

# **Linguagem para especificação de aplicações com base em Redes de Petri Adaptativas.**

Joel Rodrigues Alvares LEAL<sup>1</sup>, Almir Rogério CAMOLESI<sup>2</sup>

Joel\_19lakers@hotmail.com, camolesi@femanet.com.br

Instituto Municipal de Ensino Superior de Assis (IMESA)

Fundação Educacional do Município de Assis (FEMA) – Assis/SP (Brasil)

**RESUMO:** Uma técnica utilizada para auxiliar os projetistas na modelagem de aplicações com comportamento modificável é a tecnologia adaptativa. A tecnologia adaptativa envolve um dispositivo não-adaptativo (subjacente) já existente em uma camada adaptativa que permite realizar mudanças no comportamento da aplicação definida.

Foi desenvolvido uma linguagem para especificação de aplicações com fundamento em Redes de Petri Adaptativas. Tal linguagem deve possibilitar que um projetista utilize-se, posteriormente, de um tradutor da linguagem produzida para um Modelo Lógico.

**PALAVRAS-CHAVE:** Rede de Petri; Tecnologia Adaptativa; Adaptativo; Linguagem; Modelo Lógico.

**ABSTRACT:** A technique used to assist designers in modeling applications with modifiable behavior is adaptive technology. Adaptive technology involves a non-adaptive device (underlying) existing in an adaptive layer that allows for changes in the behavior of the specific application. A language specification for application on the grounds of Adaptive Petri nets was developed. Such language should enable a designer to use later, a translator of the language produced for Logical Model.

**KEYWORDS:** Petri net; Adaptive technology; Adaptive; Language; Logic Model.

Aplicações complexas são caracterizadas por componentes e aspectos cuja estrutura e comportamento, comumente, podem modificar-se (CAMOLESI, 2007). Tais aplicações possuem um comportamento inicial definido por um conjunto de ações que desempenham suas funções elementares e podem ter o seu comportamento modificado durante a execução para dar suporte a novas funcionalidades. Tais modificações são

decorrentes dos estímulos de entrada a que são submetidos no sistema e/ou da ocorrência de suas ações internas.

Uma técnica utilizada para auxiliar os projetistas na modelagem de aplicações com comportamento modificável é a tecnologia adaptativa (NETO, 1993). A tecnologia adaptativa envolve um dispositivo não-adaptativo (subjacente) já existente em uma camada adaptativa que permite realizar mudanças no comportamento da aplicação definida. É possível citar, por exemplo, trabalhos relacionados a reconhecedores sintáticos adaptativos (NETO; KOMATSU, 1988), os *Statecharts* Adaptativos (ALMEIDA, 1995) - empregados na modelagem de sistemas reativos - e a modelagem de aplicações complexas com base no ISDL Adaptativo (CAMOLESI; NETO, 2004a). O desenvolvimento de tecnologia adaptativa aplicado a sistemas de dispositivos adaptativos dirigidos por regras vem sendo pesquisada com totais preocupações a fim de que o usuário consiga gerenciar novos dispositivos adaptativos.

Camolesi (2007) propôs um gerador de ambientes (metambiente) que possibilita a geração automática de ambientes para o projeto de aplicações adaptativas. Tal gerador fundamenta-se nos conceitos de Tecnologia Adaptativa e permite a definição de dispositivos adaptativos dirigidos por regras (NETO, 2001).

Uma das etapas necessárias para o desenvolvimento do gerador de ambientes apresentado por Camolesi (2007) é a construção de ferramentas de edição de aplicações adaptativas com base nos dispositivos gerados. O trabalho de (GÜSTACHOW, 2013) teve por objetivo a realização de uma ferramenta que permite a edição de aplicações adaptativas com base no Modelo Lógico proposto em (CAMOLESI, 2007). Tal ferramenta possibilita que os especialistas em dispositivos adaptativos avaliem os seus dispositivos estendidos para os conceitos adaptativos de uma forma rápida, porém não muito amigável. Tal ferramenta é importante, pois auxilia os especialistas em dispositivos adaptativos no desenvolvimento de seus trabalhos, uma vez que a mesma não necessita de trabalho de extensão ao ser definido um novo dispositivo.

Güstachow, 2013 sugere no seu trabalho a definição de uma linguagem com fundamentos dispositivos adaptativos para auxiliar os projetistas de aplicações a realizar suas tarefas quando utilizar-se dos conceitos de Tecnologia Adaptativa. Neste contexto é que se apresenta este trabalho que tem como foco a proposta de uma linguagem para especificações adaptativas usando conceitos de Redes de Petri Adaptativas.

A ferramenta existente para o projeto de aplicações adaptativas com base na proposta do Modelo Lógico apresentado por Camolesi, 2007 é simples para utilização, porém o uso da mesma não é amigável ao usuário. Devido a isto a produtividade dos projetistas ao

realizarem suas especificações é muito baixa. Além da produtividade destaca-se que o uso da mesma é de difícil compreensão dos elementos representados no Modelo Lógico, devido ao alto nível de abstração da representação obtida. Ao utilizar tal ferramenta, o projetista necessita inicialmente realizar a especificação de forma manual (geralmente em papel), depois realizar o mapeamento manual para o modelo lógico. A ferramenta desenvolvida por Gústachow (2013) é muito útil para especialistas em dispositivos adaptativos testarem seus novos dispositivos, porém em um ambiente profissional de desenvolvimento torna-se inviável.

Este projeto tem o propósito de disponibilizar uma linguagem de especificação para aplicações adaptativas e uma ferramenta de tradução que tem como entrada a especificação produzida e gera na saída a representação da especificação no formato do Modelo Lógico definido por (CAMOLESI, 2007).

Desenvolver e disponibilizar uma linguagem para especificação de aplicações com fundamento em Redes de Petri Adaptativas (Camolesi, 2007). Tal linguagem deve possibilitar que um projetista utilize-se, posteriormente, de um tradutor da linguagem produzida para o Modelo Lógico proposto em (CAMOLESI, 2007). Uma vez obtida a especificação da aplicação no formato do Modelo Lógico poder-se-á utilizar das ferramentas de simulação obtidas com o *framework* adaptativo para geração de ferramentas de simulação desenvolvido por (REIS, 2013) e realizar simulações das aplicações definidas.

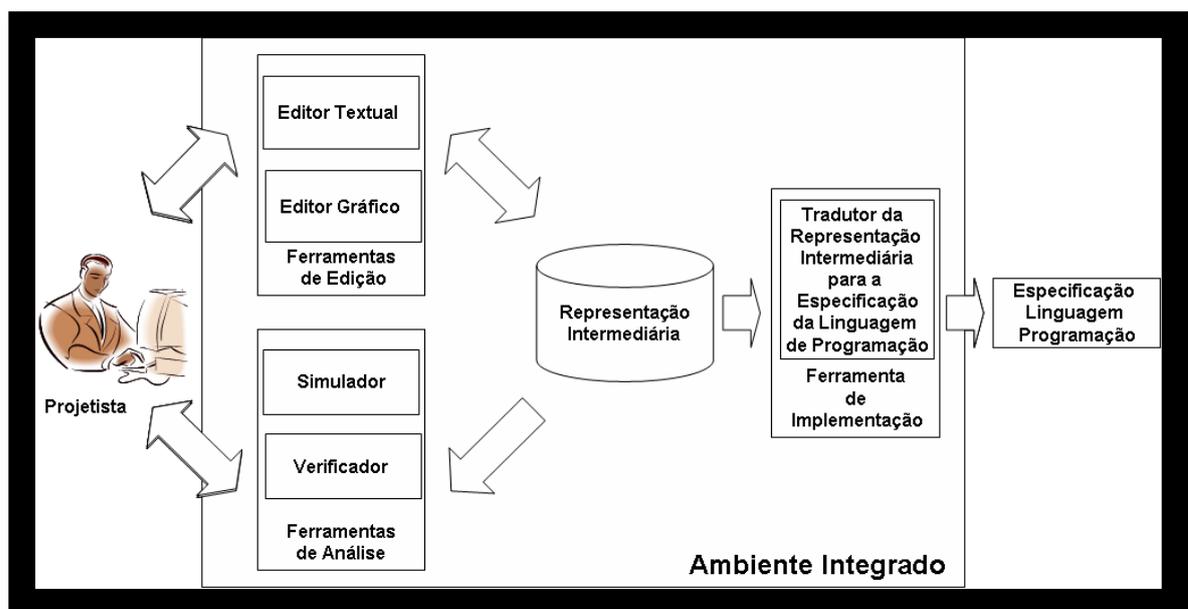
O projeto fundamenta-se na Tecnologia Adaptativa, a qual permite que um dispositivo adaptativo seja capaz de se auto modificar, ou seja, seu comportamento mudar dinamicamente em tempo de execução e sem influência externa, isso quando ele detecta uma situação em que para ele não é mais possível dar continuidade a sua execução sem antes ele alterar seu comportamento.

A maior vantagem dos dispositivos adaptativos é a sua facilidade de uso, sua relativa simplicidade e o fato de ser muito rara a necessidade de existir uma descrição completamente especificada do comportamento do dispositivo para que ele possa ser utilizado. Em lugar disso, sua operação pode ser descrita de forma incremental, e seu comportamento, programado para se alterar dinamicamente em resposta aos estímulos de entrada recebidos. Adicionalmente, como o conjunto de regras que definem o seu comportamento também se altera ao longo da sua operação, pode-se dizer que o próprio

dispositivo adaptativo programa de alguma forma a representação do conhecimento adquirido através de sua sequencia recebida de estímulos de entrada.

Dispositivos adaptativos podem ser empregados em uma grande variedade de situações, principalmente em segmentos complexos da solução de um problema, no qual sejam realizadas tomadas não triviais de decisão.

Para auxiliar o projeto de aplicações que use tecnologia adaptativa, faz-se necessária a utilização de um ambiente que integra um conjunto de ferramentas que dê suporte aos projetistas na realização de seu trabalho. A Figura 1 (CAMOLESI, 2007) ilustra a arquitetura geral de tal ambiente, que é organizado em 3 (três) grupos de ferramentas: Ferramentas de Edição, Ferramentas de Análise e Ferramenta de Implementação.



**Figura 1. Ambiente para o projeto de aplicações usando Tecnologia Adaptativa.**

Valendo-se das Ferramentas de Edição, um projetista de aplicações utilizando-se de um dispositivo adaptativo específico poderá realizar a especificação de sua aplicação com o auxílio de um editor de texto qualquer ou de um editor gráfico. Para possibilitar o intercâmbio das especificações produzidas, os editores devem gerar objetos no Modelo Lógico. Caso a especificação seja produzida em um editor textual, esta deverá ser compilada para transformar a codificação realizada no formato definido para o modelo lógico. Depois de realizada a especificação, o projetista de aplicações poderá utilizar as Ferramentas de Análise. Tais ferramentas utilizam a codificação da especificação (no

formato do modelo lógico) como base e permitem a realização de uma análise do comportamento da aplicação adaptativa em desenvolvimento. Por fim, depois de especificada e analisada a representação de uma aplicação, o projetista pode se utilizar das Ferramentas de Produção de Representações Físicas e gerar uma representação de uma aplicação em um determinado padrão de linguagem de representação física e obter a aplicação desejada.

A partir de todos os fundamentos e técnicas estudadas, foi desenvolvido um analisador léxico para uma linguagem de rede de petri adaptativa, tal analisador foi implementado utilizando o *framework JFlex*.

Primeiramente foi definido a linguagem , onde Pn são os estados, Am os arcos e Tn as transições, onde n é o número dos estados ou das transições e m o peso de cada arco, os caracteres ( , ) , ->, e ; são os separadores, a figura 2 ilustra a linguagem escrita.

```
(p0)(a1)->(t0)->(a1)(p1);
(p1)(a1)->(t1)->(a1)(p3);
(p3)(a1)->(t2)->(a1)(p5);
//Podemos inserir comentários desta forma
(p0)(a1)->(t0)->(a1)(p2);
(p2)(a1)->(t3)->(a1)(p4);
(p4)(a1)->(t4)->(a1)(p5);
```

**Figura 2. Linguagem definida para Rede de Petri adaptativa.**

A linguagem descrita anteriormente, quando analisada, separa em *tokens ARCO, ESTADO, TRANSICAO, ERROR, SEPARADOR, COMENTARIO*.

A figura 3 mostra a classe onde é definido o que é cada *token*.

```
package analisador;

import static analisador.Token.*;
%%
%class Lexer
%type Token

P = [p|P]
A = [a|A]
T = [t|T]
B = [/]
H = ([\])
I = ([\()])
C = ([ ]*->[ ]*)
```

```

COMENT = ([A-Z a-z0-9 ]+)

L = [a-zA-Z ]
D = [0-9]
WHITE=[ \t\r\n]
%{
public String lexeme;
}%
%%
{WHITE} {/*Ignore*/}

/*Separadores */
(";" | ":" | "[" | "]" | "{" | "}" | {C})    {lexeme = yytext(); return
SEPARADOR;}

/* Comentarios */
{B}{B}{COMENT} {lexeme = yytext(); return COMENTARIO;}

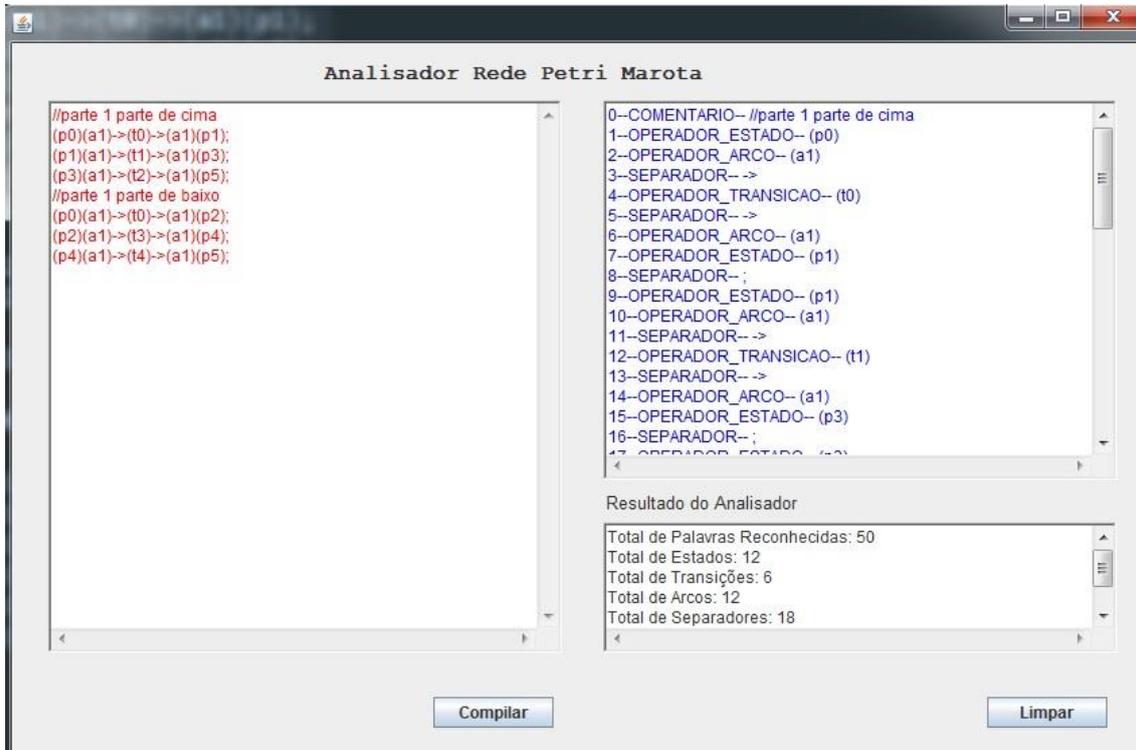
{I}{P}{D}+{H} {lexeme = yytext(); return ESTADO;}
{I}{T}{D}+{H} {lexeme = yytext(); return TRANSICAO;}
{I}{A}{D}+{H} {lexeme = yytext(); return ARCO;}

_ {return ERROR;}

```

**Figura 3. Expressão regular para definição de cada token**

A figura 4 é o analisador pronto com toda a análise feita, identificando cada parte do código escrito.



**Figura 4. Interface do analisador e uma análise realizada**

Com base na no analisador desenvolvido e nas técnicas estudadas, podemos concluir que a tecnologia adaptativa é uma área muito abrangente e tem muito o que ser estudado, já a rede de petri é o contrário, além de uma área já muito estudada porém pouco utilizada no desenvolvimento de sistemas, com ela podemos prever várias falhas que em alguns testes não encontramos, e tais falhas são descobertas antes do desenvolvimento do sistema.

## REFERENCIA BIBLIOGRÁFICA

ALMEIDA, J.R. STAD - Uma ferramenta para representação e simulação de sistemas através de statecharts adaptativos. Tese de Doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo, 1995.

CAMOLESI, A.R. e NETO, J.J. Modelagem Adaptativa de Aplicações Complexas. XXX Conferencia Latinoamericana de Informática - CLEI'04. Arequipa - Peru, Setiembre 27 - Octubre 1, 2004.

CAMOLESI, A.R. Proposta de um gerador de ambientes para a modelagem de aplicações usando Tecnologia Adaptativa. Tese de Doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

GÜTSCHOW. W.E. Sistema de Gestão de Formalismos e de Projetos Adaptativos, Exame de Qualificação - Trabalho de Conclusão de Cursos do Curso Bacharelado em Ciência da Computação (IMESA-FEMA), Assis/SP, Julho, 2013.

NETO, J. J. *Adaptive Rule-Driven Devices - General Formulation and Case Study*. Lecture Notes in Computer Science. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata 6th International Conference, CIAA 2001, Springer-Verlag, Vol.2494, pp. 234-250, Pretoria, South Africa, July 23-25, 2001.

NETO, J. J. *Contribuições à metodologia de construção de compiladores*. Tese de Livre Docência, USP, São Paulo, 1993.

NETO, J.J.; KOMATSU, W. *Compilador de Gramáticas Descritas na Notação de Wirth Modificada*. Anais EPUSP - Engenharia de Eletricidade - série B, vol. 1, pp. 477-517, São Paulo, 1988.

REIS, F.M.B. *Framework para simulação e verificação de aplicações adaptativas*. Relatório Final Projeto PIBIC edital (IMESA-FEMA), IMESA-FEMA, Assis/SP, Agosto, 2013.