WORKSHOP DE ENGENHARIA DE REQUISITOS 98 (WER98)

UMA FERRAMENTA BASEADA EM REGRAS HEURÍSTICAS PARA GERAR ESPECIFICAÇÕES DIAGRAMÁTICAS DE REQUISITOS ORIENTADAS A OBJETO

Este é um trabalho do Grupo de Bancos de Dados Inteligentes (BDI),
Instituto de Informática,
Universidade Federal do Rio Grande do Sul (UFRGS).

OBS: Recomenda-se a visualização dessa página em uma resolução mínima de 800x600.

UMA FERRAMENTA BASEADA EM REGRAS HEURÍSTICAS PARA GERAR ESPECIFICAÇÕES DIAGRAMÁTICAS DE REQUISITOS ORIENTADAS A OBJETO

Júlio Hartmann (CIC/UFRGS)
Carolina Sturm Trindade (CPGCC/UFRGS)
Stanley Loh (UCPEL, ULBRA, CPGCC/UFRGS)
José Mauro V. de Castilho (UFRGS) - *in memoriam*

Endereço para Correspondência:
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Compus do Vale – Bloco IV
Avenida Bento Gonçalves, 9500
Porto Alegre, RS – Brasil
CEP 91501-970

RESUMO

Este trabalho apresenta uma ferramenta para a geração automática de diagramas de especificação de requisitos orientados a objeto a partir de textos. São apresentadas as bases de conhecimento desenvolvidas, manualmente, na forma de regras heurísticas, as quais foram obtidas através de estudos de caso e análise de procedimentos junto a especialistas. A utilização de conhecimento humano diferencia a ferramenta apresentada da maior parte das ferramentas CASE existentes, que somente atuam nas etapas seguintes à Engenharia de Requisitos no processo de Engenharia de Software. As especificações geradas pela ferramenta são representações de diagramas conceituais orientados a objeto na metodologia OMT (*Object Modelling Technique*), e o texto analisado é obtido a partir de restrições sintáticas à linguagem natural.

PALAVRAS CHAVES: Engenharia de Requisitos, ferramentas CASE, OMT, Engenharia de Software, Inteligência Artificial, Regras Heurísticas.

ABSTRACT

This article present an automatic tool for requirement elicitation and its object-oriented diagrammatic specification through the use of human knowledge. It presents the knowledge bases manually developed as heuristic rules. The use of human knowledge differentiates this tool from existing CASE tools, that are only useful in the next steps to Requirement Engineering in the Software Engineering process. OMT (Object Modelling Technique) conceptual diagram representation are the automatic tool's output. Sintatic restrictions to natural language (Portuguese) are used for obtaining the input texts.

KEYWORDS: Requirement Engineering, CASE tools, OMT, Software Engineering, Artificial Intelligence, Heuristic Rules.

* Este trabalho é parcialmente apoiado por CNPq/PROTEM e FAPERGS.

Voltar para o topo

1 Introdução

Segundo [1], 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. [2] define Engenharia de Requisitos como o processo de elicitar, analisar, e codificar/documentar requisitos do sistema.

Em [3] é proposta uma abordagem para a Engenharia de Requisitos baseada em conhecimento (figura 1), a qual foi utilizada para desenvolver a ferramenta apresentada neste trabalho. Segundo [3], Engenharia de Requisitos abrange as atividades de: *1.) coleta dos requisitos do sistema; 2.) análise e identificação dos requisitos; 3.) especificação e documentação dos requisitos; 4.) 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. Em geral, tais requisitos, uma vez coletados e identificados, são especificados através de diagramas, resultando no que se convencionou chamar de Especificações Diagramáticas. 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, como pode ser visto em [4]. Há também os diversos tipos de diagramas Orientados a Objetos, que especificam os objetos, suas estruturas e métodos, e outros tantos diagramas.

Para cada método, existem ferramentas CASE (*Computer Aided Software Engineering*) 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.

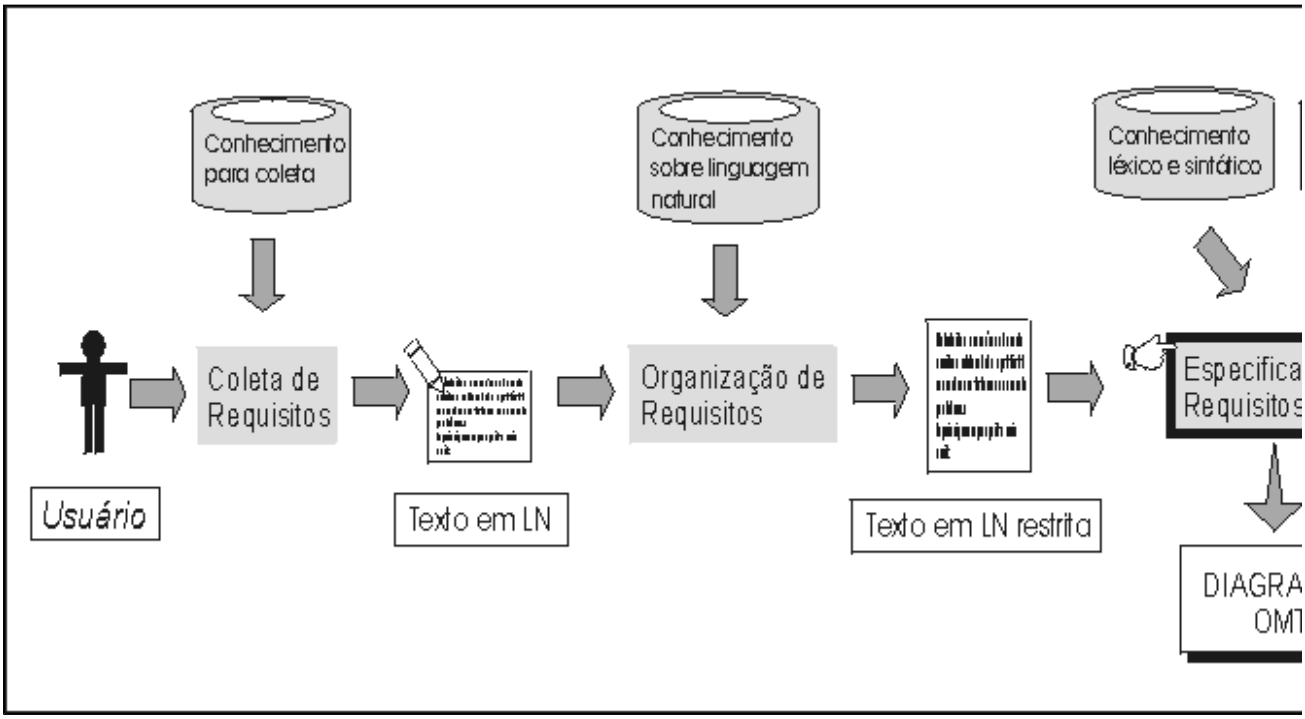

Figura 1: Visão Geral da Abordagem para a Engenharia de Requisitos

Neste trabalho é apresentada uma ferramenta baseada em conhecimento para gerar automaticamente especificações diagramáticas de requisitos de sistemas de informação. A ferramenta se utiliza de bases de conhecimento (extraído de

especialistas humanos e representado manualmente na forma de regras heurísticas) e gera especificações diagramáticas. Os diagramas gerados são aqueles abordados pela metodologia OMT (*Object Modelling Technique*) [5]: Diagrama de Objetos (DO), Diagrama de Estado (DT), Diagrama de Fluxo de Dados (DFD). Entretanto, a abordagem utilizada neste trabalho permite que outros tipos de especificações diagramáticas possam ser utilizadas, como é discutido em [3]. Trabalho semelhante, realizado com diagramas Entidade-Relacionamento (ER), é apresentado em [7].

Na figura 1 é apresentada uma visão geral da abordagem utilizada para a Engenharia de Requisitos baseada em conhecimento. A ferramenta foi desenvolvida para realizar automaticamente a etapa de especificação de requisitos, dentro da abordagem proposta; a elicitação (coleta, análise e identificação) não é tratada neste trabalho. A abordagem completa, que inclui o processo de coleta e organização dos requisitos, pode ser encontrada em [3].

Voltar para o topo

2 A ferramenta desenvolvida

Esta seção apresenta a ferramenta de geração automática de especificações diagramáticas a partir de textos semi-estruturados. A ferramenta foi desenvolvida em Delphi 3; possui uma interface para interação com o usuário (cuja entrada são textos escritos em Linguagem Natural (LN) restrita), além de realizar o processamento dos textos e a apresentação dos resultados (especificações diagramáticas). Os resultados não são apresentados de forma gráfica, mas sim na forma de tabelas de banco de dados, ou através da impressão de uma saída textual representando os diagramas. Além disso, a ferramenta possui um banco de casos. Esse banco de casos armazena diversos casos já devidamente processados e testados, podendo os mesmos ser utilizados pelo usuário como uma uma biblioteca de consulta para suportar a solução de novos problemas.

O processamento dos textos é feito basicamente em três etapas. Primeiro, é feito um reconhecimento léxico de todas as palavras do documento, para fazer sua classificação gramatical. Em seguida, é chamada uma rotina de análise sintática, que separa as frases em componentes sintáticos, tais como: sujeito, verbo, objeto direto, objeto indireto, complemento nominal, etc. Após a separação sintática das frases, são aplicadas as regras heurísticas que foram automatizadas e, assim, é obtido o resultado final, que é armazenado na forma de tabelas de bancos de dados. Um esquema do funcionamento da ferramenta é apresentado na figura 3.

Voltar para o topo

2.1 A interface com o usuário

A interface da ferramenta é uma interface visual de janelas, complementada por menus e botões. Sua função principal é dirigir um usuário leigo na coleta de requisitos.

A entrada da ferramenta são textos em LN restrita, ou seja, um subconjunto da LN associado ao contexto da especificação de requisitos. São reconhecidas somente as sentenças que respeitam a sintaxe descrita na figura 2. Para a obtenção de tais textos, há duas alternativas: uma é a utilização de ferramentas automatizadas que coletem e organizem os requisitos, como é sugerido nas conclusões, gerando-se os textos em LN restrita a partir de textos em LN ou de entrevistas com usuários; a outra é a realização manual dessas tarefas.

Os textos são separados em duas partes: premissas básicas e características funcionais. Na primeira parte são entradas as frases que expressam as características gerais do sistema em si; na segunda, constam as frases sobre o funcionamento do sistema, com ênfase nas interações do mesmo com entidades externas. Ainda com relação aos textos voltados às características funcionais, sugere-se que não se descreva procedimentos ou funções que serão utilizados em outras rotinas ou funções, ou também por outras pessoas [6].

Há restrições à sintaxe do texto de entrada; as frases devem seguir as sintaxes permitidas (as quais são apresentadas na seção 2.2). Além disso, o texto descritivo das premissas básicas do sistema é menos restrito que o das características funcionais; neste último, é imposta mais uma restrição: o sujeito das frases deve ser um agente interno ao sistema sendo modelado. Isso acontece devido à necessidade de definição de fronteiras para a especificação dos requisitos; é necessária a definição de quais os objetos que são entidades externas. No exemplo da seção 3, esta restrição é melhor discutida.

O acionamento de um botão faz com que o texto seja processado. O resultado desse processamento é apresentado na forma de tabelas de banco de dados; por exemplo, os resultados do DFD são armazenados e apresentados através das

tabelas de processos, de entidades externas, de depósitos e de fluxos. Pode, ainda, ser gerada uma saída textual na qual os diagramas de especificação de requisitos são descritos, a partir destas tabelas.

A interface do sistema se caracteriza por ser bastante interativa. Há algumas opções que podem ser ligadas ou desligadas para aumentar ou diminuir esta interatividade no processamento do texto.

Ao se processar um texto novo, recém entrado no programa, deve-se ter o cuidado de ligar a opção de Auto-Inserção de palavras. Esta opção faz com que a cada vez que uma palavra não for reconhecida no texto seja apresentada uma janela com a opção de inseri-la automaticamente em uma tabela (uma das tabelas do Dicionário, que é discutido na próxima seção), de acordo com a sua classe gramatical.

Outras interações com o usuário serão discutidas no decorrer do artigo.

Voltar para o topo

2.2 O reconhecimento léxico e sintático

O reconhecimento léxico é feito através da identificação da classe gramatical de cada palavra presente no texto, o que é feito com a ajuda do Dicionário da ferramenta. Este Dicionário é implementado em um banco de dados relacional, através de diversas tabelas que armazenam as palavras pertencentes à cada classe gramatical. Por exemplo, existem as tabelas de verbos, de substantivos, de adjetivos, de artigos, etc. Algumas tabelas armazenam outros dados, além da classe gramatical, que são utilizadas pelo programa para, durante a análise léxica e sintática, extrair do texto informações que o processamento heurístico utiliza. Na tabela de verbos, por exemplo, são armazenados também o tipo (informação utilizada no processamento heurístico) e a transitividade (dado utilizado pelo processamento sintático) de cada verbo.

Existem alguns grupos de expressões (formadas por mais de uma palavra) que merecem uma atenção especial na análise léxica devido à ênfase no seu tratamento por parte das heurísticas do diagrama de objetos (na atribuição de cardinalidades às associações), e que por isso são colocados em uma tabela separada. No texto, estas expressões aparecem entre colchetes, que são colocados automaticamente ao se acionar um botão. Exemplos dessa expressões são: "mais de um", "somente um", "todo", "cada", etc.

O processamento sintático é feito sobre uma tabela de *tokens*, classificados gramaticalmente, os quais são provenientes da análise léxica. As seguintes sintaxes frasais são reconhecidas:

Sujeito + Verbo + Objeto Direto
Sujeito + Verbo + Objeto Direto + Objeto Indireto *e/ou* Complemento Nominal
Sujeito + Verbo de ligação + Verbo no particípio + Objeto Indireto
Sujeito + Verbo de ligação + Complemento Nominal
Sujeito + Verbo + Objeto Direto, Objeto Direto, ..., **e** Objeto Direto
Sujeito + Verbo + Ob. Direto, ..., **e** Ob. Direto + Ob. Indireto, ..., **e** Ob. Indireto

Figura 2: Sintaxes admitidas pela ferramenta

Como as frases apresentam um estrutura bem definida, e relativamente fixa, os elementos sintáticos são reconhecidos pela posição dos tokens na frase. Além disso, algumas informações armazenadas no Dicionário auxiliam neste reconhecimento. Por exemplo, na identificação do objeto indireto. O objeto indireto se caracteriza por iniciar com uma preposição; só que nem toda preposição marca o seu início, pois podem haver complementos nominais na frase. O Dicionário armazena, para cada verbo, a informação de quais as preposições que geralmente caracterizam seu objeto indireto. Exemplo: na frase "*Busca na ficha de fornecedor os dados dos fornecedores*", extraída de um caso de teste, o objeto indireto "*na ficha de fornecedor*" é identificado pela presença da preposição "*em*" (na contração "na" = "em" + "a"); não é confundido com os complementos nominais "*de fornecedor*" e "*dos fornecedores*".

Voltar para o topo

2.3 O processamento heurístico

Os passos para o desenvolvimento da base de conhecimento heurístico implementado pela ferramenta são descritos em [6]. O programa processa os componentes sintáticos das frases segundo esta base de conhecimento heurístico. A base é implementada através de procedimentos no programa, que imitam o funcionamento das regras. As regras são baseadas na estrutura "Se condição, então Ação". Por exemplo [6]: "*Caso o verbo se identifique como sendo do grupo de entrada de dados, então deve-se i) criar um processo de nome igual verbo da frase seguido de Objeto Direto; ii) criar um fluxo de dados a partir do objeto direto da frase; iii) criar uma entidade externa com o nome do objeto indireto, iv) direcionar o fluxo, com sentido da entidade externa para o processo criado".*

Até o momento foram implementadas as regras para gerar o DO e DFD, faltando ainda as regras para a geração do DTE, as quais estão em estado de desenvolvimento.

O processamento heurístico implementado no sistema está baseado na classificação realizada nos verbos. Dependendo do tipo da frase e do verbo encontrado será procurada uma heurística para seu tratamento; o processamento é feito frase a frase. Os verbos são classificados em grupos: Entrada, Saída, Leitura, Modificação e Cálculo. Além destes grupos, há os verbos de ligação e outros verbos que não se encaixam em nenhum dos grupos anteriores e são colocados no grupo de verbos especiais.

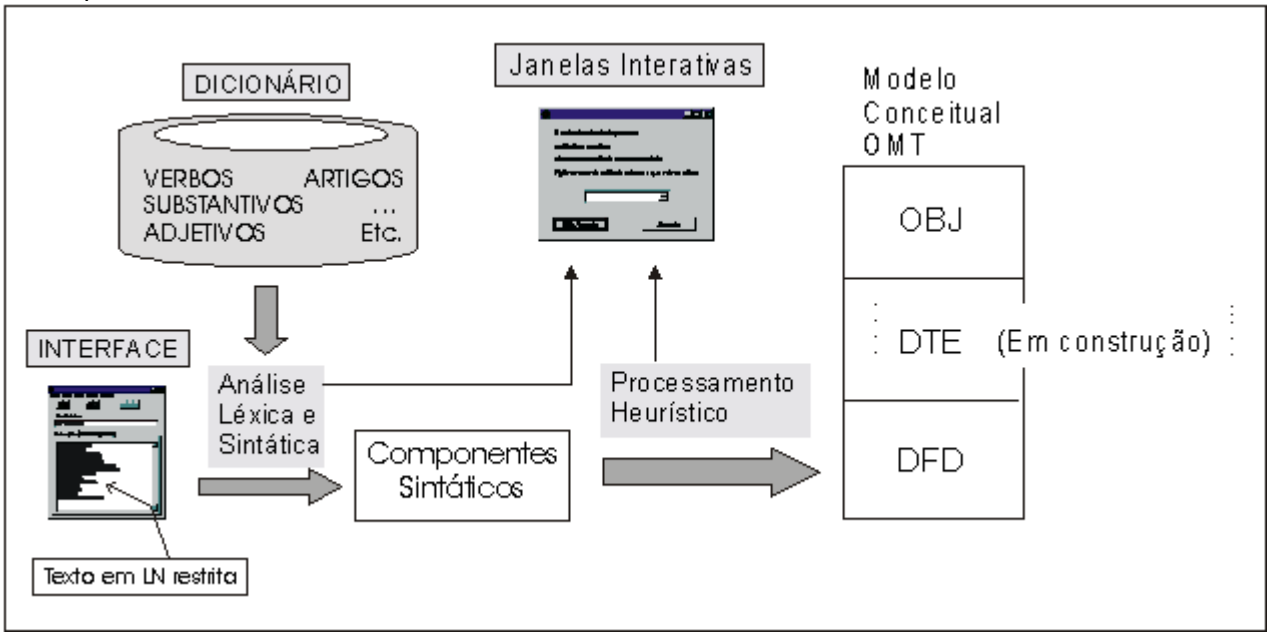

Figura 3: Funcionamento da ferramenta

A geração da especificação do diagrama de objetos é feita a partir do processamento das duas partes do texto entrado, que são "Premissas básicas" e "Características Funcionais" (como introduzido na seção 2.1); em contrapartida, as heurísticas para a geração de DFD's são executadas sobre somente uma parte do texto de entrada, aquela que descreve um caso de uso do sistema (as "Características Funcionais"), contendo frases descrevendo os processos organizacionais e seus relacionamentos com as entidades externas. O conjunto total das regras heurísticas para a geração de DFD´s está publicado em [6], e está integralmente implementado na ferramenta.

O texto de premissas básicas do sistema é formado basicamente por frases com verbos especiais (não caracterizam processamento, apenas associação lógica) ou de ligação, enquanto no texto de funcionalidades quase todos os verbos pertencem a um dos grupos.

As regras heurísticas de geração de DO's identificam as classes, seus atributos, operações e associações. Casos especiais de associação entre as classes, como agregação e herança (os tipos de associação entre classes aqui abordados são largamente discutidos em [5]), também são identificados. Seu conjunto total ainda não foi publicado; será divulgado, futuramente, em uma dissertação de mestrado.

Como exemplo, tomemos a frase "*acervo é composto por uma ou mais fitas*". O verbo identificado ("compor") possui uma heurística de tratamento específico, junto com o verbo "formar", dentro do grupo de verbos especiais; essa frase cria (se já não existirem) as classes "acervo" e "fita", e uma associação de agregação entre as duas classes (a classe "fita" é componente da classe "acervo").

Expressões como "uma ou mais" (da frase de exemplo do parágrafo anterior), são tratadas por heurísticas que determinam as cardinalidades das associações identificadas; nem sempre, entretanto, este tipo de expressão está presente nas frases tratadas. Como é comum uma mesma associação entre duas classes aparecer várias vezes em um texto, estas

cardinalidades são atribuídas à medida em que são encontradas suas evidências.

Mesmo assim, pode acontecer que algumas cardinalidades ainda estejam indefinidas ao final do processamento, por não haver informação suficiente no texto para determiná-las. Neste caso, são feitos interações com o usuário do sistema, com perguntas do tipo: "-*Quantas fitas podem ser entregues a um cliente?*" ou "-*Quantos médicos atendem a um paciente?*".

São fornecidas opções restritas de resposta: um ou mais, somente um, etc., e, a partir das respostas escolhidas, os diagramas são completados.

Voltar para o topo

2.4 O processo de construção da ferramenta

Na figura 4 é esquematizado o processo de construção da ferramenta baseada em regras heurísticas; as fases do processo, resumidamente, foram as seguintes: *1.) estudo de casos e análise de procedimentos junto ao especialista; 2.) definição das regras heurísticas; 3.) validação das regras heurísticas; 4.) implementação das regras heuristicas na ferramenta; 5.) validação da ferramenta desenvolvida*. Este processo foi executado duas vezes – uma vez para cada diagrama implementado.

As três primeiras etapas foram dispendidas na elaboração das regras heurísticas; tais regras representam o conhecimento extraído do especialista, e passaram, após sua definição, por um processo de validação manual. Esta validação foi conseguida com a realização de experimentos com pessoas leigas, que desenharam os diagramas a partir dos textos, guiados apenas pelas heurísticas. O resultado obtido foi um documento textual descritivo das regras heurísticas. A realização dessas etapas foi justificado pela obtenção de um conjunto fechado e validado de regras, preparadas para serem implementadas.

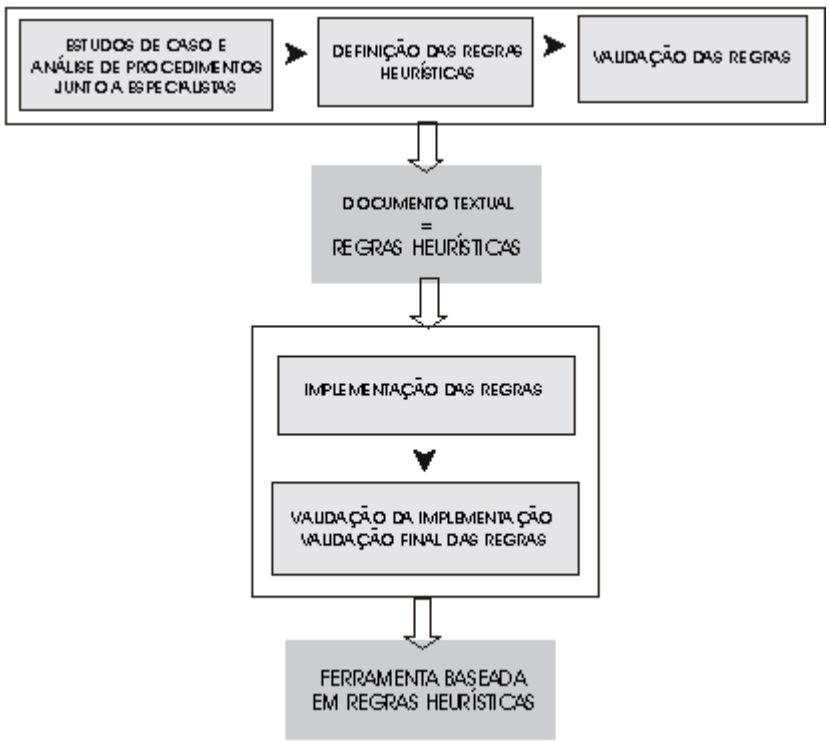

Figura 4: Processo de construção para a ferramenta baseada em regras heurísticas

A etapa seguinte do processo foi a construção da ferramenta propriamente dita, na qual as regras foram implementadas. A implementação foi feita integralmente de forma procedural; ou seja, foram codificados procedimentos que simulam a aplicação das regras. Houve a necessidade de validação da ferramenta, o que foi feito rodando-se casos de teste com resultados já previamente conhecidos, os quais haviam sido obtidos manualmente. Foi feita a comparação de resultados, e ciclos de correção de defeitos na implementação foram realizados, até que se obtivessem automaticamente os mesmos resultados que haviam sido obtidos manualmente, nas fases anteriores do processo. Além disso, novos casos de teste foram realizados, para validar a geração de DFD's em conjunto com o diagrama de objetos.

Voltar para o topo

3 Exemplo

Premissas Básicas	Características Funcionais
Aluno faz matrícula em curso. Curso possui currículo. Currículo é formado por disciplinas. Disciplina possui turmas. Turma tem número de vagas. Aluno tem dados cadastrais. Aluno possui histórico. Aluno pertence a turma. Aluno possui ordem de matrícula. Aluno possui carteirinha. Aluno possui boletim escolar.	*Informa o boletim escolar ao funcionário. Consulta as disciplinas no vetor de possibilidades. Escolhe a turma no vetor de possibilidades. Escreve na folha de matrícula as disciplinas e as turmas. Solicita ao representante de comissão de carreira uma assinatura. Apresenta ao funcionário a folha de matrícula e o comprovante de pagamento da taxa de matrícula. Recebe do funcionário os dados do aluno. Solicita ao funcionário a carteira.*

Figura 5: Exemplo de entrada de requisitos para a ferramenta

Nesta seção é apresentado um exemplo de geração automática obtida utilizando a ferramenta. São apresentados diagramas correspondentes às especificações diagramáticas textuais geradas. O texto em linguagem natural restrita (figura 5) foi extraído de manuais de rotinas da universidade, e refere-se ao processo de pré-matrícula dos alunos.

As figuras 6 e 7 ilustram os resultados obtidos para os diagramas de objetos e funcional, respectivamente, apresentados pelo programa através de uma descrição textual. Observa-se a relação entre os métodos gerados para cada classe no diagrama de objetos e o respectivo processo no DFD.

No diagrama da figura 6, observam-se as classes e atributos gerados no processamento do texto. As classes ("boletim escolar", "matrícula", "curso", etc.) e os atributos ("dados cadastrais" , "dados de aluno") são obtidos a partir da identificação dos substantivos e de seus complementos. As associações e as operações (métodos) de cada classe são identificadas de acordo com os verbos e sua classificação em grupos (os grupos dos verbos são apresentados na seção 2.3). Toda frase gera uma ou mais associações entre classes ou associa atributos a uma classe. Por exemplo, a primeira frase do texto da figura 5 é: "*Aluno faz matrícula em curso*". A partir dessa frase são geradas as classes "aluno", "matrícula" e "curso"; observa-se no diagrama da fig. 6 a associação identificada: aluno é relacionado com curso, e a classe matrícula é definida como um atributo dessa associação.

As operações são identificadas a partir das frases que contém verbos não pertencentes ao grupo especial; a maior parte destas frases pertence ao texto de "Características Funcionais" (figura 5): são frases que expressam entrada, saída, leitura, modificação ou cálculo dos dados. Na frase: "*solicita ao representante da comissão de carreira uma assinatura*", por exemplo, é identificada a operação "solicita assinatura", que é atribuída à classe "representante de comissão de carreira".

A operação "solicita assinatura" corresponde a um processo no DFD gerado ("solicitar assinatura", ver figura 7). Nesse diagrama, a classe "representante de comissão de carreira" aparece como uma entidade externa ao sistema. O atributo dessa classe ("assinatura"), identificado na frase exemplo do parágrafo acima, aparece como um fluxo de dados, que é recebido da entidade externa pelo processo "solicitar assinatura" e passado aos outros processos.

Verifica-se que o texto de características funcionais deve ser escrito com todas as frases utilizando o mesmo sujeito, como introduzido na seção 2.1; neste exemplo, é escrito representando a ação do aluno que realiza a pré-matrícula. Os diagramas são gerados mantendo este ponto de vista: o DFD representa os fluxos de dados entre os processos executados pelo próprio aluno.

Para a modelagem de sistemas de informação, deve-se utilizar as frases cujos sujeitos representam um agente interno ao sistema que se deseja automatizar (como "operador" ou "funcionário"); assim, os diagramas modelarão os objetos e processos que se deseja automatizar, junto com processos que continuarão sendo realizados manualmente (que o "funcionário", por exemplo, continuará realizando).

O exemplo foi escolhido para facilitar a visualização e compreensão dos resultados; para se automatizar o processo de matrícula de uma universidade utilizando a ferramenta desenvolvida, deveria ser utilizado um texto em LN restrita tendo o "funcionário" como sujeito das frases. Assim, o modelo seria gerado sob este ponto de vista.

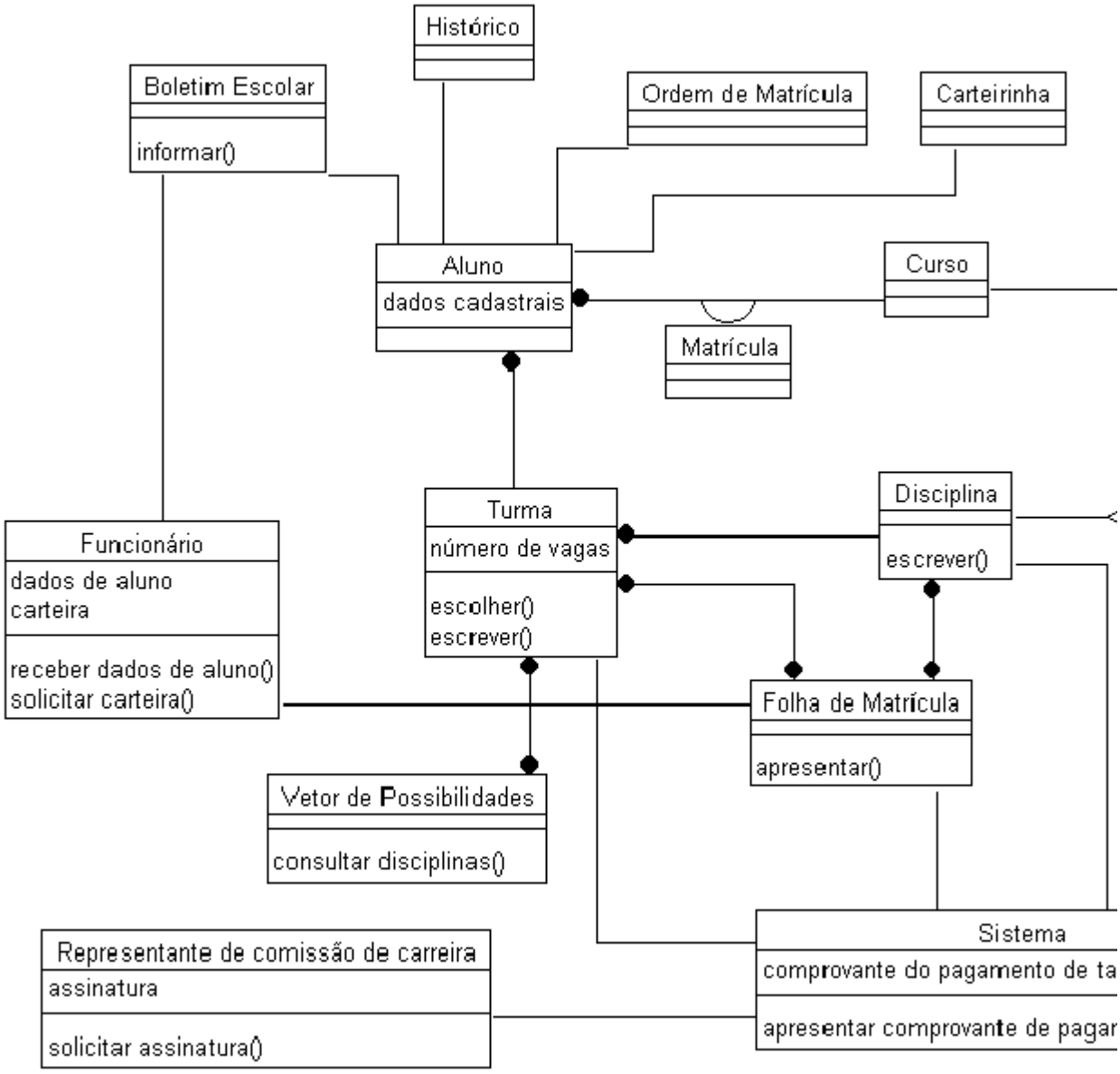

Figura 6: Ilustração do Diagrama de Objetos (DO) gerado

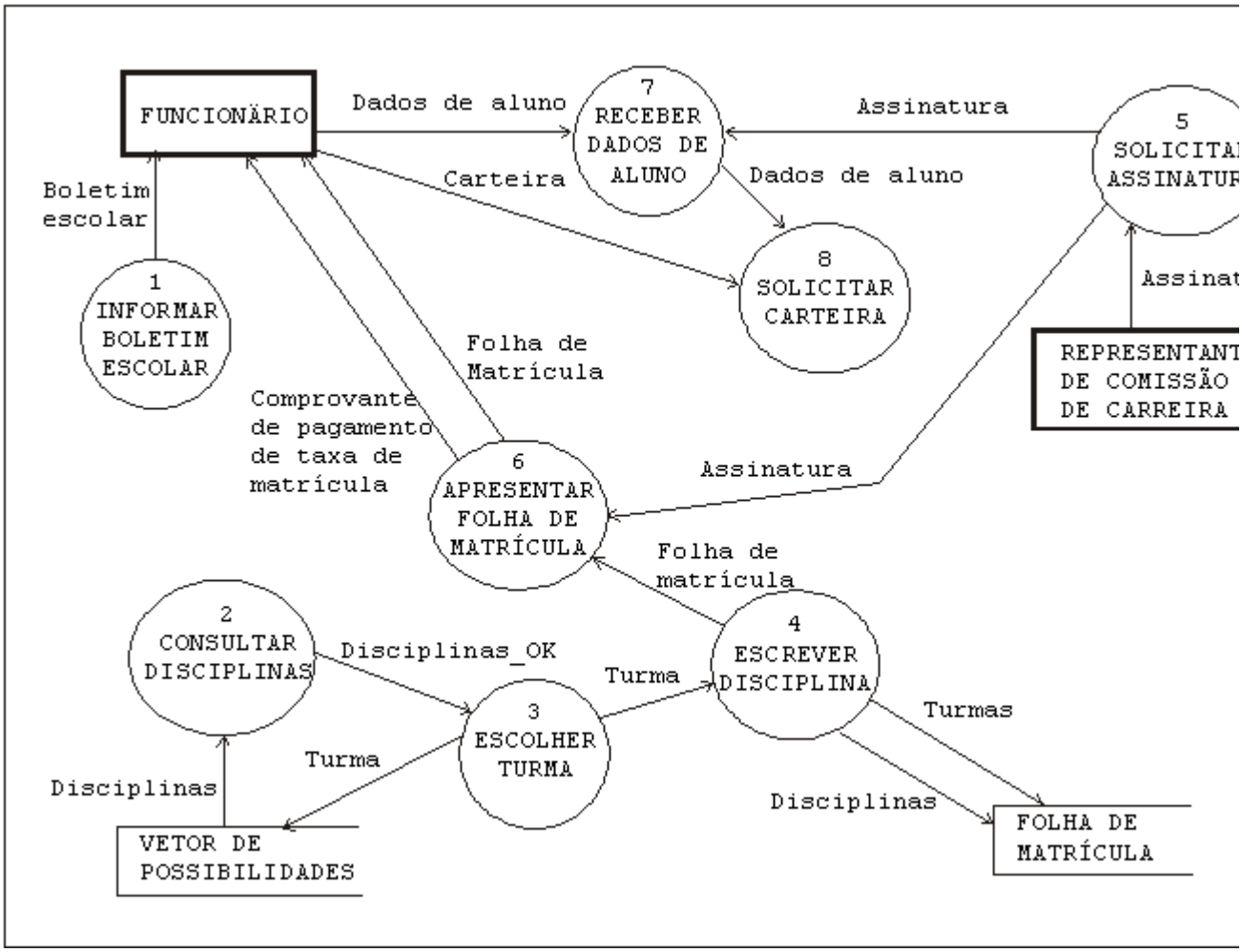

Figura 7: Ilustração do Diagrama de Fluxos de Dados (DFD) gerado

---

Voltar para o topo

4 Conclusões

Este artigo apresentou uma ferramenta para a automatização do processo de modelagem a partir da identificação de requisitos, a qual foi construída a partir da implementação de conhecimento heurístico. Este conhecimento é representado por regras heurísticas, as quais são coletadas e representadas manualmente de maneira simples, contribuindo para automatizar em parte tal processo.

A ferramenta mostra-se bastante prática na geração dos diagramas, ao menos nos casos de teste, que foram bastante trabalhados. Nesses casos, que foram obtidos em manuais e livros técnicos, constataram-se resultados bastante próximos daqueles obtidos através da construção manual dos diagramas com a utilização das heurísticas. Seu processamento automático, entretanto, é muito mais rápido e menos custoso: o processo manual de aplicação das regras heurísticas é trabalhoso, pois além de se aplicar as heurísticas uma a uma, é necessário que se procure e corrija os erros cometidos em cada etapa.

O mecanismo de auto-inserção de palavras facilita o processo de entrada de novos textos. Porém, nem todas as informações do Dicionário podem ser adicionadas automaticamente com este mecanismo, o que dificultou a entrada de alguns casos de teste; quando se entra um texto novo, precisa-se verificar se todas as informações necessárias ao seu processamento já estão no Dicionário. Em contrapartida, constatou-se que alguns textos praticamente não precisaram de inserção de palavras, pois utilizavam um vocabulário bastante semelhante a outros casos já anteriormente testados.

A perda de informações existente na transformação da LN para a LN restrita de entrada ainda não foi estudada suficientemente. No momento, pode-se dizer que as perdas estão controladas para o estudo em questão.

Os diagramas de objeto gerados pela ferramenta estão em processo de validação junto a especialistas, mas pode-se ao menos observar que estes mantém a coerência com os respectivos DFD's. Novas heurísticas podem ser adicionadas ainda, ou mesmo novas construções frasais podem ser permitidas no texto, para aumentar o grau de precisão na atribuição de cardinalidades às associações, bem como na definição do tipo destas (associação *versus* agregação, associação *versus* herança, etc.). O problema mais crítico no processo é a identificação de uma mesma associação entre duas classes que aparece em mais de uma frase no texto, pois podem existir várias associações estre estas classes, mesmo que a cardinalidade das duas seja igual; o resultado gerado pode ser errôneo, com a junção de associações diferentes. Para aperfeiçoar estes resultados, pode ser feita uma avaliação semântica mais complexa dos verbos.

No caso do DFD, foram processados automaticamente pela ferramenta vários casos já testados com pessoas leigas, que não conheciam DFD, mas que utilizaram a base de regras heurísticas; o resultado obtido foi praticamente o mesmo. Os diagramas gerados foram analisados por especialistas e considerados de boa qualidade. Em alguns casos, apareceram pequenas falhas nos diagramas, como um dado que chega a um processo, não é passado para o próximo pois o programa não detecta esta necessidade (provavelmente por que já havia um fluxo entre estes dois processos), e continua seu fluxo a partir do processo seguinte. Contudo, foi constatado ser esta uma falha na construção das regras heurísticas, e não na implementação do sistema.

Avalia-se, ainda, que os resultados gerados pela ferramenta ou obtidos com a aplicação manual das regras não tem a mesma precisão de diagramas construídos por especialistas ao analisarem o mesmo texto, já restrito sintaticamente. Contudo, há a possibilidade de se aperfeiçoar as bases de conhecimento até que se obtenha a qualidade desejada. A vantagem é poder obter um esboço inicial dos diagramas, diminuindo assim o trabalho do especialista (o analista de sistemas).

A ferramenta apresentada foi desenvolvida especificamente para a etapa de especificação diagramática, a partir de requisitos já previamente coletados, organizados e transcritos em LN restrita. Há a necessidade de integração com outras ferramentas de Engenharia de Requisitos, atuantes nas etapas anteriores do processo. Por exemplo, em [8] é apresentada uma ferramenta automatizada que entrevista usuários e documenta as informações fornecidas, gerando textos em LN como resultado. No momento, estes textos ainda devem ser transformados para a LN restrita por um especialista humano, para que possam ser utilizados como entrada para a ferramenta aqui proposta. Este processo de organização dos requisitos, no entanto, pode ser automatizado com a ajuda de ferramentas para o tratamento de LN, como a apresentada em [7]. Em [3] é apresentada uma visão integrada da Engenharia de Requisitos assistida por ferramentas automatizadas.

Outra possibilidade é a utilização da ferramenta desenvolvida em integração com uma ferramenta CASE de projeto orientado a objetos, que permita a edição e conseqüente efinamento dos diagramas gerados. Nesse caso, esta não seria utilizada com o objetivo específico de substituir o especialista (o analista de sistemas), mas sim de auxiliá-lo.

Como trabalho futuro, considera-se a validação dos experimentos realizados com o diagrama de objetos, além da realização de trabalho semelhante com o diagrama de estados. É considerado, também, o desenvolvimento de *parafraseadores* baseados em conhecimento (ferramentas que fazem o processo inverso ao apresentado: geram textos em LN a partir de diagramas, a fim de validar os requisitos junto ao usuário).

Voltar para o topo

5 Referências Bibliográficas

[1] DAVIS, Alan M.; HSIA, Pei. *Giving voice to requirements engineering*. IEEE Software, v. 11, n. 2, Março 1994.

[2] FRASER, Martin D. at al. *Informal and Formal requirements specification languages: bridging the gap*. IEEE Transactions on Software Engineering, v. 17, n. 5, Maio 1991.

[3] CASTILHO, J. M. V.; LOH, S.; TRINDADE, C. S. *Uma Abordagem Baseada em Regras Heurísticas para Construção de Especificações Diagramáticas de Requisitos*. **Anais.** Workshop Ibero-americano de Engenharia de Requisitos e Ambientes de Software - IDEAS 98. Torres-RS, 1-3 de Abril de 1998.

[4] MARTIN, J.; McCLURE, C. *Técnicas Estruturadas e CASE*. Prentice-Hall, 1991.

[5] RUMBAUGH, J., et al. *Object-Oriented Modelling and Design.* Englewood Cliffs, NJ.: Prentice-Hall, 1991.

[6] 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)

[7] BIGOLIN, N. M.; CASTILHO, J. M. V. *Um estudo sobre ferramentas de apoio para o projeto de sistemas de banco de dados.* **Anais.** VIII Simpósio Brasileiro de Engenharia de Software. Curitiba, 25-28 de Outubro de 1994.

[8] 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.

Voltar para o topo