



Pós-Graduação em Ciência da Computação

“ASTRO - Uma Ferramenta para Avaliação de Dependabilidade e Sustentabilidade em Sistemas Data Center”

Por

Bruno Silva

Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, MARÇO/2011



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

BRUNO SILVA

“ASTRO - Uma Ferramenta para Avaliação de
Dependabilidade e Sustentabilidade em Sistemas
Data Center”

*ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO
PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA
DA COMPUTAÇÃO.*

ORIENTADOR(A): Paulo Romero Martins Maciel

RECIFE, MARÇO/2011

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Silva, Bruno

ASTRO- uma ferramenta para avaliação de dependabilidade e sustentabilidade em sistemas *data center* / Bruno Silva - Recife: O Autor, 2011.

vii, 99 folhas : il., fig., tab.

Orientador: Paulo Romero Martins Maciel.

Dissertação (mestrado) Universidade Federal de Pernambuco. Cln. Ciência da Computação, 2011.

Inclui bibliografia.

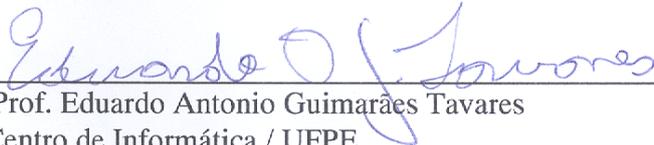
1. Computação tolerante a falhas . 2. Avaliação de sustentabilidade. 3. Redes de Petri. 4. Diagrama de blocos. I. Maciel, Paulo Romero Martins (orientador). II. Título.

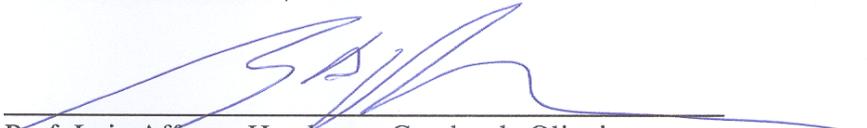
004.2

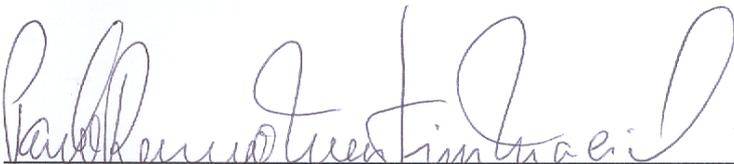
CDD (22. ed.)

MEI2011 – 044

Dissertação de Mestrado apresentada por **Bruno Silva** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**ASTRO - Uma Ferramenta para Avaliação de Dependabilidade e Sustentabilidade em Ambientes Data Center**”, orientada pelo **Prof. Paulo Romero Martins Maciel** e aprovada pela Banca Examinadora formada pelos professores:


Prof. Eduardo Antonio Guimarães Tavares
Centro de Informática / UFPE


Prof. Luiz Affonso Henderson Guedes de Oliveira
Departº de Engenharia de Computação e Automação / UFRN


Prof. Paulo Romero Martins Maciel
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 2 de março de 2011.


Prof. Nelson Souto Rosa
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

À minha família.

Agradecimentos

- Ao meu bom Deus, por ter me dado essa oportunidade.
- Ao meu orientador, Paulo Romero Martins Maciel. A idéia inicial deste projeto foi dele e durante todo mestrado Paulo me incentivou e ensinou, sempre com bastante objetividade, sobre pesquisa, engenharia e escrita. Os seus sábios conselhos foram de grande importância para que eu me transformasse em um engenheiro mais capacitado.
- Aos amigos do MoDCS, com quem convivi nos últimos, pelo companheirismo e períodos de descontração. Em especial: Edurado Tavares, Gustavo Callou, Julian Menezes, Jair Figueredo e Sérgio Galdino.
- Aos Amigos, família e esposa, Suellen Cunha, pelo suporte necessário durante todo o período do mestrado.

Amai-vos uns aos outros como eu vos amei
—BIBLIA SAGRADA. (Jo 15,12-17)

Aspectos de sustentabilidade têm recebido grande atenção pela comunidade científica, devido às preocupações com a satisfação das necessidades atuais de energia sem comprometer, por exemplo, recursos não-renováveis para as gerações futuras. Na verdade, uma crescente demanda de energia é uma questão que tem impactado a forma de como os sistemas são concebidos (*data centers*, por exemplo), no sentido de que os projetistas necessitam verificar vários *trade-offs* e selecionar uma solução viável considerando a utilização da energia e outras métricas, tais como confiabilidade e disponibilidade. As ferramentas são importantes neste contexto para automatizar várias atividades de projeto e obter resultados o mais rápido possível. Este trabalho apresenta um ambiente integrado, denominado, ASTRO, que contempla: (i) Diagramas de Blocos de Confiabilidade (RBD) e Redes de Petri Estocásticas (SPN) para avaliação de dependabilidade, (ii) um método baseado na avaliação do ciclo de vida (LCA) para a quantificação do impacto da sustentabilidade. ASTRO foi concebido para avaliar infra-estruturas de centros de dados, mas o ambiente é genérico o suficiente para avaliar sistemas em geral. Além disso, um estudo de caso é fornecido para demonstrar a viabilidade do ambiente proposto.

Palavras-chave: avaliação de dependabilidade, avaliação de sustentabilidade, *data center*, diagramas de bloco de confiabilidade, redes de petri estocásticas.

Abstract

Sustainability has received great attention by the scientific community, due to concerns for meeting current needs of energy without compromising, for instance, non-renewable resources for future generations. Indeed, the higher demand for energy is an issue that has impacted the way systems are conceived (e.g. data centers), in the sense that designers need to verify several trade-offs and select the feasible solution considering energy utilization and other metrics, such as dependability. Tools are important in this context to automate several design activities and provide results as fast as possible. This paper presents an integrated environment, namely, ASTRO, which contemplates: (i) Reliability Block Diagrams (RBD) and Stochastic Petri Nets (SPN) for dependability evaluation; and (ii) a method based on life-cycle assessment (LCA) for quantification of sustainability impact. ASTRO has been conceived to evaluate data center infrastructures, but the environment is generic enough to evaluate general systems. Besides, a real world case study is provided to demonstrate the feasibility of the proposed environment.

Keywords: dependability evaluation, sustainability evaluation, data center systems, reliability block diagrams, stochastic petri nets.

Identificação

Mestrando

Bruno Silva

Orientador

Paulo Romero Martins Maciel

Título

ASTRO - Uma Ferramenta para Avaliação de Dependabilidade e Sustentabilidade em Sistemas Data Center

Área de concentração

Avaliação de Desempenho

Linha de pesquisa

Sistemas Data Center

Sumário

1	Introdução	1
1.1	Motivação.	1
1.2	Objetivos	3
1.3	Trabalhos relacionados	4
1.4	Estrutura da dissertação	6
2	Conceitos Básicos	7
2.1	infraestrutura <i>Data Center</i>	7
2.2	Dependabilidade	9
2.2.1	Equações de dependabilidade	9
2.3	Sustentabilidade	11
2.3.1	Exergia	12
2.4	Modelos adotados	12
2.4.1	Modelo Energético.	13
2.4.2	Modelos de Sustentabilidade	15
2.4.2.1	TCO - Custo total da infraestrutura	17
2.4.3	Modelos RBD	18
2.4.3.1	Funções estruturais RBD.	19
2.4.3.2	Funções lógicas RBD	20
2.4.4	Modelos SPN	20
2.4.4.1	Componente Simples	21
2.4.4.2	Modelo de Componente Simples considerando Primeiro e Segundo <i>Moment Matching</i>	23
2.4.4.3	<i>Cold standby</i>	25
2.4.4.4	Falha em modo comum.	26
2.4.4.5	Componente de Agregação	27
2.4.4.6	Composição do modelos	27
2.5	Considerações Finais	29
3	Visão geral da Ferramenta	30
3.1	Arquitetura da ferramenta	30
3.2	Ambiente de avaliação de infraestruturas de potência <i>data center</i> .	33
3.3	Ambiente de avaliação de infraestruturas de resfriamento <i>data center</i> .	34
3.4	Ambiente de avaliação RBD	36

3.5	Ambiente de Avaliação SPN	37
3.6	Considerações finais.	37
4	Ambiente de Avaliação SPN	39
4.1	Descrição das funcionalidades.	39
4.1.1	Simulação estacionária.	40
4.1.2	Simulação transiente.	42
4.1.3	Experimentação de cenários.	44
4.2	Estrutura do ambiente SPN	45
4.2.1	Interface gráfica	45
4.2.2	Gerenciador de simulação.	49
4.2.3	Simulador SPN.	50
4.2.3.1	Esquema de simulação	50
4.2.3.2	Avaliador de expressões.	52
4.2.3.3	Avaliador de critério de parada	53
4.2.3.4	Removedor de período transiente.	54
4.2.3.5	Analisador de estatísticas.	56
4.3	Considerações Finais.	57
5	Ambiente de Avaliação RBD	58
5.1	Funcionalidades do ambiente RBD.	59
5.1.1	Avaliação de métricas de dependabilidade.	59
5.1.2	Experimentação de cenários.	60
5.1.3	Avaliação de importância dos componentes.	61
5.1.4	Avaliação de dependabilidade por limites.	62
5.1.5	Geração de expressões lógicas e estruturais.	63
5.2	Estrutura do ambiente RBD.	64
5.2.1	Interface gráfica.	64
5.2.2	Gerenciador de avaliação RBD.	67
5.2.3	Avaliador de importância.	67
5.2.4	Núcleo de avaliação e Avaliador RBD padrão.	67
5.2.5	Avaliador RBD por limites.	67
5.3	Considerações finais	68
6	Ambientes de Avaliação <i>Data Center</i>	69
6.1	Ambiente de avaliação da infraestrutura energética para sistemas data center.	69

6.2	Ambiente de resfriamento para sistemas data center.	71
6.3	Funcionalidades dos ambientes <i>data center</i> .	71
6.4	Estrutura dos Ambientes de Avaliação <i>Data Center</i> .	73
6.4.1	Editor Gráfico.	74
6.4.2	Tradutor SPN/RBD.	75
6.4.3	Avaliador de Sustentabilidade.	78
6.4.4	Biblioteca de Componentes.	78
6.5	Considerações Finais.	79
7	Estudos de caso.	80
7.1	Estudo de caso I	80
7.1.1	Descrição das arquiteturas.	80
7.1.2	Resultados	82
7.2	Estudo de Caso II	84
7.2.1	Descrição das Arquiteturas.	85
7.2.2	Resultados	87
7.3	Considerações Finais.	90
8	Conclusão	92
8.1	Limitações e trabalhos futuros	93

Lista de Figuras

2.1	Arquitetura Data Center	8
2.2	a) Sistema exemplo; b) Potência nominal; c) Potência atual.	13
2.3	Esquema de fluxo energético.	15
2.4	Exemplo RBD	18
2.5	Modelo do Componente Simples	22
2.6	Componente simples com uma falha representada por distribuição Erlang	23
2.7	Modelo Cold standby.	26
2.8	Exemplo de modo de falha comum.	26
2.9	Modelo do componente lógico simples	27
2.10	Composição em série de dois componentes básicos não-redundantes	28
3.1	ASTRO - Funcionalidades	31
3.2	ASTRO - Ambiente do sistema de potência <i>data center</i> .	32
3.3	ASTRO - Estrutura dos Componentes internos.	32
3.4	Interface gráfica do ambiente de potência.	34
3.5	Conversão entre modelos	35
3.6	Ambiente de resfriamento.	35
3.7	Conversão no ambiente de resfriamento.	36
3.8	Ambiente RBD na ferramenta ASTRO	37
3.9	Exemplo de Resultados transientes na ferramenta ASTRO. (Tempo X Confiabilidade)	38
4.1	Simulação estacionária.	41
4.2	Resultado da simulação transiente	42
4.3	Parâmetros da simulação transiente	43
4.4	Experimentação de cenários.	44
4.5	Resultado de experimentação SPN.	45
4.6	Estrutura ambiente SPN	46
4.7	Ambiente SPN	46
4.8	Componentes Gráficos do Ambiente SPN	47
4.9	Funcionamento do gerenciador de simulação.	50
4.10	Método das médias em lotes. (Tempo de simulação X valor da métrica)	55
5.1	Interface gráfica do ambiente RBD	58
5.2	Exemplo de avaliação RBD.	60

5.3	Experimentação de cenários.	61
5.4	Importância de confiabilidade	62
5.5	Exemplo de resultado da avaliação de dependabilidade por limites.	63
5.6	Exemplo de modelo RBD e suas respectivas funções lógicas e estruturais.	64
5.7	Principais componentes do ambiente RBD.	65
5.8	Interface RBD - Área de desenho.	65
5.9	Interface RBD - Área de rótulos.	66
5.10	Interface RBD - Árvore estrutural.	66
5.11	Interface RBD - Área de menus.	67
6.1	Conversão de modelos do ambiente de potência para SPN/RBD.	70
6.2	Interface gráfica do ambiente de resfriamento.	71
6.3	Conversão de modelos do ambiente de resfriamento para SPN/RBD.	72
6.4	Parâmetros globais na avaliação de sustentabilidade.	73
6.5	Parâmetros para cada componente do editor energético.	74
6.6	Resultado de exemplo para avaliação de sustentabilidade em estrutura <i>data center</i> .	75
6.7	Estrutura dos ambientes energéticos.	75
6.8	Ambiente de avaliação da estrutura de potência <i>data center</i>	76
6.9	Tradução do modelo energético para RBD	77
6.10	Geração da expressão de falha do modelo SPN a partir do modelo RBD.	78
7.1	Infra-estruturas de potência <i>data center</i> .	81
7.2	A1 (a), modelo RBD Respectivo (b) modelo SPN respectivo (c).	82
7.3	Comparação: disponibilidade, consumo de exergia da vida e TCO.	85
7.4	Arquitetura 1.	86
7.5	Arquitetura 2.	87
7.6	Arquitetura 3.	88
7.7	Arquitetura 4.	89
7.8	Arquitetura 5.	90
7.9	Resultados de potência: comparando Exergia, disponibilidade e TCO.	91
7.10	Resultados de potência e resfriamento: comparando Exergia, disponibilidade e TCO.	91

Lista de Tabelas

1.1	Comparação das características entre as ferramentas relacionadas.	6
2.1	Equações de energia de saída para diferentes dispositivos.	15
2.2	Atributos das transições do componente simples.	22
2.3	As condições que representam os estados do componente simples.	22
2.4	Atributos da transição do componente simples com falha representada por distribuição Erlang	24
2.5	Estados do componente simples com falha representada por distribuição Erlang.	24
2.6	modelo Cold standby. - atributos das Transições.	25
2.7	Atributos do componente de agregação.	27
2.8	Expressões de guarda das transições de uma composição em série.	28
7.1	Expressões de guarda para o modelo A1	82
7.2	Resultados para todas as arquiteturas.	83
7.3	Comparação dos resultados utilizando diferentes ferramentas.	83
7.4	Importância dos componentes.	84
7.5	Parâmetros da infra-estrutura de potência	87
7.6	Parâmetros da infra-estrutura de resfriamento	88
7.7	Resumo dos resultados	88

1

Introdução

Este capítulo apresenta a contextualização do trabalho destacando os principais aspectos que serão abordados ao longo da dissertação, mostrando a importância do trabalho em termos de dependabilidade e sustentabilidade. Em seguida, são apresentados os objetivos do ambiente proposto. Por fim, são apresentados os principais trabalhos relacionados.

1.1 Motivação.

Os *data centers* têm suas raízes nos grandes *Mainframes* do início da indústria de computação. Os primeiros sistemas de computação eram complexos para operar e manter, e, assim como os *data centers* atuais, exigiam um ambiente especial para o seu funcionamento. Muitos foram os recursos e métodos criados para acomodar e organizar os *data centers*, tais como *racks*, para montar os equipamentos, pisos elevados, e bandejas de cabos [8].

Com o passar do tempo, houve um crescimento significativo no número e tamanho dos *data centers*, principalmente, devido à adoção de *cloud computing* [57] como plataforma para novas aplicações baseadas na web. Assim, em empresas que dependem fortemente da internet para as suas operações, interrupções de serviço podem ser muito caras, facilmente chegando a custar milhões de dólares por horas [45].

Para atender aos requisitos de alta disponibilidade para serviços de internet, altos investimentos devem ser aplicados, incluindo novos equipamentos para fornecer redundância. No entanto, o projeto de uma infraestrutura que adota os mais sofisticados equipamentos incorre em um grande custo [45]. Dependendo do contexto, é melhor aplicar redundância utilizando equipamentos mais baratos, o que pode resultar em uma maior disponibilidade do que utilizar um único dispositivo poderoso e caro [9]. Este princípio

é amplamente praticado na indústria, em empresas que utilizam arquiteturas de hardware redundantes para mascarar ou detectar falhas e aumentar a disponibilidade de serviços. Portanto, os projetistas precisam verificar vários fatores para encontrar uma solução viável no projeto de infraestruturas *data center*, considerando custos e métricas de dependabilidade.

Nesse contexto, é importante ressaltar que o apoio de ferramentas de software é muito importante para obter métricas de dependabilidade, tendo em vista que não é trivial avaliar sistemas complexos sem o uso de ferramentas especializadas. Técnicas de modelagem, com uma forte base matemática, como *Stochastic Petri Nets* (SPN) [44] e *Reliability Block Diagrams* (RBD) [47] podem ser adotadas para avaliar a confiabilidade de tais infraestruturas. Vale ressaltar que um projetista de infraestruturas pode estar familiarizado com uma determinada técnica de modelagem, tal qual Diagramas de Blocos de Confiabilidade (RBD), mas pode não ter o mesmo conhecimento em outras técnicas de modelagem, por exemplo, Redes de Petri Estocásticas (SPN). Além disso, dependendo do contexto uma determinada técnica de modelagem pode ser mais apropriada que outra. Assim, os instrumentos de avaliação da dependabilidade devem fornecer uma gama de diferentes técnicas de avaliação que são escolhidas de forma adequada para cada problema.

Sabe-se que à luz do aquecimento global, o esgotamento dos combustíveis fósseis e o aumento dos custos de energia, trouxeram questões de sustentabilidade em foco, e, como mencionado anteriormente, uma forma de aumentar a disponibilidade de sistemas é através da adição de redundância. No entanto, uma vez que a redundância leva a utilização adicional de recursos e energia, seu uso geralmente deverá ter um impacto negativo sobre a sustentabilidade. Assim, além de uma análise de dependabilidade e custos, uma avaliação de sustentabilidade também é importante para determinar uma arquitetura viável para a construção de infraestruturas *data center*.

infraestruturas *data center* sustentáveis são aquelas construídas usando um quantidade estritamente necessária de materiais e energia ao longo da sua vida [14]. De fato, dois *data centers* com disponibilidade semelhantes e diferentes arquiteturas podem ter impactos em termos de sustentabilidade muito diferentes.

Dado que uma avaliação de sustentabilidade precisa ser realizada para construção de ambientes *data center* sustentáveis, precisamos escolher qual métrica de sustentabilidade utilizar. Indicadores de sustentabilidade variam conforme o tipo de sistemas que monitoram, um indicador eficaz deve ser baseado em dados acessíveis, deve ser relevante, de fácil compreensão e de confiança [41].

Métricas importantes para comparar equipamentos são: consumo de energia, utilização de material, impacto ambiental e prejuízos naturais irreversíveis para as próximas gerações, tanto na produção como na fase operacional. Portanto, um determinado equipamento pode ser considerado a melhor opção, mesmo tendo um maior consumo de energia, uma vez que teve um menor impacto negativo em termos de sustentabilidade. Como um índice para avaliação de sustentabilidade, a exergia é uma métrica que pode ser utilizada. Exergia é definida como a fração máxima de energia que poderia ser, teoricamente, convertida em trabalho útil [24]. Por exemplo, a exergia contida em um kJ de petróleo é maior que a exergia contida em um kJ de água morna. Portanto, estratégias para lidar com questões conflitantes, tais como elevadas disponibilidades, altos custos econômicos e ambientais, são fatores primordiais para infraestruturas *data center* sustentáveis. Neste contexto, *Life-cycle assessment* (LCA) é uma abordagem importante para quantificar a sustentabilidade ambiental em termos de consumo de exergia [24].

Tendo em vista os fatores mencionados acima, a ferramenta ASTRO é apresentada para apoiar a avaliação de dependabilidade e sustentabilidade de infraestruturas *data center*, adotando um modelo híbrido de dependabilidade que permite a utilização de RBD, SPN e modelos de alto-nível para avaliação de sistemas *data center*. ASTRO se propõe a utilizar um método baseado na avaliação *Life-cycle assessment* (LCA) [15] para quantificar o impacto da sustentabilidade relacionado a infraestruturas *data center* redundantes, considerando questões da confiabilidade e custos econômicos.

1.2 Objetivos

Esta seção descreve os objetivos do trabalho, destacando as soluções apresentadas pela ferramenta para solucionar cada um deles. Os principais objetivos da ferramenta são:

- **Avaliar dependabilidade em sistemas *data center***

A ferramenta ASTRO apresenta dois ambientes especiais que permitem aos usuários não especializados em modelagem avaliarem estruturas de potência e resfriamento *data center* em termos de dependabilidade.

- **Avaliar sustentabilidade em sistemas *data center*.**

Além da avaliação de dependabilidade, a ferramenta proposta permite a avaliação do sistema *data center* em termos de sustentabilidade, permitindo avaliar não ape-

nas a eficácia do *data center*, mas também os impactos da utilização da infraestrutura para as gerações futuras.

- **Permitir que diferentes tipos de modelagem sejam utilizados.**

A ferramenta ASTRO apresenta dois tipos de ambientes, um ambiente RBD e outro SPN. O ambiente RBD permite a avaliação de dependabilidade utilizando equações de forma fechada, o que deixa a tarefa de avaliação mais rápida. O ambiente SPN apesar de geralmente demorar um pouco mais para apresentar os resultados, aceita modelos baseados em estados, permitindo que comportamentos dependentes sejam avaliados.

- **Permitir avaliação de sistemas genéricos (não necessariamente *data center*).**

Apesar da ferramenta possuir ambientes alto nível voltados para ambientes *data center*, os ambientes RBD e SPN permitem avaliação de dependabilidade em sistemas genéricos, sejam eles *data center* ou não.

1.3 Trabalhos relacionados

Uma boa definição de desenvolvimento sustentável é fornecido pela Comissão Brundland [1] que define sustentabilidade como: “Conseguir atender às necessidades do presente sem comprometer a capacidade das gerações futuras satisfazerem suas próprias necessidades”. Este conceito é fácil de entender, mas a sua quantificação não é fácil.

Nos últimos anos, alguns trabalhos têm sido desenvolvidos para avaliar a infraestrutura dos *data centers* e um subconjunto deles também analisou o impacto sobre a sustentabilidade. [59] propõe uma anatomia de risco para detectar um ponto único de falha nos sistemas de distribuição de energia dos *data center*, bem como um mecanismo para melhorar os investimentos com níveis de risco aceitáveis. Em outras palavras, o trabalho propõe uma técnica para identificar os gargalos do sistema de energia e explica como adotar melhorias nesse sistema de forma ótima, considerando um determinado nível de risco. Em [51], os autores propõem uma plataforma para avaliação de *data centers* inteligentes, levando em consideração aspectos como: refrigeração, estrutura de fornecimento de energia e componentes de TI. *Data center* inteligente é uma solução onde cada um dos subsistemas do *data center* é concebido com suporte à operação inteligente, ou seja, utilizando um controle adaptativo, a fim de permitir maior eficiência e gerenciamento aprimorado dos seus recursos. Um coeficiente de desempenho que inclui os componentes é proposto para medir a eficiência global do fluxo de energia.

Wiboonrat [58] adota uma abordagem baseada em diagramas de blocos de confiabilidade (RBD) e simulação Monte Carlo para comparar diferentes topologias *data center*. Em [46], Patterson avalia o impacto da temperatura sobre a eficiência energética. A análise realizada indica a existência de uma temperatura ideal para a operação dos *data centers*. [34] descreve as oportunidades de integração entre os sistemas de resfriamento, sistemas de aquecimento e de fornecimento de energia. Os autores demonstram que os atributos de segurança, tais como confiabilidade, se beneficiam de uma abordagem integrada.

Existem algumas ferramentas acadêmicas e comerciais para a avaliação da dependabilidade de sistemas em geral. BlockSim [4] permite a modelagem do sistema usando RBD e *Fault Trees* [53], ou mesmo uma combinação das duas técnicas de modelagem, mas não utilizam modelos baseados em estados, dificultando o uso de modelos que possuam dependências. Relex [5] e a ferramenta Sharpe [48] fornecem avaliação de dependabilidade através da adoção de RBD, árvores de falha e modelos SPN, assim como a ferramenta proposta. No entanto, esses softwares obtêm métricas de confiabilidade apenas por análise numérica, este método não é adequado quando o sistema modelado é muito complexo [11]. A ferramenta TimeNET [31] apresenta um ambiente para avaliação de desempenho, ela também pode ser utilizada para avaliação de métricas de dependabilidade utilizando modelos SPN, porém não suporta modelos RBD. Da mesma forma que o TimeNET, a ferramenta GreatSPN [21] apresenta um ambiente para avaliação de desempenho para propósito geral que pode ser utilizada para avaliação de dependabilidade, ela suporta avaliações utilizando modelos baseados em cadeias de Markov e simulação SPN. Porém, assim como o TimeNET, não modela RBD.

Diferente dos trabalhos anteriores, este trabalho propõe um ambiente integrado para a quantificação do impacto da sustentabilidade com base na avaliação do ciclo de vida (LCA) [15] no contexto do projeto do *data center*, considerando questões de dependabilidade, sustentabilidade e custos econômicos. ASTRO também permite uma modelagem híbrida, no sentido de que RBD e SPN podem ser adotados sempre que sejam adequados. A Tabela 1.1 mostra um comparativo entre as principais ferramentas que permitem a avaliação de dependabilidade e suas características.

Apesar dessas ferramentas tratarem de dependabilidade, nenhuma delas lida com questões de sustentabilidade e dependabilidade em um único ambiente. A ferramenta proposta utiliza um modelo híbrido que utiliza avaliação RBD e simulação SPN para o cálculo de métricas de dependabilidade, visando o contexto de *data centers*. A ferramenta proposta permite a modelagem de sistemas genéricos, ou seja, que não se re-

stringem apenas ao domínio de *data centers*.

Além disso, a ferramenta é proposta para a quantificação do impacto da sustentabilidade com base na avaliação do *Life-Cycle Assessment (LCA)* [15] relacionado com infraestruturas redundantes em *data centers*, considerando questões de dependabilidade e custos econômicos.

Tabela 1.1: Comparação das características entre as ferramentas relacionadas.

Ferramenta	Fórmulas Fechadas RBD	Simulação SPN	Suporte RBD e SPN	Interface Alto nível	Fórmulas Fechadas para Avaliação de Sustentabilidade	Avaliação Numérica SPN
Astro	X	X	X	X	X	
Sharpe	X		X			X
Relex	X					
BlockSim	X					
TimeNET		X				X
GreatSPN		X				X

1.4 Estrutura da dissertação

A dissertação é organizada da seguinte forma: O Capítulo 1 apresenta a introdução do trabalho mostrando seus principais objetivos. O Capítulo 2 mostra os conceitos fundamentais do trabalho. O Capítulo 3 apresenta a arquitetura da ferramenta proposta. O Capítulo 4 apresenta o editor SPN, suas funcionalidades e estrutura. O Capítulo 5 apresenta o editor RBD. O Capítulo 6 mostra os editores voltados para o ambiente *data center*. O Capítulo 7 apresenta os estudos de caso e o Capítulo 8 apresenta a conclusão.

2

Conceitos Básicos

O capítulo ora exposto apresenta os principais conceitos da dissertação em comento. Primeiramente, é apresentada uma visão geral dos sistemas data centers. Posteriormente, abordam-se os conceitos de dependabilidade. Em seguida, introduz-se conceitos relacionados à sustentabilidade. Por fim, são apresentados os modelos utilizados neste trabalho para avaliação de dependabilidade e sustentabilidade.

2.1 infraestrutura *Data Center*

Um *data center*, é o local onde são concentrados os computadores e *softwares* responsáveis pelo processamento de dados de uma empresa ou organização [16]. Os sistemas *data center* (Figura 2.1) possuem os seguintes subsistemas: (i) infraestrutura de TI, (ii) infraestrutura de refrigeração, e (iii) infraestrutura de energia.

A infraestrutura de TI é composta por três componentes principais - servidores, rede e dispositivos de armazenamento. Servidores de virtualização podem ser utilizados para melhorar o compartilhamento dos recursos de hardware e facilitar migração. Os dispositivos de armazenamento (por exemplo, matrizes de armazenamento) são tipicamente ligados através de uma *Storage Area Network* (SAN). Os servidores também podem se conectar a sistemas de arquivos remotos usando *Network Attached Storage* (NAS) através de Ethernet. A infraestrutura de rede é normalmente possui três camadas [13]: uma camada de acesso composto de switches para conexão com servidores, um camada de agregação com switches para conectividade dentro do centro de dados, e uma camada central de roteadores que ligam o *data center* à internet. Os serviços de software são normalmente organizados em uma arquitetura multi-camadas com camadas separadas para servidores web, servidores de aplicações e banco de dados.

A infraestrutura de refrigeração do *data center* é responsável pela refrigeração da

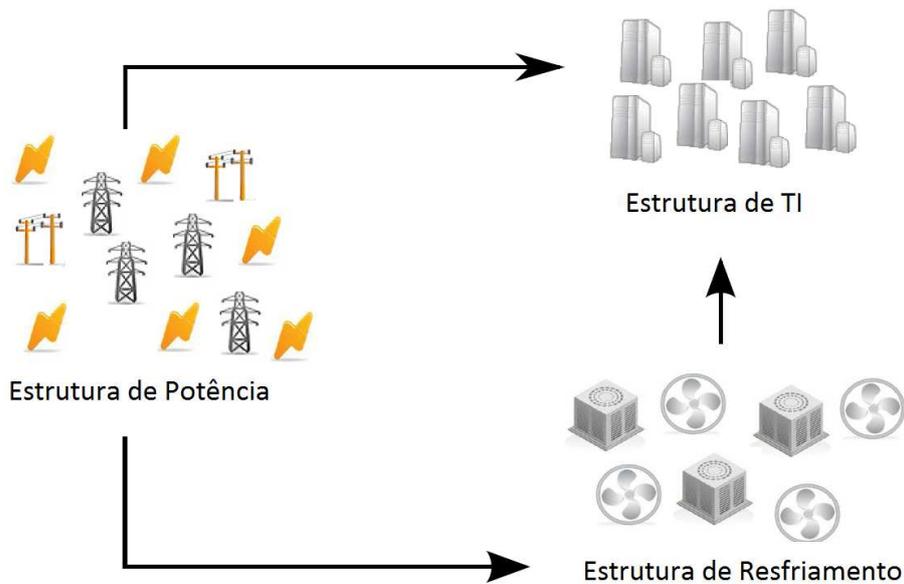


Figura 2.1: Arquitetura Data Center

infraestrutura de TI, de modo a evitar o superaquecimento. Normalmente, vários *Computer Room Air Conditioning* (CRAC) transferem o calor dissipado por dispositivos de TI para um sistema de distribuição de água refrigerada. Os *Chillers*, tipos especiais de compressores, extraem o calor do sistema de água refrigerada e o transfere para o ambiente através de torres de resfriamento ou transferidores de calor. A infraestrutura de refrigeração pode ser responsável por 10-50 % do consumo total de energia do *data center* [16].

A infraestrutura elétrica [28] é responsável pelo fornecimento ininterrupto de energia condicionado à tensão e frequência correta para o equipamento de TI hospedado em racks de *data center*. Da concessionária de energia elétrica, a potência, normalmente, passa por transformadores, chaves de transferência, e *Uninterruptible Power Supplies* (UPS), unidades de distribuição de energia (PDU) e, finalmente, filtros de linha. Tolerância a falhas está diretamente relacionada a replicação de recursos ou replicação de tempo, portanto, espera-se um consumo maior de energia com o uso da mesma. As UPS's fornecem energia condicionada e backup em caso de falhas de curta duração. Geradores ou outras fontes de energia local devem ser usadas para cortes mais longos, ou mesmo para auxiliar a demanda total de energia em regime regular. A redundância é uma questão essencial nas infraestruturas de potência, dado que a sua disponibilidade afeta os serviços do sistema global.

2.2 Dependabilidade

A dependabilidade de um sistema pode ser entendida como a habilidade de disponibilizar um conjunto de serviços com um determinado nível de confiança [55]. Na verdade, a dependabilidade está também relacionada com conceitos como tolerância a falhas e confiabilidade. Sem perda de generalidade, a confiabilidade é a probabilidade de que o sistema fornecerá um conjunto de serviços durante um determinado período de tempo [36]. Em sistemas tolerantes a falhas, a confiabilidade fornece a probabilidade de funcionamento mesmo quando há componentes com defeito. A disponibilidade é também um outro conceito importante, que quantifica o efeito misto dos processos de falhas e reparo em um sistema. Em geral, a disponibilidade e confiabilidade são conceitos relacionados, mas diferem no sentido de que o primeiro pode considerar a manutenção de componentes que falharam [26].

Essas métricas podem ser calculados através de modelos combinatórios, como RBD, ou os modelos baseados em estados, por exemplo, SPN (só para citar os modelos adotados neste trabalho). RBDs permitem representar redes de componentes e estabelecer equações de forma fechada. No entanto, modelos RBD apresentam inconvenientes para manipulação de dependências de falhas e reparação que são freqüentemente encontrados na representação de políticas de manutenção e mecanismos redundantes. Especialmente aquelas baseadas em métodos de redundância dinâmica. Também deve ser salientado que o RBD poderia ser avaliado através de simulação, sempre que distribuições mais complexas do que a exponencial são adotadas.

Por outro lado, os métodos baseados em estados podem facilmente considerar dependências, permitindo assim a representação de mecanismos complexos redundantes, bem como políticas de manutenção sofisticadas. No entanto, esses métodos sofrem com a explosão de estados [39]. Alguns desses formalismos permitem tanto a análise numérica e simulação estocástica, e SPN é um dos modelos mais importantes dessa classe.

2.2.1 Equações de dependabilidade

Se um projetista, por exemplo, estiver interessado em calcular a disponibilidade (A) de determinado dispositivo ou sistema. Ele precisa do tempo até a falha (TTF) e tempo de reparo (TTR). Considerando-se que o tempo de atividade e inatividade não estão disponíveis, a opção mais acessível é a média. Se o avaliador precisar apenas do valor médio, as métricas comumente adotadas são o tempo médio de falha ($MTTF$) e Tempo

Méio para Reparo ($MTTR$) (outros valores centrais podem também ser adotados). No entanto, se também está interessado na variação da disponibilidade, o desvio padrão do tempo até a falha ($sd(TTF)$), e seu respectivo desvio padrão do tempo de reparo ($sd(TTR)$) permitem uma estimativa da variação da disponibilidade.

A disponibilidade (A) é obtida pela análise ou simulação do estado estacionário. Ela é dada pela seguinte equação:

$$A = \frac{MTTF}{MTTF + MTTR} \quad (2.1)$$

Outro conceito útil é a disponibilidade instantânea, que indica a disponibilidade para um determinado instante (t), definida por:

$$A(t) = \frac{\mu}{\mu + \lambda} + \frac{\lambda}{\mu + \lambda} e^{-(\mu + \lambda)t} \quad (2.2)$$

Onde μ indica a taxa de reparo ($1/MTTR$), e λ corresponde à taxa de falha ($1/MTTF$). Considera-se que o tempo de falha e de reparo seguem distribuições exponenciais.

Assuma $f(t)$ uma função de densidade de probabilidade. Assim, através da análise ou simulação transiente, a confiabilidade (R) é obtida. Então, o $MTTF$ pode ser calculado como o desvio padrão do tempo até a falha (TTF):

$$MTTF = \int_0^{\infty} R(t)dt = \int_0^{\infty} t f(t)dt \quad (2.3)$$

$$sd(TTF) = \sqrt{\int_0^{\infty} t^2 f(t)dt - (MTTF)^2} \quad (2.4)$$

Deve-se ter em mente que, para a computação da confiabilidade de um determinado serviço, ou sistema, a atividade de reparação do respectivo serviço não deve ser representada. Além disso, considerando o cálculo da indisponibilidade, dada por $UA = 1 - A$, e a Equação 2.2, a seguinte equação é derivada:

$$MTTR = MTTF \times \frac{UA}{A} \quad (2.5)$$

assim, o desvio padrão do tempo de reparação (TTR) pode ser calculado como segue:

$$sd(TTR) = sd(TTF) \times \frac{UA}{A} \quad (2.6)$$

2.3 Sustentabilidade

Dentre os indicadores de sustentabilidade, a Avaliação do Ciclo de Vida (LCA) [15, 2] é uma abordagem bem comum para quantificação do impacto da utilização de um determinado sistema ou serviço. Neste trabalho, adotamos uma abordagem baseada em LCA para estimar o impacto sobre a sustentabilidade de um equipamento em termos de consumo de exergia (ver Seção 2.3.1).

Em tal abordagem, um inventário de diferentes materiais e processos envolvidos no ciclo de vida do sistema - desde a extração de matérias-primas, fabricação, operação, até o fim de vida - é compilado. Esse inventário é, então, traduzido em diferentes impactos ambientais através de um conjunto de fatores de impacto padronizados ou pré-determinados. Por exemplo, conhecendo o uso total de electricidade em kWh em todo o ciclo de vida (inventário) e a composição média da composição de energia elétrica em termos de kg de CO₂ equivalente por kWh (fator de impacto), uma estimativa do potencial de aquecimento global (Impacto ambiental), em kg de CO₂ equivalente pode ser calculada.

Coletar os dados necessários para a elaboração de inventário de uma LCA detalhado baseado em processo pode ser demorado e dispendioso, devido à grande quantidade de informações necessárias e à dificuldade de obter essas informações através de cadeias de suprimentos complexas e diversificadas. Abordagens baseadas em LCA - como LCA racionalizado [32] e *economic input-output*(EIO) LCA [33] - têm sido desenvolvidas. No presente trabalho, utilizamos uma abordagem híbrida baseada em EIO-LCA para quantificar o impacto ambiental em termos de consumo de exergia. A abordagem adotada é semelhante à utilizada por Shah et al [49, 50], pois essencialmente divide o ciclo de vida em duas fases: (1) fase embutida, que envolve todos os impactos relacionados às decisões de projeto do produto (incluindo a extração de material, fabricação e impactos da cadeia de abastecimento, bem como descarte) e, (2) fase operacional, que envolve todos os impactos referentes a decisões durante o uso do produto (como os ciclos de operação e manutenção). Para a fase embutida, um modelo de EIO-LCA foi desenvolvido para quantificar os impactos ambientais em termos de composições setoriais médias como uma função da atividade econômica trans-setorial. Para a fase operacional, um modelo detalhado operacional é construído levando-se em conta os ganhos de eficiência do dispositivo individual, disponibilidade e consumo energético.

2.3.1 Exergia

A exergia foi escolhida para quantificar o impacto ambiental em termos de métricas termodinâmicas (também chamada de energia disponível para utilização), ela representa a parcela máxima teórica da energia que poderia ser convertida em trabalho. Esta métrica foi escolhida porque apresenta uma visão unificada do total do material e da quantidade da energia envolvidos na fabricação ou operação de um determinado sistema.

Por exemplo, a exergia contida em um kJ de óleo é maior do que a exergia contida em um kJ de água morna. O óleo pode ser usado facilmente para mover um automóvel, mas a água quente não. Tal situação representa uma maior energia dispensada pela natureza na produção do óleo. Assim, exergia é o verdadeiro potencial de cada tipo possível de manifestação de energia. A utilização da exergia quantifica tanto a quantidade como a qualidade da energia e dos materiais utilizados em um equipamento específico.

Apesar de exergia ser um importante índice para avaliação, sua adoção deve ser feita criteriosamente. Por exemplo, a exergia da radiação solar é alta - uma vez que, teoricamente, pode ser convertida em trabalho útil com uma alta eficiência - mas o impacto da sua utilização é pequeno. Portanto, seu consumo não é tão prejudicial quanto a utilização de outras fontes de alta exergia. Outra característica do saldo exergético é recorrente da comparação com o processo mais eficiente teoricamente possível. Às vezes, não há alternativa tecnológica para apresentar um determinado consumo exergético. Apesar destas imperfeições, o consumo de exergia é um importante índice para expressar de uma forma simples tanto o consumo energético quanto o impacto ambiental de um equipamento. Vários trabalhos [52, 60, 22] apresentam resultados e conclusões importantes sobre a utilização deste índice como critério para comparar diferentes projetos tecnológicos.

A próxima seção descreve os detalhes sobre o modelo de sustentabilidade adotado. O ambiente ASTRO computa apenas a exergia associada com a energia utilizada durante a fabricação e as fases operacionais, como uma primeira estimativa.

2.4 Modelos adotados

Esta seção apresenta os modelos adotados pelo ambiente ASTRO para quantificação de sustentabilidade em sistemas *data center*. Inicialmente, o modelo energético é apresentado e em seguida, o modelo de sustentabilidade é detalhado. O mesmo é representado pelas equações para estimar o consumo de exergia embutida e operacional. Em seguida, os modelos de dependabilidade são descritos: eles são representados por

turas energética e de refrigeração para atender à demanda do sistema em termos de energia elétrica e carga térmica, admitindo que todos os componentes estão funcionando corretamente no início da operação do sistema.

Seja G uma tupla, que representa o modelo energético, representado a seguir:

$$G = (N, V, w, f_p, f_\eta, f_e, s, t)$$

Onde:

- N representa o conjunto de nós. ;
- $V \subseteq (N \times N)$ denota o conjunto de arestas.;
- $w : V \rightarrow \mathbf{R}$ denota os pesos das arestas.;
- $f_p : N \rightarrow \mathbf{R}^+$ corresponde a função que relaciona cada nó com sua respectiva potência máxima;
- $f_\eta : N \rightarrow \{k \in \mathbf{R} | 0 \leq k \leq 1\}$ é a função que relaciona cada nó com sua eficiência;
- $f_e : N \rightarrow \{0, 1\}$ é a função que indica o estado de cada nó do modelo;
- $s \in N$ é o nó fonte ou nó raiz.;
- $t \in N$ é o nó de destino.;

A Figura 2.3 mostra um componente (a) que recebe energia de um conjunto de n elementos. A Equação 2.7 representa a fração da potência que (a) recebe partir de um determinado componente (j).

$$p_{aj} = \frac{p_{max_j} \times w_{ja} \times e_j}{\sum_{i=1}^n p_{max_i} \times w_{ia} \times e_i} \quad (2.7)$$

Onde p_{aj} representa a potência que a recebe do j -ésimo componente, p_{max_j} é a potência máxima do j -ésimo componente, w_j é o peso do arco a partir de j para a ; e_j é o estado do componente. Se a quantidade de energia que o dispositivo está fornecendo é maior que sua capacidade máxima, então dizemos que esse componente não está operacional. Um modelo de energia é operacional se o nó destino está operacional.

A energia que cada componente consome depende da quantidade de trabalho que cada componente realiza (potência atual) e do tipo do componente. A Tabela 2.1 apresenta as equações de consumo de energia para cada tipo de componente considerado nesse trabalho. Nestas equações, η representa a eficiência do equipamento de acordo com a segunda lei da termodinâmica; Q_{IN} é a energia térmica; Q_F é a energia térmica de fluidos CRACs; COP é o coeficiente de desempenho; Q_q corresponde à carga térmica que flui para a torre de resfriamento; $\mu = \frac{PotenciaMaximaResfriamento}{ConsumoMaximoPotencia}$.

O leitor interessado deve consultar [42][19] para mais detalhes sobre a estrutura e o funcionamento das máquinas térmicas.

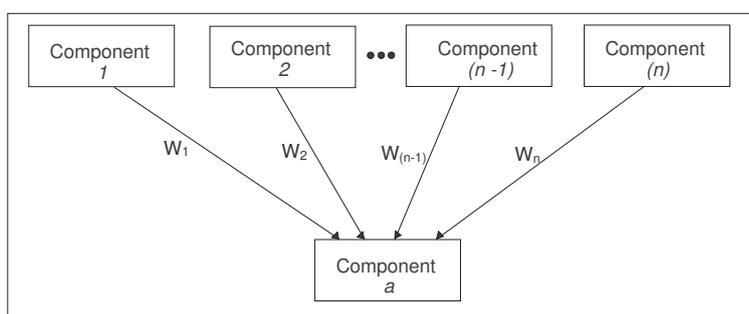


Figura 2.3: Esquema de fluxo energético.

Tabela 2.1: Equações de energia de saída para diferentes dispositivos.

Dispositivo	Consumo de Energia
Elétrico	$\frac{potenciaAtual}{\eta}$
CRAC	$\frac{1}{\eta} \times Q_{IN}$
Chiller	$Q_F \times \left(1 + \frac{1}{COP}\right)$
Torre Resfriamento	$Q_q \times \left(1 + \frac{1}{\mu}\right)$

2.4.2 Modelos de Sustentabilidade

ASTRO quantifica o impacto ambiental em termos de métricas de exergia (também chamada de energia disponível). Como discutido por Lettieri et al. [38], para a maioria dos sistemas de TI e componentes de infraestrutura relacionados, uma análise do consumo de exergia do ciclo de vida (LCEA) provê um compromisso razoável entre a quantidade de tempo necessário para a análise e precisão relativa da análise. Dependendo dos resultados da LCEA, os modelos mais detalhados podem ser construído para

qualquer uma das fases do ciclo de vida de um determinado sistema. Os parágrafos a seguir apresentam as equações adotadas para estimar os consumos de exergia embutidos (X_{emb}) e operacional (X_{op}).

A exergia embutida é obtida da seguinte forma:

$$X_{emb} = E_{man} \times [Ef_{man} + (1 - Ef_{man}) \times (1 - C_{reuse})] \quad (2.8)$$

Onde E_{man} é a energia necessária para a fabricação de todos os n equipamentos da infraestrutura (veja Equação 2.8); Ef_{man} (a eficiência de fabricação relacionada à segunda lei da termodinâmica) representa a parcela de exergia disponível que foi de fato incorporada aos equipamentos; e C_{reuse} é a fração da exergia que não é usada no próprio equipamento, mas reutilizados em outros processos.

$$E_{man} = \sum_{i=1}^n E_{eq_i} \quad (2.9)$$

No caso específico da energia elétrica, toda a energia pode ser convertida em trabalho, teoricamente, para a variável E_{man} na Equação 2.8 significa a exergia total durante a fase de manufatura. A fração E_{man} da exergia é consumida pois é incorporada nos equipamentos. A fração complementar $(1 - E_{man})$ pode ser reutilizada em outros processos (C_{reuse}) ou ser desperdiçada ou destruída ($1 - C_{reuse}$). Assim, outra maneira de entender a Equação 2.8 é como a soma dos exergia incorporada nos equipamentos mais a exergia perdida ou destruída. Ou, numa terceira via, X_{emb} é a exergia total acessível menos a exergia reutilizada em outros processos.

Neste trabalho, a energia necessária para a fabricação de cada equipamento (E_{eq_i}), foi estimada utilizando EIO-LCA em conjunto com a ferramenta disponível em [6].

Em relação à exergia operacional, a seguinte equação é adotada

$$X_{op} = \frac{K_{amb}}{K_{comp}} \times (1 - Ef_{deliv}) \times P_{in} \times T_{life} \times A \quad (2.10)$$

Em que K_{amb} é a temperatura ambiente; K_{comp} é a temperatura de funcionamento do componente; Ef_{deliv} é a eficiência na entrega em relação a todos os equipamentos (Equação 2.11); P_{in} é o potência total de entrada; T_{life} é o período de interesse (ou seja, o tempo de vida); e A é a disponibilidade da infraestrutura.

$$Ef_{deliv} = \prod_{i=1}^n Ef_{eq_i} \quad (2.11)$$

O consumo de exergia operacional, como indicado pela Equação 2.10, pode ser en-

tendida como a fração do calor dissipado pelos equipamentos que não podem ser, teoricamente, transformada em trabalho útil, ou seja, a destruição de exergia durante a conversão de energia elétrica em energia térmica. Assim, a exergia total do sistema (X_{total}) é a soma das exergias embutida e operacional:

$$X_{total} = X_{emb} + X_{op} \quad (2.12)$$

2.4.2.1 TCO - Custo total da infraestrutura

O custo total de infraestrutura (TCO) é uma técnica que tem como objetivo compreender o verdadeiro custo da compra de um determinado bem ou serviço a partir de um determinado fornecedor [27]. Além do preço de dispositivos, TCO pode incluir elementos tais como logística, custos com pesquisa e qualificação de fornecedores, transporte, recebimento, inspeção, rejeição, substituição, o tempo de inatividade causada por falha, bem como os custos de descarte.

No que diz respeito aos sistemas de TI, TCO pode ser considerado como um modelo abrangente que ajuda as organizações a compreender e gerenciar os custos diretos e indiretos, tendo em conta o ciclo de vida dos dispositivos [23]. Neste trabalho, uma abordagem que considera dispositivos *data center* é adotada para quantificar o TCO. A Equação 2.13 demonstra o cálculo do custo inicial, que considera o preço de varejo dos aparelhos.

$$CustoInicial = \sum_{i=1}^n Eq_{rp_i} \quad (2.13)$$

No qual n é o número de dispositivos e Eq_{rp_i} representa o preço de venda do equipamento i . Adicionalmente, o custo para manter o sistema em operação também é considerado. A Equação 2.14 é adotada para calcular o custo operacional.

$$CustoOperacional = E_{input} \times T \times E_{cost} \times A \quad (2.14)$$

Onde E_{input} é a entrada de energia elétrica; T é que o período adotado; E_{cost} custo corresponde ao preço da unidade de energia; A é a disponibilidade do sistema. O TCO é representado pela soma dos custos iniciais e operacionais, como mostrado na Equação 2.15.

$$TCO = CustoInicial + CustoOperacional \quad (2.15)$$

2.4.3 Modelos RBD

Diagrama de Blocos de Confiabilidade (RBD) é um modelo combinatorial que foi inicialmente proposto como uma técnica para calcular a confiabilidade em sistemas grandes e complexos, utilizando diagramas de blocos intuitivos [55]. Em geral, os diagramas de blocos fornecem uma descrição gráfica dos componentes do sistema e conectores, que podem ser adotados para determinar o estado geral do sistema, dado o estado de seus componentes. A estrutura do RBD estabelece a interação lógica entre os componentes definindo quais combinações de elementos ativos são capazes de sustentar o funcionamento do sistema. Essa técnica também foi estendida para calcular as métricas de dependabilidade, tais como a disponibilidade e manutenibilidade.

Os blocos são geralmente organizadas utilizando as seguinte composições: série, paralelo, ponte, blocos k -out-of- n , ou, ainda, uma combinação das abordagens anteriores.

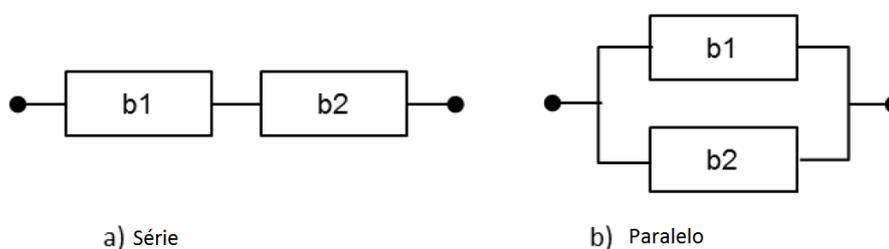


Figura 2.4: Exemplo RBD

A Figura 2.4 mostra dois exemplos, em que os blocos são organizados em série (Figura 2.4(a)) e paralelo (Figura 2.4(b)). No arranjo série, se um único componente falhar, o sistema todo não funciona mais. Assumindo um sistema com n elementos, a confiabilidade é obtido por:

$$R_s(t) = \prod_{i=1}^n R_i(t) \quad (2.16)$$

em que $R_i(t)$ é a confiabilidade do bloco b_i .

Para um arranjo paralelo (ver Figura 2.4(b)), pelo menos um componente deve ser operacional para que todo sistema esteja operacional. Levando-se em conta n componentes, a confiabilidade do sistema é dada por:

$$R_p(t) = 1 - \prod_{i=1}^n (1 - R_i(t)) \quad (2.17)$$

em que $R_i(t)$ é a confiabilidade do bloco b_i . Para outros exemplos e equações de

forma fechada, o leitor deve referir-se à [36].

2.4.3.1 Funções estruturais RBD.

Funções estruturais RBD são funções matemáticas discretas que relacionam o estado de funcionamento dos componentes de um determinado sistema com o estado de funcionamento do sistema. Seja x_i a variável que representa o estado do componente i para $1 \leq i \leq n$. Onde n é o número de componentes do sistema. Então o estado x_i do componente i é dado por:

$$x_i = \begin{cases} 1, & \text{se o componente } i \text{ está funcionando,} \\ 0, & \text{se o componente } i \text{ está falho.} \end{cases} \quad (2.18)$$

O vetor $\mathbf{x} = (x_1, x_2, \dots, x_n)$ representa o estado de todos os componentes. O estado do sistema é determinado pelos estados dos componentes. Seja ϕ a variável que representa o estado do sistema como um todo.

$$\phi = \begin{cases} 1, & \text{se o sistema está funcionando,} \\ 0, & \text{se o sistema está falho.} \end{cases} \quad (2.19)$$

Se o estado de todos os componentes é conhecido, então o estado do sistema também é conhecido. Assim, o estado do sistema é uma função determinística do estado de todos os componentes. Assim,

$$\phi = \phi(\mathbf{x}) = \phi(x_1, x_2, \dots, x_n) \quad (2.20)$$

onde $\phi(\mathbf{x})$ é a função estrutural do sistema.

As regras de formação das funções estruturais são mostradas a seguir:

- **Componentes em série**

Sejam n componentes x_1, x_2, \dots, x_n em série, a função estrutural ϕ desses componentes é representada por:

$$\phi(\mathbf{x}) = \prod_{i=1}^n x_i = \min\{x_1, x_2, x_3, \dots, x_n\}. \quad (2.21)$$

- **Componentes em paralelo**

Sejam n componentes x_1, x_2, \dots, x_n em paralelo, a função estrutural ϕ desses componentes é representada por:

$$\phi(\mathbf{x}) = 1 - \prod_{i=1}^n (1 - x_i) = \max\{x_1, x_2, x_3, \dots, x_n\}. \quad (2.22)$$

2.4.3.2 Funções lógicas RBD

As funções lógicas possuem o mesmo objetivo das funções estruturais: indicar um relacionamento entre os estados dos componentes e o estado do sistema. Porém, em alguns casos não é fácil simplificar funções estruturais para uma forma minimizada [36]. No caso das funções lógicas, técnicas de minimização booleanas podem ser utilizadas para simplificar ou minimizar as funções geradas.

- **Componentes em paralelo**

Seja $\mathbf{x} = (x_1, x_2, \dots, x_n)$ o vetor que representa o estado de n componentes que compõem um determinado sistema. A função lógica (S) que representa o funcionamento do sistema em paralelo é:

$$S_{paralelo}(\mathbf{x}) = (x_1 \vee x_2 \vee \dots \vee x_n) \quad (2.23)$$

- **Componentes em série**

Seja $\mathbf{x} = (x_1, x_2, \dots, x_n)$ o vetor que representa o estado de n componentes que compõem um determinado sistema. A função lógica (S) que representa o funcionamento do sistema em série é:

$$S_{serie}(\mathbf{x}) = (x_1 \wedge x_2 \wedge \dots \wedge x_n) \quad (2.24)$$

O leitor pode verificar mais detalhes sobre funções lógicas e estruturais em [36]. Na Seção 5.1.5, pode-se observar um exemplo de geração de funções lógicas e estruturais.

2.4.4 Modelos SPN

Redes de Petri (PN) [43] são uma família de formalismos muito bem adaptados para vários tipos de modelagens sistemáticas, tais quais sincronização, concorrência, mecanismos de comunicação. Adicionalmente, atrasos determinísticos e probabilísticos são naturalmente representados utilizando redes de Petri temporizadas e estocásticas. Em geral, uma rede de Petri é um grafo bipartido dirigido, em que locais (representados por círculos) denotam estados locais e transições (descrito como retângulos) representam

ações. Arcos (arestas) conectam lugares às transições e vice-versa. Este trabalho adota uma extensão específica das redes de Petri, nomeada, *Stochastic Petri Nets* (SPN) [40], que permite a associação de atrasos probabilísticos às transições usando distribuições exponenciais. Seu respectivo espaço de estados é isomorfo às cadeias de Markov de tempo contínuo [53]. Além disso, SPN permite a adoção de técnicas de simulação para a obtenção de métricas de confiabilidade como uma alternativa à geração de cadeias de Markov. No caso da utilização de simulação, outras distribuições de probabilidade podem ser associadas às transições.

As próximas seções descrevem os blocos propostos em SPN para a obtenção de métricas de confiabilidade, bem como a abordagem para combiná-los. O leitor pode obter detalhes sobre o formalismo das Redes de Petri em [30].

2.4.4.1 Componente Simples

O componente simples é um bloco de construção representativo utilizado pela ferramenta ASTRO, ele é caracterizado pela ausência de redundância, ou seja, o componente pode estar em dois estados, funcionando ou não. Para calcular a sua disponibilidade, o *TTF* e *TTR* devem ser representados. Se ambos *TTF* e *TTR* são exponencialmente distribuídos, o *MTTF* e *MTTR* são os únicos parâmetros necessários para calcular a sua disponibilidade. A confiabilidade também é computado diretamente por $R(t) = \exp\frac{-t}{MTTF}$, considerando que a atividade de reparação não é permitida.

O respectivo modelo SPN do “componente simples” é mostrado na Figura 2.5. Este componente tem dois parâmetros (não representado na figura), são eles o X_MTTF e X_MTTR , que representam os atrasos associados às transições $X_Failure$ e X_Repair respectivamente. Ambas as transições são exponencialmente distribuídas (*exp*) e adotam a política de concorrência servidor único (*ss*) (ou servidor infinito (*is*), dependendo da aplicação). O rótulo “X” é instanciado de acordo com o nome do componente, como por exemplo *Componente_Failure* e *Componente_Repair*. Quando os atrasos associados com as transições X_Repair ou $X_Failure$ são dependentes de marcação, constantes adicionais, como o X_MTTR_i ou X_MTTF_i , são considerados. Quando não associados, o leitor deve assumir que os atrasos de ambas as transições não são dependentes de marcação, bem como a política de concorrência seja servidor único(*ss*).

Tabela 2.2 apresenta os atributos relacionados às transições do modelo de componentes simples.

Os Lugares X_ON e X_OFF são os componentes do modelo que representam os estados de atividade e inatividade, respectivamente. O componente simples também

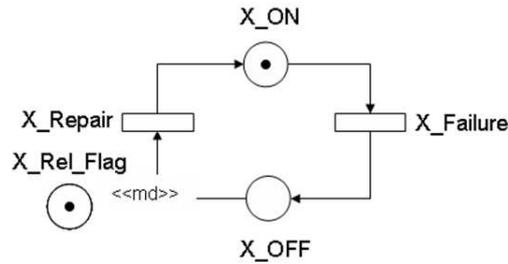


Figura 2.5: Modelo do Componente Simples

inclui um simples arco de X_OFF para X_Repair com multiplicidade dependendo do local de marcação. A multiplicidade é definida através da expressão $IF(\#X_Rel_Flag = 1) \ 2 \ ELSE \ 1$, onde o local de X_Rel_Flag modela a avaliação de confiabilidade ou disponibilidade. Assim, se a condição $\#X_Rel_Flag = 1$ é falsa (sem *token* no lugar X_Rel_Flag), então a avaliação se refere à disponibilidade. Caso contrário, a avaliação é da confiabilidade. Essa abordagem nos permite parametrizar o modelo, pois auxilia a avaliação do sistema, considerando reparação (ou seja, a disponibilidade) ou não (ou seja, confiabilidade).

Tabela 2.2: Atributos das transições do componente simples.

Transição	Tipo	Atraso	Marcação	Concorrência
$X_Failure$	exp	$X_MTTF(_i)$	constant/dependent	SS
X_Repair	exp	$X_MTTR(_i)$	constant/dependent	SS

A Tabela 2.3 mostra as condições que representam os estados operacionais e de falha. Um componente está operacional se o número de *tokens* ($\#$) no lugar X_ON é maior que 0, e falho caso contrário. Assim, se $\#X_Rel_Flag = 1$, $P\{\#X_ON > 0\}$ significa a disponibilidade do componente (avaliação do estado estacionário). Se $\#X_Rel_Flag = 0$, então $P\{\#X_OFF > 0\}$ permite a computação confiabilidade do componente, se a avaliação transitória é efetuada e a marcação inicial é de $\#X_ON = 1$ e $\#X_OFF = 0$. Deve-se ressaltar que o modelo apresentado é limitado (tem apenas dois estados), reversível e vivo [43]. As duas últimas propriedades somente se apresentam se $\#X_Rel_Flag = 0$.

Tabela 2.3: As condições que representam os estados do componente simples.

Estado	Condição
Operacional	$\#X_ON > 0$
Falho	$\#X_ON = 0$

A exergia operacional é representada pela Equação 2.10. Dado que a disponibilidade do componente simples adotado no formalismo SPN é representado por $P\{\#X_ON >$

0} (se $\#X_Rel_Flag = 0$), então a exergia operacional é calculada por $X_{op} = \frac{K_{amb}}{K_{comp}} \times (1 - E_{f_{deliv}}) \times P_{in} \times T_{life} \times P\{\#X_ON > 0\}$. Da mesma forma, a exergia operacional relacionada com os blocos de construção descritas nas próximas seções é computada. Neste caso, a expressão deve substituir a disponibilidades (variável A) na Equação 2.10. No entanto, por razões de concisão, tais expressões não são explicitamente descritas nas seções seguintes.

2.4.4.2 Modelo de Componente Simples considerando Primeiro e Segundo *Moment Matching*

A variação do componente simples inicial apresentado na Seção 2.4.4.1 considera o refinamento das transições, a fim ter em conta o primeiro e segundo *moment matching*. Neste modelo particular, a transição $X_Failure$ do modelo mostrado na Figura 2.5 foi refinado para representar um distribuição Erlang (transição-s Erlang). O parâmetro X_MTTF (μ) associado a este evento não tem uma distribuição exponencial e a relação ($X_MTTF/\text{desvio padrão} - \frac{\mu}{\sigma} \in \mathbb{N}, \frac{\mu}{\sigma} \neq 1, (\mu, \sigma \neq 0)$) não é um número inteiro igual a 1. De acordo com o método de *moment matching*, esta transição deve ser refinado como transição-s Erlang [54] (veja a Figura 2.6).

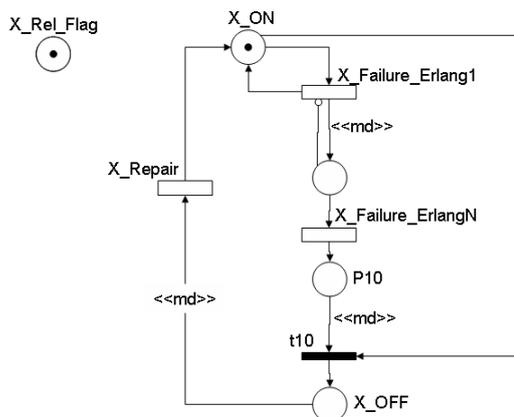


Figura 2.6: Componente simples com uma falha representada por distribuição Erlang

As transições temporizadas $X_Failure_Erlang1$ e $X_Failure_ErlangN$ representam as fases exponencial de distribuição de Erlang. Os arcos que conectam o lugar X_ON à transição de tempo $X_Failure_Erlang1$ e o local p_{10} à transição imediata t_{10} tem multiplicidade dependente da marcação, e sua expressão no arco é $(\gamma - 1) = [(\frac{\mu}{\sigma})^2 - 1]$ e $\gamma = (\frac{\mu}{\sigma})^2$, respectivamente (ver Tabela.2.4). O atraso atribuído a transições $X_Failure_Erlang1$ e $X_Failure_ErlangN$ é de $R \mu_1 = \mu/\gamma$. Vale observar os elementos de controle que são adotados neste modelo, tal que um token é mantido armazenado no lugar X_ON durante

o disparo de todas as fases Erlang, e, em seguida, é retirado apenas quando a transição t_{10} dispara, terminando toda a fase de disparo e completando a execução da transição Erlang.

Para descrever tal comportamento, um arco simples da transição $X_Failure_Erlang1$ para X_ON e outro a partir desse lugar para a transição t_{10} tem que ser adicionado, bem como um arco inibidor do local de saída da transição $X_Failure_Erlang1$ e volta para a mesma transição. Neste caso, o arco inibidor evita a transição da primeira fase ($X_Failure_Erlang1$) disparando sem limites do modelo Erlang. Esta transição só poderia ser ativada novamente quando nenhum *token* permanece no local de saída da transição $X_Failure_ErlangN$. Poderíamos argumentar que após a remoção de todos os *token* do local de saída da transição $X_Failure_ErlangN$, a transição $X_Failure_Erlang1$ poderia imediatamente disparar novamente. No entanto, deve-se ter em mente que as transições imediatas têm prioridade de disparo sobre as temporizadas, assim a transição $X_Failure_Erlang1$ não poderia ser disparada novamente antes da transição t_{10} disparar.

Tabela 2.4: Atributos da transição do componente simples com falha representada por distribuição Erlang

Transição	Tipo	Atraso	Peso	Concorrência
$X_Failure_Erlang1$	exp	$\frac{X_MTTF}{\gamma}$	-	SS
$X_Failure_ErlangN$	exp	$\frac{X_MTTF}{\gamma}$	-	SS
X_Repair	exp	MTEP	-	SS
t_{10}	im	-	1	-

Esse componente também inclui um arco do lugar X_OFF para transição X_Repair com multiplicidade dependente de marcação. A Multiplicidade do arco é definida pela expressão $IF(\#X_Rel_Flag1=1) 0 \text{ ELSE } 1$, onde o local de X_Rel_Flag permite a avaliação de métricas de confiabilidade e disponibilidade como descrito nas seções anteriores. A Tabela 2.5 apresenta as condições do estado que representam os estados do componente: operacional e falho.

Tabela 2.5: Estados do componente simples com falha representada por distribuição Erlang.

Estado	Condição
Operacional	$\#X_ON > 0$
Falho	$\#X_OFF > 0$

Muitas outras variações deste modelo foram definidos, incluindo outros refinamentos (sub-hiperexponencial e hypoexponential) na transição $X_Failure$ (ver Figura 2.5), refinamentos na transição X_Repair (ver Figura 2.5) e a inclusão de uma cobertura de falha e imperfeições de reparação.

A expressão de forma fechada para disponibilidade de “componente simples” RBD, com TTF distribuído de acordo com uma distribuição Erlang e TTR exponencialmente distribuído é $A = \frac{\gamma \times \mu}{(\gamma \times \mu) + MTTR}$.

Depois de computação disponibilidade do componente, é possível calcular a sua exergia operação, como mostrado na seção anterior.

2.4.4.3 Cold standby

Um componente com redundância em *Cold standby* é utilizado quando existe um módulo não ativo de reserva que espera para ser ativado quando o módulo (principal) ativo falhar. Assim, quando o módulo principal falhar, a ativação do módulo de reposição leva um certo tempo para ser ativado. Este período é chamado Tempo Médio para ativar (MTActivate). Representar a redundância *Cold standby* através de RBD não é fácil, uma vez que a estrutura do sistema é dinâmico, ou seja, ela muda ao longo do tempo, e o tempo para ativar o mecanismo de reposição deve ser representado. Assim, o modelo de *Cold standby* é apresentado em SPN.

A Figura 2.7 representa o modelo de rede de Petri deste componente, que inclui quatro lugares, são eles, X_{ON} , X_{OFF} , X_{Spare1_ON} , X_{Spare1_OFF} que representam os estados operacionais e falhos dos módulos principais e de reserva, respectivamente. O módulo de reposição (Spare1) é inicialmente desativado, uma vez que não existem *tokens* inicialmente armazenados nos locais X_{Spare1_ON} e X_{Spare1_OFF} . Quando o módulo principal falha, a transição $X_{Activate_Spare1}$ é disparada. Este disparo de transição indica ao módulo de reposição para começar a operação. Como mencionado anteriormente, além de parâmetros MTTF e MTTR, também é necessário representar o tempo médio para ativar. Vale observar que o MTTF e MTTR associados ao módulo principal podem ser diferentes do da reposição. Os atributos relacionados a cada transição do modelo descrito nesta seção são apresentadas na Tabela 2.6. Como cada distribuição de disparo da transição é independente de marcação, as transições dependentes de marcação são omitidas.

Tabela 2.6: modelo Cold standby. - atributos das Transições.

Transição	Tipo	Delay	Pesos	Concorrência
$X_{Failure}$	exp	X_{MTTF}	-	SS
X_{Repair}	exp	X_{MTTR}	-	SS
$X_{Activate_Spare1}$	exp	MTActivate	-	SS
$X_{Failure_Spare1}$	exp	X_{MTTF_Spare1}	-	SS
X_{Repair_Spare1}	exp	X_{MTTR_Spare1}	-	SS
$X_{Deactivate_Spare1}$	im	-	1	-

Se o lugar X_{Rel_Flag} está marcado (um *token*), o peso do arco do lugar X_{OFF} para

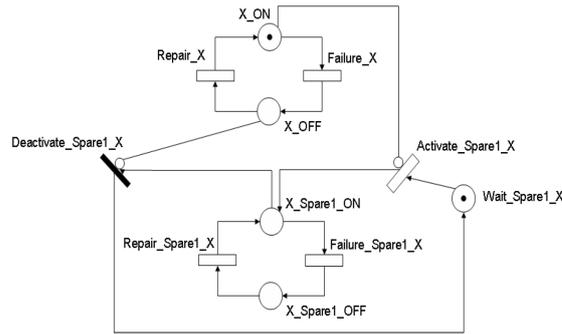


Figura 2.7: Modelo Cold standby.

a transição X_{Repair} é igual a 2, caso contrário, o peso do arco é 1 - este peso do arco é dependente de marcação. Assim, a expressão $P\{\#X_{ON}=1\}$ calcula a disponibilidade do componente enquanto $\#X_{Rel_Flag} = 0$, e a confiabilidade do componente quando $\#X_{Rel_Flag} = 1$, assim como nos modelos anteriores.

2.4.4.4 Falha em modo comum.

Uma característica proeminente do modelo SPN refere-se à representação de falha de modo comum (CMF), que normalmente ocorre quando os eventos não são estatisticamente independentes. Neste caso, uma falha de um componente pode afetar outras peças do sistema, por exemplo, impactando o tempo até a falha. Representar este modo de falha comum através RBD não é trivial, devido as mudanças da estrutura do sistema ao longo do tempo. No contexto da SPN, a combinação de blocos de componente simples podem modelar tal do modo de falha, associando atrasos dependente de marcação das transições de componentes relacionados.



Figura 2.8: Exemplo de modo de falha comum.

A Figura 2.8 mostra um exemplo composto por dois componentes, em que a falha de C1 causa impacto no MTTF de C2. Assumindo uma taxa de falha λ , C1 pode afetar, por exemplo, C2 da seguinte forma: $C2_MTTF = 1/(\lambda \times (\#C1_ON + 1))$, onde $\#C1_ON$ representa o número de *tokens* no lugar C1_ON. Componentes adicionais podem ser levados conta e os seus respectivos MTTRs podem ser ajustados para considerar o atraso de uma falha dependente de marcação, semelhante ao componente C2.

2.4.4.5 Componente de Agregação

O modelos propostos também levam em conta um elemento de agregação (Figura 2.9), que representa os estados de atividade (X_{ON}) e inatividade (X_{OFF}) de um subsistema ou sistema (por exemplo, um grupo de componentes simples). Transições $X_{Failure}$ e X_{Repair} possuem um parâmetro associado (não é mostrado na figura), são eles, X_{Fail_GE} e X_{Repair_GE} , respectivamente, que são expressões de guarda. Mais especificamente, uma transição (por exemplo, $X_{Failure}$) só pode disparar se a expressão guarda respectiva (por exemplo, X_{Fail_GE}) é verdadeira. Essas transições são transições imediatas (ver Tabela 2.7).

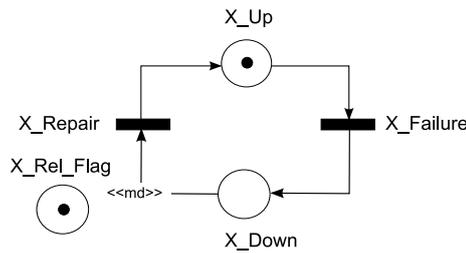


Figura 2.9: Modelo do componente lógico simples

Da mesma forma como o modelo do componentes simples, se a condição $\#X_{Rel_Flag}=1$ é falsa (sem *token* no lugar X_{Rel_Flag})então a avaliação se refere à disponibilidade. Caso contrário, a avaliação se refere a confiabilidade.

Tabela 2.7: Atributos do componente de agregação.

Transição	Tipo	Expressão da Guarda
$X_{Failure}$	im	X_{Fail_GE}
X_{Repair}	im	X_{Repair_GE}

2.4.4.6 Composição do modelos

A composição do modelos é realizada através de modelos básicos e um componente de agregação (Figure 2.9), que é adotado para representar a relação entre os componentes usando expressões de guarda nas transições imediatas. Sem perda de generalidade, os subsistemas são combinados em série, paralelo, e utilizando composições hierárquicas. Modelos RBDs são similarmente combinadas usando equações bem estabelecidas de forma fechada disponíveis para aqueles composições. Como exemplo, a composição da série é apresentada da seguinte forma, considerando blocos SPN.

Composição em blocos. Quando dois elementos de uma composição estão em série significa que se apenas um deles falhar toda a composição também falha, ou seja,

o sistema está operacional se ambos os componentes se encontram operacionais. A Figura 2.10 apresenta um equipamento chamado de dispositivo, ele é composto por dois componentes básicos não redundantes, $C1$ e $C2$, dispostos em série. Como mencionadas anteriormente, os lugares X_{Up} e X_{Down} representam os estados operacionais e falhos, respectivamente.

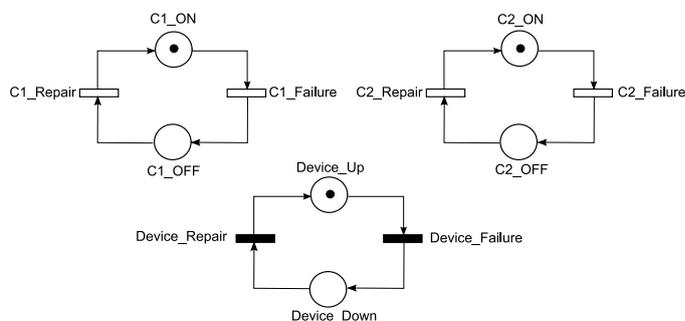


Figura 2.10: Composição em série de dois componentes básicos não-redundantes

A fim de especificar completamente o modelo de componente de agregação, é necessário definir funções de habilitação, através de expressões de guarda, associadas às transições imediatas: $Device_Repair$ e $Device_Failure$. A expressão guarda de transição $Device_Failure$ é representada pela condição que representa um estado de falha. Da mesma forma, a expressão de guarda associada a transição $Device_Repair$ é definida pela expressão lógica que descreve as condições referentes ao estado operacional. A Tabela 2.8 apresenta as expressões de guarda do modelo de agregação representado na Figura 2.10. Na primeira linha, a transição de falha é habilitada quando $C1$ ou $C2$ está em falha.

Tabela 2.8: Expressões de guarda das transições de uma composição em série.

Transição	Expressão de guarda
$Device_Failure$	$(\#C1_ON=0) \text{ OR } (\#C2_ON=0)$
$Device_Repair$	$\text{NOT}((\#C1_ON=0) \text{ OR } (\#C2_ON=0))$

Deve-se ressaltar que, assim como nos casos anteriores, sempre que se queira avaliar a confiabilidade, a marcação de lugar X_{Rel_Flag} (não representado na figura) deve ser igual a um ($\#X_{Rel_Flag} = 1$). Portanto, a confiabilidade do sistema é representado pelo resultado relacionado com a expressão $P\{\#Device_Up > 0\}$, utilizando a avaliação transiente. A disponibilidade do sistema é expresso pelo resultado da avaliação de estado estacionário da mesma expressão, mas apenas quando $\#X_{Rel_Flag} = 0$.

Além disso, essa abordagem pode ser adotada para combinar componentes N representada por um único modelo de agregação de componentes.

2.5 Considerações Finais

Este capítulo apresentou os principais conceitos que envolvem esta dissertação. Primeiramente, foi brevemente apresentada a estrutura dos data centers. Posteriormente, foram abordados os conceitos de dependabilidade. Em seguida, foram apresentados conceitos relacionados à sustentabilidade. Por fim, foram mostrados os modelos utilizados neste trabalho para avaliação de dependabilidade e sustentabilidade.

3

Visão geral da Ferramenta

Este capítulo apresenta a arquitetura da ferramenta proposta e descreve suas principais funcionalidades.

3.1 Arquitetura da ferramenta

ASTRO é uma ferramenta de modelagem de dependabilidade e sustentabilidade, que permite a utilização de diagramas de bloco de confiabilidade (RBD), redes de Petri estocásticas (SPN) e ambientes de alto nível para avaliação de infraestruturas *data center*. Além de possuir editores de alto nível voltados para ambientes *data center*, a ferramenta permite a avaliação de dependabilidade de sistemas genéricos para usuários que conheçam princípios de modelagem SPN e RBD. Os usuários têm a possibilidade de reuso de resultados entre os modelos, facilitando o trabalho de avaliação. A ferramenta ASTRO foi completamente escrita em JAVA, sendo assim podemos executá-la em sistemas que possuam a máquina virtual JAVA, seja Linux, Windows, Mac ou Solaris. As principais funcionalidades da ferramenta são apresentadas na Figura 3.1 e mostradas a seguir:

- **Avaliação da infraestrutura de fornecimento energético de sistemas *data center*.** A ferramenta ASTRO disponibiliza um ambiente para avaliação de infraestruturas de potência em sistemas *data center*. Neste ambiente, modelos de alto nível podem ser criados para representar a estrutura do sistema de potência do *data center*. Assim, pode-se avaliar o sistema em termos de dependabilidade e sustentabilidade sem a necessidade do usuário utilizar modelos SPN ou RBD diretamente. Neste contexto, os componentes do sistema de potência do *data center* são representados por "ícones", assim, apenas a estrutura do sistema precisa ser definida

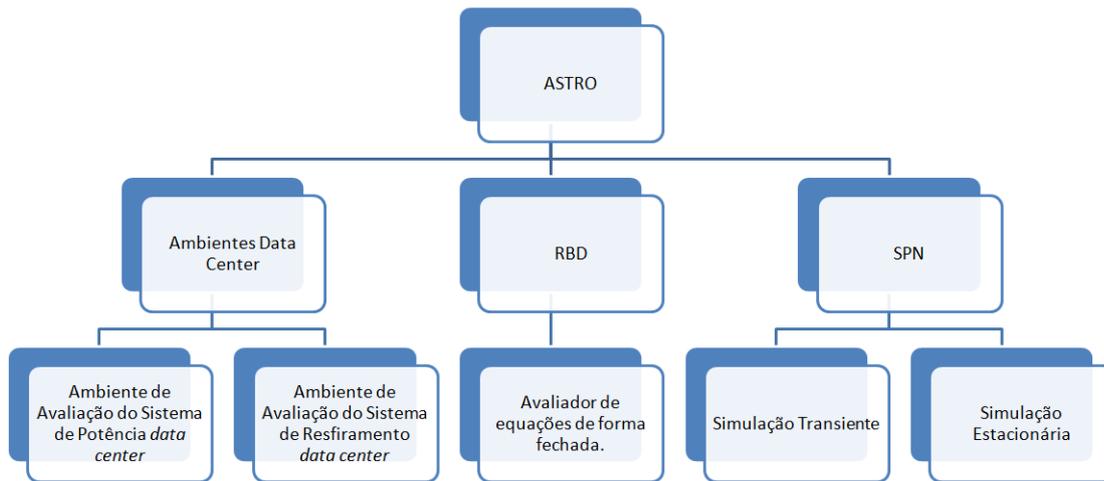


Figura 3.1: ASTRO - Funcionalidades

para que o sistema seja avaliado.

- **Avaliação da infraestrutura de resfriamento de sistemas *data center*.** A ferramenta ASTRO apresenta um ambiente para avaliação da infraestrutura de resfriamento de sistemas *data center*. Este ambiente é similar ao ambiente do sistema de potência, porém com a adição de componentes especiais que representam máquinas térmicas. Além disso, o mesmo também permite a avaliação de métricas de sustentabilidade e dependabilidade;
- **Simulação estacionária e transiente de modelos SPN.** Os usuários também têm a possibilidade de executar avaliação de dependabilidade em sistemas genéricos utilizando redes de Petri estocásticas (SPN). As métricas de dependabilidade são obtidas através de simulação estacionária ou transiente.
- **Modelagem e avaliação de modelos RBD.** A partir do ambiente RBD, a ferramenta ASTRO permite a representação e avaliação de diagramas de blocos de confiabilidade. Além da avaliação de modelos RBD, o ambiente também permite funcionalidades como: experimentação de cenários, geração de funções lógicas e estruturais, avaliação do impacto dos componentes e avaliação de RBD por limites.

3.1. ARQUITETURA DA FERRAMENTA

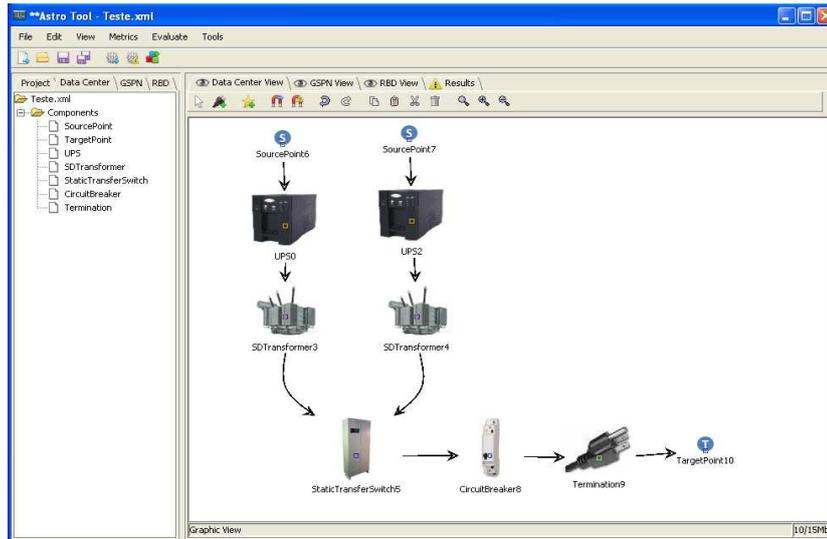


Figura 3.2: ASTRO - Ambiente do sistema de potência *data center*.

A Figura 3.3 apresenta uma visão geral dos componentes que compõem a estrutura da ferramenta ASTRO. São eles: interface gráfica, ambiente de resfriamento, ambiente de potência, ambiente SPN, ambiente RBD e avaliador de sustentabilidade. O avaliador de sustentabilidade juntamente com os ambientes de resfriamento e de potência serão explicados em detalhes no capítulo 6. Os ambientes de SPN e RBD serão explicados nos capítulos 4 e 5 respectivamente.



Figura 3.3: ASTRO - Estrutura dos Componentes internos.

Note que a partir da interface gráfica todos os componentes podem ser acessados, com exceção do avaliador de sustentabilidade que só é possível ser acessado através dos editores de Resfriamento e de potência. Uma breve descrição dos componentes da arquitetura é apresentada a seguir. Apesar do avaliador de sustentabilidade aparecer

3.2. AMBIENTE DE AVALIAÇÃO DE INFRAESTRUTURAS DE POTÊNCIA *DATA CENTER*.

como um componente separado na Figura 3.3, os editores de potência e resfriamento possuem uma instância do avaliador de sustentabilidade em suas respectivas estruturas internas.

3.2 Ambiente de avaliação de infraestruturas de potência *data center*.

Em termos de abstração, o ambiente de avaliação de infraestruturas de potência *data center*, também chamado de ambiente de potência, fornece uma visão de alto nível, no qual o usuário define os componentes do sistema *data center* de forma estrutural e a partir dela pode-se avaliar métricas de sustentabilidade e dependabilidade. O ambiente permite que o usuário especifique a estrutura do *data center* através de ícones que representam cada componente do sistema, descrevendo a conexão de componentes através dos arcos direcionados. Além disso, cada componente do *data center* tem informações relacionadas a dependabilidade e sustentabilidade (Seção 2.4). A Figura 6.8 mostra o editor de sistema de potência, note que uma visão estrutural da infraestrutura de potência é definida pelo usuário. Além disso, é importante salientar que os valores de dependabilidade e sustentabilidade obtidos pela avaliação de um determinado sistema podem ser armazenadas para uso posterior. Por exemplo, suponha um sistema S1, que é composto por dois sub-sistemas idênticos, S2 e S3. Para avaliar o sistema S1, é possível avaliar separadamente S2 e S3, e, em seguida, avaliar S1 reutilizando os resultados obtidos anteriormente.

Pode-se destacar duas funcionalidades principais do ambiente de potência:

- *Tradução de estruturas de potência data center em SPN e RBD.*

A fim de avaliar estruturas de potência *data center* em termos de dependabilidade, os mesmos devem ser traduzidos para RBD ou SPN. Cada componente do *data center* é convertido em um componente RBD ou SPN. Além disso, a estrutura do RBD ou SPN é definida de acordo com a estrutura *data center* a ser traduzida, considerando aspectos de energia e conexões entre os componentes. A Figura 3.5 mostra a tradução de estruturas *data center* para RBD e SPN.

- *Avaliação do impacto da utilização de estruturas de potência data center em termos de sustentabilidade.*

Essa funcionalidade é responsável por estimar os impactos da utilização do sistema de potência *data center* em termos de sustentabilidade (ver Seção 2.4.2).

3.3. AMBIENTE DE AVALIAÇÃO DE INFRAESTRUTURAS DE RESFRIAMENTO *DATA CENTER*.

Essa funcionalidade também calcula o custo total de propriedade que é composto pelo custo de se adquirir a infraestrutura somado ao custo de operação da mesma. A Figura 6.6 mostra o resultado de uma avaliação de sustentabilidade. Essa funcionalidade é discutida com mais detalhes na Seção 6.3.

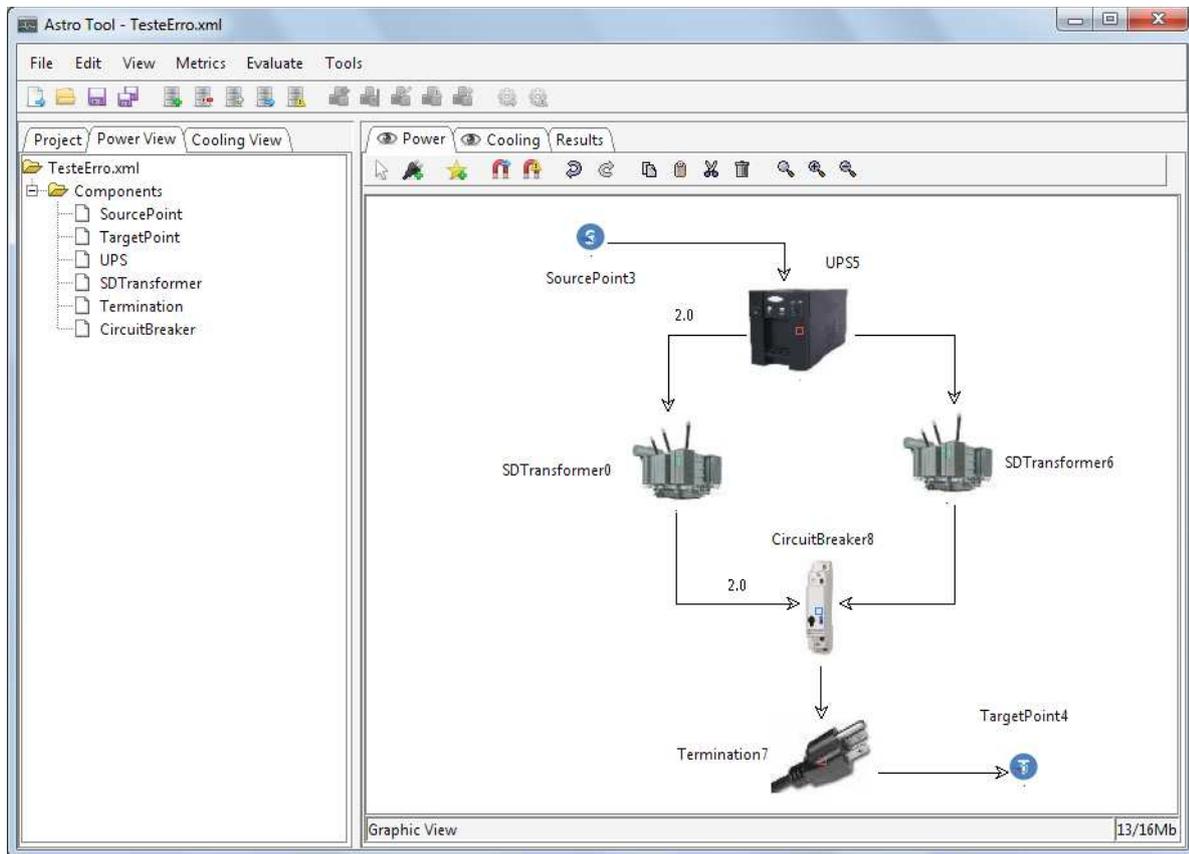


Figura 3.4: Interface gráfica do ambiente de potência.

3.3 Ambiente de avaliação de infraestruturas de resfriamento *data center*.

O ambiente de avaliação de infraestruturas de resfriamento *data center*, também chamado de ambiente de resfriamento, é similar ao ambiente de potência. A principal diferença entre os mesmos é que o ambiente de refrigeração utiliza componentes representados por máquinas térmicas e a passagem de energia entre os componentes se dá em forma de calor. Assim como o ambiente de potência, o ambiente de resfriamento

3.3. AMBIENTE DE AVALIAÇÃO DE INFRAESTRUTURAS DE RESFRIAMENTO DATA CENTER.

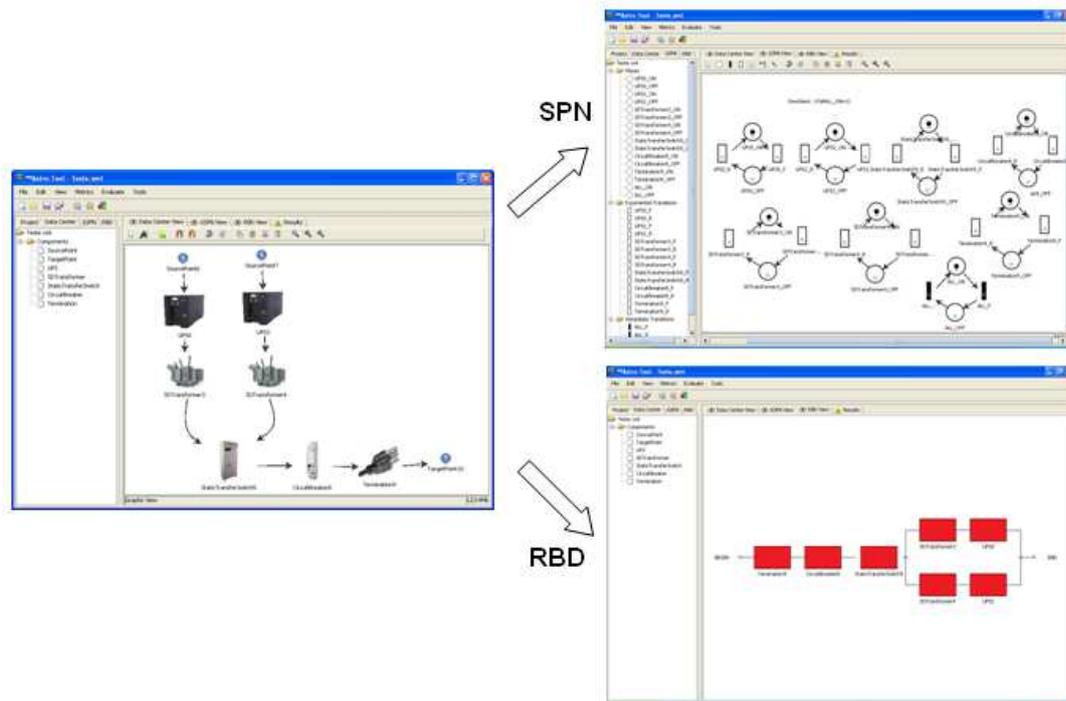


Figura 3.5: Conversão entre modelos

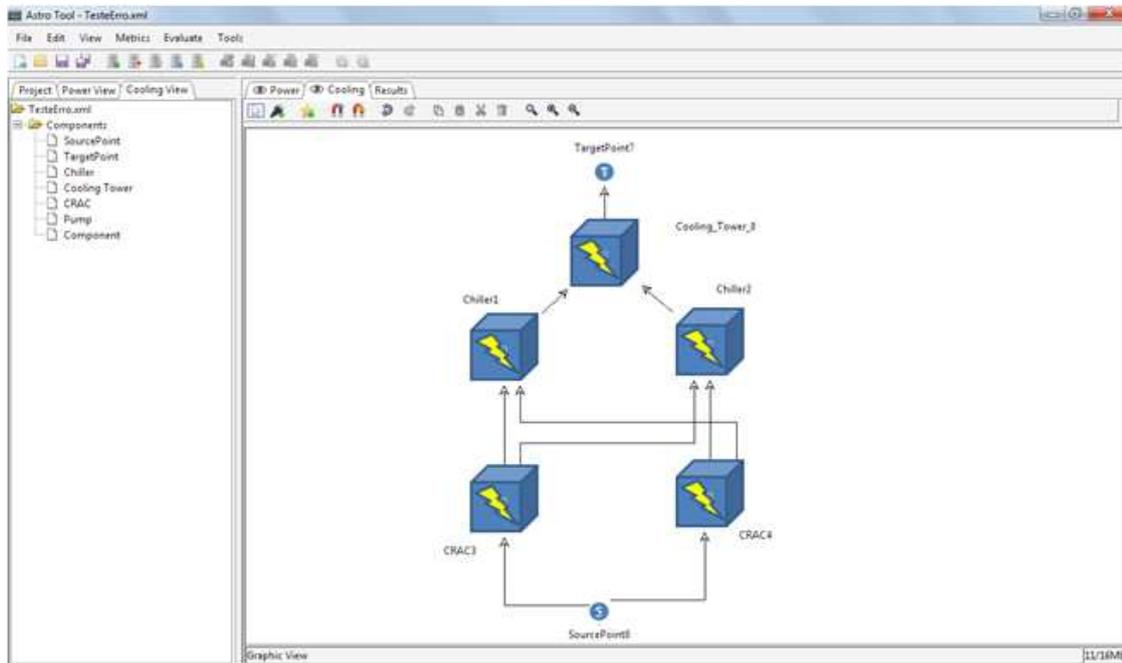


Figura 3.6: Ambiente de resfriamento.

é responsável pela tradução de modelos de alto nível para RBD ou SPN (Figura 6.3), além de quantificar o impacto da sustentabilidade do modelo em termos de exergia.

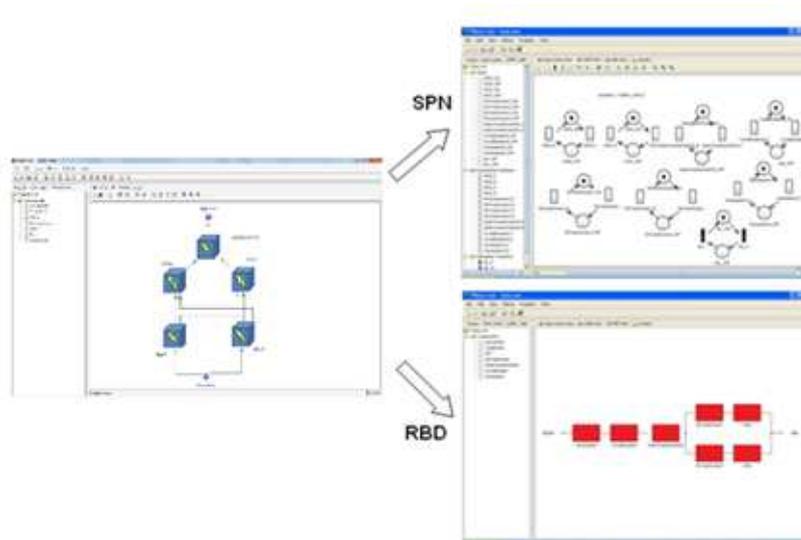


Figura 3.7: Conversão no ambiente de resfriamento.

Todos os recursos presentes nos ambientes RBD e SPN são acessíveis pelos ambientes de potência e resfriamento, permitindo que os usuários avaliem dependabilidade sem a necessidade de acessar o editor RBD ou SPN diretamente.

3.4 Ambiente de avaliação RBD

O Ambiente RBD permite a realização de avaliação de dependabilidade utilizando diagramas de blocos de confiabilidade (RBD). Os tipos de configurações suportados pelo editor RBD são: série, paralelo, K out of N e Bridge. RBDs utilizam equações de forma fechada, por isso os resultados são normalmente obtidos mais rapidamente do que utilizando simulação SPN. No entanto, existem muitas situações (por exemplo, dependência entre os componentes), no qual a modelagem baseada em RBD é bem mais difícil de ser realizada que na modelagem SPN. Utilizando o ambiente RBD é possível avaliar diferentes cenários para um mesmo modelo RBD de forma automática através da funcionalidade de avaliação de cenários (ver Seção 5.1.1). Além disso, funcionalidades proeminentes (ver Seção 5.1) como geração de funções lógicas e estruturais, avaliação de importância dos componentes e avaliação de dependabilidade por limites são providos pela ferramenta ASTRO.

A Figura 3.8 mostra o editor RBD da ferramenta.

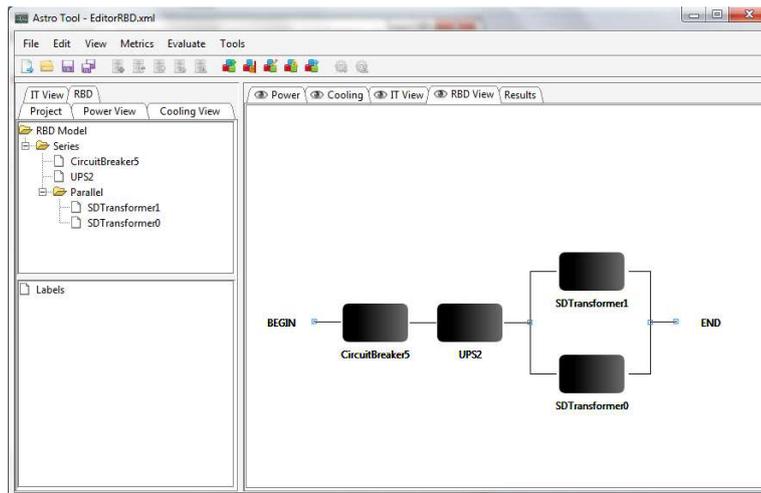


Figura 3.8: Ambiente RBD na ferramenta ASTRO

3.5 Ambiente de Avaliação SPN

O ambiente de avaliação SPN é responsável por permitir a criação, depuração e simulação de modelos SPN. A criação de modelos é feita de maneira gráfica utilizando um editor próprio. A avaliação de dependabilidade do ambiente SPN é feita adotando técnicas de simulação, tais como transiente e estacionária [11]. Métricas dependentes do tempo são obtidas através de simulações transiente, enquanto que as métricas de estado estacionário são resultado de simulações estacionárias. A Figura 3.9 mostra um exemplo de resultado de uma simulação transiente. Os modelos SPN podem ser obtidos a partir de dados dos ambientes de avaliação para sistemas *data center* ou criados pelo usuário diretamente. O Ambiente SPN é mostrado em detalhes no Capítulo 4.

3.6 Considerações finais.

Este capítulo apresentou os principais componentes e funcionalidades do ambiente ASTRO, permitindo ao usuário ter uma visão geral da ferramenta. Inicialmente foi mostrada a arquitetura da ferramenta, em seguida foram apresentados os ambientes de avaliação de sistemas *data center*, ou seja, os ambientes de potência e resfriamento. Por fim, foram mostrados os ambientes responsáveis pela avaliação de dependabilidade, são eles: ambientes RBD e SPN.

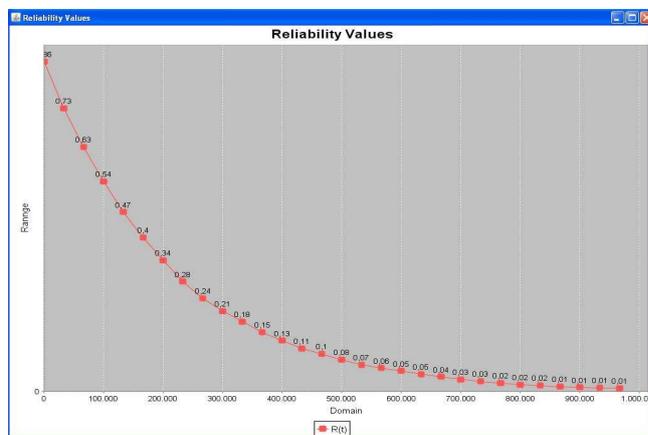


Figura 3.9: Exemplo de Resultados transientes na ferramenta ASTRO. (Tempo X Confiabilidade)

4

Ambiente de Avaliação SPN

Este capítulo descreve em detalhes o ambiente de avaliação SPN, ou simplesmente ambiente SPN. Primeiramente, são mostradas as principais funcionalidades do ambiente, das quais podemos destacar as 3 mais importantes que são a (i) simulação estacionária, (ii) simulação transiente, e (iii) experimentação de cenários. Em seguida, são abordados os principais componentes do ambiente SPN. São eles: interface gráfica, gerenciador de simulação e simulador de eventos discretos.

4.1 Descrição das funcionalidades.

O ambiente de avaliação SPN permite a modelagem e avaliação de métricas de desempenho em sistemas genéricos utilizando-se redes de Petri estocásticas [44], ele é recomendável para usuários que conheçam o formalismo SPN e que desejam modelar sistemas genéricos, não necessariamente *data center*. O ambiente SPN utiliza simulação para avaliação de modelos SPN e há duas formas de simulação distintas no ASTRO: simulação estacionária e simulação transiente. Ambas as formas utilizam as mesmas funções básicas, na primeira forma de simulação são calculadas métricas que não dependem do tempo para serem avaliadas, enquanto na segunda é preciso estabelecer o(s) instante(s) de tempo em que a métrica deve ser avaliada. A terceira funcionalidade é a experimentação de cenários, que permite ao usuário executar várias simulações automaticamente variando um determinado parâmetro do modelo, assim é possível experimentar diferentes cenários de forma automática. As próximas seções apresentam as funcionalidades do ambiente SPN.

4.1.1 Simulação estacionária.

A simulação estacionária permite a avaliação de desempenho do sistema depois que os efeitos transitórios iniciais se passarem e um estado estável seja alcançado. A simulação do estado estacionário pode ser utilizada para responder a perguntas típicas como: qual será o número médio de pessoas esperando numa determinada fila? Qual a probabilidade do sistema estar funcionando em um tempo qualquer? São questões como essas que são solucionadas através da simulação estacionária.

Na avaliação estacionária, a ferramenta ASTRO simula o modelo disparando as transições de acordo com as distribuições de probabilidades associadas a cada transição e as regras de disparo do modelo SPN.

A Figura 4.1 apresenta a tela de parâmetros utilizados na simulação estacionária da ferramenta ASTRO, são eles:

- **Confidence Level %.** (Nível de confiança.) É a probabilidade de que o intervalo de confiança (ver Seção 4.2.3.3) contenha o verdadeiro valor da métrica buscada [35].
- **Max. Relative Error.** (Erro máximo relativo.) Indica o tamanho relativo do intervalo de confiança em termos percentuais, ou seja, é a razão entre tamanho do intervalo de confiança e o valor médio da amostra. Seja a a diferença do maior valor e menor valor do intervalo de confiança, e \bar{x} o valor médio da amostra. O Erro máximo relativo ε é dado por:

$$\varepsilon = \frac{a}{\bar{x}}. \quad (4.1)$$

- **Min. # firing for each Transition.** (Número mínimo de disparos para cada transição.) Pode ser utilizado como um limitante inferior no processo de simulação, dado que exige que cada transição dispare um determinado número de vezes para que a rodada de simulação (ver Seção 4.2.3.3) acabe. Essa funcionalidade pode ser utilizada quando as taxas de disparo das diferentes transições da rede são muito diferentes.
- **Warm-up period (# Runs).** (Período de *warm-up*.) Refere-se ao número de disparos iniciais que precisam ser descartados para remoção do período transiente (ver Seção 4.2.3.4).
- **Run Size (# of firing)** (Tamanho da rodada). Utilizado pela técnica das médias em lotes para determinar o tamanho de cada lote (ver Seção 4.2.3.4).

4.1. DESCRIÇÃO DAS FUNCIONALIDADES.

- **Max. simulation real time (sec).** (Tempo máximo de simulação (Segundos).) Tempo real medido em segundos para limitar a espera do usuário em relação ao processo de simulação. É utilizado caso se queira parar a simulação quando o tempo máximo de simulação for alcançado, mesmo que o processo de simulação ainda não tenha atingido um valor com o nível de confiança e erro desejados. Por exemplo, suponha que o usuário não queira esperar mais que duas horas para obter o resultado da simulação, ele deve especificar o tempo de 7200 segundos (duas horas) neste campo.
- **Experiment** (Habilitar experimentação de cenários.) Habilita a opção de **experimentação de cenários**, essa funcionalidade é explicada em mais detalhes na Subseção 4.1.3.
- **Batch Means** (Utilizar método de médias em lotes.) Habilita a remoção do período transiente utilizando o método de médias em lote, vide Seção 4.2.3.4.

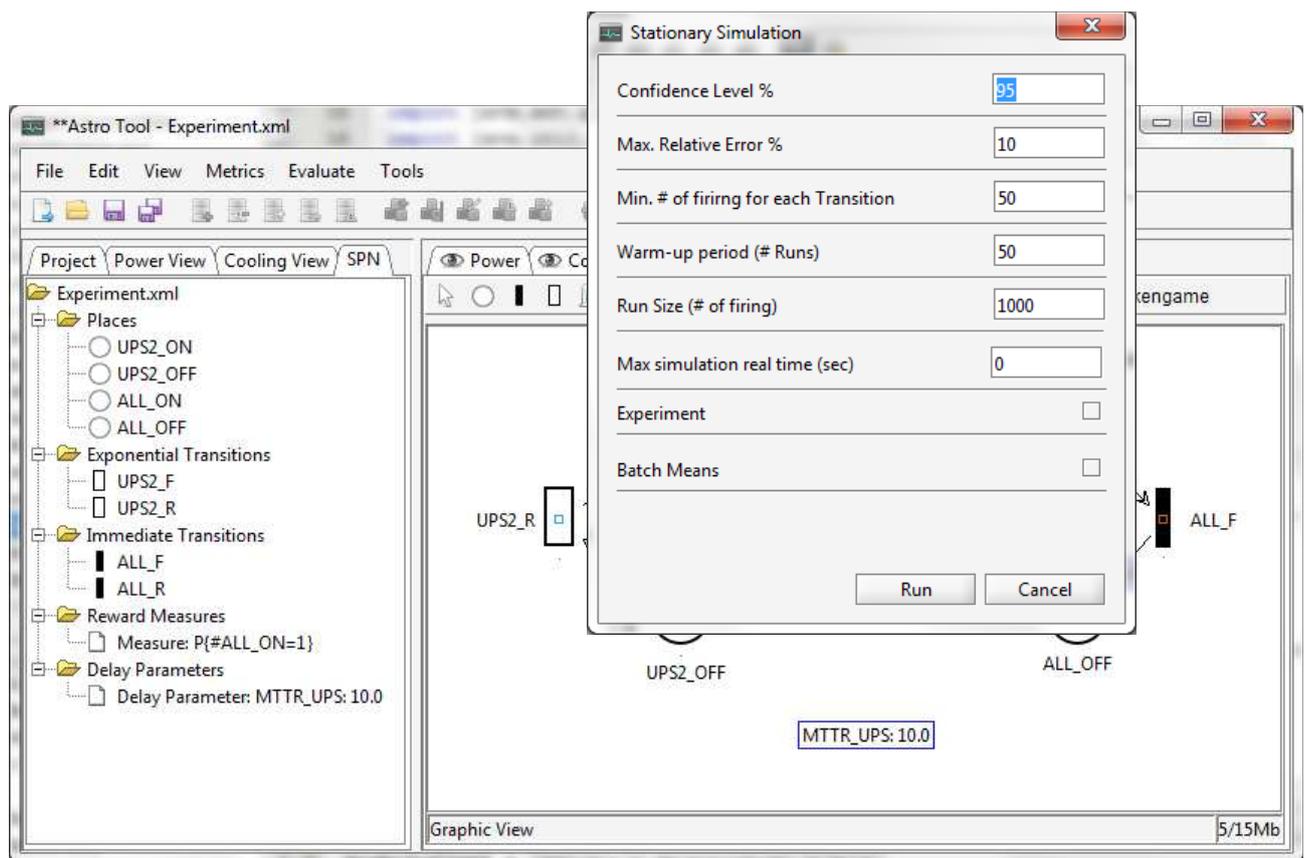


Figura 4.1: Simulação estacionária.

4.1.2 Simulação transiente.

A simulação transiente permite a análise do comportamento de um determinado sistema a partir do instante inicial até um determinado instante de tempo. Então, deve-se executar a simulação transiente para responder a perguntas do tipo: Qual é a probabilidade do sistema ainda estar operacional após uma semana de funcionamento? As medidas de desempenho são computadas do instante inicial até o instante de tempo especificado. As mesmas funções básicas do simulador de eventos discretos são utilizadas nas simulações transiente e estacionária. Uma diferença importante entre os dois processos de simulação (simulação estacionária e transiente) é a funcionalidade de remoção do período transiente (ver Seção 4.2.3.4), que é utilizada apenas pelo processo de simulação estacionária. Além disso, na simulação transiente a análise do critério de parada é feita no(s) instante(s) de tempo especificados pelo usuário, enquanto que na simulação estacionária o processo encerra quando (e se) o regime estacionário for alcançado, considerando-se o nível de confiança e o erro definidos.

A Figura 4.3 apresenta a tela de entrada dos parâmetros considerados na simulação transiente, são eles:

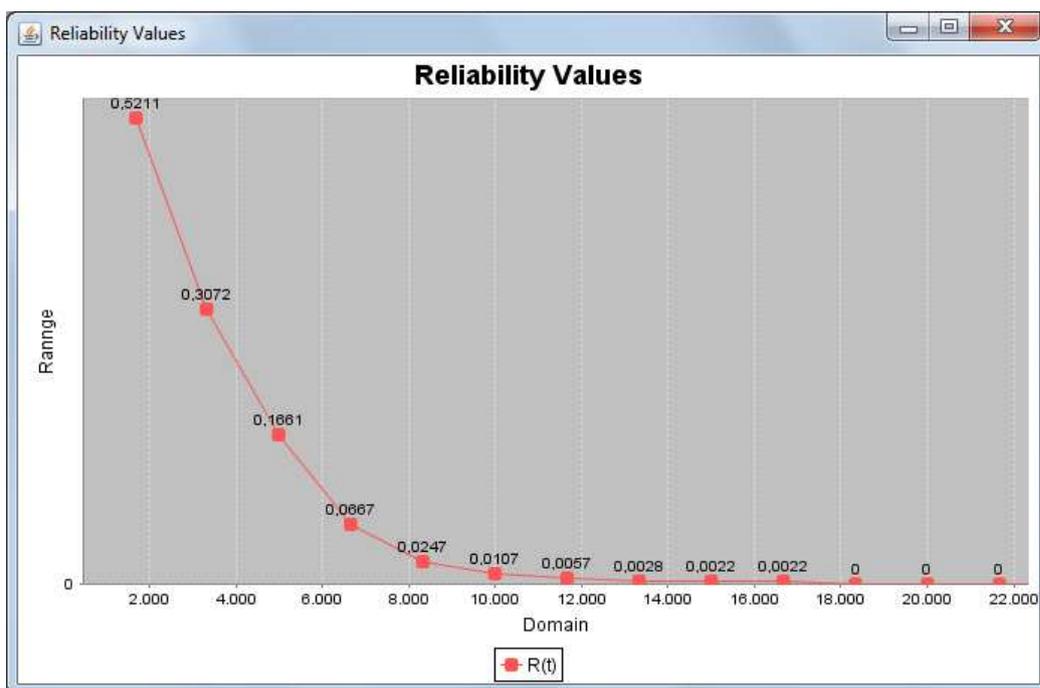


Figura 4.2: Resultado da simulação transiente

- **Confidence Level %**(Nível de confiança.) Corresponde ao mesmo parâmetro utilizado na simulação estacionária (ver Seção 4.1.1).

4.1. DESCRIÇÃO DAS FUNCIONALIDADES.

- **Max. Relative Error. %**(Erro máximo relativo.) Corresponde ao mesmo parâmetro utilizado na simulação estacionária (ver item 4.1.1).
- **Simulation Time.**(Tempo de simulação.) Especifica o tempo em que uma determinada métrica deve ser avaliada em unidades de tempo do modelo. Por exemplo, caso se queira avaliar um determinado sistema no instante 1000, deve-se especificar este valor no parâmetro **tempo de simulação**.
- **# Sampling Points**(Número de pontos de amostragem.) Especifica o número de pontos linearmente espaçados do tempo inicial de simulação até o **tempo de simulação** que serão avaliados.
- **Max. Simulation Real Time (sec).**(Tempo máximo de simulação (Segundos).) Corresponde ao mesmo parâmetro utilizado na simulação estacionária (ver item 4.1.1).
- **File Containing Results.**(Arquivo de resultados.) Arquivo de saída dos resultados da simulação transiente. No arquivo pode-se visualizar o instante de tempo, o valor da métrica avaliada e o respectivo intervalo de confiança.

A Figura 4.2 apresenta a tela de resultados de uma simulação transiente de um sistema considerando um período de 22,000 unidades de tempo e 13 pontos de amostragem, apesar de não mostrado na figura o eixo das abscissas apresenta o tempo de simulação (t) e o eixo das ordenadas apresenta o valor da métrica avaliada.

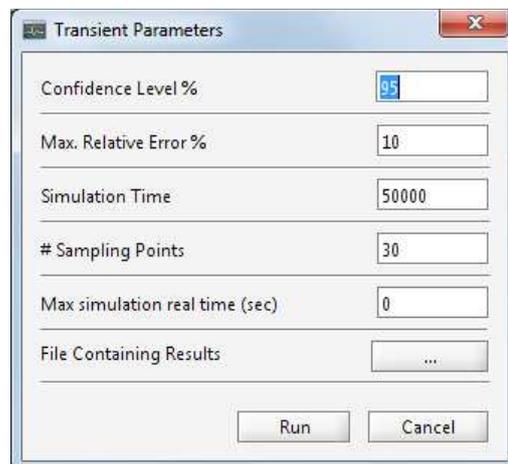


Figura 4.3: Parâmetros da simulação transiente

4.1. DESCRIÇÃO DAS FUNCIONALIDADES.

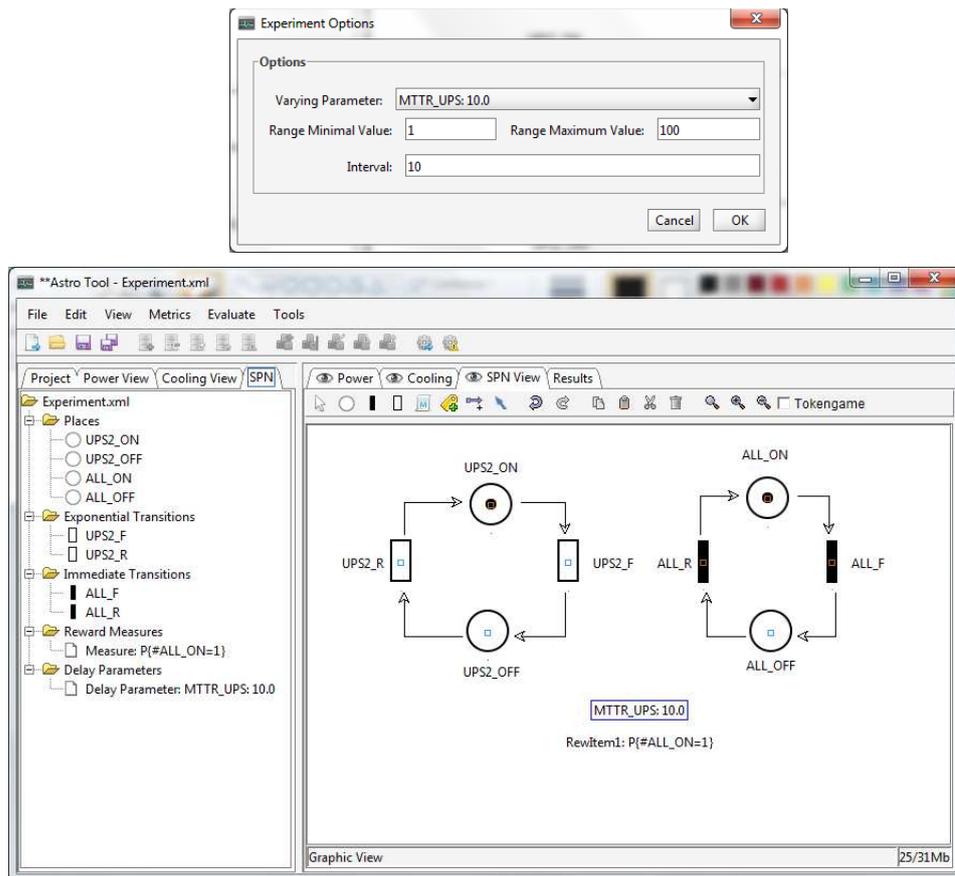


Figura 4.4: Experimentação de cenários.

4.1.3 Experimentação de cenários.

Experimentação de cenários é a funcionalidade na qual o programa executa várias simulações automaticamente variando um determinado parâmetro do modelo SPN. Suponha uma rede de petri que calcula a disponibilidade de um determinado sistema composto por vários equipamentos. Podemos variar linearmente, por exemplo, o tempo de falha de um determinado equipamento e avaliar o impacto desta variação na disponibilidade do sistema final. A variação dos parâmetros se dá de forma linear, no qual são selecionados o menor valor do intervalo, o maior valor e o passo.

Para executar a experimentação de cenários deve-se executar a simulação estacionária e habilitar a opção *Experiment*, ver Figura 4.1. Na Figura 4.4 são mostrados os parâmetros necessários para executar a experimentação de cenários, são eles:

- **Varying Parameter.** (Parâmetro variado.) Seleciona entre os parâmetros criados pelo usuário. A criação dos parâmetros é melhor explicada na subseção 4.2.1.

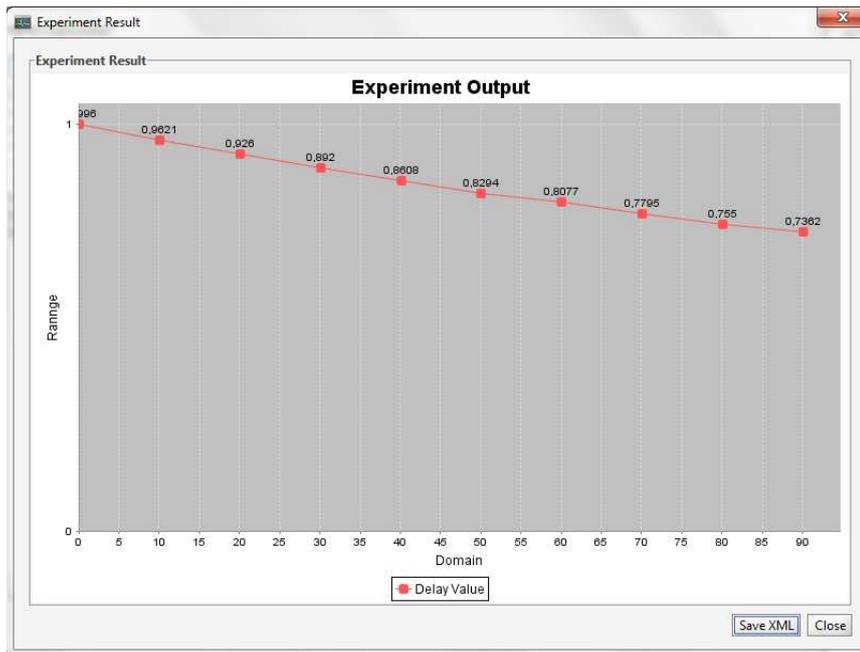


Figura 4.5: Resultado de experimentação SPN.

- **Range Minimal Value.** (Valor mínimo.) Menor valor do parâmetro selecionado.
- **Range Maximum Value.** (Valor máximo.) Maior valor do parâmetro selecionado.
- **Interval.**(Intervalo.) Passo linear utilizado para variar o parâmetro.

A Figura 4.5 apresenta a tela de resultados de uma experimentação de cenários do ambiente SPN. Apesar de não mostrado na figura o eixo das abscissas apresenta o valor do parâmetro variado (t) e o eixo das ordenadas apresenta o valor da métrica avaliada.

4.2 Estrutura do ambiente SPN

A Figura 4.6 apresenta a estrutura interna do ambiente SPN, seus componentes são apresentados nas próximas seções.

4.2.1 Interface gráfica

Esta subseção descreve brevemente a interface gráfica do ambiente SPN, a mesma é apresentada na Figura 4.7. A interface é composta por quatro áreas: uma barra de menu na parte superior, uma barra de ícones logo abaixo, no painel esquerdo temos uma visão geral do modelo e no direito a área de desenho. No painel esquerdo podemos visu-

4.2. ESTRUTURA DO AMBIENTE SPN



Figura 4.6: Estrutura ambiente SPN

alizar os componentes presentes no modelo que está sendo editado: lugares, transições e métricas, enquanto no painel direito está presente a representação gráfica do modelo SPN propriamente dito. Nas barras de menus e de ícones podemos acessar as principais funções do ambiente SPN apresentadas na Seção 4.1.

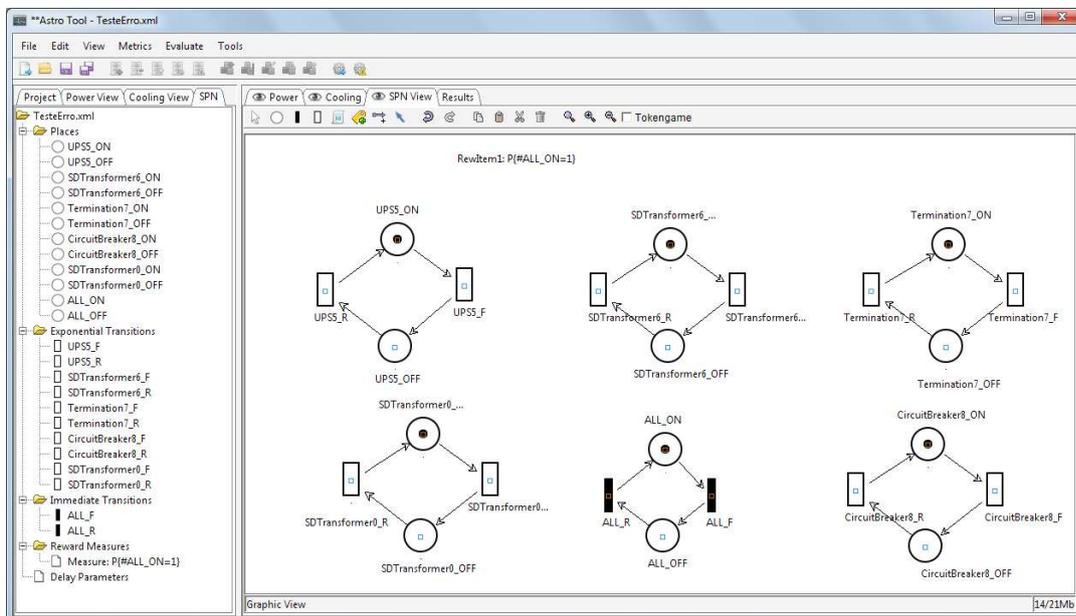


Figura 4.7: Ambiente SPN

A área de desenho possui alguns ítems importantes que merecem ser ressaltados (Figura 4.8).



Figura 4.8: Componentes Gráficos do Ambiente SPN

Os componentes da área de desenho são apresentados a seguir:

1. Componente lugar.

Os componentes que representam os lugares são representados por círculos (ver Figura 4.8), possuem duas informações principais:

- **Nome.** É mostrado como um identificador junto ao componente. O nome é importante para definições de métricas e expressões de guarda.
- **Marcação inicial.** Pode ser um número natural ou um rótulo (Ver Seção 5) pré-definido pelo usuário.

2. Componente transição imediata.

Os componentes que representam transições imediatas são representados por retângulos preenchidos (ver Figura 4.8), possuem três informações principais:

- **Prioridade.** É um número natural que define a precedência de disparo para as transições imediatas. O valor padrão é 'um' e a prioridade aumenta com o aumento do valor.
- **Peso.** No caso de duas ou mais transições, de mesma prioridade, habilitadas simultaneamente, o peso é a informação utilizada para o cálculo da probabilidade de disparo das mesmas.
- **Função de habilitação.** A função de habilitação, também chamada de guarda, é uma expressão dependente de marcação que deve ser avaliada como verdadeira para que a transição possa disparar.

3. Componente transição temporizada.

A definição de SPN [40] só admite distribuições exponenciais associadas às transições temporizadas. Porém, utilizando simulação é possível associar outras distribuições de probabilidade, além da exponencial, à estas transições. Nesse caso,

técnicas baseadas em avaliação do espaço de estados não são possíveis para avaliação de métricas [31].

Os componentes que representam transições temporizada são representados por retângulos não preenchidos (ver Figura 4.8). As distribuições de probabilidade suportadas na simulação SPN são: Exponencial, Normal, Erlang, Lognormal. Nas próximas versões da ferramenta ASTRO outras distribuições de probabilidade serão consideradas. As transições temporizadas possuem parâmetros relacionados às distribuições de probabilidades associadas [25], são eles:

- **Distribuição Exponencial.** O parâmetro da transição relacionado a distribuição exponencial é a média do tempo de disparo.
- **Distribuição Normal.** Possui dois parâmetros, são eles: (i) média do tempo de disparo e (ii) desvio padrão do tempo de disparo.
- **Distribuição Erlang.** Os parâmetros associados são: (i) média do tempo de disparo e (ii) números de fases.
- **Distribuição Lognormal.** Possui dois parâmetros: (i) média do logaritmo natural do tempo de disparo e (ii) o desvio padrão do logaritmo do tempo de disparo.

Além das informações relativas a cada distribuição de probabilidade associada, o tipo do serviço é um parâmetro utilizado nas transições temporizadas. Os tipos de serviço utilizados nas transições temporizadas são: servidor infinito ou servidor simples. O servidor infinito considera a execução simultânea de ações, conforme o grau de habilitação [43] da transição correspondente. O servidor simples considera execução seqüencial de ações, ou seja, os tempos de disparo de cada transição são determinados seqüencialmente.

4. Medidas de desempenho.

A medida de desempenho, ou simplesmente métrica, define o que deve ser computado durante a simulação. É possível calcular, por exemplo, a probabilidade de termos uma determinada quantidade de *tokens* num determinado lugar. Outros tipos de métricas também podem ser definidas, como: o número médio de *tokens* num determinado lugar ou taxa de disparo de uma determinada transição, por exemplo. As métricas possuem os seguintes atributos:

- **Nome da métrica.** Identificador único da métrica.

- **Expressão.** Uma gramática foi adotada [31] para suportar a criação de métricas e expressões de guarda (Seção 4.2.3.2) na ferramenta ASTRO. Uma medida de desempenho é uma expressão que possui números, parâmetros de marcação e de tempo, operações algébricas e as seguintes medidas básicas:
 - $P\{< condicao_logica >\}$. Corresponde a probabilidade de $< condicao_logica >$ ser verdadeira.
 - $E\{< funcao_marcacao >\}$. Valor esperado da função dependente de marcação $< funcao_marcacao >$.

Funções dependentes de marcação são definições do tipo $\#Pn$, e são relacionadas ao número de *tokens* no lugar Pn . Condições lógicas geralmente contêm comparações entre funções dependente de marcação e números. Exemplos de medidas de desempenho são $E\{\#P5\}$ e $P\{\#P2 > 0\}$;

5. Rótulos.

Um rótulo é uma constante que pode ser adicionada às marcações e aos tempos de disparo para facilitar a usabilidade do modelo. Por exemplo, suponha que seja criado um modelo para calcular a confiabilidade de um determinado sistema. Nesse modelo específico os tempos de reparo de todos os componentes são sempre os mesmos. Pode-se então criar um rótulo, por exemplo, nomeado *MTTR* e atribuí-lo às transições de reparo do modelo e sempre que quisermos modificar o valor do tempo de disparo dos componentes basta modificarmos o valor do rótulo *MTTR*. Em outras palavras, funciona de modo similar à uma variável num programa de computador. Os atributos dos rótulos são:

- **Nome.** O nome do rótulo é um identificador para ser referenciado em marcações e tempos de disparos.
- **Expressão.** A expressão é substituída internamente por cada ocorrência do nome do rótulo.

As próximas seções tratam sobre o gerenciamento dos processos de simulação e do simulador SPN.

4.2.2 Gerenciador de simulação.

O gerenciador de simulação é responsável por receber os parâmetros necessários para executar o processo de simulação e passá-los para o simulador SPN. A execução da simu-

lação pode ser realizada em apenas um passo como nos casos da simulação estacionária e transiente, ou pode ser realizada em vários passos como no caso da experimentação de cenários.

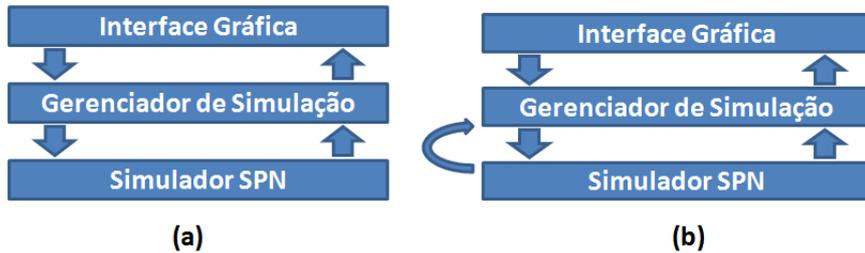


Figura 4.9: Funcionamento do gerenciador de simulação.

Na Figura 4.9 podemos visualizar o funcionamento do gerenciador de simulação nas duas situações em que é executado. No primeiro modo de funcionamento (Figura 4.9(a)) o gerenciador de simulação inicia a execução do simulador SPN utilizando os parâmetros recebidos pela interface gráfica. Assim que o processo de simulação é finalizado, os resultados são passados para a interface gráfica.

O segundo modo de funcionamento (ver Figura 4.9(b)) do gerenciador de simulação é responsável pela execução da experimentação de cenários na ferramenta. Neste modo de funcionamento são realizadas várias simulações modificando um dos parâmetros do modelo SPN (ver Seção 4.1.3). Uma vez que o processo de simulação é terminado, os resultados são passados para a interface gráfica.

4.2.3 Simulador SPN.

Simulação é o processo através do qual um modelo é avaliado numericamente, e os dados resultantes desse processo são utilizados para avaliar medidas de interesse [18]. O simulador SPN é responsável por avaliar métricas de desempenho utilizando SPN's. A estrutura do simulador pode ser vista na Figura 4.6, o simulador é composto pelos seguintes componentes: avaliador de expressões, avaliador de critério de parada, analisador de estatísticas e uma lista de eventos escalonados. Apesar de ser utilizado pelo simulador SPN, o modelo SPN não é um componente do simulador (Figura 4.6).

4.2.3.1 Esquema de simulação

Uma simulação utilizando um modelo discreto do sistema é chamado de simulação de eventos discretos. Este processo é distinto das simulações de evento contínuo, no qual o estado do sistema assume valores contínuos. O termo eventos discretos se refere

aos eventos que provocam mudança no estado do sistema. Uma simulação de eventos discretos pode usar valores de tempo discretos ou contínuos.

Antes de mostrar o esquema de simulação do ASTRO, alguns componentes do simulador de eventos discretos do ASTRO precisam ser destacados. São eles:

- ***Scheduled Event List***. Lista encadeada de eventos. Cada evento envolvido no processo de simulação está associado a uma transição. Esse evento pode ocorrer diversas vezes, portanto, é preciso guardar para cada evento uma referência para a transição do modelo e o tempo gerado pelo gerador de números aleatórios para que a ação ocorra. Esses eventos são guardados no componente *Scheduled Event List* e são ordenados em ordem crescente de tempo.
- **Variável de tempo global**. Essa variável armazena guarda o tempo de simulação, é incrementada com o tempo de cada transição que é disparada.
- **gerador de números aleatórios**. Como cada transição temporizada possui uma distribuição de probabilidade associada, um número aleatório é gerado para cada nova transição habilitada segundo a distribuição de probabilidade associada pelo usuário.

O esquema de simulação segue os seguintes passos:

1. **Inicialização do simulador**. As variáveis do simulador, incluindo o tempo global e variáveis de controle, são iniciadas com valor padrão.
2. **Verificação das transições habilitadas**. Para cada transição habilitada do modelo, é gerado um número aleatório segundo a distribuição de probabilidade associada. A lista de eventos é então atualizada.
3. **Disparo da transição com eventos associado de menor tempo**. A transição com menor tempo é disparada, a variável de tempo global do programa é atualizada e o valor da métrica (ver Seção 4.2.1) é avaliado.
4. **Atualização da *Scheduled Event List***. Dado que um transição foi disparada, a *Scheduled Event List* precisa ser atualizada. Os eventos associados às transições que se tornaram desabilitadas são removidos da *Scheduled Event List*. Em seguida, os eventos restantes da *Scheduled Event List* têm seus tempos associados subtraídos pelo tempo associado a transição disparada no passo anterior. Caso se utilize

transições exponenciais, então os tempos não precisam ser atualizados, pois a distribuição de probabilidade exponencial é *Memoryless* [35]. Em seguida, os eventos associados às transições que se tornaram habilitadas após o disparo da transição são adicionados à *Scheduled Event List*. Seus respectivos tempos de disparo são gerados segundo as distribuições de probabilidade associadas às transições.

5. **Analisa o critério de parada.** Caso o critério de parada seja atingido, a simulação é encerrada. Caso contrário volta-se ao passo 2.

4.2.3.2 Avaliador de expressões.

O avaliador de expressões de expressões da ferramenta ASTRO obedece a gramática apresentada a seguir, a mesma é utilizada para definição das métricas de desempenho.

- **Métricas de desempenho (Seção 4).** Item nomeado $\langle metrica \rangle$ da gramática apresentada a seguir.
- **Expressões de Guarda.** As expressões de guarda estão relacionadas ao item $\langle expressao \rangle$ da gramática. O item $\langle valor_real \rangle$ não pode utilizar os itens $\langle metrica \rangle$.

Notação utilizada:

<i>“simbolo”</i>	Símbolo terminal.
$\langle simbolo \rangle$	Símbolo não terminal.
$expr1 expr2$	Expressão 1 ou Expressão 2.
$expr$	Qualquer número de ocorrências.
$[expr]$	Expressão opcional.
$\langle metrica \rangle$	Definição de uma determinada métrica.

Gramática utilizada:

$\langle metrica \rangle$:	$\langle expressao \rangle$
$\langle expressao \rangle$:	$\langle valor_real \rangle$ “-”($\langle expressao \rangle$) ($\langle expressao \rangle$) ($\langle expressao \rangle$) $\langle op_num \rangle$ ($\langle expressao \rangle$)
$\langle valor_real \rangle$:	$\langle parametro_real \rangle$ $\langle constante_real \rangle$

	< <i>metrica_simples</i> > (Apenas para definições de métricas)
	< <i>integer_value</i> >
< <i>parametro_real</i> >	: < <i>identificador</i> >
< <i>constante_real</i> >	: < <i>digito</i> >{< <i>digito</i> >}“.”{< <i>digito</i> >}[< <i>expoente</i> >]
	{< <i>digito</i> >}“.”< <i>digito</i> >{< <i>digito</i> >}[< <i>expoente</i> >]
	< <i>digito</i> >{< <i>digito</i> >}[< <i>expoente</i> >]
< <i>expoente</i> >	: (“E” “e”) (“+” “?” “”) < <i>digito</i> > {< <i>digito</i> >}
< <i>metrica_simples</i> >	: “P”{< <i>condicao_logica</i> >}“”
	“E”{< <i>funcao_marcacao</i> >}“”
< <i>condicao_logica</i> >	: < <i>comparacao</i> >
	“NOT”(< <i>condicao_logica</i> >)
	(< <i>condicao_logica</i> >)
	(< <i>condicao_logica</i> >)“OR”(< <i>condicao_logica</i> >)
	(< <i>condicao_logica</i> >)“AND”(< <i>condicao_logica</i> >)
< <i>comparacao</i> >	: (< <i>funcao_marcacao</i> >) < <i>op_comp</i> > (< <i>funcao_marcacao</i> >)
< <i>op_comp</i> >	: “=” “=” “>” “<” “>=” “<=”
< <i>funcao_marcacao</i> >	: (< <i>funcao_marcacao</i> >)< <i>op_num</i> >(< <i>funcao_marcacao</i> >)
	(< <i>funcao_marcacao</i> >)
	< <i>valor_inteiro</i> >
< <i>op_num</i> >	: “+” “-” “*” “/” “^”
< <i>valor_inteiro</i> >	: < <i>constante_inteira</i> >
	< <i>parametro_inteiro</i> >
	< <i>marcacao</i> >
< <i>constante_inteira</i> >	: < <i>digito</i> >{< <i>digito</i> >}
< <i>parametro_inteiro</i> >	: < <i>identificador</i> >
< <i>marcacao</i> >	: “#”< <i>nome_lugar</i> >
< <i>nome_lugar</i> >	: < <i>identificador</i> >
< <i>identificador</i> >	: < <i>letra</i> >{< <i>letra</i> > < <i>digito</i> >}
< <i>letra</i> >	: “A” “B” ... “Z” “a” “b” “...” “z”
< <i>digito</i> >	: “0” “1” ... “9”

4.2.3.3 Avaliador de critério de parada

Um dos principais desafios no processo de simulação é determinar quando deve-se parar a simulação. Caso a simulação encerre antes do tempo necessário os resultados podem não ser precisos, caso seja muito longa um grande quantidade de tempo pode ser

desperdiçada. A ferramenta ASTRO executa a simulação até o intervalo de confiança da média (ver Seção 4.2.3.5) atingir uma largura determinada pelo usuário.

Em muitos casos as observações do processo de simulação não são independentes, existem métodos para calcular a variância da média de observações correlacionadas [37], ASTRO utiliza o método das replicações independentes para avaliação do critério de parada.

Replicações independentes. Este método realiza simulações independentes utilizando diferentes valores de semente para geração de números aleatórios. O método consiste em realizar m repetições de tamanho $n + n_0$ cada, onde n_0 é o comprimento da fase transiente. As primeiras n_0 observações de cada repetição são descartadas. As etapas do método apresentado são as seguintes:

1. **Calcular a média de cada replicação:** Seja x_{ij} a j -ésima observação da replicação i .

$$\bar{x}_i = \frac{1}{n} \sum_{j=n_0+1}^{n_0+n} x_{ij} \quad i = 1, 2, \dots, m \quad (4.2)$$

2. **Calcular a média global das replicações:**

$$\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i \quad (4.3)$$

3. **Calcular a variância das médias replicadas:** O cálculo da variância é mostrado na Seção 4.2.3.5.

4.2.3.4 Removedor de período transiente.

Em simulações de estado estacionário a fase transiente da simulação deve ser removida do calculo final da computação. Existem várias técnicas de remoção do período transiente [12], a ferramenta ASTRO permite a utilização de duas delas:

- **Long run.** Uma rodada longa é utilizada para que a presença das condições iniciais não afetem o resultado final. Este método tem duas desvantagens: pode utilizar recursos desnecessariamente, além disso, é difícil garantir que a rodada seja longa o suficiente para remover o período transiente.
- **Médias em lote.** O método das médias em lotes executa uma longa simulação, dividindo-a em várias amostras de mesmo tamanho. Cada amostra é chamada de

lote. Esse método requer o estudo da variância da média de cada lote como função do tamanho do lote. Como mostrada na Figura 4.10, cada simulação de tamanho N pode ser dividida em m médias de tamanho n , de forma que $m = \lfloor N/n \rfloor$. Onde $\lfloor b \rfloor$ denota o valor truncado de b para o menor inteiro mais próximo. Seja x_{ij} a j -ésima observação do lote i .

1. Para cada lote, calcular a média do lote.

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_{ij} \quad i = 1, 2, 3, \dots, m \quad (4.4)$$

2. Calcular a média global.

$$\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i \quad (4.5)$$

3. Computar a variância das médias dos lotes.

$$Var(\bar{x}) = \frac{1}{m-1} \sum_{i=1}^m (\bar{x}_i - \bar{\bar{x}})^2 \quad (4.6)$$

Aumenta-se n e repete-se os passos de 1 a 3, para $n = 3, 4, 5, \dots$ o intervalo transiente é encontrado quando o valor de n resultar em uma queda brusca na variância. O leitor mais interessado pode verificar mais detalhes sobre a técnica apresentada em [37].

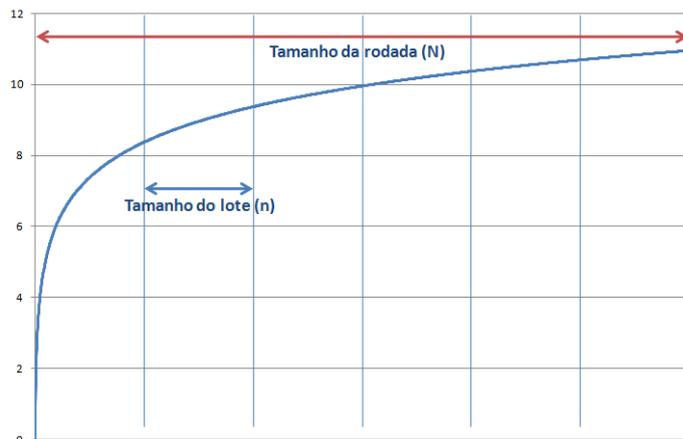


Figura 4.10: Método das médias em lotes. (Tempo de simulação X valor da métrica)

4.2.3.5 Analisador de estatísticas.

Este componente analisa as amostras geradas pelo simulador para fornecer dados ao analisador de critério de parada e removedor de período transiente. O analisador de estatísticas é responsável por fornecer os seguintes dados:

Seja x_{ij} a j -ésima observação da replicação ou lote i .

- **Média da replicação \bar{x}**

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_{ij} \quad i = 1, 2, 3, \dots, m \quad (4.7)$$

- **Média global $\bar{\bar{x}}$**

$$\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i \quad (4.8)$$

- **Variância amostral iterativa s_n^2 .**

O cálculo da variância é utilizado de forma iterativa [20], ou seja, não é necessário guardar os valores das amostras como acontece no cálculo padrão da variância. Sendo assim um número pequeno de variáveis são guardadas para a computação da variância. O cálculo da variância iterativa s_n^2 é mostrado a seguir:

$$s_n^2 = \frac{(n-2)s_{n-1}^2 + (x_n - \bar{x}_n)(x_n - \bar{x}_{n-1})}{n-1}, \quad n > 1 \quad (4.9)$$

Onde n é o número da iteração, x_n é o valor da amostra da iteração n , \bar{x}_n é o valor médio das amostras na iteração n .

- **Intervalo de confiança**

Seja a média \bar{x} , sua variância $Var(\bar{x})$, o intervalo de confiança para a média com $100(1 - \alpha)\%$ de confiança é dado por:

$$[\bar{x} - z_{1-\alpha/2}Var(\bar{x}), \bar{x} + z_{1-\alpha/2}Var(\bar{x})] \quad (4.10)$$

Onde $z_{1-\alpha/2}$ corresponde ao ponto percentual $(1 - \alpha/2)$ da distribuição normal [37]. A variância da média amostral de n observações independentes pode ser obtida das observações como segue: $Var(\bar{x}) = \frac{Var(x)}{n}$.

- **Estimativa do número de replicações.** Segundo [29] o número de replicações estimado para que o intervalo de confiança possa atingir o valor desejado é dado pela expressão.

$$R \geq \left(\frac{z_{\alpha/2} \cdot s_0}{\varepsilon} \right)^2 \quad (4.11)$$

Onde R corresponde ao número de replicações, $z_{\alpha/2}$ é o valor no qual $P(-z_{\alpha/2} < Z < z_{\alpha/2}) = 1 - \alpha$, Z corresponde a distribuição normal padronizada, s_0 é a variância medida no instante da estimativa e ε é o erro esperado.

4.3 Considerações Finais.

Este capítulo apresentou em detalhes o ambiente de avaliação SPN. Inicialmente foi mostrado o processo de simulação estacionária. Em seguida, as funcionalidades de simulação transiente e experimentação de cenários foram mostradas. Por fim, foram apresentados os principais componentes do ambiente SPN.

5

Ambiente de Avaliação RBD

Como mencionado na Seção 3.4, o ambiente RBD permite a avaliação de confiabilidade e disponibilidade utilizando diagramas de blocos de confiabilidade. Adicionalmente, o ambiente RBD permite o cálculo da importância dos componentes de um determinado sistema, geração de funções estruturais e lógicas, bem como avaliação dos limites de confiabilidade do sistema. A Figura 5.8 mostra uma visão do ambiente RBD, de tal forma que o modelo RBD apresentado possui componentes arranjados em série e paralelo.

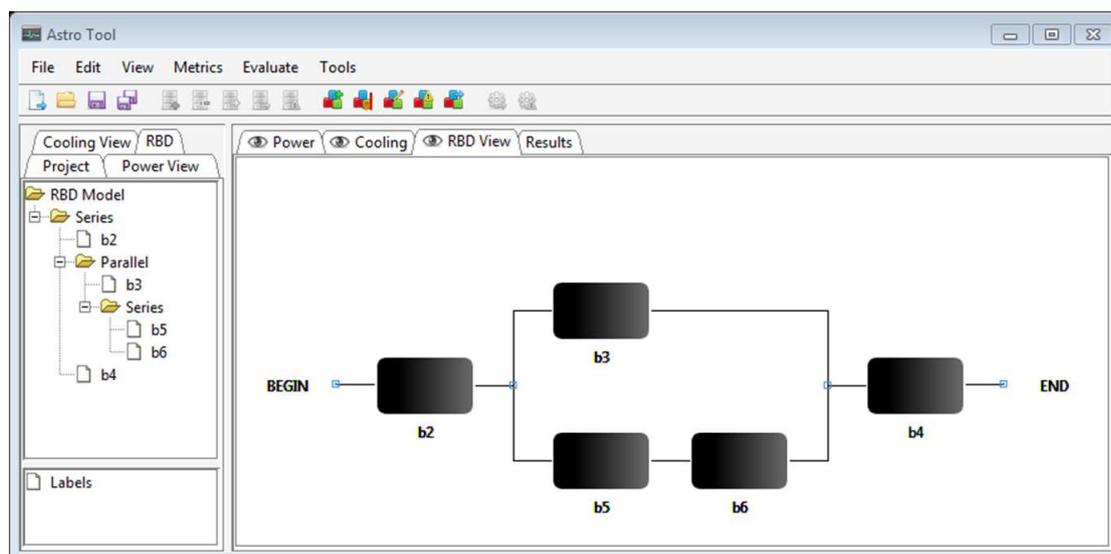


Figura 5.1: Interface gráfica do ambiente RBD

5.1 Funcionalidades do ambiente RBD.

Esta seção apresenta as funcionalidades do ambiente RBD, são elas: (i) avaliação de métricas de dependabilidade, (ii) experimentação de cenários, (iii) avaliação de importância dos componentes, (iv) avaliação dos limites de confiabilidade de um determinado sistema e (v) geração de expressões lógicas e estruturais.

5.1.1 Avaliação de métricas de dependabilidade.

Essa funcionalidade permite o cálculo de métricas de dependabilidade (ver Seção 2.2.1) utilizando modelos RBD. O ambiente RBD avalia métricas que podem ser estacionárias ou dependentes do tempo. As métricas estacionárias (ver Figura 5.2) são apresentadas a seguir:

- **Mean Time to Failure.** Tempo médio de falha (MTTF).
- **Mean Time to Repair.** Tempo médio de reparo (MTTR).
- **Steady-State Availability.** Disponibilidade associada ao modelo RBD.
- **Availability (number of 9's).** Disponibilidade associada ao modelo RBD apresentada em número de noves.

O cálculo do número de noves obedece a expressão $-\log(1 - x(\%)/100)$ [36]. Onde x é o valor percentual que se queira representar em número de noves. Além das métricas estacionárias, o ambiente RBD também permite a avaliação de métricas dependentes do tempo. As métricas dependentes do tempo (ver Figura 5.2) são apresentadas a seguir:

- **Reliability (9's).** Confiabilidade ($R(t)$) associada ao tempo **TIME** (ver Figura 5.2). Observe que o valor da confiabilidade é apresentada em número de noves.
- **Availability (9's).** Disponibilidade instantânea ($Av(t)$) apresentada em número de noves associada ao tempo **TIME** (ver Figura 5.2).

Observe que as métricas dependentes do tempo podem ser apresentadas em modo textual ou gráfico (ver Figura 5.2). O gráfico apresentado relaciona os valores das métricas dependentes do tempo (eixo das ordenadas) e seus respectivos tempos (eixo das abscissas).

5.1. FUNCIONALIDADES DO AMBIENTE RBD.

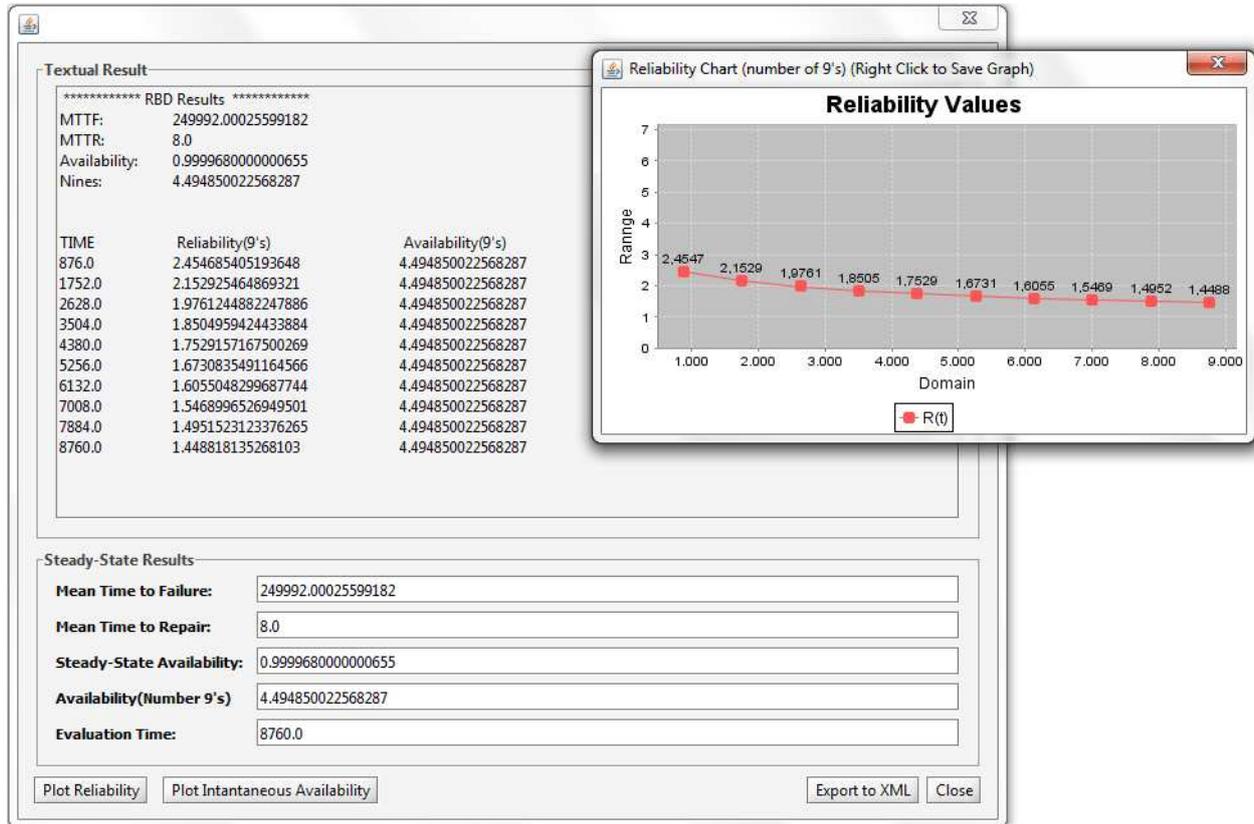


Figura 5.2: Exemplo de avaliação RBD.

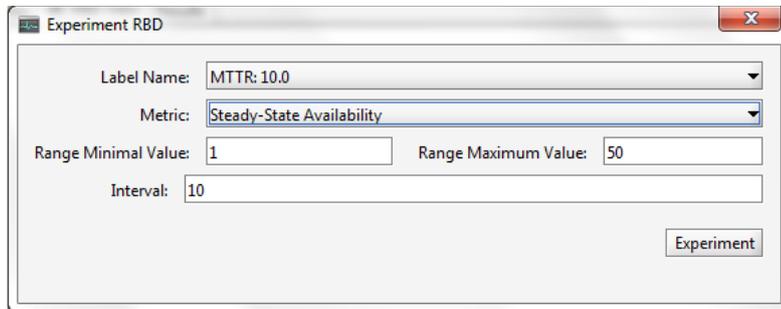
5.1.2 Experimentação de cenários.

Similarmente ao que ocorre no ambiente SPN, a experimentação de cenários no ambiente RBD é a funcionalidade na qual o ASTRO executa várias avaliações variando um determinado parâmetro do modelo. Rótulos (ver Subseção 2) são utilizados como variáveis que têm seu valor alterado ao longo do processo de experimentação. A Figura 5.3(a) apresenta os parâmetros da experimentação RBD, as entradas são:

- **Label Name.** Corresponde ao nome do rótulo.
- **Metric.** Corresponde à métrica a ser avaliada.
- **Range Minimal Value.** Corresponde ao menor valor da métrica a ser avaliada.
- **Range Maximum Value.** Corresponde ao maior valor da métrica a ser avaliada.
- **Range Minimal Value.** Este parâmetro corresponde ao passo incremental utilizado no processo de experimentação.

5.1. FUNCIONALIDADES DO AMBIENTE RBD.

Na Figura 5.3(b) é possível ver um exemplo de resultado da experimentação de cenários no ambiente RBD. No eixo das abscissas são apresentados os valores associados ao rótulo. No eixo das ordenadas temos os valores das métricas relacionadas a cada valor do rótulo. Apesar do resultado ser apresentado em forma gráfica, pode ser convertido para um arquivo no formato XML.



Experiment RBD

Label Name: MTTR: 10.0

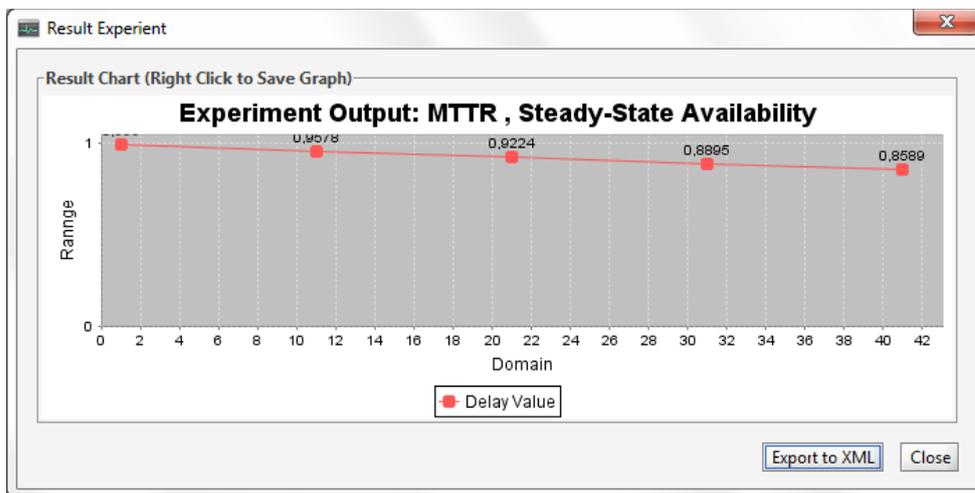
Metric: Steady-State Availability

Range Minimal Value: 1 Range Maximum Value: 50

Interval: 10

Experiment

(a) Experimentação de cenários - Parâmetros de entrada.



(b) Experimentação de cenários - Resultado exemplo.

Figura 5.3: Experimentação de cenários.

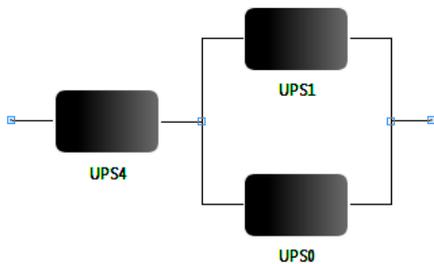
5.1.3 Avaliação de importância dos componentes.

Projetistas são frequentemente confrontados com a tarefa de identificar o componente menos confiável do sistema, a fim de melhorar o projeto. Nesse caso, melhorar ou aplicar redundância ao componente menos confiável afeta significativamente a confiabilidade do sistema. Para permitir a avaliar a importância dos componentes em termos de confiabilidade, a ferramenta ASTRO oferece a funcionalidade avaliação de importância dos componentes. A importância de confiabilidade (I_{R_i}) de um componente i de um

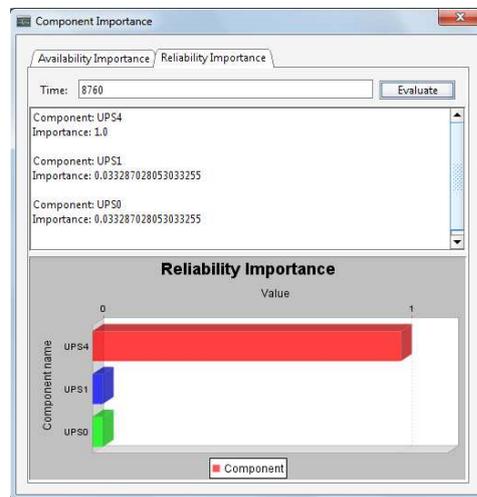
sistema s é dada por [25]:

$$(I_{R_i}) = \frac{\partial R_s}{\partial R_i} \quad (5.1)$$

R_s é a confiabilidade do sistema e R_i é a confiabilidade do componente i . Outra representação para importância dos componentes é apresentada na Secção 5.2.3. A Figura 5.4(a) apresenta um sistema composto por 3 componentes, cada um dos componentes é avaliado quanto a sua importância em termos de confiabilidade. A Figura 5.4(b) apresenta o resultado da importância dos componentes mostrados na Figura 5.4(a). O parâmetro de entrada para a avaliação é o tempo. O resultado mostra o valor da importância de cada componente em modo textual e gráfico. O resultado da importância de cada componente é normalizado em relação à importância do componente de maior valor.



(a) Importância de confiabilidade - Exemplo.



(b) Importância de confiabilidade - Exemplo de resultado.

Figura 5.4: Importância de confiabilidade

5.1.4 Avaliação de dependabilidade por limites.

Avaliação de dependabilidade por limites é um método para calcular as métricas de dependabilidade (por exemplo, confiabilidade) em modelos RBD's grandes e complexos. Essa técnica pode fornecer aproximações para o valor exato da métrica mais rapidamente do que resolver todas as equações de forma fechada. Em suma, os limites são obtidos utilizando caminhos mínimos e/ou cortes mínimos. Caminho mínimo é o número mínimo de componentes em estado operacional que garantem o funcionamento

5.1. FUNCIONALIDADES DO AMBIENTE RBD.

do sistema, ao passo que corte mínimo é o número mínimo de componentes em estado de falha que levam à falha do sistema. Em outras palavras, ao invés de apresentar um valor exato para uma determinada métrica, a ferramenta apresenta limitantes superior e inferior para o valor da métrica. Dentro desses limites o valor exato está contido. Os limites são calculados de forma iterativa, assim, o ambiente ASTRO mostra os limites estimados de acordo com o número de passos determinado pelo usuário, a medida que o número de passos aumenta os valores dos limitantes superior e inferior vão se aproximando do valor exato (ver gráfico da Figura 5.5), caso o usuário não se satisfaça com o intervalo dos limites apresentado ele pode aumentar o número de passos até que o valor exato seja encontrado.

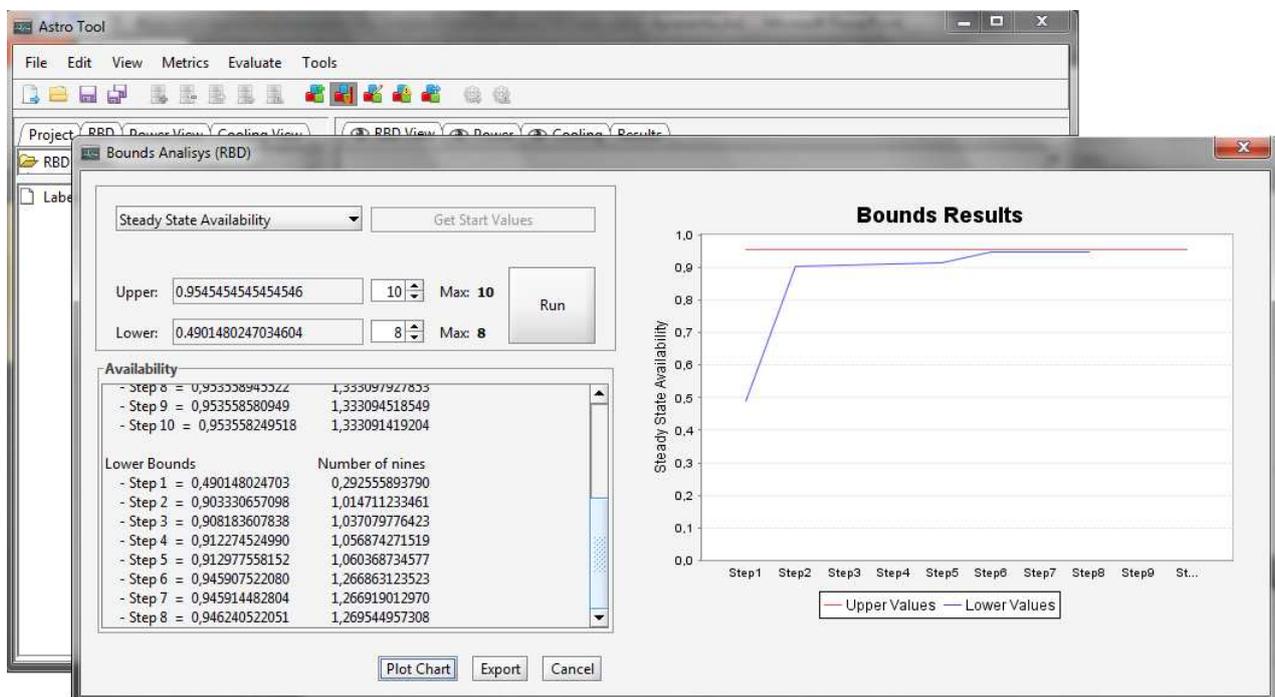
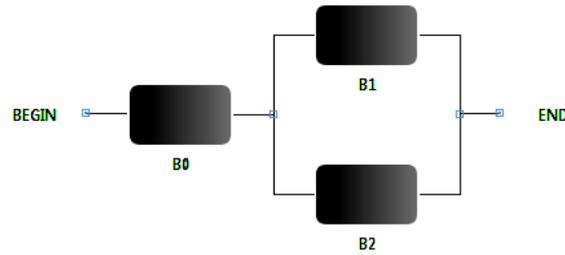


Figura 5.5: Exemplo de resultado da avaliação de dependabilidade por limites.

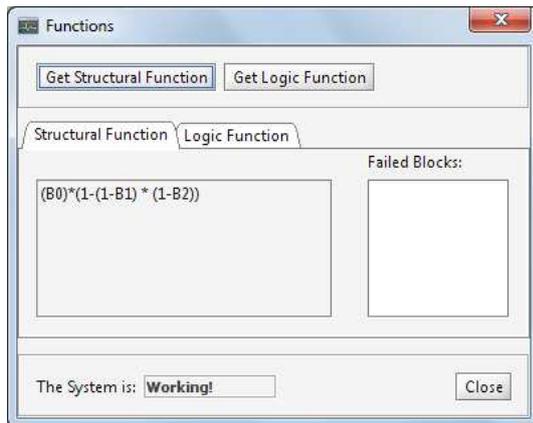
5.1.5 Geração de expressões lógicas e estruturais.

Funções estruturais e lógicas são formas alternativas de representar matematicamente o sistema, na qual a primeira adota função algébrica e a segunda utiliza a expressão lógica. Tais representações podem indicar componentes do sistema que podem ser replicados para aumentar a disponibilidade global de uma maneira mais simples. As Seções 2.4.3.1 e 2.4.3.2 apresentam em detalhes a funcionalidade em destaque. A Figura 5.6 apresenta um modelo RBD e suas respectivas funções estrutural e lógica

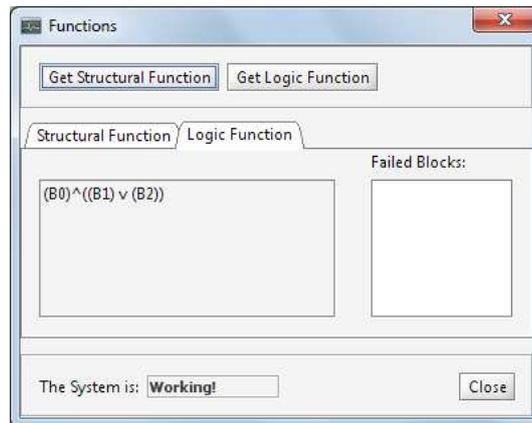
geradas pela ferramenta ASTRO.



(a) Exemplo de Modelo RBD.



(b) Função Estrutural Exemplo.



(c) Função Lógica Exemplo.

Figura 5.6: Exemplo de modelo RBD e suas respectivas funções lógicas e estruturais.

5.2 Estrutura do ambiente RBD.

A Figura 5.7 mostra a estrutura do ambiente RBD. As próximas seções apresentam os componentes destacados na Figura 5.7, com exceção dos Geradores de funções lógicas e estruturais. A descrição das funções lógicas e estruturais RBD são mostradas na Seção 2.4.3.

5.2.1 Interface gráfica.

A interface gráfica do ambiente RBD é apresentada na Figura 5.8. A interface gráfica possui seguintes componentes: área de desenho, área de rótulos, árvore estrutural do estrutura RBD, barra de menus e barra de ícones.

1. Área de desenho.



Figura 5.7: Principais componentes do ambiente RBD.

A área de desenho permite ao usuário adicionar e configurar os componentes RBD. As informações associadas aos componentes (ver Figura 5.8) podem ser representadas em termos de taxas (falha($1/MTTF$) e reparo($1/MTTR$)) ou tempos ($MTTF$ e $MTTR$). Os componentes são adicionados em série, paralelo, *bridge* e *k out of n*.

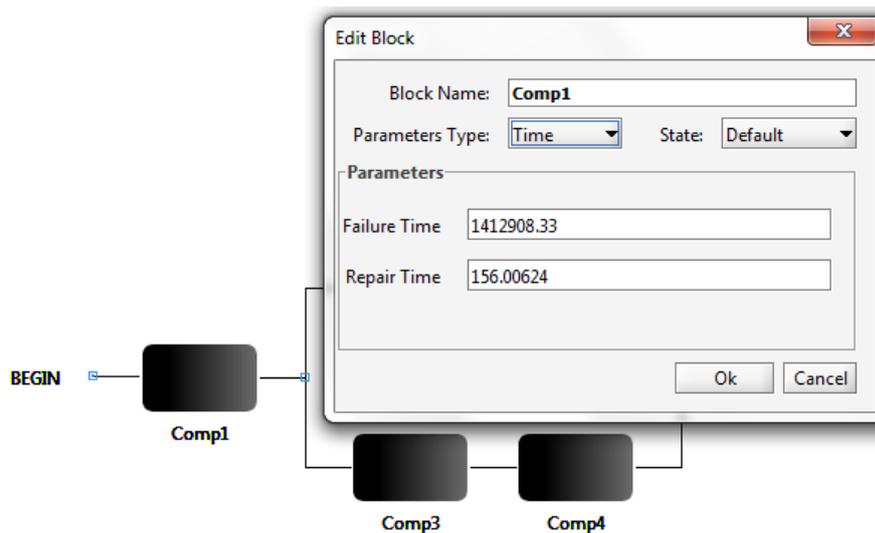


Figura 5.8: Interface RBD - Área de desenho.

2. Área de rótulos.

Assim como na interface SPN (ver secção 5), um rótulo RBD é uma constante que pode se adicionada às propriedades dos componentes para facilitar a usabilidade do modelo. Os rótulos RBD podem ser adicionados aos tempos de falha e de recuperação de cada componente. Os atributos dos rótulos são:

- **Nome.** Corresponde ao identificador do rótulo.
- **Valor.** Valor real utilizado em cada ocorrência do rótulo.

A Figura 5.9 apresenta a área de rótulos de um modelo RBD.

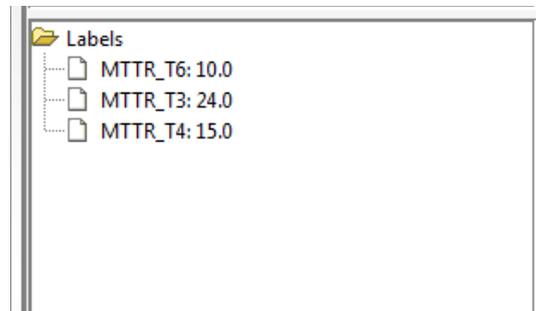


Figura 5.9: Interface RBD - Área de rótulos.

3. **Árvore estrutural do RBD.**

A árvore estrutural RBD permite além do acesso às propriedades dos componentes, uma rápida visualização da estrutura do modelo RBD em forma de árvore. A Figura 5.10 apresenta a árvore estrutural de um modelo RBD de exemplo.

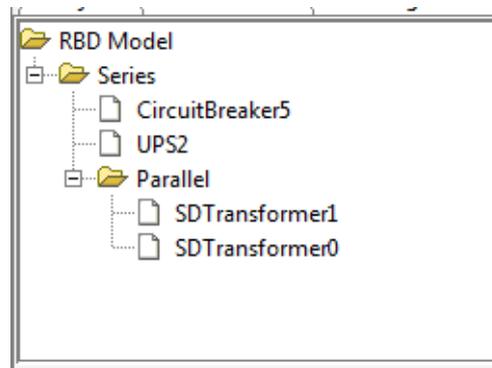


Figura 5.10: Interface RBD - Árvore estrutural.

4. **Barra de ícones.**

Através da barra de ícones as funcionalidades RBD podem ser acessadas. As funcionalidades do ambiente RBD são explicadas em detalhes na secção 5.1. A Figura 5.11 apresenta a barra de ícones do ambiente RBD.



Figura 5.11: Interface RBD - Área de menus.

5.2.2 Gerenciador de avaliação RBD.

O gerenciador de avaliação RBD é o componente responsável por gerenciar o funcionamento dos seguintes componentes: avaliador de métricas de dependabilidade, avaliador de importância dos componentes e avaliador de dependabilidade por limites. Além disso, o gerenciador de avaliação RBD é responsável por controlar o processo de experimentação de cenários. O gerenciamento do processo de experimentação de cenários é análogo ao que ocorre no ambiente SPN (ver Seção 4.2.2).

5.2.3 Avaliador de importância.

O avaliador de importância dos componentes executa os cálculos mostrados na Seção 5.1.3 de forma simplificada. Ao invés de calcular a derivada parcial mostrada na definição padrão, a ferramenta executa o cálculo mostrado a seguir:

$$I_{R_i} = R_s(1_i, \mathbf{p}^i) - R_s(0_i, \mathbf{p}^i) \quad (5.2)$$

Onde (X_i, \mathbf{p}^i) representa o vetor de componentes quando o componente i está no estado X , R_s representa a confiabilidade do sistema. Essa forma de cálculo é mostrada em [36] e facilita bastante o cálculo da importância dos componentes.

5.2.4 Núcleo de avaliação e Avaliador RBD padrão.

O núcleo de avaliação é uma interface entre o gerenciador de avaliação e os dois avaliadores RBD (ver Figura 5.7). Este componente permite ao gerenciador de avaliação selecionar o tipo de avaliação que deve ser utilizada, ou seja, avaliação por limites ou avaliação RBD padrão.

A avaliação RBD padrão executa os cálculos mostrados na Seção 2.4.3. Os resultados obtidos pelo método padrão são exatos, pois utilizam equações fechadas.

5.2.5 Avaliador RBD por limites.

Existem vários métodos para avaliação de métricas de dependabilidade em modelos RBD, esses métodos normalmente geram valores exatos como resultado. Para sistemas

complexos, pode ser muito custoso encontrar o valor exato da avaliação.

Existem técnicas de avaliação que ao invés de gerar resultados exatos, produzem um intervalo no qual o valor exato está contido. Esse intervalo é formado pelos limites superior e inferior. Apesar do resultado ser apresentado como um intervalo, esse método é bem mais rápido que os que geram valores exatos [36].

O método utilizado para encontrar esses limites é o método da soma dos produtos disjuntos [10], que provê sucessivos limitantes inferiores e superiores. Em linhas gerais, o algoritmo é executado em passos e à medida que o número de passos aumenta intervalo no qual o valor exato está contido fica mais estreito. O resultado de uma avaliação utilizando o método por limites é apresentado na Figura 5.5.

5.3 Considerações finais

Este capítulo apresentou a estrutura e funcionalidades do ambiente RBD. As funcionalidades apresentadas foram: avaliação de métricas de dependabilidade, experimentação de cenários, avaliação de importância dos componentes, avaliação de dependabilidade por limites e geração de expressões lógicas e estruturais. Além das funcionalidades, os componentes internos do ambiente RBD foram apresentados. Inicialmente, foi mostrada a interface gráfica com seus subcomponentes. Podemos destacar, gerenciador de avaliação RBD, avaliador de importância dos componentes, núcleo de avaliação, avaliador RBD, avaliador RBD por limites.

6

Ambientes de Avaliação *Data Center*

A ferramenta ASTRO além de permitir a avaliação de dependabilidade em sistemas genéricos, a mesma também possui dois ambientes especiais para avaliação de sustentabilidade e dependabilidade em sistemas *data center*. Os dois ambientes avaliam, respectivamente, as infraestruturas de resfriamento e de alimentação energética dos sistemas *data center*. Ambos os ambientes consideram a estrutura do *data center* para gerar modelos de dependabilidade e avaliar o impacto do uso dessas infraestruturas em termos de sustentabilidade e custos econômicos. Este capítulo apresenta os ambientes de avaliação *data center* destacando suas funcionalidades e estrutura.

6.1 Ambiente de avaliação da infraestrutura energética para sistemas *data center*.

O ambiente de avaliação da infraestrutura energética para sistemas *data center* (ver Seção 2.1), ou simplesmente ambiente de potência, fornece uma visão de alto nível para a modelagem de infraestruturas de potência do *data center*. Nesse ambiente o projetista define os componentes do sistema de energia do *data center*, seus respectivos atributos de dependabilidade e sustentabilidade, bem como sua estrutura. Basicamente, o ambiente fornece ícones que representam os equipamentos do sistema de energia, de tal forma que o projetista especifica a relação entre os componentes através de arcos direcionados. Por exemplo, se um componente ‘A’ aponta para o componente ‘B’, então a energia passa do componente ‘A’ para o ‘B’.

Neste ambiente, o modo de falha é definido adotando pontos de origem (chamados de *SourcePoints*) e de destino (chamados de *TargetPoints*), que indicam a direção do fluxo de energia que deve ser mantido para manter o sistema funcionando. Para que o sistema

6.2. AMBIENTE DE RESFRIAMENTO PARA SISTEMAS DATA CENTER.

de energia de um determinado *data center* esteja funcionando, todos os *TargetPoints* precisam receber energia proveniente de pelo menos um *SourcePoint*. Quando a energia passa de um componente para o outro parte dela se transforma de calor, então não somente a estrutura de um determinado sistema energético estabelece se o mesmo está operacional ou não, mas também as eficiências e potências máximas de cada componente do sistema também devem ser consideradas na hora de estabelecer seu funcionamento.

Como mencionando anteriormente, uma das características proeminentes dos ambientes de avaliação *data center* é que a avaliação de qualquer sistema pode ser armazenada para uso posterior. Por exemplo, suponha um sistema S1, que é composto por dois sub-sistemas idênticos, S2 e S3. Para avaliar o sistema S1, é possível avaliar separadamente S2 e S3, e, em seguida, avaliar S1 reaproveitando os resultados obtidos anteriormente.

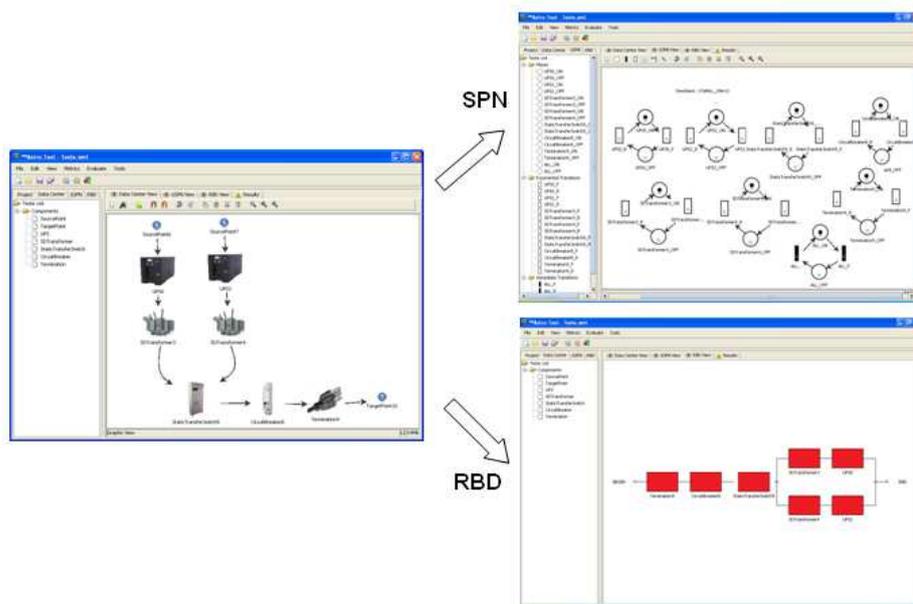


Figura 6.1: Conversão de modelos do ambiente de potência para SPN/RBD.

Como apenas estruturas de alto nível são adotadas neste ambiente, as representações do sistema de energia devem ser convertidas em RBD ou SPN para permitir a avaliação de dependabilidade. Todas as funcionalidades dos ambientes RBD e SPN podem ser acessadas diretamente dos ambientes de avaliação de potência.

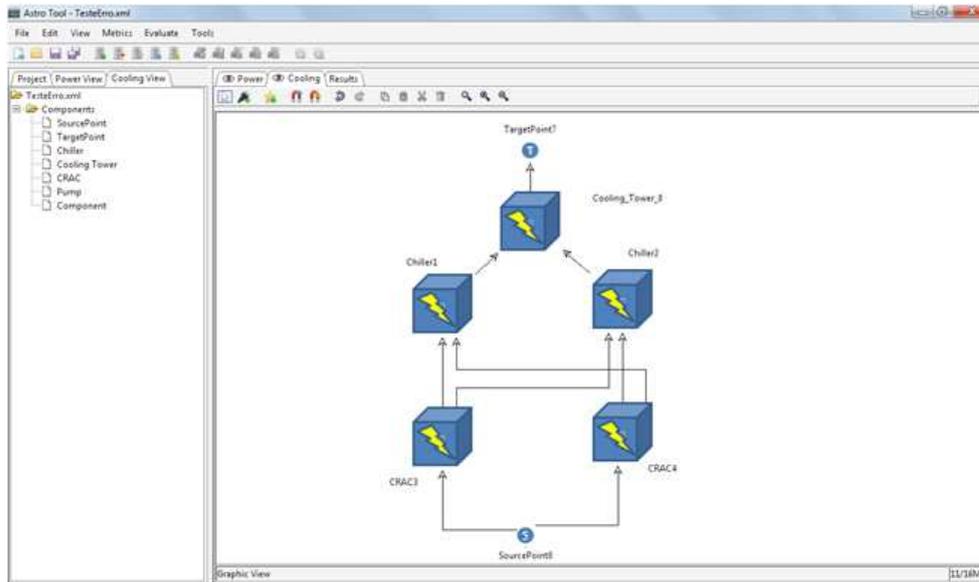


Figura 6.2: Interface gráfica do ambiente de resfriamento.

6.2 Ambiente de resfriamento para sistemas data center.

O ambiente do sistema de resfriamento (Figura 6.2) é muito semelhante ao ambiente do sistema de potência, no sentido de que o projetista pode adotar modelos de alto nível para representar o sistema de resfriamento para ambientes *data center*. No entanto, as relações entre os componentes modelados no ambiente de resfriamento representam fluxos de calor e não de energia elétrica. Da mesma forma, este ambiente também fornece funcionalidades para traduzir os modelos de alto nível em modelos RBD ou SPN (Figura 6.3) e também permite a avaliação de sustentabilidade e custos econômicos em sistemas *data center*.

Apesar de serem diferentes, os ambientes de avaliação *data center* possuem a mesma estrutura interna e as mesmas funcionalidades. Portanto, as próximas seções serão relacionadas aos dois ambientes de avaliação.

6.3 Funcionalidades dos ambientes *data center*.

Os Ambientes de avaliação *data center* possuem duas funcionalidades principais, a primeira corresponde a tradução de modelos de energia em modelos SPN e RBD, permitindo usuários não especializados abstrair detalhes da modelagem SPN e RBD. A partir dos ambientes de avaliação *data center* é possível acessar todas as funcionalidades

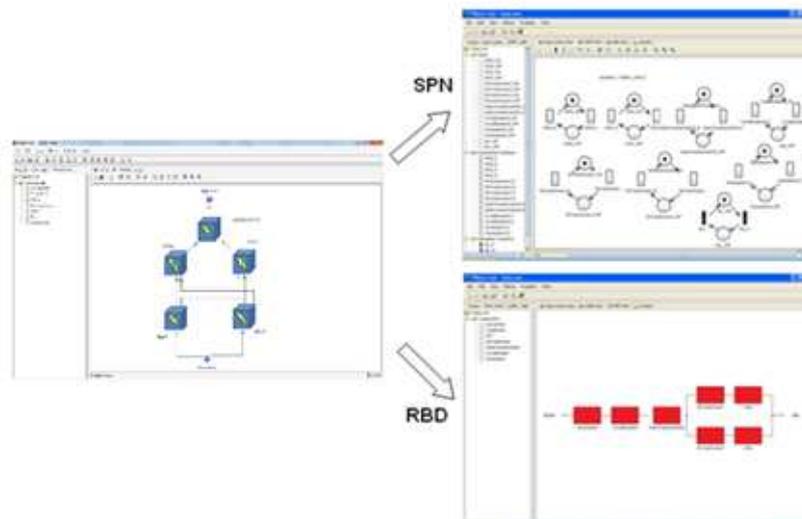


Figura 6.3: Conversão de modelos do ambiente de resfriamento para SPN/RBD.

dos ambientes RBD e SPN, tornando possível avaliar o sistema *data center* sem ter que necessariamente acessar os ambientes de baixo nível. A segunda funcionalidade corresponde a avaliação de sustentabilidade em termos de consumo exergético.

- **Tradução de modelos *data center* em SPN e RBD.**

Essa funcionalidade traduz cada componente do modelo energético em um componente do modelo SPN/RBD, a estrutura do modelo de energia define a estrutura do modelo RBD e a condição de falha do modelo SPN. A conversão entre modelos dos ambientes *data center* em SPN e RBD é realizada sem a necessidade de parâmetros de conversão. A Seção 6.4.2 apresenta os algoritmos utilizados na tradução dos modelos *data center* em SPN e RBD.

- **Avaliação de sustentabilidade.**

Os ambientes *data center* permitem avaliação do impacto da infraestrutura *data center* em termos de consumo de exergia. Essa funcionalidade também calcula o custo total da infraestrutura (TCO), que leva em conta o custo para construir o sistema (custo inicial) e os custos para manter o sistema em modo de operação (custo operacional). Para realizar a avaliação da sustentabilidade, o projetista deve especificar, para cada componente a energia incorporada na fase de fabricação e sua respectiva eficiência (ver Figura 6.5 (a)), além de outros parâmetros de sustentabilidade que são compartilhados por todos os componentes.

6.4. ESTRUTURA DOS AMBIENTES DE AVALIAÇÃO *DATA CENTER*.

Os parâmetros de entrada e resultados apresentados a seguir são descritos na Seção 2.4.2. A Figura 6.4 mostra os parâmetros requeridos para avaliação de sustentabilidade da ferramenta ASTRO.

The screenshot shows a window titled "Sustainability Evaluation" with a close button (X) in the top right corner. The window is divided into four main sections, each with a title bar and a list of input fields:

- Embedded Exergy Parameters:**
 - Manufacturing second-law efficiency: 0.5
 - Fraction of waste reused (%): 10
- Operational Exergy Parameters:**
 - System Output Power (kw): [empty field]
 - Total Lifetime (h): [empty field]
- Dependable Parameters:**
 - Availability: 0.9999600000001279
 - Automatic Compute Availability by: RBD SPN
- TCO Parameters:**
 - Electricity Cost (per KWH): 0.11

At the bottom of the window, there are three buttons: "OK", "Cancel", and "Evaluate".

Figura 6.4: Parâmetros globais na avaliação de sustentabilidade.

A Figura 6.5 apresenta os parâmetros utilizados em cada componente para avaliação de sustentabilidade, note que além dos parâmetros de sustentabilidade também são requeridos os parâmetros de dependabilidade para avaliação do sistema *data center*.

A Figura 6.6 apresenta um exemplo de resultado para avaliação de sustentabilidade em sistemas *data center*.

6.4 Estrutura dos Ambientes de Avaliação *Data Center*.

Os dois ambientes de avaliação *data center* possuem a mesma estrutura, a Figura 6.7 mostra a estrutura dos dois ambientes, são eles: Editor Gráfico (Interface Gráfica), Mod-

6.4. ESTRUTURA DOS AMBIENTES DE AVALIAÇÃO *DATA CENTER*.

The screenshot shows a configuration window for an energy component. The fields are as follows:

Field	Value
Type	UPS
Picture	/gifs/UPS.jpg
MTTR	8.0
MTTF	250000.0
Max. Power (kW)	5.0
Embedded Energy (GJ)	3.1392
Retail Price	3600.0
Efficiency (%)	95.3

Figura 6.5: Parâmetros para cada componente do editor energético.

elo Energético (ver Seção 2.4.1), Tradutor para Modelos RBD e SPN, Avaliador de Sustentabilidade e Biblioteca de Componentes. Eles serão apresentados na próximas subseções.

6.4.1 Editor Gráfico.

A Figura 6.8 mostra a interface gráfica do ambiente *data center*. Note que o editor gráfico fornece ao usuário uma visão da estrutura do sistema *data center*.

Na interface gráfica do ambiente podemos destacar três componentes principais: barra de ícones, a área de desenho à direita e a árvore de componentes.

- *Barra de Ícones.* Através da barra de ícones pode-se adicionar os componentes à estrutura *data center* que está sendo editada, além de acessar as funções de tradução para SPN e RBD. A funcionalidade de avaliação de sustentabilidade é acessada através do menu *Evaluate* na barra de menus principal do programa.
- *Área de desenho.* Utilizando a área de desenho o projetista pode montar a infra-

6.4. ESTRUTURA DOS AMBIENTES DE AVALIAÇÃO *DATA CENTER*.

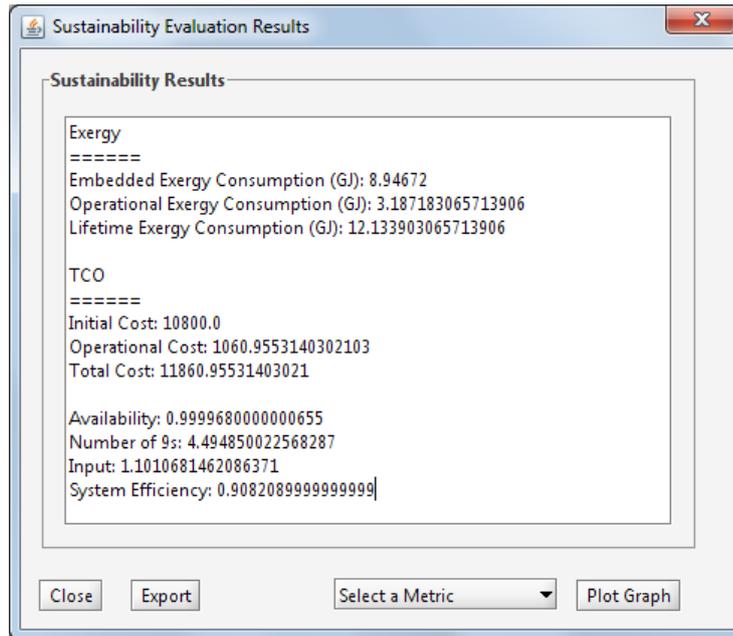


Figura 6.6: Resultado de exemplo para avaliação de sustentabilidade em estrutura *data center*.



Figura 6.7: Estrutura dos ambientes energéticos.

estrutura do sistema *data center*, os componentes da infraestrutura podem ser utilizados a partir da biblioteca dos componentes ou criado pelo usuário.

- *Árvore dos Componentes*. A árvore dos componentes apresenta os itens que podem ser adicionados à área de desenho.

6.4.2 Tradutor SPN/RBD.

Este componente é responsável pela tradução do modelo de energia em modelos RBD e SPN. Existem dois algoritmos de conversão utilizados pelo componente, tradução de modelo de energia para RBD e tradução de modelo RBD para SPN. Desta forma, a ferramenta ASTRO não traduz o modelo de energia em SPN diretamente. Inicialmente,

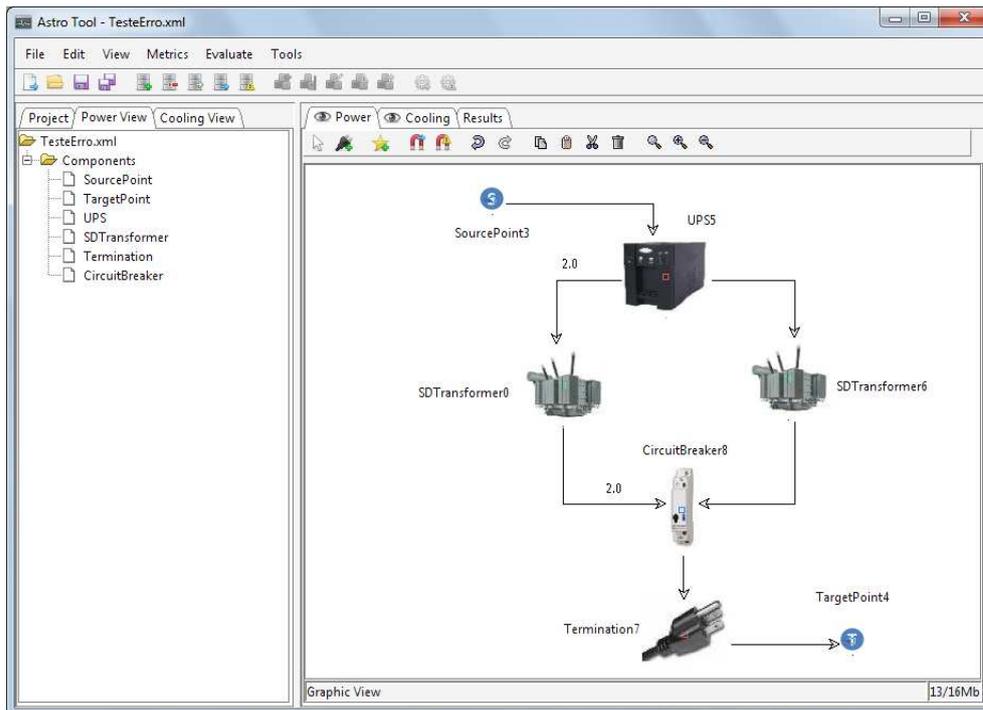


Figura 6.8: Ambiente de avaliação da estrutura de potência *data center*

o modelo de energia é convertido em RBD para em seguida o modelo RBD resultante ser convertido para SPN. Apesar da tradução dos modelos de energia em SPN ser realizada de forma indireta, o processo de tradução é transparente para o usuário. As próximas subseções vão apresentar os processos de tradução de modelos energéticos para RBD e SPN.

- **Tradução do modelo de energia para RBD.**

A Figura 6.9 mostra o algoritmo de tradução do modelo de energia em RBD. Antes do algoritmo ser executado, a variável RBDModel precisa ser inicializada. O nó inicial a ser passado como parâmetro para algoritmo na primeira chamada é o nó destino (ver Seção 2.4.1). A função `addBlock()` adiciona o bloco em série ou em paralelo dependendo das restrições de energia, caso o nó pai seja capaz de fornecer energia suficiente para alimentar o filho sem a ajuda de outro componente, então ele é adicionado em paralelo com os outros pais. Caso contrário, o componente é adicionado em série. A forma de distribuição de energia entre os componentes é mostrada na Seção 2.4.1, pode-se notar que o algoritmo apresentado corresponde a uma busca em profundidade (*depth-first search*) na estrutura *data center* a ser traduzida.

6.4. ESTRUTURA DOS AMBIENTES DE AVALIAÇÃO DATA CENTER.

```
1 traducao-energetico-rbd(nodeModeloEnergia)
2 {
3   Pais := nodeModeloEnergia.getPais();
4   if(Pais.vazio())
5   {
6     retorna RBDModel;
7   }
8
9   for-each(pai : Pais)
10  {
11    blocoRBD := criarBlocoRBD(pai);
12    RBDModel.addBlock(blocoRBD);
13    traducao-energetico-rbd(pai);
14  }
15 }
```

Figura 6.9: Tradução do modelo energético para RBD

- **Tradução do modelo RBD para SPN.**

A tradução do modelo RBD para SPN é executado em dois passos, o primeiro passo consiste em converter todos os componentes do modelo RBD em componentes básicos do modelo SPN (ver Seção 2.4.4.1), na conversão o MTTF e o MTTR de cada componente RBD é passado para o componente SPN. O próximo passo é gerar o componente de agregação (ver Seção 2.4.4.5) e sua expressão de falha e recuperação. A expressão de recuperação é a negação da expressão de falha, portanto a principal função do algoritmo de tradução do modelo RBD em SPN é gerar a expressão de falha do componente de agregação a partir do modelo RBD, dado que a conversão dos componentes RBD em SPN é trivial. A Figura 6.10 mostra o algoritmo de geração da expressão de falha do modelo SPN a partir do modelo RBD. O algoritmo apresentado corresponde a uma busca em profundidade (*depth-first search*) no modelo RBD passado como parâmetro. Observe que é feita uma verificação na variável **blocoRBD** passada como parâmetro. Se a variável corresponder a um bloco unitário, então o componente é adicionado a expressão de falha. Se a variável corresponder a um conjunto de componentes em série ou paralelo, então cada componente do conjunto é adicionado a expressão de falha.

6.4. ESTRUTURA DOS AMBIENTES DE AVALIAÇÃO DATA CENTER.

```
1  traducao-rbd-falha-spn(blocoRBD)
2  {
3    if(blocoRBD.ehBlocoUnitario())
4    {
5      retorna "(" + blocoRBD.getName() + "_ON=0"+" ";
6    }
7
8    if(blocoRBD.ehCadeiaSerieOuParalelo())
9    {
10     expr := "(";
11     for-each(newBlock : blocoRBD.filhos())
12     {
13       if(blockChain instanceof BlockSeries)
14       {
15         expr = expr + traducao-rbd-falha-spn(newBlock) + "OR";
16       }
17       else if(blockChain instanceof BlockParallel)
18       {
19         expr = expr + traducao-rbd-falha-spn(newBlock) + "AND";
20       }
21     }
21   }
21   expr := expr + ")";
21   retorna expr;
21 }
```

Figura 6.10: Geração da expressão de falha do modelo SPN a partir do modelo RBD.

6.4.3 Avaliador de Sustentabilidade.

Avaliador de Sustentabilidade é o componente que realiza a avaliação de sustentabilidade na ferramenta ASTRO, o mesmo calcula o custo e a exergia operacional de uma determinada infraestrutura utilizando as equações mostradas na Seção 2.4.2.

6.4.4 Biblioteca de Componentes.

A ferramenta ASTRO possui uma biblioteca de componentes, os componentes podem ser ajustados de acordo com cada projeto em particular, e novos componentes podem ser adicionados de acordo com a necessidade do projetista. Dado que projetos previamente avaliados podem ser reutilizados, o usuário pode avaliar sistemas complexos dividindo este sistema em pequenos subsistemas e reutilizando os valores previamente

calculados.

6.5 Considerações Finais.

Este capítulo apresentou os ambientes de avaliação *data center* da ferramenta AS-TRO. Inicialmente, foram apresentadas as funcionalidades do ambiente, são elas: avaliação de sustentabilidade em sistemas *data center* e tradução de modelos *data center* para SPN e RBD. Em seguida, foi apresentada a estrutura do ambiente *data center*. Foram mostrados os principais componentes dos ambientes *data center*, são eles: Editor Gráfico, Tradutor SPN/RBD, Avaliador de Sustentabilidade e Biblioteca de Componentes.

7

Estudos de caso.

Este capítulo apresenta dois estudos de caso para validar a ferramenta proposta. No primeiro estudo de caso, cinco infra-estruturas de energia *data center* são consideradas (ver Figura 7.1) com redundância crescente, de tal forma que cada arquitetura sucessiva tem um componente adicional duplicado. O segundo estudo de caso considera cinco arquiteturas *data center* contendo subsistemas de energia e refrigeração na mesma estrutura. Ambos os estudos de caso são analisados em termos de dependabilidade e sustentabilidade. Para os dois estudos de caso, estima-se: (i) disponibilidade, (ii) o impacto sobre a sustentabilidade em termos de exergia e (iii) o custo total de propriedade (TCO).

7.1 Estudo de caso I

Nesse estudo de caso, cinco infra-estruturas de energia *data center* são consideradas [56] (ver Figura 7.1). Para cada arquitetura, estima-se: disponibilidade, impacto em termos de sustentabilidade e custo total da infra-estrutura (TCO).

7.1.1 Descrição das arquiteturas.

Na arquitetura de referência, A1, não existem componentes redundantes. As arquiteturas A2, A3, A4 e A5 possuem dispositivos redundantes adicionais, como um UPS (A2, A3, A4, A5), um transformador (*transformer*) (A3, A4, A5), um subpainel (*subpanel*) (A4, A5) e um filtro de linha (*rack power strip*) (A5). A arquitetura A5 corresponde à arquitetura de referência (A1) com todos os componentes redundantes. Nesta representação, um STS (*static transfer switch*) não é necessário, uma vez que dispositivos de TI modernos possuem sistema de fiação dupla (*dual corded*).

As taxas de falha e de recuperação para os componentes foram obtidas de [3]. A

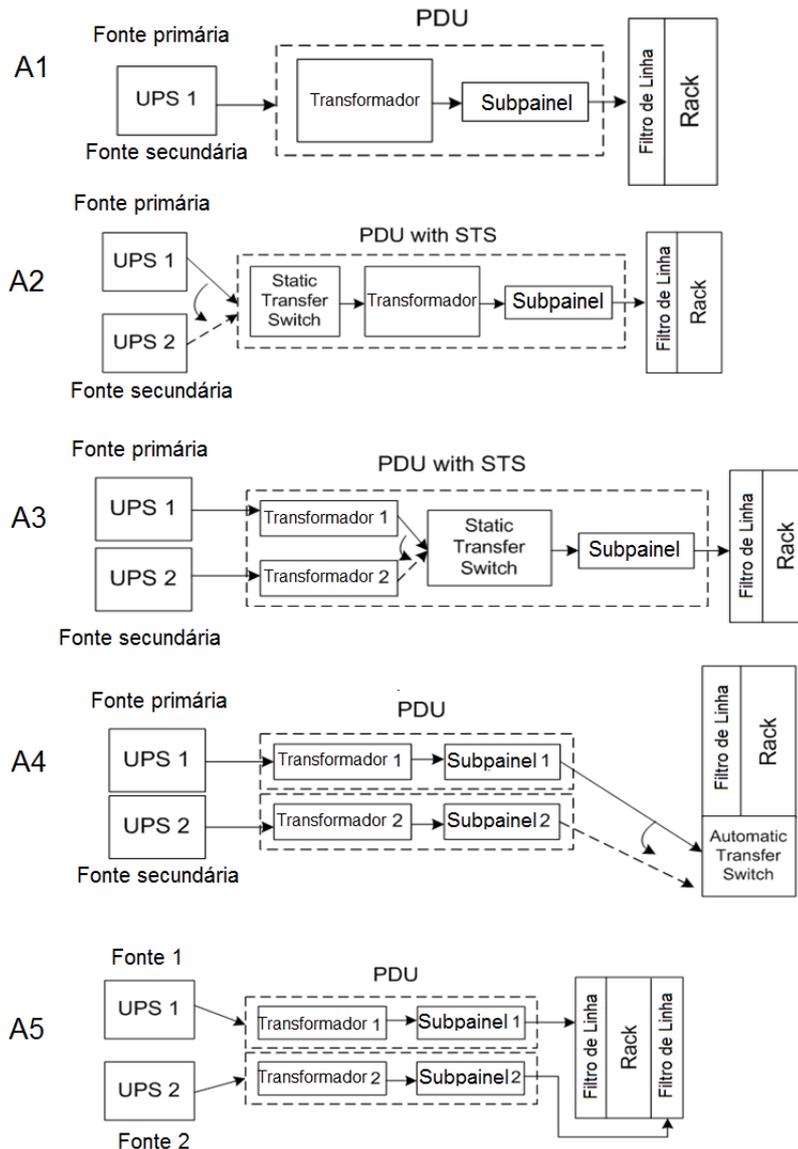


Figura 7.1: Infra-estruturas de potência *data center*.

Figura 7.2 mostra o modelo correspondente para a arquitetura A1 na interface gráfica do ambiente de potência, bem como as respectivos modelos RBD e SPN obtidos automaticamente.

Quanto ao modelo SPN (Figura 7.2c), o MTTF relacionado ao componente da fonte de alimentação ininterrupta (UPS) está associado à transição $UPS0_F$; o MTTR é relacionado à taxa de disparo da transição $UPS0_R$. Além disso, o lugar $UPS0_ON$ representa o estado operacional do componente da UPS, enquanto $UPS0_OFF$ é o estado de falha. A mesma representação é adotada para todos os outros componentes da arquitetura.

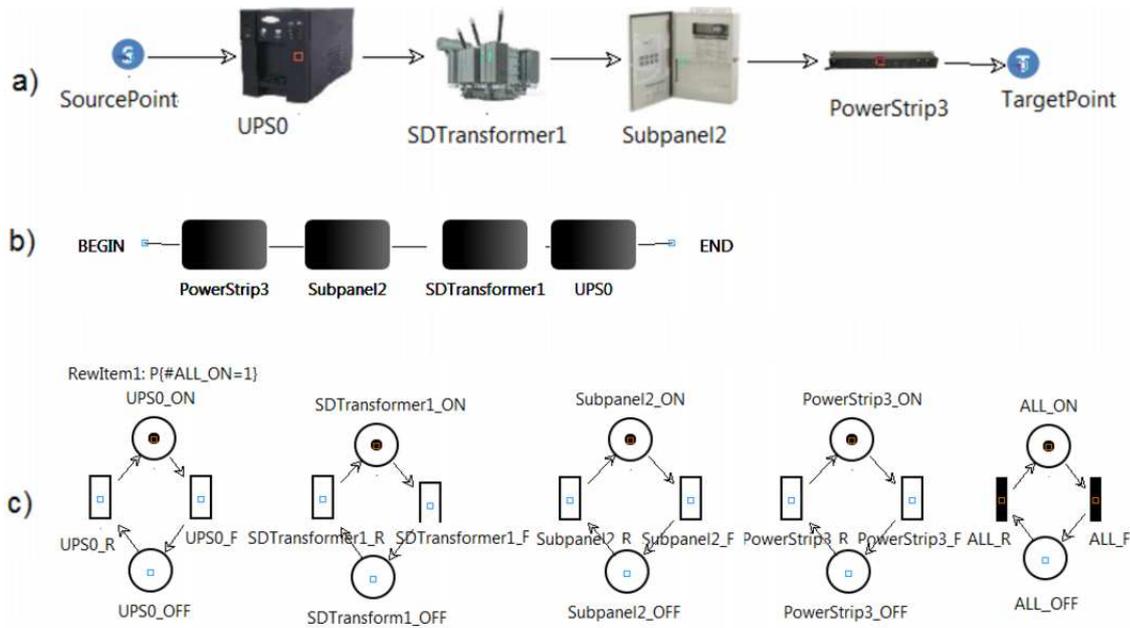


Figura 7.2: A1 (a), modelo RBD Respectivo (b) modelo SPN respectivo (c).

tura. Além disso, um token no lugar *ALL_ON* representa que o sistema está operacional, e um token no lugar *ALL_OFF* representa que o sistema falhou. Este comportamento é determinado pelas expressões de guarda nas transições *ALL_F* e *ALL_R* conforme demonstrado na Tabela 7.1.

A Figura 7.2(b) representa o modelo RBD relacionado à arquitetura A1. Observe que o modelo RBD apresenta uma composição em série, porque não há nenhuma redundância.

Tabela 7.1: Expressões de guarda para o modelo A1

Transição	Expressão de Guarda
ALL_F	$((\#PowerStrip3_ON=0)OR(\#SubPanel2_ON=0)OR(\#SDTransformer1_ON=0)OR(\#UPS0_ON=0))$
ALL_R	negação da expressão de guarda anterior.

7.1.2 Resultados

A Tabela 7.2 apresenta o resumo dos resultados da avaliação das cinco arquiteturas previamente detalhadas, em que o símbolo $9's$ representa o número de noves calculados através da seguinte equação: $-\log(1 - Avail(\%)/100)$. Como esperado, a disponibilidade aumenta com a adição de equipamentos redundantes. No entanto, note que o aumento da disponibilidade de A1 para A2 é mínimo, uma vez que, embora haja um UPS

repetido, um STS, também é introduzido em série. O número de noves dobra de A1 a A5, como esperado, uma vez que A5 representa essencialmente duas instâncias da A1 ligadas em paralelo. A disponibilidade dessas arquiteturas foram calculados utilizando (RBD) e (SPN) e os resultados obtidos são praticamente idênticos.

Tabela 7.2: Resultados para todas as arquiteturas.

Arquitetura	Disponibilidade(%) (9's)	LTE.(GJ)	TCO (USD)	Effic (%)
A1	99.98556 (3.84)	37.99	20038.43	93.3
A2	99.98627 (3.86)	44.28	24516.38	92.8
A3	99.99731 (4.57)	44.62	25068.09	92.8
A4	99.99989 (5.99)	45.03	25268.50	92.8
A5	99.99999 (7.68)	42.06	24590.67	93.3

Além dos resultados apresentados, foi feito um estudo comparativo utilizando outras duas ferramentas que permitem avaliação de dependabilidade. As ferramentas utilizadas para comparação foram: TimeNET [31] e Sharpe [48]. A Tabela 7.3 apresenta os valores de disponibilidade em termos percentuais para cada arquitetura apresentada utilizando as ferramentas citadas acima.

Tabela 7.3: Comparação dos resultados utilizando diferentes ferramentas.

Arquitetura	Astro	TimeNET	Sharpe
A1	99.98556	99.9892	99.985
A2	99.98627	99.98032	99.98596
A3	99.99731	99.99712	99.99715
A4	99.99989	99.999904	99.999931
A5	99.99999	99.999997	99.999977

O teste t emparelhado [47] foi realizado para comparação dos resultados entre o Astro e as ferramentas citadas acima. Os testes compararam os resultados das ferramentas ASTRO e TimeNET, e ASTRO e Sharpe. A comparação entre ASTRO e o TimeNET mostrou que a diferença entre as médias das duas amostras, considerando 95% para o intervalo de confiança, está entre $[-0,0037174;0,004621]$. Como o valor '0' está dentro do intervalo de confiança, pode-se inferir com 95% de certeza que estatisticamente não há evidências significativas que refutem a afirmação de que o resultado da ferramenta ASTRO é compatível com os resultados apresentados pela ferramenta TimeNET. A mesma afirmação pode ser dita entre os resultados da comparação entre as ferramentas ASTRO e Sharpe. Pois, utilizando um nível de confiança de 95% a diferença entre as médias das duas amostras está entre $[-0,0001475;0,000452]$.

A Tabela 7.4 apresenta os resultados da importância de confiabilidade para cada componente das cinco arquiteturas considerando um período de 1 ano. Note que para as arquiteturas A1 e A5, os valores das importâncias dos componentes são bem próximos.

O que indica que essas arquiteturas não apresentam um dispositivo crítico. Quando adicionamos o STS e o ATS (Arquiteturas A2 à A5), pode-se notar que a adição desses componentes fazem com que os componentes que não são duplicados fiquem com uma importância bem menor em relação ao componentes duplicados.

Tabela 7.4: Importância dos componentes.

Arquitetura	UPS1	UPS2	Transf. 1	Transf. 2	STS	Subpainel 1	Subpainel 2	Filtro Linha 1	Filtro Linha 2	ATS
A1	1.0	X	0.9715	X	X	0.9711	X	0.9663	X	X
A2	0.0332	0.0332	0.9702	X	1.0	0.9697	X	0.9649	X	X
A3	0.0387	0.0387	0.0376	0.03767	1.0	0.9697	X	0.9649	X	X
A4	0.0446	0.0446	0.0434	0.0434	X	0.0434	0.0434	0.9833	X	1.0
A5	1.0	1.0	0.9715	0.9715	X	0.9711	0.9711	0.9663	0.9663	X

Além disso, este estudo adota exergia para quantificar o impacto de sustentabilidade para cada arquitetura. A Tabela 7.2 resume os resultados obtidos utilizando a ferramenta ASTRO. Nesta tabela, *TCO* é o custo total de propriedade, que é baseado no preço de venda do componentes e os custos operacionais durante um período de 5 anos. *LTE* é o consumo de exergia (exergia operacional + exergia embutida), assumindo também um período de cinco anos.

Para uma melhor visualização, a Figura 7.3 mostra uma comparação entre a disponibilidade, o Custo Total de Propriedade (TCO), bem como a LTE. A1 representa a arquitetura de referência, note que o aumento de A1 para A2 em termos de TCO reflete o alto custo da UPS em comparação com os outros componentes. As variações de exergia são principalmente devidas ao hardware adicional, uma vez que a diferença no consumo de energia operacional é insignificante neste período de tempo. Além disso, a partir da Figura 7.3, pode ser visto que a disponibilidade aumenta significativamente de A2 para A3 (quase 20%) e de A3 para A4 (cerca de 35%), equanto o aumento do impacto em termos de sustentabilidade é desprezível. Assim, a diminuição da redundância nem sempre leva a uma redução significativa do impacto de sustentabilidade. O impacto sobre a sustentabilidade diminui A4 e A5, uma vez que um STS não é exigido em A5.

7.2 Estudo de Caso II

Neste estudo de caso, cinco arquiteturas *data centers* formadas por infra-estruturas de refrigeração e energia são analisadas. Para cada arquitetura, estima-se: disponibilidade, sustentabilidade e impacto do custo total de propriedade.

Arquiteturas de energia são responsáveis pelo fornecimento ininterrupto de energia condicionada aos dispositivos hospedados na infra-estrutura de refrigeração, bem

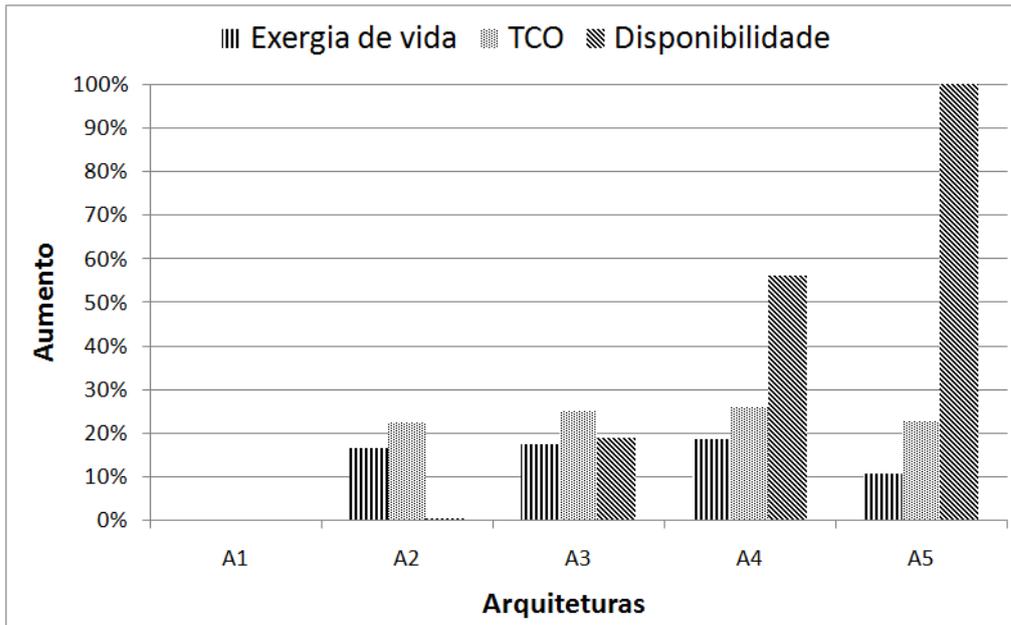


Figura 7.3: Comparação: disponibilidade, consumo de exergia da vida e TCO.

como para os racks de TI. A partir do sistema de energia, tipicamente, a energia passa pelo (UPS), unidades de distribuição de energia (*power distribution units* PDUs), composto de transformador abaixador (SDT) e subpainéis elétricos, STS's, e finalmente, *rack PDU's*. Além disso, os dispositivos podem ser replicados, a fim de aumentar a disponibilidade da arquitetura de potência.

7.2.1 Descrição das Arquiteturas.

A arquitetura 1 (ver Figura 7.4) corresponde à arquitetura de base, que é composta de dois sistemas de potência e duas infra-estruturas de refrigeração. Neste sistema, uma subestação elétrica fornece a potência de entrada para cada sistema de alimentação (composto por uma fonte de alimentação ininterrupta (UPS), uma unidade de distribuição de energia (PDU) e uma caixa de junção). Um gerador a diesel fornece energia para cada sistema de resfriamento (composto por uma torre de resfriamento, um *chiller*, duas bombas e três CRACs).

Na arquitetura 1, a energia é conduzida por uma fonte de alimentação ininterrupta (UPS), pela unidade de distribuição de energia (PDU) (composto por um transformador abaixador (SDT) e um subpainel elétrico), caixas de junção, e, finalmente, *rack PDUs* (unidades de distribuição de energia para os *racks*). Adicionalmente, considerando a arquitetura, quando um UPS falha, a energia também pode também seguir por um STS.

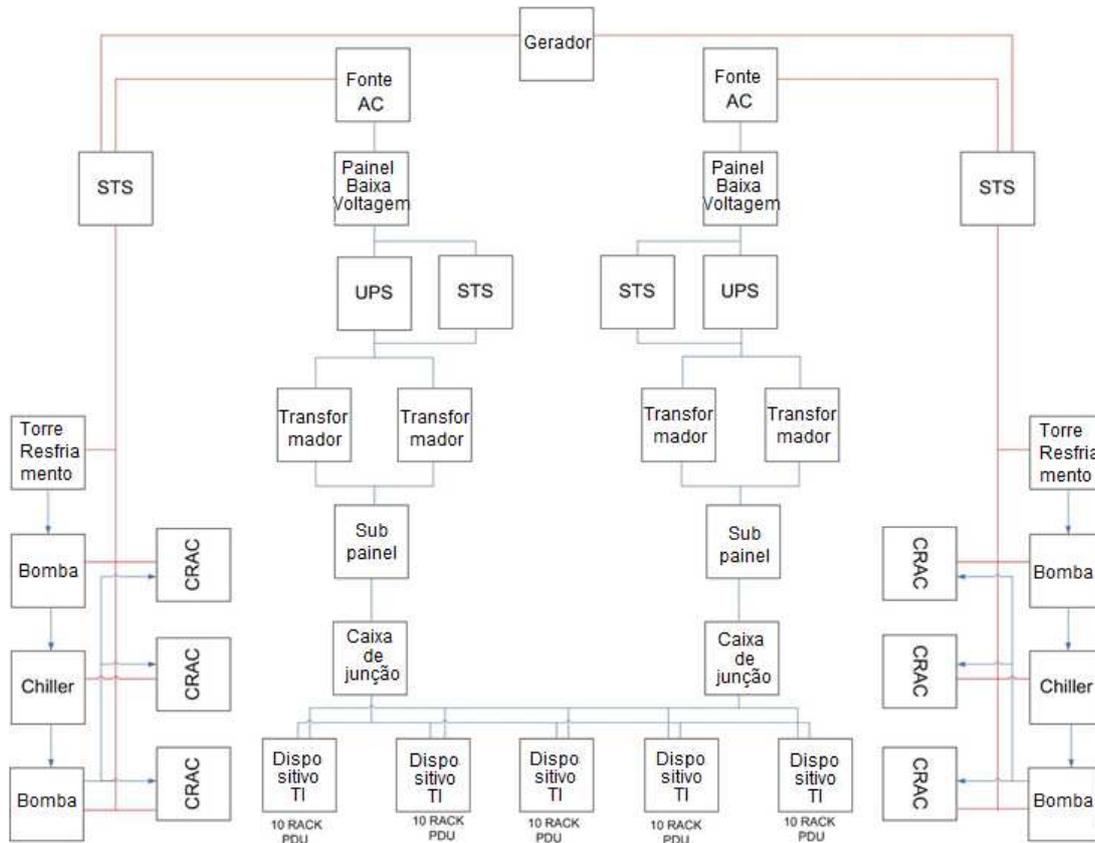


Figura 7.4: Arquitetura 1.

A diferença da arquitetura 2 para a arquitetura de referência são as conexões do STS (ver Figura 7.5). Através desta arquitetura duas subestações de energia elétrica fornecem alimentação para dois sistemas de potência. Quando ocorre uma falha em uma subestação elétrica, a outra é capaz de fornecer energia para ambos os sistemas de potência.

Quanto à arquitetura 3 (Figura 7.6), a diferença deste sistema para a arquitetura de referência são também as conexões do STS. Dessa forma, na arquitetura, duas subestações elétricas diferentes (por exemplo, fonte de alimentação e um gerador a diesel) podem fornecer energia para cada sistema de potência. O STS é capaz de escolher entre a fonte AC e o gerador à diesel. É importante afirmar que as duas subestações de energia elétrica são as fontes primárias de cada sistema de potência, e os geradores a diesel representam uma fonte em *cold standby*.

A arquitetura 4 é semelhante à arquitetura 3, porém com apenas uma subestação elétrica, como mostrado na Figura 7.7. Finalmente, a arquitetura 5 (Figura 7.8) corresponde à arquitetura de referência (ver Figura 7.4) sem transformadores redundantes.

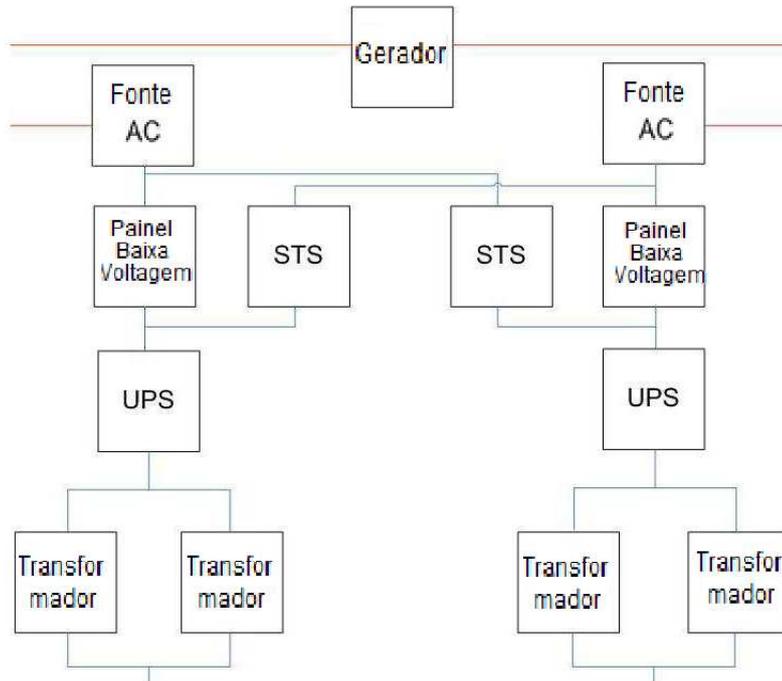


Figura 7.5: Arquitetura 2.

A Tabela 7.5 mostra os valores adotados de MTTFs e MTTRs dos dispositivos de potência (fonte AC, gerador, TI *Rack*, caixa de junção, STS, subpainel, transformador, UPS e painel de alimentação). Além disso, a Tabela 7.6 apresenta os valores adotados de MTTFs e MTTRs dos dispositivos de refrigeração (*chiller*, torre de resfriamento e CRACs).

Tabela 7.5: Parâmetros da infra-estrutura de potência

Equipamento	MTTF (hs)	MTTR (hs)
Fonte AC	4.380	8
Gerador	2.190	8
TI <i>Rack</i>	378.148.937	4
Caixa de junção	26.120.000	8
STS	240.384	8
Subpainel	1.520.000	8
Transformador	763.201	8
UPS	250.000	8
Painel de alimentação	1.520.000	8

7.2.2 Resultados

A Tabela 7.7 apresenta o resumo dos resultados e a avaliação das estruturas de potência e refrigeração dessas cinco arquiteturas. É importante afirmar que todas as arquiteturas utilizam a mesma estrutura de resfriamento. Assim, há apenas um resultado sobre

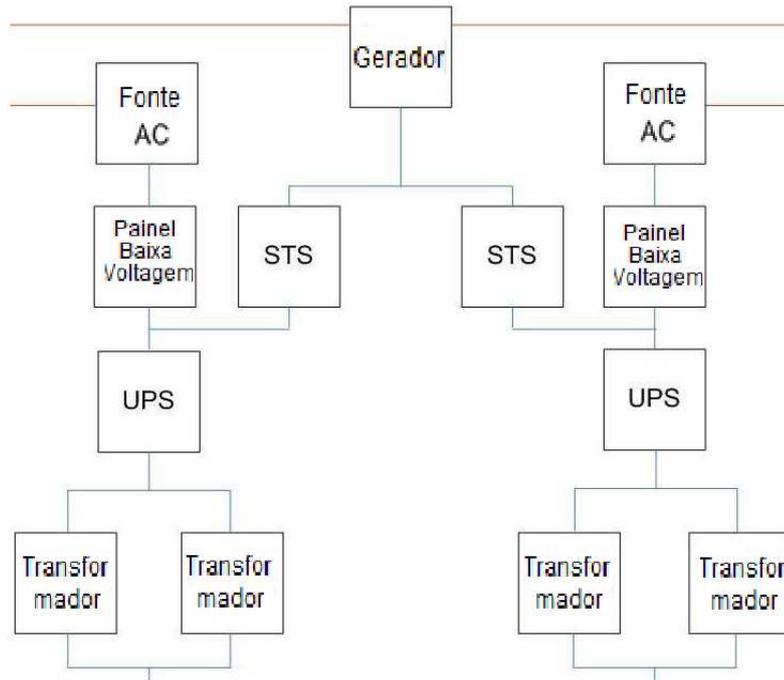


Figura 7.6: Arquitetura 3.

Tabela 7.6: Parâmetros da infra-estrutura de resfriamento

Equipamento	MTTF (hs)	MTTR (hs)
Chiller	18,000	48
Torre de resfriamento	24,816	48
CRAC	37,059	8

a arquitetura de resfriamento.

Tabela 7.7: Resumo dos resultados

Arquitetura	Disp.(%) (9's)	LTE	TCO(USD)	Downtime(h)
A1	99.9996665 (5.4769)	2084.88	787953.05	0.029213
A2	99.9996670 (5.4775)	2495.59	800502.60	0.029174
A3	99.9999988 (7.9088)	2459.21	799406.73	0.000108
A4	99.9794548 (3.6872)	2353.35	784295.20	1.799758
A5	99.9996243 (5.4251)	2084.28	786853.64	0.032909
Resfriamento	99.9979352 (4.6851)	1856.19	174141.46	0.180877

Quanto às infra-estruturas de potência, como esperado, a disponibilidade aumenta quando o nível de redundância é incrementado. Assim, a menor disponibilidade foi obtida na arquitetura A4, que corresponde à arquitetura com a menor quantidade de dispositivos (uma fonte AC está ausente). Além disso, também é possível notar que a ausência dos transformadores redundantes (em A5) não são significativas para a disponi-

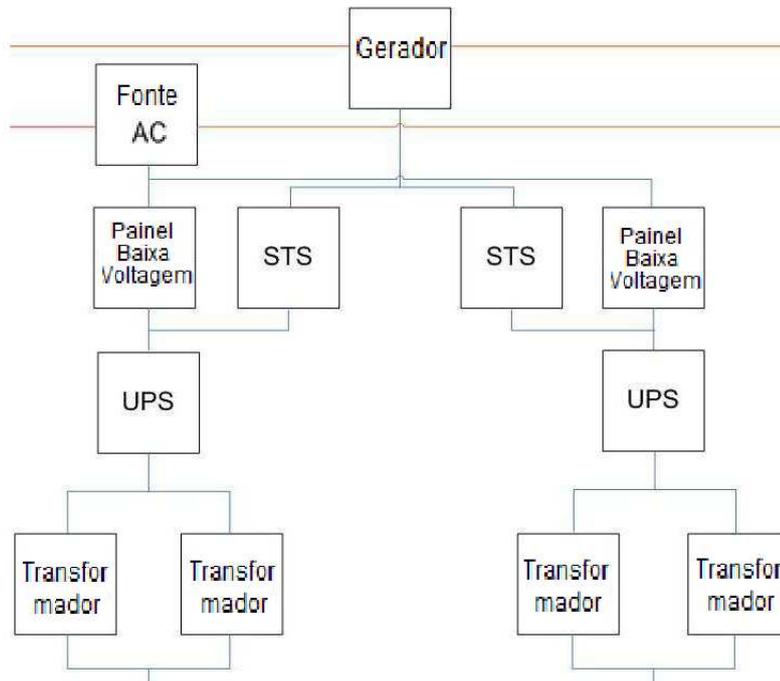


Figura 7.7: Arquitetura 4.

bilidade do sistema. Adicionalmente, A3 corresponde a arquitetura com maior disponibilidade. O leitor deve lembrar que A3 é a arquitetura em que o gerador pode ser adotado para suportar todo o sistema de energia, bem como todo o sistema de resfriamento. Portanto, isso mostra que o gerador em *cold standby*, melhora a disponibilidade.

Para uma melhor visualização, a Figura 7.9 mostra uma comparação entre a disponibilidade, o custo total de propriedade, bem como a exergia vida, tendo em conta os resultados das arquiteturas de potência sem considerar o sistema de resfriamento. A arquitetura de referência corresponde à arquitetura A1. É possível notar que uma opção interessante corresponde ao sistema A3, que é cerca de 2% mais caro que a arquitetura de referência, tem a maior disponibilidade (quase 45% superior ao A1), assim como tem um menor consumo de exergia que o sistema A2.

Além disso, a Figura 7.10 apresenta a mesma comparação levando em conta os sistemas de refrigeração e de potência. Devido ao fato de que consideramos os sistemas de refrigeração e de potência em série os resultados de disponibilidade do sistema como um todo tiveram uma redução considerável. Além disso, nessa comparação, o TCO e o consumo de exergia do sistema de refrigeração não alteraram significativamente o resultado obtido para o sistema de energia.

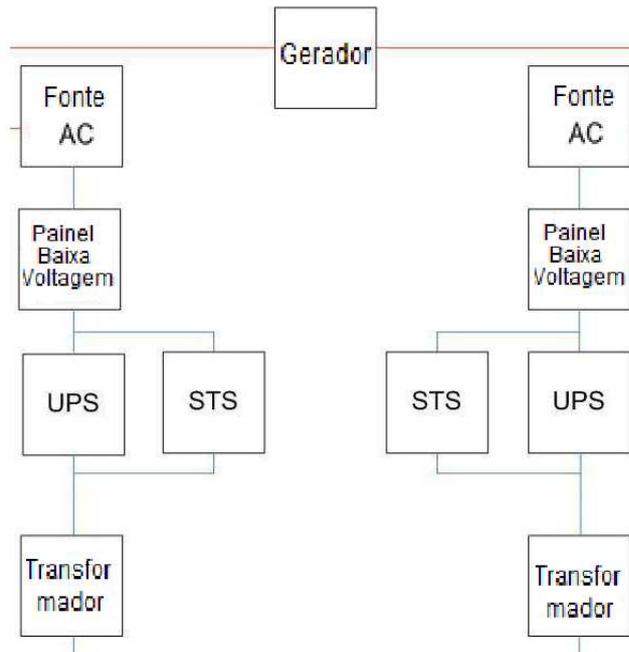


Figura 7.8: Arquitetura 5.

7.3 Considerações Finais.

Este capítulo apresentou dois estudos de caso para mostrar as funcionalidades da ferramenta proposta. No estudo de caso 1, foi possível verificar a avaliação de cinco arquiteturas de potência *data center*, com nível de redundância crescente. No segundo estudo de caso, foram avaliadas cinco arquiteturas compostas por sistemas de refrigeração e potência para ambientes *data center*. A partir dos dois estudos de caso, foi possível observar que utilizando a ferramenta ASTRO é possível avaliar sistemas *data center* considerando aspectos como dependabilidade, sustentabilidade e custos econômicos.

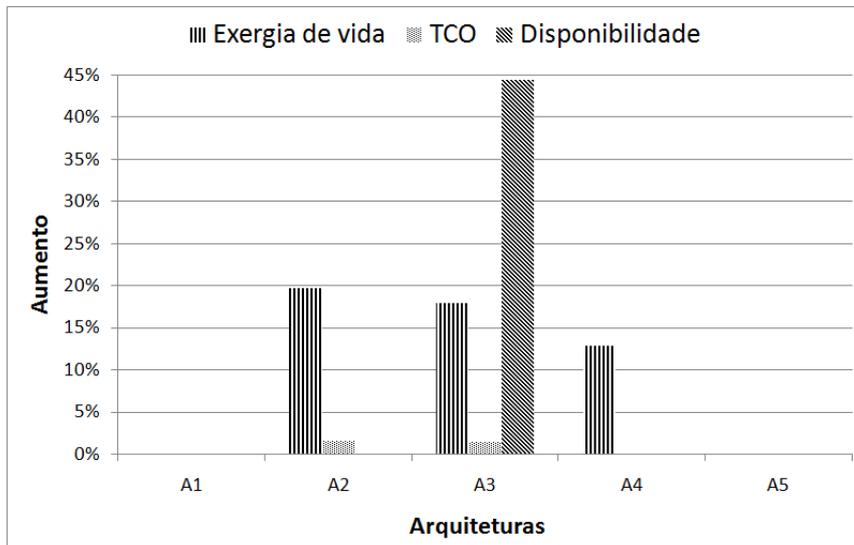


Figura 7.9: Resultados de potência: comparando Exergía, disponibilidade e TCO.

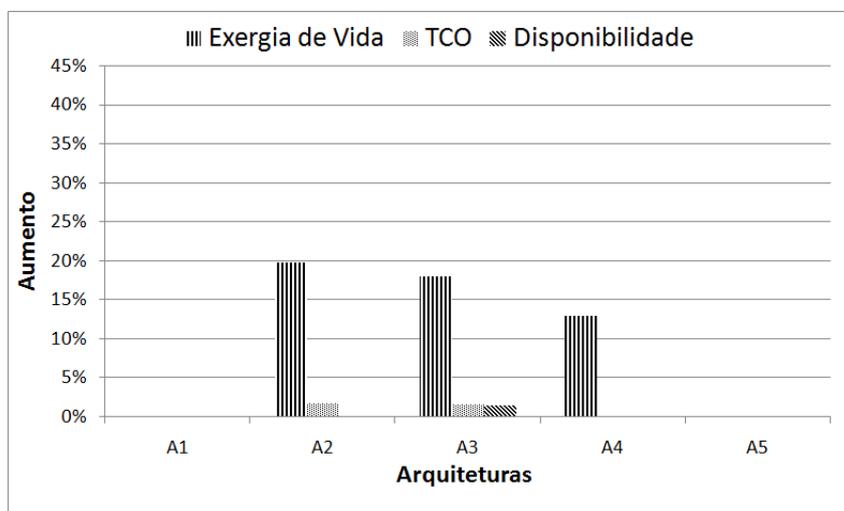


Figura 7.10: Resultados de potência e resfriamento: comparando Exergía, disponibilidade e TCO.

8

Conclusão

Um princípio de projeto amplamente utilizado para tolerância a falhas é a introdução de redundância para melhorar a disponibilidade. No entanto, dado que o aumento da redundância leva a utilização adicional de recursos e energia, espera-se um impacto negativo sobre a sustentabilidade. Neste contexto, projetistas precisam verificar vários *trade-offs* e selecionar uma solução viável, considerando métricas de dependabilidade e sustentabilidade. Assim, o uso de ferramentas é importante para automatizar as atividades de projeto e fornecer resultados tão rápidos quanto possível.

Atualmente, projetistas *data center* não possuem ferramentas de apoio à avaliação integrada de sustentabilidade e dependabilidade em infra-estruturas *data center*. De fato, duas arquiteturas *data center* diferentes com disponibilidades semelhantes podem ter impactos em termos de sustentabilidade muito diferentes.

Nesse contexto, este trabalho apresentou um ambiente integrado que considera as vantagens das redes de Petri estocásticas (SPN) e diagramas de blocos de confiabilidade (RBD) para analisar infra-estruturas *data centers*, tendo em conta aspectos de dependabilidade e sustentabilidade. A ferramenta apresentada possui 2 ambientes que permitem a avaliação de arquiteturas *data center*, são eles: ambiente de avaliação de infra-estruturas de potência em sistemas *data center* e ambiente de avaliação de infra-estruturas de resfriamento em sistemas *data center*. Com o uso desses ambientes, usuários não especializados em técnicas de modelagem SPN e RBD podem avaliar um determinado sistema *data center* conhecendo apenas sua estrutura. É importante ressaltar que a partir desses ambientes de avaliação é possível fazer uma análise do custo total da infra-estrutura (TCO) e uma estimativa do impacto do uso desse sistema em termos de consumo exergético.

Além dos editores de avaliação específicos para ambientes *data center*, foi apresentado que a ferramenta ASTRO possui dois ambientes de avaliação de dependabilidade, são eles: ambiente de avaliação RBD e ambiente de avaliação SPN. Os ambientes de

avaliação de dependabilidade permitem ao usuário avaliar métrica de suma importância para o projeto de sistemas em geral (não apenas *data center*). Métricas importantes, por exemplo: disponibilidade, confiabilidade, *MTTF* e *MTTR* são obtidas a partir do uso desses ambientes de avaliação.

Dois estudos de caso foram apresentados para demonstrar a viabilidade da ferramenta proposta. Nesses estudos de caso, 10 arquiteturas foram avaliadas em termos de sustentabilidade e dependabilidade. Foi mostrado que com o uso da ferramenta o projetista tem a possibilidade de escolher de acordo com as suas restrições, qual arquitetura *data center* melhor se adequa às suas necessidades. Sejam elas custos, disponibilidade, restrições de energia ou relacionadas à sustentabilidade.

8.1 Limitações e trabalhos futuros

No decorrer do desenvolvimento deste trabalho, algumas limitações da ferramenta foram verificadas. Essas limitações, descritas abaixo, são boas oportunidades para trabalhos futuros:

- **Utilização de métodos markovianos [17] para avaliação de dependabilidade.** A ferramenta ASTRO utiliza apenas simulação para avaliação de modelos SPN. Portanto, pretende-se que em versões futuras a ferramenta possa fazer análises transientes e estacionárias utilizando modelos SPN.
- **Utilização de outras métricas para avaliação de sustentabilidade.** Além da avaliação de sustentabilidade considerando consumo exergético, pretende-se em versões futuras adicionar outras métricas de sustentabilidade. Como por exemplo, consumo equivalente de CO_2 .
- **Adição de outros modelo de dependabilidade.** Pretende-se que em versões futuras da ferramenta outros modelos sejam incluídos a ferramenta ASTRO. Podemos destacar, por exemplo, árvores de falha e cadeias de markov.

Assinaturas

Recife, 10 de fevereiro de 2011

Bruno Silva
(Mestrando)

Paulo Romero Martins Maciel
(Orientador)

Referências Bibliográficas

- [1] (1987). *World Commission on Environmental and Development (WCED), Our Common Future*. Oxford University Press.
- [2] (2006). Iso 14064: Environmental management: Life cycle assessment - principles and framework. Technical report.
- [3] (2008). *IEEE Gold Book 473, Design of Reliable Industrial and Commercial Power Systems*.
- [4] (2010). Blocksim tool. <http://www.reliasoft.com/BlockSim>.
- [5] (2010). Relex tool. <http://www.relex.com>.
- [6] (2011). Economic input-output life cycle assessment. <http://www.eiolca.net/>.
- [7] Abd-Allah, A. (1997). Extending reliability block diagrams to software architectures. In *Technical Report: USC-CSE-97-501*, Los Angeles, CA, USA.
- [8] Augarten, S. (1984). *Bit by Bit: An Illustrated History of Computers*. Houghton Mifflin Co., Boston, MA, USA.
- [9] Avizienis, A., Laprie, J., and Randell, B. (2001). Fundamental Concepts of Dependability. *Technical Report Series-University of Newcastle upon Tyne Computing Science*.
- [10] Balan, R. O. and Traldi, L. (2003). Preprocessing minpaths for sum of disjoint products. *IEEE Transaction on Reliability*, pages 289–295.
- [11] Balbo, G. and Chiola, G. (1989). Stochastic petri net simulation. In *WSC '89: Proceedings of the 21st conference on Winter simulation*, pages 266–276, New York, NY, USA. ACM.
- [12] Banks, J. (1999). Introduction to simulation. In *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future - Volume 1*, WSC '99, pages 7–13, New York, NY, USA. ACM.
- [13] Barroso, L. and et al (2009). Web search for a planet: The google cluster architecture.

- [14] Bash, C. and et al (2008). The sustainable information technology ecosystem. In *ITherm '08*.
- [15] Baumann, H. and Tillman, A. (2004). The hitch hiker's guide to lca. In *Studentliteratur AB*.
- [16] Belady, C. (2007). In the data center, power and cooling costs more than the it equipment it supports. *Electronics Cooling Magazine*.
- [17] Bolch, G., Greiner, S., de Meer, H., and Trivedi, K. S. (2006). *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. John Wiley and Sons, New York, NY, USA, 2 edition.
- [18] Cassandras, C. G. and Lafortune, S. (2006). *Introduction to Discrete Event Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [19] Cengel, Y. A. and Boles, M. A. (2003). *Thermodynamics an Engineering Approach*. 5 edition.
- [20] Chan, T. F., Golub, G. H., and Leveque, R. J. (1983). Algorithms for Computing the Sample Variance: Analysis and Recommendations. *The American Statistician*, **37**(3), 242–247.
- [21] Chiola, G., Franceschinis, G., Gaeta, R., and Ribaudò, M. (1995). Greatspn 1.7: Graphical editor and analyzer for timed and stochastic petri nets. *Performance Evaluation*, **24**(1-2), 47 – 68. Performance Modeling Tools.
- [22] De Meester, B., Dewulf, J., Verbeke, S., Janssens, A., and Van Langenhove, H. (2009). Exergetic life-cycle assessment (ELCA) for resource consumption evaluation in the built environment. *Building and Environment*, **44**(1), 11–17.
- [23] Degraeve, Z. and Roodhooft, F. (1999). Effectively selecting suppliers using total cost of ownership. *Journal of Supply Chain Management*, **35**(1), 5–10.
- [24] Dincer, I. and Rosen, M. A. (2007). *EXERGY: Energy, Environment and Sustainable Development*. Elsevier Science.
- [25] Ebeling, C. (1997a). *An Introduction to Reliability and Maintainability Engineering*. McGraw-Hill, Boston, MA.

- [26] Ebeling, C. (1997b). *An Introduction to Reliability and Maintainability Engineering*. Waveland Press.
- [27] Ellram, L. (1995). Total cost of ownership: an analysis approach for purchasing.
- [28] Fan, X. and et al (2007). Power provisioning for a warehouse-sized computer.
- [29] Fishman, G. S. (1971). Estimating sample size in computing simulation experiments. *MANAGEMENT SCIENCE*, **18**(1), 21–38.
- [30] German, R. (2000). *Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets*. John Wiley & Sons, Inc., New York, NY, USA.
- [31] German, R., Kelling, C., Zimmermann, A., and Hommel, G. (1995). Timenet: a toolkit for evaluating non-markovian stochastic petri nets. *Performance Evaluation*, **24**(1-2), 69 – 87. Performance Modeling Tools.
- [32] Graedel, T. (1998). *Streamlined Life-Cycle Assessment*,. Prentice Hall.
- [33] Hendrickson, C. and et al (????). *Environmental Life Cycle Assessment of Goods and Services: An Input-Output Approach*.
- [34] Herold, K. and Radermacher, R. (2002). Integrated power and cooling systems for data centers. In *ITHERM*.
- [35] Jain, R. (1991). *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling / Raj Jain*. Wiley, New York :.
- [36] Kuo, W. and Zuo, M. J. (2003). *Optimal Reliability Modeling - Principles and Applications*. Wiley.
- [37] Law, A. M. (1976). Confidence intervals in discrete event simulation: A state-of-the-art survey. In *Proceedings of the 76 Bicentennial conference on Winter simulation, WSC '76*, pages 53–54. Winter Simulation Conference.
- [38] Lettieri, D. (2009). Lifetime exergy consumption as a sustainability metric for information technologies.
- [39] Lilius, J. (1998). Efficient state space search for time petri nets. *Electronic Notes in Theoretical Computer Science*, **18**, 113 – 133. MFCS'98 Workshop on Concurrency.

- [40] Marsan, M., Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G. (1998). Modelling with Generalized Stochastic Petri Nets. *ACM SIGMETRICS Performance Evaluation Review*, **26**(2).
- [41] Misra, K. B., editor (2008). *Handbook of Performability Engineering*. Springer.
- [42] Moran, M. J. and Shapiro, H. N. (2006). *Fundamentals of Engineering Thermodynamics*. John Wiley Sons, Inc., 5 edition.
- [43] Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proc. IEEE*, **77**(4), 541–580.
- [44] Natkin, G. (1990). Matrix Product Form Solution For Closed Synchronized Queuing Networks. In *Petri Nets and Performance Models*, page 29. IEEE Computer Society.
- [45] Patterson, D. (2002). A simple way to estimate the cost of downtime.
- [46] Patterson, M. (2008). The effect of data center temperature on energy efficiency. In *ITHERM'08*.
- [47] Rausand, M. and Høyland, A. (2004). *System Reliability Theory - Models, Statistical Methods, and Applications*. Wiley Series in Probability and Statistics. Wiley, second edition edition.
- [48] Sahner, A. R. and Trivedi, S. K. (1986). Reliability modeling using sharpe. Technical report, Durham, NC, USA.
- [49] Shah, A. and et al (2009). The environmental impact of data centers. In *ASME InterPAC*.
- [50] Shah, A. and Patel, C. (2009). Designing environmentally sustainable cooling solutions using exergo-thermo-volumes. In *Int. J. Energy Res.*
- [51] Sharma, R. K. and et al (2008). On building next generation data centers: energy flow in the information technology stack. In *Compute '08*.
- [52] Thormark, C. (2002). A low energy building in a life cycle—its embodied energy, energy need for operation and recycling potential. *Building and environment*, **37**(4), 429–435.

- [53] Trivedi, K. (????). *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*.
- [54] Trivedi, K. (2006). *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. John Wiley and Sons, 2nd Edition.
- [55] Trivedi, K. and et al (1994). Reliability analysis techniques explored through a communication network example. In *International Workshop on Computer-Aided Design, Test, and Evaluation for Dependability*.
- [56] V., A. (2003). Comparing availability of various rack power redundancy configurations.
- [57] Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. (2009). A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, **39**(1), 50–55.
- [58] Wiboonrat, M. (2008a). An empirical study on data center system failure diagnosis. In *ICIMP'08*.
- [59] Wiboonrat, M. (2008b). Risk anatomy of data center power distribution systems. In *ICSET'08*.
- [60] Yang, L., Zmeureanu, R., and Rivard, H. (2008). Comparison of environmental impacts of two residential heating systems. *Building and Environment*, **43**(6), 1072–1081.