

6 Conclusões

Nesta tese apresentamos uma abordagem para automação parcial de atividades associadas à verificação e validação de requisitos. Neste capítulo estruturamos as seções de forma a apresentar na seção 6.1 um resumo das principais características da abordagem proposta, contribuições desta tese e aspectos da ferramenta para suporte à abordagem. Na seção 6.2 comparamos nosso trabalho com outros de alguma forma relacionados, e discutimos limitações e restrições na seção 6.3, onde também relacionamos trabalhos futuros.

6.1. Abordagem proposta e contribuições

A estratégia proposta trabalha com requisitos escritos em linguagem natural, dado que isto propicia maior envolvimento de clientes e usuários no processo de requisitos como um todo, e especificamente nas atividades de verificação e validação. As estratégias propostas consideram também aspectos próprios do desenvolvimento distribuído de software, com interessados desempenhando atividades em ambientes geograficamente separados.

A abordagem proposta está baseada em:

- uso de técnicas de tratamento da linguagem natural para automação parcial de atividades de verificação e validação
- uso de agentes de software encapsulando ferramentas utilizadas no tratamento da linguagem natural como *Web services*
- agentes de software sendo utilizados como agentes pessoais, agindo de forma autônoma e apoiando atividades necessárias a verificação e validação de requisitos

As contribuições desta tese são oriundas de duas vertentes de trabalho: (i) uso intensivo de técnicas de processamento da linguagem natural para avaliação de documentos gerados ou manipulados no processo de requisitos, e (ii) uso de

agentes de software e *web services* para possibilitar o reuso de ferramentas disponibilizadas pela comunidade de processamento da linguagem natural e também para possibilitar que algumas ações necessárias ao processo de V&V sejam realizadas de forma autônoma.

Com uso de técnicas de processamento da linguagem natural criamos procedimentos para geração de visões de requisitos, criação ou atualização do léxico da aplicação e detecção de erros, omissões e discrepâncias no conjunto de requisitos.

Com agentes de software desenvolvemos uma estratégia para uso de agentes pessoais dos participantes do processo de requisitos, permitindo que ajam de forma autônoma em nome do ator humano que representam. A estrutura criada possibilita também que mudanças em requisitos sejam detectadas pelos agentes participantes da plataforma, que podem então tomar decisões ou notificar o ator a quem estão associados.

6.1.1. Estratégias com uso de técnicas de processamento da linguagem natural

A **geração de visões de requisitos** (seção 3.1) está intimamente relacionada ao agrupamento de requisitos. Utilizando uma estratégia para extração de termos relevantes do próprio documento de requisitos, uma taxonomia em dois níveis é criada e utilizada para categorizar os requisitos, gerando agrupamentos onde os elementos possuem características comuns. A criação da taxonomia é semi-automática e está baseada em técnicas de extração de terminologia e na identificação de colocações.

Visões textuais e gráficas dos requisitos são geradas a partir dos agrupamentos, possibilitando aos participantes das atividades de V&V o trabalho sobre grupos menores e possivelmente menos complexos de requisitos. Para a geração de visões gráficas utilizamos mapas de tópicos, o que permite aos usuários a navegação e rápida percepção de relacionamentos entre requisitos.

Também possibilitamos que os usuários solicitem agrupamentos de requisitos simplesmente fornecendo um termo ou expressão relacionada ao seu trabalho. Desta forma ele pode criar agrupamentos de acordo com seu interesse ou perfil de atuação profissional e avaliar tais agrupamentos em relação a

características como completude, por exemplo.

O objetivo na criação dos agrupamentos é apoiar atividades de V&V, mas tais agrupamentos também podem ser utilizados para, por exemplo, definir alocação de requisitos a componentes de software ou mesmo atribuir agrupamentos a incrementos de software, no caso de se utilizar o desenvolvimento incremental de software.

A criação ou atualização automática do léxico da aplicação (seção 3.2) está baseada na identificação de termos não dicionarizados e na identificação de atores e recursos referidos nos documentos.

A identificação de termos não dicionarizados baseia-se na premissa que termos ausentes de dicionários usuais da língua fazem referência a termos próprios do domínio da aplicação ou da própria área de sistemas de informação. Tais termos, por possibilitarem diferentes interpretações, devem constar do léxico da aplicação. Esta estratégia pode ser utilizada também para apoiar a criação de um léxico para a organização, se aplicada a conjuntos de documentos relacionados aos negócios da organização.

A necessidade de léxicos abrangentes e contextualizados se faz mais presente em ambientes distribuídos de desenvolvimento, nos quais problemas de comunicação gerados por diferenças culturais, fusos horários e mesmo diferentes competências lingüísticas podem dificultar a realização de atividades do processo de requisitos. O léxico é uma ferramenta extremamente necessária para evitar ambigüidades dos termos próprios ao domínio da aplicação, considerando os problemas que poderiam decorrer de uma interpretação equivocada dos requisitos.

A identificação de atores e recursos para comporem o léxico da aplicação está fundamentada extração de sintagmas nominais do texto de documentos gerados ou manipulados pelo processo de requisitos. A estratégia proposta é baseada na utilização de um *POS tagger*, que identifica as classes gramaticais das palavras do texto, e na extração de sintagmas nominais que atendem a um conjunto de padrões pré-definidos.

Os padrões utilizados para a identificação de atores basearam-se na premissa que atores são entidades ativas, e que em documentos técnicos podem ser referidos por profissões ou papéis que desempenham. Os padrões foram definidos tendo por base um conjunto de terminações da língua portuguesa, terminações essas que referenciam funções ou profissões. Após um conjunto de experimentos,

identificamos um total de 82 padrões que remetem a substantivos (componentes de sintagmas nominais) com terminações específicas. Para a identificação de recursos, utilizamos um conjunto menor de padrões, apenas 9. O conjunto de padrões para recursos é menor, pois tais padrões são mais gerais que aqueles utilizados na identificação de atores.

A **deteção de erros, discrepâncias e omissões em requisitos** (seção 3.3) é realizada através da identificação de requisitos candidatos à duplicidade e de identificação de omissões em requisitos não funcionais.

Utilizamos medidas de similaridade entre documentos para identificar pares de requisitos candidatos à duplicidade. Utilizamos um índice único para a identificação da similaridade, dado pela média aritmética dos índices do coseno, Jaccard e Dice. Estas métricas são usuais na área de processamento de linguagem natural para identificar documentos de conteúdo similar. Se o índice obtido for maior que o limiar definido em 0,90, o par de requisitos é indicado como candidato à duplicidade e deverá ser analisado pelos participantes do processo de verificação. O limiar foi definido empiricamente após vários experimentos com documentos de requisitos.

Para a identificação de omissões restringimos nossa área de atuação aos requisitos não funcionais. Utilizando técnicas de análise de conteúdo, criamos um dicionário de categorias as quais correspondem diretamente a requisitos não funcionais extraídos de catálogos públicos. A cada uma dessas categorias é associado um conjunto de termos usualmente utilizados na organização para fazer referências àquele requisito não funcional. O documento de requisitos é então analisado em relação ao dicionário de requisitos não funcionais e termos associados, sendo geradas tabelas de frequência de termos. Essas tabelas são consolidadas nas categorias do dicionário de RNF's, possibilitando ao engenheiro de requisitos uma análise de omissões em relação aos RNF's que seriam desejáveis ou essenciais para a aplicação em pauta.

Para a análise de omissões também pode ser utilizada a comparação de dois documentos em relação ao dicionário de RNF's. Se a organização dispõe de um documento de requisitos que possa ser considerado padrão num determinado domínio de aplicação, então este documento pode ser utilizado como base para comparação da presença ou ausência de RNF's no documento sendo avaliado.

6.1.2. Estratégias com uso de agentes de software

Utilizamos agentes de software como agentes pessoais de representantes de participantes do processo de desenvolvimento, como observadores do ambiente de execução para detectar eventos relevantes, como comunicadores e como encapsuladores de serviços via ferramentas criadas em diferentes linguagens de programação.

Agentes pessoais são especializados e representam participantes com um perfil específico de atuação. Tais agentes são autônomos e, ao detectar ações relevantes por parte do humano que representam, executam ações de forma autônoma em nome deste humano. Esse é o caso do agente *Manager*, que representa o gerente do projeto e, uma vez definida a data para a verificação dos requisitos, modifica o estado do documento de requisitos, executa os procedimentos de pré-verificação e notifica os humanos envolvidos.

Como observador do ambiente de execução, o agente *Observador* identifica quando ocorrem eventos de modificação na base de requisitos e propaga essa informação para os demais agentes através do *blackboard*. Os agentes interessados em tal evento podem, então, tomar decisões baseados nessa informação. A propagação automática da ocorrência de eventos como este é fundamental para ambientes distribuídos de desenvolvimento, para garantir que toda a equipe seja informada de modificações em requisitos e desenvolva seu trabalho a partir de uma mesma versão.

No ambiente de execução, agentes e humanos colaboram e trocam informações. O agente *Comunicador* é responsável pelo envio de mensagens aos humanos, quando solicitado pelos demais agentes. Já a comunicação entre agentes se faz através do ambiente de execução, via *blackboard*.

Agentes também oferecem serviços a outros agentes ou humanos através de *Web services*, e no ambiente de execução o agente *Léxico* é um representante típico deste papel. Os serviços oferecidos por este agente incorporam ferramentas de processamento da linguagem natural, parte das quais disponibilizadas livremente pela comunidade da área. Quando solicitado por outros o agente *Léxico* executa o serviço solicitado. O uso de *Web services* possibilitou a incorporação rápida de várias ferramentas de processamento da linguagem natural

que, em outras situações, necessitariam ser desenvolvidas ou obrigariam a um trabalho de adaptação.

Na plataforma de agentes utilizada nesta tese encontramos as propriedades de autonomia, reatividade e colaboração. A plataforma utilizada oferece flexibilidade para gerenciamento de agentes, possibilitando dinamicamente a remoção ou inserção de agentes no ambiente de execução. Esta característica possibilita que agentes pessoais representem fielmente o conjunto de participantes no processo de desenvolvimento - os interessados. Agentes possuem continuidade temporal, monitorando e notificando de forma imediata a ocorrência de eventos relevantes aos interessados.

6.2.

Comparação com trabalhos relacionados

Não encontramos, na literatura pesquisada, uma proposta para apoio às atividades de verificação e validação de requisitos que utilizasse de forma tão intensiva quanto a estratégia proposta neste trabalho técnicas de processamento da linguagem natural para apoio às atividades de verificação e validação de requisitos. O uso de agentes de software como apoio às atividades do próprio processo de desenvolvimento já foi abordado em diversos trabalhos. A comparação com trabalhos relacionados será feita de forma pontual, efetuando comparações em diferentes aspectos do nosso trabalho.

Agrupamentos de requisitos: na extensa pesquisa que fizemos encontramos poucos trabalhos abordando este tópico. A proposta descrita em [Hsia92] define incrementos apoiada na identificação de tipos de dados abstratos e numa técnica para agrupar requisitos que modifiquem ou utilizem um mesmo tipo abstrato de dados. O objetivo da abordagem é identificar requisitos que, agrupados, sejam alocados a um mesmo incremento; a técnica é proposta para sistemas com ênfase em manter a integridade dos dados, por exemplo um sistema para automação de bibliotecas. Esta técnica para agrupar requisitos exige que no momento do registro dos requisitos, sejam definidos os tipos abstratos de dados que irão ser utilizados, trazendo para a fase de requisitos uma definição que normalmente seria realizada numa etapa posterior do processo de desenvolvimento.

O trabalho pioneiro de Palmer e Liang [Palmer92] sobre indexação e agrupamento de requisitos defende a idéia que o agrupamento de requisitos agiliza a detecção de conflitos, incompletude, inconsistências e imprecisão internamente aos agrupamentos, e externamente entre os agrupamentos. Palmer propõe a classificação de requisitos em dois níveis de hierarquia: os requisitos são inicialmente agrupados num nível mais alto tendo por base os verbos indicativos de funcionalidades e um thesaurus de verbos construído previamente. Os grupos assim considerados são então subdivididos considerando-se uma medida de similaridade entre documentos, o coseno.

O trabalho mais recente descrito em [Chen05] apresenta uma abordagem baseada no agrupamento de requisitos para construção de modelos de *features*. Relações de dependência entre dois requisitos são identificadas se ambos acessam um mesmo recurso; são identificados 5 tipos básicos de relações de dependência, e a cada tipo é atribuído um peso. É construído um grafo no qual requisitos representam os nodos e relações de dependência são representadas por arcos. Os agrupamentos de requisitos são derivados desses grafos e gerados considerando os pesos atribuídos a cada relação de dependência. Nesta abordagem, requisitos podem estar presentes em mais de um agrupamento.

Nossa proposta para agrupamento de requisitos tem os mesmos objetivos da idéia de Palmer e Liang; também utilizamos uma taxonomia estruturada em dois níveis para o agrupamento dos requisitos. A taxonomia utilizada, no nosso caso, é construída a partir do próprio documento de requisitos. Os temas do primeiro nível são identificados através de medidas que tem por base a frequência dos termos no documento, e os do segundo nível são obtidos através do uso de colocações, ou palavras que co-ocorrem no texto com uma frequência maior que o esperado pelo acaso.

Geração de visões de requisitos: a literatura aponta poucas propostas com aplicação de técnicas de visualização a requisitos.

Na área de aspectos alguns trabalhos guardam semelhanças com o nosso: em [Baniassad04] é apresentada uma técnica para identificação e análise de aspectos em documentos de requisitos. Visões são construídas expondo comportamentos do sistema, através de análise léxica. A estratégia proposta por [Silva06] para integração de características transversais utiliza visualização para mostrar relacionamentos transversais entre requisitos.

Recentemente a comunidade de Engenharia de Requisitos mostrou interesse na aplicação de técnicas de visualização para requisitos: a 14th IEEE International Requirements Engineering Conference (2006) colocou, como um dos co-eventos, um workshop destinado especificamente a tratar do tema visualização em requisitos - o International Workshop on Requirements Engineering Visualization. Isto indica o grau de interesse atual da comunidade em trabalhos orientados à geração de visualizações em requisitos.

Um dos trabalhos apresentados [Ozkaya06] nesse evento trata do tema visualização de relacionamentos entre requisitos de forma geral. O autor registra que as ferramentas disponíveis para a fase de requisitos fazem pouco ou nenhum uso da visualização, e faz uma rápida avaliação de sua experiência no desenvolvimento de uma ferramenta para registro e visualização da rastreabilidade de requisitos. Infelizmente não são apresentados detalhes ou mesmo imagens geradas no decorrer do seu trabalho.

Criação do léxico da aplicação: parte da nossa estratégia para construção do léxico da aplicação utiliza sintagmas nominais, com vários trabalhos relacionados na literatura. Já para a estratégia que utilizamos para extrair termos próprios do domínio da aplicação, mesmo sendo uma estratégia simples baseada na comparação com dicionários da língua, não encontramos trabalhos relacionados.

No contexto de processamento de linguagem natural e recuperação de informação, o uso de sintagmas nominais para recuperação de informações é encontrado em muitos trabalhos. [Parreiras03] propõe seu uso em indexação de textos científicos e [Pérez03] os utiliza para a obtenção de conceitos que irão compor mapas conceituais. No contexto do processo de requisitos, trabalhos orientados à exploração de aspectos da linguagem natural ainda não são muito frequentes.

Em [Harmain00] é descrito o CM-Builder, uma ferramenta *case* que avalia documentos de requisitos escritos em linguagem natural (língua inglesa) buscando relações semânticas nas sentenças de requisitos, visando à construção de um modelo inicial de classes em UML através da identificação de sintagmas nominais. O resultado inclui classes, atributos e relacionamentos, agrupados em modelos de classes e armazenados em arquivos cujo formato é adequado à manipulação posterior por ferramentas normalmente utilizadas nas etapas de

projeto do sistema. Os autores enfatizam que os resultados devem ser vistos como apoio ao trabalho do analista ou engenheiro de software, devendo ser refinados para efetivamente contribuir para um modelo inicial de classes.

A abordagem proposta por Boyd et al [Boyd05] utiliza um subconjunto controlado da linguagem natural para o registro de requisitos, com o objetivo de reduzir a ambigüidade. Esse trabalho investiga a expressividade sintática e semântica de um sub-conjunto da língua inglesa para uso em documentos de requisitos, focando especificamente nos verbos (ações). A identificação das entidades (sujeitos das ações) é realizada via sintagmas nominais.

Nosso trabalho aproxima-se do trabalho de [Harmain00], já referido, mas nossos objetivos são diferentes: visamos apoiar o processo de requisitos, enquanto eles buscam apoiar o processo de projeto do sistema. Utilizamos como fontes de informação não só documentos de requisitos, mas também outros documentos que sejam manipulados ou gerados no processo de requisitos – basta que tais documentos utilizem a linguagem natural.

Enquanto a abordagem descrita em [Boyd05] utiliza um sub-conjunto restrito da língua natural, nossa abordagem não restringe o uso da língua portuguesa nos documentos que manipula. Nosso objetivo é identificar atores/sujeitos e recursos/objetos relevantes, de forma a apoiar a construção ou atualização do léxico da aplicação. O processo criado também é útil na construção de um léxico para o domínio da organização, pois em organizações utilizando desenvolvimento distribuído de software dificuldades derivadas de diferentes capacidades lingüísticas, diferenças culturais e *delays* de comunicação tornam mais presente a necessidade de um léxico abrangente.

Nossa abordagem utiliza também uma *stop list* constituída de termos não relevantes para o domínio considerado, permitindo ajustes nos termos a serem extraídos e gerando resultados mais precisos na avaliação futura de documentos de um mesmo domínio. A incorporação de um lematizador também possibilitaria a extração de sinônimos. Assim como enfatizado por Harmain e Gaizauskas [Hairmain00], consideramos que os resultados obtidos não prescindem do processo de revisão e avaliação por especialistas do domínio; os sujeitos e objetos extraídos através do processo proposto constituem um subsídio importante para a construção ou atualização do léxico da aplicação.

Identificação de duplicidade em requisitos: a utilização de medidas de

similaridade para identificar duplicidade em requisitos é tema de ao menos dois trabalhos na área de requisitos: [Park00] e [Dag01] relatam resultados de trabalhos com uso de diferentes medidas de similaridade. Em [Park00] é utilizada a medida do cosseno do ângulo formado pelos documentos, sendo que a matriz termo-documento é gerada utilizando-se a frequência não dos termos, mas de expressões com afinidade léxica. Tais expressões são obtidas usando uma combinação de um parser para identificar relações entre palavras e uma técnica denominada de *sliding window*, que varre o texto considerando "janelas" de até cinco palavras para análise.

Os índices que utilizamos já foram objeto de trabalho investigativo da similaridade entre requisitos [Dag01]; nesse estudo foi avaliado um conjunto de requisitos escritos em linguagem natural (língua inglesa). Tais requisitos foram representados por matrizes termo-documento, sendo desconsiderados termos relacionados numa *stoplist* composta por artigos, pronomes, advérbios e outros. O estudo conclui que o uso desses indexadores, em particular Dice e cosseno, resulta em alto grau de acerto na identificação de requisitos duplicados e baixa taxa de falsos positivos (requisitos indicados como duplicados, mas que na realidade não são).

Identificação de omissões em RNF's: utilizamos uma estratégia baseada na Análise de Conteúdo para identificar omissões em RNF's. Análise de conteúdo também foi utilizada em [Fantechi05] para identificar inconsistências em documentos de requisitos. Nessa abordagem o documento de requisitos é processado por um *parser* gramatical, e são extraídas triplas sujeito - ação - objeto (SAO). Esses elementos são inseridos num dicionário de categorias e a cada elemento é atribuído um peso, de acordo com a classe gramatical. A criação do dicionário segue a estrutura do documento: se requisitos funcionais são colocados separadamente de requisitos não funcionais, o dicionário estará estruturado em duas partições. O documento de requisitos é analisado, e a cada requisito é atribuído um valor que computa a presença das categorias de cada partição do dicionário no requisito. Esses valores são depois analisados, e são verificados os requisitos que apontem pesos relevantes para a partição indevida; por exemplo, um requisito funcional que apresente peso relevante para termos que constam no dicionário da partição de requisitos não funcionais.

Um problema apresentado por essa abordagem está na construção do

dicionário: se requisitos forem colocados numa seção incorreta, os componentes SAO que irão gerar as categorias da partição incluirão triplas SAO na partição incorreta, e a análise resultará inconsistente. Nossa abordagem trabalha com um dicionário construído previamente, abordando apenas requisitos não funcionais, e considerando expressões utilizadas na organização para registro dos RNF's.

6.3. Limitações e trabalhos futuros

Nosso trabalho apresenta limitações que relacionamos a seguir, juntamente com trabalhos futuros sugeridos para ampliar e consolidar as estratégias propostas.

Verificação de Requisitos: nossa estratégia incorpora de forma automática a busca por duplicidade em requisitos. Também apóia a identificação de omissões em requisitos não funcionais. Futuramente podemos investigar com uso de técnicas de processamento da linguagem natural a detecção automática de conflitos entre requisitos. Em relação aos resultados da verificação, uma possível ampliação envolveria a criação de um formulário para uso dos humanos na identificação de erros, discrepâncias e omissões em requisitos. Este formulário deveria ser pensado de forma a viabilizar de forma automática a consolidação, por um agente de software, dos relatórios gerados pelo grupo de verificadores. As tabelas para a análise de omissões de RNF's poderiam ser ampliadas de modo que o engenheiro de requisitos pudesse indicar as comparações que, executadas de forma automática, lhe subsidiassem de forma mais intensa a análise de omissões e de discrepâncias.

Controle das modificações em requisitos: nossa estratégia destaca especificamente um agente para controlar a ocorrência de modificações em requisitos, e propagar essa informação aos demais agentes da plataforma. Este aspecto seria mais bem atendido com a incorporação de um controlador de versões, de forma que cada um dos sites distribuídos pudesse manter um repositório próprio dos artefatos e documentos sendo manipulados. Isto exigiria um esforço maior para manter a consistência entre repositórios, o que poderia ser obtido através de um agente local ao site que ou executaria de forma autônoma operações de *check-in* após um determinado prazo ou notificaria o humano responsável pela modificação local da inconsistência detectada.

Rastreabilidade entre requisitos funcionais e requisitos não funcionais:

uma vez definidas as expressões utilizadas no registro de RNF's, é possível criar automaticamente a matriz de rastreabilidade entre requisitos funcionais e não funcionais. Este é um trabalho futuro facilmente implementável.

Léxico da aplicação: estender a estratégia de construção ou atualização do léxico para símbolos do tipo verbo e estado, não abarcados nesta proposta, e realizar experimentos visando verificar a adequação da estratégia criada para construção de um léxico para a organização, utilizando diferentes tipos de documentos.

Validação de requisitos: o apoio oferecido às atividades de validação é limitado à geração de visões parciais de requisitos, permitindo ao participante do processo executar seu trabalho avaliando conjuntos menores de requisitos. As visões também podem ser geradas a partir da solicitação do validador, a partir de um tema de seu interesse. Isto poderia ser expandido para tratar também outros artefatos utilizados na validação, como *mock ups* e protótipos criados para esta fase.