



## 5º Congresso de Pós-Graduação

### UMA FERRAMENTA PARA GERAÇÃO AUTOMÁTICA DE DIAGRAMA DE CLASSES A PARTIR DA ESPECIFICAÇÃO DE REQUISITOS EM LINGUAGEM NATURAL

#### Autor(es)

---

WILSON CARLOS DA SILVA

#### Orientador(es)

---

LUIZ EDUARDO GALVÃO MARTINS

#### 1. Introdução

---

Toda a base para a construção do software é organizada pela Engenharia de Requisitos. A Engenharia de Requisitos dá suporte à fase inicial do ciclo de vida do software. De acordo com Kotonya 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 especificação de requisitos é um documento, ou artefato, utilizado para a comunicação entre os clientes, usuários, engenheiros de requisitos e, em algumas situações, são utilizados para comunicação com os desenvolvedores de software. A maioria das especificações de requisitos é escrita em linguagem natural (LN), sendo complementada, às vezes, por outros tipos de notações, tais como diagramas, fórmulas, modelos formais. A especificação de requisitos em LN facilita a comunicação humana, pois podemos descrever as minúcias dos problemas e suas respectivas soluções. De acordo com uma pesquisa on-line conduzida pela Universidade de Trento (Mich, Franch, and Novi Inverardi, 2003), a maioria dos documentos disponíveis para análise dos requisitos está em LN ou são informações obtidas através de entrevistas. Conseguiram mediante esta pesquisa chegar aos seguintes números: · 79% dos documentos são escritos em LN comum. · 15% dos documentos são escritos em LN estruturada e · Somente 5% dos documentos são escritos em linguagem formal. De acordo com os dados apresentados nesta pesquisa, fica evidente a grande quantidade de documentos em LN no processo de elicitação de requisitos. Existem várias técnicas que auxiliam os engenheiros de requisitos na elicitação dos requisitos, para que posteriormente os requisitos possam ser especificados da melhor maneira possível. Segundo Davis and Hickey (2003), essas técnicas são métodos utilizados pelos engenheiros de requisitos para determinar as necessidades dos clientes e usuários, essas necessidades sendo identificadas claramente, o sistema poderá ser construído com uma grande chance de atender as expectativas e as reais necessidades dos usuários.

#### 2. Objetivos

---

Os engenheiros de requisitos contam com várias técnicas que os auxiliam no processo de elicitação, análise,

especificação, validação e gerenciamento de requisitos. Os requisitos são a base para o desenvolvimento do software, os quais na maioria das vezes são descritos em linguagem natural. O objetivo deste trabalho é desenvolver uma ferramenta para auxiliar os engenheiros de requisitos a identificarem com maior facilidade classes, atributos e métodos a partir dos requisitos descritos em linguagem natural, utilizando-se do processamento da linguagem natural, gerando um diagrama de classes conceitual da UML (Unified Modeling Language). Além desta característica a ferramenta permitirá a configuração de padrões lingüísticos, que facilitam a criação de diagramas mais próximos daqueles criados por modelador humano.

### 3. Desenvolvimento

Quando a o termo “crise do software” foi definido na década de 1960, um grande esforço foi direcionado para encontrar as causas do problema. A investigação concluiu que a deficiência na especificação de requisitos estava entre as mais importantes causas do problema. O desenvolvimento da especificação de requisitos em muitos casos parece trivial, mas é provavelmente a parte do processo onde ocorrem mais falhas do que qualquer outra (Schwartz 1975). Conforme mencionamos na introdução, grande parte dos requisitos é especificada utilizando LN. A utilização da LN traz consigo o problema da ambigüidade. De acordo com Daniel Berry e Erik Kamsties (2004) a ambigüidade é um fenômeno do mundo real que é estudado por uma gama de disciplinas tais como: lingüística, filosofia, direito e não poderia deixar de mencionar engenharia de software. Daniel Berry and Erik Kamsties (2004) trazem em seu artigo que não existe uma especificação sem ambigüidade. O que existe na verdade, é uma especificação eficaz. Uma especificação eficaz é uma especificação que é entendida pela maioria das pessoas envolvidas no processo de desenvolvimento do software, permitindo que o software realize o que a maioria espera que ele realize. A motivação para estudos relacionados ao PLN é revolucionar a forma que o computador é utilizado. A maioria do conhecimento humano é armazenada na forma de LN, computadores que puderem entender a LN poderão acessar toda essa informação de forma mais prática e rápida. Segundo Allen (1995) a linguagem é um dos aspectos fundamentais do comportamento humano e é um componente crucial em nossas vidas. Na forma escrita ela serve de base de conhecimento que passa de uma geração para outra. Na forma falada serve como nosso principal meio de comunicação no dia-a-dia com as outras pessoas. As formas de conhecimento apresentadas a seguir são essenciais para o entendimento da LN: conhecimento fonético, conhecimento morfológico, conhecimento sintático, conhecimento semântico, conhecimento pragmático, conhecimento de mundo. A ferramenta proposta neste trabalho será focada no conhecimento sintático em conjunto com padrões lingüísticos. A figura 1 apresenta a arquitetura utilizada pela ferramenta mostrando uma visão geral do processo. A figura 2 apresenta um diagrama de atividades mostrando as atividades executadas na utilização da ferramenta para auxiliar o engenheiro de requisitos no processo de geração do diagrama de classes conceitual.

Figura 1. Arquitetura da ferramenta

Figura 2. Atividades desenvolvidas no ambiente da ferramenta

O PLN é um fator preponderante no projeto, pois iremos utilizá-lo como ferramenta base para o desenvolvimento de todo o processo. O engenheiro de requisitos selecionará o arquivo contendo as descrições dos requisitos e/ou documentos fornecidos pelo cliente em formato texto. Será gerado um arquivo idêntico ao original, contendo as marcações “TAG” que identificam a classificação gramatical da palavra dentro da frase, a partir desse arquivo com as marcações a ferramenta irá classificar as palavras previamente identificadas em três categorias: classes, atributos, métodos. Para que a classificação das palavras seja feita de uma forma eficaz, iremos utilizar o que chamamos de padrões lingüísticos. Os padrões lingüísticos são na verdade, os mapeamentos de palavras de acordo com sua classificação gramatical dentro de uma frase. O componente MXPOST que adiciona as marcações “TAG” nos arquivos texto contendo as descrições dos requisitos foi desenvolvido por um grupo de pesquisa da Universidade de Pensilvânia. A ferramenta levará em consideração a análise sintática realizada pelo componente MXPOST. Como a ferramenta foi desenvolvida para levar em consideração os padrões lingüísticos para fazer o mapeamento, não realizará a análise semântica dos textos em LN, a qual ficará a cargo do engenheiro de requisitos que poderá interagir na ferramenta para realizar os devidos ajustes. A tabela 1 apresenta alguns exemplos de padrões lingüísticos definidos na ferramenta. Esses padrões poderão ser configurados pelo

engenheiro de requisitos para que a ferramenta se adapte ao seu estilo de escrever/documentar os requisitos.

#### Tabela 1. Padrões Lingüísticos definidos na ferramenta

Após a classificação das palavras nas categorias classes, atributos e métodos, o engenheiro de requisitos poderá interagir com a pré-classificação realizada automaticamente pela ferramenta, para fazer ajustes e até mesmo mudar a classificação de algumas palavras e descartar palavras que estão fora do domínio da aplicação. O diagrama será atualizado de acordo com as mudanças realizadas pelo engenheiro de requisitos, que poderá continuar o processo de ajustes após atualização do diagrama de classes conceitual, inclusive selecionando novos arquivos contendo descrições de requisitos para adicionar no diagrama atualizado com base em outros arquivos de descrições dos requisitos. O diagrama de classes conceitual poderá ser exportado para arquivos XML, no padrão que ferramentas de modelagens consolidadas no mercado possam importar e dar continuidade no processo.

## 4. Resultados

---

A ferramenta proposta neste trabalho está em uma fase avançada de desenvolvimento. As seguintes funcionalidades estão implementadas: realizar marcações no texto de entrada em LN, realizar análise dos padrões lingüísticos definidos previamente e classificar as palavras nas categorias de classes, atributos ou operações, exibir o texto de entrada no momento de captura para facilitar o rastreamento por parte do engenheiro de requisitos da palavra em seu real contexto, permitir a alteração e exclusão de palavras que estão fora do domínio da aplicação, permitir o vínculo dos atributos e operações nas suas classes respectivas, gerar diagrama de classes conceitual com base nos dados capturados pela ferramenta e alterados pelo analista de requisitos, permitir alterar e/ou adicionar classes, atributos e operações no diagrama de classes gerado, permitir adicionar associações entre as classes e realizar a exclusão das classes, atributos, operações e associações do diagrama de classes. É de suma importância a realização de testes e validação da ferramenta desenvolvida para auxiliar os engenheiros de requisitos, para de fato identificar os pontos onde podemos melhorá-la e qual a efetiva contribuição da ferramenta no contexto da Engenharia de Requisitos. Temos algumas opções para realização destes testes: no ambiente acadêmico utilizando alunos do mestrado nas disciplinas de engenharia de requisitos, ou desenvolver um projeto hipotético com alguns engenheiros de requisitos experientes, ou disponibilizar a ferramenta em um site, convidando empresas de desenvolvimento de software a utilizar a ferramenta no nível experimental, depois realizar uma avaliação on-line dos resultados percebidos por eles.

## 5. Considerações Finais

---

Vários estudos têm proposto recentemente a utilização de ferramentas lingüísticas para suporte a Engenharia de Requisitos. Existem duas razões principais para que isso ocorra. Primeiro, o progresso feito na área de PLN. Segundo, a necessidade de prover aos engenheiros de requisitos ferramentas que dêem suporte nos estágios iniciais do ciclo de vida do software. A ferramenta baseada em PLN na Engenharia de Requisitos que propomos neste trabalho pode trazer uma série de contribuições, dentre elas podemos destacar: permitir que os engenheiros de requisitos se concentrem mais no problema do que na modelagem, auxiliar na localização de ambigüidades e contradições nos documentos de especificação de requisitos, extrair diretamente dos textos em LN, elementos para gerar modelos conceituais, auxiliar na documentação dos requisitos, rastreabilidade entre os textos em LN e os artefatos produzidos, tradução dos documentos em várias línguas. A possibilidade de trabalhar com os padrões lingüísticos traz um diferencial importante em relação aos demais projetos na área de PLN na engenharia de requisitos. Algumas funcionalidades ainda faltam ser implementadas na ferramenta: recursos para identificação automática dos relacionamentos entre as classes e configuração dos padrões lingüísticos.

## Referências Bibliográficas

---

ALLEN, James **Natural Language Understanding**. The Benjamin/Cummings Publishing Company Inc,

1995.

BERRY, D.M.; KAMSTIES, E. **Ambiguity in Requirements Specification** In:LEITE, J. C. S. P.; DOORN, J. H. **Perspectives on Software Requirements**, Series: The Springer International Series in Engineering and Computer Science , Vol. 753, Kluwer Academic Publishers, 2004, Páginas 7-44.

DAVIS, A.M., HICKEY, A.M. **Elicitation Technique Selection: How Do Experts Do It**, Proceedings of the 11th International Requirements Engineering Conference, IEEE Computer Society Press, 2003.

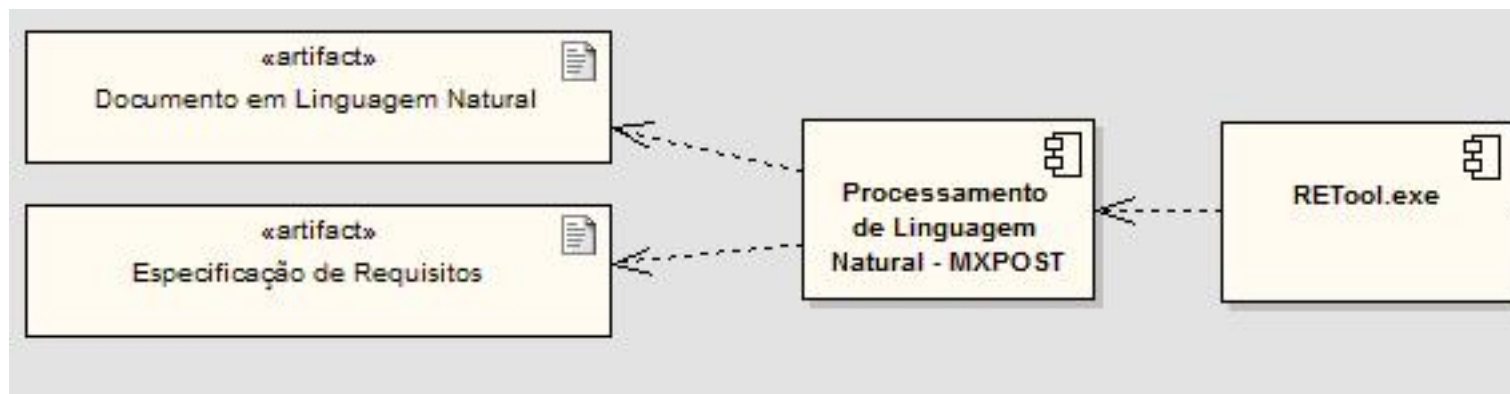
KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering: process and techniques**, New York: John Wiley & Sons Ltda, 1998.

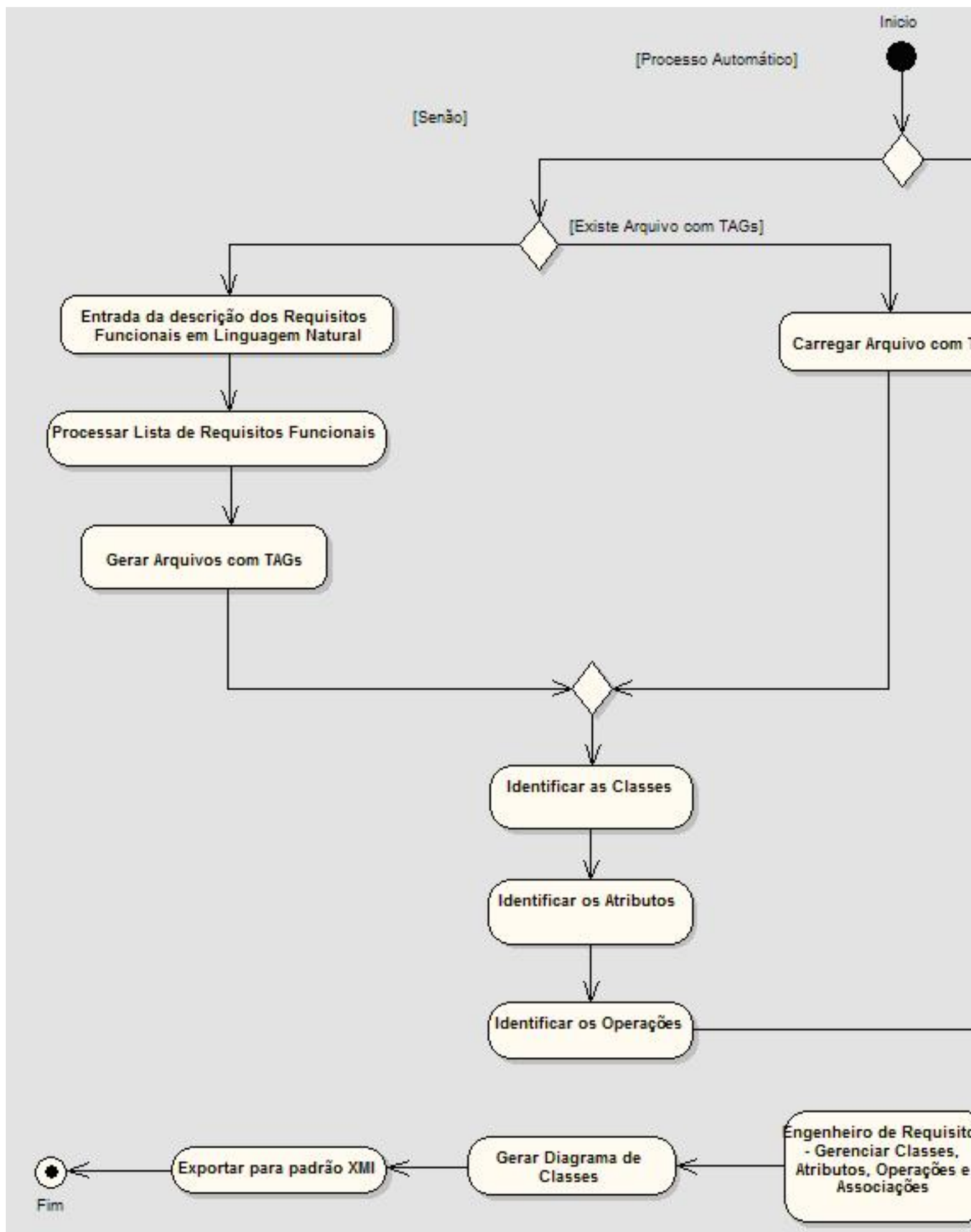
MICH, L., FRANCH, M. and NOVI INVERARDI, P. **Requirements Analysis using Linguistic Tools: Results of an On-line Survey**. Technical Report 66, Department of Computer and Management Sciences, 2003.

SCHWARTZ, J.I. **Construction of Software, Problems and Practicalities**, in Pratical Strategies for Developing Large Software Systems, E. Horowitz, ed., Addison-Wesley, Reading, Mass, 1975.

## Anexos

---





### Padrões Linguísticos

[Dois Pontos ou Vírgula ou Conjunção] e [Substantivo ou Nome Próprio] e [Vírgula ou Conjunção ou Ponto ou Preposição Artigo]

[Substantivo] e [Verbo ou Ponto]

[Substantivo] e [Adjetivo] → Gera nome composto

[Verbo] e [Artigo] e [Substantivo ou Nome Próprio] → Gera nome composto