

UM SISTEMA DE SIMULAÇÃO PARA O CÁLCULO DE PROCESSOS

Waldomiro Soares Yuan ⁽¹⁾

Aloysio de Castro Pinto Pedroza ⁽¹⁾⁽²⁾

(1) COPPE/UFRJ - Prog. Eng. Elétrica
Centro de Tecnologia - Bloco H - Sala H-321
Cidade Universitária - Ilha do Fundão
21945, Rio de Janeiro, RJ
C.P.: 68504
Fax.: (021) 290-6626 Tel.: (021) 260-5010

(2) EE/UFRJ
Dep. de Eletrônica

Resumo

Em 1980, Robin Milner propôs uma teoria para descrever sistemas concorrentes que enfatizava principalmente o comportamento externo de um sistema. Esta teoria foi inicialmente chamada de Cálculo de Sistemas Comunicantes (CCS) e atualmente é conhecida como Cálculo de Processos (CP). Este trabalho apresenta o projeto e a implementação de um sistema para a simulação de especificações feitas em CP. Esta ferramenta permite o carregamento ou salvamento dos arquivos de especificação, simula o comportamento de uma especificação, apresenta os estados das simulações executadas e possibilita a alteração da especificação.

Abstract

In 1980, Robin Milner proposed a theory to describe concurrent systems that gave more emphasis on the external aspects of a system. This theory was initially called Calculus of Communicating Systems (CCS) and now it is known as Process Calculus. This work presents a system for the simulation of concurrent system specifications written in Process Calculus. This tool allows the loading or the saving of specification files, simulates the behavior of a specification, presents the states of executed simulations and allows the modification of the specification.

I . INTRODUÇÃO

Em 1980, Robin Milner propôs uma nova teoria , chamada de Cálculo de Sistemas Comunicantes (CCS), que dava mais importância à comunicação e à observação, e, conseqüentemente, ao comportamento externo do sistema modelado deixando sua estrutura interna com certo grau de liberdade [1]. Além disso, seus conceitos se mostraram tão úteis que serviram de base para a definição de uma FDT ("Formal Description Techniques") da ISO: a linguagem LOTOS [2].

Após sofrer algumas extensões [3,4], os conceitos do CCS foram atualizados por Robin Milner e esta teoria passou a ser chamada, por seu autor, de Cálculo de Processos [5].

O Cálculo de Processos ou CP não está confinado somente ao estudo da comunicação e concorrência; seu uso se aplica também ao estudo de arquiteturas, de métodos e de linguagens de programação. Além disso, o CP pode ser empregado para projetar ou para investigar o comportamento de sistemas concorrentes.

Este trabalho consiste do projeto, desenvolvimento e implementação de um sistema para a simulação de especificações de sistemas concorrentes reais ou hipotéticos. Assim, o Sistema de Simulação facilitará a síntese das especificações em CP dos sistemas concorrentes e permitirá a análise do comportamento de sistemas existentes através de suas especificações em CP [6].

Neste trabalho, a seção II esboça os operadores do Cálculo de Processos que podem ser utilizados em uma especificação, a seção III apresenta os principais módulos do Sistema de Simulação, a seção IV descreve a estrutura de dados e a implementação deste sistema; a seção V comenta a simulação de alguns exemplos; e a seção VI apresenta os comentários finais.

II . O CÁLCULO DE PROCESSOS

O Cálculo de Processos é um formalismo destinado a descrever sistemas concorrentes a partir do seu comportamento externo. Este cálculo é composto por um pequeno conjunto de operadores que pode modelar qualquer sistema concorrente, dependendo do interesse e do nível de abstração desejado. Em geral, um Sistema é composto por diversos módulos, chamados Agentes, que possuem identidade própria e podem agir de modo independente e concorrente ou interagir entre si. Cada agente possui pontos de interações, chamados Portas, através dos quais o agente pode trocar informações com o ambiente em que se encontra ou com outro agente do sistema.

No Cálculo de Processos, um agente é tratado como uma caixa preta, da qual um observador do ambiente externo não pode visualizar os estados, somente as ações em que este se engaja. Assim, para descrever um sistema, basta determinar o comportamento deste, tal como percebido pelo observador externo, deixando sua estrutura interna com certo grau de liberdade.

Além disso, somente as comunicações realizadas entre o sistema modelado e o ambiente são visíveis. As comunicações efetuadas entre os agentes do sistema não são diretamente observáveis e, por isso, são chamadas de ações internas. Tais ações são representadas por 'T' e também são conhecidas por ações invisíveis ou transições silenciosas.

A especificação de um sistema em CP apresenta uma forma textual e segue o estilo funcional. Além disso, esta é composta pela seqüência de definições de todos os agentes deste sistema. Nesta seqüência, o primeiro agente definido representa o comportamento inicial do sistema. A figura 1 apresenta a especificação do protocolo Bit-Alternado que será usada como exemplo. O comportamento inicial deste protocolo é dado pelo agente AB.

```

AB := ( EMISSOR | CANAL_MSG | CANAL_ACK | RECEPTOR )
      \ (EMSAP, RMSAP, EASAP, RASAP)

EMISSOR := (INICIO_E(0) | TIMER) \ (TIME, TIMEOUT, RESET)
INICIO_E(b) := IN(msg).ENVIA(msg, b)
ENVIA(msg, b) := ^EMSAP(msg, b).^TIME.ENVIANDO(msg, b)
ENVIANDO(msg, b) := TIMEOUT.ENVIA(msg, b)
                  + EASAP(x).ECHECK(msg, b, x)
ECHECK(msg, b, x) := if (x=b)
                    then ^RESET.INICIO_E(modulo(b+1, 2))
                    else ENVIANDO(msg, b)

TIMER := TIME . (^TIMEOUT.TIMER + RESET.TIMER)

RECEPTOR := RECEBE(0)
RECEBE(b) := RMSAP(msgx, x).^RASAP(x).RCHECK(msgx, x, b)
RCHECK(msgx, x, b) := if (x=b)
                    then ^OUT(msgx).RECEBE(modulo(b+1, 2))
                    else RECEBE(b)

CANAL_MSG := EMSAP(msg, b).TRANS(msg, b)
TRANS(msg, b) := T.^RMSAP(msg, b).CANAL_MSG
               + T.CANAL_MSG
               + T.^RMSAP(msg, b).^RMSAP(msg, b).CANAL_MSG

CANAL_ACK := RASAP(b).ACKING(b)
ACKING(b) := T.^EASAP(b).CANAL_ACK
            + T.CANAL_ACK
            + T.^EASAP(b).^EASAP(b).CANAL_ACK

```

Figura 1 : Especificação do Protocolo Bit-Alternado.

A definição de um agente é realizada através de uma equação de comportamento do agente, cujo formato é o seguinte:

<Nome> <Argumentos> := <Expressão>

Onde <Nome> é o nome através do qual o agente é referenciado; <Argumentos> é um elemento opcional que contém a lista de variáveis a serem ligadas aos parâmetros passados na chamada do agente; e <Expressão> contém a descrição do comportamento do agente e é chamada expressão de comportamento do agente.

A expressão de comportamento do agente é construída através da aplicação de alguns elementos básicos: agente inativo, tipo especial de agente que não admite nenhuma ação e serve para finalizar o comportamento de um agente

(representado por \circ); recursividade, técnica onde um agente chama a si próprio e é útil para a representação de comportamento repetitivo e/ou infinito; e operadores, são construtores que montam expressões mais complexas a partir de expressões mais simples.

As operações do CP utilizadas pelo Sistema de Simulação são as seguintes (maiores explicações podem ser encontrados em [2,5,6]):

Constante de Agente: operação que substitui o nome de um agente pela sua expressão de comportamento. Na figura 1, o agente AB é composto pelos agentes EMISSOR, CANAL_MSG, CANAL_ACK e RECEPTOR.

Prefixo (.): operação que indica a possibilidade de acontecer uma ação sobre um agente para evoluir em seu comportamento. Na figura 1, o agente TIMER pode executar a ação TIME e seu comportamento posterior será representado por ^TIMEOUT.TIMER + RESET.TIMER.

Somação (+): operação que combina expressões de comportamento indicando alternativas de comportamento. A expressão ^TIMEOUT.TIMER + RESET.TIMER pode executar duas ações: ^TIMEOUT ou RESET. Neste exemplo, em qualquer caso, o comportamento posterior será representado por TIMER.

Condição (if-then-else): operação que possibilita a expressão do comportamento de um agente em função de uma condição. Na figura 1, o comportamento do agente ECHECK depende do valor da variável x com relação à variável b.

Restrição (\): operação que esconde determinadas portas de um agente e proíbe toda ação visível através delas.

Exemplo : Seja o agente $A := ^R1.A1 + ^R2.A2$

Então, na expressão $A\{R2,R3\}$, só pode acontecer a ação ^R1.

Rerotulação (/): operação utilizada para realizar a renomeação dos nomes das portas de um agente.

Exemplo : Seja o agente $A := R.A1 + \hat{R}.A2 + RR.A3$

Então, na expressão $A[L/R]$, podem acontecer as ações \underline{L} , $\hat{\underline{L}}$ e \underline{RR} .

Composição (|): operação que realiza a composição entre dois agentes. Assim, na composição, podem acontecer as ações individuais de cada agente componente e as interações possíveis entre os agentes componentes. Para dois agentes interagirem diretamente, é necessário que os mesmos tenham rótulos complementares e esta interação produz uma ação interna \underline{T} (sincronização ou comunicação).

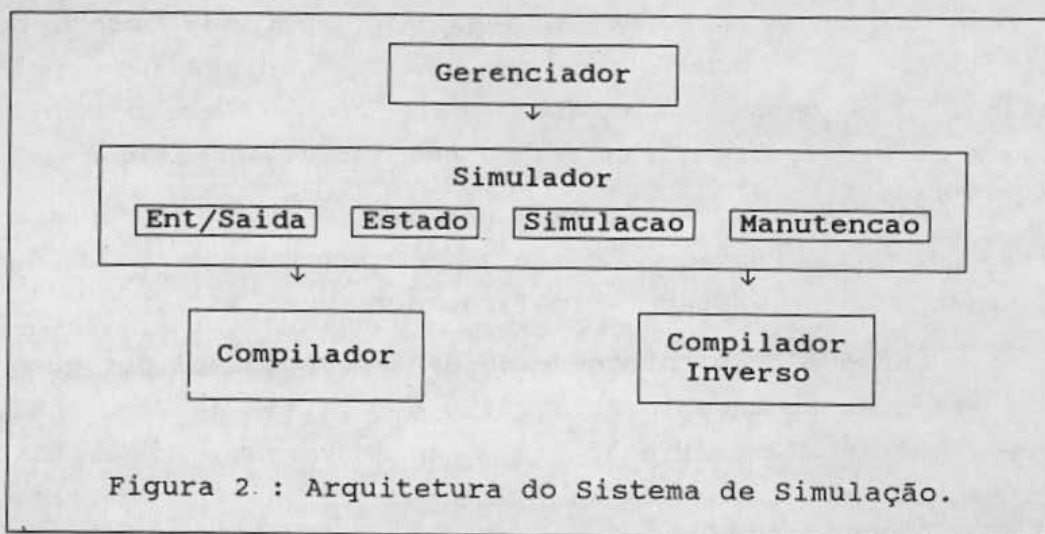
Exemplo : Sejam $\left(\begin{array}{l} A := R(x).A1(x) + L(y).A2(y) \\ B := \hat{R}(5).B1 \end{array} \right.$

Então: $\left(\begin{array}{l} A | B \xrightarrow{R(9)} A1(9) | B \\ A | B \xrightarrow{L(9)} A2(9) | B \\ A | B \xrightarrow{\hat{R}(5)} A | B1 \\ A | B \xrightarrow{T} A1(5) | B1 \end{array} \right.$

Ou seja, as ações, que os agentes A e B podiam sofrer isoladamente ($R(x)$ e $L(y)$, para A , e $\hat{P}(5)$, para B), $A|B$ também pode sofrer, alterando somente o agente que sofreu a ação. As ações pertencentes a agentes diferentes (A e B), que possuam rótulos complementares (R e \hat{R}), podem produzir uma ação interna \underline{T} .

III . O SISTEMA DE SIMULAÇÃO

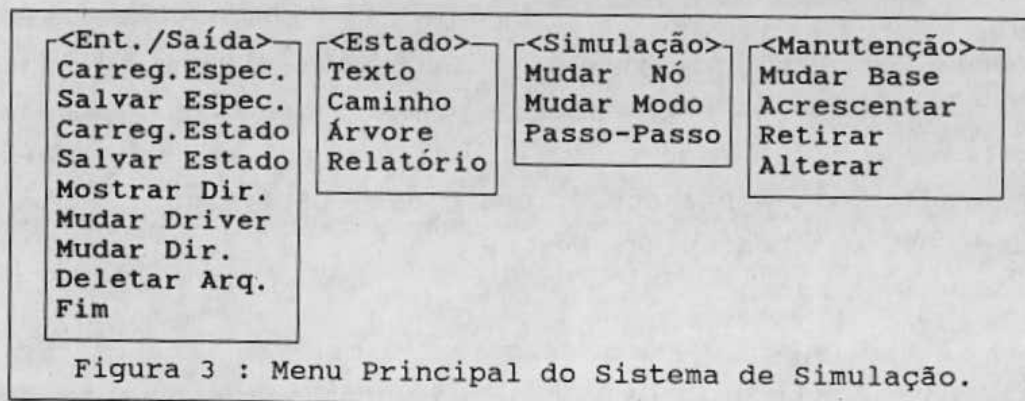
O Sistema de Simulação para o Cálculo de Processos é uma ferramenta que permite a síntese ou a análise de sistemas reais ou hipotéticos. Este sistema é formado por quatro módulos principais: Gerenciador, Simulador, Compilador e Compilador Inverso. Sua arquitetura está apresentada na figura 2, onde as setas indicam dependência entre os módulos.



O módulo Compilador gera uma estrutura interna a partir de uma especificação. Esta estrutura descreve o comportamento da especificação e está num formato próprio para ser manipulado pelo Sistema de Simulação.

O módulo Compilador Inverso executa a conversão inversa do módulo Compilador. Assim, pode-se obter a "string" de uma estrutura para ser apresentada ao usuário num formato inteligível.

O Gerenciador controla as interações entre o usuário e o Sistema de Simulação e gerencia os serviços oferecidos pelo Sistema de Simulação, a fim de atender as solicitações requeridas pelo usuário.



A figura 3 apresenta o menu principal do Sistema de Simulação que contém todos os serviços oferecidos pelo mesmo. Este menu é controlado pelo módulo Gerenciador e todos os serviços são executados pelo módulo Simulador.

Assim, de acordo com os tipos de serviços oferecidos, o módulo Simulador é composto pelos seguintes submódulos: Entrada e Saída, Estado, Simulação e Manutenção.

O submódulo de Entrada e Saída é responsável por quase todo tipo de tratamento de arquivo do Sistema de Simulação, principalmente o carregamento e o salvamento do arquivo fonte que contém uma especificação em CP. No carregamento, este submódulo utiliza o módulo Compilador para obter sua estrutura interna e inicializa as variáveis necessárias para preparar o sistema para a simulação. No salvamento, este submódulo utiliza o módulo Compilador Inverso para obter a especificação da estrutura interna, caso esta sofra alguma alteração.

O submódulo de Estado contém todas as estruturas estáticas e dinâmicas onde estão a configuração e os elementos de uma especificação, inclusive a estrutura compilada durante o carregamento da especificação. Além disso, este submódulo possui alguns serviços que permitem a visualização da especificação, do estado atual e dos outros estados simulados. Para isso, o mesmo utiliza o módulo Compilador Inverso.

O submódulo de Simulação executa a simulação da especificação carregada. Durante a simulação, este submódulo constrói uma árvore, chamada árvore de simulação, onde os nós são as expressões de comportamento dos estados alcançados pela simulação e os ramos são as ações executadas pelo simulador (transições). Esta árvore contém todos os estados simulados e também é mantida pelo submódulo de Estado.

Além disso, o submódulo de Simulação mantém uma variável no submódulo de Estado que contém a estrutura de

dados do nó a ser simulado no próximo ciclo de simulação, chamado nó atual. Inicialmente, o nó atual contém a estrutura do comportamento inicial da especificação, que é a estrutura do agente base (primeiro agente definido). A medida que cada ciclo de simulação é executado, o nó e o ramo executado é adicionado à árvore de simulação e o nó atual é alterado para conter a estrutura da expressão de comportamento do filho da ação executada.

O submódulo de Entrada e Saída também permite o salvamento e o carregamento do estado do Sistema de Simulação em um arquivo para não perder a simulação realizada até o momento. Assim, posteriormente, este arquivo pode ser carregado e a simulação prosseguida.

O submódulo de Manutenção permite a alteração da especificação carregada. Para isso, este submódulo atua diretamente na estrutura interna da especificação, pertencente ao submódulo de Estado. Uma vez realizada qualquer alteração, o sistema é reinicializado.

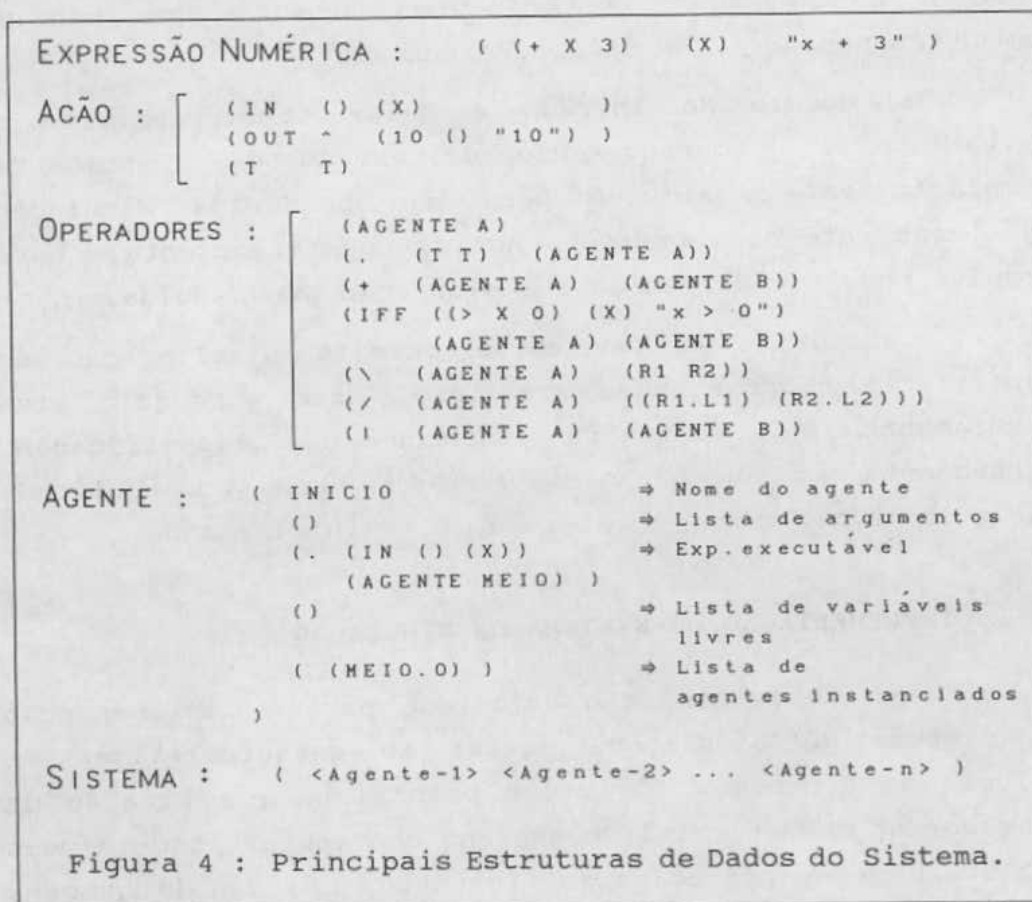
IV . IMPLEMENTAÇÃO DO SISTEMA DE SIMULAÇÃO

A linguagem de programação utilizada para a implementação do Sistema de Simulação, devido às características do Cálculo de Processos, deve ser própria para a criação de operadores, deve permitir o emprego de recursividade e suas variáveis devem ser atipadas. Portanto, uma das linguagens que apresenta tais características é o LISP.

A linguagem LISP possibilita a um programa manipular outro, uma vez que esta utiliza a mesma estrutura para representar programas e dados (listas). Assim, durante a compilação, estruturas executáveis associadas às expressões de comportamento são montadas e, durante a simulação, tais estruturas são avaliadas dentro de determinados ambientes a fim de apresentar o comportamento da expressão ao usuário.

A figura 4 apresenta as principais estruturas de dados utilizadas pelo Sistema de Simulação, a saber: Expressão

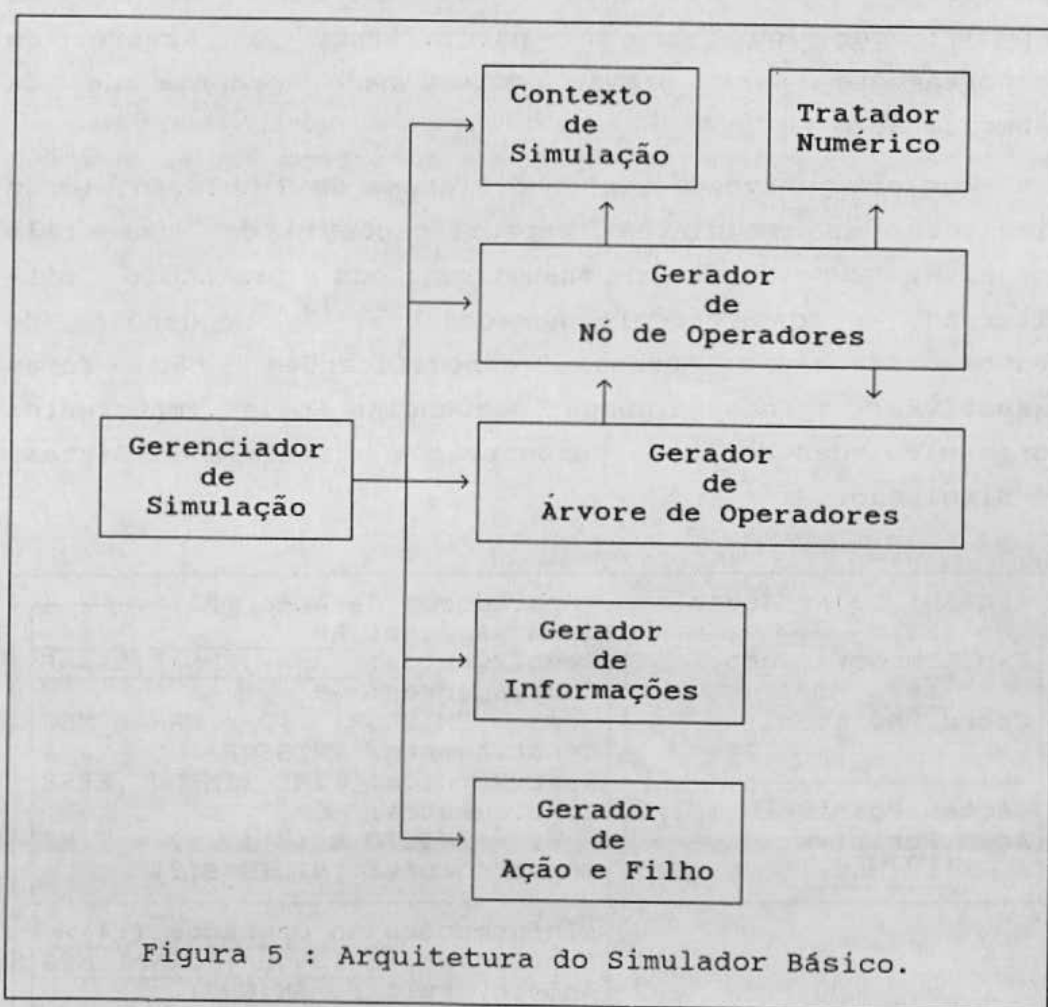
Numérica, Ação, Operadores, Agente e Sistema. O formato destas estruturas foram projetadas para facilitar sua manipulação por parte do sistema. Assim, por exemplo, as estruturas dos operadores são elementos executáveis do LISP que podem ser avaliadas em determinados ambientes para obter informações ou outras estruturas durante a simulação.



Para executar a simulação, o Sistema de Simulação utiliza o Simulador Básico cuja arquitetura está apresentada na figura 5.

O Gerenciador de Simulação controla a seqüência das tarefas de simulação do nó atual; o Contexto de Simulação contém todas as variáveis utilizadas pelo simulador básico; o Tratador Numérico realiza todo tipo de tratamento das expressões numéricas; o Gerador de Nó de Operadores monta a estrutura de nó do operador mais externo de uma expressão;

o Gerador de Árvore de Operadores monta a estrutura da árvore de operadores de uma expressão (estrutura que contém o encadeamento dos operadores utilizados para obter as ações possíveis); o Gerador de informações gera todas as informações necessárias para que o usuário escolha uma ação; e Gerador de Ação e Filho gera a estrutura da ação executada e do filho desta ação.



Assim, durante um ciclo de simulação, o Gerenciador monta a árvore de operadores do nó atual, utilizando o Gerador de Árvore de Operadores; apresenta a lista de ações possíveis ao usuário, com o auxílio do Gerador de Informações; executa a ação escolhida, usando o Gerador de

Ação e Filho; e adiciona a estrutura da ação executada e do filho gerado à árvore de simulação do Sistema de Simulação.

V . ANÁLISE DOS RESULTADOS

O processo de validação de uma especificação deve mostrar que o sistema modelado atende a requisitos pré-estabelecidos de projeto. Para isso, realiza-se testes exaustivos para verificar todo o comportamento da especificação ou gera-se parcialmente a árvore de comportamento para provar determinada propriedade da especificação [8].

Com o objetivo de testar o Sistema de Simulação, foram simuladas as seguintes especificações: de uma fila infinita, do Produtor-Consumidor, do protocolo Bit-Alternado e do protocolo Abracadabra. As seqüências de testes simuladas nestas especificações não foram exaustivas; apenas algumas seqüências mais importantes foram simuladas a fim de demonstrar a utilidade do Sistema de Simulação.

<pre><PASSO: Todas Ações > <Nó (F1)> Exp.Comportamento: AB Coord. Nó Atual:</pre>	<pre><Operadores da Ação (F3)> Const.Agente: AB Rest.Rótulos: EMSAP,RMSAP,EASAP Comp.agentes: #1 = EMISSOR; #2 = CANAL_MSG Const.Agente: EMISSOR Rest.Rótulos: TIME,TIMEOUT,RESE Comp.agentes: #1 = INICIO_E (?); #2 = TIME Const.Agente: INICIO_E(?)</pre>
<pre><Ações Possíveis (F2)> Ação Positiva: IN(MSG)</pre>	<pre><Informações do Operador (F4)> Exp.Escopo: EMISSOR CANAL MSG Ações Permit.: IN(MSG) ; Ações Rest.: EMSAP(MSG,B) ; RA</pre>

Figura 6 : Tela da Simulação do Protocolo Bit-Alternado.

A figura 6 contém a tela apresentada durante a

simulação do protocolo Bit-Alternado. A janela F1 contém informações da expressão de comportamento do estado atual; a janela F2, a lista de ações possíveis; a janela F3, a lista de operadores utilizados para obter a ação destacada em F2; e a janela F4 apresenta informações adicionais do operador destacado em F3.

Assim, através do Sistema de Simulação, pode-se simular o comportamento de uma especificação, visualizar os diversos estados simulados e gerar um relatório de todos os estados simulados. Além disso, o submódulo de manutenção permite correções da especificação durante a depuração (atuando diretamente na estrutura de dados sem a necessidade de novas compilações) e possibilita a simulação de parte da especificação facilitando sua análise.

VI . CONCLUSÃO

O Cálculo de Processos parece ser um instrumento natural para a síntese e a análise de sistemas abertos, quando se quer enfatizar principalmente as características externas, deixando a estrutura interna com um maior grau de liberdade. Além disso, o conjunto de operadores utilizado é suficiente para especificar qualquer sistema concorrente assíncrono [4,5].

O Sistema de Simulação desenvolvido permite o carregamento de uma especificação em CP e sua simulação, apresentando o seu comportamento; possibilita a visualização de todos os estados simulados da especificação e o salvamento do estado do Sistema de Simulação em um arquivo a fim de não perder as simulações realizadas; e pode-se alterar os elementos da especificação, a fim de corrigi-la, ou simular parte da mesma.

Além disso, o Sistema de Simulação apresenta um bom conjunto de mensagens de erro e uma interface com o usuário que se mostrou bastante prática, facilitando, respectivamente, o processo de depuração e de análise de

uma especificação.

Este Sistema de Simulação foi desenvolvido para microcomputadores da linha IBM-PC e é próprio para simulações de sistemas de pequeno porte, de cunho didático. O Sistema de Simulação torna-se lento quando se analisa especificações de maior complexidade. Portanto, este sistema deverá ser transportado para estações de trabalho de alta capacidade para que especificações de sistemas de grande porte possam ser avaliadas.

VII . REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MILNER, R., A Calculus of Communicating Systems, Lectures Notes in Computer Science, Vol.92, Springer-Verlag, Berlin, 1980.
- [2] BOLOGNESI, T. e BRINKSMA, E., "Introdution to the ISO Specification Language LOTOS", Computer Networks and ISDN Systems, Vol.14, pp.25-59, 1987.
- [3] MILNER, R., "Calculi for Synchrony and Asynchrony", Theoretical Computer Science, Vol.25, pp.267-310, 1983.
- [4] CARCHIOLO, V., STEFANO, A., FARO, A. e PAPPALARDO, G., "ECCS and LIPS: Two Language for OSI Systems Specification and Verification", ACM Transaction on Programming Languages and Systems, Vol.11, No.2, pp.284-329, April 1989.
- [5] MILNER, R., Communication and Concurrency, Prentice Hall International, London, 1989.
- [6] YUAN, W.S., "Um Sistema de Simulação para o Cálculo de Processos", Tese de Mestrado, COPPE/UFRJ, 1991.
- [7] PEDROZA, A.C.P. e VALIN, P.R.O., "Um Simulador de Protocolo de Comunicação para a Linguagem ESTELLE", VI Simpósio Brasileiro de Telecomunicações, Campina Grande, PB, 1988.
- [8] FIALHO, S.V., PEDROZA, A.C.P., LEÃO, J.L.S. e OLIVEIRA Jr., R.C., "Uma Ferramenta para Verificação Automática de Programas Concorrentes", VII Simpósio Brasileiro de Telecomunicações, Florianópolis, SC, 1989.