

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Sistemas de Computação

SSC546 – Avaliação de Sistemas Computacionais

Parte 1 - Aula 6

Sarita Mazzini Bruschi

Material baseado nos slides de:
Marcos José Santana
Regina Helena Carlucci Santana

Soluções para o Modelo

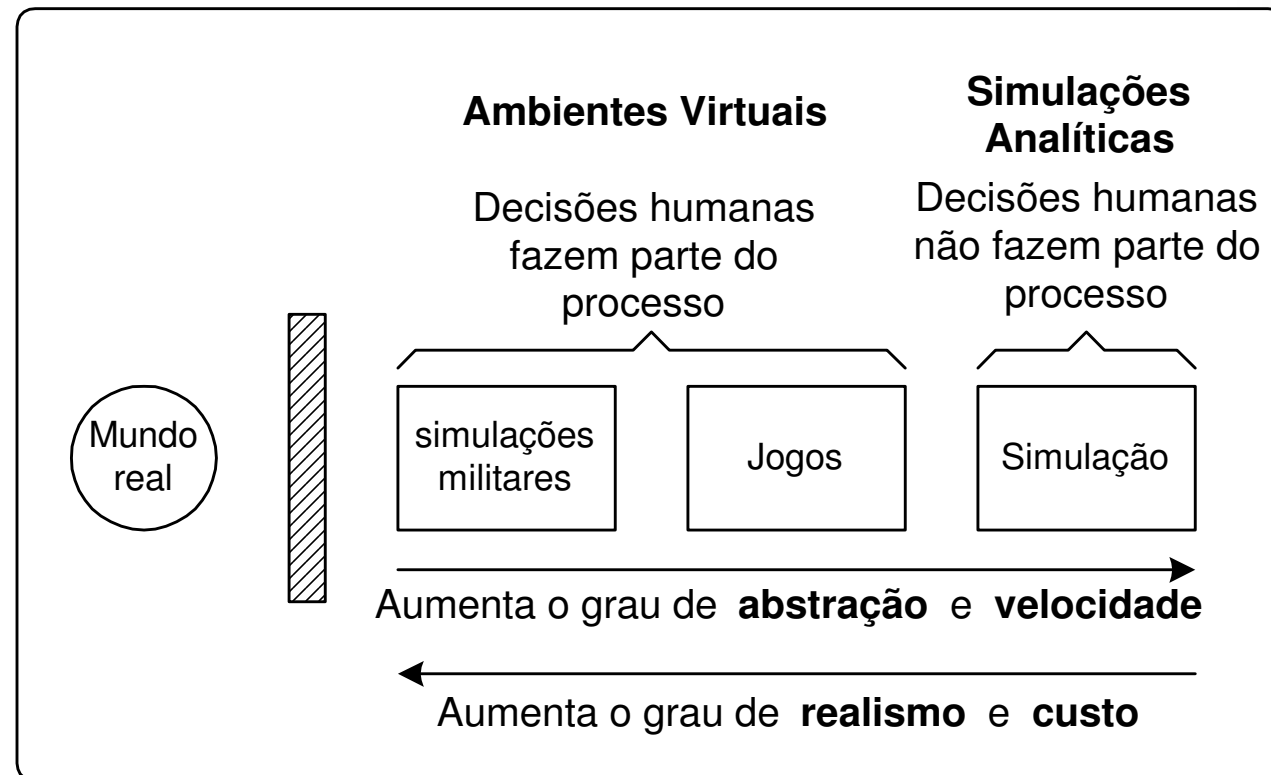
- Solução por Simulação
 - Modelos de simulação são programas de computador nos quais a operação de um sistema e sua carga são descritas utilizando-se algoritmos apropriados.

Solução por Simulação

- Simulação - o que é?
 - construção de um programa computacional que implemente um modelo;
 - o modelo é suposto ser uma representação válida do sistema em estudo.

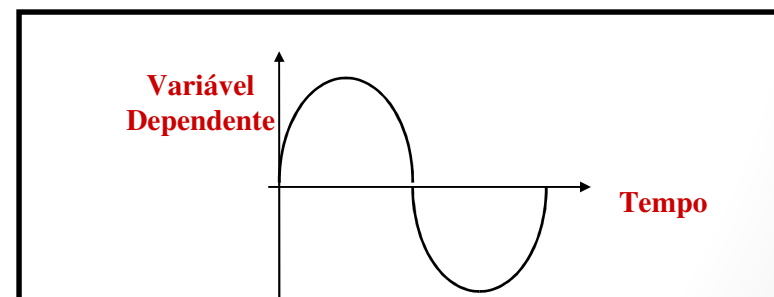
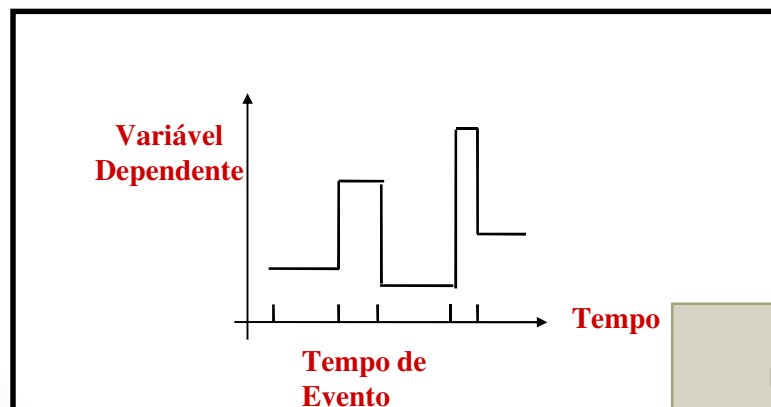
Simulação - Utilização

- Criação de ambientes virtuais
- Avaliação de desempenho de sistemas complexos



Tipos de Simulação

- Discreta X Contínua
 - Discreta - variáveis dependentes variam discretamente em pontos específicos do tempo simulado (tempo de evento)
 - Contínua - variáveis dependentes podem variar continuamente ao longo do tempo simulado

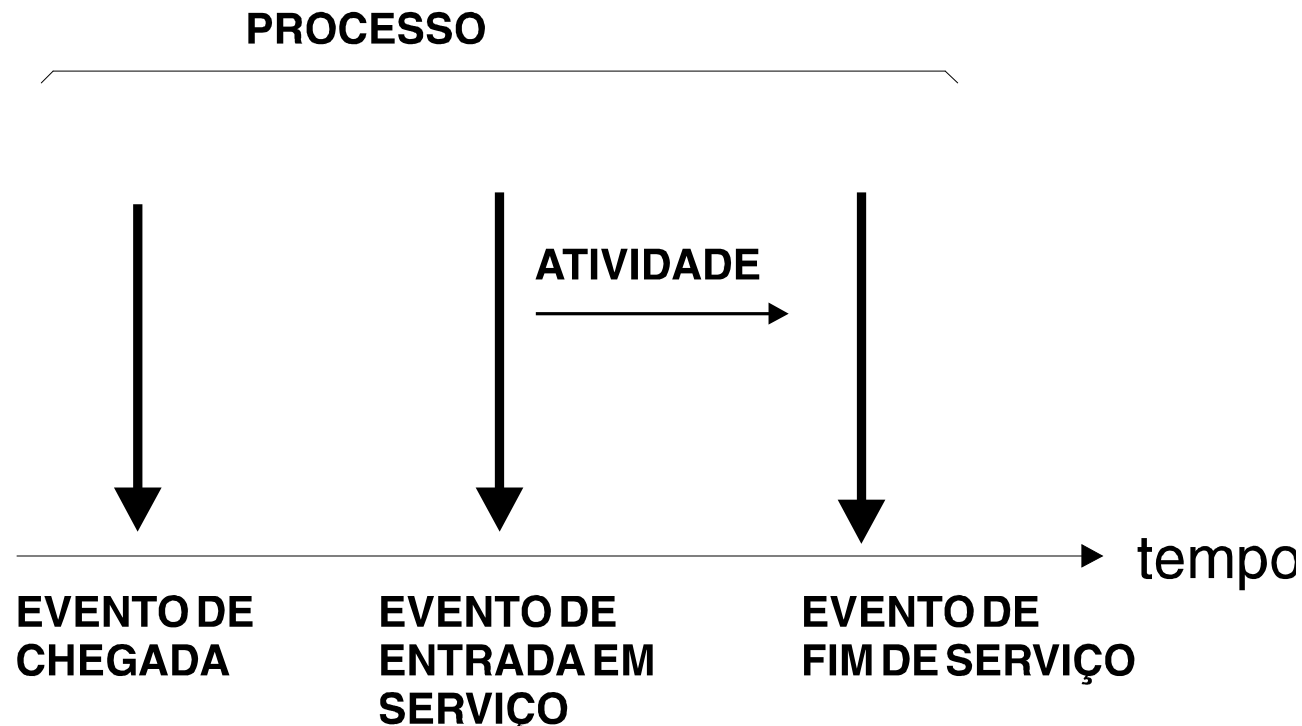


Sistemas Computacionais

Simulação Discreta

Tipos de Simulação

- Processos X Eventos X Atividade



Tipos de Simulação

- **Tipos de Dados de Entrada**

1. Estocásticos

- Dados Probabilísticos
- Amostragem baseada em Distribuições que modelam os dados

2. Determinísticos

- Conjunto fixo de dados
- Ex. Traces

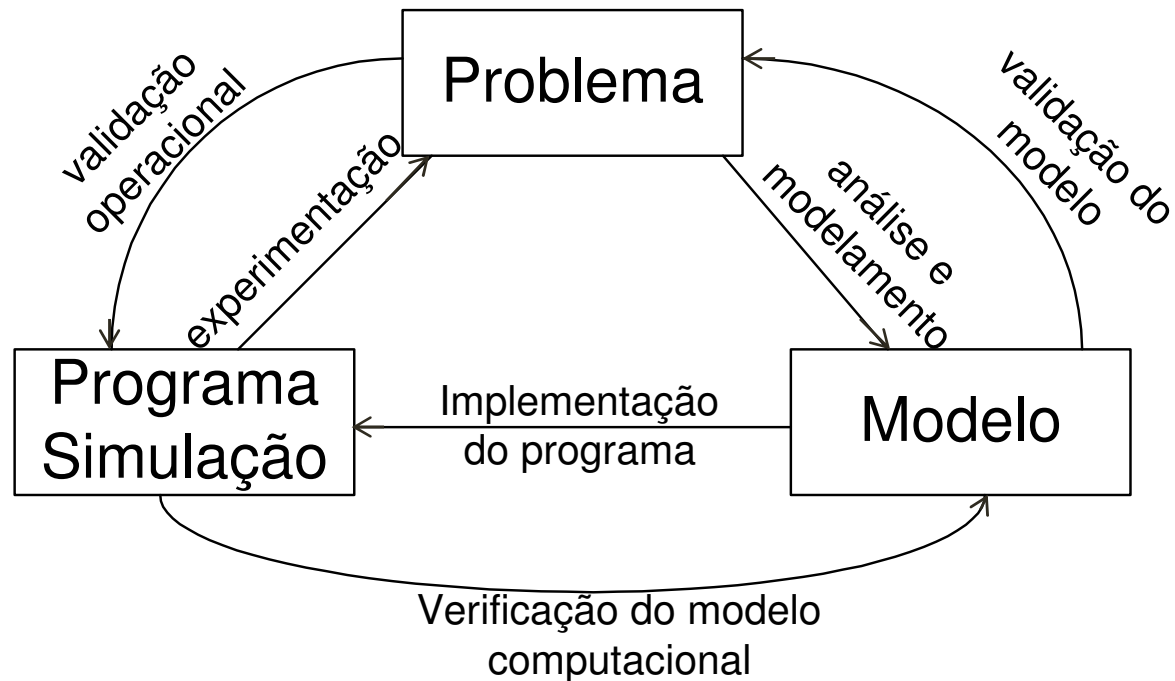
Tipos de Simulação

- Simulações Estocásticas
- Dados de entrada e/ou parâmetros do modelo são probabilísticos
- Custos do Planejamento Estatístico
 - warm-up
 - usar corretamente os números aleatórios
 - cálculo de intervalos de confiança
 - Etc.

Fases de uma Simulação

1. Estudo do sistema e definição dos objetivos;
2. Construção do modelo;
3. Determinação dos dados de entrada e saída;
4. Tradução do modelo;
5. Verificação do programa de simulação;
6. Validação do programa (modelo) de simulação;
7. Experimentação;
8. Análise dos resultados;
9. Documentação.

Solução por Simulação



Fases de uma Simulação

1. estudo do sistema e definição dos objetivos;
2. construção do modelo;
3. determinação dos dados de entrada e saída;
4. **tradução do modelo;**
5. verificação do programa de simulação;
6. validação do modelo de simulação;
7. experimentação;
8. análise dos resultados;
9. documentação.

Desenvolvimento de simulação

- Conhecimentos necessários para o desenvolvimento de uma simulação seqüencial
 - Modelagem
 - Programação / Linguagens para simulação
 - Probabilidade e estatística para análise dos resultados

Tradução do modelo

- **Software para simulação**
 - Linguagens de programação de uso geral
 - Construção do ambiente e do programa
 - Linguagens de simulação
 - Aprendizado de novas linguagens
 - Oferece suporte para a implementação da simulação
 - Exemplos
 - SIMSCRIPT (eventos)
 - GPSS, SIMULA (processos)

Tradução do modelo

- **Software para simulação**
 - Extensões funcionais
 - Uso de linguagens conhecidas
 - Implementa o ambiente
 - Disponibiliza os recursos da linguagem hospedeira
 - Exemplos: SMPL, SIMPACK (eventos)
 - Pacotes de uso específico
 - Voltados a um ou mais domínios de aplicação
 - Pouco flexíveis

Exemplos de Pacotes de Simulação

Software	ARENA	OPNET Modeler	QueGAUSS	PROVISA	WorkFlow Analyzer
Proprietário	Systems Modeling Corporation	MIL 3, Inc.	Aptech Systems, Inc.	AT&T	Meta Software Corp.
Aplicações Típicas	Ferramenta de Simulação geral, molda-se a muitas aplicações diferentes.	Modelagem de protocolos de comunicação de redes.	Sistemas de filas.	Manufatura com capacidade finita de escalonamento	Melhorias de processos de negócios
Plataformas	PC's (DOS) e Estações (UNIX)	DEC-Alpha, HP-UX, IBM, SUN	IBM PC, SUN, IBM RISC.	IBM PC, SUN, HP.	IBM, MAC, SUN.
Preço (U\$) Versão Padrão	Contato com o vendedor	25.000	275	Contato com o vendedor	10.000

Tradução do modelo

- **Software para simulação**
 - Ambientes para Simulação Automáticos
 - Facilitam a elaboração de programas de simulação
 - Pouco conhecimento de simulação ou programação
 - Usuário deve conhecer as técnicas de modelagem
 - Geram o programa automaticamente
 - Oferecem ferramentas que orientam o usuário na descrição, coleta de dados e análise de resultados

Desenvolvimento do Programa

- Depende da Abordagem escolhida (processo, atividade ou evento)
- Eventos
 - Define-se uma lista de eventos futuros (LEF)
 - Todos os eventos a serem executados devem estar nessa lista em ordem de time stamp
 - Dentro de um loop, o próximo evento é retirado da lista, executado e são colocados na lista eventos ativados pela execução

Simulação Orientada a Eventos

- Mudanças de estado no sistema ocorrem devido aos seguintes eventos:
 - Um cliente chega para pagar suas contas
 - Um cliente já foi atendido por um caixa e parte do sistema.

Simulação Orientada a Eventos

- **Evento Chegada de Cliente**

Escalone a próxima chegada

Se o caixa está ocupado então

Aumente o número de clientes na fila

Senão

Torne o estado do caixa ocupado

Escalone o fim de atendimento para $\text{tempo_corrente} + \text{tempo_atendimento}$

Simulação Orientada a Eventos

- **Evento Término de Atendimento**

Se existe algum cliente na fila então

Retire um cliente da fila para atendimento

*Escalone o fim de atendimento para tempo_corrente
+tempo_atendimento*

Senão

Torne o estado do caixa desocupado

Simulação Orientada a Eventos

- Escalone a primeira chegada

Selecione o primeiro evento

Avance o tempo de simulação para o tempo de ocorrência do evento

Se final de simulação então

Imprima estatísticas

Senão

Se evento = Chegada de Cliente então

Trata chegada de Cliente

Senão

Trata término de atendimento

Simulação - Exemplo

- Controle de tráfego aéreo
- Simulação orientada a eventos:

chegada do avião

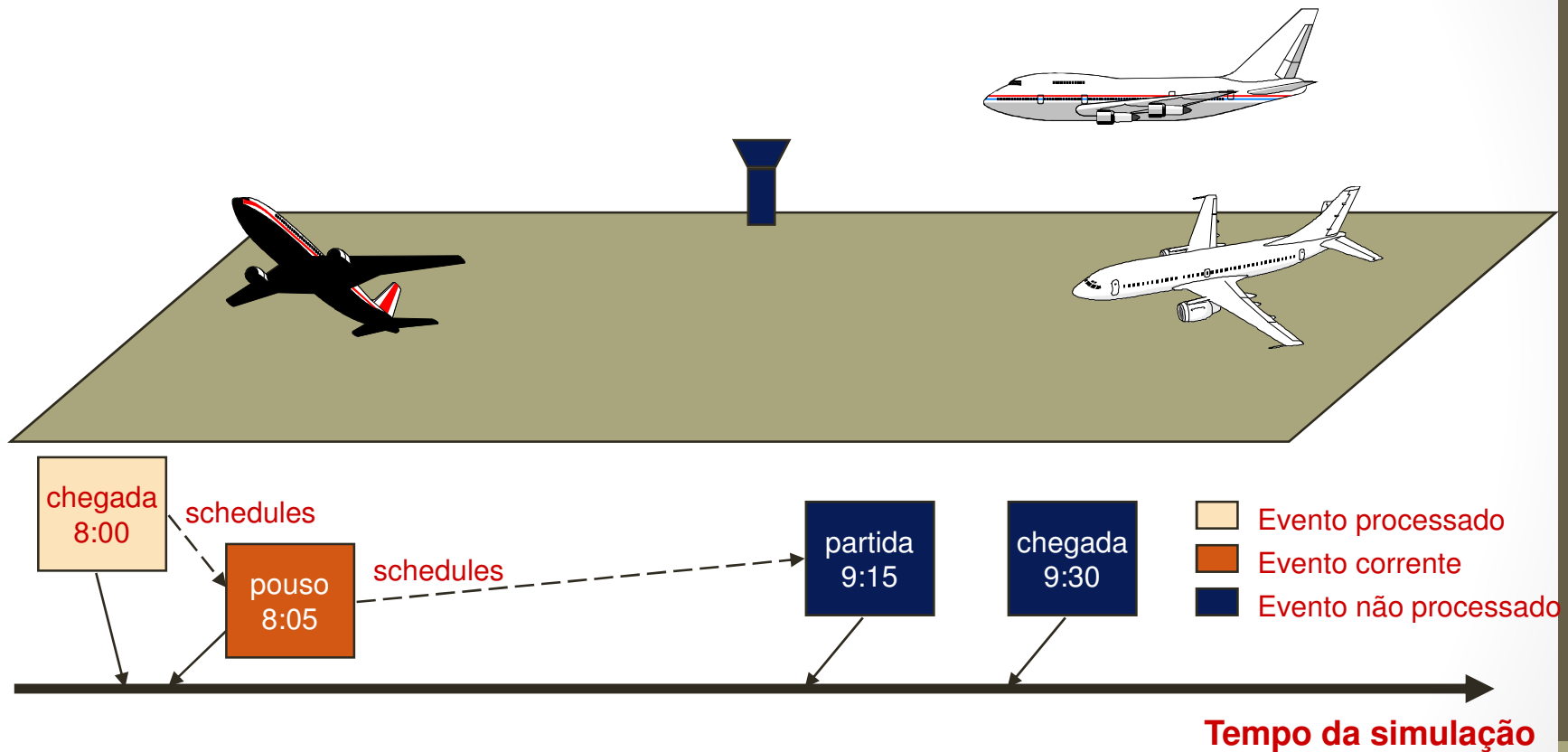
escalona

pouso

escalona

partida

Simulação - Exemplo



Eventos não processados são armazenados em uma lista de eventos futuros (LEF)
Eventos são processados seguindo a ordem de time stamp

Simulação - Exemplo

Modelo do
sistema físico

Aplicação

- variáveis de estado
- código modelando o comportamento do sistema
- entrada e saída (interface com o usuário)

Chamadas
para escalonamento
dos eventos

Chamadas para
gerenciamento dos
eventos

Independente da
aplicação

Simulação

- Gerencia a lista de eventos
- Avança o tempo da simulação

Procedimentos para gerenciamento dos eventos

Variáveis de estado

```
Integer: NoAr;  
Integer: NaTerra;
```

```
Evento Chegada  Evento Pouso  Evento Partida  
{               {             {  
    ...          ...           ...  
}
```

Aplicação

Simulação

Laço de processamento dos eventos

Agora = 8:45

Lista de Eventos Futuros (LEF)

9:00

9:16

10:10

Enquanto (simulação não é encerrada)

E = evento com menor time stamp na LEF

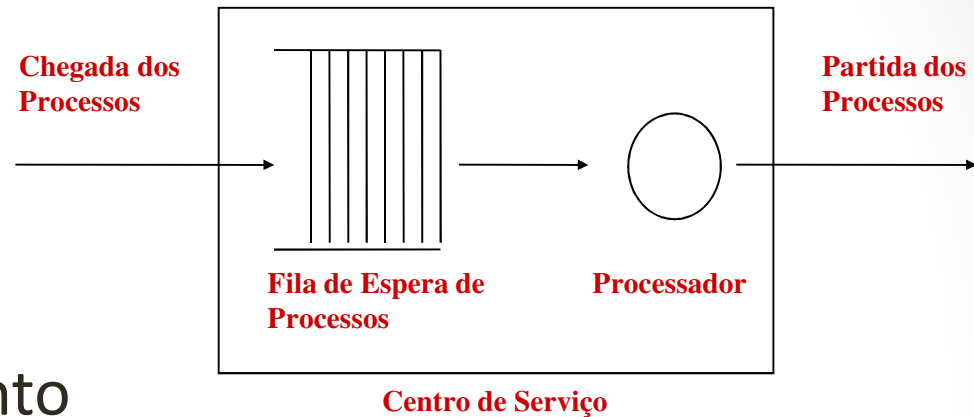
Retire E da LEF

Agora = time stamp de E

chamada ao manipulador do procedimento

Simulação - Exemplo

- Fila M/M/1
- Eventos:
 1. Chegada
 2. Início atendimento
 3. Final do atendimento
- Programa em SMPL



```

/* ----- */
#include <dir.h>
#include <stdio.h>
#include <conio.h>
#include "smpl.h"
#include "rand.h"

main()
{
    /* definicoes */
    float Te = 10000;
    int Event = 1, Customer = 1;
    real Ta1 = 10, Ts1 = 10;
    int Server1;
    FILE *p, *saida;
    saida = fopen("saida.out","w");
    if ((p = sendto(saida)) == NULL)
        printf("Erro na saida\n");

    /* prepara o sistema de simulacao e da nome ao modelo */
    smpl(0," Exemplo M/M/1");

    /* cria e da nome as facilidades */
    Server1 = facility("Servidor1",1);

    /* escalona a chegada do primeiro cliente */
    schedule(1,0, Customer);

```

```

while ( (time() < Te) )
{
    cause(&Event,&Customer);
    switch(Event)
    {
        case 1:
            schedule(2,0.0,Customer);
            schedule(1,expntl(Ta1),Customer);
            break;
        case 2:
            if (request(Server1,Customer,0) == 0)
                schedule(3,expntl(Ts1),Customer);
            break;
        case 3:
            release(Server1, Customer);
            break;
    }
}

/* gera o relatorio da simulacao */
report();
fclose(saida);
}
/* ----- */

```

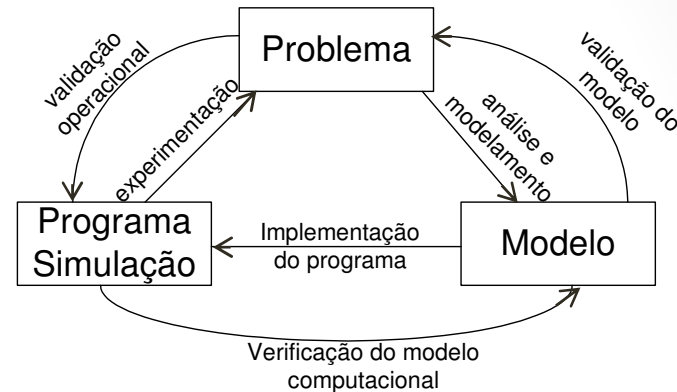
Fases de uma Simulação

1. estudo do sistema e definição dos objetivos;
2. construção do modelo;
3. determinação dos dados de entrada e saída;
4. tradução do modelo;
5. verificação do programa de simulação;
6. validação do modelo de simulação;
7. experimentação;
8. análise dos resultados;
9. documentação.

Validação de uma Simulação

- Uma das fases mais difíceis
- Nunca pode-se dizer que um modelo é totalmente válido
 - Válido para condições particulares
- **Obtêm-se uma confiança no modelo desenvolvido para este conjunto de condições**

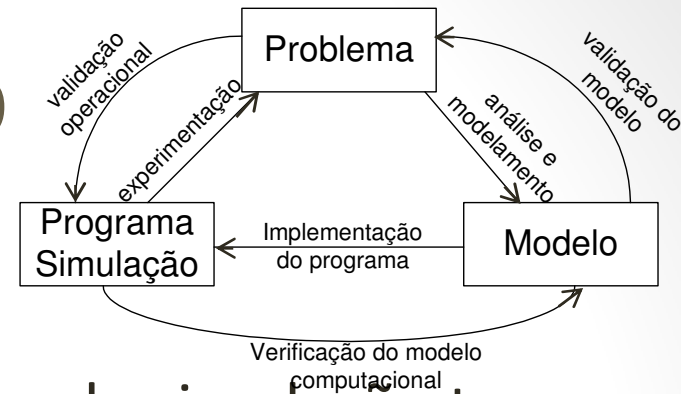
Verificação e validação



- Validação:
 - garante que as suposições feitas para delinear o modelo são válidas
- Verificação do modelo computacional:
 - o programa deve representar todas as características do modelo de uma forma correta

Fases da Validação

- Validade operacional:
 - verifica se a saída do programa de simulação tem a precisão necessária para a aplicação e nos domínios desejados
- Validação dos dados:
 - deve-se assegurar que os dados necessários para a construção do modelo, sua avaliação e testes são precisos
 - Aquisição de dados apropriada
 - Análise cuidadosa



Técnicas para Verificação e Validação

- Comparação com outros modelos
 - Resultados da simulação sendo validados são comparados com resultados de outras simulações já validadas.
- Teste de degeneração
 - Partes do modelo são retiradas ou os parâmetros de entrada são selecionados apropriadamente.
 - Ex.: Tamanho da fila aumenta quando o tempo entre chegadas diminui

Técnicas para Verificação e Validação

- Teste de condições extremas
 - As saídas da simulação devem ser compatíveis com condições extrema
 - Ex.: Se número de clientes é zero -> tempo de utilização e fila = 0
- Validade aparente
 - Pessoas que têm um bom conhecimento do sistema sendo simulado analisam os resultados.
 - Ex.: verifica se a relação entre entrada e saída são coerentes

Técnicas para Verificação e Validação

- Valores fixos
 - Valores fixos permitem a comparação dos resultados da simulação com resultados calculados à mão
- Validação através de dados existentes
 - Se dados sobre o sistema existem, parte deste dados pode ser utilizados na construção do sistema e parte para a validação

Técnicas para Verificação e Validação

- Validação interna
 - A simulação é executada várias vezes. Se resultados variam muito -> desconfiança (falta de consistência)
- Gráficos da operação
 - O comportamento do modelo é mostrado conforme o tempo passa. Consistência pode ser verificada a cada passo

Técnicas para Verificação e Validação

- Teste de comparação
 - Pessoas que conhecem bem a operação do sistema simulado são questionadas qual é a saída real e qual é a saída do modelo.
- Variabilidade de parâmetros
 - Mudanças equivalentes no modelo e no sistema real devem predizer diferenças no resultados equivalentes
- Precisão
 - Resultados obtidos na simulação e no sistema real são comparados

Fases de uma Simulação

1. estudo do sistema e definição dos objetivos;
2. construção do modelo;
3. determinação dos dados de entrada e saída;
4. tradução do modelo;
5. verificação do programa de simulação;
6. validação do modelo de simulação;
7. **experimentação;**
8. **análise dos resultados;**
9. documentação.

**Itens já vistos nas
primeiras aulas**