

**Boletim Técnico da Escola Politécnica da USP**  
**Departamento de Engenharia de Sistemas**  
**Eletrônicos**

ISSN 1517-3542

**BT/PSI/0014**

---

**Projeto de um Sistema de  
Sinalização de Tronco E1: Um  
Estudo de Caso usando o  
Ambiente *Ptolemy***

---

**Oscar W. Peña Guilarte  
Marius Strum**

**São Paulo – 2000**

O presente trabalho é um resumo da dissertação de mestrado apresentada por Oscar W. Peña Guilarte, sob orientação do Prof. Dr. Marius Strum.: "Projeto de um Sistema de Sinalização de Tronco E1: Um Estudo de Caso usando o Ambiente Ptolemy", defendida em 27/04/00, na Escola Politécnica.

A íntegra da dissertação encontra-se à disposição com o autor e na Biblioteca de Engenharia de Eletricidade da Escola Politécnica/USP.

#### FICHA CATALOGRÁFICA

Peña Guilarte, Oscar W

Projeto de um sistema de sinalização de tronco E1 : um estudo de caso usando o ambiente Ptolemy / O.W. Peña Guilarte, M. Strum. -- São Paulo : EPUSP, 2000.

p. -- (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia de Sistemas Eletrônicos, BT/PSI/0014)

1. Ambiente PTOLEMY 2. Sinalização de tronco E1 3. Sistemas digitais - Projetos I. Strum, Marius II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos III. Título IV. Série

ISSN 1517-3542

CDD 005.438  
621.3857  
004.1

# Projeto de um Sistema de Sinalização de Tronco E1: Um Estudo de Caso usando o Ambiente *Ptolemy*

Oscar W. Peña Guilarte <sup>\*</sup>

Orientador: Marius Strum <sup>\*\*</sup>

Departamento de Engenharia Eletrônica – PEE  
Laboratório de Microeletrônica – LME  
Escola Politécnica da Universidade de São Paulo  
Av. Prof. Luciano Gualberto, Travessa 3, N° 158  
05508-900 São Paulo SP Brasil  
Telefone: (011) 818-5310 r. 304  
Fax: (011) 8185585  
E-mail: <sup>\*</sup>[vilarte@lme.usp.br](mailto:vilarte@lme.usp.br) , <sup>\*\*</sup>[strum@lme.usp.br](mailto:strum@lme.usp.br)

## Resumo

Neste artigo apresentamos um estudo sobre a automação do projeto de sistemas digitais integrados através do ambiente *Ptolemy*. Para este fim projetamos um sistema de sinalização de tronco E1, usado para a comunicação entre centrais telefônicas em altas velocidades utilizando transmissão digital PCM de 30 canais e portadora E1 de 2.048 Mbps. O ambiente *Ptolemy* não permite realizar a síntese automática de uma arquitetura hardware/software a partir da especificação comportamental do sistema. Nossa trabalho explorou então um conceito de projeto que pode ser chamado de “refinamento progressivo”, já proposto por outros grupos de pesquisa. Segundo este método desenvolve-se inicialmente um modelo executável do sistema alvo, para que este possa ser validado (comportamento funcional, temporal e restrições da especificação). A partir deste geram-se outros modelos em níveis de abstração cada vez mais baixos, culminando com a obtenção de modelos que possam ser utilizados para sintetizar uma implementação hardware e/ou software do sistema, tais como descrições VHDL para as ferramentas de síntese de hardware (por exemplo o Behavioral Compiler<sup>TM</sup> ou ainda o Design Compiler<sup>TM</sup> da Synopsys) ou programas em C ou C<sup>++</sup> que são executados em núcleos de microprocessadores. Neste trabalho geramos dois modelos do nosso sistema, um híbrido baseado em 3 domínios de simulação, que foi simples de ser capturado e provou ser eficiente para validar a especificação do sistema, e o segundo baseado no domínio de geração de código CGC, que serve para criar uma implementação em software (programa em C) do nosso sistema alvo. As vantagens e limitações do método são discutidas no final do artigo.

## Abstract

In this article we present a study about the design automation of integrated digital systems using the *Ptolemy* environment. For this purpose we designed an E1 signaling system, used for high speed communication between telephone central offices according to the digital transmission 30 channels PCM standard, designated as E1 trunk at 2.048 Mbps". The *Ptolemy* environment does not allow to automatically synthesize a hardware/software architecture from a behavioral specification. In our work we explored a design concept called "progressive refinement", already proposed by other research groups. According to this method, the first design step requires the development of an executable model of the target system that will be used to validate its functional and timing behavior (including specification constraints). Once this step is completed, other models, at lower levels of abstraction, are created from this one, ending with a model that can be used to synthesize the system's hardware and/or software implementation, as VHDL descriptions for hardware synthesis tools like the Behavioral Compiler<sup>TM</sup> or the Design Compiler<sup>TM</sup> from Synopsys, or still C or C<sup>++</sup> programs to be executed on microprocessor cores. In this work we generated two models of our system, one that was based on three simulation domains, which was simple to be captured and efficient to validate the system's functionality, and the second based on the CGC code generation domain, which serves to create a software implementation (C program) of our target system. The advantages and limitation of this method are discussed at the end of the article.



## 1. Introdução.

No mundo atual a complexidade dos circuitos integrados de aplicação específica (ASIC) vem aumentando progressivamente, levando consigo a necessidade de pesquisas de novas metodologias de projeto que iniciem pelo nível de descrição comportamental do circuito integrado (CI) ou sistema digital, e que sejam amplamente suportadas pelas ferramentas de software. Atualmente o projeto e especificação de sistemas de aplicação específica é alvo fundamental de pesquisas.

Existem muitos ambientes de projeto destinados à modelagem, simulação e geração de códigos para síntese de sistemas heterogêneos, um deles é o *Ptolemy*, o qual permite descrever sistemas digitais completos e gerar o código que descreve as partes a serem implementadas em hardware e software, utilizando tecnologia orientada a objeto [1-3]. Além disso é considerado como uma infra-estrutura compacta de ferramentas, onde são construídos diferentes mecanismos chamados “Domínios” que realizam um modelo computacional para cada tipo de sistema.

Dentre os sistemas, podemos fazer menção ao sistema de telefonia, onde existe atualmente uma forte tendência a transformação dos atuais sistemas telefônicos em redes inteiramente digitais tanto na transmissão como na comutação [4].

Este artigo apresenta a modelagem de um Sistema de Sinalização de Tronco E1 para telefonia digital, utilizando o *Ptolemy* como ambiente de projeto. É feita uma caracterização de alguns comportamentos explorando diferentes domínios do *Ptolemy*. Apresenta uma estimativa de implementação através dos domínios de geração de código.

O artigo está organizado da seguinte forma. A seção 2 descreve o sistema de Sinalização de tronco E1. A seção 3 apresenta a modelagem e simulação do sistema, assim como o modelo de geração de código para uma implementação. Na seção 4 é feita uma discussão do desempenho do *Ptolemy* baseado nos modelos construídos. Na seção 5 são apresentadas as conclusões do trabalho. As referências estão na seção 6.

## 2. Descrição do Sistema de Sinalização de Tronco E1.

A Sinalização de tronco E1 é o sistema utilizado na Europa e em alguns países sul-americanos como Brasil e outros, para a comunicação à altas velocidades entre centrais telefônicas utilizando transmissão digital PCM de 30 canais e a portadora E1 de 2.048 Mbps. O termo (E1) significa que é um sistema Europeu de hierarquia 1. Este sistema tem dois enlaces PCM de 30 canais para a comunicação com a central telefônica, e vários enlaces com protocolo MVIP (*Multi-Vendor Integration Protocol*) para a comunicação com as diferentes aplicações associadas a esta tecnologia. Cada canal é amostrado a 8 kHz e transmitido a 64 Kbit/s, o que resulta numa velocidade de 2.048 Mbit/s para transmitir os 32 canais. O sistema utiliza: lei de composição *A-law*, Sinalização de Linha R2-Digital, Sinalização multi-freqüencial entre Registradores (MFC), multiplexagem por divisão no tempo (TDM) e comutação compatível com ambiente ST-BUS e MVIP. Pode ser dividido nos seguintes blocos funcionais: enlace PCM, bloco de sincronização, bloco de Sinalização de Linha, bloco de Sinalização entre Registradores e Comutador.

### 2.1 Enlace PCM de 30 Canais.

O enlace PCM de 30 canais consiste em um quadro de 256 bits transmitidos a uma taxa de repetição de 8000 Hz o qual resulta em uma freqüência de transmissão de 2.048 Mbit/s. Um quadro básico é dividido em 32 canais numerados de 0 - 31, como mostra a figura 1. Cada canal é de 8 bits, transmitidos a 64 Kbit/s, onde o primeiro bit enviado é o bit mais significativo (MSB) identificado como bit 1. Este enlace possui 30 canais destinados para a transmissão de

dados de voz ou dados digitais e 2 canais destinados para controle (*time slot*). Estes canais de controle estão situados no canal “0” (zero) e no canal 16 respectivamente. O canal “0” é chamado de canal de sincronização e é reservado para o alinhamento de quadro e para a conservação da informação da comunicação. O canal 16 é chamado de canal de Sinalização e pode funcionar em qualquer dos dois modos (sinalização por canal comum (CCS) ou sinalização associada ao canal (CAS ou sinalização de bit ABCD), através do qual são enviadas e recebidas informações referentes aos canais de voz.

## 2.2 Sincronização.

Num enlace PCM de vários canais é importante que o canal transmitido seja corretamente identificado no lado receptor. Para garantir tal propósito é utilizado um sinal padrão, chamado sinal de alinhamento de quadro (FAS), gerado no lado transmissor e usado como referência para todos os bits subseqüentes até a chegada de um novo FAS. Também existe o sinal que não contém o alinhamento de quadro (NFAS) que serve para ajudar na sincronização da comunicação. FAS e NFAS são transmitidos no canal “0” de quadros consecutivos. Ou seja, o sinal FAS será transmitido no canal “0” dos quadros pares, começando pelo quadro “0”, e o sinal NFAS será enviado no canal “0” mas dos quadros ímpares começando no quadro 1. O bit 2 destes sinais é utilizado para diferenciar um do outro, para o FAS o bit2 = 0, no entanto que para o NFAS bit2 = 1.

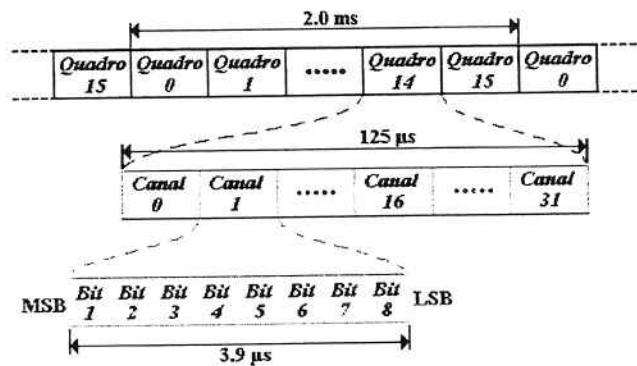


Figura 1. Formato do enlace PCM 30.

O alinhamento de quadro começa pela busca da seqüência de bits 0011011, os quais aparecem nos últimos 7 bits do FAS. O bit 1 do FAS pode ser o resto do alinhamento de multi-quadro CRC-4, ou um bit de uso internacional. No NFAS, o bit 1 pode ser utilizado para uso internacional. O bit 3 designado com a letra “A” é chamado indicação de alarme remoto (RAI), e é utilizado para indicar o estado da sincronização de quadros do enlace entre terminais. Em condições normais de operação este bit “A” deve estar colocado em “0”, no entanto que em condição de alarme será colocado a “1”. Este formato é ilustrado na figura 2.

Além do alinhamento de quadro, existe outro tipo sinal chamado “alinhamento de multi-quadro”, que facilita a identificação dos quadros dentro de um multi-quadro. O propósito de um algoritmo de Sinalização de multi-quadro está em oferecer um esquema que permita uma associação de uma Sinalização específica ABCD *nibble* com o apropriado canal do sistema PCM 30.

Canal 0 (zero)										
<i>M</i>	<i>Quadro 0</i>	C1	0	0	1	1	0	1	1	FAS
<i>l</i>	<i>Quadro 1</i>	0	1	A	X	X	X	X	NFAS	
<i>t</i>	<i>Quadro 2</i>	C2	0	0	1	1	0	1	1	FAS
<i>q</i>	<i>Quadro 3</i>	0	1	A	X	X	X	X	NFAS	
<i>a</i>										
<i>d</i>										
<i>r</i>	<i>Quadro 15</i>	0	1	A	X	X	X	X	NFAS	
<i>o</i>										

*Cn - Bit de uso internacional.*  
*A - Bit indicador de Alarme.*  
*X - Bit de uso nacional.*

Figura 2. Formato do canal 0 para os quadros pares e ímpares.

O canal 16 é reservado para a comunicação da Sinalização da informação associada ao canal (CAS) ou Sinalização ABCD. Isto quer dizer que a informação referente a cada canal de voz é enviada a través do canal 16 e de quadro em quadro, utilizando 4 bits para um canal de voz, portanto cada canal 16 é dividido em dois *nibble* (4 bits) para sinalizar o estado de dois canais de voz. Deste modo torna-se necessário um alinhamento de multi-quadro para podermos interpretar corretamente a Sinalização presente, assim como saber pelo número do quadro a quem pertence a informação contida no canal 16.

A Sinalização CAS consiste de 16 quadros, numerados de 0 a 15, o qual resulta numa taxa de repetição de 2 ms. O alinhamento de multi-quadro (MFAS) está baseado na procura do sinal ou seqüência de bit “0000”, o qual aparece nos 4 bits mais significativos do canal 16 do quadro “0” de um multi-quadro. O bit 6 desse canal é o bit de alarme, indicativo da perda de alinhamento de multi-quadro quando seu valor é “1”, e usualmente designado com a letra “Y”. Os bits 5, 7 e 8, usualmente designados “X”, são bits de reserva que normalmente tem valor “1” se não são usados. O canal 16 do resto dos quadros (1 - 15) são reservados para a sinalização dos bits ABCD dos 30 canais de voz. Os 4 bits mais significativos são reservados para sinalizar os canais de 1 a 15 e os outros 4 bits menos significativos são para sinalizar os canais de 16 a 30. O formato do canal 16 do quadros 0 a 15 é mostrado na tabela 1.

Tabela 1 Formato do canal 16 para os quadros do 0 - 15.

Canal 16 Quadro 0	Canal 16 Quadro 1	Canal 16 Quadro 2	Canal 16 Quadro 3	...	Canal 16 Quadro 14	Canal 16 Quadro 15
0000XYXX	abcd	abcd	abcd	Abcd	abcd	abcd
	Can al 1	Can al 17	Can al 2	Can al 18	Can al 3	Can al 19

### 2.3 Sinalização de Linha.

O bloco de Sinalização de Linha é o encarregado de processar os dados recebidos dos canais de controle 16 de cada quadro, nos quais estão contidas as informações referentes ao controle dos 30 canais de voz. A Sinalização de Linha é um conjunto de sinais destinados a efetuar a ocupação e supervisão enlace a enlace dos circuitos que interligam duas centrais de comutação telefônica, e que opcionalmente permite o envio dos sinais de tarifação. O sistema utiliza Sinalização R2 Digital, constituída por sinais padrões regidos por normas e especificações, definidas pela ITU-T, estabelecidas nos sistemas de telecomunicações de cada país, nosso caso está referido ao sistema TELEBRÁS do Brasil. Este sistema utiliza dois canais de sinalização no sentido “terminal destino” ( $a_f$  e  $b_f$ ) e dois canais de sinalização no sentido “terminal origem” ( $a_b$  e  $b_b$ ). Estes canais são usados na troca de informações entre junções com enlace PCM, e refletem as condições de operação do equipamento de comutação dos terminais

(saída ou entrada) e as condições de enlace dos assinantes (chamador ou chamado). As especificações e normas que regem os sinais padrões desta sinalização estão contida na referência [5].

#### 2.4 Sinalização entre Registradores.

O bloco de Sinalização entre Registradores é o encarregado de processar os sinais recebidos dos canais de voz, uma vez que o canal tenha ocupado a linha, para estabelecer a comunicação com o assinante chamado, ou com a aplicação solicitada. A Sinalização entre Registradores é o conjunto de sinais correspondentes ao envio e à recepção das informações, devidamente ordenadas, destinadas ao estabelecimento das chamadas através dos órgãos de comutação, das informações referentes as condições particulares dos assinantes chamador e chamado e das informações referentes aos circuitos de órgãos envolvidos.

Primeiramente definiremos como **registrador** ao órgão, conjunto de órgãos ou função de uma central telefônica capaz de receber, interpretar, armazenar algarismos e informações, e controlar a troca de informações desde e para outras centrais. A Sinalização multi-freqüencial (MFC) é um sistema compelido, ou seja, que cada sinal transmitido em um sentido depende de uma resposta no sentido oposto, sem o que não será interrompido, até o limite de tempo permitido pela temporização dos equipamentos de comutação.

Neste sistema são utilizadas 12 freqüências, divididas em dois grupos de seis, denominados grupo de freqüências altas e grupo de freqüências baixas. Cada sinal é formado pela união de duas freqüências, dentro de um mesmo grupo, seja o de freqüências altas ou baixas. Cada freqüência é designada por um índice e um peso, de modo que cada sinal pode ser identificado pela soma do índice e do peso atribuído às freqüências que o constituem. As freqüências altas são transmitidas para frente, no sentido do estabelecimento da cadeia de comutação. As freqüências baixas são transmitidas para trás, como resposta às primeiras. Cada conjunto de sinais (para frente e para trás) tem dois esquemas de significados diferentes, utilizando as mesmas combinações de freqüências. A passagem de um esquema para outro é indicada por meio de um dos sinais reservados para esse fim. As normas, indicações e esquemas para cada um destes sinais aparecem na referência [6].

### 3. Modelagem do Sistema através do *Ptolemy*.

Para modelar um sistema através do *Ptolemy* deve-se escolher um modelo de computação capaz de descrever seu comportamento, contendo todas, ou a maior quantidade possível das especificações do sistema [3]. Neste trabalho desenvolvemos 2 modelos. O primeiro resultou da partição do nosso sistema em módulos que, pelas suas características funcionais, pudessem ser modelados e simulados de forma ótima no *Ptolemy*. O segundo visou a obtenção de um modelo que pudesse ser usado como ponto de partida de uma implementação real do sistema todo. Ao iniciar o trabalho, dispúnhamos de dois conjuntos de normas referentes as especificações de sinalização de linha e a sinalização entre registradores, para a rede nacional de telefonia via terrestre, descritas numa publicação (41 e 39 páginas respectivamente) do **Sistema de Práticas TELEBRÁS** [5, 6], do *data sheet* do componente **MT9075A** da **MITEL** onde está descrito um algoritmo para o protocolo de sincronização [7] e uma apostila publicada por professores do laboratório de comunicações e sinais da Universidade de São Paulo (USP), que oferece informações fundamentais sobre as partes que compõem um sistema de sinalização de tronco E1 [4].

### 3.1 Modelo visando a captura e simulação do sistema

A fim de capturar as características do sistema criamos um modelo “*híbrido*” que se baseou em 3 domínios: *Synchronous DataFlow* (SDF), *Finite State Machine* (FSM) e *Discrete Event* (DE). O SDF foi adotado como domínio principal, formando o *universo SDF* do sistema (mais alta hierarquia do sistema dentro do *Ptolemy*). Esta escolha deveu-se ao fato de ser compatível com a descrição de fluxo de dados, cujo comportamento é o mais ressaltado no nosso sistema, além de ser o domínio mais maduro e testado, possuindo uma ampla biblioteca de estrelas primitivas, característica que facilita a captura das especificações do sistema com maior eficiência [3]. Os domínios FSM e DE foram utilizados para descrever algumas funções específicas (tarefas), que pelas suas características não podiam ser modeladas no domínio SDF como estrelas ou galáxias independentes.

Com o objetivo de se obter uma representação modular, que facilite a tarefa de partição visando uma implementação ou a procura de um modelo próximo a uma implementação, o sistema foi modelado num nível de tarefas, onde os blocos funcionais que o conformam foram modelados em forma de estrelas e galáxias. Isto implica que alguns comportamentos não possam ser capturados por um único domínio. Portanto é preciso utilizar outros domínios em conjuntos capazes de capturar tais comportamentos. Este é o caso do bloco de sincronização e a comunicação entre os blocos de Sinalização de Linha e Sinalização entre Registradores. A figura 3 mostra a representação do modelo, com os domínios nos quais foram modelados cada bloco.

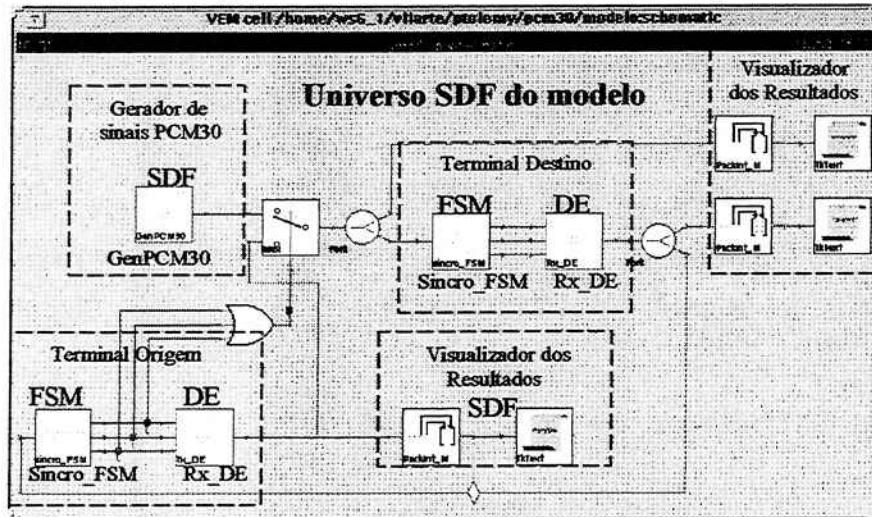


Figura 3. Modelo do sistema nos domínios de simulação do *Ptolemy*.

Para a simulação foi preciso acrescentar o modelo de um bloco gerador de sinais PCM30 para fornecer a entrada de dados do nosso sistema, com sinais de controle predefinidas nos canais 0 e 16. Este bloco foi modelado dentro em uma galáxia, chamada *GenPCM30*, no domínio SDF.

O bloco de sincronização foi modelado no domínio FSM por apresentar um comportamento do tipo máquina de estados. A estrela que o descreve chama-se *Sincro\_FSM*. Esta estrela é uma combinação de FSM com SDF como modelo concorrente [8][9], oferecendo três principais vantagens:

- Heterogeneidade: Onde diversos modelos podem coexistir e interagir pela combinação hierárquica.
  - Modularidade: Diferentes partes de um sistema pode ser modelado separadamente, escolhendo-se a melhor técnica de modelagem apropriada.

- Extensível: Outros modelos concorrentes podem ser incluídos.

Dentro dos estados das máquinas de estados secundárias (vide figura 4) estão presentes diferentes módulos construídos com estrelas e galáxias em SDF, que modelam as partes do bloco de sincronização. A figura 4 mostra as máquinas de estados contidas na estrela *Sincro\_FSM*.

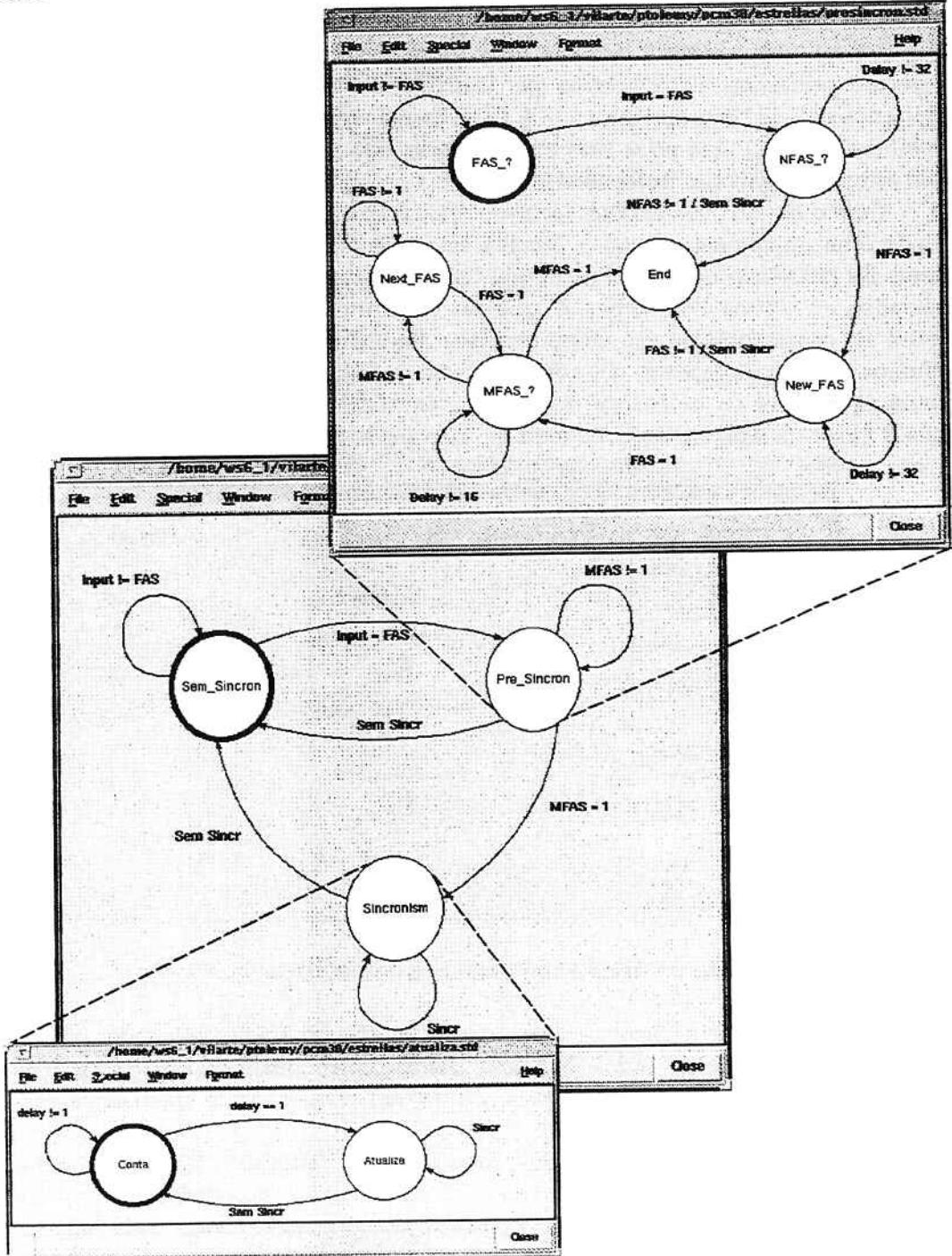


Figura 4. Representação das máquinas de estados dentro da estrela *Sincro\_FSM*.

Os blocos de Sinalização de Linha e Sinalização entre Registradores foram modelados no domínio DE dentro da galáxia *Rx\_DE*. A figura 5 mostra a estrutura interna desta galáxia. O domínio DE foi escolhido especificamente para modelar a comunicação entre estes blocos, cujo

comportamento pode ser identificado como uma comunicação irregular ou assíncrona de dados. Este comportamento está baseado em que a estrela *Signal\_R2\_Digital*, que contém o modelo do bloco de Sinalização de Linha (R2-Digital), troca sinais com a estrela *Register*, quem modela o bloco de Sinalização entre Registradores, de maneira que não necessariamente ambas estrelas possuam todas as entradas prontas para serem executadas. Este comportamento não conseguiu ser modelado em outros domínios testados (SDF, DDF, BDF, FSM), pois eles não cumprem as exigências do mesmo. Este é um ponto interessante no modelo, onde se observa que ao modelar um sistema partindo do nível de tarefa, vão aparecer comportamentos com condições cada vez mais específicas, o que obriga fazer uma escolha mais eficiente dos domínios para capturar tais comportamentos.

Para validar o sistema compelido do bloco Sinalização entre Registradores, foi colocado o mesmo modelo do Sistema fazendo a função de terminal de “Origem” de forma tal que forneça os sinais de entrada do sistema, uma vez alcançada a sincronização. Esta realimentação deve ter uma estrela de demora não nula para evitar ambigüidades no funcionamento do domínio SDF. As outras estrelas que compõem o modelo de simulação formam parte da visualização dos resultados.

O ambiente *Ptolemy* demostrou ser eficiente para validar as especificações do sistema. A ampla biblioteca de estrelas primitivas presentes nos três domínios adotados permitiu a rápida obtenção de um modelo. A representação gráfica ou esquemática das máquinas de estados oferecem uma visão clara da descrição dos blocos com comportamentos deste tipo.

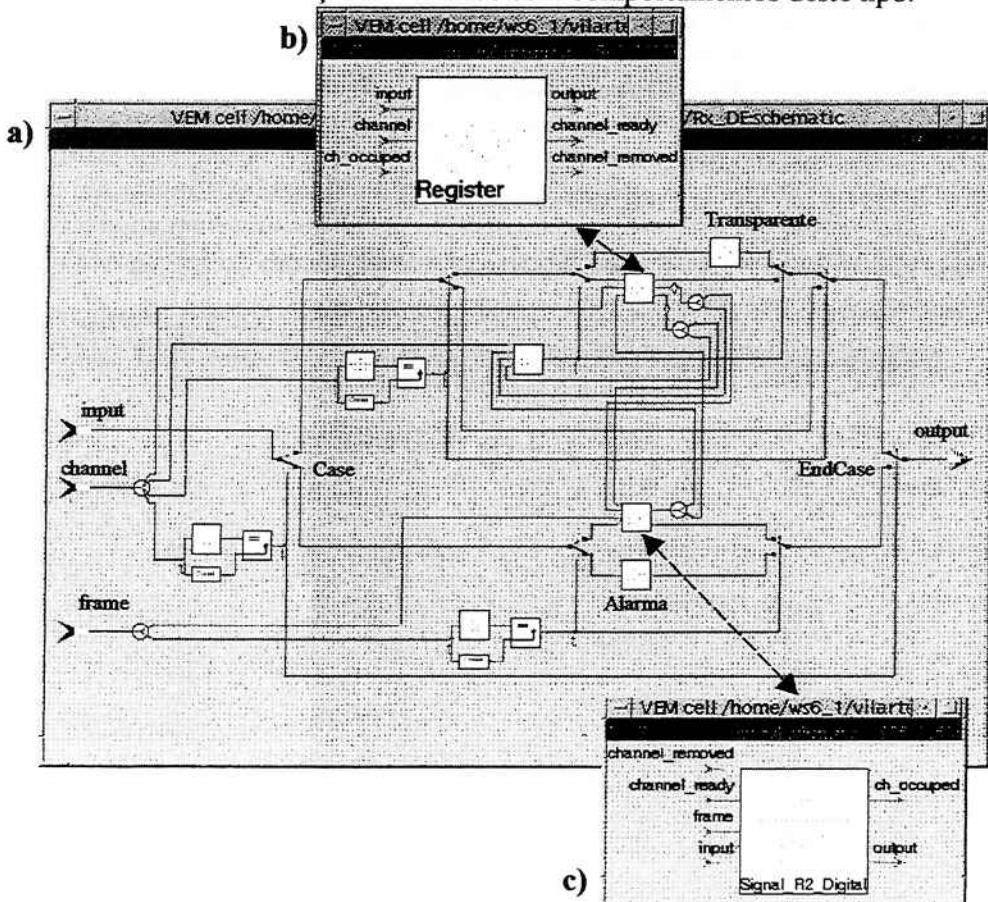


Figura 5. a) estrutura interna da galáxia *Rx\_DE*. b) estrela que modela o bloco de Sinalização entre Registradores. c) estrela que modela o bloco de Sinalização de Linha.

### 3.2 Modelo visando a implementação do sistema.

Uma vez simuladas e verificadas todas as especificações do sistema, partimos então na busca de um segundo modelo a partir do qual pudéssemos fazer uma implementação. Esta exploração é possível devido a variedade de opções (domínios) fornecidas pelo *Ptolemy*. Através dos domínios de geração de código o *Ptolemy* oferece algumas possibilidades para implementar um sistema em Hardware, em Software, ou mistos (*Hardware-Software*). Decidimos partir para uma implementação puramente software. Para isto o *Ptolemy* fornece o domínio CGC, o qual gera os códigos do sistema modelado na linguagem C. Este domínio segue unicamente a semântica do domínio SDF, o que torna mais complicada a captura de todas as especificações do sistema a ser modelado. Isto implica que alguns blocos tenham que ser modelados numa mesma estrela, de maneira que o trabalho manual é incrementado pela quantidade de código que deverá ser digitado.

Em nosso caso, o bloco de Sincronização foi modelado numa estrela chamada *Code FSM*, assim como também foram incluídos os blocos de Sinalização de Linha e Sinalização entre Registradores numa única estrela chamada *Rx*. A figura 6 ilustra o modelo do sistema no domínio CGC.

Ao executar o modelo em CGC, o *Ptolemy* gera um arquivo na linguagem C e outro arquivo executável. Estes arquivos poderiam ter sido uma implementação do sistema em software, mas o código gerado contém, além do código do sistema, os códigos referentes aos blocos geradores de sinais PCM e os blocos visualizadores de resultados. Isto implica que é necessário realizar mais um refinamento para depurar o código do sistema, de maneira que possa ser sintetizável pelas ferramentas apropriadas.

O domínio CGC, que tem como característica fundamental gerar um código C, mostrou-se limitado ao basear-se unicamente na semântica do SDF. Com tudo isto esta segunda modelagem foi mais fácil, pois a maioria dos códigos do modelo de simulação, os quais foram escritos em SDF, serviram como base à realização deste; e só foram mudados aqueles que estavam escritos no FSM e DE. Por outra parte as especificações do modelo validadas em cada rodada foram obtidas corretamente.

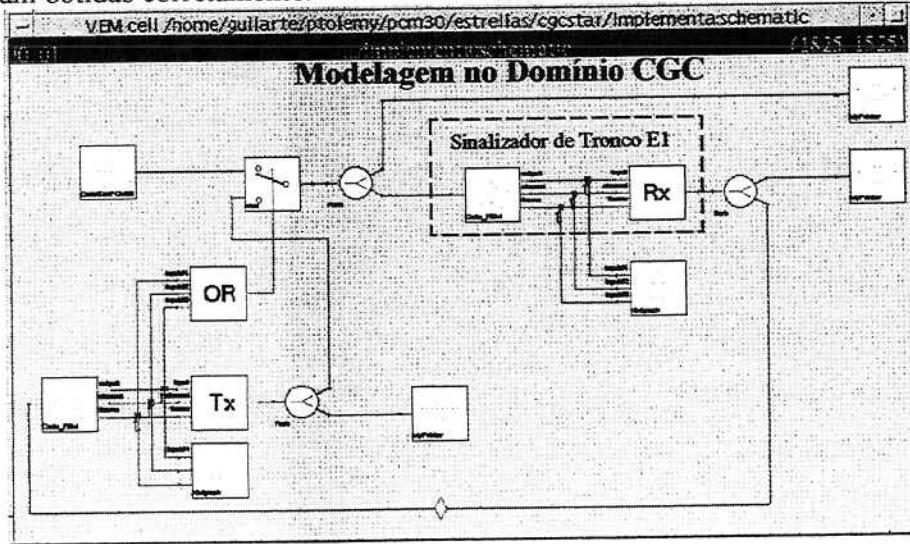


Figura 6. Modelo do sistema no domínio CGC.

#### 4. Discussão.

O modelo “*híbrido*”, criado a partir da documentação disponível, teve o propósito de gerar um instrumento que nos permitisse validar a especificação de forma rápida e eficiente. Esta tarefa é comum e importante como interface de comunicação entre os projetistas de sistemas e os projetistas de sua implementação<sup>1</sup>. Entretanto o modelo mostrou-se ruim para servir de base para uma implementação. Para resolver este problema pode-se adotar a estratégia de **refinamento progressivo**, que consiste em escrever modelos cada vez mais próximos aos usados como base de uma implementação, como por exemplo em VHDL nível RTL para implementação em hardware ou C para software. O segundo modelo foi desenvolvido visando uma implementação puramente software. No nosso trabalho, a existência do modelo *híbrido* facilitou (e acelerou) a criação do segundo modelo. A figura 7 mostra um diagrama de fluxo do processo de refinamento progressivo no *Ptolemy*.

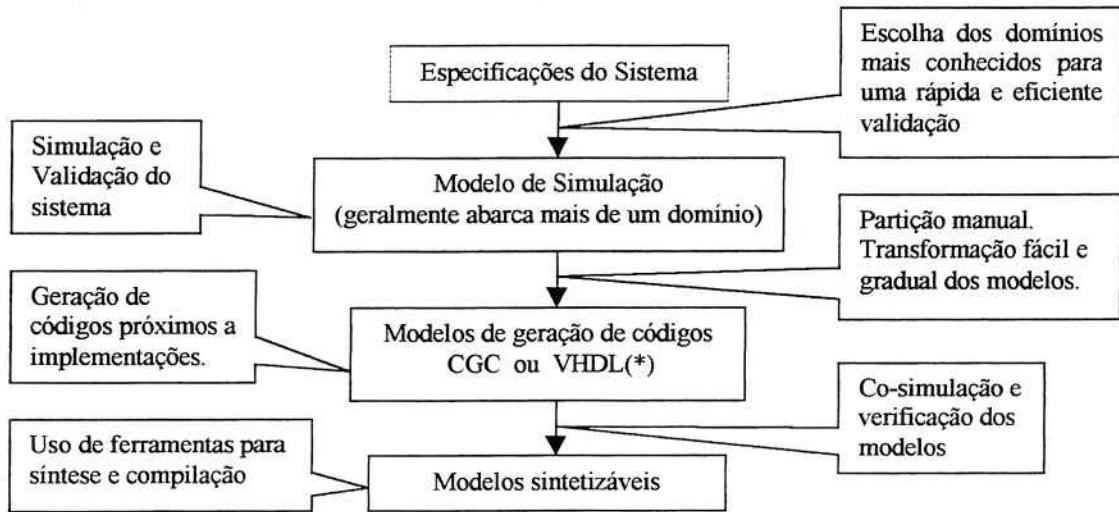


Figura 7. Processo de refinamento progressivo no *Ptolemy* (\* só na versão 0.7).

Componentes básicos de uma especificação no nível de sistema são chamados de tarefas. A descrição de uma aplicação particular a nível de tarefas possui duas características [10]. As tarefas estão num nível alto de abstração, como operações atômicas ou instruções, o que permite que aplicações complexas sejam descritas mais naturalmente. Não existe um critério fixo de como o sistema será implementado, sendo mais flexível para qualquer uma das possíveis aplicações. Entretanto, ao modelar-se um sistema partindo do nível de tarefas, aparecem comportamentos em cada módulo, o que obriga uma exploração dos domínios para sua captura. Cada comportamento pode ser capturado por mais de um domínio. É tarefa do projetista procurar métricas que permitam fazer a escolha certa. O *Ptolemy* fornece uma boa infra-estrutura para a simulação e validação de sistemas, mas as implementações são restritas aos domínios que seguem uma semântica similar ao SDF. Isto torna a tarefa de partição mais complexa.

A modelagem próxima a uma implementação puramente software, apresentada neste artigo, é um simples exercício para explorar um dos domínios de geração de código. Juntando-se vários módulos em um, modelando-os numa mesma estrela, conseguiu-se capturar todos os comportamentos no CGC, mas o modelo ficou complicado para fazer uma partição.

<sup>1</sup> Trata-se da relação cliente/prestador de serviços

## 5. Conclusões.

Apresentamos neste artigo a modelagem de um Sistema de Sinalização de Tronco E1 para telefonia digital, utilizando o *Ptolemy* como ambiente de projeto. Dois modelos foram desenvolvidos, um visando sua captura para fins de simulação e outro visando sua implementação em software. Ambos modelos foram validados através de simulações funcionais que abarcaram todas as especificações de uma operação completa numa central telefônica, obtendo-se bons resultados.

Um ambiente para o projeto automático de sistemas digitais requer ferramentas para realizar uma grande variedade de tarefas. Resumidamente podemos indicar a captura e validação da especificação inicial, partição em módulos de hardware e/ou de software, co-simulação da descrição particionada, definição da interface entre os módulos, síntese dos módulos hardware, software e da interface e validação final.

O resultado deste trabalho demonstra que o ambiente *Ptolemy* é eficiente apenas para a captura e validação das especificações de um sistema, não provendo ferramentas para realizar as demais tarefas de uma metodologia de projeto *top/down*. O fato do *Ptolemy* oferecer uma grande variedade de domínios de simulação e de geração de código, cada um deles sendo baseado em diferentes modelos computacionais, faz com que mesmo a captura da especificação inicial se torne uma tarefa árdua, pois o projetista precisa escolher um ou diversos domínios para gerar o primeiro modelo do seu sistema.

Estamos no momento desenvolvendo um modelo em C que efetivamente sirva de implementação para nosso sistema e estamos também estudando critérios e métricas que auxiliem o projetista a realizar a escolha do(s) domínio(s) mais adequado(s) para capturar as especificações de um sistema, visando tanto sua validação como sua subsequente partição e síntese.

## 6. Referências.

- [1] KALAVADE, A.; LEE, E. Hardware/Software Codesign Using *Ptolemy*. ROZENBLIT, J.; BUCHENRIEDER, K. **Codesign**. IEEE Press, 1995, New York. P.397-413.
- [2] BUCK, J. T. et al. ***Ptolemy 0.7 User's Manual***. California, Department of EECS, University of California at Berkeley, v.1, 1997, (<http://Ptolemy.eecs.berkeley.edu/Ptolemyclassic/almagest/index.html>)
- [3] RUSSO Jr, A. **Uma contribuição ao estudo do projeto “Top-Down” de sistemas digitais sobre silício.** São Paulo, 1998. 153p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo.
- [4] CABRAL Jr, E. F.; ETIENNE, P. J. **Telefonia digital**. São Paulo, PEE/EPUSP, 1998.
- [5] ESPECIFICAÇÕES DE SINALIZAÇÃO DE LINHA PARA A REDE NACIONAL DE TELEFONIA VIA TERRESTRE. **Sistema de Práticas TELEBRÁS.** Conversores de sinalização telefônica. p.01-41, emissão 02, Abr. 1987
- [6] ESPECIFICAÇÕES DE SINALIZAÇÃO ENTRE REGISTRADORES PARA A REDE NACIONAL DE TELEFONIA VIA TERRESTRE. **Sistema de Práticas TELEBRÁS.** Conversores de sinalização telefônica. p.01-39, emissão 02, Abr. 1987
- [7] MT9075A , E1 Single Chip Transceiver. Data sheet. MITEL, Dec. 1997.

- [8] LEE, B.; LEE, E. A. Interaction of Finite Machine and Concurrency Models. **Proceeding of thirty Second Annual Asilomar Conference on Signal, Systems, and Computers**. Pacific Grove, California, Nov. 1998.
- [9] GIRAUT, A.; LEE, B.; LEE, E. A. Hierarchical finite state machines with multiple concurrency models. **Technical Memorandum UCB/ERL M97/57**, Department of EECS, University of California at Berkeley, Berkeley, California, Aug. 1997.
- [10] KALAVADE, A. P. **System-level Codisign of mixed Hardware-Software Systems**. University of California at Berkeley, Berkeley, California, 1996. 194p. Tese (Doutorado).



## BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

- BT/PSI/0001 – Observabilidade Topológica de Osawa em Redes não Lineares – ARMANDO HANDAYA, FLÁVIO A. M. CIPPARRONE
- BT/PSI/0002 – Desenvolvimento de uma Microbalança de Quartz para Detectar Gases – ROBERTO CHURA CHAMBI, FRANCISCO JAVIER RAMIREZ FERNANDEZ
- BT/PSI/0003 – Sistema para Desenvolvimento de Sensores Inteligentes – ANTONIO CARLOS GASparetti, FRANCISCO JAVIER RAMIREZ FERNANDEZ
- BT/PSI/0004 – A 1.6GHz Dual Modulus Prescaler Using the Extended True Single-Phase Clock CMOS Circuit Technique (ETSPC) – JOÃO NAVARRO SOARES JÚNIOR, WILHELMUS ADRIANUS M. VAN NOIJE
- BT/PSI/0005 – Modelamento em Linguagem VHDL de uma Unidade de Policiamento para Redes Locais ATM – ÉDSON TAKESHI NAKAMURA, MARIUS STRUM
- BT/PSI/0006 – Otimização das Operações Coletivas para um Aglomerado de 8 Computadores usando uma Rede Ethernet 10 Mbps baseada em Hub – MARTHA TORRES, SERGIO TAKEO KOFUJI
- BT/PSI/0007 – Short Temporal Coherence Optical Source With External Fiber Optics Cavity – CARMEM LÚCIA BARBOSA, JOSÉ KEBLER DA CUNHA PINTO
- BT/PSI/0008 – Hydrogenated Carbon Films Used as Mask in Wafer Processing With Integrated Circuits: Post-Processing – JUAN M. JARAMILLO O., RONALDO D. MANSANO, EDGAR CHARRY R.
- BT/PSI/0009 – Redes Neurais em VLS – ANTONIO RAMIREZ HIDALGO, FRANCISCO JAVIER RAMIREZ FERNANDEZ
- BT/PSI/0010 – Caracterização de Filmes Obtidos a Partir da Deposição por Plasma de Hexametildissilazana – SANDRINO NOGUEIRA, MARIA LÚCIA PEREIRA DA SILVA
- BT/PSI/0011 – InterFace: A Real Time Facial Animation System – JOSÉ DANIEL RAMOS WEY, MARCELO KNORICH ZUFFO
- BT/PSI/0012 – Análise de Desempenho Dinâmico de Sistemas de Transmissão em Corrente Contínua Multiterminais Série Utilizando GTO – MARIA APARECIDA DOS SANTOS, WALDIR PÓ
- BT/PSI/0013 – Estudo de Dispositivos Miniaturizados para Controle do Escoamento de Fluidos – ELIPHAS WAGNER SIMÕES, ROGÉRIO FURLAN



