

Área:

CV ( )

CHSA ( )

ECET ( X )

## **AMBIENTE DE GERAÇÃO DE DIAGRAMAS DE CLASSES UML A PARTIR DE DESCRIÇÃO EM LINGUAGEM NATURAL**

*Rafael Tôrres Anchiêta (bolsista I.C.V), Raimundo Santos Moura (Orientador, Depto de  
Informática e Estatística – UFPI)*

### Introdução

Na fase inicial do desenvolvimento de *software*, o cenário ideal para pesquisadores é possibilitar ao usuário final especificar as funcionalidades (requisitos) de um sistema através de uma língua natural escrita ou falada, pois toda a base para a construção do software é organizada pela engenharia de requisitos. Ela dá suporte à fase inicial do ciclo de vida do software.

De acordo com Kontoya and Sommerville (1998), a engenharia de requisitos engloba todas as atividades envolvidas na descoberta, documentação e manutenção de um conjunto de requisitos para um sistema computacional.

A Engenharia de Requisitos é considerada por alguns autores como a parte mais crítica no desenvolvimento de software, pois a qualidade do produto final depende fortemente da qualidade dos requisitos (Ferguson, R.; Lami, G. 2006). Fernandes, M.; Machado, R.; Seidman, S.; afirmam que 85% dos problemas de software têm origem na atividade de elicitação de requisitos.

No sentido de facilitar a fase de inicial do desenvolvimento de software foi implementado um protótipo para identificar padrões a partir do Processamento de Linguagem Natural (PLN), usando descrições textuais da língua portuguesa. A ideia é obter os requisitos funcionais de um sistema computacional, visando reduzir o tempo gasto dos engenheiros de requisitos em analisar toda a descrição informal definida pelo usuário final.

### Metodologia

Este trabalho foi dividido em seis etapas. Na primeira etapa, foi feita uma pesquisa e estudo sobre processamento de linguagem natural, analisando e identificando padrões linguísticos que envolvam elementos gráficos dos diagramas UML (*Unified Modeling Language*). Na segunda etapa, foi feita uma pesquisa pelos principais ambientes acadêmicos da área de PLN usados para etiquetar textos e os ambientes acadêmicos usados para especificar requisitos de *software*. Na terceira etapa, foi feita uma coleta de várias especificações requisitos para o desenvolvimento de um protótipo que atenda os requisitos funcionais das especificações de requisitos. Na quarta etapa, foi implementado o protótipo que a partir de uma descrição textual dos requisitos funcionais do sistema gera um diagrama de classe UML. Na quinta etapa, foram realizados alguns experimentos. Por fim na sexta etapa elaboramos o relatório final com discussões e resultados.

### Resultados e Discussões

Para realizarmos etiquetagem de um texto é necessária a definição de um conjunto finito de etiquetas (*tags*) e essas etiquetas devem ter um significado linguístico associado.

O processo de etiquetagem consiste em realizar a associação das palavras que aparecem em

Área:

CV ( )

CHSA ( )

ECET ( X )

um dado texto a uma determinada etiqueta do conjunto finito de etiquetas. Essa associação é feita de acordo com o algoritmo do etiquetador utilizado. Neste projeto utilizamos a ferramenta ANTLR (Parr, T., 2007), em conjunto com o dicionário de palavras desenvolvido através do projeto ICV – UFPI “**Dicionário Eletrônico da Língua Portuguesa para Apoiar Pesquisas do Laboratório de Engenharia de Software e Informática Industrial (EaSI)**”, executado em paralelo.

A ferramenta ANTLR (*Another Tool for Language Recognition*) é um gerador de *parser* que automatiza a construção de reconhecedores para DSLs. Ele gera uma gramática combinada que especifica tanto o *parser* quanto as regras léxicas da linguagem. O formato dos *tokens* da linguagem desejada é definido através de expressões regulares. Já as regras gramaticais do *parser* são descritas através de uma gramática livre de contexto, na notação EBNF (*Extended Backus Naur Form*).

O protótipo foi desenvolvido utilizando a linguagem de programação Java utilizando a tecnologia J2SE, ele recebe como entrada uma descrição textual qualquer escrita em português e, após analisar alguns padrões textuais pré-definidos, gera as possíveis classes, atributos e métodos. A Figura 1 mostra uma visão geral do processo.

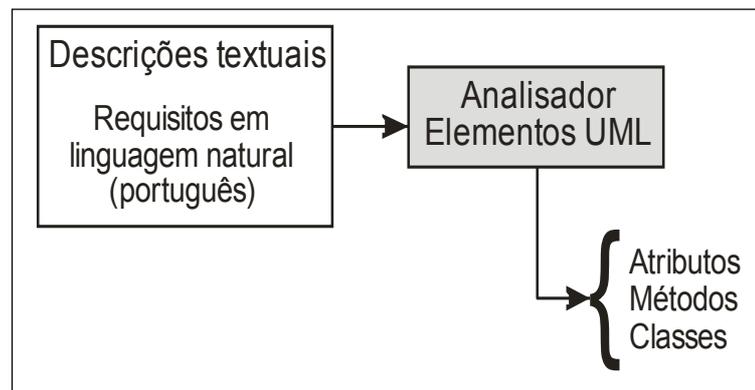


Figura 1: Elementos UML a partir de descrições textuais: visão geral

O protótipo possui uma interface bastante simples, com opções para criar, abrir e salvar um documento, o protótipo permite abrir arquivos nos formatos: Arquivo Texto Comum (TXT), *Portable Document Format* (PDF), e *Document* (DOC) e salvar arquivos no formato (TXT).

O protótipo também possui um módulo para importar arquivos no formato (XMI) *XML Metadata Interchange* que é um *Extensible Markup Language* (XML) usado para troca de informações entre programas de modelagem que tem por base o UML, além de uma opção para executar o módulo de análise do requisito funcional inserido na ferramenta.

O módulo de análise separa as palavras (*tokens*) e consulta uma estrutura de dados do tipo *HashMap* que armazena o dicionário de palavras. Para isso utilizou-se a ferramenta ANTLR em conjunto com o ambiente integrado NetBeans IDE. Após a separação das palavras que chamamos de fase léxica, a fase sintática é executada para identificar os possíveis padrões gramaticais da

Área:

CV ( )

CHSA ( )

ECET ( X )

descrição. Os padrões identificados são apresentados ao usuário na forma de um diagrama de classe UML com as possíveis classes, atributos e métodos da descrição.

A Tabela 1 apresenta os padrões textuais usados para especificar as regras gramaticais dos elementos UML usados. Nesta tabela a sintaxe utilizada está de acordo com a notação EBNF (*Extended Backus Naur Form*).

Tabela 1: Elementos UML: padrões textuais

Candidato à	Padrões Linguísticos
Atributo	(Conjunção ou Dois Pontos ou Vírgula) e [Substantivo e [Preposição + Artigo] e Substantivo
Atributo	(Conjunção ou Dois Pontos ou Vírgula) e Substantivo
Método	Verbo e Artigo e Substantivo
Classe com nome composto	Substantivo e Adjetivo
Classe com nome simples	Substantivo

#### Conclusão

A principal contribuição deste projeto é o desenvolvimento de um protótipo para mostrar os resultados de um processo de análise de requisitos. O protótipo foi desenvolvido em linguagem de programação Java com o uso do ambiente NetBeans IDE e da ferramenta de geração de analisadores léxico-gramaticais ANTLR. Faz-se uso também de um dicionário de palavras com cerca de 9.000 raízes para classificar todas as palavras da descrição quanto à classe gramatical.

Como trabalhos futuros podem se incluídos: identificar a mesma classe, atributo e/ou método quando se tratar de palavras no plural e singular e aprimorar os padrões gramaticais para identificar os elementos UML.

#### Referências Bibliográficas

- Kontoya and Sommerville (1998), **Requirements Engineering: Processes and Techniques**.  
Fernandes, M.; Machado, R.; Seidman, S. (2009) **A Requirements Engineering and Management Training Course for Software Development Professionals**. 22th Conference on Software Engineering Education and Training.  
Ferguson, R.; Lami, G. (2006) **An Empirical Study on The Relationship Between Defective Requirements and Test Failures** In. Software Engineering Workshop, v0 p7-10 IEEE Computer Society, april.  
Parr, T. (2007), **The Definitive ANTLR Reference: Building Domain-Specific Languages. The Pragmatic Programmers**. (PDF available in <http://www.praprog.com/titles/tpantlr/the-definitive-antlr-reference>).

**Palavras-chave:** Engenharia de Requisitos. Processamento de Linguagem Natural. Engenharia de Software.