



*ATIFS*

Eliane Martins  
Instituto de Computação - Unicamp  
eliane@ic.unicamp.br

6° Workshop do Projeto ATIFS - INPE - 5/12/2003



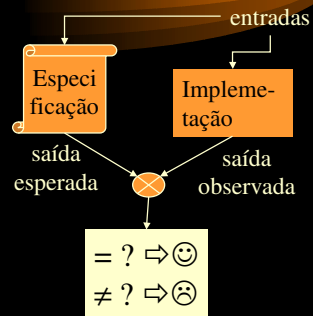
*O que é*

- Ambiente de *Testes e Injeção de Falhas* por Software (ATIFS)
  - projeto conjunto com o Instituto Nacional de Pesquisas Espaciais (INPE)
- Objetivos
  - desenvolvimento de técnicas de testes para sistemas reativos tomando por base o modelo de comportamento
  - desenvolvimento de ferramentas que apoiem as técnicas propostas
  - desenvolvimento de metodologias de teste com base nas ferramentas desenvolvidas

2

## ATIFS - Tipos de testes

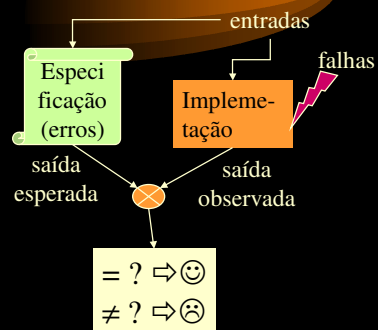
- Testes de conformidade:
  - verificar se o sistema foi implementado conforme o que foi especificado
- Porquê?
  - Importante determinar se o sistema executa as funções requeridas
  - ☞ funcionalidade



3

## ATIFS - Tipos de testes

- Injeção de falhas
  - introdução deliberada de falhas, de forma controlada
- Porquê?
  - a implementação não apresenta um comportamento inadequado em presença de falhas de outros componentes?
  - ☞ confiabilidade, disponibilidade



4

## Aspectos Pesquisados

- Foco dos estudos no ATIFS:
  - geração de testes a partir de modelo formal:
    - FSM, EFSM
  - apoio a execução de testes:
    - arquitetura distribuída de testes
    - injeção de falhas (comunicação)
  - análise de resultados dos testes:
    - FSM + injeção de falhas
    - EFSM (FSM + aspecto dados) + injeção de falhas
  - metodologias de teste
    - testes de conformidade + injeção de falhas

- explosão de estados  
- explosão do nº de casos de teste gerados  
- tratamento do aspecto dados  
- tratamento das falhas: como gerá-las automaticamente?

5

## Aspectos Pesquisados

- Foco dos estudos no ATIFS:
  - geração de testes a partir de modelo formal:
    - FSM, EFSM
  - apoio a execução de testes:
    - arquitetura distribuída de testes
    - injeção de falhas (comunicação)
  - análise de resultados dos testes:
    - FSM + injeção de falhas
    - EFSM (FSM + aspecto dados) + injeção de falhas
  - metodologias de teste
    - testes de conformidade + injeção de falhas

- independência da IUT  
- facilidade de expandir e adaptar a diferentes plataformas  
- sincronização entre testadores  
- localização do injetor

6

## Aspectos Pesquisados

- Foco dos estudos no ATIFS:
  - geração de testes a partir de modelo formal:
    - FSM, EFSM
  - apoio a execução de testes:
    - arquitetura distribuída de testes
    - injeção de falhas (comunicação)
  - análise de resultados dos testes com base no modelo formal
    - FSM + injeção de falhas
    - EFSM (FSM + aspecto dados) + injeção de falhas
  - metodologias de teste
    - testes de conformidade + injeção de falhas

- fornecer veredito + diagnóstico  
- consideração de aspectos: dados e falhas

7

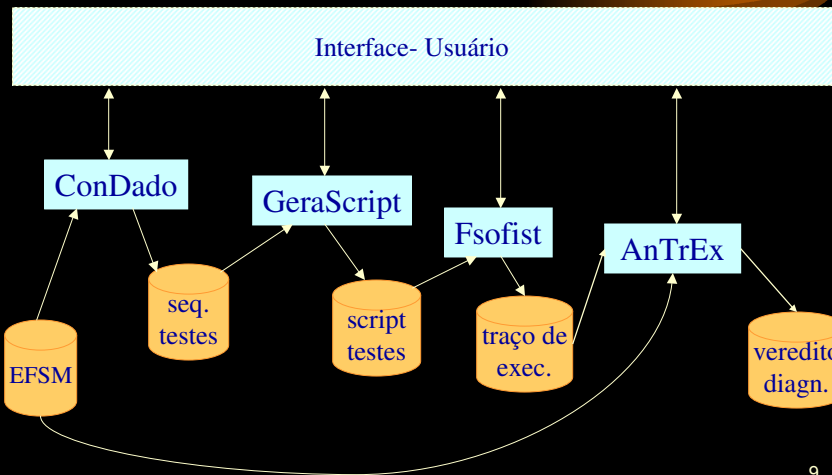
## Aspectos Pesquisados

- Foco dos estudos no ATIFS:
  - geração de testes a partir de modelo formal:
    - FSM, EFSM
  - apoio a execução de testes:
    - arquitetura distribuída de testes
    - injeção de falhas (comunicação)
  - análise de resultados dos testes:
    - FSM + injeção de falhas
    - EFSM (FSM + aspecto dados) + injeção de falhas
  - metodologias de teste
    - testes de conformidade + injeção de falhas

- estratégia de testes com base nas ferramentas existentes e que combine tanto os testes de conformidade quanto a injeção de falhas

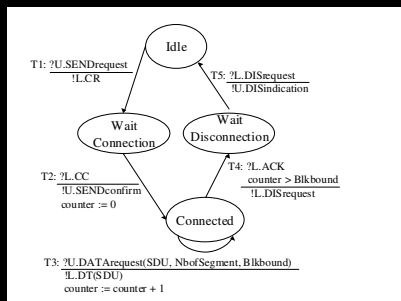
8

## Arquitetura do ATIFS



9

## Condado



Modelo de estados

### Sem restrições para a geração de testes:

- senddata (U, SENDrequest)
- recdata (L, CR)
- senddata (L, CC)
- recdata (U, SENDconfirm)
- senddata (U, DATArequest, "xHn\*e", 2, 14)
- recdata (L, DT)
- senddata (L, ACK)
- recdata (L, DISrequest)
- senddata (L, DISrequest)
- recdata(U, DISindication)

Casos de teste

10

## *ATIFS : ConDado (atual)*

- Geração de testes:
  - **controle:**
    - testes de transição de estados
    - pb da explosão do nº de testes: uso de restrições para reduzir nº de testes gerados
  - **dados:**
    - testes de sintaxe e testes de partição de equivalência

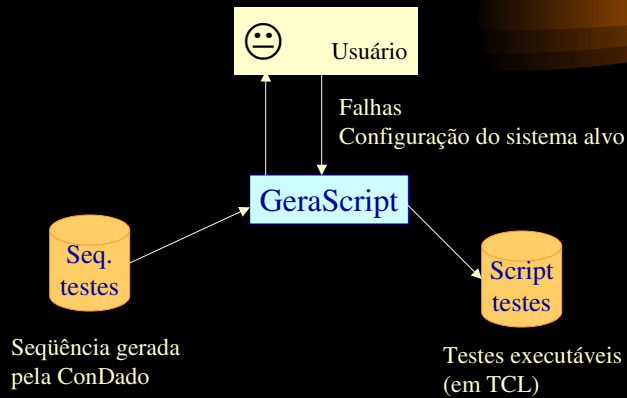
11

## *ATIFS : ConDado (perspectivas)*

- Integração com outras ferramentas:
  - uso de XML como formato de entrada
- Geração de testes:
  - **controle:**
    - implementar algoritmos mais eficientes e que forneçam outros critérios de cobertura
  - **dados:**
    - testes de partição de equivalência → testes de domínios (SeDados)
  - **falhas:**
    - é possível definir as falhas a partir do modelo?

12

## GeraScript



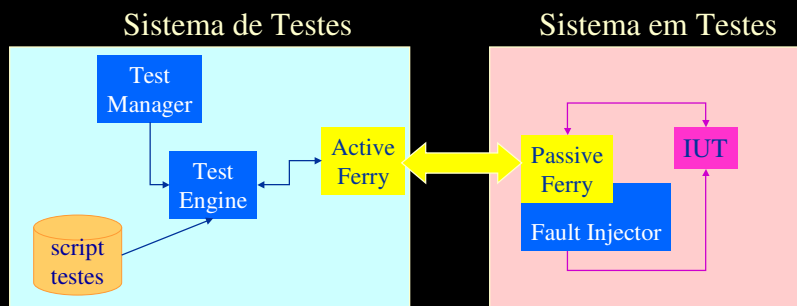
13

## GeraScript (perspectivas)

- Geração de testes executáveis em diversas linguagens:
  - uso de notação independente de linguagem (XML?)
  - usuário teria vários “tradutores” para os diferentes tipos de linguagem

14

- Execução de testes de sistemas de comunicação
- Arquitetura ferry-clip para testes de protocolos + capacidade de injeção de falhas



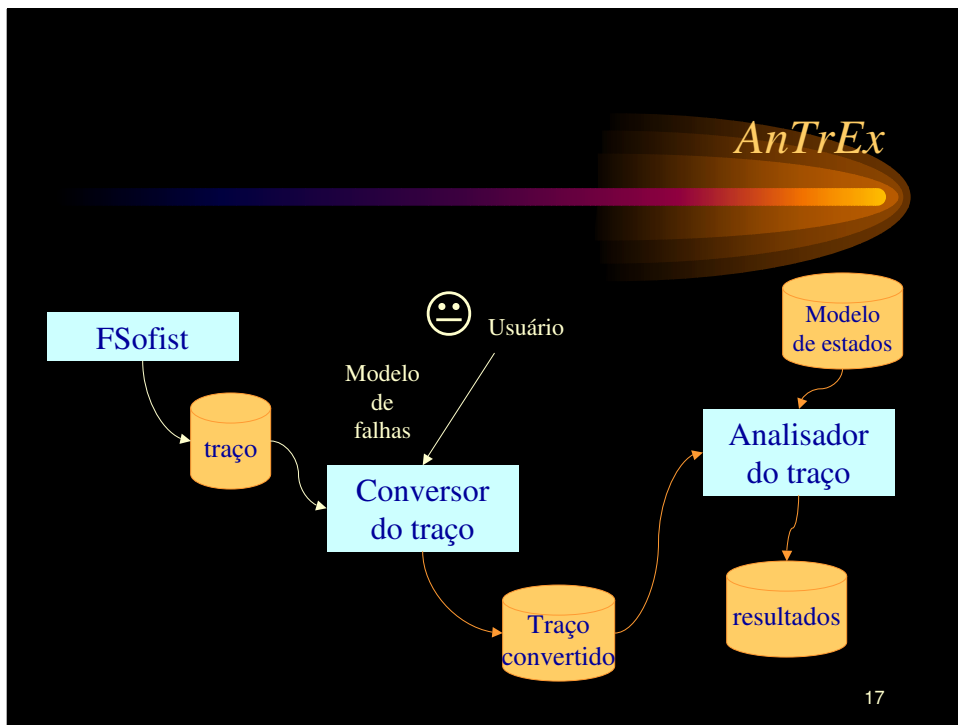
15

## Fsofist (perspectiva)

- Extensão para testes de IUT multiponto
  - experimento do computador de bordo
  - protocolo de conferência
- Apoio aos testes de interoperabilidade
- Avaliação em estudos de caso, medindo, por exemplo:
  - portabilidade
  - facilidade de expansão

16





- ## AnTrEx (perspectivas)
- Análise de traço cobrindo aspectos controle e dados
    - levar em conta parâmetros de interações
    - levar em conta as guardas das transições
  - Verificação de aspectos não funcionais:
    - tolerância a falhas
- 18

## *Metodologias de teste*

- Testes de aplicações Web
  - definição de estratégia e aplicação em estudo de caso
- Testes de sistemas espaciais
  - definição de metodologia para testes de sistemas espaciais que englobe tanto os testes de conformidade quanto os de injeção de falhas

19

## *Mais informações*

[www.inpe.br/~atifs](http://www.inpe.br/~atifs)

20