

Engenharia de Software - SBU  
Engenharia: Software  
Engenharia: Requisitos  
Especificações: Requisitos

# Uma Abordagem Baseada em Regras Heurísticas para Construção de Especificações Diagramáticas de Requisitos

CNPq 1.03.03.00-6

José Mauro Volkmer de Castilho (in memoriam)

Stanley Loh\*\*

Carolina Sturm Trindade\*

## Resumo

Este trabalho apresenta uma abordagem para elicitación e especificación de requisitos com o uso de ferramentas automatizadas baseadas em conhecimento. As especificações geradas são representadas em notações diagramáticas (diagramas), e as bases de conhecimento contêm regras heurísticas coletadas através de estudos de casos e análise de procedimentos. A vantagem do uso de conhecimento declarativo (representado pelas regras heurísticas) é que as tarefas referentes ao processo de Engenharia de Requisitos dificilmente poderiam ser realizadas automaticamente através de processos algorítmicos. Neste artigo, são discutidos dois experimentos de especificação segundo a abordagem (um deles gera um diagrama E-R e outro, um DFD).

## Abstract

This work presents an approach to requirements elicitation and specification through the use of automatic tools and knowledge bases. Specifications are represented in diagrammatic notations and knowledge bases use heuristic rules. The main benefit to the Requirements Engineering process is the use of declarative knowledge instead of procedural one. In this article, experiments that generate an E-R Model and a Data Flow Diagram are discussed.

---

\* [sturm@inf.ufrgs.br](mailto:sturm@inf.ufrgs.br) Instituto de Informática-UFRGS, Caixa Postal 15064-CEP 91501-970 Porto Alegre, RS

\*\* [loh@inf.ufrgs.br](mailto:loh@inf.ufrgs.br) - UCPEL, ULBRA, CPGCC/UFRGS

## 1 INTRODUÇÃO

Segundo [2], o objetivo principal da Engenharia de Requisitos é identificar as necessidades do sistema de software a ser construído, e definir o comportamento externo de uma solução computacional para ele. [3] define Engenharia de Requisitos como o processo de elicitar, analisar, e codificar/documentar requisitos do sistema.

- Mais precisamente, Engenharia de Requisitos abrange as atividades de:
  - coleta dos requisitos do sistema,
  - análise e identificação de requisitos,
  - especificação e documentação dos requisitos,
  - validação dos requisitos.

Na literatura sobre engenharia de software, encontram-se diversas abordagens para engenharia de requisitos baseadas em enfoques ou métodos diferentes, os quais representam os requisitos do sistema através de linguagens ou notações próprias.

Na maioria dos casos, são usados diagramas para especificar os requisitos, resultando no que se convencionou chamar de Especificações Diagramáticas. As várias notações ou métodos permitem especificar tipos específicos de requisitos.

Por exemplo, há o Diagrama Entidade-Relacionamento (DER), que permite especificar dados e seus relacionamentos, e o Diagrama de Fluxo de Dados (DFD), que permite representar processos. Há também os vários tipos de diagramas Orientados a Objetos, que especificam objetos, suas estruturas e métodos, e outros tantos diagramas, como pode ser visto em [6] (diagramas de transição-estado, diagramas de navegação, diagramas de ação, etc).

Para cada método, existem ferramentas CASE que auxiliam na construção das especificações. Apesar de sua grande utilidade, a maioria destas ferramentas somente apóia o desenho e a verificação de tais diagramas, não automatizando a própria tarefa de identificar e representar os requisitos.

Isto é devido às características das próprias atividades da Engenharia de Requisitos, que são muito dependentes da ação de especialistas humanos, os quais utilizam sua experiência e seu conhecimento, adquiridos ao longo do tempo, para realizar tais tarefas. Tais características tornam difícil a construção de software baseado em métodos procedurais (algoritmos) para mecanizar estas tarefas.

A abordagem tomada neste trabalho utiliza sistemas baseados em conhecimento para realizar automaticamente a elicitação (coleta, análise e identificação) e a especificação dos requisitos de um sistema. As ferramentas de software propostas e/ou implementadas

nesta abordagem se utilizam de bases de conhecimento (extraído de especialistas humanos e representado na forma de regras heurísticas) para realizar tais processos.

As regras heurísticas, conforme [4], não garantem o sucesso do processo (ao contrário dos algoritmos), mas seu poder está justamente na possibilidade de utilizar soluções testadamente efetivas em novas situações. Uma regra heurística deve indicar a situação em que pode ser usada e a ação recomendada (exemplo: quando estiver chovendo, deve-se usar o guarda-chuva). Assim, pode-se generalizar as regras, isto é, se uma regra R funcionou numa situação S, então é provável que uma regra R' (semelhante a R) possa ser utilizada numa situação S' (semelhante a S).

A vantagem das heurísticas é que elas representam um conhecimento declarativo e não procedural, permitindo que soluções possam ser aplicadas em casos onde não se sabe claramente que passos seguir ou por onde começar. Além disto, a abordagem aqui apresentada pretende ser independente de enfoque ou método, permitindo a construção de diversas especificações de requisitos diferentes, o que dá uma maior liberdade na realização dos sistemas.

O presente trabalho propõe uma abordagem para Engenharia de Requisitos baseada em conhecimento heurístico para a realização das diversas etapas. A proposta é validada com a apresentação de dois experimentos de construção de especificações diagramáticas (DER e DFD) dentro da abordagem.

O texto está organizado como segue: na próxima seção é apresentada uma visão geral da abordagem. Depois, serão apresentados os trabalhos anteriores do grupo, que baseiam a abordagem proposta. A seção seguinte apresenta a construção de diagramas E-R e a construção de DFD's a partir de textos. As conclusões discutem as contribuições desta abordagem, suas limitações e possibilidades de aperfeiçoamento e extensão.

## 2 VISÃO GERAL DA ABORDAGEM

A primeira etapa desta abordagem tem por objetivo coletar um conjunto de informações (nas quais os requisitos devem estar incluídos), através de interações com o usuário, representante da organização ou ambiente a ser favorecido. Como resultado, tem-se um documento escrito em Linguagem Natural (LN), contendo as informações fornecidas pelo usuário, na forma como este as forneceu. A figura 1 apresenta os principais componentes e processos da abordagem para Engenharia de Requisitos apresentada neste artigo.

Depois vem a etapa de organização dos requisitos, onde o documento em linguagem natural será analisado e transformado para uma linguagem mais restrita (um subconjunto da LN). A finalidade de tal processo é simplificar a linguagem e começar a separação das partes do documento por tipo de requisito (requisitos sobre dados, operações, restrições de integridade, etc).

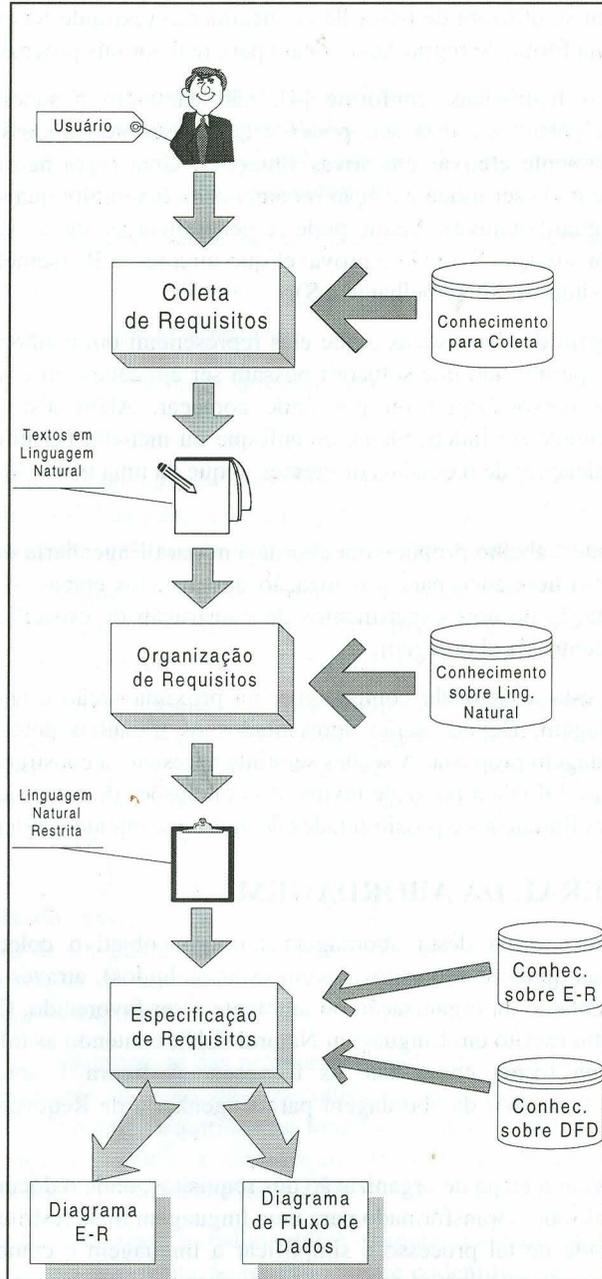


Figura 1: Visão Geral da Abordagem

A terceira etapa da abordagem se preocupa em identificar e representar os requisitos conforme o modelo ou linguagem de especificação escolhido. É tomado como entrada o documento em LN restrita e o resultado da etapa é um diagrama (ou especificação diagramática).

Para a validação dos diagramas, a abordagem ainda não possui ferramentas, mas este é um passo futuro.

Cada etapa desta abordagem é assistida por uma ou mais bases de conhecimento, contendo regras heurísticas. Este conhecimento é extraído de especialistas através da análise de casos, ou seja, o trabalho do especialista é observado em vários experimentos e cada passo documentado.

Após esta coleta inicial de regras, busca-se generalizar as principais regras ou conselhos que o especialista utiliza, agrupando-as por semelhança ou assumindo sua validade para um número maior de situações.

A cada novo experimento, o especialista deve utilizar somente as regras existentes ou então modificá-las, assim refinando e aperfeiçoando a base de conhecimento. Se o especialista necessitar usar uma regra que não existe, então esta deve ser acrescentada à base.

Após um número razoável de experimentos, quando se notar que o conjunto de regras converge, isto é, não aumenta mais, nem surgem regras mais específicas (detalhamentos das regras genéricas), a base de conhecimento é dada como definida e passa-se então ao processo de validação das regras através de experimentos feitos por pessoas leigas ou com uso de ferramentas automatizadas.

### **3 COLETA E ORGANIZAÇÃO DOS REQUISITOS**

As duas primeiras etapas desta abordagem já estão sendo assistidas por ferramentas de software, sendo que há uma base de conhecimento heurístico para cada uma destas etapas.

Para a coleta de requisitos, existe uma ferramenta automatizada (apresentada em [5]), que entrevista usuários e documenta as informações fornecidas. A base de conhecimento é constituída de um conjunto de perguntas e de regras que relacionam as perguntas entre si. Existem perguntas mais genéricas para coletar as primeiras informações. Após, a ferramenta utiliza regras mais detalhadas, que analisarão as respostas das primeiras perguntas e formularão novas perguntas ao usuário. Por fim, a ferramenta extrai palavras-chave de todas as respostas fornecidas e gera as últimas perguntas ao usuário.

A seguir são apresentados exemplos de perguntas e regras que formam a base de conhecimento para a coleta dos requisitos.

- Quem são os clientes da Organização ?
- Como a Organização atende \_\_\_\_\_ ? (completar com as respostas da primeira pergunta)
- Fale mais sobre \_\_\_\_\_ ? (completar com palavras-chave)

O produto resultante desta etapa de coleta é um documento em Linguagem Natural livre, contendo as respostas fornecidas pelos usuários, nas quais devem estar incluídos os requisitos do sistema, se o processo ocorreu com sucesso.

Após a etapa de coleta, este documento deverá ser analisado e transformado para textos em linguagem natural restrita (um subconjunto da LN). Nesta segunda etapa, as frases mais complexas são decompostas em frases mais simples, para facilitar a etapa seguinte de identificação e documentação dos requisitos.

Durante este processo de transformação/decomposição, certamente algumas informações se perdem (por exemplo, as conjunções que conectam logicamente as orações). Entretanto, tal atitude é intencional, ou seja, apenas alguns tipos de frases (as que supostamente contêm os requisitos do sistema) devem permanecer. Esta seleção é feita conforme os tipos de requisitos que deverão ser documentados nos diagramas de especificação (por exemplo, requisitos sobre dados, sobre processos, sobre restrições de integridade).

Exemplos de formatos permitidos para as frases da LN restrita aparecem no quadro a seguir (as orações compostas devem ser decompostas para estes formatos).

SUJEITO VERBO COMPLEMENTO (frase que contém requisitos de dados)  
VERBO COMPLEMENTO (frase que contém requisitos sobre processos)  
SUJEITO “deve” VERBO COMPLEMENTO (frase que contém restrição de integridade)

## 4 ESPECIFICAÇÃO DE REQUISITOS

Neste artigo, a etapa de especificação de requisitos só será exemplificada para os diagramas tipo E-R e DFD. Entretanto, a abordagem proposta aqui permite que outros tipos de especificações diagramáticas possam ser utilizadas, como será discutido nas conclusões.

### 4.1 Construção de Diagramas E-R

A especificação de Diagramas E-R é feita, nesta abordagem, de maneira automática por um sistema baseado em conhecimento, chamado de Assistente Projetista.

O Assistente Projetista é uma ferramenta baseada em técnicas de Processamento de Linguagem Natural que usa regras heurísticas para dirigir a análise sintática das frases do texto pré-formatado (descrevendo os requisitos de dados do sistema) e gerar um diagrama

E-R correspondente. O Assistente Projetista (AP) é composto por: um Analisador Morfológico, um Dicionário Morfológico, um Gerador de Dicionário, um Dicionário Prolog, um Analisador Sintático e Regras Heurísticas. A figura 2 mostra a arquitetura do AP.

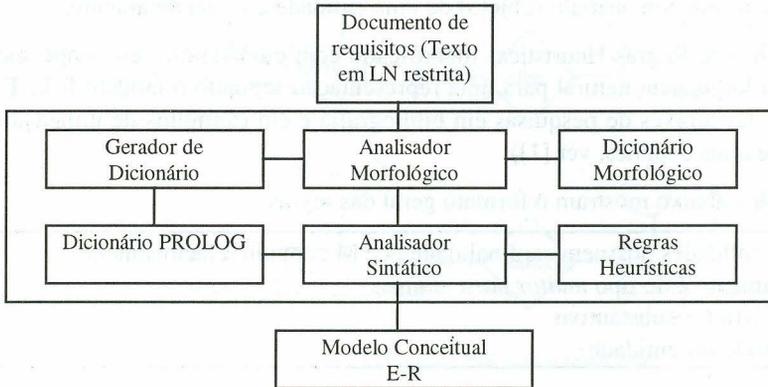


Figura 2 Arquitetura do AP

O analisador morfológico permite analisar morfológicamente uma frase em linguagem natural, usando um dicionário morfológico. São identificados os substantivos, adjetivos, verbos, e outras informações, como gênero, singular, etc, para que o tradutor de frases possa fazer o reconhecimento das estruturas das frases em linguagem natural, e assim executar a tradução correta [1].

O dicionário morfológico armazena palavras da língua portuguesa, particionadas, em estruturas de radicais (que armazena os radicais das palavras da língua portuguesa) e na estrutura de terminações (que armazena as respectivas terminações).

O analisador sintático está escrito em PROLOG, e precisa usar um dicionário para realizar sua tarefa. Para que o processamento sintático, e o próprio código do analisador, não fiquem grande demais, o “gerador de dicionário” recebe do analisador morfológico apenas as palavras do texto que vai ser analisado, com as respectivas classificações morfológicas, e as transforma em “fatos” PROLOG.

O analisador sintático usa Gramáticas de Cláusulas Definidas (DCG), e efetua a análise sintática de forma *ascendente*, que pareceu mais eficiente para a situação disponível.

Para poder definir o tipo de regras que compõem a gramática, é necessário levar em consideração o tipo especial de oração ou construção frasal que expressa requisitos (aqui neste caso, requisitos de dados, pensando em modelagem conceitual de Sistemas de Bancos

de Dados). Levou-se em conta a semântica do modelo E-R que será usada. A mínima representação de uma aplicação deve ser composta de um relacionamento bitransitivo. Por exemplo, um verbo bitransitivo é uma forte indicação da presença de duas entidades (sujeito e objeto) e um relacionamento (verbo transitivo); um verbo de ligação é uma forte indicação da presença de um atributo (objeto) de uma entidade ou relacionamento.

Um conjunto de Regras Heurísticas identificado com cuidado dirige o mapeamento dos requisitos em linguagem natural para uma representação segundo o modelo E-R. Estas regras foram obtidas através de pesquisas em bibliografia e em exemplos de utilização do modelo E-R (para mais detalhes, ver [1]).

Os exemplos abaixo mostram o formato geral das regras:

Regra 1: **Se** duas entidades possuem cardinalidade N e M com um relacionamento **então** o relacionamento é do tipo *muitos para muitos*.

Regra 2: **Se** a palavra for substantivo **então** a palavra pode ser entidade

Para fazer o mapeamento do texto a seguir para o diagrama E-R correspondente (figura 3), foram utilizadas 10 regras heurísticas. Em [1] estão definidas vinte e três regras, mas o conjunto certamente é ainda incompleto. A tarefa de completá-lo vai depender do rumo dos trabalhos futuros.

Texto:

*As publicações são editadas por editoras diferentes que possuem nomes, endereços. As publicações possuem um código, um título e refere-se a um tópico. Cada tópico possui um código, um nome, uma área de pesquisa. Nas publicações interessam os livros, os quais possuem uma tiragem. Os livros são adotados por diversas universidades. Cada universidade possui um nome, e uma cidade.*

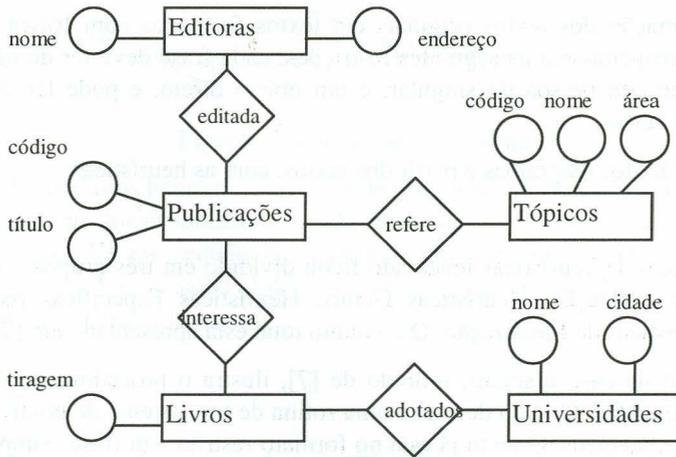


Figura 3 Diagrama E-R para o texto dado.

As interfaces de entrada e saída foram desenvolvidas com uma ferramenta chamada Guide, disponível em ambiente Unix. A interface de entrada é um editor de texto e a interface de saída é um editor gráfico. Todos os módulos deste protótipo foram implementados. O Analisador Morfológico e o Gerador de Dicionário foram implementados em C e ambiente UNIX. O Analisador Sintático e as regras heurísticas foram implementadas em Prolog.

A versão do Assistente Projetista que foi implementada apresenta algumas limitações. As estruturas de frases da língua portuguesa, possíveis de serem usadas pelo protótipo, são simples, porém o domínio de conhecimento não é limitado para um assunto específico, devido às eficientes estruturas de representação dos dicionários. Nem todas as regras heurísticas do Assistente Projetista foram implementadas, devido à abrangência do sistema, pois muitas regras dependem de interação do sistema com o usuário durante a modelagem. Porém, as regras implementadas parecem suficientes para traduzir requisitos expressos em frases com estrutura simples da língua portuguesa para o modelo E-R.

## 4.2 Construção de DFD's

Neste experimento de construção de especificações diagramáticas tipo DFD, foram seguidos os seguintes passos:

- coleta de regras heurísticas, feita com análise de casos e reconhecimento de padrões;

- obtenção de textos, de duas formas: (i) solicitação a usuários, para que descrevessem detalhadamente suas rotinas de trabalho na execução de determinada tarefa que se quer especificar em DFD; (ii) textos obtidos em um manual de procedimentos disponível;

transformação dos textos originais em textos formados com frases mais simples, onde as frases respeitassem as seguintes restrições: cada frase deve ter de um único verbo principal, na terceira pessoa do singular, e um objeto direto, e pode ter objeto indireto, dependendo do verbo;

construção dos diagramas a partir dos textos, com as heurísticas.

O conjunto de heurísticas levantado ficou dividido em três grupos, que configuram uma ordem de aplicação: Heurísticas Gerais, Heurísticas Específicas para Grupos de Verbos, e Heurísticas de Finalização. O conjunto total está apresentado em [7].

O estudo da caso a seguir, retirado de [7], ilustra o procedimento de extração de DFD a partir de textos. O texto descreve uma rotina de um sistema de controle de estoques de fábrica de embalagens. O texto já está no formato restrito, em frases simples preparadas manualmente.

Texto que descreve a rotina: “Compra de Produtos das Fábricas”:

- *Verifica na ficha do produto a quantidade dos produtos estocados.*
- *Anota o nome dos produtos com baixo estoque e o nome dos fornecedores.*
- *Busca no fichário de fornecedores os dados dos fornecedores.*
- *Negocia a compra com as fábricas fornecedores.*
- *Anota em uma ficha o nome dos produtos encomendados, a quantidade de cada produto, o valor unitário de cada produto, o valor total da compra, a data do pedido e a data de entrega.*
- *Entrega a ficha para o funcionário da seção de contabilidade.*

A seguir, cada frase é analisada com heurísticas gerais e com heurísticas dos grupos de verbos, gerando partes do diagrama, que serão depois combinadas com as heurísticas de finalização.

Por exemplo, para o processamento da primeira frase são usadas a regra 1 (enquadramento em grupo de verbos, no caso “verifica”, verbo de *leitura*), a regra 2 (cria nome do processo, no caso “verifica qtd prod est”), regra 3 (numeração de processo, 1), regras 4 e 5 (cria fluxo para cada objeto direto, e insere objeto no dicionário de termos), regra 7 (cria depósito, objeto indireto, “ficha de produto”), regras de leitura 2 e 4 (define direção dos fluxos). O resultado para a frase aparece na figura 4.



Figura 4 Resultado do processamento

Depois de usadas as heurísticas gerais e de grupos de verbos, analisa-se a integração dos processos parciais correspondentes a cada frase com as heurísticas de finalização. Os processos são analisados em seqüência: primeiro o processo de entrada (4, na figura 5), depois o de leitura (1) e os de saída (4 de novo, e 6), e por último os de modificação (2, 3 e 5). O DFD resultante aparece a seguir.

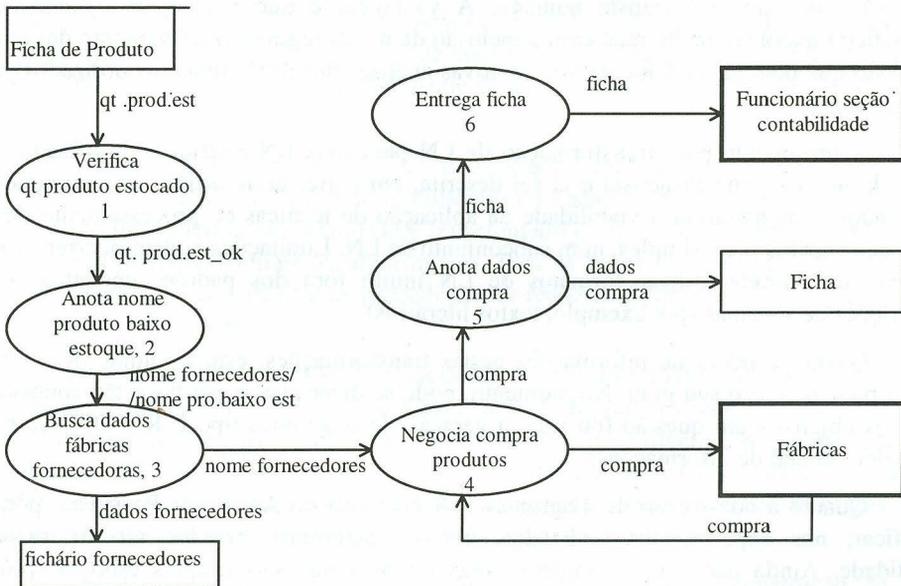


Figura 5 Análise dos processos

Com base no procedimento descrito acima está sendo implementada uma ferramenta protótipo em Delphi para a construção de especificações tipo DFD (incluindo reconhecimento sintático de frase simples e geração de representação diagramática com heurísticas), como parte de uma dissertação de mestrado. A mesma dissertação estuda a implementação, seguindo a abordagem baseada em heurísticas, de ferramentas para construção das especificações diagramáticas orientadas a objeto tipo OMT.

## 5 CONCLUSÕES

Este artigo apresentou uma abordagem para realizar a Engenharia de Requisitos, na qual são utilizadas ferramentas automatizadas construídas com bases de conhecimento. Nesta abordagem, o conhecimento é representado por regras heurísticas, as quais são coletadas e representadas de maneira simples, contribuindo para automatizar em parte tal processo.

Alguns experimentos foram realizados seguindo-se a abordagem descrita neste artigo.

No caso da coleta de requisitos, a ferramenta mostrou-se útil para coletar um bom volume de informações. Entretanto, ficou claro que a mesma não tem capacidade ainda para substituir um especialista humano. A vantagem é que a base de conhecimento heurístico poderá ser melhorada com a inclusão de novas regras e o refinamento das atuais, processo que poderá ser feito através de novas análises dos procedimentos utilizados pelos especialistas.

A ferramenta para transformação da LN para uma LN restrita (simplificada) está inserida no Assistente Projetista e já foi descrita, em parte, neste artigo. Os experimentos realizados demonstraram a viabilidade da aplicação de técnicas de processamento de LN para gerar textos mais simples, num subconjunto da LN. Limitações podem ocorrer quando o texto de entrada utilizar formatos de LN muito fora dos padrões encontrados em descrições de sistemas (por exemplo, textos literários).

Quanto à perda de informações nestas transformações, estudos ainda deverão ser feitos para avaliar o seu grau. No momento, pode-se dizer que as perdas estão controladas para os objetivos em questão (ou seja, a geração de diagramas tipo E-R e DFD com um esqueleto inicial de informações).

Quanto à construção de diagramas E-R com uso do Assistente Projetista, pôde-se verificar, nos experimentos realizados, que os diagramas gerados são de razoável qualidade. Ainda não estão completos, mas também não são simples esboços, pois já contemplam análises mais sofisticadas como o caso de uma entidade possuir vários relacionamentos. Algumas limitações foram detectadas para os casos de frases ambíguas, o que pode levar a erros no diagrama, devido a mal-interpretações. Entretanto, a vantagem principal é a geração de um diagrama inicial com bastante informações (mesmo que com alguns problemas), o que não impede a intervenção de um especialista humano para revisar o diagrama gerado.

Para a construção de DFD's ainda não há uma ferramenta de software que faça automaticamente tal atividade. Contudo, a base de regras heurísticas já foi testada com pessoas leigas, que não conheciam DFD's, e os resultados mostraram-se bastante satisfatórios. Os diagramas gerados pelos leigos foram analisados por especialistas e considerados de boa qualidade. Tais experimentos deixam clara a viabilidade do

desenvolvimento de ferramentas automatizadas que gerem DFD's a partir de textos em LN restrita. Para tanto, bastará incorporar, além das regras, um analisador morfológico-sintático (como no Assistente Projetista).

De forma geral, pode-se dizer que a referida abordagem comprova a viabilidade de serem utilizadas ferramentas automatizadas para a Engenharia de Requisitos, com o objetivo de aumentar a produtividade e a qualidade das especificações geradas. Mesmo que os diagramas gerados pelo estado atual das ferramentas ainda não possam ser comparados aos gerados por especialistas, há a possibilidade de se aperfeiçoar as bases de conhecimento até que se obtenha a qualidade desejada.

Mais experimentos deverão ser feitos para validar e aperfeiçoar as regras heurísticas utilizadas e para avaliar também a qualidade dos diagramas gerados (incluindo, quando completos e corretos eles são). Também deverá ser comparado o uso desta abordagem em relação aos meios tradicionais (manuais) de construção de diagramas. No momento, pode-se dizer (sem se considerar a qualidade) que o processo fica mais rápido com o uso de ferramentas automatizadas.

Outra vantagem desta abordagem é a possibilidade de realizar o processo de Engenharia de Requisitos com uso de ferramentas baseadas em conhecimento heurístico, o qual é muito mais fácil de ser adquirido e representado do que o conhecimento procedural utilizado nos métodos formais ou algorítmicos. Para as bases de conhecimento utilizadas nesta abordagem, bastou analisar os procedimentos realizados por especialistas para coletar regras heurísticas eficientes.

Esta abordagem também se diferencia das demais ao possibilitar o uso concorrente de diversos métodos para as etapas da Engenharia de Requisitos. Isto permitirá construir ferramentas que façam a verificação ou a integração entre os diversos diagramas gerados. Também poderão ser acrescentadas, de forma incremental, novas bases de conhecimento para outros métodos de especificação. Está em estudo a identificação de regras mais genéricas que possam servir em vários métodos (ou em classes de métodos).

Como trabalhos futuros também estão em planejamento a construção de uma ferramenta para especificação de requisitos com diagramas orientados a objetos e o desenvolvimento de *parafraseadores* baseados em conhecimento (ferramentas que geram textos em LN a partir de diagramas, a fim de validar os requisitos junto ao usuário).

## 6 Referências Bibliográficas

- [1] BIGOLIN, N. M. Aplicação de técnicas de tratamento de linguagem natural para suportar o projeto de bancos de dados. Curso de Pós-Graduação em Ciência da Computação, UFRGS. Fevereiro de 1994. (Dissertação de Mestrado)
- [2] DAVIS, Alan M. & HSIA, Pei. "Giving voice to requirements engineering". IEEE Software, v.11, n.2, March 1994.

- [3] FRASER, Martin D. et alli. "Informal and formal requirements specification languages: bridging the gap". IEEE Transactions on Software Engineering, v.17, n.5, May 1991.
- [4] LENAT, D. B. "The nature of heuristics". Artificial Intelligence, v.19, n.2, October 1982.
- [5] LOH, S.; POETA, C. R.; CASTILHO, J. M. V. "Automating the Software Requirements Elicitation Process: proposal and prototype". Proceedings. I International Conference on Information Engineering. Buenos Aires, Argentina, 6-8 de Dezembro de 1994.
- [6] MARTIN, J.; McCLURE, C. Técnicas Estruturadas e CASE. Prentice-Hall, 1991.
- [7] TRINDADE, C. S. Definição de regras heurísticas para extrair diagramas de fluxo de dados a partir de textos. Curso de Bacharelado em Informática, ULBRA. Novembro de 1995. (Trabalho de Conclusão).

Artigo originalmente publicado em: : Jornadas Iberoamericanas de Ingenieria de Requisitos y Ambientes de Software (1998, Abr. 1-3 : TORRES, RS) . Anais. Porto Alegre: Instituto de Informática/UFRGS, 1998. . VOL. 2. P. 212-223.