MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-15182-NTC/374

EXPERIÊNCIAS EM PROJETOS E USO DE TÉCNICAS DE VERIFICAÇÃO E VALIDAÇÃO DE SOFTWARE EM APLICAÇÕES ESPACIAIS NO INPE

Ana Maria Ambrósio
Fátima Mattiello-Francisco
Luciana Seda Cardoso
Valdivino Santiago
Ronaldo Arias
Nandamudi Lankalapalli Vijaykumar
Geilson Loureiro

INPE
São José dos Campos
2008

Publicado por: **esta página é responsabilidade do SID**

Instituto Nacional de Pesquisas Espaciais (INPE)
Gabinete do Diretor – (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 – CEP 12.245-970
São José dos Campos – SP – Brasil
Tel.: (012) 3945-6911
Fax: (012) 3945-6919
E-mail: pubtc@sid.inpe.br

Solicita-se intercâmbio
We ask for exchange

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-15182-NTC/374

EXPERIÊNCIAS EM PROJETOS E USO DE TÉCNICAS DE VERIFICAÇÃO E VALIDAÇÃO DE SOFTWARE EM APLICAÇÕES ESPACIAIS NO INPE

Ana Maria Ambrósio
Fátima Mattiello-Francisco
Luciana Seda Cardoso
Valdivino Santiago
Ronaldo Arias
Nandamudi Lankalapalli Vijaykumar
Geilson Loureiro

INPE
São José dos Campos
2008

RESUMO

O caráter crítico de software em aplicações espaciais requer investigações em métodos sistemáticos e processos de teste acompanhados de ferramentas para verificação e validação (V&V). Este relatório descreve projetos, de pesquisa e de desenvolvimento em ferramentas, metodologias e processos para melhorar a qualidade dos sistemas de software das missões espaciais brasileiras, que estão em andamento no Instituto Nacional de Pesquisas Espaciais (INPE) atualmente. Alguns projetos abordam exclusivamente técnicas de teste de software como geração de dados de teste, seleção de dados de teste, ferramentas para apoiar a execução de testes, análise de resultados de testes e também de apoio ao gerenciamento de testes. Outros projetos cobrem um espectro mais abrangente no processo de verificação no ciclo de vida do software ou podem ser entendidos a uma visão de sistema. Este relatório apresentada trabalhos em andamento listando os projetos, ferramentas, metodologias, processos e técnicas de teste de software exploradas. Vantagens e desvantagens no desenvolvimento dos projetos percebidas pelas equipes são também discutidas.

Palavras chaves: Testes de Software, Aplicações Espaciais, Testes de Conformidade, Injeção de Falhas por software, Ferramentas de teste, Processo de V&V.

Categorias e assuntos:
D.2.0 [Software Engineering]: General – Standards
D.2.4 [Software Engineering]: Software/Program Verification – validation.
D.2.9 [Software Engineering]: Management – software quality assurance (SQA).

Termos Gerais: Teste, Verificação, Validação, Padrões.

ABSTRACT

The critical feature of space applications requires investments of time and effort for defining systematic testing methods and processes associated to tools to support verification and validation (V&V). This report describe the research and development projects related to tool design, methodologies and processes definition to improve the quality of software and systems of the Brazilian Space Missions which are currently being carried on at the National Institute for Space Research. Some projects focuses exclusively on software testing techniques as, test data generation, test data selection, support to test execution, test result analysis and support to test management. Other projects cover a broader verification processes in the whole software lifecycle or may be applied at a system point of view. This report presents the ongoing projects, tools, methodologies, processes and testing techniques being applied in the current projects. The project advantages and drawbacks perceived by the working teams are also discussed.

SUMÁRIO

Pág.

LISTA DE FIGURAS.....13
LISTA DE TABELAS.....14
LISTA DE SIGLAS E ABREVIATURAS.....15
1. INTRODUÇÃO 9
2. TÉCNICAS de V&V 12
3 PROJETOS DE PESQUISA EM V&V 15
3.1 ATIFS - Ambiente de Teste com Injeção de Falhas por Software.....15
3.2 QSEE – Qualidade de Software Embarcado em Aplicações Espaciais.....17
3.3 GTSC - Geração de Teste a partir de StateCharts.....20
4 EXPERIÊNCIAS COM V&V NOS PROJETOS COMAV e SATCS.....21
4.1 COMputador de Bordo AVançado (COMAV).....21
4.2 Sistema de Controle de SATélites (SATCS).....22
4.2.1 Processo e Técnicas de V&V.....23
4.2.2 Controle e organização dos testes.....24
4.2.3 Sistema de Controle de Casos de Testes (SCCT).....25
5. O PROCESO DE VVI no projeto QSEE.....25
5.1 Metodologia CoFI 26
5.2 Resultados 28
6. CONCLUSÕES E TRABALHOS FUTUROS 30
AGRADECIMENTOS 32
REFERÊNCIAS BIBLIOGRÁFICAS.....32

LISTA DE FIGURAS

1 - Arquitetura do Projeto ATIFS 16

2 - Visão geral do QSEE ….....17

LISTA DE TABELAS

1 – Projetos do INPE relacionados com atividades e técnicas de V&V.....14
2 – Passos da metodologia CoFI.……..21
3 – Serviços X número de modelos criados em VVI noE.....22
4 – Experimentos de Injeção de falhas x erros detectados.....22

LISTA DE SIGLAS E ABBREVIAÇÕES

AOCS	Attitude and Orbit Control Subsystem
ATIFS	Ambiente de Teste com Injeção de Falhas por Software
CCSDS	Cooperation Committee for Space Data Standardization
CEA	Ciências Espaciais e Atmosféricas
CMMI	MPS.BR-A
COFI	Conformace and Fault Injection
COMAV	COMputador AVançado
ECSS	European Cooperation for Space Standardization
ETE	Engenharia de Tecnologias Espaciais
EFSM	Extended Finite State Machine
FSM	Finite State Machine
GTSC	Geração Automática de Casos de Teste Baseada em Statecharts
IUT	Implementation Under Test
LIT	Laboratório de Integração e Teste
MME	Modelos de Estados
MPS.BR-A	Modelo de Negócio para Melhoria de Processo de Software Brasileiro
OBDH	On-Board Data Handling
PCO	Point of Control and Observation
PETS	Parameterized Executable Test Suite
PLAVIS	Plataforma para Validação e Integração de Software em Aplicações Espaciais
QSEE	Qualidade do Software Embarcado em Aplicações Espaciais
QSEE-TAS	QSEE – Teste Automatizado de Software
SATCS	Sistema de Controle de SATélites
SCCT	Sistema de Controle de Casos de Testes
SPAC	Processamento e Análise de dados Científicos

SWPDC	SoftWare embedded into the Payload Data Handling Computer
SWIFI	Software Implemented Fault Injection
TC	Telecommand
TM	Telemetry
UML	Unified Modeling Language
V&V	Verificação e Validação
XSLT	Extensible Stylesheet Language Transformation

1. INTRODUÇÃO

O Instituto Nacional de Pesquisas Espaciais (INPE) é um órgão específico singular do Ministério de Ciência e Tecnologia. Sua principal finalidade é "promover e executar estudos, pesquisas científicas, desenvolvimento tecnológico e capacitação de recursos humanos, nos campos da Ciência Espacial e da Atmosfera, das Aplicações Espaciais, da Meteorologia e da Engenharia e Tecnologia Espacial, bem como em domínios correlatos consoantes com a política definida pelo Ministério" [Portaria N° 435 do MCT, art. 3°]. De maneira abrangente, o INPE tem como missão contribuir para que a sociedade brasileira possa usufruir os benefícios propiciados pela tecnologia espacial.

Satélites artificiais e balões estratosféricos têm sido desenvolvidos pelo INPE. Estes veículos espaciais em diferentes graus de complexidade são regidos por sistemas computacionais que possuem software dedicado. Muitos deles estão embarcados na eletrônica de vôo e são desenvolvidos especificamente para cada missão. Outros tipos de software residem em solo e se destinam à operação remota do veículo espacial. Programas de satélites, por exemplo, contam com o apoio de sistemas de software desde a fase de concepção e desenvolvimento até a fase de operação da missão em órbita. Na área de controle de satélites, existem sistemas de software projetados para realizar monitoração e controle em solo e em bordo além de efetuar o controle de órbita e atitude do veículo espacial. A bordo de veículos espaciais encontram-se basicamente três categorias de sistemas computacionais: uma para controle, aquisição e processamento de dados das cargas úteis do satélite; uma para controle de atitude e órbita do satélite (Atittude and Orbit Control Subsystem - AOCS) e uma para a implementação das tarefas de supervisão (On Board Data Handling - OBDH) do conjunto de subsistemas que compõem o veículo espacial, ao longo da vida útil da missão, em cooperação com os sistemas de solo. Em termos do controle remoto por solo, sistemas computacionais para geração e transmissão de telecomandos, bem como para recepção, armazenamento e visualização de telemetrias, são típicos em aplicações espaciais. A característica comum de todo software próprio de aplicações espaciais é de ser reativo e possuir funções de comunicação, pois interagem continuamente com sensores e atuadores, recebem e emitem comandos para a realização de suas funções.

A linha de pesquisa de Testes de Software vem sendo desenvolvida na Engenharia de Tecnologias Espaciais (ETE) do INPE desde a década de noventa, em decorrência da capacitação na área de tolerância a falhas dada à equipe responsável pelo desenvolvimento dos sistemas de software do segmento solo e espacial da Missão Espacial Completa Brasileira, no início dos anos 80. O projeto Ambiente de Testes baseado em Injeção de Falhas por Software (ATIFS) foi o primeiro projeto de Pesquisa e Desenvolvimento (P&D) entre o INPE e o Instituto de Computação da Universidade de Campinas (UNICAMP) com objetivo de explorar técnicas e métodos de testes para a validação de software em aplicações espaciais (www.inpe.br/atifs). Ainda em andamento, este projeto permite que protótipos de ferramentas para geração automática de testes a partir de modelos de estados, execução de testes com injeção de falhas e análise de resultados, desenvolvidos no âmbito da universidade sejam utilizados no domínio da área espacial. Como desdobramento do projeto ATIFS outras competências em Verificação, Validação e Testes de software existentes em universidades no país e outros grupos de pesquisa dentro no INPE, a ETE concebeu, em 2002, o projeto PLAVIS – Plataforma para Validação e Integração de Software em Aplicações Espaciais (www.labes.icmc.usp.br/plavis), o qual foi financiado em primeira instância pelo CNPq/Universal e posteriormente pelo CAPES/Cofecub, com a participação francesa no projeto. O produto final do projeto foi prover o INPE com uma plataforma integrada de VV&T para aplicações espaciais [25]. Em 2004, visando explorar especificamente a abordagem de V&V Independente de Software, a ETE, em conjunto com a Ciências Espaciais e Atmosféricas (CEA) do INPE e o Instituto de Computação da Unicamp, criou o projeto Qualidade do Software Embarcado em Aplicações Espaciais (QSEE) (http://www.cea.inpe.br/qsee/) financiado pela FINEP. Tendo por meta o engajamento da indústria brasileira de software como fornecedor do software embarcado em missões espaciais para o INPE, o projeto utiliza os processos de verificação e validação recomendados pelas normas European Cooperation for Space Standardization (ECSS), para guiar a relação cliente/ fornecedor na aquisição de software embarcado [19].

No seu terceiro ano de execução e, com término previsto para dezembro de 2007, o projeto QSEE produzirá um procedimento para aceitação de software que poderá ser adotada por um laboratório de ensaios de software embarcado de

forma independente da equipe de desenvolvimento do software. Como parte de um processo de V&V independente, foi aplicada na prática, a metodologia de teste Conformace and Fault Injection (CoFI) [4], concebida no projeto ATIFS como uma tese de doutorado.

Em 2006, por iniciativa do Laboratório de Integração e Teste (LIT), foi proposto um laboratório de teste de software (LAVVISE) cujo objetivo é prestar serviços de teste de software em um contexto de V&V independente para software embarcado em equipamentos de satélites, aviões, médicos, automobilísticos que atendam as necessidades do INPE e da indústria.

Devido à importância de atividades de V&V em software de aplicações espaciais, outros projetos estão em andamento na instituição, como o de Geração Automática de Casos de Teste Baseada em Statecharts (GTSC) que prevê a geração automática de casos de teste a partir de diagramas Statecharts da UML e uma extensão para acesso via web (WebPerformCharts), que acrescenta análise de desempenho [21].

Visando a exploração tecnológica para o desenvolvimento de software de supervisão de bordo, a ETE apóia o projeto COMputador AVançado (COMAV). O projeto COMAV visa desenvolver um computador de bordo para ser utilizado em futuros satélites, incorporando novas tecnologias de hardware e software para esse tipo de aplicação, seja em termos de metodologia e processo de desenvolvimento, seja em termos de componentes de hardware.

Na área de controle e monitoração remota de satélites está sendo desenvolvido na ETE o projeto Sistema de Controle de SATélites (SATCS). O projeto SATCS explora a utilização de metadados e design patterns para obter o máximo de reuso do software de controle e operação de satélites em solo de uma missão para outra, visando obter mínimos custos com operação de satélite mas mantendo a grau de qualidade exigido para tal atividade [13].

Este artigo está organizado da seguinte maneira: a seção 2 discute as técnicas de V&V exploradas e aplicadas nos projetos citados; a seção 3 apresenta as principais características dos projetos ATIFS, QSEE e GTSC; a seção 4 descreve os aspectos de V&V dos projetos COMAV e SATCS. A seção 5 apresenta alguns resultados da mais recente experiência em V&V independente realizada no INPE; e finalmente a seção 6 discute algumas lições aprendidas nos diversos projetos

apresentados. Devido à falta de espaço no artigo, as referências bibliográficas citadas referem-se apenas aos trabalhos publicados das equipes envolvidas.

2. TÉCNICAS de V&V

A Tabela 1 relaciona as técnicas e atividades de V&V exploradas nos projetos apresentados, quais sejam: revisões, teste estrutural, teste baseado na especificação, teste baseado em modelos de estados (*Finite State Machine* (FSM), *Extended Finite State Machine* (EFSM) e *Statecharts*), técnica de injeção de falhas por software; execução automatizada de teste, processo de V&V de software, V&V Independente de software, metodologia para geração de teste de software aplicações espaciais e gerenciamento e controle automatizado dos testes de um dado produto de software.

Tabela 1. Projetos do INPE relacionados com atividades e técnicas de V&V.

	ATIFS	COMAV	GTSC	PLAVIS	QSEE	SATCS
Revisões		X			X	X
Teste estrutural		X				X
Geração de teste a partir da especificação		X			X	X
Geração automática de Teste a partir de modelo de estados	X		X	X	X	
Injeção de Falhas	X				X	
Execução automatizada de teste	X	X			X	
Processo de V&V	X	X			X	X
V&V Independente					X	
Metodologia de geração de teste	X		X		X	
Gerenciamento automatizado de teste						X

Revisões e testes são técnicas de V&V regularmente adotadas nos projetos de software espacial desenvolvidos no INPE. O teste caixa-branca ou teste estrutural e a integração incremental do software são os tipos de teste mais comumente aplicados nos sistemas de software desenvolvimentos na ETE. Os testes de sistema, nos quais os casos de teste são criados a partir de uma especificação textual são igualmente aplicados no software para voar ou controlar remotamente satélites.

Os projetos de pesquisa como o ATIFS, o PLAVIS, o QSEE, o GTSC exploram o teste caixa-preta e a geração automática de teste baseado em modelo de estados (FSM, EFSM e Statecharts). A ferramenta Condado [17] que gera casos

de teste automaticamente a partir de um modelo em EFSM tem sido aplicada (usada ou estendida) nos projetos citados neste parágrafo. A técnica de injeção de falhas incorporada como complemento para testes baseados em EFSM, proposta inicialmente em [16] é totalmente pertinente em aplicações espaciais como forma de avaliar o comportamento do software frente à falhas de hardware causadas pela radiação do ambiente espacial e em função da criticidade da aplicação. Esta combinação tem sido aplicada tanto na concepção de uma arquitetura de teste [12], como na metodologia de teste COFI [9].

Ferramentas para apoio à execução de testes são sempre implementadas com maior ou menor formalização. No projeto ATIFS foi proposta uma ferramenta para execução de teste de sistemas de comunicação com forte apelo ao reuso [5]. No projeto QSEE, a ferramenta QSEE-TAS [23] transmite e recebe comandos para software embarcado em computadores de bordo de satélite e utiliza um protocolo proprietário do INPE para comunicar com outros computadores do satélite através de interfaces RS-232 ou USB sendo capaz de gerar relatos automatizados dos testes executados. Esta ferramenta poderá ser utilizada para teste de uma família de projetos espaciais que usem o mesmo protocolo. No SATCS, ferramentas para executar os testes são produzidas ao longo do desenvolvimento para todos os elementos de software dos vários níveis estabelecidos. Tais ferramentas são classificadas como: simuladores, stubs e guiadores.

Com o envolvimento nos projetos de pesquisa que exploraram o uso de ferramentas para geração automática a partir de especificações formais (como ATIFS e o PLAVIS), percebemos que não bastam as ferramentas, mas é preciso orientar o testador a criar as especificações formais, e convêm que estas especificações ou modelos sejam orientadas a objetivos de testes. Além disso, lidar com testes a partir de um único modelo do comportamento do software todo, quando este desempenha muitas funções distintas, não parece ser o caminho mais indicado. Sob esta motivação definimos a metodologia de teste CoFI, que é apresentada na seção 5.1.

Por outro lado, uma metodologia de teste não é suficiente quando se trata do desenvolvimento de sistemas complexos e grandes havendo necessidade de se estabelecer um Processo de V&V com atividades, documentos e regras para o bom gerenciamento de V&V nas várias fases do desenvolvimento. Em [10] foi definido um processo que lida com teste de conformidade e injeção de falhas. No

projeto QSEE o processo de V&V aplica normas de garantia de qualidade em aplicações espaciais baseadas na norma ECSS_Q_80 com marcos e revisões pré-estabelecidos [20]; além disso, procura estabelecer um procedimento próprio para validação independente para a fase de aceitação do software desenvolvido por terceiros. No projeto SATCS um processo de V&V que complementa o processo de desenvolvimento está sendo estabelecido. A documentação e os artefatos do projeto são criados como apoio de uma ferramenta comercial (Enterprise Architect) e uma ferramenta foi desenvolvida para gerenciamento e controle dos casos de teste criados e executados em cada fase de V&V. Este processo e a ferramenta para gerenciamento dos testes são apresentadas na seção 4.

3 PROJETOS DE PESQUISA EM V&V

Nesta seção uma descrição das principais características e uma discussão das vantagens e desvantagens das técnicas exploradas nos projetos de pesquisa correntes no INPE é apresentada.

3.1 ATIFS - Ambiente de Teste com Injeção de Falhas por Software

A inovação do projeto ATIFS é combinar teste de conformidade com injeção de falhas por software [18]. O projeto consta de um conjunto de ferramentas (em protótipos) integradas para apoiar as seguintes atividades de teste: geração automática, execução e análise de resultados. A figura 1 mostra a Arquitetura do Projeto ATIFS.

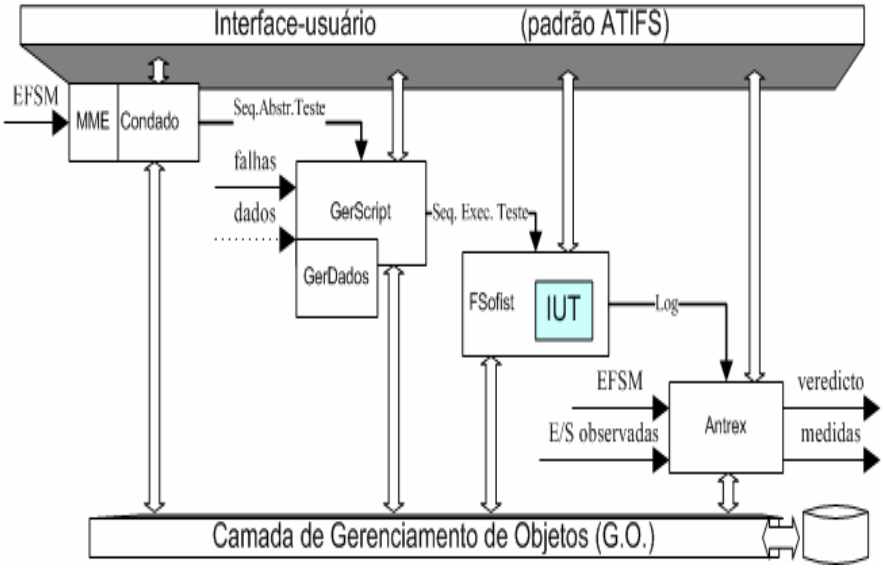

Figura 1. Arquitetura do Projeto ATIFS.

Levando em conta a reprodutibilidade dos testes, as seguintes ferramentas foram construídas mostrando a viabilidade da combinação das técnicas [15]:

Condado – (Controle e Dados) implementa o aspecto de teste de conformidade a partir de uma especificação formal, gerando casos de teste automaticamente a partir de Extended Finite State Machine (EFSM ou FSM), através de um algoritmo de busca em profundidade que percorre caminhos que começam e terminam no estado inicial, permitindo a detecção de falhas de saída [17]. A EFSM deve ser fortemente conexa. O critério de teste é cobrir todas as combinações de entradas. O editor de Modelos de Estados (MME) estende a ferramenta Condado com interface gráfica para edição da EFSM. O testador pode selecionar os estados ou caminhos que desejar cobrir os testes a fim de evitar explosão no número de casos de teste gerados. Evoluções desta pesquisa têm sido na aplicação de algoritmos evolutivos para se encontrar os melhores caminhos que representem casos de testes efetivos em menor número [7], [8].

GerDados – gera parâmetros para as entradas de uma transição. As entradas são descritas em uma linguagem similar à ASN.1. Dois tipos de testes são usados: teste de sintaxe (para gerar formatos válidos das transições) e testes de partições de equivalência (para gerar valores de dados). Os valores são escolhidos aleatoriamente dentro do domínio de entrada de cada parâmetro, conforme consta da especificação. Valores inválidos são cobertos pela técnica de injeção de falhas [29].

GerScript – faz a ligação entre os testes de conformidade e o planejamento dos experimentos de injeção de falhas a serem realizados. O testador escolhe as falhas, o padrão de repetição das mesmas e o momento de injetá-las. A seqüência de testes produzidas pela Condado (chamada Seqüência Abstrata de Teste em analogia com o padrão ISO9646) é acrescida das falhas. A seqüência abstrata pode ser editada, assim, novos testes podem ser criados manualmente. Por fim a seqüência é transformada em um script executável, o qual é entrada para a ferramenta FSofist.

Fsofist – implementa uma arquitetura baseada na arquitetura ferry-clip (para protocolos de comunicação) à qual, injetores de falhas de comunicação foram incluídos, visando a máxima independência entre o Sistema de Teste e o Sistema em Teste [12]. Permite comunicação ponto-a-ponto e multiponto para atender teste de computadores de bordo [5], [24]. Lê e executa o script executável. Armazena as entradas fornecidas e as respectivas saídas observadas.

Antrex - analisa automaticamente o resultado de cada caso de teste, comparando os caminhos da mesma EFSM fornecida à Condado com os caminhos da EFSM gerada pela Fsofist quando o caso de teste foi executado. É responsável pelo veredicto de conformidade e pelas medidas estatísticas obtidas pela análise do traço de execução [26]. O traço de execução compreende: as interações (entradas e saídas) observadas; o histórico dos testes realizados, contendo a identificação dos testes, dos métodos e dos resultados; e as informações de auxílio ao diagnóstico das falhas encontradas.

Um processo e uma metodologia de teste que acomoda as facilidades propostas pelas ferramentas do ATIFS foram propostos em [4]. Futuramente planeja-se trabalhar na re-utilização de projeto de testes para aplicações espaciais (test design) com base no crescente investimento em padrões de serviços para aplicações espaciais verificados nos trabalhos do Consultative Committee for Space data Systems (CCSDS) e no European Cooperation for Space Standardization (ECSS) [11].

3.2 QSEE – Qualidade de Software Embarcado em Aplicações Espaciais

O projeto QSEE promove uma experiência no INPE em realizar um processo de outsourcing no desenvolvimento de software embarcado em computadores de cargas úteis de satélites [22]. Neste projeto, o INPE atua como cliente e o fornecedor é uma empresa brasileira de desenvolvimento de software avaliada formalmente com nível de maturidade CMMI-3 e MPS.BR-A. A empresa, DBA Engenharia de Sistemas, não possuía experiência em desenvolvimento de software para aplicações espaciais, sendo o seu mercado principal produtos e serviços na área de tecnologia da informação. O QSEE usa as normas ECSS para guiar o relacionamento entre cliente e fornecedor [20]. Este projeto constitui a primeira experiência real do INPE em interagir com uma empresa brasileira para o fornecimento de software para computadores de bordo de cargas úteis de satélite e, por ser um projeto de transferência de tecnologia, explora abordagens inovadoras como V&V independente no desenvolvimento de software crítico na área espacial. A Figura 2 mostra uma visão geral do projeto de relacionamento entre os “stakeholders” do QSEE [19].

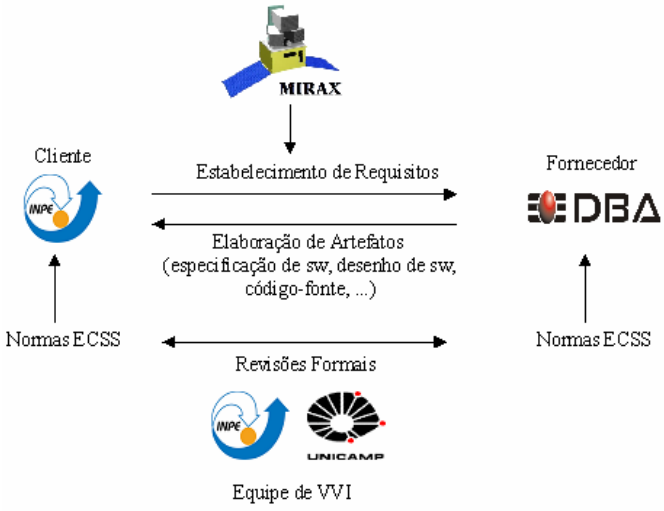

Figura 2. Visão geral do QSEE.

O software piloto desenvolvido e validado neste projeto, denominado SWPDC (do inglês, SoftWare embedded into the Payload Data Handling Computer), será embarcado na carga útil da missão de satélite de astronomia Monitor e Imageador de Raios X (MIRAX). O MIRAX está em desenvolvimento no INPE em cooperação

com instituições dos EUA, Alemanha e Holanda e objetiva monitorar uma larga região ao redor do plano Galático central para observação de Raios X.

Um dos objetivos do projeto QSEE é estabelecer para o INPE, um procedimento de validação e aceitação de software desenvolvido por terceiros. Para tanto existe uma equipe de Verificação e Validação Independente (VVI), formada pelo INPE e a UNICAMP, que aplicou a metodologia de teste COFI e a ferramenta Condado/MME para geração de testes de software (VVI). Os casos de testes foram aplicados como parte do processo de aceitação do software desenvolvido pela DBA. Detalhes da metodologia e resultados obtidos são apresentados na seção 5.

Para apoiar a execução e controle dos testes, as seguintes ferramentas foram implementadas:

QSEE-TAS (Qualidade do Software Embarcado em Aplicações Espaciais – Teste Automatizado de Software) foi projetada para execução de casos de testes e geração de relato de testes de software embarcado em experimentos de satélites científicos. De uma forma simplificada, a QSEE-TAS permite que as funcionalidades do experimento científico ou tecnológico sejam validadas de forma remota por meio do canal de comunicação real de operação em vôo. As principais funcionalidades da QSEE-TAS são:

- especificação de itens de teste1, casos de teste e passos de teste;
- codificação de mensagens de protocolos de comunicação customizáveis;
- comunicação por meio de canais RS-232, USB e TCP/IP;
- execução de casos de teste com veredicto semi-automático;
- inclusão de dados com falhas, controle de tempo para transmissão de mensagens;
- relatórios de especificações de teste e relato de teste, por sessão de teste, customizáveis via XSLT.

SPAC (Processamento e Análise de dados Científicos) funciona como um módulo da QSEE-TAS. Ele permite a extração de dados científicos capturados durante a execução dos testes e a visualização, por exemplo, na forma de espectros de energia. Sua arquitetura permite a inclusão de novos módulos que

[1] Aqui, Item de Teste tem o significado de conjunto de casos de teste.

podem estendê-lo de acordo com as necessidades específicas do experimento científico em questão. As principais funcionalidades da SPAC são:

• manipulação de canais digitais e analógicos;
• recuperação dos resultados da execução dos casos de testes;
• extração de dados de acordo com o tipo de informação (dados científicos, de diagnóstico, de teste, de descarga de memória e de housekeeping);
• visualização de dados científicos, por meio de espectros de energia, e visualização de temperaturas e relatos de eventos.

EPPSimulator (Simulador de EPP) simula os processadores front-end que lidam diretamente com os eventos detectados pelas câmeras de Raios X do MIRAX. O EPPSimulador pode ser comandado remotamente pela QSEE-TAS (TCP/IP), ou mesmo usando um cliente Telnet. Seu uso tem se provado essencial na validação do Software SWPDC, uma vez que é possível simular condições de comportamento anormal do sistema por meio de injeção de falhas por software.

3.3 GTSC - Geração de Teste a partir de StateCharts

O objeto de desenvolvimento deste projeto é uma ferramenta que permitirá ao desenvolvedor de software modelar software complexo em diagramas statecharts da UML e, a partir dessa modelagem, possibilitará a geração de casos de testes de acordo com critérios de adequação de testes tais como: todas as configurações, todas as transições, todos os caminhos simples, todos os caminhos livres de laços e todos os caminhos-k-configurações.

Como statecharts permitem uma representação mais realista de software de sistemas reativos complexos, com características como hierarquia e paralelismo, este projeto pressupõe que o desenvolvedor de software modele todo o sistema em statecharts e, a partir desse modelo, a geração de casos de testes é realizada de acordo com os critérios escolhidos.

As seguintes ferramentas estão sendo usadas no escopo do GTSC:

PerformCharts - (Performance and Statecharts) é uma ferramenta que foi desenvolvida no LAC para determinar medidas de desempenho de um sistema reativo especificado em statecharts [27], [28]. O funcionamento da PerformCharts é como descrito a seguir. Uma vez que um sistema reativo é especificado em

statecharts, o primeiro passo é converter a especificação para uma cadeia de Markov. A cadeia de Markov é resolvida por métodos numéricos gerando as probabilidades limite que já são medidas de desempenho [2]. Estas probabilidades correspondem à porcentagem de tempo ocupado por cada estado do sistema especificado. A cadeia de Markov tem uma correspondência com uma Máquina de Estados Finita (MEF) cuja visualização é obtida por meio de diagrama de estados e transições. Em uma primeira abordagem do projeto GTSC, a PerformCharts foi integrada à ferramenta Condado [17]. A PerformCharts gera somente a MEF e esta MEF alimenta a ferramenta Condado, a qual gera os casos de teste. Outros métodos de geração de casos de teste serão incorporados no projeto GTSC.

O ambiente GTSC está em desenvolvimento e integrará uma série de ferramentas, de modo que o desenvolvedor de software possa ter um ambiente de modelagem comportamental de software em Statecharts para gerar casos de teste. O GTSC tem usado duas especificações de software para computadores de experimentos científicos de satélites em desenvolvimento no INPE como estudo de caso [21].

4 EXPERIÊNCIAS COM V&V NOS PROJETOS COMAV e SATCS

O COMAV e o SATCS são projetos de pesquisa e desenvolvimento tecnológico internos do INPE, cujo objetivo é investigar/experimentar o potencial de novos paradigmas tecnológicos antes de serem aplicados as missões espaciais do INPE.

4.1 COMputador de Bordo AVançado (COMAV)

A crescente disponibilidade de processadores com maior capacidade de processamento e de outros recursos de hardware (memória, componentes lógicos programáveis, etc.) para utilização em ambiente espacial faz com que muitas das

operações anteriormente realizadas em solo passem a serem feitas em bordo. Além disso, novas aplicações estão continuamente sendo incorporadas no processamento de bordo, produzindo sistemas cada vez mais complexos. Neste cenário, as atividades de V&V de software ganham cada vez mais importância.

O projeto COMAV tem como objetivo desenvolver um sistema computacional a ser utilizado em futuras aplicações espaciais do INPE, que incorpore novas tecnologias, seja em termos de metodologia e processos de desenvolvimento, seja em termos de componentes e soluções.

As principais tarefas relacionadas a V&V de software do projeto COMAV são:

- aplicação das normas (ECSS) e ferramentas de auxílio ao processo de engenharia de software;
- aplicação de testes de módulos, integração e subsistema (caixa branca e caixa preta);
- desenvolvimento de um equipamento de testes do computador de bordo que facilite a execução e permita automação dos testes.

4.2 Sistema de Controle de SATélites (SATCS)

O SATCS foi projetado para ser facilmente configurável e customizado para atender diferentes satélites. Possui um núcleo de controle que proverá as principais funcionalidades para suportar o planejamento, a monitoração, o controle e a avaliação da performance do satélite levando em conta as necessidades de operação de cada missão espacial [14].

O SATCS é orientado a objetos e está estruturado em uma arquitetura composta por camadas: Software Básico, Aplicação e Automação. A camada de Software Básico provê funções básicas como comunicação de dados, tratamento de eventos de tempo, conversão de formatos de tempo, modelagem de estruturas de dados por meio de metadados. A camada Aplicação concentra todas as aplicações para controlar um satélite, como por exemplo: telemetria, telecomando, medidas de distâncias, dinâmica de vôo e controle e monitoração da estação terrena. A camada de Automação provê a automatização de atividades de operação dos satélites. O propósito desta camada é automatizar as tarefas rotineiras de operação do satélite, visando reduzir horas de trabalho dos

operadores e aumentar a confiabilidade da operação, uma vez que a probabilidade de um controlador cometer erros na execução de tarefas repetitivas é comprovadamente maior.

4.2.1 Processo e Técnicas de V&V

O processo de V&V estabelecido para desenvolvimento do SATCS visa permitir construir um sistema com qualidade e confiabilidade e de fácil aplicação para cada versão customizada. Este processo levou em conta a experiência adquirida pelo grupo, ao longo de quase vinte anos, no desenvolvimento desse tipo de sistema [1], [3].

As atividades de V&V são executadas em paralelo ao longo de todas as fases de desenvolvimento do software, da especificação de requisitos do sistema até as linhas de código da unidade de software. O sistema é decomposto em subsistemas, cada subsistema em módulos e cada módulo em classes. O processo adota três métodos de verificação: revisões, acompanhamento da rastreabilidade de requisitos e testes.

As revisões são feitas no final de cada fase do processo de desenvolvimento. Somente nas fases referentes ao sistema, os revisores são externos. Nos níveis internos do sistema, ou seja, nas fases de desenvolvimento dos subsistemas, as revisões são feitas pelo pessoal interno ao grupo de desenvolvimento. Os clientes do sistema participam de todas as revisões.

A rastreabilidade dos requisitos cobre os aspectos: Requisitos versus Requisitos; Requisitos versus Projeto e Requisitos versus Testes. A verificação “Requisitos versus Requisitos” tem a finalidade de demonstrar que todos os requisitos de nível superior estão sendo cobertos pelos requisitos de nível inferior e que todos os requisitos de nível inferior têm uma origem em requisitos do nível superior. A verificação “Requisitos versus Projeto” objetiva comprovar que todos os requisitos estão cobertos na solução de projeto e se cada elemento do projeto atende a pelos menos um requisito. A verificação ”Requisitos versus Testes” deve comprovar se foram definidos casos de testes para todos os requisitos e se cada caso de teste está relacionado à pelo menos um requisito.

Os testes são planejados e executados em todas as fases de desenvolvimento do sistema. O planejamento dos testes é realizado durante a especificação do

produto em fase do desenvolvimento. Deste modo, o planejamento segue uma estratégia “top-down”. Em contra partida, a execução dos testes e o registro dos resultados são feitos em uma estratégia “bottom-up”, à medida que os produtos previstos para cada fase são concluídos.

4.2.2 Controle e organização dos testes

Algumas regras foram estabelecidas para reduzir o esforço de projetar e executar os testes:

- definir uma estrutura comum de diretórios para todas as máquinas de desenvolvimento;
- manter em uma das máquinas, manipulada somente pelo gerente de configuração, as unidades de software seguras e as respectivas ferramentas de teste. Uma unidade de software é segura somente quando os resultados dos testes forem considerados satisfatórios pelo gerente do projeto;
- padronizar os nomes das unidades de software: classes, programas executáveis, ferramentas de teste, arquivos de dados, bibliotecas, guiadores, simuladores, “stubs”, etc, utilizando a sigla do subsistema mais uma identificação do tipo da unidade;
- estabelecer um conjunto mínimo de informações a serem documentadas sobre os testes dos elementos de níveis mais baixos (módulos e classes), já que o executor dos testes é o próprio desenvolvedor do ambiente de testes;
- definir uma estratégia de teste juntamente com a estratégia de desenvolvimento, de forma a reduzir o número de ferramentas de teste (guiadores, simuladores e “stubs”) a serem desenvolvidos para cada elemento de software e reusar seqüências de teste (conjunto de casos de testes) já especificadas em fases anteriores. Por exemplo, após o projeto das classes de um módulo a estratégia de desenvolvimento das mesmas deve seguir os passos: (i) Identificar as classes folhas a serem desenvolvidas primeiramente; (ii) Agrupar as classes criando componentes de testes básicos (pacotes de classes) usando como critério de agrupamento a coesão funcional. (iii) Estabelecer uma ordem de desenvolvimento dos mesmos levando em conta a dependência entre os pacotes. Assim podem ser definidos quais pacotes devem ser desenvolvidos de forma seqüencial ou podem ser desenvolvidos em paralelo. As ferramentas de

testes a serem utilizadas como guiadores e “stubs” já são implementadas durante o desenvolvimento das classes pelo desenvolvedor.

4.2.3 Sistema de Controle de Casos de Testes (SCCT)

O SCCT, desenvolvido “in-house”, é uma ferramenta de apoio à organização e documentação de testes nas várias fases do desenvolvimento. O SCCT permite cadastrar elementos de software de diferentes níveis: sistema, subsistema, módulos e classes; e estabelecer uma hierarquia entre eles. Para cada elemento associa-se as etapas de teste, o ambiente de teste, as ferramentas de testes e os casos de testes. Cada etapa de teste e caso de teste recebe um código com base no elemento de software do qual eles fazem parte. Esta ferramenta também permite registrar os resultados das execuções dos casos de testes e gerar relatórios da situação dos vários elementos de software em relação a cobertura de execução dos testes.

Dois tipos de testes são obrigatórios para todos os elementos de software: teste de integração e teste do elemento como uma caixa preta. Pelo menos uma etapa de teste deve ser associada a cada um destes tipos de teste. Os testes de sistema (validação e aceitação) são obrigatórios para os elementos do tipo sistema e do tipo subsistema que sejam utilizados por outros subsistemas. Os testes de unidade são aplicados somente às classes que não dependem de outras classes.

5. O PROCESO DE VVI no projeto QSEE

Uma das metas do projeto QSEE era estabelecer um procedimento para aceitação de software desenvolvido por terceiros.

A primeira atividade do processo VVI foi elaborar um Plano de VVI contendo, entre outros a definição dos três níveis de integração do software embarcado requeridos: instrumento, sub-sistema e sistema; visando conscientizar o fornecedor de que a entrega só se completa após o último nível de integração estar validado pelo cliente.

Um plano e uma especificação de teste foram criados para cada nível de integração (IVPlanIns, IVSpecIns, IVPlanSub, IVSpecSub, IVPlanSis, IVPlanSis).

Atividades de Revisões formais foram aplicadas no final de cada fase: especificação de requisitos, projeto preliminar, projeto detalhado, e entrega do produto de software, com participação de pessoas externas na banca para revisão dos documentos.

Como parte do produto do SWPDC, o fornecedor entregou o código, os documentos de projeto e manual de operação junto com documentos Plano de Teste do Software e Relatos dos Testes do Software executados por ele. Os casos de teste recebidos foram repetidos pelo cliente, e um primeiro conjunto de Relatório de Não-Conformidade (RNC) foi produzido.

A geração de testes realizada pela equipe de VVI aplicou a metodologia CoFI para cada um dos níveis de integração Para concepção dos casos de teste para esta etapa, a equipe VVI contou com 2 pessoas mais a consultoria da Dra. Eliane Martins da UNICAMP.

5.1 Metodologia CoFI

A metodologia CoFI, definida em [4] sofreu algumas simplificações quando aplicada no projeto QSEE, um caso real completo. Os passos da metodologia aplicados no projeto QSEE são apresentados na Tabela 2.

Tabela 2. Passos da metodologia CoFI.

Passo	Descrição
p1	identificar os serviços oferecidos pela implementação em teste (do inglês, Implementation Under Test - IUT), para os quais os testes serão gerados. Associar estes serviços a propósitos de testes.
p2	identificar os usuários dos serviços e os meios físicos de comunicação usados pela IUT.
p3	identificar as entradas, as saídas e as variáveis operacionais relacionadas a cada serviço. Identificar a arquitetura de teste (as ferramentas disponíveis para execução dos testes e como estas se relacionam com a IUT) e os pontos de controle e observação (como e onde fornecer as entradas e observar as saídas).
p4	representar o comportamento normal (cenário base e alternativos) e as exceções especificadas (cenários excepcionais) de cada serviço em máquinas de estados.
p5	criar uma matriz de transição do cenários base, como primeiro passo para criar os cenários de caminhos furtivos. Este passo inclui: • transcrever o diagrama de estados gerado no passo 4 para uma matriz, • completar os campos vazios da matriz, sempre que possível.
p6	representar o comportamento do sistema frente a caminhos furtivos, representado em máquinas de estados as células que foram completadas na matriz, no passo anterior. Este passo permite levantar eventos possíveis que ocorrem em estados em que não são esperados.
p7	escolher o modelo de falhas físicas que representa as conseqüências que anomalias do ambiente podem causar no sistema em teste e identificar as falhas que podem ser executadas pelo sistema de teste disponível.
p8	representar o comportamento do sistema frente às falhas físicas.

Continua

Tabela 2 - Conclusão

	Este passo permite levantar eventos que vão exercitar os mecanismos de tolerância à falhas da IUT.
p9	gerar automaticamente casos de teste de conformidade a partir de cada uma das máquinas de estado criadas: normais, exceções especificadas, caminhos furtivos e tolerância a falhas.

É importante ressaltar que a ferramenta Condado foi usada para geração automática de casos de teste de conformidade a partir de modelos em FSM (p9). Contou-se com as soluções da ferramenta QSEE-TAS e EPPSimulators para a injeção de falhas por software. As falhas físicas importantes no ambiente espacial são: comunicação, memória e processador (p7). Os serviços identificados são listados na Tabela 3.

5.2 Resultados

Onze serviços foram identificados, para os quais foram criadas FSMs para os seguintes grupos de cenários: normais (Nor), exceções especificadas (Exc), caminhos furtivos (CF), falhas de comunicação (Co) e falhas de memória e processador (MPr). Os serviços e o número de especificações formais são mostrados na Tabela 3.

Tabela 3. Serviços X número de modelos criados em VVI no QSEE.

Serviços		FSMs					Total
		Nor	Exc	CF	Co	MPr	
S1	Inicializa	2	1	1	1	1	6
S2	Gera dados Científicos	2	2	1	1	1	7
S3	Gera dados de Housekeeping	3	3	3	1	1	11
S4	Gera dados de Teste	2	4	4	1	1	12
S5	Gera dados de Diagnóstico	2	4	4	2	2	14
S6	Descarga de Memória	5	3	5	2	1	16
S7	Muda modo de operação	1	0	0	0	1	2
S8	Carrega e executa Programa	1	5	4	3	2	15
S9	Sintaxe de msg OBDH	1	0	0	1	0	2
S10	Sintaxe de msg EPP	1	0	0	1	0	2
S11	Comandos especiais	4	0	0	2	4	10
Total		24	24	22	13	14	97

Os casos de teste gerados foram aplicados no software desenvolvido pelo fornecedor DBA. Os erros detectados por tipo de falha são mostrados na Tabela 4. Outros 319 casos de teste gerados a partir de modelos do comportamento Normal (Nor) e das Exceções especificadas (Exc) resultaram somente 12 erros [6]. Uma FSM combinando o comportamento dos vários serviços foi também criada.

Tabela 4. Experimentos de Injeção de falhas x erros detectados.

Tipos de Falhas	Experimentos de injeção de Falhas	Erros Detectados
Comunicação	283	31
Processador	80	5
Memória	88	3
Total	451	39

Os resultados mostraram que a CoFI foi capaz de identificar um número significante de erros com um número relativamente pequeno de casos de teste. Os dados sugerem também que a metodologia CoFI foi mais efetiva para detectar prováveis erros quando o foco foi em injeção de falhas do que em conformidade [6].

6. CONCLUSÕES E TRABALHOS FUTUROS

Lições aprendidas com VVI no QSEE

Com uma equipe independente para produzir os casos de teste, os testes são mais efetivos, pois o propósito é encontrar erros ao invés de provar que o sistema está correto. Percebemos neste projeto, que o VVI consolidou, logo no início do desenvolvimento, as regras e exigências do cliente para efetuar a aceitação do produto mesmo ao longo do desenvolvimento, garantindo o compromisso de toda a equipe do projeto com a qualidade de cada entrega.

O uso da metodologia CoFI se mostrou positivo pelo fato de tratar classes de erros e lidar com cada classe de uma vez; o uso de modelos formais permite expressar precisamente o objetivo do teste, facilitando dirimir dúvidas quando a interpretação do caso de teste que causa polêmica. Muitos casos de teste levaram a combinações de eventos não pensadas pelos desenvolvedores. Uma vez que a metodologia baseia-se na documentação fornecida para a criação dos modelos formais, inconsistências na documentação também são detectadas, assim não apenas o código é validado mas também, a consistência da informação fornecida nos documentos que acompanham o produto de software.

Lições aprendidas com Projeto ATIFS

Projetos de pesquisa e desenvolvimento em cooperação com a universidade permitiu ao INPE, manter-se atualizado no estado da arte dos métodos e técnicas de V&V com reduzido investimento. Por outro lado, a universidade encontra no INPE motivação e desafios de trabalhos práticos. A combinação de testes de conformidade com uso de especificações formais e injeção de falhas é vantajosa para aplicações espaciais. O uso de padrões e reutilização em teste é outra vantagem dos conceitos do ATIFS e uma tendência nos projetos espaciais [5].

Lições aprendidas com SATCS

A experiência prática tem confirmado que uma grande parte do tempo de desenvolvimento de software é gasta com a preparação e execução dos testes. O processo V&V adotado para o projeto SATCS tem permitido o desenvolvimento dos vários elementos de software de uma forma mais confiável. A estratégia de iniciar os testes pelas folhas e ir crescendo tem gerado bons resultados, permitindo que erros sejam encontrados nas fases iniciais onde o impacto é menor. A tendência é querer fazer tudo e depois testar, mas a experiência mostra que se economiza tempo e se obtém produtos de maior qualidade quando se realiza uma boa tarefa de planejamento.

Há carência de ferramentas automatizadas de geração de casos de teste, execução e análise dos resultados obtidos. Muitas vezes, é impossível realizar manualmente, principalmente os testes de regressão necessários a cada alteração. Por exemplo, uma mudança insuficientemente testada de um serviço na camada “Software Básico” do SATCS, pode acarretar efeitos colaterais, e problemas mais complexos nas camadas superiores, dificultando a identificação da sua origem. Isto seria evitado se todos os testes de regressão fossem devidamente aplicados.

Há carência de uma metodologia de teste para orientar a geração de casos de teste de sistemas que fazem uso intenso de meta-dados onde, o conhecimento está documentado na base de dados, muitos problemas detectados são oriundos do cadastramento incorreto dos dados.

Em geral, nos projetos de desenvolvimento de software do INPE, ferramentas de teste eram construídas sem a preocupação de serem re-utilizadas. Hoje esta preocupação é maior.

AGRADECIMENTOS

Gostaríamos de agradecer a equipe de software da DSS/INPE que tem colaborado no SATCS e a equipe da DEA/INPE envolvida no COMAV; ao empenho de todos os bolsistas do projeto QSEE e GTSC. Agradecemos também à Dra. Eliane Martins da UNICAMP pelas contribuições aos vários projetos de pesquisa em V&V junto ao INPE.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Ambrosio, A M.; Gonçalvez, L.S.C.; Cardoso, P.E. “Test Strategy For A Satellite Control System Developed According To An Object-Oriented Methodology”. In: Proceedings of the First Symposium on Data Collecting (SCDI 1999). Foz do Iguaçu – SP – Brasil, 1999.

[2] Andrade, V.M.B.; Carvalho, S.V.; Vijaykumar, N.L. Software Development for Analysis of Performance Systems using Markov Models. XXIX Brazilian Symposium of Operational Research , Salvador, BA, Brazil, October 1997.

[3] Ambrosio, A.M.; Gonçalvez, L.S.C.; Cardoso, P.E. - "An Experience in Testing an Object-oriented Satellite Control System" – 33rd Havaii International Conference on System Sciences 2000 (HICSS-33). Publicado em Software Quality Journal, Vol. 8, No.4, 271-283 - special issue on Distributed Systems Testing, edited by Anna Liu and Paddy Nixon - Kluwer Academic Publisher, 1999.

[4] Ambrosio, A. M. CoFI – uma abordagem combinando teste de conformidade e injeção de falhas para validação de software em aplicações espaciais. Tese de doutorado, INPE, São José dos Campos, 2005.

[5] Ambrosio, A M.; Martins,E ; Mattiello-Francisco, MF; Silva, C.S.; Vijaykumar, N.L. “On the use of test standardzation in communication space applications” – In: Proceedings of the 8th International Conference on Space Mission Operations - SPACEOPS 2004– Montreal, Canadá, Publisher: American Institute of Aeronautics and Astronautics (AIAA), May 2004.In: www.spaceops2004.org. In CD (ISBN 1-56347-708-4).

[6] Ambrosio, A.M.; Mattiello-Francisco, F.; Santiago, V. A.; Silva, W.P.; Martins, E. Designing Fault Injection Experiments using State-based Model to test a Space Software. Third Latin-American Symposium on Dependable Computing (LADC) 2007. To appear in a number of the series LNCS - Lecture Notes in Computer Science. Springer

[7] B. T. Abreu, E. Martins, and F. L. de Sousa. Generalized Extremal Optimization: an attractive alternative for test data generation. In Proc. of the Genetic and Evolutionary Computation Conference 2007. London, England, 2007.

[8] B. T. Abreu, E. Martins, and F. L. de Sousa. Generalized Extremal Optimization: a competitive algorithm for test data generation. In Proc. of the 21th Brazilian Symposium on Software Engineering, João Pessoa, Brazil, 2007. To appear.

[9] Ambrosio, A M.; Martins,E.; Vijaykumar, N.L.; Carvalho S.V. Systematic Generation of Test and Fault Cases for Space Application Validation. Proceedings of the 9th ESA Data System in Aerospace (DASIA), 30 Mai – 2 Jun. 2005, Edinburgh, Scotland. Noordwijk: ESA Publications, 2005.

[10] Ambrosio, A. M.; Martins, E.; Vijaykumar, N.L.; de Carvalho, S.V. A Conformance Testing Process for Space Applications Software Services. JACIC - Aerospace Computing, Information, and Communication, Vol.3, N.4, pp.146-158. Publisher American Institute of Aeronautics and Astronautics - AIAA, April 2006, USA.

[11] Ambrosio, A. M.; Martins, E.; Space services: textual to formal description. Proceedings of the 8th Conference on Space Operations (SpaceOps 2006), 19 Jun – 23 Jun. 2006, Rome, Italy. American Institute of Aeronautics and Astronautics - AIAA, 2006.

[12] M. R. R. Araújo. FSofist- Uma ferramenta para teste de protocolos tolerantes

a falhas. Dissertação de mestrado Instituto de Computação – Unicamp, out/2000.

[13] Cardoso, L.S.; Ferreira, M.G.V.; Orlando V. An Intelligent System for Automatic Flight Operation Plans Generation for to Satellite Control Activities at INPE. SpaceOps 2006

[14] [Cardoso, P.E.; Barreto, J.P.; Dobrowolski, K.M.; Cardoso, L.S.; Hoffmann, L.T. A Ground Control System for CBERS 3 and 4 Satellites. SPACEOPS – 2006

[15] Martins, E.; Ambrosio, AM.; Mattiello-Francisco, M.F. ATIFS: a testing tool set with software fault injection. Proceedings of York Computer Science Yellow Report 2003 - Workshop SofTest 2003: UK Testing Research II – Dept. of Computer Science at York, UK, 4-5 Sept 2003.

[16] Martins, E. ATIFS: um ambiente de Teste baseado em Injeção de Falhas por Software. Relatório Técnico DCC-95-24. Dezembro 1995. Departamento de Ciência da Computação - Universidade Estadual de Campinas.

[17] Martins, E.; Sabião, S.B.; Ambrosio, A.M. ConData: a Tool for Automating Specification-based Test Case Generation for Communication Systems. Software Quality Journal, Vol. 8, No.4, 303-319, edited by Anna Liu and Paddy Nixon - Kluwer Academic Publishers, 1999.

[18] Martins, E.; Mattiello-Francisco, M.F. A tool for fault injection and conformance testing of distributed systems. In Dependable Computing – First Latin American Sympsium, LADC 2003 São Paulo – Brasil, October 2003 Proceedings – Springer, Lectures Notes in Computer Science 2847 – Edited by G Goos, J Hartmanis, J van Leeuwen

[19] Mattiello-Francisco, M.F.; Santiago, V. A.; Ambrosio, A.M.; Costa, R.; Jogaib, L. Verificação e Validação na terceirização de software embarcado em aplicações espaciais. Anais do V Simpósio Brasileiro de Qualidade de Software, SBQS-2006 —29 Maio a 03 Junho 2006, Vila Velha - ES, Brasil. pp. 367- 374.

[20] Mattiello-Francisco, M.F; Santiago, V.; Ambrósio, A.M.; Jogaib, L.; Costa, R. A Brazilian Software Industry Experience in Using ECSS for Space Application Software Development. 14th ISPE International Conference on Concurrent Engineering (CE2007), 16 – 20 de Julho/2007, São José dos Campos, SP, Brazil. Editors: Geilson Loureiro and Richard Curran. Springer, 2007.

[21] Santiago, V.; Amaral, A.S.M.; Vijaykumar, N.L.; Mattiello-Francisco, M.F.; Martins, E.; Lopes, O.C. A Practical Approach for Automated Test Case Generation using Statecharts. Em: Second International Workshop on Testing and Quality Assurance for Component-Based Systems in the IEEE International Computer Software and Applications Conference, 2006, Chicago-EUA, p. 183-188.

[22] Santiago, V.; Mattiello-Francisco, F.; Costa, R.; Silva, W.P.; Ambrosio, A.M. QSEE Project: An Experience in Outsourcing Software Development for Space Applications. Em: The 19th International Conference on Software Engineering & Knowledge Engineering (SEKE), 2007, Boston-EUA, p. 51-56.

[23] Silva, W.; Santiago, V. ; Mattiello-Francisco, M.F.; Passos, D. QSEE-TAS: Uma Ferramenta para Execução e Relato Automatizados de Testes de Software para Aplicações Espaciais –XX Simpósio Brasileiro de Engenharia de Software - Sessão de Ferramentas, Outubro/2006, Florianópolis-SC. Porto Alegre-RS: SBC, 2006, pp. 43-48.

[24] Silva, C.S. Uso de Arquitetura Ferry na Validação de Sistemas Embarcados. Dissertação (Mestrado em Computação Aplicada). INPE, São José dos Campos, SP. 2005.

[25] Simão, A.S.; Ambrosio, A.M.; Fabri, S.C.P.F.; Amaral, A.M.S.; Martins E.; Maldonado, J.C. Plavis/FSM: an environment to integrate FSM-based testing tools. In: Simpósio Brasileiro de Engenharia de Software - Sessão de Ferramentas, 19. SBES 2005. 3-7 outubro 2005. Uberlândia, MG. Universidade Federal de Uberlândia.

[26] M.R.Stefani. “Análise de traço com geração de diagnósticos para teste de comportamento de uma implementação de protocolo de comunicação em presença de falhas”. Dissertação de Mestrado. Instituto de Computação – Unicamp, mai/1997.

[27] Vijaykumar, N. L. Statecharts: Their use in specifying and dealing with Performance Models. Tese de Doutorado. Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, Brazil, 1999.

[28] Vijaykumar, N.L.; Carvalho, S.V.; Abdurahiman, V. On proposing Statecharts to specify Performance Models. International Transactions in Operational

Research (ITOR) , 9(3), pp.321-336, May 2002

[29] Uber, F.R. “Integrating Domain Testing into Extended Finite State Machine Testing,” Master dissertation, Computing Institute – Campinas University, Brazil, Oct. 2001. (in Portuguese)

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.