Vers la spécification des exigences de sécurité des systèmes d'information

Salim CHEHIDA [1], Mustapha kamel RAHMOUNI [2]
[1] Département d'Informatique, Université de Mostaganem, Algérie
salimchehida@yahoo.fr
[2] Département d'Informatique, Université d'Oran Es-Sénia, Algérie
kamel_rahmouni@yahoo.fr

Abstract. De plus en plus notre société est dépendante d'Internet, Où le besoin de sécurité s'applique pour le commerce électronique, l'accès distant à une machine, le transfert de fichier, l'accès à certaines parties d'un site contenant des données confidentielles et d'autres applications. La spécification fonctionnelle des systèmes d'information n'est pas suffisante, la conception et la réalisation de ces systèmes doivent tenir compte, en plus des besoins fonctionnels, des différentes exigences de sécurité. La prise en compte des contraintes de sécurité (authentification, intégrité, confidentialité, non répudiation, disponibilité, etc.) au niveau de la modélisation constitue l'un des principaux challenges pour les concepteurs des SI (particulièrement lorsqu'ils sont connectés au web). UML s'est imposé comme le langage standard pour la modélisation des vues multiples d'un système à l'aide d'un ensemble de mécanismes d'extension. Dans ce travail, nous proposons une démarche et un ensemble d'extensions d'UML pour la spécification des exigences de sécurité des systèmes d'information.

Mots clés : Sécurité des systèmes d'information, Modélisation, Spécification, UML, UMLsec.

1. Introduction

La généralisation de la connexion à l'Internet offre des possibilités nouvelles et prometteuses, elles introduisent également un certain nombre de risques dont il faut prendre conscience, en mesurer les conséquences éventuelles, et en connaissance de cause prendre les mesures adéquates. Les entreprises se trouvent désormais confrontées au contrôle efficace de la confidentialité, de l'intégrité et de la disponibilité de ces informations. La sécurité à posteriori des SI (Firewall, Antivirus, etc.) peut donner des résultats mais elle ne peut remplacer à elle seule une véritable politique de sécurité. Nous pensons que l'élaboration d'une politique de sécurité pour un système d'information doit se faire en même temps que la modélisation, et que le modèle final doit intégrer les spécifications de sécurité. La sécurité des systèmes d'information doit donc commencer par l'élaboration d'un « modèle » en identifiant : Quelles sont les menaces ? Que doit-on protéger ? Pourquoi ? C'est donc le modèle - répondrant à ces

trois questions- qui donne un sens au mot « sécurité ». [22] UML est un langage standard pour visualiser, spécifier, construire et documenter un système logiciel. Malgré sa richesse, ce langage n'est pas adapté à tous les domaines ; il utilise un ensemble des mécanismes d'extension (les stéréotypes, les étiquettes et les contraintes) pour modéliser des différents aspects du système. *UMLsec* est une extension d'UML proposée par Jürjens (Munich University of Technology) comprennent un ensemble des profils pour la sécurité au niveau des modèles conceptuels. Pour la spécification des exigences de sécurité, *UMLsec* introduit deux mécanismes dans les diagrammes d'activité afin de sécuriser des transactions électronique ; le stéréotype **<<**fair exchange**>>** est utilisé pour assurer un échange équitable lors d'une transaction électronique et le stéréotype <<provable>> pour assurer la non répudiation dans les transactions de e-commerce. Le présent article propose une nouvelle démarche et un ensemble d'extensions (en plus de profils UMLsec) permettant la spécification des exigences de sécurité des systèmes d'information avec UML. Dans ce travail, nous avons aussi présenté des exemples de spécification de quelques besoins de sécurité du système ANEM (Agence Nationale d'Emploi) pour la mise en œuvre des extensions et de la démarche proposée.

2. Mécanismes d'extension d'UML

Ces mécanismes comprennent les stéréotypes, les étiquettes et les contraintes.

2.1 Stéréotype

Les stéréotypes permettent d'étendre la sémantique des éléments de modélisation et de définir de nouvelles classes d'éléments, en plus du noyau prédéfini par UML. En d'autre terme, un stéréotype est utilisé pour définir une utilisation particulière d'éléments de modélisations ou pour modifier la signification de ces éléments ; l'élément stéréotypé et son parent ont une structure identique mais une sémantique différente. Le nom du stéréotype est placé entre guillemets. Une icône peut être associée à un stéréotype (aucune icône n'est prédéfinie par UML) [17].

2.2 Etiquette

Une étiquette ou valeur marquée est une paire (nom, valeur) qui ajoute une nouvelle propriété à un élément de modélisation. Les propriétés permettent l'extension des attributs des éléments [18]. La spécification d'une valeur marquée prend la forme : nom = valeur. Une valeur marquée est indiquée entre accolades.

2.3 Contrainte

Une contrainte est une relation sémantique entre les éléments de modélisation. UML ne spécifie pas une syntaxe particulière pour les contraintes, qui peuvent ainsi

être exprimées en langage naturel, en pseudo-code, par des expressions de navigation ou par des expressions mathématiques. Chaque contrainte est indiquée entre accolades et placée près de l'élément (stéréotypé ou non) auquel elle est associée.

3. La démarche

La démarche est définie par une séquence d'étapes successives et ordonnées permettant la spécification des exigences de sécurité d'un système d'information. La figure suivante présente les différentes étapes de notre démarche.

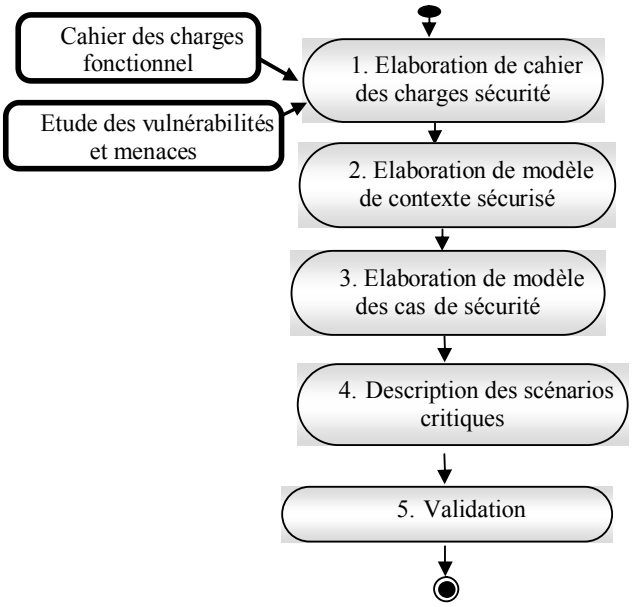

Fig. 1. La démarche de spécification des exigences de sécurité

- L'élaboration de cahier des charges est la première étape de la démarche. Cette étape jeu un rôle très important, c'est le point de départ de la spécification. Elle consiste à effectuer un premier repérage des différentes exigences de sécurité en utilisant le texte.
- La modélisation de contexte sécurisé consiste à définir les différents services de sécurité attendus du système considéré comme une boîte noire. L'objectif principal de cette étape est donc de préparer le terrain aux étapes de spécification des cas de sécurité et de description des scénarios critiques.

- Le modèle des cas de sécurité permet de structurer les services de sécurité fournis par le système (toujours envisagé comme une boite noire) pour les différents acteurs en un ensemble de cas de sécurité. Les cas de sécurité permettent de définir ce qu'on attend du système en terme de sécurité. Par exemple : Assurer l'authentification d'un utilisateur, Assurer l'intégrité et la confidentialité des informations échangées, Assurer la non répudiation d'une transaction,...
- Les scénarios critiques consistent à décrire les interactions ou les actions qui incluent un risque en mettant en jeu les différentes services ou propriétés de sécurité spécifiées par les cas de sécurité. Par exemple : les scénarios qui permettent d'assurer la non répudiation dans les transactions; il garantir que si une action est exécutée, elle ne peut pas être niée (elle est prouvée), les scénarios qui permettent d'assurer un échange équitable lors d'une transaction, les scénarios qui spécifient les interactions permettant l'échange des informations critiques (nécessite une confidentialité et une intégrité).
- Après l'identification des cas de sécurité et les scénarios critiques, le chef de projet valide ces cas avec le client ou les acteurs concernés. Si l'ensemble des exigences assurées par les cas de sécurité ne répond pas aux besoins de cahier des charges, l'équipe de spécification doit reprendre la spécification et corriger les erreurs.

4. Cahier des charges sécurité

Sur Internet, il est beaucoup plus difficile d'évaluer la sûreté des affaires. En outre des menaces sérieuses ont émergés du fait que le commerce électronique utilise un réseau public pour effectuer des transactions ayant un caractère critique. Un cahier des charges sécurité consiste à identifier les menaces, les vulnérabilités du système ainsi que les risques et les attaques possibles sur le système.

4.1 Menace

Une menace est un danger qui existe dans l'environnement d'un système indépendamment de celui-ci. Il représente l'ensemble des actions de l'environnement d'un système pouvant entraîner des pertes financières. Un système informatique sera d'autant plus menacé que les informations qu'il contient auront une valeur à la fois pour leur propriétaire et pour d'autres entités. [2]

4.2 Vulnérabilité

Une vulnérabilité est une erreur ou faille dans un système informatique permettant à un attaquant de porter atteinte à la sécurité de ce système, c'est-à-dire à son fonctionnement normal, à la confidentialité et l'intégrité des données qu'il contient et même à la disponibilité de ce système. Ces vulnérabilités sont la conséquence de faiblesses dans

la conception, la mise en œuvre ou l'utilisation d'un composant matériel ou logiciel du système, mais il s'agit généralement de l'exploitation de bugs logiciels.

4.3 Risque et attaque

Le risque est la combinaison de la menace et de la vulnérabilité. En l'absence de vulnérabilité, les menaces n'exposent à aucun risque. De même, en l'absence de menaces, la vulnérabilité n'expose à aucun risque. Mesurer un risque consiste à tenter d'identifier la probabilité qu'un événement dommageable survienne. [19] Parmi les risques, notons:

- vol d'informations confidentielles
  Les informations doivent être protégé car l'espionnage industriel est une réalité. Les données financières doivent aussi être protégées, ainsi que l'accès aux fichiers confidentiels.
- Modifications de données de haute importance
  Il s'agit des modifications stratégiques mais non détectables dans l'immédiat. Il y a une certaine gravité si ces modifications ne sont découvertes que trop tardivement.
- Rebonds
  Dans le cadre d'une attaque en règle, utilisation d'un système local à un établissement pour rebondir en cascade vers un autre site. L'établissement est donc responsable car étant le dernier dans la chaîne.
- Déni de service
  La qualité des systèmes informatique et du réseau doivent être garantie et maintenue. Ceci peut entraîner une perte de confiance dans l'outil informatique et réseau mais aussi une perte d'argent par les retards accumulés si les performances de l'outil informatique se dégradent.
- Destruction ou corruption d'informations
  Ceci peut occasionner des pertes de temps considérables pour restaurer les systèmes et les bases de données. De plus si des sauvegardes ne sont pas régulièrement assurées, cela peut devenir catastrophique.
- mascarade d'identité
  Cela peut porter un lourd préjudice à l'image d'un établissement, quant aux relations ultérieures avec des partenaires.
- Utilisation frauduleuse de ressources
  Intrusion extérieure ou intérieure sur une machine sans autorisation et utilisation de ressources réservées.

5. Modèle de contexte sécurisé

De nombreux auteurs, comme G.Bouch dans [Object solutions [21]] ou plus récemment P.Roques et F.vallée dans [UML en action [1]], ont préconisé l'utilisation de

diagramme de collaboration pour représenter de façon synthétique les différents besoins fonctionnels d'un système. Après l'élaboration du cahier des charges sécurité, on peut donc présenter les différentes exigences de sécurité sur un diagramme, que l'on peut qualifier de modèle de contexte sécurisé (secure context model). Le diagramme de collaboration est utilisé de la façon suivante :

- Le système est représenté par un objet central, cet objet est entouré par d'autres objets symbolisant les différents acteurs.
- Les objets sont reliés par des liens, sur chaque lien sont montrés des messages en sortie de système pour représenter les différents services de sécurité assurés par le système.

La figure suivante présente un exemple de modèle de contexte sécurisé.

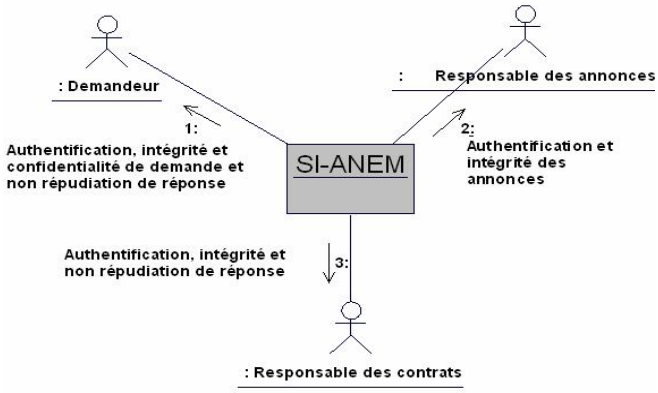

Fig. 2. Modèle de contexte sécurisé

6. Modèle des cas de sécurité

Cette étape consiste à identifier les cas de sécurité à partir de modèle de contexte sécurisé. Un cas de sécurité représente un service de sécurité rendu par le système pour un ou plusieurs acteurs. Pour cela, on utilise les cas d'utilisation de manière différente en introduisant les notions de cas de sécurité et de diagramme des cas de sécurité. Un cas de sécurité spécifie un comportement attendu du système pour répondre à des besoins de sécurité sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire en terme de sécurité informatique sans définir comment il le fait.

Les cas de sécurité sont absolument distincts des cas d'utilisation ; ils ne produisent pas une valeur ajoutée fonctionnelle mais ils recouvrent en effet tous service de sécurité dont un utilisateur bénéficie.

La figure 3 présente un exemple de modèle des cas de sécurité.

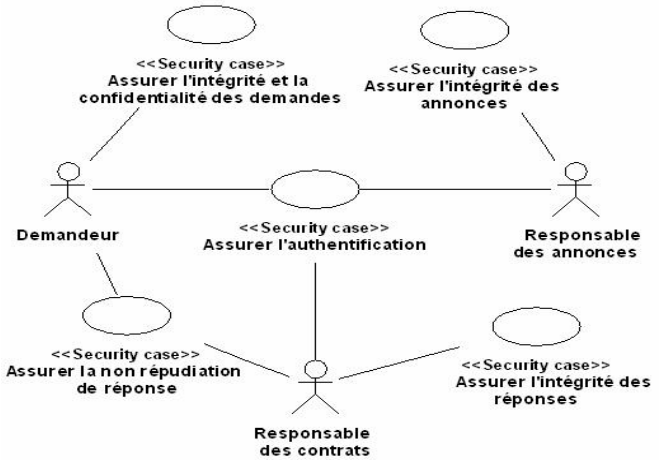

Fig. 3. Modèle des cas de sécurité

7. Scénarios critiques

Un scénario critique représente une succession particulière d'enchaînement (séquence d'actions d'interactions entre les acteurs et le système) qui incluent un risque en terme de sécurité informatique. Pour souligner ce risque, nous allons associer les propriétés de sécurité spécifiées par les cas de sécurité sur les interactions entre le système considéré comme une boîte noire et les différents acteurs. Par exemple : les scénarios qui permettent d'assurer la non répudiation dans les transactions électronique et les scénarios qui spécifient les interactions avec échange des informations critiques.

Pour la description des scénarios critiques, nous avons utilisé deux diagrammes dynamiques d'UML ; le diagramme d'activité qui est très utile en cas des actions parallèles [1] et le diagramme de séquence qui permet de bien visualiser les actions critiques.

La figure dans la page suivante présente un exemple de modèle qui souligne des échanges critiques.

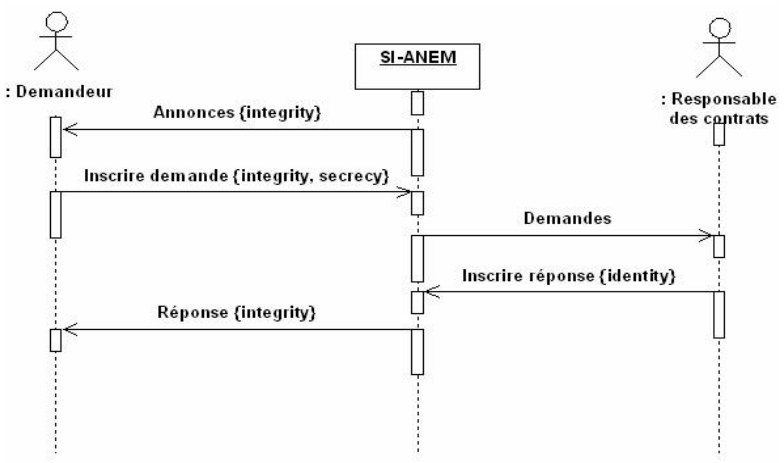

Fig. 4. Modèle des interactions critiques

Nous avons utilisé trois contraintes pour les interactions entre le système et les acteurs :

- La contrainte {secrecy} pour assurer la confidentialité des interactions.
- La contrainte {integrity} pour assurer l'intégrité des interactions.
- La contrainte {identity} pour assurer l'identité des parties lors de l'exécution d'une action d'interaction entre un acteur et le système.

La figure suivante présente un exemple de modèle qui exprime la non répudiation d'une transaction.

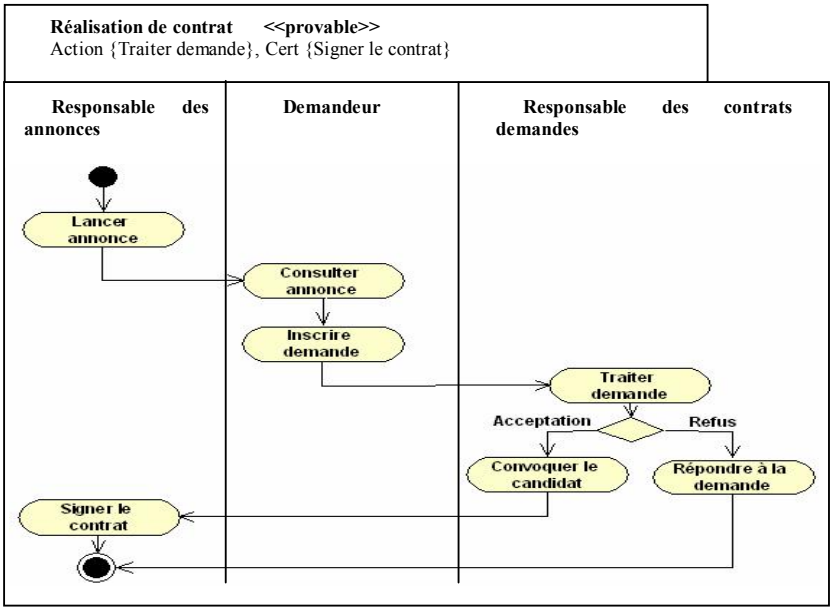

Fig. 5. Le stéréotype <<provable>>

Ce stéréotype d'*UMLsec* permet d'assurer la non répudiation dans les transactions de e-commerce ; il garantit que si une action est exécutée, elle ne peut pas être niée. Le package << provable >> définit deux étiquettes {action} et {cert} [10]. Pour tous scénario d'un diagramme d'activité : Un état d'activité contenu dans l'étiquette "cert" n'est atteint que si l'état d'activité contenu dans l'étiquette "action" est atteint avant lui.

8. Conclusion

UML n'est pas une notation fermée : elle est générique, extensible et configurable par l'utilisateur [17]. En cas de besoin, des précisions peuvent être apportées au moyen de mécanismes d'extension. Cet article présente des nouvelles profils d'UML pour la spécification des contraintes de la sécurité informatique, mais la porte reste ouverte pour d'autres extensions dans les différentes phases de développement.

La démarche proposée se fait par itération. Après l'identification des cas de sécurité, On valide ces cas. Si l'ensemble des exigences assurées par les cas de sécurité ne répond pas aux besoins de cahier des charges sécurité, on doit reprendre la spécification et corriger les éventuelles erreurs. La spécification des exigences de sécurité, quant à elle, produit des modèles pour le court terme, afin de définir les contraintes de sécurités imposées aux systèmes d'information après la capture des menaces provenant de l'environnement de ce système. Si l'environnement exige de nouvelles donnes en matière de sécurité, il convient d'intégrer la spécification des nouvelles exigences pour améliorer la sécurité de système. Les points importants qui reste à développer : Elaboration des nouvelles extensions d'UML pour les autres phases de développement ainsi que l'intégration de ces extensions dans un processus de développement.

9. Références

[1] P. Roques et F. Vallee, « *UML en action* », Eyrolles, (2002).

[2] Robert Longeon et Jean-Luc Archimbaud ; *« Guide de la sécurité des systèmes d'information a l'usage des directeurs »*, Cours CNRS, Site : http://www.cnrs.fr/Infosecu

[3] I. Jacobson, G. Booch, J. Rumbaugh, « *Le processus unifié de développement logiciel* », Eyrolles, (2000).

[4] A.Cockburn, *« Rédiger des cas d'utilisation efficaces »*, Eyrolles, (2001).

[5] P.Roques, « *UML par la Pratique* », Eyrolles, 2ème édition (2003).

[6] R.Medina, « *L'Extreme Programming* », Cours Crossbow Labs, (2008).

[7] P. kruchten , « *The Rational Unified Process : An Introduction* », Addison-Wesley, Second Edition (2000).

[8] S. Meng « *Security Requirements Analysis and Modeling of Distributed Systems* », Thèse de Master , Munich University of Technology Department of Informatics, Software & Systems Engineering, (2004).

[9] E.Maiwald, « *Sécurité des réseaux* », Campus Press, (2001).

[10] J. Jurjens, *« Secure Systems Development with UML: a Foundation »*, Thèse de doctorat, Munich University of Technology, (2003).

[11] S.Chehida, « Modélisation sécurisée des systèmes d'information Etude de cas: ANPE », Mini-projet Université d'Oran Es-Sénia – Ecole doctorale STIC, (2007).

[12] B.Debbabi, M.S.Boudjelda, « Le processus unifié de développement logiciel RUP », cours. (2007).

[13] G. Picard, « *le processus unifié*», Cours ENS Mines Saint-Etienne, (2008).

[14] P. Roques, « *Modéliser un site e-commerce*», Eyrolles, (2002).

[15] K.Scott, «*Unified Process Explained* », Addison-Wesley, (2002).

[16] C.Larman, « *UML et les Design Patterns* », Campus Press, (2002).

[17] Alain Muller et Nathalie Gaertner , *« Modélisation objet avec UML »*, Eyrolles (2004).

[18] Gerson Sygné, *« UML ï Mécanismes dêextension »*, cours (2005).

[19] Baroudi Rachid, *« Sécurité dûn système de remboursement par UMLsec »*, Mini projet Université d'Oran Es-Sénia – Magister, (2006).

[20] I. Jacobson, G. Booch, J. Rumbaugh, « *Unified Modeling Language Users Guide*», Addison Wesley Longman, (1999).

[21] G. Booch, *« Object Solutions: Managing the Object-Oriented Project»*, Addison Wesley, (1996).

[22] CNRS, « Sécurité informatique : numéro 31,… ,35 », Site : http://www.cnrs.fr/Infosecu, Revue (2001).