

# O Uso de Hipertexto na Elicitação de Linguagens da Aplicação

Julio Cesar Sampaio do Prado Leite<sup>§</sup>

Ana Paula M. Franco<sup>¶</sup>

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

R. Marquês de S. Vicente, 225.

Rio de Janeiro 22453

5/90

## Resumo

Este artigo tem como objetivo analisar a adequabilidade do uso de hipertexto como suporte para elicitación de linguagens da aplicação. O artigo descreve sucintamente o problema da elicitación de linguagens da aplicação, e a estrutura do Léxico Ampliado da Linguagem da aplicação. Este léxico é o documento para o qual o suporte de hipertexto é usado. A análise da adequabilidade é vista, principalmente, sob o ponto de vista da autoria do hipertexto. A linguagem da biblioteca da PUC é usada como exemplo.

## 1 Introdução

Pesquisas recentes sobre o desenvolvimento de software vêm ressaltando a importância da explicitação do conhecimento nas tarefas do processo de desenvolvimento de software. Diferentes estratégias vêm sendo sugeridas de como abordar o problema. A princípio, podemos destacar as estratégias apoiadas em resultados ou pesquisas em Inteligência Artificial [Lucena 87], e resultados e pesquisas na formalização do processo de desenvolvimento de software<sup>1</sup> [Maibaum 87].

Dentro da estratégia voltada para a Inteligência Artificial, alguns trabalhos têm se apoiado em sistemas de transformação como suporte à operacionalização de mudanças de representação [Smith 85]. Seguindo a linha que emprega sistemas de transformação, a proposta de Neighbors [Neighbors 84] para a construção de software, através da composição de componentes reutilizáveis, é impar. Neighbors, ao propor que cada problema

<sup>§</sup>Apoio do Cnpq e da Faperj

<sup>¶</sup>Apoio do Cnpq

<sup>1</sup>Estas estratégias foram discutidos num workshop sobre o processo de software realizado no Rio de Janeiro em dezembro de 1988 [Leite 89b].

tenha a sua própria linguagem, distancia-se das propostas na área, que advogam linguagens de especificação de uso geral. Na visão de Neighbors, a produção dessas linguagens específicas, chamadas linguagens de domínio, é função de um processo chamado análise de domínios. Este processo seria, basicamente, uma generalização da análise de requisitos.

A análise de domínio, segundo Neighbors, produz os objetos e operações de uma classe de aplicações. Com base nestes objetos e operações, uma gramática generativa é formada e tem sua semântica definida por componentes específicos escritos em uma outra linguagem. Para demonstrar que estes componentes poderiam ser reutilizados ao nível de abstração da análise, Neighbors construiu um protótipo, chamado Draco que, usando uma rede de linguagens de domínios previamente definidos, gera programas executáveis. Essa geração é parcialmente automática. A automatização é conseguida principalmente por um gerador de analisadores sintáticos, por um sistema transformador e por um sistema macro expensor. O sistema macro expensor é o responsável pela anotação das árvores geradas pelo analisador sintático. A anotação dessas árvores é feita com os componentes previamente definidos para a gramática em questão. Ao sistema transformador cabe otimizar árvores geradas pelo analisador sintático.

O aprofundamento dos estudos iniciados por Neighbors resultou no que hoje é conhecido como o paradigma Draco, onde o fator principal é o uso do conhecimento específico como base para o desenvolvimento de software. O estudo do paradigma Draco e seus problemas teóricos e práticos é o principal tema de nosso projeto de pesquisa na PUC-RIO. No paradigma Draco, o desenvolvimento de qualquer software requer que se faça uma análise de domínio. Com base nesta análise uma linguagem é construída, e usada para especificar qualquer problema no domínio em questão. Estas especificações são então transformadas pelo sistema Draco em programas executáveis. Neste artigo lidaremos com um subproblema relacionado com análise de domínios e portanto não nos deteremos nos problemas relacionados à especificação nem aos problemas existentes no esquema transformacional do paradigma Draco.

Um dos problemas que estamos investigando e que será apresentado neste artigo refere-se à *produção de linguagens da aplicação*. A produção de linguagens da aplicação é uma simplificação do problema original da análise de domínios. Linguagens da aplicação, conforme definiu Leite em [Leite 89], são linguagens que têm todas as características das linguagens de domínio, só que tratam apenas de uma ou duas instâncias daquele domínio. Esta estratégia redutora tem por objetivo ganhar mais experiência na aquisição e produção de linguagens orientadas a problemas, antes de generalizar os resultados para linguagens de domínio.

As linguagens já desenvolvidas por nosso grupo de pesquisa tem servido basicamente para explorar possíveis métodos. Além da linguagem da biblioteca, há uma linguagem para a custódia de ações e uma linguagem para agenda de consultórios médicos. Todas estas linguagens foram desenvolvidas até o ponto de se obter uma BNF da linguagem. Também foram produzidas algumas especificações (programas) usando a linguagem conforme descrita na BNF. Somente a linguagem da agenda chegou ao ponto de produzir uma semântica formal, no caso usando a linguagem Pascal.

Nossa investigação sobre a produção de linguagens da aplicação encontra-se no princípio e os primeiros resultados se concentram na parte referente a elicitacão<sup>2</sup> da linguagem.

<sup>2</sup>Elicitar: [Var. eliciar + clarear + extrair.] V. t. d. 1. descobrir, tornar explícito, obter o máximo de

Estamos usando um novo enfoque para elicitación. O enfoque, de orientação sintática e com influência da semiótica, tem como sustentáculo o que chamamos de Léxico Ampliado da Linguagem (LAL). O Léxico Ampliado tem por objetivo descrever precisamente a semântica dos símbolos próprios da linguagem da aplicação. A descrição é feita usando-se um conjunto restrito da linguagem natural.

O presente trabalho não pretende apresentar o ciclo completo do processo de construção de linguagens da aplicação. No que se segue, centramos nossa atenção na fase de construção do LAL e examinamos o conceito de hipertexto aplicado a este processo de construção. Os resultados obtidos até agora reforçam nossa convicção de que os métodos que estamos investigando se coadunam perfeitamente para a produção de linguagens da aplicação. A sedimentação de procedimentos, de heurísticas e de uma representação base é indispensável para que a construção de linguagens da aplicação, e posteriormente, de linguagens de domínio, se façam de maneira eficiente e com suporte de software.

Nossa apresentação é organizada da seguinte maneira: a) introdução ao conceito de linguagens da aplicação e técnicas atualmente utilizadas na elicitación de linguagens (Seção 2), b) apresentação do léxico ampliado da linguagem (Seção 3), c) o exemplo da linguagem da biblioteca da PUC (Seção 4) e d) o uso do hipertexto como ferramenta de apoio (Seção 5). Finalizamos com algumas observações sobre o processo utilizado e um esboço de futuros trabalhos.

## 2 Linguagens da Aplicação

Foi Leite [Leite 89] quem cogitou da necessidade de estudar, de maneira mais detida, as linguagens de aplicação. Ao contrário das propostas predominantes na literatura, desenvolveu a tese de que, em lugar do uso de uma linguagem de especificação geral, linguagens específicas devam ser usadas. Esta postura quanto a especificação em linguagens próprias do problema foi originalmente proposta por Neighbors que chamou o trabalho de levantamento dessa linguagem de **análise de domínios**. A estratégia de pesquisa proposta por Leite procura estudar a análise de domínios centrando-se no aspecto da linguagem do domínio. Esta estratégia, deixando de lado os aspectos de generalização característicos das linguagens de domínios, baseia-se na hipótese de que resultados da pesquisa na produção de linguagens da aplicação poderão ser usados em métodos de análise de domínios.

A proposta de produção de linguagens da aplicação pretende, a longo prazo, delinear métodos e técnicas para suportar a elicitación dessas linguagens, bem como sua posterior formalização. O ponto de partida do trabalho na elicitación de linguagens da aplicação tem como base uma combinação de uma visão antropológica e de uma visão influenciada pela semiótica. Leite acredita que a linguagem é um reflexo, e a maior expressão de um determinada cultura, que por sua vez é determinada por um meio social. Portanto, para uma determinada aplicação existe um meio social com vários atores que, nos seus respectivos papéis, moldam, de certa maneira, uma cultura própria. Neste meio social, no qual a aplicação se realiza, existe uma linguagem própria dos atores. Esta linguagem é usada por atores no dia a dia de seu convívio social e é uma extensão da linguagem natural normalmente utilizada.

informações para o conhecimento do objeto em questão.

As linguagens da aplicação utilizam termos e construções específicos ao meio social. Apesar de expressos em linguagem natural, estes termos tem um significado diferente do significado corrente. Por exemplo, na área de aplicação de custódia de ações, o termo *bater na posição* não significa martelar ou golpear num determinado lugar, mas significa crédito de títulos na conta-mãe. A seguir descrevemos em linhas gerais o método atualmente usado para a elicitacão de linguagens da aplicação.

## 2.1 A Elicitacão de Linguagens da Aplicacão

A idéia central usada na elicitacão de linguagens da aplicacão, é a captura de termos próprios da linguagem em questão. Portanto, o principal objetivo a ser perseguido pelos engenheiros de software nessa tarefa é a identificacão de palavras ou frases peculiares ao meio social da aplicacão sob estudo. Somente após a identificacão dessas frases e palavras é que se procurará o seu significado. A estratégia de elicitacão é ancorada na sintaxe da linguagem e é formada por três fases de coleta de fatos:

- identificacão de símbolos da linguagem,
- identificacão da semântica de cada símbolo,
- identificacão das regras de produçãõ de uma gramática que gere a linguagem da aplicacão.

Para cada uma dessas fases, estamos pesquisando heurísticas que nos auxiliem no cumprimento do objetivo de cada fase, bem como para proceder à validacão e modelagem dos fatos coletados. Convém observar que não é nosso objetivo, neste trabalho, discorrer sobre o *uso* da linguagem elicitada nem sobre sua implementacão. Seguindo a idéia do paradigma Draco, as linguagens da aplicacão passam a ser as linguagens de especificacão dos problemas. Aqui nos detemos numa das fases do processo de construçãõ dessas linguagens. A seguir detalharemos as principais heurísticas utilizadas na elicitacão de linguagens da aplicacão.

### 2.1.1 Identificacão de Símbolos

Através de uma estratégia de abordagem (p.ex. entrevistas, observacão, leitura de documentos) [Leite 90], o engenheiro de software anota as frases ou palavras que *parecem ter um significado especial na aplicacão*. Estas palavras são, em geral, palavras chaves que são usadas com frequência pelos atores da aplicacão. Quando uma palavra ou frase parecer ao engenheiro de software sem sentido, ou fora de contexto, há indícios de que esta palavra ou frase deve ser anotada. O resultado dessa fase é uma lista de palavras e frases.

A grande diferença entre a elicitacão advogada por Leite e a elicitacão comumente praticada por analistas de sistemas é o enfoque. Enquanto na análise de sistemas, as estratégias de abordagem são usadas com o objetivo de elicitare as funções do sistema em estudo e suas saídas e entradas, na elicitacão de linguagens da aplicacão, as estratégias de abordagem são usadas com o objetivo de elicitare símbolos. Na elicitacão de linguagens da aplicacão uma das principais heurísticas é justamente a de *não procurar identificar funções da aplicacão observada, mas apenas os seus símbolos*.

### 2.1.2 Identificação da Semântica

Com base na lista de símbolos o engenheiro de software entrevista atores da aplicação, procurando entender o que cada símbolo significa. Esta fase é a fase na qual o Léxico Ampliado da Linguagem é usado como um sistema de representação. Esta representação requer que para cada símbolo sejam descritos noções e impactos. Noção é o que significa o símbolo, impacto descreve os efeitos do uso/ocorrência do símbolo na aplicação, ou do efeito de algo na aplicação sobre o símbolo. Esses efeitos, muitas vezes, caracterizam restrições impostas ao símbolo ou que o símbolo impõe. A descrição de impactos e noções é orientada pelos princípios de *vocabulário mínimo* e *circularidade*.

O vocabulário mínimo prescreve que ao descrever uma noção ou um impacto, esta descrição deve minimizar o uso de símbolos externos à linguagem, e que quando estes símbolos externos são usados, devem procurar ter uma representação matemática clara (p.ex. conjunto, união, interseção, função). A circularidade prescreve que as noções e os impactos devem ser descritos usando símbolos da própria linguagem. As experiências têm demonstrado que, na explicação da noção e do impacto, os atores da aplicação usam, naturalmente, do princípio de circularidade. A obediência ao vocabulário mínimo é de responsabilidade do engenheiro de software. O resultado dessa fase é o Léxico Ampliado da Linguagem.

### 2.1.3 Identificação de Regras Gramaticais

Com base nos símbolos elicitados e explicados, o engenheiro de software procurará descrever, operacionalmente, os processos da aplicação. Ou seja com base no LAL, o engenheiro de software procurará descrever, tarefas e procedimentos existentes na aplicação. Esta fase, a que chamamos de verbalização, tem dois objetivos: fornecer subsídios para a construção das regras da gramática e validar as fases anteriores.

Nesta fase o engenheiro de software deve, novamente, usar uma estratégia de abordagem, e procurar descrever o elicitado usando os símbolos já identificados. Neste caso, o engenheiro de software estará com sua atenção voltada para o entendimento dos diversos processos que ocorrem na aplicação, ou seja, sua principal preocupação é saber como a aplicação funciona.

O enfoque utilizado para montar a verbalização é, ao contrário do utilizado na identificação dos símbolos, um enfoque mais próximo dos métodos da análise de sistemas. Apesar dessa proximidade, a verbalização ainda não é a especificação do sistema, já que a linguagem ainda não está inteiramente definida. A verbalização, utilizando-se do Léxico e da investigação das funções da aplicação, procura descrever situações existentes na aplicação com o objetivo precípuo de prover auxílio na identificação das regras da gramática.

O resultado dessa fase é uma gramática, sendo a verbalização um resultado intermediário. Não nos aprofundaremos, neste trabalho, sobre a identificação de regras gramaticais.

### 2.1.4 Validação

As três fases de identificação são interligadas por um ciclo de validação. A validação pode ser feita de maneiras distintas [Leite 90] sendo que atualmente usamos principalmente uma



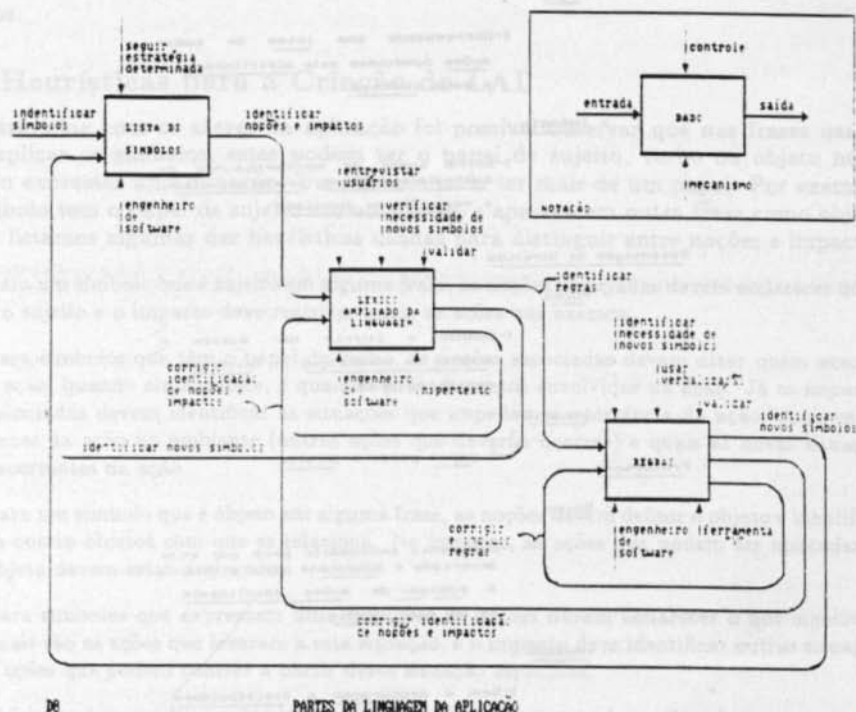


Figura 1: O Processo de Validação, um Datagrama SADT

validação informal com algumas nuances de validação por pontos de vista. Basicamente o ciclo de validação se apoia em três tipos de retroalimentação, a saber:

- necessidade de identificar novos símbolos,
- correção de noções e impactos e
- correção ou inclusão de regras.

O ciclo de validação está descrito por um datagrama apresentado na Figura 1. Um datagrama é um modelo SADT [Ross 77] em que a ênfase está na perspectiva de dado. No diagrama do tipo datagrama, as setas são atividades e os retângulos são objetos. São obrigatórias as entradas e os controles são opcionais. No diagrama da Figura 1 procuramos representar as principais retroalimentações entre os componentes de uma linguagem de aplicação.

A seguir detalharemos a notação usada no LAL e introduziremos os princípios que justificam o uso de hipertexto como uma ferramenta adequada como suporte para a formação do léxico.

- Posição da Conta / Posição de Custódia / Posição:
- Noção :
- 1-Corresponde aos lotes de todas ações possuídas pela distribuidora e seus clientes
- Impacto :
- 1-Sofre todas as movimentações de custódia (operações, transferências e exercício de direitos)
- Prescrição de Direitos :
- Noção :
- 1-Quando o cliente não exerce o direito de subscrição ou dividendo dentro do prazo
- Impacto :
- 1-O papel perde o direito
- Procuração :
- Noção :
- 1-Documento necessário para que seja levantada e bloqueada para a venda a posição de ações escriturais de um dado cliente
- Impacto :
- 1-Sem a procuração a distribuidora não poderá vender ações escriturais de um cliente

Figura 2: Parte do Léxico da Custódia de Ações

### 3 O Léxico Ampliado da Linguagem

O Léxico Ampliado é formado por um conjunto de descrições. O objetivo dessas descrições é prover a semântica dos símbolos da linguagem. A descrição semântica dos símbolos segue um sistema de representação onde cada símbolo deve ser descrito através de noções e impactos. A representação do Léxico é simples, cada símbolo é uma entrada e as noções e impactos são itens dessa entrada. Nestes itens, toda referência a outras entradas (princípio da circularidade) deve ser sublinhada. A Figura 2 mostra uma parte deste léxico da linguagem de custódia de ações<sup>3</sup>.

Quando este estudo começou, não existia um conceito bem definido que identificasse exatamente o que são noções e impactos. As primeiras experiências de elicitação de linguagens foram realizadas de uma forma bastante intuitiva. Esta estratégia experimental foi de grande importância, pois, a partir delas, chegou-se a algumas heurísticas sobre o melhor uso do sistema de representação especialmente na diferenciação entre noção e

<sup>3</sup>Esta linguagem foi produzida por Regina Ferreira e Maria Lúcia Astorga como trabalho de curso no seminário sobre Draco no segundo semestre de 1989 na PUC-RIO.

impacto. A seguir detalharemos algumas heurísticas identificadas para a descrição dos símbolos.

### 3.1 Heurísticas para a Criação do LAL

Nas entrevistas com os atores da aplicação foi possível observar que nas frases usadas para explicar os símbolos, estes podem ter o papel de sujeito, verbo ou objeto numa frase, ou expressar uma situação. Um símbolo pode ter mais de um papel. Por exemplo: um símbolo tem o papel de sujeito em uma frase, e aparece em outra frase como objeto. Abaixo listamos algumas das heurísticas usadas para distinguir entre noções e impactos.

- Para um símbolo que é sujeito em alguma frase, as noções associadas devem esclarecer quem é o sujeito e o impacto deve registrar quais as ações que executa.
- Para símbolos que têm o papel de verbo, as noções associadas devem dizer quem executa a ação, quando ela acontece, e quais os procedimentos envolvidos na ação. Já os impactos associados devem identificar as situações que impedem a ocorrência da ação, quais os reflexos da ação no ambiente (outras ações que deverão ocorrer) e quais as novas situações decorrentes da ação.
- Para um símbolo que é objeto em alguma frase, as noções devem definir o objeto e identificar os outros objetos com que se relaciona. No impacto, as ações que podem ser aplicadas ao objeto devem estar declaradas.
- Para símbolos que expressam uma situação, as noções devem esclarecer o que significa e quais são as ações que levaram a esta situação, e o impacto deve identificar outras situações e ações que podem ocorrer a partir desta situação específica.
- O Léxico deve ser feito cuidadosamente, à medida que o engenheiro de software vai aumentando seu conhecimento sobre a aplicação. Durante este processo de eliciação da linguagem, o engenheiro de software está sempre fazendo anotações, revendo e reescrevendo noções e impactos, e anotando novos símbolos relevantes para a aplicação.

Através do LAL é possível observar a integridade e variedade de locuções e termos desta linguagem, própria de uma aplicação. Pois, além da descrição de cada símbolo, os sinônimos estão explicitamente indicados, como também os relacionamentos entre os diversos símbolos.

As experiências têm demonstrado que os atores de uma determinada aplicação seguem naturalmente o princípio de circularidade, quando tentam explicar o significado do símbolo para o ambiente. Portanto, esta idéia de se ater aos termos da linguagem e o uso do Léxico apontam caminho próprio para a definição e validação da linguagem específica, base do conhecimento da aplicação em questão.

## 4 O Léxico Ampliado da Biblioteca PUC

Um dos experimentos de eliciação de uma linguagem para uma aplicação específica foi a eliciação da linguagem de biblioteca da PUC. Este exercício foi feito no primeiro semestre de 89 durante o seminário sobre o Paradigma Draco na PUC-RIO e revisto no princípio de 1990. O primeiro experimento foi feito sem que as heurísticas estivessem totalmente delineadas, sendo que algumas delas surgiram no decorrer do trabalho da



biblioteca e das outras linguagens já elicítadas. A revisão foi feita usando-se um conjunto de heurísticas mais estáveis. A seguir procuraremos apresentar uma série de observações sobre a elaboração da primeira versão da linguagem biblioteca e da sua posterior revisão.

#### 4.1 Elaboração da Primeira Versão da Linguagem Biblioteca

No primeiro experimento, executado pela co-autora e outra mestranda, Claudia Curcio, a identificação das palavras e frases usadas pelos bibliotecários e usuários da biblioteca foi feita a partir do conhecimento prévio do ambiente, e de observação e entrevistas com os bibliotecários. Como produto desta fase obteve-se uma lista inicial de símbolos para esta aplicação.

Através do conhecimento prévio da aplicação, alguns símbolos já estavam listados antes da primeira entrevista com a bibliotecária do Departamento de Informática. Duas entrevistas foram realizadas com esta bibliotecária. Em ambas as entrevistas a postura foi de uma atitude passiva, isto é, foi deixado que a bibliotecária falasse livremente sobre a biblioteca. No decorrer da entrevista eram anotados os símbolos julgados próprios pelas entrevistadoras.

A fase seguinte foi feita usando a abordagem de entrevistas. Foram feitas 2 entrevistas com a bibliotecária supervisora da biblioteca central da PUC, com o objetivo de descrever a semântica dos símbolos através do LAL. Esta fase foi a parte mais difícil do trabalho, pois não havia uma explicação clara do que era noção e impacto. O problema maior era o de distinguir o que seria descrito como noção e o que seria descrito como impacto. Muitos símbolos deixaram de apresentar impactos no LAL porque este conceito não tinha sido claramente explicitado.

Em seguida, com base no LAL foi feita a verbalização e posteriormente, a descrição da gramática da linguagem em BNF [Pagan 81]. Também nesta fase foram encontradas dificuldades, principalmente na tentativa de identificação de estruturas de controle próprias à linguagem. A BNF gerada foi baseada em uma linguagem procedural simplificada e não foi utilizado nenhum suporte para sua verificação.

#### 4.2 A Revisão da Linguagem Biblioteca

A revisão da linguagem foi feita pela co-autora e teve-se somente até a metade da terceira fase, produzindo uma primeira verbalização. Nesta revisão não se produziu uma descrição em BNF. No primeiro experimento, a preocupação maior era a de identificar os símbolos. Na revisão, a preocupação durante as entrevistas era validar as descrições associadas a cada símbolo.

Com heurísticas mais claras a respeito dos conceitos de noção e impacto, a revisão foi efetuada usando-se a abordagem de entrevistas e observação. As entrevistas, em número de 6, foram realizadas com a bibliotecária supervisora da biblioteca central. As entrevistas foram objetivas, isto é, eram reunidas dúvidas sobre os símbolos e suas descrições para compor a agenda da entrevista. As últimas duas entrevistas foram basicamente para validar o LAL e a verbalização.

**- ACERVO / COLEÇÃO****- Noção:**

- 1- Conjunto total de obras da biblioteca.
- 2- Conjunto organizado onde cada elemento (obra) tem sua localização determinada por um número de chamada.

**- Impacto:**

- 1- Pode-se fazer aquisição e o registro de novas obras no acervo.
- 2- Uma obra pode sumir do acervo (obra está desaparecida).

**- AQUISIÇÃO****- Noção:**

- 1- Tarefa realizada pelo bibliotecário.
- 2- Compra ou recebimento de doações de obras para o acervo.

**- Impacto:**

- 1- Bibliotecário deve fazer o registro e o processamento-técnico da obra.

**- ASSUNTO****- Noção:**

- 1- Matéria de que trata uma obra.

**- Impacto:**

- 1- Dado para localização de livros, folhetos, teses ou publicações-PUC.

Figura 3: Parte do Léxico da Biblioteca

### 4.3 Observações Gerais

Inicialmente, este processo de entrevistas não foi encarado com naturalidade pelos bibliotecários. A princípio, quando o bibliotecário foi informado que a pessoa que o estava entrevistando era da área de informática, teve alguma dificuldade em compreender que a pessoa estava interessada na sua linguagem, e não em um sistema automatizado ideal para o ambiente da biblioteca. À medida que o bibliotecário foi acompanhando o processo de elaboração do LAL, não só compreendeu este processo de elicitación de sua linguagem, como também, auxiliou na sua validação.

Foi observado que esta proposta de elicitación minimiza a dificuldade de comunicação envolvida nesta tarefa. Seguindo os princípios de vocabulário mínimo e circularidade, o engenheiro de software consegue conversar com atores envolvidos na aplicação, usando uma mesma linguagem, num mesmo nível de abstração. Deste modo, são os próprios atores quem guiam o engenheiro de software na identificação e descrição dos símbolos. A Figura 3 dá uma idéia sobre o LAL.

Atualmente, o LAL da biblioteca conta com um total de 80 símbolos, 209 noções e 145 impactos, sendo que na primeira versão tinha-se 56 símbolos, 118 noções e 27 impactos. As duas versões da elicitación da linguagem de biblioteca foram feitas usando como ferramenta, somente um editor de textos comum. No decorrer deste processo de elaboração do LAL, observamos que um sistema de hipertexto poderia ser uma forma de representação adequada para o LAL. A seguir descrevemos sucintamente a idéia de hipertexto e demonstramos a adequabilidade do seu uso no processo de elicitación de

linguagens da aplicação.

## 5 Hipertexto e o Léxico Ampliado da Linguagem

Um hipertexto é uma forma de organizar informações através de uma rede de nós conectados por ligações. Os sistemas de hipertexto têm duas funções básicas: uma relacionada a criação de nós e ligações e outra relacionada a navegação por esta rede. A idéia de hipertexto está intimamente ligada a produção de documentos eletrônicos, isto é, baseadas em computador. A possibilidade do tratamento da informação explorando as capacidades de armazenamento, recuperação e apresentação disponíveis nos computadores tem propiciado um crescente interesse por parte de pesquisadores e indústria em sistemas de software para suporte a hipertexto [Lima 89], [Meira 89], [Kelner 89], [CACM 90].

O nosso enfoque é o de demonstrar como o LAL se caracteriza como um hipertexto, e assumiremos que o leitor tem noções básicas sobre hipertexto. Em seguida, serão analisadas algumas questões relacionadas ao processo de autoria, ou seja, de criação de nós e ligações no hipertexto, e logo depois será apresentado um exemplo do uso do Hyperties como uma ferramenta na elaboração do LAL.

### 5.1 O problema da autoria

Os problemas relacionados à autoria trazem, em geral, as maiores dificuldades no uso do hipertexto como um meio de organizar informações [Raymond 88]. Uma destas dificuldades está relacionada à necessidade de fragmentação da informação. Um nó do hipertexto deve representar a expressão individual de uma única idéia ou conceito. Logo, a fragmentação é uma tarefa difícil, principalmente quando estas idéias e conceitos não estão bem definidos. A fragmentação deve ser feita cuidadosamente para que o hipertexto seja uma representação fiel da informação original. Outra dificuldade está relacionada à criação de ligações no hipertexto. Um dos objetivos principais do hipertexto é motivar a exploração das informações conectadas. Assim, informações isoladas, mal ligadas ou ligadas num contexto errado não têm utilidade e podem levar a um caminho errado durante o processo de navegação. Portanto, a fragmentação e a definição de ligações entre os nós do hipertexto, em geral, não são um problema trivial.

No caso do LAL observa-se que sua concepção é feita fundamentalmente com base na fragmentação e na marcação. Portanto, enquanto autoria do hipertexto estes problemas não existem para o LAL. Primeiro, porque a descrição de um símbolo representa um único conceito. A informação já é modularizada por natureza, logo, não existem dificuldades quanto à fragmentação. Como é seguido o princípio de circularidade na descrição de um símbolo, o engenheiro de software também não tem grande dificuldade na definição das ligações, pois, estas estão presentes naturalmente.

No nosso caso, cada símbolo, entrada do léxico, deve ser representado como um nó do hipertexto que contém as noções e impactos associados àquele símbolo. Como a descrição destas noções e impactos segue o princípio da circularidade, os símbolos presentes nestas descrições, palavras e frases sublinhadas no Léxico, devem ser usados para conectar os diversos nós. Deste modo, o hipertexto é uma forma natural de expressar o LAL, e os mecanismos de navegação, além da facilidade de visualização da circularidade, fornecem

uma dinâmica importante no processo de elicitação da linguagem, principalmente no que concerne à validação.

Usando o hipertexto como uma forma de representação do LAL, o engenheiro de software tem a sua disposição a lista completa dos símbolos, facilidades para percorrer esta lista e principalmente, facilidades para percorrer as descrições dos símbolos através das ligações definidas. A seguir reportaremos a implementação do LAL da biblioteca no sistema Hyperties.

## 5.2 O Uso do Hyperties na LAL da Biblioteca da PUC

O Hyperties é um sistema de hipertexto desenvolvido na Universidade de Maryland [Kreitzberg 88]. O Hyperties possui duas ferramentas básicas: uma para autoria e outra para navegação. Com a ferramenta de autoria, AUTHOR, é possível criar bases de dados compostas por artigos e gráficos, que podem ser facilmente ligados. Usando a ferramenta de navegação, BROWSER, os usuários podem facilmente percorrer estas informações ligadas seguindo caminhos de interesse.

A ferramenta de autoria do Hyperties tem comandos para criar e editar artigos (nós), havendo a possibilidade de importar um artigo preparado em um outro editor. Um artigo do Hyperties é identificado por um nome, pode ter uma lista de sinônimos associados e é composto por cabeçalho, descrição e conteúdo. Para definir uma ligação, durante a edição o usuário escreve no conteúdo do artigo a palavra ou frase usada para referência entre " ". Após a edição do artigo, essas ligações são explicitamente criadas. O comando do Hyperties que resolve as ligações busca, entre os artigos criados, se algum tem o mesmo nome ou é sinônimo da palavra ou frase marcada e cria a ligação. Caso não encontre, permite que o usuário defina esta referência como sinônimo de um artigo que já existe ou crie um novo artigo. O Hyperties dispõe de comandos para alterar as ligações e visualizar a lista de sinônimos de cada artigo.

A ferramenta de navegação é usada para apresentar o conteúdo dos artigos na tela e percorrer as ligações. As ligações de um artigo aparecem realçadas durante esta apresentação. O usuário pode se posicionar sobre uma das ligações e percorrê-la. O BROWSER busca o artigo que a ligação referencia e o apresenta na tela. Durante este processo de navegação, o Hyperties tem um comando que permite visualizar o caminho que já foi percorrido e sempre há a possibilidade de retornar ao artigo anterior.

A figura 4 mostra uma entrada do Léxico representada como um nó, article, do Hyperties. Esta entrada contém as noções e impactos associados ao símbolo que, por sua vez, têm ligações para outras entradas (nós), ou seja, símbolos marcados. A ferramenta de navegação permite o acesso direto à entrada correspondente a partir da seleção de um símbolo marcado. Este exemplo é apresentado na Figura 5.

A forma de organização do Hyperties é bem adequada para a aplicação em questão. Os nós constituem artigos, articles, que por sua vez constituem uma entrada no sumário, index (Figura 6). Deste modo, o engenheiro de software tem uma nova possibilidade para percorrer os símbolos do LAL através do index. Uma outra vantagem é que o usuário pode definir sinônimos para os artigos (nós), o que é extremamente útil no nosso caso.

Nesta versão atual do LAL, apenas exploramos a vantagem de navegação, uma vez que o LAL estava pronto, mas pudemos observar as facilidades que a ferramenta provê

CONSULTA

PAGE 1 OF 1

Noção

- Leitura de um exemplar dentro da biblioteca.
- Usuário tem a posse do exemplar.

Impacto

- Exemplar está na biblioteca.
- Exemplar não está na estante (obra está na estante).
- Após a consulta, bibliotecário deve **colocar obra na estante.**

EXTRA

Figura 4: Uma Entrada no Léxico

COLOCAR OBRA NA ESTANTE

PAGE 1 OF 1

Noção

- Tarefa realizada pelo bibliotecário após a consulta, devolução e o processamento técnico de um exemplar.
- Bibliotecário coloca o exemplar em seu lugar na biblioteca.
- Caso haja dúvida, bibliotecário consulta o catálogo topográfico.

Impacto

- Obra está na estante.

**RETURN TO "CONSULTA"**

EXTRA

Figura 5: Navegação



## INDEX of ARTICLES

INDEX - All entries	
25.	DICIONARIO
17.	*t CLASSIFICAÇÃO
18.	*t COBRANÇA DE MULTAS / COBRAR MULTA
19.	*t COLOCAR OBRA NA ESTANTE
20.	*t CONSULTA
21.	*t DAR CABEÇALHO DE ASSUNTOS
22.	*t DATA DE PUBLICAÇÃO
23.	*t DATA PARA DEVOLUÇÃO
24.	*t DEVOLUÇÃO
25.	DICIONARIO
26.	*t EDITORA
27.	*t EDIÇÃO
28.	*t EMPRÉSTIMO
29.	*t EMPRÉSTIMO PARA XEROX
30.	ENCICLOPEDIA
31.	*t EXEMPLAR
32.	*t FICHA POR AUTOR - FICHA POR ORDEM ALFABETICA

Press ESC to leave window and access commands.

EDIT	ADD	DELETE	RENAME			
INDEX	SYNONYMS	LINKS	GRAPHICS	PRINT	BROWSE	QUIT

Figura 6: Índices

durante o processo de autoria. O engenheiro de software tem a possibilidade de verificar quais os símbolos que um símbolo refere, e quais os símbolos que foram referidos por um determinado símbolo. Neste processo de elicitação de uma linguagem, geralmente, quando um símbolo faz referência direta a um outro símbolo, este outro símbolo também deve fazer alguma referência ao primeiro.

Embora o uso do Hyperties auxilie a elicitação e validação da linguagem, faltam algumas facilidades importantes neste processo. A primeira é uma própria restrição do Hyperties, pois não dispõe de mecanismos de janela para apresentação de um nó, ARTICLE. É importante também que mecanismos para fazer algumas verificações automáticas estejam presentes. Num sistema simples de verificação, poderia-se, por exemplo, apontar as palavras ou frases que constituem símbolos e que não foram marcadas, e permitir ao usuário escolher se quer ou não criar a ligação. Faltam também facilidades de formatação associadas ao hipertexto, por exemplo: no caso do LAL é interessante que o texto seja impresso no mesmo formato usado originalmente, ver Figuras 2 e 3.

## 6 Conclusão e Trabalhos Futuros

Seguindo o princípio de conhecimento específico do paradigma Draco, nossa pesquisa vem se concentrando no estudo relacionado ao problema da análise de domínios. Em particular estamos interessados em estudar métodos para apoiar as atividades de análise de domínio e nossa estratégia é a de estudar o problema em uma escala reduzida antes de atacar o problema na sua totalidade. Esta redução é o que chamamos o problema da produção de linguagens da aplicação. O objetivo final do processo de produção de uma linguagem da aplicação é obter uma gramática, um conjunto de transformações horizontais e a semântica descrita em forma de componentes executáveis. De posse desses itens, programas executáveis seriam produzidos pelo sistema Draco.

No nosso artigo centramos nossa atenção em uma das partes do processo de produção de linguagens da aplicação. Esta parte tem como base o Léxico Ampliado da Linguagem. Apresentamos uma série de heurísticas relacionadas à eliciação do LAL e mostramos a adequabilidade do enfoque de hipertexto para o LAL. Usamos como exemplo uma das linguagens já elicitadas por nosso grupo de pesquisa, a linguagem da biblioteca da PUC. Os resultados obtidos são animadores e indicam que o LAL tem um papel importante na eliciação de linguagens da aplicação.

Como os resultados obtidos são parciais e são direcionados a uma das partes do problema, há vários pontos que necessitam de maior pesquisa. Sob o ponto de vista do LAL algumas heurísticas, já observadas, poderiam ser base para uma análise automática inteligente. Se tivéssemos a possibilidade de associar tipos aos nós (símbolos) do hipertexto poderíamos fazer com que cada tipo tivesse um conjunto de regras específicas para facilitar a edição e a verificação automática das descrições associadas aos símbolos. Por exemplo, um tipo A poderia ter a seguinte regra: todos os símbolos do tipo A devem ter pelo menos uma noção associada que informe quem executa a ação. Um símbolo que tem o papel de verbo em alguma frase poderia estar associado a este tipo. Deste modo, haveria a possibilidade do ambiente informar ao engenheiro de software que ele deve escrever e/ou consultar se existe uma noção dizendo quem executa a ação.

Além do aspecto de busca de heurísticas, há também a necessidade de desenvolvimento de ferramentas que complementem o hipertexto e sejam específicas ao LAL. Desta forma, poderíamos ter facilidades de levantamentos estatísticos sobre o Léxico com o objetivo de conhecer melhor não só a aplicação, como a própria estrutura do Léxico.

Sob o ponto de vista do restante do processo de produção de linguagens da aplicação, há também vários problemas a serem pesquisados, sendo os principais aqueles referentes à identificação das transformações horizontais, à identificação de estruturas de controle próprias da linguagem, além, é lógico, do problema de representação da semântica dos componentes.

## Referências

- [CACM 90] *Communications of the ACM* 33, 3. Mar. 1990.
- [Kelner 89] Kelner, J., Cavalcanti, A. e Prado, A. Linda: Uma Linguagem de Autoria para Sistemas de Hipertexto. in *Anais do III Simpósio Brasileiro de Engenharia de Software* (Recife, PE, 1989), 124-136.
- [Kreitzberg 88] Kreitzberg, C.B. and Hoffman, P. *Hypertics User's Manual*. Cognectics Corporation and University of Maryland, 1988.
- [Leite 89] Leite, J.C.S.P. Elicitation of Application Languages. In *Monografias em Ciência da Computação, PUC-RIO*, no. 30. 1989.
- [Leite 89b] Leite, J.C.S.P. e Lucena, C.J.P. The Rio Workshop on the Software Process Rio de Janeiro, Dec. 14-15. In *Monografias em Ciência da Computação, PUC-RIO*, no. 26. 1989.
- [Leite 90] Leite, J.C.S.P. Elicitação de Requisitos. In *Anais do XXIII Congresso Nacional de Informática, Sucesu*, Ago. 1990. (a aparecer).

- [Lima 89] Lima, M.J.D. Hipertexto e suas Aplicações. *Projeto Final de Curso*. Departamento de Informática - UFRJ, Maio 1989.
- [Lucena 87] Lucena, C.J.P. *Inteligência Artificial e Engenharia de Software*. J. Zahar, Rio de Janeiro, 1987.
- [Maibaum 87] Maibaum, T.S.E. and Turski, W. *The Specification of Computer Program*. Addison-Wesley, Workingham, 1987.
- [Meira 89] Meira, S., Kelner, J., Albuquerque, E., Martins, J., Melo, A. e Vasconcelos, A. Hipertexto: O Projeto do Sistema H. in *Anais do III Simpósio Brasileiro de Engenharia de Software* (Recife, PE, 1989), 152-170.
- [Neighbors 84] Neighbors, J. The Draco Approach to Constructing Software from Reusable Components. *IEEE Transactions on Software Engineering*, SE-10, 9, Sep. 1984, pp 564-573.
- [Pagan 81] Pagan, F. *Formal Specification of Programming Languages: A Panoramic Primer* Prentice-Hall, New Jersey, 1981.
- [Smith 85] Smith, D., Koek, G., Westfold, S. Research on Knowledge-Based Software Environments at Kestrel Institute. *IEEE Transactions on Software Engineering*, SE-11, 11, Nov. 1985, pp 1278-1295.
- [Raymond 88] Raymond, D.R. and Tompa, F. Hypertext and the Oxford English Dictionary. *it Communications of the ACM* 31, 7 (Jul 1988), 871-878.