ANAIS

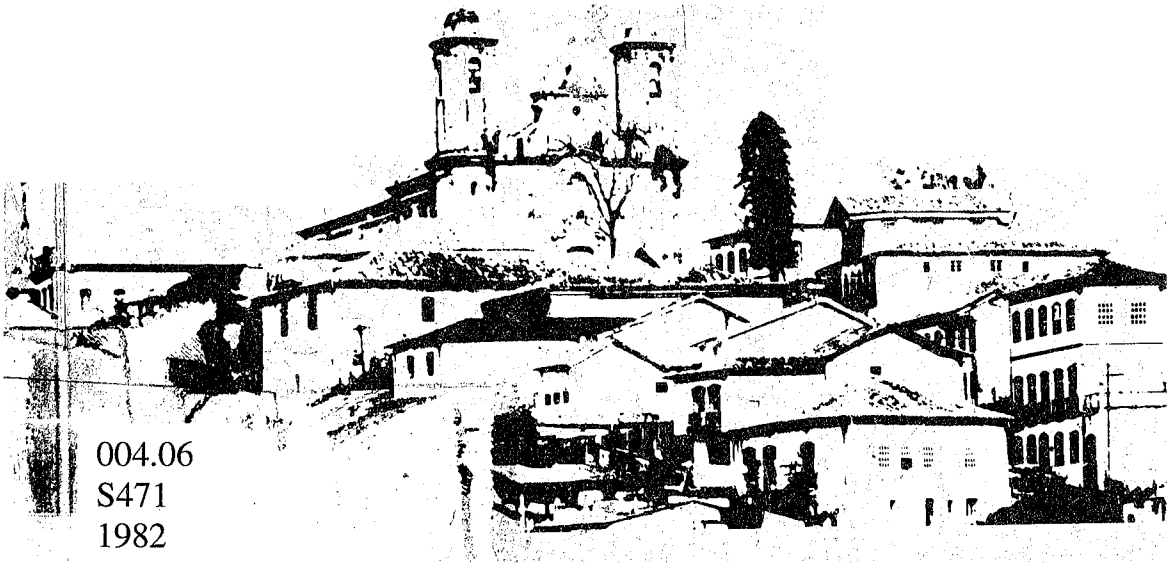
Volume I

Trabalhos apresentados no

IX Seminário Integrado de

"Software" e "Hardware" - SEMISH

EDITORES - L. J. BRAGA-FILHO, E. G. DE SIMONE E N. MEISEL



SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

DIRETORIA

Presidente: Luiz de Castro Martins
Vice-Presidente: Sílvio Davi Paciornick
Secretário-Geral: Sueli Mendes dos Santos
1º Secretário: Estevam Gilberto de Simone
2º Secretário: Ivan Moura Campos
Tesoureiro; Therezinha da Costa Ferreira Chaves

CONSELHO

Carlos Inácio Zamitti Mammana
Carlos José Pereira de Lucena
Cláudio Zamitti Mammana
Clésio Saraiva dos Santos
Henrique Pacca Luna
Ivan da Costa Marques
João Antônio Zuffo
Luiz Julião Braga-Filho
Mário Dias Ripper
Wilson de Pádua Paula Filho

A SBC tem como finalidade:

a) Incentivar atividades de ensino, pesquisa e desenvolvimento em computação no Brasil; b) zelar pela preservação e aprimoramento do espírito crítico, responsabilidade profissional e personalidade nacional da comunidade técnico-científia que atua no setor de computação no país; c) ficar permanentemente atenta à política governamental que afeta as atividades de computação no Brasil, no sentido de assegurar a emancipação tecnológica do país; d) especificamente, e enquanto for de interesse da sociedade, promover um seminário integrado de "software" e "hardware" nacionais; e) promover, por todos os meios, academicamente legítimos, através de reuniões, congressos, conferências e publicações, o conhecimento, informações e opiniões que tenham por objetivo a divulgação da ciência e os interesses da comunidade de computação.

Maiores informações: Av. Venceslau Brás, 71 Fundos Casa 27
22290 - Rio de Janeiro, RJ

II CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

UNIVERSIDADE FEDERAL DE OURO PRETO

12 a 16 de julho de 1982

ANAIS

VOLUME I - TRABALHOS APRESENTADOS NO IX SEMINÁRIO INTEGRADO DE "SOFTWARE" E "HARDWARE" - SEMISH

Editores: L.J. Braga Filho, E.G. de Simone e N. Meisel

Pág. 455 a 462

DUALIDADE E A CONSTRUÇÃO DE PROGRAMAS CONCORRENTES

Michael Stanton

PUC/RJ - Departamento de Informática
R. Marquês de S. Vicente, 225
22453 - Rio de Janeiro, RJ

SUMÁRIO

O objetivo principal deste trabalho é demonstrar a grande utilidade da dualidade entre o uso de monitores, e o uso de troca de mensagens na construção de programas concorrentes. Dualidade é vista como a ferramenta que permite o projetista de um programa concorrente escolher o estilo mais útil para facilitar sua compreensão e intuição, com a finalidade de escolher uma estrutura. A estrutura resultante, por dualidade, poderá ser transformada para o outro estilo para implementação. É mostrada a aplicação da ferramenta a uns exemplos.

ABSTRACT

The main aim of this report is to show how the design of concurrent programs may be assisted by the aid of the duality which exists between the use of monitors and the exchange of messages. Duality is seen as a tool to be used by a concurrent program designer, and which allows him to select the programming style most likely to aid his understanding and intuition, in order to arrive at an appropriate structure. Using duality, this structure may then be transformed into the other style for implementation. A real application is discussed.

1. INTRODUÇÃO

No livro, "The Architecture of Concurrent Programs" [BH77,p192] Brinch Hansen discutiu o problema da estruturação de programas concorrentes, e propôs um roteiro para identificar uma estrutura ideal, em termos de atividades simultâneas e estruturas de dados. Mais recentemente Gentleman [GEN81] ressaltou a importância de identificar formas restritas e úteis de estruturar programas concorrentes, para que estas formas pudessem ser usadas rotineiramente como blocos básicos para a montagem de estruturas maiores. Gentleman ainda enfatizou o papel importante para a estruturação de programas concorrentes de uma visão antropomórfica, fazendo uma analogia entre conjuntos de processadores assíncronos e organizações de pessoas. Esta analogia ajuda escolher quais deverão ser as atividades paralelas e qual é a natureza das interações entre estas num programa concorrente.

Além de discussões sobre a estrutura abstrata de programas concorrentes, existem divergências quanto a sua realização. Há dois principais modelos, que são usados para combinar os conceitos de paralelismo, sincronização e comunicação, e que definem um estilo de programação. O mais conhecido utiliza monitores para controlar acesso a variáveis compartilhadas [BH77 , HOA74, LAM80 , WIR77] , enquanto o segundo modelo utiliza o movimento explícito de dados na forma de troca de mensagens [BH70, BH79, HOA78, GEN81]. Este segundo modelo tem vantagens evidentes para um ambiente de memória distribuída, embora para implementação de concorrência num ambiente de memória global qualquer dos dois modelos seja válido. Num estudo importante [LAU79], Lauer e Needham demonstraram que sob certas restrições existe uma dualidade entre os dois modelos, e que um programa realizado segundo um modelo tem seu correspondente direto segundo o outro.

O objetivo deste artigo é mostrar como a dualidade apontada por Lauer e Needham pode ser utilizada como ferramenta na estruturação de programas concorrentes, por possibilitar o projeto de uma estrutura usando um dos dois estilos mencionados, e realizá-la usando o estilo dual. Em particular, daremos exemplos de programas que foram projetados usando troca de mensagens e depois realizados usando monitores.

2. A DUALIDADE ENTRE OS DOIS MODELOS

Lauer e Needham [LAU79] fizeram uma comparação entre a estrutura de programas feitos usando os dois estilos de monitores, e de troca de mensagens. Como existe diversidade na especificação de troca de mensagens, resumimos aqui as suposições feitas por Lauer e Needham. Seu modelo admite a existência de di

versos pórticos que possam receber mensagens. O remetente envia uma mensagem a um pórtico e depois pode esperar a resposta desta mensagem. O receptor, que também é o dono dos pórticos, pode esperar receber uma mensagem através de qualquer elemento num subconjunto dos pórticos, que pode ser alterado dinâmica mente. Esta alteração dinâmica permite atrasar arbitrariamente a recepção de mensagens dirigidas a um dado pórtico. Convencionalmente neste modelo, uma mensagem é respondida antes da recepção da próxima mensagem. A relação entre o remetente e o receptor é assimétrica, e o receptor desconhece a identidade do remetente. Devemos notar ainda que quando um remetente se bloqueia esperando a resposta, este bloqueio ocorre (logicamente) na fila de mensagens no pórtico do receptor e portanto antes da recepção da mensagem.

Com estas suposições, Lauer e Needham concluiram que os dois estilos de programação usando monitores e troca de mensagens são duais sob a transforma - ção

processo	<---------->	mensagem
monitor	<---------->	processo
chamar/retornar	<---------->	enviar/responder

Isto implica que a cada programa, escrito em termos de processos(do 1º tipo)e monitores, corresponde um programa dual, escrito em termos da troca de mensagens entre processos (do 2º tipo). É interessante observar que a mesma palavra processo tradicionalmente tem sido usada com dois significados bem diferentes, como revelado pela transformação de dualidade. Esta observação talvez explique certa relutância contra a aceitação da tese de Lauer e Needham. Gentleman [GEN81], por exemplo, ao defender programação baseada na troca de mensagens, negou a relevância desta tese. Porém pretendemos mostrar que o conteúdo do seu artigo não suporta esta conclusão.

O modelo de troca de mensagens de Gentleman tem algumas diferenças daquele assumido por Lauer e Needham. Primeiro, embora continue assimétrica a relação entre remetente e receptor, o receptor passa a conhecer a identidade do remetente, para poder responder às mensagens em ordem arbitrária. Segundo, a comunicação requer a sincronização de ambas as partes, e logo, para não bloquear nunca o receptor, o envio de uma mensagem bloqueia o remetente até a chegada da resposta. Terceiro, em vez de manter filas de mensagens para pórticos diferentes, o receptor recebe todas as mensagens na ordem de sua chegada. Em particular é incomum bloquear a recepção de mensagens de uma determinada fonte, e impossível bloquear mensagens de um determinado tipo. Devemos observar aqui que estas características são as mesmas que tem um monitor, que recebe chamadas das suas entradas em ordem de chegada, mas não precisa retornar controle na mesma ordem, podendo bloquear a execução de uma entrada em qualquer ponto a-

través do comando espera. Esta execução seria retomada mais tarde quando for conveniente. Na sua discussão, Lauer e Needham reconhecem que esta característica de um monitor não tem correspondente no seu modelo de troca de mensagens . Na opinião deste autor, o modelo de Gentleman corresponde neste ponto. Embora um programa no estilo de Gentleman não seja transformado para uso de monitores por troca de certas palavras chave [LAU79 p14], a transformação não é muito mais complexa. Em particular nos exemplos de administradores dados no artigo de Gentleman, o comando receber é sempre seguido de um comando de casos, que substitui o comando de casos de pórticos de Lauer e Needham, e cujos braços correspondem às entradas de um monitor. Estamos portanto dispostos a reconhecer dualidade entre programas usando monitores e programas usando troca de mensagens no estilo de Gentleman.

A maneira mais vívida de ver a dualidade entre os estilos é de usar uma notação comum. Em particular pretendemos padronizar uma notação gráfica que representa as relações estruturais entre os componentes de um programa concorrente. O exemplo da fig. 1 é tirado da página 195 de BH77.

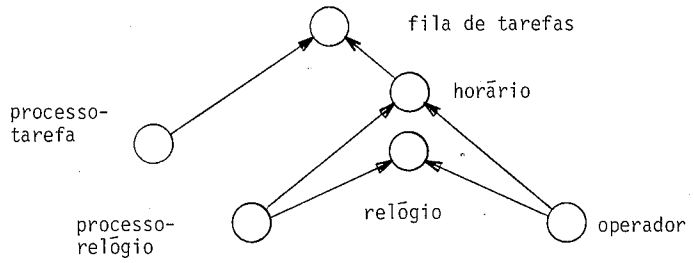


Figura 1: uma representação gráfica de uma estrutura concorrente

Brinch Hansen [BH77] utiliza a notação da figura 1 com o seguinte significado. Cada círculo representa ou um processo (do 1º tipo) ou um monitor. Cada seta representa "direitos de acesso", ou seja o direito de chamar entradas do monitor apontado pela seta. Monitores então são distinguidos por serem apontados por pelo menos dois outros componentes, enquanto processos (do 1º tipo) não são apontados por nenhum outro componente. Propomos adotar esta mesma notação para o modelo usando troca de mensagens. Neste caso valerá a interpretação a seguir. Cada círculo representa um processo (do 2º tipo). Cada seta representa uma relação de comunicação assimétrica, com a cabeça apontando o receptor das mensagens, que deverá respondê-las. (Por coincidência, ou não, Gentleman [GEN81] utiliza exatamente esta notação). Com estas convenções, podemos esta-

belecer a estrutura de um programa concorrente utilizando um ou outro modelo . O resultado da análise será um gráfico da estrutura, que agora possa ser interpretado de acordo com o modelo mais conveniente para fins de realização.

3. DUALIDADE COMO FERRAMENTA

No final da última secção sugerimos uma notação gráfica para programas concorrentes que incorpora a noção de dualidade. Daqui falta um passo pequeno para o resultado principal deste trabalho - o reconhecimento de dualidade como fernamenta poderosa para projetar um programa que será implementado usando um dos modelos aqui descritos. O primeiro passo é produzir um gráfico da estrutura na nossa notação. Deve-se observar que o raciocínio que gera este gráfico poderá usar características do modelo dual para chegar à estrutura desejada. A implementação segue direto de uma interpretação do gráfico em termos do modelo usada na implementação. Ou seja, o projeto pode ser feito em termos de um modelo, e implementado usando o outro. Aliás foi esta descoberta no contexto da implementação de um canal de comunicação duplex (veja na próxima secção) que motivou este trabalho.

Outra consequência de dualidade é a possibilidade de aproveitar os esforços de outros autores que trabalhavam em termos do modelo dual. Por exemplo, o administrador e os trabalhadores de Gentleman, desenvolvidos em termos da troca de mensagens [GEN81], têm seus correspondentes diretos em termos de monitores. Os gráficos de estrutura apresentados no seu artigo precisa meramente de interpretação segundo o outro modelo.

4. UM EXEMPLO

Na implementação do protocolo X.25, Farran [FAR81A] empregou o modelo de monitores [FAR81B]. Parte do programa envolve a implementação de um canal duplex para transmissão de pacotes com controle de fluxo. A estrutura (simplificada) adotada é mostrada na figura 2.

Neste gráfico incompleto, as interfaces desta subestrutura com às vizinhas se fazem através dos monitores INTx1, INRx1, INTx2 e INRx2. O monitor FLUXO é usado para controlar o fluxo de pacotes que passa pelo canal. Os processos (do 1º tipo) Rx e Tx são responsáveis respectivamente para recepção e transmissão relativas à interface interna. Quando foi elaborada esta estrutura, o seguinte defeito foi reconhecido - o processo Tx, responsável para transmissão busca seu pacote ou de INTx2 ou de FLUXO. Se não tiver um pacote num destes monitores, o processo espera neste monitor, apesar da existência de um pacote no outro monitor, com consequente perda de paralelismo, e, logo, desempenho.

A solução correta foi percebida somente depois de conhecido o estudo de

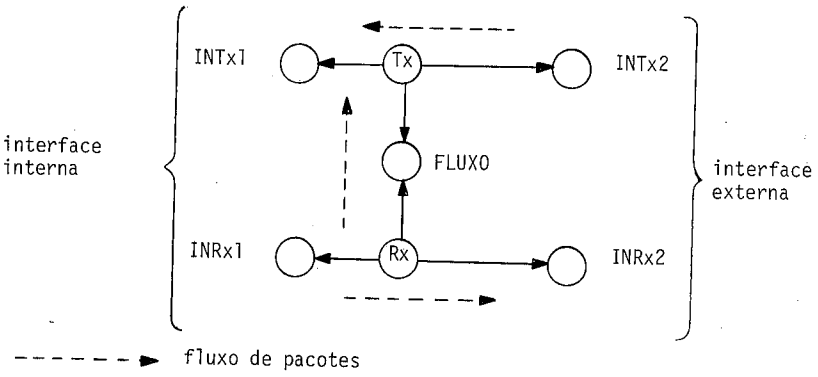


Figura 2: canal duplex de comunicação de pacotes (primeira versão)

Lauer e Needham. Foi tentada uma solução em termos da troca de mensagens. Na verdade isto combina muito bem com a aplicação, por tratar de transmissão de pacotes. A figura 3 mostra a nova estrutura.

Deve-se notar que o complexo formado pelos componentes Rx, Tx e FLUXO agora foi substituido pelo que Gentleman chama de um administrador (GERENTE) e quatro mensageiros (Rx1, Rx2, Tx1 e Tx2) para o envio de mensganes. Estes mensageiros não processam os pacotes transmitidos; todo processamento é centralizado em GERENTE.

Uma vez conhecida a estrutura, sua implementação em termos de monitores se torna fácil.

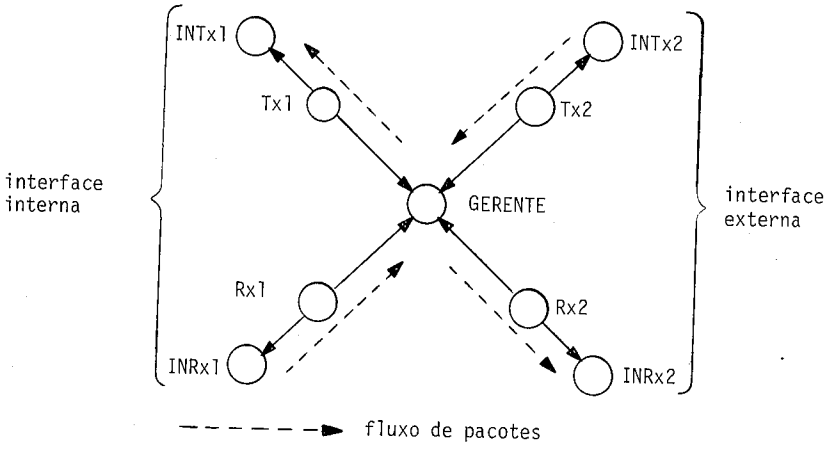


Figura 3: canal duplex de comunicação de pacotes (versão final)

5. CONCLUSÕES

No uso da ferramenta descrita aqui não há nenhuma consideração objetiva para preferir um ou outro dos modelos para a análise que gera o gráfico da estrutura. Por outro lado, para uma atividade intuitiva como projetar estruturas concorrentes, as considerações subjetivas são muito importantes. Este autor está convencido do que o modelo de troca de mensagem oferece mais apoio à intuição, devido ao seu maior grau de antropomorfismo, quando comparada com seu dual - organizações humanas funcionam na base da troca hierarquizada de mensagens. Gentleman [GEN81] também apoia esta visão.

6. AGRADECIMENTOS

Este trabalho resultou de longas discussões com Yussef Farran e Paulo Medeiros sobre a implementação de protocolos de comunicação. O autor agradece o apoio financeiro da FINEP, e a Ademar Passos pelos desenhos.

7. REFERENCIAS

[BH70] BRINCH HANSEN, P. "The nucleus of a multiprogramming system", CACM *13*(4), 238-250, 1970.

[BH77] BRINCH HANSEN, P., "*The architecture of concurrent programs*", Prentice-Hall, Englewood Cliffs, New Jersey, 1977.

[FAR81A] FARRAN L.,Y.E., "Especificação de uma implementação da recomendação X.25 do CCITT para um Sistema IBM/370", tese de mestrado, PUC/RJ , 1981.

[FAR81B] FARRAN L.,Y.E.e STANTON,M.A. "Programação Concorrente numa linguagem de alto nível - uma implementação ", *Anais do 8º SEMISH*, Florianópolis, pp 125-135, 1981.

[GEN81] GENTLEMAN,W.M., "Mesage passing between sequential processes: the reply primitive and the administrator concept", *Software-Practice and Experience*, *11*(5), 435-466, 1981.

[HOA74] HOARE,C.A.R., "Monitors: an operating system structuring concept" , CACM *17*(10), 549-557, 1974.

[HOA78] HOARE,C.A.R., "Communicating sequential processes", CACM *21*(8), 666-677, 1978.

[LAM80] LAMPSON,B.W. e REDELL,D.D., "Experience with processes and monitors in Mesa", CACM *23*(2), 105-117, 1980.

[LAU79] LAUER,H.E. e NEEDHAM,R.M., "On the duality of operating system structures", *Second International Symposium on Operating Systems*, IRIA, Rocquencourt, France, outubro de 1978, reimpresso em *Operating Systems Review 13*(2), 3-19, 1979.

[WIR77] WIRTH,N., " Modula, a language for modular multiprogramming" , *Software - Practice and Experience* 7(1), 3-35, 1977.