

DESENVOLVIMENTO DE UMA FERRAMENTA DE TESTES DE SOFTWARE EM UM PROGRAMA DE PARCERIA ESCOLA-EMPRESA

André Schinzel Braga¹

Resumo: Este artigo apresenta o resultado de um trabalho de pesquisa com o objetivo de desenvolver uma ferramenta de testes para sistemas em tempo real. A ferramenta (chamada TLM – Time Line Monitor) foi desenvolvida em paralelo com um projeto conjunto entre o CEFET-PR e uma empresa do setor de telecomunicações. Graças a esta parceria, foi possível avaliar a ferramenta em um ambiente industrial e empregar no projeto técnicas pouco utilizadas fora do meio acadêmico.

Palavras-Chave: Sistemas em tempo real, Sistemas dedicados, Testes de Software

Abstract: This paper presents the results of a research with the objective to develop a tool for testing real-time systems. The tool (called TLM – Time Line Monitor) has been developed in parallel with a project conducted by CEFET-PR and a telecommunications company. With this partnership, it was possible to evaluate the tool on a industrial environment, and use during the project some techniques not used outside the academic environment.

Keywords: Real-time systems, Embedded systems, Software tests

1. Introdução

O Centro Federal de Educação Tecnológica do Paraná desenvolve um programa de cooperação com uma série de indústrias do Estado, visando a uma maior interação entre os meios acadêmico e industrial. No decorrer dos últimos quatro anos, o Laboratório de Engenharia de Software (LES), do Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial, desenvolveu projetos em parceria com uma indústria do setor de telecomunicações, visando a aprimorar equipamentos até então produzidos.

Estes projetos envolvem o desenvolvimento de hardware e software por uma equipe de professores, alunos de graduação e pós-graduação, segundo exigências comuns à área industrial, tais como custos, níveis de qualidade e

.....
¹ CPGEI-CEFET/PR – asbraga@computer.org

cronogramas. Por outro lado, permitem a aplicação e avaliação de técnicas e tecnologias ainda não difundidas no meio industrial e restritas ao meio acadêmico.

O projeto de um equipamento complexo é uma tarefa propensa a erros, os quais devem ser identificados e corrigidos com rapidez. Ao final do desenvolvimento, o equipamento deve ser rigorosamente testado visando a localizar falhas no seu funcionamento, evitando-se assim comercializar produtos de baixa qualidade. Um fato bem conhecido em engenharia de software é que a maioria dos defeitos de um programa está localizada em funções que raramente são executadas. Portanto, para comprovar que um equipamento é confiável, é preciso garantir que todas as partes do seu programa foram testadas.

Conhecendo a importância dos testes no desenvolvimento de um equipamento, iniciou-se uma pesquisa para desenvolver uma ferramenta de apoio a essa atividade, com ênfase especial no teste de restrições temporais de sistemas computacionais, ou seja, procurando-se determinar tempos de execução de programas. Essa pesquisa foi realizada como trabalho de mestrado em paralelo ao desenvolvimento dos projetos industriais no LES [Braga 99]. Este artigo apresenta a interação entre ambos os projetos: o ambiente industrial avaliando a ferramenta de testes segundo critérios reais e o ambiente acadêmico fornecendo as técnicas de testes mais adequados para o equipamento em questão.

A ferramenta desenvolvida chama-se TLM (*Time Line Monitor*), é composta por um programa e um equipamento de hardware, e foi empregada com êxito durante os testes e depuração dos projetos desenvolvidos no LES.

2. Conceitos básicos

Na atividade de testes de um equipamento, há dois aspectos que devem ser considerados:

O aspecto lógico, ou seja, verificar se o sistema executa corretamente todas as funções exigidas. Por exemplo: um equipamento de segurança para controlar uma máquina deve desligá-la ao identificar situações de perigo, tais como: aproximação de pessoas, aumento de temperatura ou defeito mecânico. Para funcionar corretamente, o equipamento não deve desligar a máquina se não houver perigo, nem mantê-la funcionando em caso de risco iminente.

O aspecto temporal, ou seja, avaliar se o sistema executa as funções exigidas dentro do prazo estipulado. No exemplo anterior, o equipamento será totalmente inútil se desligar a máquina muito tempo após a situação de perigo ser identificada, pois poderá ocasionar desastres.

Um sistema computacional que possui restrições de tempo em sua especificação é chamado Sistema em Tempo Real (STR). Em um STR, o comportamento temporal é tão importante quanto o comportamento lógico. Atualmente, os STR têm aplicações em diversas áreas, entre elas: automação industrial, telecomunicações, engenharia biomédica, controle de embarcações e aeronaves, telemetria e eletrônica de consumo. Dependendo das áreas de aplicação, alguns STR são considerados de segurança crítica, pois falhas no seu funcionamento podem conduzir a graves perdas humanas, econômicas ou ambientais.

Nesses casos, é fundamental garantir que o sistema atende às suas especificações funcionais e temporais. Por esse motivo, durante o desenvolvimento de software para STR deve-se necessariamente incluir etapas de testes das restrições temporais. Daí a importância da medição de tempos de execução.

Muitos STR enquadram-se também na definição de Sistema Dedicado, sistema computacional que faz parte de um equipamento cuja finalidade direta não é a computação (processamento de dados) [Manley 94], [Douglass 97]. Este é o caso de muitos equipamentos das áreas referidas acima, em que o usuário não sabe ou não precisa saber que está operando um computador.

Sistemas dedicados possuem algumas características próprias que exigem técnicas especiais em seu desenvolvimento. Em geral, por razões de custo, os recursos computacionais (poder de processamento e memória) são escassos, consumidos quase totalmente pelas tarefas que executam. A interação com o ambiente é forte, utilizando dispositivos de hardware como sensores, conversores e atuadores; por outro lado, a interação com o usuário é fraca, raramente possuindo teclado, monitor de vídeo e outros recursos deste tipo. É comum que o equipamento funcione continuamente por longos períodos de tempo e em ambiente hostil, sujeito a choques, vibrações, umidade e descargas eletromagnéticas.

3. Técnicas de testes

Devido à importância dos testes no desenvolvimento de um sistema, é fundamental que estes sejam realizados seguindo procedimentos e técnicas eficientes. Procedimentos de teste para o aspecto lógico de um sistema computacional são bastante difundidos e empregados [Pressman 92, pág. 595-661], mas ao estudar as principais referências e ferramentas para testes observa-se uma falta de procedimentos eficientes para o teste temporal [Braga 99]. A partir desta observação, iniciou-se uma pesquisa com o objetivo de desenvolver uma ferramenta de medição de tempos, aplicável aos sistemas desenvolvidos no LES.

Para testar um sistema computacional, é preciso executá-lo a partir de um conjunto de entradas bem conhecido que permita revelar falhas no funcionamento. Cada entrada utilizada no teste é chamada vetor de teste e deve ser cuidadosamente planejada para avaliar se os requisitos do usuário foram implementados corretamente. Caso um teste realizado com um vetor não revele nenhuma falha, não se pode garantir que o sistema está livre de falhas, entretanto, aumenta-se a confiança no sistema [Maldonado 98].

A confiabilidade de um sistema pode ser medida pela relação entre o número testes realizados e o número de falhas identificadas. Entretanto, se os casos de teste não forem bem projetados, pode-se exercitar sempre as mesmas funcionalidades do sistema, fazendo com que a confiabilidade real do produto permaneça constante. Desta forma, o procedimento correto ao detectar-se a saturação da curva de confiabilidade é adotar uma técnica mais rigorosa para projeto de casos de teste (Figura 1).

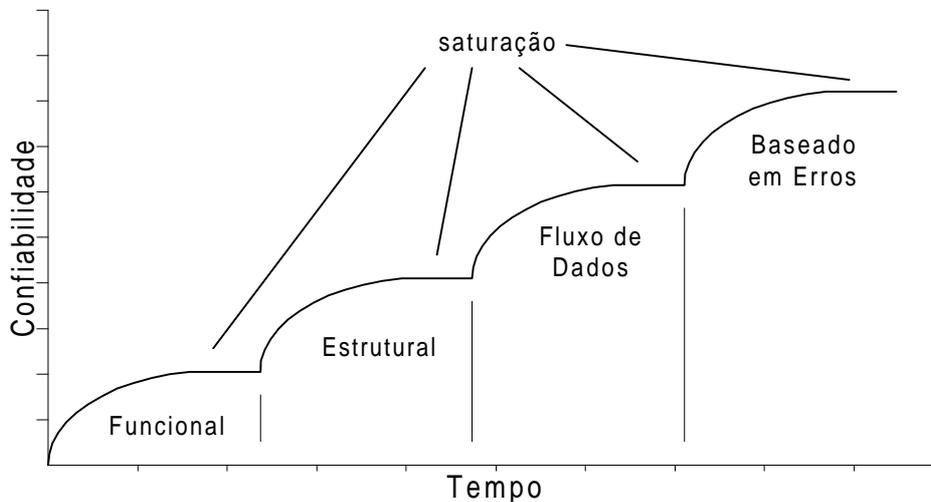


Figura 1 - Evolução da confiabilidade com as técnicas de projeto de casos de teste ²

Em geral, as técnicas para projeto de casos de teste são complementares, ou seja, devem ser utilizadas em conjunto para permitir um aumento da confiabilidade do sistema. Alguns exemplos destas técnicas são:

- Teste Funcional: os casos de teste são definidos a partir da especificação do sistema;
- Teste Estrutural: projeta-se os casos de teste a partir da estrutura de controle (tomadas de decisão) do sistema;
- Teste de Fluxo de Dados: os casos de teste são projetados a partir da definição e uso de variáveis;
- Teste Baseado em Erros: utiliza-se os erros mais freqüentes cometidos no desenvolvimento para definir os casos de teste.

Para o teste de restrições temporais, a proposta da pesquisa é adotar as mesmas técnicas, devidamente adaptadas para identificar falhas temporais. Para a implementação da primeira versão do TLM utilizam-se as técnicas de teste funcional e estrutural, ficando as restantes como propostas para versões futuras.

4. Time Line Monitor

Ao iniciar o desenvolvimento do TLM, foi feita uma grande pesquisa bibliográfica procurando identificar quais os métodos de projeto mais adequados para a ferramenta. Todo o processo de desenvolvimento foi direcionado para

.....
² [Mathur 96]

obter uma ferramenta confiável, precisa, fácil de ser utilizada e adaptável, pois deverá ser utilizada em teste de diversos tipos de sistemas diferentes. Com a tecnologia empregada, a adaptação pode ser realizada facilmente através de mudança na programação do sistema. O TLM pode medir tempos com uma resolução melhor que 10^{-7} s.

Com a experiência adquirida no desenvolvimento dos projetos no LES, foi possível identificar, entre outros, dois problemas relacionados com os testes de sistemas dedicados e em tempo real: a interação entre processos e o uso de recursos do sistema.

Um STR típico executa uma série de atividades em paralelo (chamadas processos); por exemplo: um programa para controlar uma máquina pode ser responsável por manter a velocidade de rotação constante, o consumo de energia dentro de limites adequados e a segurança dos operadores. Durante a operação do sistema, há uma forte interação entre os processos; por exemplo: para que o consumo de energia seja diminuído, pode ser necessário diminuir a velocidade. Neste caso, o processo de controle do consumo deve comunicar-se com o controle da velocidade.

A comunicação entre todos os processos é de fundamental importância para o correto funcionamento de todo o sistema. Se eles não trocarem mensagens na ordem correta e nos momentos determinados, haverá falhas na operação do sistema. Ao testar um programa composto por dezenas de processos, é fundamental reconhecer com rapidez erros na seqüência das mensagens entre os mesmos e assim corrigi-los. Para auxiliar esta atividade, utiliza-se uma representação gráfica da comunicação, chamada Linhas de Tempo, onde cada processo é representado por uma linha vertical e as mensagens entre eles por setas. Assim, torna-se mais fácil para um analista compreender o funcionamento dos processos.

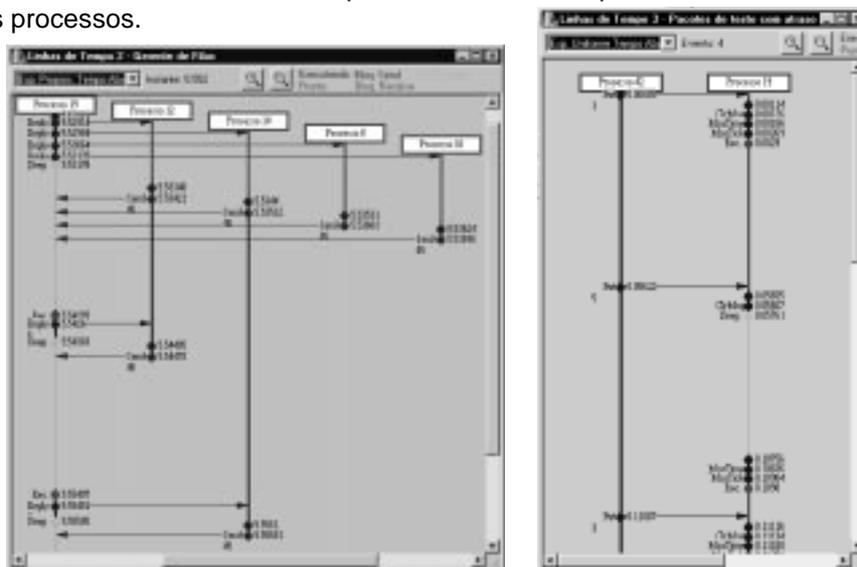


Figura 2 – Linhas de tempo monitoradas com o TLM

Para auxiliar esta compreensão, uma das capacidades do TLM é acompanhar (monitorar) todas as interações entre os processos e outros eventos relevantes, armazenando o instante de cada ocorrência. Após a monitoração, o TLM apresenta as linhas de tempo na tela de um computador, facilitando a análise. A Figura 2 apresenta dois exemplos de linhas de tempo obtidas a partir da monitoração de vários processos realizada pelo TLM. Cada evento é representado por um círculo com a indicação do instante da sua ocorrência. Analisando a seqüência dos eventos, a comunicação entre os processos e o instante de cada ocorrência, é possível avaliar o comportamento lógico e temporal do sistema que está sendo testado e assim localizar erros na implementação.

Conforme descrito na seção 2, sistemas dedicados possuem recursos computacionais muito limitados; portanto, é preciso ter certeza que são estes os recursos são utilizados da forma mais otimizada possível. Se a programação não for feita de forma criteriosa, com cuidado especial para a otimização, é possível que um determinado processo utilize demais os recursos disponíveis, enquanto outros processos não têm os necessários.

A falta de recursos livres para os processos causa falhas durante a operação do sistema, pois os tempos envolvidos nas interações entre processos tornam-se muito elevados. Para auxiliar a localização de erros deste tipo, o TLM permite a análise de histogramas, que informam o tempo total durante o qual os processos executaram. Através desta informação, o analista pode determinar quais processos estão consumindo mais recursos.

A Figura 3 apresenta um exemplo de histograma gerado a partir de uma monitoração do TLM. É possível observar claramente que o processo número 19 ocupa os recursos do sistema durante 53% do tempo. Esta taxa de utilização é muito elevada, pois não deixa recurso livre por tempo suficiente para os demais processos operarem. A partir desta informação, foi possível corrigir o processo, diminuindo a taxa de utilização do processo 19 para apenas 5%.

É importante ressaltar que, se erros desse tipo não forem identificados e corrigidos, há o risco de se ter que adicionar mais recursos ao sistema, aumentando a sua complexidade e custo. Aumento de complexidade invariavelmente implica em aumento no número de erros e aumento de custo é inaceitável em projetos industriais.

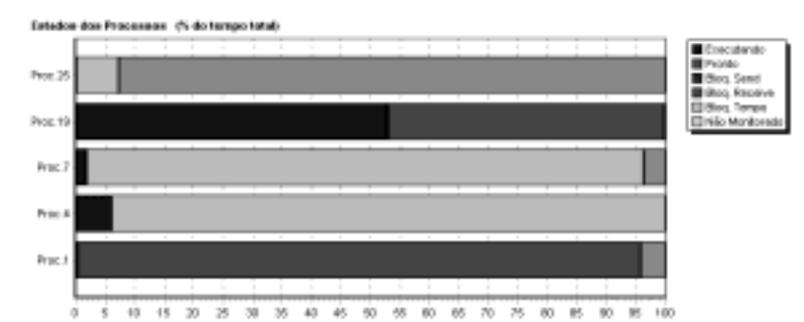


Figura 3 – Taxa de uso de recursos do sistema

5. Considerações Finais

Utilizando o TLM, foi possível testar os projetos em desenvolvimento no LES, avaliando o seu desempenho e a correção lógica e temporal. Estes projetos também foram de fundamental importância para identificar quais são as principais dificuldades envolvidas no teste de equipamentos complexos.

Ao aplicar as técnicas de teste funcional e estrutural nos equipamentos sob teste, foi possível identificar um grande número de falhas, e, através das linhas de tempo e dos histogramas, identifica-se diretamente qual o processo responsável por estas falhas. Com esta capacidade, vários erros puderam ser corrigidos com precisão e rapidez.

A principal contribuição do TLM como trabalho de pesquisa é a adaptação das técnicas de teste funcional e estrutural para avaliar não apenas o aspecto lógico do sistema, mas também o aspecto temporal. Essas técnicas são largamente empregadas durante o teste lógico de sistemas computacionais, mas não foi encontrada nenhuma referência sobre o seu uso industrial para identificar falhas temporais.

Na continuação deste trabalho, propõem-se empregar outras técnicas de teste, por exemplo, o teste de fluxo de dados e os testes baseados em erros, e a integração do TLM com outras ferramentas de apoio ao desenvolvimento de software.

Agradecimentos

O autor gostaria de agradecer ao Prof. Dr. Douglas P. B. Renaux pelo apoio, companheirismo e orientação e à Profa. Faimara R. Strauhs pelo suporte e estímulo para a realização deste trabalho.

Referências

- BRAGA, A. S.; **TLM – Uma ferramenta de apoio ao teste de restrições temporais em sistemas dedicados operando em tempo real**. Dissertação de mestrado no CPGEI-CEFET/PR – Curitiba-PR – Abril/1999.
- DOUGLASS, B. P.; **Real-Time UML: Developing Efficient Objects for Embedded Systems**. Addison-Wesley Longman, Inc. - 1ª edição - 1997.
- MALDONADO, J. C.; Travassos, G. H.; **Teste de Software Orientado a Objetos**. IX Conferência Internacional de Tecnologia de Software, Curitiba - Jun/98
- MANLEY, J. H.; **Embedded Systems**. In: Encyclopedia of Software Engineering. Marciniak, J. J. (editor) - John Wiley & Sons, Inc. - 1994 - pág. 454-458.
- MATHUR, A.; **New Trends in Software Quality Monitoring and Achievement During Software Development**. VII Conferência Internacional de Tecnologia de Software, Curitiba - Jun/96
- PRESSMAN, R. S.; **Software Engineering - A Practitioner's Approach**. McGraw-Hill, Inc. 3ª edição – 1992 - N. York, EUA.