

Extração de requisitos apoiada por técnicas de Inteligência Artificial embutidas em ferramentas de apoio

Cristiane Aparecida Lana¹, Lucas Gonçalves Cunha¹, José Luis Braga¹ e Antônio de Pádua Albuquerque Oliveira²,

¹Departamento de Informática, UFV, PPGCC – Programa de Pós-Graduação em Ciência da Computação, Viçosa, MG
cristiane.lana@ufv.br, zeluis@dpi.ufv.br, lucas.cunha@ufv.br

²Departamento de Informática e Ciências da Computação, UERJ, Rio de Janeiro, RJ
padua@ime.uerj.br

Abstract. A linguagem peculiar e a pluralidade de visões distintas exigem conhecimento e experiência do engenheiro de requisitos para o sucesso da atividade de extração de requisitos em domínios específicos. Neste trabalho, são apresentadas as ferramentas i*Get e TEKBS, que utilizam técnicas de inteligência artificial para auxiliarem o engenheiro de requisitos na extração de metas concretas e flexíveis, no contexto de iStar. A i*Get utiliza as “ações concretas” definidas no LAL enquanto a TEKBS as “ações flexíveis” combinadas com os sinônimos dos termos que representam essas ações, obtidos a partir do Wordnet. Os resultados são transformados em base de fatos da ferramenta CLIPS, e então processados usando uma base de regras de análise disparadas pela máquina de inferência do CLIPS. Resultados preliminares mostram que um número maior de requisitos são extraídos pelo engenheiro, melhorando o conhecimento sobre o domínio.

Keywords: Knowledge-based systems; iStar; softgoals and hardgoal; Léxico

1 Introdução

A tecnologia exerce um papel crescente na economia e as empresas têm crescido mais fortemente dependentes da entrega bem sucedida de Sistemas de Informação – SI [1]. Entretanto, no processo de desenvolvimento de SI falhas no modelo de requisitos e consequentemente nos projetos são comuns. Estudos como [1] ressaltam que modelos de requisitos com requisitos incorretos, incompletos ou conflitantes são considerados as grandes causas de insucesso dos projetos de software. De forma geral, os insucessos estão associados à falta de compreensão do negócio pelo analista de sistema, ao mau entendimento da finalidade do sistema, e às falhas de comunicação entre os analistas envolvidos [2].

Para alguns autores como [3] os fatores de insucesso estão mais intimamente associados aos requisitos de atributos de qualidade (Metas Flexíveis¹), tais como consistência, completude, confiabilidade e facilidade de compreensão, que a grande maioria das vezes são tratados de forma secundária e informal do ponto de vista de elicitação, sendo difícil sua consideração ao longo do ciclo de vida do software.

Na busca de minimizar os problemas enfrentados no processo de elicitação e consequentemente criar melhores modelos de requisitos, surgem vários grupos de pesquisa no exterior e no Brasil (*Tropos Group*, *SERG*, *REQENG*, *Requirements Engineering Research Group*, *PUC-RIO (ER)*, *UFPE (LER)*, entre outros) que investigam formas de melhorar o processo de elicitação de requisitos. Desses esforços surgiu a Engenharia de Requisitos Orientada a Metas² – GORE [4]. A abordagem por metas, por ser baseada em intencionalidade, proporciona: (i) a estabilidade de metas em comparação com os requisitos que as implementam; (ii) a habilidade de considerar soluções alternativas; (iii) a possibilidade de verificação da plenitude dos requisitos; (iv) a rastreabilidade a partir do contexto organizacional para os requisitos; e (v) um nível de visão mais alto dos requisitos se comprado com especificações tradicionais [5].

Apesar de os modelos intencionais³ conseguirem uma maior representatividade quando comparados a outros modelos, as dificuldades de extrair informações do domínio e construir modelos ainda são as mesmas, por serem obtidas com o uso de linguagem natural.

Apresenta-se neste artigo uma abordagem de melhoria do processo de elicitação de requisitos nas fases iniciais do ciclo de vida do software, utilizando técnicas de Inteligência Artificial e sinonímia, tendo o Método ERi*c (Engenharia de Requisitos Intencional) como modelo para Engenharia de Requisitos. Com essa abordagem são projetadas e implementadas duas ferramentas e uma técnica. A primeira ferramenta *i*Goal Elicitation Tool* - *i*Get* tem por finalidade acompanhar o Engenheiro de Requisitos – ER na extração e modelagem de metas, apoiando o ER nas fases iniciais de criação de modelos de sistemas baseados em agentes. A segunda ferramenta e a técnica chamada de *Knowledge-based Softgoal Elicitation Technique* – TEKBS que busca auxiliar na elicitação de Metas Flexíveis - MFs⁴ por meio de sinonímia na elaboração o modelo *i*Star*. A TEKBS pode ser complementar a *i*Get* utilizando como sua entrada o artefato produzido pela *i*Get*, mas pode também ser utilizada isoladamente quando sua entrada for qualquer outro tipo de documento, expresso em XML - *eXtensible Markup Language*⁵, por exemplo. Resultados parciais de aplicação das ferramentas em um problema conhecido⁶, mostram uma melhoria na identificação de requisitos, o que é o objetivo da pesquisa.

¹ O termo Meta Flexível é utilizado em português como sinônimo de softgoal. Uma descrição mais detalhada de softgoal pode ser encontrada em Chung (2000); Oliveira (2008) e Mylopoulos et. al. (1992).

² Metas são objetivos de alto nível do negócio, organização ou sistema. Elas expressam a razão do sistema proposto e guia decisões em vários níveis dentro do projeto.

³ Tropos, NFR Framework, KAOS, ERi*c, Framework i*, são exemplos de algumas abordagens que enquadram na proposto da GORE.

⁴ Neste trabalho usaremos a sigla MFs para Metas Flexíveis e MF para Meta Flexível.

⁵ <http://www.w3schools.com/xml/>

⁶ *Expert Committee System*

Este artigo, além da introdução, possui mais quatro seções. Na segunda seção são apresentadas as tecnologias e conceitos necessários para o entendimento do estudo. Na terceira seção são apresentados os resultados e as discussões. Na quarta seção exibe os trabalhos correlatos. E por fim, a quinta seção descreve as conclusões deste trabalho.

2 Tecnologia e conceitos de suporte

Nesta seção são apresentadas as tecnologias e os conceitos de suporte necessários para o entendimento deste artigo.

2.1. Framework i*

O Framework i* foi introduzido por [6], é um modelo para modelagem organizacional que é mais expressivo para descrição de requisitos que envolvam muitos atores e que possa ser empregada nas primeiras fases do ciclo de vida do desenvolvimento do software [7]. Atores são considerados intencionais em consequência de terem desejos e necessidades, e são considerados estratégicos, por estarem preocupados com as oportunidades e vulnerabilidades dos problemas [8].

O Framework i* possui dois níveis de abstração: O Modelo SD (*Strategic Dependency* - Dependência Estratégica) e o Modelo SR (*Strategic Rationale* – Razão Estratégica). No Modelo SD um ator depende de outro para satisfazer uma meta, para executar uma tarefa, fornecer um recurso ou satisfazer uma Meta Flexível - MF [6]. Meta Flexível está relacionada com atributos de qualidade [7] e as metas, tarefas e recursos estão relacionados com as funcionalidades do sistema [6]. O ator que depende de outro é chamado de *Depender* enquanto o que recebe a requisição é chamado de *Dependee* e o objeto de dependência é chamado de *Dependum*.

O Modelo SR descreve os interesses dos *stakeholders* e suas motivações e como eles podem ser tratados por diferentes configurações de sistemas e ambientes [6]. Ele fornece um nível detalhado para modelar relações intencionais internas dos atores. Elementos intencionais, metas, tarefas, recursos e MFs, aparecem em Modelos SR não somente como dependências externas, mas também como elementos internos organizados em sua maioria em estruturas hierárquicas com relacionamentos meios-fim, decomposição de tarefas e relacionamento de contribuição⁷ [6].

2.2. Léxico Ampliado de Linguagem - LAL

O LAL⁸ é um modelo de representação de termos usados na linguagem da aplicação [9], cujo foco é capturar o vocabulário do contexto organizacional. A geração do

⁷ Relação meios-fim, que estão relacionadas ao alcance das metas, a decomposição de tarefa, que dá detalhes a respeito de como as tarefas serão desenvolvidas e realizadas, além de representar utilização e disponibilização de recursos e relacionamentos de contribuição que fornece a colaboração entre as metas flexíveis, que pode ser negativa ou positiva.

⁸ Para estruturar o domínio em linguagem natural no LAL é utilizada a ferramenta Cenário e Léxicos – C&L, uma ferramenta colaborativa desenvolvida pela PUC-RIO.

léxico ocorre por meio da aplicação de técnicas de elicitación como pode ser visto em [10]. O LAL é composto por símbolos⁹ que são identificados por nomes, representados por duas descrições, uma chamada de noção e a outra de impacto. A noção explica o significado literal do símbolo, e o impacto descreve os efeitos do uso ou ocorrência do símbolo no domínio sob estudo.

A construção do LAL é baseada nos princípios de circularidade e vocabulário mínimo. A circularidade descreve a ideia de possuir termos em função dos já existentes. O vocabulário mínimo reduz o uso de palavras fora do léxico, isto é, quanto menos termos fora do contexto da aplicação forem usados nas descrições melhor. No LAL, os símbolos podem ser classificados gramaticalmente pelas seguintes heurísticas: se o símbolo indica prática de uma ação, ele é classificado como sujeito; se sofre a ação, é classificado como objeto; se é uma situação em um dado momento, é classificado como estado; se representa uma ação, ele é classificado como verbo[9].

2.3. Requisitos não funcionais e Metas Flexíveis

Diversas são as pesquisas que consideram MFs como sinônimo de RNFs [11] [12]. Porém, o presente artigo utiliza MFs como considerada em Mylopoulos, Chung e Nixon [13], um conceito mais abstrato para representar a intencionalidade dos atores com relação à qualidade. Assim, as MFs são vistas como um conceito primitivo para modelagem de RNFs. Por isso, Chung [14] e Oliveira [9] definem o que vem a ser MFs e como elas se diferem dos RNFs.

Para eles, MF é uma espécie de meta que é razoavelmente aceita para denotar a falta de precisão na percepção de satisfação de RNFs. Uma MF representam necessidades que são parte da intencionalidade do ator e os RNFs são características de qualidade implementadas por funções de software.

De acordo com Rawlins e Chung [15], uma análise baseada em MFs pode ser um complemento útil para abordagens baseadas nos requisitos funcionais. Sendo assim, a não elicitación das MFs ou sua elicitación tardia pode ser tão crítica para o sucesso de um sistema quanto a não elicitación de um requisito funcional.

Por ser diferente de um RNF, a representação da MFs é expressa usando a notação Tipo[tópico], onde o tipo é um atributo de qualidade e o tópico determina em que característica do sistema se espera aquela qualidade [14].

3. Método Eri*c alterado por i*GET e TEKBS

Este trabalho é uma extensão do trabalho de Oliveira [9]. O Método Eri*c foi criado com a finalidade de melhorar a qualidade dos requisitos dos projetos de software. O Eri*c é um processo manual que contempla todas as fases de elicitación, deve ser

⁹ Os símbolos de uma linguagem são termos específicos de um domínio, podem ser gírias ou termos técnicos, que normalmente não são de conhecimento comum. Por exemplo, no *Expert Committee* os termos *autor de trabalho*, *membro de comitê de programa*, *chair* são específicos deste domínio.

executado pelo ER e ficará dependente da sua expertise para extrair das descrições dos *stakeholders* as metas e MFs explícitas e implícitas.

Para tentar minimizar os possíveis erros que podem surgir do processo manual, duas ferramentas de apoio foram desenvolvidas. A primeira é a *i*Get* que automatiza parte das tarefas contempladas na primeira etapa “Elicitar as Metas dos Atores” do Método ERi*c, ver Fig 1. A segunda é a TEKBS que busca melhorar, a partir dos dados elicitados pela *i*Get*, a extração semiautomática de MFs utilizando sistemas com base de conhecimento e sinonímia, ver Fig 1. Na cor verde é mostrado o *workflow* da etapa de elicitação do Método ERi*c [9]. Na cor azul aparecem as atividades introduzidas pela *i*Get*, e na cor rosa as introduzidas pela TEKBS. Ambas as ferramentas foram desenvolvidas na linguagem CLIPS [16].

A fim de demonstrar o funcionamento e validar a *i*Get* e a TEKBS, foi utilizada uma versão simplificada do problema *Expert Committee - EC* [17], conhecido e já utilizado em outros trabalhos na área. A escolha do EC está relacionada à existência de estudos, como o de [1] que exploram o seu contexto, incluindo a elicitação de metas e MFs realizada de forma manual por Oliveira [9].

O EC pode ser descrito resumidamente como: “O EC é um sistema multi-agente aberto, utilizado para apoiar o gerenciamento de submissões e ao processo de revisão realizado em conferências ou workshops. Por ser um sistema baseado em agentes de software, o EC pode auxiliar os pesquisadores nas atividades de envio de trabalhos e processo de revisão que são as que mais consomem tempo. Os agentes de EC são assessores de software que assumem diferentes papéis, mas os principais são: autor do trabalho, revisor, membro do comitê do programa e chair. Diversas são as atividades que precisam ser realizadas e elas são executadas de acordo com as prioridades, contudo, o processo de realização das revisões é feito mediante negociação entre o chair, os membros do comitê do programa e os revisores.” [18].

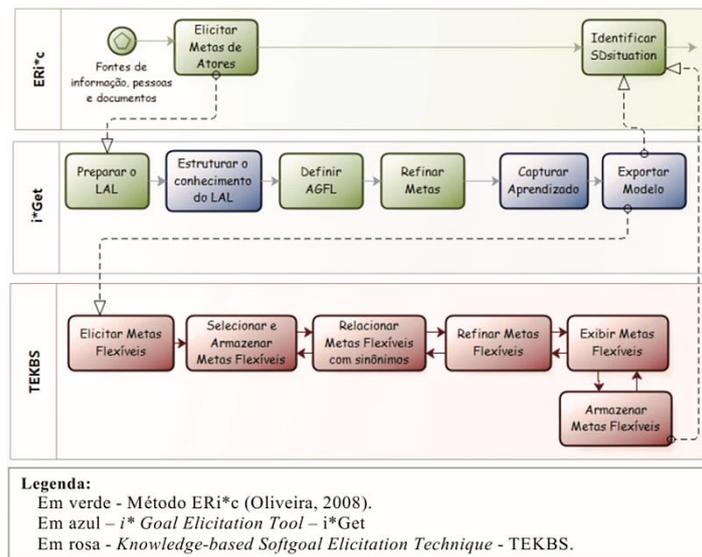


Fig. 1. Visão Geral do *i*Get* e da TEKBS inseridas no Método ERi*c.

3.1. i*GET – i* Goal Elicitation Tool

A i*Get foi proposta por [19] e é uma ferramenta que auxilia o ER na extração de requisitos nas etapas iniciais de criação de modelos de requisitos que utilizam a notação de metas para modelagem de software. Foi construída para estender e semiautomatizar¹⁰ parte das atividades do ERI*c [9], respeitando as definições contidas no ERI*c. É composta por quatro módulos (*Extracting, Refining, Learning e Exporting*) que cobrem as seis fases¹¹ da ferramenta, ver Fig. 1. O *Module Extracting* é o primeiro módulo da i*Get onde o conhecimento do LAL é formatado e inserido na base para que seja possível realizar a extração das metas. Nesta primeira atividade são listados todos os atores presentes nos símbolos (*author, chair, committee member, general coordinator, researcher e reviewer*), além de identificar as Metas Concretas - MC¹² que são elicitadas por ordem de inserção na base e MFs que são extraídas após a elicitacão das MCs. Por serem originadas de símbolos¹³ de tipos diferentes as MCs apresentam formatos distintos como ilustra a Tabela 1.

Tabela 1. Parte das Metas Concretas elicitadas antes do refinamento

Identificados	Descrição da meta
gen97	To conflict be decided by committee member
gen96	To article be published by chair
gen95	To article be accepted
gen90	Because chair want(s) article be reviewed by reviewer

Após a elicitacão das MCs é preciso extrair as MFs do contexto do EC. De acordo com as heurística do ERI*c [9] *toda MF possui uma MC associada*, a Fig. 2 mostra algumas MFs que são associadas a algumas MCs da Tabela 1 e que serão utilizadas posteriormente pela TEKBS.

O *Module Refining* tem a função de equalizar as MCs, agrupando as metas (concretas e flexíveis) por ator, excluindo as redundâncias e unificando a redacão. Usando o problema EC já disponível no LAL, o processo inicia-se com 33 MCs e 12 MFs elicitadas pelo *Module Extracting* e após o processo de refinamento, 13 MCs e 11MFs permaneceram na base. Com o refinamento das MCs, as MFs relacionadas às MCs duplicadas, que serão excluídas, são transferidas para MCs equivalentes mantendo a coerência dos dados e melhorando a qualidade das metas (concretas e flexíveis) e consequentemente dos modelos finais.

¹⁰ A i*Get é semiautomatizada, depende de interaçãõ externa com o ER

¹¹ O módulo *Extracting* cobre as atividades: (i) preparar o LAL; (ii) Estruturar o conhecimento do LAL; e (iii) Definir a Agentes vindo do Léxico- AGFL onde as metas são elicitadas.

¹² Uma Meta Concreta está relacionada a Funcionalidade do sistema.

¹³ Metas Concretas podem ser provenientes de símbolos do tipo sujeito, tipo verbo, tipo objeto e tipo estado (Oliveira, 2008).

```

Softgoal : Type(quality) Topic[review]
Associated Goal:conflict be decided by committee member

Softgoal : Type(punctuality) Topic[publication]
Associated Goal:article be published by chair

Softgoal : Type(correct) Topic[evaluation]
Associated Goal:article be accepted by null

```

Fig. 2. Metas Flexíveis Extraídas do EC.

Um exemplo de refinamento pode ser visto na Tabela 2 que foi obtida a partir da Tabela 1 de MCs do EC. Considerando a Tabela 1 foram excluídas as MCs *gen95* e *gen96*, por serem metas duplicadas na base de dados, e as demais tiveram sua redação unificada, alterando a posição do sujeito na frase que inicialmente se encontrava no final para o início, tornando-o reponsável pela meta.

Tabela 2. Parte das Metas Concretas elicitados após o refinamento

Identificados	Descrição da meta
gen97	chair want(s) article be reviewed by reviewer
gen90	committee member want(s) conflict be decided

O *Module Learning* tem por finalidade classificar as metas presentes nas descrições dos símbolos, fornecendo ao ER um mecanismo de indicação da classificação de possíveis metas baseando-se nos históricos das metas já existentes na base da *i*Get*. Para que a classificação fosse possível, inicialmente foi utilizada a técnica de Normalização Morfológica (*Stemming*) [20] e o algoritmo de normalização de Porter [21]. Com a técnica e o algoritmo foi possível identificar e criar os radicais dos verbos, que podem ser classificados segundo o seu Fator de Certeza - FC [22], que é baseado no histórico de ocorrências das ações em elicitções anteriores, seus valores começam a aparecer depois de algumas execuções da ferramenta. Para o exemplo, as metas elicitadas são inseridas na base com FC neutro, sendo modificado na medida em que novas elicitções ocorrerem.

Ao final da execução do módulo de aprendizado, as metas encontram-se na base de produção e para que a informação não seja perdida, elas são armazenadas para uso posterior, criando o histórico de uso.

Por fim, temos o *Module Exporting*. É ele o responsável por disponibilizar as metas para o ER possibilitando a agregação dos demais elementos do Modelo SD e SR. Para a exportação foi utilizada a linguagem *iStarML* [19] [23], onde as metas existentes na memória da *i*Get* foram formatadas seguindo a sintaxe da linguagem e disponibilizadas para o ER. Para facilitar o entendimento do ER foi elaborada uma representação gráfica das metas em *iStarML* com a utilização do *jUCMNav* [24], que auxilia o ER na criação de modelos orientados a metas.

3.2. TEKBS - Knowledge-based Softgoal Elicitation Technique

A TEKBS foi desenvolvida por [25] e é considerada não somente uma ferramenta que auxilia o ER na extração de MFs utilizando sinônimos, mas também uma técnica de elicitação de MFs uma vez que é possível utilizar outros tipos de entradas além dos originados na i*Get. Sua atuação é nas primeiras fases do processo de elicitação de requisitos que utilizam a notação de metas para modelagem de software.

A TEKBS enquanto ferramenta foi desenvolvida para estender o ERI*c [9]. Ela é composta por oito módulos (*Module Extracting*, *softgoalrootFile*, *Synonymous*, *Inserting*, *Display Softgoals*, *Single Deleting*, *Duplicate Softgoal* e *BaseFile*) que cobrem as seis atividades que a compõem, ver Fig. 1.

A primeira atividade da TEKBS, “*Elicitar Metas Flexíveis*” é responsável por capturar as MFs do contexto, que são provenientes das ações flexíveis identificadas na base recebida, além de ser a atividade onde ocorre a estruturação das informações do domínio recebido de acordo com a linguagem CLIPS. A segunda atividade “*Selecionar e Armazenar Metas Flexíveis*” é constituída do “*Module Extracting*”, que escolhe somente as MFs da base de dados inicial¹⁴ e do “*Module SoftgoalrootFile*” que é responsável por armazenar as MFs extraídas. No caso do EC, foram extraídas da i*Get e armazenadas da base da TEKBS as MFs raízes¹⁵: *quality [review]*; *punctuality [publication]*; *acknowledge [committee]*; *honest [review]*; *quality [article]*; *no delay [review]*; *fair [review]*; *e correct [evaluation]*, que serão utilizadas para buscar sinônimos e descobrir novas MFs.

A próxima atividade é responsável por “*Relacionar Metas Flexíveis com Sinônimos*” e possui duas etapas: a primeira é buscar os sinônimos na base de dados do Wordnet utilizando o aplicativo JAVAS¹⁶ e a segunda é extrair os sinônimos da base de pesquisa do JAVAS para o TEKBS que é executada pelo *Module Synonymous*. A Fig. 3 mostra toda a sequência das atividades, assim como os artefatos de entrada e saída.

A etapa executada pelo JAVAS é externa a TEKBS e por isso é necessário sair da TEKBS e após a pesquisa no Wordnet carregar o arquivo retornado pelo JAVAS. Somente após o carregamento da base de pesquisa o *Module Synonymous* será acionado e os sinônimos inseridos na Base de Sinônimos – BS. No entanto, não se pode garantir que todos os sinônimos extraídos do Wordnet sejam realmente sinônimos das MFs raízes, sendo assim, foi feita a avaliação de aderência ou acoplamento¹⁷ das MFs raízes com as MFs sinônimas. Essa análise é primordial para

¹⁴ A base inicial pode ser composta de outras metas, como MCs, por isso a necessidade de selecionar somente as MFs.

¹⁵ Considera-se MFs Raízes as MFs que foram extraídas da base de dados da i*GET

¹⁶ O JAVAS é um aplicativo desenvolvido a partir da biblioteca Jsoup que fornece uma Application Programming Interface – API para manipular dados, leitura em HTML e criar arquivos como extensão do CLIPS.

¹⁷ A análise de aderência (relevância do sinônimo para o contexto) ou acoplamento (o nível de interdependência entre o Tipo e o tópico da MF e entre as MFs) é para garantir que todos os sinônimos elicitados são de fato sinônimos das MFs raízes e que fazem parte do contexto do EC. Exemplo: a MF sinônima *admit [rating]* foi elicitada, mas ela não faz parte do contexto do EC, sendo considerada “*Fora do contexto*”.

garantir que a atividade *Refinar Metas Flexíveis* seja feita de forma a manter as informações elicitadas pela TEKBS alinhadas contendo somente MFs pertencentes ao domínio, neste caso o EC. O processo de análise será descrito na subseção 3.2.1.

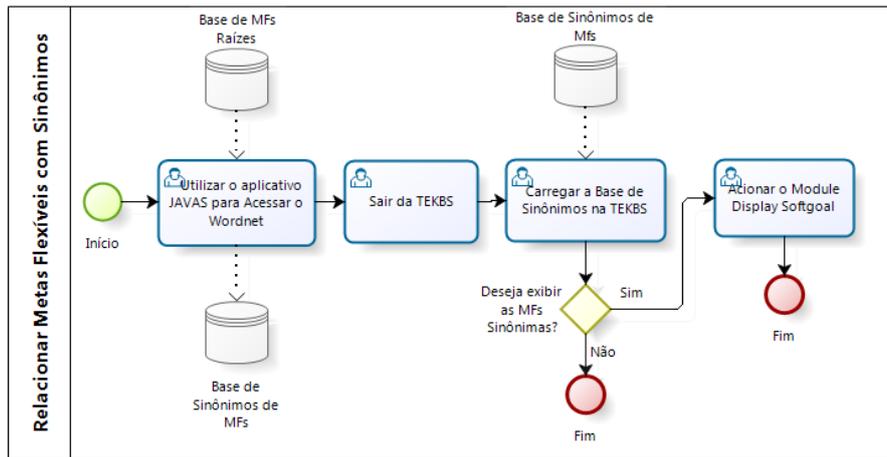


Fig. 3. Fluxo das atividades Relacionar Metas Flexíveis com sinônimos

A quarta atividade “*Refinar Metas Flexíveis*“ é executada após a análise de contexto das MFs. Por meio dela é possível: (i) excluir MFs ou sinônimos duplicados utilizando o *Module Duplicate Softgoal*; (ii) excluir MFs que não fazem parte do escopo do problema por meio do *Module Single Deleting*; e (iii) inserir MFs que fazem parte do escopo e que não estavam presentes inicialmente com o *Module Inserting*. A exclusão e a inserção podem ser justificadas pelos processos de mudança comuns em projetos de software e a decisão de inserir ou excluir é de responsabilidade do ER.

Para o EC, a primeira atividade foi excluir as MFs sinônimas que não faziam parte do escopo e excluir as duplicidades, ver Fig. 4. Após a elicitación, 1827 MFs sinônimos foram inseridas na base de dados, dessas somente 306 permaneceram após as exclusões e depois do refinamento baseado na análise de contexto a base ficou com 80 MFs pertencentes ao contexto do EC.

```

*** You chose Module PrintSoftgoal ***
Id:(gen1827) SynSoftgoal: set [reappraisal]
Id:(gen1826) SynSoftgoal: adjust [reappraisal]
Id:(gen1825) SynSoftgoal: slump [reappraisal]
Id:(gen1824) SynSoftgoal: decline [reappraisal]
↓
*** You chose Module PrintSoftgoal ***
Id:(gen306) SynSoftgoal: promptness [rating]
Id:(gen305) SynSoftgoal: recognise [rating]
Id:(gen304) SynSoftgoal: recognize [rating]
Id:(gen303) SynSoftgoal: admit [rating]

```

Fig. 4. Parte da Memória de trabalho antes e após as exclusões anteriores ao refinamento.

As atividades cinco e seis, *Armazenar Metas Flexíveis* e *Exibir Metas Flexíveis*, nesta ordem, são executadas ao longo do processo de elicitación, ou quando o ER achar necessário. *Armazenar Metas Flexíveis* oferece a opção de criar uma nova base de sinônimos para ser utilizada futuramente na própria TEKBS.

3.2.1. Análise de contexto das Metas Flexíveis sinônimas

Para realizar o processo de análise de acoplamento ou aderência entre as MFs, foram seguidos cinco passos: (i) elicitare os sinônimos das MFs raízes utilizando a base de sinônimos de língua inglesa Wordnet; (ii) analisar o significado de cada parte da MF sinônima e verificar se condizem com o significado das partes da MFs raízes; (iii) selecionar somente os tipos e os tópicos que mantiverem o sentido das MFs raízes; (iv) combinar os tipos e os tópicos selecionados no item 3, gerando novas MFs; e (v) avaliar se as novas MFs elicítadas fazem parte do contexto do domínio.

Considerando o domínio do EC, ainda foram executados mais dois passos: (vi) das MFs que pertencem ao contexto, quais são novas e quais já haviam sido elicítadas por Oliveira [9] que realizou o processo do EC manualmente e quais já haviam sido elicítadas pela i*Get.; e (vii) comparar os resultados, tomando como hipótese a obtenção de sinônimos das bases Macmillan ou Thesaurus.

As 306 MFs sinônimas analisadas junto ao contexto do EC foram organizadas em 17 grupos, destes um total de 8 grupos (144 MFs) foram descartados por serem grupos repetidos. Uma nova seleção foi realizada com os 9 grupos (162 MFs) restantes considerando as descrições: (i) não pertencer ao contexto do EC, consideradas como “Fora de Contexto”; (ii) pertencer ao contexto, mas é uma MF repetida dentro do grupo de combinação analisado, chamada de “repetida”; e (iii) ser pertencente ao contexto do EC nomeada de “Válidas”. O processo de análise mais detalhado pode ser visto em [25].

O Gráfico 1 mostra o resultado da segunda análise onde 52,47 % (85) MFs foram descartadas considerando as descrições *i* e *ii* e 47,53 % (77) permaneceram na base por pertencerem ao contexto do EC.

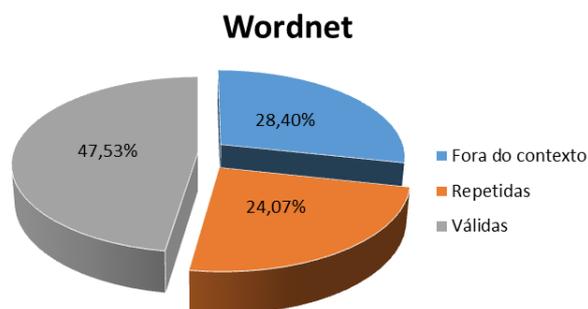


Gráfico 1. Metas Flexíveis por grupo de análise

Para evitar redundância na base da TEKBS o resultado do processo de elicitación foi comparado com as MFs inicialmente elicitadas pela i*Get. Assim, das 77 MFs, um total de 6,49% (5) já haviam sido elicitadas e foram excluídas. As demais 93,51% das MFs elicitadas eram novas e auxiliaram no entendimento do domínio e na descrição de requisitos.

Ao comparar o processo da TEKBS com o processo manual de Oliveira¹⁸ [9], que inicialmente elicitou 37 MFs, tem-se ao final da análise 21 MFs que foram elicitadas por [9] desconsiderando as duplicidades. Dessas, 3 MFs são interseção entre TEKBS e [9]. Assim, totalizam 18 MFs elicitadas pelo processo manual e 74 MFs elicitadas com auxílio da TEKBS, sendo que as 18 MFs podem ser alcançadas por combinações das MFs elicitadas pela TEKBS.

Por fim, comparou-se a aplicação da TEKBS com outros dicionários sendo utilizados o Macmillan e o Thesaurus. Para o passo 1 da análise de contexto, o processo foi manual por não possuírem API ou por ela não estar disponível. Após, foram seguidos os passos 2-6 utilizados para análise com o Wordnet. Na análise de significado dos grupos de sinônimos o Thesaurus foi o melhor dicionário para elicitare MFs com o mesmo significado das MFs raízes. Na análise de significado por grupo de combinações uma nova seleção foi realizada como mostra o Gráfico 2 e o Thesaurus continuou sendo o que mais possibilitou elicitare MFs pertencentes ao contexto do EC.

Comparando-os com o processo manual de Oliveira [9] e com a i*Get, o Wordnet chega a um total de 7 MFs que já tinham sido elicitadas, o Macmillan a um total de 3 e o Thesaurus somente a 2 o que, preliminarmente, confirma a sua superioridade no processo de elicitación de sinônimos.

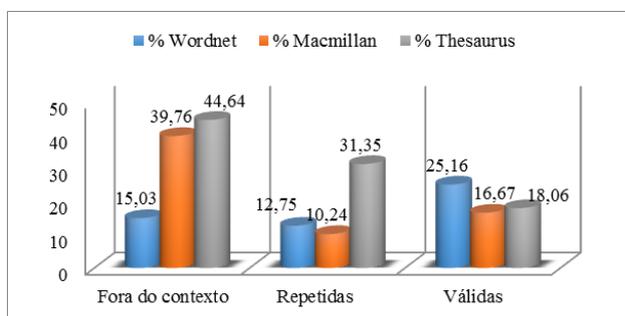


Gráfico 2. Metas Flexíveis selecionadas por grupos de análise e por dicionário.

4. Trabalhos Correlatos

O trabalho de Baia et al. [26] utiliza a estratégia de investigação do vocabulário do domínio descrito por meio do LAL em busca de sinônimos e características que

¹⁸ Para a comparação com Oliveira foi utilizada a Tabela 4.2.16 contida em, na página 147, onde são reunidas todas as MCs e MFs elicitadas.

apontam requisitos de transparências nas descrições do domínio. Nos três trabalhos, o conhecimento descrito por meio do LAL é transformado e investigado pelas regras de inferência. A distinção entre o trabalho de [19] e de [25] para este trabalho está no objetivo final da investigação. Em Baia [26], o objetivo é encontrar requisitos de transparência nos modelos *i**, enquanto o de [19] e de [25] é buscar por metas (concretas ou flexíveis) para a construção dos modelos que serão representados em notação *i**. Além disso, existe outra diferenciação do trabalho de [26] para o trabalho de [25]. Ambos utilizam a sinonímia, porém no trabalho de [25] a sinonímia é utilizada associada ao CLIPS para identificar MFs, facilitando a elicitação e permitindo testar, com intervenção do engenheiro de requisitos, se um sinônimo encontrado é de fato relevante no contexto do domínio da aplicação, evitando assim a associação de um sinônimo fora do contexto do problema a alguma MF já elicitada.

Em [27] foi utilizada uma abordagem baseada em tecnologia da Web semântica para melhorar a qualidade de uma Especificação de Requisitos de Software -SRS. Essa abordagem tem como base a ontologia e o LAL que foram utilizados para criar a OntoSRS e o OntoLEL que têm como função preservar a consistência, a exatidão, a rastreabilidade, a organização e a não ambiguidade de um SRS. Em [27] as autoras procuram contribuir com um processo de especificação orientado para a obtenção de um SRS de qualidade. A abordagem apresentada em [27] difere da abordagem descrita por [25] pelo fato de não elicitar MFs e sim alguns RNFs. Por outro lado, as abordagens de [19] e [25] assim como esta, utilizam o LAL para facilitar o entendimento da linguagem natural. No entanto, somente a abordagem de [25] faz uso de sinônimos para preservar a não ambiguidade.

5. Considerações Finais

O processo de elicitação é complexo e necessita da expertise do engenheiro de requisitos na busca de dados e informações nos diversos mecanismos utilizados para extrair requisitos. A utilização da IA torna a elicitação do domínio mais próximo da realidade dos atores interessados e facilita o entendimento do ER.

A *i*Get* e a TEKBS são utilizadas como guias, orientando o ER e proporcionando uma exploração mais abrangente do domínio. A *i*Get* também fornece um mecanismo capaz de gerar base de conhecimento expansível, além de suporte a utilização do Método ER*i*c*.

Na TEKBS, em seu processo de verificação de acoplamento das MFs foi possível perceber que o processo de elicitação utilizando o Wordnet é promissor e vem facilitar a fase de entendimento do problema. Apesar de a pesquisa feita utilizando o dicionário de língua inglesa Thesaurus ter sido a que mais permitiu elicitar MFs sinônimas válidas para o contexto do EC. Ademais, ela auxilia o ER na extração de MFs não declaradas explicitamente por meio do uso de sinônimos contribuindo para que um número maior de MFs sejam elicitadas e que ocorra uma diminuição de omissões de MFs.

Os resultados alcançados com o EC são auspiciosos para auxiliar o ER não somente na elicitação, mas no entendimento do problema e na produção de um documento de requisitos mais claro e com mais qualidade. No entanto, a falta de trabalhos relativos à

validação de descrições em i*, implicou em dificuldade de validação das metas extraídas por meio da i*Get e consequentemente gerou dificuldades de validação da própria ferramenta. Considerando a TEKBS, apesar de existirem trabalhos que abordem a elicitação, sua grande maioria está voltada para os RNFs e não para MFs o que aumentou a dificuldade de comparação com as MFs elicítadas. Todavia, por ser um processo dependente do ER, as MFs elicítadas utilizando o EC foram consideradas essenciais para o entendimento do problema. Dentro das consultas realizadas nenhum trabalho foi localizado na mesma linha de pesquisa¹⁹, o que sugere que a linha de pesquisa requer mais investigação.

Referências Bibliográficas

1. Jun, L., Qiuzhen, W., Qingguo, M.: The effects of project uncertainty and risk management on IS development project performance: A vendor perspective. *International Journal of Project Management* 29, 923-933 (2011)
2. Ernst, N., Borgida, A., Jureta, I.J., Mylopoulos, J.: An Overview of Requirements Evolution. *Evolving Software Systems*, pp. 3-32. Springer (2014)
3. Cysneiros, L.M.: Requisitos não funcionais: da elicitação ao modelo conceitual. Departamento de Informatica, vol. Doutor, pp. 224 f. Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2001 (2001)
4. van Lamsweerde, A.: Goal-oriented requirements engineering: a roundtrip from research to practice [engineering read engineering]. In: 12th IEEE International Requirements Engineering Conference, pp. 1-4. IEEE Computer Society, (2004)
5. Regev, G., Wegmann, A.: Where do goals come from: the underlying principles of goal-oriented requirements engineering. In: 13th IEEE International Conference on Requirements Engineering, pp. 353-362. IEEE, (2005)
6. Yu, E.S.K.: Modelling strategic relationships for process reengineering. *Computer Science*, vol. Phd, pp. 131. University of Toronto, Toronto (1995)
7. Trondal, J., Peters, B.G.: The rise of European administrative space: lessons learned. In: *Journal of European Public Policy*, pp. 295-307. (2013)
8. Alencar, F.M.R., Silva, C.T.L.L., Lucena, M., Castro, J.F.B., Santos, E., Ramos, R.A.: Improving the Understandability of I* Models. In: 10th International Conference on Enterprise Information Systems, pp. 129-136. (2008)
9. Oliveira, A.P.A.: Engenharia de Requisitos Intencional: Um Método de Elicitação, Modelagem e Análise de Requisitos. Departamento de Informática, vol. Doutor, pp. 261f. Pontifícia Universidade Católica do Rio de Janeiro, 2008 (2008)
10. Goguen, J.A., Linde, C.: Techniques for requirements elicitation. In: *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, pp. 152-164. IEEE, (1993)
11. Cysneiros, L., Yu, E., Leite, J.C.S.P.: Cataloguing non-functional requirements as softgoal networks. In: *Proc. of requirements engineering for adaptable*

¹⁹ Trabalhos que utilizem LAL, IA, sinonímia e o i* para auxiliar no processo de elicitação de metas concretas e flexíveis.

- architectures. 11th international requirements engineering conference, pp. 13-20. (2003)
12. Vinay, S., Aithal, S., Sudhakara, G.: Effect of Contribution Links on Choosing Hard Goals in GORE Using AHP and TOPSIS. *Emerging Research in Electronics, Computer Science and Technology*, pp. 737-745. Springer (2014)
 13. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. *Software Engineering, IEEE Transactions on* 18, 483-497 (1992)
 14. Chung, L., Nixon, B., Yu, E., Mylopoulos, J.: *Non-functional Requirements in Software Engineering*. Kluwer Academic Publishers (2000)
 15. Rawlins, M.C., Chung, L.: In Pursuit of Better EDI: An Introduction to the Use of Non-Functional Requirements in Designing EDI Standards and Architectures. for X12 Strategic Implementation Task Group and UN EDI Standards Committee (1999)
 16. Riley, G. A Tool for Building Expert Systems. 2013. Disponível em: < <http://clipsrules.sourceforge.net/> >. Acesso em: 25 de março de 2014.
 17. Deloach, S.A., Wood, M.F., Sparkman, C.H.: Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering* 11, 231-258 (2001)
 18. Garcia, A.F.: From Objects to Agents: an Aspect Oriented Approach. Departamento de Informática, vol. Doutor, pp. 298 f. Pontificia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004 (2004)
 19. Cunha, L.G.: Mapeamento de especificações LAL - Léxico Ampliado de Linguagem em modelos intencionais. Inédito. Departamento de Informática, vol. Dissertação Universidade Federal de Viçosa, 2014 (2014)
 20. Barion, E.C.N., Lago, D.: Mineração de textos. *Revista de Ciências Exatas e Tecnologia* 3, 123-140 (2008)
 21. Harman, D.: How effective is suffixing? *JASIS* 42, 7-15 (1991)
 22. Negnevitsky, M.: *Artificial intelligence: a guide to intelligent systems*. Pearson Education (2005)
 23. Cares, C., Franch, X., Perini, A., Susi, A.: Towards interoperability of iStar models using iStarML. *Computer Standards & Interfaces* 33, 69-79 (2011)
 24. JUCMNAV. jUCMNav: Juice up your modelling! , 2014. Disponível em: < <http://cserg.site.uottawa.ca/ucm/bin/view/ProjetSEG/WebHome> >. Acesso em: 06 de dezembro de 2014.
 25. Lana, C.A.: Técnica baseada em sinônimos para auxiliar na elaboração de modelo iStar. Departamento de Informática, vol. Mestrado, pp. 141f. Universidade Federal de Viçosa, 2014 (2014)
 26. Baía, J.W., Braga, J.L.: Uso de Sinônimos na Identificação de Atributos de Transparência. In: *Workshop em Engenharia de Requisitos* pp. 94-104. (2013)
 27. Castañeda, V., Ballejos, L.C., Calusco, M.L.: Improving the Quality of Software Requirements Specifications with Semantic Web Technologies. In: *Workshop em Engenharia de Requisitos -WER*, pp. 1-14. (2012)