



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**PADRONIZAÇÃO XML DE INTERFACE PARA WEB-
PERFORMCHARTS, BASEADA EM W3C, E
INCORPORAÇÃO DE NOVOS MÉTODOS DE GERAÇÃO
DE TESTES PARA SOFTWARE CRÍTICO DE
APLICAÇÕES ESPACIAIS**

Ana Paula de Oliveira Garcia

Relatório Final de Iniciação Científica,
orientada pelo Dr. Nandamudi
Lankalapalli Vijaykumar e Dr. Gian
Ricardo Berkenbrock, aprovada em 03
de agosto de 2019.

INPE

São José dos Campos

2020



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**PADRONIZAÇÃO XML DE INTERFACE PARA WEB-
PERFORMCHARTS, BASEADA EM W3C, E
INCORPORAÇÃO DE NOVOS MÉTODOS DE GERAÇÃO
DE TESTES PARA SOFTWARE CRÍTICO DE
APLICAÇÕES ESPACIAIS**

Ana Paula de Oliveira Garcia

Relatório Final de Iniciação Científica,
orientada pelo Dr. Nandamudi
Lankalapalli Vijaykumar e Dr. Gian
Ricardo Berkenbrock, aprovada em 03
de agosto de 2019.

INPE

São José dos Campos

2020

RESUMO

Este trabalho tem como objetivo a implementação de uma interface textual baseada na linguagem de marcação SCXML e incorporar outros métodos de geração de testes já desenvolvidos, como *H-Switch Cover*, *Breadth First Search* e *Depth First Search* na ferramenta WEB-PerformCharts. O projeto facilita o uso da ferramenta através de uma interface XML padrão W3C para usuários interessados que já trabalham com UML e Statecharts. Do trabalho concluído até o presente momento teve-se a realização de pesquisas bibliográficas das principais áreas relacionadas ao projeto de pesquisa, incluindo os conceitos básicos de Statecharts, sistemas reativos, PcML e SCXML e o estudo aprofundado da ferramenta WEB-PerformCharts. A próxima etapa do trabalho será a validação do interpretador PcML, o desenvolvimento da ferramenta já com a realização de testes para assim disponibilizar a mesma no servidor providenciado pela equipe da UFSC – Universidade Federal de Santa Catarina, escrita do manual de usuário e também a documentação da ferramenta para que assim, atualizações e/ou incrementações com outras funcionalidades sejam simples.

Palavras-chave: Software crítico. Testes automáticos. WEB-PerformCharts.

ABSTRACT

This work aims to implement a textual interface based on the SCXML markup language and incorporate other test generation methods already developed, such as H-Switch Cover, Breadth First Search and Depth First Search in the WEB-PerformCharts tool. The project facilitates the use of the tool through a standard W3C XML interface for interested users who already work with UML and Statecharts. From the work completed to the present moment, bibliographic research has been carried out in the main areas related to the research project, including the basic concepts of Statecharts, reactive systems, PcML and SCXML and the in-depth study of the WEB-PerformCharts tool. The next stage of the work will be the validation of the PcML interpreter, the development of the tool with tests already carried out so as to make it available on the server provided by the UFSC team – *Universidade Federal de Santa Catarina*, writing the user manual and also the documentation of the tool so that updates and / or increments with other features are simple.

Key words: Critical software. Automatic tests. WEB-PerformCharts.

LISTA DE FIGURAS

Figura 1.1 - Exemplo de uma MEF	12
Figura 1.2 - Representação de um Statechart.....	13
Figura 2.1 - Especificação em Statecharts de um sistema com duas máquinas e um reparador.	17
Figura 2.2 - Exemplo da linguagem de marcação utilizada no projeto.....	19
Figura 2.3 - Diagrama PcML	21
Figura 2.4 - Arquitetura da ferramenta WEB-PerformCharts.....	23
Figura 2.5 - Exemplo do funcionamento de um Sistema Colaborativo.....	25

LISTA DE ABREVIATURAS E SIGLAS

INPE – Instituto Nacional de Pesquisas Espaciais

LABAC – Laboratório Associado de Computação e Matemática Aplicada
MEF – Máquinas de Estados Finitos

PcML – *PerformCharts Markup Language* **UML** – Linguagem de Modelagem Unificada

W3C – *World Wide Web Consortium*

XML – *Extensible Markup Language*

SCXML – Linguagem de marcação - *State Chart XML*

HTML – *HyperText Markup Language*

PHP – *HyperText PreProcessor*

SUMÁRIO

1. INTRODUÇÃO	12
1.1. OBJETIVO GERAL.....	12
1.2. OBJETIVOS ESPECÍFICOS	14
2. DESENVOLVIMENTO.....	15
2.1. METODOLOGIA.....	15
2.2. STATECHARTS	16
2.2.1. INTRODUÇÃO	16
2.2.2. DESENVOLVIMENTO.....	18
2.3. SCXML	19
2.3.1. INTRODUÇÃO	19
2.3.2. DESENVOLVIMENTO.....	20
2.4. PCML E INTERPRETADOR.....	21
2.4.1. INTRODUÇÃO	21
2.4.2. DESENVOLVIMENTO.....	22
2.5. WEB-PERFORMCHARTS	23
2.5.1. INTRODUÇÃO	23
2.5.2. DESENVOLVIMENTO.....	24
2.6. SISTEMAS COLABORATIVOS	25
3. CONCLUSÃO	26
REFERÊNCIAS BIBLIOGRÁFICAS.....	27

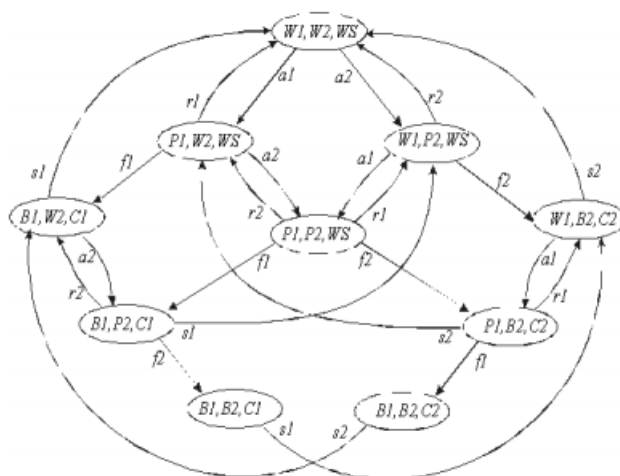
1. INTRODUÇÃO

1.1. OBJETIVO GERAL

Software para aplicações espaciais é considerado crítico devido ao alto custo e tempo dispendido para embarcar em equipamentos que vão a bordo de satélites, foguetes e balões estratosféricos. O INPE é responsável por fabricar satélites e balões estratosféricos para avaliação de diversos fenômenos no espaço, terra e nos oceanos. Os equipamentos que estão a bordo são embarcados com software para fazer uma série de tarefas, desde housekeeping até o seu controle. Devido à particularidade de tal software, é necessário considerar que este seja muito bem qualificado para evitar eventuais problemas. Assim, uma vez que os satélites e/ou balões estejam no espaço, inviabiliza, muitas vezes, a sua correção in loco.

A atividade que qualifica um software, em particular, crítico, é a de Validação por meio de testes [Delamaro et al., 2007] e [Ammann & Offutt, 2008]. O Laboratório Associado de Computação e Matemática Aplicada (LABAC) do Instituto Nacional de Pesquisas Espaciais (INPE) desenvolveu uma ferramenta para geração automática de testes de software, WEB-PerformCharts [Arantes et al., 2008] e [Arantes et al., 2014], na qual a especificação de um software é modelada por meio de Máquinas de Estados Finitos (MEF) [Lee & Yannakakis, 1996] ou Statecharts [Harel, 1987] que conseguem representar sistemas reativos. Quando a especificação está modelada em Statecharts, a ferramenta converte para uma MEF, a partir da qual se obtêm testes aplicando alguns critérios implementados.

Figura 1.1 - Exemplo de uma MEF

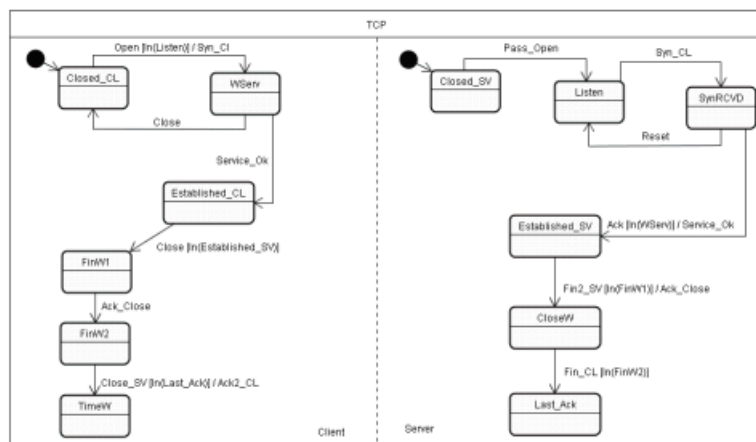


Fonte: Amaral, 2005.

Uma MEF consiste de estados, eventos (ou entradas) e transições. O sistema, representado como uma MEF, está em algum estado num certo instante e quando um evento é habilitado, ocorre uma transição para um outro estado. Os Statecharts são uma extensão de MEFs com a adição de mecanismos para representar atividades paralelas, condições de guardas e até eventos considerados internos onde a reação ocorre sem uma intervenção externa.

Até o momento, está sendo utilizada uma interface textual, baseada em XML, consistindo de tags para se referir aos estados, eventos, transições e condições de guarda. Esta linguagem foi chamada de PcML (PerformCharts Markup Language) [Amaral, 2005]. Ao acionar a ferramenta WEB-PerformCharts e depois de ter acesso via login e senha, é necessário fazer um upload de um arquivo PcML onde há o modelo de uma especificação de software. Este modelo pode ser tanto MEF quanto Statecharts. A ferramenta chama um interpretador para gerar um programa principal em C++. O programa principal é executado e gera uma MEF (se o modelo da especificação estiver em Statecharts). Com a MEF, via ferramenta, é escolhido um critério (Transition Tour, Switch Cover, UIO, DS) [Sidhu & Leung, 1989] para se obter as sequências de testes.

Figura 1.2 - Representação de um Statechart



Fonte: Amaral, 2005.

Quando esta linguagem, PcML (PerformCharts Markup Language) foi proposta, não havia nenhum padrão. Com a popularidade dos Statecharts e, em particular, quando o seu uso foi incorporado na linguagem UML para modelar comportamento de uma classe, a W3C propôs uma linguagem de marcação – State Chart XML (SCXML).

1.2. OBJETIVOS ESPECÍFICOS

A proposta do projeto de Iniciação Científica - Padronização XML de Interface para WEB-PerformCharts, baseada em W3C, e Incorporação de novos Métodos de Geração de Testes para Software Crítico de Aplicações Espaciais – tem como objetivo principal a implementação da interface baseada no padrão SCXML e a incorporação de outros métodos de geração de testes já desenvolvidos como H-Switch Cover [Souza et al., 2017], Breadth First Search e Depth First Search [Mariano et al., 2016] e [Mariano et al., 2019] na ferramenta.

Desta forma, a proposta do projeto coloca a ferramenta WEB-PerformCharts utilizando uma interface XML padrão W3C, assim, então, facilitando o seu uso por usuários interessados que já trabalham com UML e Statecharts.

2. DESENVOLVIMENTO

2.1. METODOLOGIA

O projeto em questão tem como objetivo principal a implementação de um algoritmo para a automação de testes como uma tentativa de reduzir custos associados a esses, aumentando a detecção de falhas e diminuir os ciclos de testes, assim facilitando o uso da ferramenta WEB-PerformCharts através de uma interface XML padrão W3C para usuários que já trabalham com UML e Statecharts.

Dessa forma, para atingir o objetivo proposto, foi necessário seguir a seguinte metodologia:

- (1) Conhecimento de Statecharts.
- (2) Conhecimento da linguagem SCXML.
- (3) Conhecimento da linguagem PcML e de seu interpretador.
- (4) Conhecimento da ferramenta WEB-PerformCharts
- (5) Utilização da ferramenta, com alguns exemplos, para se familiarizar com o seu uso.
- (6) Fazer uma comparação entre as linguagens PcML e SCXML e, então, encontrar os aspectos comuns para serem reutilizados na adaptação do interpretador.
- (7) Estender o interpretador para lidar com os aspectos diferentes entre as linguagens PcML e SCXML.
- (8) Validação do interpretador e gerar vários casos de exemplos.
- (9) Colaboração em conjunto com a equipe da Universidade Federal de Santa Catarina – UFSC – para colocar a ferramenta no servidor que será disponibilizado por eles no Campus Joinville.
- (10) Especificação, desenvolvimento e realização de testes.
- (11) Disponibilização da ferramenta.
- (12) Escrita de um manual de usuário.
- (13) Documentação da ferramenta para facilitar a atualização e/ou incrementação com outras funcionalidades.

2.2. STATECHARTS

2.2.1. INTRODUÇÃO

Statecharts são modelos baseados em *Communicating Finite State Machines* apropriados para modelar sistemas reativos complexos, pois são uma técnica formal de especificação do comportamento destes sistemas que estendem diagramas de estado com conceitos de decomposição hierárquica de estados fornecendo noções de profundidade (abstração); ortogonalidade que permite a representação de atividades paralelas; e interdependência e sincronismo através de comunicação do tipo broadcasting [Harel, 1987]. Além disso, Statecharts são fundamentados nos seguintes elementos básicos: Estados, Eventos, Condições, Ações, Expressões, Variáveis, Rótulos e Transições.

Estados são usados para descrever componentes (e suas possíveis situações) de um determinado sistema. Os estados de um statechart (que representam os valores das variáveis do sistema em um determinado instante) podem ser classificados em dois grupos: básicos e não-básicos. Os estados básicos são aqueles que não possuem subestados. Já os não-básicos são decompostos em subestados. Essa decomposição pode ser de dois tipos: OR ou AND. Se a decomposição é do tipo OR, então o sistema sempre estará em um único subestado em um certo instante. Entretanto, se a decomposição é do tipo AND, o estado estará em mais de um subestado, simultaneamente.

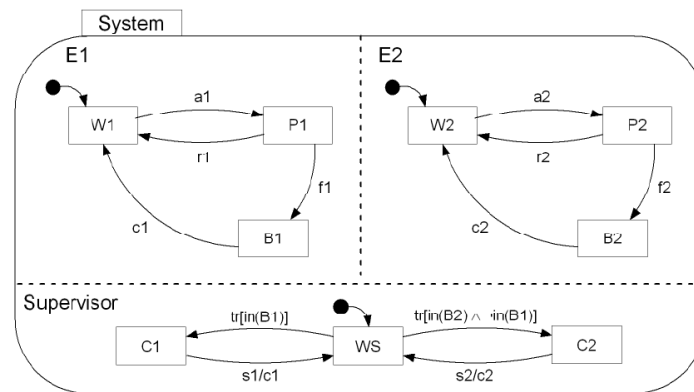
Eventos são considerados a entidade que causa uma interferência no comportamento atual do sistema, levando esse sistema a outro comportamento. Eventos podem ser externos ou internos. Os externos são aqueles que devem ser estimulados explicitamente, já os internos são automaticamente acionados pelo próprio sistema. Opcionalmente, a um evento pode ser anexada uma condição (entre parênteses), também chamada de condição-guarda, de maneira que o evento só ocorrerá se satisfeita aquela determinada condição. Os statecharts proporcionam alguns eventos especiais como true (condição) e false (condição), abreviados na notação statecharts para tr (condição), fs (condição), respectivamente.

O elemento ação é considerado para representar os efeitos do paralelismo em statecharts (a influência de um estado paralelo em outro, também ortogonal). Ações podem ser uma mudança de uma expressão, uma mudança de uma variável ou eventos que são disparados em outros componentes paralelos. Eventos como ação são considerados internos, ou seja, não é necessário o seu estímulo explícito.

As transições são a representação gráfica para denotar uma mudança de estado dentro do sistema. Rótulos nas setas indicam eventos e opcionalmente condições de guarda e ações para prover algum significado adicional.

“Considerando um sistema que contenha duas máquinas e um reparador, como mostra a imagem abaixo.

Figura 2.1 - Especificação em Statecharts de um sistema com duas máquinas e um reparador.



Fonte: Amaral, 2005.

O exemplo da Figura 2.1 representa um Sistema de Manufatura com duas máquinas e um reparador para consertar quando as máquinas quebram. Os componentes paralelos (sub-estados AND da raiz System) são E1, E2, Supervisor e são estados XOR (Ou-Exclusivos). Os componentes E1 e E2 têm suas lógicas representadas por uma MEF e seus subestados. O componente Supervisor é acionado quando há quebra das máquinas com prioridade para conserto da E1 quando as duas estão quebradas. Os estados W1, W2 e WS (dos componentes E1, E2 e Supervisor respectivamente) são os estados iniciais do System. No caso de geração de testes, como os métodos implementados dependem de uma MEF, esta representação Statecharts é convertida para uma MEF plana já mostrada na Figura 1.1. Detalhes e algoritmo desta conversão podem ser encontrados em [Amaral et al., 2003] e [Arantes et al., 2014].

2.2.2. DESENVOLVIMENTO

No projeto em questão, aprofundou-se estudando a técnica *Statecharts*, a qual é utilizada para a especificação de um software. Dessa forma, pode-se observar que a técnica gráfica de *Statecharts* é uma extensão à MEF, entretanto, permite a representação da composição hierárquica de estados, atividades paralelas, sincronismo e interdependência através de comunicação entre componentes *broadcast*.

Tem-se que quando uma especificação é modelada através de *Statecharts*, a ferramenta (WEB-PerformCharts) converte para uma MEF e, a partir daí, obtêm-se os testes aplicando alguns critérios já implementados. Portanto, até o atual momento, é utilizada a interface textual, baseada em XML, a qual consiste de *tags* para se referir aos estados, eventos, transições e condições de guarda. A linguagem em questão é conhecida como PcML.

Quando esta linguagem, PcML (PerformCharts Markup Language) foi proposta, não havia nenhum padrão. Com a popularidade dos *Statecharts* e, em particular, quando o seu uso foi incorporado na linguagem UML para modelar comportamento de uma classe, a W3C propôs uma linguagem de marcação - State Chart XML (SCXML).

2.3. SCXML

2.3.1. INTRODUÇÃO

SCXML significa Statechart XML, que é uma notação de máquina de estado para abstração de controle. É uma linguagem de marcação baseada em XML que fornece um ambiente de execução baseado em máquinas de estados genéricas que se baseiam em gráficos de estados de [Harel, 1987], também é capaz de descrever MEF complexas.

Figura 2.2 - Exemplo da linguagem de marcação utilizada no projeto.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="mfeeX.xsl"?>
<MFEE>
  <STATES>
    <STATE NAME="IdleWaitingSync" TYPE="inicial"/>
    <STATE NAME="CountingTimeWaitingExpid" TYPE="normal"/>
    <STATE NAME="CountingTimeWaitingType" TYPE="normal"/>
    <STATE NAME="CountingTimeWaitingSize" TYPE="normal"/>
    <STATE NAME="CountingTimeWaitingData" TYPE="normal"/>
    <STATE NAME="CountingTimeWaitingChecksum" TYPE="normal"/>
    <STATE NAME="IdleWaitingSync" TYPE="final"/>
  </STATES>
  <EVENTS>
    <EVENT NAME="EB9" VALUE="1"/>
    <EVENT NAME="StartTimingCount" VALUE="1"/>
    <EVENT NAME="NotEB9" VALUE="1"/>
    <EVENT NAME="WaitingTimeExpired" VALUE="1"/>
    <EVENT NAME="Timeout" VALUE="1"/>
    <EVENT NAME="ExpidRec" VALUE="1"/>
    <EVENT NAME="TypeRec" VALUE="1"/>
    <EVENT NAME="SizeRec" VALUE="1"/>
    <EVENT NAME="DataRec" VALUE="1"/>
    <EVENT NAME="ChecksumRec" VALUE="1"/>
    <EVENT NAME="CommandReceived" VALUE="1"/>
  </EVENTS>
</MFEE>
```

Fonte: Autor.

A notação SCXML é um candidato para a linguagem de controle dentro de várias linguagens de marcação provenientes do W3C. Este publicou recentemente a recomendação *Statechart XML (SXCML)* que define uma especificação completa para representar os diagramas de estados por um arquivo XML. O SCXML define várias classes e interfaces Java para controlar um aplicativo.

Nota-se que todas as *tags* devem ter início e fim, como pode ser observado no exemplo, <STATES> indica o início e </STATES> indica o final dessa informação.

2.3.2. DESENVOLVIMENTO

Como já descrito anteriormente, o desenvolvimento consistirá em um profundo estudo da interface padrão SCXML, pois o XML define todos os estados, transições, eventos e variáveis através do uso na máquina de estados. Além dos elementos estruturais dela, a linguagem SCXML também define a semântica de execução dessa máquina de estados.

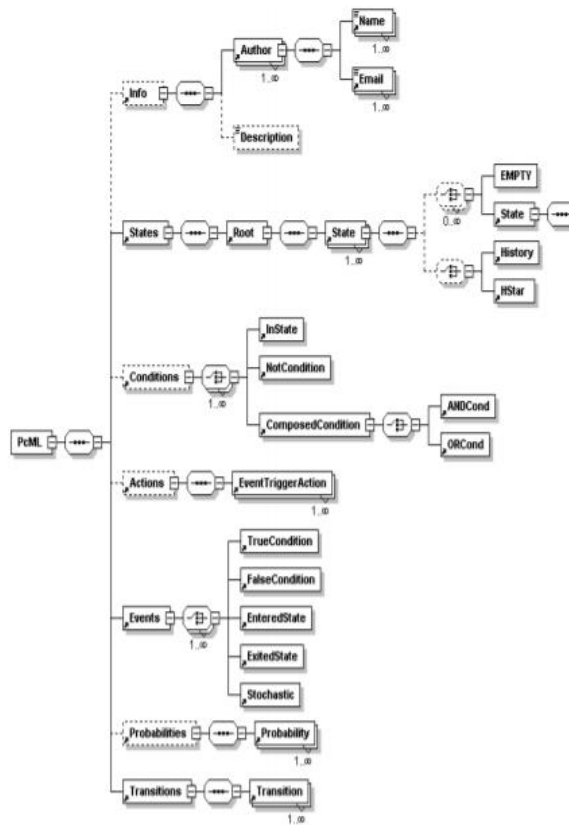
Para se fazer uso do software desenvolvido (WEB-PerformCharts), é necessário a criação de um módulo principal na linguagem de programação C++ a fim de especificar o modelo para o qual se deseja fazer a avaliação de desempenho, fazendo chamadas a métodos necessários para converter uma representação em uma cadeia de Markov e calcular as medidas de desempenho. A solução inicial encontrada para tal problema seria através de uma interface gráfica, ainda não disponível. Dessa forma, teve-se uma solução intermediária a qual seria através de uma interface textual. Assim, uma linguagem de marcação baseada em XML foi criada com o objetivo de especificar e tratar modelos de desempenho.

2.4. PCML E INTERPRETADOR

2.4.1. INTRODUÇÃO

O projeto propõe uma linguagem de marcação, PcML [Amaral, 2004] que significa PerformCharts Markup Language, para ser usada na especificação do sistema. É uma linguagem de marcação baseada em XML, cujas *tags*, atributos e outras características representam os elementos usados em Statecharts na especificação de sistemas reativos com o propósito de avaliação de desempenho [Ray, 2002],[Maruyama, 2002].

Figura 2.3 - Diagrama PcML



Fonte: Amaral et al, 2004.

2.4.2. DESENVOLVIMENTO

Como mencionado anteriormente, foi necessário a transformação da especificação em PcML do sistema em um programa C++ para poder gerar as suas medidas de desempenho. Para atingir tal objetivo, foi preciso criar algumas aplicações na linguagem Java. O programa em C++ consiste, basicamente, de chamar as funções para que:

- (1) criem estruturas de dados que correspondem à representação do sistema em Statecharts;
- (2) simulem a dinâmica do sistema através de estímulos dos eventos criando a cadeia de Markov;
- (3) resolvam a cadeia de Markov a fim de obter as probabilidades limites que são a base das medidas de desempenho.

O código na linguagem Java cria uma árvore hierárquica de nós/objetos, em que a busca aos nós é realizada através do uso de expressões XPath. Em seguida, são mostradas algumas linhas da descrição do sistema em PcML e o código correspondente em C++ gerado.¹

¹ Amaral, Ana Silvia M.; Veloso, R. R.; Vijaykumar, N. L.; Lisboa, C. R.; Oliveira, E. L. **Uma linguagem de marcação para statecharts para ser usada em avaliação de desempenho baseada em xml.** XXXVI Simpósio Brasileiro de Pesquisa Operacional, pp. 1-10. 2004.

2.5. WEB-PERFORMCHARTS

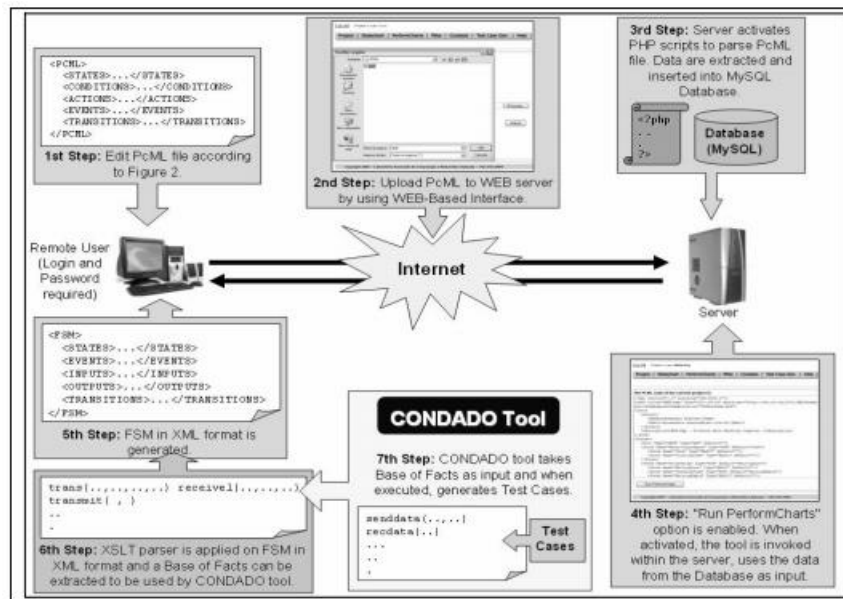
2.5.1. INTRODUÇÃO

Inicialmente desenvolvida para avaliar o desempenho de sistemas reativos, representados em gráficos de estados [Vijaykumar et al., 2002] e [Santiago et al., 2008], a ferramenta WEB-PerformCharts, desenvolvida pelo Instituto Nacional de Pesquisas Espaciais (INPE), tem o objetivo de permitir que diferentes equipes trabalhem em um mesmo projeto através de uma interface baseada na web e na aplicação de banco de dados por meio da internet. Idealizada para ajudar *software testers* que trabalham em locais diferentes para cooperarem em projetos em comum e, assim, aproximar suas experiências a fim de beneficiar a qualidade do software.

Ainda em desenvolvimento, porém conta com a utilização de tecnologias como HTML, PHP e MySQL. Dessa forma, ela pode ser hospedada em um servidor Apache com um servidor operacional Linux de modo a ficar totalmente livre de custos com pacotes de softwares.

O WEB-PerformCharts ainda permite a geração de casos de teste baseados em modelos por meio de métodos formais, ou seja, gráficos de estados de linguagem formal. A ferramenta em questão apresenta quatro critérios de testes diferentes para gerar casos de teste baseados em modelo, fornecendo maior flexibilidade ao designer de teste. Ela converte uma especificação em Statecharts que é uma representação sintética de sistemas críticos e complexos, em uma Máquina de Estados Finitos.

Figura 2.4 - Arquitetura da ferramenta WEB-PerformCharts



Fonte: Arantes et al, 2010.

2.5.2. DESENVOLVIMENTO

No INPE, temos vários setores responsáveis pela construção de equipamentos que vão a bordo de satélites em que o software é embarcado, dessa maneira, utilizando a ferramenta em questão, a realização de testes acaba sendo facilitada. A ferramenta proposta baseia-se na especificação de sistemas reativos utilizando a técnica *Statecharts* e na geração de casos de teste para a mesma de acordo com alguns critérios disponíveis.

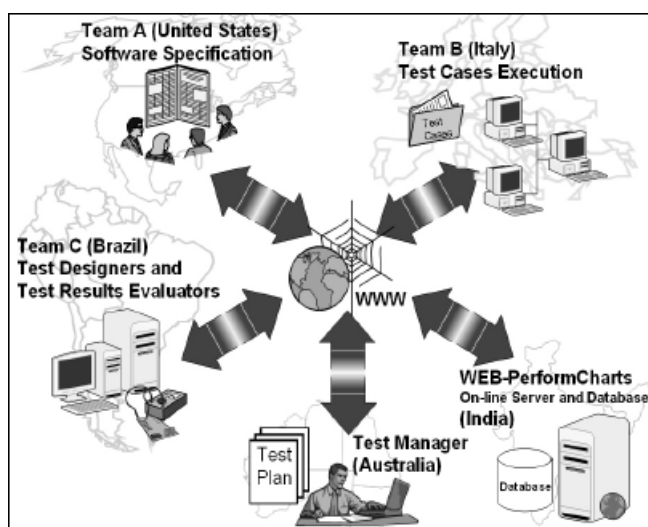
Como já mencionado anteriormente, o principal objetivo do desenvolvimento do WEB-PerformCharts é aproximar as equipes combinando sua experiência e know-how para beneficiar a qualidade do software. Assim, inicialmente, a ferramenta interpreta uma especificação em PcML, transformando esta em uma cadeia de Markov ou em uma MEF no padrão XML e, a partir da MEF, os critérios de teste são aplicados para, então, gerar casos de teste.

2.6. SISTEMAS COLABORATIVOS

O principal objetivo de sistemas colaborativos é ajudar usuários de diferentes locais envolvidos em um mesmo projeto, de modo a ter um grande apoio à comunicação, coordenação e cooperação. Basicamente, eles são aplicativos e/ou ferramentas que operam em redes que auxiliam o trabalho em equipe, assim, ajudando na troca de informações.

As aplicações neste sentido possuem diversos meios de promover qualquer usuário de internet, permitindo uma grande economia de custos e tempo, aumentando o trabalho em conjunto e promovendo a eficiência e qualidade. Uma vez que todos os dados manipulados por um indivíduo podem ser imediatamente percebidos por todos os outros usuários em locais remotos.

Figura 2.5 - Exemplo do funcionamento de um Sistema Colaborativo



Fonte: Arantes et al, 2008.

As aplicações WEB-based apresentam grandes vantagens, pois oferecem soluções de baixo custo, uma vez que nesta arquitetura o usuário pode utilizar qualquer sistema operacional e não requer nenhum outro software proprietário. Sem mencionar que, nos dias atuais, a grande maioria das pessoas possuem um fácil acesso à internet.

3. CONCLUSÃO

A importância de testes para a garantia da qualidade de produtos de software é clara nos dias atuais. Porém, é fato conhecido que as atividades associadas ao processo de teste demandam um tempo significativo do esforço de desenvolvimento de um produto de software. A automação de testes surgiu como uma tentativa de reduzir custos associados aos testes, aumentar a detecção de faltas e diminuir os ciclos de teste. Embora automação de testes não seja a “bala de prata” para resolver todos os problemas relativos a testes, se for corretamente planejada e implementada, ela pode ajudar a obter custo e eficiência das atividades do processo de teste durante o ciclo de vida de desenvolvimento do software. Ambientes que possibilitem a geração automática de casos de testes, a partir da modelagem de softwares complexos, são relevantes em um contexto prático de desenvolvimento de software. A automação da geração de casos de testes é de alguma forma facilitada se o sistema em questão for representado por uma técnica formal, como Statecharts, MEFs ou por outras. Atualmente, existe um ambiente denominado WEBPerformCharts que gera casos de testes caixa preta baseados em alguns critérios quando a especificação de um software é modelada ou em uma MEF ou em Statecharts.

Um projeto descentralizado é uma tendência muito comum para empresas amplamente dispersas nos dias modernos, uma vez que pode resultar em uma grande economia de tempo e custos, assim, diminuindo as necessidades de viagens e infraestrutura. Desse modo, a fim de promover tal tendência para geração de casos de teste, a ferramenta WEB-PerformCharts foi planejada para ser usada via internet em uma abordagem de desenvolvimento distribuído.

A decisão de usar um banco de dados online como método de armazenamento permite que os designers de teste compartilhem seus projetos e, então, facilitando o controle de versões, já que seu gerenciamento é mais fácil do que copiar vários arquivos locais de diversos computadores.

A ferramenta WEB-PerformCharts possui dois níveis de acesso:

- Administradores, com acesso total a qualquer projeto;
- Usuários, com acesso apenas aos projetos criados por eles mesmos.

Os resultados apresentados no trabalho sugerem que a transição pode ser reconhecida como um critério rápido para obter um conjunto de casos de teste, mas gera maior número de casos de teste para aplicações complexas. Dessa forma, a principal contribuição deste projeto é viabilizar uma ferramenta de suporte ao processo de teste em um ambiente distribuído por meio do desenvolvimento de uma ferramenta baseada na web.

Dessa forma, o próximo passo do projeto é o estudo e a validação do interpretador de PcML, o desenvolvimento da ferramenta e a realização de testes, e assim, disponibilizar a ferramenta WEB-PerformCharts e colocá-la no servidor disponibilizado pela equipe da UFSC e, por fim, a escrita de um Manual de Usuário, juntamente com a documentação da ferramenta.

REFERÊNCIAS BIBLIOGRÁFICAS

- Amaral, Ana Silvia M. S.; Veloso R. R.; Vijaykumar, N. L., **PcML Reference Manual. Relatório Técnico a ser publicado, Instituto Nacional de Pesquisas Espaciais**, São José dos Campos, SP, Brasil. 2004.
- Amaral, Ana Silvia M. S.; Vijaykumar, N. L.; Martins, E. **Geração Automática de Casos de Teste de Conformidade para Software de Aplicações em Protocolos de Comunicação.** *Workshop em Computação Aplicada*, pp. 1-6. 2003.
- Amaral, Ana Silvia M.; Veloso, R. R.; Vijaykumar, N. L.; Lisboa, C. R.; Oliveira, E. L. **Uma linguagem de marcação para statecharts para ser usada em avaliação de desempenho baseada em xml.** *XXXVI Simpósio Brasileiro de Pesquisa Operacional*, pp. 1-10. 2004.
- Arantes, A. O.; Vijaykumar, N. L.; Santiago Junior, V. A. **From Statecharts to Test Case Generation on Web.** *Workshop dos cursos de computação aplicada do inpe, 7. (worcap).*, 2007, São José dos Campos. On-line. IBI: <8JMKD3MGP8W/37J5C78>. Disponível em: <<http://urlib.net/rep/8JMKD3MGP8W/37J5C78>>.
- Arantes, A.; Santiago Júnior, V. A.; Vijaykumar, N. L. **Tool support for generating model-based test cases via web.** *Int. J. Web Engineering and Technology*. Vol. 9, N. 1, pp. 1-35. 2014.
- Arantes, A.; Vijaykumar, N. L.; Carvalho, A. R. **Automatic Teste Case Generation Through a Collaborative Web Application.** *Proceeding of the IASTED International Conference*, pp. 1-6. 2008.
- Arantes, A.; Vijaykumar, N. L.; Santiago Junior, V. A.; Guimarães, D. **WEB-PerformCharts: A Collaborative WEB-based tool for Test Case Generation from Statecharts.** *Proceedings of iiWAS2008*, pp. 1-8. 2008.
- Ferreira, D. F.; Nunes, M. K. P.; Ferreira, É.; Arantes, A.; Vijaykumar, N. L. **Integração de métodos de teste à ferramenta WEB-PerformCharts.** *Simpósio De Iniciação Científica e Tecnológica, 12., 2010*, São Paulo, 2010.
- Harel, D. **Stacharts: A visual formalism for complex systems.** *Science of Computer Programming* 8, North-Holland, pp. 1-44. 1987.
- Lisboa, C. R.; Santana, M. J.; Vijaykumar, N. L.; Carvalho, S. V.; Carlucci, R. H. **Statecharts Estocásticos e Queuing Statecharts: Novas Abordagens para Avaliação de Desempenho Baseadas em Especificação Statecharts.** *Simpósio Brasileiro de Engenharia de Software*, pp. 1-16. 2001.
- Marinke, R.; Vijaykumar, N. L.; Senne, Edson L. F. **Geração de Testes Estruturais para aplicações Multithreads: Abordagem por Statecharts.** *II Simpósio de Tecnologia da Informação da Região Noroeste do Rio Grande do Sul*, pp. 1-6. 2020.
- Maruyama, H. **XML and Java – Developing Web Application.** CA: Addison Wesley, 2002.

Vijaykumar, N. L.; Souza, E. **Geração de casos de teste para sistemas da área espacial usando critérios de teste para máquinas de estados finitos.** *INPE - Instituto Nacional de Pesquisas Espaciais*, pp. 1 -137. 2010.