FaTC2: An Object-Oriented Framework for Developing Fault-Tolerant Component-Based Systems

Fernando J. Castor de Lima Filho
Paulo Asterio de C. Guerra
Cecília Mary F. Rubira

{fernando, asterio, cmrubira}@ic.unicamp.br

ICSE 2003 – Workshop on Software Architectures for Dependable Systems
Motivation

- The construction of systems with high dependability requirements out of software components represents a major challenge
  - Few assumptions can be made about the level of confidence of off-the-shelf components
  - An architectural approach is required
- Exception handling is a well-known technique for leveraging the task of incorporating fault tolerance into software systems
Motivation (2)

- Component-based systems introduce challenges which are not addressed by traditional (language-based) exception handling systems (EHSs)
- Some of these challenges are:
  - Traditional EHSs lack support for attaching exception handlers to architectural elements (components, connectors, configurations)
  - In an architecture, exception propagation does not necessarily follow the method invocation chain
  - An architectural-level EHS should support the attachment of handlers to components without requiring modifications to them
Objectives

- To create an architectural-level EHS which leverages the construction of fault-tolerant component-based systems
- To devise a reusable implementation of the EHS by means of an object-oriented framework
  - Based on the concept of idealised fault-tolerant component
Idealised Fault-Tolerant Component
C2 Architectural Style

- We use the C2 architectural style in order to represent component-based systems
  - Integration of heterogeneous off-the-shelf components
- A C2 architecture is composed by components, connectors and interconnections
  - Layered
  - Elements in an architecture communicate by means of asynchronous messages
  - Each component may have its own control thread
- Tools which support the development of C2 applications:
  - ArchStudio
  - C2.FW framework
A Simple C2 Architecture

Component 1

Requests

Connector 1

Component 2

Connector 2

Component 3

Component 4

Notifications
Overview of FaTC2

- An extension of the Java™ version of the C2.FW framework
  - C2.FW lacks support for fault tolerance.
- Introduces forward error recovery in the original framework by means of an EHS

\[ \text{C2.FW} + \text{Exception Handling System} = \text{FaTC2} \]

- FaTC2 is based on the concept of idealised C2 component (iC2C)
Idealised C2 Component

- A structuring concept for the incorporation of exception handling in component-based systems
- Equivalent, in structure and behavior, to the idealised fault-tolerant component
- Defined according to the C2 style
Overall Structure of an iC2C

- **iC2C_top**
  - NormalActivity
  - Normal behavior & Error detection

- **iC2C_internal**
  - AbnormalActivity
  - Error diagnosis & recovery

- **iC2C_bottom**
Description of FaTC2

- The concept of iC2C is employed for defining exception handling contexts
  - NormalActivity component: normal behavior and error detection
  - AbnormalActivity component: error treatment
- Connections between normal and abnormal parts are managed by FaTC2
  - Developers focus on implementing the normal and abnormal behavior of the system
  - Abstracts the interaction protocol
Description of FaTC2 (2)

Provided by the application developer
Description of FaTC2 (3)

Provided by FaTC2

iC2C_top

iC2C_internal

NormalActivity

AbnormalActivity

iC2C_bottom
Exception Handling at the Architectural Level

- FaTC2 defines an architectural-level EHS for component-based systems

- Main features:
  - Separates exception handlers from normal behavior
  - Handlers may be attached to components, connectors and configurations
  - Exception propagation according to the execution flow of the application
Exception Definition

- Architectural exceptions are data objects implemented as simple Java exceptions
- FaTC2 wraps exceptions as C2 notifications
Handler Definition and Attachment

- The AbnormalActivity component of an iC2C defines an architectural-level exception handler
- Handlers may be attached to components, connectors and configurations
- FaTC2 supports the definition of multiple exception-handling contexts
Handler Definition and Attachment (3)

- Normal behavior
- Internal exception handlers
- External exception handlers
- AbnormalActivity

Diagram showing the flow between different levels of exception handling in an IC2C system.
Handler Definition and Attachment (2)

A C2 configuration

Exception handlers for the whole configuration

iC2C_top

iC2C_internal

iC2C_bottom

AbnormalActivity
Exception Propagation

Normal Activity

Abnormal Activity

External iC2C

(...)
Exception Propagation

Service Request

Normal Activity

(...)

Abnormal Activity

External iC2C
Exception Propagation

Exception Raised

Normal Activity

(...)

Abnormal Activity

External iC2C
Exception Propagation

Unable to handle exception. Re-raising

Unable to handle exception. Re-raising

Normal Activity

(...)

AbnormalActivity

External iC2C
Exception Propagation

Unable to handle exception. Re-raising

(....)

Normal Activity

AbnormalActivity

External iC2C
Continuation of the Flow of Control

Abnormal Activity

Normal Activity

Exception handled successfully. Returning to normal

Green iC2C receives return-to-normal request

Returning to normal

Continuation of the Flow of Control

Abnormal Activity

Normal Activity

Exception handled successfully. Returning to normal

Green iC2C receives return-to-normal request

Returning to normal
Continuation of the Flow of Control

The NormalActivity component of the green iC2C handles the request.
FaTC2 and C2.FW

Original (C2.FW)
- <<Interface>> DelegateBrick
- AbstractC2DelegateBrick
- IC2CBottomConnector
- IC2CTopConnector
- IC2CInternalConnector
- INormalActivityComponent
- IC2C
- IC2CCompositeBrick

Extension
- FaTC2
- <<Interface>> Component
- AbstractAbnormalActivityComponent
- IC2CCompositeBrick
- IC2CInternalConnector
- IC2CTopConnector
- IC2CBottomConnector
- IAbnormalActivityComponent
- AbstractNormalActivityComponent
- IC2CBrick
- AbstractNormalActivityComponent

(...)
Conclusions

- Our contributions:
  - Definition of an architectural-level EHS for component-based applications
  - Construction of a reusable implementation for this EHS by means of the FaTC2 framework
  - Extension of the C2.FW framework with forward error-recovery

- Architectural-level exception handling is not a replacement for language-level exception handling
Work in Progress

- Asynchronous iC2C
- Some of the features defined by the EHS are still not supported by FaTC2
  - Hierarchical handler search
  - Attachment of handlers to arbitrary configurations
- Evaluation of the EHS
Contact Information

Fernando J. Castor de Lima Filho
fernando@ic.unicamp.br

Paulo Asterio de Castro Guerra
asterio@ic.unicamp.br

Cecília Mary Fischer Rubira
cmrubira@ic.unicamp.br
Related Work

[Cook:1999:HRU]   [Issarny:2001:ABE]