Regras Heurísticas para Transformação Automática de Textos em Diagramas de Fluxo de Dados

Carolina Sturm Trindade*
Universidade Luterana do Brasil

Rodrigo Leal de Moraes*
Universidade Luterana do Brasil

Stanley Loh**
Universidade Católica de Pelotas
Universidade Luterana do Brasil

Endereço
Av. Cel. Lucas de Oliverira, 1365/301
Porto Alegre - 90440-011
RS - Brasil
E-mail: loh@atlas.ucpel.tche.br

RESUMO

Este trabalho apresenta um conjunto de Regras Heurísticas para extrair Diagramas de Fluxo de Dados (DFD's) a partir de textos escritos em linguagem natural (língua portuguesa). A necessidade de ferramentas, manuais e automatizadas, para assistir o analista de sistemas na tarefa de criar DFD's, durante a especificação dos requisitos, foi a principal motivação para o desenvolvimento deste trabalho. A metodologia usada para a definição da regras heurísticas baseou-se na análise de estudos de casos realizados pelos autores. Éapresentado um estudo de caso para avaliação das regras definidas e dos DFD's criados a partir delas.

Palavras chave: análise de requisitos, comunicação usuário-analista, desenvolvimento de sistemas, formalização, heurísticas.

ABSTRACT

This work presents a set of heuristic rules to extract Data Flow Diagrams (DFD's) from texts written in natural language (portuguese). The main motivation for this work was the need for tools to support the System Analyst at the DFD's creation, during requirements specification. The heuristic rules was defined through case studies. A case is presented to evaluation of the rules and the resulting DFD's.

Keywords: heuristics, informal-formal transformation, systems development, requirements analysis, user-analyst communication.

* Mestrando na UFGRS
** Professor da ULBRA e UCPEL

Apoio parcial: ULBRA, UCPEL, FAPERGS

1 INTRODUÇÃO

O processo de Formalização, atividade realizada pelo analista durante a Fase de Especificação no ciclo de desenvolvimento do *software*, é o processo que transforma as informações contidas no Documento de Requisitos em informações expressas no Documento de Especificação [LOH91]. O Documento de Requisitos é aquele que descreve informalmente quais informações deverão ser introduzidas e geradas pelo sistema e, também, aquelas que identificam os usuários da Organização. Já o Documento de Especificação descreve formalmente as funções e as informações que serão desempenhadas e manipuladas pelo sistema.

Em geral, os usuários usam as linguagens informais, por exemplo a Linguagem Natural (LN), porque essas são um modo familiar de comunicação. Por exigir menos treino, o modelo informal facilita o processo de obtenção de requisitos, tornando-se mais indicado para a definição e representação dos requisitos do sistema. Já os analistas utilizam as linguagens formais porque essas são mais apropriadas aos métodos computacionais. A linguagem formal, na maioria das vezes, utiliza uma notação matemática, o que proporciona informação escrita de modo explícito e preciso. Dessa maneira, o uso de uma linguagem formal apresenta algumas vantagens, como por exemplo: clareza, não-ambigüidade, independência de implementação e completude da especificação [CAS95].

Entretanto, embora as linguagens informais favoreçam a criação de documentos mais concisos, e ainda, sejam indicadas para a criação do Documento de Requisitos, elas apresentam alguns problemas, tais como: exigem maior entendimento do contexto, favorecem o aparecimento de redundâncias e contradições no Documento de Especificação. Esses fatores, por exemplo, podem contribuir para que uma determinada característica deixe de ser especificada, refletindo, conseqüentemente, no projeto e/ou na implementação de um sistema [CAS95]. As linguagens formais, por sua vez, também apresentam alguns inconvenientes, promovendo uma certa resistência à especificação formal [MIR91]. O primeiro problema relaciona-se ao fato de que a atividade de escrever uma especificação formal requer um conhecimento intensivo, exigindo um período de aprendizado maior. Isso torna a tarefa cansativa e, conseqüentemente, mais propensa a erros. Assim, citam-se como principais problemas do processo de formalização: a distância entre as Linguagens Informal e Formal, a dificuldade de interpretar as Linguagens Naturais e a falta de ferramentas para auxiliar nesse processo de transformação.

Visando minimizar os problemas encontrados no processo de formalização, diversas técnicas e mecanismos são pesquisados e desenvolvidos. [FRA91], por exemplo, argumenta que a criação de métodos e de mecanismos de transformação da especificação informal para formal tornaria o processo de desenvolvimento e de verificação de especificação formal transparente ao usuário leigo, facilitando o *"feedback"* para o analista. Também é referenciado o uso de ferramentas automatizadas para a obtenção de dados na fase de análise [LOH94] e

para a Derivação de Especificação Formal [BAL78], [FRA91], [MIR91], [BIG94] e [AMO95].

O protótipo desenvolvido por [BAL78] produz especificação formal (saída) a partir de especificação informal (entrada) baseando-se na descrição e no contexto em que a especificação ocorre. Já [MIR 91] desenvolveu uma ferramenta interativa - *SPECIFIER* - que deriva especificações formais de tipos de dados e programas a partir de descrições informais através de técnicas de resolução de problemas comuns fazendo uso de tabelas e analogias. [FRA 91], por sua vez, propõe mapear automaticamente especificações VDM a partir de um conjunto de especificações em DFD. [AMO95] utiliza uma estrutura categorizada e de refinamento evolutivo de requisitos informais para requisitos funcionais, metas, demandas de implementação, teoremas e axiomas que são declarados e orientados para uma especificação formal.

Finalmente, cita-se a ferramenta de [BIG94] - Assistente Projetista- que tem a finalidade de transformar os requisitos expressos em Linguagem Natural (língua portuguesa) diretamente para um modelo conceitual, no caso o modelo Entidade-Relacionamento (ER). O Assistente Projetista evoluiu a partir da criação do "Protótipo Padrão" e do "Protótipo para Formalização". O "Protótipo Padrão" tem a função de transformar os requisitos expressos em LN em um conjunto de frases Padrão, tornando a fase de projeto conceitual mais simples e precisa, pois a solução do problema fica dividida em duas etapas. Num primeiro momento, a LN é transformada, através da utilização de heurísticas, para linguagem Padrão. Numa segunda etapa, a linguagem Padrão é convertida para linguagem Formal. O "Protótipo para Formalização" transforma requisitos expressos em LN para uma linguagem Formal, no caso, lógica de primeira ordem. Dessa forma, o "Assistente Projetista" é a evolução do Protótipo Padrão e do Protótipo para Formalização, visto que transforma requisitos expressos em LN diretamente para um modelo conceitual (modelo ER), através de técnicas sofisticadas e interfaces gráficas para entrada e saída de requisitos.

A disponibilidade de ferramentas automatizadas para auxiliar a criação de desenhos de Diagramas de *software* também é discutida, uma vez que acarreta alguns benefícios, como por exemplo, a economia de tempo despendido entre a produção manual e a produção automatizada [PRO 91]. Para o problema da criação automática de um *layout* para uma ferramenta de desenvolvimento de *software*, no caso o DFD, a partir de linguagem de especificação de requisitos, é proposto um critério para desenho de DFD baseado na descrição de algoritmos de localização e direcionamento.

A literatura carece de materiais e de técnicas que objetivam orientar e apoiar os analistas na tarefa de elaborar DFD's. Ao criar um DFD, durante a fase de modelagem (segunda fase no ciclo de desenvolvimento de *software*), o analista emprega habitualmente regras que estão fundamentadas em experiências e intuições próprias. Em função do exposto, visando contribuir ao desenvolvimento de novos métodos para o processo de formalização durante o

desenvolvimento de *software*, pretende-se definir um conjunto de regras heurísticas para extrair Diagramas de Fluxo de Dados (DFD's) a partir de textos escritos em língua portuguesa.

A escolha por regras heurísticas para auxiliar a formalização fundamenta-se no fato de que elas são adequadas para guiar o desenvolvimento de atividades as quais se valem da experiência pessoal, da observação e da inteligência humana. Ao contrário dos algoritmos, adequados para tarefas estruturadas e programáveis. Como criação de DFD's é uma atividade não estruturada optou-se pelo uso de heurísticas, ou seja, regras do pensamento que reduzem o trabalho requerido para se obter uma solução normalmente satisfatória para um determinado problema. Finalmente, existem outras pesquisas, como [FRA91], [BIG94] e [AMO95] que também sugerem o uso da heurística para auxiliar no processo de formalização.

2 METODOLOGIA

O método utilizado para a definição das heurísticas foi a análise de casos e o reconhecimento de padrões. Para utilizar as heurísticas faz-se necessário ter como entrada textos escritos Linguagem Natural restrita (um subconjunto do português). O produto (saída) das aplicações das heurísticas é o Diagrama de Fluxo de Dados.

Para realizar os estudos de casos criaram-se vários DFD's a partir de textos. Observou-se o procedimento mental envolvido neste processo, a fim de identificar um conjunto de padrões e transcrevê-los para um conjunto inicial de heurísticas. Depois, realizaram-se testes de avaliação e de complementação das heurísticas.

Os textos utilizados como entrada devem descrever rotinas de trabalho na seguinte forma: 1) o texto deve ser formado por frase simples, apresentando como formato o verbo e complemento verbal; 2) cada frase deve ser composta por um único verbo principal que deverá estar conjugado na terceira pessoa do singular; 3) cada frase deve ter, obrigatoriamente, pelo menos um objeto direto; 4) a presença de objeto indireto irá depender do tipo de verbo presente na frase; 5) não deve-se utilizar pronome, devendo o mesmo ser substituído pelo seu correspondente cada vez que aparecer na frase; 6) na descrição de uma rotina não deverão aparecer os procedimentos realizados por outras pessoas.

3 REGRAS HEURÍSTICAS

Seguindo sugestões do método heurístico para quebrar problemas complexos em subproblemas, as heurísticas definidas foram divididas em três grupos: heurísticas gerais, heurísticas específicas para grupo de verbos e heurísticas de finalização. A relação estabelecida entre estes grupos é a seguinte: deve-se primeiramente utilizar as heurísticas gerais, depois as heurísticas específicas para grupo de verbos e finalmente as heurísticas de finalização. Para a utilização das heurísticas também é necessário que o usuário tenha como pré-requisito um conhecimento relativo a Objeto Direto (OD) e Objeto Indireto (OI).

3.1 Heurísticas Gerais

Analisar cada frase separadamente, identificar o tipo do verbo e enquadrá-lo em um dos grupos sugeridos: Entrada de dados, Saída de dados, Leitura de dados, Modificação dos dados, Cálculo, Grupo Especial e Outros. Após determinar-se o grupo do verbo, este deverá usar as heurísticas relativas ao seu grupo. Então, tem-se que:

- no grupo relativo à ***Entrada de dados*** encontram-se os verbos que indicam uma entrada de dados para o sistema. Exemplos: "pedir", "perguntar", "receber", "requisitar" e "solicitar";
- no grupo relativo à ***Saída de dados*** encontram-se os verbos que indicam uma saída de dados do sistema para a entidade externa. Exemplos: "avisar", "comunicar", "devolver", "emitir", "encaminhar", "entregar", "enviar", "fornecer", "informar"e "confirmar";
- no grupo relativo à ***Leitura de dados*** encontram-se os verbos que indicam uma leitura feita sobre os dados (agrupados em algum lugar do sistema). Exemplos: "analisar", "buscar", "conferir", "consultar", "controlar", "decidir", "procurar", "relacionar", "reunir" e "verificar";
- no grupo relativo à ***Modificação dos dados*** encontram-se os verbos que indicam uma alteração realizada sobre os dados do sistema. Exemplos: "arquivar", "atualizar", "buscar", "cadastrar", "desarquivar", "guardar", "registrar", "reservar", "acrescentar", "adicionar", "anexar", "anotar", "assinar", "datilografar", "escrever", "elaborar", "extrair", "listar", "preencher", "relacionar", "rubricar" e "separar";
- no grupo relativo ao ***Cálculo*** encontram-se os verbos que indicam uma operação de cálculo sobre os dados envolvidos no DFD. Exemplos: "adicionar", "calcular", "extrair", "somar", "subtrair", "multiplicar", "dividir" e "totalizar";
- no ***Grupo Especial***, destacam-se os verbos fazer, proceder, efetuar e providenciar. Estes verbos necessitam de um complemento, uma vez que sozinhos não informam o tipo de operação que deve ser realizada. Exemplos: fazer as anotações equivale a "anotar"; fazer os registros equivale a "registrar", proceder a conferência equivale a "conferir" e fazer a relação equivale a "listar";
- no grupo dos ***Outros*** encontram-se todos aqueles verbos que não se enquadram em nenhum dos grupos anteriores

Se aparecer algum verbo diferente dos anteriormente citados, então deve-se verificar o significado do verbo e procurar substituir por algum sinônimo. Se este verbo pode ser substituído por algum dos verbos sugeridos, então o verbo considerado diferente terá tratamento idêntico aos demais verbos do grupo ao qual deve pertencer. Caso contrário, vai para o grupo dos Outros.

As demais heurísticas gerais visam a criação e a numeração de processos, definição de fluxos, criação de depósitos e de entidades externas.

- **Heurística para criação e numeração do processo:** o verbo e OD da frase darão nome ao processo. Então, para a frase *"Solicita o código do livro ao aluno"*, o processo

criado será denominado **"Solicita código do livro"**. Os processos também são numerados à medida em que são criados. Para cada processo há um número único. Assim, o primeiro processo a ser criado recebe o número 1, o segundo processo recebe o número 2 e assim sucessivamente.

- **Heurística para criar Fluxo de Dados:** o OD da frase sempre dará origem a um fluxo de dados. A orientação deste fluxo, entretanto, somente será determinada após o uso das heurísticas específicas para grupo de verbos. Assim, para a frase *"Solicita o código do livro ao aluno"*, o fluxo de dados criados será denominado **"código do livro"**.
- **Heurística para criar Entidade Externa:** uma entidade externa será criada sempre que o OI acompanhar um verbo pertencente ao grupo de Entrada de dados ou ao grupo de Saída de dados. Então, para a frase *"Solicita o código do livro ao aluno"*, será criado uma entidade externa chamada de **"aluno"**, uma vez que o verbo *"solicita"* pertence ao grupo entrada de dados e *"aluno"* é o OI da frase.
- **Heurística para criar depósitos:** uma das maneiras estabelecidas para a criação de depósitos é associar o OI aos verbos pertencentes aos grupos de Leitura de dados ou Modificação de dados. Além disso, também deve-se observar, através do Dicionário de Termos que é criado, se o OI já fora usado em frases anteriores.
- **Heurística para criar Dicionário de Termos:** todos os fluxos, depósitos e entidades externas criados são inseridos no Dicionário de Termos que, por sua vez, será útil para a elaboração do Dicionário de Dados que, certamente, deverá ser criado pelo analista.

3.2 Heurísticas Específicas para Grupo de Verbos

Essas heurísticas fundamentam-se no princípio que as operações estão baseadas em cinco grupos de ações, conforme sugerido por [LOH91]: entrada de dados, saída de dados, modificação de dados, leitura de dados e cálculo. Cada grupo possui características próprias, conforme o próprio nome sugere. Entretanto, salienta-se que nos grupos de entrada de dados e de saída de dados obtém-se os elementos que fazem a interface entre o sistema e o mundo exterior. Os demais grupos caracterizam-se em manipular os dados de entrada e criar os dados de saída. Assim essas heurísticas definem o sentido do fluxos dos dados e a criação ou não de um depósito ou de um fluxo de dados a partir do OI. Então, para a frase *"Solicita o código do livro ao aluno"* tem-se: 1) o fluxo de dados, *código do livro,* criado pela heurísticas gerais, será orientado para dentro do processo; 2) o sentido do fluxo será da entidade externa *"aluno"*, também criada pelas heurísticas gerais, para o processo *"solicita código do livro"*.

3.3 Heurísticas de Finalização

Objetivam fazer a ligação entre os processos do DFD, cuidando para que não se tenham os chamados "buracos negros" ou "processos de geração expontânea". Os processos, ao serem analisados nessas heurísticas, devem ainda obedecer uma seqüência: primeiramente todos os processos de entrada, após todos os processos de leitura, todos os processos de saída, todos

os processos de modificação e por fim todos os processos de cálculo. Obrigatoriamente nem todos os grupos de processos estarão presentes na construção do DFD. No entanto, é fundamental manter esta ordem seqüencial estipulada, bem como a ordem numérica estabelecida nas heurísticas gerais.

4 ESTUDO DE CASO

O texto a seguir descreve a rotina *"Compra produtos das fábricas"*, que servirá de entrada para a utilização das heurísticas:

"- 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 fornecedoras.*
- *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 da entrega.*
- *Entrega a ficha para o funcionário da contabilidade."*

O resultado da utilização das heurísticas é o DFD da Figura 1.

4.1 Heurísticas utilizadas:

Utilizaram-se primeiro as heurísticas gerais, depois as específicas para grupos de verbos e por último as de finalização.

- Cada frase foi analisada separadamente. O verbo de cada frase juntamente com o seu OD dão origem ao nome do processo. Os processos são numerados de acordo com a ordem em que aparecem as frases.
- Todos os depósitos, entidades externas e fluxos criados pela primeira vez foram inseridos no dicionário de termos.
- Todos os OD originaram um fluxo de dados.
- Para os verbos enquadrados no grupo de entrada e saída de dados tem-se um OI que dá origem a uma entidade externa.
- Para os grupos de leitura e modificação de dados, o OI dá origem a um depósito, no caso de aparecer pela primeira vez, ou o OI dá origem a um fluxo de dados, no caso de já existir no dicionário de termo como sendo um fluxo de dados com o nome igual ao OI.
- Para o grupo de cálculo, o OI dá origem o um fluxo de dados.
- Para evitar os "buracos negros" e os "processos de geração expontânea" faz-se:
  - Um processo "P_n" que possue fluxo de saída sem destino definido terá este fluxo direcionado para um destes casos: a) um processo "P_{n+x}", isto é, para um processo de numeração superiro ao processo "P"; b) um depósito; c) uma entidade externa.
  - Um processo "P_n" que possue um fluxo de entrada sem origem definida terá este fluxo ligado a um destes casos: a) um processo "P_{n-x}" de numeração inferior, isto é, para um processo criado anteriormente ao processo que está sendo analisado b) um depósito; c) uma entidade externa.
- Frase número 1: o OI deu origem a o depósito de dados "Ficha do produto"; o verbo foi enquadrado no grupo de leitura de dados, por isso : a) o fluxo com nome igual ao OD tem sentido do depósito para o processo, b)criou-se um fluxo orientado para fora do processo e com o nome igual ao OD seguido da palavra "OK".
- Frase número dois: o verbo "anota" pertence ao grupo de saída de dados; como não há OI, não cria-se depósito e direciona-se o fluxo para fora do processo;

- Frase número três: o verbo foi enquadrado no grupo de modificação de dados; o seu fluxo é orientado para o depósito criado a partir do OI.
- Frase número quatro: o verbo "negociar" é tratado como um verbo do grupo de entrada e de saída de dados. Por isso, o fluxo "compra" é direcionado da entidade externa para o processo e e vice-versa.Como o processo de numeração seguinte é de modificação, cria-se um outro fluxo de saída com sentido para o processo número 5.
- Frase número cinco: o OD "dados da compra" deve ser direcionado para fora do processo, em direção OI , isto é, ao depósito "ficha", .
- Frase número seis: o OD "ficha" deve ser direcionado para a entidade externa "funcionário da contabilidade". Um fluxo denominado "ficha" também entra para este processo porque: "ficha" é um fluxo criado por processo de saída, logo deve-se procurar um fluxo ou depósito com este nome nos processo de numeração anterior e direcioná-lo para este processo de saída (um depósito "ficha" foi criado pelo processo de número 5).

O Dicionário de Termos resultante foi o seguinte:

{ ...
quantidade de produtos estocados - qtd.pro.est.- (fluxo)
ficha do produto (depósito)
nomes dos produtos com baixo estoque - nome prod. baixo est.- (fluxo)
nome fornecedores (fluxo)
dados fornecedores (fluxo)
fichário de fornecedores (depósito)
compra (fluxo)
fábricas fornecedoras (EE)
dados *compra* = nome dos produtos encomendados, quantidade de cada produto, valor unitário de cada produto, valor total da compra, data do pedido e data da entrega.
ficha (depósito)
funcionário contabilidade (EE) ...}

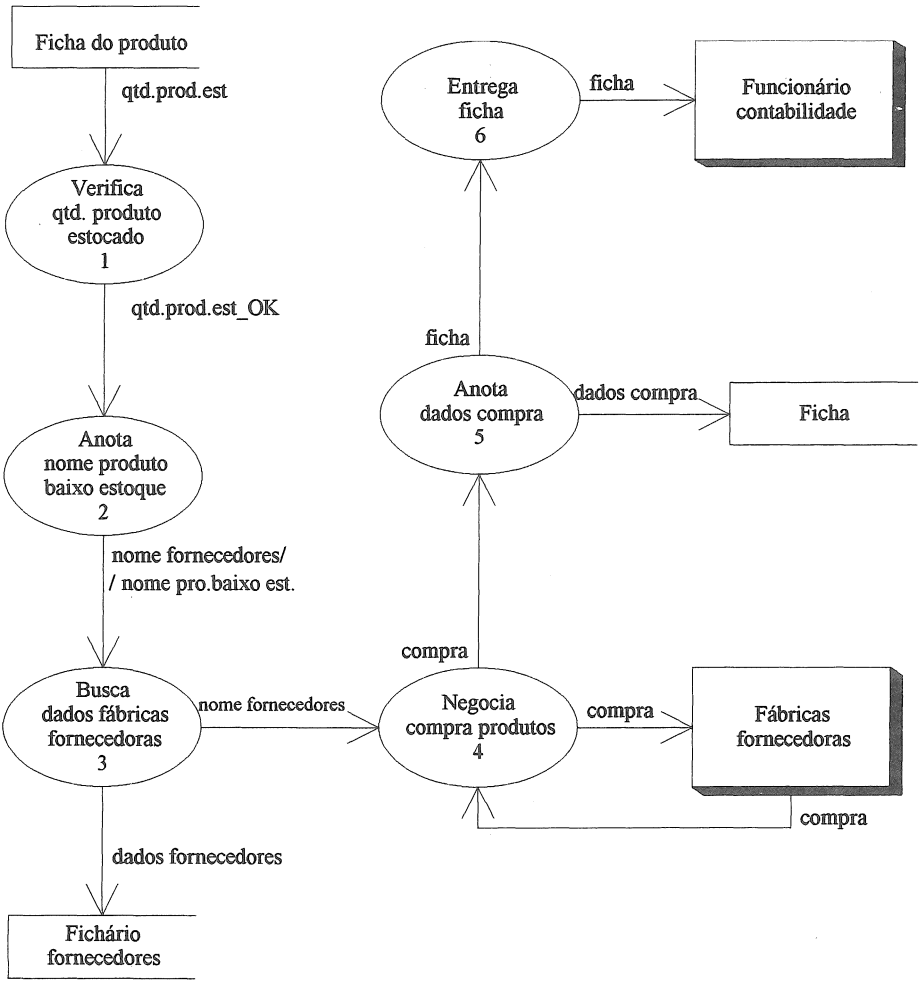

Figura 1 DFD resultante para o Estudo de Caso "Compra produtos das fábricas"

5 CONCLUSÃO

Desenvolveu-se um conjunto de regras heurísticas para assistir o desenvolvimento de DFD's a partir de textos escritos em língua portuguesa, com a finalidade de automatizar tal processo. As heurísticas apresentadas foram divididas em três grupos: Heurísticas Gerais, Heurísticas Específicas para Grupo de Verbos e Heurísticas de Finalização. Entretanto, ressalta-se que não espera-se que os DFD's obtidos mediante a utilização das heurísticas aqui propostas sejam totalmente corretos, livres de alterações e nem que modelem corretamente todas as funções que o sistema deverá executar e todas as interações entre elas. Isto até seria praticamente impossível, uma vez que um DFD estará finalizado quando for aceitável pelo

usuário [YOU 92]. Ressalta-se que as regras heurísticas não substituem a presença do Analista de Sistemas e não garantem a solução para todos os casos. Contudo, isto não invalida as heurísticas criadas pois, conforme [DEM 89], é normal que os primeiros esboços de DFD's não sejam perfeitos. Pelo contrário, o autor alega que o primeiro desenho é importante para auxiliar na verificação das falhas mais "grotescas", sendo normal os sucessivos refinamentos para que se alcance uma idéia perfeita e completa, uma vez que *"idéias perfeitas não brotam, desenvolvem-se"*. Por isso, não pretende-se chegar a respostas definitivas, mas sim, fazer das heurísticas uma ferramenta capaz de auxiliar na tarefa de obtenção de DFD a partir de textos escritos em português.

O método usado para a obtenção das heurísticas fundamentou-se em estudos de casos realizados pelos autores. Procurou-se criar DFD's corretos, evitando buracos negros e processos de geração expontânea e buscando a criação e a nomeação de processos, fluxos de dados, depósitos e entidades externas. Entretanto, outro procedimento considerado também importante para a análise da correção do DFD não foi determinante para a definição das heurísticas: o particionamento de DFD em níveis e, conseqüentemente, a consistência entre níveis. Assim, não foram criadas heurísticas específicas visando o particionamento dos DFD'. Porém, constatou-se que, por exemplo, um processo que fosse "explodir" geraria uma série de novos procedimentos. Isso exigiria, então, a criação de novos textos para gerar um novo DFD. Restaria, assim, verificar o balanceamento entre o processo que "explodiu" e os seus processos "filhos".

Uma característica que deve ser levada em consideração no desenvolvimento dos DFD's é a de não torná-los demasiadamente complexos. Como este aspecto está relacionado ao particionamento do DFD, mecanismos de balanceamento para verificar a consistência dos dados não foram elaborados. Heurísticas específicas para esses casos não foram definidas. O cuidado, neste sentido, direcionou-se para o trabalho de textos não muito extensos, uma vez que o número de frases está relacionado com o número de processos criados.

Os experimentos foram realizados com pessoas que possuiam algum conhecimento de DFD e com outras que não tinham conhecimento de DFD ou que nunca haviam criado DFD. Esses indivíduos recebiam as heurísticas e os textos e realizavam sozinhos a construção dos DFD's. Constatou-se, então, que a criação dos DFD's foi conforme o esperado, isto é, a maioria das pessoas entendia o que estava escrito, não aplicando erroneamente as heurísticas. Além disso, com base nos testes realizados, constatou-se que as heurísticas mostraram-se bastante esclarecedoras para algumas pessoas que ainda tinham dúvidas quanto à criação de DFD's. Deste modo, crê-se que as heurísticas definidas podem ser proveitosas no treinamento de analistas na tarefa de criar DFD a partir de textos.

Sob o aspecto de produtividade obtiveram-se a algumas conclusões, fundamentadas na análise dos experimentos e dos pareceres obtidos junto aos grupos que testaram as heurísticas. Foi constatado de que o material descritivo ocasionava algum "embaraço" no momento da

construção dos DFD's para aqueles indivíduos que não estavam habituados com as regras heurísticas criadas. Para amenizar esse problema, criou-se um esquema gráfico. Ao adotá-lo, observou-se que a produtividade apresentou melhora, visto que, as pessoas já não "se perdiam" quando necessitavam manusear o material.

Embora as heurísticas pareçam ser claras e capazes de gerar um esboço de DFD razoável para os casos testados, constata-se que elas ainda são restritas. Os textos dos quais as heurísticas foram extraídas são simples: não apresentam todas as expressões possíveis permitidas pela língua portuguesa. Também não foram realizados testes baseados em métricas estabelecidas por estudiosos para avaliar a qualidade dos DFD's. Entretanto, os DFD's resultantes foram avaliados por pessoas experientes que os julgaram de boa qualidade.

Finalmente, crê-se que o trabalho desenvolvido será útil para os estudos de construção de ferramentas que auxiliarão o analista na atividade de extrair DFD a partir de textos. As heurísticas também poderão ter grande contribuição para o desenvolvimento de trabalhos envolvidos na construção de protótipos de sistemas capazes de transformar automaticamente textos em DFD's. Nesse caso, a automatização do processo de formalização dos requisitos seguiria um rumo semelhante ao desenvolvido por [BIG 94], envolvendo o processamento de linguagem natural. Nesse trabalho, [BIG 94] desenvolveu uma ferramenta que traduz os requisitos expressos em linguagem natural, coletados na fase de análise de requisitos, para criar especificações segundo o modelo ER. Logo, a partir de estudos de tratamento da linguagem natural, poderia-se construir ferramentas que tivessem como característica a entrada de requisitos expressos em língua portuguesa e que, a partir desses requisitos, produzissem automaticamente DFD's como saída. Assim, a automatização da transformação dos requisitos seria facilitada e a validação dos requisitos seria feita de modo transparente para o usuário.

6 BIBLIOGRAFIA

[AMO95] AMOROSO, E. G.. Creating Formal Specifications from Informal Requirements Documents. Software Engineering Notes, New York, v. 20, n. 1, p. 67-70, jan. 1995.

[BAL78] BALZER, R. et alii.. Informality in Program Specifications. IEEE Transactions on Software Engineering, New York, v. SE-4, n. 2, mar. 1978.

[BIG94] BIGOLIN, N. M. Aplicação de Técnicas de Tratamento da Linguagem Natural no Apoio ao Projeto de Banco de Dados. Porto Alegre: CPGCC da UFRGS, 1994. 117 p. Dissertação de Mestrado.

[CAS95] CASTRO, J. F. B de. Introdução à Engenharia de Requisitos. Porto Alegre: Instituto de Informática da UFRGS. 1995. 43 p. (Curso apresentado na XIV Jornada de Atualização em Informática no XV Congresso da Sociedade Brasileira de Computação).

[DEM 89] DeMARCO, T.. Análise Estruturada e Especificação de Sistemas. Rio de Janeiro: Campus, 1989. 333 p.

[FRA91] FRASER, M. D.; KUMAR, K.; VAISHNAVI, V. K.. Informal and Formal Requirements Specification Languages: Bridging the Gap. IEEE Transactions on Software Engineering, New York, v. 17, n. 10, p. 1126-1140, oct. 1991.

[LOH91] LOH, S.. Uma Linguagem Comum entre Usuários e Analistas para Definição de Requisitos de Sistemas de Informação. Porto Alegre: CPGCC da UFRGS, 1991. 180 p. Dissertação de Mestrado.

[LOH94] LOH, S.; POETA, C. R.; CASTILHO, J. M. V. de. Automating the Software Requirements Elicitation Process: Proposal and Prototype. 1994. (Relatório Técnico).

[MIR91] MIRIYALA, K.; HARANDI, M. T.. Automatic Derivation of Formal Software Specifications from Informal Descriptions. IEEE Transactions on Software Engineering, New York, v. 17, n. 10, p. 1126-1140, oct. 1991.

[PRO91] PROTSKO, L. B.; SORENSON, P. G.; TREMBLAY, J. P.; SCHAEFER, D. A.. Towardsthe Automatic Generation of Software Diagrams. IEEE Transactions on Software Engineering, New York, v. 17, n.1, p. 10-21, jan. 1991.

[YOU92] YOURDON, E.. Análise Estruturada Moderna. 3.ed. Rio de Janeiro: Campus, 1992. 836 p.