

# 4

## VERIFICAÇÃO E VALIDAÇÃO DE MODELOS DE SIMULAÇÃO

Esse capítulo trata das questões relativas a verificação e a validação de modelos computacionais de simulação. Inicialmente aborda-se as principais técnicas envolvidas no processo de verificação dos modelos de simulação. Tais procedimentos procuram garantir que o programa computacional por trás do modelo esteja isento de erros. Na segunda parte deste capítulo, trata-se das técnicas de validação de modelos. Neste caso, o interesse é garantir que o programa computacional, agora isento de erros, tenha um comportamento semelhante ao do sistema modelado, permitindo a realização de inferências estatísticas a partir de modelos válidos e confiáveis.

### Tópicos

- 4.1 Introdução**
- 4.2 Verificação e Validação de Modelos de Simulação**
- 4.3 Técnicas de Verificação de Modelos de Simulação**
- 4.4 Técnicas de Validação de Modelos de Simulação**
- Sumário**

### 4.1 Introdução

Durante o desenvolvimento de um modelo de simulação é preciso estar seguro de que o mesmo esteja sendo corretamente implementado. Subentende-se, portanto, que o mesmo encontre-se sem erros de sintaxe e/ou de lógica, bem como seja representativo do sistema real ou projetado. Estes dois passos são conhecidos como verificação e validação de um modelo e costumam ser apontados como os primeiros passos de um processo mais amplo, que inicia pela análise e tratamento dos dados que serão utilizados pelo modelo e culmina com a análise dos resultados da simulação. Na prática, verifica-se que estas ações acabam por se estenderem ao longo de todas as etapas de um projeto de simulação.

### 4.2 Verificação e Validação de Modelos de Simulação

A qualidade e a validade de um modelo de simulação são medidas pela proximidade entre os resultados obtidos pelo modelo e aqueles originados do sistema real. Uma vez que uma série de pressupostos e simplificações sobre o comportamento do sistema real costuma ser realizada no desenvolvimento do modelo, qualquer tomada de decisão com base em seus resultados deve ser precedida de uma avaliação de sua qualidade e apropriação. Esta avaliação está subdividida em

duas etapas. A primeira consiste em avaliar se estes pressupostos e estas simplificações foram corretamente implementadas no modelo computacional. A segunda é saber se, apesar dos pressupostos e das simplificações implementadas, o modelo ainda é válido, isto é, comporta-se à semelhança do sistema real. As duas etapas são chamadas de *verificação* e *validação*, respectivamente. Em outras palavras, validação relaciona-se com a representatividade dos pressupostos enquanto que verificação diz respeito à correção, isto é, ausência de erros, das implementações computacionais. Pode-se dizer também que verificação é a etapa na qual assegura-se que o modelo realiza o que se pretende que seja realizado.

Segundo Jain (1991), um modelo computacional de simulação pode se encontrar em uma das seguintes categorias:

1. Não-validado e não-verificado
2. Não-validado e verificado
3. Validado e não-verificado
4. Validado e verificado

Por exemplo, um modelo não-validado e verificado, seria aquele em que todos os pressupostos estão corretamente implementados, isto é, computacionalmente falando, o programa está correto, mas seus resultados se encontram longe da realidade do sistema real. O objetivo principal das duas etapas, que se aborda a seguir, é dispor de modelos que se encontrem na última das quatro categorias.

### 4.3 Técnicas de Verificação de Modelos de Simulação

Descreve-se a seguir algumas técnicas destinadas à verificação de modelos de simulação computacional. Estas técnicas baseiam-se em dois conjuntos de elementos:

1. Métodos típicos de correção de programas (*debugging*), presentes na literatura que trata do desenvolvimento de programas de computadores;
2. Procedimentos especialmente relacionados ao desenvolvimento de modelos de simulação.

Algumas das técnicas que se descreve serão acompanhadas de exemplos e pequenos modelos de simulação desenvolvidos no Arena.

#### 4.3.1 Uso de Modelos Determinísticos

Um problema comum que se apresenta durante a verificação dos modelos de simulação é o da variabilidade das respostas do programa devido ao comportamento das variáveis aleatórias. Tais variáveis, necessárias para representar os processos estocásticos que ocorrem nos sistemas do mundo real, podem dificultar o trabalho de verificação. Obviamente que a análise e a busca de erros em programas determinísticos é mais fácil do que naqueles que contém variáveis aleatórias. Uma técnica bastante útil consiste em atribuir valores determinísticos a estas variáveis. Com isso, fica mais simples a previsão de respostas do sistema e, como consequência, a verificação e a busca de erros nos vários módulos do programa.

Para exemplificar o emprego desta técnica, desenvolve-se um modelo para o sistema abaixo especificado. A figura 4.1 mostra de maneira esquemática a idéia do sistema.

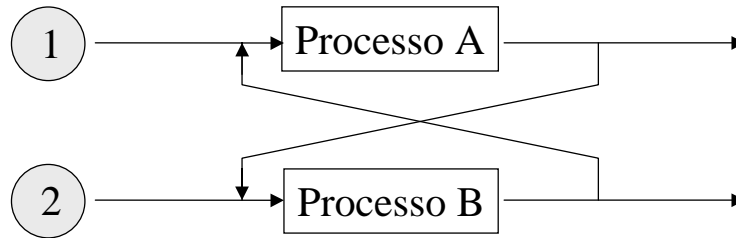


Figura 4.1: Modelo para teste de valores aleatórios versus valores determinísticos.

Suponha que num sistema existam duas entradas (1 e 2), que dão acesso a dois recursos que realizam os processos A e B. Considere a existência de dois tipos de entidades que devem passar por estes processos. Se uma determinada entidade for admitida pela entrada 1 do sistema, ela realiza o processo A e na sequência o processo B. Do contrário, se uma entidade é admitida no sistema via entrada 2, ela primeiro realiza o processo B e depois o A. A ordem dos processos não é importante e eles são independentes. O modelo deste sistema é bastante simples. No entanto, devido ao alto grau de variabilidade associado aos eventos que definem as chegadas e aos tempos das atividades relacionadas aos processos, mesmo um sistema simples como este, pode trazer dúvidas quanto à correção do modelo desenvolvido. Observem-se os parâmetros e os resultados da simulação destes modelos (modelos: “determXrandom.doe” e “randomXdeterm.doe”).

$$\text{Tempos entre Chegadas (entradas 1 e 2, em minutos)} = \text{EXPO (4)}$$

Tempo de Processo (processos A e B, em minutos) = EXPO (2)

De acordo com o modelo criado, espera-se que os resultados apresentem valores semelhantes, quanto as variáveis de desempenho adotadas, isto é, espera-se que o número de entidades que saem do sistema a partir do processo A deva ser próximo ao número das que saem de B. Além disso, os tempos no sistema para aquelas que entram via porta 1, também devem ser semelhantes aos das entidades que entram via porta 2. Cinco experimentos (simulações) foram realizados. Cada experimento é simulado por 480 minutos. Os resultados estão na tabela 4.1.

Experimento	Entidades Processadas		Tempo no Sistema	
	Saindo por A	Saindo por B	Entrando em 1	Entrando em 2
1	111	94	18.12	17.48
2	104	110	40.64	42.27
3	92	107	85.13	76.20
4	100	101	30.65	33.11
5	103	102	58.33	61.23

Tabela 4.1: Resultados das simulações com parâmetros não determinísticos

As diferenças apresentadas nos resultados dos cinco experimentos devem-se a alta variabilidade imposta aos parâmetros empregados no modelo. No capítulo cinco serão vistos mais detalhes associados às diversas distribuições de probabilidades empregadas em modelos de simulação. Especificamente quanto à distribuição exponencial, pode-se adiantar que é uma das

distribuições sobre a qual cuidados redobrados devem ser tomados no seu emprego, devido a grande variabilidade de seus resultados.

Para um analista que observa tais resultados sem considerar tais propriedades das distribuições, é comum que a primeira dúvida que vem à sua cabeça seja a possibilidade de problemas na construção do modelo. Neste caso, executar a simulação substituindo os parâmetros por valores determinísticos pode ajudar. Na tabela 4.2 observa-se um novo resultado da simulação quando os seguintes valores foram utilizados:

Tempos entre Chegadas (entradas 1 e 2, em minutos)	= 4
Tempo de Processo (processos A e B, em minutos)	= 2

Experimento	Entidades Processadas		Tempo no Sistema	
	Saindo por A	Saindo por B	Entrando em 1	Entrando em 2
1	120	120	4	4

Tabela 4.2: Resultados da simulação com parâmetros não determinísticos

É fácil verificar que os resultados acima estão de acordo com as expectativas do analista. Portanto, as diferenças observadas anteriormente devem-se as variações provocadas pelos números gerados das distribuições de probabilidades empregadas.

### 4.3.2 Variações sobre os Dados de Entrada

Variar os dados de entrada do modelo e verificar se as respostas são adequadas e consistentes, é outra técnica de análise bastante comum. Se os dados de entrada sofrem variações, o esperado é que os resultados sofram variações condizentes. Por exemplo, reduzindo-se o tempo entre as chegadas de entidades no sistema, é esperado que ocorra um aumento no número de elementos nas filas devido a um maior congestionamento. Por outro lado, aumentando-se a capacidade de atendimento de determinado recurso, acredita-se que o tempo médio de espera das entidades na fila deste recurso seja reduzido. Fica claro que, as respostas dos modelos às alterações aqui sugeridas e exemplificadas dependem dos sistemas modelados e de outras alterações que por ventura tenham sido realizadas.

Para exemplificar, toma-se o modelo utilizado na sessão anterior. Dobrando-se a taxa de chegadas junto a entrada 2, isto é, reduzindo-se o tempo entre chegadas de EXPO(4) para EXPO(2), a expectativa, mesmo considerando as variações associadas as distribuições exponenciais empregadas, é de que ocorram o dobro de saídas a partir do processo A. Os resultados de cinco experimentos, são mostrados na tabela 4.3.

Os resultados, apesar das variações apresentadas, mostram, claramente, a tendência esperada. Para o desenvolvedor do modelo, o emprego desta técnica de verificação, trás evidências de que o modelo, por reagir às modificações na direção esperada, está bem programado.

Experimento	Entidades Processadas	
	Saindo por A	Saindo por B
1	143	75
2	131	66
3	145	65
4	130	60
5	133	68

Tabela 4.3: Resultados das simulações com redução nos TEC junto à entrada dois.

### 4.3.3 Uso de Rotinas de Rastreamento (Trace)

As rotinas de rastreamento ou acompanhamento são módulos, existentes na maioria das linguagens de simulação, que permitem uma visualização da lista dos eventos ocorridos, ordenados pelo tempo de sua ocorrência. Ao lado dos eventos, a lista também apresenta as modificações ocorridas nos conteúdos das inúmeras variáveis e atributos deles dependentes. Geralmente tais rotinas podem ser ativadas e desativadas pelo usuário, uma vez que sua ativação provoca uma carga adicional ao processamento. Este tipo de ferramenta é extremamente útil para a verificação de erros, especialmente quanto à lógica da modelagem.

Na figura 4.2 vê-se parte da listagem de uma rotina de rastreamento (trace) emitida pela linguagem de simulação SIMAN, que faz parte do ambiente ARENA. Pode-se observar que os módulos de comandos encontram-se ordenados ao longo do tempo. A cada ponto discreto no tempo, um conjunto de módulos é ativado. Por exemplo, no tempo 0,0000 inicia-se a simulação. Uma entidade é criada e imediatamente uma próxima é programada para ocorrer 6,8397 unidades de tempo (minutos neste caso) após a chegada desta entidade. A entidade criada é controlada internamente no Arena sob identidade 2.

Após a criação, a entidade é enviada a uma fila onde verifica a disponibilidade de um recurso chamado MAQUINA. Como o recurso encontra-se disponível, a entidade toma conta deste recurso por um tempo associado a uma distribuição normal com média 5 e desvio-padrão 1. O valor associado a esta passagem de tempo é de 5,20685 min. Neste instante, o relógio é avançado de 0,0000 para 5,20685, pois não existem outros eventos programados para acontecer neste período. O recurso MAQUINA é liberado e a entidade passa por um bloco contador. O contador, denominado *Conta Pecas*, é então incrementado em 1. A entidade deixa o sistema e o tempo (relógio da simulação) é avançado para 6.8397 (tempo do próximo evento programado) quando uma nova entidade é criada.

Geralmente, o nível de detalhamento de uma rotina de acompanhamento pode ser predefinido, isto é, se desejável, pode-se acompanhar apenas as modificações de parte das variáveis e entidades que se encontrem no sistema.

SIMAN System Trace Beginning at Time: 0.0			
Seq#	Label	Block	System Status Change
TIME: 0.0 ENTITY: 2			
1	CREATE		Next creation scheduled at time 6.8397
2	QUEUE		Entity 2 sent to next block
3	SEIZE		Seized 1 unit(s) of resource MAQUINA Delaying for time NORM( 5,1 )
4	DELAY		Delayed by 5.20685 until time 5.20685
TIME: 5.20685 ENTITY: 2			
			Releasing resources
5	RELEASE		MAQUINA available increased by 1 to 1 Updating counter Conta Pecas
6	COUNT		Counter CONTA PECAS incremented by 1 to 1 Disposing entity
7	DISPOSE		Disposed entity 2
TIME: 6.8397 ENTITY: 3			
8	CREATE		Next creation scheduled at time 9.23668
9	QUEUE		Entity 3 sent to next block
10	SEIZE		Seized 1 unit(s) of resource MAQUINA Delaying for time NORM( 5,1 )
11	DELAY		Delayed by 4.09239 until time 10.9321
TIME: 9.23668 ENTITY: 2			
12	CREATE		
...			
TIME: 10.9321 ENTITY: 3			
			Releasing resources
15	RELEASE		MAQUINA available increased by 1 to 1
...			
16	COUNT		Counter CONTA PECAS incremented by 1 to 2

Figura 4.2: Listagem de uma rotina de rastreamento.

A definição deste nível vai depender do interesse do usuário. O usuário pode programar as rotinas de acompanhamento de tal forma que somente quando da ocorrência de determinados eventos ela seja ativada, mostrando ao analista detalhes do processamento. Por exemplo, num modelo de uma célula de manufatura, o analista poderia estar particularmente interessado sobre os acontecimentos (eventos) que ocorrem sobre o estado de uma área de armazenagem intermediária entre dois postos de trabalho. Quando uma variável de estado controladora desta

área (número de peças, por exemplo) alcançasse um determinado valor, o *trace* seria ativado revelando valores de variáveis, atributos das peças que ali se encontram etc..

#### 4.3.4 Aplicação de Testes de Continuidade

Estes testes são aplicados sobre várias rodadas de simulação do mesmo modelo. A cada rodada aplica-se uma pequena alteração sobre os parâmetros de entrada. Para cada um dos parâmetros, uma pequena alteração, geralmente provoca uma pequena mudança nos resultados. Qualquer mudança mais radical nos resultados deve ser investigada, pois, quase sempre, são provocadas por erros no programa. Um cuidado que deve ser tomado envolve o emprego de distribuições de probabilidade que possam provocar grandes variações sobre os resultados, como a exponencial, por exemplo. O uso destas distribuições, associado a testes de continuidade, pode gerar dúvidas sobre os resultados obtidos. As diferenças foram causadas pelas alterações sobre os parâmetros ou pela variação atribuída às distribuições? Neste caso, o emprego de parâmetros determinísticos ou um controle sobre a função geradora de números aleatórios do simulador, pode ser a solução.

Na tabela 4.4 temos um comparativo entre os resultados das simulações empregando-se as duas técnicas. O modelo usado foi o apresentado na sessão 4.3.1.

Tempo entre Chegadas (TEC)		TEC Exponencial Número de Saídas		TEC Determinístico Número de Saídas	
Entrada 1	Entrada 2	Saindo por A	Saindo por B	Saindo por A	Saindo por B
4	4	111	94	120	120
4	4,4	102	96	108	120
4	4,8	86	122	100	120
4	5,2	86	122	92	120
4	5,6	97	118	85	120
4	6,0	71	103	80	120

Tabela 4.4: Resultados das simulações para o teste de continuidade

Nas colunas que indicam o número de saídas que ocorrem no sistema a partir de A e B, observa-se uma tendência clara ao declínio das ocorrências em A acompanhando a redução nas chegadas originadas na entrada 2. A redução é devido a um aumento dos tempos entre estas ocorrências (TEC). É oportuno salientar, no entanto, que os números das colunas 3 e 4, não evidenciam tão claramente esta tendência. O motivo é o emprego da distribuição exponencial, a qual mascara em parte esta tendência, bem evidenciada quando se empregam valores determinísticos, como se vê nas colunas 5 e 6.

#### 4.3.5 Emprego de Testes de Degenerescência

Estes testes consistem em verificar o funcionamento do modelo quando sobre os parâmetros aplicam-se valores extremos (menores ou maiores) do que os permitidos.

Exemplifica-se este teste com um modelo de simulação de um roteador de tráfego de pacotes em uma rede de computadores. Um roteador é um equipamento que processa pacotes



originados de diversas fontes e os envia a um ou mais destinos. No modelo exemplo, este roteador recebe pacotes originados em três diferentes portas de entrada, os processa e, posteriormente envia-os aos seus destinos. Uma das características destes equipamentos é a presença de *buffers* os quais permitem que enquanto um pacote é processado, um outro que chegue ao roteador aguarde e não seja descartado. Um problema que existe é o da definição do tamanho ideal do *buffer*. Se grande, a possibilidade de descarte de pacotes é pequena. No entanto, um *buffer* muito grande poderá significar que, dependendo da situação (muito tráfego, por exemplo) um pacote poderá permanecer muito tempo no roteador antes de ser processado. Para alguns tipos de aplicações, que utilizam as redes e que sejam sensíveis a este retardo (tráfego de voz, por exemplo), esta é uma situação não desejável. Um modelo de simulação é uma excelente ferramenta para testar possíveis soluções a este problema. Ele deve permitir que mesmo em situações extremas, seja possível observar qual seria o comportamento do sistema real ao qual se refere. A figura 4.3 é a representação esquemática de um roteador (Modelo “Roteador Simples.doe”).

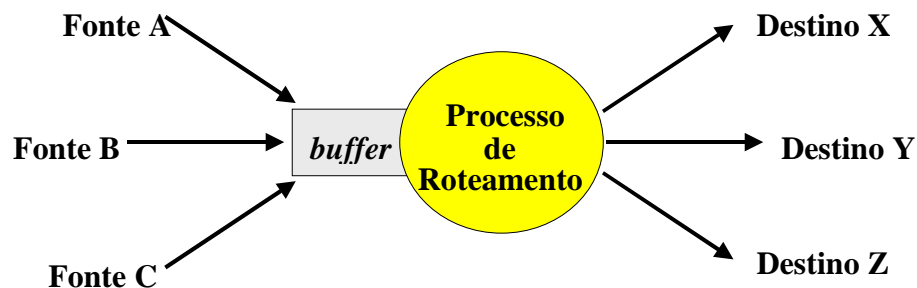


Figura 4.3: Representação de um roteador de tráfego de pacotes em uma rede.

No modelo criado é possível testar algumas situações não muito comuns de funcionamento deste roteador. Por exemplo, testar vários tamanhos de *buffer*, desde tamanho zero, o que pode provocar um grande descarte de pacotes, até o tamanho infinito, em que não haja possibilidade de descarte. Outro teste que pode ser executado seria a chegada de uma grande quantidade de pacotes de alta prioridade e todos com o mesmo destino ou destinos. Enfim, embora o modelo seja simples, diversas situações de maior complexidade, que não costumam ser comuns ao sistema real, podem ser testadas. Em todas elas, o modelo funciona.

Apesar destes casos extremos não serem muitas vezes representativos, eles ajudam a consolidar a modelagem e à descoberta de vários erros que em condições normais de funcionamento do modelo não apareceriam. É importante que a verificação ocorra sobre várias combinações destes valores extremos.

#### 4.3.6 Execução de Casos Simplificados

Uma forma especialmente sugerida para a verificação de modelos é a execução de casos simplificados. No caso de sistemas de manufatura, por exemplo, significa a modelagem de sistemas com poucas máquinas, reduzido número de peças, uma determinada política de gerenciamento de filas, ausência inicial de um sistema de transportes, etc. Para uma rede de computadores, trabalhar com a geração de poucos pacotes, um número reduzido de fontes



geradoras, de nós intermediários e de nós destino, facilita as análises e comparações. Considere, no entanto, que a utilização de modelos reduzidos para casos simplificados não garante que modelos mais complexos funcionem de modo correto. Para uma maior segurança, embora fazendo uso de modelos simplificados, garanta que os testes efetuados possuam suficiente complexidade e aderência ao sistema real.

#### 4.3.7 Verificação de Consistência

Os testes de consistência servem para verificar se o modelo produz resultados similares quando aos parâmetros de entrada são atribuídos valores equivalentes em diferentes execuções. Por exemplo: num modelo de um sistema de produção o setor das máquinas é alimentado por três esteiras. Cada esteira deposita no *buffer* das máquinas, em média, 30 peças por hora para serem processadas. Este modelo deve produzir resultados semelhantes a outro em que às três esteiras são substituídas por uma única cuja média seja de 90 peças/hora. A figura 4.4 abaixo, mostra uma ilustração de cada um dos modelos criados no ARENA.

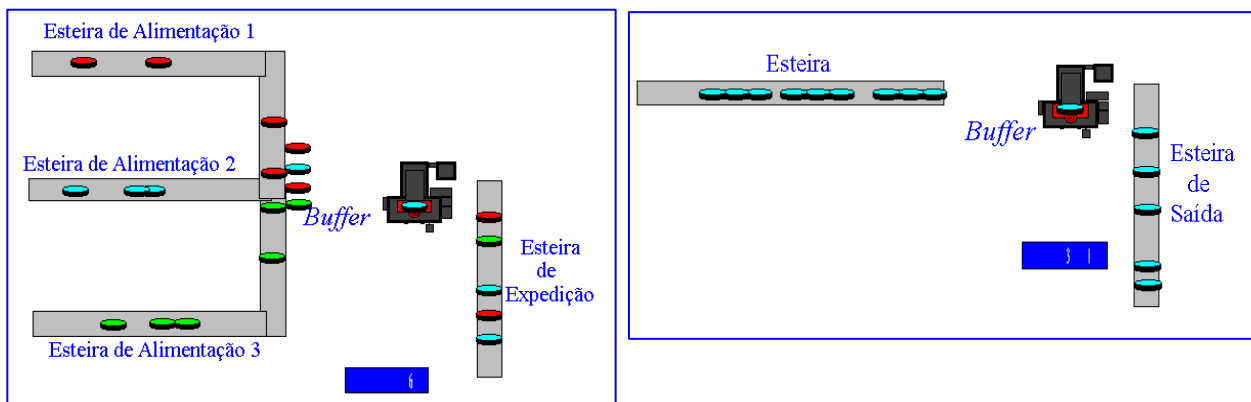


Figura 4.4: Modelos dos sistemas de esteiras modelados no ARENA.

Os resultados apresentados pelos dois modelos mostram consistência. Um cuidado deve ser novamente observado. Quando empregadas distribuições de probabilidade como aqui se fez, as similaridades entre os resultados serão verificadas para experimentos de razoável duração. Por exemplo, simulando-se estes sistemas por apenas 480 min., serão verificadas diferenças nos resultados que podem levar a acreditar que os modelos não geram resultados similares. Quando, no entanto, simulados por 4800 min., as respostas serão muito próximas (ver modelos “Esteiras 3x1.doe” e “Esteiras 3x1a.doe”).

#### 4.3.8 Independência de Sementes Geradoras de Números Aleatórios

As sementes geradoras de números aleatórios utilizados nas funções geradoras de variáveis aleatórias, não devem influenciar o comportamento geral dos resultados. Diferenças

entre as diversas rodadas de simulação são esperadas, mas tais diferenças, considerando-se as médias de diversas amostras (rodadas) devem ser mínimas, caracterizando que estas são devidas aos processos aleatórios envolvidos no sistema real e bem representados no modelo.

#### **4.3.9 Rotinas de Verificação**

As rotinas de verificação são elementos extras adicionados ao programa ou modelo, que permitem ao modelador averiguar a ocorrência de determinados tipos de erros, se eles existirem. Paralelamente, é possível a emissão de alarmes, sinais ou relatórios, indicando a presença de problemas. Por exemplo, se as probabilidades de um conjunto de eventos devem somar 1, o programa poderá verificar isto emitindo avisos ao usuário no caso da ocorrência de valores diferentes do esperado. Um outro exemplo consiste na contabilidade de entidades geradas e servidas. No caso da rede de computadores, por exemplo, o modelo contaria o número de pacotes enviados pelas fontes bem como o número de pacotes recebidos nos diversos destinos. A diferença entre o número de pacotes enviados e o número de pacotes recebidos, acrescidos daqueles perdidos nas diversas rotas, deve ser igual zero. Alguma coisa diferente de zero, indicaria erros de programação ou modelagem. De fato, manter uma contabilidade sobre o número de entidades geradas e retiradas do modelo durante a simulação constitui uma rotina de segurança essencial. O ARENA possui este tipo de rotina. Possui, também, um controle da situação e localização de todas as entidades dentro do modelo durante e ao término de cada rodada de simulação. Desta forma é possível, a qualquer momento ou ao final da simulação, ter conhecimento dos valores de qualquer das variáveis de estado, das entidades que ainda se encontrarem no modelo bem como de todos os demais elementos do modelo.

#### **4.4 Técnicas de Validação de Modelos de Simulação**

Segundo Raj Jain (1991), a validação consiste em assegurar que o modelo e os inúmeros pressupostos e simplificações adotados no seu desenvolvimento sejam razoáveis e, se corretamente implementados, tenham um comportamento e produzam resultados semelhantes aqueles observados nos sistemas reais. O pressuposto básico desta e de outras considerações a propósito do processo de validação de modelos, considera que, de alguma maneira, o analista ou modelador estará comparando um modelo com o mundo real. Este aspecto traz a tona alguns problemas de ordem prática e teórica.

As questões práticas podem ser resumidas pela quase total impossibilidade de se obter, mesmo que seja esta a intenção, um modelo que tenha um comportamento idêntico ao sistema real, considerando que este, de fato, exista.

O segundo ponto importante desta discussão, é que poucas são as oportunidades em que se constrói um modelo para que este atue de forma idêntica a do sistema real. Na maioria das vezes, a construção de um modelo visa justamente a análise e a observação da atuação do sistema (exista este ou não), em condições diferenciadas daquelas observáveis na prática. Exceto por situações em que a experimentação visa a busca de pequenos detalhes (ajuste fino) sobre o funcionamento do sistema real.

A maioria das análises procura verificar o desempenho de inúmeras e diversificadas alternativas para um novo projeto, seja ele baseado num sistema real existente ou não; ou ainda, a simulação de um modelo de sistema real, sobre o qual se aplicam formas de operação e gerência totalmente diferenciadas. Em ambas as situações, mesmo com um sistema real presente, são poucas as possibilidades para a realização de comparações detalhadas para fins de validação.

Quanto aos aspectos teóricos, convém salientar que um modelo inclui uma série de pressupostos e hipóteses sobre o comportamento de um sistema, traduzidos na forma de regras, as quais procuram conduzir o comportamento dos elementos e objetos presentes no sistema. Sendo assim, se o emprego de qualquer técnica, incluindo-se a modelagem e simulação, objetiva o aprendizado e o progresso, então é melhor que tais pressupostos e hipóteses sejam testados de alguma maneira [PIDD, 1995]. Estes aspectos, se não observados, podem levar o analista a cometer erros clássicos tais como, aceitar os resultados de uma experimentação bem executada, a qual se baseia em hipóteses falsas. No emprego da modelagem e simulação, tal pode ocorrer quando se conclui a partir de um modelo não válido.

Basicamente, o processo de validação deve abordar três aspectos principais:

1. Os que envolvem as simplificações e os pressupostos adotados na modelagem do sistema;
2. Os que consideram os parâmetros utilizados como entrada de dados e as distribuições utilizadas para representarem os aspectos de aleatoriedade;
3. Os que envolvem as considerações adotadas quando das análises e conclusões formuladas diante dos resultados obtidos pelas simulações.

Cada um destes aspectos deve ser avaliado e submetido a testes de validação, os quais podem ser realizados de diversas maneiras. É bastante comum a realização da validação a partir de uma macro visão. Em outras palavras, o modelo e o sistema são vistos como uma espécie de “caixa-preta”. Para fins de comparação, não se observa o comportamento interno de ambos, mas, e tão somente, os resultados produzidos.

#### 4.4.1 O Conhecimento e a Intuição de Especialistas

Segundo vários autores (BALCI, 1994; PIDD, 1995 e SARGENT, 1994), a análise e a avaliação de modelos por parte de especialistas é a melhor e a mais simples forma de validá-los. O processo de validação, na prática, deve ser iniciado na fase de concepção estendendo-se até a fase de experimentação. Desta maneira, todos os três aspectos, anteriormente mencionados, estarão contemplados no processo. Não é preciso ter um modelo pronto para a execução para, só então, dar início ao processo de validação.

As simplificações e os pressupostos adotados no modelo devem ser examinados assim que um projeto preliminar estiver disponível. Uma técnica bastante empregada neste aspecto é chamada de *Brainstorming Meeting*, isto é, reuniões contando com a participação de vários elementos que tenham ligações e conhecimentos do sistema e que possam dar suas opiniões. Os aspectos envolvendo os parâmetros de entrada, e suas distribuições, podem ser discutidos nestas

reuniões e, posteriormente, validados durante o desenvolvimento do modelo. Os resultados devem começar a ser examinados e avaliados tão logo um modelo executável esteja disponível.

Os conhecidos testes de *Turing* podem ser aplicados, se possível. Estes testes consistem da apresentação de dois conjuntos de resultados a especialistas. Um conjunto contém resultados do sistema real enquanto que o outro do modelo. Se o especialista não conseguir distinguir os relatórios emitidos pelo modelo computacional daqueles advindos do sistema real é por que não existem muitas razões para invalidar o trabalho de modelagem.

Testes estatísticos também podem e devem ser aplicados sempre que possível. No caso da existência do sistema real, testes comparativos entre resultados deste e do simulado podem ser levados a efeito pela aplicação de um teste-*t*.

#### 4.4.2 Medições Obtidas em Sistemas Reais

Embora seja o método mais confiável e preferido na validação de um modelo, nunca se deve esquecer o fato de que se estará comparando observações realizadas em sistemas distintos. Toda a noção de validação aqui discutida implica na comparação entre dois conjuntos de observações: um proveniente do modelo e outro proveniente do sistema real. As observações provenientes do modelo devem ser inteiramente conhecidas e validas (pelo menos do ponto de vista do seu criador), isto é, não existem desculpas para que o modelador de um sistema não compreenda seu comportamento a ponto de não poder validar suas observações. Já com relação ao sistema real, o conjunto de observações nele originado, pode ter interpretações diferenciadas, as quais dependem, fundamentalmente, do observador.

Desta forma, a aplicação da técnica de comparação com medições obtidas em um sistema real deve sempre ressaltar o fato de que tais observações são provenientes de uma forma distinta e não necessariamente única, de avaliar as respostas deste sistema.

Uma vez que a simulação é uma técnica que pode ser aplicada à construção de modelos de sistemas ainda não existentes, nem sempre tais medições podem ser tomadas. Outras vezes, mesmo com a existência do sistema real, fatores econômicos e outros impedem este exercício. Técnicas estatísticas aplicadas a verificações sobre médias populacionais ou sobre diferenças de médias de duas populações costumam ser aplicadas para testar se existem diferenças entre os resultados obtidos no sistema modelado e no modelo real. As distribuições de entrada costumam ser testadas utilizando-se técnicas de aderência ou testes estatístico como o Qui-quadrado ou o K-S. Outra técnica que pode ser adotada é a verificação do comportamento do modelo diante de situações tipicamente conhecidas para o sistema real. Quando não se têm especialistas e um sistema real para observar, uma prática é a modelagem do sistema para fins de avaliação analítica do mesmo.

#### 4.4.3 Resultados Teóricos

Em alguns segmentos do conhecimento, o emprego de técnicas de modelagem analítica é muito comum. Tome-se, por exemplo, algumas áreas da ciência da computação. São bem conhecidas e empregadas técnicas como a Teoria das Filas ou as Redes de Petri, para a

modelagem e a análise de desempenho de sistemas computacionais. Neste segmento, é muito comum também o emprego de mais de uma técnica para a validação dos resultados obtidos. Por exemplo, modelagem analítica de um sistema e validação dos resultados por meio da simulação deste mesmo sistema. É comum, neste caso, que o modelo analítico empregue uma versão mais simplificada do sistema. É importante ressaltar, no entanto, que o emprego de modelos analíticos para validar modelos de simulação devem ser utilizados com certo cuidado, uma vez que ambos podem não ser representativos do sistema real.

É interessante, neste momento, citar alguns autores que são referências no que diz respeito à validação de modelos. Raj Jain (1991), por exemplo, cita que “a validação completa de um modelo é um mito”. Na realidade, diz ele “pode-se demonstrar que um modelo não é inválido para algumas das situações comparadas. Provar que um modelo produz os mesmos resultados que o sistema original, em todas as circunstâncias, necessita uma quantidade muito grande de recursos”. Na mesma linha de pensamento pode-se citar Michael Pidd (1994). Ele afirma que “na melhor das hipóteses, um modelador deve ficar satisfeito quando as observações realizadas no modelo mostram características idênticas as observações do sistema real, isto é, que os dois conjuntos de observações não são idênticos, mas, suficientemente similares para o propósito em mente”. Cita ainda este último: “...é sempre possível que outras observações realizadas nos mesmos sistemas (modelo e real) apresentem uma discordância acentuada. Desta forma, um modelo válido é aquele irrefutável segundo pressupostos bem especificados”.

Pode-se acrescentar que tal prova não é, na maioria das situações necessária. Portanto, a questão da validação de modelos é importante, mas deve-se ter sempre em mente suas limitações e seus limites. O importante é que se faça um delineamento das principais situações sob as quais se deseja avaliar o comportamento do modelo em relação ao sistema real. O objetivo principal é aumentar a confiança nos resultados apresentados pelo modelo para que se possa inferir com a máxima confiança sobre o comportamento do sistema real.

## 4.5 Sumário

Ao longo deste capítulo abordou-se os principais aspectos relacionados com a verificação e a validação de modelos computacionais voltados a simulação de sistemas. Conforme foi salientado, estes são passos fundamentais para que se possa dar prosseguimento, de forma consistente, a todos os aspectos e etapas envolvidas em projetos de simulação de sistemas. No capítulo que segue discutem-se os principais elementos relacionados ao tratamento de dados tendo em vista a representatividade de modelos voltados à simulação de sistemas.