

## PROGRAMAÇÃO LÓGICA E ENGENHARIA DE SOFTWARE

Teresa Cristina de Aguiar

COPPE - Engenharia de Sistemas e Computação  
Universidade Federal do Rio de Janeiro  
Caixa Postal 68511  
21945 - Rio de Janeiro - RJ

### RESUMO

Este artigo apresenta duas aplicações da programação lógica em Engenharia de Software: prototipagem rápida e sistemas especialistas que auxiliam no processo de desenvolvimento de software. Tem como objetivo discutir o potencial da lógica e sua adequação a estas duas aplicações.

### 1. INTRODUÇÃO

O interesse pela programação lógica tem crescido nos últimos anos devido principalmente ao fato da lógica simbólica estar sendo utilizada no projeto Japonês de quinta geração. No entanto, antes disso, já se percebia que o formalismo da lógica poderia dar uma significativa contribuição à teoria e prática da computação. [HOGGER]

A diferença fundamental entre a programação lógica e a convencional é que a programação lógica requer a descrição da estrutura lógica de problemas, enquanto que na programação convencional há necessidade de se descrever como o computador deve agir para resolver um problema. Esta separação entre lógica e controle, proposta pela programação lógica, possibilita, segundo [KOWALSKI-79], que se alcancem alguns dos objetivos da Engenharia de Software: a construção de programas mais confiáveis, mais fáceis de serem manipulados e mais fáceis de se adaptarem a novos problemas. Para alcançar estes objetivos a Engenharia de Software tem se empenhado em melhorar o processo

de desenvolvimento de software, utilizando recursos como os da programação lógica.

Este artigo apresenta duas aplicações da programação lógica em Engenharia de Software: prototipagem rápida e sistemas especialistas que auxiliam no processo de desenvolvimento de software. Tem como objetivo discutir o potencial da lógica e sua adequação a estas duas aplicações.

Este artigo é parte de um trabalho que vem sendo realizado pela linha de pesquisa Engenharia de Software, da COPPE-UFRJ, com os seguintes objetivos: a) utilizar os recursos da Inteligência Artificial para auxiliar o processo de desenvolvimento de software, gerando, como consequência, produtos de alta qualidade, e b) dar o apoio da Engenharia de Software necessário à Inteligência Artificial.

## 2. APLICAÇÕES DA PROGRAMAÇÃO LÓGICA NA ENGENHARIA DE SOFTWARE

### 2.1- PROTOTIPAGEM RÁPIDA

Já não há dúvidas da necessidade de se descrever os requisitos lógicos de um sistema em sua primeira fase de desenvolvimento. Uma série de métodos têm sido criados com este objetivo. Quando a descrição dos requisitos é realizada numa linguagem lógica processável permite-se a análise e execução destes requisitos. Isso é de extrema importância, uma vez que não basta que na fase inicial de desenvolvimento de um sistema se descreva os requisitos - muitas vezes eles não expressam o desejado, assim como podem apresentar inconsistências. Esta capacidade de execução torna as linguagens lógicas ferramentas adequadas para construção de protótipos. Possibilita que antes

da fase de projeto e implementação se possa verificar se a especificação corresponde à visão que o usuário tem do sistema. Este tipo de verificação na fase inicial do desenvolvimento diminui bastante os custos do projeto.

A adoção desta estratégia implica na modificação do ciclo tradicional de desenvolvimento de software (especificação, codificação, testes, reformulação, recodificação e operação).

A nova estratégia propõe os seguintes passos:

1. Especificação inicial - É definida uma especificação parcial.

2. Testes interpretativos - A especificação (que ainda não é um programa) é testada através de interpretação, utilizando-se uma máquina de inferência.

3. Reformulação incremental - Se o comportamento mostra-se incompleto ou incorreto, a especificação é revisada e continua-se a partir do segundo passo.

4. Codificação - Se a especificação está correta, continua-se com o tradicional desenvolvimento de aplicações.

A codificação pode ser dispensada em certos casos, quando desejado, pois a análise executável tem a possibilidade de produzir diretamente o comportamento desejado para o sistema.

Considerando que a prototipagem rápida vem sendo desenvolvida com o uso de técnicas tradicionais, qual a vantagem de se utilizar a programação lógica nesta área? Os protótipos são geralmente elaborados na mesma linguagem de programação que é utilizada para a implementação. Empregando a programação lógica é possível implementar protótipos através de linguagens semanticamente equivalentes a diagramas de fluxo de dados [KOWALSKI-84]. [SIMPLICIO] descreve uma ferramenta capaz de

executar especificações de requisitos do sistema através da lógica.

## 2.2- SISTEMAS ESPECIALISTAS

A tecnologia de sistemas especialistas vem contribuindo para solucionar uma série de falhas que podem ocorrer com os especialistas de uma área: permite que o conhecimento, uma vez adquirido, não seja esquecido; facilita a transferência de conhecimentos; possibilita a geração de resultados mais consistentes, uma vez que o sistema especialista não está sob o efeito de fatores emocionais; a geração da documentação é facilitada, já que existe um mapeamento entre a forma que o conhecimento é representado no sistema e a descrição em linguagem natural desta representação. Os sistemas especialistas têm ainda a grande vantagem do baixo custo de operação, apesar do custo de desenvolvimento ser ainda bastante alto.

O desenvolvimento de sistemas especialistas de apoio à Engenharia de Software tem como objetivo armazenar o conhecimento de especialistas em algumas das etapas do processo de desenvolvimento de software, de forma a dar assistência aos especialistas ou prover informações ou auxílio àqueles que não têm acesso a um especialista da área. A lógica é uma das técnicas possíveis de serem utilizadas para representar o conhecimento de um sistema especialista.

Têm sido desenvolvidos sistemas especialistas em lógica com os seguintes objetivos:

- Possibilitar que programas sejam escritos numa linguagem de tal forma que permitam uma verificação e manutenção eficientes, além de utilizar de forma mais eficiente os recursos

que a linguagem de programação dispõe. O sistema especialista auxiliaria o usuário na programação e no uso mais eficiente da linguagem, dispensando-o de maiores treinamentos e da consulta a manuais. Um sistema especialista com este objetivo é apresentado em [ROMAGNOLI].

- Auxiliar o desenvolvimento da especificação de projeto, incorporando o conhecimento de um método para desenvolver este tipo de especificação. Um sistema com este objetivo é apresentado em [PESSOA].

### 3. LÓGICA E SUAS APLICAÇÕES

Segundo [BALDWIN] um sistema lógico tem vantagens importantes: a) o poder de expressão; b) as regras de inferência são corretas e completas, apesar de na prática muitos sistemas modelados não utilizarem o conjunto completo de regras de inferência e não exibirem todas as características de completeza.

No entanto o poder expressivo das fórmulas bem formadas e as características próprias das regras de inferência resultam em sérios problemas. Um deles é o longo tempo para construção de uma prova a partir de um objetivo e de premissas iniciais. Não existe um algoritmo para construir uma prova num tempo razoável, o que torna a lógica difícil de ser utilizada em grandes domínios. Outros problemas resultam do rigor na utilização das regras de inferência. Estas regras não permitem um raciocínio baseado em suposição ou por "default". O raciocínio baseado em probabilidade também não é permitido. É difícil, da mesma forma, utilizar a lógica em domínios nos quais proposições são verdadeiras somente num determinado momento. Para manipular proposições com um componente tempo, ou baseadas em probabilidade, há necessidade de se acrescentar mais fórmulas bem

formadas, aumentando-se o tempo de busca.

Apesar das limitações, as linguagens lógicas possuem um potencial que ainda não foi totalmente explorado. Há necessidade, portanto, de desenvolverem-se mecanismos que melhorem seu desempenho, o que já vem sendo realizado, uma vez que estão em fase de desenvolvimento uma série de trabalhos com este objetivo. Um deles é apresentado em [CASANOVA].

A partir da análise das aplicações apresentadas verifica-se que o gerador de protótipos é a que melhor justifica a utilização de uma linguagem lógica, uma vez que permite que se explore a semelhança entre a forma de se especificar e a forma de representação das linguagens lógicas. Além do fato de que, ao se utilizar uma linguagem processável, permite-se a imediata execução do que foi especificado, sem a necessidade de se utilizar outra linguagem, simplificando-se muito o problema de prova de correção.

Com relação ao desenvolvimento de sistemas especialistas não fica clara a vantagem da utilização da lógica. Existem outros métodos para o desenvolvimento desses sistemas e seria interessante se aqueles que optam pela lógica justificassem a escolha, analisando a possibilidade de utilização de outros métodos.

Percebe-se, no entanto, que de uma forma geral, a escolha das ferramentas utilizadas para o desenvolvimento de sistemas especialistas não tem sido realizada de maneira muito correta. Segundo [WATERMAN], muitos sistemas especialistas foram construídos com ferramentas escolhidas por motivos como: a) familiaridade do engenheiro do conhecimento com a ferramenta. b)

dentre as ferramentas compatíveis com o hardware do desenvolvedor, a escolhida era a mais eficiente; c) a ferramenta foi desenvolvida e depois foram escolhidas aplicações para testá-la.

Com relação ao desenvolvimento de sistemas especialistas, [KOWALSKI-83] afirma que PROLOG é uma excelente ferramenta de implementação. O que se pode dizer, independente de se utilizar uma linguagem lógica ou não para implementação, é que há certas dificuldades em se desenvolver sistemas especialistas. Primeiro, a dificuldade em se reunir um grupo de desenvolvimento competente, devido à falta de recursos financeiros. Segundo, as limitações atuais inerentes à tecnologia de Inteligência Artificial. E por fim o longo tempo necessário ao desenvolvimento.

Além destes problemas, um outro surge como uma séria dificuldade: a aquisição do conhecimento. Caso o método não seja descrito de uma maneira formal, a única forma de se adquirir o conhecimento sobre ele é através de consultas a um especialista, o que pode se tornar uma árdua tarefa.

É importante que antes de se iniciar a construção de um sistema especialista, seja realizada uma avaliação da viabilidade de sua construção. Segundo [WATERMAN], esta avaliação implica em se determinar não só se é possível construí-lo, mas também se é apropriado e justificado. Por não ser realizada esta avaliação têm sido desenvolvidos muitos sistemas que não podem ser chamados de especialistas, apesar de receberem esta denominação.

Com relação à aplicação de sistemas especialistas à Engenharia de Software, há ainda o problema do conhecimento em muitas áreas da Engenharia de Software não estar suficientemente

sedimentado, o que limita a construção de sistemas especialistas fora do ambiente acadêmico.

#### 4. CONCLUSÃO

A lógica possui sérias restrições. No entanto estas limitações não impedem que as linguagens lógicas se tornem ferramentas eficientes em certos domínios de problemas, podendo tornar a computação mais eficiente. Quanto à Engenharia de Software, é uma área que como outras, pode vir a se beneficiar dessa nova tecnologia de software. Mas cabe também à Engenharia de Software o importante papel de fornecer, não só à área de linguagens lógicas como à Inteligência Artificial como um todo, recursos que aumentem a qualidade dos programas gerados e do processo de desenvolvimento de sistemas.

#### REFERÊNCIAS

[BALDWIN] D. Baldwin, G.M.Kasper: Toward Representing Management-domain Knowledge; Decision Support Systems 2, 159-172; North-Holland; 1986.

[CASANOVA] M.A.Casanova, M.E.M.T.Walter: Implementação de Interpretadores PROLOG Combinando Expansão em Profundidade com Expansão em Amplitude; Anais do 3o Simpósio Brasileiro de Inteligência Artificial; Rio de Janeiro; Novembro 1986.

[DOYLE] Jon Doyle; Expert Systems and the "Mith" of Symbolic Reasoning; IEEE Transaction on Software Engineering; Vol. SE-11; Num. 11; Novembro 1985.

[HOGGER] C.J. Hogger; Introduction to Logic Programming; Academic

Press, Inc; 1984.

[KOWALSKI-79] R.Kowalski; Algorithm = Logic + Control; Communications of the ACM; Vol. 22; Num 7; Julho 1979.

[KOWALSKI-83] R.Kowalski; Logic Programming; Information Processing 83, R.E.A. Mason (ed); Elsevier Science Publishers B.V.; North-Holland; 1983.

[KOWALSKI-84] R. Kowalski; AI And Software Engineering; Datamation; vol. 2; 1984.

[MELLISH] W.F.Clocksln, C.S.Mellish; Programming in Prolog, Springer-Verlag; 1984.

[PESSOA] T.E.C.Pessoa, R.C.B.Martins, D.Shwabe; Um Sistema Especialista para o Método de Jackson; Anales de las 16as Jornadas Argentinas de Informática e Investigación Operativa; Buenos Aires; 1986.

[RICH] E. Rich; Artificial Intelligence; International Student Edition; Mc Graw-Hill Book Company; 1986.

[ROBINSON] J.A.Robinson; Logic Programming - Past, Present and Future; New Generation Computing; OHMSHA LTD; 1983.

[ROMAGNOLI] G.S.Romagnoli, M.T.M.P.Silva, M.C.Filho; Um Sistema Especialista para Programação em NATURAL; Anais do 3o Simpósio Brasileiro de Inteligência Artificial; Rio de Janeiro; Novembro 1986.

[SIMPLÍCIO] F.C.Simplício Filho; Um Método de Análise por Protótipos Baseado em Representação de Conhecimentos; Tese de Mestrado; PUC - RJ; Agosto 1986.

[WATERMAN] Donald A. Waterman; A Guide to Expert Systems, Addison-Wesley Publishing Company; 1985.