

-URI-

Universidade Regional Integrada do Alto Uruguai e das Missões

Departamento de Engenharias e Ciência da Computação

Ciência da Computação / Sistemas de Informação

# *Modelagem e Simulação de Sistemas*

Braulio Adriano de Mello

**Santo Ângelo, Outubro de 2001**

Última atualização: 10/10/2007

# Sumário

1. A Simulação de Sistemas.....	4
1.1 Um pequeno histórico .....	5
2. Modelos de simulação .....	7
2.1 Introdução .....	7
2.2 Sistema .....	8
2.3 Modelo .....	8
2.4 Cenários .....	9
2.5 Parâmetros do Modelo .....	9
3. Propriedades dos modelos de Simulação .....	11
4. Classificação dos modelos de simulação.....	13
4.1 Modelos Determinísticos.....	13
4.2 Modelos Estocásticos.....	13
4.3 Modelos Estáticos .....	13
4.4 Modelos Dinâmicos .....	14
5. Classificação dos Modelos quanto às mudanças que sofrem suas variáveis de estado .....	15
5.1 Modelagem para simulação discreta .....	17
5.1.1 Simulação orientada a evento.....	18
5.1.2 Simulação Orientada ao Exame da Atividade .....	18
5.1.3 Simulação Orientada a processo.....	19
5.2 Modelagem para simulação contínua.....	19
5.3 Exemplo: construção e execução de um modelo .....	19
5.3.1 Enunciado do sistema .....	19
5.3.2 Modelagem e simulação .....	20
5.4 Exercícios.....	24
6. Modelos baseados em redes de filas.....	26
6.1 Centros de serviço .....	26
7. Algoritmos de escalonamento.....	30
8. Estágios para a construção de modelos de representação .....	31
8.1 Trabalho: Implementação de uma ferramenta de simulação .....	32
9. Paradigmas de Simulação orientada a objeto.....	35
9.1 Introdução .....	35
9.2 Simulação orientada a objetos .....	36
9.3 Proposta de classificação de paradigmas de simulação apresentada em [COP97].....	37
9.3.1 Quanto ao atendimento de solicitações.....	38
9.3.2 Classificação das entidades quanto à forma de descrição dos eventos .....	38
9.3.3 Classificação quanto ao mecanismo utilizado pelas entidades para troca de informações.....	38
9.3.4 Classificação quanto à autonomia da entidade .....	39
9.3.5 Classificação das entidades quanto ao tempo de permanência no sistema .....	39
9.3.6 Classificação das entidades quanto à capacidade de tomar iniciativas .....	39
9.3.7 Classificação quanto à ótica pela qual é feita a descrição do comportamento das entidades.....	40
10. Solução do Modelo .....	41
11. Uso do computador com o Arena.....	43
11.1 ARENA.....	43

11.2	O ambiente do ARENA.....	43
11.3	Construção de modelos.....	44
11.4	O exercício do modelo.....	45
11.5	Template.....	46
11.6	Trabalho para ser entregue no final da disciplina.....	48
12.	Probabilidade e estatística em simulação .....	49
12.1	Coleta de dados .....	49
12.2	Estatística descritiva.....	49
12.2.1	Agrupamento de dados .....	49
12.2.2	Estimação de parâmetros.....	51
12.3	Geração de números e amostras aleatórias .....	51
12.3.1	Amostras aleatórias .....	51
12.3.2	Teste de aleatoriedade .....	52
12.4	Distribuições de Probabilidade .....	52
12.4.1	Distribuição uniforme .....	52
12.4.2	Distribuição exponencial.....	53
12.4.3	Distribuição de Poisson.....	53
12.4.4	O método de Monte Carlo .....	53
12.5	Inferência estatística.....	54
13.	Estudo estatístico .....	55
13.1	Estimação do intervalo de confiança .....	55
13.2	Hipóteses de teste .....	56
13.3	Problemas estatísticos relacionados com a simulação .....	57
14.	Referências Bibliográficas .....	58
	Anexo I.....	59

# 1. A Simulação de Sistemas

A simulação pode ser vista como o estudo do comportamento de sistemas reais através do exercício de modelos. Um modelo incorpora características que permitem representar o comportamento do sistema real [LAW1982]. Sistema pode ser interpretado como uma coleção de itens entre os quais se possa encontrar ou definir alguma relação de funcionalidade.

O termo simulação é bastante genérico, visto que são variadas as formas e mecanismos utilizados para a representação do comportamento de sistemas. O termo simulação também possui grande abrangência em relação ao seu campo de aplicação. Por exemplo, a simulação pode ser aplicada na indústria, organizações públicas, na representação de software/hardware, entre outros.

Os benefícios da simulação de sistemas, aliados à agilidade oferecida pelos meios computacionais, têm sido largamente utilizados como ferramenta auxiliar na solução de problemas diversos. Justifica-se tal afirmação considerando que, com o uso de um computador, uma grande quantidade de eventos pode ser executada em curto espaço de tempo.

De modo geral, o uso da simulação é recomendado principalmente em dois casos. Primeiro, quando a solução de problemas é muito cara ou mesmo impossível através de experimentos. E em segundo, quando os problemas são muito complexos para tratamento analítico. Com o uso da simulação, principalmente quando se observam características estocásticas, sistemas podem ter seu comportamento representado com maior fidelidade e realismo.

São fatores que tornam desejável o uso de técnicas de simulação [FIL95] aliadas aos benefícios computacionais:

- Tempo: em computador é possível realizar experimentos que, se executados sobre o sistema real, poderiam consumir anos;
- Custo: embora a simulação em computador exija recursos humanos e alguns equipamentos, geralmente o custo se mantém muito abaixo se comparado à execução de experimentos sobre o sistema real;
- Impossibilidade de experimentação direta: há situações em que experimentações diretas no sistema real não podem ser realizadas por questões de segurança, de tempo, de acesso, ou ainda de inexistência (sistema em construção);
- Visualização: os computadores oferecem recursos que facilitam a visualização dos resultados de uma simulação (gráficos, tabelas, entre outros), bem como do estado do sistema durante o exercício de um modelo;
- Repetição: depois de construído, um modelo de representação pode ser executado n vezes a um custo muito baixo;
- Interferência: um modelo é extremamente mais flexível para a realização de mudanças se comparado a um sistema real. Esta é uma característica bastante

desejável no estudo de sistemas com objetivos de geração de informações de apoio a tomada de decisões.

Modelos de simulação podem ser considerados como uma descrição do sistema real. O exercício (execução) de modelos de simulação em computador tem potencial para fornecer resultados mais precisos sem que seja preciso interferir no sistema real. Tais resultados, quando analisados estatisticamente, produzem informações que podem contribuir grandemente na tomada de decisões que visam a solução de problemas.

Um outro ponto relevante a considerar em simulação é a ordem de ocorrência dos eventos. Considera-se como evento (mudança de estado) qualquer acontecimento que interfere no sistema. Como a simulação trata do comportamento de sistemas, ou seja, ordem em que os eventos acontecem, é essencial que uma variável de tempo seja contabilizada. O tempo em que cada evento ocorre é estudado para avaliar se este evento deveria ocorrer naquele instante, antes ou depois.

Restringindo um pouco mais a simulação de sistemas para os meios computacionais, e considerando os modelos como um dos métodos de representação de sistemas, o seguinte conceito pode ser adotado:

*“Simulação é uma técnica numérica para realizar experiências em um computador digital, a qual envolve certos tipos de modelos lógicos que descrevem o comportamento de um sistema sobre extensos intervalos de tempo.”[xxx]*

Basicamente, o uso da simulação pode oferecer vantagens quando necessário [FIL95]:

- Estimar distribuição de variáveis aleatórias;
- Testar hipóteses estatísticas;
- Comparar cenários representando diferentes soluções para um problema em estudo;
- Avaliar o comportamento de uma solução analítica;
- Avaliar um processo de tomadas de decisão em tempo real.

## 1.1 Um pequeno histórico

Nas décadas de 60 e 70 a simulação era excessivamente cara e utilizava ferramentas que geralmente só eram disponíveis em grandes corporações. A mão de obra precisava ser especializada, pois a construção e execução de modelos dependia de conhecimentos muito acima da média observada em usuários comuns. O grupo que trabalhava em simulação geralmente era composto por doutores que desenvolviam sistemas grandes e complexos utilizando as linguagens disponíveis na época, tais como o Fortran.

No final da década de 70 e na década de 80, os computadores foram se tornando mais rápidos e mais baratos. Nesta época, por exemplo, as linhas de montagens de carros passaram a utilizar a simulação para resolver problemas tais como de segurança e otimização da linha. Nesta mesma época, a simulação começou a ser utilizada em negócios e por estudantes e pesquisadores que descobriram seu potencial.

No final da década de 80 o valor da simulação foi reconhecido por muitas organizações. Tanto, que várias delas fizeram da simulação um requisito para que investimentos grandes pudessem ser aprovados. No entanto, organizações pequenas raramente utilizavam essa técnica.

Nos anos 90 a simulação atingiu um grau de maturidade suficiente para que seja adotada por organizações de variadas áreas e diferentes portes. É utilizada em estágios iniciais de projetos, em animações, pesquisa, entre outros. Este avanço foi principalmente possível pelo surgimento de ferramentas voltadas para a simulação e fáceis de usar, e pela disponibilidade de computadores mais rápidos e baratos.

## 2. Modelos de simulação

### 2.1 Introdução

Os problemas enfrentados pela indústria, centros de pesquisas, organizações em geral são cada vez maiores e mais complexos, exigindo, cada vez mais, técnicas e procedimentos para auxílio nos seus tratamentos. Uma destas técnicas é a modelagem para simulação cujo objetivo é representar o comportamento de sistemas através de um modelo.

A resolução de quaisquer problemas requer primeiro que estes sejam bem definidos, ou seja, especificar com precisão suficiente os objetivos. Para tanto, é necessário um bom conhecimento do sistema sobre estudo para que se possa entender o problema sem dubiedade.

A situação não é diferente quando se quer resolver problemas usando simulação. O sistema deve ser bem conhecido para que se possa detectar os problemas relacionados e traçar as metas a serem alcançadas.

A simulação de sistemas pode ser desmembrada em algumas etapas básicas [FIL95], também discutidas no âmbito de pesquisa operacional. São elas:

- Problema: identificação de um problema apresentado pelo sistema em estudo, bem como das partes (ou sub-sistemas) que interferem no problema identificado (envolve coleta de amostras);
- Estudo do sistema: descrever o sistema em termos de componentes, atividades, entidades, eventos, restrições, propósitos que motivaram o uso da simulação para solução de problemas;
- Modelo: construção do modelo de representação do sistema em estudo. O modelo precisa ser validado;
- Solução: realizar experimentos sobre o modelo construído, e utilizar as informações resultantes para propor soluções ao problema detectado inicialmente;
- Operacionalizar solução: Executar alterações no sistema real com base nos resultados de simulação;
- Realimentação: re-elaboração parcial do trabalho de acordo com resultados obtidos na interferência sobre o sistema real.

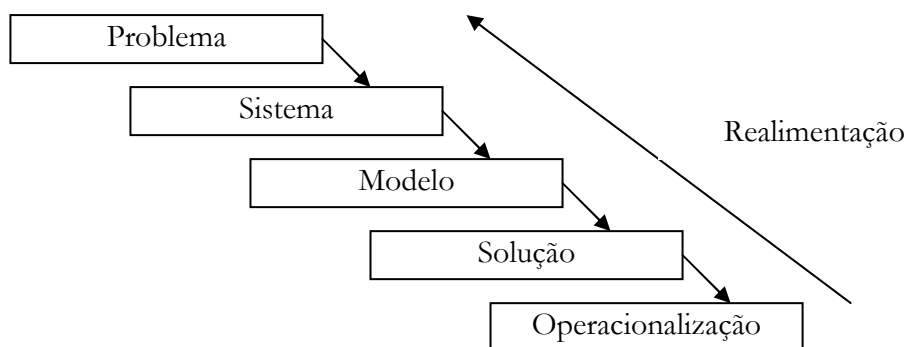


Figura 2.1 – Etapas de Simulação (Fonte: [FIL95])

Os propósitos essenciais do estudo das técnicas de simulação de sistemas se concentram principalmente nas atividades de estudo do sistema real, construção do modelo, e execução de experimentos sobre o modelo. São também os propósitos principais desta disciplina.

## 2.2 Sistema

Para os propósitos desta disciplina, o termo sistema pode ser definido como um conjunto estruturado de itens (componentes ou sub-sistemas) entre os quais se possa encontrar ou definir alguma relação de funcionalidade. O valor dos atributos de um componente, num dado instante de tempo, definem o estado do modelo.

## 2.3 Modelo

De modo geral, os modelos são utilizados para solucionar problemas que seriam muito caros em uma solução experimental ou muito complicados para tratamento analítico.

A construção de um modelo pode visar a solução de um problema específico dentro de um sistema. Deste modo, pode existir um número variado de modelos para um mesmo sistema, cada modelo respeitando as características de um problema particular.

Identificado um problema dentro de um sistema para o qual se pode construir um modelo, é preciso que se tenha uma estrutura organizada para descrever este modelo. Para isso, é utilizada uma linguagem de simulação capaz de fornecer esta estrutura, bem como permitir a tradução do modelo em uma forma aceitável para seu processamento em computador. Neste caso, o computador será utilizado para exercitar o modelo de forma a produzir saídas que possam ser analisadas, a fim de que decisões relacionadas com a resolução do problema possam ser tomadas. Com a ajuda de computadores é possível simular grande quantidade de eventos em curto espaço de tempo.

Através do exercício de um modelo que representa um sistema é possível fazer uma estimativa do comportamento futuro deste sistema. Ao exercício de um modelo dá-se o nome de simulação.

Vale ressaltar que o experimento não pode ser comparado com a simulação. Enquanto experimento é caracterizado por tentativas de mudanças no próprio sistema, simulação é exercida



sobre uma representação do sistema. Também, o experimento pode ter um custo muito elevado, o que justifica, em grande parte dos casos, a construção de modelos para simulação.

Como exemplos práticos da aplicação da simulação pode-se citar o estudo dos efeitos de variações no meio ambiente, mudanças complexas em um determinado ambiente, ou ensino de habilidades (simulador de voo).

Diante dos resultados do exercício de um modelo são realizadas análises de situações novas sobre as quais se tem pouco conhecimento, permite fornecer dados para concepção de novas regras, ou ganho de experiência no projeto do próprio modelo.

Modelos também devem incorporar elementos reais de forma simples, mantendo uma relação custo X benefício coerente. Ou seja, evitar a construção de um modelo muito complexo que dificulte ou mesmo impossibilite a sua compreensão e manipulação. Evitar também a construção de modelos demasiadamente simples que não representem as propriedades desejáveis do sistema em estudo.

De acordo com a ênfase dada aqui para simulação com o uso de computadores, pode-se definir a **simulação** como uma técnica numérica par realizar experiências em um computador digital, as quais envolvem certos tipos de modelos lógicos que descrevem o comportamento de um sistema sobre extensos intervalos de tempo. Por **sistema** entende-se como uma coleção de itens entre os quais se possa encontrar ou definir alguma relação, que são objeto de estudo na construção do modelo. Então, os **modelos** são descrições de um sistema, considerando que eles são concebidos através de uma abstração da realidade, ou seja, apenas aspectos relevantes do sistema real são considerados.

## 2.4 *Cenários*

Dentro do contexto de otimização, existem objetivos a serem alcançados pelo sistema que deve trabalhar satisfazendo restrições impostas pelo meio ambiente. Desta forma, a solução do problema em estudo deve ser a que melhor atinge os objetivos dentre as que satisfazem as restrições. A observação e o acompanhamento do comportamento do modelo, operando com uma determinada solução, são feitas para avaliar se as restrições estão sendo satisfeitas e se os objetivos estão sendo alcançados. Com frequência, uma solução para o problema pode representar uma configuração para o sistema ou uma política de operação do sistema. Assim, as diferentes alternativas de solução do problema devem ser enumeradas para a definição dos cenários a serem avaliados durante os experimentos de simulação [FIL95].

Pode acontecer que os cenários em comparação sejam tão distintos que são desenvolvidos dois ou mais programas para representa-los. Mesmo em tais casos considera-se que há um único modelo que seleciona o programa a ser executado de acordo com o cenário desejado.

## 2.5 *Parâmetros do Modelo*

Os parâmetros definem o cenário de cada corrida (simulação). São controlados externamente pelo usuário no início de cada corrida e portam-se como constantes durante a corrida de simulação. Podem representar uma configuração específica para o sistema ou uma determinada política de operação do sistema. Também podem ser sementes de inicialização dos geradores de variáveis (aleatórias) independentes.

Entre estes parâmetros estão a quantidade e a distribuição da demanda de serviço, os algoritmos de escalonamento, o número médio de vezes que um usuário visita um dispositivo, o

número de usuários em um sistema fechado e a distribuição do tempo entre chegadas de usuários em um sistema aberto. Alguns destes parâmetros caracterizam a carga do sistema, como por exemplo, a distribuição do tempo entre chegadas de usuários.

Pode-se dizer que a habilidade mais importante de um projetista é o entendimento de como o sistema se comporta, a capacidade de estimar o conjunto inicial dos parâmetros do sistema e a habilidade de realizar análises de modificação de parâmetros com precisão.

### 3. Propriedades dos modelos de Simulação

Para abordar o projeto e aplicação de modelos de simulação em computador, é requisito uma definição de modelo de simulação. Deste modo, um modelo científico pode ser definido como uma abstração de um sistema real, que possa ser utilizado com os propósitos de predição e controle. A finalidade de um modelo científico é permitir ao analista determinar em que proporções uma ou mais mudanças em determinados aspectos de um sistema poderão afetar outros aspectos deste sistema ou do sistema como um todo.

A utilidade de um modelo está diretamente relacionada à sua capacidade de incorporar elementos reais de forma simples. No entanto, um modelo deve conter os aspectos importantes do sistema real, sem que sua complexidade impossibilite a compreensão e manipulação do mesmo. Essa característica pode ser difícil de ser alcançada, pois modelos realísticos raramente são simples, e modelos simples raramente são realísticos.

No que se refere aos modelos matemáticos para a simulação em computadores, estes são constituídos por 4 elementos: componentes, variáveis, parâmetros e relações funcionais.

O componente representa a parte do modelo sobre a qual são realizados os estudos de simulação. Os componentes dos modelos tendem a variar largamente, dependendo do sistema que está sendo simulado. Por exemplo, em um sistema de gerenciamento de redes, podem ser considerados componentes os objetos gerenciados, os agentes de gerenciamento e as aplicações gerentes.

As variáveis são utilizadas nos modelos de simulação para relacionar um componente com outro. Tais variáveis podem ser classificadas como variáveis exógenas, variáveis de estado e variáveis endógenas.

As variáveis exógenas são as variáveis independentes ou de entrada do modelo. São consideradas como tendo sido previamente determinadas e fornecidas, independentemente do sistema do qual está sendo construído o modelo. Desta forma estas variáveis podem ser vistas como atuantes sobre o modelo, mas não influenciadas por ele (a direção causa-efeito ocorre somente no sentido variável exógena -> sistema).

As variáveis exógenas podem ser classificadas como controláveis e não controláveis. As variáveis controláveis são as variáveis ou parâmetros que podem ser manipuladas ou controladas pelos elementos encarregados da decisão ou de estabelecer o programa de ação em relação ao sistema.

Variáveis não controláveis são geradas pelas circunstâncias nas quais o sistema modelado existe e não pelo próprio sistema ou pelos elementos encarregados das decisões a ele relativas.

As variáveis de estado descrevem o estado de um sistema ou de um de seus componentes, quer no início de um determinado período de tempo, quer no seu término, ou ainda durante o decorrer de um certo período. Elas interagem com as variáveis exógenas e com as endógenas, de acordo com as relações funcionais previamente estabelecidas.

O valor das variáveis de estado durante um determinado período de tempo pode

dependem do valor de variáveis exógenas de um período de tempo anterior de variáveis de saída de períodos precedentes. Quando um componente utiliza em sua entrada parte de sua saída ocorre um loop no modelo. Estes loops podem ser utilizados para introduzir retardos na simulação.

Em um sistema de gerenciamento de redes, as variáveis de estado podem incluir número de pedidos na fila, resultado de uma operação de gerenciamento ou número de objetos selecionados.

Variáveis endógenas são as variáveis dependentes ou de saída do sistema. São geradas pela interação das variáveis exógenas e de estado, de acordo com as características operacionais do sistema. Em um sistema de gerenciamento de redes, são variáveis endógenas a média de operações realizadas com sucesso ou o conjunto de informações resultantes da ocorrência de eventos nos objetos gerenciados.

Dependendo da finalidade de uma simulação, uma variável pode ser classificada como variável exógena, de estado ou endógena. (Citar exemplo de agente de gerenciamento dentro do sistema)

Quanto às variáveis exógenas, elas podem ser utilizadas de dois modos diferentes em experiências de simulação. Na primeira, elas são tratadas como parâmetros. Neste caso, os parâmetros devem ser determinados pelas condições do problema ou pelos elementos encarregados da decisão. Tais condições ou elementos devem ser previamente lidos pelo computador como dados de entrada, ou geradas internamente se forem variáveis estocásticas (através de métodos pré-definidos).

As relações funcionais que descrevem as interações das variáveis com os componentes de um modelo são duplas: identidades e características operacionais. Ambas são usadas para gerar o comportamento do sistema. As identidades podem tomar a forma quer de definições quer de proposições tautológicas relativas aos componentes do modelo. Para um sistema de gerenciamento, a métrica do tráfego entre uma estação gerente e um recurso remoto pode ser considerado um exemplo. (Em uma firma, lucro = vendas - custo)

Uma característica operacional é uma hipótese, geralmente uma equação matemática que relaciona as variáveis endógenas e de estado de um sistema com suas variáveis exógenas.

Diferentemente dos componentes e variáveis que podem ser diretamente observados no sistema real, os parâmetros das características operacionais somente podem ser obtidos através da inferência estatística.

## 4. Classificação dos modelos de simulação

Existem diferentes classificações propostas para os modelos de simulação. Aqui é utilizado um sistema que classifica os modelos de simulação em determinísticos, estocásticos, estáticos e dinâmicos.

### 4.1 *Modelos Determinísticos*

Nos modelos determinísticos não se permite às variáveis exógenas e endógenas serem variáveis randômicas. Também, as características operacionais devem ser relações exatas e não funções de densidade de probabilidade.

Os modelos determinísticos são menos computacionalmente exigentes que os modelos estocásticos. Estes modelos podem freqüentemente ser resolvidos por técnicas como a do cálculo de máximos e mínimos (técnica analítica).

Em resumo, métodos analíticos são mais eficientes que os métodos de simulação na resolução de problemas determinísticos.

### 4.2 *Modelos Estocásticos*

Modelos são estocásticos quando pelo menos uma das características operacionais é dada por uma função de probabilidade. Geralmente são mais complexos que modelos determinísticos. Deste modo, a simulação é mais adequada como método de análise e solução para os modelos estocásticos.

Estes modelos são adequados à geração randômica de dados para serem utilizados nos estágios de observação ou de testes.

### 4.3 *Modelos Estáticos*

Modelos estáticos não consideram a variável de tempo. Nas pesquisas operacionais, geralmente a maioria dos trabalhos nas áreas da programação linear, não linear e na teoria dos jogos, têm sido relacionadas com modelos estáticos. No entanto, o uso de simulação pode não ser recomendado pelo fato de que a maioria dos modelos estáticos são completamente determinísticos. Assim, as soluções podem normalmente ser obtidas por técnicas analíticas diretas.

## ***4.4 Modelos Dinâmicos***

Os modelos matemáticos que tratam de interações variáveis com o tempo são chamados modelos dinâmicos (ocorrência de uma ordem temporal entre eventos).

## 5. Classificação dos Modelos quanto às mudanças que sofrem suas variáveis de estado

Os modelos de um sistema podem ser classificados como modelos de mudança discreta e modelos de mudança contínua. Estes termos descrevem o modelo e não o sistema. Um sistema pode ser modelado tanto através de um modelo de mudança discreta (ou discreto) quanto um modelo de mudança contínua (ou contínuo).

O tempo é a principal variável independente em um modelo de simulação dinâmico. As demais variáveis (variáveis de estado) podem ser consideradas como funções do tempo, portanto são variáveis dependentes.

As variáveis de estado do sistema são classificadas de acordo com os valores que tomam ao longo da simulação. Os modelos dinâmicos são classificados de acordo com os instantes de tempo em que as variáveis de estado têm seus valores alterados. Um modelo é discreto se todas as variáveis de estado têm seus valores alterados apenas em um número contável de instantes de tempo. Um modelo é contínuo se todas as variáveis de estado têm seus valores alterados a qualquer instante de tempo. Um modelo é misto se algumas variáveis de estado têm os seus valores alterados a qualquer instante de tempo e outras apenas em um número contável de instantes de tempo [FIL95].

**Modelo de mudança discreta:** é aquele em que as variáveis dependentes variam discretamente em pontos específicos do tempo simulado, referidos como tempo de evento. A variável tempo pode ser contínua ou discreta em tais modelos, dependendo se as mudanças discretas nas variáveis dependentes podem ocorrer em qualquer ponto do tempo real ou unicamente em pontos predeterminados.

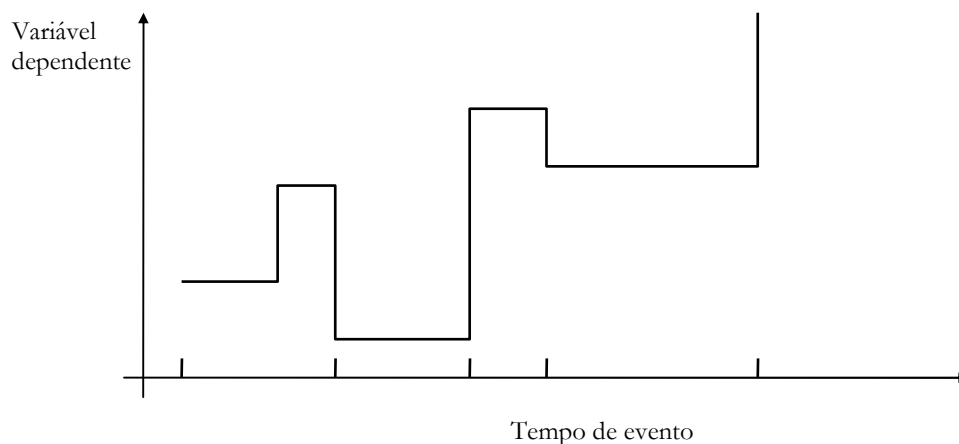
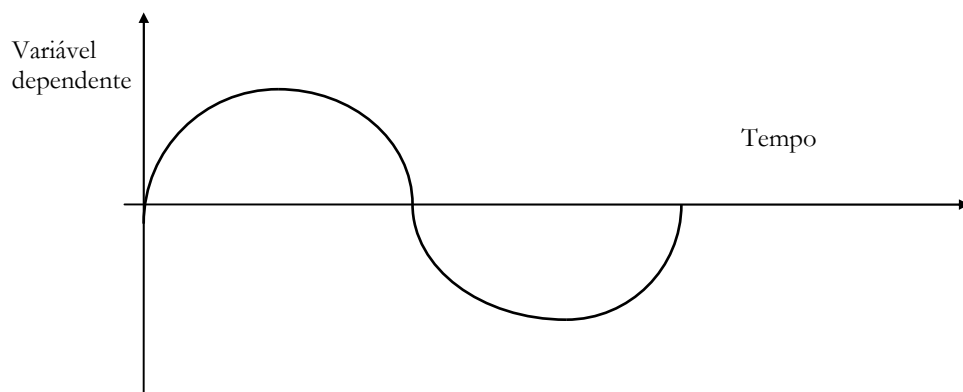
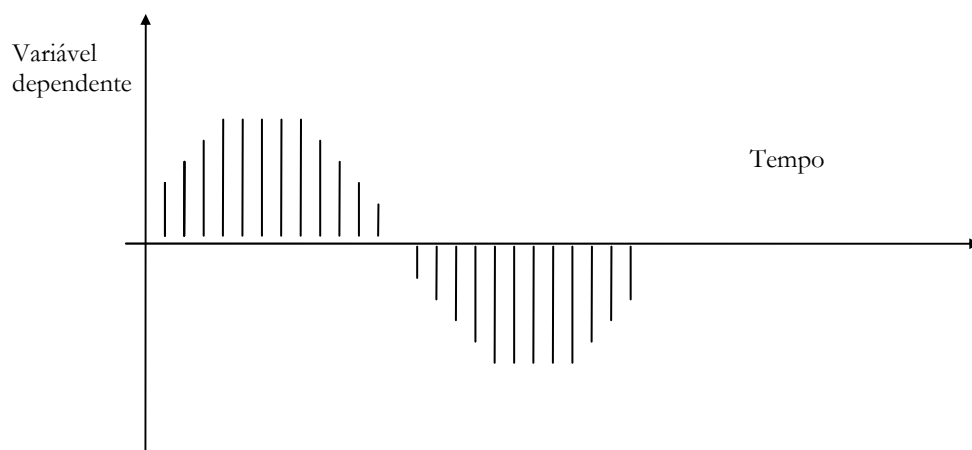


Figura 1 - Variável dependente em um modelo discreto

**Modelo de mudança contínua:** é aquele em que as variáveis dependentes podem variar continuamente ao longo do tempo simulado. Um modelo contínuo pode ser tanto contínuo no tempo ou discreto no tempo, dependendo se os valores das variáveis dependentes estão sempre disponíveis em qualquer ponto do tempo simulado, ou apenas em pontos específicos.



*Figura 2 - Variável dependente em um modelo contínuo com tempo contínuo*



*Figura 3 - Variável dependente em um modelo contínuo com tempo discreto*

Algumas vezes pode ser útil modelar um sistema discreto por um modelo contínuo, considerando as entidades do sistema como um agregado e não como entidades individuais. Podemos encontrar um exemplo de tal conveniência na análise do movimento das partículas de um gás.



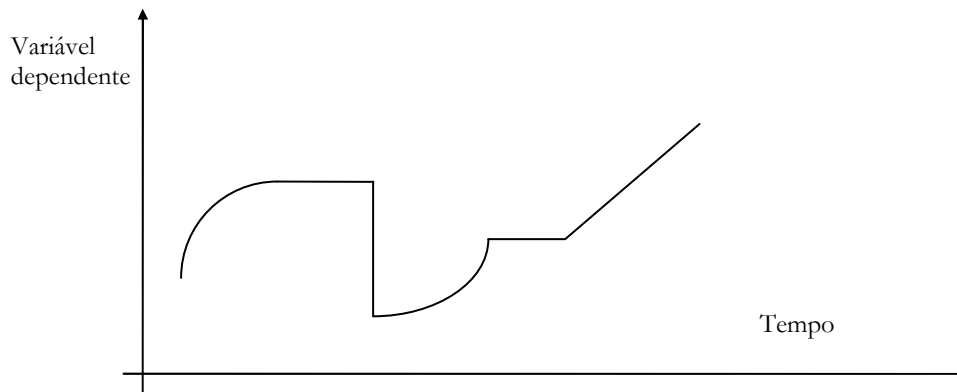


Figura 4 – Modelo de mudança combinada

**Modelo de mudança combinada:** é aquele em que as variáveis dependentes podem variar discretamente, continuamente, ou continuamente com saltos discretos superpostos. A variável tempo pode ser discreta ou contínua.

O aspecto mais importante de uma simulação combinada surge nas mudanças de variações discretas para contínuas e vice-versa. Uma linguagem de simulação combinada deve conter meios de detectar tais ocorrências e modelar suas consequências.

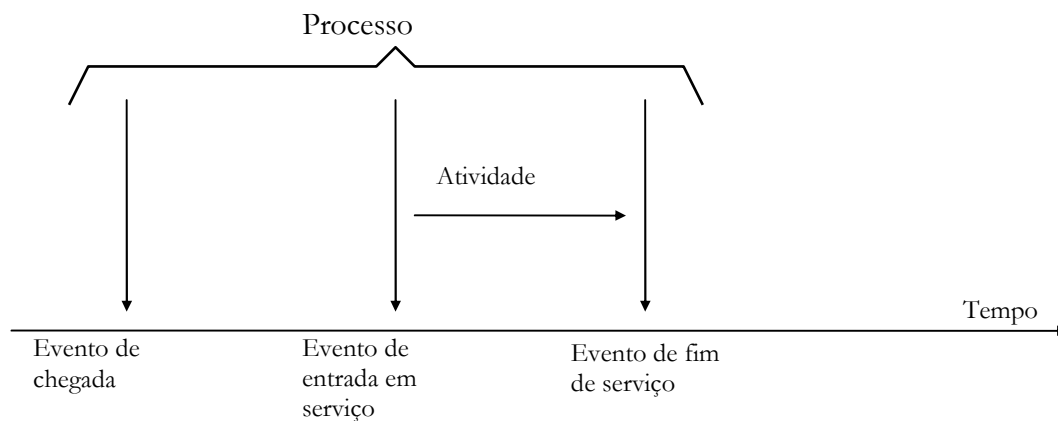
Em resumo, chama-se simulação discreta ao exercício de um modelo discreto. De forma análoga, de simulação contínua ao exercício de um modelo contínuo. E, de simulação combinada ao exercício de um modelo combinado.

## 5.1 Modelagem para simulação discreta

Os objetos em um sistema discreto são chamados *entidades*. Existem vários tipos de entidade e cada uma tem vários tipos de características ou *atributos*. Embora possam estar envolvidas em diferentes tipos de atividades, pode ser conveniente agruparmos as entidades baseados em um atributo comum. Grupos de entidades são chamados *arquivos* ou *conjuntos*.

O objetivo de um modelo para simulação discreta é reproduzir as atividades das entidades engajadas e, a partir daí, conhecer algo sobre o comportamento e desempenho do sistema. Isto é conseguido quando definimos os estados do sistema e construímos atividades que o movem de um estado a outro. O estado de um sistema é definido em termos de valores numéricos dados aos atributos das entidades. Um sistema está em determinado estado, quando todas as suas entidades estão em estados consonantes com o domínio dos valores dos atributos que definem aquele estado.

Em simulação discreta, o estado do sistema só pode mudar nos tempos de eventos. Uma vez que o estado do sistema permanece constante entre tempos de eventos, uma descrição completa do estado do sistema pode ser obtida avançando o *tempo simulado* de um evento a outro. Este mecanismo é usado na maioria das linguagens para simulação discreta.



*Figura 5 - Relações entre evento, processo e atividade*

A formulação de um modelo para simulação discreta pode ser realizada de três formas:

- Pela definição das mudanças nos estados que podem ocorrer em cada tempo de evento;
- Pela descrição das atividades nas quais as entidades do sistema se envolvem e;
- Pela descrição do processo através do qual as entidades do sistema fluem.

Um evento acontece em um ponto isolado do tempo, no qual decisões devem ser tomadas de forma a iniciar ou terminar um atividade. Um processo é uma seqüência ordenada de eventos e pode englobar várias atividades. Estes conceitos levam naturalmente a três alternativas de visão de um modelo para simulação discreta:

- Modelagem orientada a evento
- Modelagem orientada ao exame da atividade
- Modelagem orientada a processo

### **5.1.1 Simulação orientada a evento**

Na simulação orientada a evento, um sistema é modelado pela definição das mudanças que ocorrem no tempo de evento. A tarefa do modelador é determinar os eventos que podem causar a mudança no estado do sistema e então desenvolver a lógica associada com cada tipo de evento. A simulação do sistema é produzida pela execução da lógica associada a cada evento, em uma seqüência ordenada no tempo.

### **5.1.2 Simulação Orientada ao Exame da Atividade**

Neste caso, o modelador descreve as atividades nas quais as entidades do sistema estão engajadas e prescreve as condições que causam o início e o fim de uma atividade. Os eventos que iniciam ou terminam uma atividade não são escalonados pelo modelador, mas iniciados a partir das condições especificadas para a atividade. À medida que o tempo simulado avança, as condições para início ou fim de uma atividade são examinadas. Para nos assegurarmos de que cada atividade é levada em consideração, é necessário examinar todo o conjunto de atividades a cada avanço do tempo.

### 5.1.3 Simulação Orientada a processo

Muitas estruturas de modelos para simulação incluem seqüências de eventos as quais ocorrem em padrões definidos, por exemplo, uma fila de entidades esperando por um servidor. A lógica associada com tal seqüência de eventos pode ser generalizada e definida por uma única afirmação. Uma linguagem para simulação pode então traduzir tal afirmação na seqüência de eventos associada. Uma linguagem orientada a processo emprega tais afirmações para modelar o fluxo das entidades no sistema. Estas afirmações definem uma seqüência de eventos que é automaticamente executada pela linguagem de simulação.

Exemplo do professor:

- crie entidades chegando a cada  $T$  unidades de tempo
- espere pelo professor
- avance o tempo simulado pelo tempo de serviço
- libere o professor
- termine as atividades da entidade

## 5.2 Modelagem para simulação contínua

Em um modelo para simulação contínua, o estado do sistema é representado por variáveis dependentes que mudam continuamente no tempo. Um modelo para simulação contínua é construído pela definição das equações que definem as relações entre suas variáveis de estado, cujo comportamento dinâmico simula o sistema real.

## 5.3 Exemplo: construção e execução de um modelo

Esta seção apresenta o desenvolvimento e execução de um modelo simples de um sistema de atendimento de alunos por professores.

### 5.3.1 Enunciado do sistema

Ao chegar na sala do professor, o aluno deve esperar em uma fila até que chegue a sua vez de ser atendido. Quando chega a sua vez, o aluno conversa com o professor e depois abandona a sala dando oportunidade para que um novo aluno possa ser atendido. Apenas por uma questão de simplicidade, vamos assumir que o tempo de chegada dos alunos e o tempo gasto por cada aluno em atendimento são conhecidos. Os são valores dados na Tabela 1.

*Tabela1 - Tempo de chegada de alunos e tempo de atendimento*

Aluno	Tempo de chegada	Tempo de atendimento
1	6,0	9,0
2	24,0	7,5
3	26,0	10,0
4	35,0	6,5
5	44,0	6,0
6	52,0	5,0
7	58,0	9,0
8	70,0	7,5
9	77,0	6,5
10	89,0	11,0

O objetivo é realizar uma simulação manual do processo apresentado acima, de forma a determinar o tempo médio que um estudante passa tirando dúvidas com o professor e a percentagem do tempo que o professor fica ocupado tirando dúvidas.

### 5.3.2 Modelagem e simulação

O sistema acima pode ser definido pelo professor e pelos estudantes. Não são considerados estudantes que não têm dúvidas. Assume-se então, que o processo de estudo do aluno, através do qual ele percebe suas dúvidas não é importante para o nosso objetivo. Ao excluirmos este processo do escopo do sistema, assumimos que o aparecimento de alunos para atendimento são entradas do sistema, bem como a demanda de tempo de atendimento para cada aluno.

Neste processo de simulação usa-se a linguagem natural, considerando que os alunos são atendidos por ordem de chegada.

Definição dos estados do sistema:

- número de estudantes à espera de atendimento
- ocupação ou não do professor

É importante frisar que existem outros modos de definir o estado de um sistema. Por exemplo, com base no número total de estudantes no sistema.

A mudança de estado do sistema pode ocorrer devido a dois eventos:

- um estudante chega para atendimento
- um estudante acaba de tirar suas dúvidas e parte do sistema

Para ilustrar a simulação pode-se traçar o retrato dinâmico do estado do sistema, pelo processamento dos eventos de chegada e saída de estudantes ordenados no tempo. A simulação manual do exemplo, correspondente às entradas da tabela inicial, pode ser apresentada de acordo com duas visões. A visão do estudante (descrição orientada a processo) e a visão da ocorrência ordenada dos eventos, no caso da visão do professor (descrição orientada a evento).

A tabela seguinte apresenta a visão do estudante, assumindo inicialmente: que não havia estudantes no sistema, que o professor estava desocupado e que o primeiro aluno com dúvida chegou no tempo 2,4 minutos. Na tabela, as colunas 1 e 2 foram tiradas da tabela inicial. A coluna 3 depende se o aluno anterior já partiu do sistema. O valor da coluna vai ser o maior valor entre o tempo de chegada do estudante e o tempo de partida do aluno imediatamente anterior. A coluna 4 é a soma da coluna 3 com o valor do tempo de atendimento dado na tabela inicial. A

coluna 5 computa o tempo que o estudante passa na fila, simplesmente subtraindo do tempo que o estudante começa a ser atendido (coluna 3), o tempo em que o estudante chegou ao sistema (coluna 2). De forma similar, a coluna 6 computa o tempo que um estudante passa no sistema, subtraindo seu tempo de chegada (coluna 2), do seu tempo de saída (coluna 4). O tempo médio que um estudante passa no sistema pode ser facilmente calculado pela tabela abaixo, sendo seu valor igual a 9,75 minutos.

*Tabela 2 - Simulação Manual: a visão do estudante*

Aluno	tempo de chegada a sala	tempo de entrada em atendimento	tempo de saída da sala	tempo de espera na fila	tempo total gasto
1	6,0	6,0	15,0	0,0	9,0
2	24,0	24,0	31,5	0,0	7,5
3	26,0	31,5	41,5	5,5	15,5
4	35,0	41,5	48,0	6,5	13,0
5	44,0	48,0	54,0	4,0	10,0
6	52,0	54,0	59,0	2,0	7,0
7	58,0	59,0	68,0	1,0	10,0
8	70,0	70,0	77,5	0,0	7,5
9	77,0	77,5	84,0	0,5	7,0
10	89,0	89,0	100,0	0,0	11,0

Uma descrição do exemplo orientada a evento pode ser vista na tabela abaixo. Nesta tabela, os eventos foram listados em ordem cronológica, e nela podemos facilmente calcular a percentagem para ocupação do professor, cujo valor é igual a 78%. Para manter a ordem cronológica dos eventos, foi necessário manter um calendário de futuros eventos a serem processados, o que foi conseguido colocando em ordem correta os eventos de chegada e partida de estudantes. A ordem correta do processamento dos eventos foi obtida deste calendário, que vai sempre fornecer com exatidão o próximo evento a ser processado.

*Tabela 3 - Simulação Manual: visão do professor.*

Tempo de evento	Aluno	tipo do evento	número de alunos na fila	número de alunos na sala	estado do professor	tempo de ocupação do prof.
0,0	-	início	0	0	livre	-
6,0	1	chegada	0	1	ocupado	-
15,0	1	saída	0	0	livre	9,0
24,0	2	chegada	0	1	ocupado	-
26,0	3	chegada	1	2	ocupado	-
31,5	2	saída	0	1	ocupado	-
35,0	4	chegada	1	2	ocupado	-
41,0	3	saída	0	1	ocupado	-
44,0	5	chegada	1	2	ocupado	-
48,0	4	saída	0	1	ocupado	-
52,0	6	chegada	1	2	ocupado	-
54,0	5	saída	0	1	ocupado	-
58,0	7	chegada	1	2	ocupado	-
59,0	6	saída	0	1	ocupado	-
68,0	7	saída	0	0	livre	44,0

70,0	8	chegada	0	1	ocupado	-
77,0	9	chegada	1	2	ocupado	-
77,5	8	saída	0	1	ocupado	-
84,0	9	Saída	0	0	livre	14,0
89,0	10	chegada	0	1	ocupado	-
100,0	10	Saída	0	0	livre	11,0

Qualquer que seja a visão de descrição do exemplo, a lógica associada ao processo de chegada e partida de estudantes depende do estado do sistema. No caso da chegada do estudante, se o professor está livre ele se torna ocupado, entrando o estudante em atendimento, tendo seu tempo de partida escalonado para o tempo corrente mais seu tempo de atendimento. Se o professor estiver ocupado, o estudante entra na fila de espera, aumentando o número de alunos em espera. No caso de partida de um estudante, se existe algum outro estudante em espera, o estado do professor ainda continua ocupa, o número de estudantes na fila é decrementado de uma unidade, o novo estudante entra em atendimento, sendo sua partida escalonada para o tempo corrente mais seu tempo de atendimento. Contudo, se a fila de espera estiver vazia, simplesmente o estado do professor é alterado para livre.

### Segunda etapa do exercício:

Alterar o modelo incluindo mais uma entidade permanente denominada 'Reprografia'. O tempo de chegada nesta entidade permanente é igual ao tempo de saída do atendimento na entidade 'Professor'. Os tempos de atendimento na Reprografia são apresentados na Tabela 4 a seguir.

*Tabela 4 – Tempo de atendimento na Reprografia*

Aluno	Tempo de atendimento
1	3,0
2	4,5
3	7,0
4	2,0
5	15,0
6	11,0
7	8,0
8	3,0
9	2,0
10	5,0

Objetivos:

1. Média de tempo que os alunos permanecem no modelo
2. Tempo médio de espera na fila da Reprografia
3. Percentual de ociosidade da Reprografia

### Terceira etapa do exercício:

Alterar o modelo incluindo mais uma entidade permanente denominada 'Biblioteca'. Esta entidade permanente deve ser incluída entre a entidade de atendimento do professor e a entidade Reprografia. Assim, o aluno, após ser atendido pelo professor se dirige a Biblioteca e em seguida

a Reprografia. O tempo de chegada na entidade permanente Biblioteca é igual ao tempo de saída do atendimento na entidade 'Professor'. O tempo de chegada na entidade permanente Reprografia passa a ser o tempo de saída de atendimento da entidade Biblioteca.

A entidade Biblioteca possui uma característica diferente das outras. A biblioteca não possui limite de atendimentos simultâneos em função do espaço disponível para acesso ao acervo. Assim que o aluno chega na Biblioteca o atendimento sempre é imediatamente iniciado. Neste caso é essencial tomar cuidado com a ordem de saída dos alunos da Biblioteca. A inexistência de fila pode fazer com que a ordem de saída seja diferente da ordem de chegada, o que irá interferir no comportamento da próxima entidade permanente. Os tempos de atendimento na Biblioteca são apresentados na Tabela 5 a seguir.

*Tabela 5 – Tempo de atendimento na Biblioteca*

Aluno	Tempo de atendimento
1	8,0
2	30,0
3	2,0
4	7,0
5	18,0
6	1,0
7	20,0
8	10,0
9	9,0
10	8,0

Objetivos:

4. Média de tempo que os alunos permanecem no modelo
5. Tempo médio de espera na fila da Reprografia
6. Percentual de ociosidade da Biblioteca e da Reprografia

Obs. A Biblioteca, assim como as demais entidades, é considerada ociosa quando não houver nenhuma entidade temporária fazendo uso dos seus serviços.

#### **Quarta etapa do exercício:**

A entidade permanente Professor deve ser alterada para atender, simultaneamente, até dois alunos. Ou seja, representar o comportamento de um setor de atendimento com dois professores.

Utilizar os mesmos tempos de chegada e de atendimento para simular o modelo com esta nova característica.

Objetivos:

7. Média de tempo que os alunos permanecem no modelo
8. Tempo médio de espera na fila da Reprografia
9. Percentual de ociosidade da Biblioteca e da Reprografia

#### **Quinta etapa do exercício:**

Nesta etapa os alunos número 4 e número 7 retornam para a Biblioteca para buscar mais bibliografia após serem atendidos na entidade Reprografia. Ao saírem da biblioteca (usando o mesmo tempo de atendimento da primeira passagem) passam novamente pela Reprografia (também utilizando o mesmo tempo de atendimento da primeira passagem) e então abandonam o modelo.

O retorno destes alunos à entidades permanentes anteriores deve ser realizado de forma consistente no tempo de simulação.

Objetivos:

10. Média de tempo que os alunos permanecem no modelo
11. Tempo médio de espera na fila da Reprografia
12. Percentual de ociosidade da Biblioteca e da Reprografia

Obs. A Biblioteca, assim como as demais entidades, é considerada ociosa quando não houver nenhuma entidade temporária fazendo uso dos seus serviços.

## 5.4 Exercícios

1. Um porto possui um ancoradouro que pode acomodar somente um barco a cada vez. Barcos chegam entre 30(+5) horas cada, e esperam na ordem de chegada para descarregar. A descarga leva 25(+20) horas. Assim que um barco termina sua descarga ele deixa o ancoradouro. Simule este sistema para 10 barcos.
2. Dados os seguintes tempos de chegada para um centro de serviço único (0,2; 0,6; 2,2; 2,6; 3,0) e os tempos de serviço correspondentes (0,8; 0,4; 1,2; 0,2; 0,2), faça uma tabela para a simulação do sistema, com uma visão orientada a evento, e uma outra com uma visão orientada a processo, calculando quando o serviço começa, o tempo de partida, e o tempo de espera na fila de cada um dos cinco usuários. Sabendo que o tempo total de simulação é de 4,0 unidades de tempo, calcule a utilização, o tempo médio de espera, o tamanho médio da fila e a vazão do centro de serviço.
3. Imagine vários microcomputadores ligados entre si por um caminho fechado unidirecional (rede em anel). Para trocarem mensagens, as várias estações devem esperar pelo recebimento de uma sequência padrão de bits, chamada permissão. Quando de posse da permissão, a estação pode enviar uma mensagem, passando a permissão logo a seguir para a próxima estação no anel. Se uma estação receber a permissão e não tiver mensagens para enviar, deve passar a permissão à frente, imediatamente. A cada mensagem recebida, a estação deve enviar à estação emissora um reconhecimento de recepção da mensagem (um outro padrão especial de bits). Sabendo-se que o tamanho das mensagens é variável, descreva o sistema especificando quem são as entidades, quais são as atividades que estas entidades estão envolvidas, qual o processo por que passa uma entidade e quais os eventos associados ao sistema.
4. Um orelhão é usado por pessoas que querem telefonar. O intervalo médio de chegada dessas pessoas é de 15 minutos (+10). Elas gastam de 5 a 15 minutos telefonando. Simular este sistema para 10 pessoas.
5. Em um banco, um caixa, ao atender um cliente para saque de dinheiro, deve ir até o cadastro para conferir a assinatura do cheque, no que leva um tempo uniformemente distribuído com média CONFERE. Sabendo que existem três caixas para saque, que os clientes escolhem um dos caixas com a mesma probabilidade, que o tempo de atendimento ao cliente, além do



tempo de conferência de assinatura, é exponencialmente distribuído com média ATENDE, e que os clientes chegam ao banco com um tempo entre chegadas também exponencialmente distribuído, mas com média CHEG, faça uma descrição do sistema usando a orientação a evento, e uma outra usando a orientação a processo.

6. Escreva um programa em Pascal, ou uma outra linguagem qualquer, para a simulação do problema do professor tirando dúvidas apresentado anteriormente. Use orientação a evento sabendo que o tempo entre chegadas e atendimento são randômicos, e não os dados na tabela.
7. Uma universidade mantém um setor de reprografia com 3 máquinas de xerox, sendo que uma das máquinas tem rendimento 30% inferior às outras duas. E destas duas, uma permite a cópia automática de folhas avulsas, inclusive frente e verso. A máquina mais lenta alcança 20 cópias por minuto, se de um único original. Em geral, as duas máquinas manuais são utilizadas para cópia de material encadernado, e a automática para folhas avulsas, contudo, podendo também ser utilizada em material encadernado. Este tipo de material consome tempo adicional entre as trocas de original, naturalmente. O setor mantém um arquivo onde são armazenados materiais deixados pelos professores. Cerca de 30% das cópias são feitas a partir do material arquivado. Os pedidos para cópias são atendidos na ordem de chegada, salvo + - 5% que são considerados urgentes, e que geralmente consomem pouco tempo (poucas cópias de poucos originais). Através do exercício de modelos de simulação, avalie a capacidade de atendimento deste setor diante de aspectos do sistema que considere importante para seu bom funcionamento. (e.g. limite de usuários do xerox para que a fila se mantenha com média inferior a 10 pessoas).
8. O balcão de uma sorveteria  $\sim e$  atendido por um atendente. Os fregueses chegam de acordo com uma distribuição uniforme com intervalo [1-5] minutos. Eles são atendidos em uma base de acordo com a disciplina de fila First In First Out. O tempo de atendimento de cada cliente é dado pela distribuição uniforme com intervalo [1-8]. Construa um ou mais modelos de simulação de modo a determinar o se é viável contratar um segundo atendente para a sorveteria. Justifique sua resposta utilizando argumentos tais como ociosidade dos atendentes, entre outros.

## 6. Modelos baseados em redes de filas

Os sistemas, cuja modelagem, simulação e análise são consideradas, são compostos de recursos. São exemplos de recursos a UCP de um computador ou um enlace de comunicação. Os usuários fazem uso destes recursos ao requererem seus serviços. Durante o tempo que um usuário está recebendo um serviço, outros usuários podem requisitar o mesmo serviço, o que causará a contenção para o uso do recurso, resultando em filas ou linhas de espera. O nível de contenção e a quantidade de recursos requisitados vão afetar decisivamente o comportamento do sistema [SOARES92].

A contenção de recursos dentro de um sistema é um ponto fundamental na operação do mesmo. Se não há contenção, a análise de desempenho é usualmente fácil de se efetuar. Entretanto, sistemas de computação, redes de comunicação, entre outros, geralmente exibem algum grau de contenção.

Um analista raramente construirá um modelo de um sistema sem considerar modelos previamente construídos. Normalmente se utilizam as características gerais de um modelo na construção de um outro modelo. Pode-se usar estruturas de redes de filas na construção de modelos utilizados na modelagem discreta orientada a processo.

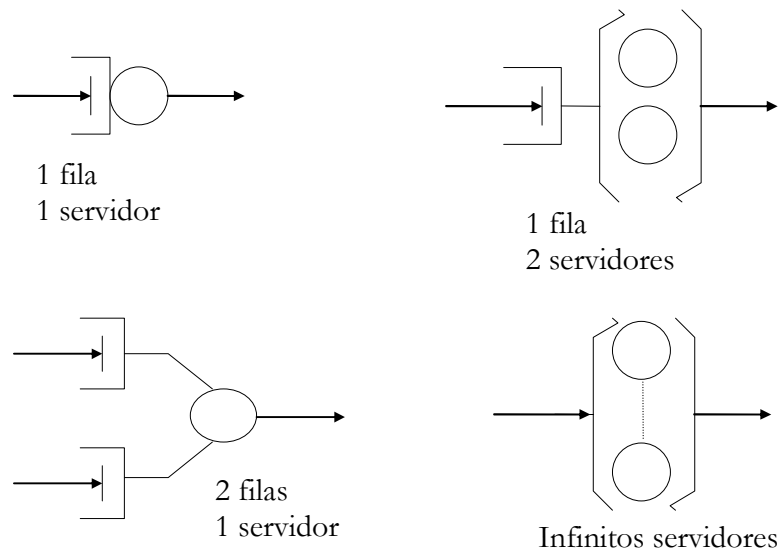
Outros exemplos:

- Redes de Petri;
- Diagramas lógicos.

### 6.1 *Centros de serviço*

Uma rede de filas básica consiste de entidades chamadas *centros de serviço* e um conjunto de entidades chamadas *usuários*, que recebem os serviços nos centros.

Um centro de serviço consiste de um ou mais *servidores*, correspondentes a recursos no sistema modelado, e uma área de espera (fila) para usuários que estão requisitando o serviço. Alguns analistas chamam os centros de serviços de servidores ou filas. A UCP de um computador é um exemplo de recurso que pode ser modelado como um centro de serviço. Um usuário corresponde a uma entidade que circula pelo sistema modelado. Alguns analistas se referem aos usuários como tarefas ou transações. Um programa de computador é um exemplo de entidade que pode ser modelada por um usuário.



*Figura 6.1 – Centros de serviço*

A rede de filas mais simples corresponde a um único centro de serviço. O sistema pode ser muito complexo, mas o modelo é simples. É claro que modelos com um único centro de serviço não podem capturar toda a complexidade dos sistemas reais. Contudo, às vezes, tais modelos oferecem informações bastante úteis. Um único centro de serviço em um modelo de rede de filas possui vários mecanismos relacionando os usuários com a área de espera e o servidor.

Por exemplo, em um modelo de prestação de serviços hipotético existe um único servidor para o atendimento de clientes (entidades temporárias). O atendimento é individual, ininterrupto e é realizado de acordo com a ordem de chegada dos clientes. Quando diversos usuários aguardam por um serviço e o servidor fica livre, uma decisão deve ser tomada sobre qual usuário deve entrar em serviço (disciplina de fila). Encerrado o atendimento de um cliente, um outro cliente da fila pode então ser atendido. Caso a fila esteja vazia, o servidor permanece ocioso até a chegada do próximo cliente [FIL95].

Supondo que ocorram a chegada e o atendimento de  $n$  clientes com intervalos de chegada  $r_1 \dots r_n$  contados a partir do instante 0, e sejam  $s_1 \dots s_n$  os valores dos tempos de atendimento dos clientes. Denote por  $r = (r_i)$  o vetor de intervalos entre chegadas e por  $s = (s_i)$  o vetor de tempos de atendimento. Assim:

- Intervalo entre chegadas:  $r_i$ ;
- Tempo de atendimento:  $s_i$ ;
- Onde:  $i = 1 \dots n$

A partir desses dados, são determinados para cada cliente, os instantes de chegada, o início e término de atendimento ou partida do sistema, dados pelas seguintes expressões:

- Chegada:  $a_i = a_{i-1} + r_i$
- Início de atendimento:  $b_i = \max \{ c_{i-1}, b_i \}$   $i = 1 \dots n$
- Término de atendimento:  $c_i = b_i + s_i$

Desta forma, o instante de chegada de cada cliente é calculado a partir do instante de chegada do cliente anterior adicionando-se o tamanho do intervalo correspondente. O atendimento de um cliente tem início ou no instante de sua chegada, caso o servidor esteja ocioso por não haver cliente no sistema, ou no instante de término do atendimento do cliente anterior, caso contrário. Finalmente, o instante de término de atendimento é obtido somando-se o instante de início de atendimento ao tempo de atendimento.

A partir desses valores, definem-se os tempos de espera na fila e de permanência no sistema para cada cliente:

- Tempo de fila:  $w_i = b_i - a_i$
- Tempo de sistema:  $u_i = c_i - a_i$
- Onde:  $i = 1...n$

O tempo de espera de um cliente é o tempo durante o qual ele permanece na fila. O tempo de sistema é o tempo durante o qual o cliente permanece no sistema e compreende o tempo de espera e o tempo de atendimento:  $u = w + s$ .

Exemplo: sejam  $r=(1,2,1,7,1,4)$  e  $s=(4,2,1,4,3,4)$

*Tabela 6.1 – Exemplo de calculo de a, b, c, u, w, e v*

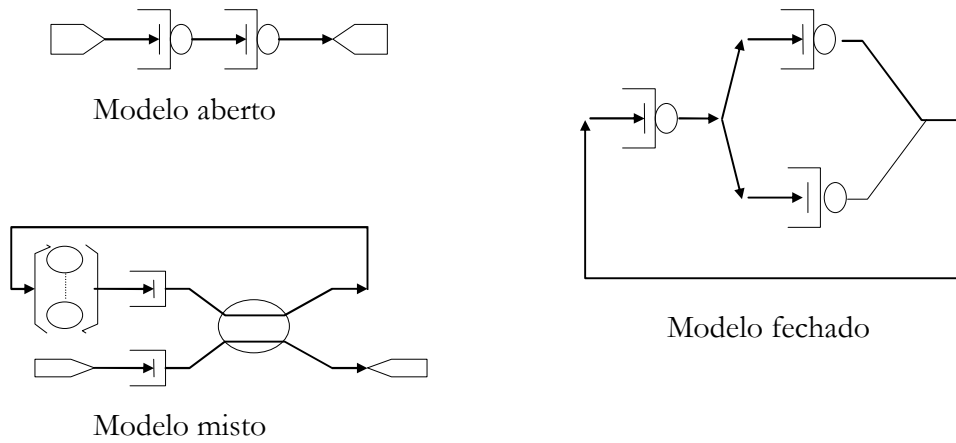
I	r	s	a	b	c	u	W	V
1	1	4	1	1	5	4	0	-
2	2	2	3	5	7	4	2	2
3	1	1	4	7	8	4	3	3
4	7	4	11	11	15	4	0	-
5	1	3	12	15	18	6	3	3
6	4	4	16	18	22	6	2	2

Um centro de serviço pode conter vários servidores. Um servidor pode atender um usuário com uma taxa constante, ou a taxa de serviço pode ser dependente do número de usuários na fila de espera. A fila de espera pode ter uma capacidade infinita ou finita. Pode também existir diferentes tipos de usuários (classe de usuários).

Centros de serviços são recursos ativos. Centros que não realizam serviço são os centros passivos. Centro de serviços passivos são uma extensão bastante poderosa às redes de filas convencionais. A memória de um computador é um exemplo de recurso que poderia ser modelado como um centro de serviço passivo.

Os recursos passivos permitem o compartilhamento de um número finito de itens do recurso (fichas do recurso passivo).

Existem variados tipos de centros de serviços que são interconectados:



*Figura 6.2 – Tipos de modelos*

Quando um usuário deixa um centro de serviço, algum mecanismo deve ser fornecido de forma a determinar para qual centro irá a seguir. Isto é **roteamento**.

## 7. Algoritmos de escalonamento

Algoritmos de escalonamento são utilizados para decidir que usuário deve entrar em serviço quanto um servidor se encontra disponível. São conhecidos como disciplinas da fila. São elas:

- FCFS (first come first served): os usuários entram em serviço exatamente na mesma ordem que chegam ao centro de serviço correspondente;
- LCFS (last come first served): tem a estrutura de uma pilha onde o último usuário a chegar ao centro é o que entrará em serviço assim que um servidor estiver disponível. Alguns consideram esta disciplina preemptiva (se um usuário chegar ao centro com esta disciplina e um outro usuário estiver em serviço, este último é retirado do atendimento, colocado no topo da pilha, entrando o novo usuários em serviço. Assim que este completar seu serviço, o usuário no topo da pilha entra novamente em atendimento).
- RR (round robin): um usuário escalonado é atendido por um intervalo de tempo pequeno, um quantum. Se o serviço não é completado neste tempo, ele é colocado no final da fila e outro usuário é atendido até que esgote seu quantum ou termine o serviço;
- PS (processos sharing): todos os usuários dividem a capacidade do centro de serviço, como se executassem em paralelo (comparada a RR, tem quantum = 0);
- IS (infinite server): não existe fila, todos os usuários são servidos assim que chegam ao centro de serviço;
- PRTY (nonpreemptive priority): usuário em atendimento não é afetado quando um usuário de mais alta prioridade chega ao centro;
- PRTYPR (preemptive-resume priority): a chegada de um usuário com mais alta prioridade a um centro tira de serviço o de prioridade mais baixa, que só reentrará em serviço assim que todos os usuários com prioridades maiores que a sua forem atendidos.

## 8. Estágios para a construção de modelos de representação

O processo de construção de modelos que representam sistemas reais pode ser dirigido por uma seqüência de estágios de forma a simplificar, agilizar, e melhorar a fidelidade do modelo. Considerando a estreita relação entre a simulação e a orientação a objetos, como discutido anteriormente neste texto, as técnicas de engenharia de software oferecem contribuições relevantes no trabalho de projeto de modelos.

Neste contexto, a seguir são brevemente apresentados os estágios por que um projetista passa ao construir modelos para simulação de sistemas. São eles:

1. Identificação do problema: neste estágio, o problema é discutido no escopo do sistema real. Por exemplo, a existência de uma ociosidade exagerada em um equipamento que pertença a uma linha de montagem, ou a necessidade de reestruturação no fluxo de informações em um departamento administrativo, ou o desempenho insuficiente de um protocolo de comunicação em um multicomputador. Uma vez identificado um problema, pode-se avaliar a viabilidade do uso da simulação para o trabalho de busca de alternativas coerentes. Esta avaliação acontece em consonância com os propósitos da simulação.
2. Formulação do problema: após identificado, o problema é então ‘formulado’ com vistas à simulação. Tomando como exemplo o protocolo de comunicação do item anterior, o buffer designado para o armazenamento de mensagens entrantes pode estar sendo insuficiente. Devido à natureza evolucionária da simulação, a definição do problema é um processo contínuo, que ocorre durante todo o estudo;
3. Objetivos: na seqüência, o objetivo é fazer com que as políticas do protocolo sejam manipuladas de forma que o número de mensagens no buffer não ultrapasse um determinado limite. A partir daí são definidos os objetivos específicos, estágio que enumera as características internas do sistema real que precisam ser representadas no modelo. Por exemplo, a frequência com que o protocolo recebe mensagens para enviar, tamanho das mensagens, etc.
4. Construção do modelo: um modelo vai consistir de uma descrição estática e de uma descrição dinâmica. A descrição estática define os elementos do sistema e suas características. A descrição dinâmica vai definir o modo como os elementos do sistema interagem causando mudanças no estado do sistema no decorrer do tempo. O modelador deve conhecer bem a estrutura e as regras de operação do sistema e saber extrair o essencial do sistema, sem incluir detalhes desnecessários. A quantidade de detalhes incluída no modelo deve ser baseada no propósito para o qual foi construído;
5. Determinação dos dados de entrada e saída: geralmente os valores de entrada são de início hipotéticos ou baseados em alguma análise preliminar. Depende do objetivo do modelo;

6. Tradução do modelo: diz respeito à tradução do modelo para uma forma reconhecida pelo computador;
7. Verificação: consiste em determinar se o modelo traduzido executa no computador como esperado;
8. Validação: tem por objetivo comprovar que ele representa o sistema real com fidelidade suficiente para garantir a obtenção de soluções satisfatórias para o problema original. Normalmente são utilizados cenários em que o comportamento real já é conhecido previamente e que pode ser comparado com o comportamento obtido com a simulação. Também pode envolver uma comparação da estrutura do modelo e do sistema, e comparações do número de vezes que decisões fundamentais ou tarefas dos subsistemas são realizadas. Outros modelos do mesmo sistema também podem ser utilizados na validação;
9. Plano de tática e estratégia: refere-se ao estabelecimento de condições experimentais para a execução da simulação. Ou seja, o plano estratégico consiste no desenvolvimento de um projeto experimental eficiente tanto para explicar as relações entre o resultados simulados e as variáveis controladas, quanto para determinar a combinação dos valores das variáveis controladas que minimizariam ou maximizariam a resposta simulada. Já o plano tático consiste em determinar como cada simulação deve ser realizada para se obter o máximo de informações sobre os dados;
10. Experimentação e análise dos resultados: exercitação do modelo e interpretação dos resultados;
11. Implementação dos resultados e documentação do modelo.

### ***8.1 Trabalho: Implementação de uma ferramenta de simulação***

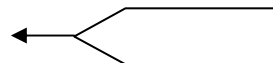
Este trabalho tem como objetivo a especificação e construção de uma ferramenta de simulação, baseada no conceito de ‘centros de serviço’, que suporte a execução de modelos em computador. As características do tipo de modelos que a ferramenta deve suportar são descritas nesta seção, além das características da ferramenta. A linguagem de implementação é de livre escolha do aluno, desde que seja uma linguagem convencional e não de simulação.

As entidades permanentes que a ferramenta deve oferecer para a construção de modelos são identificadas pelas figuras a seguir.

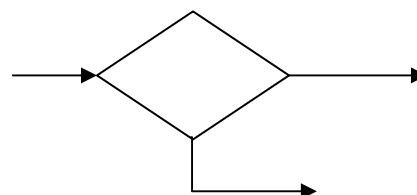
(a) Chegada de entidades temporárias



(b) Saída de entidades temporárias

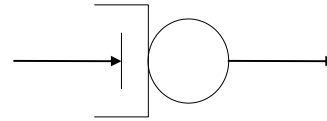


(c) Roteamento

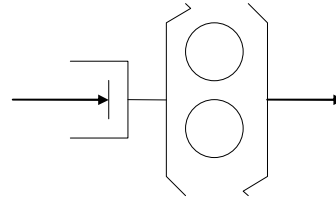




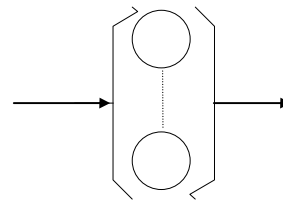
(d) Centro de serviço (1 servidor)



(e) Centro de serviço (2 servidores)



(f) Centro de serviço (infinitos servidores)



A entidade (a), que representa a chegada de entidades temporárias no modelo, obedece a um padrão de chegada definido por uma distribuição de probabilidade uniforme. Os atributos mínimo e máximo devem ser fornecidos pelo projetista do modelo.

A entidade (b) representa a saída de entidades temporárias do modelo e implementa apenas um contador acumulador.

A entidade (c) representa a escolha de uma rota no modelo que as entidades temporárias podem seguir. Esta entidade não consome tempo e o projetista deve fornecer, como atributo, o percentual de entidades temporárias que desviam a rota (para baixo).

A entidade (d) representa um centro de serviço com um servidor. Implementa contenção e o tempo de atendimento é dado por uma distribuição uniforme. Os atributos mínimo e máximo da distribuição precisam ser fornecidos pelo projetista do modelo.

A entidade (e) representa um centro de serviço com dois servidores. Implementa contenção e o tempo de atendimento é dado por uma distribuição uniforme para ambos os servidores. Os atributos mínimo e máximo da distribuição são fornecidos pelo projetista do modelo e podem ser diferentes para cada servidor.

A entidade (f) representa um centro de serviço com infinitos servidores. Não implementa contenção e o tempo de atendimento é dado por uma distribuição uniforme para ambos os servidores. Os atributos mínimo e máximo da distribuição são fornecidos pelo projetista do modelo e são os mesmos para todos os servidores.

Na construção de um modelo o projetista pode combinar qualquer quantidade de entidades permanentes e em qualquer ordem. A ferramenta deve manter a consistência do modelo durante a execução.

Para a construção de modelos deve ser definida uma estrutura para que o projetista, utilizando um arquivo texto, construa a especificação do modelo. Tomando como exemplo a estrutura a seguir:

CS1-1 4,8 CS1-3

Nesta estrutura, o centro de serviço com um servidor número 1 (CS1-1) atende as entidades temporárias em tempos que variam entre 4 e 8 unidades de tempo. Após atendidas, as entidades são encaminhadas para o centro de serviço com um servidor número 3 (CS1-3).

A definição de estruturas para especificar todas as informações relevantes para a construção de modelos faz parte do trabalho. Para executar o modelo, a ferramenta deve fazer a leitura das especificações a partir do documento texto e então executar a simulação.

Os resultados da simulação devem ser gravados em um arquivo de saída com nome definido pelo projetista do modelo no momento da execução. Após a execução (ou exercício) de um modelo, a ferramenta deve informar os seguintes resultados:

- Ociosidade das entidades permanentes, individualmente por servidor;
- Ociosidade média das entidades permanentes;
- Tempo médio de espera das entidades temporárias na fila de cada entidade permanente;
- Tempo médio de atendimento das entidades temporárias em cada entidade permanente;
- Tempo médio de toda a permanência, no modelo, das entidades temporárias.

Um estudo de caso simples deve ser implementado e executado na ferramenta construída. A existência deste estudo é relevante para a validação da ferramenta.

O instrumento de apresentação do trabalho é o artigo. Para tanto, deve ser escrito um texto usando o formato da SBC ([www.sbc.org.br](http://www.sbc.org.br)) contendo: título, identificação dos autores, resumo, introdução, desenvolvimento, conclusão, e referências bibliográficas. Basta um artigo por grupo de trabalho.

A avaliação é individual por aluno. De posse do artigo, o aluno deve demonstrar domínio sobre a ferramenta considerando: uso, detalhes de implementação, e justificativas em relação às decisões de projeto da ferramenta.

## 9. Paradigmas de Simulação orientada a objeto

Este capítulo foi retirado na íntegra da tese de doutorado [COP 97] desenvolvida no programa de pós-graduação em computação do Instituto de Informática da UFRGS.

Com o objetivo de adaptar o conteúdo aos propósitos da disciplina de simulação, considerando também níveis adequados de exploração dos temas para graduação, algumas partes das abordagens originais foram omitidas e/ou remodeladas (simplificadas).

### 9.1 Introdução

A orientação a objetos tornou-se popular nos últimos anos como uma abordagem poderosa para o desenvolvimento de sistemas computacionais complexos. Reusabilidade e extensibilidade estão entre os benefícios prometidos quando se usa seus conceitos.

Particularmente, a reusabilidade pode ser abordada em termos de composição e tecnologias de geração. As tecnologias de composição caracterizam-se pelos conjuntos de componentes atômicos que, no caso ideal, são reutilizados sem alterações. No caso ideal dispõem-se de componentes passivos que são operados (compostos) por um agente externo. Exemplos de tais componentes podem ser esqueletos de programas, subrotinas, funções e classes de objetos.

As tecnologias de geração são mais difíceis de caracterizar porque os componentes reutilizáveis não são facilmente identificáveis como entidades concretas. Os itens reutilizáveis são inseridos em um programa, como por exemplo os geradores de aplicações.

Em orientação a objetos, as classes abstratas e os frameworks são tipos de projetos reutilizáveis. Uma classe abstrata é uma classe que não fornece a implementação de todas as operações que definem a interface, necessitando de redefinições em subclasses.

Um framework funciona como um molde para a construção de aplicações ou subsistemas dentro do domínio de uma aplicação. Basicamente, aplicações específicas são construídas especializando as classes do framework para fornecer a implementação de alguns métodos, enquanto a maior parte da funcionalidade da aplicação é herdada.

Considerando que um modelo de simulação pertence à classe dos sistemas de software, nada mais natural do que aplicar esses (reusabilidade e extensibilidade) conceitos no desenvolvimento de sistemas de simulação. Deve ficar claro, entretanto, que existe uma grande diferença entre um paradigma de simulação, isto é, as idéias e recursos usados na construção de um modelo, e um paradigma de projeto e implementação aplicado ao desenvolvimento de sistemas de simulação. Linguagens orientadas a objetos podem ser aplicadas na implementação de sistemas de simulação que utilizam conceitos de modelagem distintos. Ainda que todos possam ser chamados de sistemas orientados a objetos, pode haver confusão quanto ao significado do termo *simulação orientada a objetos*.

## 9.2 *Simulação orientada a objetos*

Em geral, um projeto de simulação progride através de seqüências lógicas e estruturadas de atividades. Para melhorar esse processo, o conceito de ambiente de simulação envolve uma linguagem de simulação e ferramentas de suporte. A linguagem de modelagem é a maior contribuição para um ambiente de simulação porque influencia diretamente a maioria dos outros recursos.

Quando se usa conceitos de orientação a objetos para desenvolver um sistema, o mesmo é organizado como uma coleção de objetos discretos que incorporam tanto as estruturas de dados como o comportamento. Esta definição é coerente como a idéia de que modelos de simulação podem ser vistos como uma coleção de entidades que interagem entre si. Levando este fato em consideração, pode-se inferir que um modelo de simulação normalmente incorpora conceitos de orientação a objetos, uma vez que este paradigma é uma forma natural de se construir modelos de simulação porque pode-se facilmente mapear entidades em objetos. Essa idéia é reforçada como o fato de que a primeira linguagem a incorporar conceitos de orientação a objetos foi a linguagem Simula.

Entidades de um modelo de simulação podem ser vistas como agregações de uma estrutura estática e de uma estrutura dinâmica. Ambas são descritas utilizando-se rotinas e estruturas de dados. No entanto, as abordagens clássicas de simulação (orientação a eventos, orientação a processos e orientação a atividades) geram um código onde a descrição individual de cada entidade não é facilmente identificável. Na abordagem orientada a eventos, por exemplo, a descrição do comportamento pode encontrar-se espalhada em diversas rotinas de evento. Em um modelo de tamanho considerável, a identificação de quais rotinas descrevem o comportamento de uma determinada entidade pode não ser trivial. Além disso, não existe relação visível entre a descrição do comportamento, composta de rotinas, e a descrição da estrutura estática, em geral composta por variáveis e estruturas de dados. Na abordagem orientada a processos, o problema é minimizado no que diz respeito à descrição do comportamento na medida em que uma única rotina, uma rotina de processo, descreve todo o ciclo de vida da entidade. O problema persiste, porém, no que diz respeito ao relacionamento entre a estrutura estática e a estrutura dinâmica da entidade. Além desses aspectos, nenhuma das abordagens possui construções que permitam identificar com clareza os mecanismos de comunicação utilizados pelas entidades para troca de informações entre outros fatores. Este tipo de projeto produz, em geral, código muito rápido mas difícil de ler e de manter.

A descrição individualizada das entidades de simulação é um problema de projeto de software. A solução está relacionada com a escolha de uma metodologia de projeto/implementação que enfatize a modularização e o encapsulamento. Diversos autores apontam como melhor solução o uso da abordagem orientada a objetos. A compatibilidade entre a orientação a objetos e os formalismos de simulação discreta tem sido destacada. O mapeamento de entidades em objetos é uma forma natural de se agregar a descrição das entidades.

Ao mapear-se entidades de simulação em objetos, entretanto, alguns aspectos tem de ser levados em consideração. O estado de um objeto é descrito pelo conjunto de seus atributos de dados e seus valores possíveis. Para isso, existem duas perspectivas (propostas por diferentes autores [COP97]):

- o estado de um objeto é descrito pelo conjunto de seus atributos de dados e seus valores possíveis
- o estado de uma entidade de simulação é definido pelo subconjunto de seus atributos que são especificamente usados para descrever o comportamento da entidade.

Por exemplo, em um objeto que representa um buffer, o estado pode ser definido apenas pelo número de peças armazenadas na fila. No que se refere às mensagens, em orientação a objetos uma mensagem usualmente corresponde à ativação de um processo do objeto receptor. Em modelos de simulação, entretanto, a troca de mensagens é mais complexa. São necessárias mensagens temporizadas de maneira a permitir o escalonamento de eventos futuros. Mensagens não temporizadas são úteis apenas para operações que não implicam em avanço do tempo simulado.

Considerando os aspectos relativos à troca de mensagens, existem dois tipos de entidades:

- entidades de receptor passivo, onde a entidade emissora determina o instante de tempo em que o receptor irá processar a mensagem;
- entidades de receptor ativo, onde a entidade receptora armazena as mensagens em um buffer para processamento posterior.

Deste modo, pode-se identificar claramente que um ambiente de simulação deve distinguir o paradigma de simulação da estratégia de projeto/implementação oferecida. Desta forma, pode-se definir ambientes de simulação orientados a objetos como sendo aqueles que oferecem os conceitos de orientação a objetos como uma estratégia de projeto e implementação, independentemente dos paradigmas de simulação suportados pelo ambiente.

### ***9.3 Proposta de classificação de paradigmas de simulação apresentada em [COP97]***

As entidades são o elemento fundamental de um modelo de simulação. Pode-se dizer que um modelo de simulação nada mais é do que a descrição do comportamento de um conjunto de entidades. Desta forma a identificação das entidades que compõem um modelo, bem como a análise dos aspectos de seu comportamento que são relevantes para o problema em questão são, conseqüentemente, as etapas mais importantes do processo de modelagem.

No processo tradicional de modelagem, após o estudo do problema, define-se o paradigma de simulação que será usado e adequa-se todas as entidades do modelo ao paradigma escolhido. Quando as entidades são usadas como unidade de modularização, mapeadas para objetos, precebe-se que é possível identificar cada entidade como um modelo individual. Trabalhando-se desta forma, não existe a necessidade de todas as entidades serem descritas segundo o mesmo paradigma de simulação. Para cada entidade pode-se adotar o paradigma mais conveniente. Essa flexibilidade facilita não apenas a descrição das entidades como também seu reuso.

A seguir é apresentada uma classificação para entidades de modelos de simulação discreta baseada nos diferentes aspectos envolvidos na sua descrição [COP97]. Tais aspectos são identificados e as respectivas abordagens de modelagem apresentadas. A classificação diz respeito às entidades e não ao modelo porque considera-se as entidades como pequenos modelos individuais que, agrupados hierarquicamente ou não, formam modelos mais complexos. Classificam-se as entidades:

- Quanto ao atendimento de solicitações
- Quanto à forma de descrição dos eventos
- Quanto ao mecanismo utilizado para troca de informações

- Quanto à autonomia da entidade
- Quanto ao tempo de permanência no sistema
- Quanto à capacidade de tomar iniciativas
- Quanto à ótica pela qual é feita a descrição do comportamento das entidades

### 9.3.1 Quanto ao atendimento de solicitações

Durante uma simulação, uma entidade é freqüentemente solicitada a fornecer informações ou executar ações. Dependendo do tipo de entidade, esta poderá ter a capacidade de escolher entre atender ou não a uma solicitação. Entidades de receptor passivo não têm poder de escolha. Neste caso a entidade reage no momento da solicitação, sendo que a entidade emissora determina, através de uma solicitação, quando a reação irá ocorrer. Entidades de receptor ativo, por outro lado, possuem buffers ou filas para armazenar as solicitações recebidas, podendo escolher o momento em que as mesmas serão atendidas ou simplesmente recusadas. Pode-se considerar ainda uma terceira categoria: Entidades de receptor semi-ativo. Estas entidades caracterizam-se por possuir uma fila de mensagens, ou seja, a entidade emissora não determina o momento em que a solicitação será atendida. A entidade receptora, porém não tem capacidade para alterar a ordem em que as mensagens serão atendidas, ou seja, as mensagens serão atendidas na ordem em que foram recebidas, se houver disponibilidade.

### 9.3.2 Classificação das entidades quanto à forma de descrição dos eventos

O comportamento de uma entidade em um modelo de simulação discreto é sempre descrito em termos de eventos, ou seja, das ações instantâneas que provocam alterações nos estados do modelo. Existem, porém, diferentes abordagens para a descrição desses eventos. Estas abordagens, orientação a eventos, a processos e a atividades, são chamadas de visões de mundo, e a cada uma está associada uma estratégia ou algoritmo de simulação.

### 9.3.3 Classificação quanto ao mecanismo utilizado pelas entidades para troca de informações

Em um modelo de simulação as entidades se comunicam de maneira a trocar informações e serviços. Diferentes técnicas podem ser usadas para efetivar tal comunicação.

O método mais simples é aquele no qual as entidades podem acessar os atributos de dados umas das outras livremente, obtendo assim as informações de que necessitam. Este método será chamado de comunicação livre. Quando o modelo é orientado a processos, são necessários mecanismos de sincronização entre os processos, de maneira a garantir que a informação correta está sendo obtida no momento em que está disponível.

Outras técnicas prevêm o encapsulamento das informações de cada entidade. Neste caso as entidades não têm acesso direto aos atributos de dados umas das outras. Para estas situações, duas técnicas bastante conhecidas são o uso de portas e de mensagens.

O sistema de troca de mensagens parte do princípio no qual toda entidade possui um identificador único. Uma mensagem será composta, em geral, pelo identificador da entidade emissora, identificador da entidade destino, identificador do tipo de mensagem e uma lista de parâmetros. Neste tipo de sistema, toda a troca de informações entre entidades só é possível através de mensagens. Para que o sistema funcione é preciso que se estabeleçam protocolos de comunicação entre as entidades que precisam conversar.

As mensagens podem ser síncronas ou assíncronas. Quando se usam mensagens síncronas, a entidade que envia a mensagem aguarda o tratamento da mesma por parte da entidade receptora. Utilizando-se mensagens assíncronas a entidade emissora continua seu processamento independentemente do tratamento à mensagem enviada se efetuar ou não. Maiores informações em [COP97].

Tanto as portas como as mensagens podem ser usadas com entidades de receptor ativo, semi-ativo ou passivo. No sistemas de comunicação livre, porém, as entidades são de receptor passivo, obrigatoriamente, visto que as entidades não têm nenhum controle sobre o acesso a suas informações.

É importante não confundir a forma com que as entidades atendem as solicitações (receptor ativo, semi-ativo ou passivo) com o fato da troca de mensagens ser síncrona ou não. Uma mensagem síncrona pode ser enviada para uma entidade que se utiliza de um receptor ativo. Neste caso a entidade emissora corre o risco de ter de aguardar até que o destinatário resolva atender sua solicitação antes de poder prosseguir em sua execução. Sinais de controle podem ser acrescentados ao protocolo de comunicação das entidades de forma a otimizar essas situações.

#### **9.3.4 Classificação quanto à autonomia da entidade**

A classificação das entidades quanto a sua autonomia está diretamente ligada ao uso do modelo do ator. Quanto se usa uma linguagem orientada a objetos convencional como C++ ou Object Pascal, os objetos são passivos e compartilham uma única "thread" de execução. Neste caso todas as mensagens são síncronas e as entidades possuem receptor passivo. Quando se usa o modelo do ator, os objetos possuem sua "thread" própria de execução, podem enviar mensagens de forma assíncrona e as entidades podem trabalhar com receptores ativos ou semi-ativos.

As entidades de um modelo de simulação serão consideradas autônomas na medida em que forem mapeadas para objetos. O uso de entidades autônomas associado com mensagens não tipadas, atribui um certa independência à entidade, permitindo, entre outras coisas, que esta não conheça, em tempo de compilação, com quem irá se comunicar durante a execução, que instâncias de um tipo de entidade não previstas em tempo de compilação possam ser ativadas durante a execução ou, ao contrário, que entidades previstas possam ser desativadas sem a necessidade de se interromper o processo de simulação. O uso de entidades autônomas permite a interação com as mesmas, por parte do usuário, durante a simulação, seja para interferir no seu ciclo de vida ou na forma pela qual a entidade reage às mensagens ou simplesmente para obter informações.

#### **9.3.5 Classificação das entidades quanto ao tempo de permanência no sistema**

Uma entidade pode ser permanente ou temporária. Entidades permanentes existem durante todo o tempo de simulação. Se imaginarmos o modelo de um guichê de atendimento onde clientes chegam, aguardam em uma fila, são atendidos e vão embora, os "clientes" são considerados entidades temporárias pois chegam, são atendidos e desaparecem do sistema. O guichê, por outro lado, é considerado uma entidade permanente pois existirá durante toda a simulação.

#### **9.3.6 Classificação das entidades quanto à capacidade de tomar iniciativas**

Existem entidades ativas e entidades passivas. Por entidades ativas entende-se aquelas capazes de tomar a iniciativa de solicitar um serviço ou desencadear uma seqüência de ações. Entidades passivas, por outro lado, são capazes apenas de responder a solicitações. No exemplo do guichê de atendimento, pode-se projetar "cliente" como uma entidade ativa capaz de solicitar serviços. O guichê, neste caso, poderia não tomar iniciativas sendo apenas ocupado por clientes

em certos momentos. Em algumas linguagens de simulação as entidades passivas são chamadas de recursos. Esta classificação, porém, não é muito rígida. Por vezes uma entidade pode ser passiva em relação a uma dada categoria de entidades e ativa em relação a outra. Por exemplo, guichê é passivo em relação aos clientes mas poderia não ser em relação a uma entidade funcionário.

### **9.3.7 Classificação quanto à ótica pela qual é feita a descrição do comportamento das entidades**

Um modelo de simulação, em geral, é composto por entidades que prestam serviços (servidores) e por entidades que solicitam serviços (clientes). Independente da visão de mundo utilizada, o modelo de simulação pode ser descrito tanto do ponto de vista das entidades clientes como das entidades servidores. Se for adotada a ótica do cliente as entidades cliente serão modeladas como entidades ativas e as servidoras como passivas. Se for escolhida a ótica do servidor, irá ocorrer o contrário.

Para melhor entendimento pode-se utilizar como exemplo a linha de produção de uma peça qualquer. Nesta, blocos de material são manipulados em estações de trabalho até que sejam considerados acabados. Pode-se imaginar, então, os blocos como entidades temporárias (clientes), que são atendidas por estações de trabalho modeladas como entidades permanentes (servidores).

Se o modelo for descrito utilizando-se a ótica do cliente, o mesmo será descrito do ponto de vista das peças. Cada bloco de material que entrar no sistema terá conhecimento sobre todas as operações pelas quais deve passar até que seja considerado pronto e será capaz de requisitar, sucessivamente, os serviços necessários. Se for utilizada a ótica do servidor, os blocos de material conterão apenas uma identificação do tipo de peça que deve ser produzido, e cada estação de trabalho saberá o que fazer com tal tipo de peça e para qual estação deve ser enviada depois de processada.

O limite entre as duas abordagens, entretanto, não é bem definido. Nem todos os problemas de simulação se enquadram na situação proposta. Alguns modelos poderão, inclusive, adotar uma abordagem mista onde para algumas entidades será escolhida a ótica do cliente e para outras a ótica do servidor. Uma mesma entidade pode assumir uma postura de servidor perante algumas entidades e de cliente perante outras. A escolha irá depender exclusivamente do tipo de problema.

Algumas linguagens de simulação impõem não apenas a visão de mundo mas também a ótica de descrição do comportamento das entidades.



## 10. Solução do Modelo

A técnica analítica é o método geralmente mais rápido e portanto o preferido, quando aplicável. O problema com o método é que muitas hipóteses de simplificação devem ser feitas para que possamos resolver o modelo. Simulação é mais geral e pode ser aplicada em situações mais complexas. O preço a ser pago pela generalidade é que um tempo maior é exigido na obtenção de medidas de desempenho precisas.

Outra vantagem da solução analítica é a obtenção do resultado exato para o modelo sobre estudo, o que não acontece com a simulação. Por outro lado, devido às simplificações, o modelo pode não ser uma representação precisa do sistema real e o resultado exato obtido ser desprovido de qualquer utilidade.

A **solução analítica** de um modelo de redes de filas exige que algumas restrições sejam satisfeitas. São algumas delas:

- As decisões de roteamento devem ser feitas através da especificação de um conjunto de probabilidades de ramificação
- A posse simultânea de recursos não é permitida
- Somente recursos ativos são permitidos
- Somente fontes, classes e sorvedouros são permitidos na rota
- As disciplinas das filas só podem ser FCFS, PS, LCFS e IS
- Disciplinas de filas com prioridade não são permitidas
- As únicas medidas de desempenho disponíveis são: utilização, vazão, comprimento médio da fila, tempo médio na fila, distribuição do comprimento da fila.

Embora muitos modelos analíticos não possam ser resolvidos por conta da violação de restrições expostas acima, alguns destes modelos podem ser resolvidos através de técnicas especiais, sem uso de simulação. Estas são as **técnicas de aproximação**.

A **Simulação** é um método de solução que tenta imitar o comportamento do sistema, se constituindo em um experimento estatístico que observa o comportamento do modelo no decorrer do tempo. O modelo pode ser exercitado com dados obtidos de medidas do sistema ou com valores aleatórios gerados a partir de distribuições, com a finalidade de representar o tempo de chegada de usuários, tempos de serviço e probabilidades de roteamento. Os valores aleatórios gerados não são de fato randômicos, mas produzidos por um algoritmo, de acordo com a **distribuição de probabilidade**.

À medida que a simulação é executada, os usuários circulam entre os vários recursos. Durante o exercício, medidas de desempenho são calculadas, a partir de dados observados, baseadas nos valores aleatórios utilizados durante a execução. Para muitas simulações, as saídas aleatórias (medidas de desempenho) atingem um estado estacionário. Estar em um estado estacionário não significa que o estado permanecerá o mesmo a partir de então, mas

simplesmente que as que as distribuições associadas com os estados do sistema convergiram para um estado limite.

Na simulação é possível associar atributos aos usuários. Nos atributos são armazenadas informações sobre o usuário respectivo. Por exemplo: identificação do usuário, contadores, tempo de serviço, entre outros. Além dos atributos, um modelo de simulação pode conter variáveis globais, que podem ser utilizadas para a identificação de qualquer tipo de condição, estando seus valores disponíveis aos usuários em qualquer tempo. Um exemplo de variável global seria aquele onde a variável é usada para catalogar o número de usuários presentes em uma cadeia aberta.

A simulação também permite o uso de diferentes tipos de distribuições para o tempo entre chegadas, para os tempos de serviço, para as demandas de trabalho, para a alocação e criação de fichas, e para as atribuições de valores às variáveis. Também, são permitidas disciplinas de filas mais complexas, e podem ser preemptivas.

Pode-se observar também algumas desvantagens na simulação, principalmente relacionadas com os problemas estatísticos próprios da simulação.

## 11. Uso do computador com o Arena

Como visto anteriormente, a simulação de sistemas é realizada através da repetição de um determinado processo que representa o sistema real através de um modelo. Deste modo, a simulação deve ser executada em quantidade suficiente para que os resultados obtidos tenham um grau de confiabilidade aceitável. Uma das principais ferramentas utilizadas na execução destas repetições é o computador.

Linguagens voltadas à simulação começaram a surgir na década de 60. Como exemplo, pode-se citar a linguagem GPSS, utilizada por um longo período. Ao longo do tempo as linguagens foram evoluindo até que, na década de 80, começaram a surgir programas com capacidade de executar a simulação visual. São exemplos: ARENA, TAYLOR e PROMODEL.

Cada programa de simulação possui uma característica básica que os diferencia uns dos outros, ou seja, a visão do sistema real. Este termo significa a forma com que o programa foi concebido, ou como ele vê um sistema a ser simulado.

### 11.1 ARENA

O ARENA foi criado pela empresa Systems Modeling em 1993 e ele é o sucessor de um outro produto, o SIMAN, desenvolvido em 1982. Este produto foi uma evolução da arquitetura do GPSS e, durante anos, foi líder entre os produtos de simulação para uso geral. Em 1990 ele recebeu um complemento chamado de CINEMA que adicionava maiores habilidades gráficas. Este conjunto foi melhorado e, a partir de 1993, passou a se denominar ARENA.

O ARENA utiliza uma interface gráfica para que o usuário possa fazer uso de módulos utilizados na descrição de sistemas reais. Esta interface gráfica facilita grandemente o uso do ambiente.

O ARENA também disponibiliza o *Input Processor* e o *Output Processor*. O primeiro permite analisar dados reais do funcionamento do processo e escolher a melhor distribuição estatística que se aplica a ele. O segundo é uma ferramenta com diversos recursos que permite analisar dados coletados durante a simulação.

### 11.2 O ambiente do ARENA

Após inicializado, o ARENA apresenta uma tela cinza chamada de *Contents Region* e nela é que o modelo vai sendo construído. Acima é apresentada a barra de ferramentas e um menu secundário. No lado esquerdo está a *panel area*, onde são apresentados os painéis com os templates (botões que representam os módulos disponíveis).

Também são disponibilizados alguns modelos de simulação como exemplos. Após o reconhecimento inicial do ambiente, é aconselhável o estudo de alguns desses modelos exemplos.

Para executar o modelo são apresentados na parte inferior esquerda do monitor alguns ícones voltados para o controle da execução.

### 11.3 Construção de modelos

Nesta seção são apresentados os principais passos para a construção de um modelo de simulação no ARENA, através de um exemplo. Como exemplo é utilizado um sistema de pedágio com as seguintes distribuições de probabilidade:

- Veículos chegam ao pedágio a intervalos EXPO(30);
- A duração do atendimento é de GAMM(6.98,1.51).

Para que se possa simular um sistema com o uso do ARENA é preciso descrever:

- O fluxo dentro do sistema
- As durações, distâncias, velocidades, etc.

A técnica utilizada pelo ARENA para se montar qualquer modelo é a programação visual. Ou seja, o fluxo do sistema é montado na tela do computador na forma de um **diagrama de blocos**.

No caso do problema do pedágio é preciso fornecer ao ARENA o diagrama de blocos com a ocorrência dos eventos:

- O veículo chega (bloco ARRIVE)
- O veículo é atendido (bloco SERVER), o que compreende as seguintes etapas:
  - Entra na fila e permanece até chegar sua vez
  - Atendimento
  - Desocupação do posto do pedágio
- O Veículo sai do sistema (bloco DEPART).

Para a construção do modelo, inicialmente é preciso inicializar um modelo clicando em *menu principal (arquivo)* e *novo*. Em seguida clicar na orelhinha (attach) e inserir o template COMMON que contém os principais blocos (ou módulos) para se construir o diagrama de blocos na *contents area*.

Em seguida deve-se clicar sobre o módulo **arrive** e clicar novamente sobre a área do modelo (onde o modelo será construído). Assim já está adicionado ao modelo o módulo **arrive**, cuja função é gerar usuários para o modelo, ou seja, representa a chegada de veículos.

Após a inserção desse módulo no modelo, é preciso fornecer informações à este módulo. Para isso, de um duplo clique sobre o retângulo que contém a palavra Arrive. Assim é apresentada uma janela de configuração do módulo deste.

Nesta janela, inicialmente deve ser fornecido o nome do componente do modelo (ou estação). Neste caso o nome pode ser 'chegada'. A seguir, levar o cursor até o campo *Time Between* e escolher uma das opções (distribuições de probabilidade). Neste caso, de acordo com o enunciado do sistema, deve ser a distribuição EXPO(Mean), onde a variável Mean deve ser substituída por 30.

Em seguida, na seção abaixo (Leave Data), deve-se fornecer dados de saída do usuário, isto é, o nome da próxima estação de trabalho (ou componente). Para tal, deve-se digitar ‘atendimento’ (por exemplo) no campo *Station*, que será o nome dado para o componente de serviço. Para campos não preenchidos o ARENA assume valores default. Por fim, confirmar as alterações da janela.

O próximo módulo será do tipo SERVER (disponível no template COMMON). Para inserir este módulo no modelo realizar a mesma sequência de passos utilizada para inserir o módulo ARRIVE. Este bloco vai representar todos os eventos que ocorrem no posto de atendimento: entrada na fila, pagamento e saída do posto.

Para alterar os campos deste módulo deve-se dar um duplo clique sobre o mesmo e a respectiva janela será apresentada. Então preencher os campos conforme o seguinte:

- O nome da estação (ou componente) é ‘atendimento’
- O próximo campo a ser preenchido seria Resource, onde deve-se colocar o nome do trabalhador que opera a estação de trabalho (caixa, por exemplo). No entanto o campo Resource foi preenchido automaticamente pelo ARENA com ATENDIMENTO-R. Trata-se de uma iniciativa do ARENA em dar o nome do recurso igual ao nome da estação de trabalho acrescido da letra R.
- Fornecer agora a duração do atendimento no pedágio preenchendo o campo Process Time com GAMM(6.98,1.51).
- Na seção onde se fornece dados sobre a saída da transação clique no campo ao lado de onde se lê station e digite SAIDA, para dizer que o veículo, após ser atendido pelo servidor do pedágio, se dirige para o local denominado SAIDA.
- Confirme as alterações realizadas.

Para alguns dos campos, valores não foram fornecidos. Nestes casos o Arena assume valores default. São eles:

- Capacity type=capacity. Ou seja, o servidor estará disponível o tempo todo. A opção schedule pode ser utilizada para informar os momentos de indisponibilidade do servidor
- Para configurar um atendente disponível no bloco servidor, o Capacity deve ser igual a 1.

O último bloco é o bloco DEPART. O procedimento para incluir este bloco na Contents Region é o mesmo dos anteriores. Com um duplo clique neste bloco a sua tela de apresentação é mostrada. Basta fornecer o nome do bloco (geralmente é saída) e confirmar.

## **11.4 O exercício do modelo**

Até este ponto, a estrutura do modelo e a configuração de cada um dos blocos está concluída. Contudo, ainda deve-se fornecer outros dados tais como a duração da simulação ou o tipo de figura utilizada. Para isso usa-se o bloco SIMULATE.

Utiliza-se o procedimento já conhecido para inserir um bloco SIMULATE e mostrar a sua tela de apresentação. O quadro vermelho que aparece acima do bloco é a figura que será utilizada na simulação. Esta figura fará o papel do usuário do modelo para os quais os centros de serviço prestam seus serviços.

Neste bloco, o campo Length of Replication deve ser preenchido com a duração da simulação. Uma unidade corresponde a um segundo de simulação real. Confirmar a configuração do bloco.

Agora o modelo pode ser executado. Para isso, basta pressionar no botão RUN que fica na parte inferior esquerda do monitor. Neste momento, o ARENA irá conferir o modelo e informar se existe algum tipo de erro. Se não existe, o modelo é executado, e ao final de sua execução, os resultados da simulação são apresentados.

Como exercício, pode-se alterar o tempo entre chegada dos usuários de expo(30) para expo(5). Neste caso, o tamanho da fila não será suficiente. Para resolver este problema, pode-se alterar o número de atendentes do servidor alterando o campo Capacity de 1 para 3, por exemplo. Assim o módulo de atendimento fica com 3 atendentes. O modelo pode então ser executado novamente, e os resultados observados.

No exemplo acima, não existe interligação entre os módulos. O conhecimento da rota de um usuário foi dada pelo preenchimento do campo Station da seção Leave Data dos blocos Arrive e Server. No entanto, é aconselhável se fazer a interligação dos blocos para facilitar o entendimento do modelo.

Para fazer a interligação é necessário pressionar na opção connect dentro dos módulos Arrive e Server. Em seguida, pressionar no ícone para interligação que fica na barra de ferramentas e fazer a interligação a partir das setas que aparecem nas extremidades dos blocos.

(Fazer alterações no modelo sobre distribuições, número de elementos do servidor, etc, e analisar os resultados).

## 11.5 *Template*

Um módulo é uma unidade que pode ser utilizada para construir partes de um modelo de simulação. Já um painel é um conjunto de módulos destinados a modelar um tipo específico de sistema.

Um **template** é então o conjunto de um ou mais painéis que compreendem todos os módulos necessários para a construção de um tipo específico de modelo.

No ARENA, o painel COMMON contém módulos básicos para modelar o funcionamento da maioria dos sistemas. É composto por módulos lógicos e de dados. O painel TRANSFER contém dispositivos específicos de transferência como, por exemplo, transportadores, esteiras, etc.

O painel SUPPORT contém módulos especializados e módulos básicos para criação de lógica em outros níveis.

O módulos lógicos do painel COMMON são os seguintes:

- ARRIVE: para a criação de entidades que chegam no sistema;
- DEPART: libera as entidades que deixam o sistema;
- SERVER: servidor básico, utilizado para representar a prestação de um serviço à um usuário por um determinado tempo;
- INSPECT: Fornece a opção de enviar uma fração de entidades para um módulo alternativo. Utilizado quando se deseja representar falha na prestação de serviços à um percentual dos usuários que o solicitam;
- ADVANCED SERVER: fornece capacidade adicional de modelação além do servidor básico (SERVER);
- PROCESS: representa o processamento do recurso no ADVANCE SERVER

- ENTER: representa a entrada de um ADVANCE SERVER;
- LEAVE: representa a saída de um ADVANCE SERVER;

Os módulos de dados do painel COMMON são:

- EXPRESSIONS: permite a definição de expressões para o modelo;
- QUEUE: permite a definição de filas adicionais para os módulos servidores;
- RECIPES: permite a definição de dados relacionados a uma localização;
- SEQUENCES: permite a definição de seqüências utilizadas pelas entidades no sistema;
- SETS: fornece a possibilidade de se definir conjuntos de recursos, filas, etc.
- SIMULATE: define informações de projeto, número de replicações e símbolo de animação durante a simulação;
- STATICS: permite a definição de dados pré estabelecidos, ou estáticos, usados no módulo recipes;
- TATISTICS: permite ao usuário definir e solicitar estatísticas adicionais;
- STORAGE: permite a definição de armazenamento, que são áreas de permanência de entidades;
- VARIABLES: Permite a definição de variáveis;

### **EXERCÍCIO:**

- Modelar uma fábrica de pipas da seguinte forma:
  - Chegada entre pedidos com uma distribuição EXPO(12)
  - Tempo de corte TRIA(8,10,12)
  - Tempo de costura TRIA(10,11.5,14)
  - Tempo de inspeção igual a 7.5, sendo que o índice de rejeição é igual a 15%
  - Tempo de deslocamento entre Corte e costura igual a 2
  - Tempo de deslocamento entre inspeção e costura igual a 5
  - Tempo de deslocamento entre costura e inspeção igual a 5
  - Tempo de deslocamento entre inspeção e remessa igual a 5 minutos
  - O tempo de simulação deve ser de 960 minutos.

## ***11.6 Trabalho para ser entregue no final da disciplina***

Até a última semana de aula, do respectivo semestre letivo, deve ser entregue um trabalho de modelagem e simulação de um sistema qualquer à escolha do aluno (existente ou hipotético). Como ponto fundamental do trabalho, é importante que a descrição em texto do sistema esteja coerente com os objetivos, com os modelos estático e dinâmico, e resultados do exercício do modelo.

O trabalho pode ser feito em grupos, contudo a avaliação é INDIVIDUAL. No momento da apresentação cada aluno deve demonstrar domínio sobre os detalhes de funcionamento e implementação do trabalho.

No momento da apresentação do trabalho o aluno deve ter em mãos (impresso) a parte escrita, no formato de artigo, de acordo com o seguinte:

- Título
- Identificação dos autores
- Resumo (100- 300 palavras)
- Introdução (identificação da área, identificação do problema, solução proposta no corpo do artigo, organização/estrutura do artigo)
- Simulação (breve fundamentação sobre princípios, aplicações, ..., da simulação)
- Descrição informal (texto) do sistema em estudo (sistema que será simulado)
- Tarefa de abstração do sistema real para levantamento de requisitos, seguido do projeto e construção do modelo de representação. Também, o estudo de variabilidade. Este item pode ser subdividido em múltiplas seções no artigo.
- Exercício do modelo de simulação e avaliação dos resultados
- Conclusões e aspectos que podem dar continuidade ao trabalho
- Bibliografia

O Anexo I apresenta um exemplo comentado da apresentação escrita do trabalho aqui solicitado. No texto deste exemplo são discutidos aspectos técnicos que precisam ser considerados nos estágios de desenvolvimento de uma simulação, como discutido na seção 7.7 desta apostila.



## 12. Probabilidade e estatística em simulação

### 12.1 *Coleta de dados*

Aquisição de dados é o processo de obtenção de dados sobre um fenômeno de interesse (tanto utilizando o sistema, quanto o modelo). Na aquisição de dados pode-se utilizar documentos, questionários, pesquisa de campo e experimentação física.

A experimentação física é comumente o método de obtenção de dados mais caro e que consome mais tempo. O processo inclui medição, gravação e edição de dados. Um grande cuidado deve ser tomado ao se planejar a experiência, de forma a assegurar que os dados obtidos sejam representativos.

Para determinar o comportamento de um sistema, podemos observá-lo enquanto ele trabalha. Isto é uma medida de desempenho, freqüentemente utilizadas na condução de avaliação de desempenho de sistemas de computação. Veja bem, medidas de desempenho são aquelas obtidas do sistema, enquanto avaliação de desempenho são resultados (medidos ou estimados) obtidos do modelo.

No que se refere à experimentação, mesmo quando medidas são possíveis e suficientes para se entender o funcionamento do sistema, é difícil utilizar os dados obtidos para prever o comportamento futuro do sistema sob mudanças de carga e configuração. Enquanto se tenta projetar um novo sistema, medidas são raramente de alguma utilidade. Nestes casos só resta trabalhar com o modelo para aquisição de dados.

### 12.2 *Estatística descritiva*

Tanto a coleta de dados para a definição das entradas do modelo, quanto nas coletas de dados de desempenho a partir do modelo, encontramos o problema de como converter os dados obtidos em uma forma usável. Sempre há o interesse pelo tratamento projetado para resumir características importantes de um conjunto de dados. Tais tratamentos normalmente resumem os dados às custas da perda de alguma informação contida nos mesmos.

#### 12.2.1 **Agrupamento de dados**

O agrupamento em classes é um dos métodos para transformação de dados. Neste método, os dados são resumidos pela tabulação do número de dados que ocorreram dentro de cada classe. Esse tipo de tabela é chamado de tabela de freqüência.

No exercício do professor, os dados agrupados em classes são os tempos de espera dos estudantes para atendimento pelo professor. Uma tabela de distribuição obtida na resolução do problema poderia ser como a apresentada a seguir:

Tempo de espera do aluno	Número de alunos
0 -> 10	8
10 -> 20	14
20 -> 30	21
30 -> 40	11
40 -> 50	8
50 -> 60	7
> 60	5

Classes com sem limite inferior ou superior fixados são chamadas de classes abertas. Veja exemplo na tabela acima.

Outra, é a tabela de frequência acumulada, que é obtida pela adição sucessiva das frequências da tabela anterior.

Tempo de espera do aluno, menor que:	Número acumulado de alunos
10	8
20	22
30	43
40	54
50	62
60	69
$\infty$	74

Outra variação é obtida pela conversão da tabela cumulativa, pela divisão de cada classe cumulativa pelo número total de dados.

Tempo de espera do aluno, menor que:	Distribuição de frequência
10	0,108
20	0,297
30	0,581
40	0,730
50	0,838
60	0,932
$\infty$	1

As tabelas são muitas vezes apresentadas sob a forma de gráficos de forma a facilitar a interpretação dos dados. A forma gráfica mais comum é o histograma, que apresenta as classes de frequência como retângulos, cujos comprimentos são proporcionais à frequência nas classes.

### 12.2.2 Estimação de parâmetros

**População** é o conjunto de dados que consiste de todas as observações possíveis de uma variável aleatória. **Amostra** contém apenas parte de todas as observações possíveis.

Outro método de resumir um conjunto de dados é vê-lo como uma amostra, que é então utilizada para a estimação de parâmetros da população associada. Os parâmetros da população usualmente de maior interesse são a média (parâmetro) e a variância (estatística)

Estatísticas baseadas em observação - interessa apenas o valor (sexo das pessoas que frequentam um determinado lugar)

Estatísticas sobre variáveis persistentes no tempo - valores das amostras são definidos sobre o tempo (número de pessoas em uma fila)

## 12.3 Geração de números e amostras aleatórias

O método mais prático para a geração de amostras aleatórias a partir de uma dada distribuição é primeiro gerar uma ou mais amostras uniformemente distribuídas, entre 0 e 1, e então transformar as amostras uniformes nas novas amostras desejadas. Amostras independentes, uniformemente distribuídas no intervalo 0 a 1, são chamadas **números aleatórios**.

São três os métodos para a obtenção de números aleatórios:

- através de uma leitura de uma tabela de números aleatórios;
- uso de dispositivos físicos que geram sinais aleatórios; e
- emprego de uma equação recursiva, que gera o  $i$ -ésimo número randômico a partir do  $(i-1)$ -ésimo número. Como os números são gerados a partir de uma equação determinística, são chamados de pseudo-aleatórios

### 12.3.1 Amostras aleatórias

Para a geração de amostras aleatórias existem quatro métodos fundamentais que podem ser utilizados.

O **método da transformada inversa** é a técnica mais simples. Ele parte do pressuposto de que a variável aleatória  $R=F(x)$  (onde  $F(x)$  é a função de distribuição de probabilidade da variável aleatória  $X$ , de onde se quer a amostra) é uniformemente distribuída no intervalo  $[0,1]$ . Assim, para gerar uma amostra aleatória da distribuição  $X$ , nós geramos primeiro um número aleatório  $r$  e depois resolve-se a equação:  $r = F(x)$ .

O **Método da aceitação / rejeição** consiste na geração de amostras a partir de uma distribuição. Algumas destas amostras são então rejeitadas de forma que as amostras restantes tenham a distribuição desejada. Métodos de aceitação e rejeição foram desenvolvidos tanto para distribuições contínuas quanto para discretas, sendo a eficiência do método dependente do número de rejeições exigidas.

O **método da composição** parte do pressuposto que a função densidade da variável cuja amostra se deseja, pode ser obtida da soma ponderada de funções de distribuição. Este método pode então ser usado para selecionar uma das subdistribuições e, o mesmo procedimento ou um outro qualquer, para se obter uma amostra da subdistribuição. Pode-se ver assim que um método para pode ser dependente de outro.

Quanto aos métodos que empregam **probabilidades especiais**, algumas vezes a distribuição de onde se deseja obter uma amostra possui alguma propriedade especial ou está relacionada com outra distribuição, a partir da qual é mais fácil obter uma amostra aleatória. Tal propriedade ou relação é então usada na geração desejada.

### 12.3.2 Teste de aleatoriedade

A validade estatística dos resultados de uma simulação depende do grau de aleatoriedade dos números aleatórios gerados. Exatamente por isso, vários procedimentos estatísticos foram desenvolvidos para testar esta aleatoriedade. Infelizmente, contudo, mesmo estes testes não garantem que a sequência gerada seja apropriada para uma particular aplicação.

## 12.4 Distribuições de Probabilidade

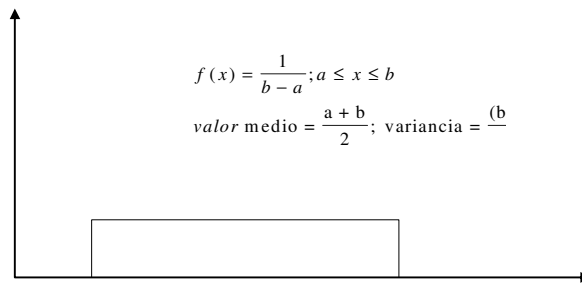
Distribuições de probabilidade ocorrem em quase todas as partes. Usuários chegam a um centro de serviço de acordo com um certo padrão chamado *distribuição de chegada*. A quantidade de serviço requerida segue um padrão correspondente, que chamamos de *distribuição de demanda de trabalho*. No caso especial, e ao mesmo tempo típico, em que o servidor tem uma capacidade fixa, o valor da distribuição da demanda de trabalho pode ser dividido pela capacidade, na determinação de um padrão chamado *distribuição de tempo de serviço*. Filas passivas têm também distribuições associadas, correspondentes ao número de fichas requeridas por um usuário.

Uma solução para a introdução desta variabilidade estatística em modelos consiste na observação e medida de padrões que ocorrem durante um certo período de operação do sistema. Os valores específicos são armazenados e posteriormente utilizados como entradas para o modelo de simulação. Também pode-se introduzir uma distribuição aleatória nos modelos, o que é mais viável.

O modo mais simples de se representar uma distribuição é através do uso de funções de distribuição e funções de densidade de probabilidade.

### 12.4.1 Distribuição uniforme

A distribuição uniforme especifica que os valores compreendidos entre um mínimo e um máximo são equiprováveis. O uso da distribuição uniforme geralmente significa um completo desconhecimento da variável aleatória, conhecendo-se apenas seus limites.



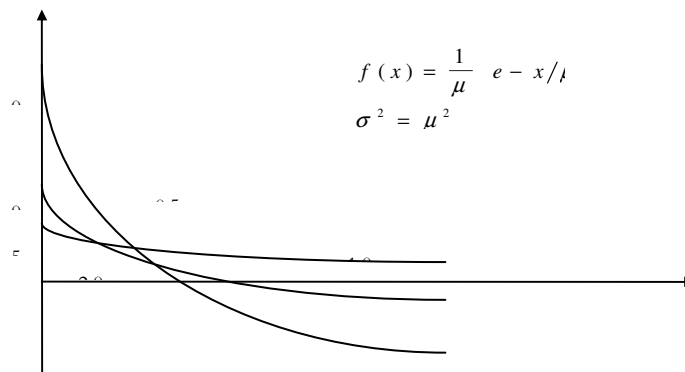
Função de densidade de probabilidade uniforme

### 12.4.2 Distribuição exponencial

O intervalo entre ocorrências tem uma distribuição exponencial se:

- a probabilidade de acontecer um evento em um intervalo de tempo  $\Delta t$  pequeno é proporcional ao tamanho deste intervalo;
- a probabilidade de ocorrência de mais de um evento neste intervalo é nula e;
- a ocorrência de um evento é independente da ocorrência de outros.

Uma propriedade importante da distribuição exponencial é que ela não tem memória, ou seja, se uma atividade vem ocorrendo há  $t$  unidades de tempo, então a probabilidade que esta atividade termine em  $\Delta t$  unidades de tempo é a mesma que se a atividade tivesse começado no instante  $t$ , qualquer que seja  $t$ . O uso da distribuição exponencial supõe uma grande variância.



### 12.4.3 Distribuição de Poisson

A distribuição de Poisson é uma distribuição discreta. Se a duração do intervalo de tempo entre chegadas é exponencialmente distribuída e se as chegadas ocorrem uma de cada vez, então o número de chegadas em um intervalo fixo qualquer tem uma distribuição de Poisson.

### 12.4.4 O método de Monte Carlo

Em [PRA99] o Método de Monte Carlo é definido como:

*“Uma maneira de se transformar um conjunto de números aleatórios em outro conjunto de números (variáveis aleatórias), com a mesma distribuição da variável considerada”*

Em resumo, o este método oferece uma garantia de que, quando um processo de geração aleatória de números é exercido para uma grande massa de dados, os valores simulados guardam uma estreita semelhança com os valores reais no que se refere a variáveis randômicas. Ou seja, a duração média obtida dos números sorteados tende a ser idêntica à duração do processo na vida real (sistema real).

## ***12.5 Inferência estatística***

Em estudos de simulação, inferências ou predições, concernentes ao comportamento do sistema sobre estudo, devem ser realizadas, baseadas nos resultados experimentais obtidos da simulação. Devido ao fato de que a simulação possui elementos aleatórios, as saídas de uma simulação são amostras observadas de variáveis aleatórias, e em conseqüência, qualquer assertiva que se fizer quanto à operação do sistema, baseada em resultados de simulação, deve levar em consideração a variabilidade inerente destas saídas. Esta variabilidade é levada em consideração pelo uso de intervalos de confiança ou através de hipóteses de teste.

Para a definição desta variabilidade, pode ser feita uma análise estatística detalhada (durante o processo de obtenção de informação) sobre a precisão e sensibilidade do modelo. Basicamente, deve-se estudar que saída seria obtida caso a simulação fosse realizada novamente ou se tivesse tido um tempo de execução maior. Quando isto é feito, assume-se que o modelo de simulação é estocástico e que os elementos aleatórios do modelo vão produzir saídas que são probabilísticas.

Em [SOARES92] é apresentado um estudo mais aprofundado sobre a inferência estatística em estudos de simulação.

## 13. Estudo estatístico

Os estudos de simulação concernentes ao comportamento do sistema sobre estudo devem ser realizados com base nos resultados experimentais obtidos da simulação. Como a simulação possui elementos aleatórios, deve-se levar em consideração a variabilidade desses resultados. A forma de se considerar esta variabilidade é através dos intervalos de confiança ou hipóteses de teste.

O intervalo de confiança é um intervalo onde o parâmetro da população deve se encontrar, como uma probabilidade dada, chamada nível de confiança.

No desenvolvimento do intervalo de confiança para a média, assume-se que as amostras são independentes e identicamente distribuídas.

### *13.1 Estimação do intervalo de confiança*

Como resultado de pesquisas, existem métodos utilizados para estimação de intervalos de confiança, entre eles o método de replicações independentes, divisão em subintervalos, o método regenerativo e o espectral. Um método não pode ser considerado melhor do que o outro, mas um pode ser mais apropriado que outro para situações determinadas.

No **método das replicações independentes**: Neste método a simulação é repetida várias vezes, partindo sempre do mesmo estado inicial, variando a seqüência de números aleatórios gerados a cada replicação. Normalmente, a seqüência de números aleatórios para a segunda replicação começa onde a seqüência da primeira replicação terminou, e assim por diante. A média de cada subintervalo pode ser calculada pela divisão do somatório de amostras do subintervalo pelo número de amostras deste intervalo.

Esse método tem a propriedade desejável que as amostras são independentes, além disso, o método pode ser usado tanto para análise do estado transitório quanto do permanente. Como desvantagem, cada replicação contém um segmento inicial de partida que pode não ser representativo na análise do comportamento permanente.

No **Método da divisão em Subintervalos** o procedimento consiste na divisão de uma única execução da simulação em subintervalos. Cuidados devem ser tomados na estimação da largura dos subintervalos e quanto ao cálculo em suas fronteiras (um mesmo valor não pode entrar no cálculo em dois subintervalos).

A vantagem do uso de subintervalos na estimativa dos intervalos de confiança é que uma única execução pode ser feita na obtenção da estimativa, além disso, apenas um período transiente é incluído na saída. A desvantagem do procedimento é a necessidade do estabelecimento da largura do subintervalo, de forma a torná-los independentes. Ainda outra desvantagem, diz respeito ao cálculo nas fronteiras de uma rajada. Cuidados devem ser tomados

na computação das médias, quando uma observação se estende por mais de uma rajada, por exemplo, uma chegada no subintervalo  $i$  que parte do sistema no subintervalo  $i+1$ .

O **método regenerativo** divide uma execução em intervalos chamados ciclos. Um ciclo começa quando um estado específico do sistema é atingido e, neste estado, o comportamento futuro do modelo é independente do passado. Quando um retorno é feito a tal estado, o ciclo termina e uma observação independente de cada quantidade de interesse é obtida. Ao se definir o ciclo desta forma, se obtém amostras independentes do modelo, evitando o problema da covariância encontrado no uso de subintervalos. Um problema estatístico diferente surge entretanto, o comprimento do ciclo não é determinístico, mas sim uma variável **aleatória**. O estado de regeneração deve ter as seguintes propriedades: 1- o modelo volta periodicamente ao estado de regeneração. 2- quando o modelo entra no estado de regeneração, o comportamento futuro do modelo depende apenas deste estado. Todo ciclo tem um comportamento estatisticamente idêntico a qualquer outro ciclo.

É desejável que o estado de regeneração ocorra frequentemente em uma simulação de tamanho razoável. Sem esta propriedade, o método não pode ser aplicado com segurança. Também é desejável que o estado seja facilmente detectado pela simulação. Por esta razão, os pacotes de modelagem, que oferecem o método regenerativo para o cálculo do intervalo de confiança, geralmente só permitem como estados de regeneração aqueles especificados apenas pelo número de usuários em cada nó, entendendo que características adicionais dos estados são especificadas implicitamente pela linguagem.

O Em resumo, as vantagens do método regenerativo estão baseadas na obtenção de variáveis aleatórias independentes e identicamente distribuídas para cada ciclo.

O **método espectral**, ao contrário dos outros métodos, não exige que as amostras sejam independentes e identicamente distribuídas. Ele leva em consideração a correlação entre os dados.

Como vantagem sobre as replicações independentes, o método espectral permite fazer uma única simulação ao invés de várias execuções. Isto não exige preocupação com a estimação do estado inicial. Este método se aplica a todas as análises do comportamento de equilíbrio dos modelos simulados, que utilizam redes de fila estendidas, sendo independente da existência ou não de estados de regeneração.

## 13.2 Hipóteses de teste

Em algumas aplicações de simulação, o objetivo é decidir se uma afirmação feita sobre um parâmetro é falsa ou verdadeira. Por exemplo, nós poderemos querer saber se as mudanças em uma regra de atendimento qualquer reduz o tempo de espera em uma fila. Estas comparações são às vezes realizadas usando testes de hipóteses, que devem levar em consideração a variação natural das medidas em simulação.

O procedimento geral de teste de hipóteses pede a definição de uma hipótese nula e uma hipótese substituta. A hipótese nula é usualmente estabelecida com o objetivo de se determinar se ela pode ou não ser rejeitada em favor da hipótese substituta.



São dois os tipos de erro que podem ser cometidos ao se aplicar um critério de decisão de rejeição. Primeiro, uma hipótese nula pode ser rejeitada quando ela é verdadeira. Em segundo, uma hipótese não nula pode ser aceita quando ela é falsa. Uma regra de decisão pode então ser avaliada pelas probabilidades associadas aos erros citados.

### ***13.3 Problemas estatísticos relacionados com a simulação***

A resolução de um modelo de simulação pode apresentar variados erros de diferentes fontes. Uma particular fonte de erro é a estimação imprecisa dos parâmetros de entrada do modelo. Para a execução do modelo, deve-se fornecer valores para as distribuições do tempo de serviço, distribuições do tempo de chegada, probabilidades de rotas, número de fichas de um recurso passivo, população da cadeia, disciplina das filas, entre outros. Os valores fornecidos são simples estimativas dos parâmetros reais. Alguns destes parâmetros são críticos na solução do modelo, e pequenos erros cometidos em sua estimativa podem redundar em grandes erros nas medidas de desempenho.

Existem também erros inerentes ao modelo. A estrutura do modelo pode estar incorreta, podendo omitir alguns elementos chaves importantes, ou seja, o modelo pode conter alguns erros lógicos.

Ao se resolver um modelo por simulação, nós introduzimos também, erros concernentes à variabilidade estatística dos resultados, concernentes ao transitório da simulação, à geração de amostras aleatórias, entre outros.

Inicialmente, se as condições iniciais da simulação não estão de acordo, podem ocorrer problemas. O estado mais comumente utilizado é o vazio. Também pode-se iniciar uma simulação do estado de maior probabilidade de ocorrência ou na média do estado de equilíbrio.

Para reduzir a probabilidade de erros na estimativa da média no estado estacionário pode-se utilizar um período de aquecimento. Para isso, se estabelece um ponto de truncamento, antes do qual, nenhum valor obtido é incluído na estatística.

Também é importante a decisão quanto ao comprimento da sequência de simulação e número de replicações de uma simulação. O uso de poucas sequências de grande comprimento geralmente produz uma melhor estimativa da média no estado permanente, do que o uso de várias sequências de comprimento pequeno. Isto acontece devido ao fato que menos amostras no estado transiente são contabilizadas e/ou menos dados são truncados. Quanto maior o transitório, maior deve ser a sequência de simulação. Contudo, o reduzido número de amostras correspondentes a poucas replicações pode piorar nossa estimação da variância da média. O uso de várias pequenas execuções pode, no entanto, introduzir uma polarização devido às condições iniciais.

O fim de uma simulação pode ser determinado pelo tempo, número de amostras que é determinado pelo número de elementos que entram no sistema, ou número de entidades que será processada pelo sistema.

## 14. Referências Bibliográficas

- [JAIN91] JAIN, Raj. “The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling”, John Willey & Sons, Canada, 1991.
- [SOARES92] SOARES, Luiz Fernando Gomes. “Modelagem e Simulação Discreta de Sistemas”, Editora Campus, Rio de Janeiro, 1992.
- [NAYLOR71] NAYLOR, T.; Balintfy, J.; BURDICK, D.; CHU, K. “Técnicas de Simulação em Computadores”, Editora Vozes, São Paulo, 1971.
- [STRACK85] STRACK, J. “Modelagem e Simulação de Sistemas”, Editora LTC, Rio de Janeiro, 1985.
- [LAW82] LAW, A. ; KELTON, W.: “Simulation Modeling and Analysis”. Editora McGraw-Hill Book, New York, 1982.
- [COP 97] COPSTEIN, B. SIMOO: Plataforma Orientada a Objetos para Simulação Discreta Multi-Paradigma. 1997. 137 f. Tese de Doutorado, UFRGS, Curso de Pós-Graduação em Ciência da Computação. CPGCC da UFRGS, Porto Alegre.
- [FER95] FERSCHA, A. Parallel and Distributed Simulation of Discrete Event Systems. In: Handbook of Parallel and Distributed Computing. McGraw-Hill, University of Vienna, Vienna, Áustria, 1995.
- [FIL01] FILHO, Paulo José de Freitas. 2001. 322 fls. Introdução à Modelagem e Simulação de Sistemas: Com Aplicações em Arena. Visual Books Editora, Florianópolis, SC.
- [FIL95] FILHO, Clovis Perin. Introdução à simulação de sistemas. 1995. 159f. Campinas, SP: Editora da UNICAMP.
- [PRA99] PRADO, Darci. 1999. 122 fls. Teoria das Filas e da Simulação. Editora de Desenvolvimento Gerencial, Belo Horizonte, Série Pesquisa Operacional, vol2.

## **Anexo I**

# Uso da simulação para avaliação de desempenho dos algoritmos do Elevador e do Menor Seek Primeiro, utilizados no deslocamento do braço de HDs

Braulio Adriano de Mello  
(braulio@inf.ufrgs.br)

Universidade Federal do Rio Grande do Sul  
Instituto de Informática - UFRGS  
Disciplina de Modelagem e Simulação de Sistemas Discretos  
Prof.: xxxxxxxxxxxx

## 1. Introdução

A simulação se apresenta como uma ferramenta adequada para o estudo do comportamento de sistemas em geral. Seus benefícios estão amparados na possibilidade de realizar experimentos sobre um sistema real sem que seja preciso interferir nesse sistema. Para isso, são utilizados modelos de representação.

Como exemplo do uso da simulação, neste trabalho é apresentado um exercício de representação do comportamento de dois algoritmos utilizados no deslocamento do braço de suporte das cabeças de leitura de discos rígidos. Os algoritmos são: elevador, e menor seek primeiro [TAN 95].

Partindo do estudo do comportamento de ambos os algoritmos, é realizada a definição do modelo conceitual dos algoritmos, e em seguida, a implementação do modelo utilizando a linguagem de simulação Simscript [SIM 89]. Diante dos resultados alcançados perante o exercício (execução) dos modelos implementados, é então desenvolvido um breve estudo de comparação de desempenho entre os algoritmos considerando uma situação de demanda de acesso ao disco previamente especificada e delimitada.

O trabalho apresenta inicialmente uma descrição do sistema real (algoritmos), seguida da definição e delimitação do modelo conceitual, com base em um propósito definido (objetivo). São então abordadas as especificações sobre a coleta da amostra utilizada como base para definições das probabilidades e demais aspectos de implementação. Em seguida, são apresentados alguns aspectos de avaliação dos resultados alcançados visando a comparação de desempenho entre os algoritmos simulados, as considerações finais e referências.

## 2. Descrição do sistema real

O deslocamento do braço de discos rígidos é determinado por algoritmos que podem implementar formas distintas de escalonamento. Estes algoritmos interferem diretamente na performance apresentada no trabalho de leitura e gravação de dados nos discos de armazenamento.

O trabalho de acesso aos dados do disco pode ser feito considerando o bloco como a unidade de alocação utilizada pelo sistema operacional. Ou seja, para gravar um arquivo o sistema operacional deve alocar tantos blocos quanto necessário, sendo que parte do espaço do último bloco alocado pode não ser totalmente utilizado. Isto é conhecido como fragmentação interna.

Para se medir o desempenho do algoritmo que controla o deslocamento do braço, de modo a alcançar os blocos do disco que contém a informação desejada, é preciso conhecer o cilindro em que o respectivo bloco se encontra. Como os blocos que contém as informações de um arquivo em particular podem estar em cilindros diferentes no disco (fragmentação externa), entende-se que o braço deve se deslocar alcançando os blocos desejados para que o acesso solicitado tenha sucesso.

O tempo necessário para que o braço possa alcançar o cilindro desejado é um dos principais fatores que determinam o desempenho do disco. Este tempo é conhecido como tempo de seek. A latência rotacional (tempo para que o setor procurado se posicione embaixo da cabeça de leitura) e o tempo de transferência dos dados também influenciam no desempenho geral do disco.

Considerando a mesma latência rotacional, o mesmo tempo de transferência de dados e a mesma velocidade de deslocamento do braço, resta então a ordem de busca dos cilindros como fator diferenciador quanto ao desempenho, sendo este o objeto de estudo deste trabalho.

Serão considerados, para este trabalho, dois algoritmos distintos que determinam as regras de busca dos cilindros que contém os blocos desejados, sendo: algoritmo do *menor seek primeiro*, e algoritmo do *elevador*.

É considerada também a existência de uma tabela, indexada pelo número do cilindro, com

todas as requisições pendentes para cada cilindro. Desta forma, qualquer um dos algoritmos citados poderá identificar, de acordo com sua política, qual o próximo cilindro a ser alcançado. A tabela é dinâmica, ou seja, a medida em que o braço do disco vai alcançando os cilindros solicitados, novas requisições podem ser recebidas pela tabela.

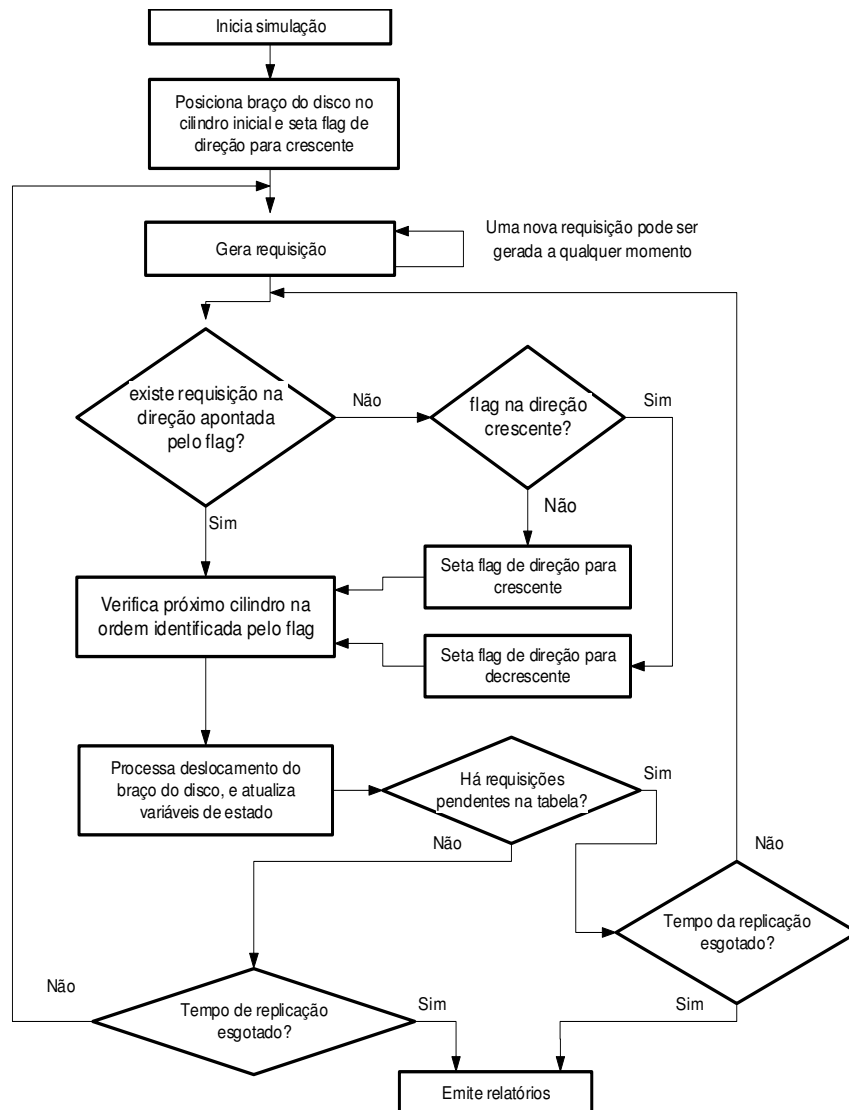


Figura 2.1 - Fluxograma do algoritmo do Elevador

### Modo de Operação dos Algoritmos de deslocamento do braço

O algoritmo do menor seek primeiro, partindo do cilindro em que se encontra, desloca o braço do disco para o cilindro mais próximo, se houver requisição pendente. Este algoritmo apresenta bom desempenho, no entanto tende a manter o braço no meio do disco cometendo, de certa forma, injustiça com os cilindros mais próximos do início ou fim do disco.

O algoritmo do elevador faz com que o braço do disco se desloque apenas em uma direção enquanto houver requisições naquela direção. Por exemplo, se houver as requisições para os cilindros 4, 20, 32, 40 e 64, e o braço está no cilindro 32 quando ocorre uma requisição para o cilindro 30, o

braço segue até os cilindros 40 e 64. Quando não mais houver requisições para cilindros na direção atual, troca de direção e procede da mesma forma. Ou seja, depois de atender a requisição para o cilindro 64 o braço será deslocado ao cilindro 30, seguindo em direção ao endereço inicial enquanto houver requisições nesta direção.

O algoritmo do elevador corrige a injustiça com os cilindros mais afastados do centro do disco, contudo tende a prejudicar a performance em comparação ao algoritmo do menor seek primeiro. O que nem sempre acontece, dependendo da ordem das requisições.

O comportamento de ambos os algoritmos é apresentado na figura 2.1 e figura 2.2, respectivamente.

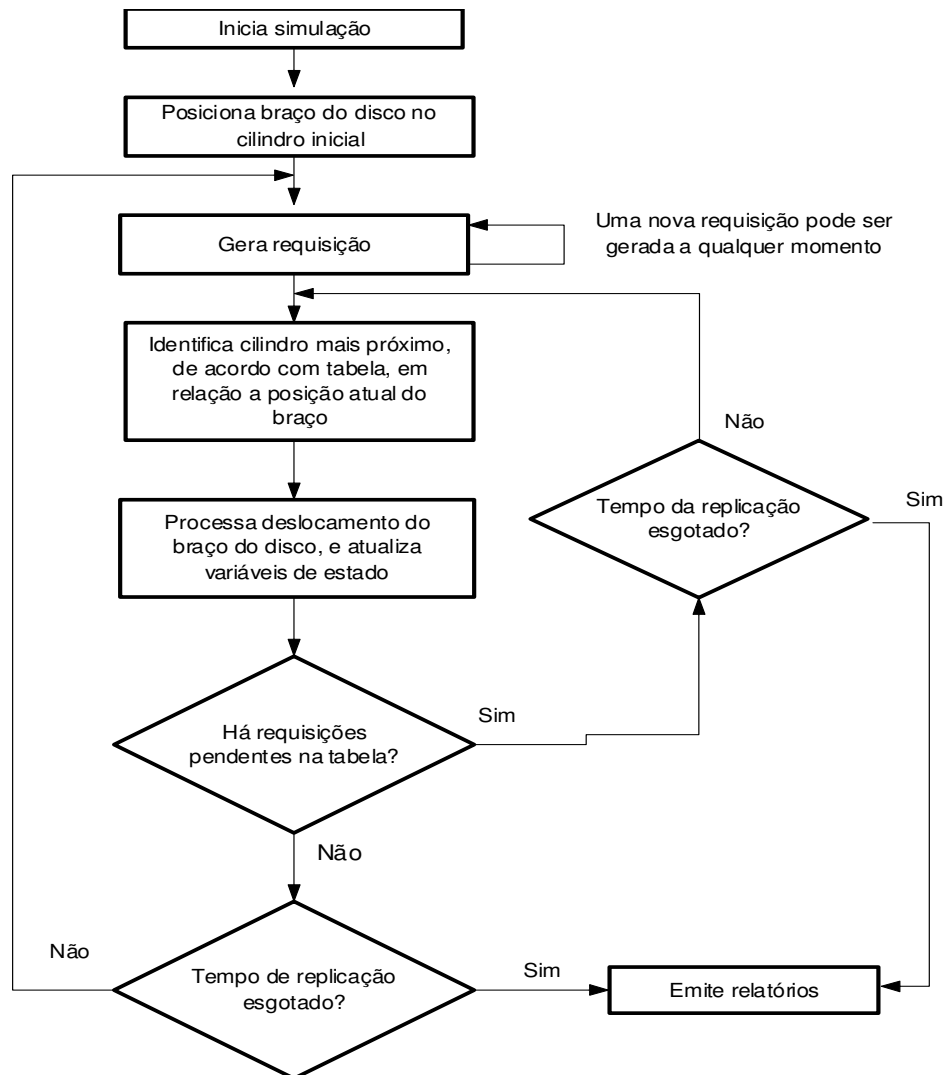


Figura 2.2 - Fluxograma do algoritmo do Menor Seek Primeiro

- Utilização do braço do disco.

### 3. Objetivos

#### Objetivo Geral:

Avaliar o desempenho entre o algoritmo do *menor seek primeiro* e o algoritmo do *elevador*.

#### Objetivos Específicos:

- Tamanho máximo, médio e mínimo da fila de requisições pendentes;

Bytes por setor: 512

Setores por trilha (default): 63

Default heads: 16

Cilindros: 16.383

Setores: 19.925.880

RPM: 7.200

Latência rotacional: 8,32 msec (a média rotacional é de 4,16 msec)

É conveniente modelar o tempo entre chegadas das requisições para leitura/gravação de blocos através de uma distribuição exponencial. Em consequência, o número destes eventos em um intervalo de tempo segue uma distribuição de Poisson. Como, neste trabalho, os resultados do experimento serão coletados levando em consideração o número de requisições que são emitidas em intervalos fixos de tempo (1 segundo), a preferência na escolha da distribuição de probabilidade indica a Poisson.

Assumindo o disco em estudo como tendo alto nível de fragmentação externa [TAN 95] e pouco espaço livre, o número do cilindro onde está o bloco solicitado segue uma distribuição uniforme de 1 até 16.383. Este número representa o número de cilindros do disco, de acordo com especificações do fabricante.

Para o cálculo do tempo gasto pela braço do disco para se deslocar de um cilindro para outro são utilizados valores fornecidos pelo fabricante, sendo:

**Full stroke:** A metade do tempo gasto pelo disco para se deslocar do primeiro cilindro de dados até o último cilindro de dados, e novamente até o primeiro. Este tempo é dado pela média de 100 experimentos como descrito, sendo: **18,6 mls.**

**Número de cilindros:** 16.383.

### 4. Modelo conceitual

Para a definição do ambiente de simulação que visa avaliar os dois algoritmos de deslocamento do braço propostos, são utilizadas as especificações técnicas do disco Seagate ST310210A [BAR 00]. O manual deste produto fornece as seguintes especificações:

Média (time seek): 8,5 msec(Read); 9,5 msec(Write)

Seek time (track-to-track): 1,2 msec(Read); 1,9 msec(Read)

Step Time: Significa o tempo para mover a cabeça de leitura de um cilindro para o cilindro adjacente. Corresponde ao seek time track-to-track.

**Tempo de seek:** Tempo necessário para, a partir de um cilindro, posicionar a cabeça de leitura no cilindro imediatamente posterior ou anterior. O tempo gasto varia no intervalo de **1,2 a 1,9 mls.**

Então, o tempo geral de seek entre os cilindros solicitados pelas requisições é calculado da seguinte forma:

$$\text{Tempo seek} = \text{randi.f}(1,2,1,9,1) + ((18,6 / 16383) * \text{abs.f}(\text{cilindro requisitado} - \text{cilindro atual}))$$

#### Descrição do Modelo

São modelados como **entidades permanentes** do sistema o braço do disco, cilindros (num total de 16.383 de acordo com fabricante do disco utilizado).

São **entidades temporárias** do modelo as requisições de acesso à blocos que são emitidas.

**Variáveis:** identificador do cilindro requisitado; identificador da posição atual do braço do disco; estrutura de dados que armazena as requisições pendentes; número de requisições pendentes; flag de direção (para o algoritmo do elevador); demais variáveis de estado necessárias para demonstrar o estado do sistema nos instantes do tempo de simulação.

**Eventos:** chegada da requisição na estrutura de dados; início do atendimento da

requisição; início do deslocamento do braço do disco; fim do deslocamento do braço do disco; fim do atendimento da requisição.

#### **Abstração do sistema (granularidade):**

É relevante obter como resultado valores relativos à fila de requisições e utilização do braço de cada algoritmo para um mesmo número de requisições emitidas. Considerando para isso, as mesmas condições de processamento, fragmentação do disco, espaço livre do disco, entre outros. Deste modo, não foram considerados aspectos tais como diferentes exigências de acesso ao disco e nível de fragmentação do disco. Tais aspectos poderiam tornar o modelo excessivamente complexo diante do objetivo proposto.

**Eventos exógenos:** tempo entre chegadas das requisições de acesso (distribuição de Poisson com média  $E(x)$ ); tempo de atendimento de cada requisição; identificação do cilindro requisitado (distribuição uniforme com  $\min=1$  e  $\max=16.383$  [BAR 00]).

**Eventos endógenos:** ordem (instante) em que a requisição chega na estrutura de dados; tempo em que inicia o atendimento à cada uma das requisições; tempo em que o braço inicia a busca do cilindro requisitado; tempo em que este cilindro é alcançado; tempo em que a requisição abandona o modelo.

## **5. Experimento para coleta da amostra**

O número de requisições de acesso ao HD está diretamente relacionado ao modo como o computador está sendo utilizado. Ou seja, finalidade dos processos que estão sendo executados. Quando disparado um processo que indexa um banco de dados por exemplo, naturalmente o acesso ao disco é expressivo. Por outro lado, quando disparado um processo cuja finalidade não exige acesso constante ao disco, a situação é inversa.

Os autores que trabalham com teoria de Sistemas Operacionais (SO) abordam estes aspectos e, quando discutidos itens estratégicos de SO, valorizam o estudo do tipo de aplicação

que será utilizada [TAN 95]. Em função disso, propostas não tão usuais podem ser mais adequadas de acordo com a finalidade.

Para este trabalho, foi assumido um processamento com emissão de requisições de acesso ao disco variando entre altas e baixas taxas. Para a execução do experimento foi utilizado um programa simples cujo objetivo era criar, gravar e ler arquivos aleatoriamente (distribuição uniforme [TAN 95]), mantendo o tamanho dos arquivos menores do que uma unidade de alocação do disco (4kb neste caso). Durante a execução deste programa, foram utilizados outros programas com o objetivo de provocar concorrência pelo uso do processador (calculadora por exemplo). Isto interferiu no grau de multiprogramação, cujo resultado pode ser observado (Fig.1) nas variações do número de requisições.

Para os propósitos deste trabalho é importante que a mesma situação seja fornecida para ambos os algoritmos de deslocamento do braço do disco, pois o objetivo é avaliar/comparar a performance destes. Alterando as distribuições de probabilidade, a mesma avaliação também pode ser realizada combinando situações diferentes de estado do disco e nível de acesso ao disco. Por exemplo: alta e baixa taxa de fragmentação; frequência de acesso ao disco; disco com pouco, médio ou muito espaço livre.

#### **Cálculo da Média**

$$E(X) = \sum_{i=1}^n x_i \cdot p(x_i)$$

Tendo os valores da amostra coletada através do experimento (Fig.1), e considerando que a variável  $X$  é discreta, calcula-se então a Esperança Matemática (Média) com o uso da fórmula:

A tabela (Tab.1) apresenta o número de requisições de acesso ao disco que ocorreram durante um segundo ( $X$ ), e o a frequência de  $X$  durante o experimento.

$X$	$P(X)$	$X \cdot P(X)$
2	7/511	14/511
4	8/511	32/511
6	58/511	348/511



7	1/511	7/511
8	61/511	488/511
9	1/511	9/511
10	90/511	900/511
11	2/511	22/511
12	75/511	900/511
13	2/511	26/511
14	61/511	854/511
15	8/511	120/511
16	55/511	880/511
17	7/511	119/511
18	29/511	522/511
19	3/511	57/511
20	14/511	280/511
22	11/511	242/511
24	11/511	264/511
26	3/511	78/511
28	3/511	84/511
30	1/511	30/511
		E(X)=12,2818

Tabela 5.1 - Tabela de frequência da ocorrência de requisições de acesso

Então, de acordo com a amostra coletada,  $E(X) = 12,2818$ , sendo:

$$E(x) = \frac{\sum_{j=1}^k P(x)_j \cdot X_j}{n}$$

Sendo que, para a Distribuição de Poisson,  $E(X) = V(X) = \alpha$ , então  $\alpha = 12,2818$  em:

$$P(x) = \frac{e^{-\alpha} \alpha^i}{i!}$$

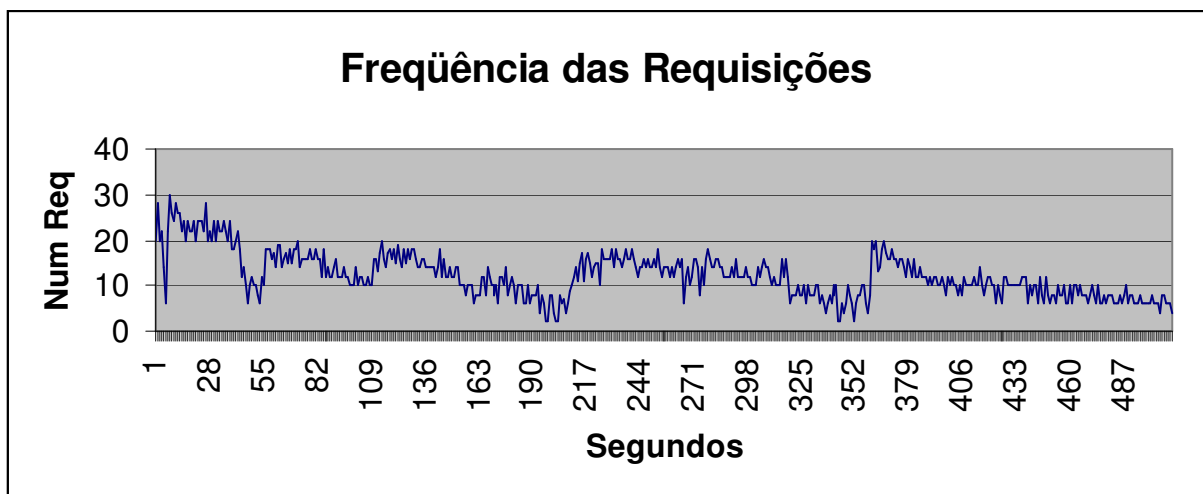


Figura 5.1 - Frequência das Requisições obtidas através de experimento a cada instante de tempo

- Desvio padrão:

$$\sigma = \sqrt{V(x)} = \sqrt{12,2818} = 3,5045$$

- Grau de confiança:

Na aplicação do teste do Chi-Quadrado o  $\chi^2$  calculado se manteve acima do  $\chi^2$  observado, rejeitando a hipótese.

Justifica-se este aspecto pelo fato de que o experimento foi realizado considerando interferências expressivas no grau de multiprogramação, intercaladas com instantes com baixa interferência. Assim, houve grande variação na frequência observada (como pode ser observado na Tab.1).

Deste modo, para estabelecer grau de confiança é mais adequado que se faça o experimento para diferentes situações de uso do computador individualmente. Por exemplo, combinando alto/médio/baixo grau de multiprogramação e alto/médio/baixo acesso ao disco. Considerando estes aspectos, pode-se alcançar uma relação mais homogênea entre valores esperados e valores observados no teste do chi-quadrado.

No entanto, neste caso basta que a mesma situação seja submetida aos 2 algoritmos pelo fato de que o objetivo deste trabalho é o estabelecimento de uma relação de desempenho entre ambos.

## 6. Exercício da Simulação

Os modelos dos algoritmos de deslocamento do braço (elevador e menor seek primeiro) foram implementados utilizando a linguagem de simulação Simscript [SIM 89]. Ambas as implementações diferem apenas no modo como é feito o acesso ao próximo número de cilindro requisitado, de acordo com o respectivo algoritmo em estudo.

Para cada modelo, foram executadas 9 replicações distintas cujos resultados são estudados na seção seguinte. Em cada replicação, foi simulado o atendimento de 5000 requisições de acesso aos cilindros do disco, gerando resultados em termos de tamanho de fila do recurso BRAÇO (braco do disco) e sua utilização.

## 7. Apresentação dos resultados

Para questões de comparação do desempenho entre os dois algoritmos de deslocamento, neste trabalho foi utilizado como parâmetro de referência o tamanho médio da fila. Foi feita a escolha por este parâmetro pelo fato de que uma das formas bastante utilizadas em Sistemas Operacionais para medir o desempenho de algoritmos deste tipo é baseado no número de requisições atendidas [TAN 95].

Nas perspectivas futuras algumas considerações são feitas a respeito da continuidade do trabalho apontando outros parâmetros de avaliação.

A tabela 6.1 apresenta os resultados (tamanho médio da fila) das 9 replicações

executadas utilizando o modelo de representação do algoritmo do elevador, sendo calculado então o tamanho médio da fila entre as 9 replicações ( $E(X)$ ), e um intervalo de confiança de 90%.

X	P(X)	X*P(X)	(X - E(X))	(X - E(X)) <sup>2</sup>	(X - E(X)) <sup>2</sup> * P(X)
0,00132958	0,11111111	0,000147731	2,12956E-05	4,53501E-10	5,0389E-11
0,00084762	0,11111111	0,00009418	-0,000460664	2,12212E-07	2,35791E-08
0,00095374	0,11111111	0,000105971	-0,000354544	1,25702E-07	1,39669E-08
0,00111293	0,11111111	0,000123659	-0,000195354	3,81634E-08	4,24037E-09
0,00125250	0,11111111	0,000139167	-5,57844E-05	3,1119E-09	3,45767E-10
0,00103760	0,11111111	0,000115289	-0,000270684	7,32701E-08	8,14112E-09
0,00165512	0,11111111	0,000183902	0,000346836	1,20295E-07	1,33661E-08
0,00179678	0,11111111	0,000199642	0,000488496	2,38628E-07	2,65142E-08
0,00178869	0,11111111	0,000198743	0,000480406	2,30789E-07	2,56433E-08
$E(X) = 0,001308284$			$Var(X) = 1,15847E-07$		

Tabela 6.1 - Resultados do exercício do modelo do algoritmo do Elevador

### Algoritmo do Elevador:

Tendo  $Var(X) = 1,15847E-07$ , então:

$$\text{Desvio Padrão} = \sqrt{1,15847E-07} = 0,000340363$$

Para um intervalo de confiança de 90%,  $\alpha = 0,1$  e  $R(\text{número de replicações}) = 9$ , tem-se:

$$GL = R - 1 = 9 - 1 = 8 \text{ (Graus de Liberdade).}$$

$$\alpha = 0,1 / 2 = 0,05$$

Utilizando a Tabela dos valores percentis ( $tp$ ) da distribuição  $t$  de Student:

$$\text{Para } \alpha = 0,05 \text{ e } GL = 8, \quad t = 1,86$$

Então:

$$E(X) - 1,86 * 0,000340363 \leq \text{Intervalo} \leq E(X) + 1,86 * 0,000340363$$

$$E(X) - 0,000633076 \leq \text{Intervalo} \leq E(X) + 0,000633076$$

$$0,000675209 \leq \text{Intervalo} \leq 0,00194136$$

A tabela 6.2 apresenta os resultados (tamanho médio da fila) das 9 replicações executadas utilizando o modelo de representação do algoritmo do menor seek primeiro, sendo calculado

então o tamanho médio da fila entre as 9 replicações ( $E(X)$ ), e um intervalo de confiança de 90%.

X	P(X)	X*P(X)	(X - E(X))	(X - E(X)) <sup>2</sup>	(X - E(X)) <sup>2</sup> * P(X)
0,00107974	0,11111111	0,000119971	0,000155904	2,43062E-08	2,70069E-09
0,00087517	0,11111111	9,72411E-05	-4,86656E-05	2,36834E-09	2,63148E-10
0,00085494	0,11111111	9,49933E-05	-6,88956E-05	4,7466E-09	5,274E-10
0,00105578	0,11111111	0,000117309	0,000131944	1,74093E-08	1,93437E-09

0,00094403	0,11111111	0,000104892	2,01944E-05	4,07816E-10	4,53128E-11
0,00073557	0,11111111	0,00008173	-0,000188266	3,54439E-08	3,93821E-09
0,00100161	0,11111111	0,00011129	7,77744E-05	6,04886E-09	6,72096E-10
0,00086902	0,11111111	9,65578E-05	-5,48156E-05	3,00475E-09	3,33861E-10
0,00089866	0,11111111	9,98511E-05	-2,51756E-05	6,33809E-10	7,04232E-11
E(X) = 0,000923836			Var(X) = 1,04855E-08		

Tabela 6.2 - Resultados do exercício do modelo do algoritmo do Menor Seek Primeiro

**Algoritmo do Menor Seek Primeiro:**

Tendo  $\text{Var}(X) = 1,04855E-08$  , então:

$$\text{Desvio Padrão} = \sqrt{1,04855E-08} = 0,000102399$$

Para um intervalo de confiança de 90%,  $\alpha = 0,1$  e  $R(\text{número de replicações}) = 9$ , tem-se:

$$GL = R - 1 = 9 - 1 = 8 \text{ (Graus de Liberdade).}$$

$$\alpha = 0,1 / 2 = 0,05$$

Utilizando a Tabela dos valores percentis ( $t_p$ ) da distribuição  $t$  de Student:

$$\text{Para } \alpha = 0,05 \text{ e } GL = 8, t = 1,86$$

Então:

$$E(X) - 1,86 * 0,000102399 \leq \text{Intervalo} \leq E(X) + 1,86 * 0,000102399$$

$$E(X) - 0,000190462 \leq \text{Intervalo} \leq E(X) + 0,000190462$$

$$0,000733374 \leq \text{Intervalo} \leq 0,001114297$$

Considerando o tamanho médio da fila nas 9 replicações (para 5000 requisições cada) de cada um dos algoritmos de deslocamento simulados nas condições especificadas, pode-se constatar que:

- Considerando  $E(X)$ , o algoritmo do menor seek primeiro mantém a média da fila de requisições em torno 70% do tamanho médio da fila mantida pelo algoritmo do elevador para atender o mesmo número de requisições (objetivo principal);
- A intersecção entre o intervalo de confiança encontrado para ambos os algoritmos (para 90%) é igual ao intervalo de confiança do algoritmo do menor seek primeiro. Isto reforça a afirmação inicial de que, dependendo da ordem em que as requisições são emitidas no que diz respeito ao número do cilindro, o algoritmo do elevador tente a se aproximar do desempenho do outro algoritmo simulado.

do braço de discos rígidos. Para o desenvolvimento deste estudo, foram utilizadas algumas técnicas de modelagem, simulação e avaliação de resultados conhecidas.

Foram estudados o algoritmo do elevador e o algoritmo do menor seek primeiro. Como resultado principal, foi observado o melhor desempenho do segundo algoritmo (em torno de 30%) se comparado ao primeiro, levando em consideração o tamanho médio da fila como parâmetro de avaliação.

Como complemento/continuidade deste estudo, é proposta a realização de novas simulações (exercícios) considerando diferentes situações. Por exemplo, combinando alto/médio/baixo grau de multiprogramação e alto/médio/baixo acesso ao disco. Intercalando com níveis distintos de fragmentação e espaço vazio do disco, entre outros.

**8. Considerações finais**

Neste trabalho foi apresentado um estudo de dois algoritmos utilizados para o deslocamento

**9. Referências bibliográficas**

[BAR 00] Barracuda ATA II Family. **Product Manual.** Publication Number:

20400151-001, Rev. A, February,  
Seagate Technology, 2000.  
Encontrado no site  
[http://www.seagate.com/support/d  
isc/specs/model\\_st3.html](http://www.seagate.com/support/disc/specs/model_st3.html)

- [TAN 95] TANEMBAUM, A.: **Sistemas Operacionais Modernos**. Prentice-Hall do Brasil, Rio de Janeiro, 1995.
- [SIM 89] SIMSCRIPT. **UNIX SIMSCRIPT II.5 - User's Manual**. CACI Products Company, Canadá, 1989.
- [KIV 87] KIVIAT, P. J.; Villanueva, R.; Markowitz, H. M. **SIMSCRIPT II.5 Programming Language**. CACI Products Company, 1987.
- [SIM 85] SIMSCRIPT. **SIMSCRIPT II.5 Reference Handbook**. CACI Products Company, 1985.
- [SPI 85] SPIEGEL, M. R. **Estatística**. 2ª Edição, McGraw-Hill, São Paulo, 1985.