Model-Driven Simulation, Animation and Analysis

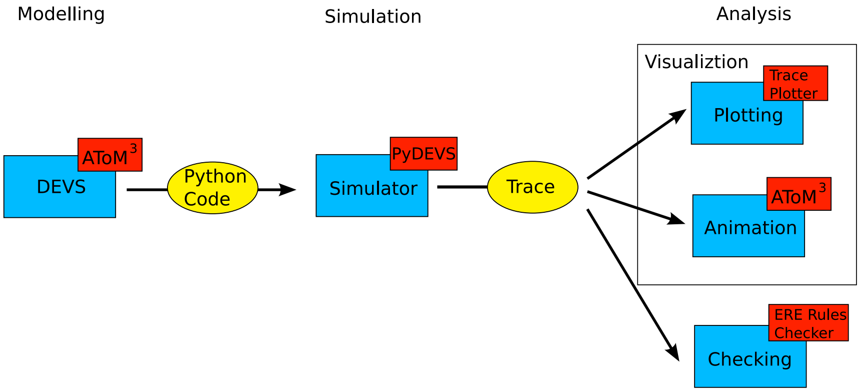
Ximeng Sun

MSDL - School of Computer Science
McGill University

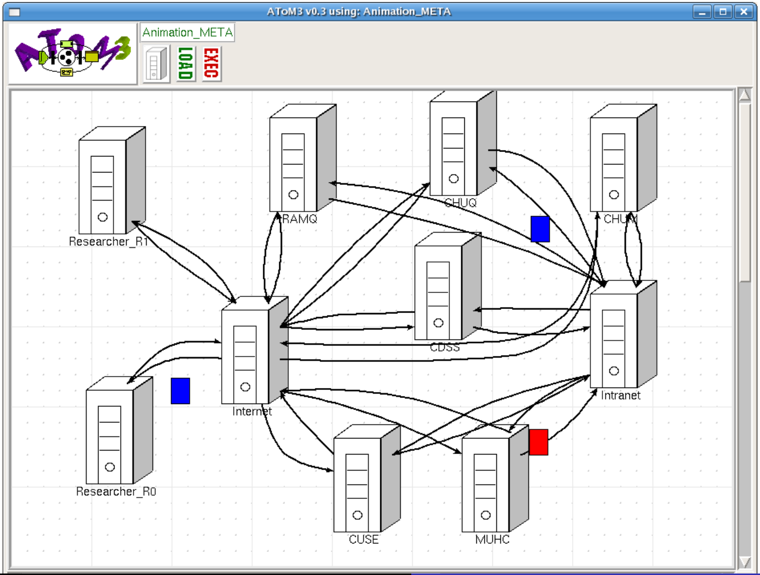
August 28, 2006

Outline

Description of The Simulation-based Approach



Demo


AToM3 v0.3 using: Animation_META
Animation_META
LOAD
EXEC
RAMQ
CHUQ
CHUM
Researcher_R1
CDSS
Internet
Intranet
Researcher_R0
CUSE
MUHC

Outline

Introduction

- Aim: Assessing and refining use cases to ensure that the specified functionality meets the dependability requirements of the system.
- Method:
  1. Mapping use cases to DA-Charts model;
  2. Perform probability analysis of the model using AToM^3.
- Details: S. Mustafiz, X. Sun, J. Kienzle, H. Vangheluwe. Model-Driven Assessment of Use Cases for Dependable Systems. *ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems*, October 2006, Genova, Italy.

Introduction

- Aim: Assessing and refining use cases to ensure that the specified functionality meets the dependability requirements of the system.
- Method:
  1. Mapping use cases to DA-Charts model;
  2. Perform probability analysis of the model using AToM^3.
- Details: S. Mustafiz, X. Sun, J. Kienzle, H. Vangheluwe. Model-Driven Assessment of Use Cases for Dependable Systems. *ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems*, October 2006, Genova, Italy.

Introduction

- Aim: Assessing and refining use cases to ensure that the specified functionality meets the dependability requirements of the system.
- Method:
  1. Mapping use cases to DA-Charts model;
  2. Perform probability analysis of the model using AToM^3.
- Details: S. Mustafiz, X. Sun, J. Kienzle, H. Vangheluwe. Model-Driven Assessment of Use Cases for Dependable Systems. *ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems*, October 2006, Genova, Italy.

Background

- Dependability: Property of a computer system such that reliance can justifiably be placed on the service it delivers.
    - Reliability: Measure a system's aptitude to provide service and remain operating as long as required.
    - Safety: Determined by the lack of catastrophic failures it undergoes.
    - Availability
    - Maintainability
    - Confidentiality
    - Integrity
- Fault tolerance: Means of achieving system dependability.
    - Error detection: Detection of exceptional situations
    - System recovery: Describing the interactions with the environment

Background

- Dependability: Property of a computer system such that reliance can justifiably be placed on the service it delivers.
  - Reliability: Measure a system's aptitude to provide service and remain operating as long as required.
  - Safety: Determined by the lack of catastrophic failures it undergoes.
  - Availability
  - Maintainability
  - Confidentiality
  - Integrity
- Fault tolerance: Means of achieving system dependability.
  - Error detection: Detection of exceptional situations
  - System recovery: Describing the interactions with the environment

Background

- Dependability: Property of a computer system such that reliance can justifiably be placed on the service it delivers.
  - Reliability: Measure a system's aptitude to provide service and remain operating as long as required.
  - Safety: Determined by the lack of catastrophic failures it undergoes.
  - Availability
  - Maintainability
  - Confidentiality
  - Integrity
- Fault tolerance: Means of achieving system dependability.
  - Error detection: Detection of exceptional situations
  - System recovery: Describing the interactions with the environment

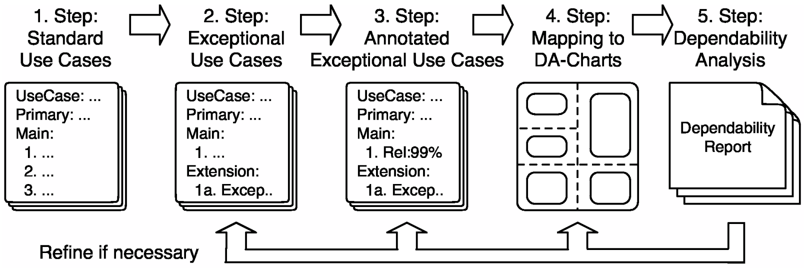
Background

- Dependability: Property of a computer system such that reliance can justifiably be placed on the service it delivers.
    - Reliability: Measure a system's aptitude to provide service and remain operating as long as required.
    - Safety: Determined by the lack of catastrophic failures it undergoes.
    - Availability
    - Maintainability
    - Confidentiality
    - Integrity
- Fault tolerance: Means of achieving system dependability.
    - Error detection: Detection of exceptional situations
    - System recovery: Describing the interactions with the environment

Background

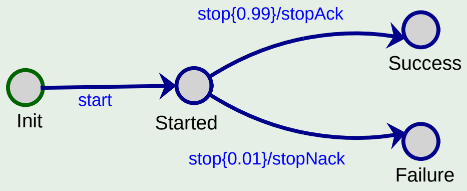
- Dependability: Property of a computer system such that reliance can justifiably be placed on the service it delivers.
    - Reliability: Measure a system's aptitude to provide service and remain operating as long as required.
    - Safety: Determined by the lack of catastrophic failures it undergoes.
    - Availability
    - Maintainability
    - Confidentiality
    - Integrity
- Fault tolerance: Means of achieving system dependability.
    - Error detection: Detection of exceptional situations
    - System recovery: Describing the interactions with the environment

Model-Driven Process for Assessment and Refinement of Use Cases

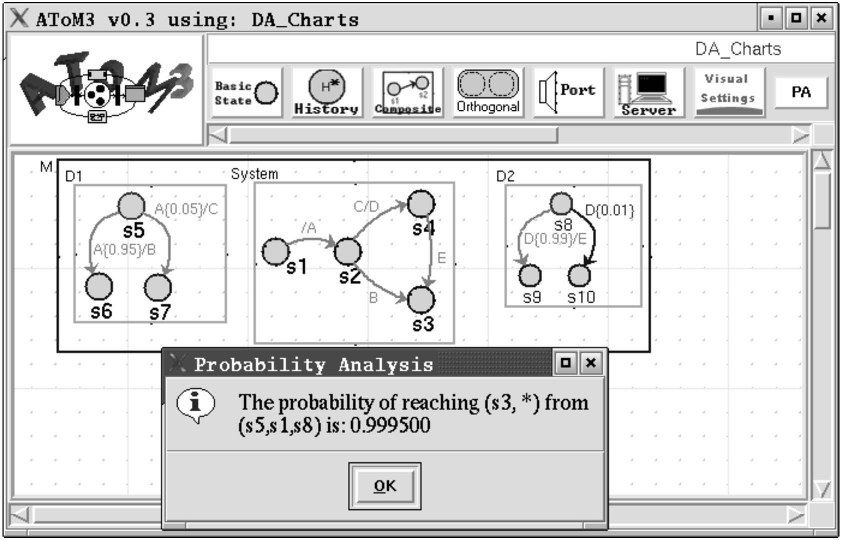


DA-Charts

- **D**ependability **A**ssessment **Charts**: Probabilistic extension of the Statecharts formalism.
- A state can transition to one of two possible target states: a *success* state with probability *p* and a *failure* state with probability *1-p*.
- Syntax: *event[condition]{probability}/action*



DA-Charts in *AToM*^{3}



The probability of reaching (s3, *) from (s5,s1,s8) is: 0.999500

Outline

Elevator Arrival Use Case

Use Case: ElevatorArrival

Main Success Scenario:

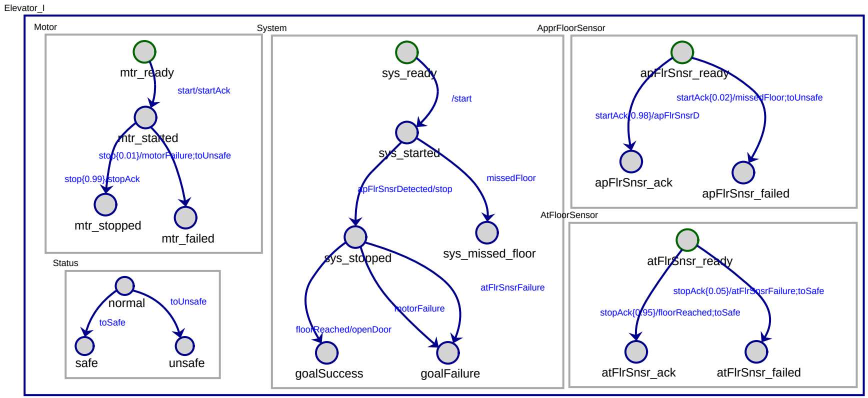
1. System asks motor to start moving towards the destination floor.
2. System detects elevator is approaching destination floor. Reliability:0.98 Safety-critical
3. System requests motor to stop. Reliability:0.99 Safety-critical
4. System receives confirmation elevator is stopped at destination floor. Reliability:0.95
5. System requests door to open.
6. System receives confirmation that door is open.

Extensions:

- 2a. Exception{MissedFloor}
- 4a. Exception{MotorFailure}
- 6a. Exception{DoorStuckClosed}

Elevator Arrival Use Case with Failures

DA-Charts Model



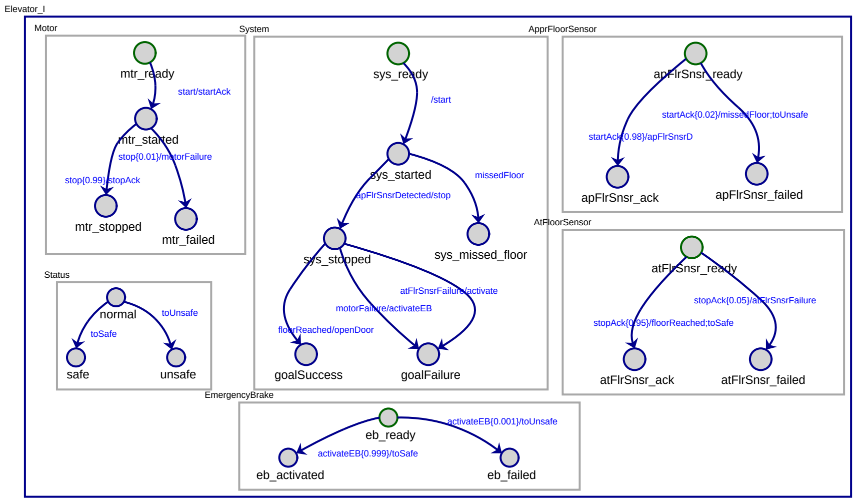
Elevator Arrival Use Case with Failures

Analysis

- Safety Analysis:
  - The system is unsafe if the approaching floor sensor fails to detect the destination floor, or if the motor fails to stop when told to do so.
  - The safe probability is caculated (reaching state *safe* from initial state *sys_ready*): **97.02%**
- Reliability Analysis:
  - Probability of reaching the *goalSuccess* state: **92.169%**

Elevator Arrival Use Case with Failures and Handlers

DA-Charts Model



Elevator Arrival Use Case with Failures and Handlers

Analysis

- Safety Analysis:
  - It's now safe **97.02%** of the time, with an increase of 0.9742%
  - Would be more safe if *missedFloor* exception would be detected and handled.
- Reliability Analysis:
  - No change
  - Could be refined to detect the failure of *AtFloorSensor*

Outline

DA-Charts Related

- Integrating hierarchy and history into DA-Charts
- Automate the process of mapping use cases to DA-Charts

Thank You!

Question?