

Sistema de Controle de Trens - Desenvolvimento de uma Aplicação Simulada para Tolerância a Falhas

Marcelo André Minghelli
Fernanda Krueel Denardin
Ingrid E. S. Jansch-Pôrto
{mingheli, fernanda, ingrid }@inf.ufrgs.br

Instituto de Informática
Universidade Federal do Rio Grande do Sul

Resumo

Este artigo apresenta um tópico experimental em desenvolvimento no projeto de “sistemas distribuídos de alta confiabilidade para aplicações críticas”. O aspecto abordado constitui-se em um enfoque preliminar da aplicação que será utilizada para testes das técnicas e ferramentas estudadas e desenvolvidas no decorrer do projeto.

Como aplicação básica optou-se por trabalhar com um sistema de controle de trens, que na vida prática enquadra-se na classe de aplicações críticas, atuando no transporte de pessoas ou cargas, onde qualquer tipo de perda ou acidente é totalmente indesejável.

Será apresentado como o sistema foi projetado e desenvolvido até o presente momento e o estado atual de desenvolvimento da simulação deste.

Abstract

This paper presents an experimental topic which is being developed in the context of the “high reliability distributed systems for critical applications project”. The chosen aspect is a preliminary approach of the envisaged application that will be used to verify the techniques and tools studied and developed during the project.

As a basic application we have chosen to work with a train control system, which in fact is one of the critical applications, once they transport passengers or cargo and any kind of loss is completely undesirable.

In this paper, we will present how this system has been designed and developed to this moment, and the actual development state of this system simulation.

1- Introdução

Alta confiabilidade e disponibilidade sempre foram requisitos essenciais para uma classe de sistemas computacionais classificados como sistemas críticos. Esses sistemas são os utilizados em situações onde eventuais falhas podem por em risco vidas humanas (centrais nucleares, transporte, controle de indústrias, medicina) ou provocar grandes danos financeiros (transações bancárias e bolsas de valores).

Atualmente, um dos projetos de pesquisa do grupo de tolerância a falhas da UFRGS¹ é o projeto SisDAC - Sistemas Distribuídos de Alta Confiabilidade para aplicações críticas

¹ Maiores informações podem ser obtidas na página do grupo de tolerância a falhas em <http://www.inf.ufrgs.br/gpesquisa/tf/>.

[JAN97] . O principal objetivo deste projeto é justamente aplicar técnicas de tolerância a falhas a sistemas distribuídos utilizados em situações críticas, visando alcançar um grau elevado de confiabilidade e disponibilidade em tais sistemas.

Para isso, um dos trabalhos em andamento é dedicado a preparar o suporte para o desenvolvimento e a construção de uma aplicação com características de funcionamento ideais para testes do projeto. A idéia inicial era de montar um sistema de controle de trens físicos baseados em modelos HO, entretanto diante das dificuldades enfrentadas para especificar e adquirir partes no mercado nacional, optou-se por, inicialmente, simular este sistema, conforme será apresentado neste artigo. Cabe ressaltar que a idéia de se trabalhar com trens não é original; um trabalho tomando por base este modelo foi desenvolvido nacionalmente no grupo de ciência da computação da UNICAMP por Elbson Quadros e Cecília Rubira [QUA97].

2- O Sistema de Controle de Trens

O projeto SisDAC prevê o desenvolvimento e a integração de métodos e técnicas de tolerância a falhas aplicáveis a sistemas de controle críticos. A viabilidade das soluções propostas deve ser visualizada através do uso de um sistema de controle automático de trens simulado. Um dos pontos importantes da definição deste protótipo é que todos os componentes da ferrovia ou do trem devem ser tolerantes a falhas.

O funcionamento geral do sistema supõe que o conjunto de trilhos será dividido em malhas, onde cada uma possuirá uma unidade de controle própria. A interação entre estas unidades de controle é estabelecida de forma a comportar-se de acordo com as propriedades dos sistemas distribuídos, permitindo a comunicação de todas as partes e o maior controle organizado do sistema, além do usufruto das facilidades oferecidas pelo modelo. Também as dificuldades associadas ao modelo tais como ausência de memória global, ausência de relógio global e conhecimento limitado das informações referentes às outras malhas, são de interesse do estudo do caso-exemplo.

Será permitido ao usuário alterar fatores como velocidade, trajeto, número de trens na ferrovia e destino de cada um deles, antes do início de cada simulação.

3- A Ferramenta SIMOO MET

Para a implementação deste projeto optou-se por utilizar a ferramenta SIMOO [COP97], desenvolvida por Bernardo Copstein em sua tese de doutorado, no CPGCC (*Curso de Pós-Graduação em Ciência da Computação*) da UFRGS. Esta ferramenta constitui-se em *framework* para simulação discreta orientada a objetos de propósitos gerais, que utiliza um diagrama de classes hierárquico enriquecido com recursos para a construção de modelos de simulação. Uma grande vantagem no uso desta ferramenta é que ela encontra-se disponível, para uso, no Instituto de Informática.

A ferramenta possui uma biblioteca de elementos de visualização que permite a criação de um fundo para a simulação e a movimentação de imagens (tipo *bitmap*), adicionando um aspecto visual e gráfico ao sistema simulado.

Pelas constatações decorrentes do estudo inicial da ferramenta, acredita-se que a sua adoção, altamente motivada pelas facilidades de obtenção e fácil acesso ao seu autor, revelou-se positiva.

4- Apresentação do Sistema

4.1- Uma Visão Geral do Sistema

O foco do presente trabalho consiste na implementação e controle de um modelo de ferrovia utilizando um ambiente de simulação. O modelo inicial definido para a simulação possui um conjunto de trilhos e conectores e dois trens percorrendo a mesma ferrovia, mas em sentidos opostos. O objetivo de ambos os trens (que são identificados como “Azul” e “Vermelho”) é percorrer o trilho até chegar na estação de cor diferente e retornar à sua própria estação por um número determinado de vezes, sem que haja colisão entre eles, terminando por estacionar junto à sua estação após a última volta. Por definição, o trem *Azul* representa um trem de passageiros e o *Vermelho*, um trem de carga.

4.2- Estrutura do Sistema

Cada trem possui uma unidade de controle própria que gerencia a sua movimentação e faz uma análise de colisão. Quando é estabelecido o processo de simulação, o relógio de simulação é inicializado e são criados os elementos gráficos de fundo, juntamente com os dois trens, o que permite a visualização da simulação. A partir daí, o controle central envia uma mensagem para cada controle de trem que coordena independentemente o melhor trajeto durante uma volta. Durante cada volta, o controle realiza a movimentação no setor, a escolha do próximo setor, a escolha do setor a ser colocado na fila de previsão (que será detalhado adiante) e faz o deslocamento dentro da fila. Cada trecho de trilho em reta ou curva corresponde a um setor. Estas rotinas são executadas até que se complete uma volta.

Com relação a tolerância a falhas, o sistema só foi programado para possíveis falhas de controle que causem colisão entre os dois trens. Estas falhas de controle podem colocar dois trens em sentidos opostos em um mesmo setor de trilho ou colocar um trem em uma rota indefinida, fazendo com que a simulação não termine. Como o sistema atual constitui-se em uma versão inicial de teste, ainda não foi implementada a simulação de falhas nos trilhos, conectores e trens, ou seja, os elementos correspondentes às falhas físicas de um modelo real.

4.3- Lógica de Movimentação

A movimentação dos trens é baseada na biblioteca de elementos gráficos de visualização do SIMOO. Há duas rotinas de movimentação por setor de trilhos: uma para curvas e outra para retas. Para retas, o processo é bastante simples: a rotina reta é ativada por uma mensagem e recebe como parâmetros o *bitmap* correspondente ao sentido da reta, a posição inicial em coordenadas cartesianas em relação ao *bitmap* de fundo (x e y) e o valor dos incrementos de x e y em cada movimentação.

Para a movimentação em curvas, o processo é um pouco mais complicado devido às limitações do simulador. Como o simulador não rotaciona os *bitmaps*, é necessário que se criem tantos *bitmaps* quantos forem necessários para dar a impressão de movimento na curva. Neste modelo foram necessários nove *bitmaps* para uma representação real. Assim, para cada curva devem ser indicados os nomes e as coordenadas dos nove *bitmaps* correspondentes. Para dar a impressão de movimentação, os *bitmaps* são criados e apagados, funcionando como quadros em uma animação.

4.4- A Previsão de colisão

A previsão de colisão utiliza um método relativamente simples. Como cada setor de trilho e cada conector são identificados, é possível determinar os próximos movimentos do trem. Para evitar uma colisão com um trem em sentido oposto, é necessário saber se um determinado setor de trilho vai ser utilizado pelo outro trem, antes de completar-se a movimentação. Para isso, cada controlador de trem possui uma fila contendo a identificação do setor atual, o próximo setor e uma previsão do movimento seguinte. Como há um trajeto que deve ser realizado, mas não é possível que ambos os trens utilizem simultaneamente o mesmo setor, a cada movimentação é feita uma comparação entre os elementos da fila.

Na figura 1, é representado um modelo utilizado na simulação. Há um trilho composto de onze setores de trilho (1 a 11), sendo oito em curva e três em reta, seis conectores (A até F) e os trens *Vermelho* e *Azul*. Nesta figura, são mostradas duas situações na simulação: com ambos os trens na sua posição inicial e com ambos os trens no setor dez após duas movimentações.

Analisando a figura 1, verifica-se que os dois trens, se programados para realizar o movimento básico, devem utilizar o setor dez simultaneamente na terceira movimentação. Neste caso, essa colisão já poderia ser prevista e evitada no início da volta. Mas um dos dois deve ter a preferência na passagem. Como o trem *Azul* transporta passageiros, supõe-se ser prioritário o conforto destes mediante a redução do tempo de viagem; logo, ele terá a preferência de passagem. No início da volta do trem *Vermelho*, a previsão de colisão indicaria o setor dez. Então, é ativada uma rotina para selecionar uma rota alternativa, até que ele possa utilizar o setor dez, com base na otimização dos parâmetros tempo e distância. Assim, o elemento de previsão na fila é alterado e, então, é executada a movimentação dos dois trens.

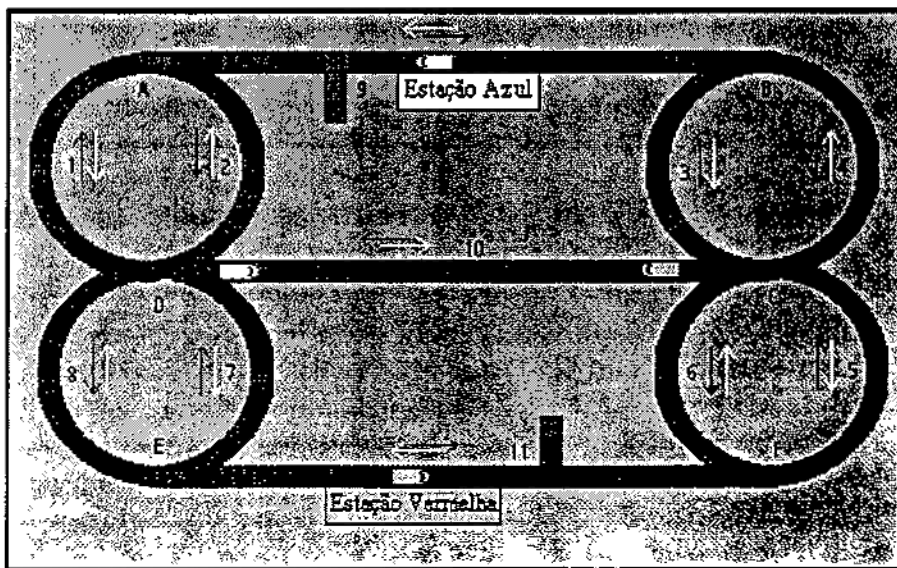


FIGURA 1 - Visualização Gráfica do Sistema de Controle de Trens

5- Conclusão e Extensões Futuras

Este modelo simplificado constitui-se apenas em uma experiência inicial, se considerados os objetivos do projeto. Há várias etapas a serem cumpridas, se for considerado o que o projeto necessita para subsidiar os testes das demais técnicas de tolerância a falhas que serão estudadas.

Algumas perspectivas de trabalhos futuros, que podem ser incorporados em pouco tempo, referem-se à extensão da malha ferroviária, bem como o número de estações e de trens trafegando. Além disso, seria interessante a simulação abrir a possibilidade de se injetar falhas, a cada período de tempo, em algum setor de trilho ou em conectores com defeito, já que em um modelo real, estes elementos não são à prova de falhas.

Também deve ser implementado um controle auxiliar para que o usuário possa acessar o sistema, em tempo de execução, e fazer alterações na rota, velocidade ou destino de um determinado trem.

Até o presente momento, este trabalho teve por objetivos principais a exploração e a identificação dos recursos de uma ferramenta nova, o SIMOO, além do aprendizado das características de um novo tipo de aplicação, do ponto de vista dos autores. A partir destes elementos, torna-se viável a especificação de um novo ambiente com características e potencialidade adequadas ao projeto no qual se enquadra. Os resultados decorrentes deste permitirão observar o comportamento do sistema físico correspondente associado a todas as facilidades de atualização e alteração proporcionadas pelos sistemas computacionais.

6 - Créditos

Os autores Fernanda e Marcelo devem sua participação neste projeto ao financiamento de bolsas DTI e Iniciação Científica, respectivamente, no contexto do programa RHAЕ do CNPq.

7- Referências Bibliográficas

- [COP97] COPSTEIN, Bernardo. **SIMOO: Plataforma Orientada a Objetos para Simulação Discreta Multi-Paradigma**. Tese de Doutorado. Porto Alegre: CPGCC da UFRGS, 1997.
- [QUA97] QUADROS, Elbson; RUBIRA, Cecília. **Uma experiência prática em programação orientada a objetos, distribuída e confiável**. In: Simpósio de Computadores Tolerantes a Falhas, 7. Anais. Campina Grande: SBC, 1997.
- [JAN97] JANSCH-PÔRTO, Ingrid; WEBER, Taisy; WEBER, Raul. **SisDAC - Sistemas Distribuídos de Alta Confiabilidade para aplicações críticas**. Projeto submetido e aprovado pelo programa RHAЕ/CNPq. Em execução desde 1997.