.6 : Anonymisation en cascade : de l’universalité jusqu’à l’unicité.

3.2.3.4 L’anonymisation dans les pays européens

3.2.3.4.1 L’anonymisation en France

En 1995, le CHU de Dijon, ainsi que d’autres établissements de santé de la région Bourgogne ont choisi l’algorithme de hachage SHA (*Standard Hash Algorithm*) pour transformer, d’une manière irréversible (anonymisation), les variables d’identification : nom, prénom, date de naissance et sexe. Le but est d’obtenir un identifiant strictement anonyme, mais toujours le même pour un patient donné [Quantin *et al*. 1998]. Néanmoins, même si SHA est mathématiquement irréversible, il ne résiste pas aux attaques “ponctuelle” ou “par dictionnaire ”. En effet, supposons qu’un tiers malveillant, Bob, a accès à une base de données médicale anonyme *BDMA*.

Dans une attaque ponctuelle, Bob voudrait, par exemple, savoir si Alice a le SIDA. La première étape consiste à appliquer l’algorithme de hachage aux variables identifiantes d’Alice afin d’obtenir l’identifiant anonyme associé à Alice : NA_{Alice}. La deuxième étape consiste à chercher si NA_{Alice} existe dans la base de donnée anonyme des personnes ayant le SIDA $BDMA_{\text{SIDA}}$.

Dans une attaque par dictionnaire, Bob pourrait appliquer l'algorithme de hachage à une liste de variables identifiantes (dictionnaire) pour disposer d'une table $T_{id\text{-}Num}$ reliant des variables identifiantes à des codes anonymes. En faisant un simple rapprochement entre la base de données médicales anonymes et la table, il pourrait facilement déduire les informations médicales des personnes dont les noms figurent sur son dictionnaire (figure 3.7).

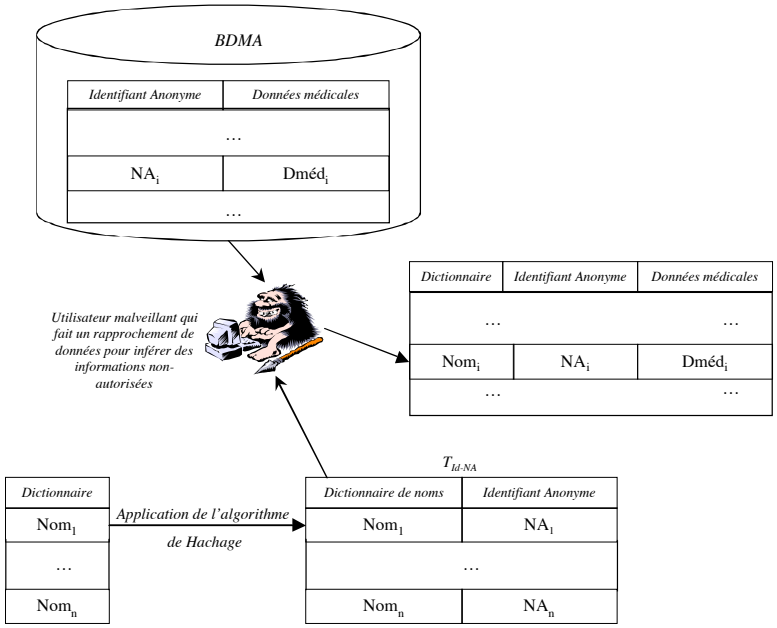


Figure 3.7 : Attaque par dictionnaire.

Pour illustrer la procédure d'anonymisation utilisée dans la région Bourgogne, considérons maintenant le cas où les hôpitaux transmettent des données médicales au Département d'Informations Médicales (DIM). Celui-ci effectue des analyses médico-économiques, avant d'archiver toutes les données médicales de tous les patients (aux niveaux des agences régionales d'hospitalisation par exemple).

Si on n'utilise qu'une fonction de hachage du côté des hôpitaux, les personnes en charge des traitements statistiques et médico-économiques (au DIM) peuvent remonter aux identités par une simple attaque ponctuelle (ils disposent des données médicales anonymes et de l'algorithme de hachage). Par ailleurs, du côté du DIM, si on n'utilise qu'une fonction de hachage avant l'archivage, les employés des hôpitaux pourront retrouver les identifiants (utilisés dans les archives) de n'importe lequel de leurs patients, ainsi que toutes les informations concernant ce patient, y compris celles provenant des autres établissements (rappelons que la loi [Loi 2002] donne au patient le droit d'interdire que certaines de ces données soient communiquées à d'autres professionnels de santé "voir section 3.2.2.1.5").

Pour prévenir ce type d'attaques, deux clés ont été ajoutées à l'algorithme de hachage SHA. La première clé k_1, utilisée par tous les émetteurs des données (hôpitaux et médecins), est concaténée à l'identité. Une fonction de hachage est ensuite appliquée au résultat : $empreinte_1 = \mathcal{H}(k_1 \mid identité)$. Cette opération produit une empreinte qui varie d'une identité à l'autre, mais qui est toujours la même pour un patient donné. Les informations transmises au centre de traitement des fichiers (DIM) en vue de leur rapprochement sont ainsi devenues strictement anonymes et les personnes qui assurent les traitements centralisés ne peuvent pas lever l'anonymat à l'aide d'une attaque par dictionnaire puisqu'elles ne connaissent pas la clé k_1. De

l'autre côté de la communication, les informations reçues par le DIM sont hachées par le même algorithme mais avec une seconde clé k_2, qui n'est pas communiquée aux hôpitaux : $empreinte_2 = \mathcal{H}(k_2 + empreinte_1)$ (voir figure 3.8).

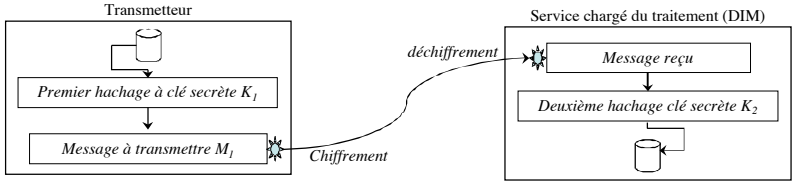


Figure 3.8 : Procédure de double hachage des informations traitées par le DIM.

Le protocole présenté en Bourgogne s'avère complexe et risqué. En effet, il nécessite une distribution de la même clé secrète à tous les fournisseurs d'informations (médecins libéraux, hôpitaux, cliniques, etc.), tout en supposant que cette clé doit rester secrète. Certes, une personne qui ne connaît pas une des clés secrètes (k_1 ou k_2) est dans l'impossibilité d'effectuer les transformations. Néanmoins, si une clé est corrompue, le niveau de sécurité est considérablement réduit. De même, si un jour il s'avère que l'algorithme (ou la longueur de la clé) n'est plus efficace, comment faire le rapprochement entre les identifiants avant et après changement de l'algorithme ou de la clé (sachant que les empreintes dépendent de la clé supposée être toujours la même et chez tous les fournisseurs d'informations) ? Si ce problème survient, la seule solution envisageable consiste à appliquer une autre transformation à toute la base de données, solution qui n'est guère aisée.

Étudions à présent une autre procédure qui a été élaborée par le CESSI de la CNAM-TS pour le PMSI (Programme de Médicalisation des Systèmes d'Information) privé : la procédure FOIN (Fonction d'Occultation d'Informations Nominatives). Comme en Bourgogne, la procédure de la CNAM utilise une fonction de hachage à sens unique (SHA) avec une clé des deux côtés. L'originalité de cette méthode réside dans l'utilisation de la technique de *Fragmentation-Redondance-Dissémination* ou FRD [Fabre *et al*. 1996]. Les étapes de cette technique de tolérance aux intrusions, déjà expliquée, sont les suivantes :

- découper l'information (clé secrète) en fragments de telle sorte que des fragments isolés ne puissent fournir d'information significative ;
- ajouter de la redondance pour empêcher que la modification ou destruction de quelques fragments n'ait pas de conséquence pour les utilisateurs autorisés ;
- isoler les fragments les uns des autres par dissémination de sorte qu'une intrusion (la corruption d'une partie de la clé secrète) dans une partie du système ne fournisse que des fragments isolés.

En utilisant cette technique de FRD pour protéger la clé secrète nécessaire à la fonction d'anonymisation, FOIN fragmente la clé secrète en N images de telle sorte qu'elle ne peut être reconstruite qu'à partir d'un certain nombre s (dit seuil de reconstitution) d'images différentes. En fait, cette notion de seuil repose sur le schéma à seuil de Shamir qui peut se résumer ainsi :

- le partage du secret k (clé secrète) est fait sur N images distinctes ;
- il s'effectue à l'aide d'un polynôme $P(x)$ de degré "s-1", s étant le seuil de reconstruction : $P(x) = a_0 + a_1x + .. + a_{s-1}x^{s-1}$;
- il suffit de rassembler s images distinctes parmi les N images pour permettre de recalculer le secret k ;
- les calculs se font dans un corps de Galois (corps fini offrant certaines propriétés mathématiques) ;

- le secret k est l'ensemble des coefficients du polynôme : (a_0, a_1, .., a_{s-1}). La figure 3.9 présente un exemple avec s=2 ($P(x) = a_0 + a_1x$).

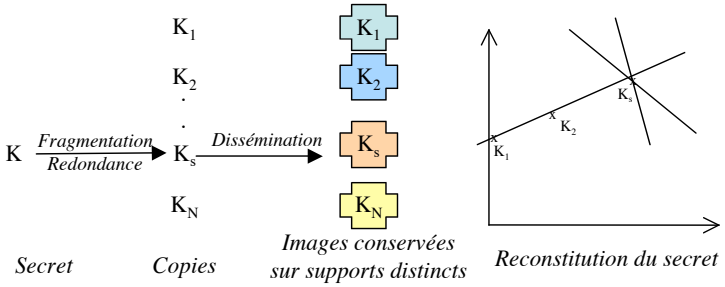


Figure 3.9 : Fragmentation-Redondance-Dissémination de la clé secrète.

Les images de la clé secrète sont disséminées sur un nombre de supports distincts. La première image sera placée dans la fonction d'anonymisation (dans le logiciel distribué aux transmetteurs d'informations), les autres sont données à des personnes de confiance, comme le responsable de l'application ou le directeur de la CNAM. Ainsi, même s'il existe N images (fragments) de la clé, la présence de "s-1" personnes de confiance est suffisante pour reconstituer le secret (figure 3.10).

Comme en Bourgogne, et pour se prémunir contre toute attaque ponctuelle ou par dictionnaire, la fonction d'anonymisation FOIN est utilisée à deux niveaux : une première fois dans les hôpitaux, avant de transmettre les données médicales des patients et une deuxième fois, avant l'archivage de ces données.

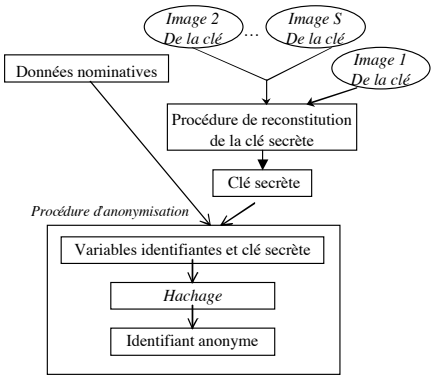


Figure 3.10 : Procédure FOIN.

Néanmoins, le cas où s = 2 est confronté aux mêmes faiblesses de la procédure du CHU de Dijon. Si en plus, N > 2, les images seront détenues par N personnes et n'importe laquelle pourra reconstituer la clé (puisqu'une des deux images nécessaires est présente dans le logiciel). De plus, la technique de FRD ne résout que le problème du stockage "longue durée". En effet, la clef reste vulnérable au vol par un utilisateur malveillant qui réussit à la lire ou la copier lorsqu'il l'utilise durant les traitements.

3.2.3.4.2 L'anonymisation des données sur le cancer en Allemagne

Le RNC (*Répertoire National du Cancer*) allemand est un registre épidémiologique fondé en 1953. De nos jours, il contient les données de plus de deux millions de patients atteints du cancer. Le but majeur est de pouvoir effectuer des statistiques médicales et des recherches épidémiologiques sur le cancer. Dans les années quatre-vingt, et pour des raisons techniques, seuls les fichiers centralisés ont été informatisés. Pour chaque cas, les détails suivants ont été enregistrés :

- identification personnelle du patient ;
- établissement, historique, stade, diagnostic et thérapie ;
- autres traitements et suivis médicaux ;
- historique individuel et familial ;
- décès et résultat de l'autopsie.

La procédure d'enregistrement transite par deux étapes et à travers deux institutions [Blobel 1996] :

- Dans un premier temps, un site de confiance rassemble les données auprès des médecins, dentistes et centres de suivi. Les données identifiantes sont d'abord chiffrées par une clé de session *IDEA* générée aléatoirement. Cette clé est elle-même chiffrée par une clé publique *RSA* d'une longueur minimale de 640 bits. Par ailleurs, et pour permettre une association non-ambiguë des informations au fichier du patient, un identifiant anonyme est généré à partir de certaines données personnelles du patient. Cet identifiant est en fait généré par l'application de la procédure de hachage à sens unique "*MD*5", suivi d'un algorithme de chiffrement symétrique (*IDEA*). Cet identifiant est le même, sur tout le territoire allemand, pour un patient donné. Le format obtenu est nommé "*format de chaînage*".
- Le site de confiance transmet au site d'enregistrement à la fois les données d'identité sous forme chiffrée et les données épidémiologiques en clair. Le site d'enregistrement enregistre le fichier dans la base de données du NCR et rassemble les données appartenant au même patient. Il procède de la manière suivante : un numéro aléatoire est ajouté à l'identifiant, et le résultat est chiffré par IDEA. Le format obtenu est nommé "*format de stockage*".

Sur demande et pour des raisons scientifiques, l'exploitation des données anonymisées du registre est possible, mais restreinte en temps (durée) et en quantité (nombre de fois). Dans certains cas particuliers, un comité de conseil peut autoriser le déchiffrement des données identifiantes.

Nous commenterons cet exemple après avoir présenté celui de la Suisse assez similaire.

3.2.3.4.3 Le traitement statistique des données médicales en Suisse

Du point de vue statistique, il n'est pas nécessaire de savoir à qui appartient un fichier médical. Néanmoins, en Suisse, l'Office Fédéral des Statistiques (OFS), responsable de la collecte des données médicales auprès des hôpitaux, a besoin de reconnaître si deux fichiers différents correspondent à la même personne. L'implémentation suisse propose de hacher les identifiants dans les hôpitaux avant de les transmettre à l'OFS ; puis à la réception, les données médicales sont chiffrées par la clé secrète de l'OFS [Jeanneret *et al*. 2001] (figure 3.11).

La première étape consiste à remplacer les données d'identité par un identifiant personnel, le code anonyme de lien. Les données sur lesquelles le calcul repose doivent être disponibles et constantes dans le temps. Trop de restrictions sur le choix peut conduire à des collisions, tandis qu'une multitude de données peuvent être non disponibles ou, du moins, changer au cours du

temps. Le choix s'est porté sur un ensemble minimal d'identifiants : la date de naissance, le sexe, le nom et le prénom.

Puisque la transformation cryptographique doit être appliquée tout le temps et par tous les hôpitaux, l'utilisation d'une clé secrète n'est pas la solution la mieux adaptée. À l'inverse, l'utilisation des empreintes[22] (hachage des données identifiantes) satisfait mieux ces objectifs, avec l'inconvénient de ne pas résister aux attaques par dictionnaires. Il convient donc de chiffrer les données d'identité, d'abord durant la transmission de l'hôpital à l'OFS, ensuite dans les bases de données. Les étapes de cette procédure sont les suivantes (figure 3.11) :

- Hachage des variables identifiantes : *Hachage*[*Var-ID*] = *Empreinte*.
- Génération en arrière-plan (dans l'ordinateur de l'hôpital) d'une clé de session : *c*.
- Chiffrement de l'empreinte avec IDEA en employant *c* : $IDEA[Empreinte]_c$; et chiffrement de *c* par la clé publique de l'OFS, (notée *E*) en utilisant l'algorithme RSA : $RSA[c]_E$.
- Transmission de la clé de session (chiffrée), de l'empreinte (chiffrée) et des données médicales (chiffrées) à l'OFS.
- À la réception, déchiffrement de la clé secrète (de session) c par RSA en employant la clé privée de l'OFS notée "*D*" ;
- Déchiffrement de l'empreinte (avec IDEA et la clé c), et re-chiffrement de cette empreinte avec une clé *k* (fragmentée) pour donner le lien anonyme utilisé comme code personnel (code de liaison uniforme) lors du stockage des données médicales au niveau de l'OFS.

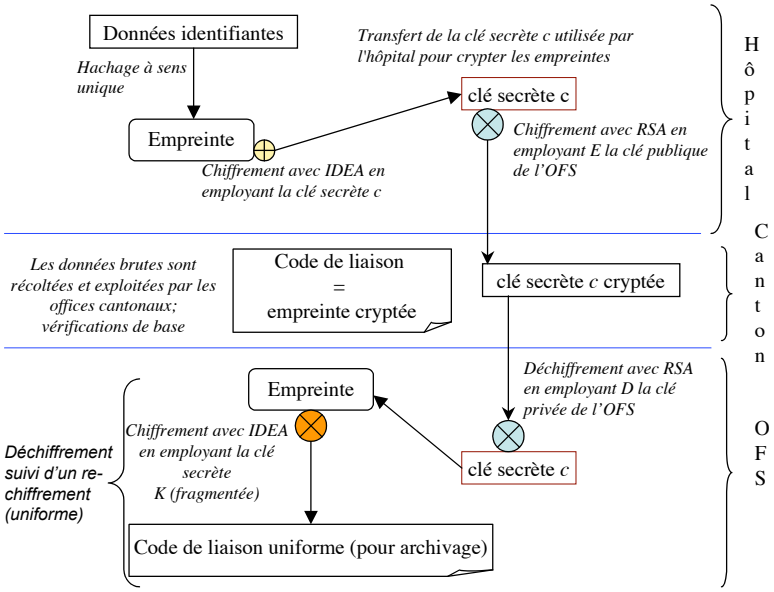


Figure 3.11 : Transformation des données identifiantes en Suisse.

Les solutions allemandes et suisses se ressemblent, c'est pourquoi on se contentera de discuter la deuxième. Comparons la procédure suisse et la fonction FOIN de la CNAM-TS :

[22] Rappelons qu'une empreinte a été définie dans la section 3.2.3.4.1 (page 81) comme étant un code anonyme qui varie d'une identité à une autre mais qui est toujours le même pour un patient donné.

- les données transmises sont chiffrées dans la solution suisse, tandis qu'elles ne le sont pas dans FOIN ;
- FOIN utilise la technique Fragmentation-Redondance-Dissémination qui vise, non seulement la confidentialité et l'intégrité, mais aussi la disponibilité. Dans la procédure suisse, même si K est fragmentée en plusieurs parties, elle ne peut être reconstituée qu'en présence de toutes les personnes disposant de ces fragments (de K). La reconstitution est faite par un simple ou exclusif "$K = K_1 \oplus .. \oplus K_N$" C'est donc un cas particulier de FOIN, où on considère $N = s$ (seuil de reconstitution).

Par ailleurs, les trois opérations effectuées au niveau de l'OFS (retrouver la clé secrète *c*, retrouver l'empreinte, et chiffrement avant archivage) ne doivent jamais être visibles aux utilisateurs de l'OFS. Cependant, comment s'assurer qu'elles ne sont jamais enregistrées sur aucun support ? Un cheval de Troie opérant pour une personne malveillante, aussi bien interne qu'externe, pourrait récupérer les valeurs de la clé secrète *c* ou des *empreintes*, et effectuer par la suite une attaque par dictionnaire. Pour pallier ce type de risques, nous pensons que ces étapes (phase de *calcul*) doivent être faites par un module matériel bien protégé. Des mécanismes inviolables de contrôle d'accès, éventuellement matériels, pourront renforcer la protection de ce module de façon à ce que seules les personnes de confiance puissent réaliser l'opération composite *calcul* ; le serveur d'autorisation leur donne les droits correspondants sans qu'elles puissent lire ou copier, ni l'*empreinte*, ni les clés secrètes *K* et *c* ; des droits distribués sont donnés aux composants matériels ; chaque composant effectue les opérations qui lui sont destinées.

Cette analyse montre, les avantages et les faiblesses de chacune des solutions actuelles et renforce notre volonté d'une démarche analytique préalable des risques, des besoins, des exigences ainsi que des objectifs de sécurité, avant de recourir aux solutions de sécurité assurant l'anonymisation. Complétons ainsi cette analyse par une étude détaillée d'un ensemble de scénarios du domaine médical.

3.2.3.5 Scénario 1 : transfert des données médicales

La sensibilité des informations échangées entre professionnels de santé (par exemple, le laboratoire d'analyses et le médecin) met en évidence le besoin de confidentialité et d'intégrité des données transitant sur le réseau de soins. La figure 3.12 schématise une des solutions qui consiste à utiliser un chiffrement asymétrique. Ainsi, en supposant que le destinataire légitime est le seul à disposer de la clé privée, personne d'autre ne peut déchiffrer le message transitant par le réseau et ainsi accéder aux données personnelles en clair.

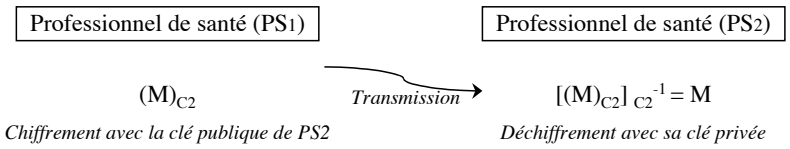


Figure 3.12 : Échange de données chiffrées entre professionnels de santé.

Si les données transmises sont volumineuses, il est préférable d'utiliser un chiffrement hybride, c'est-à-dire :

- *Du côté de l'émetteur* : générer aléatoirement une clé de session, considéré comme clé secrète valable pour la transmission en cours ; chiffrer le message avec cette clé de session, et chiffrer cette clé avec la clé publique du destinataire ; enfin émettre à la fois le message chiffré et la clé de session chiffrée ;

- *Du côté du destinataire* : déchiffrer la clé de session en utilisant sa clé privée, et déchiffrer par la suite le message avec cette clé de session.

Le chiffrement des données médicales transitant sur le réseau est actuellement une pratique courante entre les professionnels de santé, notamment en utilisant S-MIME.

Selon la classification donnée précédemment, l'*objectif* est une anonymisation réversible, tandis que l'*exigence* est la robustesse à l'inversion.

3.2.3.6 Scénario 2 : unions professionnelles

Le transfert des données relatives aux activités des médecins vers les unions professionnelles se fait à des fins d'évaluation de l'activité des médecins. Une première exigence consiste donc à cacher les identités du patient et du médecin. Toutefois, l'anonymat des médecins doit pouvoir être levé pour l'évaluation de leurs comportements en vue de la qualité de soins. En effet, l'article L4134-4 du code de la santé publique ainsi que l'article 81 de la loi 94-43 [Loi 94] précisent que « les médecins conventionnés exerçant à titre libéral dans la circonscription de l'union sont tenus de faire parvenir à l'union les informations mentionnées à l'article L. 161-29 du code de la sécurité sociale relatives à leur activité, sans que ces informations puissent être nominatives à l'égard des assurés sociaux ou de leurs ayants-droit ou, à défaut, à condition qu'elles ne comportent ni leur nom, ni leur prénom, ni leur numéro d'inscription au Répertoire national d'identification des personnes physiques. Ces informations ne sont pas nominatives à l'égard des médecins ». Ces textes ajoutent que « L'anonymat (des médecins) ne peut être levé qu'afin d'analyser les résultats d'études menées dans le cadre de l'évaluation des comportements et des pratiques professionnelles en vue de la qualité des soins ».

Les *objectifs* sont alors :

- l'anonymisation inversible de l'identité du médecin ; seule une autorité habilitée à évaluer les comportements des médecins pourrait rétablir les identités réelles des médecins ;
- l'anonymisation inversible des données nominatives du patient, seuls les médecins-conseils de la sécurité sociale pourront lever cet anonymat ; en effet, l'article L161-29 du code de la sécurité sociale ajoute : « seuls les praticiens-conseils et les personnels sous leur autorité ont accès aux données nominatives (des patients) issues du traitement susvisé, lorsqu'elles sont associées au numéro de code d'une pathologie diagnostiquée ».

Cette manière de faire évite les risques suivants (au niveau des unions professionnelles) :

- un utilisateur malhonnête qui tente d'avoir plus de détails sur les activités d'un médecin alors que la finalité de son traitement ne le justifie pas ; par exemple, dans le cadre d'une étude relative au fonctionnement du système de santé, il n'est pas nécessaire d'accéder aux identités (respect du principe du moindre privilège) ;
- atteinte à l'intimité des patients dans la mesure où ceux-ci peuvent confier des informations à certains professionnels de santé, sans pour autant avoir forcément envie de les communiquer aux autres professionnels de santé ou personnes en charge des traitements au sein des unions.

Le scénario décrit dans cette section est résumé dans la figure 3.13.

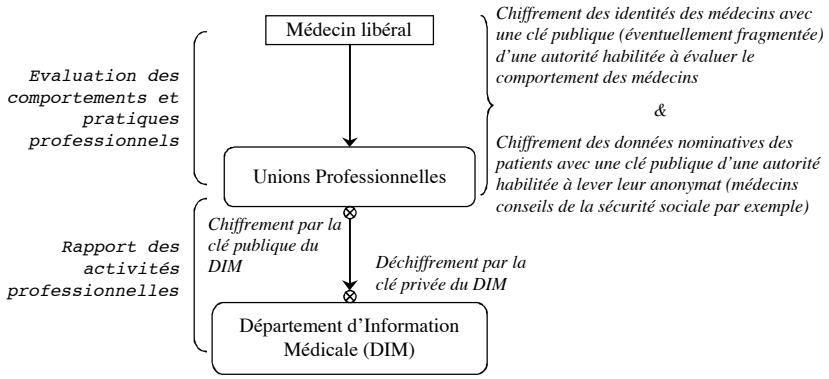


Figure 3.13 : Manipulations des identités au niveau des unions professionnelles.

3.2.3.7 Scénario 3 : Programme de Médicalisation des Systèmes d'Information "PMSI"

Le Programme de Médicalisation des Systèmes d'Information (PMSI) est un système d'analyse de l'activité des établissements de santé dont la finalité est l'allocation des ressources tout en diminuant les inégalités budgétaires. Le PMSI a été expérimenté depuis 1983, et généralisé dans les hôpitaux publics et privés participant au service public par la circulaire du 24 juillet 1989 [Circulaire 1989] pour l'activité de MCO (Médecine, Chirurgie, Obstétrique). Son utilisation à des fins budgétaires a été formalisée par la circulaire du 7 décembre 1996 [Ordonnance 1996]. Il a été étendu aux établissements privés par les ordonnances du 24 avril 1996. La circulaire du 9 mars 1998 [Circulaire 1998] a généralisé le PMSI aux établissements publics ayant une activité de Soins, de Suite et de Réadaptation. Une multitude de textes ont été élaborés pour réglementer le fonctionnement du PMSI. Citons à titre indicatif, la loi du 31 juillet 1991 [Loi 1991], le décret du 27/07/94 [Décret 1994] ainsi que les arrêtés des 20/09/1994, 22/07/1996 et 29/07/1998 [Arrêté 1998].

Dans la pratique, chaque séjour d'un patient donne lieu à un recueil standardisé de données de nature administrative (dates d'entrée et de sortie, date de naissance, nom et prénom, par exemple) et de nature médicale (diagnostics, actes codés). Les séjours sont ensuite classés selon l'indicateur médico-économique "Groupe Homogène de Malades" (GHM). Les patients d'un GHM donné sont considérés comme ayant mobilisé des ressources de même ampleur. Chaque année une échelle des coûts affecte un coût relatif à chaque GHM, mesuré en points ISA pour "Indice Synthétique d'Activité". Les données du PMSI des établissements publics sont anonymisées, puis transmises semestriellement aux Agences Régionales de l'Hospitalisation (ARH) qui les utilisent pour l'allocation budgétaire. Celles des établissements privés sont transmises trimestriellement à la CNAM-TS (Caisse Nationale de l'Assurance Maladie des Travailleurs Salariés), en attendant de devenir un outil d'allocation de ressources. Plus précisément, tout séjour hospitalier effectué dans la partie court séjour d'un établissement fait l'objet d'un Résumé de Sortie Standardisé (RSS), constitué d'un ou plusieurs Résumés d'Unité Médicale (RUM). Le RUM contient des données (administratives et médicales) concernant le séjour d'un patient dans une unité médicale donnée. À partir des RUM récupérés et validés, le Département d'Information Médicale (DIM) construit le fichier des Résumés de Sortie Standardisés (RSS) à l'aide d'un logiciel regroupeur. Les services des statistiques et des études épidémiologiques reçoivent du médecin responsable du Département d'Informations Médicales (DIM), les données médicales et administratives figurant sur les Résumés de Sortie Anonymisés (RSA). La procédure générale est donnée sur la figure 3.14, tandis que les détails du fonctionnement du PMSI sont donnés en annexe B.

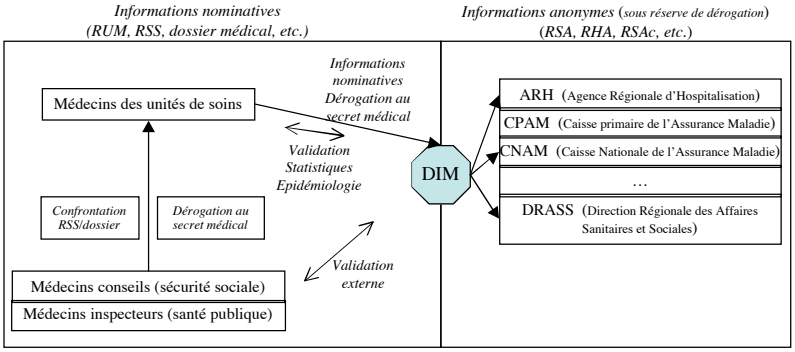


Figure 3.14 : Frontières des données nominatives, anonymes et anonymisables.

Étant donné que la finalité du PMSI est purement médico-économique (et non pas directement épidémiologique), le *besoin* est de pouvoir effectuer des trajectoires de soins par le biais d'une pseudonymisation ; l'*objectif* est une anonymisation irréversible ; et les *exigences* sont un chaînage universel (toujours et partout le même identifiant pour un patient donné) ainsi que la robustesse à la réversion et aux inférences (déductives, inductives, abductives, etc.).

3.2.3.8 Scénario 4 : traitement des maladies à déclaration obligatoire

Les maladies dont la surveillance est nécessaire à la conduite et à l'évaluation de la politique de santé publique (le SIDA, par exemple), ou qui nécessitent une intervention urgente locale (méningite, choléra, rage) sont des maladies à déclaration obligatoire. À l'origine, les fichiers des patients séropositifs sont nominatifs[23], mais ils sont anonymisés (*anonymisation irréversible*) avant toute transmission.

Les *besoins* sont divers : prévention, production de soins, veille sanitaire, analyses épidémiologiques, etc. L'objectif principal est l'irréversibilité de la fonction d'anonymisation. Le chaînage universel et la robustesse à la réversion et aux attaques par inférence constituent les principales exigences.

À cet égard, le type de protection doit dépendre des objectifs. En effet, s'agit-il d'obtenir, année par année, un état exhaustif du nombre de séropositifs pour connaître les tendances et l'évolution de l'épidémie, ou d'évaluer, de façon globale, l'impact des actions de prévention ? S'agit-il encore d'instituer une véritable surveillance épidémiologique de l'évolution des cas d'infection par le VIH, du stade de la découverte de la séropositivité, à l'apparition éventuelle du SIDA avéré ? Dans ce cas, l'objectif est de mesurer de façon fine l'impact des actions thérapeutiques et de prévention nécessitant un suivi des cas, en particulier un rapprochement avec le système de déclaration obligatoire des personnes séropositives.

Ce choix d'objectifs comporte des conséquences importantes tant sur la nature des données susceptibles d'être collectées que sur leur durée de conservation et les liens éventuels avec d'autres systèmes de surveillance. Il implique en conséquence des choix en terme de protection de données.

---

[23] Il existe des fichiers directement nominatifs, comportant des informations relatives à des personnes séropositives, qu'il s'agisse des dossiers médicaux tenus dans les services hospitaliers, les cabinets médicaux, ou les laboratoires, des fichiers de remboursement détenus par les organismes de sécurité sociale ou encore des fichiers constitués par les associations d'aides aux personnes séropositives. Notons toutefois qu'il existe des centres de dépistage anonymes dans lesquels il n'y a aucune possibilité de retrouver les identités des personnes séropositives.

Appliquer l'anonymisation à la source et disposer de mesures de sécurité adéquates ne dispensent pas de s'interroger sur la pertinence des autres informations appelées à figurer sur la déclaration de séropositivité. Il s'agit en particulier, du code postal de résidence, la profession et l'origine géographique.

- Le code postal de domicile : si l'objectif est de mieux cibler les actions de prévention locale, sa collecte semble nécessaire. Néanmoins, la pertinence du recueil de cette donnée n'est pas à ce jour réellement démontrée. En outre, sa collecte et son expiration pourraient être de nature à permettre une localisation géographique précise surtout dans les petites communes. Dès lors le recueil sous une forme aussi détaillée que le code postal du lieu de résidence des personnes séropositives peut paraître excessif au regard des objectifs recherchés et il est probablement plus judicieux d'utiliser le code du département au lieu du code postal.
- La profession : même si elle peut constituer une indication de la situation sociale des personnes séropositives afin de mieux cibler les actions de prévention, il ne paraît pas nécessaire de disposer du détail de la profession précise des personnes concernées. Dès lors, une simple mention des catégories socio-professionnelles selon la nomenclature de l'INSEE paraît être pertinente.
- L'origine géographique : il serait peut-être suffisant de mentionner si la personne est originaire d'un pays où la transmission hétérosexuelle est prédominante ou si elle a eu des relations sexuelles avec une personne originaire ou ayant vécu dans un pays où la contamination hétérosexuelle est prédominante.

Actuellement en France, il semble que le but est d'assurer un suivi des cas et de mesurer l'évolution du SIDA. Il est donc nécessaire de collecter des données individualisées afin de permettre de détecter les doublons, de vérifier les données auprès des professionnels de santé, etc. En ce qui concerne l'appauvrissement des données, l'Institut de Veille Sanitaire a montré que l'utilisation d'informations indirectement nominatives figurant sur les déclarations du SIDA (initiales du nom et prénom, date de naissance, département de domicile) permet de repérer plus de 99% des doublons (la suppression des initiales réduirait ce chiffre à 20% de doublons). Un appauvrissement trop important des données peut donc fausser les statistiques et remettre en cause la fiabilité scientifique de la surveillance épidémiologique.

3.2.3.9 Scénario 5 : traitements des données statistiques

En aucun cas, les données médicales à caractère personnel ne peuvent être manipulées pour des traitements à des fins non-épidémiologiques, en l'occurrence, des traitements purement statistiques ou à des fins de publications scientifiques. À cet égard, non seulement ces données doivent être anonymisées, mais il doit être impossible de les ré-identifier. Ainsi, s'imposent l'irréversibilité de l'anonymisation ainsi que la robustesse aux inférences. En effet, même après anonymisation, les identités peuvent être déduites par un statisticien en combinant plusieurs requêtes ou en complétant son raisonnement par des hypothèses ou par des informations externes au système. Pour illustrer ce type d'attaques, étudions un exemple extrait de [Cuppens 2003]. Soit une base de données relationnelles, interrogée sous SQL, et contenant la relation *Analyse*(*Patient*, *H/F*, *Age*, *Mutuelle*, *Leucocyte*). Pour un patient donné, "*Analyse*" fournit :

- des informations d'ordre administratif, à savoir s'il s'agit d'un homme ou d'une femme (attribut H/F), son âge (attribut Âge) et le nom de sa mutuelle (attribut Mutuelle),
- les résultats d'analyses médicales, à savoir l'attribut leucocyte, qui donne pour chaque patient son taux de leucocytes par « mm^3 » de sang. Pour simplifier la présentation de l'exemple, nous considérons seulement cet attribut leucocyte, mais la démarche que nous présentons s'appliquerait naturellement pour dériver d'autres résultats d'une analyse médicale (par exemple taux d'hémoglobine, plaquette, urée, sucre, etc.).

Le tableau 3.6 donne l'exemple d'instance de la relation Analyse que nous considérerons dans la suite.

Patient	H/F	Age	Mutuelle	Leucocyte
Dupont	H	30	MMA	6000
Durand	F	25	LMDE	3000
Dulac	F	35	MMA	7000
Duval	H	45	IPECA	5500
Dubois	H	55	MGEN	3500
Dumont	H	38	MMA	7500
Dupré	F	32	IPECA	7200
Dupuis	F	50	MGEN	6800
Dufour	H	45	MAAF	4000
Dumas	H	40	Rempart	3800

Tableau 3.6 : Instances de la relation Analyse.

Une base de données statistiques (dans le contexte de la sécurité) est une base de données qui permet d'évaluer des requêtes qui dérivent des informations d'agrégation (par exemple, les moyennes) mais pas des requêtes qui dérivent des informations particulières, en l'occurrence, des informations relatives à une personne donnée. Par exemple, si on considère la relation présentée dans le tableau 3.10, la requête "quelle est la moyenne du taux de leucocytes des patients ayant plus de 30 ans ?" va être permise alors que la requête "quel est le taux de leucocytes de Dupont ?" ne le sera pas.

Le problème associé à ce type de base de données réside en ce qu'il est souvent possible de faire des inférences à partir de requêtes autorisées pour déduire des réponses à des requêtes qui ne le seraient pas. Comme en fait état [Denning 1979] : « les statistiques contiennent des traces des informations initiales ; un espion sera capable de reconstruire ces informations par le traitement d'un nombre suffisant de statistiques ». C'est un cas particulier important de déduction d'informations sensibles par inférence.

Supposons, par exemple, qu'un utilisateur *U* soit autorisé à effectuer des requêtes statistiques (uniquement) et envisage de découvrir le taux de leucocyte de Dubois. Supposons également que *U* sache par ailleurs que Dubois est un adhérent masculin de la MGEN. Considérons maintenant les requêtes 1 et 2 suivantes :

1) SELECT COUNT (Patient)
   FROM Analyse
   WHERE H/F = 'H' AND Mutuelle = 'MGEN' ;

Résultat : 1.

2) SELECT SUM (Leucocyte)
   FROM Analyse
   WHERE H/F = 'H' AND Mutuelle = 'MGEN' ;

Résultat : 3500.

La sécurité de la base de données a été clairement compromise, même si l'utilisateur *U* a émis uniquement des requêtes statistiques autorisées. Comme le montre l'exemple, si l'utilisateur peut trouver une expression booléenne qui identifie un certain individu, alors les informations concernant cet individu ne sont plus sécurisées. Ce fait suggère que le système doit refuser de répondre à une requête pour laquelle la cardinalité de l'ensemble des statistiques est inférieure à une certaine borne *b*. De même, il suggère que le système devrait également refuser de répondre si cette cardinalité est supérieure à une borne $N - b$ (où N est la cardinalité

de la relation initiale) parce que la compromission ci-dessus peut également être obtenue à partir de la suite de requêtes 3-6 suivantes :

3) SELECT COUNT (Patient)
FROM Analyse
Résultat : 10.
4) SELECT COUNT (Patient)
FROM Analyse
WHERE NOT (H/F = 'H' AND Mutuelle = 'MGEN') ;
Résultat : 9 ; 10 - 9 = 1.
5) SELECT SUM (Leucocyte)
FROM Analyse
Résultat : 54300.
6) SELECT SUM (Leucocyte)
FROM Analyse
WHERE NOT (H/F = 'H' AND Mutuelle = 'MGEN') ;
Résultat : 50800 ; 54300 – 50800 = 3500.

Malheureusement, se contenter de restreindre les requêtes à celles pour lesquelles l'ensemble des statistiques possède une cardinalité c telle que $b \leq c \leq N - b$ n'est pas une solution pour éviter la compromission en général. D'autres exemples peuvent être trouvé dans [Cuppens 2004]. La référence [Denning 1979] montre que, pour pratiquement toutes les bases de données statistiques, un traceur global peut toujours être trouvé. Un traceur global est une expression booléenne qui peut être utilisée pour trouver la réponse à toute requête non-autorisée, c'est-à-dire une requête faisant intervenir une expression inadmissible.

Ainsi la sécurité dans les bases de données statistiques est un problème réel. Plusieurs suggestions apparaissent dans la littérature, mais il est difficile de décider si l'une d'entre elles est vraiment satisfaisante. Par exemple, une solution serait "la permutation des données", c'est-à-dire les valeurs des attributs des *n-uplets* sont permutées de sorte que la précision globale de la statistique est conservée. Mais même si une valeur particulière (par exemple, un taux de leucocytes particulier) est identifiée, il n'existe aucun moyen de savoir à quel individu appartient cette valeur. La difficulté inhérente à cette approche réside dans la recherche des ensembles d'entrées dont les valeurs peuvent être permutées de cette façon.

Une autre solution pourrait être le brouillage, qui consiste à modifier les réponses aux requêtes statistiques en y ajoutant du "bruit" aléatoire pour rendre plus difficile le recoupement entre requêtes.

3.2.3.10 Scénario 6 : études épidémiologiques focalisées

Le PMSI traite des informations médico-administratives, économiques et statistiques, afin de réaliser des analyses pertinentes des bases de données régionales et nationales (voir annexe B). Naturellement, les données traitées sont anonymes, et même si elles sont souvent chaînables, il n'y a généralement aucun moyen de lever l'anonymat. À l'inverse, dans d'autres types d'études, il est souvent souhaitable de revenir à l'identité réelle des patients afin d'améliorer la qualité des soins. Prenons à titre d'exemple, certaines études épidémiologiques focalisées : protocoles de recherche en cancer, maladies génétiques ou maladies rares (dites aussi orphelines).

Si, par exemple, ces études épidémiologiques mettent en évidence la situation suivante : les patients de la catégorie "C" ayant subi certains traitements "T_{avant}" ont une espérance de vie considérablement réduite s'ils ne suivent pas le traitement "$T_{recouvrement}$". Dans de telles

situations, il faudra remonter aux identités réelles pour que les patients puissent profiter de ces résultats. Il s'agit ainsi d'une anonymisation inversible, c'est-à-dire un chiffrement avec une clé (secrète), de telle sorte que seules des personnes habilitées à lever l'anonymat (*médecins conseils, médecins inspecteurs, médecins traitants*) détiennent la clé de déchiffrement, et seulement quand c'est nécessaire (figure 3.15).

Dans le cas des protocoles de recherche sur le cancer, le processus commence par un typage (stade de la maladie), puis par une identification du protocole correspondant au patient (s'il existe), enfin, selon le protocole, le patient est enregistré dans un registre régional, national, voire international. Les études épidémiologiques et statistiques faites sur ces registres peuvent dégager de nouveaux résultats concernant les patients d'un certain protocole. Dans le but de raffiner les études et faire avancer la recherche scientifique, il est parfois utile de remonter aux identités réelles des patients pour les identifier, faire des recoupements entre plusieurs données déjà recueillies, et les compléter *a posteriori*.

Dans le cas de maladies génétiques, les études se font sur des données médicales anonymisées. Toutefois, si de nouveaux résultats sont découverts, il est envisageable de remonter aux identités des patients, voire de leurs familles pour leur faire de nouveaux tests, ou de leur faire suivre de nouveaux traitements. Le principe est le même pour les maladies orphelines.

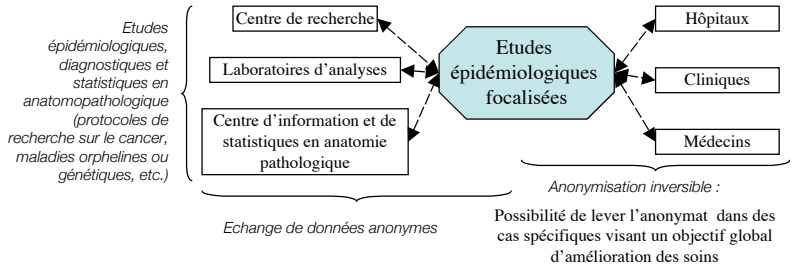


Figure 3.15 : Anonymisation dans le cadre des études épidémiologiques focalisées.

3.2.3.11 Une nouvelle solution générique

3.2.3.11.1 Schéma général

Les avantages et les faiblesses des différentes procédures d'anonymisation qui existent dans les pays européens ont été discutés. Par ailleurs, on a montré, à travers des scénarios, que toute anonymisation nécessite une étude préalable judicieuse, identifiant de manière claire et explicite les besoins, les objectifs ainsi que les exigences de sécurité à atteindre. À cet égard, la fonction d'anonymisation doit être adaptée aux différents composants de la démarche présentée. Cette analyse a permis d'aboutir à une solution qui peut aider les différents acteurs en informatique de santé à mieux définir les objectifs et exigences en ce qui concerne l'anonymisation des identifiants des personnes, et à proposer des solutions adaptées aux attentes. La figure 3.16 schématise notre proposition [Abou El Kalam *et al*. 2003c, Abou El Kalam *et al*. 2004a, Abou El Kalam *et al*. 2004b].

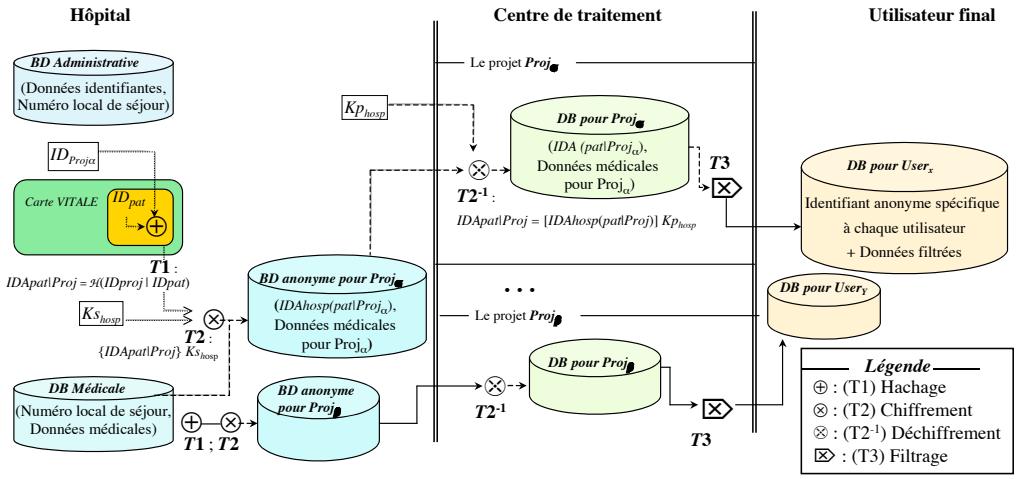


Figure 3.16 : Procédure d'anonymisation proposée.

Détaillons les traitements effectués ainsi que les vues disponibles au niveau des hôpitaux, des centres de traitements et des utilisateurs finaux.

3.2.3.11.2 Transformations au niveau des établissements de soins

À l'hôpital, trois types de bases de données peuvent être distinguées : une base de donnée administrative accessible par les personnels administratifs, chacun selon ses fonctions ; une base de donnée médicale dont l'accès est restreint aux personnels soignants (en charge des patients) ; ainsi que des bases de données anonymes, chacune contient les informations nécessaires et suffisantes pour un projet donné. Un projet désigne une entité de traitement des données anonymes tel le PMSI, le DIAM (Dispositif Informationnel de l'Assurance Maladie), les associations de personnes diabétiques, les centres des études cliniques, etc.

Le passage de la base de données médicale à une base anonyme (destinée à un certain projet) nécessite l'application de deux transformations, T1 et T2, aux données à transférer.

La transformation **T1** : consiste à obtenir "*IDApat|Proj*", un identifiant anonyme par personne et par projet, à partir des deux identifiants :

- "*IDproj*", l'identifiant du projet, qui est détenu par les établissements de soins (hôpitaux, cliniques) ;
- "*IDpat*", l'identifiant anonyme unique et individuel du patient, nous suggérons que cet identifiant soit détenu par le patient sur la carte Vitale[24] ; une longueur de 128 bits nous paraît suffisante pour éviter des collisions (risque que deux personnes différentes aient le même identifiant).

Au niveau de l'hôpital, et lors de l'alimentation des bases de données anonymes (par projet), l'utilisateur (employé de l'hôpital par exemple) envoie *IDproj* (l'identifiant du projet concerné

[24] Même si la carte Vitale appartient à l'assuré et donc peut correspondre à plusieurs personnes (l'assuré et ses ayants-droits), l'identifiant ID_{pat} est individuel. Dès lors, sur une même carte peuvent figurer plusieurs identifiants correspondants à l'assuré et aux ayants-droit de moins de seize ans. Ceci sous entend un dispositif de classification de ces identifiants (par numéro d'ordre par exemple).

par la base de donnée) à la carte ; celle-ci contient déjà *IDpat* (l'identité du patient donnant son consentement pour l'exploitation de ses données médicales par le projet). La procédure T1 consiste à appliquer une fonction de hachage (MD5 ou SHA par exemple) à (*IDproj* | *IDpat*), la concaténation de *IDproj*) et *IDpat* :

(T1) $IDApat|Proj = \mathcal{H}(IDproj \mid IDpat)$

La transformation T1, réalisée au sein de la carte Vitale du patient, et produisant l'empreinte $\mathcal{H}$(IDproj | IDpat), vise les objectifs suivants :

- un patient n'apparaît dans une base de donnée anonyme que si cela est obligatoire (par exemple pour le PMSI) ou s'il donne son consentement à travers la fourniture de son identifiant (pour une étude de nature médico-commerciale, par exemple) ;
- l'identifiant anonyme *IDApat|Proj* n'utilise aucun secret dont la divulgation porterait atteinte à la vie privée des autres personnes (contrairement à l'utilisation d'une clé secrète commune pour tous les patients). De plus, puisque le calcul de l'empreinte IDApat|Proj s'effectue au niveau de la carte, *IDpat* reste toujours au sein de la carte ; il n'est jamais stocké isolement ; et il n'est utilisé que pour créer une entrée dans la base anonyme pour un projet donné (au niveau de l'hôpital) ;
- puisque *IDproj* est spécifique à chaque projet, les risques de rapprochements non-autorisés des données de deux projets différents sont écartés, ou du moins sont peu vraisemblables ; de plus, les bases de données anonymes (par projet) sont isolées de l'extérieur de l'hôpital et sont soumises à des mesures strictes de contrôle d'accès ;
- il est possible que chaque projet puisse faire des rapprochements de données concernant un même patient (car pour un patient donné et un projet donné, l'empreinte *IDApat|Proj* est toujours la même).

Néanmoins, la transformation T1 ne permet pas de se prémunir contre certaines attaques où les intrus essayent de faire des rapprochements d'informations (concernant un projet donné) détenus par deux hôpitaux différents. En effet, supposant que le patient Paul a été traité à Rangueil et à Purpan, et que dans chacun de ces deux hôpitaux, Bob est consentant de l'utilisation de ses données pour un projet "Proj". Supposant qu'un employé de Purpan, nommé Bob, sait que l'empreinte X (=*IDAPaul|Proj*) correspond à Paul. Supposant en plus, que Bob arrive à s'emparer de la base de donnée anonyme, concernant le projet "Proj", et détenue par Rangueil. Dans ce cas, l'utilisateur malveillant Bob peut facilement établir le lien entre le patient Paul et ses données médicales (concernant le projet "Proj") détenues par Rangueil (mais aussi celles détenues par Purpan, puisque Bob travaille à Purpan).

Afin de contrer ce type d'attaques, nous introduisant la *transformation asymétrique* **T2** au niveau de l'hôpital. Ainsi, avant de stocker les données dans les bases de données anonymes spécifiques à chaque projet, l'hôpital chiffre (chiffrement asymétrique) l'identifiant *IDApat|Proj* avec une clé $Ks_{hôp}$ spécifique à l'hôpital ; ("{}K" désigne un chiffrement avec K) :

(T2) $IDAhôp(pat|Proj) = \{IDApat|Proj\}\ Ks_{hôp}$

Si on reprend le scénario précédent, l'utilisateur malveillant Bob ne peut revenir aux identités des personnes car il ne dispose pas de la clé de déchiffrement Kp_{Purpan}. En effet, chaque hôpital détient sa clé $Ks_{hôp}$, tandis que $Kp_{hôp}$ n'est détenue que par les projets.

Les deux transformations (T1 et T2), effectuées au niveau des hôpitaux, permettent d'avoir une grande robustesse vis-à-vis d'attaques ayant pour but de lever l'anonymat (ou de faire des rapprochements) de façon non autorisée. Pour autant, la procédure proposée reste assez flexible. En effet, si deux hôpitaux (*hôpa* et *hôpb*) décident de fusionner un jour, il est tout à fait possible de relier les données concernant chaque patient ; que ces données proviennent de *hôpa* ou de *hôpb*.

Il suffit que chaque hôpital déchiffre ses données avec sa clé "$Kp_{hôp}$", puis chiffre le résultat avec la clé privée $Ks_{hôp_{ab}}$ du nouvel hôpital. Si $IDA_{hôp_a}(pat|Proj)$ (respectivement $IDA_{hôp_b}(pat|Proj)$) désigne un identifiant anonyme au sein de l'hôpital $hôp_a$ (respectivement $hôp_b$) ; "[]K" désignant le déchiffrement avec la clé K :

- Le traitement effectué sur les anciennes données de l'hôpital $hôp_a$ est :
$$\{ [IDA_{hôp_a}(pat|Proj)] \, Kp_{hôp_a} \} \, Ks_{hôp_{ab}} ;$$
- Le traitement effectué sur les anciennes données de l'hôpital $hôp_b$ est,
$$\{ [IDA_{hôp_b}(pat|Proj)] \, Kp_{hôp_b} \} \, Ks_{hôp_{ab}} ;$$

Ainsi, les codes de liaisons obtenus seront les mêmes dans les deux établissements (pour chaque base de donnée anonyme associé à un certain projet).

Pour les utilisateurs internes aux établissements de soins, les mécanismes de contrôles d'accès doivent interdire tout accès non-autorisé, tandis que des mécanismes de détection et de tolérance aux intrusions doivent renforcer les autres mesures de sécurité.

3.2.3.11.3 Transformations au niveau des centres de traitements

Les données contenues dans les bases de données anonymes (au niveau des établissements) subissent des transformations qui dépendent de l'identifiant anonyme $IDA_{proj|pat}$ et de la clé $Ks_{hôp}$. Pour retrouver les données qui lui sont destinées, chaque centre de traitement (correspondant à un projet) déchiffre les données qui lui sont envoyées par la clé $Kp_{hôp}$ de l'hôpital transmetteur :

$$\begin{aligned} & [IDAhôp(pat|Proj)] \, Kp_{hôp} \\ \text{et d'après } \mathbf{(T2)}, \quad & = [\{IDA_{pat|Proj}\} \, Ks_{hôp}] \, Kp_{hôp} \\ & = IDA_{pat|Proj} \end{aligned}$$

Le centre de traitement retrouve ainsi les informations suffisantes et nécessaires aux traitements qu'il effectue. Ces informations sont associées aux identifiants anonymes $IDA_{pat|Proj}$, ce qui permet à chaque projet de chaîner les données de chaque patient.

Par ailleurs, avant leur distribution aux utilisateurs finaux (recherche scientifique ciblées, publications, Web, presse), et afin de respecter le plus possible le principe du moindre privilège, les informations transférées peuvent éventuellement subir un traitement de filtrage ciblé pour chaque catégorie d'utilisateurs. Il peut, par exemple, s'agir d'une agrégation, d'un appauvrissement des données, etc.

Si de plus, l'objectif de sécurité est d'interdire à deux (ou plusieurs) utilisateurs finaux de recouper les informations, il convient d'appliquer une autre anonymisation (hachage, avec MD5 par exemple) avec une clé secrète $K_{util|proj}$.

$$IDA_{pat|util} = \mathcal{H}(IDA_{pat|Proj} \mid K_{util|proj})$$

Dans ce cas, si le but est de permettre à l'utilisateur de faire des chaînages dans le temps (par projet), la clé $K_{util|proj}$ doit être stockée au niveau du centre de traitement, de façon à pouvoir la réutiliser, à chaque fois que celui-ci souhaite transmettre d'autres informations à cet utilisateur. À l'inverse, si le centre souhaite empêcher le chaînage dans le temps par les utilisateurs, la clé est générée aléatoirement à chaque distribution.

3.2.3.11.4 Discussion

La solution que nous proposons garantit les points suivants :

- le patient doit donner explicitement son consentement pour toute utilisation non-obligatoire, mais souhaitable, de ses données ;
- les clés et identifiants utilisés dans les diverses transformations (ID_{proj}, ID_{pat}, $Ks_{hôp}$, $Kp_{hôp}$, $IDA_{pat|Proj}$ et $IDA_{pat|util}$) sont détenus par des personnes différentes, et dans des endroits différents : ID_{proj} ne concerne qu'un projet parmi d'autres ; ID_{pat} est spécifique à un patient, et cette information n'est connue que par lui ; $Ks_{hôp}$ est spécifique à l'hôpital ; et $IDA_{pat|util}$ n'est destinée qu'à un utilisateur (ou type d'utilisateur) d'un projet donné. Il est donc pratiquement impossible de pouvoir faire des désanonymisations non autorisées ;
- la solution résiste aux attaques par dictionnaire et à tous les niveaux : établissements de soins, centres de traitements, et utilisateurs finals ;
- la séquence d'anonymisation (anonymisation en cascade) que nous proposons à différents niveaux, combinée avec des mécanismes de contrôles d'accès, permet de garantir, en toute robustesse, l'exigence de non inversibilité ;
- les identifiants anonymes générés étant spécifiques à un secteur particulier (projet, domaine d'activité, centre d'intérêt, branche professionnelle, établissement, etc.), il est possible d'adapter la solution à chaque secteur (par exemple lorsque le centre de traitement est le seul utilisateur) ;
- il est possible de fusionner les données de deux (voire de plusieurs) établissements sans compromettre la flexibilité et la sécurité ;
- la manière selon laquelle l'information est distribuée et utilisée par l'utilisateur final est importante. Notre solution peut être adoptée pour tenir compte de la finalité du traitement ;
- si un utilisateur final (chercheur dans le domaine des maladies orphelines par exemple) découvre une information qui nécessiterait de remonter aux identités des patients, il doit renvoyer ses résultats à l'hôpital qui, lui seul (par l'intermédiaire du professionnel soignant par exemple), peut établir le lien entre les variables identifiantes, les numéros locaux de séjours, et les données médicales de ses patients.

Par ailleurs, nous pensons que la solution idéale n'existe pas, et nous suggérons de compléter notre solution, selon le cas étudié, par une combinaison de solutions techniques et organisationnelles :

- l'accès aux données doit être parfaitement contrôlé. Une *politique de contrôle d'accès* doit être définie et mise en place pour que les données ne soient accessibles qu'aux seuls utilisateurs habilités ;
- la spécification du système d'information et de l'architecture du réseau doit obéir à une politique globale de sécurité, et donc doit être adaptée aux besoins ;
- La définition de la politique de sécurité doit inclure une analyse approfondie des risques d'abduction ;
- la constitution de sous-bases de données régionales ou thématiques doit être contrôlée.
- Il convient d'utiliser (si cela est possible) des anonymisations thématiques, de sorte que même si un utilisateur parvient à casser l'anonymisation, les risques d'abduction soient limités à un thème donné ;
- Il faut séparer les données d'identité des renseignements proprement médicaux. Bien entendu ce mécanisme ne peut être appliqué que dans des contextes particuliers. Un exemple est donné dans l'annexe B ;

- Nous préconisons l'utilisation de solutions complémentaires comme l'*appauvrissement* des données ;
- Il faut surveiller les utilisations qui sont faites des données, en définissant et en mettant en œuvre des outils de *détection d'intrusion* ; en particulier, ces outils doivent permettre de détecter les requêtes, voire les enchaînements de requêtes, ayant un but malveillant (inférence de données, abus de pouvoir, *etc*.) ;
- nous préconisons également l'utilisation d'autres techniques comme le brouillage ou le filtrage, de façon à ne pas répondre à des requêtes statistiques si l'information demandée est trop précise ; etc.

Dans cette section, nous avons analysé le problème d'anonymisation dans le domaine médical et nous avons proposé des solutions flexibles et adaptées aux besoins, objectifs et exigences de sécurité de ce domaine. Compte tenu de la ressemblance entre la nature, la structure, et le fonctionnement des domaines médical et social, nous pouvons constater que cette démarche est tout à fait applicable dans la sphère sociale.

Par ailleurs, les données anonymisées (ou anonymes) doivent être recensées comme sensibles et donc doivent être intégrés dans la description de la politique de sécurité. Ainsi, avec des données nominatives ou non, on revient au problème global qui est la définition et la spécification de la politique de sécurité.

Rappelons que dans la première section de ce chapitre nous suggérons une méthodologie pour établir des politiques de sécurité. La première étape consiste à obtenir certains éléments de description structurels (comme les rôles) ou fonctionnels (comme les processus), à identifier de manière explicite les informations à protéger (anonymes ou pas), et à caractériser les menaces (altération, inférence, etc.). La deuxième étape s'appuie sur ces descriptions pour formuler les objectifs de sécurité en terme de confidentialité, d'intégrité et de disponibilité des ressources et services de l'organisation. Un règlement de sécurité vient ensuite décrire comment l'organisation fonctionne tout en satisfaisant les objectifs de sécurité tracés.

Chapitre 4. Le modèle Or-BAC

Préambule

Dans le chapitre précédent, nous avons proposé une méthode permettant de définir une politique de sécurité pour des organisations complexes en général et pour les SICSS en particulier. Nous avons appliqué notre méthodologie à différents scénarios des domaines social et médical. Nous avons également exprimé le besoin de modéliser la politique de sécurité, notamment pour contribuer au processus de preuve et de vérification, nécessaire pour avoir une confiance élevée dans le système.

Après avoir rappelé les lacunes des politiques et modèles de sécurité existants, ce chapitre commence par présenter les concepts nécessaires pour exprimer des politiques de sécurité pour des systèmes complexes, coopératifs et distribués comme les SICSS. Le but étant de :

- tenir compte du contexte et de l'interopérabilité,
- prendre en compte toute amélioration, changement ou mise à jour des éléments en relation avec la sécurité,
- réaliser un bon compromis entre le respect du principe du moindre privilège et la flexibilité du contrôle d'accès, de façon à ne pas gêner le travail du personnel, tout en respectant les droits des usagers.

Ensuite, nous imbriquons ces concepts pour construire le modèle Or-BAC, dont une représentation entité-relation a été développée dans le cadre du projet MP6 [Abou El Kalam *et al*. 2003a]. Le concept d'organisation est central dans Or-BAC. Celui-ci n'utilise que des entités abstraites pour exprimer la politique de sécurité, sans se soucier la manière selon laquelle les différentes sous-organisations implémentent et instancient ces entités (rôles, groupes, activités, contexte). Cette façon de procéder réduit considérablement la complexité de la spécification de la politique de sécurité, offre plus de flexibilité et facilite l'interopérabilité des composants du système ou de l'organisation, sans pour autant compromettre la sécurité. Par ailleurs, Or-BAC n'est pas restreint aux permissions, mais permet également de définir des interdictions, des obligations et des recommandations.

La deuxième partie de ce chapitre présente le modèle Or-BAC en utilisant une notation UML [Abou El Kalam & Deswarte 2004].

4.1. Motivation

Les politiques et modèles de sécurité qui existent dans la littérature (et que nous avons décrits dans le chapitre précèdent) ne prennent pas en compte les points suivants :

- des règles qui spécifient des permissions ou des interdictions contextuelles ; pourtant, dans le domaine médical, les professionnels de santé ont des permissions spéciales dans des contextes spécifiques comme l'urgence ; de même, les personnes autorisées peuvent, dans certains cas (échéance par exemple), forcer l'accès à Net-entreprises, pour préparer ou valider les déclarations, ou pour effectuer des télépaiements ;
- des règles qui spécifient des obligations ou des recommandations : les modèles et politiques de contrôle d'accès classiques sont généralement limités aux permissions ; certains incluent des interdictions, et plus récemment quelques modèles de politique de sécurité ont ajouté des obligations [Bettini *et al*. 2002, Damianou *et al*. 2001] ; dans les SICSS, il convient aussi de prendre en compte des recommandations ;
- des règles spécifiques à l'organisation ; en particulier, l'organisation peut être structurée en plusieurs sous-organisations, qui ont chacune leur propre politique de sécurité : le système de santé est organisé en plusieurs domaines (médical, paramédical, recherche, etc.), le domaine médical regroupe les hôpitaux, les cliniques et les médecins libéraux, chaque hôpital est organisé en services, etc. ; la politique de sécurité devra ainsi proposer un cadre homogène, permettant de spécifier plusieurs politiques de sécurité au sein d'une même organisation ; il est évident que chaque structure de la sphère santé-social doit implémenter sa propre politique interne de sécurité tout en respectant l'ensemble des contraintes imposées par la politique globale de sécurité.

La section suivante présente un ensemble de concepts permettant de spécifier une politique de sécurité capable de prendre en compte ces différents points, et supporter ainsi la richesse, la complexité et l'hétérogénéité des SICSS.

En outre, nous tenons à préciser que ce mémoire ne décrit pas la version initiale du modèle Or-BAC (représentation entité-relation, et logique du premier ordre), à laquelle nous avons contribué dans le cadre du projet MP6 [Abou El Kalam *et al.* 2003a ; Abou El Kalam *et al.* 2003b]. En revanche une version qui utilise UML et la logique déontique [Abou El Kalam & Deswarte 2004] sera présentée ici. Celle-ci nous semble plus détaillée et plus riche en expressivité.

4.2. Concepts de base du modèle Or-BAC

4.2.1. Organisations

L'organisation est l'entité centrale des politiques et modèles de sécurité que nous proposons. Dans le domaine médical, nous pouvons considérer les organisations : "clinique du Languedoc", "équipe de l'unité des soins intensifs de l'hôpital Rangueil", etc. De la même manière, dans le secteur social, les établissements, les entreprises ainsi que les organismes de protection sociale, sont des exemples d'organisations. Une organisation peut être définie comme une entité ayant un rôle professionnel ou statutaire bien défini, ou encore, un groupe structuré d'entités actives, c'est-à-dire de sujets (utilisateurs, équipes, ou autres) jouant certains rôles (figure 4.1).

Il est important de noter qu'un groupe quelconque de sujets n'est pas nécessairement considéré comme une organisation. Autrement dit, le fait que chaque sujet joue un rôle dans l'organisation correspond à un certain accord entre les sujets pour former une organisation.

La figure 4.1 exprime que les utilisateurs et les organisations sont considérés comme des *sujets* (entités actives), et qu'ils peuvent à ce titre, jouer des rôles. La figure 4.2 donne l'exemple d'un utilisateur Jean qui joue le rôle "cardiologue", et de l'organisation US21 qui joue le rôle "unité de soins intensifs". De la même manière, dans le cadre de Net-entreprises, les rôles "personne inscrite", "personne non-inscrite", "administrateur", "mandataire social", "dirigeant" et "personne autorisée", sont joués par des utilisateurs, tandis que les rôles "établissement déclaré", "tiers déclarant" sont joués par des organisations.

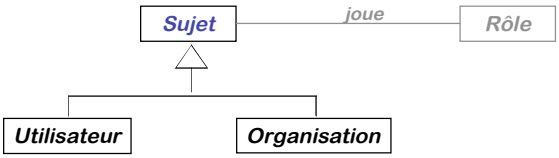


Figure 4.1 : Relation d'héritage entre les organisations et les sujets.

Figure 4.2 : Ébauche d'un diagramme d'objets représentant les rôles joués par les sujets.

4.2.2. Rôle dans Organisation (RdO)

Le contrôle d'accès basé sur les équipes, C-TMAC (section 2.7, page 55), considère deux relations binaires (utilisateur, rôle) et (utilisateur, équipe). Un utilisateur peut donc activer n'importe lequel de ses rôles dans n'importe laquelle de ses équipes. Dans la pratique, même si un utilisateur a le droit de jouer plusieurs rôles, il n'a pas forcément le droit de les jouer dans n'importe laquelle de ces équipes. Le modèle Or-BAC traite ce problème en ajoutant la notion de "*rôle dans organisation*" notée *RdO*.

La figure 4.3 illustre l'exemple d'un utilisateur Pierre qui joue les RdO "infirmier à l'hôpital de Purpan", "radiologue assistant à l'hôpital de Rangueil", mais pas forcément "radiologue assistant dans Purpan", ni "infirmier dans Rangueil". Selon une modélisation C-TMAC, ces rôles pourraient être autorisés, même s'ils ne sont pas conformes à la réalité.

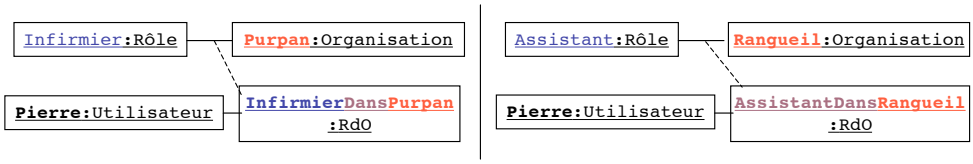


Figure 4.3 : Ébauches de diagrammes d'objets représentant des instances de RdO.

Nous avons fait le choix de représenter l'entité RdO par une classe-association. En effet, d'une part, un RdO est une association entre le rôle et l'organisation, et d'autre part, il a des attributs, notamment les identifiants du rôle et de l'organisation. Un RdO possède les caractéristiques d'une classe et d'une association, et peut à ce titre, participer à d'autres relations, en l'occurrence la relation qui associe les utilisateurs aux RdO (figure 4.4).

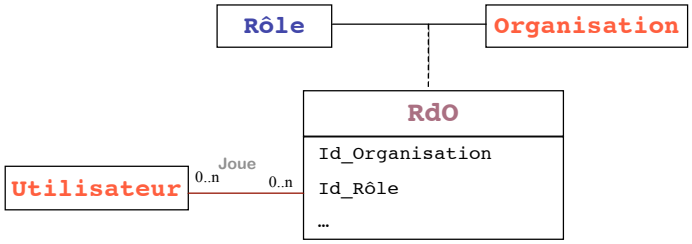


Figure 4.4 : Ébauche du diagramme de classe représentant la classe association RdO.

4.2.3. Vue dans Organisation (VdO)

Dans le modèle Or-BAC, les *objets* représentent des entités non actives comme les fichiers, les formulaires imprimés, etc. Dans le domaine médical, nous avons ainsi à considérer des objets comme les dossiers administratifs ou les dossiers médicaux des patients.

Nous avons déjà expliqué que les rôles sont des entités abstraites qui permettent de structurer les sujets et de faciliter la mise à jour de la politique de sécurité quand un nouvel utilisateur est ajouté. Dans la mesure où il est également nécessaire de structurer les objets et d'ajouter de nouveaux objets au système, nous considérons qu'une entité (abstraite) comparable au rôle pour les sujets est nécessaire pour les objets. Nous l'appelons v*ue*.

Ainsi, de la même manière que les sujets ayant une fonction commune sont représentés dans la politique de sécurité par des rôles, *les objets qui satisfont une propriété commune sont spécifiés à travers des vues*. Les vues sont donc pour les objets ce que sont les rôles pour les sujets (figure 4.5).

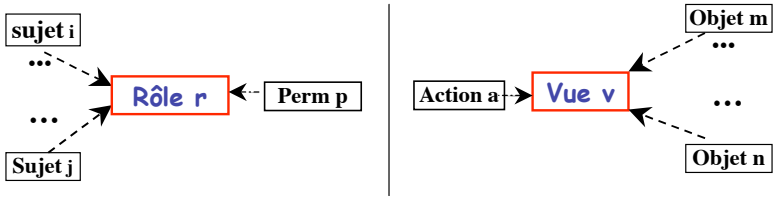


Figure 4.5 : Similitudes entre les rôles et les vues.

Outre sa nature intuitive extraite du fonctionnement normal des organisations, la notion de vue offre plusieurs autres avantages :

- Elle *facilite la structuration* et *permet d'abstraire les objets* dans la spécification de la politique de sécurité.
- Elle *aide à réduire les erreurs d'administration*. En effet, des entités comme les rôles, et les vues, ainsi que des associations comme (*action*, *vue*) et (*rôle, permission*) restent relativement fixes dans le système d'information ; elles sont donc gérées par l'administrateur. En revanche, certaines associations comme (*objet, vue*), par exemple, l'affectation de nouveaux dossiers aux vues correspondantes, changent souvent ; elle peuvent ainsi être gérées localement (par le personnel d'accueil de l'hôpital par exemple).
- Elle *aide à réduire les coûts d'administration*. Le coût des associations (*action, objet*) (sans passer par la vue) est de l'ordre de N_A*N_O, où N_A est le nombre d'actions et N_O est le nombre d'objets. Alors qu'avec la notion de vue, le coût est N_A+N_O, et ceci pour chaque vue.

Dans la mesure où les *vues* caractérisent la manière dont les *objets* sont utilisés dans l'*organisation*, nous introduisons la classe-association "*Vue dans Organisation*", notée *VdO*, et nous associons les objets aux VdO (figure 4.6).

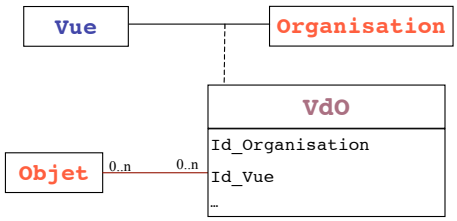


Figure 4.6 : Ébauche du diagramme de classe représentant la classe association VdO.

Cette manière de structurer les organisations, les objets et les vues permet d'exprimer *qu'une même vue peut être définie différemment suivant l'organisation considérée.* Le but est de permettre à des organisations de donner des définitions différentes à une même vue. Supposons à titre d'exemple que l'hôpital de Purpan utilise un système de fichiers, et que l'hôpital de Rangueil utilise une base de données ; la vue "dossier médical" peut être définie à Purpan comme un ensemble de documents textes, tandis qu'à Rangueil, cette même vue correspond à des attributs ou à des tables de la base. Le premier diagramme d'objets de la figure 4.7 exprime que "l'hôpital Purpan utilise le fichier F_{31}.txt comme un dossier médical", tandis que le deuxième diagramme illustre que "l'hôpital Rangueil utilise la table dossier médical comme un dossier médical".

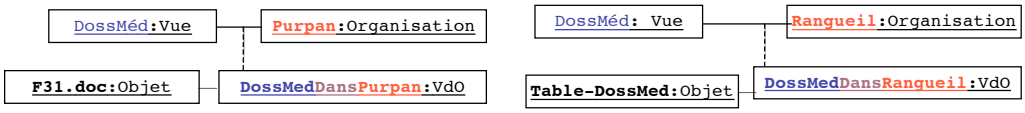


Figure 4.7 : Ébauches de diagrammes d'objets représentant des instances de VdO.

4.2.4. Activité dans Organisation (AdO)

Les politiques de sécurité spécifient les accès aux entités passives accordés aux entités actives et régulent les actions opérées sur le système. Dans notre politique, l'entité *Action* englobe principalement les actions informatiques comme lire, écrire, etc. De la même manière que les rôles et les vues sont des abstractions des sujets et des objets, nous définissons une nouvelle entité utilisée comme abstraction des actions : l'entité *Activité*. Les activités pourront être "lecture", "modification", etc. Ainsi, les rôles associent des sujets qui remplissent les mêmes fonctions, les vues regroupent des objets qui satisfont une propriété commune, et par analogie, les activités correspondent à des actions qui ont un objectif commun.

Par ailleurs, de la même manière que nous avons présenté les notions de RdO et VdO, nous introduisons l'entité "*activité dans organisation*" (notée *AdO*) comme classes-associations entre les activités et les organisations (figure 4.8).

Remarquons que là encore, l'objectif est de permettre à des organisations de structurer différemment les mêmes activités. Si nous considérons l'activité "lecture", celle-ci peut correspondre, dans l'organisation Purpan, à l'action "lire" un fichier, mais peut tout aussi bien correspondre à l'action "select" dans Rangueil (figure 4.9).

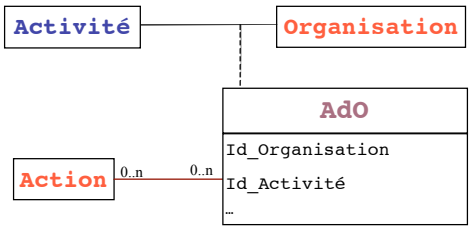


Figure 4.8 : Ébauche du diagramme de classe représentant la classe association AdO.

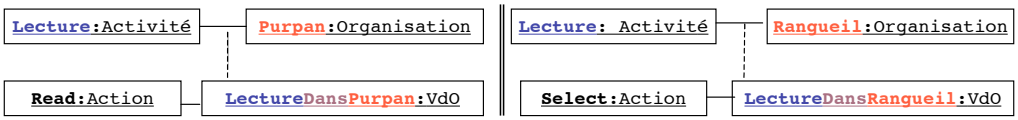


Figure 4.9 : Ébauches de diagrammes d'objets représentant des instances d'AdO.

Soulignons que le concepteur de la politique de sécurité définit les règles avec des entités abstraites (rôles, vues et activités), sans se soucier de la manière selon laquelle les organisations du domaine implémentent ou considèrent les instanciations de ces entités.

4.2.5. Le contexte

D'une manière générale, le contexte peut être défini comme toute information qui caractérise la situation d'une entité ou qui spécifie les circonstances concrètes dans lesquelles les organisations accordent des permissions à des rôles pour réaliser des activités sur des vues. En l'occurrence : qui peut déléguer ? Quand l'utilisateur a-t-il le droit d'accéder à une information ? D'où l'accès est-il possible ? Comment, où et pour quelles raisons, les informations sont-elles disponibles ?

Le contexte est ainsi l'une des notions utilisées par la politique présentée pour mieux respecter le principe du moindre privilège[25]. Dans le domaine médical, le *contexte* permettra d'exprimer des notions comme : l'urgence, les processus habituels, l'exclusion mutuelle, les attributs temporels, etc. En effet, le contexte peut être vu selon différentes facettes, selon qu'il est relié aux sujets, aux objets ou à l'utilisation elle-même. Une étude détaillée concernant l'utilisation du contexte dans les applications médicales peut être trouvée dans [Abou el Kalam & Deswarte 2002] ainsi que dans la norme européenne [CEN 1999].

4.2.5.1 Contexte et contraintes du rôle

En général, le contexte du rôle précise des valeurs que doivent prendre certaines variables contextuelles pour autoriser, interdire, obliger ou recommander à un utilisateur de jouer un rôle donné. Par exemple :

- *instant d'accès* : le rôle médecin de salle est valide pendant les heures normales de travail tandis que le rôle médecin de garde est valide la nuit ;
- *cardinalité* : c'est une contrainte sur le nombre maximal ou minimal d'utilisateurs autorisés à jouer un certain rôle, que ce soit directement ou indirectement (à travers l'héritage) ;

[25] Ce principe consiste à ne donner accès aux utilisateurs autorisés qu'aux seules ressources dont ils ont besoin pour accomplir leurs tâches, et ce, seulement pendant la durée de ces tâches.

- *exclusion mutuelle* : cette contrainte peut être *statique* ou *dynamique* ; une exclusion mutuelle statique entre deux rôles signifie qu'un utilisateur ne peut jamais jouer ces deux rôles, par exemple, dans le même établissement, être personnel soignant et comptable ; une exclusion mutuelle dynamique entre deux rôles signifie que l'utilisateur ne peut pas jouer les deux rôles simultanément, par exemple, médecin à l'hôpital et médecin travaillant pour une société d'assurance.

4.2.5.2 Contexte d'objet

Comme les rôles, les objets (ou d'une manière générale les vues) ont des attributs contextuels spécifiques. Par exemple :

- *la durée de conservation des données* : données de neurologie (70 ans), maladies héréditaires (illimitée), etc. ;
- *le lieu* : les dossiers de spécialités de chacune des unités sont situés et gérés localement dans les ordinateurs de cette unité ; dans certains cas, l'accès à ce type de dossier ne peut se faire que localement ; de manière générale, certaines organisations des SICSS associent à leurs objets (données, ressources) les emplacements où ils sont situés et gérés.

4.2.5.3 Attributs d'utilisateurs

Dans la pratique, certains attributs des utilisateurs peuvent être utilisés pour obliger ou recommander l'exécution d'une certaine action, interdire ou accorder des autorisations spécifiques ou des droits temporaires. Nous pouvons citer par exemple :

- l'affiliation à un corps de santé régional, national, ou international ;
- l'expérience dans la pratique de certains types particuliers de soins comme l'acupuncture ; etc.

4.2.5.4 Contexte de l'utilisation

Une entité peut être concrète ou abstraite. L'ensemble des entités abstraites de la politique proposée peut lui-même être partitionné en deux niveaux logiques : un premier niveau contenant les rôles, les vues et les activités, et un deuxième niveau schématisant les *coalitions* (ou les collaborations). Les organisations en font partie. En réalité, les organisations (les équipes de soins, par exemple) collaborent dans des processus spécifiques.

Le contexte d'utilisation est un concept innovant qui utilise la notion de *processus* pour gérer les accès normaux, et la notion d'*objectif d'utilisation* pour supporter les exceptions, avec plus de responsabilité. Le but est de réaliser un bon compromis entre le respect du *principe du moindre privilège* et la *flexibilité* du contrôle d'accès. Tout accès doit donc être inscrit soit dans un processus, soit dans une exception déclarant un objectif spécifique de l'utilisation prétendue. Les autorisations finales seront ainsi différentes selon que l'utilisateur aura déclaré tel ou tel objectif d'utilisation.

4.2.5.4.1 Processus

Dans le cas d'un processus de soins, le consentement (explicite ou implicite) du patient est recueilli, et l'activité de soins est enregistrée dans le système par une personne habilitée par exemple, le médecin traitant. Le processus est ainsi identifié par un patient, un motif et des organisations qui collaborent pour traiter le patient.

En outre, dans le domaine médical, différents protocoles de processus (processus-types) peuvent être énumérés. Il suffit qu'une personne habilitée instancie un de ces processus, et les

accès s'inscriront dans ce cadre. À titre d'exemple, le processus "demande d'avis neurochirurgical en urgence" peut être constitué des étapes suivantes :

- arrivée du patient aux urgences où il est accueilli par l'infirmière ;
- examen clinique par le médecin urgentiste ;
- réalisation d'un scanner par un radiologue et un manipulateur radio ;
- réalisation des examens de biologie ;
- prise en compte des résultats du scanner et demande d'avis neurochirurgical ;
- le radiologue donne son avis à l'urgentiste ;
- le biologiste donne son avis sur les examens de biologie ;
- le neurochirurgien de l'hôpital distant donne son avis à l'urgentiste ;
- le neurochirurgien de l'hôpital distant accepte le transfert du patient ;
- sortie administrative du patient ;
- préparation de l'accueil du patient dans le service de neurochirurgie ;
- accueil du patient dans le service de neurochirurgie ;
- visite au patient et organisation de la prise en charge ;
- intervention chirurgicale en neurochirurgie.

Par analogie, dans Net-entreprises, système représentatif du secteur social, nous trouvons des processus bien cadrés comme le processus d'inscription ou le processus d'accès aux services sécurisés. Ces processus ont été partiellement décrits dans la section 3.2.2.1.3, et sont donnés plus en détail dans [GIP 2002b].

4.2.5.4.2 Exception

Dans un cas d'exception, c'est-à-dire pour tout accès au système en dehors d'un processus habituel, l'utilisateur doit déclarer un objectif d'utilisation. Dans le domaine de la santé, ceci correspond à des cas non prévus dans les processus de soins, où le consentement du patient est souvent impossible. Le but est de favoriser la vie des patients et de ne pas faire obstacle au travail du personnel soignant, tout en engageant la responsabilité de l'utilisateur. De même, dans la sphère sociale, nous pouvons énumérer des cas d'exception tels que :

- à l'échéance (urgence), les personnes autorisées peuvent forcer le système pour préparer ou valider les déclarations ou les paiements via les services de Net-entreprises ;
- lors de la sélection des déclarations en mode standard, le système propose à la personne autorisée une liste de déclarations correspondant au profil de son établissement ; toutefois, cette liste n'est pas limitative et la personne autorisée peut forcer certains paramètres ;
- il doit être possible de forcer le niveau[26] de sécurité pour certains services et dans des conditions très particulières.

Afin de satisfaire les objectifs cités précédemment (flexibilité et sécurité), nous suggérons que le système effectue deux types de contrôles dans le cas d'une exception :

- *un contrôle a priori* : les règles de sécurité doivent spécifier quel utilisateur (ou rôle) a le droit de déclarer quel objectif, et dans quelles conditions ; par exemple si le médecin est absent (grève, force majeure), tout professionnel soignant peut déclarer l'objectif "*urgence non habituelle*" et accéder au dossier médical du patient ; un autre exemple est

[26] Les accès à Net-entreprises sont paramétrés par trois niveaux de sécurité. Si l'utilisateur ne possède pas le niveau requis pour un service, l'accès lui est refusé, sauf dans le cas d'exception que nous décrivons.

celui d'un médecin qui a traité autrefois un patient et qui souhaite ré-accéder à son dossier en spécifiant comme objectif *révision du diagnostic* ;

- *un contrôle a posteriori* : pour plus de flexibilité, le système peut autoriser certains accès, avec un ensemble minimal de vérifications (contrôle a priori) ; pour renforcer la sécurité, nous proposons des contrôles supplémentaires, qui seront réalisés a posteriori, notamment à travers les fonctionnalités d'audit et éventuellement par des notifications automatiques des patients ou des médecins traitants ; l'analyse des enregistrements d'audit permet de vérifier a posteriori le bien fondé des décisions prises (par exemple, le caractère d'urgence non habituel) ; à cet égard, notre démarche spécifie ce type de contrôle à l'intérieur de la politique de sécurité, au même titre que les contrôles a priori.

Notons que certaines contraintes contextuelles décrites dans ce mémoire, notamment celles sur les rôles, peuvent être vues comme analogues aux contraintes d'intégrité dans le domaine des bases de données [Godfrey *et al*. 1998]. Toutefois, les contraintes d'intégrité sont des obligations qui valident les opérations après leurs exécutions (vérification en aval), alors que le contexte que nous décrivons (sauf le contexte d'utilisation) est vérifié avant d'autoriser ou non l'accès (vérification en amont).

Les concepts de base que nous venons de présenter sont des briques nécessaires et suffisantes pour construire Or-BAC. Dans la section suivante, nous montrons comment les assembler et les relier pour représenter la politique de sécurité à l'aide de ce modèle.

4.3. Représentation d'Or-BAC en UML

Or-BAC (pour *Organization-Based Access Control*) est un modèle de sécurité qui prend en compte la richesse et les particularités des SICSS. Il peut également être appliqué à une gamme très large d'applications complexes, interopérables et distribuées.

Une première représentation d'Or-BAC avec un modèle *entité-relation* – à laquelle nous avons contribué dans le cadre du projet MP6 – peut être trouvée dans [Abou El Kalam *et al*. 2003a ; Abou El Kalam *et al*. 2003b]. Dans ce mémoire, nous proposons une description plus détaillée en utilisant UML. Le modèle UML permet de surmonter le manque d'expressivité du modèle entité-relation, notamment : la récursivité, les différents types de relations (dépendance, agrégation, héritage), le comportement dynamique, etc.

Dans la section précédente, nous avons présenté les classes associations RdO, VdO et AdO. Selon le besoin et les niveaux d'abstraction, ces concepts peuvent servir à :

- structurer les sujets, les objets et les actions par des entités abstraites ;
- identifier les rôles, vues et activités présents dans chacune des organisations du système ;
- spécifier qu'un utilisateur peut jouer différents rôles dans différentes organisations, mais pas forcément les mêmes rôles dans chacune de ces organisations ; montrer qu'une même vue peut correspondre à des objets différents selon les organisations ; montrer qu'une même activité peut être implémentée différemment dans des organisations différentes ; etc.
- masquer la manière selon laquelle les organisations implémentent les activités, instancient les vues ou utilisent les rôles.

Par ailleurs, Or-BAC considère des *permissions*, des *interdictions*, des *obligations* et des *recommandations*, et tient compte du contexte dans lequel une requête est faite. La notion de "*contexte dans organisation*", notée *CdO* peut être définie de la même manière que les RdO, VdO et AdO.

Dans le modèle Or-BAC, une politique définit des règles du type : dans le CdO *c*, l'organisation *org* accorde au RdO *r* la permission (ou l'interdiction ou l'obligation ou la recommandation) de réaliser l'AdO *a* sur la VdO *v*. Nous relions ainsi les entités "*organisation*, *RdO*, *VdO*, *AdO* et *CdO*" avec une classe-association notée "*type d'accès*" et

qui correspond à une permission, obligation, interdiction ou recommandation (figure 4.10). Bien évidemment, cette manière de faire inclut le cas particulier où la règle ne concerne qu'une seule organisation ; dans ce cas, le CdO peut préciser que les organisations (du RdO, VdO, et AdO) sont identiques.

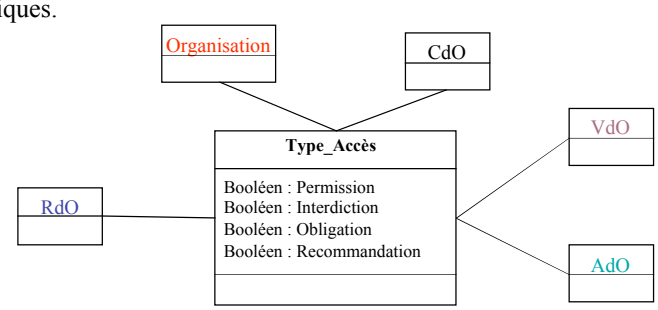


Figure 4.10 : Ébauche du diagramme de classe représentant les règles de sécurité.

D'une manière générale, une politique de sécurité comporte des faits du type : *Permission*(Purpan, Médecin-Dans-Rangueil, Lecture-Dans-Purpan, DossierMédical-Dans-Purpan, Catastrophe-Dans-Purpan) et *Permission*(Purpan, Médecin, Lecture, Dossier-médical, Médecin-traitant). La première règle implique deux organisations différentes et signifie que "l'hôpital Purpan accorde aux médecins de l'hôpital de Rangueil la permission de consulter n'importe lequel de leurs dossiers médicaux dans le contexte catastrophe (figure 4.11). La deuxième règle est plus restrictive et exprime que "l'hôpital Purpan accorde aux médecins la permission de consulter les dossiers médicaux des patients dont ils sont les médecins traitants".

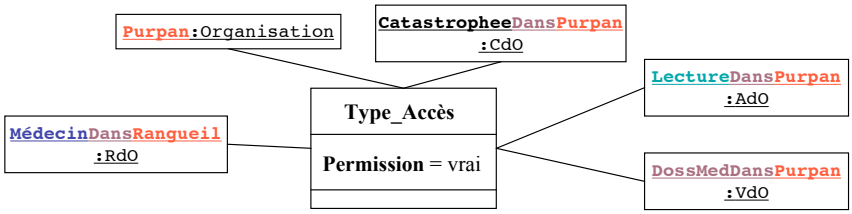


Figure 4.11 : Ébauche du diagramme d'objet représentant une règle de permission.

En outre, la politique de sécurité doit spécifier les conditions qui permettent de constater tel ou tel contexte dans telle ou telle organisation (CdO). Au moment de la requête, et avant d'accorder ou rejeter l'accès, le système doit vérifier (ou calculer) le contexte courant en fonction des relations qui existent entre le sujet demandant l'accès, l'objet invoqué, l'action demandée, et l'organisation impliquée.

Il est important de souligner que, dans Or-BAC, les règles de sécurité ne sont pas spécifiées pour chaque objet, sujet et action, mais seulement en utilisant des entités abstraites : organisations, rôles, vues, activités et contextes. Pour autant, le contrôle d'accès de bas niveau doit permettre de décrire les actions concrètes que réalisent les sujets sur les objets.

Nous distinguons ainsi deux niveaux d'abstraction (figure 4.12) :

- niveau abstrait, portant uniquement des entités abstraites (organisation, rôle, vue, activité, contexte) ; la politique de sécurité y est exprimée à travers la classe association type d'accès (*permission*, *obligation*, *interdiction* ou *recommandation*);
- niveau concret, portant sur des autorisations (ou obligations ou interdictions ou recommandations) concrètes associées, dans le contexte courant, à un utilisateur u_i, un

objet o_j et une action α_k. Ces faits (*permission*, *obligation*, *interdiction* ou *recommandation*) sont déduits, à un moment donné, par instanciation des règles de la politique de sécurité.

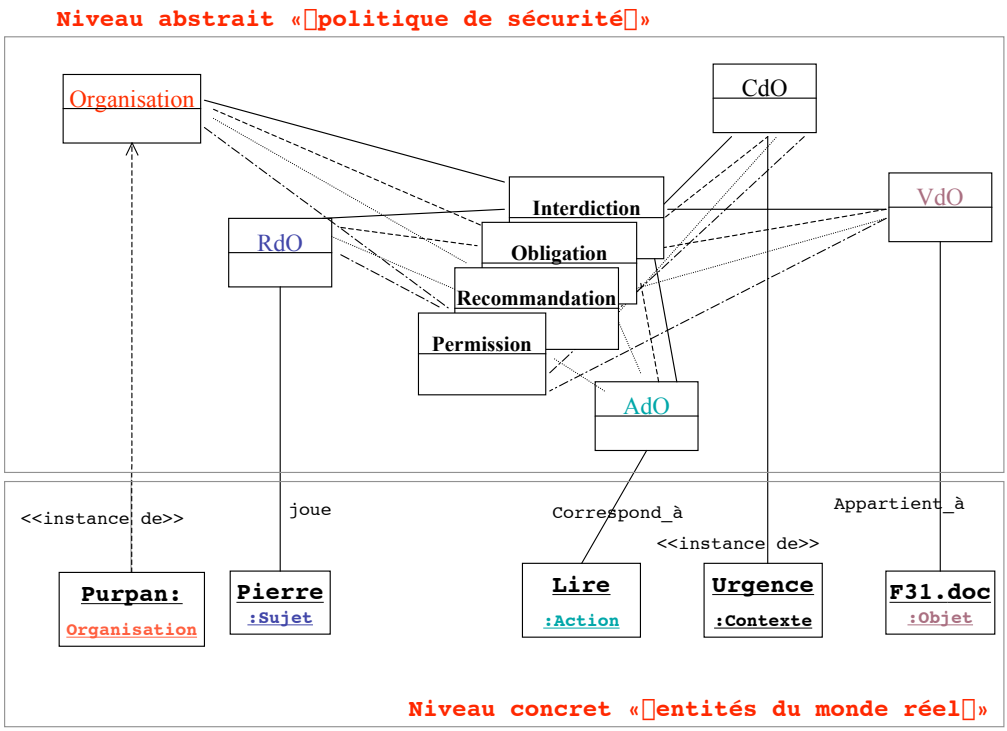


Figure 4.12 : Les deux niveaux d'abstraction du modèle Or-BAC.

En regroupant les ébauches de diagrammes de classe présentées précédemment, on obtient le diagramme de classe correspondant à Or-BAC (figure 4.13). Celui-ci s'explique ainsi :

Les éléments du système sont partitionnés en deux catégories : les éléments acteurs (auxquels la politique de sécurité attribue des droits) ou *sujets* et les entités passives ou *objets*, sur lesquels des actions sont effectuées. L'ensemble des objets contient des éléments qui ne peuvent jamais être des sujets (machines dossiers, etc.). Ces objets sont notés "$O \backslash S$". De même, puisque les sujets peuvent eux-mêmes être manipulés, ils forment une partition de l'ensemble des objets, ils sont ainsi notés : Objets qui Peuvent être Actifs "OPA". En particulier, les patients sont considérés comme des objets (lorsque l'infirmière leur fait une injection par exemple) qui peuvent être actifs en consultant leurs dossiers médicaux.

Par ailleurs, le modèle présenté considère une structure récursive sur les objets, les sujets, les activités et les organisations. Par exemple, les équipes sont aussi des sujets qui peuvent jouer des rôles et avoir des droits. L'ébauche du diagramme de classe de la figure 4.14-a montre d'une part que les utilisateurs, comme les équipes, sont des sujets (*relation d'héritage)* et d'autre part que les équipes sont composées de sujets, c'est-à-dire d'utilisateurs et d'autres équipes *(relation de composition)*. La figure 6.14-b donne la vision ensembliste correspondante. Cette manière de faire assure la flexibilité et la récursivité, dans la mesure où il est possible de former des équipes, qui à leur tour peuvent être regroupées pour former des équipes plus grandes et ainsi de suite.

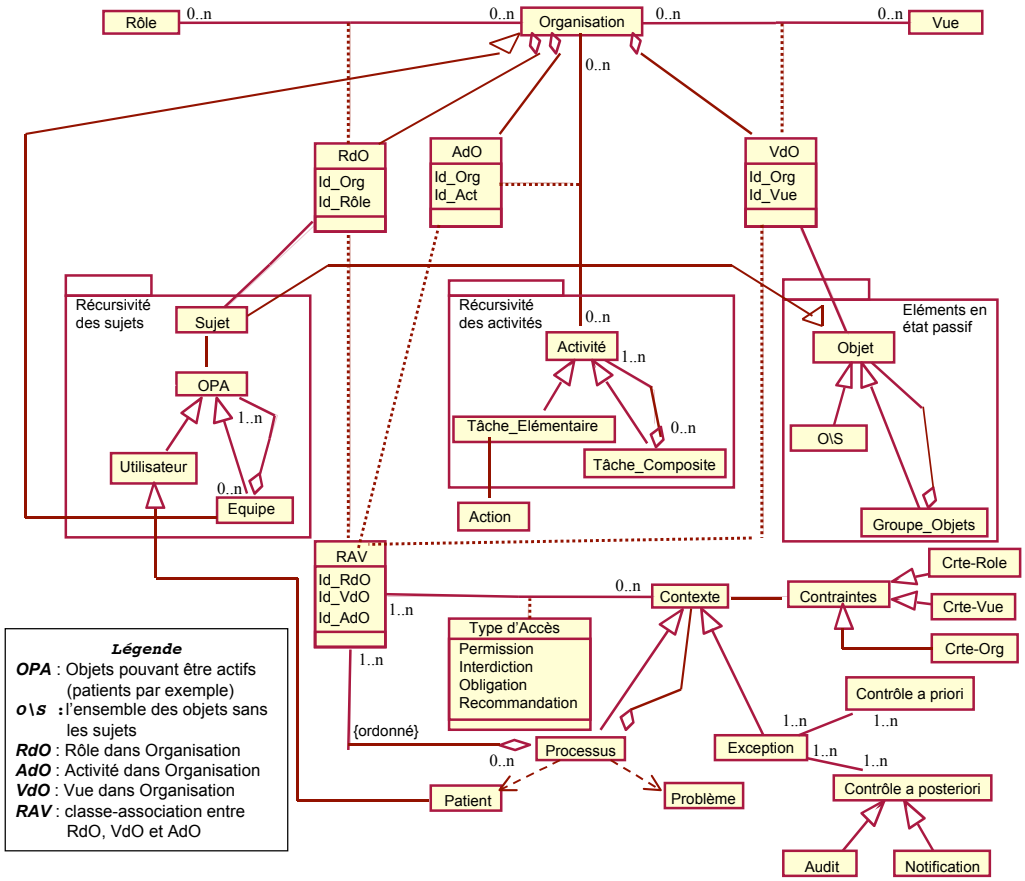
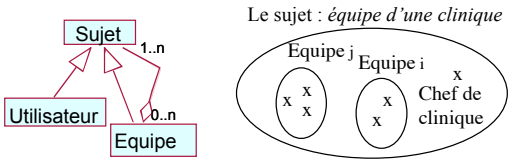


Figure 4.13 : Représentation UML du modèle Or-BAC.



Figures 4.14 (**a** et **b**) : Exemple de récursivité.

Les RAV (Rôle, Activité, Vue) ainsi que les autorisations peuvent êtres modélisés par des classes-associations. Tout processus est composé d'un ensemble ordonné de RAV. Il est identifié par un patient et un problème (la relation de dépendance exprime que toute modification du problème ou du patient entraîne un changement dans le processus). Une exception (ou objectif d'utilisation) est reliée à deux types de contrôles : un contrôle a priori et à un contrôle a posteriori. Un contexte est relié à des contraintes sur les rôles, les vues ou les organisations. Les *autorisations* sont attribuées à des RAV en tenant compte du contexte. Ce sont donc des classes-associations qui relient RAV et *contexte* et qui ont comme attribut soit une permission, soit une obligation, soit une recommandation, soit une interdiction.

Notons que le méta-modèle Or-BAC que nous avons présenté peut être appliqué, non seulement aux SICSS, mais aussi à une large gamme d'organisations et de systèmes.

Chapitre 5. Choix d'un formalisme pour Or-BAC

Préambule

Dans le chapitre précédent, nous avons présenté les concepts de base du modèle Or-BAC et nous avons proposé une représentation UML d'Or-BAC. Cette représentation offre des outils graphiques qui doivent guider le processus de spécification et de mise en œuvre de la politique de sécurité du système étudié (ce processus sera détaillé dans le chapitre 6).

Néanmoins, la simplicité de la représentation UML cache une réelle complexité de modélisation. À l'inverse, une représentation formelle doit fournir une spécification plus précise et non-ambiguë, comme elle devrait permettre une analyse plus rigoureuse notamment de la complexité, de l'adéquation entre le service attendu et la description opérationnelle, etc.

À cet égard, l'outil *MagicDraw* permet de traduire une spécification UML en langage formel *Maude* [Clavel *et al*. 2002]. Celui-ci offre des méthodes et des outils pour la vérification, le *model checking*, le calcul de complexité, etc.

L'objet de ce chapitre est de présenter un autre formalisme logique capable de modéliser les règles de fonctionnement, les objectifs de sécurité ainsi que les règles de sécurité. Ce formalisme permet d'étudier un autre volet de la vérification formelle, en offrant des outils d'aide au raisonnement sur les permissions, les interdictions, les obligations et les recommandations.

Aussi, ce chapitre est-il articulé en quatre parties.

Il commencera par présenter l'intérêt d'une approche formelle : consultation d'une politique de sécurité, étude de la cohérence de cette politique, vérification des propriétés attendues, etc.

Ensuite, il détaille les différents langages susceptibles de représenter une politique de sécurité fondée sur Or-BAC, en l'occurrence la logique du premier ordre, la logique modale, et notamment une de ses branches, la logique déontique. Celle-ci a l'avantage d'utiliser des permissions, des interdictions ainsi que des obligations.

Enfin, nous proposons un formalisme logique (fondé sur la logique déontique) pour Or-BAC. Ce formalisme sera ensuite utilisé pour représenter les règles de fonctionnement, les objectifs de sécurité ainsi que les règles de sécurité des SICSS. Par ailleurs, nous présentons des idées sur les méthodes (méthode des tableaux, logique possibiliste) d'exploitation de ce formalisme, notamment pour effectuer des vérifications et pour détecter et résoudre des conflits.

5.1. Intérêt d'une approche formelle

Le principal atout de l'utilisation d'une approche formelle dans la spécification d'une politique de sécurité réside dans l'élimination d'un certain nombre des ambiguïtés de la spécification, et d'aider l'administrateur à spécifier, à définir et à formaliser la politique de sécurité. Les profits en sont multiples : l'analyse des problèmes d'interopérabilité entre plusieurs politiques, l'interrogation de la politique par des requêtes concernant les accès, ou la manipulation de la spécification par des transformations mathématiques et avec l'assistance d'outils de preuve, notamment pour vérifier la cohérence de la politique de sécurité [Cuppens & Saurel 1996]. Ceci est particulièrement important dans les SICSS car, plus encore que dans d'autres domaines d'applications, les politiques des SICSS doivent permettre d'assurer *simultanément* les propriétés de confidentialité, d'intégrité, de disponibilité et d'auditabilité, dans un univers *dynamique* où les droits des utilisateurs doivent pouvoir varier selon le *contexte* courant. L'association d'un formalisme logique à la définition de politiques de sécurité est donc fort utile, notamment pour fournir à l'utilisateur les services que nous décrivons ci-dessous.

5.1.1. Consultation d'une politique de sécurité

Une politique de sécurité peut faire l'objet de plusieurs requêtes du type :

- Étant données certaines caractéristiques contextuelles, quelles sont pour un utilisateur jouant un certain rôle, les permissions (interdictions, obligations ou recommandations) en matière d'action sur un (des) objet(s) donné(s), ou de délégation de privilèges à une autre personne jouant un autre rôle ?
- Qui a des privilèges (et lesquels) sur un (des) objet(s) donné(s), et dans quel contexte ?
- Dans quel contexte tel utilisateur a-t-il tel privilège sur tel objet ou telle délégation de privilèges ?
- Qui peut déléguer quel privilège à un individu jouant un rôle donné, dans un contexte donné ?

5.1.2. Cohérence d'une politique de sécurité

En se basant sur un langage logique, avec une syntaxe et une sémantique bien précises, il est possible de vérifier la cohérence de la politique de sécurité. Globalement, on peut distinguer quatre types d'incohérence dans la politique de sécurité :

- Il peut s'avérer que certains objectifs de sécurité soient contradictoires les uns avec les autres.
- De la même manière, plusieurs règles de sécurité présentes dans la politique peuvent elles-mêmes se contredire.
- Il est parfois possible que les règles de fonctionnement entrent en conflit avec les objectifs et les règles de sécurité qui ont été définis.
- Enfin, étant donné que les règles de sécurité permettent de savoir comment un état de sécurité peut évoluer, et que les objectifs de sécurité permettent de savoir si un état est sûr, il est a priori souhaitable que l'on puisse vérifier s'il n'est pas possible, partant d'un état sûr et en appliquant les règles de sécurité, d'atteindre un état non-sûr (c'est-à-dire, un état qui compromet l'un des objectifs de sécurité). Notons que ce type de vérification peut être difficile, car il correspond à la résolution du problème de protection, indécidable dans le cas général [Harrison *et al*. 1976] (voir 2.2.2.2).

À cet égard, lorsqu'une politique de sécurité contient des permissions, mais aussi des interdictions, voire des obligations, il est nécessaire de s'assurer qu'elle ne peut pas générer de conflit, c'est-à-dire, garantir qu'il n'existe pas de situation dans laquelle un utilisateur aurait simultanément la permission et l'interdiction d'effectuer une action sur un objet.

5.1.3. Propriétés attendues d'une politique de sécurité

Il est nécessaire de s'assurer qu'avec une politique de sécurité définie et cohérente, il n'existe pas de situation dans laquelle un utilisateur peut violer une propriété de sécurité exigée. Par exemple :

- un utilisateur peut apprendre une information alors qu'il n'a pas l'autorisation de la connaître (confidentialité),
- il peut créer, modifier ou détruire une information alors qu'il n'en a pas l'autorisation (intégrité).

Il peut être intéressant d'identifier les éléments de la politique provoquant le non respect des propriétés de sécurité souhaitées, c'est-à-dire les sous-ensembles minimaux de règles qui génèrent l'incohérence dans certaines situations.

Par ailleurs, la modélisation formelle d'une politique de sécurité demeure primordiale si l'on souhaite vérifier que l'*implémentation* du système d'information, et en particulier des *mécanismes de contrôle d'accès*, permet bien de garantir les propriétés de sécurité souhaitées.

5.1.4. Complétude et interopérabilité

L'objectif d'une politique de sécurité est de définir les règles à respecter pour protéger le système contre les menaces identifiées lors de l'analyse des risques. Le problème de la complétude d'une politique de sécurité peut être vu comme celui de l'exhaustivité du règlement correspondant. Ceci peut être vérifié en montrant que, face à chaque risque identifié, il existe une règle spécifiée dans la politique de sécurité qui définit la conduite à tenir face à ce risque.

Des problèmes de fusion de politiques de sécurité peuvent également se poser, par exemple dans le cadre d'une restructuration entre deux organismes. Un premier aspect concerne la définition de rôles et de structures organisationnelles qui soient compatibles. Un autre aspect concerne ensuite la détection de conflits dans la politique obtenue par fusion, puis la proposition d'une méthode permettant de résoudre ces conflits. Des problèmes analogues se posent si l'on souhaite faire interopérer deux organisations dotées chacune de sa propre politique de sécurité.

5.2. Choix d'un langage de base pour formaliser Or-BAC

Le choix d'un langage formel pour la spécification d'une politique de sécurité s'effectue tout d'abord en fonction du domaine d'application de ce langage. Il semble indispensable de choisir un langage permettant, d'une part, de représenter naturellement des notions comme celles de permission, d'obligation, d'interdiction et de recommandation que l'on retrouve dans une politique de sécurité de manière générale, et d'autre part, de capturer les particularités et les concepts du système étudié, c'est-à-dire des SICSS.

Avec nos partenaires des sous-projets 3 et 4 du projet MP6, nous avons proposé d'associer à Or-BAC un formalisme logique fondé sur la logique de premier ordre [Abou El Kalam *et al*. 2003]. Dans ce mémoire, nous proposons un autre formalisme fondé sur la logique déontique [Abou El Kalam & Deswarte 2003]. Ce langage a été élaboré avec la contribution de Philippe Balbiani de l'IRIT.

Mais tout d'abord, présentons une brève description de la logique du premier ordre et de la logique modale, et plus particulièrement une de ses branches : la logique déontique.

5.2.1. Logique de premier ordre

La logique des propositions est l'étude des raisonnements dont la forme est constituée par des variables propositionnelles (p, q, r, …) et des connecteurs interpropositionnels tels que *et* : ($\wedge$), *ou* ($\vee$), *non* ($\neg$), *si … alors …*($\Rightarrow$). Malheureusement, ce type de logique ne permet pas d'effectuer des raisonnements comme le fameux syllogisme de Socrate : *tous les hommes sont mortels*, or *Socrate est un homme*, donc *Socrate est mortel* En effet, la logique des propositions, ne peut pas exprimer une assertion du type : *tous les hommes sont mortels*. C'est pourquoi la logique de premier ordre a été introduite [Kleen 1967]. Elle reprend l'ensemble des éléments de la logique propositionnelle et y ajoute des constantes (a, b, c, …), des variables (x, y, z, …), des prédicats (relations), des fonctions {$f(x, y, ..)$, $g(y, z,)$, ...)}, des quantificateurs universel $\forall$et existentiel $\exists$, etc.

Les constantes sont des symboles directement mis en correspondance avec les objets que l'on décrit ou sur lesquels on veut raisonner. Les variables peuvent être instanciées dans un ensemble de constantes. Les fonctions prennent comme arguments des variables ou des constantes pour retourner des valeurs prises parmi l'ensemble de constantes possibles. Les "*mots*" constitués des variables, des constantes et des fonctions appliquées à ces variables ou constantes forment les termes du langage : $t_1 := x$; $t_2 := f(x, y, b)$; $t_3 := a$, etc. Les prédicats ($P(t_1, .., t_n)$, $Q(t_1, .., t_n)$, ...) sont des relations qui ont pour arguments les termes du langage. Le langage prédicatif comporte toutes les formules bien formées à partir des formules atomiques ($\psi := P(t_1, .., t_n)$, $\varphi := (t_1 = t_n)$, ...), des connecteurs dyadiques de la logique propositionnelle {$\neg$, $\wedge$, $\vee$, $\supset$, $\equiv$} ainsi que des quantificateurs ($\forall$ et $\exists$). Si ψ et φ sont des formules bien formées du langage L, alors $\psi \vee \varphi$, $\psi \wedge \varphi$, $\neg \psi$, $\neg \varphi$, $\psi \supset \varphi$, $\forall x \psi$, $\exists x \varphi$, etc., sont des formules bien formées.

5.2.2. Logique modale

La logique modale est une extension de la logique classique dans laquelle, en plus des connecteurs booléens, on trouve les connecteurs intensionnels de la *nécessité* ($\Box$) et de la *possibilité* ($\Diamond$). Plus précisément, elle constitue un cadre formel pour l'étude de ces notions, en fournissant, outre une représentation explicite de celles-ci par des opérateurs modaux (pour la nécessité et pour la possibilité), la possibilité d'étudier leurs aspects intentionnels et déductifs, ainsi que leurs aspects extensionnels par la sémantique des mondes possibles (dite sémantique de Kripke). La logique modale, prise au sens large, est aujourd'hui l'un des meilleurs outils d'analyse scientifique pour l'étude de divers concepts considérés comme philosophiques. Ce sont principalement ceux de la nécessité, de la possibilité, de l'obligation, de la permission, du futur, du passé, du temps en général, du savoir, de la croyance, etc. En effet, selon le type de la modalité, diverses logiques modales peuvent être distinguées [Chella 1980 ; Catach 1989]. On trouve les modalités :

- Ontiques : « il est (nécessaire, possible, contingent, impossible) que p ».
- Temporelles : « il (sera, a été) (*toujours*, *à un moment donné*) vrai que p ».
- Epistémiques : « x (*sait*, *croit*, *doute*) que p ».
- Dynamiques : « il (sera, a été) (*nécessaire*, *possible*, *impossible*) en faisant … que p ».
- Déontiques : « il est (*obligatoire*, *permis*, *interdit*) que p ». Ces modalités, auxquelles il est opportun d'ajouter la *recommandation*, nous intéressent plus particulièrement.

Une intérprétation intuitive de toute logique modale [Kripke 1959 ; Kripke 1963] peut être fondée sur un modèle sémantique comprenant un ensemble de mondes différents, dans lesquels on considère la valeur de vérité des différentes propositions, notamment des propositions

contenant un opérateur modal. Par exemple, « *Nécessairement p* » est vrai dans un monde *w* si et seulement si « *p* est vrai dans tous les mondes *w*' directement accessibles depuis *w* ». L'idée est que tous les mondes (ou états) ne sont pas forcément directement (ou indirectement) accessibles depuis un monde *w*. Un monde *w* permet d'accéder directement à un monde *w*', seulement si toutes les propositions qui sont vraies dans *w*' sont vraies dans *w*. De la même manière, si une proposition est nécessairement vraie dans un monde *w*, elle doit être vraie dans tous les mondes *w*' auxquels *w* permet d'accéder.

5.2.3. Logique déontique

La logique modale est appelée *logique déontique* lorsque les formules "□A" et "◊A" sont lues « il est obligatoire que A » et « il est permis que A » respectivement. Le concept d'interdiction n'est pas oublié puisque la formule "□¬A" exprime justement l'interdiction de "A". Plus précisément, si $\mathbf{\Phi}$ est un ensemble de propositions atomiques *a*, *b*, *c*, … , si ¬, ∧, ∨, ⇒ et ⇔ désignent les connecteurs booléens habituels, et si **O**, **P** et **F** désignent les trois opérateurs modaux de la logique déontique (obligation, permission, interdiction), le langage de la logique déontique noté $L_O(\mathbf{\Phi})$ est l'ensemble des formules (ou expressions) construit par les règles suivantes :

- si $p \in \mathbf{\Phi}$, p est une formule,
- si *p* et *q* sont des formules, $\neg p$, $p \wedge q$, $p \vee q$, $p \Rightarrow q$ sont des formules,
- si p est une formule, **O***p*, **P***p* et **F***p* sont des formules.

En notation EBNF (Extended Backus Normal Form), L_O peut être donné sous la forme :

$$f ::= a \mid \neg f \mid f \wedge f \mid f \vee f \mid f \Rightarrow f \mid Of \mid Pf \mid Ff \mid$$

où $a \in L_O(\mathbf{\Phi})$ est une proposition atomique ; *f* est une formule du langage déontique $L_O(\mathbf{\Phi})$.

Une formule modale est une formule contenant au moins un des opérateurs modaux **O**, **P** et **F**. Les formules **O***p*, **P***p* et **F***p* désignent respectivement, "il est obligatoire que p", "il est permis que p", "il est interdit que p".

La *sémantique* associée à une logique modale normale est appelée sémantique de *Kripke*, ou encore sémantique des mondes possibles [Kripke 1963]. Un modèle de Kripke $\boldsymbol{M}$ est un triplet $(\boldsymbol{W},\mathfrak{R},\boldsymbol{V})$ où

- $\boldsymbol{W}$ est un ensemble de mondes possibles *w*,
- $\mathfrak{R}$ est une relation binaire sur W appelée relation d'accessibilité,et
- $\boldsymbol{V}$: W×Φ → {vrai, faux} est une fonction qui donne pour chaque monde $w \in$W la valeur de vérité V(*w*,*p*) de la proposition atomique *p*.

$\models_w^M p$ note le fait que la proposition *p* soit vraie dans un monde *w* dans le modèle $\boldsymbol{M}$. Cette valeur de vérité est définie de la manière suivante, qui étend le calcul propositionnel habituel :

- si $p \in \mathbf{\Phi}$, $\models_w^M p$ si est seulement si V(*w*,*p*) =vrai,
- $\models_w^M \neg p$ si est seulement si on n'a pas $\models_w^M p$,
- $\models_w^M (p \vee q)$ si est seulement si $\models_w^M p$ ou $\models_w^M q$,
- $\models_w^M \mathbf{O}p$ si est seulement si $\forall w' \in W \,/\, w \mathfrak{R} w', \quad \models_{w'}^M p$

Cette dernière définition peut se traduire par : "il est *obligatoire* que *p* est vraie dans un monde *w* si est seulement si *p* est vraie dans tous les mondes *w*' directement accessibles depuis *w*. Si on considère que l'opérateur **P** est le dual de l'opérateur **O**, c'est-à-dire $\mathbf{P}p = \neg\mathbf{O}\neg p$, on peut facilement déduire la valeur de vérité de **P**p.

- $\models^{M}_{w} \mathbf{P}p$ si est seulement si $\exists w' \in W / w \Re w'$, $\models^{M}_{w'} p$

La définition d'un système logique nécessite la considération d'axiomatiques et de règles d'inférence. À titre d'exemple, nous citons :

- La règle d'inférence : $\frac{p}{\mathbf{O}p}$ (dite Règle de Nécessité ou RN), qui signifie que si le système contient p (l'hypothèse), il contient $\mathbf{O}p$ (la conclusion).
- L'axiome K : $\mathbf{O}(p \Rightarrow q) \Rightarrow (\mathbf{O}p \Rightarrow \mathbf{O}q)$.
- L'axiome D : $\mathbf{O}p \Rightarrow \mathbf{P}q$. Si une logique déontique inclut cet axiome, alors la relation d'accessibilité $\Re$ est sérielle, c'est-à-dire : $\exists w \in W / \exists w' \in W / w \Re w'$.

5.3. Langage proposé pour Or-BAC

Dans la *section* 3.3, nous avons représenté Or-BAC à l'aide d'UML. Notre langage déontique "*L*" doit fournir une syntaxe permettant d'exprimer les instances des relations existantes entre les entités.

Chaque expression de *L* contiendra des symboles extraits d'un vocabulaire particulier classés en quatre groupes : les *symboles de constante*, les *variables individuelles*, les *symboles de relation* et les *symboles de fonction*.

5.3.1. Le langage

5.3.1.1 Constantes

Les *symboles de constante* correspondent aux instances des entités du diagramme. Ainsi, il y a autant de types θ de symboles de constante que d'entités dans notre diagramme, c'est-à-dire : *Organisation*, *Sujet*, *Objet*, *Action*, *Rôle*, *Vue*, *Activité* et *Contexte*. Nous avons par exemple les symboles de constante de type :

- Organisation, comme Purpan, Rangueil, ICU31, etc.,
- Sujet, comme Jean, Marie, ICU31, etc,
- Objet, comme F31.doc, F32.doc, F33.tex, etc.,
- Action, comme lire, écrire, select, etc.,
- Rôle, comme médecin, infirmière, unité_des_soins_intensifs, etc.,
- Vue, comme dossier_administratif, dossier_médical, dossier_chirurgical, etc.,
- Activité, comme lecture, écriture, consultation, etc.
- Contexte, comme urgence, etc.

Les constantes sont notées par des lettres minuscules comme *a*, *b* et *c*.

5.3.1.2 Variables

Les *variables* sont notées par des lettres minuscules comme *x*, *y* et *z*. Il y a des variables individuelles pour chaque type θ. Les symboles de constante de type θ et les variables individuelles de type θ seront appelés *termes-*θ.

Les symboles de relation de *L*, notées par des mots commençant par des lettres majuscules *P*, *Q*, *R*, etc., correspondent aux relations de notre diagramme. Chaque symbole de relation *P* de *L* est considéré comme un type de relation. Par exemple :

- *RdO* est un symbole de relation de type (*Organisation*, *Sujet*, *Rôle*).

- *VdO* est un symbole de relation de type (*Organisation*, *Sujet*, *Vue*).
- *Est_permis*, *Est_interdit*, *Est_obligatoire*, *Est_recommandé* sont des symboles de relation de type (*Sujet*, *Objet*, *Action*).

5.3.1.3 Formules atomiques

Avec les prédicats, les actions définissent les éléments fondamentaux du langage. Par exemple : TRANSMETTRE(*u, f, u*') : permet à *u* de transmettre le fichier *f* à *u*' ; $\text{CREER}_{\text{Org}}(u, org)$: permet de créer une organisation.

Au moyen des prédicats et des actions, on peut définir les formules atomiques : si *A* est un *prédicat* ou une *action* de type "$\lambda_1, .., \lambda_n$" et si "$t_1, .., t_n$" sont des termes de type "$\lambda_1, .., \lambda_n$" alors $A(t_1, .., t_n)$ est une formule atomique. Par exemple, le prédicat *RdO*(*Bob, médecin*, *Hôpital-Rangueil*) ; et l'action *LIRE*(*Bob,* F_5*.doc*) sont des dormules atomiques.

5.3.1.4 Fonctions

À ce stade, notre langage n'est pas assez expressif pour pouvoir comparer des entités. Dans de nombreuses applications, nous désirons dériver des informations concernant certaines propriétés des entités. D'un point de vue formel, des symboles de *fonction* sont utilisés pour décrire les attributs de ces entités. Les symboles de fonction sont notés par des lettres minuscules comme *f*, *g* et *h*. À chaque symbole de fonction *f* sont associés un domaine et un co-domaine (encore appelé domaine image de la fonction). Le domaine et le co-domaine d'un symbole de fonction dépendent de la nature des attributs qui lui sont associés. Si un symbole de fonction correspond à l'attribut *Nom*, alors son domaine est de type *Sujet* et son co-domaine est un ensemble de noms. De même, le domaine d'un symbole de fonction correspondant à un attribut *Âge* est de type *Sujet* et son co-domaine est un ensemble d'entiers positifs. Enfin, le domaine d'un symbole de fonction correspondant à l'attribut *Groupe_sanguin* est de type *Sujet* et son co-domaine est l'ensemble $\{A, AB, B, O\}$. Dans la mesure où il est possible que des sujets n'aient pas de nom ou que leur groupe sanguin soit inconnu, les symboles de fonction du langage *L* peuvent n'établir qu'une correspondance partielle entre les domaines et co-domaines associés. Dans de nombreuses situations, il nous est impossible d'attribuer une valeur unique à certains attributs d'une entité. Pour répondre à une telle situation d'un point de vue conceptuel, nous utilisons des symboles de fonction unaires ayant pour co-domaine l'ensemble des parties d'un ensemble. Pour illustrer ceci, il nous suffit de mentionner le cas de l'attribut *médecin_traitant* : le domaine du symbole de fonction associé est de type *Sujet* et le co-domaine associé est un ensemble d'ensembles finis de noms.

Afin de dériver les informations représentées par les symboles de fonction, nous devons introduire des relations binaires concrètes, notées par σ, τ et μ entre les domaines. Le type d'une relation binaire concrète est le couple correspondant aux domaines sur lesquels la relation s'applique. L'égalité est probablement la relation binaire concrète la plus simple que nous aurons à traiter. Considérons les exemples suivants :

- Si *t* et *u* sont des termes de type *Sujet*, alors *médecin_traitant(t) = médecin_traitant(u)* signifie que les sujets *t* et *u* ont au moins les mêmes médecins traitants.
- Si *t* et *u* sont des termes de type *Sujet*, alors *Âge(t) = Âge(u)* signifie que les sujets *t* et u ont le même âge.
- Si *t* et *u* sont des termes de type *Sujet*, alors *Groupe_sanguin(t) = Groupe_sanguin(u)* signifie que les sujets *t* et *u* ont le même groupe sanguin.

Bien évidemment, il existe certains cas où d'autres relations binaires doivent être considérées. Par exemple :

- Si t et u sont des termes de type *Sujet*, alors *médecin_traitant(t)* ∩ *médecin_traitant(u)* ≠ ∅ signifie que les sujets t et u ont un médecin traitant en commun.
- Si t et u sont des termes de type *Sujet*, alors $\hat{A}ge(t) < \hat{A}ge(u)$ signifie que le sujet t est plus jeune que le sujet u.
- Si t et u sont des termes de type *Sujet*, alors *Groupe_sanguin(t) ~ Groupe_sanguin(u)* signifie que les groupes sanguins de t est compatible avec celui de u.

Le langage est généré par la règle de grammaire suivante, donnée en notation EBNF, où f désigne une formule :

$$f := A(t_1, .., t_n) \mid \neg f \mid f \vee f \mid f \wedge f \mid \boldsymbol{O}f \mid \boldsymbol{P}f \mid \boldsymbol{F}f \mid \boldsymbol{R}f.$$

5.3.2. La sémantique

La sémantique utilisée est définie par le modèle $M = < W, \mathfrak{R}, V, D >$ où :

- $\boldsymbol{W}$ est un ensemble de mondes possibles w ;
- $\boldsymbol{\mathfrak{R}}$ est une relation d'accessibilité (relation binaire sur W) ;
- $\boldsymbol{D}$ est un domaine. Un domaine est un ensemble non-vide de valeurs ;
- $\boldsymbol{V}$ est une fonction qui donne les valeurs de vérité des éléments du langage : V(*Formule Atomique d'arité n*) ∈ D^n, V(*variable*) ∈ D, V(*constante*) ∈ D.

Intuitivement, (*Bob*, *médecin*, *Hôpital-Rangueil*) ∈ V(*w, RdO*) *»* signifie que dans le monde w, *Bob* joue le rôle médecin au sein de l'organisation "*Hôpital-Rangueil*". De la même manière, *«* (*Sam,* f_5*.doc*) ∈ V(*w,LIRE) »* signifie que dans le monde w, *Sam* exécute l'action *LIRE* sur le fichier f_5*.doc*.

5.3.3. Les conditions de vérité

En plus des conditions de vérité classiques relatives aux opérateurs ∧, ∨,¬, le langage présenté ajoute une condition de vérité qui permet de déterminer si une formule atomique est vraie dans un monde donné : $\models_w^M A(t_1,..,t_n)$ ssi $(V(t_1), .., V(t_n)) \in V(w, A)$.

Par exemple, $\models_w^M RdO$(*Bob, médecin*, *Hôpital-Rangueil*) ssi (*Bob, médecin*, *Hôpital-Rangueil*) ∈ V(*w, RdO*). En outre, les opérateurs modaux permettent de modifier les propriétés de la relation d'accessibilité $\mathfrak{R}$ entre les différents mondes du modèle associé à la spécification. Ils indiquent si deux mondes doivent ou non être accessibles l'un depuis l'autre. Rappelons la signification des formules déontiques (section 5.2.3) :

- $\models_w^M \mathbf{O}f$ ssi ∀w'∈W tel que wRw', $\models_{w'}^M f$: f est vrai dans tous les mondes accessibles à partir de w.
- $\models_w^M \mathbf{P}f$ ssi ∃w'∈W tel que wRw', $\models_{w'}^M f$: il existe au moins un monde accessible à partir de w où f est vrai.
- $\models_w^M \mathbf{F}f$ ssi ∀w'∈W tel que wRw', $\models_{w'}^M \neg f$: f n'est vrai dans aucun des mondes accessibles à partir de w.

Le langage proposé accepte quelques axiomes, notamment :

- $\boldsymbol{F}p \leftrightarrow \boldsymbol{O}\neg p$: l'interdiction de faire quelque chose est équivalente à l'obligation de ne pas le faire.

- $\boldsymbol{P}p \rightarrow \neg \boldsymbol{F}p$: la politique définit explicitement les permissions ; ainsi, toute chose permise est forcément non interdite (l'inverse n'est pas toujours vrai).
- $\boldsymbol{O}(p \wedge q) \leftrightarrow \boldsymbol{O}p \wedge \boldsymbol{O}q$: l'obligation de faire la conjonction de p et q est équivalente à l'obligation de faire p et l'obligation de faire q.
- $\boldsymbol{O}p \rightarrow \boldsymbol{R}p$ et $\boldsymbol{R}p \rightarrow \boldsymbol{P}p$: tout ce qui est obligatoire est recommandé, tout ce qui est recommandé est permis. La recommandation est donc une modalité plus forte que la permission et moins contraignante que l'obligation.

5.4. Utilisation du langage proposé pour spécifier une politique

5.4.1. Spécification des règles de fonctionnement

Le but de la spécification des règles de fonctionnement et des règles de sécurité est de définir les différents flux d'informations et les contrôles d'accès. Il ne s'agit de représenter que le fonctionnement pertinent vis-à-vis de la sécurité afin de pouvoir, ultérieurement, déterminer l'impact sur les objectifs de sécurité. La description se fait par le biais des opérateurs de la logique propositionnelle. Au niveau de la sémantique, les règles de fonctionnement définissent la structure (dite de Kripke) interne des mondes. Ce sont des axiomes ne contenant pas d'opérateurs modaux. Ils n'ont donc aucun impact sur les caractéristiques de la relation « $\Re$ » entre les mondes du modèle. En revanche, chaque monde est tel qu'il constitue un état de fait compatible avec les règles de fonctionnement. Ainsi, la règle $q \rightarrow r$ signifie que dans tout monde w où q est vrai, r l'est aussi.

5.4.1.1 Les sujets et les rôles

Les rôles joués dans les organisations sont facilement représentables dans le langage proposé à l'aide du prédicat "RdO". Supposons à titre d'exemple que l'hôpital Purpan habilite plusieurs sujets : Jean dans le rôle de directeur, Marie dans le rôle d'assistante administrative, ST_1 dans le rôle d'équipe chirurgicale et RT_2 dans le rôle d'équipe radiologique. Dans notre langage, ces faits sont représentés par des instances de la relation *RdO* :

- *RdO*(Purpan, Jean, directeur),
- *RdO*(Purpan, Marie, assistante_administrative),
- *RdO*(Purpan, ST1, équipe_chirurgicale) et
- *RdO* (Purpan, RT2, équipe_radiologique).

Dans ces faits, directeur, assistante_administrative, équipe_chirurgicale et équipe_radiologique sont des rôles.

Nous considérons par ailleurs qu'un attribut *Patient* associé à l'entité *Sujet* indique quels sont les patients d'un sujet. Par conséquent, notre langage inclut une fonction partielle *Patient* ayant pour domaine *Sujet* et pour co-domaine un ensemble d'ensembles finis de noms. Par exemple, les fonctions *Patient*(Purpan) et *Patient*(Michelle) retournent respectivement la liste des patients de l'hôpital Purpan et la liste des patients de Michelle qui est un professionnel soignant.

Dans le domaine social, Net-entreprises est une organisation où les entreprises (organisations) jouent les rôles : *tiers-déclarant*, *établissement-déclaré* et *établissement-adhérent*. En l'occurrence :

- *RdO*(Net-entreprises, Ernst&Young, tiers-déclarant) : pour Net-entreprises, Ernst&Young est (entre autres) un centre de gestion agréé à faire des déclarations pour des entreprises.

- *RdO*(Net-entreprises, Hôpital-Rangueil, établissement-déclaré) : Net-entreprises considère l'hôpital de Rangueil comme un établissement déclaré.
- *RdO*(Net-entreprises, Hôpital-Rangueil, établissement-adhérent) : l'hôpital de Rangueil est une organisation adhérente à Net-entreprises.

Puisque tout établissement déclarant ou déclaré doit être inscrit à Net-entreprises, une relation d'héritage existe entre ces rôles : *Sous-Rôle*(Net-entreprises, tiers-déclarant, établissement-adhérent) ; et *Sous-Rôle*(Net-entreprises, établissement-déclaré, établissement-adhérent).

5.4.1.2 Les objets et les vues

Considérons les objets appartenant aux vues suivantes :

- *dossier_administratif* : les objets qui appartiennent à cette vue fournissent des informations administratives concernant les patients comme leur nom, leur adresse, leur âge et leur numéro de sécurité sociale ;
- *dossier_médical* : cette vue correspond au dossier médical des patients, elle contient les données d'urgence, l'historique des consultations et des hospitalisations, etc. ;
- *dossier_chirurgical* : cette vue correspond aux dossiers de spécialité gérés par l'équipe chirurgicale.

Les objets appartenant à ces vues ont un attribut *Nom*. Si F_{31}.doc est un dossier appartenant à une de ces vues, *Nom*(F_{31}.doc) fournit le nom du patient correspondant. Nous supposons également que les dossiers sont directement gérés par l'hôpital Purpan qui utilise un système de fichier par exemple. Ceci se traduit dans notre modèle par des faits de la forme :

- *VdO*(Purpan, F_{31}.doc, dossier_administratif),
- *VdO*(Purpan, F_{32}.doc, dossier_médical), et
- *VdO*(Purpan, F_{33}.doc, dossier_chirurgical).

Si ST_1, l'équipe chirurgicale et RT_2, l'équipe radiologique, partagent la même base de données gérée par l'hôpital Purpan. Cela signifie qu'elles utilisent les mêmes vues que l'hôpital. Ceci peut s'exprimer de la manière suivante :

- $\forall o \forall v$ (*VdO*(Purpan, o, v) $\rightarrow$ *VdO*(ST_1, o, v)),
- $\forall o \forall v$ (*VdO*(Purpan, o, v) $\rightarrow$ *VdO*(RT_2, o, v)).

À partir des trois vues *dossier_administratif*, *dossier_médical*, *dossier_chirurgical*, nous définissons une quatrième vue, appelée *dossier_patient*. Nous supposons qu'elle a trois attributs *dossier_administratif*, *dossier_médical* et *dossier_chirurgical* tels que :

- $\forall o$ (*VdO*(Purpan, o, dossier_patient) $\leftrightarrow$ $\exists o_1 \exists o_2 \exists o_3$

  (*VdO*(Purpan, o_1, *dossier_administratif*) $\wedge$

  VdO(Purpan, o_2, *dossier_médical*) $\wedge$

  VdO(Purpan, o_3, *dossier_chirurgical*) $\wedge$

  dossier_administratif(o) = o_1 $\wedge$

  dossier_médical(o) = o_2 $\wedge$

  dossier_chirurgical(o) = o_3 $\wedge$

  Nom(o_1) = *Nom*(o_2) = *Nom*(o_3)).

La vue *dossier_patient* correspond au dossier médical complet du patient. Dans une base de données relationnelle, nous obtiendrions le dossier patient par une jointure des vues *dossier_administratif*, *dossier_médical*, *dossier_chirurgical* suivant l'attribut *Nom*.

Dans le domaine social, les déclarations (d_1, d_2, …) et paiements (p_1, p_2, …) effectués par les organisations adhérentes à Net-entreprise peuvent être considérés comme des objets appartenant aux vues : déclaration-annuelle-URSSAF, déclaration-trimestrielle-URSSAF, déclaration-mensuelle-URSSAF, déclaration-annuelle-ASSEDIC, déclaration-trimestrielle-ASSEDIC, déclaration- mensuelle-ASSEDIC, titre-de-paiement-annuel-URSSAF, titre-de-paiement- trimestriel-URSSAF, etc. Dans Or-BAC, les instances de la relation *VdO*(*organisation*, *objet*, *vue*) sont du type : *VdO*(Ernst&Young, d_{31}.txt, déclaration-mensuelle-URSSAF), *VdO*(CNRS, d_{32}.txt, déclaration-mensuelle-URSSAF).

En outre, les déclarations ainsi que les paiements associés aux unités[27] déclarées ou aux portefeuilles[28] peuvent être considérés comme des vues, en l'occurrence: *VdO*(Ernst&Young, d_{33}.txt, $F_{déclaration}$(*portefeuille-Grandes-Entreprises*)). Dans cette règle, nous avons utilisé une fonction $F_{déclaration}$ qui s'applique à un ensemble d'organisations et retourne l'ensemble des déclarations associées à ces organisations.

Dans la sphère sociale, la composition de vues peut être illustrée à travers des instanciations de la relation *:* "*Sous-Vue*(*Organisation, Vue, Vue*)", par exemple *Sous-Vue* (Ernst&Young, déclarations), $F_{déclaration}$(*portefeuille-Grandes-Entreprises*)) déclare que dans l'organisation Ernst&Young, les déclarations associées au portefeuille des grandes entreprises est une partition de l'ensemble des déclarations.

5.4.1.3 Les actions et les activités

Nous ne considérons dans les exemples ci-dessous que les activités correspondant à des accès directs aux dossiers, par exemple, la création, la consultation et l'écriture. Si nous supposons que les dossiers sont gérés par l'hôpital dans une base de données relationnelle, ces activités correspondent respectivement aux actions *insert*, *select*, *update*, etc. Ceci est représenté par les trois faits suivants :

- *AdO*(Purpan, insert, creation),
- *AdO*(Purpan, select, consultation) et
- *AdO*(Purpan, update, écriture).

Dans le domaine social, différentes activités peuvent être distinguées, à titre d'exemple nous citons :

- *Lecture* : pour visualiser une déclaration, un titre de paiement, un compte URSSAF, etc.
- *Lecture-écriture* (ou modification) : pour préparer une déclaration, modifier une déclaration déjà signée ou envoyée, etc.
- *Suppression*, pour effacer une donnée, ou radier une entreprise d'un service.
- *Signature* d'une déclaration ou d'un télé-règlement.
- *Envoi* d'une déclaration ou d'un accusé de réception.
- *Validation* des informations saisies.

Les processus où n'intervient qu'un seul sujet peuvent être vus comme des activités composites, par exemple : l'abonnement ou le désabonnement à un service ; la primo-inscription d'un administrateur ; la nomination ou la révocation d'une personne autorisée ; etc.

---

[27] Dans la section 3.2.2.1, nous avons défini une unité déclarée comme étant un sous-ensemble d'un établissement ou d'une entreprise. Elles peuvent correspondre à des découpages fonctionnels géographiques ou hiérarchiques. Ainsi, les déclarations concernant les cadres par exemple peuvent être considérées comme des vues séparées de celles concernant les autres employés.

[28] Dans la section 3.2.2.1.3.3, nous avons défini un portefeuille comme étant un ensemble d'établissements ou d'entreprises associés à une personne autorisée.

Dans Or-BAC, les associations des actions aux activités dans une certaine organisation sont données par la relation "*AdO*", par exemple :

- *AdO*(LAAS, sendMail, envoi),
- *AdO*(LAAS, openfile, consultation) et
- *AdO*(Ernst&Young, update, écriture).

5.4.1.4 La hiérarchie

L'héritage est un mécanisme qui permet de maîtriser la complexité. La règle : $\forall a \forall v \forall c$ (*Permission*(ST_1, r_1, a, v, c) → *Permission*(ST_1, r_2, a, v, c) exprime l'héritage des permissions dans ST_1 entre un rôle r_1 (par exemple médecin) et un rôle r_2 (par exemple chirurgien). Une autre manière de faire consiste à ajouter le prédicat "*Sous-Rôle*(*Organisation*, *Rôle*, *Rôle*)", ainsi pour spécifier que r_1 hérite des permissions de r_2, il suffit d'ajouter l'instance *Sous-rôle*(ST_1, r_1, r_2) à cette règle.

Précisons toutefois que dans notre modèle il est possible de spécifier que l'héritage entre deux rôles donnés ne s'applique qu'à certaines organisations. Nous pouvons par exemple spécifier qu'à l'hôpital Purpan, le rôle directeur hérite des permissions du rôle médecin : $\forall a \forall v \forall c$ (*Permission*(Purpan, médecin, a, v, c) → *Permission*(Purpan, directeur, a, v, c)).

Bien évidemment, l'héritage est spécifié au sein d'une organisation donnée, et nous n'aurions pas cette règle si, au lieu de considérer l'hôpital Purpan, nous considérions un autre établissement au sein duquel le directeur n'est pas un médecin. Il est également possible d'exprimer dans notre modèle l'héritage, entre rôles, d'interdictions, d'obligations et de recommandations.

Par ailleurs, de la même manière que nous avons spécifié la hiérarchie de rôles, nous pouvons modéliser la récursivité des vues, des activités et des organisations à travers lesnouveaux prédicats : "*Sous-Vue*(*Organisation, Vue, Vue*)", "*Sous-Activité*(*Organisation, Activité, Activité*)", "*Sous-Organisation*(*Organisation, Organisation*)". Par exemple, les vues *dossier_administratif*, *dossier_médical* et *dossier_chirurgical* sont des sous-vues de la vue *dossier_patient*. Ainsi, un rôle qui a la permission de réaliser une activité sur la vue *dossier_patient* a également la permission de réaliser la même activité sur les sous-vues précédemment citées. Ceci s'exprime dans notre langage par la règle suivante : $\forall r \forall a \forall c$ (*Permission*(*Purpan*, r, a, *dossier_patient*, c) → *Permission*(*Purpan*, r, a, *dossier_administratif*, c)). Il en est de même pour les vues *dossier_médical* et *dossier _chirurgical*.

Dans le domaine social, la spécification des accès (en amont) à Net-entreprises distingue deux rôles "de base" joués par les utilisateurs dans les organisations (section 3.2.2.1, page 69) :

- Personne inscrite,
- Personne non-inscrite.

Rien n'empêche qu'une personne inscrite puisse jouer le rôle de personne non-inscrite. Inversement, pour qu'une personne non-inscrite devienne inscrite, elle doit remplir certaines conditions (inscription).

Les rôles : dirigeant, mandataire social, administrateur et personne autorisée, sont des sous-rôles du rôle personne inscrite, ils héritent donc de ses propriétés (par exemple, l'appartenance à un établissement adhérent) et de ses privilèges (par exemple, le droit de modification de ses propres informations). Une relation d'héritage existe également entre le rôle *mandataire social* et le rôle *dirigeant*. Il est possible de faire appel au polymorphisme pour spécifier que le mandataire social est le dirigeant du siège de l'entreprise. Dans Or-BAC, cette hiérarchie de rôles peut être modélisée par les règles suivantes :

- *Sous-Rôle*(*Org*, Mandataire, Dirigeant),

- *Sous-Rôle*(*Org*, Dirigeant, Personne-inscrite),
- *Sous-Rôle*(*Org*, Administrateur, Personne-inscrite),
- *Sous-Rôle*(*Org*, Personne-autorisée, Personne-inscrite).

Si différentes personnes autorisées accèdent à différents services, le rôle "personne autorisée" est décomposé en plusieurs sous-rôles. Autrement dit, si dans une certaine organisation, les personnes qui préparent les déclarations, différent de celles qui les envoient (les modifient, ou les payent), il est nécessaire de considérer autant de sous rôles que de sous tâches distinctes des personnes autorisées. Nous pouvons ainsi déduire les règles de hiérarchie de rôles suivantes :

- *Sous-Rôle*(*Org*, Personne-autorisée-Préparer-Déclaration, Personne-inscrite),
- *Sous-Rôle*(*Org*, Personne-autorisée-Valider-Déclaration, Personne-inscrite),
- *Sous-Rôle*(*Org*, Personne-autorisée-Visualiser-Déclaration, Personne-inscrite),
- *Sous-Rôle*(*Org*, Personne-autorisée-Préparer-Paiement, Personne-inscrite),
- *Sous-Rôle*(*Org*, Personne-autorisée-Valider- Paiement, Personne-inscrite),
- *Sous-Rôle*(*Org*, Personne-autorisée-Visualiser- Paiement, Personne-inscrite).

5.4.1.5 Le contexte

Le contexte peut être exprimé par des règles de fonctionnement. En l'occurrence, l'exclusion mutuelle entre les rôles médecin de nuit et médecin de salle s'exprime par *RdO*(*u, Médecin*, *Org*)→[*RdO*(*u, Médecin-nuit, org*)↔¬*RdO*(*u, médecin-salle, org*)].

Le contexte "équipe traitant" entre un certain patient *o*, un sujet *s* et une organisation *org* peut être définie de la manière suivante : $\forall s\ \forall o\ \forall a\ CdO(org, s, o, \alpha, Equipe\text{-}traitante) \leftrightarrow \exists r\ RdO(org, s, r) \wedge Nom(o) \in Patient(s))$. Cette formule peut être interprétée ainsi : dans l'organisation *org*, le contexte "équipe traitante" est vrai entre le sujet *s* et l'objet *o* si et seulement si *s* joue un rôle dans *org* et si *o* est un dossier correspondant à un des patients traité par *org*.

De la même manière, le contexte "équipe traitante" est défini dans une certaine organisation ST_1 par :

- $\forall s \forall o \forall \alpha$ (*Définit*(ST_1, *s*, α, *o*, *équipe_traitante*) ↔ $\exists r$ (*RdO*(ST_1, *s*, *r*) ∧ *Nom*(*o*) ∈ *Patient*(ST_1)), c'est-à-dire, dans ST_1, le contexte "équipe traitante" est vrai entre le sujet *s*, l'action α et l'objet *o* si et seulement si *s* joue un rôle dans ST_1 et si *o* est un dossier correspondant à un des patients traité par l'organisation ST_1.

Le contexte peut éventuellement dépendre de contraintes temporelles. Le contexte "nuit" par exemple, peut être spécifié par la règle suivante : "$\forall s \forall o \forall \alpha$ (*Définit*(*org*, *s*, α, *o*, nuit) ↔ *Après*(20:00) & *Avant*(08:00)". L'opérateur "&" désigne la conjonction de contextes ; "*Après*" et "*Avant*" sont deux fonctions qui s'appliquent à des éléments du "*temps*" et qui retournent des contextes temporels. Elles peuvent être définies comme suit :

- "$\forall org\ \forall s\ \forall o \forall \alpha\ \forall t \in Temps,\ Définit(org, s, \alpha, o, après(t)) \leftrightarrow temps(\text{Horloge}) \geq t$". "Horloge" est une entité capable d'évaluer et de retourner le temps courant.

Le contexte "avant" peut être défini d'une façon similaire par :

- "$\forall org\ \forall s\ \forall o \forall \alpha\ \forall t \in Temps, Définit(org, s, \alpha, o, après(t)) \leftrightarrow temps(Horloge) \geq t$".

L'expression du lieu comme attribut contextuel peut être fait ainsi : "$\forall org\ \forall s\ \forall o \forall \alpha\ \forall t \in Temps,\ Définit(org, s, \alpha, o,$ Accès-Local) ↔ $@IP(s) \in IP\text{-}Local(org)$". Cette règle signifie que le contexte "Accès Local" est vrai dans l'organisation org entre le sujet *s* l'action α et l'objet *o*, si et seulement si l'adresse IP du sujet appartient à un intervalle d'adresses IP internes à *org*. Nous considérons que les sujets ont l'attribut "*@IP*"qui donne l'adresse IP du terminal

utilisé par le sujet et que les organisations ont l'attribut "*IP-Local*"qui retourne l'ensemble des adresses IP internes à cette organisation.

Pour offrir plus de flexibilité, nous avons défini l'*objectif d'utilisation* comme un contexte particulier revendiqué par des utilisateurs souhaitant intervenir dans des situations qui sortent des processus de soins habituels. Un objectif d'utilisation est caractérisé par des conditions a priori et des conditions a posteriori. À titre d'exemple, détaillons le contexte $Contexte_{UNH}$: "*Urgence Non Habituelle* (*UNH*)". Nous proposons de l'exprimer de la manière suivante : $\forall s \forall o \forall \alpha$ (*Définit*(ST_1, s, α, o, $Contexte_{UNH}$) $\leftrightarrow$ *Déclarer-Objectif* $\wedge$ *Condition-a-priori* $\wedge$ *Condition-a-posteriori*, avec :

- *Déclarer-Objectif* $\equiv$ (*créer*, s, o') $\wedge$ *VdO*(*org*, o', Vue_{UNH}) $\wedge$ *Patient-traité*(o')=*Nom*(o). Dans cette expression, nous considérons la vue "Vue_{UNH}" ayant un attribut "*Patient-traité*". Le sujet déclare l'objectif d'utilisation "urgence non habituelle" s'il insère l'objet o' dans la vue Vue_{UNH}, et si o correspond bien au patient figurant dans l'objectif déclaré.
- *Condition-a-priori* $\equiv$ *RdO*(*org*, s, *Personnel-Soignant*). Cette expression signifie que tout personnel soignant (dans une organisation) peut déclarer l'objectif d'utilisation "urgence non habituelle" (dans cette organisation).

Condition-a-posteriori $\equiv$ *Obligation*(*org*, *Système*, *Enregistrer*, $Vue_{Données\text{-}Audit}$, $Contexte_{Objectif}$), c'est-à-dire que le système (rôle) doit enregistrer les données d'urgence.

Les types de contextes existants dans la sphère sociale ne diffèrent pas de ceux énumérés précédemment, nous nous contentons ainsi de modéliser quelques exemples dans Or-BAC.

Nous commençons par exprimer le contexte temporel "déclaration-à-échéance". Pour cela, nous avons tout d'abord besoin de définir la fonction "*date*" qui s'applique à une action et un objet, et retourne la date d'exécution de l'action sur l'objet, par exemple "*date*(envoi, d_1.txt)" désigne la date d'envoi de la déclaration d_1.txt à Net-entreprises. Nous définissons par la suite la fonction "*avant-date*" qui s'applique à des triplets (d : date, α : action, o : objet), pour retourner un contexte temporel défini comme suit : $\forall s \forall o \forall \alpha$ *CdO*(*Org*, s, α, o, *avant-date*(d, α, o) $\leftrightarrow$ *date*(α, o)$\leq d$, c'est-à-dire, l'action α est exécuté sur l'objet o avant la date d. Nous avons également besoin de l'attribut "Nbre-salariés" d'une entité "organisation" pour désigner le nombre de salariés de cette organisation et de l'attribut "mois" d'une entité *date*, pour désigner le mois de cette date. Enfin, le contexte "déclaration-à-échéance" peut être défini par la règle :

$\forall s \forall o \forall \alpha$ (*CdO*(*Org*, s, envoi, o, déclaration-à-échéance) $\leftrightarrow$

[*Nbre-salariés*(*Org*) $\leq 9 \wedge$ *avant-date*(15 [(mois(*date*) *modulo* 3) = 1], α, o)] $\vee$

[$10 \leq$ *Nbre-salariés*(*Org*) $\leq 50 \wedge$ *avant-date*(15 mois(*date*)+1, α, o)] $\vee$

[*Nbre-salariés*(*Org*) $> 50 \wedge$ *avant-date*(5 février, α, o)].

En effet, une déclaration est considérée à échéance si elle appartient à l'un des trois cas suivants :

Si le nombre de salariés de l'entreprise ne dépasse pas neuf, la déclaration est trimestrielle et doit être faite avant le 15 du mois qui suit (c'est-à-dire avant le 15 des mois de janvier, avril, juillet et octobre) ; sinon, si le nombre de salariés est compris entre dix et cinquante, les déclarations sont mensuelles et doivent être faites avant le quinze du mois qui suit ; sinon (nombre de salariés supérieur à cinquante), les déclarations sont mensuelles et doivent être faites avant le cinq du mois qui suit.

Par ailleurs, puisque les autres contextes[29] identifiés dans les accès à Net-entreprises sont similaires à ceux présentés dans le scénario médical de la section précédente, il n'est pas nécessaire de les décrire à nouveau.

Une autre représentation du contexte associé à Or-BAC dans un langage logique du premier ordre a été effectuée par Cuppens et Miège [Cuppens & Miège 2003b]. Certaines différences existent toutefois entre leur vision et la nôtre. En particulier, ils catégorisent différemment le contexte et ne considèrent pas le contexte d'utilisation.

5.4.1.6 Les contraintes

L'utilisation de contraintes a été utilisée dans le contrôle d'accès à base de rôle "RBAC" [Sandhu *et al*. 1996]. Les contraintes sont exprimées dans notre modèle par des règles s'appliquant à diverses relations. Nous donnons les règles suivantes à titre d'exemple :

$\forall s$ (*RdO*(Purpan, *s*, équipe_chirurgicale) $\rightarrow$

($\exists s_1$ *RdO*(*s*, s_1, chirurgien) $\wedge$

$\exists s_2$ *RdO*(*s,* s_2, anesthésiste) $\wedge$

$\exists s_3$ *RdO*(*s,* s_3, infirmier)) .

Cette règle indique que si l'hôpital Purpan habilite *s* comme équipe chirurgicale, alors *s* habilite un chirurgien, un anesthésiste et une infirmière. Autrement dit, à l'hôpital de Purpan, toute équipe chirurgicale doit être constituée d'au moins un chirurgien, un anesthésiste et une infirmière.

Dans le formalisme logique associé à Or-BAC il est également possible de spécifier l'exclusion mutuelle entre les rôles (contrainte contextuelle), notamment pour modéliser des situations du type : au sein de l'hôpital Purpan, aucun sujet *s* ne peut être habilité à la fois comme chirurgien et comme anesthésiste : $\forall s$ $\neg$(*RdO*(Purpan, *s*, chirurgien) $\wedge$ *RdO*(Purpan, *s*, anesthésiste)).

Dans Net-entreprises différentes contraintes peuvent être énumérées, notamment celle interdisant de mettre un même établissement ou une même entreprise dans deux portefeuilles différents. Ceci peut se traduire par la règle : $\forall Org$ $\neg$[$VdO(Org, o, F_{déclaration}(portefeuille_X))$ $\wedge$ ($VdO(Org, o, F_{déclaration}(portefeuille_Y))$ $\wedge$ ($F_{déclaration}(portefeuille_X)$ $F_{déclaration}(portefeuille_Y)$].

5.4.2. Spécification des objectifs de sécurité

Les objectifs de sécurité sont exprimés par l'utilisation d'opérateurs modaux. Ces opérateurs permettent de modifier les propriétés de la relation d'accessibilité $\Re$ entre les différents mondes du modèle. Ils indiquent si deux mondes doivent être accessibles l'un depuis l'autre. F[*RdO*(*u, Pharmacien,* _)$\wedge$*CREER*(*u, ordonnance*)] est un objectif de sécurité relié aux propriétés de confidentialité et d'intégrité. Il correspond donc au point de vue de la sécurité au souci d'interdire une certaine formule. Plus précisément, il indique que, dans aucun des mondes accessibles, on ne peut trouver un utilisateur qui joue le rôle pharmacien et qui crée une ordonnance.

---

[29] Il s'agit de contextes relatifs au *lieu*, par exemple le lieu de gestion d'un type de déclarations ou de référentiels ; ou relatifs à un objectif d'utilisation, par exemple un processus d'accès aux services de Net-entreprises ou à des cas d'urgences non habituelles où l'utilisateur peut forcer l'accès et effectuer ses déclarations ou ses paiements.

5.4.3. Spécification des règles de sécurité

Du point de vue formel, une règle de sécurité est une formule modale dont les différentes clauses ne sont pas toutes des formules modales (par exemple : $r \rightarrow \mathbf{P}q$). Elle reflète la manière dont l'état de sécurité est en relation avec les différentes permissions, obligations, interdictions et recommandations qui existent dans le système.

Dans le langage proposé, les règles de sécurité prennent essentiellement la forme suivante :

$\forall s \forall \alpha \forall o \forall r \forall a \forall v \forall c$

$\mathbf{P}(org, r, a, v, c) \wedge$

$RdO(org, s, r) \wedge$

$VdO(org, o, v) \wedge$

$AdO(org, \alpha, a) \wedge$

$CdO(org, s, \alpha, o, c) \rightarrow Est\_permis(s, \alpha, o)$:

Si l'organisation *org*, dans le contexte *c*, accorde la permission au rôle *r* de réaliser l'activité *a* sur la vue *v*, si *org* habilite le sujet *s* dans le rôle *r*, si *org* utilise l'objet *o* dans la vue *v*, si *org* considère l'action α comme faisant partie de l'activité *a* et si au sein *org* le contexte *c* est vrai entre *s*, α et *o*, alors le sujet *s* a la permission de réaliser l'action α sur l'objet *o*.

Cette manière de faire couvre parfaitement les différents cas d'accès qui se présentent dans le domaine médical. Néanmoins, dans la sphère sociale, les règles de sécurité sont un peu différentes. En effet, Le tableau 5.1 rappelle les diff.érents accès possibles :

Rôle {Type}	**Types d'accès autorisés**
Personne non-inscrite {*revendiqué*}	Demand e de renseignement ; abonnement ou désabonnement ; lecture actualités ; dépôt de commentaires ; informations : techniques et informations d'ordre général sur les partenaires et le fonctionnement de Net-entreprises ; accès au plan du site.
Personne inscrite	Informations et alertes personnalisées ; modification de ses propres informations d'inscription ou d'authentification.
Dirigeant {*attribué et anonyme par défaut*} ;{*revendiqué pour être nominatif*}	Vision d'ensemble sur l'adhésion de son établissement à Net-entreprises : visualisation de l'historique des modifications ; consultation des personnes autorisées et de leurs droits ; réinitialisation de l'administrateur ; radiation de l'établissement ; limitation du nombre d'administrateurs ; interdiction (ou levée d'interdiction) de toute nouvelle inscription pour son établissement.
Mandataire social {*revendiqué*}	Radiation de son entreprise ; limitation du nombre d'administrateurs ; interdiction (ou levée d'interdiction) de toute inscription pour son entreprise.
Administrateur {*revendiqué*}	Inscription ; gestion des personnes autorisées et de leurs droits d'accès aux déclarations : nomination ou révocation de ces personnes, consultation ou modification de leurs droits, définition de leurs périmètres d'habilitation ; résiliation de son inscription.
Personne autorisée {*attribué*}	Accès (préparation, validation ou visualisation) aux déclarations ou aux paiements

Tableau 5.1 : Droits associés à chaque rôle de notre scénario social.

Ce récapitulatif nous montre de nouvelles formes de règles, que nous n'avons pas analysé précédemment. Considérons la règle la plus originale dans les accès aux services de Net-entreprises : "un *administrateur* désigne une *personne autorisée* à effectuer des *types de déclarations* selon une certaine *modalité* pour le compte d'un certain *établissement*[30]".

Cette règle peut être décomposée en deux autres règles :

- une première règle donnant à l'administrateur le droit de désigner (ou révoquer) des personnes autorisées, c'est-à-dire d'affecter des utilisateurs au rôle personne autorisé ; cette règles s'appelle règle de délégation (ou d'administration) ;
- une deuxième règle donnant à la personne autorisée le droit d'effectuer des déclarations pour le compte d'un certain établissement.

Nous proposons de représenter les règles de délégation sous la forme "*Permission* (*organisation, RdO, AdO, VdO, CdO*)", avec :

- *organisation* est la structure qui décide de déléguer, c'est-à-dire l'établissement déclarant ;
- *RdO* est la fonction qui a le pouvoir de désigner, nommer, ou révoquer des personnes habilitées, dans notre cas, c'est le rôle administrateur dans l'établissement déclarant ;
- *AdO* correspond à l'activité nomination dans l'établissement déclarant (association d'un utilisateur à un rôle) ;
- *CdO* est le processus suivi par l'administrateur pour effectuer cette nomination, notons le par exemple "inscription de nouvelles personnes autorisées" ;
- *VdO* est le "schéma de la règle de délégation" ; les objets appartenant à cette vue sont des règles.

Récapitulons, la règle de délégation (qui autorise l'administrateur à affecter des nominations) a la forme suivante : "*Permission* (*établissement déclarant*, *administrateur*, *désigner ou révoquer*, *schéma de la règle de délégation*, *inscription de nouvelles personnes autorisées*)".

En insérant un objet dans la vue "schéma de la règle de délégation", l'administrateur désigne une personne autorisée pour effectuer des types de déclarations, pour une certaine organisation, selon une certaine modalité. Par conséquent, cette vue *particulière*, notée "V*_schéma-règle-délégation*", possède les attributs[31] suivants :

> *org* : l'*organisation* dans laquelle le rôle concerné sera joué, c'est-à-dire l'établissement déclaré ; $\forall$ *schéma-délégation*$_i$ $\in$ V*_schéma-règle-délégation*, *org*(*schéma-délégation*$_i$) = étab-déclaré$_\alpha$) ;
>
> *rôle* : le *rôle* délégué, c'est-à-dire "personne autorisée" ; *Rôle*(*schéma-délégation*$_i$) = personne-autorisée ;
>
> *sujet* : le *sujet* (Claire, Philippe, …) bénéficiant de la délégation, c'est donc l'utilisateur qui sera désigné (par l'administrateur) comme personne autorisée ; par exemple, *sujet*(*schéma-délégation*$_i$) = Philippe ;
>
> *activité* : l'activité correspondant à l'action que la personne autorisée (Philippe) peut effectuer, par exemple *activité*(*schéma-délégation*$_i$) = envoyer-formulaire-déclaration ;
>
> *vue* : la *vue* déléguée, c'est-à-dire la vue que la personne autorisée va manipuler ; il peut s'agir de types de déclarations ou types de titres de paiement, par exemple, les déclarations URSSAF, DUCS, CSSS, DADS-TDS : *vue*(*schéma-délégation*$_i$) = déclarations-URSSAF, *Vue*(*schéma-délégation*$_j$) = titre-paiement-DUCS, etc.

---

[30] Rappelons que l'établissement déclarant et l'établissement déclaré ne sont pas nécessairement les mêmes, en l'occurrence dans le cas d'un tiers déclarant.

[31] Bien évidemment, les attributs de cette vue sont instanciés à chaque fois que l'administrateur désigne ou révoque une personne autorisée, autrement dit à chaque fois qu'il insère un objet dans cette vue.

contexte : le *contexte* (ou processus, ou cadre) dans lequel la déclaration ou le paiement sera effectué, par exemple, *contexte*(*schéma-délégation*$_j$) = *processus-de-paiement*, *contexte*(*schéma-délégation*$_j$) = *processus-de-déclaration*, etc.

Pour résumer, les deux règles dont nous avons besoin sont :

Première règle :

- *Permission* (*établissement déclarant*, *administrateur*, *désigner ou révoquer*, *schéma de la règle de délégation*, *inscription de nouvelles personnes autorisées*) ;

Deuxième règle qui spécifie la forme logique de la vue "schéma de la règle de délégation". Nous la définissons de la manière suivante :

- $\forall$ schéma-délégation$_i \in$ V_schéma-règle-délégation

VdO(*org*, *schéma-délégation*$_i$, *schéma-règle-délégation*) $\rightarrow$

RdO(*établ-déclaré*$_\alpha$, *Philippe*, *Personne-autorisée*) $\wedge$

Permission(*établ-déclaré*$_\alpha$, *Personne-autorisée*, *envoyer-formulaire-déclaration*, *déclarations-URSSAF*, *processus-de-déclaration*)

Même si le principe reste toujours le même, la manière selon laquelle ce type de règles peut être exprimé n'est pas unique. Elle dépend étroitement du fonctionnement et des objectifs[32] de l'organisation (entreprise) étudiée. À cet égard, il faut d'abord avoir des réponses aux questions suivantes (qui dépendent de l'organisation) :

- Est ce que l'on considère les sous-rôles[33] du rôle personne autorisée ou pas ?
- Si oui, est ce que les privilèges associés aux sous-rôles sont stables (toujours les mêmes)? Dans ce cas, il serait préférable de décomposer la règle ci-dessus en deux parties : dans une première étape, il faut associer les permissions aux rôles, tandis que dans la deuxième étape, on affecte les utilisateurs aux rôles (les sous-rôles du rôle personnes autorisé) ; sinon, il serait également possible d'associer directement des permissions aux utilisateurs.
- Est ce que l'on considère que la vue "déclaration" et différente de la vue "paiement", ou à l'inverse, on considère une vue "type de déclaration" mais avec deux modalités différentes : "déclarer" et "payer".

Cette analyse nous conduit à discuter le problème d'administration des droits, sujet intéressant qui a fait l'objet d'études récentes, et qui est souvent associé à un modèle de contrôle d'accès. En l'occurrence, les modèles ARBAC97 « *Administration Role-Based Acces Control* » [Sandhu & Bhaidipati1997 ; Sandhu & Bhaidipati1999], ARBAC99 [Sandhu & Munawer 1999] ainsi que ARBAC02 [Oh & Sandhu 2002] ont été associés au modèle RBAC, tandis que le modèle AdOr-BAC (*Administration Model for Or-BAC*) a été associé au modèle Or-BAC [Cuppens & Miège 2003a]. Les modèles d'administration associés à *RBAC* considèrent les associations : URA (*User Role Administration*), PRA (*Permission Role Administration*) tandis que AdOr-BAC redéfinit ces deux associations et ajoute l'association *UPA* « *User Permission Administration* ». Analysons, à travers des exemples adaptés à notre scénario social, comment AdOr-BAC présente ces trois associations :

URA sert à affecter des utilisateurs à des rôles. Modéliser des règles du type "l'administrateur est autorisé à affecter un utilisateur au rôle Personne Autorisée (PA) dans l'établissement déclarée *établ-déclaré*$_\alpha$", nécessite la création des vues URA et *URA-PA- établ-*

[32] Un objectif peut par exemple être : une définition séparée des personnes autorisées, puis de leurs périmètres d'habilitations, ou à l'inverse, définir ces deux tâches en même temps.

[33] Nous avons déjà expliqué dans la section 3.2.3.2.1 que les sous-rôles du rôle personne autorisée peuvent être : personne autorisée à effectuer les déclarations, personne autorisée à effectuer des paiements, personne autorisée à consulter les déclarations etc.

déclaré$_{\alpha}$. *URA* possède trois attributs : *sujet* (sujet concerné par l'affectation), *rôle* (auquel le sujet sera affecté) et *org* (l'organisation dans laquelle le sujet sera affecté). *URA-PA- étab-déclaré*$_{\alpha}$. est définie comme suit :

- $\forall\ ura \in$ URA $VdO(org, ura,$ URA-PA- étab-déclaré$_{\alpha}) \rightarrow RdO($*org*$(ura),$ *sujet*$(ura),$ *rôle*$(ura))$; avec : "*org*(*ura*) = étab-déclaré$_{\alpha}$" et "*rôle*(*ura*) = personne autorisée" (ou l'un de ses sous-rôles).

La règle autorisant l'administrateur à gérer les rôles des utilisateurs ressemble à celle donnée dans l'exemple précédent (*Permission* (*établissement-déclarant*$_{\beta}$, *administrateur*, *gestion*, *URA-PA- étab-déclaré*$_{\alpha}$, *Processus-habilitation-PA*)).

PRA sert à associer des permissions à des rôles. L'association PRA a cinq attributs : *initiateur* (organisation où la permission s'applique), *rôle* (rôle qui bénéficie de la délégation), *privilège* (l'activité déléguée), *cible* (vue concernée par la délégation) et *contexte* (dans lequel la règle s'applique). Donner une nouvelle permission à un rôle correspond en fait, à créer un nouvel objet qui se conforme à l'association PRA. Le lien entre ces objets et la relation "*Permission*" est modélisé par :

- ∀org ∀contexte,

$VdO(org, pra, PRA) \rightarrow$ *Permission* (*initiateur*(pra), *rôle*(pra), *privilège*(pra), *cible*(pra), *contexte*(pra)).

Ainsi, la règle où l'administrateur est autorisé à affecter la permission "consulter des déclarations" au rôle "personne-autorisée" peut être modélisée comme suit :

- (*Permission* (*établissement-déclarant*$_{\beta}$, administrateur, gestion, PRA-PA- étab-déclaré$_{\alpha}$, Processus-habilitation-PA))

La vue *PRA-PA- étab-déclaré*$_{\alpha}$ est définie par :

- ∀pra, VdO(établissement-déclarant$_{\beta}$, pra, PRA-PA- étab-déclaré$_{\alpha}$,) ↔

VdO(*établissement-déclarant*$_{\beta}$, *pra*, *PRA*) ∧

initiateur(*pra*) = *étab-déclaré*$_{\alpha}$, ∧

rôle(*pra*) = *personne-autorisée* ∧

privilège (*pra*) = *consultation* ∧

cible(*pra*) = *déclaration* ∧

contexte(*pra*) = *processus-vérification.*

UPA sert à affecter des permissions à des utilisateurs. Cuppens et Miège distinguent deux cas différents nommés : UPA1 et UPA2. UPA1 donne à un utilisateur le droit d'associer une permission à un autre utilisateur pour réaliser une certaine tâche sur un certain objet, tandis que UPA2 donne à un utilisateur le droit d'associer une permission à un autre utilisateur pour réaliser une certaine activité sur une vue. Dans le cadre de ce mémoire, nous donnons seulement un exemple de UPA2. Celle-ci contient cinq attributs : *initiateur* (organisation dans laquelle la permission s'applique), *sujet* (sujet qui bénéficie de la délégation), *privilège* (l'action déléguée), *cible* (l'objet concernée par la délégation) et *contexte* (dans lequel la permission s'applique). Afin d'exprimer la règle "le dirigeant de l'entreprise déclarante peut associer, à l'administrateur, le droit de mettre à jour la table des personnes autorisées", nous devons, tout d'abord, définir la vue MJTPA (pour Mise à jour de la Table des Personnes Autorisée) qui est une sous-vue de UPA2 :

- ∀upa, *VdO*(établissement-déclarant$_{\beta}$, upa, MJTPA) ↔

VdO(*établissement-déclarant*$_{\beta}$, *upa*, UPA2) ∧

RdO(*étab-déclaré*$_{\alpha}$, *sujet*(*upa*), *Personne-autorisée*) ∧

privilège(*upa*) = *consultation* ∧

sous-vue(*étab-déclaré*$_{\alpha}$, *cible*(*upa*), *table-personnes-autorisée*).

Notons tout de même que la vision présentée dans [Cuppens & Miège 2003a] considère l'attribut initiateur comme l'organisation qui décide de la délégation (*l'établissement déclarant*) tandis que dans notre vision, il s'agit de l'organisation dans laquelle la permission d'applique (*l'établissement déclaré*).

Cette section montre que le langage proposé permet de couvrir la richesse des SICSS, notamment vis-à-vis de la représentation des permissions, des interdictions, des obligations et des recommandations. Néanmoins, comme tout langage basé sur la logique déontique, des questions comme l'interprétation sémantique des formules (modèles de Kripke) s'imposent fréquemment.

À cet égard, des techniques de déduction automatique, fondées par exemple sur la méthode des tableaux [Fitting 1993], ont été proposées. Entre autres, ils permettent de vérifier si, partant d'un état sûr (qui satisfait les objectifs de sécurité), et en appliquant les règles de sécurité, on retrouve un état (une situation) ou certains objectifs de sécurité sont violés (état non sûr). Un exemple de l'application de la méthode des tableaux dans le cadre des politiques de sécurité est donné dans [Ortalo 1998].

Afin de détecter et résoudre les conflits, on peut également utiliser la logique possibiliste. Rappelons qu'un conflit peut être défini comme une situation dans laquelle un utilisateur aurait simultanément la permission et l'interdiction, ou l'obligation et l'interdiction, d'effectuer une action sur un objet (section 4.4.1.2, page 112). Par exemple, l'application d'un enchaînement de règles permet de déduire qu'*un médecin ne traitant pas un patient P n'a pas le droit de consulter son dossier médical*, alors qu'un autre enchaînement de règles permet de déduire qu'*en cas d'urgence, n'importe quel médecin peut lire le dossier médical d'un patient*. Ces deux résultats amènent à un conflit dans le cas où un médecin n'ayant jamais traité un patient veut lire son dossier alors qu'il y a urgence. Nos partenaires de MP6 proposent d'utiliser la logique possibiliste [Benferhat *et al*. 2003] pour résoudre ce type de conflits en associant, aux formules de la base, des coefficients de confiance allant de 0 à 1. Ainsi, dans notre exemple, si l'on considère l'urgence comme étant prioritaire, on donne un coefficient x_1 pour la première règle, et x_2 pour la deuxième, avec $x_1 < x_2$. La comparaison des coefficients permet de ne retenir que la deuxième règle.

Chapitre 6. Application d'Or-BAC aux SICSS et mise en oeuvre

Préambule

Le modèle Or-BAC a été associé, d'une part, à une spécification UML pouvant guider le processus de mise en œuvre (chapitre 4), et d'autre part à un formalisme logique (chapitre 5) pouvant aider à raisonner sur les permissions, obligations, interdiction et recommandation ; à détecter et résoudre les conflits, etc.

Le présent chapitre propose une démarche progressive fondée sur l'utilisation d'UML [Muller & Gaertner 2000 ; Booch *et al*. 1999] pour la définition d'une politique de sécurité – fondée sur Or-BAC – d'un système ou d'une organisation.

Cette démarche – qui peut être appliquée à différentes organisations – tient compte des aspects conceptuels, statiques et dynamiques de la politique de sécurité. Nous allons ainsi montrer comment utiliser les différents diagrammes UML pour : représenter les interactions entre les utilisateurs et le système ; spécifier les concepts de permission, interdiction ou obligation ; représenter des processus ; détailler les types de relations ; etc.

La deuxième partie présente l'analyse conceptuelle, organisationnelle et opérationnelle que nous avons suivi pour mettre en œuvre un logiciel de contrôle d'accès pour un centre de soins dentaires contenant plusieurs organisations (cabinets, services, etc.). Seuls les aspects décrivant le passage entre la politique de sécurité (fondée sur Or-BAC) et le contrôle d'accès seront abordés en détail. L'analyse fonctionnelle du logiciel est donnée en annexe D.

6.1. Démarche UML

Avant de spécifier une politique de sécurité sous une forme compatible avec Or-BAC, il faut tout d'abord décrire cette politique (fonctionnement pertinent vis-à-vis de la sécurité, analyse des risques, objectifs de sécurité, règles de sécurité, etc.). Pour cela, nous proposons de suivre la démarche suivante :

La première étape consiste à déterminer le besoin. Cette étape a été décrite dans le troisième chapitre. Rappelons tout de même qu'elle se base sur les informations recueillies sur le système (textes et lois juridiques, documents décrivant l'organisation, etc.) pour identifier les ressources à protéger, les menaces, les objectifs de sécurité ainsi que les règles de sécurité. La détermination du besoin correspond ainsi à une description, dans un langage très proche des utilisateurs, de la politique de sécurité qu'on veut modéliser et mettre en place.

La deuxième étape consiste à représenter le besoin. Dans une démarche classique, et pour des raisons de simplification et de découplage, l'espace des besoins est segmenté selon les points de vue de chaque acteur (catégorie d'utilisateurs : rôles, équipes, organisations). Cette segmentation permet d'exprimer, acteur par acteur, les attentes (en terme de services, fonctions) vis-à-vis du système. La décomposition des besoins en acteurs est appelée décomposition en *cas d'utilisation*.

Chaque cas d'utilisation apparaît comme une classe de scénarios. Chaque scénario (instance de cas d'utilisations) est, en réalité, une séquence d'événements (interactions entre les utilisateurs et le système) qui peut être représentée par un *diagramme de séquences*. Si les cas secondaires (particuliers) sont nombreux, l'utilisation des cas d'utilisation est moins pertinente et il nous semble plus judicieux d'opter pour une représentation plus abstraite à l'aide des diagrammes d'*état-transition*. En amont des cas d'utilisation, les *diagrammes d'activités* peuvent représenter clairement les acteurs et leurs responsabilités dans le fonctionnement. Lors de la construction des cas d'utilisation, il faut se poser des questions du type : Quelles sont les tâches de chaque acteur ? Quelles informations doit-il créer, modifier, détruire, lire ? L'acteur devra-t-il informer le système des changements externes ? Le système devra-t-il informer l'acteur des changements internes ?

La troisième étape est l'analyse du domaine. La modélisation par les cas d'utilisation suit un critère de décomposition fonctionnel. Si ce critère de décomposition est conservé lors du passage à l'architecture, le système sera difficilement extensible. L'étape de l'analyse du domaine doit ainsi assurer la transition vers une vision *objet*, puis montrer comment les groupes d'objets collaborant (initialement issus du domaine et complétés par des objets de conceptions) réalisent les interactions décrites dans les diagrammes de séquences (qui documentent les cas d'utilisation). Bien évidemment, la vision orientée-objet, intégrée à ce niveau de l'analyse, tient compte des principes d'encapsulation, d'abstraction et de polymorphisme.

En outre, une modélisation UML repose sur une consolidation mutuelle de deux points de vue complémentaires : dynamique et statique.

Les aspects dynamiques sont représentés par des *diagrammes de séquences* et de *collaborations* (pour les cas particuliers) et par des diagrammes *d'états-transitions* ou *d'activités* (pour les cas généraux).

Les aspects statiques sont exprimés sous la forme de cas particuliers, dans les diagrammes *objets* (comme charpente de diagrammes de *collaboration*) et sous la forme de cas généraux dans les diagrammes de *classes*.

L'analyse du domaine est complétée par deux autres étapes : la description de l'interaction Homme-Machine (IHM) et le déploiement du code exécutable (représentation des éléments de

réalisation : fichiers, modules, bibliothèques, etc., puis description de l'environnement d'exécution).

Les cas d'utilisation sont décrits de manière textuelle, agrémentés de quelques diagrammes d'interaction. À ce stade de la modélisation, les interactions représentent les principaux événements qui se produisent. Plus tard, lors de la conception, ces événements seront traduits en messages qui déclenchent des actions. Dans un premier temps, seuls les scénarios nominaux (processus de soins) sont décrits. Nous rappelons que l'étude des cas d'utilisation a pour objectif de déterminer ce que chaque acteur attend du système, et que la détermination des besoins est basée sur la représentation de l'interaction entre les acteurs et le système. Pour faciliter la compréhension, l'application de notre démarche s'appuie sur un exemple simple, que nous allons étendre au cours de l'analyse. Le tableau 6.1 résume les fonctions accomplies par chaque acteur (nous limitons l'étude aux rôles, le même raisonnement peut être suivi pour les autres acteurs : équipes et organisations) [Abou El Kalam & Deswarte 2004].

Acteur	**Rôle dans organisation**	**Modalité d'accès**	
		Activité	**Vue**
Directeur	Supervision du centre	Lecture (R)	Données de gestion administrative, technique et comptable
	Mise à jour de la table du personnel	Lecture, écriture (W), mise à jour (U), destruction (D)	Table associant les individus aux rôles qu'ils jouent dans le centre
Professionnel de santé (dentiste, secrétaire)	Prise des rendez-vous	R, W, U, non-destruction	Fichier des rendez-vous
Dentiste non-traitant	Soins d'urgence	R	Dossier du patient sauf la partie interrogatoire
Dentiste traitant	Soins□consultation, diagnostic	R, W, U, non-destruction	Toutes les informations y compris la partie interrogatoire
	Prescription	R, W, non-destruction	Ordonnance
Secrétaire	Enregistrement et gestion des informations administratives	R, U, W, D	Informations administratives
Comptable	Facturation et gestion comptable	R, W	Facture, fichiers de comptabilité

Tableau 6.1 : Forme textuelle d'un exemple de politique de sécurité.

Cette représentation peut être spécifiée, en UML, à l'aide du diagramme de cas d'utilisation de la figure 6.1.

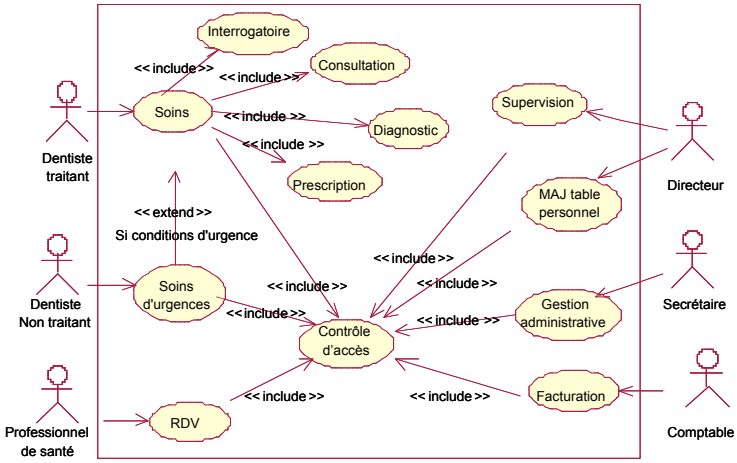


Figure 6.1: Exemple de diagramme de cas d'utilisation.

Dans ce diagramme de cas d'utilisation, nous distinguons deux types de relations :

- la relation *extend* dont le cas d'utilisation source est "soins d'urgence" et le cas d'utilisation destination est "soins" ; cette relation indique que, si les conditions d'urgences sont vérifiées, le comportement du cas d'utilisation "soins d'urgence" ajoute son comportement au cas d'utilisation "soins" ; autrement dit, les soins ne sont décrétés soins d'urgence que si certaines conditions sont vérifiées;
- la relation *include* qui permet de décomposer des comportements partageables entre plusieurs cas d'utilisation ; dans notre cas par exemple, toutes les fonctions doivent passer par la phase "contrôle d'accès".

Pour capturer les aspects temporels des interactions entre objets, chaque cas d'utilisation est ensuite décrit à l'aide d'un diagramme de séquence. À titre d'exemple, nous détaillons deux cas d'utilisation :

- "*gestion*" utilisée par la secrétaire pour modifier les informations d'un patient (figures 5.2-a et 5.2-b) ;
- "*contrôle d'accès*", inclus dans chacun des cas d'utilisation de la figure 6.1.

La figure 6.2-a exprime les séquences nécessaires à l'exécution du cas d'utilisation "gestion". La secrétaire demande au système d'information de lui fournir les informations qu'elle souhaite modifier[34]. Sa requête passe par une phase de contrôle d'accès que nous détaillons dans la figure 6.3. Si cette étape est réussie, le système lui fournit la liste des patients auxquels elle peut accéder. Une fois le patient choisi, le système affiche les informations concernant ce patient, auxquelles l'infirmière a le droit d'accéder. Dans ce cas, il s'agit d'informations d'ordre administratif. Après modification, le système effectue les sauvegardes demandées. Cet enchaînement d'opérations est d'abord décrit par un diagramme de séquence (figure 6.2-a). Celui-ci peut automatiquement être traduit en un diagramme de collaboration (figure 6.2-b). Nous expliquons par la suite que ces deux diagrammes permettent de capturer deux aspects complémentaires : temporel et spatial.

[34] Dans ce diagramme, on ne s'interesse qu'à la phase d'autorisation. Nous supposons ainsi que les phases d'identification et d'authentification se sont bien effectuées et que la secrétaire détient une preuve du succès de ces étapes. Dans la suite de cette section, d'autres diagrammes détailleront les phases d'identification et d'authentification.

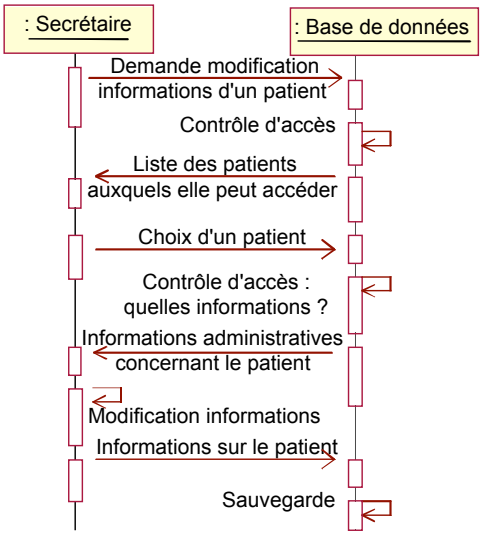


Figure 6.2-a : Exemple de diagramme de séquence.

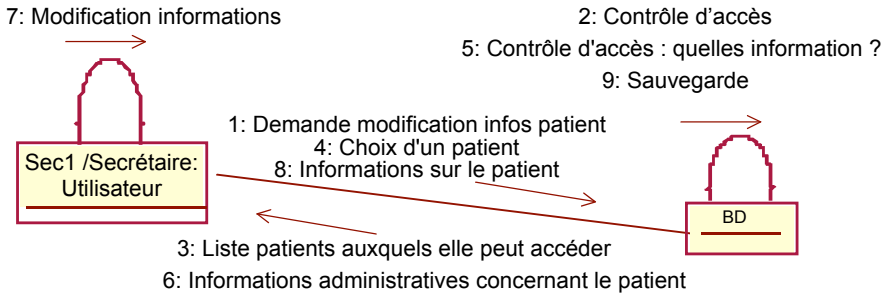


Figure 6.2-b : diagramme de collaboration correspondant.

Afin d'éclaircir les liens entre les différentes phases d'identification, authentification et autorisation, il nous semble nécessaire de décrire l'enchaînement temporel des étapes du contrôle d'accès avant d'autoriser (ou non), l'invocation d'un objet local (figure 6.3). Celui de la figure 6.4 décrit le cas où l'objet invoqué est distant (cas plus général). Ces deux diagrammes modélisent le schéma d'autorisation développé et implémenté dans le cadre du projet MAFTIA [Deswarte *et al*. 2001].

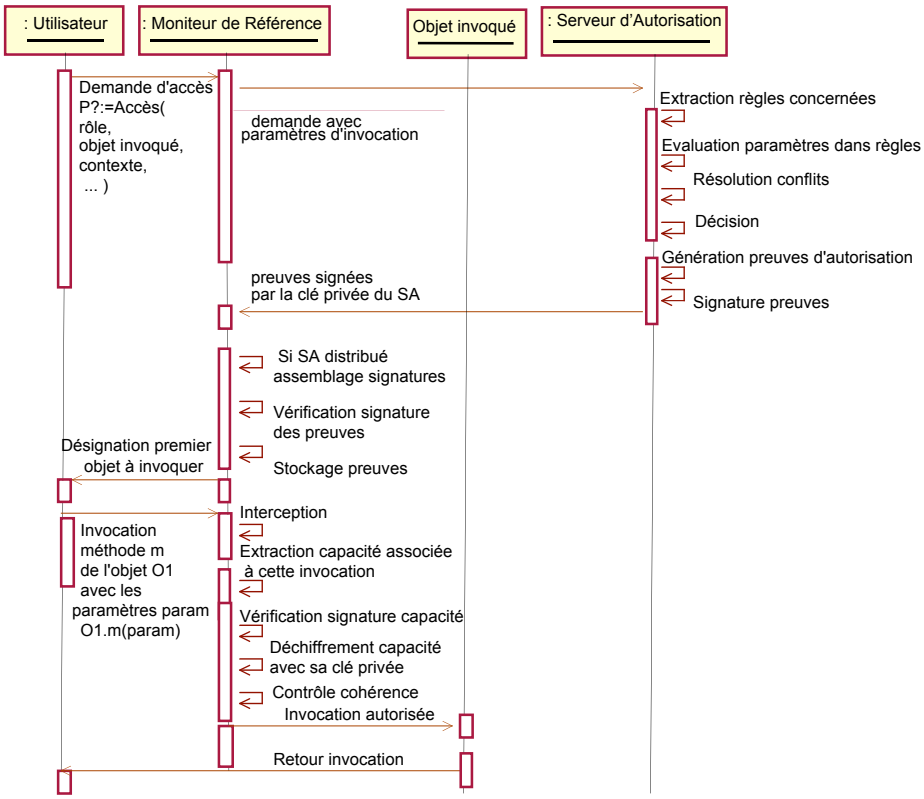


Figure 6.3 : Contrôle d'accès dans le cas d'une invocation d'un objet local.

Dans la figure ci-dessus, l'utilisateur envoie une requête avec un ensemble de paramètres comme son rôle, l'objet invoqué et le contexte. Le moniteur de référence de sa machine intercepte la requête et l'envoie au serveur d'autorisation. Ce dernier interroge la politique de sécurité et extrait les règles qui permettent de décider dans le cas courant. Il évalue les paramètres de la requête dans ces règles, résout les conflits[35] éventuels, et renvoie sa décision (est-ce-que l'action est permise, interdite, obligatoire ou recommandée ?). Ensuite, il génère un ensemble de preuves d'autorisation associées à cette action, signe cet ensemble (avec sa clé privée) et l'envoie au moniteur de référence.

Dans le projet MAFTIA, une preuve d'autorisation est définie comme un n-uplet dont les éléments sont les capacités donnant le droit à un sujet de réaliser une action. Si l'action comprend d'autres actions composites, cette preuve contient des coupons. Chaque coupon regroupe les capacités nécessaires à la réalisation de l'action composite. L'objet qui reçoit une preuve d'autorisation pour réaliser une action composite, utilise les capacités qui lui sont destinées, et achemine les autres capacités, aux autres objets intervenant dans le reste de l'action [Deswarte *et al*. 2001].

Comme il ne s'agit que d'un accès à un objet local, le moniteur de référence qui reçoit les preuves d'autorisation est celui du demandeur. Il stocke les preuves, et envoie à l'utilisateur le nom du premier objet à invoquer. Quand l'utilisateur invoque la méthode de cet objet, le

[35] Un conflit peut être, par exemple, le cas où la dérivation d'un ensemble de règles permet d'autoriser l'accès, alors qu'une autre dérivation impliquant d'autres règles l'interdit.

moniteur de référence intercepte cette invocation, extrait les capacités qui lui sont associées, vérifie leurs signatures, et les déchiffre avec sa clé privée. Il effectue également des contrôles de cohérence avant d'invoquer l'objet concerné. Le cas général où l'objet invoqué est situé sur une machine distante est présenté dans la figure 6.4.

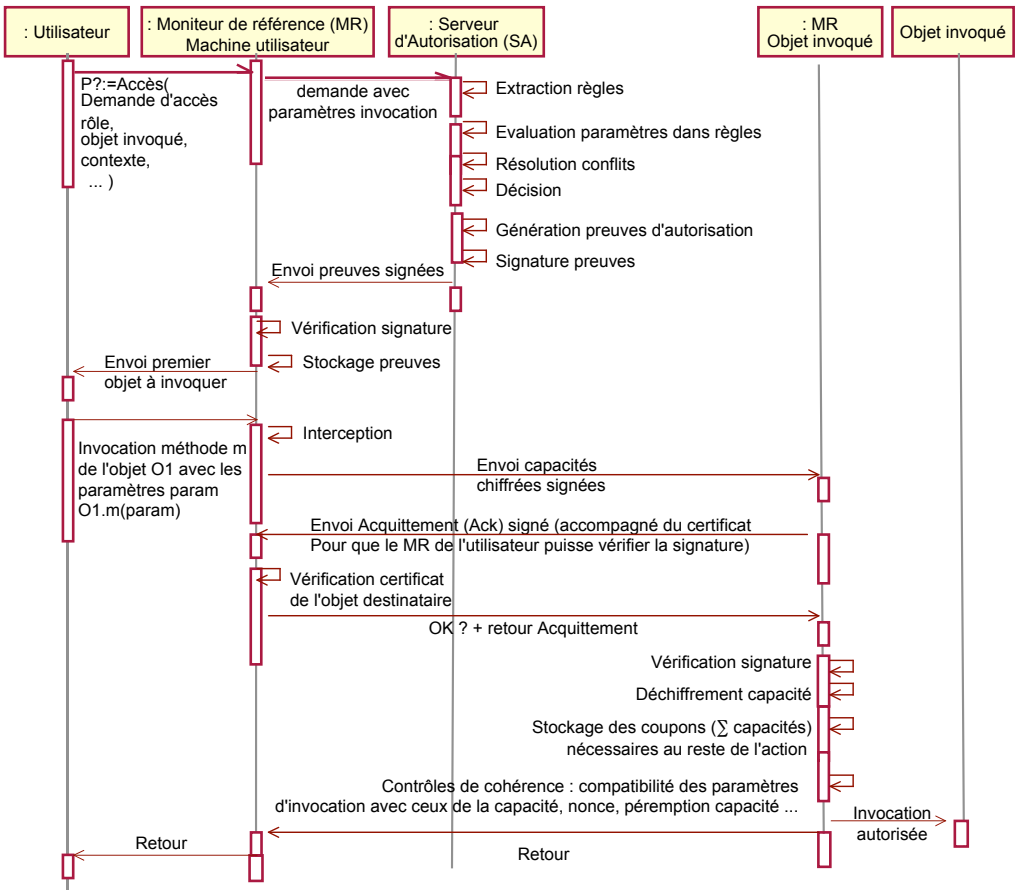


Figure 6.4 : Contrôle d'accès dans le cas d'une invocation d'un objet distant.

Il est important de signaler que, dans les schémas classiques de délégation (basés sur la procuration) un objet transmet à un autre objet certains de ses privilèges pour qu'il puisse réaliser une tâche en son nom. À l'inverse, le schéma proposé par MAFTIA applique le principe du moindre privilège car, même si un utilisateur contribue à une action composite, il ne peut effectuer que les actions qu'il a le droit d'exécuter. Chaque objet exécute sa partie de l'action avec sa propre identité. Un objet peut donc recevoir des droits ponctuels pour exécuter une partie de l'action composite, sans nécessiter que l'objet qui les lui transmet possède ces droits.

La figure 6.5 illustre le diagramme de collaboration correspondant au diagramme de séquence de la figure 6.6.

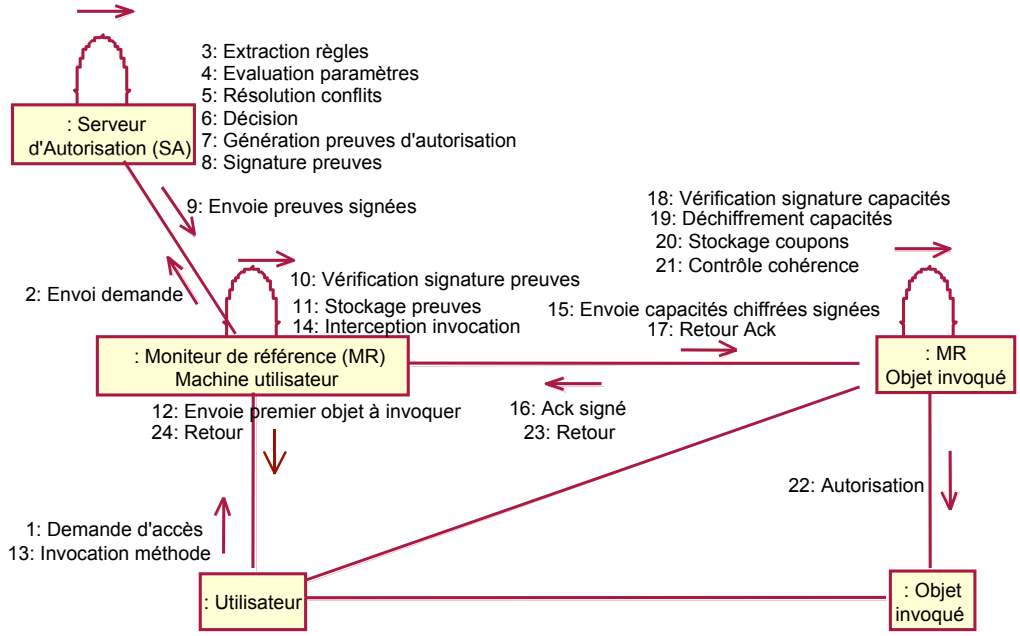


Figure 6.5: Diagramme de collaboration (invocation d'un objet distant).

Nous venons d'expliquer comment représenter une politique de sécurité à l'aide des cas d'utilisation. Nous avons détaillé, à travers des diagrammes de séquences, les étapes de certaines procédures d'accès (cas d'utilisation "gestion" et "contrôle d'accès"). Le diagramme d'activité[36] de la figure 6.6 représente un scénario récapitulatif des phases précédant la décision d'accès (est ce que l'accès est autorisé, refusé, obligatoire ou recommandé).

Un utilisateur $user_i$ commence par s'identifier et s'authentifier. Le système récupère ses attributs, lui propose de choisir un rôle, récupère et vérifie la hiérarchie et le contexte du rôle (en l'occurrence l'exclusion mutuelle). L'étape suivante consiste à choisir une organisation (ou à créer une nouvelle organisation). Dans cet exemple, nous supposons que l'utilisateur a choisi org_j et $rôle_k$, et que l'organisation est une équipe de soins. Le système vérifie si l'utilisateur a le droit de jouer $rôle_k$ dans org_j, avant de récupérer et de vérifier le contexte de l'organisation (lieu, certaines règles de fonctionnement, unité de rattachement, etc.). L'utilisateur a ensuite trois alternatives : choisir un processus de soins parmi les processus enregistrés dans le serveur et auxquels son organisation participe ; créer un nouveau processus (seulement si $rôle_k$ = médecin) ; ou déclarer, dans des cas particuliers et bien définis, un objectif d'utilisation. Dans ce dernier cas, le système vérifie les conditions *a priori* et lance le contrôle *a posteriori*. Une requête d'accès tient compte de paramètres tels que : les attributs de la connexion et de l'utilisateur, de la fonction jouée (rôle dans une organisation), de l'activité à réaliser, ainsi que du contexte (du rôle, de l'objet, de l'organisation et de l'utilisation prétendue). Le système extrait la ou les règles de sécurité concernées, évalue ensuite les paramètres de la requête dans ces règles (instanciation et déduction) et gère les conflits éventuels avant d'accorder ou de rejeter la requête d'accès.

[36] Certes, les diagrammes d'activités sont moins détaillés que les diagrammes de séquences, mais ils peuvent parfois offrir une vision glogale qui masque certains aspects, facilitant ainsi la compréhension pour certains types d'utilisateurs finaux.

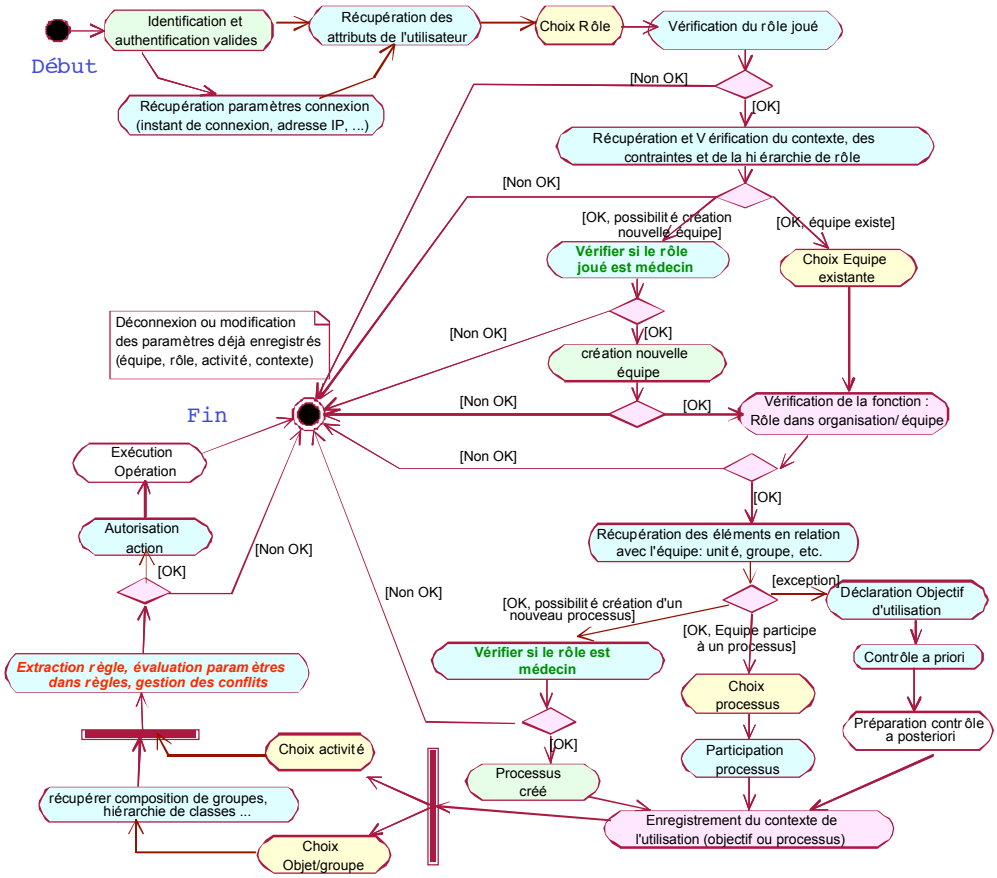


Figure 6.6 : Diagramme d'activité résumant les scénarios d'accès.

Pour résumer, UML définit les cas d'utilisation au moyen de collaborations entre objets du domaine. Chaque collaboration regroupe un contexte d'objets et une interaction entre ces objets. Les diagrammes de séquences ainsi que les diagrammes d'activité représentent les interactions en favorisant l'aspect temporel ; tandis que les diagrammes de collaboration insistent sur la structure spatiale qui permet la mise en œuvre de la collaboration d'un ensemble d'objets.

Le contexte d'objets est exprimé de manière particulière dans les diagrammes de collaboration et de manière générale dans les diagrammes de classe. Pour une application spécifique, un diagramme de classe, peut être déduit de la manière suivante :

- d'abord représenter les acteurs ainsi que leurs fonctions dans le diagramme de cas d'utilisation ;
- puis détailler chaque cas d'utilisation à l'aide de diagrammes d'interaction (celui-ci donne une représentation temporelle des interactions entre les acteurs et le système) ; chaque diagramme d'interaction correspond à un diagramme de collaboration qui insiste sur l'aspect spatial des interactions entre objets, instances des acteurs ;
- une ébauche de diagramme de classe est automatiquement déductible à partir d'un diagramme de collaboration ;
- le diagramme de classes est obtenu automatiquement par un assemblage des différentes ébauches ;

- enfin, éliminer les associations redondantes et ajuster le modèle.

L'utilisation d'UML pour spécifier une politique de sécurité permet de montrer les deux niveaux d'abstraction du modèle Or-BAC :

- *niveau abstrait* (*description générale au niveau spécification*), représenté par des diagrammes de séquences, de classes, de cas d'utilisations, etc. ;
- *niveau concret qui représente les entités du monde réel (description spécifique au niveau instance*), à l'aide de diagrammes d'objets et de diagrammes de collaboration ; ce niveau représente une instance particulière d'une interaction avec les objets, les liens, les stimulus (instances de messages) échangés, etc.

6.2. Spécification des concepts de la politique de sécurité

6.2.1. Concepts structurels

Plusieurs sections du quatrième chapitre expliquent les détails de modélisation des entités d'une politique de sécurité associée à Or-BAC (pages 101 à 110). En l'occurrence, ils montrent que les RdO (Rôle dans Organisation), VdO, AdO et CdO sont des classes associations entre l'organisation d'une part, et le rôle, la vue, l'activité et le contexte.

Un rôle (médecin par exemple) est modélisé par une classe dont le nom est celui du rôle (médecin). Les attributs de la classe décrivent les propriétés qui caractérisent les objets jouant le rôle, c'est-à-dire, les objets ayant un comportement similaire à celui identifié par le rôle. La spécialisation entre rôles peut être modélisée par un héritage (entre ces rôles).

Une organisation a été définie comme étant un groupe structuré d'entités actives (section 4.2.1, page 110). Elle peut donc être modélisée par une composition d'objets du système, ou par l'élément de spécification UML "sous-système[37]".

Les objets du système sont modélisés par des objets UML. Chaque objet appartient toujours à une classe qui peut changer au cours du temps. En l'occurrence, le fait qu'un objet puisse être tantôt actif (un patient qui consulte son dossier médical, et joue à ce titre le rôle patient) tantôt passif (lorsque l'infirmière lui fait une injection par exemple) est parfaitement modélisable à l'aide de l'héritage multiple entre classes.

Les stéréotypes d'UML peuvent également être utilisés pour préciser la sémantique de certains éléments comme les organisations, les rôles, etc.

Les contraintes peuvent être exprimées de plusieurs manières :

- par des contraintes de multiplicité sur les relations ou sur les classes ; par exemple pour restreindre le nombre d'utilisateurs jouant un rôle ou appartenant à une organisation, contraindre la structure d'une organisation, etc. ;
- par des expressions du langage OCL (*Object Constraint Languge*) ; celui-ci permet d'exprimer de manière formelle, des contraintes sous forme d'expressions booléennes prédéfinies ou personnalisées, par exemple, les contraintes sur le comportement d'un certain rôle ; néanmoins, le langage OCL reste peu expressif, il ne dispose pas d'une sémantique explicite et ne permet pas d'exprimer certaines contraintes importantes pour la sécurité comme les contraintes temporelles ;
- par du pseudo-code ou en langage naturel, bien entendu, de manière informelle.

[37] En UML, un sous-système est défini comme étant un ensemble d'éléments qui représente une unité comportementale du système physique.

6.2.2. *Concepts comportementaux*

Afin de modéliser les actions avec UML, il est possible d'utiliser les échanges de messages entre objets (invocation d'une méthode, par exemple). Dans les cas où l'on souhaite modéliser les interactions entre des organisations, les actions peuvent être modélisées par des transitions entre des diagrammes d'objets. Bien évidemment, la transition doit être étiquetée par la règle de sécurité définissant la modalité (permission, interdiction, obligation ou recommandation) de l'action, Si cette modalité est conditionnelle (par exemple, de la forme : il est permis que X seulement si Y), les conditions sont données par des expressions logiques (dans une notation proche du langage OCL).

Comme indiqué sur la figure 6.7, une règle de sécurité est représentée par une relation de dépendance entre deux sous-systèmes :

- La relation de dépendance est labellisée par le nom de la règle et possède une note avec la condition de garde.
- Le sous-système "avant" représente les conditions nécessaires pour que l'action soit réalisée, tandis que le sous-système "après" représente les effets de l'action sur le système.

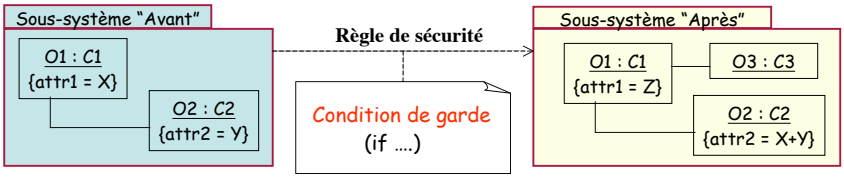


Figure 6.7: Exemple de représentation UML d'une règle de sécurité.

Il est évident que la sémantique et la nature des composants de la transition (sous-systèmes avant et après, conditions de garde) dépendent de la modalité de cette règle. Ainsi :

- Une permission est représentée par une transition dont le sous-système "avant" et les conditions de garde déterminent le scénario des actions permises ainsi que les participants (organisations, rôles, etc.) à ce scénario ; tandis que le sous-système "après" représente les effets de l'action.
- Une obligation interne est modélisée par un ensemble d'actions que le système doit assumer. Par exemple, l'obligation de redémarrer une machine quand un événement se produit.
- Une règle d'obligation externe, c'est-à-dire une obligation due à des entités externes au système, peut être modélisée par des chiens de garde observant les violations possibles de l'obligation et déterminant un plan de reprise (actions correctrices, par exemple) approprié (activable en cas de violation).
- Une règle d'interdiction peut être modélisée de manière similaire aux permissions (règle conditionnelle), mais sans état "après".

6.3. Exemple de mise en œuvre

Nous avons décrit le *méta-modèle* Or-BAC en utilisant deux visions : une vision *logique* qui sert à raisonner sur la politique de sécurité, et une vision *génie-logiciel* qui guide le processus de mise en œuvre. Dans le cadre du projet MP6, nous étudions actuellement la première tâche, tandis que dans cette section, nous présentons comment nous avons implémenté la politique de sécurité associée à Or-BAC dans un cas concret du domaine médical : centre de soins dentaires multiorganisationnel.

Le centre dentaire que nous considérons est une unité de soins (*organisation*) composée de trois cabinets (*sous-organisation*), d'un laboratoire de prothèse (*sous-organisation*) et de trois services : réception, administration et comptabilité. Chaque cabinet contient au moins un chirurgien dentiste (et parfois une assistante) ; le service administratif emploie une ou plusieurs secrétaires médicales ; au service de comptabilité sont affectés un ou plusieurs comptables ; tandis qu'un ou plusieurs prothésistes opèrent dans le laboratoire de prothèse. Selon la terminologie d'Or-BAC, nous avons :

- *Les organisations* : centre dentaire, cabinet dentaire$_i$, cabinet dentaire$_j$, cabinet dentaire$_k$, service de réception, service administratif, service de comptabilité et laboratoire de prothèse.
- *Les relations de compositions entre organisations* : *sous-organisation*(centre dentaire, cabinet dentaire$_1$), … , *sous-organisation*(centre dentaire, cabinet dentaire$_3$), *sous-organisation*(centre dentaire, service administratif), etc.
- *Les rôles* : administrateur, directeur, professionnel de santé, chirurgien dentiste, secrétaire, comptable et prothésiste.
- *Hiérarchie de rôles* : $\forall org \in Organisation$, *sous-rôle* (*org*, chirurgien dentiste, professionnel de santé), *sous-rôle*(*org*, secrétaire médicale, professionnel de santé), etc.
- *Les rôles joués dans les organisations* : *RdO*(centre dentaire, s$_1$, directeur), *RdO*(cabinet dentaire$_1$, s$_1$, chirurgien dentiste), *RdO*(cabinet dentaire$_1$, s$_2$, assistante dentaire), *RdO*(cabinet dentaire$_2$, s$_3$, chirurgien dentiste), *RdO*(service de réception, s$_6$, secrétaire médicale), *RdO*(service administratif, s$_7$, secrétaire médicale), *RdO*(service de comptabilité, s$_8$, comptable), *RdO*(laboratoire de prothèse, s$_9$, prothésiste), etc.

Étant donné que l'administrateur peut ajouter d'autres rôles ou organisations, et que le directeur peut modifier la table du personnel du centre, la liste présentée ci-dessus n'est pas limitative et peut évoluer au cours du temps. Toutefois, l'affectation d'un utilisateur à plusieurs rôles doit respecter les règles d'exclusion mutuelle. Par exemple, un sujet ne peut pas être à la fois comptable et directeur du centre.

Une analyse fonctionnelle détaillée de notre application est donnée en annexe D. Dans la suite de cette section, nous ne présentons que certains aspects reliés à la sécurité. Les droits d'accès implémentés sont ceux présentés dans le tableau 5.1 de la page 129. Nous avons ainsi les objectifs et les règles de sécurité suivants :

- Objectifs de sécurité :

  il est interdit aux administrateurs, secrétaires, comptables, prothésistes et assistantes de créer des ordonnances ;

  exceptés les dentistes traitants, les utilisateurs n'ont pas le droit d'accéder à la partie interrogatoire ;

  seul le directeur peut mettre à jour la table du personnel ;

  les diagnostics ne peuvent jamais être effacés ;

  les opérations de soins ne sont permises qu'aux utilisateurs qui sont personnels soignants en charge du patient ;

  le comptable a les droits "consultation et création" ainsi que les interdictions "non mise à jour, non destruction" sur les factures de tous les patients ;

  ..

- Règles de sécurité :

  Dans des cas d'urgence où en plus, le dentiste traitant est absent, tout autre dentiste peut accéder au dossier médical du patient en déclarant l'objectif d'utilisation *urgence*

non habituelle ; en même temps, le système enregistre automatiquement les paramètres de l'accès dans le fichier d'audit.

Les autres accès doivent s'inscrire dans des processus habituels décrits par le modèle conceptuel de traitement.

…

D'une manière générale, le logiciel que nous avons mis en œuvre réalise deux types de contrôles :

- Contrôle d'accès respectant la politique de sécurité associée à Or-BAC. En effet, il prend en considération : les rôles joués dans les organisations, les différents types du contexte, les vues, etc.
- Contrôle de cohérence et vérification de la validité des données saisies.

Pour implémenter les entités de notre application, nous avons fait le choix d'utiliser une base de données sous SQL. Nous avons ainsi les tables : utilisateur, intervention, rendez-vous, médicament, ordonnance, etc. (voir spécification fonctionnelle en annexe D).

Compte tenu de la grande quantité de données à gérer et de l'hétérogénéité de ces données, ce choix nous semble le plus adapté. À l'inverse, les mécanismes de contrôle d'accès SQL étant insuffisants pour ce type d'applications, nous les avons gérés au niveau du langage de programmation (*Visual Basic* 6).

Dans la spécification du modèle Or-BAC en UML, les notions les plus importantes sont des classes (les rôles, par exemple) ou des classes-associations (les RdO "Rôles dans Organisation", par exemple). Dans l'implémentation, une classe devient une table ayant comme champs, les attributs de la classe ; tandis qu'une classes-associations "C" entre deux classes "A" et "B" devient une table ayant comme clé primaire la concaténation des clés de "A" et "B". La figure 6.8 donne l'exemple de la classe association "Rôle-dans-Unité".

```
CREATE TABLE Rôle (
  Id_Méd Number (5),
  …
  PRYMARY KEY (Id_Rôle) )

CREATE TABLE Unité (
  Id_Unité Number (5),
  …
  PRYMARY KEY (Id_Unité) )

CREATE TABLE Rôle-Dans-Unité (
  Id_Méd Number (5) REFERENCES Médecin (Id_Rôle) ON
DELETE CASCADE,
  Id_Méd Number (5) REFERENCES Unité (Id_ Unité) ON
DELETE CASCADE,
  PRYMARY KEY (Id_Rôle, Id_Unité ))
```

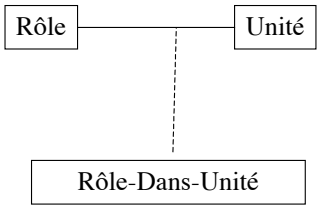


Figure 6.8 : Implémentation des RdO en bases de données.

Il est important de noter que la table "Rôle-dans-Unité" ne stocke jamais les attributs du rôle, ni ceux de l'unité ; elle fait seulement référence aux clés étrangères "Id_Rôle" et "Id_Unité". La clé primaire de la table "Rôle-dans-Unité" est la concaténation de la clé de la table "Rôle" et celle de l'unité. En outre, la contrainte "ON DELETE CASCADE" permet de maintenir l'intégrité référentielle en supprimant automatiquement de la table Rôle-dans-Unité les valeurs des clés étrangères Id_Rôle ou Id_Unité, à chaque fois que l'une de ces clés est supprimée des tables "Rôle" ou "Unité".

Pour implémenter une classe "B" qui hérite d'une classe "A", il suffit de créer deux tables "A" et "B" et de faire référence dans la table "B" à la clé de "A".

Par ailleurs, les permissions, interdictions, obligations et recommandations sont gérées au niveau du logiciel de programmation de la manière suivante :

- Les *permissions* sont implémentées à travers des capacités associées aux profils des RdO.
- La politique mise en œuvre considère que tout ce qui n'est pas explicitement autorisé est *interdit* ; dans certains cas, des *interdictions* sont exprimées à travers des capacités négatives et sont renforcées par des boîtes de dialogues indiquant à l'utilisateur qu'il n'est pas autorisé à faire l'action demandée.
- Les *obligations* sont implémentées par des actions automatiques comme l'enregistrement (automatique) de certaines données de connexion dans un fichier d'audit ou la fermeture (automatique) de l'application.
- Les *recommandations* sont pour le moment implémentées à travers l'affichage d'une boîte de dialogue avec un message "de recommandation" ainsi qu'un enregistrement automatique du choix fait par l'utilisateur. Ainsi, nous pensons qu'une recommandation est une autorisation qui, si elle n'est pas respectée, engage la responsabilité de l'utilisateur (c'est la raison pour laquelle le système enregistre le choix de l'utilisateur).

À chaque connexion, l'identification et l'authentification se font par vérification, dans la table "utilisateur", du "nom de login" et du mot de passe comme indiqué dans la figure 6.9.

```
" select * from Users where Login = '" & txtlogin.Text & "'", cn,
adOpenKeyset
If rs.RecordCount = 0 Then
    MsgBox " L'utlisateur n'existe pas ! "
    vbCritical, " Erreur de connexion "
    txtlogin.Text = ""
    Else
    If rs.Fields("Password") <> txtpassword.Text Then
        MsgBox " Mot de passe incorrect ! "
     Else
        loginconnexion = rs!login
        passwordconnexion = rs!Password
        libuser = rs!Lib_User
        profil = rs!Cod_Profil
        dateconnexion = rs!Dat_Connexion
        heureconnexion = rs!Hr_Connexion
        rs.Close
        Unload Frm_Connexion
        menuprincipal.Show 1
      End If
End If
```

Figure 6.9 : Phases d'identification et d'authentification.

Remarquons que durant les phases d'identification et d'authentification, l'utilisateur se voit déjà attribuer un profil. En effet, à chaque "nom de login" correspond un profil ; celui-ci est un numéro qui correspond à un role dans une organization "RdO".

Les règles de sécurité sont codées au niveau du langage de programmation en associant à chaque profil des permissions (ou interdictions ou obligations ou recommandations) de réaliser des activités sur des vues. La figure 6.10 exprime que seuls les utilisateurs jouant des RdO

correspondant aux profils inférieurs au profil "5" ont la permission de visualiser (VdO "*show*") les dossiers des patients (éléments de la VdO "frm_patient_U1").

```
If profil < 5 Then
    frm_patient_U1.Show 1
    Else
        MsgBox "Vous n'êtes pas autorisé à accéder à la gestion
       des patients ...", vbCritical
End If
```

Figure 6.10 : Exemple d'implémentation d'une règle de sécurité.

L'application "contrôle d'accès pour un centre dentaire", ainsi présentée et modélisée, a été implémentée en utilisant *Visual Basic* 6. Elle est disponible sous forme de démonstration. Outre l'aspect sécurité, son utilisation par tous types d'utilisateurs est facile. Elle répond à nos besoins : réalisation d'un contrôle d'accès respectant notre politique de sécurité, automatisation de la recherche d'informations autorisées sur les patients (gain de temps et disponibilité), possibilité d'ajouts de nouveaux utilisateurs, rôles, organisations, etc. Cette application est, pour le moment, restreinte à une implémentation logicielle des permissions par des listes de contrôles d'accès, et des interdictions par des capacités négatives. Elle pourrait être étendue pour intégrer de nouveaux mécanismes de contrôles d'accès.

Bien évidemment, les apports d'Or-BAC vont au délà de la mise en œuvre décrite ci-dessus ; et nous sommes actuellement en train d'implémenter Or-BAC dans un environnement – heterogène et distribué – faisant intervenir deux organisations distantes (deux hôpitaux). Le but est de montrer qu'Or-BAC fournit un cadre homogène ou chaque organisation spécialise (implémente) ses entités (activités, vues, *etc*.) comme elle le souhaite.

La première organisation utilise des documents XML alors que la deuxième utilise une base de données. Il en découle que la même vue (mais aussi la même activité) est implémentée différement dans chacun des deux hôpitaux.

Un utilisateur commence par se connecter, puis envoie au serveur d'autorisation (application client-serveur) son RdO (rôle dans une organisation), ainsi que la VdO (vue dans une organisation). Leserveur d'autorisation (SA) effectue les tâches suivantes :

- extrait les règles concernées (qui font intervenir le RdO et la VdO) ;
- construit une capacité (un objet au sens orienté-objet) contenant les paramètres utils {RdO, VdO, (P/I/O/R, Activité$_1$, contexte$_1$), (P/I/O/R, Activité$_2$, Contexte$_2$), .. , (P/I/O/R, Activité$_n$, Contexte$_n$)} ; ce qui signifie que le RdO à la permission, l'interdiction, l'obligation ou la recommandation de réaliser Activité$_1$, (resp. Activité$_2$, .. Activité$_n$) sur la VdO dans Contexte$_1$, (resp. Contexte $_2$, .. Contexte $_n$) ; un contexte peut être une durée de validité, un processus, etc. ;
- envoie la capacité à l'utilisateur.

L'utilisateur présente cette capacité au moniteur de référence[38] de l'objet demandé (une table de la base de données, par exemple) en précisant l'activité demandée lors de cet accès. Le moniteur de référence (situé sur la machine de l'objet demandé) déduit l'action à réaliser (à

[38] Le moniteur de référence regroupe les mécanismes de protection permettant de garantir le contrôle d'accès et de flux définis par la politique de sécurité. Toute tentative d'accès est réalisée via le moniteur de référence qui vérifie que chaque accès d'un sujet vers un objet est garanti par un droit d'accès.

partir de l'activité), vérifie si le contexte est vrai, et donne sa décision (permission, interdiction, obligation ou recommandation). Cette décision sera traduite par des mécanismes locaux d'accès.

Pour l'implémentation, nous avons fait les choix suivants (liste non exhaustive) :

- le langage java pour la programmation ;
- l'API JDBC pour l'accès à la base de données ;
- JAVA RMI pour l'invocation des objets distants ; pour un objet client, l'invocation est effectuée d'une manière transparente qu'elle soit faite sur un objet distant ou local ;
- Un formatage de données avant la transmission et, de l'autre côté de la communication, une adaptation aux logiciels de la machine locale de l'utilisateur lors de l'accès (pour visualisation ou modification, par exemple) ; pour cela, on utilise des Applets java ;
- Le serveur d'autorisation contient les règles de la politique de sécurité sous forme d'enregistrements d'une table ; la centralisation ainsi que l'isolation des règles de la politique de sécurité du reste de l'application, nous facilitera la tâche de vérification de la cohérence et de la complétude de cette politique.

Conclusion générale

Ce mémoire présente une démarche rigoureuse – fondée sur une politique de sécurité – pour mettre en œuvre une sécurité de bon niveau dans les systèmes d'information et de communication. Cette démarche a été appliquée aux domaines de la santé et des affaires sociales, qui présentent l'intérêt de combiner de fortes exigences de confidentialité, d'intégrité et de disponibilité, mais aussi de responsabilité.

En conclusion rappelons la logique adoptée tout au long de ce travail, et résumons les résultats obtenus.

Tout d'abord, plutôt que de dresser un état de l'art exhaustif des politiques, modèles et mécanismes de sécurité, nous avons adopté une analyse pragmatique, opposant ces méthodes aux besoins réels des SICSS. Le but est non seulement de proposer une sécurité robuste et bien fondée théoriquement, mais aussi flexible et adaptée aux demandes des usagers. Nous avons également opté pour une diversification des méthodes et outils utilisés pour la construction (progressive) de chaque solution proposée : visions génie-logiciel et formelle ; modélisation par MERISE et par UML ; développement d'un langage basé sur la logique du premier ordre, puis d'un autre basé sur la logique déontique, etc.

Le mémoire s'est organisé autour des axes suivants :

Présentation de l'état des lieux sectoriel, conceptuel et terminologique en sécurité pour les SICSS. Cette étape commence par une description sectorielle fondée essentiellement sur les textes juridiques relatifs aux SICSS, puis présente et établit les liens entre les concepts et termes classiques de la sûreté de fonctionnement en général, et de la sécurité informatique en particulier.

Discussion de politiques classiques de sécurité. Cette partie étudie en détail les avantages et les limites de chacune des principales politiques existantes. Une réflexion particulière est portée à la difficulté de l'application de ces approches aux SICSS : certaines introduisent une rigidité parfois incompatible avec les usagers, d'autres ne tiennent pas compte du contexte et ne permettent pas de représenter des interdictions et des recommandations, etc. Dans le même sens, les modèles actuels sont parfois trop complexes ou difficiles à administrer.

Élaboration de politiques de sécurité pour les SICSS. Nous avons, tout d'abord, montré que la définition d'une politique de sécurité est une étape nécessaire pour obtenir des systèmes pouvant satisfaire des exigences de sécurité élevées. Nous avons ensuite proposé une méthodologie dont les principales étapes sont : la description du système, l'identification des informations à protéger, la caractérisation des menaces, l'expression des objectifs de sécurité ainsi que des règles qui déterminent comment l'information sensible et les autres ressources sont gérées, protégées, et distribuées. Cette démarche est enfin appliquée pour établir une politique de sécurité dans un exemple de système d'information médicale, et une autre pour un exemple de la sphère sociale.

Présentation d'un nouveau modèle de sécurité. Pour spécifier les politiques développées, nous avons proposé, avec nos partenaires des sous-projets 3 et 4 de MP6, le nouveau modèle

Or-BAC. Celui-ci permet d'exprimer des permissions, des interdictions et des obligations, mais aussi des recommandations, concept fort utile dans le domaine de la santé. Or-BAC permet également de prendre en compte des informations de contexte dans l'expression des règles, afin de spécifier un contrôle d'accès fin et adapté. Il est aussi un moyen de spécifier, dans un cadre homogène, plusieurs politiques de sécurité pour des organisations différentes devant coopérer.

Modélisation formelle d'Or-BAC. Nous avons opté pour l'utilisation de la logique déontique pour fournir un cadre permettant de spécifier formellement une politique de sécurité basée sur Or-BAC. Cette spécification peut éventuellement servir à raisonner sur la politique de sécurité, sa cohérence, sa complétude, etc. Si la politique de sécurité est représentée dans un langage de la logique déontique, la méthode des tableaux nous semble idéale pour effectuer de telles vérifications. Par ailleurs, la logique possibiliste peut servir à résoudre les conflits éventuels dans la politique de sécurité.

Intégration dans un cadre de génie logiciel. Le modèle Or-BAC est d'abord représenté avec des diagrammes UML, puis intégré dans une démarche UML globale, couvrant à la fois les aspects statiques et dynamiques de la sécurité.

Développement d'un logiciel intégrant notre politique d'autorisation. Cette étape distingue deux types d'analyses : une analyse fonctionnelle (de l'application elle-même) couvrant les aspects conceptuels, organisationnels et opérationnels, préalable à la mise en œuvre ; puis indépendamment, une autre analyse sécuritaire éclaircissant le passage entre la politique de sécurité et les mécanismes de contrôle d'accès utilisés. Un logiciel de gestion d'un centre dentaire a été implémenté pour démontrer la faisabilité d'une implémentation en vraie grandeur les différentes facettes d'une politique Or-BAC.

Même si le travail présenté aboutit à une implémentation, un certain nombre de points restent à étudier. Ainsi, plusieurs perspectives de recherches peuvent être distinguées :

- Développer un scénario plus riche où plusieurs organisations de grandes tailles coopèrent, et utiliser le formalisme logique associé à Or-BAC pour effectuer un certain nombre de vérifications sur ce scénario. Les vérifications de la politique de sécurité peuvent porter sur la consistance, la complétude, la non-existence de canaux d'inférence, etc. Il peut également s'avérer important d'automatiser le test de la cohérence des politiques de sécurité, c'est-à-dire la détection des conflits. Ceux-ci peuvent éventuellement être résolus par la logique possibiliste [Benferhat *et al*. 2003].
- Afin de faciliter la conception et la manipulation de politiques fondées sur Or-BAC, il serait souhaitable d'utiliser une représentation graphique dont l'intérêt pratique peut s'avérer significatif si cette représentation est supportée par un outil d'édition.
- Éclaircir le passage de la politique aux mécanismes de sécurité (capacités distribuées, interprétations XML, etc.). Dans les bases de données, ce passage serait automatisable tandis que dans d'autres environnements, la politique serait interprétée dans un serveur d'autorisation.
- Il est nécessaire d'approfondir les études concernant la propriété de disponibilité, propriété importante dans bon nombre d'applications émergentes ou futures. Des investigations doivent être menées tant sur la façon de spécifier formellement le concept de disponibilité et le problème du déni de service au moyen d'une politique de disponibilité, que sur les mécanismes capables d'implémenter cette propriété.

Annexe A : menaces pouvant avoir des conséquences dans le monde médical

A1. Menaces pouvant porter atteinte à la confidentialité

Origine du problème	*Menace*	*Vulnérabilité*	*Conséquence*
Faute de conception.	Accès par des utilisateurs externes non autorisés aux données hébergées sur le serveur d'un hôpital.	Le nom du patient est utilisé comme identifiant.	Divulgation des informations concernant une personne.
Agrégation et centralisation des données dans un dossier accessible par un grand nombre d'utilisateurs. Faute de conception surtout pour les opérations du système socio-technique.	Divulgation non autorisée des données personnelles ; risque d'inférences, etc.	Manque de restriction d'accès aux utilisateurs (internes) du système.	Le dossier devient une ressource précieuse à divulguer, plus facilement accessible que des dossiers multiples dispersés ; risque de corruption des utilisateurs internes.
Faute de conception surtout pour les opérations du système socio-technique ; logique maligne.	Accès plus largement réparti aux informations médicales.	Manque de restriction d'accès aux utilisateurs du système.	Faire modifier les décisions qui doivent être prises (embauches, prêts, …).
Agrégation d'informations et non prise en compte du consentement du patient. Faute de conception surtout pour les opérations du système socio-technique.	Constitution de bases de données convoitées.	Contrôle d'accès insuffisant ; absence ou insuffisance de la politique de sécurité	Risque de pressions politiques pour faire légaliser l'autorisation d'accès à des entités non nécessairement médicales, pour exploitation à des fins non médicales (revente, etc.).

Origine du problème	*Menace*	*Vulnérabilité*	*Conséquence*
Faute de conception ; faute d'interaction ; logique maligne.	Possibilité d'accès direct à un ensemble de données.	Possibilités de croisement de données.	Divulgation d'informations personnelles.
Faute de conception surtout pour les opérations du système socio-technique	Pressions directes ou indirectes sur le patient		Risque de divulgation à des tiers non autorisés
Faute de conception ou de développement	Possibilité, pour un assureur possédant un simple lecteur de carte, de visualiser certaines informations sans carte CPS (Carte de Professionnel de Santé).	Certaines informations (par exemple, les informations nécessaires en cas d'urgence) restent de libre accès.	Risque de divulgation d'informations compromettantes pour l'individu comme certaines maladies invalidentes.
Faute de conception ou de développement	Interconnexion entre fichiers fiscaux et médicaux par le NIR.	Utilisation du NIR comme identifiant dans différentes bases de données.	Perte des libertés individuelles
Logique maligne	Vol de données	Manque de protection.	Utilisation délictueuse de ces données.
Faute d'interaction et parfois de conception.	Accès non intentionnel.		Divulgation non intentionnelle de ces données.
Pour gérer un système informatique, il est nécessaire d'avoir un administrateur de la base. Logique maligne.	Administrateur non digne de confiance : les droits de l'administrateur peuvent dépasser largement son besoin.	Manque de restrictions d'accès ou non respect du principe du moindre privilège.	L'administrateur peut accéder à des informations qu'il n'a pas à connaître.
Faute de conception	Chevaux de Troie.	Utilisation d'un logiciel piégé.	Transmission de données sensibles vers l'extérieur, via le logiciel piégé.
Mise en réseau des systèmes médicaux ou système médical sur réseau local relié à Internet. Logiques malignes	Risque d'attaque de l'extérieur ; risques d'usurpations d'identité	Réseau local pas assez protégé.	Atteintes aux propriétés de sécurité (divulgation ou utilisation illicite d'informations sensibles etc.).

Origine du problème	*Menace*	*Vulnérabilité*	*Conséquence*
Logiques malignes ; Fautes de conception	Interception possible des messages échangés par courrier électronique.	Protection insuffisante du réseau.	Atteinte à la confidentialité et à l'intégrité
Stockage de la clé privée d'un médecin sur un disque dur lors de sa génération par une société commerciale en ligne. Logiques malignes ; fautes de conception.	Risque de vol ou piratage de cette clé privée.	Stockage non sécurisé de la clé.	Possibilité de lire les messages dont le médecin est destinataire ou émetteur (résultats laboratoires, avis d'autres médecins sur un certain cas, etc.).
Utilisation d'un algorithme de chiffrement non publié et donc peut-être non vérifié comme solide.	Possibilité de déchiffrement ou décryptage par des personnes non autorisées.	Algorithme vulnérable ou non robuste.	Possibilité de lire les messages dont le médecin est destinataire ou émetteur.
Longueur insuffisante des clés. Logiques malignes ; Fautes de conception.	Possibilité de décryptage par des personnes non autorisées.	Longueur insuffisante des clés.	Violation de la confidentialité et de l'intégrité.
Stockage de dossiers médicaux centralisés sur sites web. Logiques malignes.	Attaques possibles de ces sites par des visiteurs (par exemple, en utilisant les traitements invisibles avec Html).	Architecture centralisé de dossiers médicaux	
Stockage éclaté des dossiers médicaux sur les sites de diverses institutions. Faute de conception	Utilisation de ces informations par un médecin outrepassant sa fonction ou son besoin.		Divulgation non autorisée des informations situées sur ces dossiers.

A2. Menaces pouvant porter atteinte à l'intégrité

Origine	*Menace*	*Vulnérabilité*	*Conséquence*
Non intentionnel (exemple : erreur de saisie entre angine et angiome) ou malveillance. *Faute d'interaction*	Information introduite ou devenue fausse dans le système.	Absence de mécanisme de contrôle des données saisies.	Prise par le médecin de décisions erronées causant du tort au patient ; erreur de diagnostic ; etc.

Origine	*Menace*	*Vulnérabilité*	*Conséquence*
Rétention délibérée, par le patient, d'informations le concernant relative à son état de santé ou à sa situation sociale, en vertu du principe du droit à l'oubli (antécédent pathologique, épisode socio-sanitaire antérieur). *Faute d'interaction humaine.*	Peur (par le patient) de violation de son intimité socio-sanitaire ; perte de confiance en ce qui concerne la capacité du système d'information médical ou social à garantir la confidentialité des informations sensibles confiées par lui.	Absence dans le système de certaines informations pourtant nécessaires d'un point de vue médical.	Risque direct de non mise à disposition par le patient d'informations sur son état de santé ou, par l'assuré de la précarité de sa situation sociale : informations pertinentes, dont peut en juger un médecin ou une assistante sociale ; risque indirect de manque de pertinence du protocole de soins, des traitements ou de la prise en charge sociale.
Non intentionnel	Risque possible de mauvais diagnostic, ordonnances, traitements.	Erreurs de conception ou de programmation des logiciels	
Faute de conception ou de développement ; fautes d'interaction ; logique maligne.	Suppression illégitime de données.	Insuffisance de la politique d'autorisation.	Possibilité pour un utilisateur d'effacer des dossiers montrant des négligences ou escroqueries.
Non intentionnel. *Faute d'interaction.*	Manipulations erronées des données ou des logiciels.	Absence de contrôles	Utilisation erronée des données, des résultats, des logiciels ; erreurs de diagnostic, etc.
Faute de conception ; logique maligne.	Vol de données		Utilisation erronée ou impossible des informations.
Non intentionnel ou malveillance	Feu, inondation, détérioration, etc.	Stockage du matériel non protégé physiquement.	Risque de destruction des informations.
Non intentionnel (exemple : patient âgé ou perturbé) ou malveillance (usurpation d'identité)	Mauvaise identification du patient.	Absence de contrôles (notamment de l'identification et de l'authentification).	Attribution erronée de diagnostics, attribution erronée ou illicite de soins ou de remboursements.
Mise en réseau des systèmes médicaux.	Plus grand risque d'attaques de l'extérieur	Manque de protection vis-à-vis des attaques extérieures.	Atteintes aux propriétés de sécurité

Origine	*Menace*	*Vulnérabilité*	*Conséquence*
Logique maligne.	Virus destinés à altérer malicieusement les dossiers médicaux.	Manque de protections contre les logiques malignes.	Altération des informations du dossier
Faute de conception ; logique maligne.	Interception, modification des courriers électroniques.	Logiciels de courrier électronique non sécurisés	Altération des informations contenues dans les messages
Stockage de dossiers médicaux centralisés sur sites web.	Attaques ou intrusions possibles de ces sites par des visiteurs.	Architecture centralisé non suffisamment sécurisées.	Atteinte à l'intégrité du SICSS visité ; prise de contrôle de l'ordinateur du visiteur, par le concepteur du site visité ; etc.

A3. Menaces pouvant porter atteinte à la disponibilité

Origine	*Menace*	*Vulnérabilité*	*Conséquence*
Informatique ou télématique. *Logique maligne ; Faute de conception.*	Défaillance informatique ou sabotage ou attaque entraînant une information non disponible dans le système.	Possibilité de retarder ou causer un dysfonctionnement du système.	Prise par le médecin de décisions erronées causant du tort au patient ; erreur de diagnostic ; impossibilité d'utiliser un dossier informatique comme preuve.
Stockage de dossiers médicaux centralisés sur sites web. *Logique maligne*	Attaques ou intrusions possibles de ces sites par des visiteurs ; etc.	Architecture centralisé non suffisamment sécurisée.	Atteinte à la disponibilité du SICSS visité ; prise de contrôle de l'ordinateur, ou attaque du disque dur du visiteur.
Mise en place de de mécanismes d'identification. *Faute d'interaction ; Faute de conception ou de développement*	Décès du médecin ; perte du code et donc risque de discontinuité des soins sur le patient.	Mise en place d'un mécanisme d'authentification n'ayant pas prévu ce cas.	Perte des données des patients.
Mise en réseau des systèmes médicaux. *Logique maligne*	Attaques de l'extérieur (DDoS)	Risques d'intrusions extérieurs (protection insuffisante).	Atteintes aux propriétés de sécurité.

Peur du manque de confidentialité. *Crainte d'une faute de conception*	Impossibilité d'échange d'informations entre les institutions de santé.	Chaque institution installe son propre système de confidentialité	Risque de non-interopérabilité.

A4. Menaces pouvant porter atteinte à l'auditabilité

Origine	*Menace*	*Vulnérabilité*	*Conséquence*
Contexte d'urgence Logique maligne	Abus de pouvoir ; Utilisation abusive de droits valables en contexte d'urgence.	Non respect du principe du moindre privilège ; politique de sécurité non adéquate.	
Pour gérer un système informatique, il est nécessaire d'avoir un administrateur de la base. *Logique maligne*	Administrateur non digne de confiance qui abuse de ses pouvoirs.	Manque de traçabilité et de procédures dissuasives.	Divulgation ou altération de donnéespersonnelles.
Non consentement du patient à faire figurer des informations médicales le concernant dans son dossier médical ; défaut d'authentification d'une intervention sur le dossier médical ; défaut d'authenticité d'une modification du dossier médical ; défaut d'intégrité d'un dossier médical ;	Information manquante dans le système, ou devenue fausse suite à une malveillance (intrusion, accès trop libre, etc.)	Absence de trace ; absence d'imputation ; défaut d'opposabilité ; absence de signature électronique ; absence de justification authentique rendant probante la nécessité de : *(i)* décider d'un protocole de soins particulier à appliquer, *(ii)* établir une prescription particulière et atypique, *(iii)* omettre certains médicaments habituellement recommandés en pareil situation (ici, à cause d'allergies non mentionnées, ou d'antécédent non rappelés,…), etc.	Impossibilité d'utiliser un dossier médical (ou social, par analogie) informatisé comme preuve pour justifier une décision, une attitude ou une action pour un professionnel de santé (ou par un assistant social, par analogie).

Annexe B : Anonymisation des données du PMSI

Dans le cadre du PMSI (Programme de Médicalisation des Systèmes d'Information), tout établissement de santé rend compte de son activité au moyen de résumés de séjour (RSS) pour les hospitalisations de court-séjour MCO, au moyen de résumés hebdomadaires (RHS) pour les séjours en soins de suite ou de réadaptation. Le chaînage des séjours est le complément nécessaire du dispositif de recueil d'informations. Il a deux rôles essentiels :

- la validation de la qualité du codage des informations fournies ;
- la réalisation d'analyses pertinentes des bases de données régionales et nationales.

Comme indiqué sur le schéma récapitulatif de la figure B1, la procédure se développe en deux étapes :

- le numéro "*de sécurité sociale*" est anonymisé et couplé avec un numéro administratif. Le bureau des admissions (ou des frais de séjour) crée (en utilisant le logiciel MAGIC, "*Module d'anonymisation et de gestion des informations de chaînage*"), à partir du fichier des "numéros de sécurité sociale" des patients admis, un fichier de "numéros anonymes". Le fichier obtenu devra être couplé au fichier des "numéros administratifs" conférés à ces patients lors de leur admission. Le résultat de cette procédure est la production d'un fichier dénommé ANO-HOSP ;
- un second couplage est effectué par le département d'information médicale (DIM) : le médecin responsable du DIM met en relation le fichier ANO-HOSP, reçu du service des admissions, et le fichier élaboré par ses soins, qui opère la liaison entre le numéro de RSS et le numéro administratif.

Cette protection d'anonymisation des résumés est renforcée, au niveau de l'ARH (Agence Régionale d'Hospitalisation), d'une nouvelle procédure de hachage des numéros anonymes.

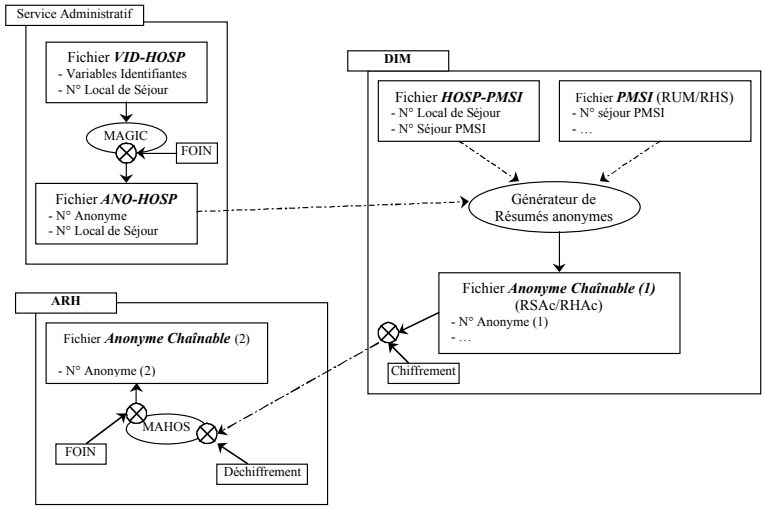


Figure B1 : Schéma récapitulatif des anonymisations du PMSI

B1. Traitements réalisés au niveau des services administratifs

B1.1 Constitution du fichier VID-HOSP

Ce fichier est réalisé par les services administratifs de l'établissement. Il est constitué des variables suivantes :

- numéro de sécurité sociale (celui de l'ouvrant droit), celui-ci est recueilli au niveau du service des admissions, afin de remplir les informations de séjours transmises aux caisses d'assurance maladie (ou pour facturation) ;
- date de naissance et sexe de la personne admise et non celui de l'assuré ;
- variable identifiant le séjour hospitalier (numéro administratif local de séjour qui n'est rien d'autre que le numéro fourni par le service des admissions de l'établissement pour identifier le séjour du patient).

B1.2 Constitution du fichier ANO-HOSP et transmission au DIM

Le logiciel MAGIC constitue à partir du fichier VID-HOSP, le fichier ANO-HOSP contenant :

- un numéro de patient anonymisé, celui-ci est constitué par le module "FOIN" ;
- un numéro local de séjour.

Ce fichier (ANO-HOSP) devra être transmis au DIM de l'établissement. MAGIC effectue les contrôles suivants :

- conformité des différentes variables ;
- absence de doublon sur le numéro d'hospitalisation dans le fichier VID-HOSP.

Une clé d'intégrité sera générée pour s'assurer qu'aucune modification du fichier ANO-HOSP n'aura été réalisée entre sa génération et son utilisation.

B2. Traitements réalisés au niveau du DIM

B2.1 Constitution du fichier HOSP-PMSI

Le DIM réalise un fichier contenant les correspondances entre le numéro de séjour PMSI (numéro de RSS, numéro séjour SSR) et le numéro d'identification de séjour administratif. Ce fichier est en général, déjà utilisé par les DIM pour permettre de faire la liaison entre les données saisies pour le PMSI et le dossier médical.

B2.2 Constitution du fichier anonyme chaînable

Cette étape produit un fichier anonyme contenant un identifiant permettant le chaînage des séjours pour un patient donné sur l'ensemble de la France. Un couplage en cascade s'effectue tout d'abord entre les fichiers ANO-HOSP et HOSP-PMSI puis entre les fichiers de données PMSI et le fichier HOSP-PMSI. Le premier couplage s'effectue sur la variable "numéro local de séjour" le second sur la variable "numéro séjour PMSI".

Lors de l'étape de couplage, le logiciel d'anonymisation réalise les contrôles suivants :

- contrôle des variables du fichier HOSP-PMSI ;

- vérification de l'absence de doublons sur le numéro d'hospitalisation dans le fichier HOSP-PMSI ;
- vérification que les numéros PMSI présents dans HOSP-PMSI sont bien dans le fichier PMSI et vice-versa ;
- vérification que les numéros locaux de séjours présents dans le fichier ANO-HOSP sont bien dans HOSP-RSS.

B2.3 Traitements réalisés au niveau de l'ARH

Lors de la réception des fichiers par l'ARH (Agence Régionale d'Hospitalisation), les étapes suivantes seront réalisées :

- déchiffrement du fichier ;
- réalisation d'une deuxième anonymisation du numéro (anonyme), crée dans l'établissement ;
- traitements MAHOS traditionnels.

La deuxième anonymisation est nécessaire pour que les numéros calculés dans l'établissement ne soient pas ceux utilisés dans les bases nationales.

Annexe C : Introduction à UML

Cette annexe a pour objectif de présenter de manière synthétique la notation UML. Elle se décompose en deux parties, la première propose un résumé d'UML, la seconde présente les différents diagrammes UML utilisés dans le chapitre 4.

C1. UML en résumé

UML a été standarisé par l'OMG (*Object Management Group*) en novembre 1997. Les *objectifs* d'UML sont :

- proposer aux utilisateurs un langage visuel près à l'emploi permettant de développer et d'échanger des modèles orienté-objet adaptés à la modélisation des systèmes informatiques ;
- fournir des mécanismes d'extension et de spécialisation des concepts de base afin de s'adapter à tout les types d'applications informatiques ;
- permettre des spécifications indépendantes d'un langage de programmation ou d'un processus de développement donné ;
- offrir une base sémantique formelle pour la compréhension du modèle ;
- encourager l'utilisation et le développement d'outils de génie-logiciel.

UML est constitué de neuf modèles semi-formels, appelés diagrammes. Il est complété par un langage de définition de contrainte, l'OCL (*Object Constraint Language*). Ce dernier permet de définir des contraintes entre les différents éléments de la notation (attributs, classes, objets, etc.).

Les différents *modèles* d'UML sont :

- *les modèles fonctionnels* (diagramme de cas d'utilisation), permettent de capturer les besoins fonctionnels ;
- *les modèles structuraux* (diagramme de classes et diagramme d'objets),décrivent l'aspect statique des objets,leurs structures, leurs attributs,leurs interfaces ainsi que leurs relations statiques au sein du système ;
- *les modèles comportementaux* (digrammes de séquences, de collaboration, d'activités et d'états) mettent l'accent sur le rôle des objets dans le système : leurs méthodes (services qu'ils rendent), interactions, collaborations avec les autres objets ainsi que leurs comportement au cours du temps ;
- *les modèles d'implémentation* (diagramme de composant et de déploiement) permettent de décrire une architecture matérielle et de définir l'architecture physique. Ils représentent les relations entre logiciel et matériel.

C2. Les diagrammes UML

C2.1 Les cas d'utilisation

Les cas d'utilisation décrivent les fonctionnalités du système (ou d'un sous-système) d'un point de vu d'un utilisateur, et recentre l'expression des besoins sur ce dernier. Les cas d'utilisation sont donc utiles pour représenter ce que doit faire un système par rapport à son environnement. Les entités de l'environnement extérieures au système sont appelées acteurs. Un diagramme de cas d'utilisation représente un ou plusieurs cas d'utilisation, les acteurs impliqués dans ces fonctionnalités ainsi que les relations entre les acteurs et les cas d'utilisation.

Un cas d'utilisation est représenté par un identificateur textuel dans une ellipse un acteur par une icône stéréotypée de personnage et les relations par des traits. Les relations peuvent être des associations entre un acteur et un cas d'utilisation, des généralisations entre les acteurs, et des généralisations, extensions et inclusions entre cas d'utilisation.

Les associations faisant intervenir un acteur et un cas d'utilisation sont en trait plein et non directionnel. La généralisation entre acteur est indiquée par une flèche triangulaire fermée, orientée vers l'acteur le plus général. Une généralisation entre cas d'utilisation est indiquée par une flèche triangulaire fermée, orientée vers le cas d'utilisation le plus général. Les autres associations entre cas d'utilisation (extension et inclusion) sont en pointillés, avec une flèche et précisés par les stéréotypes <<*extend*>> ou <<*include*>> :

- A <<*extend*>> B signifie que le comportement de A peut être étendu et que B peut faire appel au comportement de A.
- A <<*include*>> B signifie que le comportement de B est toujours présent dans A.

C2.2 Les modèles structuraux

Les *diagrammes de classes* et *d'objets* représentent la structure statique d'un système.

Dans le cas général, une classe est représentée par un rectangle divisé en trois compartiments. La case supérieure contient le nom, la case médiane contient les attributs et enfin la case inférieur contient les opérations de classe.

Par convention la notation "*O* /*R* :*C* " désigne un objet appelé *O*, instance de la classe *C*, ayant pour rôle *R*. les arguments *O* et /*R* sont optionnels. Une autre façon de différencier objet et classe est de souligner le nom de l'objet dans ce cas, seul le nom de classe dont il est instance est donné.

Des relations (ou associations) peuvent être décrites entre classes. Les relations entre objet sont des instances d'association et sont appelées lien. Une relation est représentée par un trait entre deux classes. Elle est caractérisée par un nom et/ou deux rôles (un pour chaque classe associée).Une indication de cardinalité peut être rajoutée près des deux rôle de l'association. Les associations les plus importantes sont celles qui représentent une agrégation ou une généralisation.

L'*agrégation* est une association non symétrique spécifiant qu'une classe est composée d'autres classes (classes composites). Elle est représentée par un losange vide du côté de l'agrégat. La composition est une agrégation forte, pour laquelle la destruction de l'objet composé entraîne celle des objets composites. Dans ce cas, le losange est noir.

L'*héritage* (ou généralisation) se représente par une flèche triangulaire vide pointant vers le père.

C2.3 Les modèles comportementaux

Un diagramme d'interactions exprime le comportement qui résulte d'une communication d'un groupe d'instances. Cet ensemble d'interactions peut être organisé autour d'un cas d'utilisation, d'une ou plusieurs opérations réalisées par plusieurs objets. Le but est de décrire comment les objets collaborent au cours du temps et quelles responsabilités ils assument.

Il existe deux type diagrammes d'interaction proposant chacun une présentation différente :

- *Les diagrammes de séquence* représentent des interactions entre objets, en instant sur la chronologie des envois de messages. Les objets intervenant dans l'interaction sont matérialisés par une ligne de vie, et les messages échangés au cours du temps sont mentionnés sous une forme textuelle. En UML, les interactions et les informations échangées sont appelés messages. Un message est constitué par l'émission par le demandeur de services d'un événement, et par la réception de cet événement par le fournisseur de services. L'émission d'un événement se fait par une action.
- *Les diagrammes de collaboration* montrent les relations entre les objets et sont préférables pour comprendre la responsabilité de chaque objet dans le contexte de l'interaction décrite. En revanche, ils ne montrent pas le temps sur une échelle séparée. En général un diagramme de collaboration est utilisé comme canevas pour décrire un ensemble de diagramme de séquence, chaque diagramme de séquence étant une histoire possible.

Par ailleurs, *les diagrammes des états* peuvent être utilisés pour modéliser le comportement des objets, des classes, des sous systèmes ou des acteurs[39]. Comme toutes les approches de type "système de transition", la notion d'état et de transition en sont le fondement. L'*état* représente l'ensemble des valeurs des variables du système à un instant donné. Le changement d'état est représenté par une *transition*, c'est à dire une modification d'une ou de plusieurs variables du système. Une transition peut être provoquée, soit par un événement interne (expiration d'un délai ou fin d'une activité, par exemple), soit par un événement externe (type de stimulus). Elle peut aussi correspondre à une action du système sur l'environnement (type de réponse). Une transition peut contenir des paramètres. La syntaxe est la suivante :

Nom-Evènement (*Paramètre_Evènement*) [*Garde*]/*Actions*.

Chacun de ces champs est optionnel. Des données, alors paramètres, peuvent être associées à un événement. La transition est franchie dès que l'événement survient, et que la garde (expression logique booléenne) est vraie. Lors de franchissement, des actions peuvent être effectuées. Chaque action est séparée par une virgule,les actions sont alors exécutées en séquence. Une action peut être l'appel d'une opération, l'envoi d'un signal, la création ou la destruction d'une instance, l'envoi des valeurs d'une variable, ou enfin l'arrêt d'une opération.

Des actions peuvent être aussi associées aux états. Leur définition se fait suivant la notation "*Nom_Action* /. *Liste_Action*". Pour un état, trois mots clef sont réservés au nom d'une action :

- "Entry / Liste_Action" qui définit les actions que l'objet doit effectuer dès qu'il atteint l'état,
- "Exit / Liste_Action" pour décrire les actions à effectuer lorsque l'objet sort de cet état,
- "Do / Liste_Action" qui définit la liste des actions effectuées tant que l'objet reste dans l'état auquel est associé cette action.

En outre, un état *historique*, noté H, permet de réintégrer le dernier sous-état quitté lorsqu'une transition portant sur un état composite déjà atteint auparavant est sensibilisée.

[39] En fait, tout les objets n'ont pas besoin d'un diagramme des états, seuls les objets actifs en nécessitent un. Un objet actif est un objet dont les opérations sont contraintes par son état interne.

Cette possibilité autorise la représentation de la notion de reprise après une interruption. L'état *historique**, noté H*, permet de réintégrer l'état de plus bas niveau dans la hiérarchie de décomposition.

Enfin, des diagrammes d'activités, variante des diagrammes d'états-transitions, peuvent compléter la modélisation du comportement du système et de ses entités. Dans un diagramme d'états-transitions, les états et les transitions sont mis en avant alors que dans un diagramme d'activités, ce sont les activités et les transitions qui sont mises en avant. Les deux types de diagrammes permettent ainsi d'avoir deux vues différentes sur les automates donnés. Un diagramme d'activité visualise un graphe d'activités qui modélise le comportement interne d'une méthode (la réalisation d'une opération), d'un cas d'utilisation ou plus généralement d'un processus.

Dans ce type de diagrammes, un état-action modélise une étape dans l'exécution d'un algorithme ou d'un workflow. Elle est représentée par un triangle arrondi, comme un état, mais plus étiré horizontalement avec des côtes convexes à droite et à gauche. Les états actions sont reliés par des transitions, souvent automatiques, représentés par des flèches.

Étant donné que le travail décrit dans ce mémoire n'utilise pas les modèles d'implémentation d'UML, nous n'allons pas détailler les diagrammes de composants ni les diagrammes de déploiement. Une description plus riche d'UML peut être trouvée dans [Muller & Gaertner 2000 ; Booch *et al*. 1999].

Annexe D : Contrôle d'accès pour un centre dentaire

Il est évident que lors de tout projet, la rédaction du code de programme est obligatoirement devancée d'une méthode d'analyse fournissant une vue détaillée des problèmes à traiter. La méthode "MERISE", que nous avons adoptée lors de la réalisation de notre application, est une méthode de conception et de modélisation des systèmes d'information qui repose sur une architecture en trois cycles d'abstraction [Matheron 1998] :

- *Niveau conceptuel* : définit les fonctions réalisées par l'entreprise (en l'occurrence le centre dentaire). Il répond à la question : "quoi ?" et aide ainsi à comprendre le fonctionnement et à modéliser les données, les traitements et les communications.
- *Niveau logique* (*ou organisationnel*) : complète l'analyse en décrivant les fonctions des acteurs, "qui ? quand ? où ?", ainsi que les processus de l'organisation "qui fait quel traitement, quand et où ?".
- *Niveau physique* (*ou opérationnel*) : finalise le travail en indiquant comment les données et les programmes sont implémentés et réalisés.

Dans le reste de ce chapitre, nous détaillons chacun de ces niveaux.

D1. Analyse conceptuelle

L'analyse conceptuelle que nous avons établie pour notre application consiste à fournir le dictionnaire de données, les règles de gestion, ainsi que les trois modèles conceptuels : le modèle conceptuel de communication, le modèle conceptuel de données et le modèle conceptuel de traitement.

D1.1 Dictionnaire de données

L'analyse conceptuelle commence par le regroupement des données recensées dans ce qui est communément nommé un "dictionnaire de données". L'annexe E présente le dictionnaire correspondant à notre application.

Code	*Désignation*	*Type*	*Nature*	*Longueur*
Ndossier	Numéro dossier du patient	Numérique (N)	Élémentaire (E)	6
Nmpatient	Nom patient	Alphanumérique (AN)	E	10
Prpatient	Prénom patient	A	E	10
Fonctpatient	Fonction patient	A	E	10
NSS	Numéro sécurité sociale du patient	N	E	15
Dnpatient	Date naissance patient	N	E	10
Adpatient	Adresse patient	AN	Concaténé	30

			(Co)	
Telbpatient	Tel bureau	N	E	10
Teldpatient	Tel domicile patient	N	E	10
Telppatient	Tel portable patient	N	E	10
Emailpatient	Adresse électronique patient	AN	E	20
Diagpatient	Diagnostic du patient	AN	E	30
Interpatient	Interrogatoires du patient	AN	E	30
Ndentiste	Numéro dentiste	N	E	2
Nmdentiste	Nom dentiste	A	E	10
Prdentiste	Prénom dentiste	A	E	10
Spdentiste	Spécialité dentiste	A	E	10
Ncindentiste	Numéro carte d'identité du dentiste	AN	E	8
Sexedentiste	Sexe dentiste	A	E	9
Dndentiste	Date naissance dentiste	AN	E	12
Dtembdentiste	Date d'embauche dentiste	AN	E	12
Ancdentiste	Ancienneté dentiste	N	E	2
Addentiste	Adresse dentiste	AN	Co	30
Telbdentiste	Téléphone bureau dentiste	N	E	10
Tepdentiste	Téléphone portable dentiste	N	E	10
Faxdentiste	Fax dentiste	N	E	10
Emaildentiste	Adresse électronique dentiste	AN	E	20
Ntechnicien	Numéro technicien	N	E	2
Nmtechnicien	Nom technicien	A	E	10
Prtechnicien	Prénom technicien	A	E	10
Sexetechnicien	Sexe technicien	A	E	9
Ncintechnicien	Numéro carte identité. technicien	AN	E	8
Dntechnicien	Date de naissance technicien	AN	Co	12
Dtembtechnicien	Date d'embauche	AN	E	12
anctechnicien	Ancienneté technicien	N	E	2
Teldtechnicien	Téléphone domicile technicien	N	E	10
Telptechnicien	Téléphone portable technicien	N	E	10
Emailtechnicien	Adresse électronique technicien	AN	E	20

faxtechnicien	Fax technicien	N	E	9
Nrdv	Numéro rendez-vous	N	E	2
Dtrdv	Date rendez-vous	AN	E	12
Hrdv	Heure rendez-vous	N	E	4
Obrdv	Observation rendez-vous	AN	E	30
Ninter	Numéro intervention	N	E	10
Dentsoignée	Dent soignée	A	E	3
Dtinter	Date Intervention	AN	E	12
Hinter	Heure Intervention	N	E	4
Mtcons	Motif de consultation	A N	E	30
Pltrait	Plan de traitement	AN	E	30
Nord	Numéro ordonnance	N	E	10
Dtord	Date ordonnance	AN	E	12
Nfact	Numéro facture	N	E	10
Dtfact	Date facture	AN	E	12
Libfact	Libelle facture	A	E	20
Mopay	Mode paiement	A	E	8
Mtfact	Montant facturé	N	E	5
Avfact	Avance de la facture	N	E	5
Recfact	À recevoir de la facture	N	E	5
Npro	Numéro prothèse	N	E	2
Libpro	Libellé prothèse	A	E	10
Typro	Type prothèse	A	E	10
Ppro	Prix prothèse	N	E	5
Typesoin	Type soin effectué	A	E	15
Intypesoin	Intitulé type soin	A	E	20
Libradio	Libellé radio	A	E	10
Pradio	Prix radio	N	E	3
Codmed	Code médicament	N	E	3
Libmed	Libellé médicament	A	E	10
Formemed	Forme médicament	A	E	10
Codetymed	Code type médicament	N	E	2
Libtymed	Libellé type médicament	An	E	10

D1.2 Règles de gestion

- La deuxième étape de l'analyse conceptuelle consiste à identifier les règles de gestion associées à l'application. Les règles que nous avons implémenté sont les suivantes :

  chaque intervention est procédée d' un rendez-vous ;

chaque intervention produit un sous-ensemble d'actes de l'ensemble suivant :{ordonnance, facture, rendez-vous} ;

chaque facture correspond à une seule intervention ;

une intervention peut correspondre à zéro ou plusieurs soins dentaires, ou zéro ou plusieurs prothèses ;

un médicament appartient à un "type de médicament" ;

une ordonnance peut contenir un ou plusieurs médicaments ;

une intervention est faite par un et un seul dentiste ;

un cabinet doit contenir au moins un chirurgien dentiste, celui-ci peut être aidé par des assistantes médicales ;

la hiérarchie de rôles ainsi que les règles d'exclusion mutuelle évoquées précédemment ;

etc.

D1.3 Modèle conceptuel de communication

Le Modèle Conceptuel de Communication est un schéma qui représente les échanges de flux de produits, de personnes, de valeurs ou d'informations entre les intervenants internes (au sein du centre) et externes (patients). La figure D1 détaille les flux soulevés dans notre application.

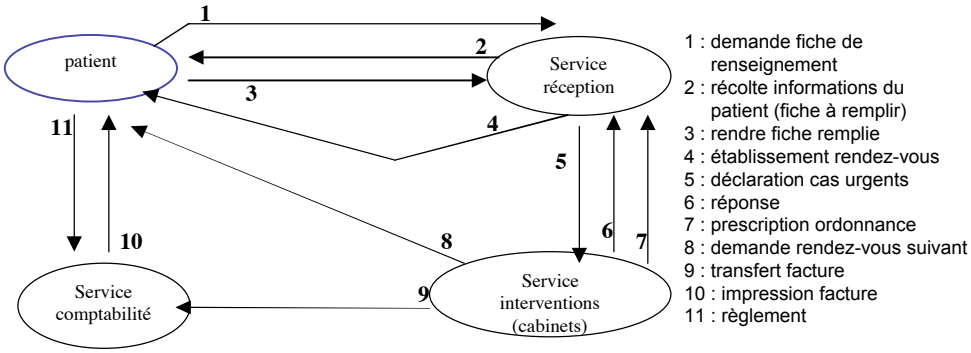


Figure D1 : Modèle conceptuel de communication de notre application.

Le graphe de dépendance fonctionnelle correspondant est donné dans la figure D2, utilisant les codes indiqués dans l'annexe E.

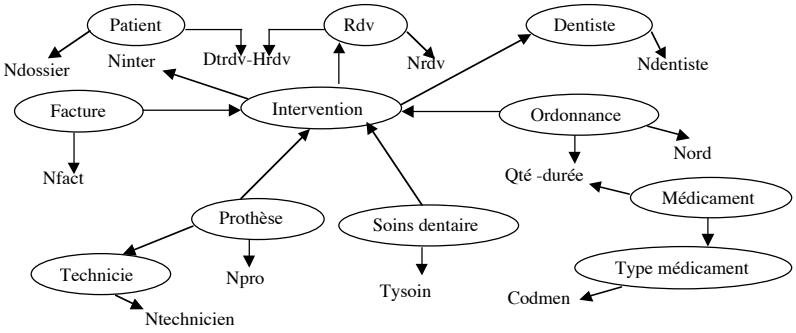


Figure D2 : Graphe de dépendance fonctionnelle correspondant à notre application.

D1.4 Modèle conceptuel de données

Le Modèle Conceptuel de Données (MCD) permet d'avoir une description statique de l'ensemble des données manipulées par l'organisation ainsi que les relations entre ces données (figure D3).

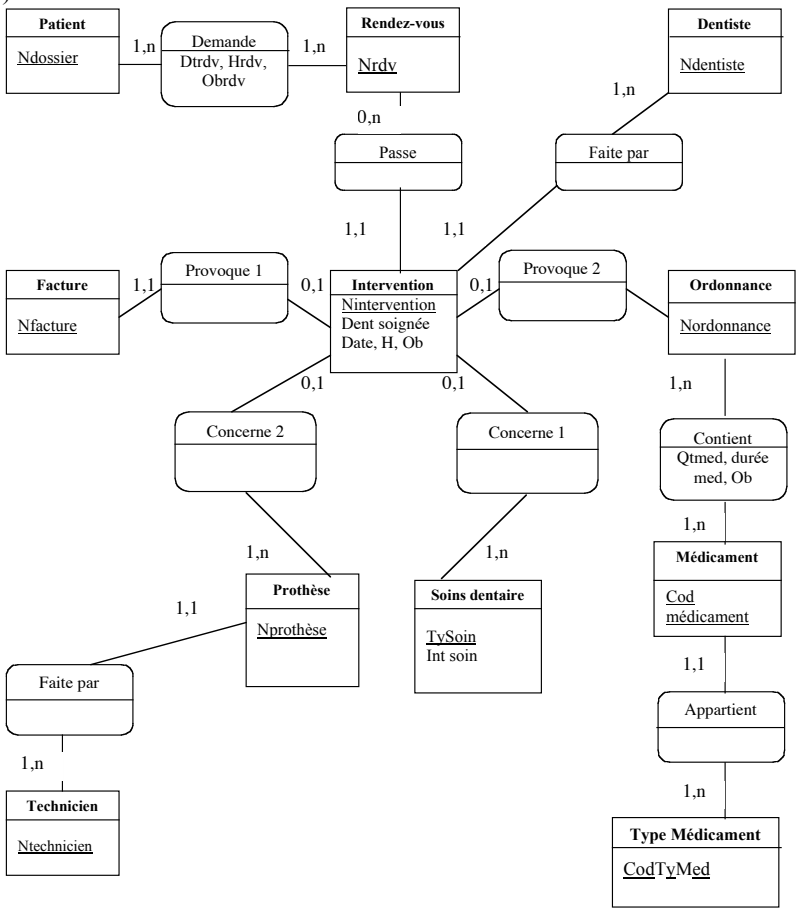


Figure D3: Modèle conceptuel de données correspondant à notre application.

D1.5 Modèle conceptuel de traitement

Le Modèle Conceptuel de Traitement (MCT) donne une description dynamique de l'organisation en décrivant les différents processus. Un processus étant une succession d'activités déclenchées par des événements et produisant des résultats. La figure D4 donne l'exemple d'un processus qui commence par l'arrivée du patient et qui s'achève par une intervention. Dans ce mémoire, et pour éviter toute lourdeur, nous n'allons pas détailler les processus spécifiques aux soins dentaires : extraction simple d'une dent ; extraction par alvéolectomie ; obturation à l'amalgame 1f, 2f ou 3f ; radio rétro alvéolaire, etc.

Après avoir présenté l'analyse conceptuelle correspondant à notre application, nous détaillons l'analyse logique (ou organisationnelle) ainsi que l'analyse physique (ou opérationnelle).

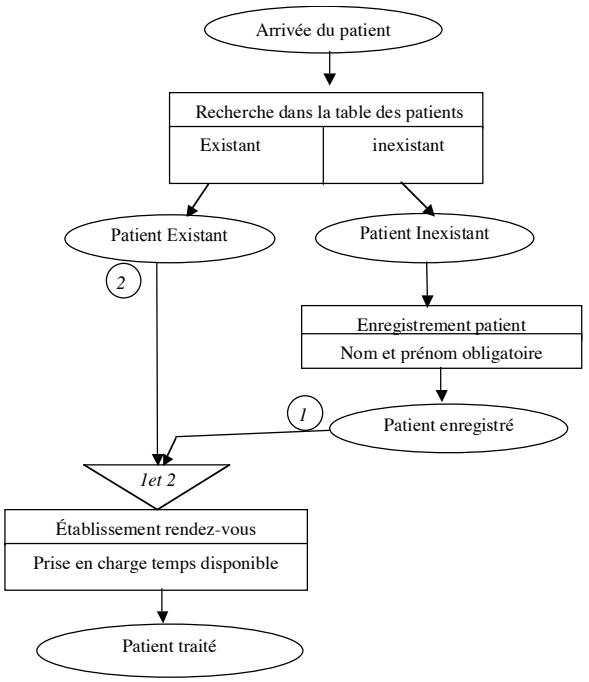


Figure D4 : Exemple de modèle conceptuel de traitement pour notre application.

D2. Analyse logique

Les modèles que nous venons de décrire sont indépendants des choix organisationnels (fichiers ou bases de données) et de programmation (langage). À l'inverse, le Modèle Logique de Données (MLD) est une représentation du modèle conceptuel en terme d'organisation de données, il traduit les données automatisées et leurs liens dans un formalisme plus proche du langage choisi pour la programmation. Le MLD qui correspond à notre choix d'implémentation est donné dans la figure D5.

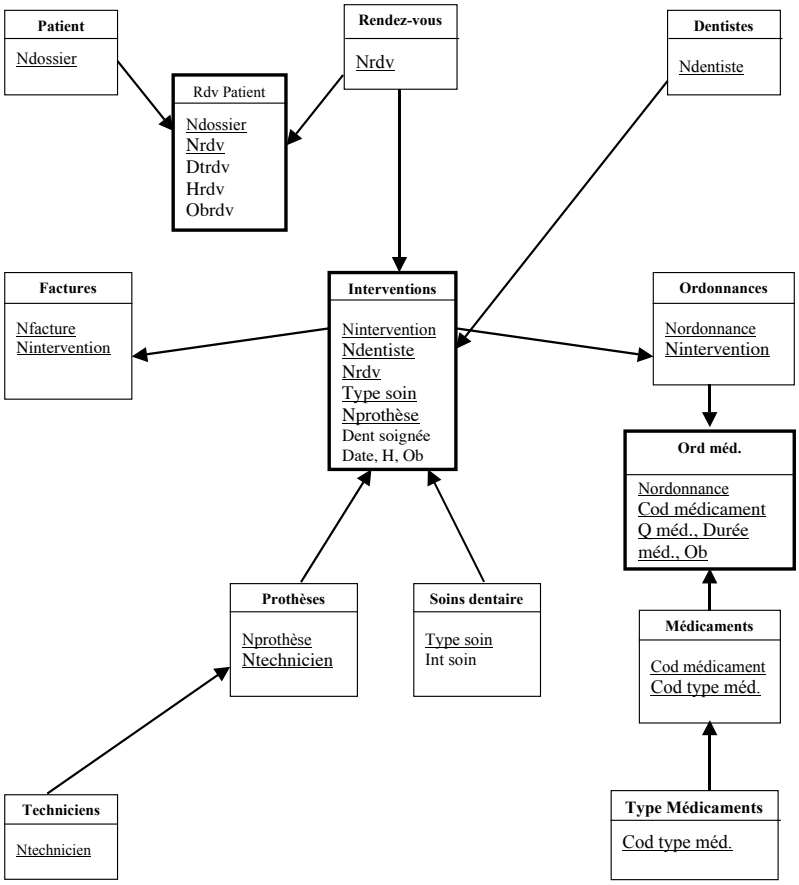


Figure D5 : Modèle logique de données pour notre application.

D3. Analyse physique

La dernière étape de notre analyse consiste à présenter la vue opérationnelle de notre application à travers un modèle physique de données. Celui-ci présente l'ensemble des données issues du modèle logique de données sous formes de tables. Quelques-unes des tables de notre implémentation sont présentées ci-dessous:

Table : patient				
Champ	*Description*	*Longueur*	*Type*	*Clé*
Ndossier	Numéro dossier	6	Numérique	Oui
Nmpatient	Nom patient	10	Texte	N
Prpatient	Prénom patient	10	Texte	N
Foncpatient	Fonction patient	9	Texte	N
NSSpatient	Numéro de sécurité sociale du patient	8	Texte	N
Dnpatient	Date naissance patient	12	Texte	N
Agepatient	Age du patient	2	Texte	N

Adpatient	Adresse du patient	30	Texte	N
Sexepatient	Sexe du patient	8	Texte	N
Telbpatient	Téléphone bureau du patient	10	Texte	N
Teldpatient	Téléphone domicile du patient	10	Texte	N
Telppatient	Téléphone portable du patient	10	Texte	N
Faxpatient	Fax du patient	10	Texte	N
Emailpatient	Email du patient	20	Mémo	N
Diagpatient	Diagnostic patient	30	Mémo	N
Interpatient	Interrogatoire patient	30	Mémo	N

Table : dentiste				
Champ	*Description*	*Longueur*	*Type*	*Clé*
Ndentiste	Numéro dentiste	2	Numérique	Oui
Nmdentiste	Nom dentiste	10	Texte	N
Prdentiste	Prénom dentiste	10	Texte	N
Spdentiste	Spécialité dentiste	10	Texte	N
Ncindentiste	Numéro CIN dentiste	8	Texte	N
Sexedentiste	Sexe dentiste	9	Texte	N
Dndentiste	Date naissance	12	Texte	N
Agedentiste	Age dentiste	2	Texte	N
Dtembdentiste	Date d'embauche	12	Texte	N
Ancdentiste	Ancienneté dentiste	2	Texte	N
Addentiste	Adresse dentiste	30	mémo	N
Telddentiste	Téléphone bureau du dentiste	10	Texte	N
Telpdentiste	Téléphone portable du dentiste	10	Texte	N
Faxdentiste	Fax dentiste	10	Texte	N
Emaildentiste	mél dentiste	30	Mémo	N

Table : rendez-vous				
Champ	*Description*	*Longueur*	*Type*	*Clé*
Nrdv	Numéro rendez-vous	2	Numérique	Oui
Dtrdv	Date rendez-vous	12	Texte	N
Hrdv	Heure rendez-vous	4	Texte	N
Obrdv	Observation rendez-vous	30	mémo	N

Table intervention				
Champ	*Description*	*Longueur*	*Type*	*Clé*
Ninter	Numéro intervention	6	Numérique	Oui

Dentsoignée	Dentsoignée	3	Mémo	N
Dtinter	Date intervention	12	Texte	N
Hinter	Heure intervention	4	Texte	N
Mtcons	Motif de consultation	30	mémo	N
Pltrait	Plan de traitement	30	Mémo	N

Table : facture				
Champ	*Description*	*Longueur*	*Type*	*Clé*
Nfact	N° facture	6	Numérique	Oui
Ninter	N° intervention	6	Numérique	N
Dtfact	Date facture	12	Texte	N
Libfact	Libellé facture	10	Texte	N
Mopay	Mode paiement	10	Texte	N
Mtfact	Montant facturé	5	Numérique	N
Avfact	Avance de la facture	5	Numérique	N
Recfact	À recevoir	5	Numérique	N

Table : ordmed				
Champ	**Description**	**Longueur**	**Type**	**Clé**
Nord	N° ordonnance	5	Numérique	Oui
Codmed	Code médicament	5	Texte	Oui
Qtmed	Quantité médicament	10	Texte	N
Dureemed	Durée médicament	2	Numérique	N

Références bibliographiques

[Abou El Kalam *2002*] A. Abou El Kalam, "Politiques de sécurité pour les systèmes d'informations médicales", *Cinquièmes Journées Doctorales en Informatiques et Réseaux* (*JDIR* 2002), Toulouse, 4-6 mars 2002, pp. 201-210.

[Abou El Kalam *et al*. 2002a] A. Abou El Kalam, F. Cuppens, Y. Deswarte, L. Merrouche, C. Saurel, G. Trouessin, *Modèles et politiques de sécurité des systèmes d'informations et de communications en santé et en social : Etat des lieux scientifico-technologique*, Rapport LAAS n° 02091, mars 2002, 32 pp.

[Abou El Kalam et al. 2002b] A. Abou El Kalam, Y. Deswarte, D. Powell, *Modèles et politiques de sécurité des systèmes d'informations et de communications en santé et en social : concepts et terminologie génériques*, Rapport LAAS n° 02268, juillet 2002, 22 pp.

[Abou El Kalam *et al*. 2002c] A. Abou El Kalam, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, R. El Baida, F. Nambot, C. Saurel, G. Trouessin, *Modèles et politiques de sécurité des systèmes d'informations et de communications en santé et en social : informations à protéger et menaces*, Rapport LAAS n° 02334, septembre 2002, 34 pp.

[Abou El Kalam & Deswarte 2002] A. Abou El Kalam, Y. Deswarte, "Contrôle d'accès basé sur les rôles, les groupes d'objets et le contexte : Étude de cas dans les systèmes d'information et de communication en santé", *Actes de la conférence Sécurité et Architecture des Réseaux (SAR'02)*, Marrakech, 8-12 juillet 2002, 11pp.

[Abou El Kalam & Deswarte 2003a] A. Abou El Kalam, Y. Deswarte, "Security model for Health Care Computing and Communication Systems", *18th International Information Security Conference* (*IFIP SEC* 2003), Athènes, 26-28 mai 2003, Kluwer Academic Publishers, pp. 277-288.

[Abou El Kalam *et al*. 2003a] A. Abou El Kalam, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, R. El-Baida, A. Miège, C. Saurel, G. Trouessin "Organization-Based Access Control", *4th International Workshop on Policies for Distributed Systems and Networks (Policy'03)*, Côme, Italie, 4-6 juin 2003, IEEE Computer Society Press, pp. 120-131.

[Abou El Kalam *et al*. 2003b] A. Abou El Kalam, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, R. El-Baida, A. Miège, C. Saurel, G. Trouessin "Un modèle de contrôle d'accès basé sur les organisations", *Cahiers francophones de la recherche en sécurité de l'information*, n° 2, Université de Montpellier I, premier trimestre 2003, pp. 30-43.

[Abou El Kalam *et al*. 2003c] A. Abou El Kalam, Y. Deswarte, G ; Trouessin, E. Cordonnier "*MP6 : Spécification d'un prototype d'anonymisation*", septembre 2003, 34 pp, Rapport LAAS 3525.

[Abou El Kalam *et al*. 2003d] A. Abou El Kalam, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, R. Elbaida, C. Saurel, G. Trouessin, "Modèles et politiques de sécurité des SICSS", *1ère Conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers* (*GISEH*), Lyon, janvier 2003, pp.268-277.

[Abou El Kalam *et al*. 2003e] A. Abou El Kalam, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, R. Elbaida, C. Saurel, G. Trouessin, "Modèles et politiques de sécurité des systèmes d'information et de communication en santé et social", à paraître dans la revue *Santé et Systémique*, édition Hermès, 2004, 21 pp.

[Abou El Kalam & Deswarte 2004a] A. Abou El Kalam, Y. Deswarte, "Modèles de sécurité pour le secteur de la santé", *Technique et Science Informatique* (*TSI*), Vol. 23, n° 3 (thématique sécurité informatique), mars 2004, Hermes, 30 pp. (Rapport LAAS 02433).

[Abou El Kalam et *al*. 2004a] A. Abou El Kalam, Y. Deswarte, G. Trouessin, E. Cordonnier, "Une démarche méthodologique pour l'anonymisation de données personnelles sensibles", *2ème Conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers*, Mons, 9-11 septembre 2004.

[Abou El Kalam et *al*. 2004b] A. Abou El Kalam, Y. Deswarte, G. Trouessin, E. Cordonnier, "Gestion des données médicales anonymisées", *Symposium SSTIC sur la Sécurité des Technologies de l'Information et des Communications*, Rennes, 2-4 juin 2004.

[AFNOR 1997] AFNOR, document de normalisation française, Fascicule de Documentation FD S 97-560.

[Ahn & Sandhu 2000] G. Ahn and R. Sandhu, "Role-Based Authorization Constraints Specification", *ACM Transactions on Information and System Security* (*TISSEC*), vol. 3, n° 4, novembre 2000, pp. 207-226.

[Ammann & Sandhu 1992] P.E. Ammann et R.S. Sandhu, "Implementing Transaction Control Expressions by checking for Absence of Access Rights", *Proceedings of the 8th Annual Computer Security Applications Conference*, San Antonio (Texas, USA), 30 novembre – 4 décembre 1992, IEEE Computer Society Press, pp. 131-140.

[Anderson 1980] J.P. Anderson, "Computer Security Threat Monitoring and Surveillance", Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, 1980.

[Anderson 1996a] R.J. Anderson, "Clinical System Security : Interim Guidelines", in *British Medical Journal*, vol. 312, n° 7023, janvier 1996, pp. 109-111.

[Anderson 1996b] R.J. Anderson, "A Security Policy Model for Clinical Information Systems", *IEEE Symposium on Security and Privacy*, Oakland, Californie, 6-8 mai 1996, IEEE Computer Society Press, pp. 30-43.

[Arrêté 1998] Arrêté du 29 juillet 1998 relatif au recueil et traitement des données d'activité médicale par les établissements de santé publics et privés financés par dotation globale visées à l'article L. 710-16-1 du même code et à la transmission aux agences régionales de l'hospitalisation et à l'État d'informations issues de ce traitement (JO, 26 août 1998).

[Audit 1998] Audit Commission, *Ghost in the Machine - An Analysis of IT Fraud and Abuse*, Audit Commission Publications, United Kingdom, ISBN 1-86240-05603, 1998.

[Barkley et al. 1999] J. Barkley, K. Beznosoz et J. Uppal, "Supporting Relationships in Access Control Using Role Based Access Control", *Proceedings of the ACM workshop on RBAC*, Fairfax, Virginia, USA, 28-29 octobre 1999, pp. 55-65.

[BCN 1999] *Droits d'accès au dossier médical informatisé*, Bulletin du Conseil National du 12 décembre 1998, BCN n° 84, juin 1999, ministère de la santé publique, 14 pp.

[Bell-LaPadula 1996] D.E. Bell, L.J. LaPadula, *Secure Computer Systems : Unified Exposition and Multics Interpretation*, Rapport technique, MTR 2997 Rev. 1, MITRE corp., Bedford (Massachussetts, USA), 1976.

[Benferhat *et al*. 1998] S. Benferhat, D. Dubois, H. Prade, "Practical Handling of Exception-Teinted Rules and Independence Information in Possibilistic Logic". *Applied Intelligence*, vol. 9, 1998, pp.101-127.

[Benferhat *et al*. 2003] S. Benferhat, R. El Baida, F. Cuppens, "A Stratification-Based Approach for Handling Conflicts in Access Control", 8th *ACM Symposium on Access Control Models and Technologies*, Côme, Italy, 2-3 juin 2003, 189-2003, ACM press.

[Bettini *et al*. 2002] C. Bettini, S. Jajodia, X. S. Wang et D. Wijesekera, "Obligation Monitoring in Policy Management", *International Workshop, Policies for Distributed Systems and Neworks* (*Policy 2002*), Monterey, Californie, 5-7 juin 2002, IEEE Computer Society Press, pp. 2-12.

[Biba 1977] K.J.Biba, *Integrity Consideration for Secure Computer Systems*, The MITRE Corporation, Technical Report ESD-TR-76-372 & MTR-3153, 1977.

[Bieber & Cppens 1991] P. Bieber et F. Cuppens, "A Definition of Secure Dependencies with the Logic of Security", 4th *IEEE Computer Security Foundations Workshop* (*CSFW*'91), Franconia, New Hampshire, USA, 18-20 June 1991, Proceedings. IEEE Computer Society Press, pp. 2-11.

[Bieber & Cppens 1992] P. Bieber et F. Cuppens, "A Logical View of Secure Dependencies", *Journal of Computer Security*, vol. 1, n. 1, 1992, pp. 99-129.

[Blobel 1996] B Blobel, "Clinical Record Systems in Oncology. Experiences and Developments on Cancer Registers in Eastern Germany", in *Personal Medical Information – Security, Engineering and Ethics*, R.J. Anderson (Editor), Springer-Verlag, ISBN 3-540-63244-1, pp. 39–56, 1997.

[Blobel & Pharow 2001] B.Blobel et P.Pharow, "The Need and Practice of User Authentication and TTP Services in Distributed Health Information Systems", *16th International Conference on Information Security (IFIP/SEC'01)*, Paris, France, 11-13 juin 2001, Kluwer Academic Publishers, pp. 61-76.

[BMA 1996] *Security in Clinical Information Systems*, British Medical Association, London, ISBN 0-7279-1048-5, 1996.

[Booch *et al*. 1999] G. Booch, J. Rumbaugh, I. Jacobson, "*The Unified Modeling Language. User Guide*", 1999, Addison Wesley, ISBN : 0-201-57168-4, 482 pp.

[Branstad *et al*. 1990] M.A. Branstad, C.P. Pfleeger, D. Brewer, C. Jahl, H. Kurth, "Apparent Differences Between the US TCSEC and the European ITSEC", *14th National Computer Security Conference*, octobre 1990, pp. 45-58.

[Brewer & Nash1989] D. Brewer, M. Nash, "The Chinese Wall Security Policy", *IEEE Symposium on Security and Privacy*, Oakland, Californie, 1-3 mai 1989, IEEE Computer Society Press, pp. 206-214.

[Catach 1989] L. Catach, *Les logiques multimodales*, Thèse de doctorat, Université de Pierre et Marie Curie (Paris 6), Paris, France, 1989, 312 pp.

[CC 1999a] *Common Criteria for Information Technology Security Evaluation*, *Part* 1*: Introduction and general model*, 60 p., ISO/IEC 15408-1 (1999).

[CC 1999b] *Common Criteria for Information Technology Security Evaluation*, *Part* 4*: Predifined Protection Profiles*, 166 pp., ISO/IEC 15408-1 (1999).

[CEN 1999] CEN/TC 251/WG I, *Norme prENV 13606-3: Health Informatics - Electronic Healthcare Record Communication*, n° 99-046, Comité Européen de Normalisation, 27 mai 1999.

[Chellas 1980] B.F. Chellas, "*Modal Logic : An Introduction*", Cambridge University Press, 1980, ISBN 0-521-29515-7, 295 pp.

[Cheng 1999] E. C. Cheng, "An Object-Oriented Organizational Model to Support Dynamic Role-Based Access Control in Electronic Commerce Applications", 32nd *Annual Hawaii International Conference on System Sciences (HICSS-32)*, Maui, Hawaii, 5-8 janvier 1999.

[Cheswik & Bellovin 1994] W.R. Cheswik, S.M. Bellovin, *Firewalls and Internet Security*, Addison-Wesley, IBSN 0-201-63357-4, 1994.

[Circulaire 1989] Circulaire DH/PMSI n° 303 du 24 juillet 1989 relative à la généralisation du Programme de médicalisatio*n* (BOMS n° 89/46), Ministère de l'emploi et de la solidarité, France.

[Circulaire 1998] Circulaire n° 153 du 9 mars 1998 relative à la généralisation dans les établissements de santé sous dotation globale et ayant une activité de soins de suite ou de réadaptation d'un recueil de RHS, ministère de l'emploi et de la solidarité, France.

[Clavel *et al*. 2002] M. Clavel, F. Duran, S. Eker, P. Lincoln, N. Marti-Oliet, J. Mesequer, J. Quesada, "Maude: Specification and Programming in Rewriting Logic", *Theoretical Computer Science*, vol. 285, n°. 2, pp. 187-243, Elsevier, Amsterdam, 2002.

[Clercq 1998] E. De Clercq, D. Deliège et C. Christoph, "Dossier médical, réseaux et système intégré de soins", *Septièmes Journées Francophones d'Informatique Médicale*, Société Belge d'Informatique Médicale : Santé et réseaux informatiques, Liège, 24-25 avril 1998, *Informatique et Santé*, 1998, Springer-Verlag France, vol. 10, pp. 56-64.

[Clark & Wilson 1987] D.Clark, D.Wilson, "A Comparison of Commercial and Military Computer Security Policies", *IEEE Symposium on Security and Privacy*, Oakland, Californie, 27-29 avril 1987, IEEE Computer Society Press, pp. 184-194.

[Code 1995a] Code de déontologie médicale, décret 95-1000 du 6 septembre 1995.

[Code 1995b] Code de la santé publique, Code de la famille et de l'aide social, Paris, Dalloz, 1995.

[CTCPEC 1993] *The Canadian Trusted Computer Product Evaluation Criteria*, Canadian System Security Center, Communication Security Establishment, Governement of Canada, version 3.0, janvier1993.

[Cuppens & Saurel 1996] F. Cuppens, C. Saurel, "Specifying a Security Policy: a Case Study", 9th *IEEE Computer Security Foundations Workshop*, Kenmare, County Kerry, Irlande, 10-12 juin 1996, IEEE Computer Society Press, pp. 123-134.

[Cuppens & Ortalo 2000] F. Cuppens, R. Ortalo, "A Language to Model a Database for Detection of Attacks", 3rd *International Workshop on Recent Advances in Intrusion Detection* (*RAID*). Toulouse, France, 2-4 octobre 2000, Springer, pp. 197-216.

[Cuppens & Saurel 1999] F. Cuppens, C. Saurel, "Toward a Formalization of Availability and Denial of Service", *in Information Systems Technology Panel Symposium on Protection Nato Information Systems in 21*st *Century*, Washington, octobre 1999.

[Cuppens & Miège 2003a] F. Cuppens, A. Miège, “Administration Model for Or-BAC”, *Workshop on Metadata for Security, International Federated Conferences* (*OTM*’03), Sicile, Italie, 3-7 novembre 2003.

[Cuppens & Miège 2003b] F. Cuppens et A. Miège, “Modelling Contexts in the Or-BAC Model”, *19th Annual Computer Security Applications Conference* (*ACSAC*’03), Las Vegas, Nevada, 8-12 décembre, 2003, IEEE Computer Society.

[Cuppens 2003] F. Cuppens, “Sécurité des bases de données”, in *Sécurité des réseaux et des systèmes répartis*, (Yves Deswarte & Ludovic Mé, eds), Traité IC2, Hermès, ISBN : 02-7462-0770-2, 264 pp, octobre 2003.

[Cury & Debar 2001] D. Curry et H. Debar, *Intrusion Detection Message Exchange Format Data Model and XML Document Type Definition*, draft-ietf-idwgidmef-xml-03.txt, février 2001.

[d’Ausbourg 1994] B. d’Ausbourg, “Implementing Secure Dependencies over a Network by Designing a Distributed Security SubSystem”, i*n Third European Symposium on Research in Computer Security* (*ESORICS*’94), (D. Gollman, Ed.), Brington, United Kingdom, Lecture Notes in Computer Science n° 875, novembre 1994, ISBN 3-540-58618-0, Springer-Verlag, pp. 249-266.

[Dacier 1993] M. Dacier, “A Petri Net Representation of the Take-Grant Model”, *in Computer Security Foundations Workshop VI*, Franconi, USA, 15-17 juin 1993, IEEE Computer Society Press, pp. 99-108.

[Dacier 1994] M. Dacier, *Vers une évaluation quantitative de la sécurité informatique*, Thèse de doctorat, Institut National Polytechnique de Toulouse, N° 971, 154 pp., 20 décembre 1994 (Rapport LAAS 94488).

[Dacier & Deswarte 1994] M. Dacier et Y. Deswarte, “Privilege Graph: an Extention to the Typed Access Matrix Model”, i*n Third European Symposium on Research in Computer Security* (*ESORICS*’94), (D. Gollman, Ed.), Brington, United Kingdom, Lecture Notes in Computer Science n° 875, novembre 1994, ISBN 3-540-58618-0, Springer-Verlag, pp. 319-334.

[Damianou *et al*. 2002] N. Damianou, N. Dulay, E. Lupu et M. Sloman. “The Ponder Policy Specification Language”, *International Workshop on Policies for Distributed Systems and Neworks (Policy 2001)*. Bristol, UK, IEE Computer Society Press, pp.18- 38, 29-31 Janvier 2001.

[Debar & Wespi 2001] H. Debar, A. Wespi, “Agregation and Correlation of Intrusion-Detaction Alerts”, 4*th International Workshop on Recent Advances in Intrusion Detection* (*RAID*), Californie (USA) , 10-12 octobre 2001, Springer, pp. 197-216.

[Décret 1994] Décret n° 94-666 du 27 juillet 1994 relatif aux systèmes d’information médicale et l’analyse de l’activité des établissements de santé publics et privés sous compétence tarifaire de l’État, modifié par le décret n° 98-63 du 2 février 1998.

[Décret 2002] Décret 2002-637 du 29 avril 2002 relatif à l’accès aux informations personnelles détenus par les professionnels et les établissements de santé en application des articles L.1111-7 et L.1112-1 du code de la santé publique.

[Degoulet 1989] P. Degoulet, J.-C. Stéphan, A. Venot et P.-J. Yvon, *Informatique et Gestion des Unités de Soins - Informatique et Santé – vol*. 1, Springer-Verlag France, pp. 257-268, juin 1989.

[Degoulet 1992] Degoulet P., Fieschi M., *Traitement de l'information médicale - Méthodes et applications hospitalières*, Collection : Manuels Informatiques Masson - Entreprise, Paris, 1992, 269 pp., ISBN: 2225825149.

[Deliège 2001] D. Deliège, "A Classification System of Social Problems - Concepts and impact on Gps' registration of problems", *Second International Conference on Social Work in Health and Mental Health Care*, Melbourne, 12-15 janvier 1998 : 25, Social Work in Health Care, 2001.

[Denning 1979] D. Denning et P. Denning, "Data Security". *ACM Computer Survey*, vol. 11, n° 3, septembre 1979, ACM Press, ISBN : 0360-0300, pp. 227-249.

[Directive 1995] Directive 95/46/CE du Parlement Européen, adoptée par le Conseil Européen le 24 juillet 1995, *On the protection of individuals with regard to the processing of personnal data and on the free movement of such data*, 1995.

[Directive 2002] Directive du Parlement Européen n° 2002/58/EC concernant "le traitement des données à caractère personnel et la protection de la vie privée dans le secteur de télécommunications électroniques", 12 juillet 2002, Journal Officiel L 201, 31-7-2002, pp. 37-47.

[Deswarte *et al*. 1999] Y. Deswarte, K. Kanoun , J.C. Laprie, "Diversity Against Accidental and Deliberate Faults", in *Computer Security, Dependability and Assurance: From Needs to Solutions*, P. Amman, B.H. Barnes, S. Jajodia, E.H. Sibley (Eds.), IEEE Computer Society Press, 1999, pp.171-181.

[Deswarte *et al*. 2001] Y. Deswarte, N. Abghour, V. Nicomette, D.Powell, "An Internet Authorization Scheme using Smartcard-based Security Kernels", *International Conference on Research in Smart Cards (e-Smart 2001)*, Cannes (France), 2001, "*Smart Card Programming and Security*", I. Attali and T. Jensen (Eds.), Springer-Verlag, LNCS 2140, pp. 71-82.

[Deswarte *et al*. 2002] Y. Deswarte, N. Abghour, V. Nicomette, D.Powell, "An Intrusion-Tolerant Authorization Scheme for Internet Applications", *Supp. to the proceedings of the 2002 Int. Conf. on Dependable Systems & Networks (DSN 2002)*, Workshop on Intrusion Tolerant System*s*, Washington (USA), 23-26 juin 2002, pp. C1.1-C1.6.

[Deswarte 2003] Y. Deswarte, "La sécurité des systèmes d'information et de communication", in *Sécurité des réseaux et des systèmes répartis*, (Yves Deswarte & Ludovic Mé, eds), Traité IC2, Hermès, ISBN : 02-7462-0770-2, 264 pp, octobre 2003.

[Fabre *et al*. 1996] J.C. Fabre, Y. Deswarte, L. Blain, "Tolérance aux fautes et sécurité par fragmentation-redondance-dissémination", *Technique et Science Informatiques* (*TSI*), Vol.15, n° 4, 1996, Hermes, pp.405-427.

[Federal Criteria 1992] *Federal Criteria for Information Technology Security*, National Institute of Standards and Technology (NIST) and National Security Agency (NSA), vol. I and II, Draft, 1992.

[Ferraiolo *et al*. 2001] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn and R. Chandramouli "A Proposed Standard for Role-Based Access Control", *ACM Transactions on Information and System Security,* vol. 4, n° 3, août 2001.

[Fitting 1993] M. Fitting, "*Basic Modal Logic*", Handbook of Logic in Artificial Intelligence and Logic Programming Logic Foundations, (D.M. Gabbay, C.J. Hogger, J.A. Robinson, Eds.). Vol. 1/5, pp.365-448, ISBN 0-19-853745-X, Oxford Science Publications, 1993.

[FMAG 1996] Bundesarztekammer. *Empfehlungen zu arztlicher Schweigepflicht, Datenschutz und Datenverarbeitung in der Arztpraxis. Koln: Bundesarztekammer*, 1996 (en allemand).

[Fray 1986] J.M. Fray, Y. Deswarte, D. Powell, "Intrusion-Tolerance Using Fine-Grain Fragmentation-Scattering", *Proc. Int. Symp. on Security and Privacy*, pp. 194-201, Oakland, Californie, USA, IEEE Computer Society Press, mai 1986.

[Garvila & Barkley 1998] S.I. Gavrila, J.F. Barkley, "Formal Specification for Role Based Access Control", *Third ACM Workshop on RBAC*, Fairfax, Virginia, USA, 22-23 octobre 1998.

[Georgiadis *et al*. 2001] C.K. Georgiadis, L. Mavridis, G. Pangalos, R.K. Thomas, "Flexible Team-Based Access Control Using Contexts", *ACM Symposium on Access Control Models and Technologies,* (*SACMAT'01*), Chantilly, Virginia, USA, 3-4 mai 2001, pp. 21-27.

[Godfrey *et al*. 1998] P. Godfrey, J. Grant, J. Gryz, J.Minker, "Integrity Constraints: Semantics and Applications", in *Logics for Databases and Information Systems,* J. Chomiki et J. Minker (Eds.), Kluwer Academic Publishers, 1998, ISBN 0-7923-8129-7.

[HRU 1976] M.A. Harrison, W.L. Ruzzo et J.D. Ullman, "Protection in Operating Systems", *Communication of the ACM,* 19(8), pp. 461-471, août 1976.

[Hu 1991] W-M. Hu, "Reducing Timing Channels with Fuzzy Time", in *Proceedigns of the 1991 IEEE Symposium on Research in Security and Privacy*, Oakland (CA), 20-22 mai 1991, pp.8-20.

[ISO 15408] Common Criteria for Information Technology Security Evaluation, norme ISO/CEI 15408, version 2.1, août 1999.

[ITSEC 1991] ITSEC , *Information Technology Security Evaluation Criteria*, v 1.2, 136 pp., ISBN 92-826-3005-6, Office des publications officielles des Communautés Européennes, Luxembourg, 1991.

[Jacob 1988] J. Jacob, "Security Specification", *IEEE Symposium on Security and Privacy*, 18-21 Avril 1988, Oakland, Californie, IEEE Computer Society, pp. 14-23.

[JCSEC 1992] JCSEC, "*The Japanese Computer Security Evaluation Criteria – Functionality Requirements*", Ministry of International Trade and Industry, Draft V1.0, août 1992.

[Jeanneret *et al*. 2001] J.P. Jeanneret, D. Olivier, J. Chiffelle, "How to Protect Patient's Rigottes to Médical Secret in Official statistic", *Information Security Solutions Europe Conference* (*ISSE*), London, UK, 26-28 Septembre 2001.

[Jones *et al*. 1976] A.K. Jones, R.J. Lipton, L. Snyder, "A Linear Time Algorithm for Deciding Security", *17th Annual Symposium on Fundations of Computer Science*, Houston, USA, 1976.

[Joseph 1988] M.K. Joseph, A. Avizienis, "A Fault Tolerance Approach to Computer Viruses", *Proc. Int. Symp. on Security and Privacy*, Oakland, Californie, USA, IEEE Computer Society Press, mai 1988, pp. 52-58.

[Katsikas 1996] S. Katsikas, D. Gritzalis, "High Level Security Policy Guidelines". in: *Data Security for Health Care*, SEISMED Consortium (Eds.), IOS Press, 1996.

[Kleene 1967] S. Kleene, "*Mathematical Logic*", John Wiley & Sons, 1967, traduit en français par A. Largeault sous le titre "*Logique mathématique*", A. Colin, Paris, ISBN : 0–19–850048–3, 1972.

[Kripke 1959] S. Kripke, "A Comleteness Theorem in Modal Logic", *Journal of Philosophical Logic*, vol. 24, 1959, pp. 1-14.

[Kripke 1963] S. Kripke, "Semantical Consideration in Modal Logic", *Acta Philosophical Logic*, vol. 16, 1963, pp. 83-94.

[Laprie 1995] J.-C. Laprie, J.Arlat, J.-P. Blanquart, A. Costes, Y. Crouzet, Y. Deswarte, J.-C. Fabre, H. Guillermain, M. Kaâniche, K. Kanoun, C. Mazet, D. Powell, C. Rabéjac et P. Thévenod, *Guide de la sûreté de fonctionnement*, 324 pp., Editions Cépaduès, Toulouse 1995.

[Lampson 1971] B. Lampson, "Protection", *5th Princeton Symposium on Information Sciences and Systems*, 1971.

[Landwehr 1983] C.E. Landwehr, "The Best Available Technologies for Computer Security", *IEEE Computer*, vol. 16, n° 7, pp. 86-100, juillet 1983.

[Lawrence 1993] L.G.Lawrence, "The Role of Roles", *Computer & Security*, vol.12, n° 1, pp. 16-21, 1993.

[Loi 1978] Loi 78-17 du 6 janvier 1978 relative à l'Informatique, aux fichiers et aux libertés, Journal officiel de la République française, pp. 227-231, décret d'application 78-774 du 17 juillet 1978, pp. 2906-2907.

[Loi 1991] Loi 91-748 du 31 juillet 1991 portant réforme hospitalière.

[Loi 1994] Loi 94-43 du 18 janvier 1994 relative à la santé publique et à la protection sociale, article 8.

[Loi 2002] Loi 2002-303 du 4 mars 2002 relative aux droits des malades et à la qualité du système de santé, article L. 1111-7.

[Matheron 1998] Jean-Paul Matheron. *Comprendre MERISE ; outils conceptuels et organisationnels*, 1998, Editions Eyrolles, 5ème édition, ISBN n° 2-212-07502-2.

[McLean 1985] J. McLean, "A comment of the basic security theorem of Bell and LaPadulla". *Information Processing Letters*, vol. 2, n° 2, pp. 67-70, Février 1985.

[McLean 1990] J. McLean, "Security Models and Information Flow". *IEEE Symposium on Research in Security and Privacy,* Oakland, California, pp. 180-187, IEEE Computer Society Press, 1990.

[Menezes *et al*. 1996] A.J. Menezes, P.C. Van Oorshot, S.A. Vanstone, *Handbook of Applied Crytography*, CRC Press, ISBN 0-8493-8523-7, 1996.

[Michel & Mé 2001] C. Michel et L. Mé, "ADeLe: an Attack Description Language for Knowledgebased Intrusion Detection", *in Proceedings of the 16th International Conference on Information Security (IFIP/SEC'01)*, Paris, France, 11-13 juin 2001, Kluwer Academic Publishers, juin 2001, pp. 353-365.

[Nicomette 1996] V. Nicomette, *La protection dans les systèmes à objets répartis*, Thèse de doctorat, Institut National Polytechnique de Toulouse, n° 1252, 177 pp., 17 décembre 1996 (Rapport LAAS 96496).

[Muller & Gaetner 2000] P.A. Muller, N. Gaertner, "*Modélisation objet avec UML*", 2000, Eyrolles, ISBN : 2-212-09122-2, 520 pp.

[Nicomette & Deswarte1997] V. Nicomette, Y. Deswarte, "An Authorization Scheme for Distributed Object Systems", in *Proc. Int. Symposium on Security and Privacy*, Oakland (CA, USA), mai 1997, IEEE Computer Society Press, pp. 21-30.

[Oh & Sandhu 2002] S. Oh, R.S. Sandhu, "A Model for Role Administration Using Organisation Structure", in *Proceeding of the 7th ACM Symposium on Access Control Models and Technologies* (*SACMAT 2002*), Monterey, Californie, 3-4 juin 2002, ACM press, pp. 155-162.

[Ordonnance 1996] Ordonnance n° 96-346 du 24 avril 1996 portant réforme de "l'hospitalisation publique et privée des systèmes d'information et à l'organisation médicale dans les hôpitaux publics".

[Ortalo 1998a] R. Ortalo, "A Flexible Method for Information System Security Policy Specification", in 5th *European Symposium on Research in Computer Security* (*ESORICS* 98), Louvain-La-Neuve, Belgique, 16-18 Septembre 1998, Lecture Notes in Computer Science n° 1485, Springer-Verlag, pp. 67-84.

[Ortalo 1998b] R. Ortalo, "Evaluation quantitative de la sécurité des systèmes d'information", Thèse de doctorat, Institut National Polytechnique de Toulouse, n° 1418, 166 pp., 18 mai 1998 (Rapport LAAS 98164).

[Ortalo 1999] R. Ortalo, Y. Deswarte, M. Kaâniche, "Experimenting with Quantitative Evaluation Tools for Monitoring Operational Security", *IEEE Transactions on Software Engineering*, Vol.25, N°5, pp. 633-650, septembre/octobre 1999.

[Powell & Stroud 2003] D. Powell, R. Stroud (Eds.), *Malicious- and Accidental-Fault Tolerance for Internet Applications: Conceptual Model and Architrecture*, Final Version, Rapport LAAS n°03011, Projet IST-1999-11583 MAFTIA, Deliverable D21, janvier 2003, 123 pp., disponible à <http://www.research.ec.org/maftia/deliverables/>.

[Quantin *et al*. 1898] C. Quantin, H. Bouzelat, FA. Allaert, AM. Benhamiche, J. Faivre et L. Dusserre, "How to ensure data security of an epidemiological follow-up: quality assessment of an anonymous record linkage procedure", *International Journal of Medical Informatics* 49 (1998) 117-122.

[Rabin 1989] M.O. Rabin, "Efficient Dispersal of Information for Security", Load Balancing and Fault Tolerance", *J. of the ACM*, vol. 36, n° 2, pp. 335-348, 1989.

[Recommendation 1992] Recommendation of the "Communication of Health Information in Hospitals", European Health Commitee CDSP (92)8, Council of Europe, Strasbourg, 21 juin 1992.

[Recommandation 1997] Recommandations du Conseil de l'Europe, R(97)5, *On The Protection of Medical Data Banks*, Council of Europe, Strasbourg, 13 février 1997.

[Résolution 1990] Résolution A/RES/45/95 Assemblée générale des Nations Unies, *Principes directeurs pour la réglementation des fichiers personnels informatisés*, 14 Décembre 1990.

[Séminaire 2001] Séminaire de la société Française de statistique, *Appariements sécurisés et statistique publique*, organisé par le groupe Statistique économique et sociale par Benoît Riandey (INED-SfdS), 28 février 2001.

[Sandhu 1992] R.S. Sandhu, "Expressive Power of the Schematic Protection Model", *Journal of Computer Security*, vol.1, n° 1, pp. 59-98, 1992.

[Sandhu *et al*. 1996] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, "Role-Based Access Control Models", *IEEE Computer*, vol. 29, n° 2, pp.38-47, février, 1996.

[Sandhu 1996] R.S. Sandhu, "Role Hierarchies and Constraints for Lattice-Bases Access Controls", in *4th European Symposium on Research in Computer Security* (*ESORICS*'96), (E. Bertino, H. Kurth, G. Martella, E. Mon,tolivo, Eds.), Rome, Italie, september 25-27, Lecture Notes in Computer Science 1146, pp. 65-79, ISBN 3-540-61770-1, Springer-Verlag, 1996.

[Sandhu & Bhamidipati 1997] R.S. Sandhu, V. Bhamidipati, "The URA97 Model for Role-Based User-Role Assignment", in *Proceeding of IFIP WG 11.3 Workshop on Database Securiy*, Californie, 11-13 août 1997.

[Sandhu & al. 1999] R.S. Sandhu, V. Bhamidipati et Q. Munawer, "The ARBAC97 Model for Role-Based Administration of Roles", *ACM Transactions on Information and System Security* (*TISSEC*), vol. 2, n° 1, février 1999, pp. 105-135.

[Sandhu & Munawer 1999] R.S. Sandhu, Q. Munawer, "The ARBAC99 Model for Administration of Roles", in *Proceeding of the* 15th *Annual Computer Security Applications Conference* (*ACSAC'99*), Phoenix, Arizona, 6-10 décembre 1999, IEEE Computer Society, pp. 229-241.

[Sandhu *et al*. 2000] R.S. Sandhu, D. Ferraiolo et R. Kuhn, "The NIST Model for Role-Based Access Control: Towards a Unified Standard", in *5th ACM Workshop on Role-Based Access Control,* Berlin (Allemagne), pp. 47-63, 2000.

[Saydjari *et al*. 1989] O.S. Saydjari, J.M. Beckman et J.R. Leaman, "LOCK Trek: Navigating Unchartered Space", *International Symposium on Security and Privacy,* Oakland (CA, USA), pp. 167-177, IEEE Computer Society Press, 1989.

[SMA 1995] Swedish Medical Association, *Information Technology: The Physician and the Patient*, Stockholm, SMA, 1995.

[Solms & Merwe 1994] S.H. von Solms et I. Van der Merwe, "The Management of Computer Security Profiles Using a Role-Oriented Approach", *Computer & Security*, vol.13, n° 8, pp. 673-680,1994.

[Synder 1981] L. Synder, "Theft and Conspiracy in the Take-Grant Model", *Journal of Computer and System Sciences*, vol. 23, pp. 333-347, 1981.

[TCSEC 1995] TCSEC, *Trusted Computer System Evaluation Criteria*, 122 pp., Department of Defense (DoD), DoD Standard, DoD 5200.28-STD, 1985.

[Thomas 1997] R.K. Thomas, "TMAC: A primitive for Applying RBAC in Collaborative Environment", *2nd ACM Workshop on RBAC*, FairFax, Virginia, USA, pp. 13-19, 6-7 novembre 1997.

[TNI 1987] TNI, *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG-005, National Computer Security Center, 31 juillet 1987, 278 pp.

[Totel 1998] E. Totel, "Politique d'intégrité multiniveau pour la protection en ligne de tâches critiques", Thèse de doctorat, Institut National Polytechnique de Toulouse, n° 1571, 145 pp., 18 mai 1998 (Rapport LAAS 98533).

[Trouessin 2000] G. Trouessin, *Towards Trustworthy Security for Healthcare Information Systems*, Report N°GT/2000.03, CESSI/CNAM, juin 2000.

[Trouessin 2001] G. Trouessin, "Sécurité et intimité des données à caractère personnel", *La Lettre d'ADELI n° 42*, juillet 2001.

[Trouessin 2002] G. Trouessin, “L’évolution des normes de sécurité vers plus d’auditabilité des systèmes d’information”, *Colloque AIM à l’HEGP : « Présent et avenir des systèmes d’information et de communication hospitaliers »*, 23-24 mai 2002 (à paraître chez Springer-Verlag).

[Wang 1999] W. Wang, “Team-and-Role-Based Organizational Context and Access Control for Cooperative Hypermedia Environments”, *Proceeding of the 10th ACM Conference on Hypertext and Hypermedia* (*Hypertext’99*), Darmstadt, Germany, pp. 37-46, February 21-25, 1999.

[Weissman 1992] C. Weissman, “BLACKER: Security for the DDN, Examples of A1 Security Engineering Trades”, *in EEE Symposium Research in Security ans Privacy*, Oakland, Californie, IEEE Computer Society Press, 4-6 mai 1992, pp. 286-292.

[Willikens *et al*. 2002] M. Willikens, S. Feriti et M. Masera, “A Context-related Autorisation and Access Control Method Based on RBAC: A Case Study from the Health Care Domain”, *ACM Symposium on Access Control Models and Technologies, (SACMAT’02)*, Californie, U.S.A., 3-4 juin 2002, pp. 177-124.

[Woodward 1995] B. Woodward, “The Computer-Based Patient Record and Confidentiality”, *New England Journal of Medicine*, v. 333, n° 21, 1995, pp. 1419-1422.

[Zakinthinos & Lee 1994] A. Zakinthinos, E.S. Lee, “The Composability of Non-Interference”, *Journal of Computer Security*, vol. 3, no. 4, pp.269-281, 1994.