

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM
INFORMÁTICA

UMA ABORDAGEM QUALITATIVA PARA
VERIFICAÇÃO FORMAL DE SOFTWARE
EMBARCADO

MARCELO MONTEIRO CUSTÓDIO

MANAUS
2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL DO AMAZONAS INSTITUTO DE
CIÊNCIAS EXATAS PROGRAMA DE PÓS-GRADUAÇÃO EM
INFORMÁTICA

MARCELO MONTEIRO CUSTÓDIO

UMA ABORDAGEM QUALITATIVA PARA VERIFICAÇÃO FORMAL
DE SOFTWARE EMBARCADO

Dissertação apresentada ao Programa
de Pós-Graduação em Informática da
Universidade Federal do Amazonas,
como requisito parcial para a obten-
ção do título de Mestre em Informá-
tica, área de concentração Engenharia
da Computação

Orientador: Prof. Dr. Raimundo Barreto
MANAUS
2008

AGRADECIMENTOS NA VERSÃO FINAL.

RESUMO

Os sistemas embarcados possuem inegável importância na sociedade atual. Eles possuem restrições temporais (quando são de tempo real), de gerência de consumo de energia, tamanho, peso etc que tornam o seu projeto e concepção mais complexos do que os sistemas convencionais. Dado o grande número de requisitos de todos os tipos, a alta complexidade dos softwares embarcados desenvolvidos bem como a grande possibilidade de catástrofes significativas em caso de falha e por fim a grande pressão de mercado por produtos cada vez mais rápido, fazem-se necessários métodos que possam assegurar uma correta, rápida porém intuitiva especificação e concepção dos projetos. O objetivo do trabalho então é prover uma abordagem que colete uma especificação de software embarcado em uma notação semi-formal, orientada a objetos e amplamente aceita pela Indústria, que é a Unified Modeling Language (UML), especificamente com seu Diagrama de Sequência, o qual é apto para capturar os aspectos dinâmicos de um sistema. Prover também um esquema de tradução dos diagramas de sequência em UML para uma notação formal, no caso a notação de Redes de Petri, o qual é adequada para verificação formal, gerando saídas de arquivos nos formatos APNN e PNML. O formato APNN é adequado para ser usado no Model Checking Kit (MCK). Por fim, prover uma interface amigável para coleta de requisitos ou consultas de propriedade em alto nível com a consequente tradução das mesmas para o formato de CTL puro adequado para ser usado no MCK e um programa em SMV e sua especificação em CTL, formatos aptos a serem usados no model checker SMV. A verificação de propriedades é apenas qua-

litativa, isto é, que verificará apenas propriedades de execução do software embarcado, em oposição às propriedades quantitativas de tempo por exemplo, comuns em softwares de tempo-real.

Palavras-chave: Sistemas Embarcados, Model Checking, Redes de Petri, UML.

ABSTRACT

Embedded systems have undeniable relevance in modern society. They have temporal constraints (as long as they are real time ones), power consumption management, size, weight, etc which make their design more complex than the design of their desktop peers. Given the huge number of requirements of all kinds, the high complexity of embedded software as well as the big possibilities of critical damages in case of flaws and, at last, the even bigger pressure of market for new products faster, it make necessary methods which can assure correct, fast but intuitive specification and conception of designs. So the goal of this work é provide an approach which gathers a specification of embedded software in a semi-formal, object-oriented and widely Industry accepted notation, that is Unified Modeling Language (UML). In special, its Sequence Diagram, which is able to capture dynamic aspects of a system. It's to provide also a translation scheme of the UML sequence diagrams to Petri Net notation, which is able to be formally verified, generating files with APNN, PNML and SMV extensions. At last, the present work aims to provide a GUI to colect requirements and proprieties queries and the further automatic translation to CTL, which is able to be used with the chosen model checker used by of this work, the SMV.

Key-words: Embedded Systems, Model Checking, Petri Net, UML.

Lista de Figuras

3.1	Exemplo de Sistema Embarcado: Sistema de Navegação da Espaçonave Apollo da NASA	31
3.2	Palm (r) Treo	31
3.3	Visões da UML	46
3.4	Um Exemplo de Diagrama de Sequência	48
3.5	Elementos Básicos das Redes de Petri	51
4.1	Método Proposto	73
4.2	Notação Diagrama de Sequência Estendida	74
4.3	(Regra 1) Conversão de envio de mensagem.	75
4.4	(Regra 2) Conversão de uma chamada de método	76
4.5	Rede Gerada da Regra 02	77
4.6	(Regra 3) Conversão de uma operação de retorno de método.	78
4.7	(Regra 4) Conversão de uma condição.	79
4.8	(Regra 5) Conversão de concorrência.	80
4.9	(Regra 6) Conversão de Escolha.	81
4.10	(Regra 7) Conversão de sincronismo.	83
4.11	(Regra 8) Conversão de confluência.	84
4.12	Pacote ambiente - parte 1 de 2	90

4.13	Pacote ambiente - parte 2 de 2	91
4.14	Pacote geradores	93
4.15	Pacote modelo.diagramadesequencia	94
4.16	Pacote modelo.programaSMV	94
4.17	Pacote modelo.rededepetri	95
4.18	Pacote parser	96
4.19	Pacote smv	96
4.20	Pacote tradutor	97
5.1	Grafo de Precedências do Simple Control Application	100
5.2	Diagrama de Sequência da Entrada. Parte 1 de 6	103
5.3	Diagrama de Sequência da Entrada. Parte 2 de 6	104
5.4	Diagrama de Sequência da Entrada. Parte 3 de 6	105
5.5	Diagrama de Sequência da Entrada. Parte 4 de 6	106
5.6	Diagrama de Sequência da Entrada. Parte 5 de 6	107
5.7	Diagrama de Sequência da Entrada. Parte 6 de 6	107
5.8	Rede de Petri Completa	109
5.9	Passo 1 - Entrada do diagrama de sequência	111
5.10	Tradução para Rede de Petri	112
5.11	Carregando infra-estrutura	113
5.12	Primeira Consulta: AG S1	115
5.13	Segunda Consulta: EF S1	117
5.14	Terceira Consulta: AG (S1 & S14 & S16 & S18 & S7)	119
5.15	Quarta Consulta: AG (S2 -> AF S19)	121
5.16	Quinta Consulta: EF (S2 & S4 & S6)	123
5.17	Sexta Consulta: AG (S16 & S6 -> AF S23)	125

5.18	Sétima Consulta: AG (S1 -> AG (!S22))	126
5.19	Oitava Consulta: AG (S1 -> EF S22)	129
5.20	Nona Consulta: AG (S22 -> EF S25)	130
5.21	Décima Consulta: A(S1 U AG S14)	132
5.22	Décima Primeira Consulta: A(S1 U EF S14)	134
5.23	Décima Segunda Consulta: A(S1 U EF (!S14))	135
5.24	Décima Terceira Consulta: A(S1 U EX S14)	137
5.25	Décima Quarta Consulta: A(S1 U AX S14)	139

Lista de Tabelas

3.1	Estruturas de Ramificação e Modos Temporais	66
4.1	Formato do Diagrama de Sequência	85
4.2	Infra-estrutura e Arquitetura do Sistema	86
4.3	Formato do Arquivo APNN	87
4.4	Formato do Arquivo PNML	89
4.5	Formato do Arquivo SMV	90

Lista de Siglas

Sigla	Significado
APNN	<i>Abstract Petri Net Notation</i>
CTL	<i>Computation-Tree Logic</i>
INA	<i>Integrated Net Analyzer</i>
PNML	<i>Petri Net Markup Language</i>
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>

Sumário

1	Introdução	17
1.1	Motivação	18
1.2	Objetivos	19
1.2.1	Objetivo Geral	19
1.2.2	Objetivos Específicos	19
1.3	Contribuições	20
1.4	Organização da Dissertação	21
2	Trabalhos Correlatos	22
2.1	Jeng	22
2.2	Fernandes	23
2.3	Eshuis	23
2.4	Bernardi	25
2.5	Yao	25
2.6	Amorin	26
2.7	Bonnefoi	27
3	Revisão Bibliográfica	28
3.1	Sistemas Embarcados	29

3.1.1	Visão Geral	29
3.1.2	Características e Requisitos	32
3.1.3	Desenvolvimento	33
3.2	Unified Modeling Language	37
3.2.1	Breve Sumário	37
3.2.2	História da UML	38
3.2.3	Conceitos UML	42
3.3	Redes de Petri	49
3.3.1	Visão Geral	50
3.3.2	Definição Formal	52
3.3.3	Propriedades Matemáticas	53
3.3.4	Limitações	55
3.3.5	Extensões	56
3.4	Model Checking	59
3.4.1	Introdução aos Métodos Formais	59
3.4.2	Definição	59
3.4.3	Os Benefícios do Model Checking	60
3.4.4	Limitações do Model Checking	61
3.4.5	Problema da Explosão do Espaço de Estados	63
3.4.6	Verificação através de CTL	64
3.4.7	SMV	67
4	Método Proposto	69
4.1	Contexto e Visão Geral	69
4.2	Descrição Detalhada	70
4.3	Notação Diagrama de Sequência Estendida	74

4.4	Regras de Tradução	74
4.4.1	Regra 1. Conversão de envio de mensagem	74
4.4.2	Regra 2. Conversão de uma chamada de método	75
4.4.3	Regra 3. Conversão de uma operação de retorno de método	75
4.4.4	Regra 4. Conversão de uma condição	76
4.4.5	Regra 5. Conversão de concorrência	76
4.4.6	Regra 6. Conversão de Escolha	76
4.4.7	Regra 7. Conversão de sincronismo	82
4.4.8	Regra 8. Conversão de confluência	82
4.5	Formato dos arquivos de entrada	82
4.5.1	Diagrama de Sequência	82
4.5.2	Infra-estrutura do Sistema	85
4.6	Formato do arquivos de saída	87
4.6.1	Arquivo APNN	87
4.6.2	Arquivo PNML	87
4.6.3	Arquivo SMV	88
4.7	Projeto do Ambiente	88
4.8	Definição das consultas	92
5	Experimento	98
5.1	Descrição	98
5.2	Diagrama de Sequência da Entrada	102
5.3	Aplicando-se as Regras de Tradução	103
5.3.1	Características da Rede Gerada	108
5.4	Utilizando a Ferramenta para Realizar o Método	110

5.5	Consultas de Propriedades do Sistema	114
5.5.1	Consulta I	114
5.5.2	Consulta II	116
5.5.3	Consulta III	118
5.5.4	Consulta IV	120
5.5.5	Consulta V	122
5.5.6	Consulta VI	124
5.5.7	Consulta VII	124
5.5.8	Consulta VIII	128
5.5.9	Consulta IX	128
5.5.10	Consulta X	131
5.5.11	Consulta XI	133
5.5.12	Consulta XII	133
5.5.13	Consulta XIII	136
5.5.14	Consulta XIV	138
6	Considerações Finais	140
6.1	Conclusões	140
6.2	Contribuições	141
6.3	Trabalhos Futuros	142
A	Arquivo XML de Entrada da Ferramenta com o Diagrama de Sequência do Experimento	147
B	Arquivo XML de Entrada da Ferramenta com a Infra-estrutura da Aplicação do Experimento	150

C	Arquivo APNN Gerado no Experimento	153
D	Arquivo PNML Gerado no Experimento	158
E	Arquivo SMV Gerado no Experimento	206

Capítulo 1

Introdução

Os sistemas embarcados possuem inegável importância na sociedade atual. Eles proporcionaram grande melhoria de qualidade da vida moderna com aplicações como freios ABS, cirurgia com robôs, injeção eletrônica de combustível, casas inteligentes, sistemas de navegação de veículos, assistentes pessoais digitais (*Personal Digital Assistants* - PDAs) e aparelhos celulares, fornos micro-ondas, computadores vestíveis etc e promoveram um igualmente grande impulso econômico. Suas maiores características são a ubiquidade e pervasividade.

O fenômeno no avanço na área de sistemas embarcados se deu basicamente devido aos avanços da tecnologia de sistemas micro-processados, bem como sua redução de preço. Tais avanços iniciaram-se na década de 1980 e resultaram em novas concepções de produtos.

Os sistemas embarcados possuem restrições temporais (quando são de tempo real), de gerência de consumo de energia, tamanho, peso etc que tornam o seu projeto e concepção mais complexos do que os sistemas convencionais. Além disso, existem vários desafios de pesquisa e requisitos não

funcionais a serem alcançados, tais como *context-awareness*, ou a capacidade de perceber o contexto ao redor e agir de acordo com a situação, inteligência, interação natural, personalização, conectividade, composição de componentes, i.e. se são construídos a partir de componentes "COTS" (*components off-the-shelf*). Porém, certamente o mais importante diz respeito à **dependabilidade** do sistema, ou seja, sua confiabilidade no que diz respeito aos tempos de resposta (se for um sistema de tempo real) e também nos aspectos de segurança, disponibilidade, manutenibilidade e robustês.

Com o tempo surgiram novas demandas de aplicações embarcadas para o mesmo *chip*, exigindo, portanto, flexibilidade de configuração e manutenção. E tudo isso feito em um tempo cada vez menor. O software veio como a solução para esses problemas. A desvantagem primária de mover as funcionalidades para software é que elas executam reconhecidamente menos rápido do que as equivalentes em hardware.

1.1 Motivação

Dado o grande número de requisitos de todos os tipos, a alta complexidade dos softwares embarcados desenvolvidos bem como a grande possibilidade de catástrofes significativas em caso de falha e, por fim, a grande pressão de mercado por produtos cada vez mais rápido, fazem-se necessários métodos que possam assegurar uma correta, rápida ainda que intuitiva especificação e concepção dos projetos.

1.2 Objetivos

1.2.1 Objetivo Geral

Preencher a lacuna de insuficiência de corretude e formalismos na área de desenvolvimento de software embarcado e não embarcado de maneira geral.

O presente trabalho visa unir o melhor dos dois mundos permitindo: (1) A especificação do software embarcado através dos Diagramas de Sequência da UML, uma notação bastante intuitiva amplamente adotada pela Indústria e Academia; (2) A verificação de propriedades através da técnica de Verificação de Modelos ou *Model Checking*, após uma tradução do modelo para Rede de Petri, uma notação adequadamente rigorosa mas não tão intuitiva quanto a da UML. Além disso, prover uma interface amigável para capturar consultas de propriedades em linguagem natural bem como sua posterior tradução para a Lógica da Árvore de Computação ou *Computation Tree Logics* (CTL).

1.2.2 Objetivos Específicos

Prover uma abordagem que colete uma especificação de software embarcado em uma notação semi-formal, orientada a objetos e amplamente aceita pela Indústria, que é a Unified Modeling Language (UML), especificamente com seu Diagrama de Sequência, o qual é apto para capturar os aspectos dinâmicos de um sistema. É também objetivo deste trabalho prover um esquema de tradução dos diagramas de sequência em UML para uma notação formal, no caso, a notação de Redes de Petri, gerando saídas de arquivos nos formatos APNN e PNML. O formato APNN é adequado para ser usado no *model checker* Model Checking Kit (MCK). Já o formato PNML é amplamente

aceito pela comunidade acadêmica e tem sido cada vez mais usado também por ser baseado em XML, um padrão da Indústria pra comunicação de dados e troca de informações entre ferramentas de software. Por fim, prover uma interface amigável para coleta de requisitos ou consultas de propriedade em alto nível com a consequente tradução das mesmas para o formato de CTL puro adequado para ser usado no MCK e um programa em SMV e sua especificação em CTL, formatos aptos a serem usados no model checker SMV. A verificação de propriedades é apenas qualitativa, isto é, que verificará apenas propriedades de execução do software embarcado, em oposição às propriedades quantitativas de tempo por exemplo, comuns em softwares de tempo-real.

1.3 Contribuições

Uma lista das principais contribuições realizadas na ferramenta desenvolvida no presente trabalho:

- Abordagem de coleta de requisitos de em linguagem semi-formal, orientada a objetos e padrão da Indústria, no caso, a UML.
- Tradução automática para modelo formal, no caso Redes de Petri.
- Verificação desse modelo via model checking.
- Escrita em arquivo da Rede de Petri gerada nos formatos APNN e no cada vez mais usado PNML.
- Escrita em arquivo da Rede de Petri gerada no formato SMV.

- Interface amigável para coleta das consultas em alto nível.
- Tradução automática dessas consultas para CTL.
- Todos os pontos anteriores aplicados a software embarcado.

1.4 Organização da Dissertação

O presente trabalho está organizado como segue. De início, é mostrada uma revisão bibliográfica dos principais conceitos relacionados, quais sejam: Sistemas Embarcados, Redes de Petri, UML - especificamente os Diagramas de Sequência - e, por fim, Métodos Formais e Model Checking. A seguir, o método de verificação formal de software embarcado proposto neste trabalho é apresentado. A fim de comprovar as descobertas e fundamentos do trabalho, é apresentado um experimento com uma aplicação simples de controle na sequência, onde são realizadas 14 consultas de propriedades do sistema. Por último, são tecidas as considerações finais, que incluem as conclusões do trabalho, as contribuições do mesmo e as possíveis melhorias para trabalhos futuros.

Capítulo 2

Trabalhos Correlatos

2.1 Jeng

O primeiro trabalho correlato é o de Jeng [21] cujas regras de tradução UML para Rede de Petri são a base das regras usadas no presente trabalho. Naquele trabalho, tomava-se como entrada um diagrama de sequência da UML estendido e, através de composição de blocos, chegava-se à Rede de Petri Lugar/Transição final. O diagrama de sequência estendido é aumentado de alguns elementos descritivos que visam a aumentar o poder expressivo da notação permitindo ou ressaltando a presença de construções tais como sincronismo, confluência, escolha etc. A notação e as regras são apresentadas no capítulo 4, sobre o Método Proposto, a partir da seção 4.3.

Tanto a notação de diagrama de sequência estendido de Jeng quanto a de Rede de Petri Lugar/Transição são usadas também no presente trabalho. Porém, no trabalho de Jeng, a abordagem de tradução foi usada para provar a eficiência de um algoritmo de *slicing* para escalonamento de tarefas e, além disso, nenhuma verificação formal foi realizada.

2.2 Fernandes

O trabalho de Fernandes [13] usa um estudo de caso sobre uma especificação de um controle de elevador. O artigo apresenta uma abordagem que pode traduzir uma dada descrição em UML em uma Rede de Petri Colorida (RPC). No presente trabalho, usam-se as Redes de Petri Convencionais. As descrições em UML devem estar especificadas na forma de Casos de Uso e de Diagramas de Sequência da UML 2.0. O modelo RPC constitui-se de uma representação única, coerente e executável de todos os comportamentos possíveis que são especificados pelos dados artefatos da UML. RPCs constituem-se de uma linguagem de modelagem formal que habilita a construção e análise de modelos de comportamento escaláveis e executáveis. Um uso combinado de UML e RPC pode ser útil em muitos projetos. O trabalho de Fernandes também se propõe a construir ferramentas de software para realizar a referida tradução.

2.3 Eshuis

Há também o de Eshuis [29]. Duas traduções de diagramas de atividade para a linguagem de entrada do NuSMV, um Model Checker simbólico, são apresentadas. Ambas traduções mapeiam um diagrama de atividades em uma máquina de estados finita e são inspiradas pela semântica de Statecharts existentes. As traduções em nível de requisitos definem máquinas de estados que podem ser verificadas eficientemente, apesar de assumirem uma hipótese de sincronismo perfeito. A tradução em nível de implementação define máquinas de estado que não podem ser verificadas tão eficientemente,

mas que são mais realísticas visto que não se valem valer de uma hipótese de perfeito sincronismo. A fim de justificar o uso de tradução em nível de requisitos, é mostrado no trabalho de Eshuis que para um grande conjunto de diagramas de atividade e determinadas propriedades, ambas as traduções são equivalentes: independentemente de qual tradução é utilizada, a saída do *model checking* é a mesma. Ademais, para algumas propriedades lineares, a tradução em nível de implementação é equivalente a uma versão um pouco modificada da tradução em nível de requisitos. Naquele trabalho, ambas as traduções são usadas para se verificar restrições de integridade de dados para um diagrama de atividades e um conjunto de diagramas de classe da UML que especificam os dados manipulados nas atividades.

Os diagramas de atividade da UML descrevem o ordenamento global de pedaços atômicos do comportamento, conhecidos como atividades. A notação é inspirada tanto por gráficos de fluxo quanto por redes de Petri. Diagramas de Atividade permitem a especificação de processos complexos de software que possuem paralelismos, laços e comportamento orientado a eventos. Eles podem ser usados para o comportamento de alguns casos de uso ou especificar o *workflow* ou processo de negócios de uma organização. Recentemente, eles foram adotados por diversas padrões de serviços da *web* para especificar uma conversação na mesma. O diagramas de atividade são também uma notação da UML para expressar aspectos dinâmicos do sistema. O presente trabalho usa outra notação para aspectos dinâmicos, a de diagram de sequência.

2.4 Bernardi

Vale citar o trabalho de Bernardi [2], que estuda o uso dos Diagramas de Sequência e Statecharts da UML para a validação e avaliação de desempenho de sistemas. É assumido que o sistema é especificado como um conjunto de Statecharts e que os Diagramas de Sequência são usados para representar “execuções de interesse”. A UML carece da falta de uma semântica formal e, por isso, não é possível aplicar, diretamente, técnicas matemáticas em seus modelos para fins de validação de sistema. Para alcançar essa meta, o trabalho de Bernardi propõe uma tradução automática de *Statecharts* e Diagramas de Sequência - também usados no presente trabalho - para Redes de Petri Generalizadas e Estocásticas e uma composição dos modelos de rede resultantes adequados para alguma meta de análise. A tradução para redes de Petri é baseada na sintaxe abstrata das colaborações da UML e em seu pacote de máquinas de estado (a partir do qual *Statecharts* e Diagramas de Sequência são derivados) e a construção de um modelo completo utiliza extensivamente composição.

2.5 Yao

O trabalho de Yao [37] visa ajudar o desenvolvimento de aplicações de software de alta qualidade por meio de uma abordagem para verificação de consistência de modelos dinâmicos da UML baseados em técnicas de Redes de Petri. A notação de Redes de Petri Coloridas Estendidas (RPCE) é usada para formalmente descrever transições de estados de objetos individuais e interações entre esses objetos e é, portanto, capaz de verificar a consistência

de modelos baseados na teoria de redes de Petri. No trabalho de Yao, são considerados, assim como no de Bernardi, diagramas de sequência e *statecharts*. A abordagem começa com uma estratégia para modelos dinâmicos da UML e então discute a tradução de *statecharts* com estados compostos para a notação RPCE. O grafo de alcançabilidade é gerado para conduzir o processo de verificação de consistência. O artigo discute todas as fases da abordagem e ilustra o conceito por meio de um exemplo.

2.6 Amarin

O trabalho de [1] apresenta uma metodologia para mapear a linguagem dos *Live Sequence Charts* (LSCs) para um modelo em Rede de Petri (RdP) equivalente como uma abordagem para análise e verificação de propriedades de sistemas embarcados. A linguagem LSC é uma linguagem para especificação de sistemas, permitindo ao projetista especificar o que deveria acontecer para todas as possibilidades de execução de um sistema, bem como modelar anti-cenários. Entretanto, a análise e verificação de propriedades do sistema não são possíveis. A fim de permitir o diagnóstico de especificações inconsistentes, além da simulação, verificação e análise devem ser consideradas. Portanto, a proposição de um modelo RdP para LSC permite a verificação e análise de sistemas descritos em LSC contribuindo para o aumento da confiança do projetista no processo de desenvolvimento do sistema e redução dos riscos que podem levar a falhas no projeto.

2.7 Bonnefoi

O trabalho de Bonnefoi [3] é sobre a aplicação de métodos formais para modelar e analisar sistemas complexos no contexto do Sistema de Transporte Inteligente - *Intelligent Transport Systems* (ITS). Ele sugere que uma metodologia de especificação baseada num conjunto de diagramas da UML para gerar um conjunto completo de modelos analisáveis. a metodologia integra os requisitos do desenvolvimento incremental e modular de sistemas complexos. A análise feita no modelo formal é conduzida por critérios qualitativos, tal como no método proposto no presente trabalho, é verificada por ferramentas de model checking. As orientações propostas são ilustradas por um estudo de caso o qual considera carros em situações de tráfico, trocando informações sobre seus estados para alcançar consistência entre suas decisões de condução.

Por fim, há também o trabalho do grupo Precise UML [16] o qual visa anexar anotações semânticas em OCL, principalmente, aos diagramas a fim de trazer um rigor maior à especificação.

O próximo capítulo, o de Revisão Bibliográfica, introduz os principais conceitos usados no presente trabalho como fundamentos para o seu desenvolvimento.

Capítulo 3

Revisão Bibliográfica

Este capítulo introduz os principais conceitos necessários para se entender esta dissertação. É dividida em 4 seções: sistemas embarcados, Unified Modeling Language - UML, Redes de Petri e Métodos Formais. A primeira seção mostra as principais características, aplicações e limitações dos sistemas embarcados. Porém, o foco desta seção é software embarcado, exatamente o objeto de estudo do presente trabalho. A seguir, a seção sobre UML detalha a importância da mesma e foca nos Diagramas de Sequência, uma das notações para se conceber a visão dinâmica de um sistema. Tais diagramas são usados no trabalho como notação para descrição em alto nível de aplicações. Na sequência, o capítulo sobre Redes de Petri introduz este importante formalismo matemático, o qual especifica rigorosamente sistemas críticos. Por fim métodos formais são apresentados, em especial a técnica de Model Checking ou Verificação de Modelos.

3.1 Sistemas Embarcados

Sistemas Embarcados estão em todas as partes, desde eletrodomésticos a espaçonaves. Atualmente, a grande parte dos sistemas tem um sistema funcionando embarcado em si. Alguns produtos até tornaram-se possíveis somente graças aos sistemas computacionais que são integrados a eles, por exemplo, telefones celulares, injeção de combustível eletrônica, freios ABS e assim por diante. Este fenômeno está de alguma forma relacionado aos avanços da tecnologia e redução de preços dos sistemas microprocessados iniciado nos anos 1980, os quais resultaram em uma nova concepção de produtos. As Figuras 3.1 e 3.2 mostram exemplos de sistemas embarcados.

3.1.1 Visão Geral

Em geral, um sistema embarcado é um sistema digital especializado o qual executa um grupo de funções dedicadas dentro de um sistema maior de tal forma que novas funcionalidades são adicionadas ao sistema externo ou as existentes são otimizadas. Tipicamente, sistemas embarcados consistem de processadores de prateleira de propósito geral (os "COTS"), ASICs e/ou FPGAs. Eles usam um computador, mas não são nem usados nem percebidos como tal (característica marcante da Ubiquidade). Virtualmente, todas as aplicações que possuem uma interface digital, tal como televisores, microondas, aparelhos de vídeo-cassete, carros etc possuem um sistema embarcado. Alguns sistemas embarcados incluem um kernel de sistema operacional, mas muitos são tão especializados que toda a lógica é implementada como um único programa. Os sistemas embarcados e a computação de mesa diferem substancialmente no que diz respeito às restrições de projeto. Por exemplo,

adicionalmente à CPU e à hierarquia de memória, há uma grande variedade de interfaces que habilitam o sistema a medir, manipular e interagir com o ambiente externo. Os sistemas embarcados tipicamente têm restrições apertadas tanto de funcionalidade quanto de implementação. Em particular, eles devem garantir confiabilidade de uma aplicação, baixo custo, requisitos de tempo-real, pequeno tamanho, baixo peso, longo ciclo de vida, baixo consumo de energia dentre outros. Certamente, tais restrições fazem o bem sucedido projeto desses sistemas uma tarefa difícil de ser completada pelas metodologias tradicionais de desenvolvimento de sistemas. Outra característica tradicional dos sistemas embarcados é que eles usualmente não operam em um ambiente estritamente controlado. Calor excessivo também é frequentemente um problema. Alguns sistemas necessitam de proteção contra vibração, choques, iluminação, flutuações no fornecimento de energia, água, corrosão, fogo e diversas outras anormalidades físicas. O software em sistemas embarcados é muito mais restringido do que o da computação de propósito geral. Por exemplo, o software embarcado não pode usar alocação dinâmica de memória não-restritiva nem memória virtual. Para algumas aplicações altamente críticas, até o uso de pilhas pode ser proibitivo. Os projetistas têm que lidar com vários dilemas visto que os sistemas embarcados têm aumentado sua complexidade e, ao mesmo tempo, as pressões do mercado por resultados mais rápidos têm diminuído o *time-to-market*. A fim de lidar com esses requisitos restritivos, metodologias de desenvolvimento apropriadas têm desempenhado um grande papel.



Figura 3.1: Exemplo de Sistema Embarcado: Sistema de Navegação da Espaçonave Apollo da NASA



Figura 3.2: Palm (r) Treo

3.1.2 Características e Requisitos

Aqui, um resumo das principais características dos sistemas embarcados comumente encontrados no Mercado:

- Especificidade de aplicação. A aplicação é conhecida antes de o projeto iniciar e flexibilidade é importante (atualizações, diferenciação, reuso).
- Processamento digital de sinais. Digitalização dos sinais incrementam o uso de *Digital Signal Processor* - DSP nos Sistemas Embarcados.
- Reatividade. É transformacional se transforma um conjunto de dados de entrada em um conjunto de dados de saída; é interativa se reage ao ambiente em sua própria velocidade ou reativa se reage continuamente ao ambiente na velocidade do próprio ambiente.
- Distributividade. Processos comunicantes em diversos processadores e hardware dedicado conectado por *links* de comunicação.
- Tempo-Real. Tarefas sob restrições de *deadline*, ou prazo de finalização, brando ou rígido. Brando se não implicar em perdas ou danos caso não seja alcançado. Rígido em caso contrário.
- Criticalidade. O grau de criticalidade depende das conseqüências que o desvio do comportamento normal possa causar ao ambiente.

Sistemas embarcados têm restrições apertadas, ou seja, têm que garantir operações reativas a eventos externos em tempo-real, de acordo com limites de tamanho e peso, consumo de energia e refrescância (*cooling*), satisfazer requisitos de segurança e confiabilidade e tudo isso com um custo bastante apertado.

Sistemas embarcados não operam em um ambiente estritamente controlado. Calor excessivo é um problema. Alguns sistemas precisam de proteção contra vibração, amortecimento, iluminação, flutuação no fornecimento de energia, água, corrosão, fogo e muitas outras anomalias físicas.

O software em sistemas embarcados também é muito mais restrito do que em computação tradicional. Em geral, software embarcado não pode usar alocação de memória dinâmica, nem memória virtual. Para algumas aplicações altamente críticas, até mesmo o uso de pilha pode ser proibido.

3.1.3 Desenvolvimento

O desenvolvimento de sistemas embarcados tem uma quantidade de características que diferem substancialmente do desenvolvimento de aplicações convencionais. O relacionamento de hardware e software de um sistema embarcado impacta diretamente o processo de desenvolvimento do sistema através de vários aspectos no ciclo de vida [7]. Sistemas embarcados possuem também severas restrições temporais e limitações físicas que devem ser levados em consideração no processo de projeto do sistema. Em termos gerais, algumas diferenças com aplicações desktop que influenciam o processo de desenvolvimento são apresentadas a seguir [7]:

- A interface homem-máquina que consiste de dispositivos de entrada e saída não deve solicitar nenhum treinamento para operar o sistema. Em outras palavras, deve ser de fácil uso por parte do usuário.
- O custo de uma simples unidade de sistema embarcado deve ser a menor possível com o propósito de reduzir custos de produção. Deste modo, o

projeto do sistema deve ser altamente otimizado para o custo do ciclo de vida e eficiência.

- O sistema embarcado tem funcionalidade fixa e estrutura rígida. O software é específico para a aplicação e reside em uma memória somente de leitura.
- A qualidade do software deve ser alta, pois não existe muita flexibilidade para mudar o software depois de ser liberado para o mercado.
- O tempo de vida de sistemas embarcados é frequentemente longo. Desta forma, é necessário uma boa porta de diagnóstico para que seja possível realizar manutenção em campo.
- Do ponto de vista do valor de negócio agregado, sistemas embarcados não são vendidos somente porque eles contam com um microprocessador potente. Sistemas embarcados são tipicamente vendidos por que eles fornecem as funcionalidades, qualidade e custo que o cliente procura. Além disso, o tempo para identificar a oportunidade de venda de produto e o tempo para lançá-lo no mercado (*time-to-market*) podem ser de extrema importância para as organizações.

Software Embarcado

O objetivo geral do projeto de software embarcado é implementar um conjunto específico de funções enquanto satisfaz restrições como desempenho, custo, consumo de energia, tamanho e peso. Estas funções podem ser implementadas como um componente de hardware ou software rodando num

componente programável. Atualmente, as funcionalidades de sistemas embarcados têm crescido em número e complexidade de tal forma que o tempo de desenvolvimento tornou-se difícil de se prever e de se controlar. Esta complexidade forçou os projetistas a levar em consideração implementações flexíveis. Ademais, os ciclos de manufatura de hardware são mais caros e consomem mais tempo. Dai, a implementação baseada em software tornou-se uma solução alternativa viável. Neste caso, o crescente poder computacional e as correspondentes reduções no tamanho e preço do processador permitiram mover mais e mais a funcionalidade para o software. Ademais, implementações baseadas em software provêm maior grau de flexibilidade do que as implementações baseadas em hardware e portanto é mais fácil atender à restrição de *time-to-market*. Análises recentes de mercado indicam que o software responde por mais de 80% do desenvolvimento do sistema. Por isso, a fim de se tornar competitivo, as companhias têm de ter um poderoso ambiente de desenvolvimento de software.

Problemas com o software embarcado

Sangiovanni-Vincentelli e Martin [31] apontaram vários problemas com o software embarcado, os quais são resumidos a seguir. O primeiro é o menor desempenho em relação ao hardware. A fim de superar isso, alguns programadores usam Assembly ou C. Entretanto, esta prática pode afetar o tempo de desenvolvimento - e, conseqüentemente, o *time-to-market*, a legibilidade e a manutenibilidade do software resultante. Em geral, software embarcado também necessita de suporte em hardware para depuração e avaliação de

desempenho. Muitas companhias têm adotado métodos orientados a objeto. Tais métodos são certamente importantes mas não suficientes para garantir qualidade e atender ao *time-to-market*. Outra desvantagem clássica é a crescente dificuldade em se verificar a corretude do projeto. Esta verificação é crítica devido às considerações de segurança em muitos domínios de aplicações. Adicionalmente, pouca atenção tem sido dada para restrições de *deadlines* estritos, pouco uso de memória e consumo de energia do software.

Desafios Concernentes às Metodologias de Projeto de Software Embarcado

De acordo com Sangiovanni-Vincentelli e Martin [32] também, uma metodologia de desenvolvimento de software para sistemas embarcados tem de considerar os seguintes desafios:

- Reuso.
- Hardware/software codesign.
- Modelar propriedades não funcionais.
- Extensivo uso de componentes de software.
- Arquitetura de sistema e de software.
- Validação e verificação em nível de sistema.
- dentre outros.

3.2 Unified Modeling Language

3.2.1 Breve Sumário

A Linguagem Unificada de Modelagem (*Unified Modeling Language* - UML) é uma linguagem visual de propósito geral que é usada para especificar, visualizar, construir e documentar os artefatos de um sistema de software. Ela captura decisões e entendimentos sobre os sistemas a serem construídos. É usada para se entender, projetar, navegar em, configurar, manter e controlar a informação sobre os sistemas. Deve ser usada com todos os métodos de desenvolvimento, estágios do ciclo de vida, domínios de aplicações e mídias. A linguagem de modelagem tem a capacidade de unificar a experiência passada sobre técnicas de modelagem e incorporar as melhores práticas de software atuais em uma abordagem padrão. A UML inclui conceitos semânticos, notações e guias. Ela tem partes estáticas, dinâmicas, de ambiente e organizacionais. Ela tem a capacidade de suportar através de ferramentas de modelagem visual interativa que têm geradores de código e geradores de relatórios. A especificação UML não define um processo padrão mas é particularmente útil com um processo de desenvolvimento iterativo. Ela também tem a capacidade de suportar a maioria dos processos de desenvolvimento orientado a objeto. A UML captura informação sobre a estrutura estática e o comportamento dinâmico de um sistema. Um sistema é modelado como uma coleção de objetos discretos que interagem a fim de realizar um trabalho que, em última instância, beneficia um usuário externo. A estrutura estática define os tipos de objetos que são importantes para o sistema e suas implementações, assim como as relações entre os objetos. O comportamento dinâmico

define a história dos objetos sobre o tempo e as comunicações sobre objetos para alcançar suas metas. Modelar um sistema a partir de vários pontos de vista separados porém relacionados permite que ele seja entendido para diferentes propósitos. A UML também contém construções organizacionais para se arranjar modelos em pacotes que permitem a equipes de software particionar sistemas grandes em partes facilmente trabalháveis, a fim de se entender e controlar as dependências entre os pacotes e gerenciar o versionamento das unidades de modelo em um complexo ambiente de desenvolvimento. Ele contém construções para se representar as decisões de implementação e para se organizar elementos de tempo de execução em componentes. A UML não é uma linguagem de programação. As ferramentas podem prover geradores de código de UML em uma variedade de linguagens de programação, bem como construir modelos a partir de Engenharia Reversa de programas existentes. A UML não é uma linguagem altamente formal criada para prova de teoremas. Há várias linguagens para isso, mas as mesmas não são fáceis de se entender ou usar para muitos propósitos. A UML é uma linguagem de modelagem discreta. Não deve ser usada para se modelar sistemas contínuos tais como os encontrados em Engenharia e na Física. A UML é projetada para ser um linguagem de modelagem de propósito geral para sistemas de eventos discretos tais como aqueles construídos por software, firmware ou lógica.

3.2.2 História da UML

A UML foi desenvolvida num esforço para se simplificar e se consolidar um grande número de métodos de desenvolvimento orientados a objeto que emer-

giram.

Métodos de desenvolvimento orientados a objeto

Métodos de desenvolvimento para linguagens de programação tradicionais, tais como Cobol e Fortran, emergiram nos anos 1970 e se massificaram nos anos 1980. Inicialmente, entre eles havia a Análise Estruturada e o Projeto Estruturado [38] e suas variantes, tais como o Projeto Estruturado de Tempo-real [35] e outros. Esses métodos originariamente desenvolvidos por Constantine, DeMarco, Mellor, Ward, Yourdon e outros, alcançaram alguma penetração na área de grandes sistemas, especialmente sistemas encomendados por órgãos governamentais nos EUA nas áreas aeroespaciais e de defesa, nas quais os oficiais insistiram em um processo de desenvolvimento organizado e com ampla documentação do projeto e implementação. Os resultados não foram tão bons quanto se esperava - muitas ferramentas de Engenharia de Software Apoiada por Computador (*Computer-aided software engineering* (CASE)) eram nada mais do que simples geradores de relatórios que extraíam projetos após a implementação estar completa - mas os métodos incluíram boas idéias que foram ocasionalmente usadas efetivamente na construção de grandes sistemas. Aplicações comerciais foram mais relutantes na adoção de ferramentas CASE e métodos de desenvolvimento. A maioria das organizações desenvolveram software internamente para suas próprias necessidades, sem a característica relação entre cliente e contratado que marcaram os grandes projetos governamentais. Sistemas comerciais foram encarados como mais simples então, seja isto verdade ou não. A primeira linguagem

orientada a objeto é reconhecidamente Simula-67, desenvolvida em 1967. Esta linguagem nunca teve uma continuidade significativa, apesar de que ela grandemente influenciou os desenvolvedores de muitas outras linguagens orientadas a objeto posteriores. O movimento orientado a objeto tornou-se ativo com a popularização da linguagem Smalltalk no início dos anos 1980, seguida por outras linguagens orientadas a objeto, tais como Objective C, C++, Eiffel e CLOS. O real uso de linguagens orientadas a objeto era limitado de início, mas a Orientação a Objetos atraiu muita atenção. Aproximadamente cinco anos depois, Smalltalk já tinha se tornado largamente conhecida, os primeiros métodos de desenvolvimento orientados a objeto foram publicados por Shlaer/Mellor [33] e Coad/Yourdon [6], seguidos de perto pelos livros de Booch [4], Rumbaugh/Blaha/Premerlani/Eddy/Lorensen [20] e Wirfs-Brock/Wilkerson/Wiener [30]. Estes livros, adicionados aos primeiros livros sobre projeto de linguagens de programação de Goldberg/Robson [14], Cox [8] e Meyer [26], iniciaram o campo de metodologias orientadas a objeto. A primeira fase foi concluída pelo fim de 1990. O livro do método “Objectory” [19] foi publicado um pouco depois, baseado no trabalho que apareceu em artigos anteriores. Tal livro adota uma abordagem um tanto quanto diferente, com seu foco em casos de uso e no processo de desenvolvimento. Nos próximos cinco anos, um grande número de livros sobre metodologias OO apareceu, cada qual com seu próprio conjunto de conceitos, definições, notações, terminologias e processo. Alguns adicionaram novos e úteis conceitos, mas no geral havia uma grande similaridade entre os conceitos propostos por diferentes autores. Muitos dos livros mais novos começaram de um ou mais dos métodos existentes acrescentando pequenas extensões ou melhorias

minoritárias. Os autores originais não estavam ociosos também; a maioria deles atualiza seu trabalho original, frequentemente incorporando as boas idéias de vários outros autores. Em geral, emergiu um *pool* de conceitos base comuns, juntamente com uma grande variedade de conceitos defendidos por um ou dois autores mas não usados comumente.

Esforço de Unificação

Houve alguns esforços iniciais na tentativa de se unificar conceitos entre os métodos. Um exemplo notável foi o do método Fusion de Coleman e seus colegas [10], o qual incluiu conceitos do método OMT [20], Booch [4] e CRC [30]. Como não envolveu os autores originais, deve ser considerado como um novo método ao invés de um substituto para vários outros existentes. A primeira tentativa bem sucedida de combinar e substituir abordagens existentes surgiu quando Rumbaugh juntou-se a Booch na Rational Software Corporation em 1994. Eles começaram por combinar conceitos dos métodos OMT e Booch, resultando em uma proposta inicial em 1995. à época, Jacobson também se juntou à Rational e começou a trabalhar com o Booch e Rumbaugh. Seu trabalho conjunto originou a Linguagem de Modelagem Unificada (UML). Houve bastante repercussão deste trabalho visto que tratava-se dos três melhores autores do momento. Em 1996, o Object Management Group (OMG) convidou pesquisadores a trazer propostas para uma abordagem padrão para a modelagem orientada a objeto. Eventualmente, todas as propostas convergiram e uma proposta final foi submetida em Setembro de 1997.

Padronização

A UML foi adotada unanimamente pela membresia da OMG como um padrão em Novembro de 1997 [15]. A OMG assumiu a responsabilidade pelo futuro desenvolvimento do padrão UML. Mesmo antes da adoção final, um número de livros foi publicado delineando os principais aspectos da UML. Muitas software houses fabricantes de ferramentas anunciaram o suporte ou a intenção de suportar a UML e muitos autores de metodologias anunciaram que usariam a notação da UML em trabalhos futuros. A aparição da UML parece ter sido bastante atrativa para o público em Computação em geral porque ela converge como foi dito o esforço de vários autores em um único reduzindo grandemente o trabalho.

3.2.3 Conceitos UML

Esta seção apresentará uma introdução aos principais conceitos e diagramas da UML. Não há uma linha rígida entre os vários conceitos e construções da UML mas, por conveniência, eles são divididos em várias visões. Uma visão é simplesmente um subconjunto de construções de modelagem da UML o qual representa um aspecto do sistema. A divisão em diferentes visões é algo de arbitrário, mas espera-se que seja intuitivo. Um ou dois tipos de diagrams provêm uma notação visual para os conceitos em cada visão. Em um nível mais elevado, as visões podem ser divididas em três áreas: classificação estrutural, comportamento dinâmico e gerenciamento de modelo. A classificação estrutural descreve os objetos no sistema e suas inter-relações. Os classificadores incluem classes, casos de uso, componentes e nodos. Os

classificadores provêm a base sobre a qual o comportamento dinâmico é construído. As visões de classificação incluem as visões estática, de caso de uso e de implementação. Definiremos as partes que compõem a UML:

- **Visões:** As Visões mostram diferentes aspectos do sistema que está sendo modelado. A visão não é um gráfico, mas uma abstração consistindo em uma série de diagramas. Definindo um número de visões, cada uma mostrará aspectos particulares do sistema, dando enfoque a ângulos e níveis de abstrações diferentes e uma Figura completa do sistema poderá ser construída. As visões também podem servir de ligação entre a linguagem de modelagem e o método/processo de desenvolvimento escolhido.
- **Modelos de Elementos:** Os conceitos usados nos diagramas são modelos de elementos que representam definições comuns da orientação a objetos como as classes, objetos, mensagem, relacionamentos entre classes incluindo associações, dependências e heranças.
- **Mecanismos Gerais:** Os mecanismos gerais provêm comentários suplementares, informações, ou semântica sobre os elementos que compõem os modelos; eles provêm também mecanismos de extensão para adaptar ou estender a UML para um método/processo, organização ou usuário específico.
- **Diagramas:** Os diagramas são os gráficos que descrevem o conteúdo em uma visão. UML possui nove tipos de diagramas que são usados em combinação para prover todas as visões do sistema.

Visões

O desenvolvimento de um sistema complexo não é uma tarefa fácil. O ideal seria que o sistema inteiro pudesse ser descrito em um único gráfico e que este representasse por completo as reais intenções do sistema sem ambiguidades, sendo facilmente interpretável. Infelizmente, isso é impossível. Um único gráfico é incapaz de capturar todas as informações necessárias para descrever um sistema. Um sistema é composto por diversos aspectos: funcional (que é sua estrutura estática e suas interações dinâmicas), não funcional (requisitos de tempo, confiabilidade, desenvolvimento, etc.) e aspectos organizacionais (organização do trabalho, mapeamento dos módulos de código, etc.). Então o sistema é descrito em um certo número de visões, cada uma representando uma projeção da descrição completa e mostrando aspectos particulares do sistema. Cada visão é descrita por um número de diagramas que contém informações que dão ênfase aos aspectos particulares do sistema. Existe em alguns casos uma certa sobreposição entre os diagramas o que significa que um destes pode fazer parte de mais de uma visão. Os diagramas que compõem as visões contêm os modelos de elementos do sistema. As visões que compõem um sistema estão expressas na Figura 3.3 e são:

- Visão de Caso de Uso. Descreve a funcionalidade do sistema desempenhada pelos atores externos do sistema (usuários). A visão de Caso de Uso é central, já que seu conteúdo é base do desenvolvimento das outras visões do sistema. Essa visão é montada sobre os diagramas de Caso de Uso e eventualmente diagramas de atividade.
- Visão Lógica. Descreve como a funcionalidade do sistema será implementada. É feita principalmente pelos analistas e desenvolvedores. Em

contraste com a visão de Caso de Uso, a visão lógica observa e estuda o sistema internamente. Ela descreve e especifica a estrutura estática do sistema (classes, objetos, e relacionamentos) e as colaborações dinâmicas quando os objetos enviarem mensagens uns para os outros para realizarem as funções do sistema. Propriedades como persistência e concorrência são definidas nesta fase, bem como as interfaces e as estruturas de classes. A estrutura estática é descrita pelos diagramas de classes e objetos. O modelamento dinâmico é descrito pelos diagramas de estado, sequencia, colaboração e atividade.

- Visão de Componentes. É uma descrição da implementação dos módulos e suas dependências. É principalmente executado por desenvolvedores, e consiste nos componentes dos diagramas.
- Visão de concorrência. Trata a divisão do sistema em processos e processadores. Este aspecto, que é uma propriedade não funcional do sistema, permite uma melhor utilização do ambiente onde o sistema se encontrará, se o mesmo possui execuções paralelas, e se existe dentro do sistema um gerenciamento de eventos assíncronos. Uma vez dividido o sistema em linhas de execução de processos concorrentes (*threads*), esta visão de concorrência deverá mostrar como se dá a comunicação e a concorrência destas threads. A visão de concorrência é suportada pelos diagramas dinâmicos, que são os diagramas de estado, sequencia, colaboração e atividade, e pelos diagramas de implementação, que são os diagramas de componente e execução.
- Visão de Organização. Finalmente, a visão de organização mostra a

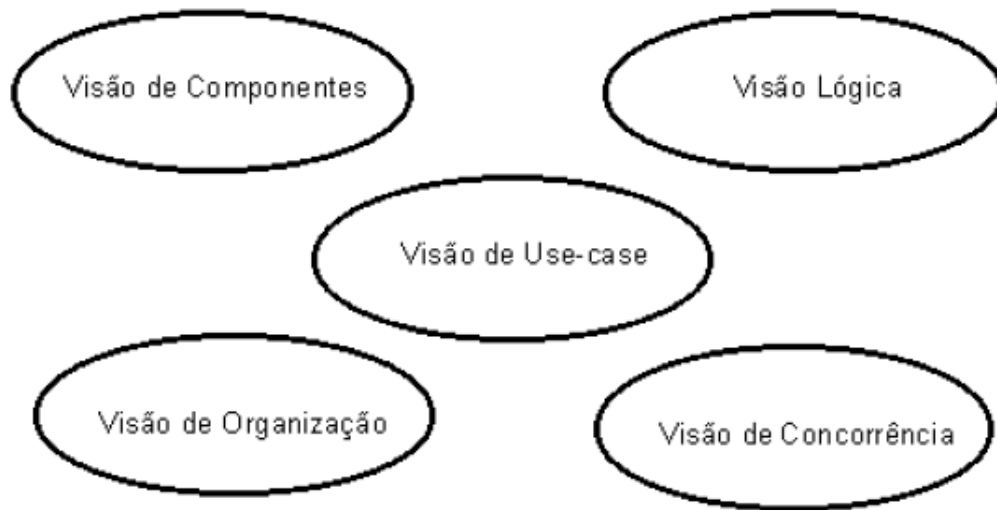


Figura 3.3: Visões da UML

organização física do sistema, os computadores, os periféricos e como eles se conectam entre si. Esta visão será executada pelos desenvolvedores, integradores e testadores, e será representada pelo diagrama de execução.

Diagramas

Os diagramas utilizados pela UML são compostos de nove tipos: diagrama de Caso de Uso, de classes, de objeto, de estado, de sequência, de colaboração, de atividade, de componente e o de execução. Todos os sistemas possuem uma estrutura estática e um comportamento dinâmico. A UML suporta modelos estáticos (estrutura estática), dinâmicos (comportamento dinâmico) e funcional. A Modelagem estática é suportada pelo diagrama de classes e de objetos, que consiste nas classes e seus relacionamentos. Os relacionamentos podem ser de associações, herança (generalização), dependência ou

refinamentos. Os modelamentos dinâmicos são suportados pelos diagramas de estado, sequência, colaboração e atividade. E o modelamento funcional é suportado pelos diagramas de componente e execução.

Diagrama de Sequência

Um diagrama de sequência mostra a colaboração dinâmica entre os vários objetos de um sistema. O mais importante aspecto deste diagrama é que, a partir dele, percebe-se a sequência de mensagens enviadas entre os objetos. Ele mostra a interação entre os objetos, alguma coisa que acontecerá em um ponto específico da execução do sistema. O diagrama de sequência consiste em um número de objetos mostrado em linhas verticais. O decorrer do tempo é visualizado observando-se o diagrama no sentido vertical de cima para baixo. As mensagens enviadas por cada objeto são simbolizadas por setas entre os objetos que se relacionam. Diagramas de sequência possuem dois eixos: o eixo vertical, que mostra o tempo e o eixo horizontal, que mostra os objetos envolvidos na sequência de uma certa atividade. Eles também mostram as interações para um cenário específico de uma certa atividade do sistema. No eixo horizontal estão os objetos envolvidos na sequência. Cada um é representado por um retângulo de objeto (similar ao diagrama de objetos) e uma linha vertical pontilhada chamada de linha de vida do objeto, indicando a execução do objeto durante a sequência, como exemplo citam-se as mensagens recebidas ou enviadas e ativação de objetos. A comunicação entre os objetos é representada como linha com setas horizontais simbolizando as mensagens entre as linhas de vida dos objetos. A seta especifica se

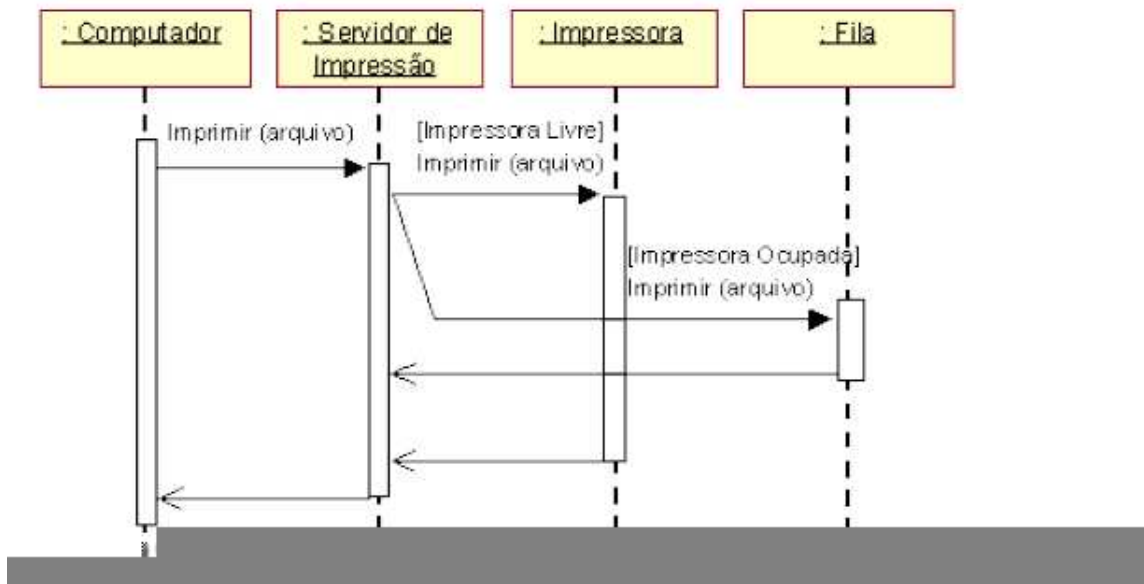


Figura 3.4: Um Exemplo de Diagrama de Sequência

a mensagem é síncrona, assíncrona ou simples. As mensagens podem possuir também números sequenciais, eles são utilizados para tornar mais explícito as sequência no diagrama. Em alguns sistemas, objetos rodam concorrentemente, cada um com sua linha de execução (*thread*). Se o sistema usa linhas concorrentes de controle, isto é mostrado como ativação, mensagens assíncronas, ou objetos assíncronos. A Figura 3.4 mostra um exemplo.

Os diagramas de sequência podem mostrar objetos que são criados ou destruídos como parte do cenário documentado pelo diagrama. Um objeto pode criar outros objetos através de mensagens. A mensagem que cria ou destroi um objeto é geralmente síncrona, representada por uma seta sólida.

Em síntese: o Diagrama de Seqüência é uma das ferramentas UML usadas para representar interações entre objetos de um cenário, realizadas através de operações ou métodos (procedimentos ou funções). Este diagrama é cons-

truído a partir do Diagrama de Casos de Usos. Primeiro, se define qual o papel do sistema (Casos de Uso). Depois, é definido como o software realizará seu papel (seqüência de operações).

O diagrama de sequência dá ênfase na ordenação temporal em que as mensagens são trocadas entre os objetos de um sistema. Entende-se por mensagens os serviços solicitados de um objeto a outro, e as respostas desenvolvidas para as solicitações.

3.3 Redes de Petri

A teoria inicial das Redes de Petri foi apresentada na tese de doutoramento *Kommunikation mit Automaten* do Dr. C. A. Petri em 1962 na Faculdade de Matemática e Física da Universidade de Darmstadt, na então Alemanha Ocidental. O trabalho de Petri atraiu a atenção de A. W. Holt que, em conjunto com outros pesquisadores, desenvolveu muito da teoria, notação e representação das redes de Petri. De 1970 a 1975, o grupo de estrutura da computação do MIT foi o mais ativo na condução da pesquisa sobre Redes de Petri. Em 1975, houve uma Conferência sobre Redes de Petri. No entanto, não houve publicação dos anais [12]. Em 1979, pesquisadores de vários países europeus reuniram-se em Hamburgo para um curso avançado sobre a *Teoria Geral das Redes de Processos e Sistemas* [5]. Seguiram-se diversos outros trabalhos propondo alterações ao modelo original, tais como redes com arcos inibidores e redes temporizadas determinísticas e estocásticas. Atualmente, Redes de Petri é considerada uma técnica para especificação de sistemas concorrentes consolidada. Grupos de pesquisa em todo o mundo têm Redes de Petri como tema, desenvolvendo estudos sobre seus aspectos teóricos e suas

aplicações. Apenas para constar, diversas técnicas de modelagem matemática de sistemas em muitas áreas da Ciência têm sido propostas. Barroca e McDermid [23] apresentam a seguinte classificação:

1. Técnicas Baseadas em Modelos. Fornecem descrições abstratas sobre estados e operações que transformam os estados. No entanto, não oferecem meios explícitos para especificar concorrência. Um exemplo é a linguagem Z [34].
2. Técnicas Baseadas em Álgebras de Processo. Tais técnicas fornecem meios explícitos para se especificar concorrência. O comportamento dos processos é representado através de comunicações observáveis. Exemplos incluem CCS [27], CSP [17] e LOTOS [18] [9].
3. Técnicas Baseadas em Lógica. Uma grande variedade de técnicas baseadas em Lógica têm sido propostas, onde se analisam as relações causais e aspectos relacionados à temporização. Um exemplo é a Lógica Modal de Ações [58].
4. Técnicas Baseadas em Redes. Essas técnicas modelam concorrência através de mecanismos implícitos de fluxo de *tokens* na rede. Este fluxo é controlado por condições que habilitam a realização de tarefas (eventos). O grande exemplo são as Redes de Petri.

3.3.1 Visão Geral

Uma rede de Petri ou rede de transição é uma das várias representações matemáticas para sistemas distribuídos discretos. Como uma linguagem de modelagem, ela define graficamente a estrutura de um sistema distribuído

como um grafo direcionado com comentários. Possui nós de posição, nós de transição e arcos direcionados conectando posições com transições. A notação básica de Redes de Petri é mostrada na Figura 3.5.

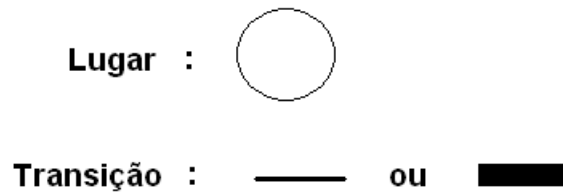


Figura 3.5: Elementos Básicos das Redes de Petri

Estes dois elementos são os vértices do grafo associado às rede de Petri. Os vértices são interligados por arcos dirigidos. Os arcos que interligam lugares às transições correspondem à relação entre as condições verdadeiras que, em um dado momento, possibilitam a execução das ações, enquanto os arcos que interligam transições aos lugares representam a relação entre as ações e as condições que se tornam verdadeiras com a execução das ações. A qualquer momento durante a execução de uma rede de Petri, cada lugar pode armazenar um ou mais *tokens*. Diferente de sistemas mais tradicionais de processamento de dados, que podem processar somente um único fluxo de *tokens* entrantes, as transições de redes de Petri podem consumir e mostrar *tokens* de múltiplos lugares. Uma transição só pode agir nos tokens se o número requisitado de *tokens* aparecer em cada posição de entrada. A execução de uma rede de Petri é não-determinística. Isso significa que múltiplas transições podem ser habilitadas ao mesmo tempo (cada uma pode ser disparada) e que nenhuma transição deve ser obrigatoriamente executada em determinado momento. Transições agem em *tokens* de entrada por um pro-

cesso denominado disparo. Quando uma transição é disparada, ela consome os *tokens* de seus lugares de entrada, realiza alguma tarefa de processamento, e realoca um número específico de *tokens* nos seus lugares de saída. Isso é feito atomicamente. Como disparos são não determinísticos, redes de Petri são muito utilizadas para modelar comportamento concorrente em sistemas distribuídos. Com essas noções em mãos, segue a definição formal de Redes de Petri.

3.3.2 Definição Formal

Uma Rede de Petri (S, T, F, M_0, W, K) é dada por:

- S , um conjunto de lugares.
- T , um conjunto de transições.
- F , um conjunto de arcos também chamados de relações de fluxo. Ele é sujeito à restrição de que nenhum arco conecta dois lugares ou transições ou, mais formalmente, $F \subseteq (S \times T) \cup (T \times S)$.
- $M_0 : S \rightarrow \mathbb{N}$, conhecido como marco inicial, no qual para cada lugar $s \in S$, existem $n \in \mathbb{N}$ tokens.
- $W : F \rightarrow \mathbb{N}^+$, conhecido como um conjunto de pesos de arco, relaciona para cada arco $f \in F$ um $n \in \mathbb{N}^+$ denotando quantos *tokens* são consumidos de um lugar por uma transição ou, alternativamente, quantos tokens são produzidos por uma transição e colocados em cada posição.
- $K : S \rightarrow \mathbb{N}^+$, conhecido como restrições de capacidade, relaciona para cada posição $s \in S$ um número positivo $n \in \mathbb{N}^+$ denotando o número

máximo de *tokens* que podem ocupar aquele lugar.

3.3.3 Propriedades Matemáticas

O estado de uma rede de Petri é representado por um vetor M , no qual o primeiro valor do vetor é a quantidade de *tokens* na primeira posição da rede, o segundo é a quantidade de *tokens* na segunda posição, e assim por diante. Tal representação descreve completamente o estado de uma rede de Petri. Uma lista de transição de estados $\sigma = \langle M_{i_0} t_{i_1} M_{i_1} \dots t_{i_n} M_{i_n} \rangle$, que pode ser simplificada em $\vec{\sigma} = \langle t_{i_1} \dots t_{i_n} \rangle$ é chamada uma sequência de disparo se cada transição satisfaz o critério de disparo, isto é, se existem *tokens* suficientes na entrada de cada transição. Nesse caso, a lista de transição de estados de $\langle M_{i_0} M_{i_1} \dots M_{i_n} \rangle$ é chamada *trajetória*, e M_{i_n} é chamado *alcançável* a partir de M_{i_0} através da sequência de disparos de $\vec{\sigma}$. Matematicamente: $M_{i_0} [\vec{\sigma}] > M_{i_n}$. Todas as sequências de disparo que pode ser atingidas em uma rede N com estado inicial M_0 são denotadas como $L(N, M_0)$. A matriz de transição de estados W^- é $|T|$ por $|S|$ e representa a quantidade de *tokens* passados por cada transição de cada posição. Similarmente, W^+ representa a quantidade de *tokens* dados por cada transição para cada posição. A soma dos dois, $W^- + W^+$, pode ser utilizada para calcular a equação mencionada anteriormente de $M_{i_0} [\vec{\sigma}] > M_{i_n}$. Ela agora pode ser escrita simplesmente como $M_0 - M_n = W^T \bullet \sigma$, no qual σ é um vetor de quantas vezes cada transição foi disparada na sequência. Vale notar que somente porque a equação pode ser satisfeita não significa que ela pode utilizada; para isso devem haver *tokens* suficientes para cada transição ser disparada, isto é, somente com a equação não é suficiente dizer que o estado M_n pode ser atingido a partir

do estado M_0 . Todos os estados que podem ser atingidos em uma rede N com estado inicial M_0 são denotados como $R(N, M_0)$. Surge o problema de alcançabilidade: é verdadeiro que $M_w \in R(N, M_0)$, no qual M_w é, por exemplo, um estado inválido tal qual um elevador se movendo enquanto a porta está aberta? A alcançabilidade dos estados pode ser representada pelo grafo de alcançabilidade no qual os destinos de um grafo direcionado representam estados (por exemplo M), e transições de arcos entre dois dos tais estados. O grafo é contruído como a seguir: o estado inicial M_0 é definido e todos as possibilidades de transição são exploradas a partir desse estado; depois a partir os estados resultantes da primeira iteração, e assim por diante. Enquanto a alcançabilidade parece ser uma boa ferramenta para encontrar estados errôneos, o grafo construído possui estados demais para problemas práticos. Por essas razões, a lógica linear temporal com o método de tableau é geralmente utilizado para provar que tais estados não podem ser alcançados. Essa lógica usa a técnica de semi-decisão para encontrar se realmente um estado pode ser alcançado, ao procurar um conjunto de condições necessárias para o estado ser alcançado e provando que tais condições não podem ser satisfeitas.

Atividade

Uma transição t de uma rede de Petri (N, M_0) está

- L_0 viva, ou morta, *sse* ela não pode ser disparada, isto é, não está em nenhum $\vec{\sigma} \in L(N, M_0)$.
- L_1 viva *sse* ela pode ser disparada, isto é, está em uma seqüência de disparo $\vec{\sigma}$ na qual $\vec{\sigma} \in L(N, M_0)$.

- L_2 viva *sse* para qualquer número k inteiro positivo, t pode ser disparado pelo menos k vezes em uma sequência de disparo $\vec{\sigma}$ na qual $\vec{\sigma} \in L(N, M_0)$.
- L_3 viva *sse* existe uma sequência de disparo $\vec{\sigma} \in L(N, M_0)$ na qual t é disparada infinitamente.
- L_4 viva, ou viva, *sse* em qualquer estado alcançável $m(\forall M \in R(N, M_0))$, t é L_1 viva.

Uma transição L_3 viva é automaticamente L_0 , L_1 e L_2 da mesma maneira. Uma rede de Petri é L_k viva *sse* toda transição pertencente é L_k viva.

3.3.4 Limitações

Existem redes de Petri nas quais as posições são limitadas em um número máximo de *tokens*. Nesse caso, a limitação é uma propriedade inerente. Apesar disso, redes de Petri podem ser definidas sem limitações com uma propriedade estrutural. Uma rede de Petri limitada não estruturalmente é k -limitada se não existe estado alcançável no qual alguma posição contém mais que k *tokens*. Uma rede de Petri é segura se ela é 1-limitada. Limitações de certas posições em uma rede inerentemente limitada podem ser feitas em uma rede inerentemente não-limitada ao utilizar uma transformada de posição, no qual uma nova posição (chamada contra-posição) é criada, e todas as transições que levam x *tokens* na posição original levam x *tokens* para a contra-posição. O número de *tokens* em M_0 agora deve satisfazer à equação *place + counter - place = boundedness*. Assim, aplicar uma transformada de posição para todas as posições em uma rede limitada, e

restringindo o estado inicial M_0 para satisfazer a equação acima, uma rede limitada pode facilmente ser transformada em uma rede não-limitada. Desta maneira qualquer análise usada em uma rede não-limitada pode ser utilizada em redes limitadas. A recíproca não é verdadeira.

3.3.5 Extensões

Existem várias extensões para redes de Petri. Algumas delas são completamente compatíveis com o modelo tradicional (por exemplo, redes de Petri coloridas), algumas adicionam propriedades que não podem ser definidas no modelo tradicional (por exemplo redes de Petri temporizadas). Se elas pudessem ser definidas no modelo tradicional, não seriam realmente extensões, e sim maneiras mais convenientes de demonstrar a mesma coisa, de forma que poderiam ser transformadas com fórmulas matemáticas para o modelo tradicional, sem perda de informação. Extensões que não podem ser transformadas para o modelo tradicional são muitas vezes poderosas, mas geralmente não possuem ferramentas matemáticas suficientes para análise como no modelo tradicional de redes de Petri. O termo redes de Petri de alto-nível é utilizado por vários formalismos de redes de Petri que estendem o formalismo posição/transição. Isso inclui redes de Petri coloridas, redes de Petri hierárquicas e outras extensões como segue:

- Em uma rede de Petri tradicional, *tokens* são indistinguíveis. Em uma rede de Petri colorida, cada token possui um valor. Em várias ferramentas para redes de Petri coloridas, os valores dos *tokens* são tipados, e podem ser testados e manipulados usando-se uma linguagem de programação funcional. Uma derivação de redes de Petri coloridas são as

redes de Petri bem formadas, nas quais os arcos e expressões de guarda são restritas para tornar a rede mais fácil de ser analisada.

- Outra extensão popular para redes de Petri é a hierarquia: hierarquia na forma de diferentes visões suportando níveis de refinamento e abstração, assim como estudado por Fehling. Outra forma de hierarquia é encontrada nas chamadas redes de Petri Objeto ou Sistemas Objeto, no qual uma rede de Petri pode conter outra rede de Petri como token, introduzindo o conceito de redes de Petri aninhadas que se comunicam ao sincronizar transições entre diferentes níveis.
- Um Sistema de Adição de Vetor com Estados (*Vector Addition System with States* - VASS) pode ser visto como uma generalização de uma rede de Petri. Considerado um autômato finito no qual cada transição é denominada por uma transição da rede. A rede de Petri é então sincronizada com o autômato finito, isto é, uma transição no autômato é feita no mesmo momento que a mesma transição na rede de Petri. É somente possível utilizar uma transição no autômato se a transição correspondente na rede de Petri está habilitada, e é somente possível disparar uma transição na rede de Petri se existe uma transição no mesmo estado no autômato.
- Redes de Petri Priorizadas adicionam prioridades para as transições, de forma que uma transição não pode ser disparada se uma transição de prioridade maior está habilitada. Desta forma, transições estão em grupos de prioridade e, por exemplo, um grupo de prioridade 3 só pode disparar se todas as transições estão desabilitadas nos grupos 1

e 2. Mesmo em um grupo de prioridade, disparos ainda assim são não-determinísticos.

- A propriedade não determinística é bem útil pois permite ao usuário abstrair um grande número de propriedades (dependendo da finalidade da rede). Em certos casos, entretanto, existe também a necessidade de modelar também temporizações. Para esses casos são utilizadas redes de Petri temporizadas, no qual existem transições que são temporizadas, e possivelmente transições não temporizadas (nesse caso, transições não temporizadas são prioritárias em relação as temporizadas). Desta forma, a propriedade de tempo também pode ser modelada além da de estrutura. Uma derivação de redes de Petri temporizadas são as redes de Petri estocásticas, que adicionam tempos não-determinísticos através de aleatoriedade ajustável nas transições. A distribuição exponencial é utilizada para cronometrar essas redes. Nesse caso, o grafo de alcançabilidade dessas redes pode ser usado como uma Cadeia de Markov.

Existem várias outras extensões para redes de Petri. Entretanto, é importante saber que, ao se adicionar complexidade nas redes devido à inclusão de propriedades, mais difícil fica utilizarem-se ferramentas tradicionais para calcular certas propriedades da rede. Por essa razão, é sempre uma boa idéia utilizar-se a rede mais simples possível para uma tarefa dada.

3.4 Model Checking

3.4.1 Introdução aos Métodos Formais

Métodos Formais referem-se a técnicas e ferramentas matematicamente rigorosas para a especificação, projeto e verificação de sistemas de software e hardware. Frequentemente, o conjunto de atividades num projeto de sistema embarcado não é especificado de uma maneira rigorosa e sem ambiguidade. Desta forma, o processo leva várias iterações para chegar ao resultado final. Por isso, o uso de um ou mais métodos formais é altamente recomendável para projetar-se sistemas embarcados.

3.4.2 Definição

A Verificação de Modelos ou *Model Checking* é a abordagem mais bem sucedida que emergiu no que diz respeito à verificação de modelos. A idéia essencial atrás de uma ferramenta de model checking é receber os requisitos do sistema ou seu projeto (modelos) e uma propriedade (chamada especificação) que o sistema resultante deve satisfazer. A ferramenta então gera como saída “SIM” se o dado modelo satisfaz as dadas especificações e gera um contra-exemplo caso contrário. Este contra-exemplo detalha porque o modelo não satisfaz a especificação. Estudando-se o contra-exemplo, o usuário da ferramenta pode identificar o erro no modelo, corrigí-lo e verificá-lo então novamente. A idéia é que, assegurando-se que o modelo satisfaça suficientemente as propriedades do sistema, pode-se aumentar o grau de confiabilidade da correção do modelo. Os requisitos do sistema são chamados modelos porque eles representam os requisitos ou projeto.

As ferramentas de Model Checking enfrentam um problema de possível explosão combinatorial de estados que deve ser abordada a fim de se resolver muitos problemas reais. Há muitas abordagens pra se combater tal problema, tal qual o uso de diagramas de decisão binários (BDDs), introduzidos por McMillan [25] e utilizados na ferramenta de model checking do presente trabalho, o SMV.

3.4.3 Os Benefícios do Model Checking

A seguir, um resumo dos principais benefícios da técnica de Model Checking, de acordo com Katoen [22]:

- Abordagem geral com aplicações para verificação de hardware, engenharia de software, sistemas multi-agentes, protocolos de comunicação, sistemas embarcado e dai por diante.
- Suporta verificação parcial: um projeto pode ser verificado contra uma especificação parcial, considerando-se apenas um subconjunto de todos os requisitos. Isso pode resultar em eficiência melhorada, desde que o usuário desta técnica possa restringir a validação para verificar apenas os mais importantes requisitos ignorando os menos importantes.
- Estudos de caso têm mostrado que a incorporação de model checking no processo de projeto não o atrasa mais do que as técnicas de Simulação e de Testes o fazem. Em muitos estudos, o uso da técnica de model checking levou a um tempo menor de desenvolvimento até. Ademais, graças a várias técnicas modernas, os verificadores de modelo ou *model checkers* são capazes de lidar com espaços de estados maiores (um

exemplo com 10^{130} estados foi reportado na literatura).

- Os *model checkers* podem potencialmente ser usados tão rotineiramente quanto os compiladores pelos projetistas visto que também não exigem tanta interação com o usuário.
- Crescente interesse por parte da Indústria: empregos são oferecidos para aqueles que têm proficiência no uso das técnicas de model checking. Além do mais, a Indústria tem construído seus próprios *model checkers*. Exemplos incluem os da Siemens e Lucent Technologies. Outras companhias que criaram grupos de pesquisa nesta área ou tem usado largamente as ferramentas incluem Intel, Cadence, Fujitsu e Dutch Railways para mencionar algumas.
- Forte fundamento matemático: modelagem, semântica, Teoria de Concorrência, Teoria de Autômatos, Estrutura de Dados, Algoritmos de Grafos etc constituem a base de model checking.

3.4.4 Limitações do Model Checking

A seguir, um resumo das principais limitações da técnica de Model Checking, de acordo com Katoen [22]:

- É menos adequado para aplicações com uso intensivo de dados, as chamadas *data-intensive applications*. Isto porque o tratamento de dados usualmente introduz espaços de estados infinitos.
- A aplicação do model checking está sujeita a questões de decidibilidade: para alguns casos particulares - como a maior parte dos sistemas de

estados infinitos - o model checking não é eficientemente computável. Verificação Formal entretanto é, em princípio, aplicado a estes sistemas.

- Usando-se model checking, um modelo do sistema é verificado ao invés do sistema real em si. O fato de um modelo ter certas propriedades não implica que a realização do mesmo também as terá. Por esse propósito, técnicas complementares tais como testes sistemáticos são necessárias.
- Apenas os requisitos declarados são verificados: não há garantias da completude das propriedades desejadas.
- Encontrar uma abstração apropriada (tal como o modelo do sistema e propriedades apropriadas em lógica temporal) requer alguma experiência.
- Como qualquer ferramenta, o software de model checking pode não ser confiável. Entretanto neste caso o impacto não é tão grande visto que as ferramentas de model checking se baseiam em algoritmos padrão e bem conhecidos.
- É possível (em geral) verificar-se generalizações com model checking. Se, por exemplo, um protocolo, após verificado, seja considerado correto para 1, 2 ou 3 processos usando-se model checking, não é possível prover uma resposta para n processos, sendo n arbitrário. Apenas para casos particulares, isto é factível. Model checking pode, entretanto, sugerir teoremas para parâmetros arbitrários que, subsequentemente, podem ser verificados usando-se alguma técnica de verificação formal.

Ainda segundo Katoen, não é possível correte absoluta com a tecnica de model checking para sistemas de tamanho realístico. Porém, apesar de

tais limitações, o model checking pode prover um aumento significativo no nível de confiança no sistema.

Visto que no model checking é muito comum modelar-se o possível comportamento do sistema como autômato de estado finito, a técnica é inerentemente vulnerável ao problema prático de que o número de estados pode exceder o total de memória disponível do computador. Isto é em particular um problema para sistemas distribuídos e paralelos que podem ter muitos estados possíveis - o tamanho do espaço de estados de tais sistemas é, no pior caso, proporcional ao produto do tamanho do espaço de estados de seus componentes individuais. Este problema é conhecido como “Problema da Explosão do Espaço de Estados” A Subseção 3.4.5 detalha melhor tal problema.

3.4.5 Problema da Explosão do Espaço de Estados

As ferramentas de Model Checking enfrentam um problema de explosão combinatorial do espaço de estados, comumente conhecida como problema da explosão de estados. Tal problema deve ser resolvido a fim de se tornar viáveis vários problemas reais. Existem várias abordagens para se combater este problema. A seguir, um resumo das mesmas.

Algoritmos simbólicos não constroem o grafo de alcançabilidade para o autômato; ao invés disso, eles representam tal grafo implicitamente usando-se uma fórmula em lógica propocional. O uso de Diagramas de Decisão Binários (*Binary Decision Diagrams* - BDDs) tornou-se popular pelo trabalho de Ken McMillan. Algoritmos de Verificação de Modelos Limitados (*Bounded Model Checking*) descrevem o autômato em termos de um número fixo de passos k e verificam se a violação de propriedade pode ocorrer em k ou menos passos.

Isto tipicamente envolve codificar o modelo restrito como uma instância do problema SAT. O processo pode ser repetido com valores cada vez maiores de k até que todas as possíveis violações tenham sido corrigidas.

Outra técnica é a de Redução de Ordem Parcial, o qual pode ser usada (em grafos explicitamente representados) para se reduzir o número de intervalos independentes de processos concorrentes que necessitam ser considerados. A idéia básica é que se não importa, para o conjunto de coisas que se pretende provar, se A ou B executado primeiro, então é perda de tempo considerar-se tanto intervalos de AB e BA .

Uma terceira técnica é a abstração. Ela tenta provar propriedades num sistema por meio de uma simplificação prévia. O sistema simplificado normalmente não satisfaz exatamente as mesmas propriedades que o original de tal forma que um processo de refinamento faz-se necessário. Geralmente, alguém requer que a abstração seja válida (as propriedades provadas para o sistema simplificado sejam também para o original); entretanto, mais frequentemente, a abstração não é completa (nem todas as propriedades verdadeiras do sistema original o são também na abstração). Um exemplo de abstração é, num programa, ignorar-se os valores de variáveis não booleanas e considerar-se apenas variáveis booleanas e o fluxo de controle do programa; tal abstração, apesar de parecer vaga, pode de fato ser suficiente para provar por exemplo propriedades de exclusão mútua.

3.4.6 Verificação através de CTL

A Lógica da Árvore de Computação ou *Computation Tree Logic* (CTL) é um tipo de Lógica Temporal de Ramificações introduzida por Clarke e Emerson

e usada em uma maneira um pouco modificada por Quielle e Sifakis para fins de model checking. Três operadores temporais são usados, a saber:

- E - pronuncia-se "para algum caminho"
- A - pronuncia-se "para todos os caminhos"e
- U - pronuncia-se "até"

Os operadores de caminho são:

- X - na próxima oportunidade
- F - eventualmente
- G - continuamente

Considere o modelo básico de computação das estruturas de Kripke, qual seja $K = (S, I, \delta, AP, L)$, onde:

S	Estados do sistema
$I \subseteq S$	Estados iniciais
$\delta \subseteq S \times S$	Relações de transição
AP	Proposições atômicas sobre os estados
$L : S \rightarrow 2^{AP}$ (rótulos)	Função de rotulamento

A Tabela 3.1 apresenta um resumo das estruturas de ramificação e modos temporais da CTL.

Dessa forma, a validade do sistema é dada por:

$$K \models \varphi \text{ sse } K, S \models \varphi \text{ para todo } s \in I, \text{ sse } I \subset \|\varphi\| K$$

tipo	fórmula φ	$K, S_0 \models \varphi$ sse ...
atômica	$p \in AP$	p é verdade de S_0
proposicional	$\neg\varphi$	$K, S_0 \not\models \varphi$
	$\varphi \vee \psi$	$K, S_0 \models \varphi$ ou $K, S_0 \models \psi$
temporal	$EX\varphi$	Existe um caminho $S_0S_1 \dots$ s.t. $K, S_1 \models \varphi$
	$AF\varphi$	Para todos os caminhos $S_0S_1 \dots$ existe $i \in \mathbb{N}$ s.t. $K, S_i \models \varphi$
	$\varphi EU\psi$	Existe um caminho $S_0S_1 \dots$ e $i \in \mathbb{N}$ s.t. $K, S_i \models \psi$ e $K, S_j \models \varphi$ para todo $0 \leq j < i$
	$AX\varphi, EF\varphi, \dots$	similar

Tabela 3.1: Estruturas de Ramificação e Modos Temporais

3.4.7 SMV

O SMV é uma ferramenta para verificar sistemas de estados finitos contra especificações em lógica temporal CTL. A linguagem de entrada do SMV foi projetada para permitir uma descrição de sistemas de estados finitos que variam dos totalmente síncronos aos completamente assíncronos, e dos detalhados aos abstratos. Alguém poderia prontamente especificar um sistema como uma máquina de Mealy síncrona ou como uma rede de processos abstratos e não determinísticos. A linguagem provê a possibilidade de descrições modulares hierárquicas e a definição de componentes reusáveis. Partindo do princípio de que a SMV é direcionado para se descrever máquinas de estado finitas, os únicos tipos de dados na linguagem são os finitos, quais sejam: Booleanos, escalares e arranjos fixos. Estruturas de dados estáticas também podem ser construídas.

O propósito primário da linguagem de programação do SMV é descrever a relação de transição de uma estrutura de Kripke finita. Qualquer expressão em cálculo proposicional pode ser usada para determinar esta relação. Isto provê uma grande quantidade de flexibilidade e, ao mesmo tempo, um certo perigo de inconsistência. Por exemplo, a presença de contradição lógica pode resultar em um *deadlock*. Isto poderia tornar certas especificações verdadeiros sem razão ao mesmo tempo que tornar a descrição não implementável. Enquanto o processo de model checking pode ser usado para verificar-se *deadlocks*, é melhor evitar-se o problema quando possível por meio do uso de uma descrição de estilo mais restrito. O SMV suporta isso através de sua sintaxe de atribuição paralela.

O próximo capítulo discorre a respeito do método desenvolvido no pre-

sente trabalho para capturar e verificar formalmente requisitos de softwares embarcados.

Capítulo 4

Método Proposto

4.1 Contexto e Visão Geral

O contexto do presente trabalho é o de software embarcado. Este está sujeito a várias restrições, como descrito na Seção 3.1, tais como possibilidade de ser distribuído ou não, de tempo-real, peso, consumo de energia etc. Muitos deles, em caso de falhas, podem gerar sérios danos ou prejuízos financeiros. Há, por isso, a necessidade de métodos rigorosos para se assegurar a corretude dos projetos de software embarcado. Há também, por outro lado, a necessidade de métodos mais ágeis e intuitivos para acelerar o desenvolvimento.

O método proposto no presente trabalho visa suprir tais lacunas. Do ponto de vista da agilidade do processo como um todo, o método utiliza-se da Linguagem de Modelagem Unificada (Unified Modeling Language - UML), em particular de seu Diagrama de Sequência (ver 3.2.3) para modelar o software capturando seus aspectos importantes. A grande vantagem da UML é a de ser semi-formal e orientada a objetos facilitando o uso. A segunda vantagem, consequência da primeira, é a de ser um padrão da Indústria.

Há ainda uma terceira vantagem que é o fato de ser uma linguagem de propósito geral, não sendo restrita a um domínio ou outro (ver 3.2.1). Isso traz a possibilidade de a mesma ser usada também no contexto de software embarcado.

Um importante aspecto na modelagem de sistema está associado com a escolha do formalismo adequado para produzir um modelo do sistema. De fato, não há “formalismo correto”, de uma perspectiva geral. De acordo com as necessidades de modelagem dos requisitos é que um formalismo específico deve ser escolhido baseando-se nas características do sistema e na experiência do projetista [24]. A massificação do uso da UML sugere que a mesma é de conhecimento da grande maioria dos projetistas de software embarcado, não sendo um problema portanto seu uso pelo presente método.

Do ponto de vista da garantia da corretude, o método proposto usa Redes de Petri, um modelo matemático preciso, conforme descrito na Seção 3.3.5, para realizar verificações formais. A idéia é traduzir o modelo em alto nível, na notação da UML, para um modelo mais rigoroso, na notação Redes de Petri. Depois, realizar consultas de propriedades com CTL. O método é *qualitativo* pois considera apenas os operadores tradicionais da CTL e não informações de tempo-real, o que o tornariam *quantitativo*.

4.2 Descrição Detalhada

O presente trabalho considera o modelo executivo cíclico para software embarcado. Ou seja, as unidade mínimas de computação são as tarefas físicas do software a ser analisado. Elas são construídas previamente, antes de serem usadas no método. E estes pedaços mínimos de software são os objetos

que compõe o diagrama de sequência da modelagem. Com isso, captura-se o aspecto ou visão dinâmica, conforme descrito na Seção 3.2.3. As tarefas são agrupadas em processos. Dessa forma, o sistema final não leva em consideração o uso de sistemas operacionais embarcados (de tempo real ou não), mas o modelo de escalonamento baseado no despachante. A seguir, a descrição passo-a-passo do método qualitativo proposto. Todas etapas descritas são apoiadas por um ambiente de software, o Ambiente de Verificação Formal de Software Embarcado, descrito na Seção 4.7.

O primeiro passo no método é a especificação das tarefas de aplicação, previamente construídas, em diagramas de sequência da UML, se o projetista ainda não tiver realizado esta etapa em fases anteriores do projeto do software embarcado. Esta especificação é apresentada no formato XML e seu formato é apresentado na Tabela 4.1. Tal especificação pode também ser gerada por uma ferramenta visual de Engenharia de Software Auxiliada por Computador (*Computer-Aided Software Engineering - CASE*). Após tem-se a tradução do formalismo de alto nível, no caso, o Diagrama de Sequência, para outro mais formal, as Redes de Petri Lugar/Transição. Para tanto, o método atual baseia-se em parte no método proposto por Jeng [21] de conversão de Diagramas de Sequência para Redes de Petri. Este método é descrito na Seção 4.3. De posse da rede de Petri traduzida pode-se gerar um arquivo no formato Abstract Petri Net Notation - APNN, que pode ser usado como entrada para o *model checker* Model Checking Kit (MCK) ou Petri Net Markup Language - PNML. Na geração do arquivo SMV, fazem-se necessárias informações a respeito da infra-estrutura do sistema. Tais informações dizem respeito aos processadores e seus processos. Também há informações sobre

os processos e suas tarefas. Seu formato é apresentado na Tabela 4.2.

Não faria sentido permitir a especificação do software em um nível de abstração mais alto e solicitar ao projetista / usuário da ferramenta um conhecimento mais aprofundado do formalismo da Computational Tree Logic - CTL. Levando-se isso em consideração, tem-se a última etapa: Através de uma interface gráfica especial, o Ambiente de Verificação Formal de Software Embarcado permite ao projetista especificar, em linguagem natural, as consultas de propriedades a serem realizadas sobre o software embarcado em questão. A fórmula é então traduzida automaticamente para o formato CTL, adequado a ser usado em algum verificador de modelos (model checker) tal como o SMV ou o MCK. O SMV é melhor comentado na Seção 3.4.7. A Figura 4.1 resume o exposto acerca do método proposto. A Seção 4.3 detalha o método de Jeng.

Outra característica do método atual é que ele pode considerar o sistema distribuído entre vários processadores e as tarefas se comunicando via um barramento compartilhado. No experimento citado no Capítulo sobre experimentos há quatro processadores em questão (ver 5).

É válido citar também que a idéia de se modelar software com Diagramas de Sequência com tradução para Redes de Petri e posterior verificação formal através de técnicas de Verificação de Modelos (Model Checking) não se aplica somente ao software que funciona no modelo de escalonamento baseado em despachante, mas pode ser aplicado virtualmente a todo e qualquer projeto de software, inclusive não embarcado.

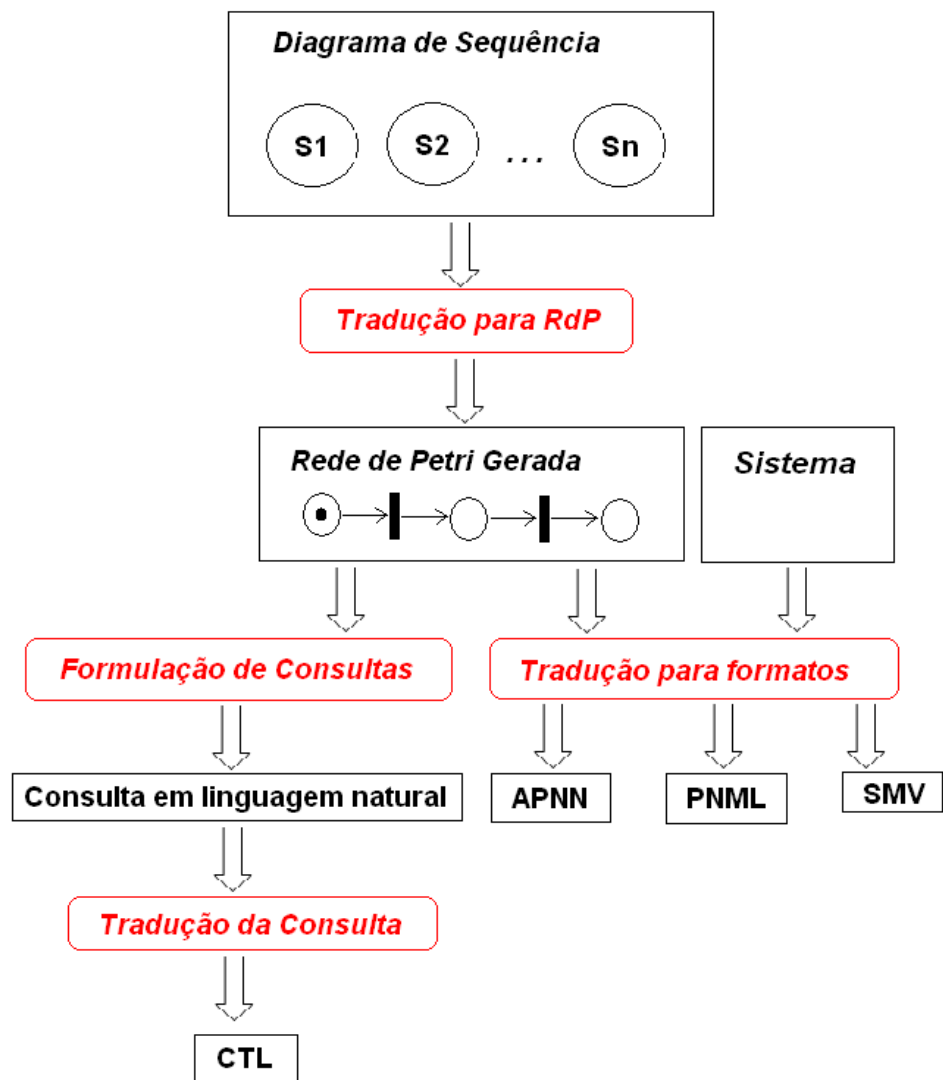


Figura 4.1: Método Proposto

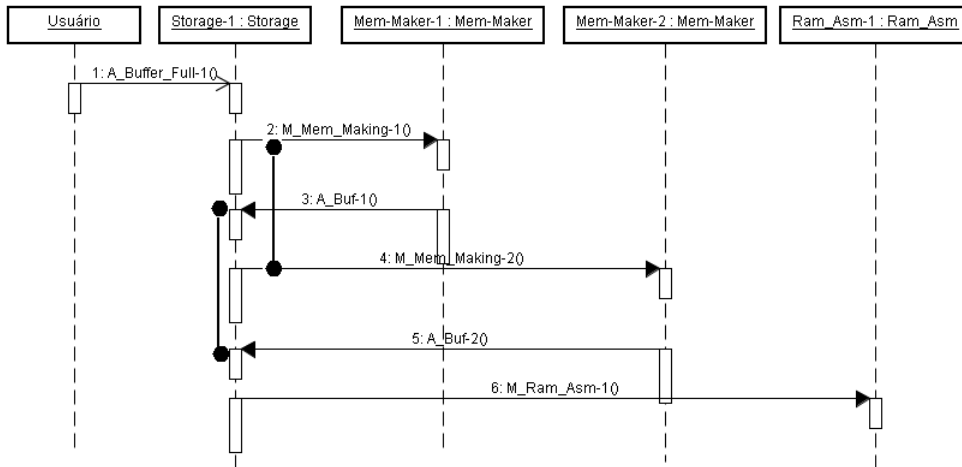


Figura 4.2: Notação Diagrama de Sequência Estendida

4.3 Notação Diagrama de Sequência Estendida

A modelagem do sistema é realizada com o diagrama de sequência, aumentando das extensões introduzidas no trabalho de Jeng [21]. A fim de melhor descrever o comportamento do sistema, Jeng estendeu o diagrama de sequência com o poder da concorrência, escolha, sincronismo e confluência. A Figura 4.2 mostra Notação Diagrama de Sequência Estendida

4.4 Regras de Tradução

O processo de tradução introduzido no trabalho de Jeng [21] consiste em oito regras. As próximas Subseções apresentam um resumo das mesmas.

4.4.1 Regra 1. Conversão de envio de mensagem

O envio de mensagem é transformado em um lugar como mostra a Figura 4.3, onde *msg* é um atributo de *Object2*. O número dentro do colchete seguindo

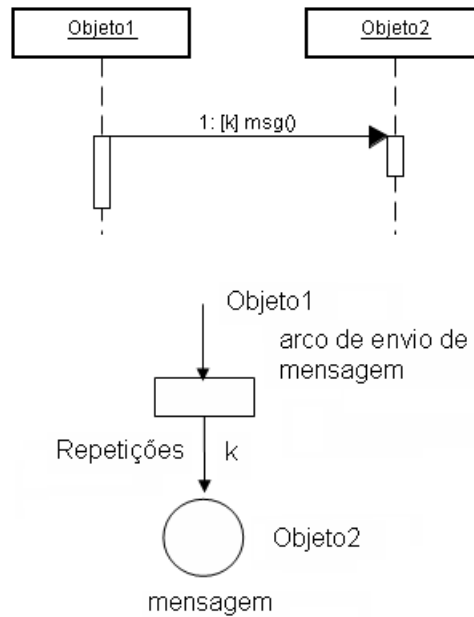


Figura 4.3: (Regra 1) Conversão de envio de mensagem.

o asterisco denota a marca de iteração e é convertido em um peso de arco.

4.4.2 Regra 2. Conversão de uma chamada de método

Uma chamada de método é transformada em um modelo básico como mostram as Figuras 4.4 e 4.5, onde o usuário pode definir um recurso a ser usado.

4.4.3 Regra 3. Conversão de uma operação de retorno de método

Uma operação de retorno de método é convertida em um lugar, quando a execução de um método é completada e o objeto então retorna um valor para alguma variável. Isto corresponde a um lugar que ganha um token,

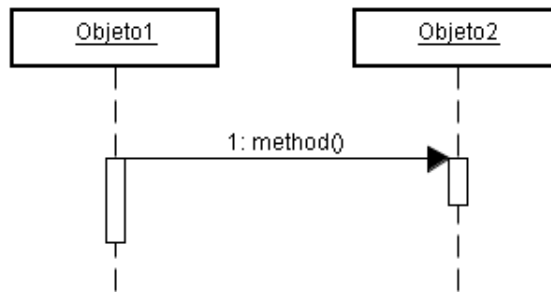


Figura 4.4: (Regra 2) Conversão de uma chamada de método

como mostra o lugar *rtn* da Figura 4.6.

4.4.4 Regra 4. Conversão de uma condição

Cada condição é transformada em um lugar que contém um dos tokens necessários para o disparo de uma transição, como ilustrado no lugar *c* da Figura 5.15.

4.4.5 Regra 5. Conversão de concorrência

Como mostrado na Figura 4.8, o intervalo rotulado por uma linha sólida com extremidades sólidas também é chamado de grupo de concorrência.

4.4.6 Regra 6. Conversão de Escolha

A Figura 4.9 mostra o grupo de escolha, o qual é denotado por um intervalo rotulado com uma linha pontilhada cujas extremidades são vazias. Somente uma mensagem neste grupo pode ser disparada por vez.

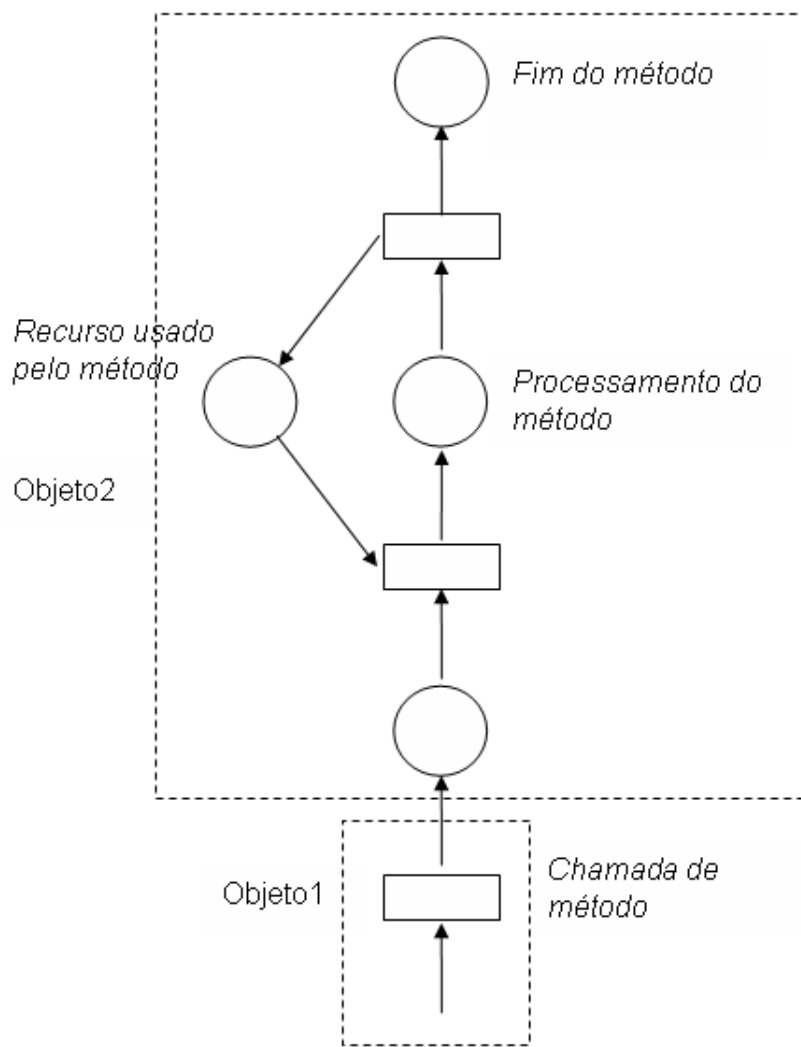


Figura 4.5: Rede Gerada da Regra 02

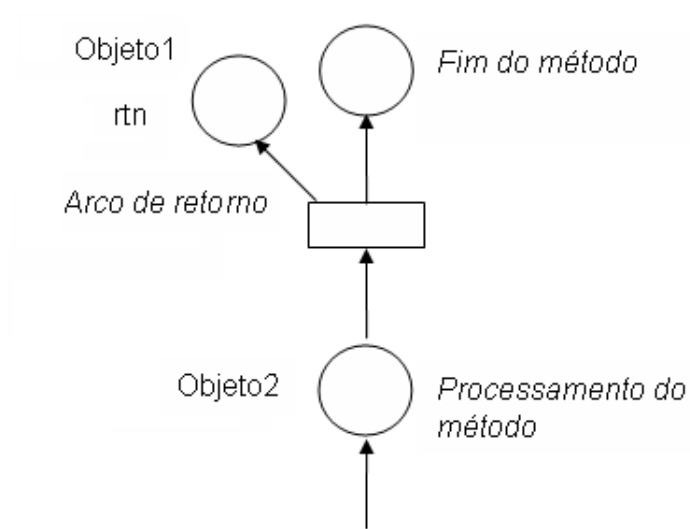
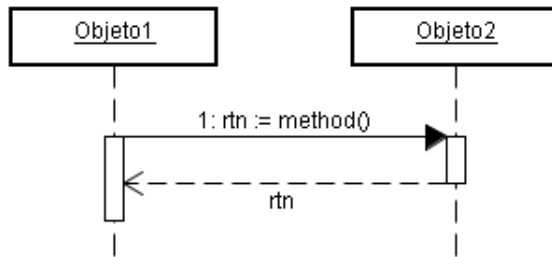


Figura 4.6: (Regra 3) Conversão de uma operação de retorno de método.

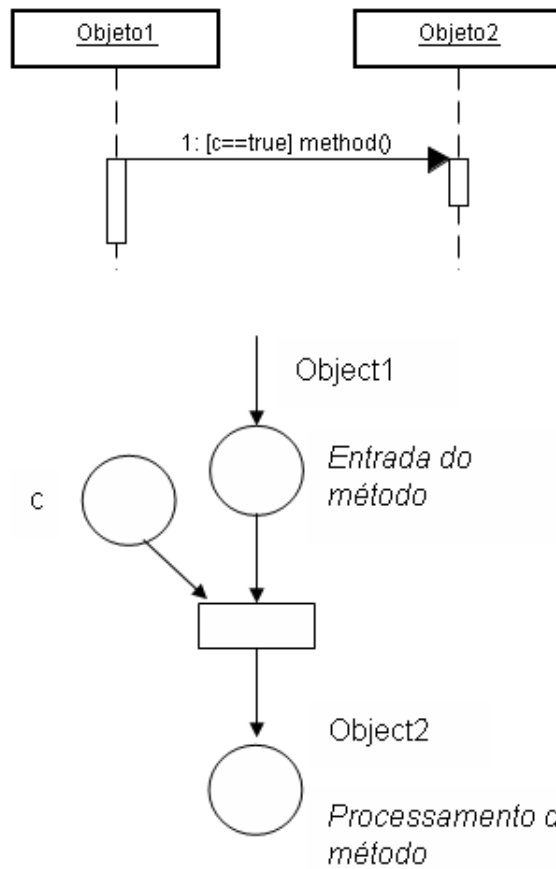


Figura 4.7: (Regra 4) Conversão de uma condição.

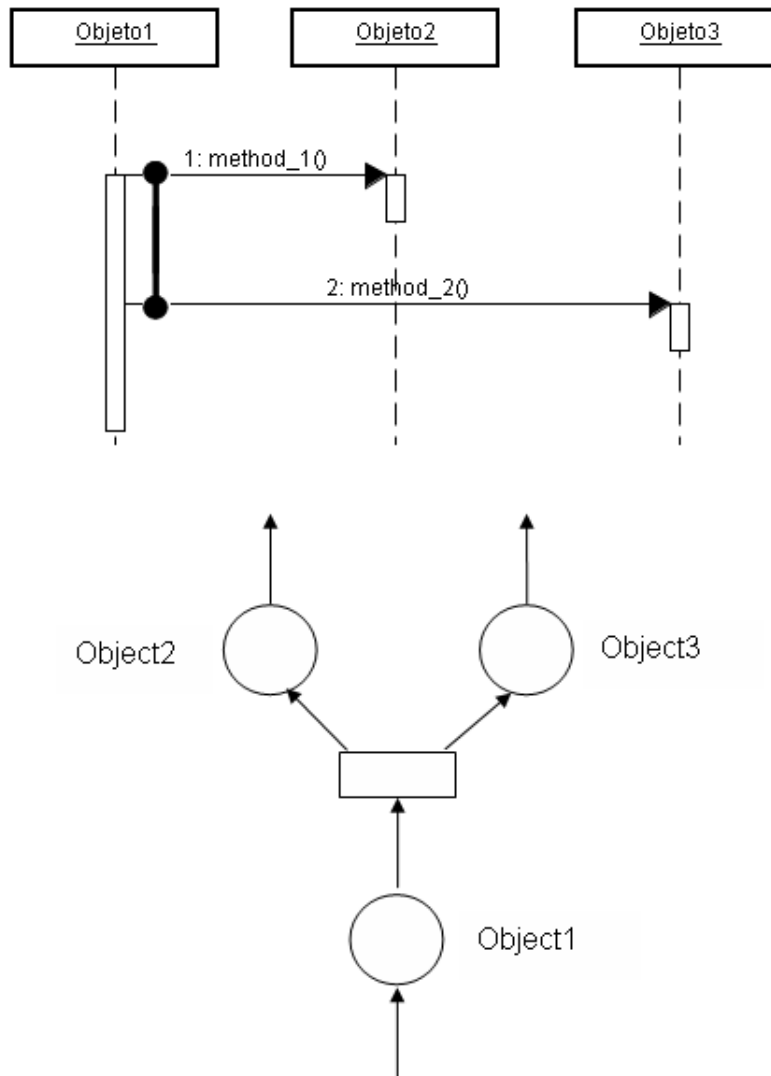


Figura 4.8: (Regra 5) Conversão de concorrência.

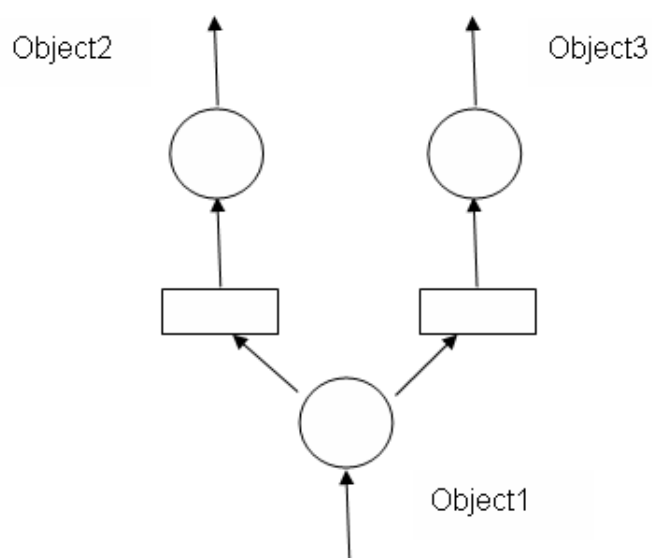
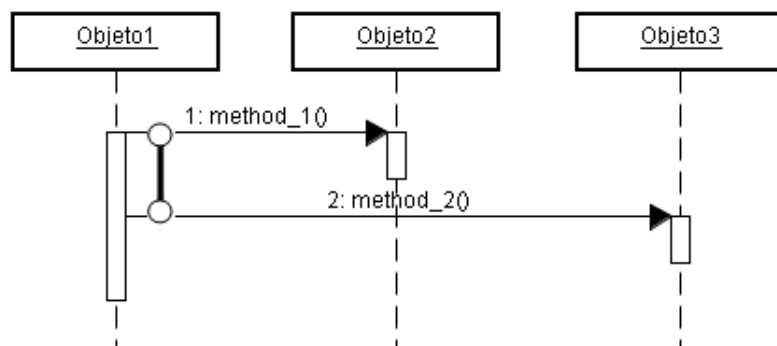


Figura 4.9: (Regra 6) Conversão de Escolha.

4.4.7 Regra 7. Conversão de sincronismo

A Figura 4.10 mostra um grupo de sincronismo, o qual é denotado por um intervalo rotulado por uma linha sólida vertical em suas setas. Após todas as mensagens no grupo de sincronismo finalizarem suas execuções, o próximo passo pode ser executado. No caso da Figura 4.10, o próximo passo é a mensagem *method3 ()*.

4.4.8 Regra 8. Conversão de confluência

Como mostra a Figura 4.11, o intervalo rotulado pela linha pontilhada vertical nas setas é chamado grupo de confluência. Diferente do grupo de sincronismo, o grupo de confluência permite que somente uma mensagem do grupo que finalize sua execução possa disparar a execução da próxima operação, no caso *method3 ()*.

4.5 Formato dos arquivos de entrada

4.5.1 Diagrama de Sequência

A seguir, na Tabela 4.1, o formato do arquivo de entrada na ferramenta contendo o diagrama de sequência tendo as tarefas de aplicação como objetos.

A notação do diagrama de sequência é bastante trivial não exigindo nenhum conhecimento novo do projetista em relação à UML. Após uma breve descrição da aplicação contida na *tag aplicacao*, seguem-se as *tags tarefa* contendo como valor cada uma das tarefas de aplicação do sistema. A seguir, a *tag mensagem* que expressa a troca de mensagens entre os objetos relevantes do domínio, no caso as tarefas descritas anteriormente. Dentro de cada

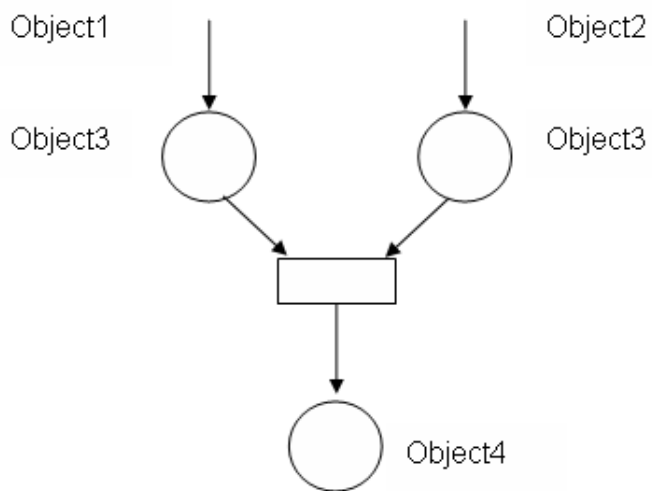
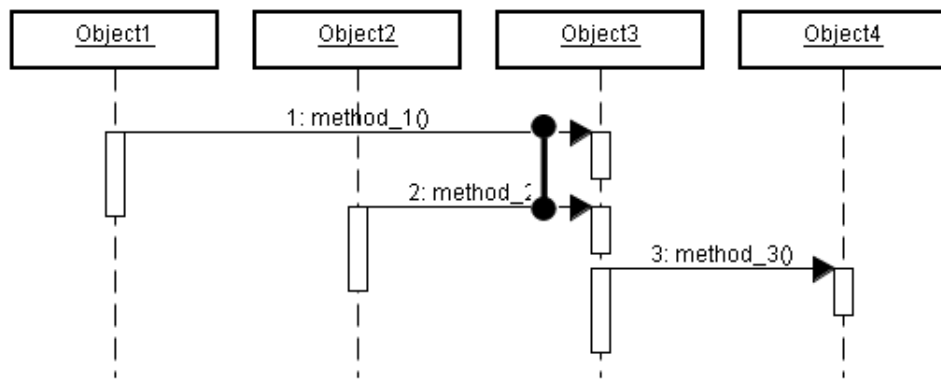


Figura 4.10: (Regra 7) Conversão de sincronismo.

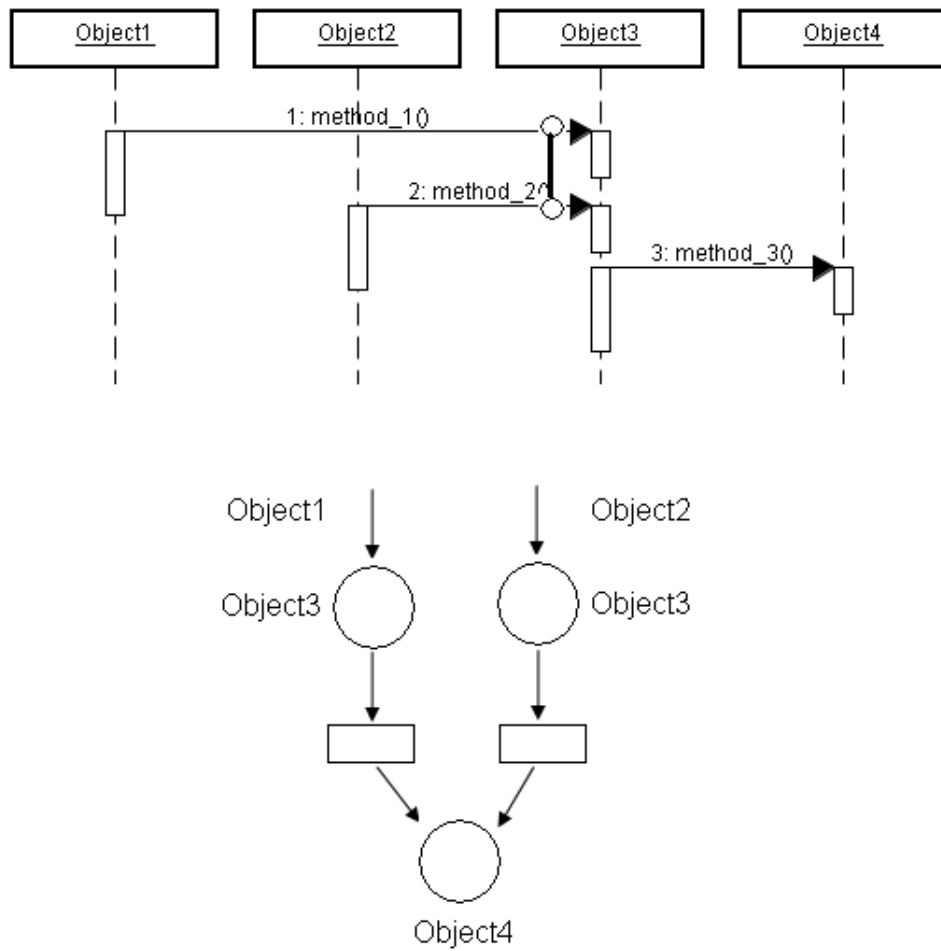


Figura 4.11: (Regra 8) Conversão de confluência.

```

<?xml version="1.0"encoding="ISO-8859-1"?>
<diagramadesequencia>
  <aplicacao>Descrição da Aplicação</aplicacao>
  <arefa>Alguma tarefa de aplicação 1</arefa>
  <arefa>Alguma tarefa de aplicação 2</arefa>
  :
  <arefa>Alguma tarefa de aplicação n</arefa>
  <mensagem>
    <de>Tarefa de Origem</de>
    <para>Tarefa de Destino</para>
  </mensagem>
</diagramadesequencia>

```

Tabela 4.1: Formato do Diagrama de Sequência

mensagem, há uma *tag* *de* com a tarefa de origem e uma *para*, com a de destino.

4.5.2 Infra-estrutura do Sistema

Este arquivo, também no formato XML, traz informações a respeito da infra-estrutura do sistema e da arquitetura da aplicação, úteis para a confecção das consultas de propriedades nas últimas etapas do método proposto neste trabalho. A seguir, na Tabela 4.2 o formato do arquivo com tal infra-estrutura.

Como se percebe pela Tabela 4.2, nas *tags* *processador*, há informações do id do processador em questão bem como dos processos que executam nele. Já nas *tags* *processo*, há informações do id do processo em questão e das tarefas que constituem tal processo. Tais tarefas foram descritas no arquivo da Tabela 4.1.

```

<?xml version="1.0"encoding="ISO-8859-1"?>

<sisistema>

  <aplicacao>Descrição da Aplicação</aplicacao>

  <processador>
    <idprocessador>Algum processador</idprocessador>
    <processos>
      <elemento>Algum processo</elemento>
    </processos>
  </processador>

  <processo>
    <id>Algum processo</id>
    <tarefas>
      <tarefa>Tarefa 1</tarefa>
      <tarefa>Tarefa 2</tarefa>
      :
      <tarefa>Tarefa n</tarefa>
    </tarefas>
  </processo>

</sisistema>

```

Tabela 4.2: Infra-estrutura e Arquitetura do Sistema

```

\beginnet{ID-REDE}

\place{ID-LUGAR} {
  \name{NOME-LUGAR}
  \init{MARCAÇÃO-INCIAL}
  \capacity{CAPACIDADE}
}

\transition{ID-TRANS} {
  \name{NOME-TRANS}
}

\arc{ID-ARCO} {
  \from{ID-ORIGEM}
  \to{ID-DESTINO}
  \weight{PESO}
  \type{TIPO}
}

\endnet

```

Tabela 4.3: Formato do Arquivo APNN

4.6 Formato do arquivos de saída

4.6.1 Arquivo APNN

A ferramenta gera um arquivo APNN (Abstract Petri Net Notation), um formato de arquivo capaz de ser usado no Model Checking Kit (MCK). Não será mostrada toda a descrição detalhada do mesmo bem como dos outros dois formatos de arquivos que vem a seguir. Para uma melhor descrição do APNN, favor referir-se a [28]. O Apêndice C traz na íntegra o arquivo APNN gerado no experimento. A seguir, na Tabela 4.3, o formato básico de um arquivo APNN.

4.6.2 Arquivo PNML

O segundo formato de saída é o Petri Net Markup Language [36]. Este formato foi escolhido por causa de sua grande adoção em várias ferramentas e, portanto, pela ampla possibilidade de compatibilidade com as mesmas. O

Apêndice D traz na íntegra o arquivo PNML gerado no experimento. Na Tabela 4.4, o formato do arquivo em questão.

4.6.3 Arquivo SMV

O formato SMV serve como entrada para o *model checker* de mesmo nome que é usada no Capítulo 5 para realizar consultas de propriedades do sistema. Ele é melhor descrito em [25] e o Apêndice E traz na íntegra o arquivo SMV gerado no experimento. A Tabela 4.5 traz o formato do arquivo SMV.

A “variável” é um dos lugares gerados na rede de Petri. O lugar inicial na rede é “lugar_Start” e ele sempre terá um *token* inicialmente. A seguir, na seção SPEC, tem-se uma fórmula CTL traduzida a partir da interface de formulação de consultas em alto nível. Além do módulo principal `main`, há um módulo para cada processo. Cada um destes módulos contém no mínimo uma variável, que é uma tarefa de aplicação do sistema. Em cada um destes módulos, novos valores das variáveis são atribuídos dentro da construção “next”.

4.7 Projeto do Ambiente

A ferramenta Ambiente de Verificação Formal de Software Embarcado foi desenvolvida utilizando-se uma metodologia iterativa baseada em prototipagem, com *feedbacks* constantes. A linguagem escolhida foi a Java por suas características de orientação a objetos e portabilidade, além da grande disponibilidade de *APIs* para construção de *GUIs*.

O pacote `ambiente` como mostrado nas Figuras 4.12 e 4.13 é responsável

```

<? xml version="1.0" encoding="ISO-8859-1" ?>

<pnml xmlns="XMLNS">

  <net id="ID-REDE" type="URI">

    <name>
      <text>NOME-REDE</text>
    </name>

    <transition id="TRANS-ID">
      <graphics>
        <position x="0" y="0" />
      </graphics>
      <name>
        <text>TRANS-NOME</text>
        <graphics>
          <offset x="0" y="0" />
        </graphics>
      </name>
    </transition>

    <arc id="ARCO-ID" source="ID-ORIGEM" target="ID-DESTINO">
      <inscription>
        <text>PESO</text>
        <graphics>
          <offset x="0" y="0" />
        </graphics>
      </inscription>
      <graphics>
        <position x="0" y="0" />
      </graphics>
    </arc>

    <place id="LUGAR-ID">
      <graphics>
        <position x="0" y="0" />
      </graphics>
      <name>
        <text>LUGAR-NOME</text>
        <graphics>
          <offset x="0" y="0" />
        </graphics>
      </name>
      <initialMarking>
        <text>MARCAÇÃO-INICIAL</text>
      </initialMarking>
    </place>

  </net>

</pnml>

```

Tabela 4.4: Formato do Arquivo PNML

```

MODULE main
VAR
  <variável> : boolean;
  <processo N> : process <nome do módulo> (<lista de variáveis>);

ASSIGN
  init(lugar_Start) := 1;
  init(<variável>) := 0;

SPEC
  <fórmula CTL>

MODULE <nome do módulo> (<lista de variáveis>)
ASSIGN
  next(<variável>) :=
    case
      <variável>=0 : 1;
      1 : <variável>;
    esac;

```

Tabela 4.5: Formato do Arquivo SMV

ambiente::CapturaRequisitos	ambiente::DistribuirTarefas
+ diagramaDeSequencia : DiagramaDeSequencia + redeDePetri : RedeDePetri + projeto : Projeto + CapturaRequisitos0 - carregaRequisitos(arquivo : String) : void - initComponents0 : void - jButton1ActionPerformed(evt : ActionEvent) : void - jButton2ActionPerformed(evt : ActionEvent) : void - jButton3ActionPerformed(evt : ActionEvent) : void - jButton4ActionPerformed(evt : ActionEvent) : void + main(args : String[]) : void	+ DistribuirTarefas0 - initComponents0 : void - jButton4ActionPerformed(evt : ActionEvent) : void - imprime0 : void - formWindowActivated(evt : WindowEvent) : void - jButton2ActionPerformed(evt : ActionEvent) : void - jButton3ActionPerformed(evt : ActionEvent) : void - jButton5ActionPerformed(evt : ActionEvent) : void + main(args : String[]) : void

Figura 4.12: Pacote ambiente - parte 1 de 2

pelas classes que coletam os processos e as tarefas, escrevem a rede de Petri gerada em formatos de arquivos de saída específicos e, por fim, formulam consultas em linguagem natural e as traduzem para CTL.

O pacote `geradores`, mostrado na Figura 4.14, atua juntamente com o `ambiente` pra escrever a rede de Petri gerada nos três formatos: APNN, SMV e PNML.

O pacote `modelo.diagramadesequencia` da Figura 4.15 traz as classes básicas para se lidar com o Diagrama de Sequência lido na entrada pelo parser (4.18 e seus elementos constituintes. Este mesmo diagrama será posteriormente traduzido para rede de Petri pelo `tradutor` (4.20).

O pacote `modelo.programaSMV` na Figura 4.16 oferece o suporte à tradução da rede de Petri para o formato SMV e o pacote `modelo.rededepetri`, na Figura 4.17, para as redes de Petri, saída do `tradutor` (4.20).

4.8 Definição das consultas

A Figura 4.1 mostra que há uma etapa de formulação de consultas em alto nível ou linguagem natural. A interface de definição de consultas foi projetada pensando-se em coletar consultas em alto nível e então traduzi-las para um formato capaz de ser executado por um *model checker*. O formato escolhido foi o do Computational Tree Logic (CTL). Basicamente, as construções CTL abordadas são as ordinárias com excessão de *weak until* (W). Como foi dito, isso facilita o processo de desenvolvimento do software embarcado, livrando o projetista da obrigação de dominar notações formais e agilizando todo o processo.

O próximo Capítulo traz o experimento proposto a fim de validar o pre-

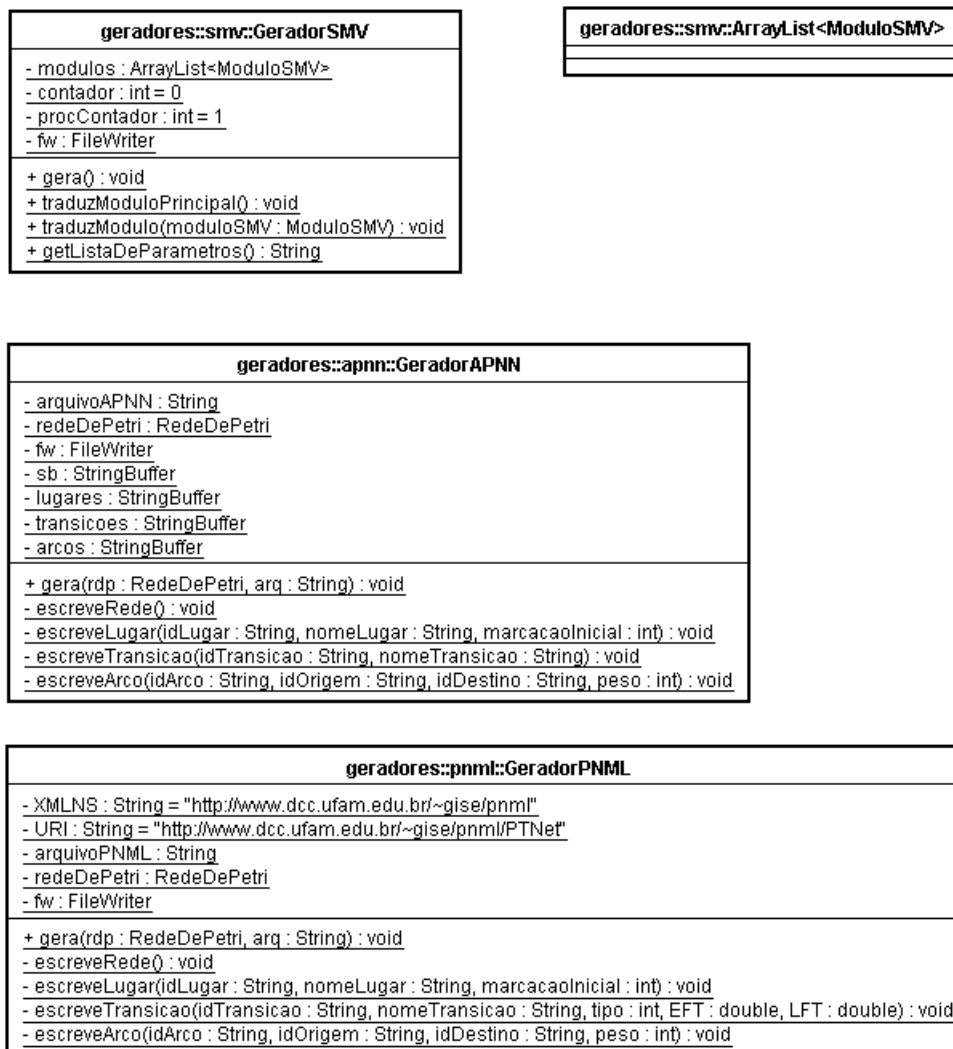


Figura 4.14: Pacote geradores

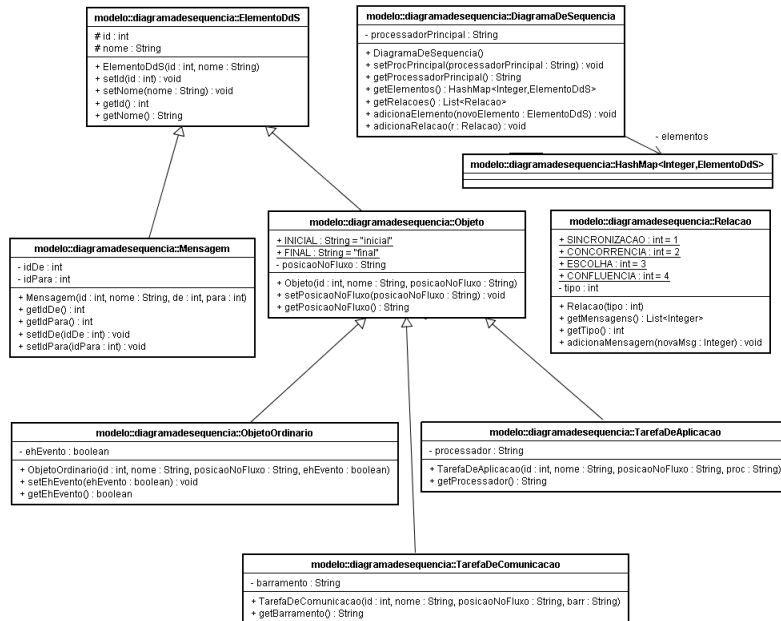


Figura 4.15: Pacote modelo.diagramadesequencia

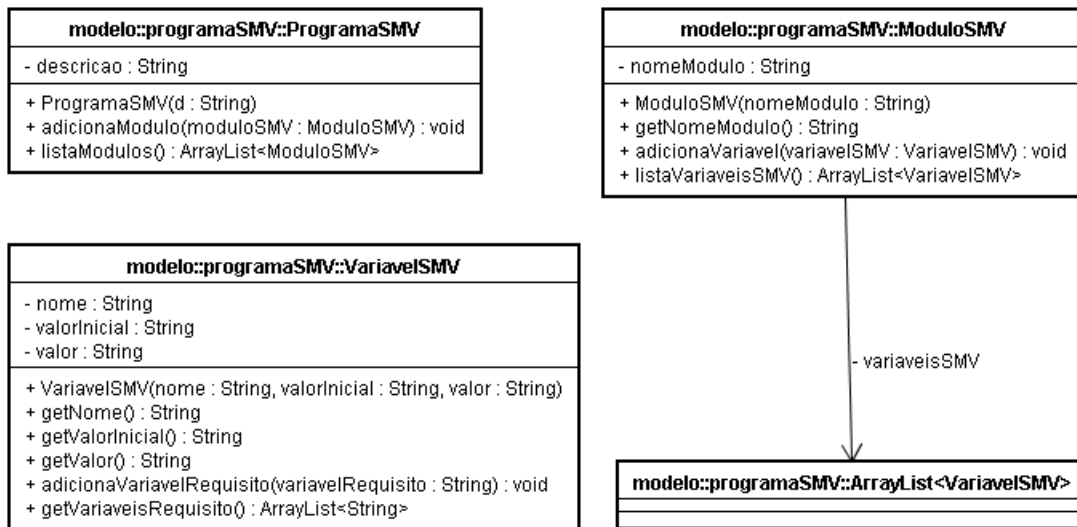


Figura 4.16: Pacote modelo.programaSMV

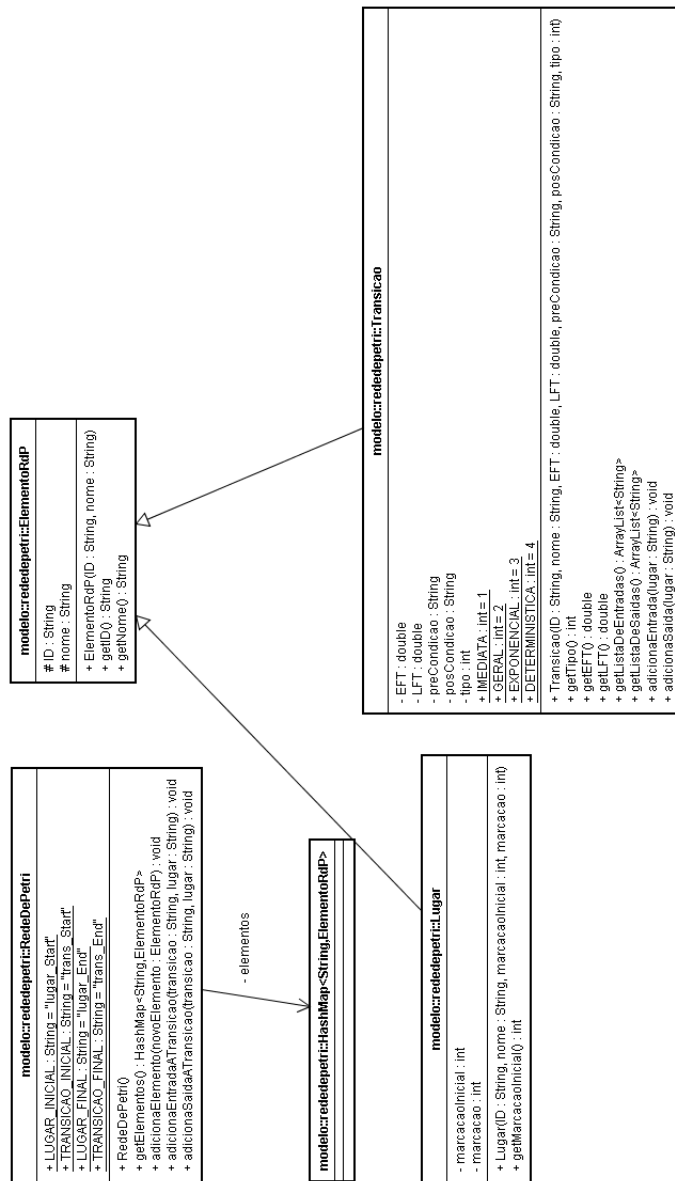


Figura 4.17: Pacote modelo.rededepetri

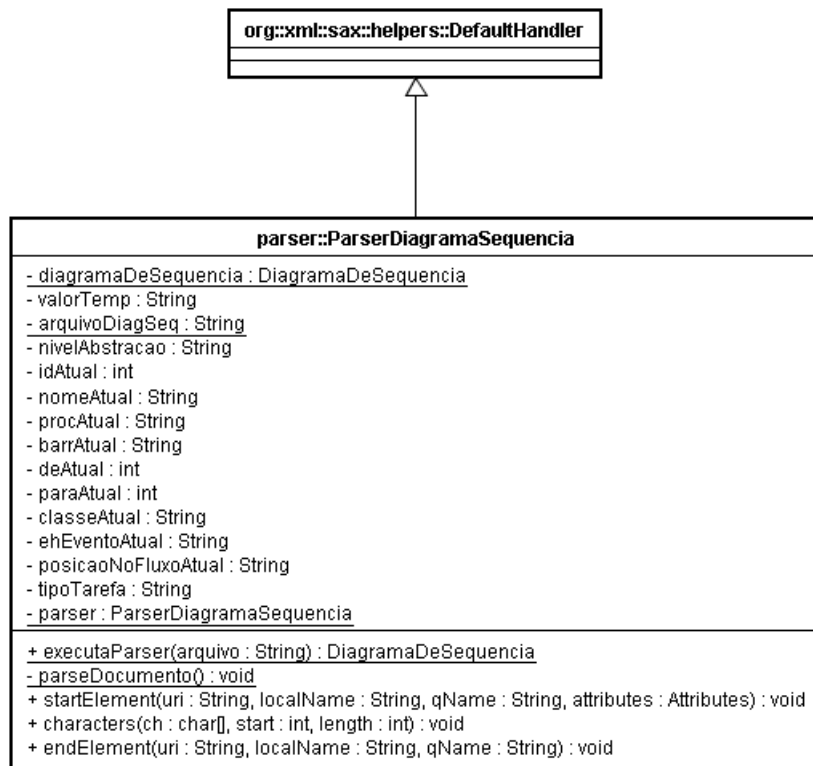


Figura 4.18: Pacote parser

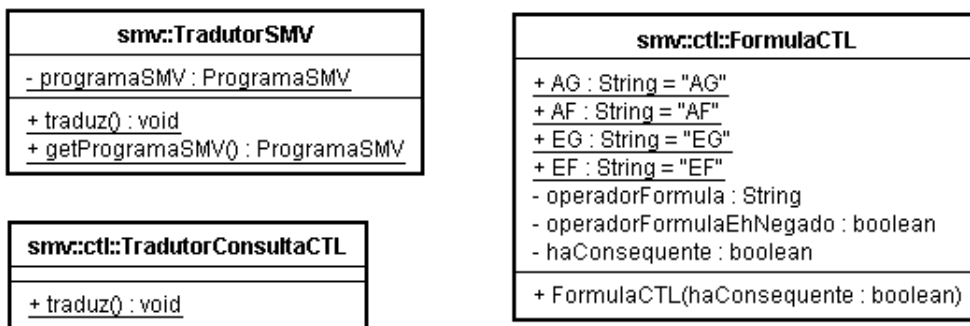


Figura 4.19: Pacote smv

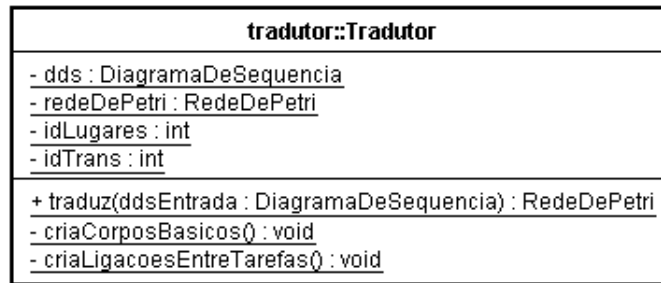


Figura 4.20: Pacote tradutor

sente método. O próximo Capítulo apresenta também capturas de tela de consultas e traduções realizadas no Ambiente de Verificação Formal de Software Embarcado. O funcionamento geral do Ambiente - que realiza o método proposto neste trabalho na sua íntegra - é melhor compreendido com o experimento do próximo Capítulo.

Capítulo 5

Experimento

5.1 Descrição

O cenário levado em consideração nos experimentos é o de uma aplicação simples de controle, a *Simple Control Application*, descrita originalmente em [11]. Ela consiste basicamente num modelo de processos P_1, P_2, \dots, P_n , onde cada processo é dividido em uma sequência de tarefas $T_{i_1}, T_{i_2}, \dots, T_{i_n}$. As tarefas são unidades computacionais indivisíveis do ponto de vista de escalonamento, neste contexto. O sistema consiste fisicamente em de um dispositivo sensorial montado sobre uma plataforma motorizada que deve detectar e rastrear objetos específicos em um ambiente. O sistema é controlado por quatro processadores conectados por meio de um barramento. O primeiro processador controla o sensor e executa dois processos: SENSOR e SENSOR_MNG. O segundo processador controla os atuadores e executa os processos ACT_CTRL e ACTUATOR_MNG. Ambos se comunicam com outro processador rodando a principal atividade de controle MAIN_CONTROL e são responsáveis pela detecção de eventos significantes

detectados no ambiente. O último processador, rodando o processo SIGNAL, é responsável pelo gerenciamento de tais eventos significantes detectados no ambiente (e.g., a geração de alarme). O fluxo de controle da aplicação é o seguinte: O processo MAIN_CONTROL comunica-se remotamente ao processo SENSOR_MNG o pedido por uma nova atividade. O SENSOR_MNG, localmente solicita dados sensoriais do processo SENSOR. Quando os dados sensoriais são retornados, o processo SENSOR_MNG extrai características significantes do objeto a ser rastreado e comunica de volta os resultados para o processo MAIN_CONTROL. A seguir, o MAIN_CONTROL envia a informação da posição para o processo ACTUATOR_MNG que se comunica com o processo ACT_CONTRL para realizar o rastreamento do objeto identificado e mantê-lo no campo de visão dos sensores. Ao mesmo tempo, o processo MAIN_CONTROL verifica a presença de eventos significantes no sistema e envia um relatório ao processador rodando o processo SIGNAL, responsável pela detecção e ativação dos alarmes e avisos. Ademais, todas as tarefas são caracterizadas pelas restrições temporais padrão, a saber, deadline, tempo de liberação, de computação etc. Entretanto, neste experimento, no processo de tradução de diagramas de sequência para rede de Petri, não serão consideradas tais restrições. Apenas as unidades computacionais mínimas, as tarefas. Para entender melhor o fluxo do sistema, a Figura 5.1 traz um grafo de precedência com os processos e suas tarefas divididos pelos quatro processadores que compõem o sistema.

Assim, a partir da Figura 5.1, tem-se a seguinte distribuição das tarefas por processo:

1. MAIN_CONTROL

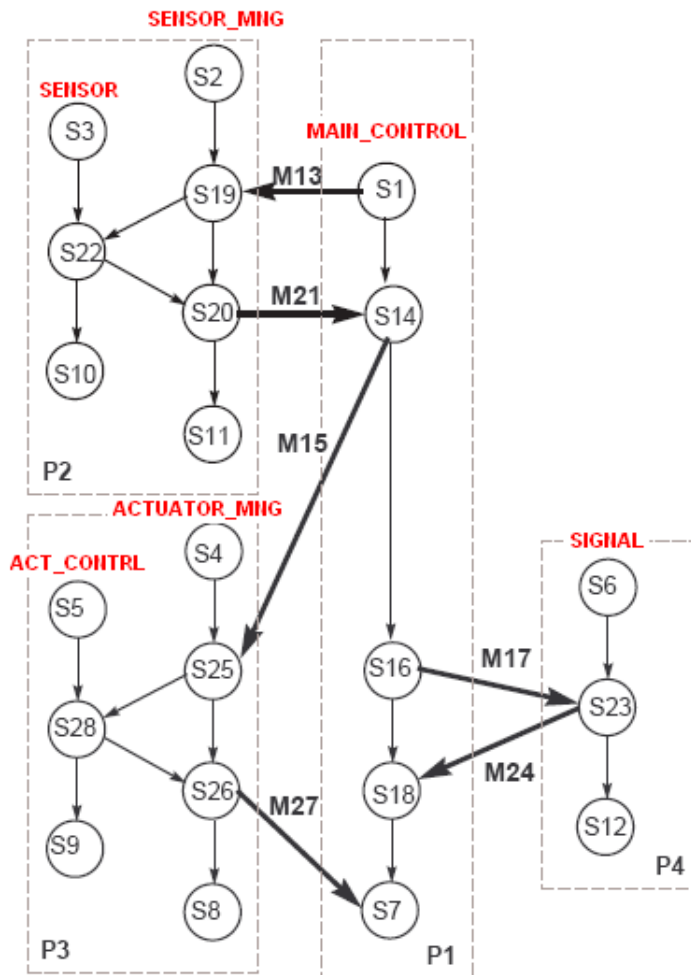


Figura 5.1: Grafo de Precedências do Simple Control Application

- S1
- S14
- S16
- S18
- S7

2. SENSOR_MNG

- S2
- S19
- S20
- S11

3. SENSOR

- S3
- S22
- S10

4. ACTUATOR_MNG

- S4
- S25
- S26
- S8

5. ACT_CONTRL

- S5
- S28
- S9

6. SIGNAL

- S6
- S23
- S12

Conforme a Figura 5.1 também, há ainda as seguintes tarefas de comunicação ou mensagens entre as tarefas ou objetos: M13, M21, M15, M17, M24 e M27. Essas não fazem parte propriamente dita de nenhum dos processos. São apenas conectores que viabilizam o fluxo da ação dos processos do sistema.

5.2 Diagrama de Sequência da Entrada

Os diagramas de sequência, conforme exposto na Seção 3.2.3, capturam os aspectos dinâmicos do sistema, enfatizando a ordem temporal dos eventos no mesmo. As partes de diagrama das Figuras 5.2, 5.3, 5.4, 5.5, 5.6 e 5.7 modelam tais aspectos no contexto do *Simple Control Application*. Como é obrigação do projetista entrar com tais diagramas e no formato de entrada adequado (ver Apêndice A), de acordo com o método do presente trabalho detalhado na Figura 4.1, os mesmos serão apenas apresentados aqui e não explicados. É válido porém dizer que, pela notação estendida introduzida na Seção 4.3, percebe-se a presença de várias linhas verticais entre mensagens

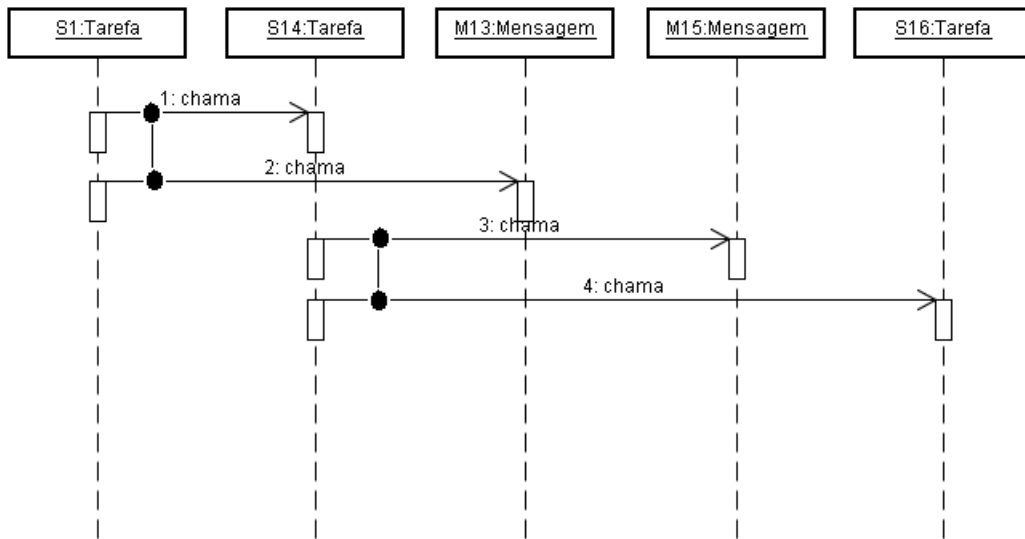


Figura 5.2: Diagrama de Sequência da Entrada. Parte 1 de 6

e círculos sólidos caracterizando-se, portanto, situações de concorrência e de junção. Neste experimento, em particular, não há situações de escolha ou de confluência. O arquivo XML com a infra-estrutura do sistema é apresentado na íntegra no Apêndice B.

5.3 Aplicando-se as Regras de Tradução

Considere o pedaço de diagrama de sequência contida na Figura 5.2. Nele, pela notação estendida introduzida na Seção 4.3, percebe-se que, pela presença de linhas verticais e círculos sólidos, há dois grupos ou situações de concorrência. A primeira é entre S1 e S14. E a segunda é entre S14 e S16. No primeiro caso, ambas as tarefas irão ser inicializadas ao mesmo tempo. Porém, elas são tarefas iniciais no fluxo de seus requisitos e portanto têm como predecessora imediata o lugar inicial. Este lugar inicial, então, será

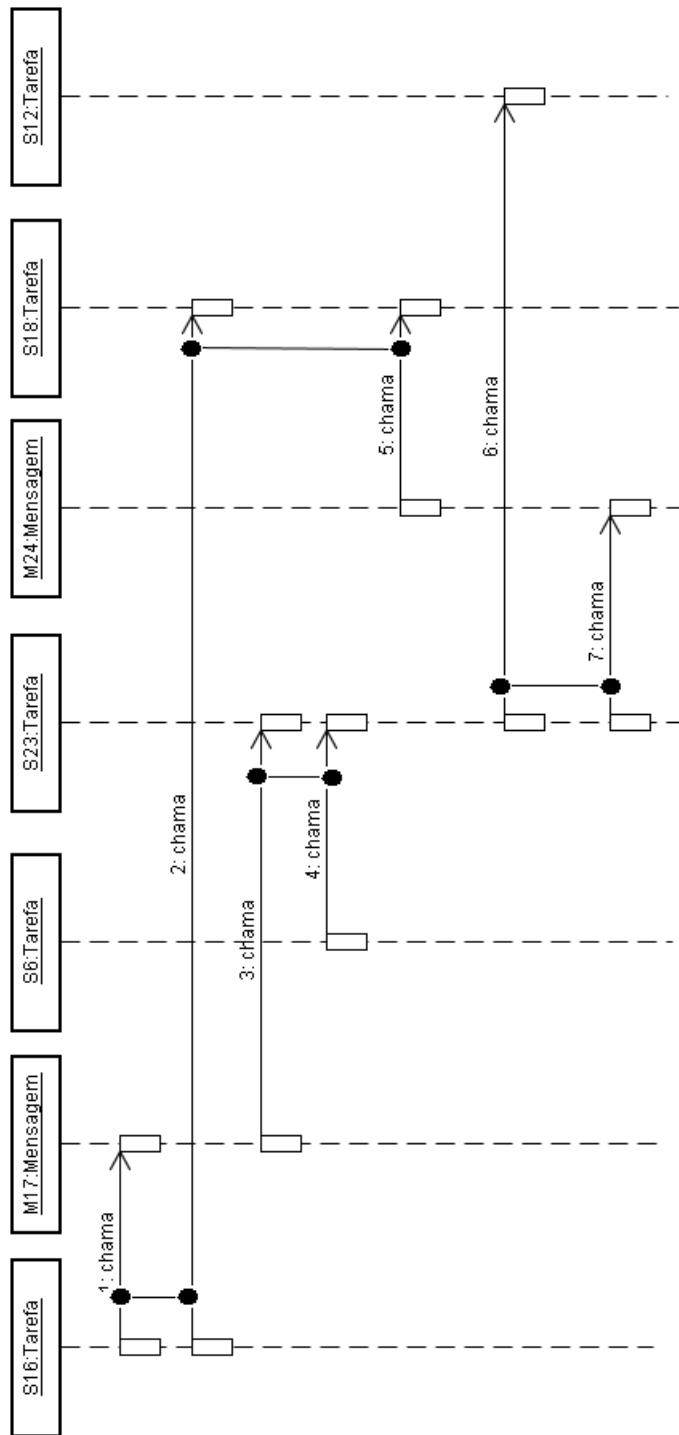


Figura 5.3: Diagrama de Sequência da Entrada. Parte 2 de 6

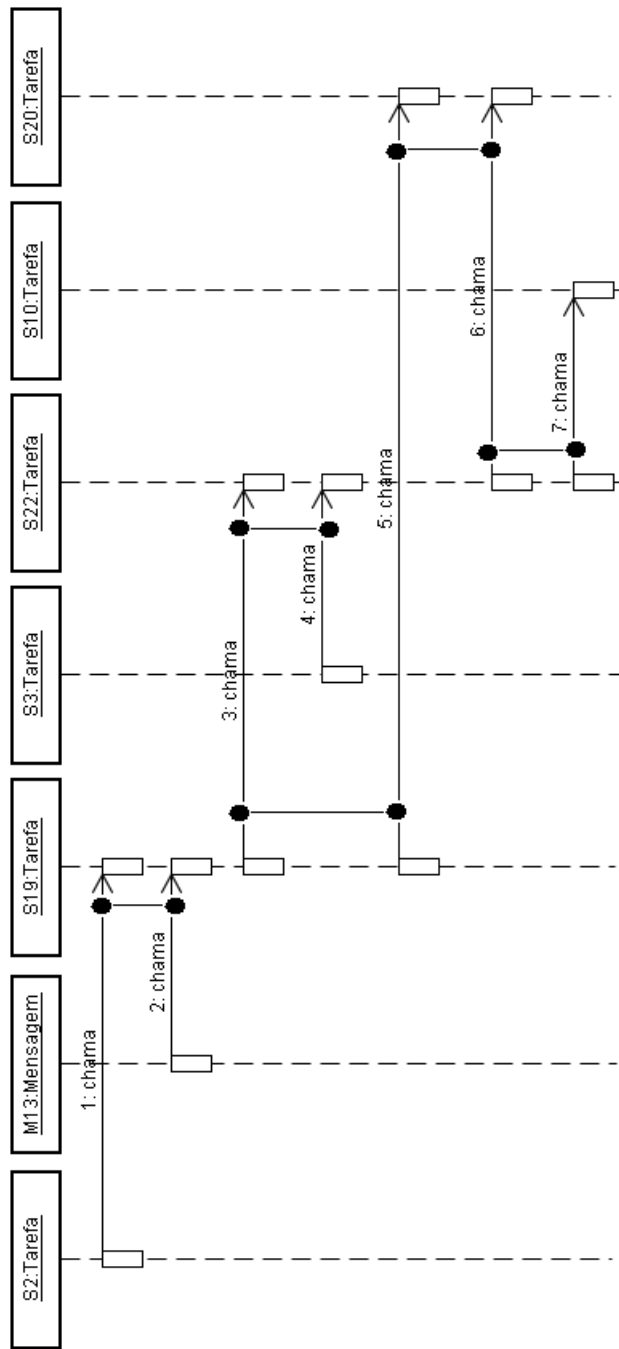


Figura 5.4: Diagrama de Sequência da Entrada. Parte 3 de 6

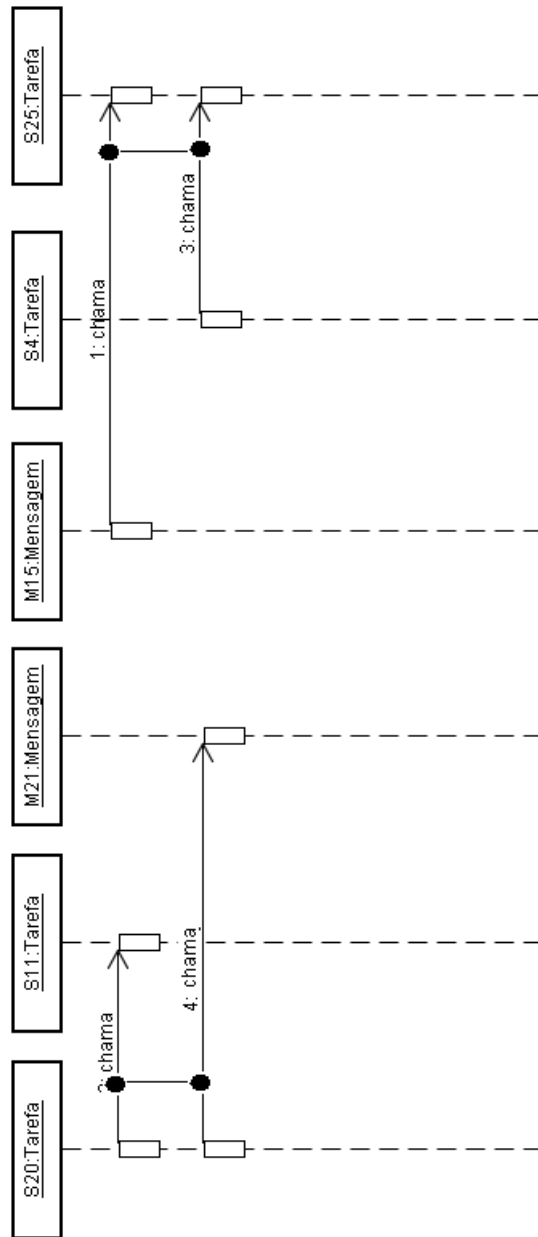


Figura 5.5: Diagrama de Sequência da Entrada. Parte 4 de 6

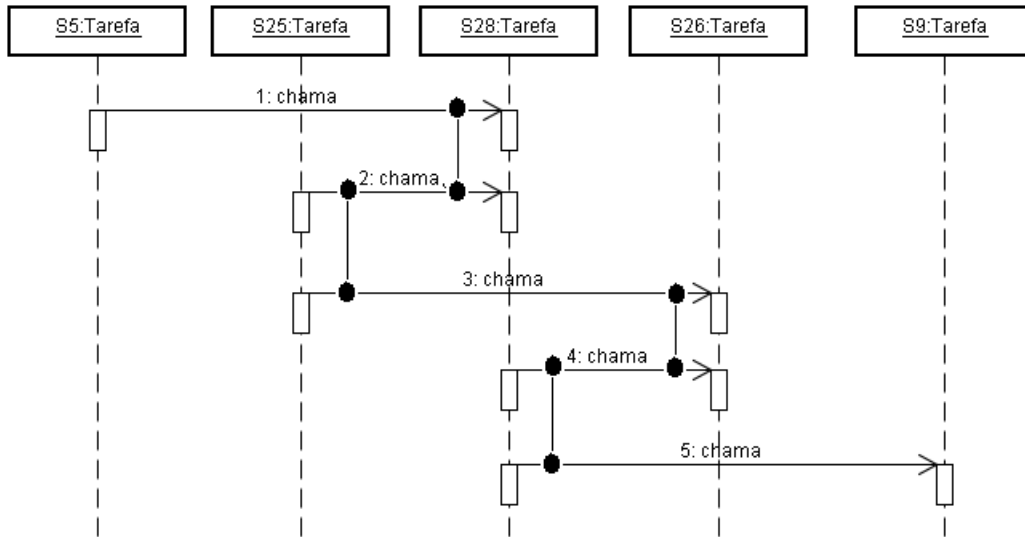


Figura 5.6: Diagrama de Sequência da Entrada. Parte 5 de 6

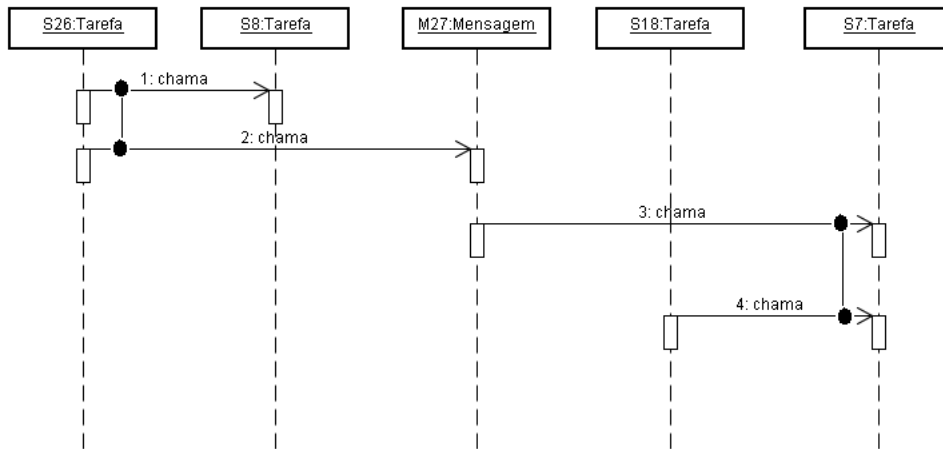


Figura 5.7: Diagrama de Sequência da Entrada. Parte 6 de 6

entrada para uma transição t que terá como lugares de saída os lugares de ganho de processador de S1 e S14. Logo, a transição t iniciará as tarefas S1 e S14 em paralelo. No segundo caso, S14 e S16 também possuem uma transição anterior comum t que os dispara em paralelo. Todas essas traduções e outras estão presentes na rede de Petri gerada na Figura 5.8.

5.3.1 Características da Rede Gerada

Usando-se o software Snoopy 2.0, transforma-se a rede para o formato PNT, o que serve de entrada para o analisador Integrated Net Analyzer (INA).

A seguir, as principais características da rede de Petri gerada, apenas citadas:

- Estaticamente livre de conflitos
- Dinamicamente livre de conflitos
- Pura
- Ordinária
- Homogênea
- Não-conservativa
- Não é uma máquina de estados
- Livre escolha
- Não viva
- Marcada com exatamente um *token*

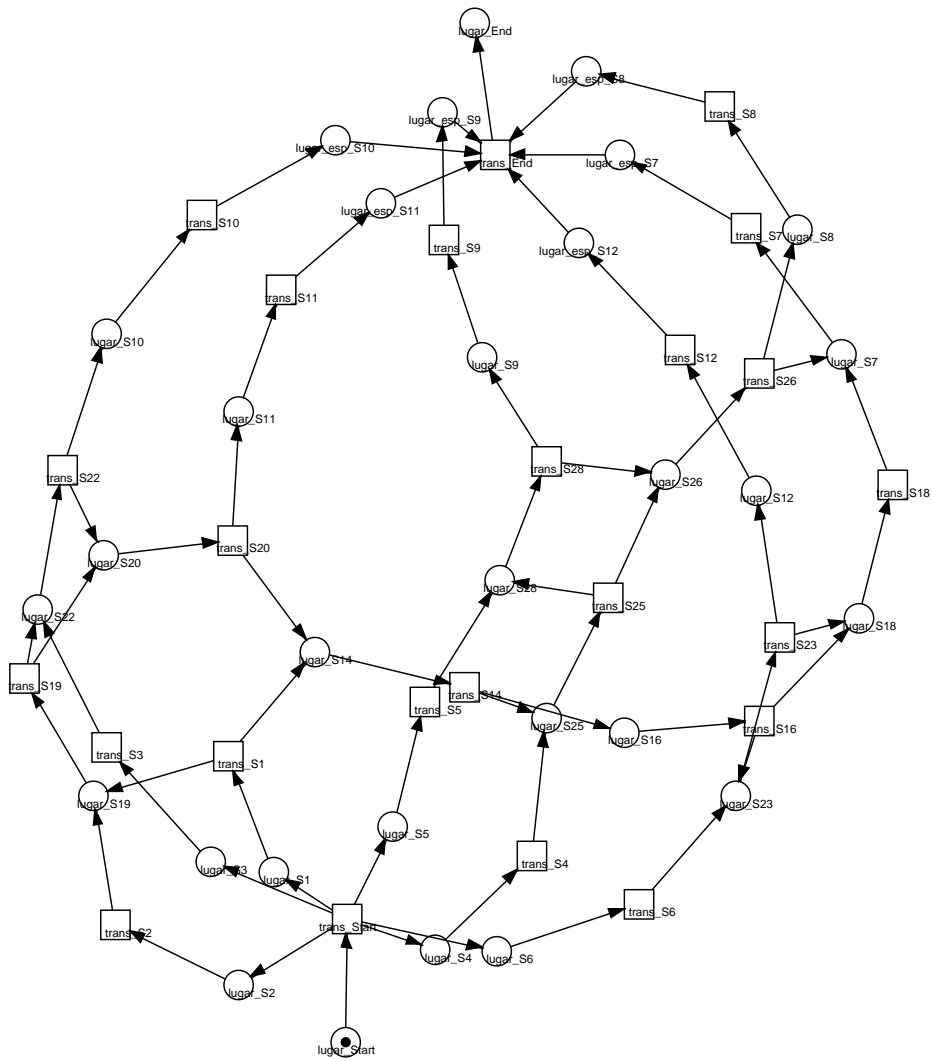


Figura 5.8: Rede de Petri Completa

- Limitada
- Estruturalmente limitada
- A rede não tem transições sem lugares antes ou sem lugares depois
- Conectada

5.4 Utilizando a Ferramenta para Realizar o Método

Esta Seção apresenta o passo-a-passo necessário para se ter o software embarcado distribuído em mãos formalmente verificado.

Passo 1 - Entrada do diagrama de sequência

A captura de tela da Figura 5.9 mostra a opção de *menu* a ser acessada a fim de se carregar o arquivo XML com o diagrama de sequência.

Passo 2 - Tradução do Diagrama de Sequência para Rede de Petri

A tradução para Rede de Petri é realizada e o modelo existe internamente ao software. A captura de tela da Figura 5.10 mostra a opção de *menu* a ser acessada bem como a janela interna que surge para gerar tais formatos. Como se percebe pela Figura, os formatos podem ser APNN ou PNML, extensamente descritos em Seções anteriores.

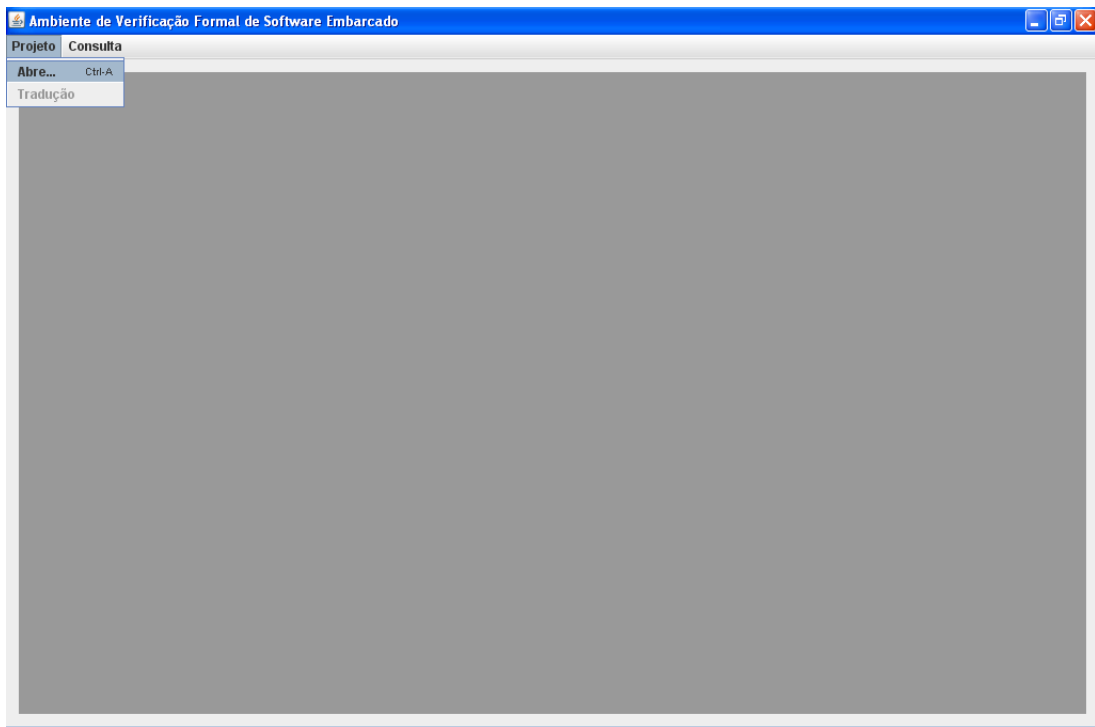


Figura 5.9: Passo 1 - Entrada do diagrama de sequência

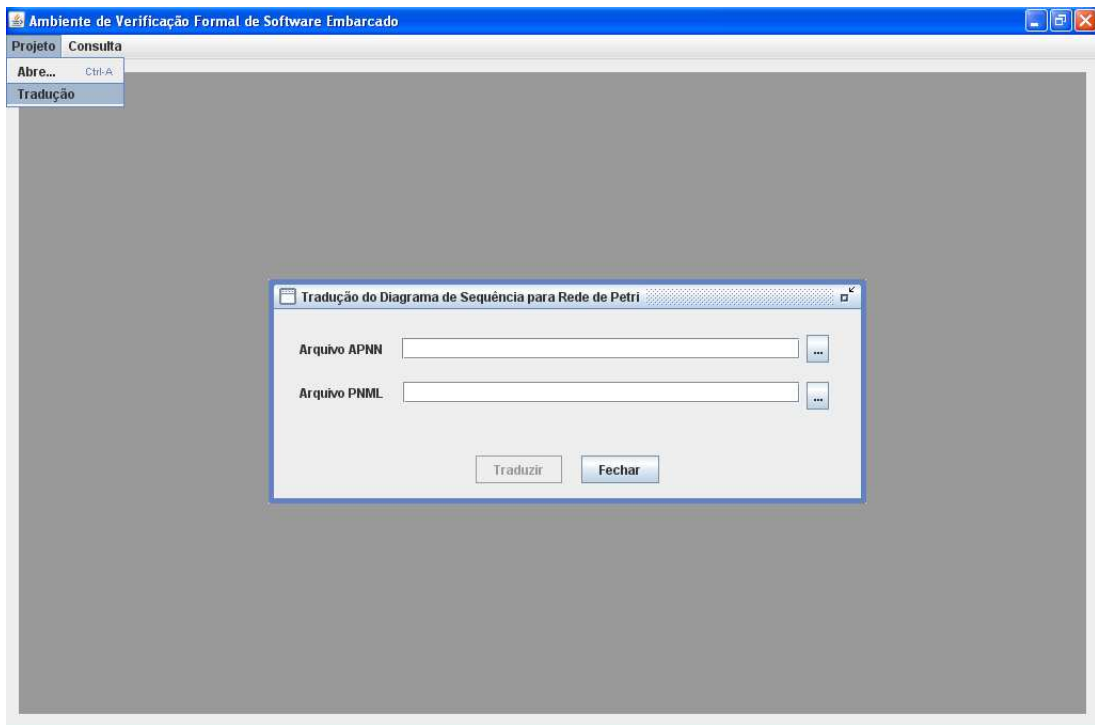


Figura 5.10: Tradução para Rede de Petri

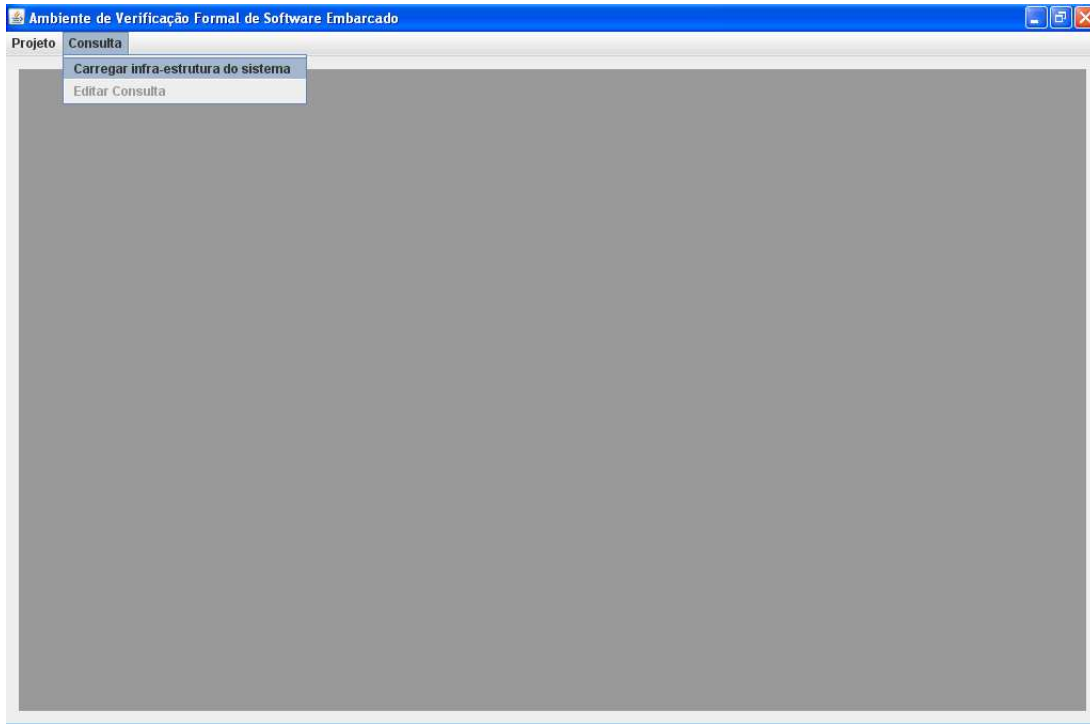


Figura 5.11: Carregando infra-estrutura

Passo 3 - Carregar infra-estrutura

A captura de tela da Figura 5.11 carrega a infra-estrutura do sistema e a arquitetura da aplicação. Tal arquivo é apresentado na íntegra no Apêndice B.

Passo 4 - Edição de Consultas

Com a infra-estrutura carregada, o *menu* “Editar Consultas” é habilitado. Escolhendo-se tal *menu*, surge a janela interna de edição de consultas. Exemplos de tela deste quarto e último passo são mostrados mais a frente, na Seção 5.5. Arquivos no formato SMV são gerados nesta etapa, após pressionar-se

o botão “CONSULTAR”.

5.5 Consultas de Propriedades do Sistema

Consultas mais triviais serão mostradas de início. As mais complexas serão deixadas para o fim. Ao total, foram realizadas **14** consultas visando verificar formalmente as mais variadas propriedades do sistema em questão, no caso, a Aplicação Simples de Controle.

É válido notar que, para utilizar a interface de geração de consultas do Ambiente de Verificação Formal de Software Embarcado, o usuário deve compreender o conceito de caminhos ou possibilidades de execução. Eles consistem em várias ramificações possíveis em que o sistema se encontra em um estado, caracterizado por valores diferentes das variáveis. Estas variáveis são as tarefas dos processos. É necessário entender também que, dentro de um caminho - ou de vários - uma variável pode estar executando apenas no próximo momento em relação ao momento atual, eventualmente ou continuamente do momento atual ao fim da execução do sistema.

5.5.1 Consulta I

A tarefa **S1**, em todos os caminhos ou possibilidades de execução, continuamente do início ao fim em tais caminhos, executa ?

A Figura 5.12 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL

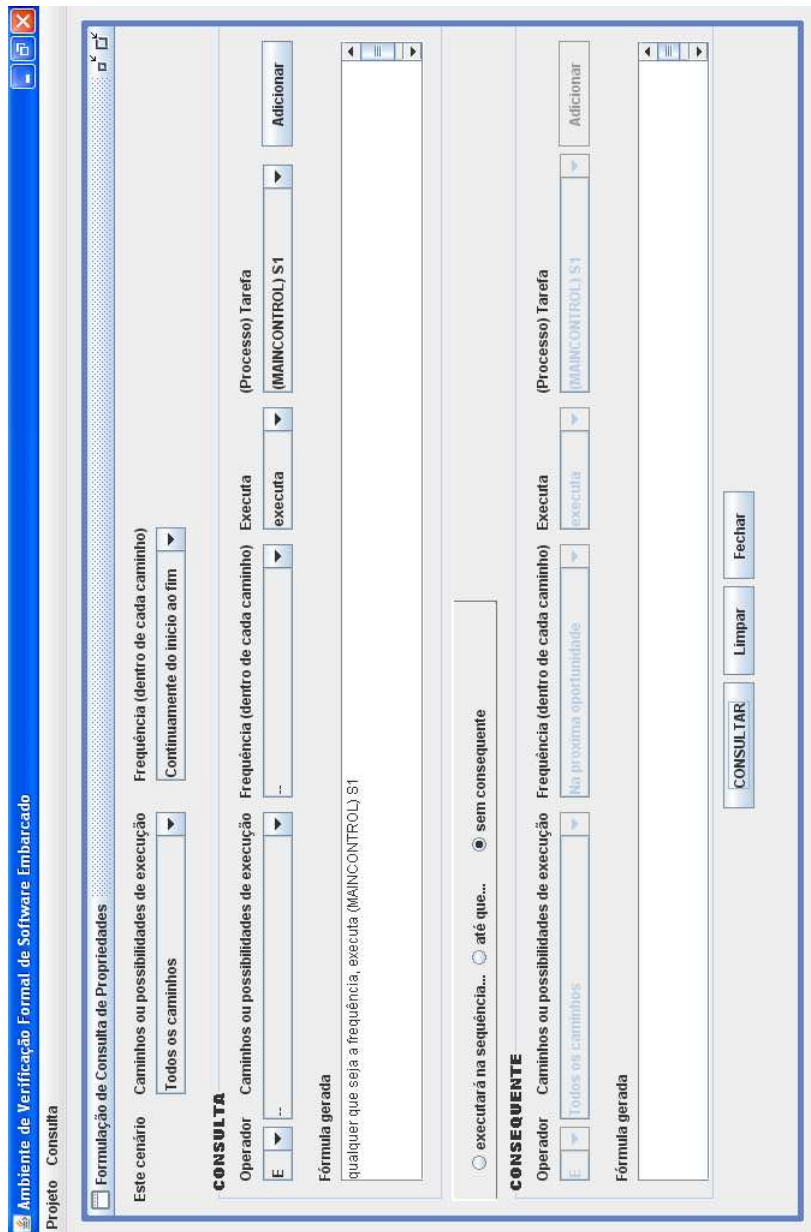


Figura 5.12: Primeira Consulta: AG S1

traduzida: AG S1.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification AG S1 is false
-- as demonstrated by the following execution sequence
state 1.1:
lugar_Start = 1
S1 = 0
S14 = 0
S16 = 0
S18 = 0
S7 = 0
S2 = 0
S19 = 0
S20 = 0
S11 = 0
S3 = 0
S22 = 0
S10 = 0
S4 = 0
S25 = 0
S26 = 0
S8 = 0
S5 = 0
S28 = 0
S9 = 0
S6 = 0
S23 = 0
S12 = 0

resources used:
processor time: 0.016 s,
BDD nodes allocated: 3053
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é FALSO e o contra-exemplo é mostrado no *log* do SMV. Logo no primeiro estado, a tarefa S1 já não é executada, tornando falha a consulta ou verificação de propriedade.

5.5.2 Consulta II

A tarefa S1, em pelo menos um caminho ou possibilidade de execução, eventualmente em tais caminhos, executa ?

A Figura 5.13 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: EF S1.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

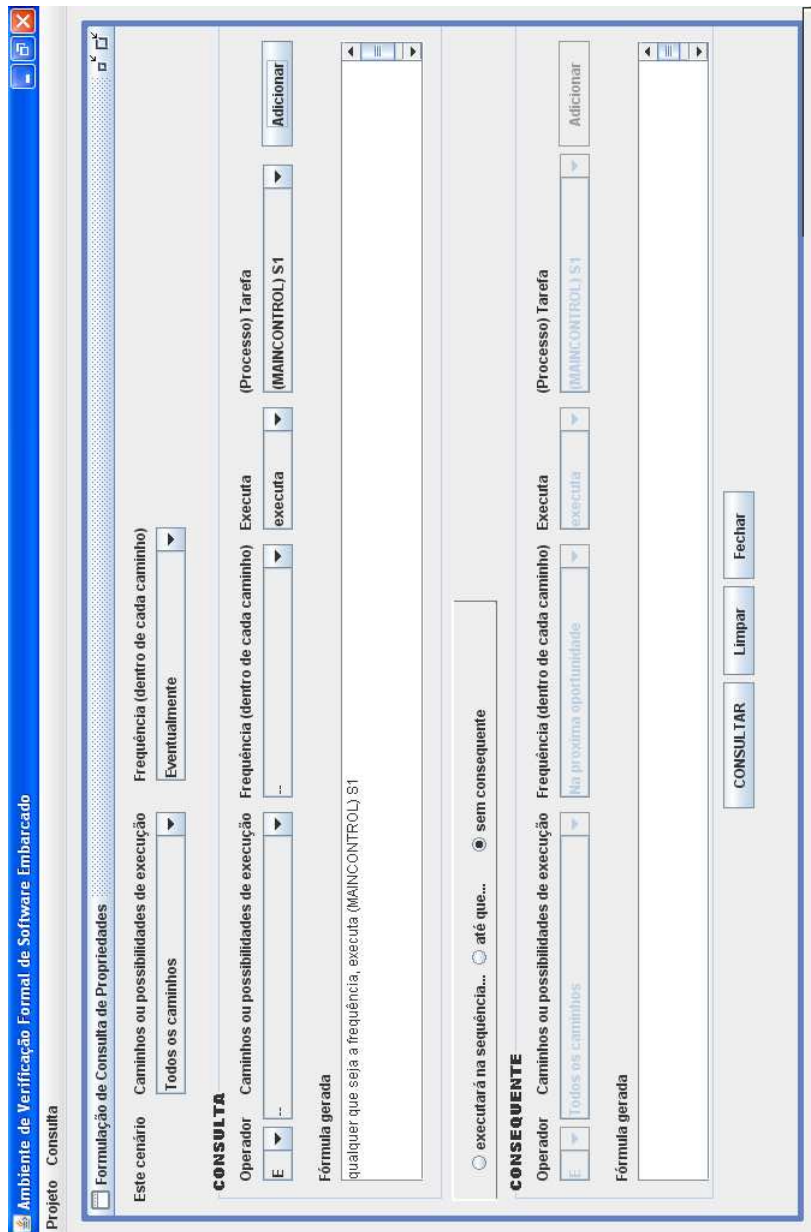


Figura 5.13: Segunda Consulta: EF S1

```

-- specification EF S1 is true

resources used:
processor time: 0 s,
BDD nodes allocated: 2927
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1

```

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV garantido o correto funcionamento do sistema pelo menos no que diz respeito à execução da tarefa S1 em algum caminho ou ramo de execução no futuro.

5.5.3 Consulta III

Em todos os caminhos ou possibilidades de execução, continuamente do início ao fim em tais caminhos, todas as tarefas do processo MAINCONTROL, a saber, S1, S14, S16, S18 e S7, serão executadas em paralelo estando portanto ativadas em todos os instantes?

A Figura 5.14 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $AG (S1 \ \& \ S14 \ \& \ S16 \ \& \ S18 \ \& \ S7)$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```

-- specification AG (S1 & S14 & S16 & S18 & S7) is false
-- as demonstrated by the following execution sequence
state 1.1:
lugar_Start = 1
S1 = 0
S14 = 0
S16 = 0
S18 = 0
S7 = 0
S2 = 0

```

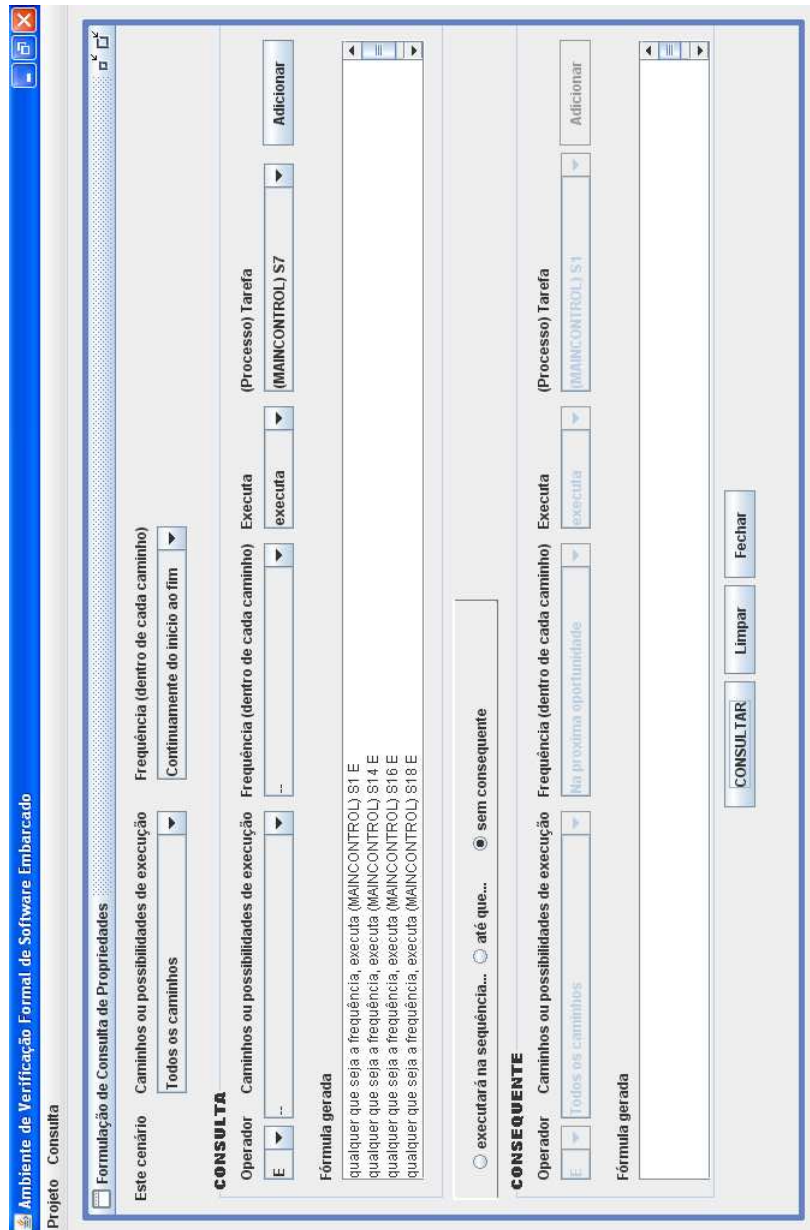


Figura 5.14: Terceira Consulta: AG (S1 & S14 & S16 & S18 & S7)


```
S19 = 0
S20 = 0
S11 = 0
S3 = 0
S22 = 0
S10 = 0
S4 = 0
S25 = 0
S26 = 0
S8 = 0
S5 = 0
S28 = 0
S9 = 0
S6 = 0
S23 = 0
S12 = 0
```

```
resources used:
processor time: 0 s,
BDD nodes allocated: 3057
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é FALSO como mostrado no *log* do SMV. Isso mostra que não é verdade que em todos os momentos tal processo terá todas suas tarefas executando em paralelo.

5.5.4 Consulta IV

Em todos os caminhos ou possibilidades de execução, o seguinte cenário ocorre: a partir do estado em que S2 tenha executado, em todos os próximos caminhos ou possibilidades de execução, eventualmente executará S19?

A Figura 5.15 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: AG (S2 -> AF S19).

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification AG (S2 -> AF S19) is true
```

```
resources used:
processor time: 0 s,
BDD nodes allocated: 2929
```

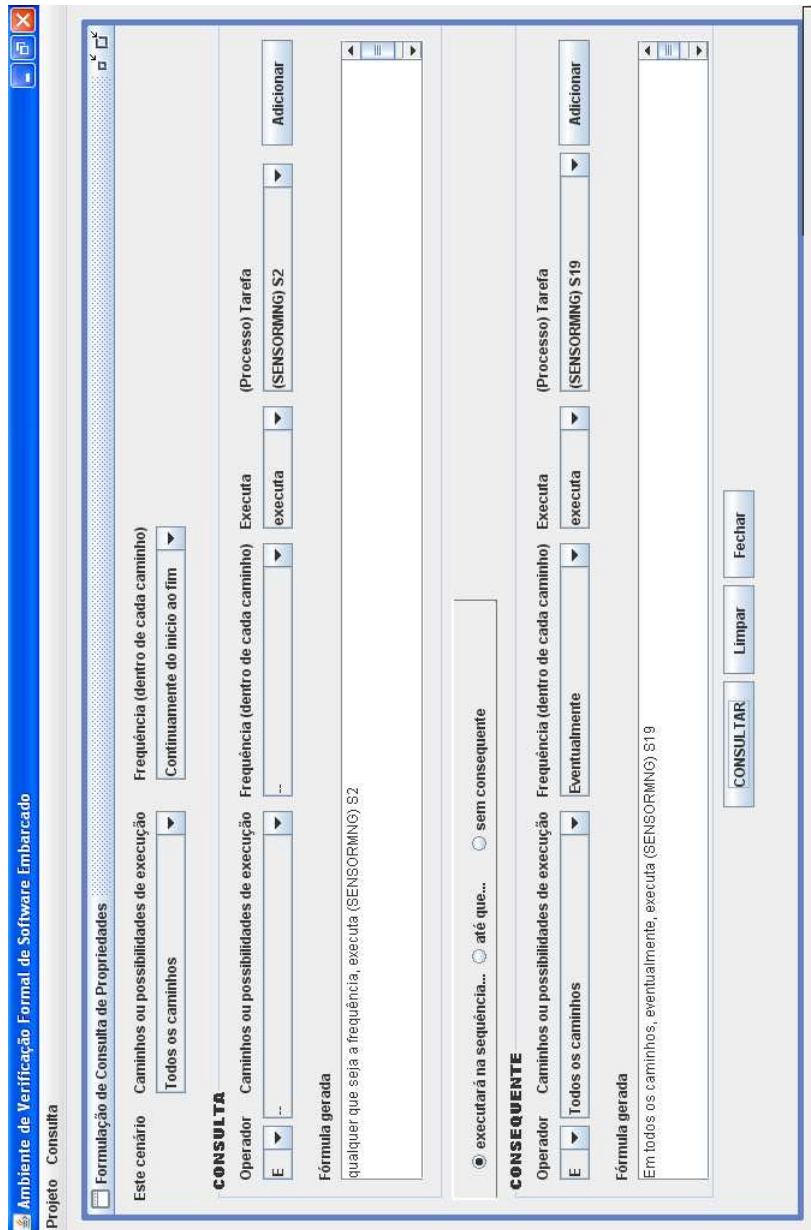


Figura 5.15: Quarta Consulta: AG (S2 -> AF S19)

Bytes allocated: 1045064

BDD nodes representing transition relation: 295 + 1

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV. Isso mostra que, após executar a primeira tarefa do processo SENSORMNG, no caso S2, a tarefa na sequência, no caso S19 será executada também.

5.5.5 Consulta V

Haverá em algum dos caminhos ou possibilidades de execução, eventualmente, um momento em que as primeiras tarefas de cada um dos processos SENSORMNG, ACTUATORMNG e SIGNAL, respectivamente S2, S4 e S6 serão executadas ao mesmo tempo?

A Figura 5.16 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: EF (S2 & S4 & S6).

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification EF (S2 & S4 & S6) is true
```

```
resources used:
```

```
processor time: 0.015 s,
```

```
BDD nodes allocated: 2996
```

```
Bytes allocated: 1045064
```

```
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV, mostrando que, sim, é possível em que algum caminho, eventualmente, as três tarefas estejam executando paralelamente.

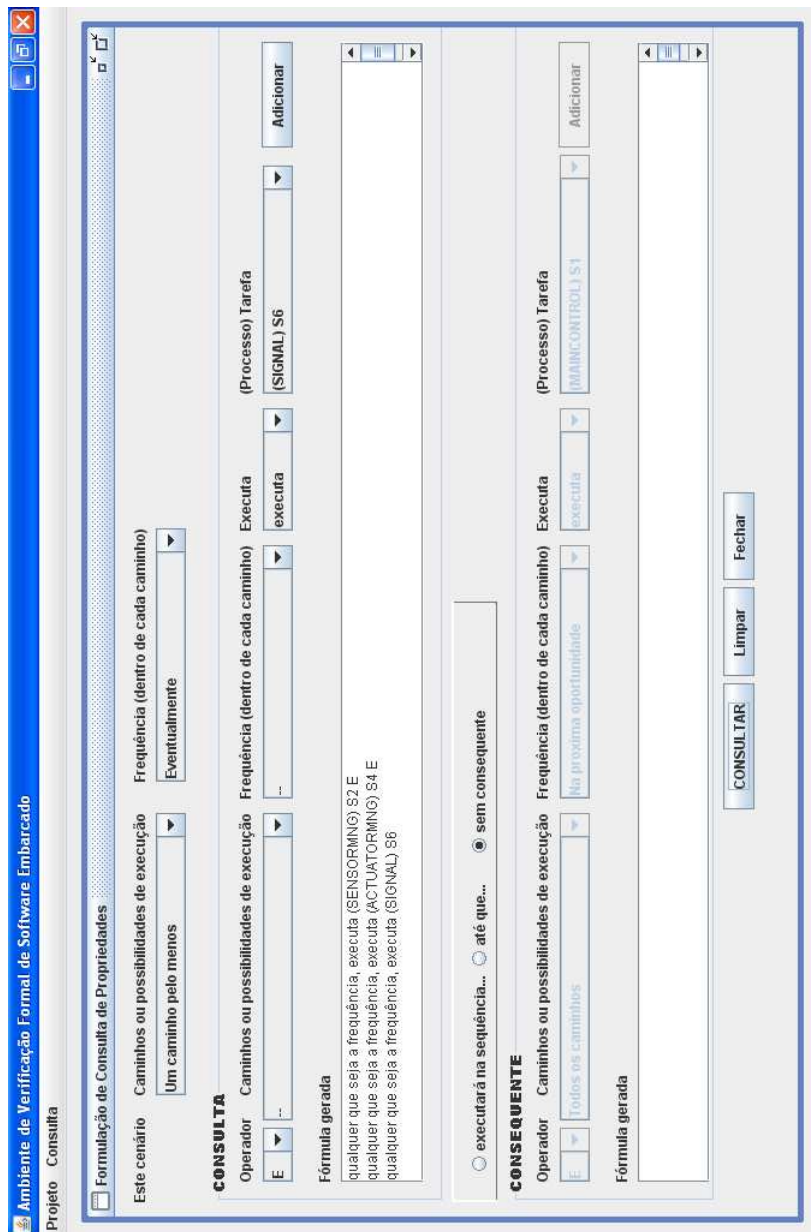


Figura 5.16: Quinta Consulta: EF (S2 & S4 & S6)

5.5.6 Consulta VI

Dado que as tarefas S6 (processo SIGNAL) e S16 (processo MAINCONTROL) são executadas, em todos os caminhos ou possibilidades de execução, eventualmente no futuro, S23 (também do processo SIGNAL) será executado?

A Figura 5.17 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $AG (S16 \ \& \ S6 \ \rightarrow \ AF \ S23)$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification AG (S16 & S6 -> AF S23) is true
```

```
resources used:
```

```
processor time: 0 s,
```

```
BDD nodes allocated: 2951
```

```
Bytes allocated: 1045064
```

```
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV, mostrando que esta propriedade demonstra a correta comunicação através do barramento entre o processo MAINCONTROL e SIGNAL.

5.5.7 Consulta VII

Considerando todos os caminhos ou possibilidades de execução, dado que a tarefa S1 executa, haverá algum momento eventualmente em todos os caminhos a partir deste momento em que a tarefa S22 não será executada?

A Figura 5.18 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $AG (S1 \ \rightarrow \ AG \ (!S22))$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification AG (S1 -> AG (!S22)) is false
-- as demonstrated by the following execution sequence
state 1.1:
```

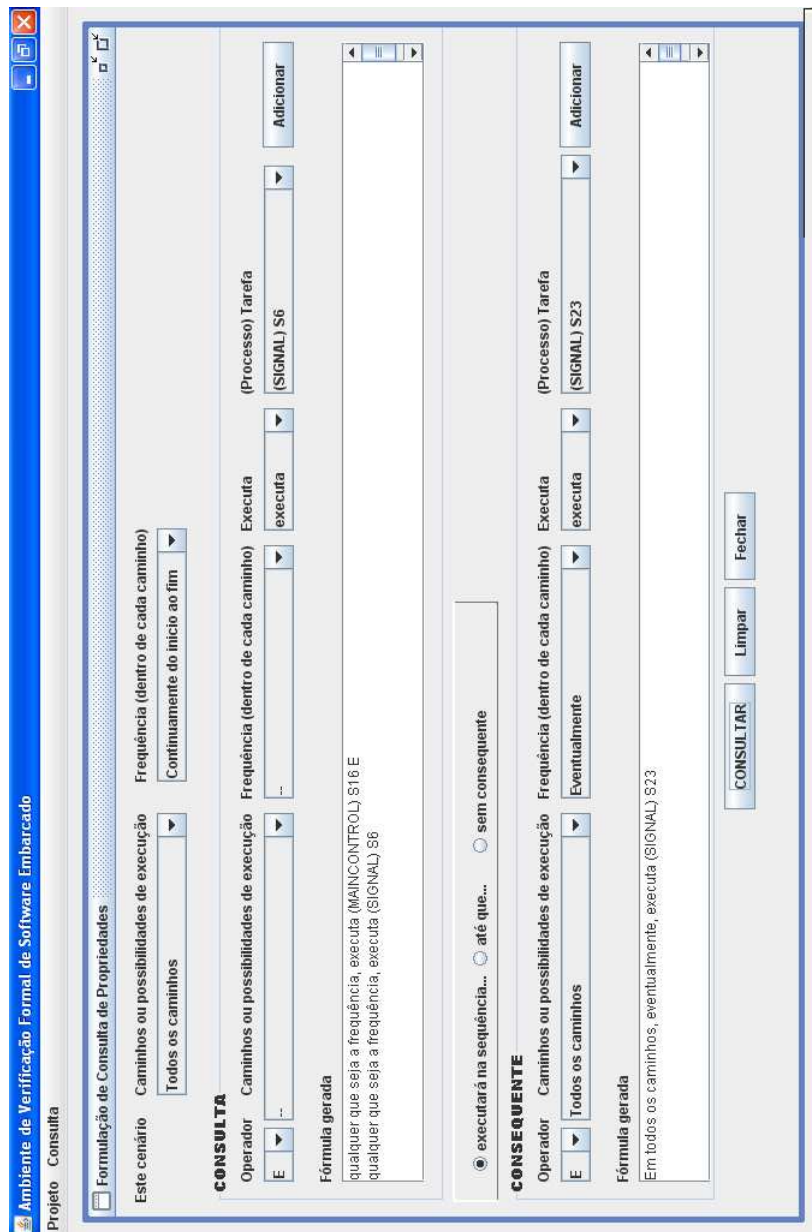


Figura 5.17: Sexta Consulta: AG (S16 & S6 -> AF S23)

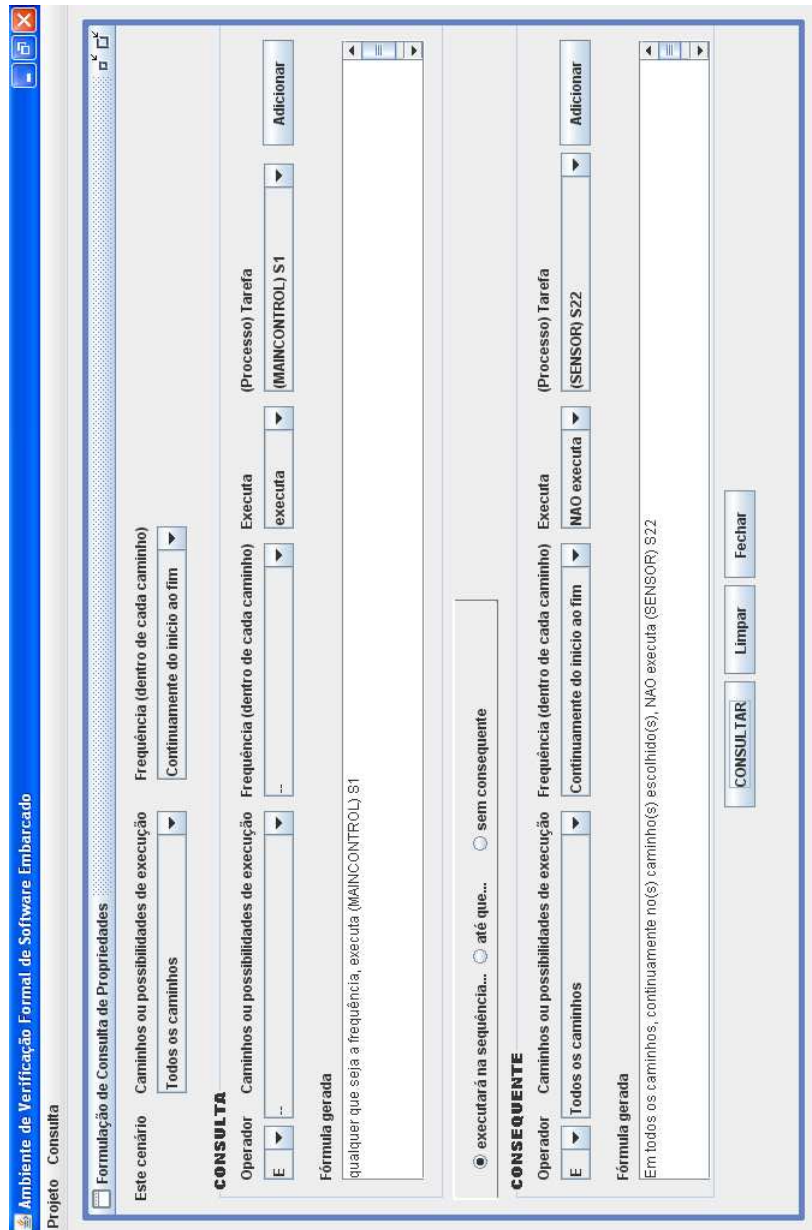


Figura 5.18: Sétima Consulta: AG (S1 -> AG (!S22))

```

lugar_Start = 1
S1 = 0
S14 = 0
S16 = 0
S18 = 0
S7 = 0
S2 = 0
S19 = 0
S20 = 0
S11 = 0
S3 = 0
S22 = 0
S10 = 0
S4 = 0
S25 = 0
S26 = 0
S8 = 0
S5 = 0
S28 = 0
S9 = 0
S6 = 0
S23 = 0
S12 = 0

state 1.2:
[executing process proc51]

state 1.3:
S1 = 1
S14 = 1
S16 = 1
S18 = 1
S7 = 1

state 1.4:
[executing process proc53]

state 1.5:
S3 = 1
S22 = 1
S10 = 1

resources used:
processor time: 0.016 s,
BDD nodes allocated: 3754
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1

```

Análise

O resultado é FALSO como mostrado no *log* do SMV, mostrando que, após a execução da primeira tarefa de MAINCONTROL, no caso, S1, a tarefa S22 pertencente ao processo SENSOR acaba sendo executada invariavelmente. E também que a comunicação entre o processo principal e o de sensor está garantida.

5.5.8 Consulta VIII

Considerando todos os caminhos ou possibilidades de execução, a partir do momento em que a tarefa S1 executa, haverá algum momento em que a tarefa S22 eventualmente será executada, considerando-se pelo menos um caminho ou possibilidade de execução do ponto em que S1 executa pra frente ?

A Figura 5.19 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $AG (S1 \rightarrow EF S22)$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification AG (S1 -> EF S22) is true

resources used:
processor time: 0 s,
BDD nodes allocated: 2927
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV, trazendo o mesmo conhecimento da fórmula anterior, mas expressado de outra maneira, em termos de que pode existir um caminho no futuro em que eventualmente S22 é executado.

5.5.9 Consulta IX

Considerando todos os caminhos ou possibilidades de execução, a partir do momento em que a tarefa S22 executa, haverá algum momento em que a tarefa S25 eventualmente será executada, considerando-se pelo menos um caminho ou possibilidade de execução do ponto em que S22 executa para frente ?

A Figura 5.20 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $AG (S22 \rightarrow EF S25)$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification AG (S22 -> EF S25) is true

resources used:
processor time: 0 s,
BDD nodes allocated: 2927
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1
```

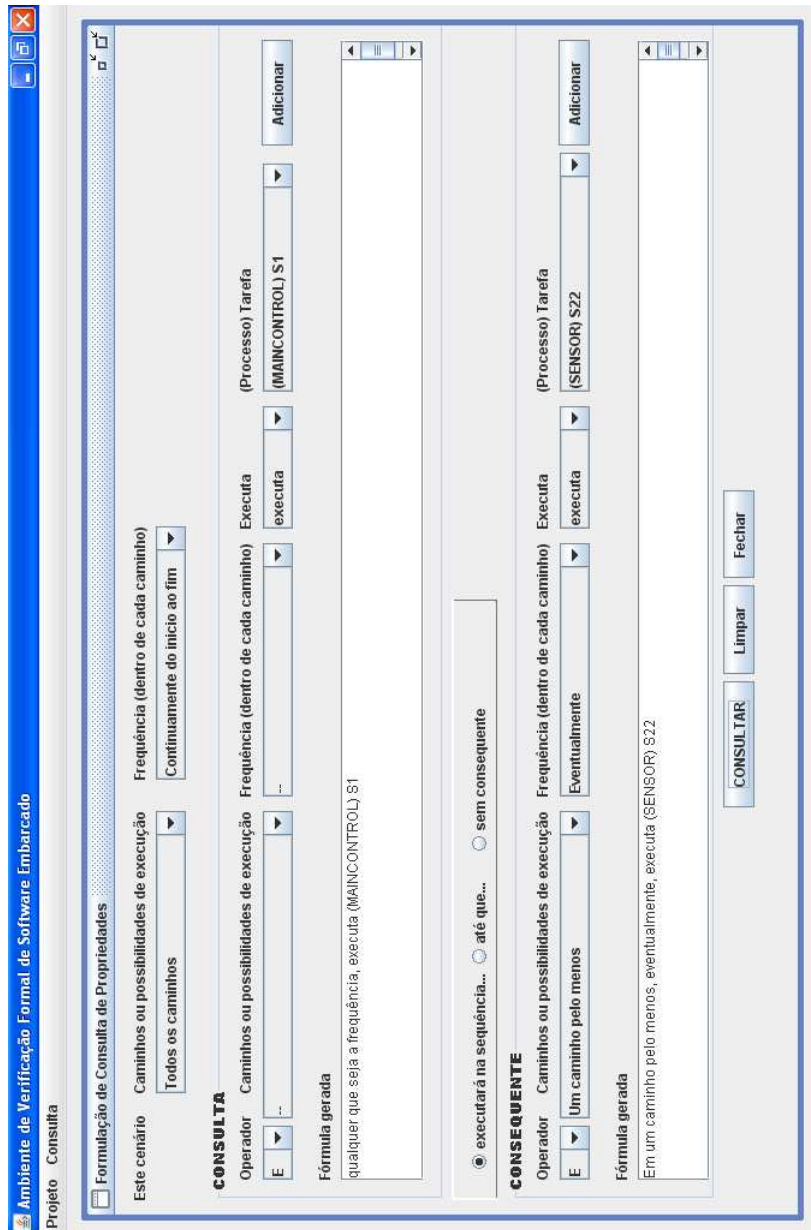


Figura 5.19: Oitava Consulta: AG (S1 -> EF S22)

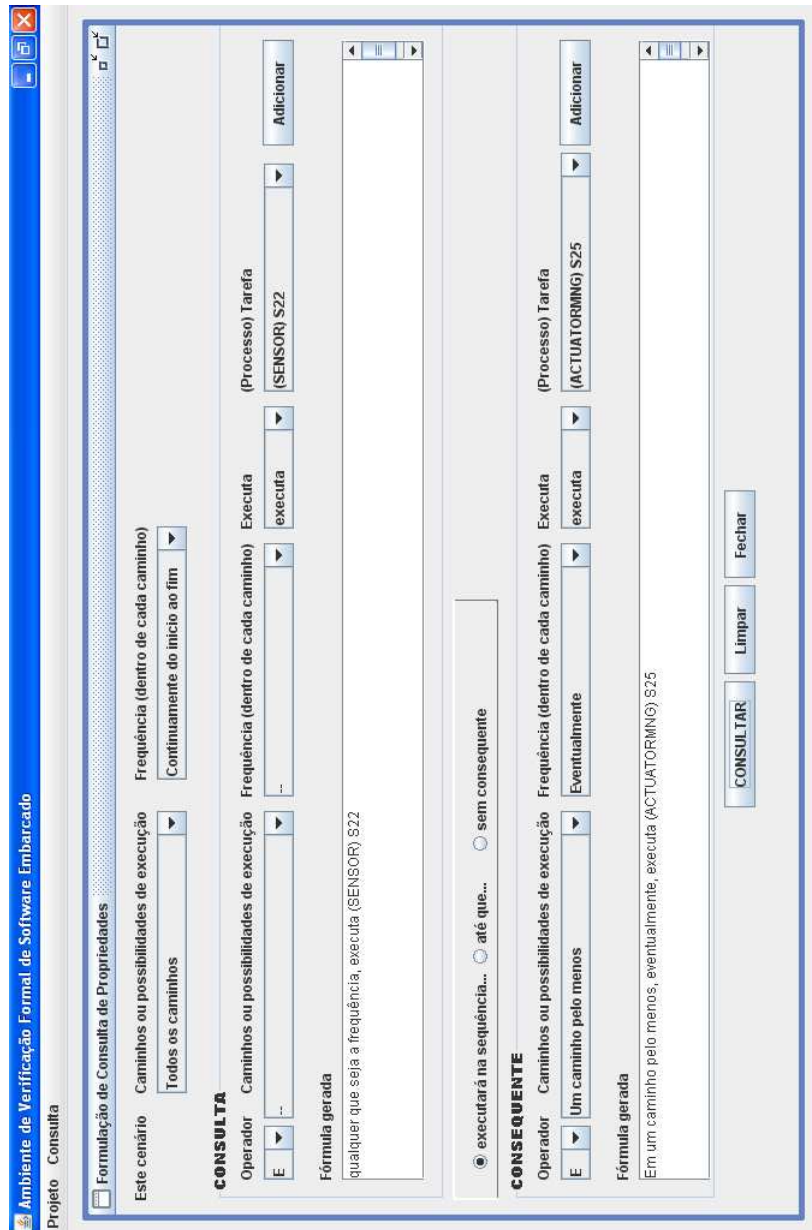


Figura 5.20: Nona Consulta: AG (S22 -> EF S25)

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV. Está garantido que, ao executar uma das tarefas do processo SENSOR, uma tarefa do processo ACTUATORMNG será executada eventualmente em alguma possibilidade de execução.

5.5.10 Consulta X

A consulta atual e as próximas são do tipo: pUq , ou "P se verificará até que Q seja verdade". A utilidade prática desse tipo de consulta de propriedade é verificar se o ordenamento e sequenciamento estabelecido entre as tarefas é de fato respeitado nos vários caminhos ou possibilidades de execução.

Considerando todos os caminhos ou possibilidades de execução, S1 é executado até que seja verdade que S14 seja continuamente do início ao fim executado em todos os caminhos?

A Figura 5.21 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $A(S1 \text{ U } AG \text{ S14})$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification A(S1 U AG S14) is false
-- as demonstrated by the following execution sequence
state 1.1:
lugar_Start = 1
S1 = 0
S14 = 0
S16 = 0
S18 = 0
S7 = 0
S2 = 0
S19 = 0
S20 = 0
S11 = 0
S3 = 0
S22 = 0
S10 = 0
S4 = 0
S25 = 0
S26 = 0
S8 = 0
S5 = 0
S28 = 0
S9 = 0
S6 = 0
S23 = 0
S12 = 0
```

```
resources used:
```

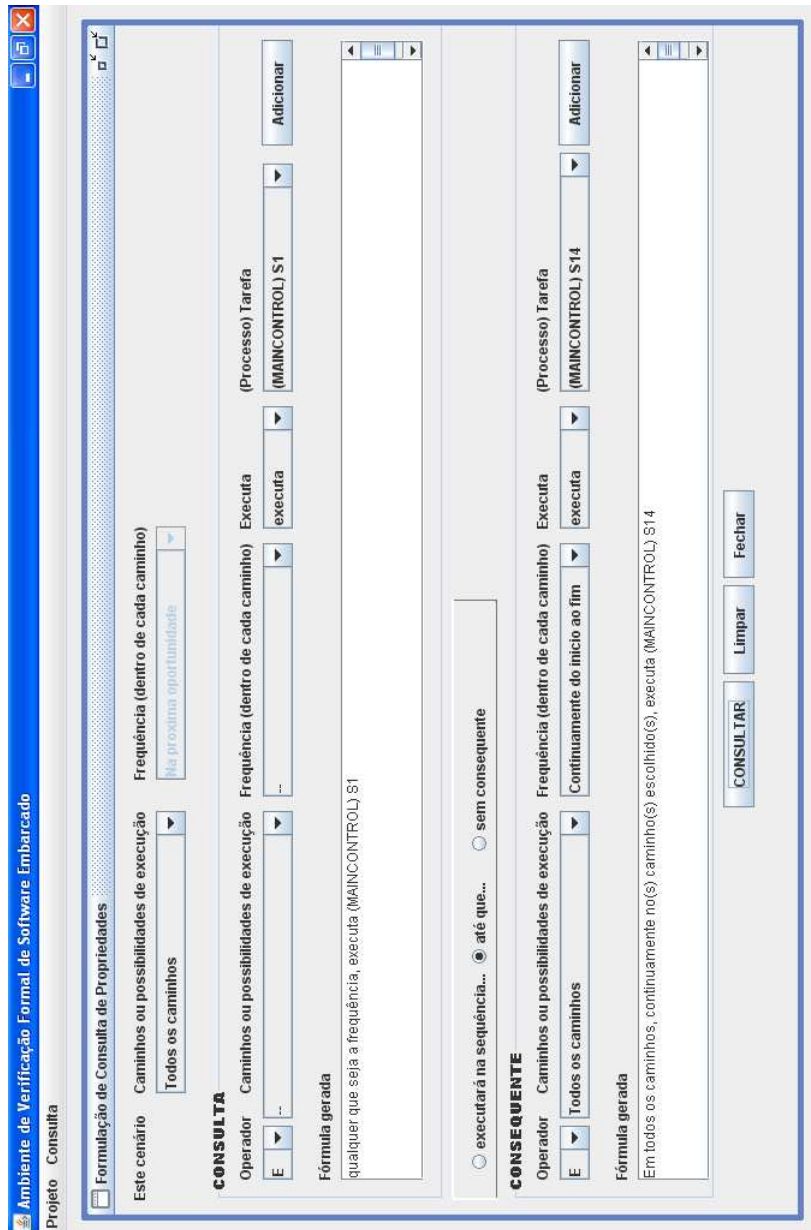


Figura 5.21: Décima Consulta: A(S1 U AG S14)

```
processor time: 0.047 s,  
BDD nodes allocated: 3063  
Bytes allocated: 1045064  
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é FALSO como mostrado no *log* do SMV. Pelo contra-exemplo, não é possível que S14 seja verdadeiro em todos os caminhos continuamente do início ao fim se logo de início é falso.

5.5.11 Consulta XI

Considerando todos os caminhos ou possibilidades de execução, S1 é executado até que seja verdade que S14 seja eventualmente executado em pelo menos um caminho?

A Figura 5.22 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $A(S1 \ U \ EF \ S14)$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification A(S1 U EF S14) is true  
  
resources used:  
processor time: 0.015 s,  
BDD nodes allocated: 2927  
Bytes allocated: 1045064  
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV. Isso demonstra que é possível que S1 seja executado até que S14 seja executado em pelo menos um dos caminhos eventualmente.

5.5.12 Consulta XII

Considerando todos os caminhos ou possibilidades de execução, S1 é executado até que seja verdade que S14 **não** seja eventualmente executado em pelo menos um caminho?

A Figura 5.23 mostra a captura de tela da consulta.

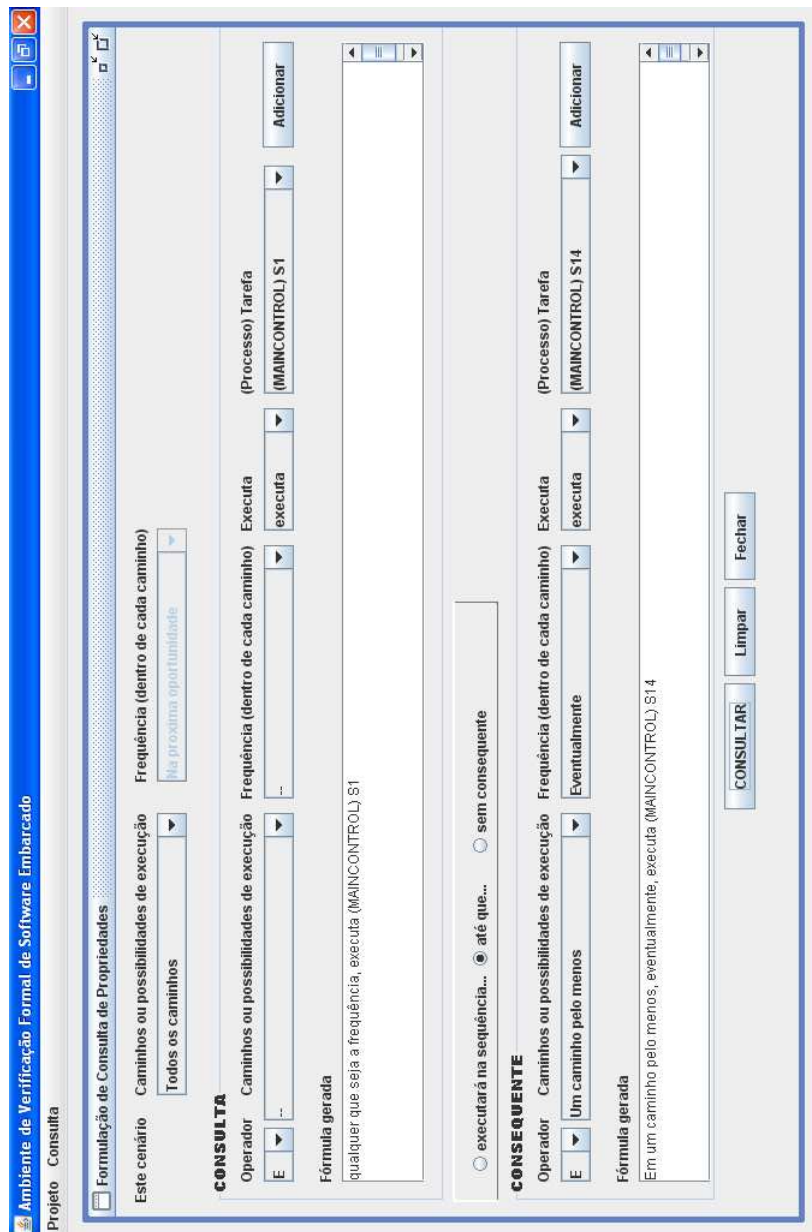


Figura 5.22: Décima Primeira Consulta: A(S1 U EF S14)

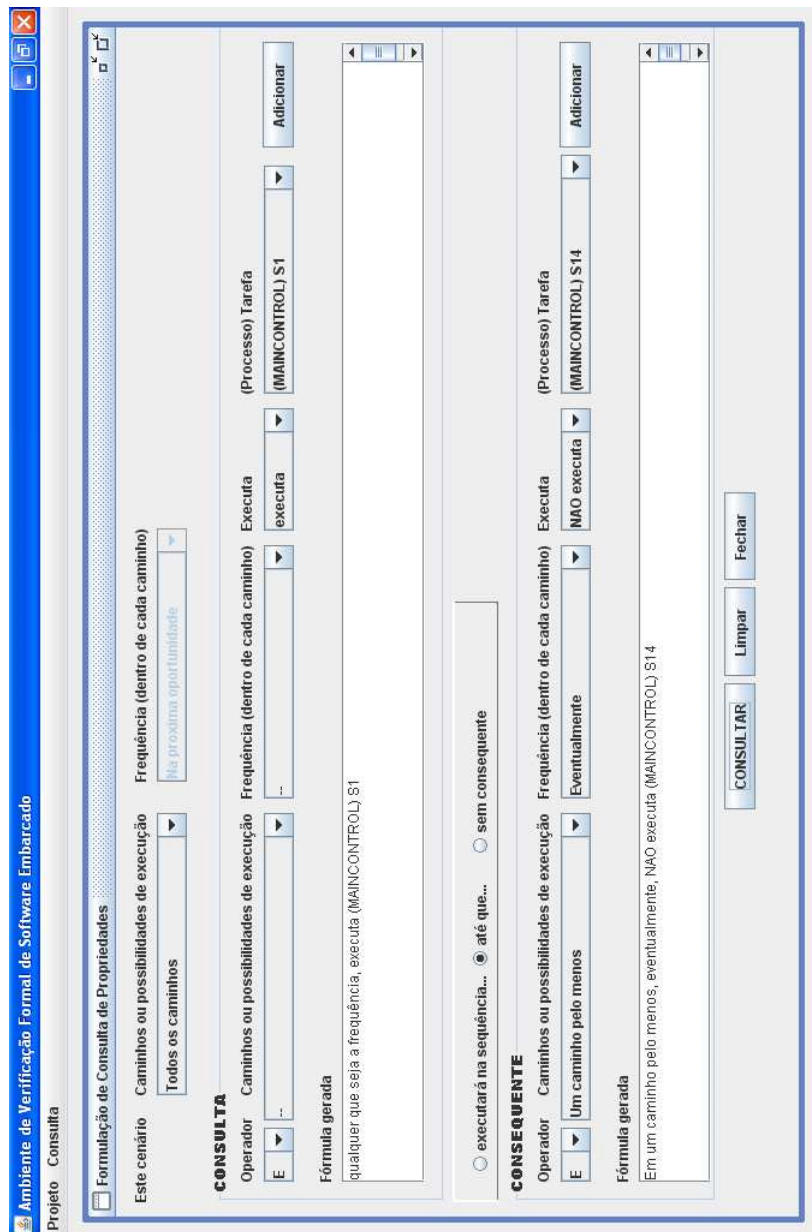


Figura 5.23: Décima Segunda Consulta: A(S1 U EF (!S14))

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $A(S1 \text{ U } EF (!S14))$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification A(S1 U EF (!S14)) is true

resources used:
processor time: 0 s,
BDD nodes allocated: 2929
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV. Isso demonstra que é possível que S1 seja executado até que S14 **não** seja executado em pelo menos um dos caminhos eventualmente.

5.5.13 Consulta XIII

Considerando todos os caminhos ou possibilidades de execução, S1 é executado até que seja verdade que S14 seja executado exatamente no próximo instante em pelo menos um caminho?

A Figura 5.24 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $A [(S1) \text{ U } (EX S14)]$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification A(S1 U EX S14) is true

resources used:
processor time: 0.016 s,
BDD nodes allocated: 2921
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é VERDADEIRO como mostrado no *log* do SMV. A tarefa S14 não só será executada após a tarefa S1 - assim como comprovado em consultas anteriores - como S14 tem a possibilidade de ser executada até mesmo *imediatamente após* S1, visto que há pelo menos um caminho ou possibilidade de execução em que isso ocorre.

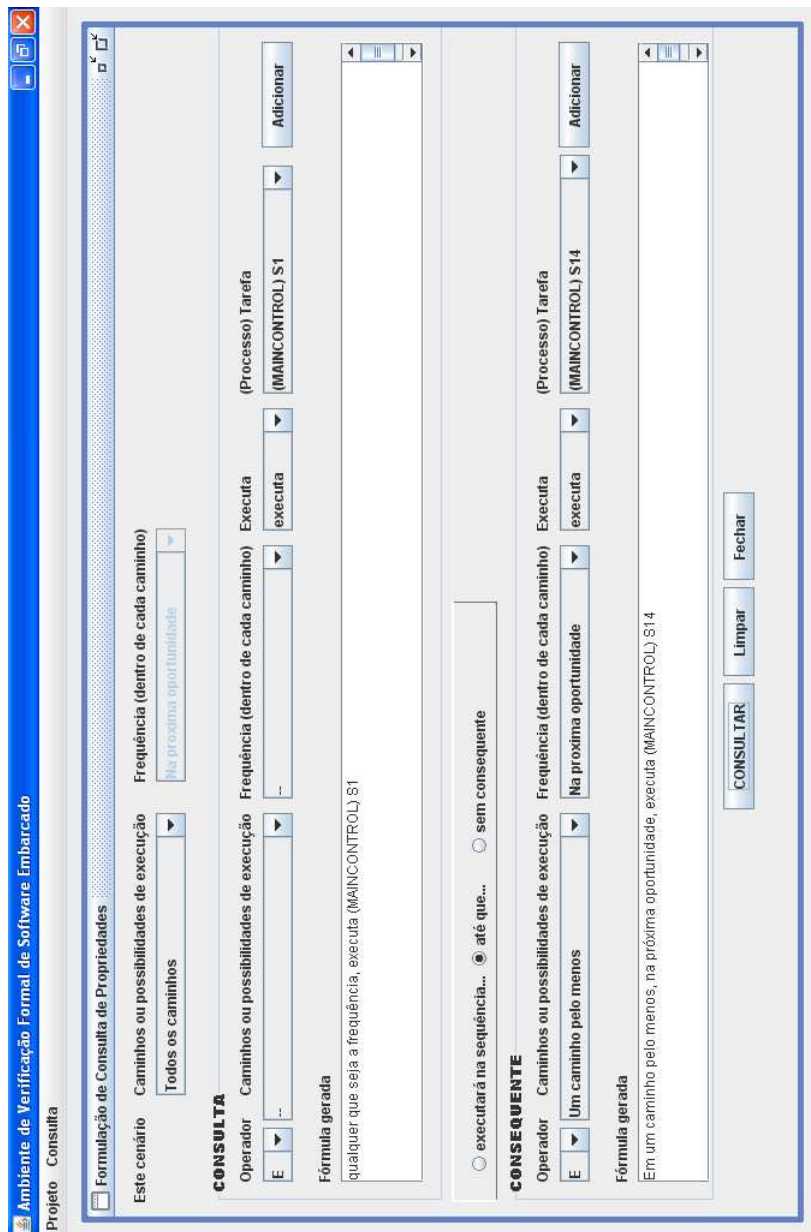


Figura 5.24: Décima Terceira Consulta: A(S1 U EX S14)

5.5.14 Consulta XIV

Considerando todos os caminhos ou possibilidades de execução, S1 é executado até que seja verdade que S14 seja executado exatamente no próximo instante em todos os caminhos?

A Figura 5.25 mostra a captura de tela da consulta.

Após pressionar-se o botão CONSULTAR, tem-se a seguinte fórmula CTL traduzida: $A [(S1) U (AX S14)]$.

Eis o *log* da execução no SMV, juntamente com os recursos alocados:

```
-- specification A(S1 U AX S14) is false
-- as demonstrated by the following execution sequence
-- loop starts here --
state 1.1:
lugar_Start = 1
S1 = 0
S14 = 0
S16 = 0
S18 = 0
S7 = 0
S2 = 0
S19 = 0
S20 = 0
S11 = 0
S3 = 0
S22 = 0
S10 = 0
S4 = 0
S25 = 0
S26 = 0
S8 = 0
S5 = 0
S28 = 0
S9 = 0
S6 = 0
S23 = 0
S12 = 0

state 1.2:

resources used:
processor time: 0.016 s,
BDD nodes allocated: 3066
Bytes allocated: 1045064
BDD nodes representing transition relation: 295 + 1
```

Análise

O resultado é FALSO como mostrado no *log* do SMV pois não é verdade que a tarefa S14 será executada *imediatamente após* a tarefa S1 em todos os caminhos ou possibilidades de execução.

O próximo Capítulo tece os últimos comentários, e ressalta os ganhos obtidos pelo presente trabalho bem como melhorias para trabalhos futuros.

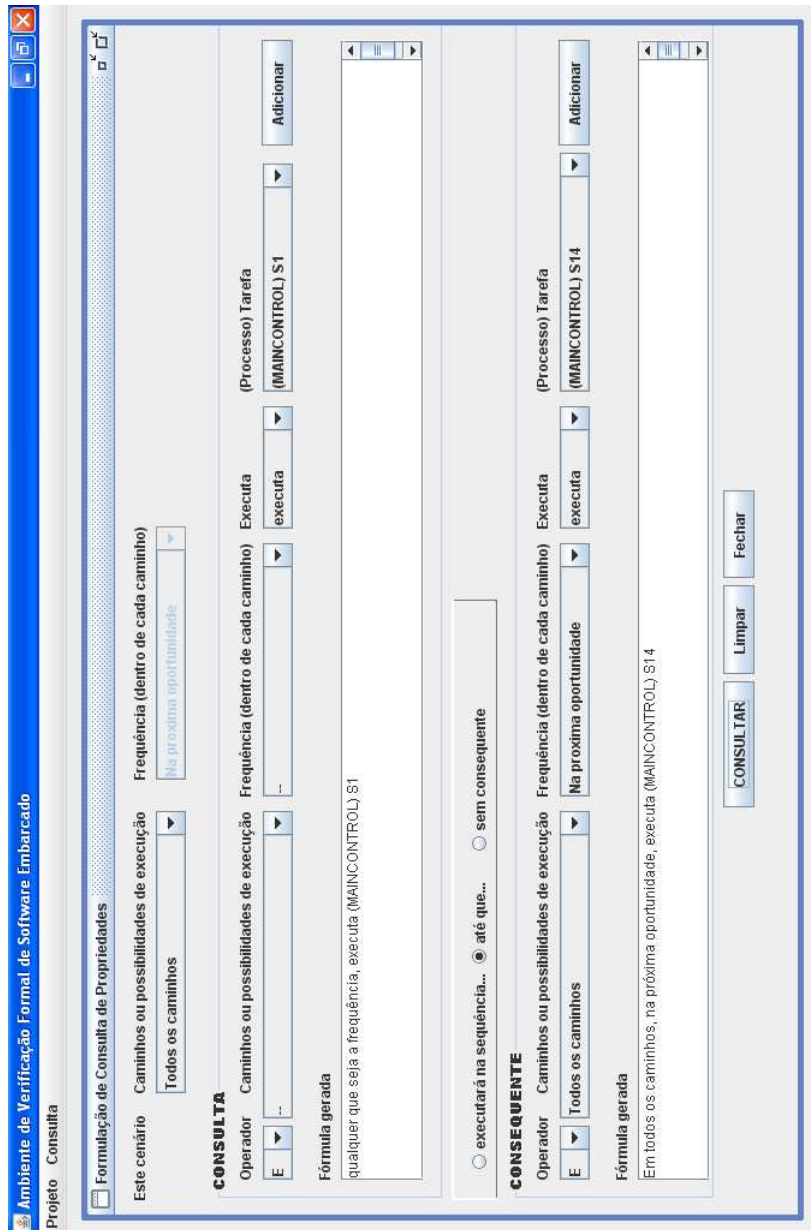


Figura 5.25: Décima Quarta Consulta: A(S1 U AX S14)

Capítulo 6

Considerações Finais

6.1 Conclusões

No universo dos softwares embarcados, o processo de desenvolvimento se torna mais árduo do que para sistemas desktop, devido a restrições de consumo de energia, peso e tamanho, tempo etc, conforme apresentado na Seção 3.1.2. Além disso, como também foi exposto, muitos desses sistemas, em caso de falhas em seu funcionamento, podem acarretar muitos danos à vida humana ou prejuízos financeiros. Portanto, fazem-se necessários métodos formais para a precisa verificação de tais sistemas logo nas primeiras fases do desenvolvimento, ainda em tempo de projeto e a conseqüente constatação de que foram corretamente projetados.

Todavia, fazem-se necessários também métodos ágeis no sentido de serem intuitivos e de fácil uso por parte do desenvolvedor principalmente por causa da crescente pressão de Mercado por novos produtos rapidamente, ou *time-to-Market*. Atendendo a tais demandas, o presente trabalho apresentou um método intuitivo de verificação formal de software embarcado, exposto no Capítulo 4. Baseando-se em tarefas e processos escritos previamente para um ambiente de abordagem de escalonamento de executivo cíclico, o método apresentado toma como entrada uma especificação destas tarefas e seus processos na notação do Diagrama de Sequência da UML, apresentado na seção 3.2.3. A grande vantagem dessa notação é a de livrar o projetista da necessidade de conhecer formalismos. Por isso ela é intuitiva e, como conseqüência, largamente utilizada pela Indústria. Para assegurar a tão esperada correteude, o presente método apresentou regras de tradução do diagrama de

seqüência para a notação Rede de Petri Lugar/Transição ou, simplesmente, Redes de Petri, apresentada na Seção 3.3. A notação Redes de Petri é suficiente formal e matematicamente rigorosa para qualquer tipo de verificação de sistemas críticos, sejam eles distribuídos ou não. Em particular, o presente trabalho adota a técnica de Verificação de Modelos ou *Model Checking*. Como exposto na seção 3.4.4, esta técnica apresenta algumas limitações no seu uso prático. O verificador de modelos ou *model checker* utilizado foi o SMV.

Por fim, não adiantaria muita coisa ter a especificação em alto nível se as consultas a propriedades ainda fossem escritas numa notação formal. Por isso, visando também agilizar o desenvolvimento, o presente método trouxe técnicas para traduzir as consultas em linguagem natural para a notação *Computational Tree Logic* (CTL), própria para verificação em *model checkers*, tais como o MCK ou o próprio SMV. Para a realização das consultas, é necessário obter informações da infra-estrutura do sistema e arquitetura da aplicação, além do Diagrama de Sequência em si. Estas informações dizem respeito aos processadores e seus processos, bem como os processos e suas tarefas.

Todos os passos do método são realizados por um ambiente de software, o Ambiente de Verificação Formal de Software Embarcado. O funcionamento da ferramenta e seu projeto foram detalhados na Seção 4.7. O experimento apresentado no Capítulo 5 valida as técnicas descritas neste trabalho.

Vários trabalhos relacionados têm sido desenvolvidos, como mostrou o Capítulo 2. Muitos deles usavam a notação de *Statechart*, Diagrama de Atividades ou mesmo o Diagrama de Sequência da UML como notação de modelagem e, então, realizam algum tipo de tradução para algum tipo de formalismo, principalmente Redes de Petri. Alguns deles utilizam a rede gerada final somente no contexto de Avaliação de Desempenho. O presente trabalho visa, como os outros, acrescentar ao estado da arte da verificação formal, em particular de software embarcado, de maneira que mais e mais práticas formais sejam usadas na Indústria.

6.2 Contribuições

A seguir, um sumário das principais contribuições obtidas pelo presente trabalho e pelo Ambiente que o realiza, cujos conceitos podem potencialmente ser usados para todo e qualquer tipo de software, embarcado ou não.

- Tradução automática de Diagrama de Sequência para um modelo formal, no caso Redes de Petri.

- Verificação desse modelo via *model checking* simbólico, através da ferramenta SMV.
- Escrita da Rede de Petri final nos formatos APNN, PNML e SMV.
- Interface amigável pra coleta de consultas em linguagem natural.
- Tradução automática dessas consultas para CTL.
- Todos os conceitos aplicados a software embarcado.
- Experimento com software embarcado distribuído.
- Ferramenta para realizar o método.

6.3 Trabalhos Futuros

Este trabalho pode ser estendido para contemplar futuramente alguns novos pontos, dentre eles:

1. Adicionar suporte à tempo-real e ao perfil UML SPT.
2. Considerar outros diagramas da UML a fim de trazer uma modelagem mais abrangente permitindo especificar o sistema sob outras visões.
3. Realizar experimentos usando processadores reais, como por exemplo o 8051 ou Pic.
4. Investigar o potencial de outras lógicas, como LTL para a construção de consultas a propriedades.
5. Investigar o potencial de variantes de Redes de Petri, tais como as Coloridas e Orientadas a Objeto.
6. Ferramenta CASE para gerar Diagrama de Sequência.
7. Inclusão do operador *weak until* (W).

Referências Bibliográficas

- [1] Leonardo Amorim, Paulo Maciel, Meuse Nogueira, Raimundo Barreto, and Eduardo Tavares. Mapping live sequence chart to coloured petri nets for analysis and verification of embedded systems. *SIGSOFT Softw. Eng. Notes*, 31(3):1–25, 2006.
- [2] Donatelli S. Bernardi S. and Merseguer J. From uml sequence diagrams and statecharts to analysable petri net models. In *Proceedings of the 3rd international Workshop on Software and Performance*, 2002.
- [3] F. Bonnefoi, L.M. Hillah, F. Kordon, and X. Renault. Design, modeling and analysis of its using uml and petri nets. *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 314–319, 30 2007-Oct. 3 2007.
- [4] Grady Booch. *Object-Oriented Analysis and Design with Applications, 1st ed.* Benjamin/Cummings, 1991.
- [5] W. Brauer. *Net Theory and Applications. Lecture Notes in Computer Science.* Springer-Verlag, 1979.
- [6] Peter Coad and Edward Yourdon. *Object-Oriented Analysis, 2nd ed.* Yourdon Press, 1991.
- [7] J. E. Cooling. *Software Engineering for Real-Time Systems.* Addison-Wesley, 2003.
- [8] Brad J. Cox. *Object-Oriented Programming: An Evolutionary Approach.* Addison-Wesley, 1986.
- [9] J. A. M. de Queiroz and P. R. F. Cunha. Sistemas distribuídos: de especificação lotos a implementações. In *IX Escola de Computação*, 1994.

- [10] Stephanie Bodoff Chris Dollin Helena Gilchrist Fiona Hayes Derek Coleman, Patrick Arnold and Paul Jeremaes. *Object-Oriented Development: The Fusion Method*. Prentice Hall, 1994.
- [11] M. DiNatale and J. A. Stankovic. Dynamic end-to-end guarantees in distributed realtime systems. In *Proceedings of the IEEE Real-Time Systems Symposium*, pages 216–227, 1994.
- [12] S. J. Chang F. Dicesare and G. Goldbogen. Failure propagation trees for diagnosis in manufacturing systems. In *IEEE Trans. on Systems, Manuf. and Cybernetics*, 1991.
- [13] Joao M. Fernandes, Simon Tjell, Jens Baek Jorgensen, and Oscar Ribeiro. Designing tool support for translating use cases and uml 2.0 sequence diagrams into a coloured petri net. In *SCESM '07: Proceedings of the Sixth International Workshop on Scenarios and State Machines*, 2007.
- [14] Adele Goldberg and David Robson. *Smalltalk-80: The Language and Its Implementation*. Addison-Wesley, 1983.
- [15] Object Management Group. Unified modeling language specification. 1998.
- [16] The Precise UML Group. puml specification. *The Precise UML Group*, 2008.
- [17] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall International, 1985.
- [18] Open Systems Interconnection. Lotos - a formal description technique based on temporal ordering of observational behaviour. Technical report, Open Systems Interconnection, 1987.
- [19] Patrik Jonsson Ivar Jacobson, Magnus Christerson and Gunnar Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1992.
- [20] William Premerlani Frederick Eddy James Rumbaugh, Michael Blaha and William Lorensen. *Object-Oriented Modeling and Design*. Prentice Hall, 1991.
- [21] MuDer Jeng and WeiZhao Lu. Extension of uml and its conversion to petri nets for semiconductor manufacturing modeling. In *IEEE International Conference on Robotics and Automation*, 2002.

- [22] Joost-Pieter Katoen. *Concepts, Algorithms and Tools for Model Checking*. 1999.
- [23] J. A. McDermid L. M. Barroca. Formal methods: Use and relevance for the development of safety-critical systems. *The Computer Journal*, 1992.
- [24] Anikó Costa Luís Gomes and Paulo Meira. From use cases to building monitoring systems through petri nets. In *IEEE ISIE*, 2005.
- [25] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publ., 1993.
- [26] Bertrand Meyer. *Object-Oriented Software Construction*. Prentice Hall, 1988.
- [27] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [28] Falko Bause Peter Kemper and Pieter Kritzing. Abstract petri net notation. 2000.
- [29] Eshuis R. Symbolic model checking of uml activity diagrams. *ACM Trans. Softw. Eng. Methodol*, 2006.
- [30] Brian Wilkerson Rebecca Wirfs-Brock and Lauren Wiener. *Designing Object-Oriented Software*. Prentice Hall, 1990.
- [31] A. Sangiovanni-Vincentelli and G. Martin. Platform-based design and software design methodology for embedded systems. In *IEEE Design and Test of Computers*, pages 23–33, 2001.
- [32] A. Sangiovanni-Vincentelli and G. Martin. A vision for embedded software. In *Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES'01)*, pages 1–7, 2001.
- [33] Sally Shlaer and Stephen J. Mellor. *Object-Oriented Systems Analysis: Modeling the World in Data*. Yourdon Press, 1988.
- [34] J. M. Spivey. *Z Notation - A Reference Manual*. Prentice Hall International, 1985.
- [35] Paul Ward and Stephen J. Mellor. *Structured Development for Real-Time Systems: Introduction and Tools*. Yourdon Press, 1985.

- [36] M. Weber and E. Kindler. The petri net markup language. In *Petri Net Technology for Communication Based Systems, LNCS 2472*, 2003.
- [37] S. Yao and S. M Shatz. Consistency checking of uml dynamic models based on petri net techniques. In *Proceedings of the 15th international Conference on Computing*, 2006.
- [38] Edward Yourdon and Larry L. Constantine. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Yourdon Press, 1979.

Apêndice A

Arquivo XML de Entrada da Ferramenta com o Diagrama de Sequência do Experimento

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<diagramadesequencia>
  <aplicacao>Simple Control App.</aplicacao>
  <mensagem>
    <de>S1</de>
    <de>S2</de>
    <de>S3</de>
    <de>S4</de>
    <de>S5</de>
    <de>S6</de>
    <de>S7</de>
    <de>S8</de>
    <de>S9</de>
    <de>S10</de>
    <de>S11</de>
    <de>S12</de>
    <de>S14</de>
    <de>S16</de>
    <de>S18</de>
    <de>S20</de>
    <de>S22</de>
    <de>S23</de>
    <de>S25</de>
    <de>S26</de>
    <de>S28</de>
  </mensagem>
</diagramadesequencia>
```

<para>S14</para>
</mensagem>

<mensagem>
<de>S14</de>
<para>S16</para>
</mensagem>

<mensagem>
<de>S16</de>
<para>S18</para>
</mensagem>

<mensagem>
<de>S18</de>
<para>S7</para>
</mensagem>

<mensagem>
<de>S1</de>
<para>S19</para>
</mensagem>

<mensagem>
<de>S14</de>
<para>S25</para>
</mensagem>

<mensagem>
<de>S16</de>
<para>S23</para>
</mensagem>

<mensagem>
<de>S2</de>
<para>S19</para>
</mensagem>

<mensagem>

<de>S19</de>
<para>S20</para>
</mensagem>

<mensagem>
<de>S20</de>
<para>S11</para>
</mensagem>

<mensagem>
<de>S20</de>
<para>S14</para>
</mensagem>

<mensagem>
<de>S19</de>
<para>S22</para>
</mensagem>

<mensagem>
<de>S3</de>
<para>S22</para>
</mensagem>

<mensagem>
<de>S22</de>
<para>S10</para>
</mensagem>

<mensagem>
<de>S22</de>
<para>S20</para>
</mensagem>

<mensagem>
<de>S4</de>
<para>S25</para>
</mensagem>

<mensagem>	<para>S9</para>
<de>S25</de>	</mensagem>
<para>S26</para>	
</mensagem>	<mensagem>
	<de>S28</de>
<mensagem>	<para>S26</para>
<de>S26</de>	</mensagem>
<para>S8</para>	
</mensagem>	<mensagem>
	<de>S6</de>
<mensagem>	<para>S23</para>
<de>S25</de>	</mensagem>
<para>S28</para>	
</mensagem>	<mensagem>
	<de>S23</de>
<mensagem>	<para>S12</para>
<de>S26</de>	</mensagem>
<para>S7</para>	
</mensagem>	<mensagem>
	<de>S23</de>
<mensagem>	<para>S18</para>
<de>S5</de>	</mensagem>
<para>S28</para>	
</mensagem>	
	</diagramadesequencia>
<mensagem>	
<de>S28</de>	

Apêndice B

Arquivo XML de Entrada da Ferramenta com a Infra-estrutura da Aplicação do Experimento

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<sisistema>

  <aplicacao>Simple Control Application</aplicacao>

  <!-- PROCESSADORES -->
  <processador>
    <idprocessador>P1</idprocessador>
    <processos>
      <elemento>MAINCONTROL</elemento>
    </processos>
  </processador>

  <processador>
    <idprocessador>P2</idprocessador>
    <processos>
      <elemento>SENSORMNG</elemento>
      <elemento>SENSOR</elemento>
    </processos>
  </processador>

  <processador>
    <idprocessador>P3</idprocessador>
    <processos>
      <elemento>ACTUATORMNG</elemento>
    </processos>
  </processador>

```

```

        <elemento>ACTUATOR</elemento>
    </processos>
</processador>

<processador>
    <idprocessador>P4</idprocessador>
    <processos>
        <elemento>SIGNAL</elemento>
    </processos>
</processador>

<processador-principal>P1</processador-principal>

<!-- PROCESSOS -->
<processo>
    <id>MAINCONTROL</id>
    <tarefas>
        <arefa>S1</arefa>
        <arefa>S14</arefa>
        <arefa>S16</arefa>
        <arefa>S18</arefa>
        <arefa>S7</arefa>
    </tarefas>
</processo>

<processo>
    <id>SENSORMNG</id>
    <tarefas>
        <arefa>S2</arefa>
        <arefa>S19</arefa>
        <arefa>S20</arefa>
        <arefa>S11</arefa>
    </tarefas>
</processo>

<processo>
    <id>SENSOR</id>
    <tarefas>
        <arefa>S3</arefa>
        <arefa>S22</arefa>
        <arefa>S10</arefa>
    </tarefas>
</processo>

<processo>
    <id>ACTUATORMNG</id>
    <tarefas>
        <arefa>S4</arefa>
        <arefa>S25</arefa>
        <arefa>S26</arefa>
        <arefa>S8</arefa>
    </tarefas>
</processo>

<processo>
    <id>ACTUATOR</id>
    <tarefas>
        <arefa>S5</arefa>
        <arefa>S28</arefa>
        <arefa>S9</arefa>
    </tarefas>

```



```
</processo>

<processo>
  <id>SIGNAL</id>
  <tarefas>
    <arefa>S6</arefa>
    <arefa>S23</arefa>
    <arefa>S12</arefa>
  </tarefas>
</processo>

</sistema>
```

Apêndice C

Arquivo APNN Gerado no Experimento

```
\beginnet{rede001}  
  
\place{P74}{\name{lugar_S26}\init{0}\capacity{1}}  
\place{P58}{\name{lugar_S25}\init{0}\capacity{1}}  
\place{P20}{\name{lugar_S23}\init{0}\capacity{1}}  
\place{P42}{\name{lugar_S22}\init{0}\capacity{1}}  
\place{P45}{\name{lugar_S20}\init{0}\capacity{1}}  
\place{P15}{\name{lugar_GB_M17}\init{0}\capacity{1}}  
\place{P10}{\name{lugar_GB_M15}\init{0}\capacity{1}}  
\place{P8}{\name{lugar_GB_M13}\init{0}\capacity{1}}  
\place{P64}{\name{lugar_S28}\init{0}\capacity{1}}  
\place{P7}{\name{lugar_M13}\init{0}\capacity{1}}  
\place{P9}{\name{lugar_M15}\init{0}\capacity{1}}  
\place{P77}{\name{lugar_GB_M27}\init{0}\capacity{1}}  
\place{P14}{\name{lugar_M17}\init{0}\capacity{1}}  
\place{P52}{\name{lugar_GB_M21}\init{0}\capacity{1}}  
\place{P23}{\name{lugar_GB_M24}\init{0}\capacity{1}}  
\place{P51}{\name{lugar_M21}\init{0}\capacity{1}}  
\place{P22}{\name{lugar_M24}\init{0}\capacity{1}}  
\place{P30}{\name{lugar_End}\init{0}\capacity{1}}  
\place{P31}{\name{lugar_GP_S12}\init{0}\capacity{1}}  
\place{P56}{\name{lugar_GP_S11}\init{0}\capacity{1}}  
\place{P50}{\name{lugar_GP_S10}\init{0}\capacity{1}}  
\place{P76}{\name{lugar_M27}\init{0}\capacity{1}}  
\place{P26}{\name{lugar_GP_S18}\init{0}\capacity{1}}  
\place{P37}{\name{lugar_GP_S19}\init{0}\capacity{1}}  
\place{P13}{\name{lugar_GP_S16}\init{0}\capacity{1}}  
\place{P6}{\name{lugar_GP_S14}\init{0}\capacity{1}}  
\place{P54}{\name{lugar_S11}\init{0}\capacity{1}}  
\place{P48}{\name{lugar_S10}\init{0}\capacity{1}}  
\place{P28}{\name{lugar_S12}\init{0}\capacity{1}}  
\place{P5}{\name{lugar_S14}\init{0}\capacity{1}}  
\place{P12}{\name{lugar_S16}\init{0}\capacity{1}}  
\place{P36}{\name{lugar_S19}\init{0}\capacity{1}}
```

```

\place{P25}{\name{lugar_S18}\init{0}\capacity{1}}
\place{P49}{\name{lugarEsp_S10}\init{0}\capacity{1}}
\place{P55}{\name{lugarEsp_S11}\init{0}\capacity{1}}
\place{P29}{\name{lugarEsp_S12}\init{0}\capacity{1}}
\place{P0}{\name{lugar_Start}\init{1}\capacity{1}}
\place{P65}{\name{lugar_GP_S28}\init{0}\capacity{1}}
\place{P75}{\name{lugar_GP_S26}\init{0}\capacity{1}}
\place{P59}{\name{lugar_GP_S25}\init{0}\capacity{1}}
\place{P46}{\name{lugar_GP_S20}\init{0}\capacity{1}}
\place{P21}{\name{lugar_GP_S23}\init{0}\capacity{1}}
\place{P43}{\name{lugar_GP_S22}\init{0}\capacity{1}}
\place{P72}{\name{lugar_GP_S9}\init{0}\capacity{1}}
\place{P18}{\name{lugar_GP_S6}\init{0}\capacity{1}}
\place{P68}{\name{lugar_GP_S5}\init{0}\capacity{1}}
\place{P81}{\name{lugar_GP_S8}\init{0}\capacity{1}}
\place{P85}{\name{lugar_GP_S7}\init{0}\capacity{1}}
\place{P34}{\name{lugar_GP_S2}\init{0}\capacity{1}}
\place{P3}{\name{lugar_GP_S1}\init{0}\capacity{1}}
\place{P62}{\name{lugar_GP_S4}\init{0}\capacity{1}}
\place{P40}{\name{lugar_GP_S3}\init{0}\capacity{1}}
\place{P71}{\name{lugarEsp_S9}\init{0}\capacity{1}}
\place{P84}{\name{lugarEsp_S7}\init{0}\capacity{1}}
\place{P80}{\name{lugarEsp_S8}\init{0}\capacity{1}}
\place{P2}{\name{lugar_S1}\init{0}\capacity{1}}
\place{P33}{\name{lugar_S2}\init{0}\capacity{1}}
\place{P17}{\name{lugar_S6}\init{0}\capacity{1}}
\place{P67}{\name{lugar_S5}\init{0}\capacity{1}}
\place{P61}{\name{lugar_S4}\init{0}\capacity{1}}
\place{P39}{\name{lugar_S3}\init{0}\capacity{1}}
\place{P70}{\name{lugar_S9}\init{0}\capacity{1}}
\place{P79}{\name{lugar_S8}\init{0}\capacity{1}}
\place{P83}{\name{lugar_S7}\init{0}\capacity{1}}

```

```

\transition{T22}{\name{trans_M24}}
\transition{T72}{\name{trans_M27}}
\transition{T49}{\name{trans_M21}}
\transition{T14}{\name{trans_M17}}
\transition{T9}{\name{trans_M15}}
\transition{T7}{\name{trans_M13}}
\transition{T65}{\name{trans_GP_S5}}
\transition{T56}{\name{trans_GP_S25}}
\transition{T59}{\name{trans_GP_S4}}
\transition{T79}{\name{trans_GP_S7}}
\transition{T21}{\name{trans_GP_S23}}
\transition{T18}{\name{trans_GP_S6}}
\transition{T42}{\name{trans_GP_S22}}
\transition{T68}{\name{trans_GP_S9}}
\transition{T76}{\name{trans_GP_S8}}
\transition{T45}{\name{trans_GP_S20}}
\transition{T62}{\name{trans_GP_S28}}
\transition{T71}{\name{trans_GP_S26}}
\transition{T67}{\name{trans_S9}}
\transition{T75}{\name{trans_S8}}
\transition{T78}{\name{trans_S7}}
\transition{T29}{\name{trans_End}}
\transition{T17}{\name{trans_S6}}
\transition{T64}{\name{trans_S5}}
\transition{T58}{\name{trans_S4}}
\transition{T38}{\name{trans_S3}}
\transition{T32}{\name{trans_S2}}
\transition{T2}{\name{trans_S1}}
\transition{T30}{\name{trans_GP_S12}}

```

```

\transition{T53}{\name{trans_GP_S11}}
\transition{T6}{\name{trans_GP_S14}}
\transition{T48}{\name{trans_GP_S10}}
\transition{T36}{\name{trans_GP_S19}}
\transition{T13}{\name{trans_GP_S16}}
\transition{T26}{\name{trans_GP_S18}}
\transition{T55}{\name{trans_S25}}
\transition{T20}{\name{trans_S23}}
\transition{T41}{\name{trans_S22}}
\transition{T44}{\name{trans_S20}}
\transition{T35}{\name{trans_S19}}
\transition{T25}{\name{trans_S18}}
\transition{T12}{\name{trans_S16}}
\transition{T28}{\name{trans_S12}}
\transition{T52}{\name{trans_S11}}
\transition{T5}{\name{trans_S14}}
\transition{T47}{\name{trans_S10}}
\transition{T3}{\name{trans_GP_S1}}
\transition{T39}{\name{trans_GP_S3}}
\transition{T33}{\name{trans_GP_S2}}
\transition{T73}{\name{trans_GB_M27}}
\transition{T23}{\name{trans_GB_M24}}
\transition{T50}{\name{trans_GB_M21}}
\transition{T0}{\name{trans_Start}}
\transition{T70}{\name{trans_S26}}
\transition{T15}{\name{trans_GB_M17}}
\transition{T61}{\name{trans_S28}}
\transition{T10}{\name{trans_GB_M15}}
\transition{T8}{\name{trans_GB_M13}}

```

```

\arc{A1} {\from{P22} \to{T22} \weight{1}\type{ordinary}}
\arc{A2} {\from{T22} \to{P26} \weight{1}\type{ordinary}}
\arc{A3} {\from{P76} \to{T72} \weight{1}\type{ordinary}}
\arc{A4} {\from{T72} \to{P85} \weight{1}\type{ordinary}}
\arc{A5} {\from{P51} \to{T49} \weight{1}\type{ordinary}}
\arc{A6} {\from{T49} \to{P6} \weight{1}\type{ordinary}}
\arc{A7} {\from{P14} \to{T14} \weight{1}\type{ordinary}}
\arc{A8} {\from{T14} \to{P21} \weight{1}\type{ordinary}}
\arc{A9} {\from{P9} \to{T9} \weight{1}\type{ordinary}}
\arc{A10} {\from{T9} \to{P59} \weight{1}\type{ordinary}}
\arc{A11} {\from{P7} \to{T7} \weight{1}\type{ordinary}}
\arc{A12} {\from{T7} \to{P37} \weight{1}\type{ordinary}}
\arc{A13} {\from{P68} \to{T65} \weight{1}\type{ordinary}}
\arc{A14} {\from{T65} \to{P67} \weight{1}\type{ordinary}}
\arc{A15} {\from{P59} \to{T56} \weight{1}\type{ordinary}}
\arc{A16} {\from{T56} \to{P58} \weight{1}\type{ordinary}}
\arc{A17} {\from{P62} \to{T59} \weight{1}\type{ordinary}}
\arc{A18} {\from{T59} \to{P61} \weight{1}\type{ordinary}}
\arc{A19} {\from{P85} \to{T79} \weight{1}\type{ordinary}}
\arc{A20} {\from{T79} \to{P83} \weight{1}\type{ordinary}}
\arc{A21} {\from{P21} \to{T21} \weight{1}\type{ordinary}}
\arc{A22} {\from{T21} \to{P20} \weight{1}\type{ordinary}}
\arc{A23} {\from{P18} \to{T18} \weight{1}\type{ordinary}}
\arc{A24} {\from{T18} \to{P17} \weight{1}\type{ordinary}}
\arc{A25} {\from{P43} \to{T42} \weight{1}\type{ordinary}}
\arc{A26} {\from{T42} \to{P42} \weight{1}\type{ordinary}}
\arc{A27} {\from{P72} \to{T68} \weight{1}\type{ordinary}}
\arc{A28} {\from{T68} \to{P70} \weight{1}\type{ordinary}}
\arc{A29} {\from{P81} \to{T76} \weight{1}\type{ordinary}}
\arc{A30} {\from{T76} \to{P79} \weight{1}\type{ordinary}}
\arc{A31} {\from{P46} \to{T45} \weight{1}\type{ordinary}}
\arc{A32} {\from{T45} \to{P45} \weight{1}\type{ordinary}}

```



```

\arc{A95} {\from{T12} \to{P26} \weight{1}\type{ordinary}}
\arc{A96} {\from{T12} \to{P15} \weight{1}\type{ordinary}}
\arc{A97} {\from{P28} \to{T28} \weight{1}\type{ordinary}}
\arc{A98} {\from{T28} \to{P29} \weight{1}\type{ordinary}}
\arc{A99} {\from{P54} \to{T52} \weight{1}\type{ordinary}}
\arc{A100} {\from{T52} \to{P55} \weight{1}\type{ordinary}}
\arc{A101} {\from{P5} \to{T5} \weight{1}\type{ordinary}}
\arc{A102} {\from{T5} \to{P10} \weight{1}\type{ordinary}}
\arc{A103} {\from{T5} \to{P13} \weight{1}\type{ordinary}}
\arc{A104} {\from{P48} \to{T47} \weight{1}\type{ordinary}}
\arc{A105} {\from{T47} \to{P49} \weight{1}\type{ordinary}}
\arc{A106} {\from{P3} \to{T3} \weight{1}\type{ordinary}}
\arc{A107} {\from{T3} \to{P2} \weight{1}\type{ordinary}}
\arc{A108} {\from{P40} \to{T39} \weight{1}\type{ordinary}}
\arc{A109} {\from{T39} \to{P39} \weight{1}\type{ordinary}}
\arc{A110} {\from{P34} \to{T33} \weight{1}\type{ordinary}}
\arc{A111} {\from{T33} \to{P33} \weight{1}\type{ordinary}}
\arc{A112} {\from{P77} \to{T73} \weight{1}\type{ordinary}}
\arc{A113} {\from{T73} \to{P76} \weight{1}\type{ordinary}}
\arc{A114} {\from{P23} \to{T23} \weight{1}\type{ordinary}}
\arc{A115} {\from{T23} \to{P22} \weight{1}\type{ordinary}}
\arc{A116} {\from{P52} \to{T50} \weight{1}\type{ordinary}}
\arc{A117} {\from{T50} \to{P51} \weight{1}\type{ordinary}}
\arc{A118} {\from{P0} \to{T0} \weight{1}\type{ordinary}}
\arc{A119} {\from{T0} \to{P3} \weight{1}\type{ordinary}}
\arc{A120} {\from{T0} \to{P18} \weight{1}\type{ordinary}}
\arc{A121} {\from{T0} \to{P34} \weight{1}\type{ordinary}}
\arc{A122} {\from{T0} \to{P40} \weight{1}\type{ordinary}}
\arc{A123} {\from{T0} \to{P62} \weight{1}\type{ordinary}}
\arc{A124} {\from{T0} \to{P68} \weight{1}\type{ordinary}}
\arc{A125} {\from{P74} \to{T70} \weight{1}\type{ordinary}}
\arc{A126} {\from{T70} \to{P77} \weight{1}\type{ordinary}}
\arc{A127} {\from{T70} \to{P81} \weight{1}\type{ordinary}}
\arc{A128} {\from{P15} \to{T15} \weight{1}\type{ordinary}}
\arc{A129} {\from{T15} \to{P14} \weight{1}\type{ordinary}}
\arc{A130} {\from{P64} \to{T61} \weight{1}\type{ordinary}}
\arc{A131} {\from{T61} \to{P72} \weight{1}\type{ordinary}}
\arc{A132} {\from{T61} \to{P75} \weight{1}\type{ordinary}}
\arc{A133} {\from{P10} \to{T10} \weight{1}\type{ordinary}}
\arc{A134} {\from{T10} \to{P9} \weight{1}\type{ordinary}}
\arc{A135} {\from{P8} \to{T8} \weight{1}\type{ordinary}}
\arc{A136} {\from{T8} \to{P7} \weight{1}\type{ordinary}}

```

```

\endnet

```

Apêndice D

Arquivo PNML Gerado no Experimento

```
<?xml version="1.0" encoding="ISO-8859-1"?>
    <value>0</value>
    </initialMarking>
    </place>
    <pnml xmlns="~/gise/pnml">
        <net id="rede001" type="~/gise/pnml/PTNet">
            <name>
                <value>Rede de Petri L/T gerada</value>
            </name>
            <place id="P74">
                <graphics>
                    <position x="0" y="0" />
                </graphics>
                <name>
                    <value>lugar_S26</value>
                </name>
                <initialMarking>
                    <value>0</value>
                </initialMarking>
            </place>
            <place id="P58">
                <graphics>
                    <position x="0" y="0" />
                </graphics>
                <name>
                    <value>lugar_S25</value>
                </name>
                <initialMarking>
                    <value>0</value>
                </initialMarking>
            </place>
            <place id="P20">
```



```

<name>
  <value>lugar_GB_M13</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<place id="P64">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S28</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<place id="P7">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_M13</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

</place>
<transition id="T22">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_M24</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
</toolspecific>
</transition>

<arc id="a1" source="P22" target="T22">
  <inscription>
    <value>1</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<arc id="a2" source="T22" target="P26">
  <inscription>
    <value>1</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```

```

</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<transition id="T72">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_M27</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a3" source="P76" target="T72">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a4" source="T72" target="P85">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<transition id="T49">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_M21</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a5" source="P51" target="T49">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```

```

</arc>
<arc id="a6" source="T49" target="P6">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<place id="P9">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_M15</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P77">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GB_M27</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P14">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_M17</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P52">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GB_M21</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P23">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GB_M23</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

```

```

<name>
  <value>lugar_GB_M24</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<transition id="T14">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_M17</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a7" source="P14" target="T14">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a8" source="T14" target="P21">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T9">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_M15</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a9" source="P9" target="T9">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```

```

    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a10" source="T9" target="P59">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<place id="P51">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_M21</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P22">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_M24</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<transition id="T7">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_M13</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a11" source="P7" target="T7">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>

```

```

</arc>
<arc id="a12" source="T7" target="P37">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<place id="P30">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_End</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P31">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S12</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P56">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S11</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P50">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S10</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P76">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

```

```

<name>
  <value>lugar_M27</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<transition id="T65">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S5</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a13" source="P68" target="T65">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a14" source="T65" target="P67">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T56">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S25</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a15" source="P59" target="T56">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```

```

    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a16" source="T56" target="P58">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T59">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S4</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a17" source="P62" target="T59">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a18" source="T59" target="P61">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T79">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S7</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

```



```

    </toolspecific>
</transition>

<arc id="a19" source="P85" target="T79">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a20" source="T79" target="P83">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T21">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S23</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>immediata</tipo>
  </toolspecific>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a21" source="P21" target="T21">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a22" source="T21" target="P20">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T18">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S6</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>immediata</tipo>
  </toolspecific>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
  </toolspecific>
</transition>

```

```

    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a23" source="P18" target="T18">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a24" source="T18" target="P17">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T42">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S22</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a25" source="P43" target="T42">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a26" source="T42" target="P42">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```

```

<transition id="T68">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S9</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a27" source="P72" target="T68">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a28" source="T68" target="P70">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<position x="0" y="0" />
</graphics>
</arc>

<transition id="T76">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S8</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a29" source="P81" target="T76">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a30" source="T76" target="P79">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```

```

        <offset x="0" y="0" />
    </graphics>
</inscription>
<graphics>
    <position x="0" y="0" />
</graphics>
</arc>

<transition id="T45">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>trans_GP_S20</value>
    </name>
    <toolspecific tool="APVFSE" version="1.0.0">
        <tipo>mediata</tipo>
        <intervalo-de-disparo>
            <eft>0.0</eft>
            <lft>0.0</lft>
        </intervalo-de-disparo>
    </toolspecific>
</transition>

<arc id="a31" source="P46" target="T45">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>

<arc id="a32" source="T45" target="P45">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>

<place id="P26">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>lugar_GP_S18</value>
    </name>
    <initialMarking>
        <value>0</value>
    </initialMarking>
</place>

<place id="P37">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>lugar_GP_S19</value>
    </name>
    <initialMarking>
        <value>0</value>
    </initialMarking>
</place>

```

```

    </initialMarking>
</place>

<place id="P13">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S16</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<transition id="T62">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S28</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a33" source="P65" target="T62">
  <inscription>
    <value>1</value>
  </inscription>
</arc>

<place id="P6">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S14</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<transition id="T71">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S16</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<arc id="a34" source="T62" target="P64">
  <inscription>
    <value>1</value>
  </inscription>
</arc>

```

```

    <position x="0" y="0" />
</graphics>
<name>
    <value>trans_GP_S26</value>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
        <eft>0.0</eft>
        <lft>0.0</lft>
    </intervalo-de-disparo>
</toolspecific>
</transition>
<arc id="a35" source="P75" target="T71">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>
<arc id="a36" source="T71" target="P74">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>
    </arc>
</transition>
<transition id="T67">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>trans_S9</value>
        <graphics>
            <offset x="0" y="0" />
        </graphics>
    </name>
    <toolspecific tool="APVFSE" version="1.0.0">
        <tipo>mediata</tipo>
        <intervalo-de-disparo>
            <eft>0.0</eft>
            <lft>0.0</lft>
        </intervalo-de-disparo>
    </toolspecific>
</transition>
<arc id="a37" source="P70" target="T67">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>
<arc id="a38" source="T67" target="P71">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>

```

```

</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<place id="P54">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S11</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<transition id="T75">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S8</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a39" source="P79" target="T75">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a40" source="T75" target="P80">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<place id="P48">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S10</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

```

```

<transition id="T78">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S7</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a41" source="P83" target="T78">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a42" source="T78" target="P84">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<transition id="T29">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_End</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a43" source="P29" target="T29">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a44" source="P49" target="T29">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```



```

    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a45" source="P55" target="T29">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a46" source="P71" target="T29">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a47" source="P80" target="T29">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a48" source="P84" target="T29">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a49" source="T29" target="P30">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<place id="P28">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S12</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</place>

```

```

</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<transition id="T17">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S6</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a50" source="P17" target="T17">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a51" source="T17" target="P21">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<transition id="T64">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S5</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a52" source="P67" target="T64">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```

```

</arc>
<arc id="a53" source="T64" target="P65">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>
<place id="P5">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S14</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>
<transition id="T58">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S4</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
</toolspecific>
</transition>
<arc id="a54" source="P61" target="T58">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>
<arc id="a55" source="T58" target="P59">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>
<transition id="T38">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S3</value>
  </name>

```

```

    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a56" source="P39" target="T38">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a57" source="T38" target="P43">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<place id="P12">
  <graphics>
    <position x="0" y="0" />
  </graphics>
</place>

  <position x="0" y="0" />
</graphics>
<name>
  <value>lugar_S16</value>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<transition id="T32">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S2</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a58" source="P33" target="T32">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```

```

<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<arc id="a59" source="T32" target="P37">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T2">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>imediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a60" source="P2" target="T2">
  <inscription>
    <value>1</value>

```

```

<graphics>
  <offset x="0" y="0" />
</graphics>
</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<arc id="a61" source="T2" target="P6">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a62" source="T2" target="P8">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<place id="P36">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S19</value>

```

```

    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P25">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S18</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<transition id="T30">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S12</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>imediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a63" source="P31" target="T30">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a64" source="T30" target="P28">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<transition id="T53">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S11</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>imediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

```

```

        <tipo>mediata</tipo>
        <intervalo-de-disparo>
            <eft>0.0</eft>
            <lft>0.0</lft>
        </intervalo-de-disparo>
    </toolspecific>
</transition>

<arc id="a65" source="P56" target="T53">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>

<arc id="a66" source="T53" target="P54">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>

<transition id="T6">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>trans_GP_S14</value>
    </name>
</transition>

        <offset x="0" y="0" />
    </graphics>
</name>
    <toolspecific tool="APVFSE" version="1.0.0">
        <tipo>mediata</tipo>
        <intervalo-de-disparo>
            <eft>0.0</eft>
            <lft>0.0</lft>
        </intervalo-de-disparo>
    </toolspecific>
</transition>

<arc id="a67" source="P6" target="T6">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>

<arc id="a68" source="T6" target="P5">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>

<transition id="T48">
    <graphics>
        <position x="0" y="0" />
    </graphics>
</transition>

```

```

</graphics>
<name>
  <value>trans_GP_S10</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
</toolspecific>
</transition>

<arc id="a69" source="P50" target="T48">
  <inscription>
    <value>1</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<arc id="a70" source="T48" target="P48">
  <inscription>
    <value>1</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<transition id="T36">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S19</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
</toolspecific>
</transition>

<arc id="a71" source="P37" target="T36">
  <inscription>
    <value>1</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<arc id="a72" source="T36" target="P36">
  <inscription>
    <value>1</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</inscription>

```



```

<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<transition id="T13">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S16</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a73" source="P13" target="T13">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<arc id="a74" source="T13" target="P12">
  <inscription>
    <value>1</value>
  </inscription>
</arc>

<graphics>
  <offset x="0" y="0" />
</graphics>
</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<transition id="T26">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S18</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a75" source="P26" target="T26">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```

```

<arc id="a76" source="T26" target="P25">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T55">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S25</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>imediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a77" source="P58" target="T55">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>

  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a78" source="T55" target="P75">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a79" source="T55" target="P65">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T20">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S23</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>

```

```

<toolspecific tool="APVFSE" version="1.0.0"> </inscription>
  <tipo>mediata</tipo> <graphics>
  <intervalo-de-disparo> <position x="0" y="0" />
  <eft>0.0</eft> </graphics>
  <lft>0.0</lft> </arc>
  </intervalo-de-disparo>
</toolspecific> <transition id="T41">
</transition> <graphics>
  <position x="0" y="0" />
<arc id="a80" source="P20" target="T20"> </graphics>
  <inscription> <name>
  <value>1</value> <value>trans_S22</value>
  <graphics> <graphics>
  <offset x="0" y="0" /> <offset x="0" y="0" />
  </graphics> </graphics>
</inscription> </name>
<graphics> <toolspecific tool="APVFSE" version="1.0.0">
  <position x="0" y="0" /> <tipo>mediata</tipo>
  </graphics> <intervalo-de-disparo>
  <eft>0.0</eft> <eft>0.0</eft>
  <lft>0.0</lft> <lft>0.0</lft>
  </intervalo-de-disparo>
</arc> </toolspecific>
<arc id="a81" source="T20" target="P31"> </transition>
  <inscription> <arc id="a83" source="P42" target="T41">
  <value>1</value> <inscription>
  <graphics> <value>1</value>
  <offset x="0" y="0" /> <graphics>
  </graphics> <offset x="0" y="0" />
</inscription> </graphics>
<graphics> </graphics>
  <position x="0" y="0" /> <offset x="0" y="0" />
  </graphics> </graphics>
</arc> </inscription>
  <position x="0" y="0" /> <graphics>
  <arc id="a82" source="T20" target="P23"> <position x="0" y="0" />
  <inscription> </graphics>
  <value>1</value> </arc>
  <graphics> <arc id="a84" source="T41" target="P50">
  <offset x="0" y="0" /> <inscription>
  </graphics> </arc>
  </graphics> </arc>
  </graphics> </arc>

```

```

    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a85" source="T41" target="P46">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T44">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S20</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>imediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a86" source="P45" target="T44">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a87" source="T44" target="P56">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a88" source="T44" target="P52">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<place id="P49">

```

```

<graphics>
  <position x="0" y="0" />
</graphics>
<name>
  <value>lugarEsp_S10</value>
<graphics>
  <offset x="0" y="0" />
</graphics>
</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<place id="P55">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugarEsp_S11</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P29">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugarEsp_S12</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>

  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P0">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_Start</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<initialMarking>
  <value>1</value>
</initialMarking>
</place>

<transition id="T35">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S19</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
</toolspecific>
</transition>

```

```

<arc id="a89" source="P36" target="T35">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a90" source="T35" target="P43">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a91" source="T35" target="P46">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T25">
  <graphics>
    <position x="0" y="0" />
  </graphics>
</transition>

</graphics>
<name>
  <value>trans_S18</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>imediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
</toolspecific>
</transition>

<arc id="a92" source="P25" target="T25">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a93" source="T25" target="P85">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```

```

<transition id="T12">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S16</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>imediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a94" source="P12" target="T12">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a95" source="T12" target="P26">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a96" source="T12" target="P15">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T28">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S12</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>imediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a97" source="P28" target="T28">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```



```

        <offset x="0" y="0" />
</graphics>
</inscription>
<graphics>
    <position x="0" y="0" />
</graphics>
</arc>
<arc id="a102" source="T5" target="P10">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>
<arc id="a103" source="T5" target="P13">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>
<place id="P75">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>lugar_GP_S26</value>
    </name>
    <initialMarking>
        <value>0</value>
    </initialMarking>
</place>
<transition id="T5">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>trans_S14</value>
    </name>
    <toolspecific tool="APVFSE" version="1.0.0">
        <tipo>mediata</tipo>
        <intervalo-de-disparo>
            <eft>0.0</eft>
            <lft>0.0</lft>
        </intervalo-de-disparo>
    </toolspecific>
</transition>
<arc id="a101" source="P5" target="T5">
    <inscription>
        <value>1</value>
    </inscription>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</arc>
<place id="P59">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>lugar_GP_S25</value>
    </name>
    <initialMarking>
        <value>0</value>
    </initialMarking>
</place>

```

```

        <offset x="0" y="0" />
    </graphics>
</name>
<initialMarking>
    <value>0</value>
</initialMarking>
</place>

<transition id="T47">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>trans_S10</value>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
        <eft>0.0</eft>
        <lft>0.0</lft>
    </intervalo-de-disparo>
</toolspecific>
</transition>

<arc id="a104" source="P48" target="T47">
    <inscription>
        <value>1</value>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</inscription>
<graphics>
    <position x="0" y="0" />
</graphics>
</arc>

        <arc id="a105" source="T47" target="P49">
    <inscription>
        <value>1</value>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</inscription>
<graphics>
    <position x="0" y="0" />
</graphics>
</arc>

<transition id="T3">
    <graphics>
        <position x="0" y="0" />
    </graphics>
    <name>
        <value>trans_GP_S1</value>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
        <eft>0.0</eft>
        <lft>0.0</lft>
    </intervalo-de-disparo>
</toolspecific>
</transition>

<arc id="a106" source="P3" target="T3">
    <inscription>
        <value>1</value>
    <graphics>
        <offset x="0" y="0" />
    </graphics>
</inscription>
<graphics>

```

```

    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a107" source="T3" target="P2">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T39">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S3</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a108" source="P40" target="T39">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a109" source="T39" target="P39">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T33">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GP_S2</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

```

```

<arc id="a110" source="P34" target="T33">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a111" source="T33" target="P33">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<place id="P46">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S20</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P21">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S23</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P43">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S22</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P72">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S9</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

```

```

</name>
<initialMarking>
  <value>0</value>
</initialMarking>
</place>

<place id="P18">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S6</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P68">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S5</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P81">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S8</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P85">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S7</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P34">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S2</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

```

```

    </initialMarking>
</place>

<place id="P3">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S1</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P62">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S4</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P40">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_GP_S3</value>

```

```

  <graphics>
    <offset x="0" y="0" />
  </graphics>
</name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P71">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugarEsp_S9</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P84">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugarEsp_S7</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

```

```

<place id="P80">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugarEsp_S8</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<transition id="T73">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GB_M27</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a112" source="P77" target="T73">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<place id="P80">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugarEsp_S8</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<transition id="T73">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GB_M27</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a113" source="T73" target="P76">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

<transition id="T23">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GB_M24</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a114" source="P23" target="T23">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</arc>

```

```

<inscription>
  <value>1</value>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</inscription>
<graphics>
  <position x="0" y="0" />
</graphics>
</arc>

<arc id="a115" source="T23" target="P22">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T50">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GB_M21</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a116" source="P52" target="T50">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a117" source="T50" target="P51">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T0">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_Start</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>

```



```

    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a118" source="P0" target="T0">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a119" source="T0" target="P3">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a120" source="T0" target="P18">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a121" source="T0" target="P34">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a122" source="T0" target="P40">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a123" source="T0" target="P62">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```

```

<arc id="a124" source="T0" target="P68">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<place id="P2">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S1</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P33">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S2</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P17">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S6</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P67">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S5</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P61">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>0</value>
  </initialMarking>
</place>

```

```

    <value>lugar_S4</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P39">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S3</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

<place id="P70">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>lugar_S9</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</place>

    <place id="P79">
    <graphics>
      <position x="0" y="0" />
    </graphics>
    <name>
      <value>lugar_S8</value>
    </name>
    <initialMarking>
      <value>0</value>
    </initialMarking>
  </place>

  <place id="P83">
    <graphics>
      <position x="0" y="0" />
    </graphics>
    <name>
      <value>lugar_S7</value>
    </name>
    <initialMarking>
      <value>0</value>
    </initialMarking>
  </place>

<transition id="T70">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S26</value>
  </name>
  <initialMarking>
    <value>0</value>
  </initialMarking>
</transition>

```

```

    </graphics>
</name>
<toolspecific tool="APVFSE" version="1.0.0">
  <tipo>mediata</tipo>
  <intervalo-de-disparo>
    <eft>0.0</eft>
    <lft>0.0</lft>
  </intervalo-de-disparo>
</toolspecific>
</transition>
<arc id="a125" source="P74" target="T70">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>
<arc id="a126" source="T70" target="P77">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>
<arc id="a127" source="T70" target="P81">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>
    <offset x="0" y="0" />
  </graphics>
</inscription>
</graphics>
</arc>
<transition id="T15">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GB_M17</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>
<arc id="a128" source="P15" target="T15">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```

```

<arc id="a129" source="T15" target="P14">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a131" source="T61" target="P72">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a132" source="T61" target="P75">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T61">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_S28</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tip>imediata</tip>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<transition id="T10">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GB_M15</value>
  </name>
  <toolspecific tool="APVFSE" version="1.0.0">
    <tip>imediata</tip>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a130" source="P64" target="T61">
  <inscription>
    <value>1</value>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

```

```

    <tipo>mediata</tipo>
    <intervalo-de-disparo>
      <eft>0.0</eft>
      <lft>0.0</lft>
    </intervalo-de-disparo>
  </toolspecific>
</transition>

<arc id="a133" source="P10" target="T10">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<arc id="a134" source="T10" target="P9">
  <inscription>
    <value>1</value>
    <graphics>
      <offset x="0" y="0" />
    </graphics>
  </inscription>
  <graphics>
    <position x="0" y="0" />
  </graphics>
</arc>

<transition id="T8">
  <graphics>
    <position x="0" y="0" />
  </graphics>
  <name>
    <value>trans_GB_M13</value>
  </name>
  <graphics>
    <offset x="0" y="0" />
  </graphics>
</transition>
</net>
</pnml>

```

Apêndice E

Arquivo SMV Gerado no Experimento

```
MODULE main
VAR
lugar_Start : boolean;
proc1 : process mPrincipal (lugar_Start, lugar_GP_S3, lugar_S3,
                           lugar_GP_S22, lugar_S22, lugar_GP_S10,
                           lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                           lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                           lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                           lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                           lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                           lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                           lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                           lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                           lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                           lugar_S23, lugar_GP_S12, lugar_S12);

lugar_GP_S3 : boolean;
lugar_S3 : boolean;
lugar_GP_S22 : boolean;
lugar_S22 : boolean;
lugar_GP_S10 : boolean;
lugar_S10 : boolean;
proc2 : process mSENSOR (lugar_Start, lugar_GP_S3, lugar_S3,
                         lugar_GP_S22, lugar_S22, lugar_GP_S10,
                         lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                         lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                         lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                         lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                         lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                         lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                         lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                         lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                         lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                         lugar_S23, lugar_GP_S12, lugar_S12);
```

```

lugar_GP_S5 : boolean;
lugar_S5 : boolean;
lugar_GP_S28 : boolean;
lugar_S28 : boolean;
lugar_GP_S9 : boolean;
lugar_S9 : boolean;
proc3 : process mACTUATOR (lugar_Start, lugar_GP_S3, lugar_S3,
                        lugar_GP_S22, lugar_S22, lugar_GP_S10,
                        lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                        lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                        lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                        lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                        lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                        lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                        lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                        lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                        lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                        lugar_S23, lugar_GP_S12, lugar_S12);

lugar_GP_S2 : boolean;
lugar_S2 : boolean;
lugar_GP_S19 : boolean;
lugar_S19 : boolean;
lugar_GP_S20 : boolean;
lugar_S20 : boolean;
lugar_GP_S11 : boolean;
lugar_S11 : boolean;
proc4 : process mSENSORMNG (lugar_Start, lugar_GP_S3, lugar_S3,
                        lugar_GP_S22, lugar_S22, lugar_GP_S10,
                        lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                        lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                        lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                        lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                        lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                        lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                        lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                        lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                        lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                        lugar_S23, lugar_GP_S12, lugar_S12);

lugar_GP_S1 : boolean;
lugar_S1 : boolean;
lugar_GP_S14 : boolean;
lugar_S14 : boolean;
lugar_GP_S16 : boolean;
lugar_S16 : boolean;
lugar_GP_S18 : boolean;
lugar_S18 : boolean;
lugar_GP_S7 : boolean;
lugar_S7 : boolean;
proc5 : process mMAINCONTROL (lugar_Start, lugar_GP_S3, lugar_S3,
                        lugar_GP_S22, lugar_S22, lugar_GP_S10,
                        lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                        lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                        lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                        lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                        lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                        lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                        lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                        lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                        lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                        lugar_S23, lugar_GP_S12, lugar_S12);

lugar_GP_S4 : boolean;
lugar_S4 : boolean;

```



```

lugar_GP_S25 : boolean;
lugar_S25 : boolean;
lugar_GP_S26 : boolean;
lugar_S26 : boolean;
lugar_GP_S8 : boolean;
lugar_S8 : boolean;
proc6 : process mACTUATORMNG (lugar_Start, lugar_GP_S3, lugar_S3,
                             lugar_GP_S22, lugar_S22, lugar_GP_S10,
                             lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                             lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                             lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                             lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                             lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                             lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                             lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                             lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                             lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                             lugar_S23, lugar_GP_S12, lugar_S12);

lugar_GP_S6 : boolean;
lugar_S6 : boolean;
lugar_GP_S23 : boolean;
lugar_S23 : boolean;
lugar_GP_S12 : boolean;
lugar_S12 : boolean;
proc7 : process mSIGNAL (lugar_Start, lugar_GP_S3, lugar_S3,
                        lugar_GP_S22, lugar_S22, lugar_GP_S10,
                        lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                        lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                        lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                        lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                        lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                        lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                        lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                        lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                        lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                        lugar_S23, lugar_GP_S12, lugar_S12);

ASSIGN
init(lugar_Start) := 1;
init(lugar_GP_S3) := 0;
init(lugar_S3) := 0;
init(lugar_GP_S22) := 0;
init(lugar_S22) := 0;
init(lugar_GP_S10) := 0;
init(lugar_S10) := 0;
init(lugar_GP_S5) := 0;
init(lugar_S5) := 0;
init(lugar_GP_S28) := 0;
init(lugar_S28) := 0;
init(lugar_GP_S9) := 0;
init(lugar_S9) := 0;
init(lugar_GP_S2) := 0;
init(lugar_S2) := 0;
init(lugar_GP_S19) := 0;
init(lugar_S19) := 0;
init(lugar_GP_S20) := 0;
init(lugar_S20) := 0;
init(lugar_GP_S11) := 0;
init(lugar_S11) := 0;
init(lugar_GP_S1) := 0;
init(lugar_S1) := 0;
init(lugar_GP_S14) := 0;

```

```

init(lugar_S14) := 0;
init(lugar_GP_S16) := 0;
init(lugar_S16) := 0;
init(lugar_GP_S18) := 0;
init(lugar_S18) := 0;
init(lugar_GP_S7) := 0;
init(lugar_S7) := 0;
init(lugar_GP_S4) := 0;
init(lugar_S4) := 0;
init(lugar_GP_S25) := 0;
init(lugar_S25) := 0;
init(lugar_GP_S26) := 0;
init(lugar_S26) := 0;
init(lugar_GP_S8) := 0;
init(lugar_S8) := 0;
init(lugar_GP_S6) := 0;
init(lugar_S6) := 0;
init(lugar_GP_S23) := 0;
init(lugar_S23) := 0;
init(lugar_GP_S12) := 0;
init(lugar_S12) := 0;

```

SPEC

<ALGUMA ESPECIFICAÇÃO DE CONSULTA CTL>

```

MODULE mPrincipal (lugar_Start, lugar_GP_S3, lugar_S3,
                  lugar_GP_S22, lugar_S22, lugar_GP_S10,
                  lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                  lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                  lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                  lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                  lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                  lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                  lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                  lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                  lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                  lugar_S23, lugar_GP_S12, lugar_S12)

```

ASSIGN

```

next(lugar_Start) :=
case
lugar_Start=0 : 1;
1 : lugar_Start;
esac;

```

```

MODULE mSENSOR (lugar_Start, lugar_GP_S3, lugar_S3,
                lugar_GP_S22, lugar_S22, lugar_GP_S10,
                lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                lugar_S23, lugar_GP_S12, lugar_S12)

```

ASSIGN

```

next(lugar_GP_S3) :=
case
lugar_GP_S3=0 & lugar_Start=1 : 1;
1 : lugar_GP_S3;

```

```

esac;
next(lugar_S3) :=
case
lugar_S3=0 & lugar_GP_S3=1 : 1;
1 : lugar_S3;
esac;
next(lugar_GP_S22) :=
case
lugar_GP_S22=0 & lugar_S3=1 & lugar_S19=1 : 1;
1 : lugar_GP_S22;
esac;
next(lugar_S22) :=
case
lugar_S22=0 & lugar_GP_S22=1 : 1;
1 : lugar_S22;
esac;
next(lugar_GP_S10) :=
case
lugar_GP_S10=0 & lugar_S22=1 : 1;
1 : lugar_GP_S10;
esac;
next(lugar_S10) :=
case
lugar_S10=0 & lugar_GP_S10=1 : 1;
1 : lugar_S10;
esac;

MODULE mACTUATOR (lugar_Start, lugar_GP_S3, lugar_S3,
                 lugar_GP_S22, lugar_S22, lugar_GP_S10,
                 lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                 lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                 lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                 lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                 lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                 lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                 lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                 lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                 lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                 lugar_S23, lugar_GP_S12, lugar_S12)

ASSIGN
next(lugar_GP_S5) :=
case
lugar_GP_S5=0 & lugar_Start=1 : 1;
1 : lugar_GP_S5;
esac;
next(lugar_S5) :=
case
lugar_S5=0 & lugar_GP_S5=1 : 1;
1 : lugar_S5;
esac;
next(lugar_GP_S28) :=
case
lugar_GP_S28=0 & lugar_S5=1 & lugar_S25=1 : 1;
1 : lugar_GP_S28;
esac;
next(lugar_S28) :=
case
lugar_S28=0 & lugar_GP_S28=1 : 1;
1 : lugar_S28;
esac;
next(lugar_GP_S9) :=

```

```

case
lugar_GP_S9=0 & lugar_S28=1 : 1;
1 : lugar_GP_S9;
esac;
next(lugar_S9) :=
case
lugar_S9=0 & lugar_GP_S9=1 : 1;
1 : lugar_S9;
esac;

MODULE mSENSORMNG (lugar_Start, lugar_GP_S3, lugar_S3,
                    lugar_GP_S22, lugar_S22, lugar_GP_S10,
                    lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                    lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                    lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                    lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                    lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                    lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                    lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                    lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                    lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                    lugar_S23, lugar_GP_S12, lugar_S12)

ASSIGN
next(lugar_GP_S2) :=
case
lugar_GP_S2=0 & lugar_Start=1 : 1;
1 : lugar_GP_S2;
esac;
next(lugar_S2) :=
case
lugar_S2=0 & lugar_GP_S2=1 : 1;
1 : lugar_S2;
esac;
next(lugar_GP_S19) :=
case
lugar_GP_S19=0 & lugar_S1=1 & lugar_S2=1 : 1;
1 : lugar_GP_S19;
esac;
next(lugar_S19) :=
case
lugar_S19=0 & lugar_GP_S19=1 : 1;
1 : lugar_S19;
esac;
next(lugar_GP_S20) :=
case
lugar_GP_S20=0 & lugar_S22=1 & lugar_S19=1 : 1;
1 : lugar_GP_S20;
esac;
next(lugar_S20) :=
case
lugar_S20=0 & lugar_GP_S20=1 : 1;
1 : lugar_S20;
esac;
next(lugar_GP_S11) :=
case
lugar_GP_S11=0 & lugar_S20=1 : 1;
1 : lugar_GP_S11;
esac;
next(lugar_S11) :=
case
lugar_S11=0 & lugar_GP_S11=1 : 1;

```

```
1 : lugar_S11;
esac;
```

```
MODULE mMAINCONTROL (lugar_Start, lugar_GP_S3, lugar_S3,
                    lugar_GP_S22, lugar_S22, lugar_GP_S10,
                    lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                    lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                    lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                    lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                    lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                    lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                    lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                    lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                    lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                    lugar_S23, lugar_GP_S12, lugar_S12)
```

```
ASSIGN
```

```
next(lugar_GP_S1) :=
case
lugar_GP_S1=0 & lugar_Start=1 : 1;
1 : lugar_GP_S1;
esac;
next(lugar_S1) :=
case
lugar_S1=0 & lugar_GP_S1=1 : 1;
1 : lugar_S1;
esac;
next(lugar_GP_S14) :=
case
lugar_GP_S14=0 & lugar_S20=1 & lugar_S1=1 : 1;
1 : lugar_GP_S14;
esac;
next(lugar_S14) :=
case
lugar_S14=0 & lugar_GP_S14=1 : 1;
1 : lugar_S14;
esac;
next(lugar_GP_S16) :=
case
lugar_GP_S16=0 & lugar_S14=1 : 1;
1 : lugar_GP_S16;
esac;
next(lugar_S16) :=
case
lugar_S16=0 & lugar_GP_S16=1 : 1;
1 : lugar_S16;
esac;
next(lugar_GP_S18) :=
case
lugar_GP_S18=0 & lugar_S23=1 & lugar_S16=1 : 1;
1 : lugar_GP_S18;
esac;
next(lugar_S18) :=
case
lugar_S18=0 & lugar_GP_S18=1 : 1;
1 : lugar_S18;
esac;
next(lugar_GP_S7) :=
case
lugar_GP_S7=0 & lugar_S26=1 & lugar_S18=1 : 1;
1 : lugar_GP_S7;
esac;
```

```

next(lugar_S7) :=
case
lugar_S7=0 & lugar_GP_S7=1 : 1;
1 : lugar_S7;
esac;

MODULE mACTUATORMNG (lugar_Start, lugar_GP_S3, lugar_S3,
                    lugar_GP_S22, lugar_S22, lugar_GP_S10,
                    lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                    lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                    lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                    lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                    lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                    lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                    lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                    lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                    lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                    lugar_S23, lugar_GP_S12, lugar_S12)

ASSIGN
next(lugar_GP_S4) :=
case
lugar_GP_S4=0 & lugar_Start=1 : 1;
1 : lugar_GP_S4;
esac;
next(lugar_S4) :=
case
lugar_S4=0 & lugar_GP_S4=1 : 1;
1 : lugar_S4;
esac;
next(lugar_GP_S25) :=
case
lugar_GP_S25=0 & lugar_S14=1 & lugar_S4=1 : 1;
1 : lugar_GP_S25;
esac;
next(lugar_S25) :=
case
lugar_S25=0 & lugar_GP_S25=1 : 1;
1 : lugar_S25;
esac;
next(lugar_GP_S26) :=
case
lugar_GP_S26=0 & lugar_S25=1 & lugar_S28=1 : 1;
1 : lugar_GP_S26;
esac;
next(lugar_S26) :=
case
lugar_S26=0 & lugar_GP_S26=1 : 1;
1 : lugar_S26;
esac;
next(lugar_GP_S8) :=
case
lugar_GP_S8=0 & lugar_S26=1 : 1;
1 : lugar_GP_S8;
esac;
next(lugar_S8) :=
case
lugar_S8=0 & lugar_GP_S8=1 : 1;
1 : lugar_S8;
esac;

```

```

MODULE mSIGNAL (lugar_Start, lugar_GP_S3, lugar_S3,
                lugar_GP_S22, lugar_S22, lugar_GP_S10,
                lugar_S10, lugar_GP_S5, lugar_S5, lugar_GP_S28,
                lugar_S28, lugar_GP_S9, lugar_S9, lugar_GP_S2,
                lugar_S2, lugar_GP_S19, lugar_S19, lugar_GP_S20,
                lugar_S20, lugar_GP_S11, lugar_S11, lugar_GP_S1,
                lugar_S1, lugar_GP_S14, lugar_S14, lugar_GP_S16,
                lugar_S16, lugar_GP_S18, lugar_S18, lugar_GP_S7,
                lugar_S7, lugar_GP_S4, lugar_S4, lugar_GP_S25,
                lugar_S25, lugar_GP_S26, lugar_S26, lugar_GP_S8,
                lugar_S8, lugar_GP_S6, lugar_S6, lugar_GP_S23,
                lugar_S23, lugar_GP_S12, lugar_S12)

ASSIGN
next(lugar_GP_S6) :=
case
lugar_GP_S6=0 & lugar_Start=1 : 1;
1 : lugar_GP_S6;
esac;
next(lugar_S6) :=
case
lugar_S6=0 & lugar_GP_S6=1 : 1;
1 : lugar_S6;
esac;
next(lugar_GP_S23) :=
case
lugar_GP_S23=0 & lugar_S16=1 & lugar_S6=1 : 1;
1 : lugar_GP_S23;
esac;
next(lugar_S23) :=
case
lugar_S23=0 & lugar_GP_S23=1 : 1;
1 : lugar_S23;
esac;
next(lugar_GP_S12) :=
case
lugar_GP_S12=0 & lugar_S23=1 : 1;
1 : lugar_GP_S12;
esac;
next(lugar_S12) :=
case
lugar_S12=0 & lugar_GP_S12=1 : 1;
1 : lugar_S12;
esac;

```

Índice Remissivo

Algoritmos de Grafos, 61
APNN, 4, 19, 71
Autômatos, 57, 61, 63

CTL, 4, 19, 20, 72, 141

Diagrama de Sequência, 140
diagrama de sequência estendido, 22

INA, 108

MCK, 4, 19, 71, 87
Model Checking, 19, 59, 141

PNML, 4, 19, 71

Redes de Petri, 4, 19, 70, 141

Simulação, 60
SMV, 4, 20, 60, 67, 72, 113, 141
software embarcado, 140
software embarcado, 34, 69

Teste de Software, 60

UML, 4, 19, 37, 69

XML, 71

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)