

# 7

## O Pronome ALL

Neste capítulo, é explorado o pronome ALL. Na Seção 7.1, o conjunto de todos os objetos do sistema é explicado. Na Seção 7.2, são mostradas as transformações de código necessárias para que o pronome ALL seja corretamente executado. Na Seção 7.3, é apresentado o nível Meta necessário para a obtenção destas transformações. Finalmente, na Seção 7.4, a implementação de um sistema com Invocação Implícita [SN92], descrito na Subseção 2.4.4, utilizando o pronome ALL é comparada com a implementação tradicional.

### 7.1

#### O Conjunto de Todos os Objetos do Sistema

Um sistema orientado a objetos é composto por objetos que interagem através de troca de mensagens. Esta troca se dá sempre de um objeto simples para outro, sendo necessário que o envio de uma mensagem a mais de um objeto seja explicitamente programado.

Um grupo de objetos é particularmente interessante para o envio de mensagens: o conjunto formado por todos os objetos do sistema. Criado na inicialização do sistema como um conjunto vazio, este é acrescido quando um objeto é criado e decrescido quando um é destruído. Poder enviar mensagens a este grupo significa poder implementar mensagens de *broadcast* que são utilizadas um grande número de arquiteturas e algoritmos. O pronome ALL pode ser usado para representar este conjunto.

Mensagens enviadas através do pronome ALL são oferecidas a todos os objetos do sistema, sendo, no entanto, tratadas apenas por aqueles que se interessarem por elas. Em outras palavras, por aqueles que disponibilizarem um tratador adequado para elas.

A Figura 7.1 mostra o cenário no nível base de três classes, onde duas utilizam o pronome ALL. As classes *B* e *C*, em algum de seus métodos, enviam respectivamente as mensagens *n* e *m* através do pronome e declaram que o fazem através do especificador <<message>> introduzido na linguagem.

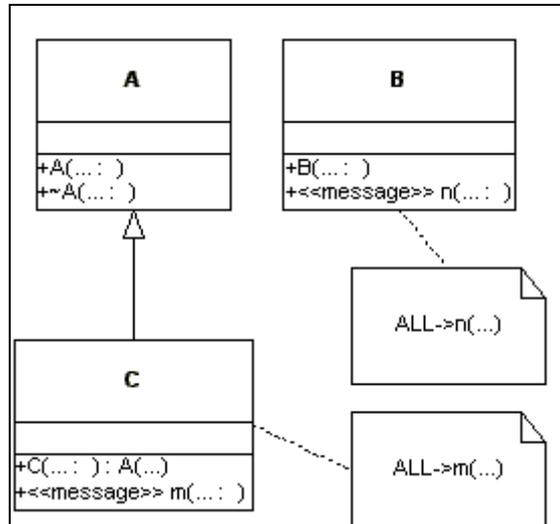


Figura 7.1 – Cenário de utilização do pronome ALL em nível base

## 7.2

### Transformações de Código

O cenário anterior deve ser transformado de modo que:

1. as classes *A*, *B* e *C* possuam uma superclasse que implemente de forma vazia um tratador para *m()* e para *n()*; e
2. todos os objetos do sistema devem ser acessíveis no momento em que uma mensagem a eles seja enviada.

Para a primeira transformação, é utilizada a mesma estratégia descrita para os pronomes anteriores.

Para a segunda transformação, todo objeto, quando criado, precisa ter sua referência armazenada em um repositório que seja acessível por todos os objetos do sistema. Além disso, todo objeto, quando destruído, precisa ter sua referência removida do mesmo repositório. Este repositório é implementado como um *singleton* [G+95] provido como suporte de tempo de execução. Um

comando de inserção neste é introduzido nos construtores das classes raízes das árvores hierárquicas e um comando de exclusão introduzido nos destrutores das mesmas classes. A restrição das classes onde são feitos os comandos de inserção e remoção é para garantir que um objeto seja inserido, ou removido, apenas uma vez, já que para que um objeto seja criado, todos os construtores de suas superclasses são executados.

O envio é transformado em um envio múltiplo para todos os objetos contidos no repositório. É interessante que a ordem de acesso a estes objetos, disponibilizada pelo repositório, não seja sempre a mesma para que as mensagens não sejam enviadas sempre aos mesmos objetos primeiro.

A Figura 7.2 mostra o cenário transformado.

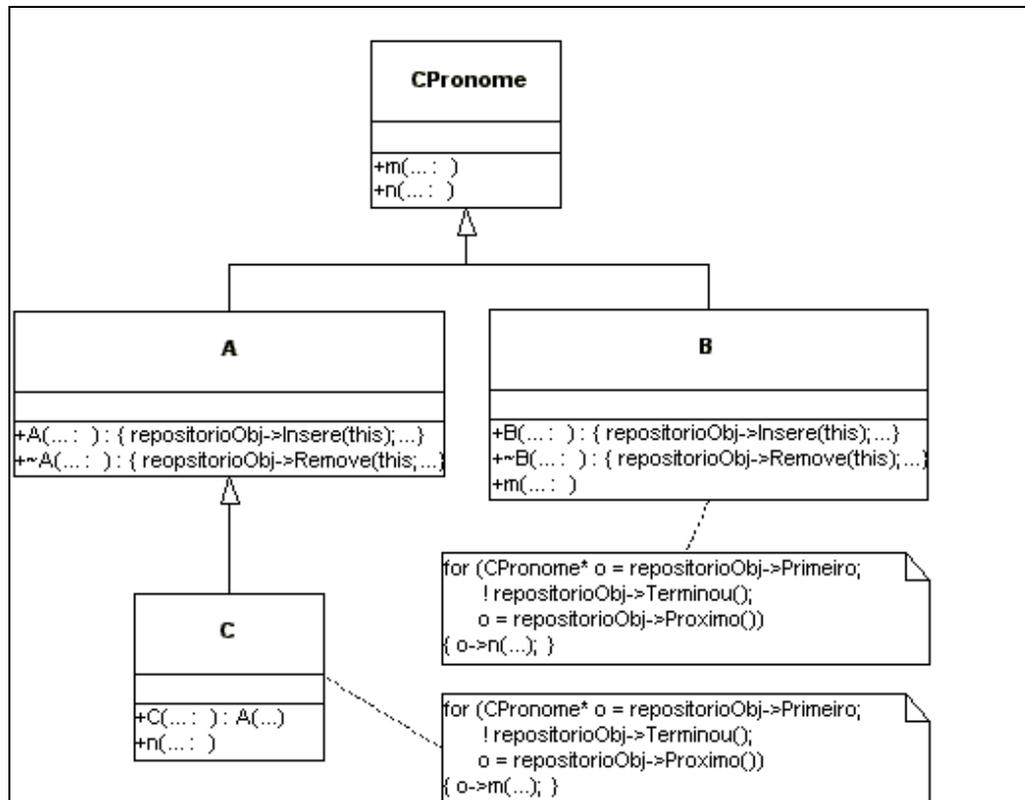


Figura 7.2 – Cenário de utilização do pronome ALL transformado

### 7.3

#### O Nível Meta

Para obter estas transformações, a todas as classes do nível base é associada a metaclasses *MCAII*, a menos da classe *CPronome* que é associada a

uma metaclassa especial denominada *MCRaizPronome*. O nível meta e sua associação com o nível base mostrado anteriormente são detalhados na Figura 7.3.

Todos os métodos utilizados na implementação deste pronome já foram descritos nos capítulos anteriores não o sendo, portanto, novamente aqui.

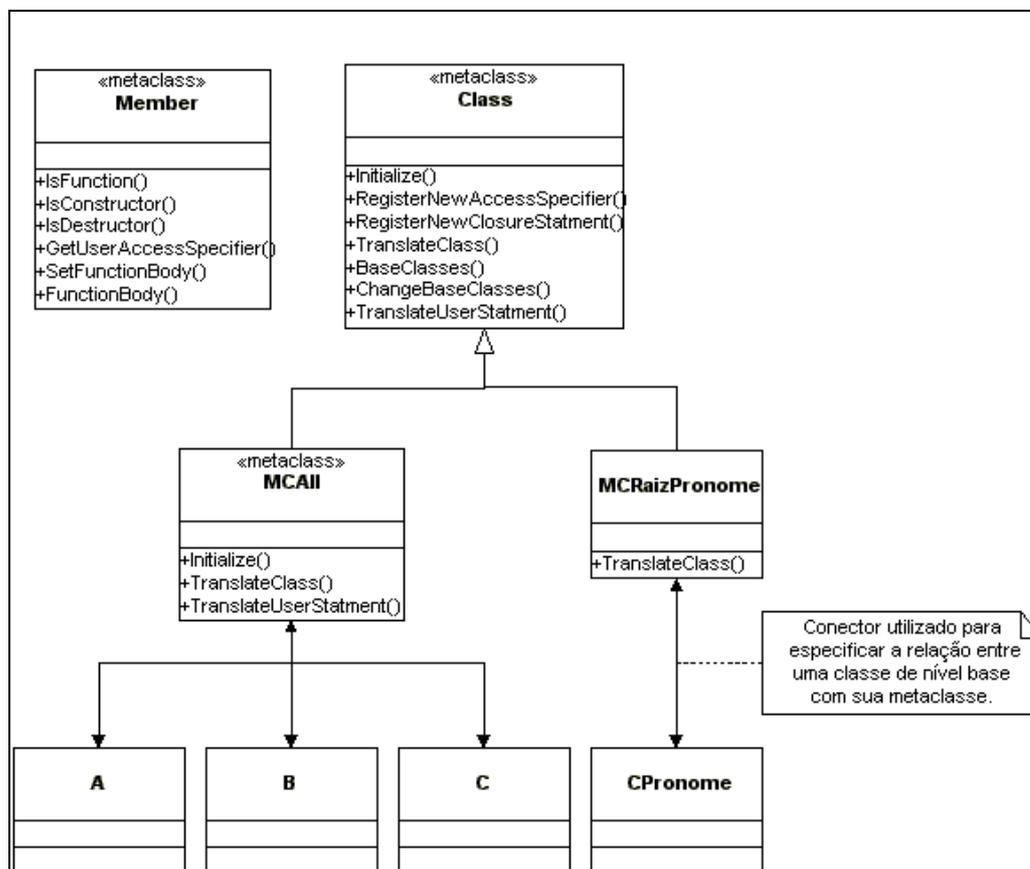


Figura 7.3 – Nível Meta para implementação do pronome ALL

Para que o especificador (message) e o novo comando (envio de mensagem para o ALL) seja identificado, a metaclassa *MCAII* redefine o método *Initialize()* e faz os dois registros:

- RegisterNewAccessSpecifier("message")
- RegisterNewClosureStatement("ALL")

As inserções da nova herança, a inserção no repositório de objetos feita nos construtores das classes raízes de árvores hierárquicas, a remoção do mesmo repositório feita nos destrutores das mesmas classes e o armazenamento das mensagens enviadas através do pronome são feitos na redefinição do método *TranslateClass()*.

Para o armazenamento das mensagens enviadas através do pronome, foi utilizado o objeto *Singleton* [G+95] *RepositorioMensagens* que armazena as mensagens declaradas com o especificador <<message>> e as disponibiliza para ser introduzidas na classe *CPronome*.

A transformação do envio efetivo da mensagem através do pronome ALL é feita redefinindo-se o método *TranslateUserStatment()* na classe *MCAII*. Este substitui o envio único de mensagem original pelo envio múltiplo da mesma mensagem para todos os objetos existentes no repositório de objetos.

Ao término da transformação da última classe associada a *MCAII*, todas as mensagens enviadas através do pronome ALL estão armazenadas.

A metaclasses *MCRaizPronome* redefine o método *TranslateClass()* e, neste, insere na classe *CPronome* uma função membro para cada mensagem armazenada no repositório, sempre com implementações vazias.

## 7.4

### Invocação Implícita Utilizando ALL

A técnica de integração referenciada como integração implícita [SN92], descrito na Subseção 2.4.4, pode ser implementada utilizando-se o pronome ALL de forma bastante direta. O anúncio de um evento é feito através do pronome ALL e os objetos que neste estiverem interessados disponibilizam um tratador adequado para tratá-los.

Os tempos de execução da implementação da técnica através de um *framework* de aplicação [SN92], utilizando o padrão *Observer* [G+95] e um mediador de eventos como *Singleton* [G+95] para captar eventos anunciados e despachá-los para os objetos interessados, e da implementação utilizando o pronome ALL podem ser comparados na Tabela 7.1 e no gráfico mostrado na Figura 7.4.

Estes tempos foram conseguidos executando-se o padrão nas duas implementações e variando-se o número de objetos do sistema não interessados no evento. Isto é, o número de objetos interessados no evento, em cujo comportamento a implementação tradicional já insere código a ser executado, permanece inalterado, sendo variado apenas o número de objetos não interessados no evento, em cujo comportamento a implementação com o pronome insere código que será executado inutilmente.

Da mesma forma que nos pronomes descritos anteriormente, cada configuração do exemplo foi executada 1000 vezes, sendo os tempos mostrados na Tabela 7.1 a média destas execuções. Os desvios padrões das amostras consideradas são apresentados na Tabela 7.2. O ambiente de execução destes experimentos foi, também, o mesmo dos pronomes anteriores. Isto é, um computador Pentium4, com 256 MB de RAM, utilizando-se Linux Mandrake 8.1.

Número de objetos não interessados no evento	Implementação através de <i>framework</i> de aplicação ( $\mu$ s)	Implementação com MAIN ( $\mu$ s)	Diferença ( $\mu$ s)	Aumento Percentual
0	1750	1761	11	0,6285
1	1763	1822	59	3,3465
10	1923	2002	79	4,1081
100	2347	2427	80	3,4086
1000	6406	6749	343	5,3543
10000	61500	65048	3548	5,7691
100000	506687	539645	32958	6,5046
1000000	2644458	2774234	129776	4,9074
10000000	13332444	13992277	659833	4,9490
100000000	77773405	81183752	3410347	4,3849
1000000000	622888224	646051127	23162903	3,7186
10000000000	3310562284	3442488002	131925718	3,9849

Tabela 7.1 – Tempos de execução da Invocação Implícita

Número de acessos ao <i>Singleton</i>	Desvios Padrões Implementação Padrão	Desvios Padrões Implementação com MAIN
0	0,2309	0,2172
1	0,2850	0,2374
10	0,2173	0,3106
100	0,3297	0,3153
1000	0,3107	0,2139
10000	0,2853	0,1830
100000	0,1983	0,2182
1000000	0,3083	0,2309
10000000	0,4290	0,2180
100000000	0,1294	0,3183
1000000000	0,2409	0,2951
10000000000	0,2273	0,2906

Tabela 7.2 – Desvios Padrões das amostras de tempo utilizadas para análise da execução da Invocação Implícita

Os dados apresentados na Tabela 7.1 são melhor visualizados no gráfico da Figura 7.4.

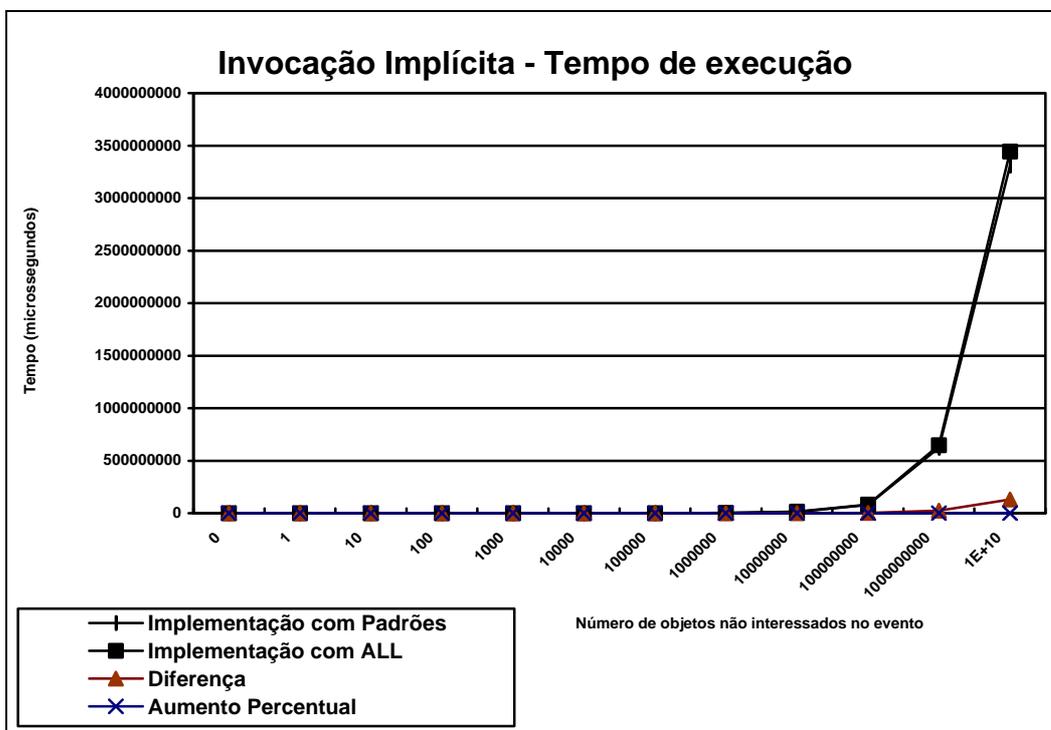


Figura 7.4 – Gráfico de comparação entre os tempos de execução da Invocação Implícita

Na Tabela 7.1 e no gráfico mostrado na Figura 7.4 pode-se verificar que o tempo de execução na implementação do padrão utilizando-se o pronome ALL é sempre um pouco maior, o que já era esperado. O aumento percentual máximo obtido, no entanto, foi de 5%, evidenciando a não degradação do sistema com o uso do pronome.