

Visualização Científica em Simulações de Sistemas de Tempo Real: Fundamentos e Apresentação de uma Ferramenta.

ALAN SALVANY FELINTO (*)
DRA TEREZA GONÇALVES KIRNER (**)

UFSCar—Universidade Federal de São Carlos, Departamento de Computação – PPG-CC
Caixa Postal 676, 13560-970 São Carlos, SP, Brasil, Telefone (0162) 748233
UFSCARC@BRFAPESP.BITNET

Abstract. This paper discusses basic concepts of scientific visualization, presenting a software tool which supports the representation and handling of graphic objects, in real-time systems simulations.

(*) Bacharel em Ciência da computação; Mestrando do Programa de Pós Graduação em Ciência da computação da UFSCar; professor da UEL (Universidade Estadual de Londrina - Paraná) e do IMESA (Instituto Municipal de Ensino Superior de Assis - São Paulo).

(**) Professora Adjunta; Coordenadora do curso de Pós Graduação em Ciência da Computação da UFSCar.

1. Introdução

Nos últimos anos, a necessidade de abordagens e ferramentas para o desenvolvimento de Sistemas de Tempo Real (STR) tem recebido atenção considerável, devido à disseminação de aplicações críticas e complexas, em diversas áreas do conhecimento.

Em linhas gerais, um STR é um sistema que precisa reagir prontamente a estímulos do meio ambiente, dentro de intervalos de tempo e sob restrições temporais precisas e previsíveis. Assim, o desenvolvimento desses sistemas deve garantir não apenas uma solução computacional correta mas também um atendimento preciso das restrições de temporização inerentes à aplicação [WOOD (1990)].

Para garantir a criação de STR confiáveis, corretos e que cumpram as suas restrições de tempo, é necessário o uso de técnicas potentes de análise, dentre as quais se destaca a simulação.

Tradicionalmente, as simulações produzem resultados que retratam o comportamento do sistema através de formas convencionais, do tipo tabelas e gráficos contendo valores de variáveis. O emprego da visualização científica permite a reprodução destes resultados, mostrando-os de forma clara, através de recursos gráficos e de animação.

Esta comunicação é resultado de estudos sobre a aplicabilidade dos conceitos e técnicas de visualização científica em simulações de STR. Com base nesses estudos, foi gerada uma ferramenta que auxilia os projetistas de software (e usuários envolvidos no projeto) a testarem e verificarem graficamente os resultados dos sistemas em desenvolvimento.

2. Fundamentos Conceituais

2.1. Visualização Científica

Conforme discussões apresentadas por McCormick [McCORMICK (1987)] e Carlbom [CARLBOM (1992)], o termo visualização científica refere-se à especificação de transformações, teorias ou modelos em representações visuais compatíveis com a realidade enfocada. Essas representações visuais incluem, especialmente, formas gráficas animadas, organizadas de maneira a expressar informações importantes para a compreensão do objeto ou problema em estudo.

Portanto, a visualização científica pode retratar tanto resultados de dados agregados, quanto o processamento lógico de uma solução, como também o comportamento dinâmico de processos físicos, monitorados através de programas ou ferramentas de "software".

Exemplos de aplicações da visualização em sistemas de monitoração e análise de performance, são descritos em "Performance Instrumentation and Visualization" [KIMELMAN (1990)]. Entre elas, podem ser destacadas aplicações nas áreas de medicina, química, engenharia, etc.

A visualização científica configura uma área de pesquisa inter disciplinar, que compreende importantes campos de pesquisa. Seus principais temas (de acordo com [CARLBOM (1992)], [HEGRON (1989)] [McCORMICK (1987)]) são: Computação Gráfica, Processamento de Imagem, Visão por Computador e Estudo de Interface do Usuário.

A principal contribuição prática da visualização científica é prover meios mais potentes para a compreensão de dados e resultados de pesquisa, utilizando para isso métodos de interpretação e representação visual.

A computação gráfica tem se preocupado, tradicionalmente, com a síntese de desenhos e imagens, obtidas a partir de um conjunto de especificações fornecidas pelo usuário. É um campo que tem se expandido bastante, incluindo atualmente, a criação, armazenamento e manipulação de modelos e imagens de objetos. Esses modelos são geralmente geométricos, mas podem também englobar modelos matemáticos para descrever a síntese do objeto ([CARLBOM (1992)], [BURGER (1989)]).

Segundo CARL(1992), o processamento de imagens enfoca a análise de dados em duas dimensões, obtidos pela digitalização das imagens do mundo real. Utiliza-se de transformações do tipo imagem para imagem, onde a nova imagem é uma versão modificada da imagem original. A transformação para uma nova imagem pode ter a finalidade de reduzir ruídos (de forma a melhorar a imagem como um todo), ou de aumentar a visibilidade de bordas ou regiões (técnica bastante útil para a detecção ou reconhecimento de objetos contidos em uma imagem).

Em contraste com o processamento de imagem, a visão por computador procura emular a interpretação humana através de uma interpretação feita via computador. Esta área lida com a análise automática, reconstrução, e reconhecimento de objetos em um cenário a partir de uma ou mais imagens. Neste sentido, a visão por computador é o inverso da computação gráfica, isto é, visa encontrar as principais características dos objetos a partir de uma imagem, em oposição à síntese de imagens que, a partir da descrição das características do objeto, cria a imagem [CARLBOM (1992)] [GONZALES (1987)].

Para o desenvolvimento de uma visualização, deve-se ter muito cuidado com a parte da representação gráfica do sistema. Isto requer, além dos conhecimentos básicos de computação gráfica, processamento de imagem, e visão em computador, o conhecimento da ergonomia de software, que conduz à criação de interfaces potentes e compatíveis com os usuários e com o sistema que está sendo desenvolvido, facilitando a interação homem-máquina [PLASTOCK (1991)] [SANTOS (1989)].

2.2. Sistemas de Tempo Real

Um sistema de tempo real pode ser visto como um conjunto de processos concorrentes, que interagem entre si, funcionando sob restrições de temporização, que precisam ser atendidas para que o sistema cumpra os seus objetivos.

De acordo com Koymans [KOYMANS (1988)], existem dois componentes essenciais em um STR: O ambiente e o sistema computacional.

O ambiente compõe-se: de uma parte física, representada pelos equipamentos que serão controlados pelo sistema computacional; e de uma interface composta por todos os pontos físicos (sensores, atuadores e o meio por onde os dados irão trafegar) que executam a troca de informações entre o equipamento e o sistema computacional.

O sistema computacional é um hardware, com um software embutido, que tem a função de monitorar e controlar a execução dos processos do ambiente do STR.

O relacionamento entre o sistema computacional e o ambiente do STR é feito por eventos e variáveis de estados do ambiente que estimulam uma resposta do sistema, e vice-versa.

Um esquema simplificado de um STR é mostrado na figura 1.

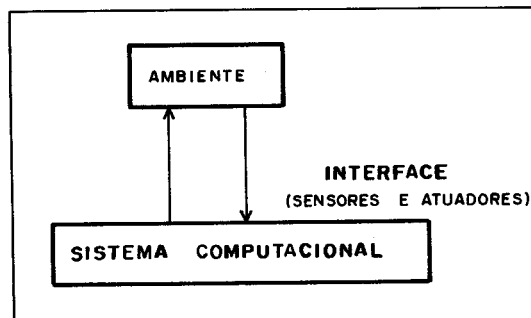


FIGURA 1: SISTEMA DE TEMPO REAL

2.3. Simulação de Sistemas de Tempo Real

A simulação pode ser genericamente definida como um método para investigar as propriedades e o comportamento de um sistema real (e/ou físico), através da construção de um segundo sistema que é um modelo do sistema original [SCHUTZ (1990)]. São feitas então experimentações com o modelo criado, e os resultados podem ser aplicados ao sistema original.

Para STR, a simulação constitui-se em uma das técnicas mais potentes, disponíveis para validar e monitorar o comportamento e os resultados parciais e finais desejados. O emprego da visualização científica permite a reprodução destes resultados, mostrando-os de forma clara, através de recursos gráficos e de animação [McCORMICK (1987)].

3. Uma Ferramenta para a Visualização de Sistemas de Tempo Real

Como resultado de estudos efetuados sobre a aplicabilidade dos conceitos e técnicas de visualizações de STR, foi construída uma ferramenta, cujo

objetivo é auxiliar projetistas (e/ou usuários) a monitorarem e testarem o comportamento e os resultados de aplicações que estão sendo desenvolvidas.

Os sub-ítems a seguir fornecem um resumo da ferramenta, sendo que sua descrição completa, juntamente com a discussão do tema deste comunicado, são encontradas em Felinto [FELINTO (1992)].

3.1. Estrutura da Ferramenta

A ferramenta implementada é aqui chamada de SVSSTR, que significa Suporte para a Visualização de Simulações de Sistemas de Tempo Real. Seus componentes básicos são: dois módulos principais, o módulo gerencial e o de criação; duas bibliotecas, a de sistemas e a de recursos gráficos; e um módulo de interface com a aplicação do usuário. Estes componentes são ilustrados na figura 2, e descritos a seguir.

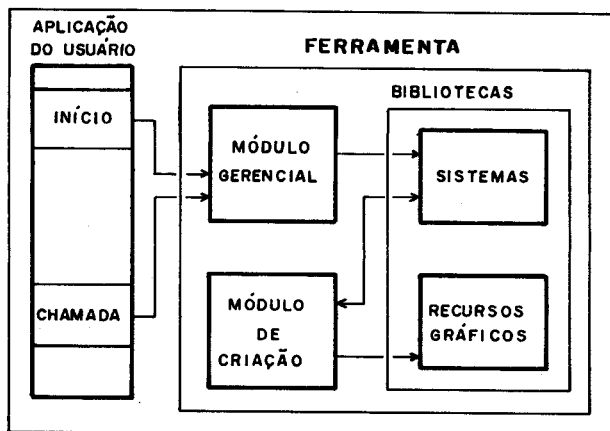


FIGURA 2: ARQUITETURA DO SVSSTR

A) Módulo Gerencial

Conforme definido no item 2.2, um STR possui dois componentes principais, que são o Ambiente e o Controle Computacional. Assim sendo, para esta ferramenta, a aplicação do usuário é responsável pelo controle computacional, e o SVSSTR faz a simulação e a demonstração gráfica do ambiente.

O módulo gerencial do SVSSTR fica responsável pelo controle do ambiente do STR simulado.

Na implementação efetuada, o controle da animação dos objetos gráficos baseia-se no conceito de estímulo e resposta, sendo que a comunicação entre a aplicação do usuário e o ambiente é realizada através dos parâmetros das chamadas dos processos.

Em outras palavras, o módulo gerencial recebe chamadas da aplicação do usuário, as quais "estimulam" os objetos. Esses objetos, após serem "estimulados", processarão as mudanças no seu comportamento, para então enviar a resposta, através do módulo gerencial, para a aplicação. A aplicação processará essa resposta, continuando a sua execução,

podendo ser feitas quantas chamadas forem necessárias.

O controle do tempo foi implementado com o apoio de fórmulas, que calculam o tempo-real que os objetos do mundo real gastam para completarem as suas tarefas. Na simulação do ambiente, o tempo gasto para a execução da tarefa de um objeto, é proporcional ao tempo-real obtido pelas fórmulas, tendo como base o tempo marcado pelo relógio interno da Estação de Trabalho utilizada.

B) Módulo de Criação do Sistema Gráfico

Este módulo tem a função de preparar o ambiente do sistema de tempo-real considerado (ver figura 3).

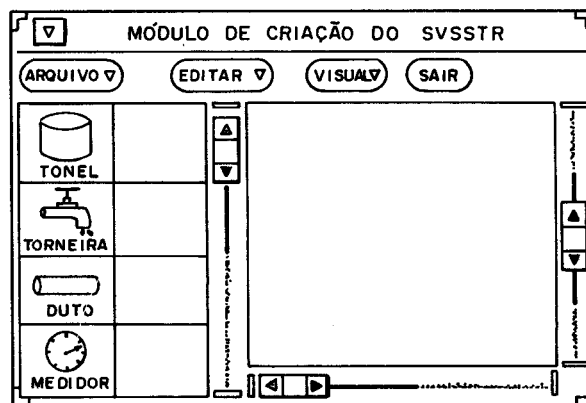


FIGURA 3: MÓDULO DE CRIAÇÃO

Para isso, é acessado uma biblioteca de recursos, que contém todos os componentes do sistema a ser simulado.

Cada componente é representado por um objeto, com características próprias de comportamento e de apresentação gráfica.

O sistema gráfico gerado por este módulo pode ser armazenado na biblioteca de sistemas, para ser utilizado posteriormente pelo módulo gerencial.

Durante a construção gráfica do ambiente do STR, o módulo de criação faz a conexão entre os objetos especificados pelo usuário. Esta conexão requer a determinação das possíveis comunicações entre os objetos do sistema, ou seja, são determinados quais são os possíveis objetos com os quais um objeto pode se comunicar. Nesta etapa, o usuário especifica o estado inicial padrão para o sistema, que, no entanto, poderá modificar-se ao longo da execução da aplicação do usuário.

C) Bibliotecas

Existem 2 bibliotecas no SVSSTR: a biblioteca de recursos e a biblioteca de sistemas.

A biblioteca de recursos contém os objetos gráficos pré-definidos, necessários para a construção dos

sistemas gráficos. São incluídos também módulos que permitem a representação do comportamento do ambiente do STR criado (como manipulação de cores, recursos para animação, controle de mouse, etc.).

A biblioteca de sistemas armazena os objetos e suas inter-conexões, referentes ao sistema que está sendo simulado. A partir de uma chamada de início (ver figura 2), o módulo gerencial acessa a biblioteca de sistemas e lê o sistema desejado.

D) Interface Entre a Aplicação e o SVSSTR

A interface entre a aplicação e o SVSSTR é realizada por meio dos parâmetros indicados nas chamadas de processos.

Assim, a chamada da aplicação ao SVSSTR, é descrita pela especificação do objeto gráfico de interesse, associado aos seguintes parâmetros: identificação do objeto; tipo da informação que se deseja manipular; ação desejada; e uma variável para a atualização ou leitura da informação resultante da ação. O padrão empregado, segue o mesmo padrão de primitivas usadas pelo sistema X-Window.

3.2 Implementação da Ferramenta

Na implementação do SVSSTR, foram utilizados o Sistema Operacional UNIX, a linguagem de programação C, o Sistema XWindow e o "toolkit" Xview, que facilitou a criação e gerenciamento de janelas e objetos gráficos [HELLER (1990)]. O protótipo atual foi construído em Estações de Trabalho SUN - Sparc 2 e incorporou recursos de manipulação de gráficos, funcionamento interativo baseado em menus e janelas, facilidades para movimentação e variação do tamanho da tela de apresentação, etc.

4. Conclusão

Nesta comunicação, foram resumidos os fundamentos teóricos relativos ao emprego da Visualização Científica em Simulações de STR, procurando-se enfatizar a relevância de simulações gráficas e animadas para o monitoramento e validação comportamental de STR.

Foi descrita também a ferramenta SVSSTR (Suporte de Visualização Científica para Simulações de Sistemas de Tempo Real), desenvolvida para auxiliar projetistas e usuários na geração de STR corretos e viáveis.

Como a Visualização Científica é uma área ainda emergente e com poucos estudos e resultados práticos, este trabalho pode prestar uma contribuição significativa aos pesquisadores, projetistas de software e profissionais envolvidos com o desenvolvimento e simulação (ou teste) de STR.

Bibliografia

- BURGER, P.; GILLIES, D. *Interactive Computer Graphics: Functional, Procedural and Device-Level Methods*. Great Britain: Addison-Wesley Publishing Company, 1989.504 p.
- CARLBOM, Ingrid; CHAKRAVARTY, Indranil; HSU, William M. Integrating Computer Graphics, Computer vision, and Image Processing in Scientific Applications. *Computer Graphics*, v.26, n.1, p.8-16, janeiro 1992.
- FELINTO, Alan S. *Suporte para Visualização de dados Científicos Obtidos a Partir da Simulação de Sistemas de Tempo Real*. São Carlos - SP. UFS-Car, Setembro 1992. Proposta de Dissertação de Mestrado (Exame de Qualificação).
- GONZALES, R. C.; WINTZ, P. *Digital Image Processing*. 2. ed. U.S.A: Addison-Wesley Publishing Company, novembro de 1987. 503 p.
- HEGRON, Gerard; PALAMIDESE, Patrizia; THALMANN, Daniel. Motion Control in Animation, Simulation and Visualization. *Computer Graphics forum*, v.8, n.4, p.347-352, dezembro, 1989.
- HELLER, D. *XView Programming Manual*. O'Reilly Associates, Inc., CA, USA, 1990.
- KIMELMAN, Doug. "Environments for Visualization of Program Execution". in: SIMMORES, Margarith; KOSKELA, Rebecca. *Performance Instrumentation and Visualization*. New York, ACM Press, 1990, p.135-146.
- KOYMANS, Ron; KUIPER, Ruurd. Paradigms for Real-Time Systems. *Lecture Notes in Computer Science* - 331, p.159-174, setembro, 1988.
- McCORMICK, B.H.; DEFANT, T.A.; BROWN, M. D. Visualization in Scientific Computing. *Computer Graphics*, v.21, n.6, p.1-14, novembro, 1987.
- PLASTOCK, Roy A. *Aplicações da Computação Gráfica*. in: *Computação Gráfica*. McGraw-Hill, 1991, p. 274-317.
- SANTOS, A. C.. An Overview of Human Factors in Computer Systems (chapter 3) - In: *The Design of Information Systems: A New Methodological Approach*, tese de doutorado, London Business School, 1989, pp. 70-92.
- SCHUTZ, W. *Real-Time Simulation in quad the Distributed Real-time System MARS*. Institut fur Technische Informatik, Technische Universität Wien, Austria, Research Report, junho, 1990.
- WOOD, Mike; BARRETT, Tom. A real-Time Primer. *Embedded Systems Programming*. p. 20-38, fevereiro, 1990.