3
Uso de estratégias orientadas a metas

A abordagem proposta para o tratamento dos requisitos de segurança está baseada na adoção de estratégias orientadas a metas para definição dos requisitos. Estratégias orientadas a metas caracterizam-se pela construção de modelos intencionais, cujo foco se encontra nas metas a serem alcançadas pelos atores, ou seja, em suas intenções e motivações. A aplicação destas estratégias orientadas a metas foi baseada na utilização do *framework* i*, apresentado anteriormente, cuja intencionalidade é distribuída entre os atores. Neste capítulo, será apresentado (a título de introdução e de forma bastante resumida) o processo de engenharia de requisitos. Em seguida, apresentamos o processo proposto por Liu, Yu e Mylopoulos [3]. Depois apresentamos a evolução do processo proposta por nós, detalhando os passos para definição de requisitos gerais e os passos específicos para definição de requisitos de segurança.

3.1 Processo de Engenharia de Requisitos

De acordo com Leite [30], o processo de engenharia de requisitos é composto de quatro macro-atividades: elicitação, modelagem, análise e gerência, conforme apresentado na Figura 3.1. A atividade de elicitação tem como principal objetivo a aquisição de informações (e requisitos), oriundas do "Universo de Informações". A atividade de modelagem tem por objetivo documentar os requisitos através de modelos. A atividade de análise tem por objetivo a verificação e validação dos requisitos modelados. A atividade de gerência é desempenhada paralelamente a estas três atividades, com o objetivo de evoluir os requisitos de forma controlada e garantir a rastreabilidade entre os requisitos.

PUC-Rio - Certificação Digital Nº 0410836/CA

Nesta dissertação, chamaremos de "Definição de Requisitos" a composição das atividades de elicitação, modelagem e análise, conforme indicado na Figura 3.1.

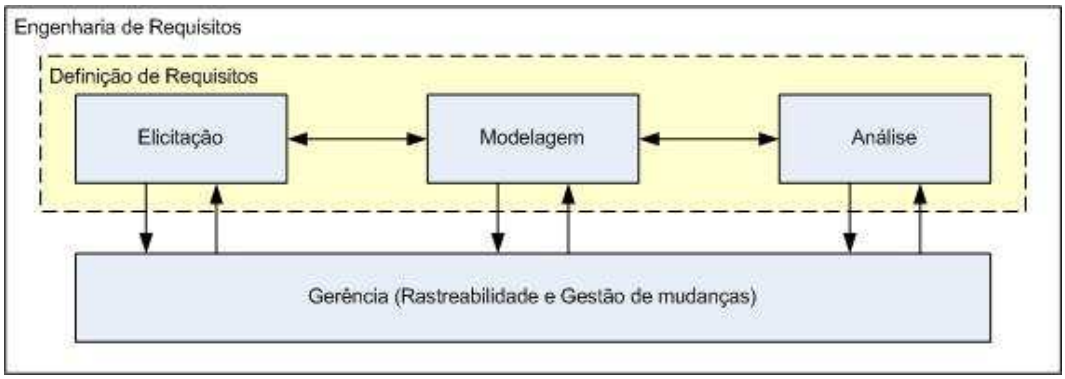

Figura 3.1 - Visão geral da Engenharia de Requisitos, adaptada de [30]

PUC-Rio - Certificação Digital Nº 0410836/CA

3.2 Definição de requisitos usando i*

Neste trabalho será utilizada uma extensão do processo de definição de requisitos usando i*, originalmente elaborado por Liu, Yu e Mylopoulos [3]. O processo tem por objetivo incorporar requisitos de segurança (não-funcionais) à definição de requisitos gerais, conforme representado na Figura 3.2.

O processo original é composto por duas partes: (i) um processo básico de definição de requisitos gerais (representado pelas caixas com linhas sólidas, à esquerda) e (ii) um processo para definição de requisitos específicos de segurança (representado pelas caixas com linhas tracejadas, à direita). O processo de definição de requisitos gerais apresenta os passos para construção e refinamento dos modelos SD e SR do *framework* i*. A este processo, são integrados os passos referentes ao processo de definição de requisitos específicos de segurança.

3.3 Evolução do processo de definição de requisitos

Nosso trabalho propõe a evolução do processo proposto por Liu, Yu e Mylopoulos em [3] com a incorporação de alguns passos ao processo de definição

de requisitos gerais. Estes passos foram introduzidos com os objetivos de: ajudar a lidar com a complexidade (para análise) dos modelos produzidos pelo *framework* i*; melhorar o entendimento dos atores e suas relações; tornar explícito o processo de definição dos requisitos não-funcionais, em especial, os de segurança; e, auxiliar a validação dos requisitos pelos usuários e demais interessados.

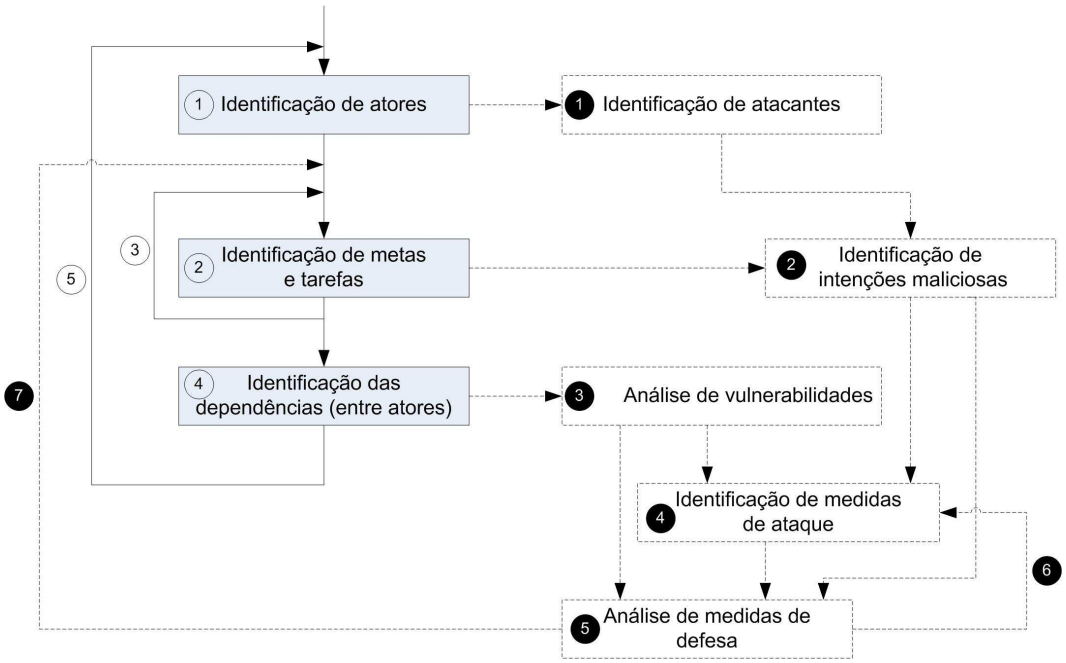

Figura 3.2 - Processo de definição de requisitos usando i* de Liu, Yu e Mylopoulos [3]

Os modelos SR produzidos em i* são grafos com vários tipos de nós (ator e seus refinamentos, meta, tarefa, recurso e meta flexível) e dois tipos de elo (meios-fim e decomposição de tarefa). Freqüentemente estes modelos tornam-se complexos (para análise): algumas tarefas alternativas para se alcançar uma mesma meta, detalhamento das tarefas alternativas através de sua decomposição. A adição dos requisitos de segurança torna estes modelos ainda mais complexos (para análise). Por este motivo foi introduzido um passo para definição de *SDSituations*. Para auxiliar o entendimento dos atores e suas relações foi introduzido um passo específico para refinamento dos atores, com elaboração de modelos SA (*Strategic Actor*). No processo original [3], os atores são modelados a partir dos passos identificação de atores e identificação de atacantes. Porém, os

PUC-Rio - Certificação Digital Nº 0410836/CA

atores (refinados ou não em posições, papéis e agentes) aparecem diretamente nos modelos SD e SR, elaborados nos passos subseqüentes. Com a introdução, feita por nós, de um passo específico para elaboração dos modelos SA, nossa intenção é que seja despendida uma atenção especial aos atores e suas relações durante a elaboração destes modelos, que servirão de referência (dos atores e relações) para o restante do trabalho.

A definição de cenários foi introduzida com o objetivo de auxiliar a validação dos requisitos. Por se tratar de uma estrutura descritiva, o uso de cenários prescinde do entendimento da semântica de uma notação gráfica rica como a de i* pelos atores validadores dos requisitos.

Os requisitos não-funcionais, particularmente os de segurança, precisam ser detalhados a um nível adequado para que análise de alternativas de operacionalização se torne viável. Para abordar esta questão, foi introduzido um passo para definição de requisitos não-funcionais.

As alterações introduzidas estão descritas a seguir:

(i) Definição de *SDsituations*

Conforme mencionado anteriormente, uma *SDsituation* é uma forma de representação estruturada de uma situação de dependência estratégica [4]. Uma *SDsituation* é composta de um ou mais elementos de dependência, e cada *SDSituation* pode ser identificada separadamente das outras, formando uma rede de interdependências. A interdependência entre *SDsituaions* pode ser física, lógica ou temporal. O uso de *SDsituations* ajuda a lidar com a complexidade inerente aos modelos de interdependências intencionais [4], sendo este o principal motivo para a sua incorporação ao processo de análise de requisitos. De fato, é mais fácil trabalhar com *SDsituations* do que com todas as dependências ao mesmo tempo, como pôde ser verificado no estudo de caso, apresentado mais adiante.

PUC-Rio - Certificação Digital Nº 0410836/CA

(ii) Definição de Cenários

Um cenário é uma descrição estruturada de uma situação que ocorre no mundo real [5]. Situações têm algumas características: são concretas e possuem metas ou objetivos; envolvem atores e necessitam de recursos; são individualmente independentes e inter-relacionadas; acontecem em um local e tempo definidos e podem ter restrições (uma restrição pode ao mesmo tempo qualificar ou designar alguma restrição ao cenário e é de certa maneira um requisito não funcional [8]). Um cenário satisfaz uma meta ou objetivo e tem um contexto que especifica suas fronteiras. Cenários são compostos por episódios que definem seus requisitos funcionais. Pode haver cursos alternativos de execução, representados por exceções, que o cenário deva considerar. O principal motivo para introdução da definição de cenários, ao processo de definição de requisitos gerais, é que são estruturas descritivas e, por isso, são mais adequados para a interação com os usuários finais, que se sentem mais confortáveis em validar um artefato descrito em linguagem natural. Um cenário usa a linguagem do "Universo de Informações". Esta linguagem é capturada através do Léxico Estendido da Linguagem (LEL) [6] que é um modelo de representação de seus termos, composto por símbolos (verbo, objeto, sujeito e estado) que representam palavras ou sentenças. Cada símbolo é identificado por um nome, ou nomes no caso de haver sinônimos, e representado por duas descrições: a primeira, chamada noção, é a denotação do símbolo, equivalente a uma descrição encontrada no dicionário; a segunda, chamada de impacto ou respostas comportamentais, é a conotação do símbolo que descreve a contextualização do símbolo no "Universo de Informações".

(iii) Refinamento de Atores

A análise dos atores e dos atacantes é um ponto fundamental para questões de segurança. Este passo tem por objetivo a construção dos modelos SA (vide seção 2.4.1) dos atores legítimos e dos atacantes.

PUC-Rio - Certificação Digital Nº 0410836/CA

Com estes modelos os relacionamentos estruturais (estáticos) entre os atores ficam evidenciados. Os relacionamentos do tipo ‘desempenha’ e ‘ocupa’ são de especial interesse para análise de segurança.

(iv) Definição de Requisitos Não-Funcionais (NFRs – *Non-Functional Requirements*)

A incorporação de um passo específico para Definição de Requisitos Não-Funcionais tem por objetivo detalhar uma forma de definição destes requisitos. Algumas questões relevantes relativas aos requisitos não-funcionais precisam de respostas explícitas. Exemplos destas questões são: “Qual a origem dos requisitos não-funcionais?”; “São requisitos de quais atores?”; “São requisitos do sistema como um todo ou cada cenário possui seus próprios requisitos não-funcionais?”; “Qual a relação entre os requisitos não-funcionais? Por exemplo: qual o impacto de requisitos de segurança sobre os requisitos de usabilidade?”. Uma outra vantagem da introdução deste passo é representar separadamente o requisito não-funcional das possíveis soluções a serem adotadas para atendê-lo. Por serem requisitos de qualidade, os requisitos não-funcionais são representados por metas flexíveis no *framework* i*, aplicando-se a eles o conceito de *satisficy* (assim como para as metas flexíveis, não se pode afirmar que os requisitos não-funcionais são plenamente satisfeitos; pode-se afirmar apenas que são “satisfeitos em um nível considerado adequado”). A elicitação dos requisitos não-funcionais será discutida detalhadamente adiante, na descrição do processo.

Além da introdução destes novos passos, foi feita uma troca na ordem de execução entre o passo 2 – “Identificação de metas e tarefas” e o passo 4 – “Identificação das dependências (entre atores)”. Esta troca se mostrou necessária para introdução dos novos passos, além de não trazer nenhum prejuízo ao processo (os modelos SD e SR são mutuamente complementares e, em [1], não há uma recomendação sobre a ordem seqüencial para elaboração destes modelos).

PUC-Rio - Certificação Digital Nº 0410836/CA

A evolução do processo, a partir da introdução destas alterações no processo original (Figura 3.2), gerou um novo fluxograma de processo no qual está baseado o estudo de caso apresentado neste trabalho, apresentado na Figura 3.3. As caixas com linhas sólidas, à esquerda na Figura 3.3, indicam os passos da definição de requisitos de gerais. As caixas sombreadas indicam os passos acrescentados ao processo original. As setas em azul foram acrescentas em virtude da inserção dos novos passos. Os passos do processo de definição de requisitos gerais são detalhados na seção 3.4.1, a seguir, enquanto os passos do processo de definição de requisitos específicos de segurança são detalhados mais adiante, na seção 3.4.2 deste capítulo.

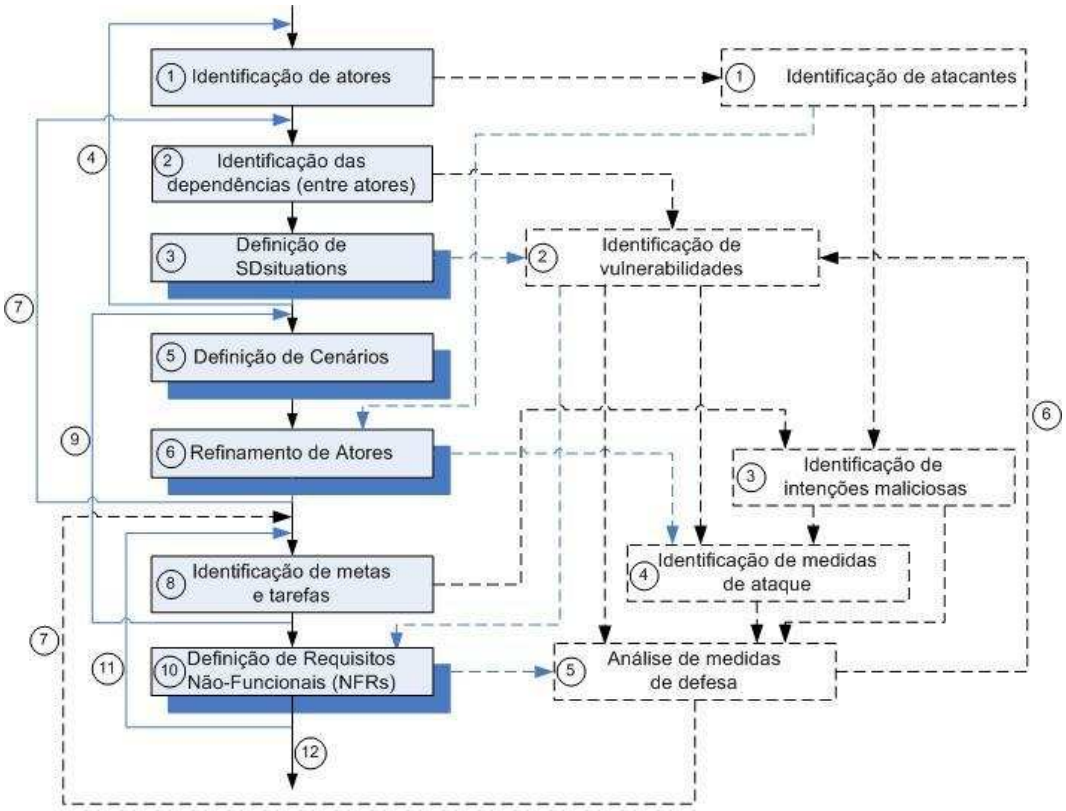

Figura 3.3 - Fluxograma do processo evoluído a partir do processo de Liu, Yu e Mylopoulos [3]

3.4 Detalhamento do processo evoluído

3.4.1 Detalhamento do processo evoluído de definição de requisitos gerais

Os passos do processo de definição de requisitos de gerais (caixas em azul, à esquerda na Figura 3.3) estão descritos detalhadamente a seguir:

PUC-Rio - Certificação Digital Nº 0410836/CA

① Identificação de atores – este passo responde a questão "quem está envolvido no sistema?". Inicialmente são identificados atores humanos, representando os *stakeholders*, juntamente com os atores computacionais. Com o progresso da definição de requisitos, à medida que forem sendo tomadas algumas decisões de projeto durante a modelagem, mais atores serão identificados (passo 7 – Figura 3.3) como agentes de software, por exemplo. Este passo, identificação de atores, situa-se principalmente na atividade de elicitação de requisitos. Em [3], neste passo já é feita o refinamento dos atores em agentes, papéis e posições. Nós preferimos postergar este refinamento para o passo 6 – refinamento de atores, após serem elaborados os modelos SD e definidas as *SDsituations* e cenários. Assim, há mais subsídios para o refinamento dos atores.
② Identificação das dependências entre atores – este passo responde a questão "como os atores se relacionam uns com os outros?". As respostas a esta questão são representadas em um modelo SD. No *framework* i* o foco é o relacionamento intencional (ex: um ator depende de outro para que uma meta seja satisfeita) e não no fluxo de informações (ex: que mensagem um ator manda para outro) entre os atores. Este passo inicia-se na atividade de elicitação de requisitos, com a identificação das dependências, avançando até a fase de modelagem de requisitos, com a elaboração do modelo SD.
③ Definição de *SDsituations* – o objetivo deste passo é definir as situações de dependências entre os atores. Esta definição é feita da seguinte forma [4]: (i) identificação dos elementos da *SDsituation* - identificar atores (*depender* e *dependee* podem ser identificados a partir do modelo SD). Os elementos de dependência correspondem aos *dependums* (metas, tarefas, recursos e metas flexiveis) no modelo SD; (ii) composição de *SDSituations* – Examinando as dependências no modelo SD, o engenheiro de requisitos deve definir que

PUC-Rio - Certificação Digital Nº 0410836/CA

dependências (*depender* + *dependum* + *dependee*) entre os atores fazem parte de uma mesma situação de dependência estratégica (*SDsituation*). Estas dependências são agrupadas formando uma *SDsituation*;

(iii) identificação de interrelacionamento entre *SDsituations* – identificação de dependências (lógica, física ou temporal) entre as *SDsituations.* Apesar de uma *SDsituation* ser composta de uma ou mais dependências, uma dependência pode ser uma outra *SDsituation.*

Este passo situa-se na fase de modelagem de requisitos.

④ Passo iterativo: identificação de novos atores – após a identificação das situações de dependência entre os atores, novos atores podem ser identificados. Para tanto, deve-se examinar todas as dependência presentes em cada situação verificando se o *dependee* pode, por si só ou com a colaboração dos outros atores já identificados, 'honrar' os acordos estabelecidos com o *depender* representados pelos relacionamentos de dependência estratégica. Ou seja, deve se verificar se o *dependee* pode, sozinho ou com a colaboração dos demais atores identificados, satisfazer as metas, desempenhar as tarefas, disponibilizar os recurso ou 'satisfazer a contento' as metas flexíveis acordadas. A impossibilidade de 'honrar' os acordos indica a necessidade de novos atores, que devem ser identificados, que colaborem para o cumprimento das metas acordadas. Uma vez identificados estes atores, a colaboração será estabelecida via relacionamento de dependência estratégica, com a complementação do modelo SD elaborado previamente, de forma iterativa. Este passo reflete uma iteração entre as ativades de modelagem e elicitação de requisitos.

⑤ Definição de cenários – o objetivo deste passo é definir os cenários correspondentes às situações de dependência no modelo SD. A seguir são apresentados os passos para definição de cenários, juntamente com algumas heurísticas, propostos em [4] e [17], com adaptações:

(i) Criação dos cenários: embora seja possível que um cenário tenha mais de uma *SDsituation*, o usual é ter um cenário para cada

PUC-Rio - Certificação Digital Nº 0410836/CA

SDsituation;

(ii) Descrição dos cenários: descrever as informações do cenário, a saber:

a) Objetivo (ou meta) - considerar o melhor elemento de dependência da situação. A primeira escolha é uma dependência de meta, a segunda de tarefa e por último uma dependência de recurso. Uma dependência de meta flexível não pode ser o objetivo do cenário por que uma meta flexível está sempre junto de outro elemento de dependência;

b) Atores - os atores do cenário são os mesmos atores da *SDsituation*;

c) Contexto – esta informação não existe em *SDsituation*, mas pode ser incluído na descrição do cenário pelo engenheiro de requisitos;

d) Pré-condições - devem ser listados todos os recursos preparados em *SDsituations* anteriores e necessários à *SDsituation* corrente;

e) Episódios – descrição funcional dos detalhes do cenário, em forma de sentença. As sentenças descrevem as tarefas desempenhadas pelos atores no cenário. Algumas tarefas podem aparecer dentro de uma dependência de tarefa em uma *SDsituation*, derivadas do modelo SD, outras aparecerão nos modelos SD dos atores (elaborados no próximo passo). Com o refinamento dos cenários no passo 6, mais tarefas poderão ser acrescentadas aos episódios;

f) Restrições – designam metas flexíveis (NFRs) que devem ser consideradas e descrevem imposições do processo;

g) Exceções – representam alternativas na definição do cenário.

PUC-Rio - Certificação Digital Nº 0410836/CA

Na primeira execução deste passo, serão definidos cenários incompletos. Com a execução do passo 9 (iterativo), os cenários serão complementados.
(6) Refinamento de atores – neste passo é construído um modelo SA. O foco do modelo SA são as relações estruturais entres os atores. Para tanto, os atores identificados anteriormente nos passos 1 e 2 são refinados em papéis, posições e agentes no modelo SA. Algumas heurísticas são usadas para construção do SA: primeiramente, atores que possuam características individuais relevantes para o contexto da aplicação e que não possam ser transferidas para outros atores devem ser desdobrados em agentes; para os demais atores no modelo SD, são criados papéis que representem a participação do ator nas *SDSituations*. Para as *SDSituations* complementares (com dependência lógica), pode ser criado apenas um papel. Para os atores que foram desdobrados em mais de um papel, pode se criar uma posição que ocupe estes papéis. Este refinamento de atores aplica-se também aos atacantes identificados no passo 1 (definição de requisitos específicos de segurança).
(7) Passo iterativo: refinamento de modelos SD, cenários e modelos SR – com o progresso da definição de requisitos os atores já identificados podem ser refinados em papéis, posições ou agentes. Os modelos SD elaborados anteriormente, passo 2, são refinados com base nos modelos SA elaborados no passo 6.
(8) Identificação de metas e tarefas – este passo responde a questão "o que os atores desejam alcançar?". As respostas a esta questão podem ser representadas por metas que capturam os objetivos de alto-nível dos atores. Para estas metas são atribuídas tarefas alternativas, através de elos meios-fim, cuja execução possibilite o alcance das metas. Tarefas, por sua vez, podem ser decompostas em sub-tarefas, recursos, submetas (metas cujo alcance seja necessário para execução da tarefa) e metas flexíveis (que definem os requisitos de qualidade para execução da tarefa).
(9) Passo iterativo: refinamento de cenários – o refinamento das metas, das metas flexíveis e das tarefas deve prosseguir até alcançar um estágio em que as decisões de desenho (*design*) possam ser tomadas com base nas informações

PUC-Rio - Certificação Digital Nº 0410836/CA

contidas nos modelos e cenários. À medida que as metas e tarefas são refinadas os episódios dos cenários correspondentes são enriquecidos, complementando a definição dos cenários.

(10) Definição de requisitos não-funcionais (NFRs) – este passo tem por objetivo integrar os requisitos não-funcionais detalhados, relativos a cada *SDsituation*, aos modelos SR de cada ator, com os interrelacionamentos inerentes a estes requisitos. A abordagem é a seguinte:

(i) Criação de meta flexíveis –.É extremamente recomendado o uso de catálogos de NFRs, como proposto em [7] e [20], para a elicitação dos requisitos não-funcionais (metas flexíveis). Catálogos são uma forma interessante de se armazenar e reusar conhecimento a respeito dos requisitos não-funcionais: como estão inter-relacionados, quais as soluções conhecidas e que impacto estas soluções tem em outros requisitos. Em geral, o uso de um Catálogo de NFRs requer a adequação destes NFRs ao contexto da aplicação. Esta adaptação pode ser feita via metódos de refinamento do próprio catálogo [7]. As soluções encontradas para um determinado contexto poderão ser adicionadas ao catálogo para uso posterior;

(ii) Decomposição de requisitos não-funcionais – uma vez elicitados os requisitos não-funcionais, é necessário refiná-los até que alcancem um baixo nível de abstração em que seja possível identificar alternativas para sua operacionalização. Este refinamento é elaborado via métodos de decomposição [7]. Os requisitos de alto nível de abstração são decompostos primeiramente por tipos e, uma vez decompostos por tipos, são novamente decompostos por tópicos. Em uma decomposição por tipos, os requisitos não-funcionais de nível de abstração mais alto são decompostos em requisitos de características mais específicas. Em uma decomposição por tópicos, ocorre uma parametrização dos requistos não-funcionais por tópicos (“um *tópico* de uma meta flexível pode se referir a uma coleção de

PUC-Rio - Certificação Digital Nº 0410836/CA

itens, tais como informações, processos, funções, eventos, interfaces, tempo, quantia monetária, etc.” [7]). A decomposição dos requisitos não-funcionias, tanto por tipos quanto por tópicos, quebra um requisito “pai” (de nível de abstração mais alto) em requisitos “filhos” usando elos de contribuição “E” (*AND*). Os métodos de decomposição possibilitam o exercício do paradigma “dividir para conquistar” [7].
(11) Passo iterativo: adição dos requisitos não funcionais aos modelos SR – Neste passo, os requisitos não-funcionais definidos no passo 10 são adicionados aos modelos SR, elaborados no passo 8. A execução destes passos deve ocorrer de forma iterativa até que os modelos estejam refinados o suficiente para que seja possível tomar decisões de desenho (*design*).
(12) Verificação de consistência entre modelos (passo final): ao final do processo é necessário verificar a consistência entre os modelos produzidos, como condição de parada para execução do processo. Caso os modelos ainda não estejam consistentes entre si, devem ser executados os passo iterativos 11, 9, 7 e 4.

PUC-Rio - Certificação Digital Nº 0410836/CA

3.4.2 Detalhamento do processo evoluído de definição de requisitos gerais

As caixas com linhas tracejadas, à direita da Figura 3.3, indicam os passos do processo de definição de requisitos específicos de segurança. Estes passos estão integrados ao **processo de definição de requisitos gerais** de forma a permitir que sejam antecipadas ameaças de atacantes potenciais e que sejam procuradas e adotadas contramedidas para proteção do sistema. Os passos relativos à definição de requisitos específicos de segurança estão descritos a seguir:

(1) Identificação de atacantes – este passo tem por objetivo identificar potenciais atacantes ao sistema. A premissa básica é que todos são considerados

"culpados até que provem sua inocência". Em outras palavras, cada um dos os atores (agentes, papéis ou posições) identificados no processo de definição de requisitos gerais podem ser considerados um potencial atacante ao sistema ou aos outros atores. Estes atacantes herdam as intenções, capacidades e relacionamentos sociais dos atores legítimos correspondentes, isto é, a hierarquia interna de metas e os relacionamentos de dependência externa no modelo. O termo atacante é usado para referenciar a fonte de qualquer ameaça. Uma fronteira é delineada para o sistema e, então, é feita uma busca exaustiva por atacantes dentro desta fronteira. Nesta abordagem, todos os atacantes são considerados atacantes "internos". Além dos "atacantes internos" derivados dos atores legítimos, identificados à esquerda da Figura 3.3 no passo 1 - Identificação de Atores, outros atacantes podem ser adicionados ao modelo. A identificação destes outros potenciais atacantes, que chamaremos de "atacantes externos", pode ser derivada das respostas para a pergunta: "Quem poderia se beneficiar de um ataque (bem-sucedido) à segurança do sistema?". O exame de informações históricas de ataques a aplicações do mesmo domínio, ou aplicações que usem o mesmo tipo de tecnologia, também pode auxiliar na identificação dos atacantes externos. Neste contexto, atacantes aleatórios como *hackers/crackers* da internet são representados como "atacantes externos" compartilhando o mesmo território que suas vítimas. Este passo é parte da atividade de elicitação de requisitos (específicos de segurança).
❷ Identificação de vulnerabilidades – Este passo visa identificar pontos de vulnerabilidade na rede de dependências (uma **vulnerabilidade** configura um risco [38, p 238] para o sistema e pode vir a ser explorado por um atacante, ou mesmo vir a ocorrer acidentalmente, sem intenção). A idéia básica é que relacionamentos de dependência trazem vulnerabilidades para o sistema e para o ator dependente (*depender*). Os elementos de dependência, presentes nos relacionamentos de dependência entre os atores, fazem parte da *superfície de ataque* do sistema (*superfície de ataque* de um sistema é definida em termos das ações que são visíveis externamente para seus usuários e dos recursos do sistema que ação acessa ou modifica [9]). Os relacionamentos de dependência constituem um bom ponto de partida para a identificação de vulnerabilidades. Atacantes em potencial podem explorar estas vulnerabilidades para atacar o

PUC-Rio - Certificação Digital Nº 0410836/CA

sistema e satisfazer suas intenções maliciosas. As questões a serem respondidas neste passo são: “Que relações de dependência são vulneráveis a ataques?”, “Qual o efeito (em cascata) se um relacionamento de dependência for comprometido?”.

A identificação de vulnerabilidades não se encerra com a identificação dos pontos de vulnerabilidade. É necessário examinar toda rede de dependências e verificar onde os relacionamentos de dependência atacados impactariam outros atores. O exame da rede de dependências pode ser iniciado a partir das *SDsituations*, identificando-se quais elementos de dependência de cada *SDsituation* são críticos em relação a determinados requisitos de segurança - uma vez determinado um ponto de vulnerabilidade, devem ser examinadas as outras *SDsituations* relacionadas (que possuam alguma dependência com a *SDstiuation* em exame) para avaliar o impacto da vulnerabilidade detectada. A modelagem de dependências em i* permite uma identificação de vulnerabilidades mais detalhada porque a falha potencial de uma dependência pode ser rastreada para o *depender* e para o *dependee* envolvidos. As vulnerabilidades identificadas nas redes de dependência (e nas respectivas *SDsituations*) sinalizam a necessidade de se adicionar metas flexíveis aos modelos SR dos atores envolvidos na dependência. É possível também identificar vulnerabilidades na infra-estrutura usada pelo sistema se recursos como internet, sistema operacional e outros softwares básicos forem tratados como recursos em i*. Este passo é parte da atividade de elicitação de requisitos (específicos de segurança).

❸ Identificação de intenções maliciosas – Para cada atacante, identificado no passo 1 – identificação de atacantes, são identificadas, ou atribuídas, metas de alto nível que os atacantes desejem satisfazer ao atacar o sistema. Esta análise pode revelar a apropriação de recursos e capacidades legítimas para uso ilícito. Através do uso ilícito de recursos ou do exercício ilícito de ações do sistema o atacante visa se beneficiar de alguma forma. As questões chave deste passo são: “Que objetivos os atacantes desejam satisfazer ao atacar o sistema?”; “De que formas um atacante poderia se beneficiar se, ilicitamente, desempenhasse as ações ou acessasse os recursos do sistema (se possuísse as capacidades de um

PUC-Rio - Certificação Digital Nº 0410836/CA

ator real)?” Para responder a esta questão as relações de dependência (nos modelos SD) e as capacidades/habilidades dos atores legítimos (nos modelos SR) devem ser examinadas em busca do benefício potencial que um atacante poderia obter apropriando-se ilicitamente de recursos valiosos ou desempenhando ilicitamente as tarefas atribuídas aos atores legítimos. Durante a execução deste passo são elaborados modelos SR dos atacantes com as intenções maliciosas dos atacantes modeladas como metas. Este passo é parte da atividade de elicitação (identificação de intenções maliciosas) e também da atividade de modelagem de requisitos, com a elaboração de modelos intencionais dos atacantes.
❹ Identificação de medidas de ataque – As medidas de ataque configuram ameaças à segurança do sistema. Este passo foca em como um atacante pode atacar os pontos de vulnerabilidade, identificados anteriormente, a fim de alcançar suas intenções maliciosas. O objetivo é identificar as diferentes medidas de ataque alternativas (meios) que possibilitem aos atacantes satisfazerem suas intenções maliciosas (fins). A pergunta chave deste passo é: “Quais as alternativas para satisfazer as intenções maliciosas?”. Durante a execução deste passo, os modelos SR dos atacantes, elaborados no passo anterior, são refinados com as alternativas de ataque modeladas como tarefas que se relacionam com as intenções maliciosas através de elos “meios-fim”. As estratégias identificadas nestes modelos SR configuram as medidas de ataque. Este passo é parte da atividade de elicitação (identificação de intenções maliciosas) e também da atividade de modelagem de requisitos, com o refinamento dos modelos intencionais dos atacantes produzidos no passo anterior.
❺ Análise de medidas de defesa – Este passo consiste em identificar e analisar as alternativas de defesa para as ameaças previamente identificadas (via medidas de ataque). São fatores necessários para o sucesso de um ataque: a motivação do atacante, as vulnerabilidades do sistema e a capacidade do atacante de efetuar o ataque. Para neutralizar um ataque hipotético, é preciso encontrar medidas que neguem suficientemente estes fatores, eliminando a possibilidade da ocorrência de ataques ou reduzindo o impacto destes ataques

PUC-Rio - Certificação Digital Nº 0410836/CA

caso venham a ocorrer.

As medidas de defesa correspondem às alternativas de operacionalização para os requisitos não-funcionais de segurança refinados. Alternativas de operacionalização, representadas por tarefas, são relacionadas via elos de contribuição (meios-fim) que representam um refinamento "OU" (*OR*) às metas flexíveis (requisitos não-funcionais). Em geral, estes elos de contribuição possuem como atributo o valor de contribuição (que pode ser positivo ou negativo).

As alternativas de operacionalização também podem constar de um catálogo de requisitos não-funcionais, uma vez que podem ser reaproveitadas, embora com potencial de reuso menor que os requisitos refinados por tipos.

A ídeia é que sejam consideradas diversas alternativas de operacionalização. Uma vez que as diversas alternativas estejam explicitadas, com as respectivas contribuições (positivas e negativas) para os demais requitos não-funcionais, a escolha deve ser feita de acordo com a prioridade dos requistos impactados.

Através da avaliação do efeito das medidas de defesa contra as ameaças é possível decidir se os impactos das ameaças foram reduzidos a um nível aceitável. Esta análise é feita iterativamente até que uma solução satisfatória seja encontrada.

Um ponto relevante é que uma vulnerabilidade pode trazer danos a um sistema sem que haja necessariamente um ataque (no sentido de haver um atacante disposto a explorar uma vulnerabilidade do sistema em proveito próprio). Por ser i* um *framework* intencional, esta questão não é diretamente abordada neste processo. Para contornar esta omissão, a análise de medidas de defesa deve considerar não apenas as vulnerabilidades do sistema passíveis de ataque (intencionais), mas também vulnerabilidades às faltas não-intencionais, como acidentes, desastres naturais, mau-funcionamento de hardware e/ou software. Identificadas as medidas de defesa alternativas é feita uma análise para escolha da(s) medida(s) mais adequada(s). Uma alternativa de operacionalização, que tem impacto positivo para o requisito não-funcional que se deseja operacionalizar, pode ter impacto negativo em outros requisitos não-funcionais.

PUC-Rio - Certificação Digital Nº 0410836/CA

A escolha da alternativa de operacionalização mais adequada é norteada pela priorização dos requisitos não-funcionais entre si, considerando-se também probabilidade de uma vulnerabilidade vir a ser explorada, ou seja, do risco que a vulnerabilidade configura vir a se concretizar. Este passo faz parte das atividades de modelagem e análise de requisitos.
❻ Passo iterativo: identificação de outras medidas de ataque – A adição de medidas de defesa pode trazer novas vulnerabilidades ao sistema. É necessário fazer um novo ciclo de identificação de vulnerabilidades, a fim do qual, uma nova análise de medidas de defesa será disparada.
❼ Incorporação das medidas de defesa escolhidas – As medidas de defesa escolhidas, dentre as alternativas estudadas, são incorporadas ao modelo finalizando a modelagem dos requisitos de segurança. Com a incorporação das medidas de defesa, os modelos SR são atualizados disparando em seqüência a atualização dos Cenários, via passo 9 (definição de requisitos gerais), integrando os requisitos de segurança com as soluções escolhidas à definição dos requisitos gerais.

Ao final deste processo, os cenários terão incorporados de forma detalhada os requisitos de segurança tratados e as respectivas soluções adotadas. Estes cenários detalhados podem ser usados na validação dos requisitos (tantos os gerais, quantos os específicos de segurança). A validação dos requisitos [33] pode ser feita através da leitura dos requisitos, expressos na forma de cenários, pelos clientes e demais interessados. Conforme mencionado anteriormente, acreditamos que para clientes e interessados que não estejam bem familiarizados com o *framework* i* seja mais fácil a validação dos requisitos através da leitura dos cenários que pela leitura dos modelos SD, SR e SA.

PUC-Rio - Certificação Digital Nº 0410836/CA