

Lidia Segre e Michael Stanton

COPPE/UFRJ  
Programa de Sistemas  
Caixa Postal 68511  
21945 Rio de Janeiro, RJ

Departamento de Informática  
PUC/RJ  
22453 Rio de Janeiro, RJ

### Sumário

Apresenta-se o atual estágio de um projeto que visa criar meios de construir software distribuído. Estes meios incluem a linguagem de programação, basicamente Modula-2 com algumas extensões, e uma linguagem de configuração, que permite definir como um programa distribuído é composto de módulos carregados (potencialmente) em estações diferentes. Entre os itens descritos se incluem a implementação da chamada remota de procedimentos e da linguagem de configuração estática de programas.

#### 1. Introdução

Os autores estão conduzindo uma investigação na construção de software para executar em ambientes distribuídos. Programas distribuídos são compostos de módulos, que são escritos na linguagem Modula-2 [Wirth 1982] com algumas extensões, e carregados em estações diferentes de um sistema distribuído de acordo com a finalidade do programa e a conveniência do seu configurador. Em [Segre 1985a], descrevemos as extensões a Modula-2 que são necessárias para permitir o uso desta linguagem para escrever módulos componentes de um programa distribuído. Estas incluem mecanismos para descrever concorrência e comunicação remota, e foi escolhido o modelo de processos e monitores (remotos) com acesso através da chamada remota de procedimentos. Para conservar o conceito de transparência de locação, mostramos que o mecanismo de chamada remota também tem que ser transparente, e definimos em [Segre 1985b] uma linguagem de configuração estática que permite configurar programas distribuídos, isto é, colocar e ligar seus módulos componentes.

Este relatório descreve a evolução das propostas contidas em [Segre 1985a, b], inclusive alguns aspectos pragmáticos decorrentes do trabalho prático de implementá-las. Além dos autores, participaram deste trabalho os alunos de mestrado Nanci dos Santos Lages (COPPE/UFRJ), Nelson Alves da Silva Filho (PUC/RJ) e Vanise Paraíso Vetromille (COPPE/UFRJ).

#### 2. Chamada remota de procedimento (CRP)

A implementação de chamada remota transparente segue o modelo de "stubs" adotado no projeto Cedar da Xerox [Birrell 1984]. A partir da especificação de uma interface remota,

escrita na forma de um módulo de definição em Modula 2, gera-se automaticamente um par de "stubs", um do cliente (importador da interface), e o outro do servidor (exportador da interface). O "stub" do cliente é um módulo (escrito em Modula-2) que implementa a interface em questão e recebe chamadas locais a seus procedimentos exportados, convertendo estas chamadas em mensagens a serem enviadas à estação onde está localizado o módulo exportador da interface. Alí, são recebidas pelo "stub" do servidor, que as transforma em chamadas aos procedimentos do módulo exportador. Ao terminar a execução da chamada, os resultados são enviados de volta à estação originária da chamada por procedimento análogo.

A implementação convencional de Modula-2 traduz uma especificação de interface, codificada num módulo de definição, para um arquivo "simbólico", que contém informação sobre os identificadores exportados [Geissmann 1983]. O gerador de "stubs" é um programa em Modula-2, que a partir deste arquivo simbólico gera dois arquivos de programa fonte em Modula-2 para os "stubs". Estes então serão compilados e usados para compor qualquer configuração em que esta interface é importada remotamente.

Os "stubs" se comunicam entre si usando um protocolo de comunicação, próprio para CRP, que é implementado neste projeto por um módulo de suporte replicado em cada estação da rede. Este módulo é responsável pela manutenção e atualização de informações sobre a localização dos módulos exportadores de interfaces importadas a cada estação, além do envio e da recepção de mensagens de chamada e retorno de chamada de procedimentos, e da recuperação de falhas de chamadas remotas. Para permitir um máximo de paralelismo, o módulo de suporte dispara um processo novo para executar cada chamada remota recebida.

Na implementação sendo desenvolvida, as informações sobre a localização dos módulos exportadores de interfaces poderão vir em forma de tabelas montadas pelo configurador estático e o carregador, ou poderão ser obtidas dinamicamente através de um servidor de nomes. Nesta implementação está sendo utilizado como subsistema de comunicação remota a rede local fabricada pela empresa Eden.

## 2. Configuração de programas distribuídos

Para alcançar o objetivo de transparência de locação, é preciso poder programar os módulos que compõem um programa distribuído sem saber em que estações estes serão carregados para execução. Esta informação é fornecida por um programa de configuração, escrito numa linguagem própria, que define a localização física dos módulos e as relações de importação e exportação de interfaces entre eles. Em [Segre 1985b], definimos uma linguagem de configuração estática, onde são conhecidos todos estes detalhes antes de iniciar a execução de qualquer componente do programa distribuído.

Atualmente está sendo implementada a linguagem

proposta, cujas declarações serão traduzidas em diretivas para carregar módulos em determinadas estações e estabelecer ligações entre exportadores e importadores. Para gerar estas diretivas o tradutor terá que reconhecer múltiplos usos do mesmo módulo, para evitar duplicação de módulos carregados, além das situações quando uma interface é importada remotamente. Neste último caso terão que ser carregados os "stubs" apropriados a esta interface, um de servidor junto ao exportador, e um de cliente para cada importador da interface.

### 3. Software de suporte

A implementação de CRP e de configuração de programas dependem da existência prévia de algum software de suporte. Os serviços básicos necessários incluem:

- . processos concorrentes e monitores;
- . comunicação remota em rede;
- . carga e controle de execução de módulos.

Para prover estes serviços, é preciso construir um sistema operacional rudimentar que execute em cada estação da rede.

### 4. Conclusão

Espera-se ter em funcionamento durante o primeiro semestre de 1987 a primeira versão do software descrito acima, para podermos então investigar mais concretamente a construção de software distribuído para uma rede local de computadores pessoais.

O trabalho descrito aqui contou com o apoio da Finep, do CNPq e da Embratel.

### Bibliografia

- [Birrell 1984] Birrell, A.D., Nelson, B.J. "Implementing remote procedure calls". ACM Trans. Comp. Sys. 2, 1 (fev 1984).
- [Geissmann 1983] Geissmann, L.B. "Separate compilation in Modula-2 and the structure of the Modula-2 compiler on the personal computer Lilith". Dissertação no. 7286, Instituto Federal de Tecnologia (ETH), Zurique, Suíça, 1983.
- [Segre 1985a] Segre, L., Stanton, M. "Sobre o uso de Modula-2 para programação em ambientes distribuídos". Anais do 5o. Simp. Des. Software Básico, Belo Horizonte (nov 1985), pág. 1-25.
- [Segre 1985b] Segre, L., Stanton, M. "Uma linguagem de configuração para o uso de Modula-2 para programação em ambientes distribuídos". Anais do 5o. Simp. Des. Software Básico, Belo Horizonte (nov 1985), pág. 26-40.
- [Wirth 1982] Wirth, N. "Programming in Modula-2". 2a. ed. Springer. 1982.