Carnegie Mellon University – University of Coimbra
Master of Software Engineering



Asterisk-Java, design patterns

Analysis of Software Artifacts, Homework #3

Ignacio Cavero, João Brito, João Viegas, Miguel Sousa

What is Asterisk-Java?

- Free open-source PBX (Private Branch Exchange) java library
  - Handles calls over ethernet or RJ11
- Provides applications with the possibility of interacting with users through standard home phones (or other types of phone lines)
- Asterisk-Java server handles call requests by executing scripts

Asterisk-Java, real-world example



- Offers a service that allows a user to enter his/her phone number on a website. The system will dial the user's number, ask the person to hold (queuing the call) and then dial customer service

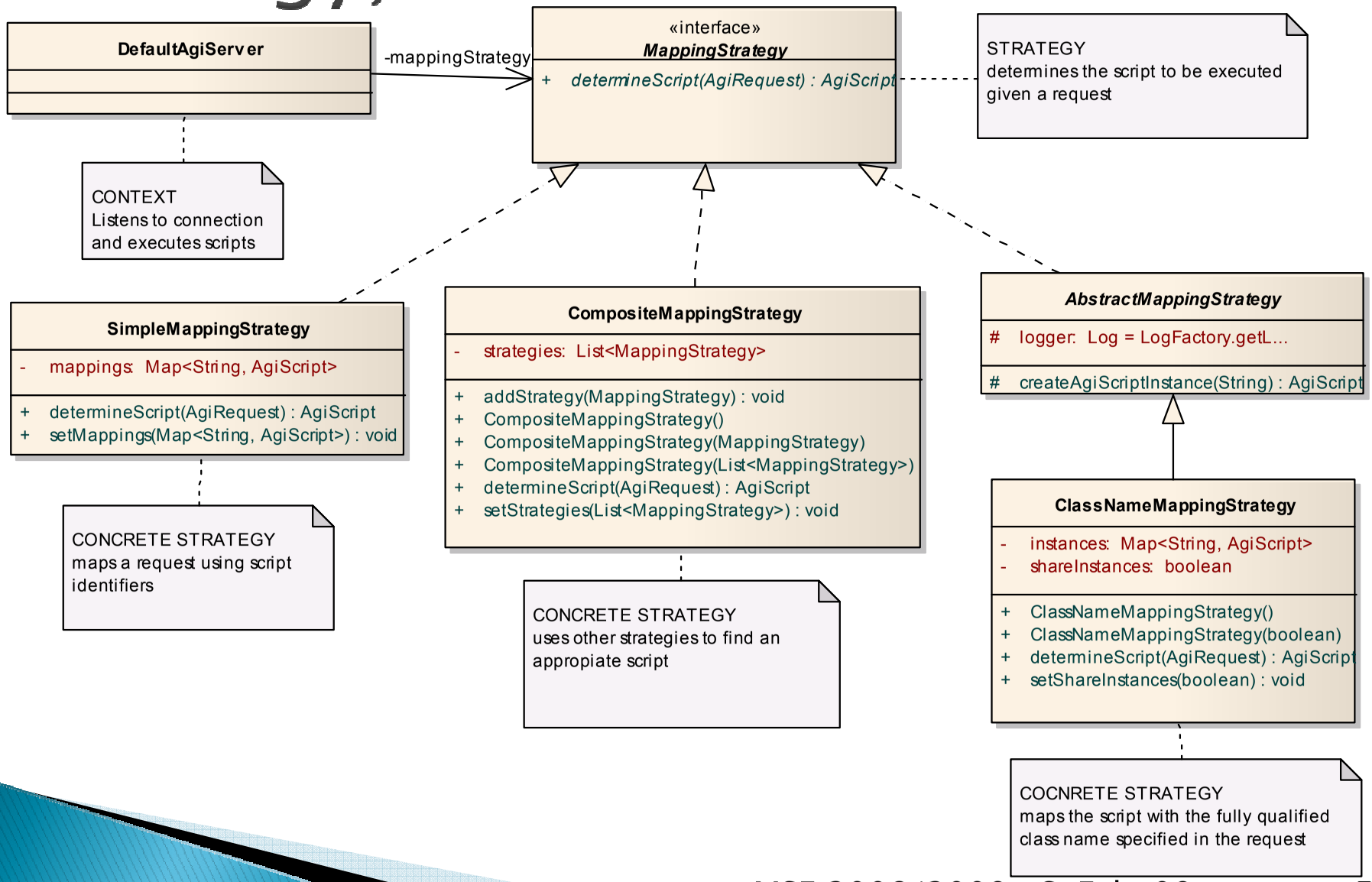
Amazon click to call

Asterisk supports this

Strategy design pattern
Script mapping

http://svn.reucon.net/repos/asterisk-java/trunk/src/main/java/org/asteriskjava/fastagi

Strategy, UML



Strategy: Rationale

- Allows different implementations of the mapping between requests and the script that is going to handle the request

- Three implementations are provided:
  - Loads classes based on specific identifiers
  - Looks in configuration files
  - Using reflection

Strategy: Pros & Cons

- Advantages
  - Adds flexibility for different types of formats in the storage of the corresponding script on the request, e.g. a class name or a previously configured identifier
  - Provides a way for applications to load scripts in completely arbitrary ways that do not depend on the request
- Disadvantages
  - Increases the configuration mechanisms necessary to make an AgiServer instance

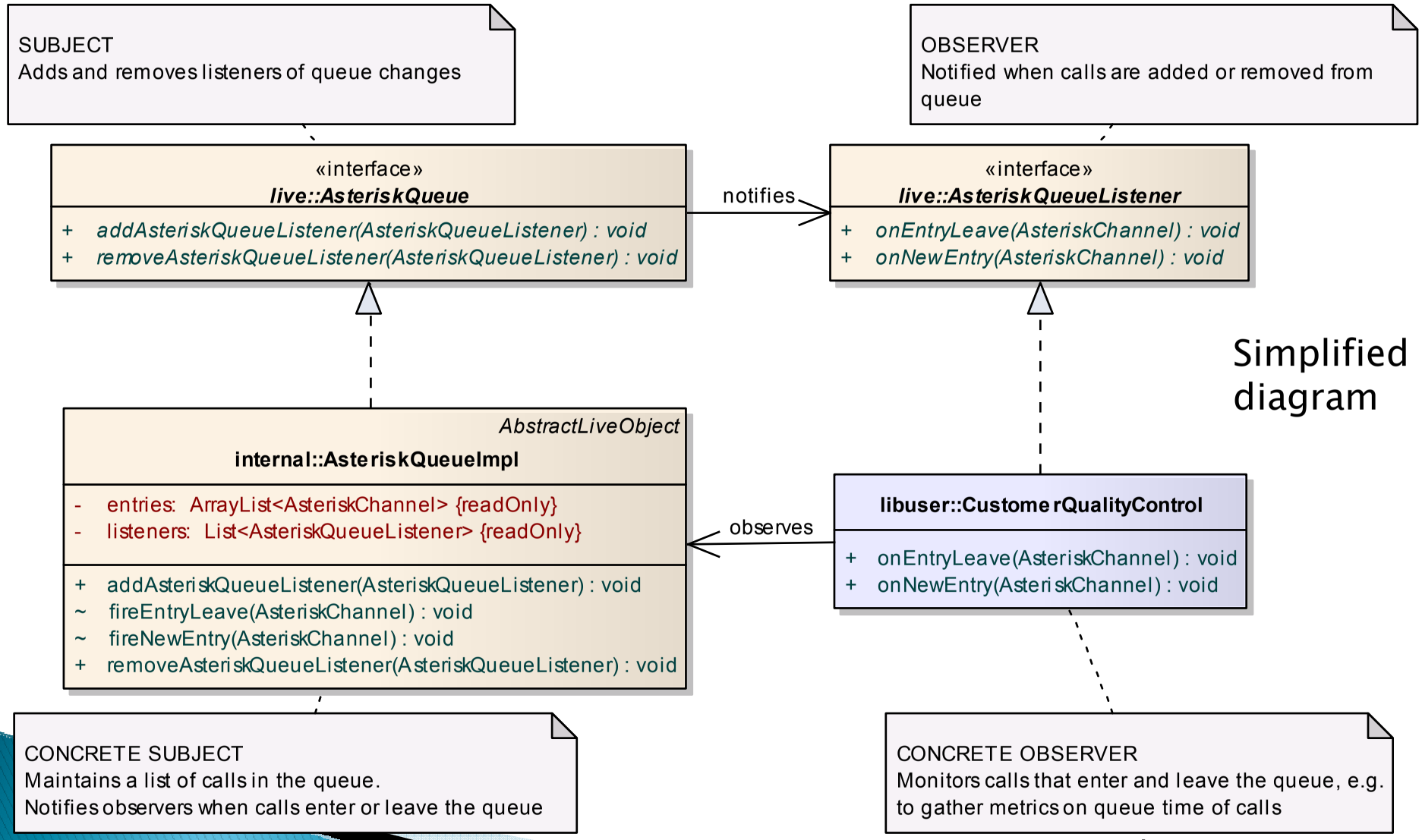
Observer design pattern
AsteriskQueue

http://svn.reucon.net/repos/asterisk-java/trunk/src/main/java/org/asteriskjava/live/AsteriskQueue.java

http://svn.reucon.net/repos/asterisk-java/trunk/src/main/java/org/asteriskjava/live/AsteriskQueueListener.java

http://svn.reucon.net/repos/asterisk-java/trunk/src/main/java/org/asteriskjava/live/internal/AsteriskQueueImpl.java

Observer, watching call's queues



Observer: Rationale

- Allows asterisk-java applications to monitor calls that enter and exit call queues
- Encapsulates implementation details of library from application developers
- Example:
  - A CustomerQualityControl observer gathers data for metrics on average time a call is in the queue

Observer: Pros & Cons

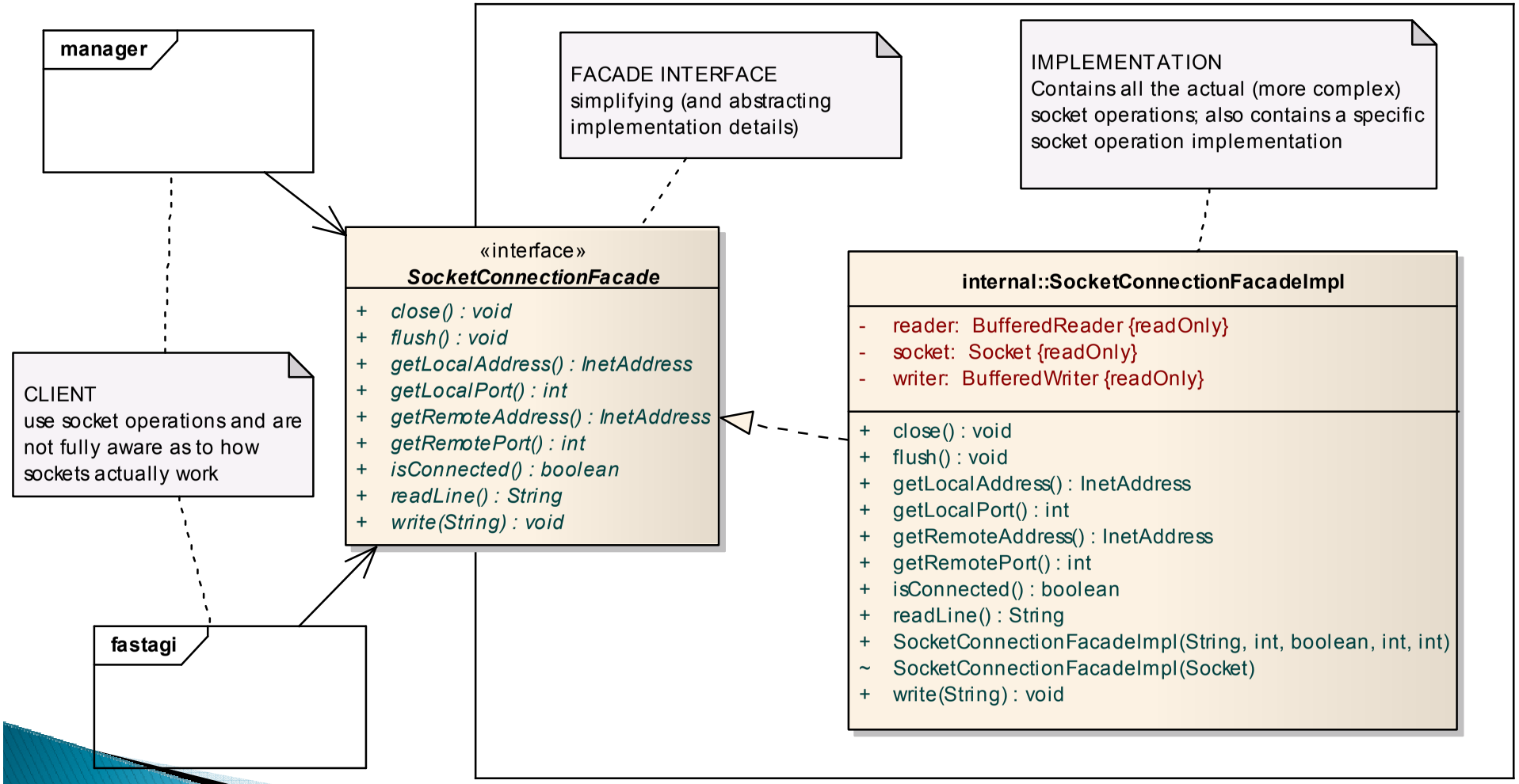
- Advantages
  - Easy to use mechanism to monitor call queues
  - Monitoring of call queues is transparent for library implementation
- Disadvantages
  - Performance might be an issue, when handling large numbers of calls
  - Library has no control over concrete observer's implementation: chance of failure situations (e.g. server hang)
  - Notifications to multiple observers are not ordered, hence no order-related dependencies can be used by applications

Facade design pattern
ServerSocketFacade

http://svn.reucon.net/repos/asterisk-java/trunk/src/main/java/org/asteriskjava/util/SocketConnectionFacade.java

http://svn.reucon.net/repos/asterisk-java/trunk/src/main/java/org/asteriskjava/util/internal/SocketConnectionFacadeImpl.java

Facade: UML


manager
FACADE INTERFACE
simplifying (and abstracting implementation details)
IMPLEMENTATION
Contains all the actual (more complex) socket operations; also contains a specific socket operation implementation
«interface»
SocketConnectionFacade
+ close() : void
+ flush() : void
+ getLocalAddress() : InetAddress
+ getLocalPort() : int
+ getRemoteAddress() : InetAddress
+ getRemotePort() : int
+ isConnected() : boolean
+ readLine() : String
+ write(String) : void
internal::SocketConnectionFacadeImpl
- reader: BufferedReader {readOnly}
- socket: Socket {readOnly}
- writer: BufferedWriter {readOnly}
+ close() : void
+ flush() : void
+ getLocalAddress() : InetAddress
+ getLocalPort() : int
+ getRemoteAddress() : InetAddress
+ getRemotePort() : int
+ isConnected() : boolean
+ readLine() : String
+ SocketConnectionFacadeImpl(String, int, boolean, int, int)
~ SocketConnectionFacadeImpl(Socket)
+ write(String) : void
CLIENT
use socket operations and are not fully aware as to how sockets actually work
fastagi

Facade: Rationale

- SocketConnectionFacade provides read and write operations on a generic socket connection
- Simpler and centralized access to socket operations
- Hides implementation details, allowing for implementation by any I/O socket communication form (e.g., UDP or TCP)

Facade: Pros & Cons

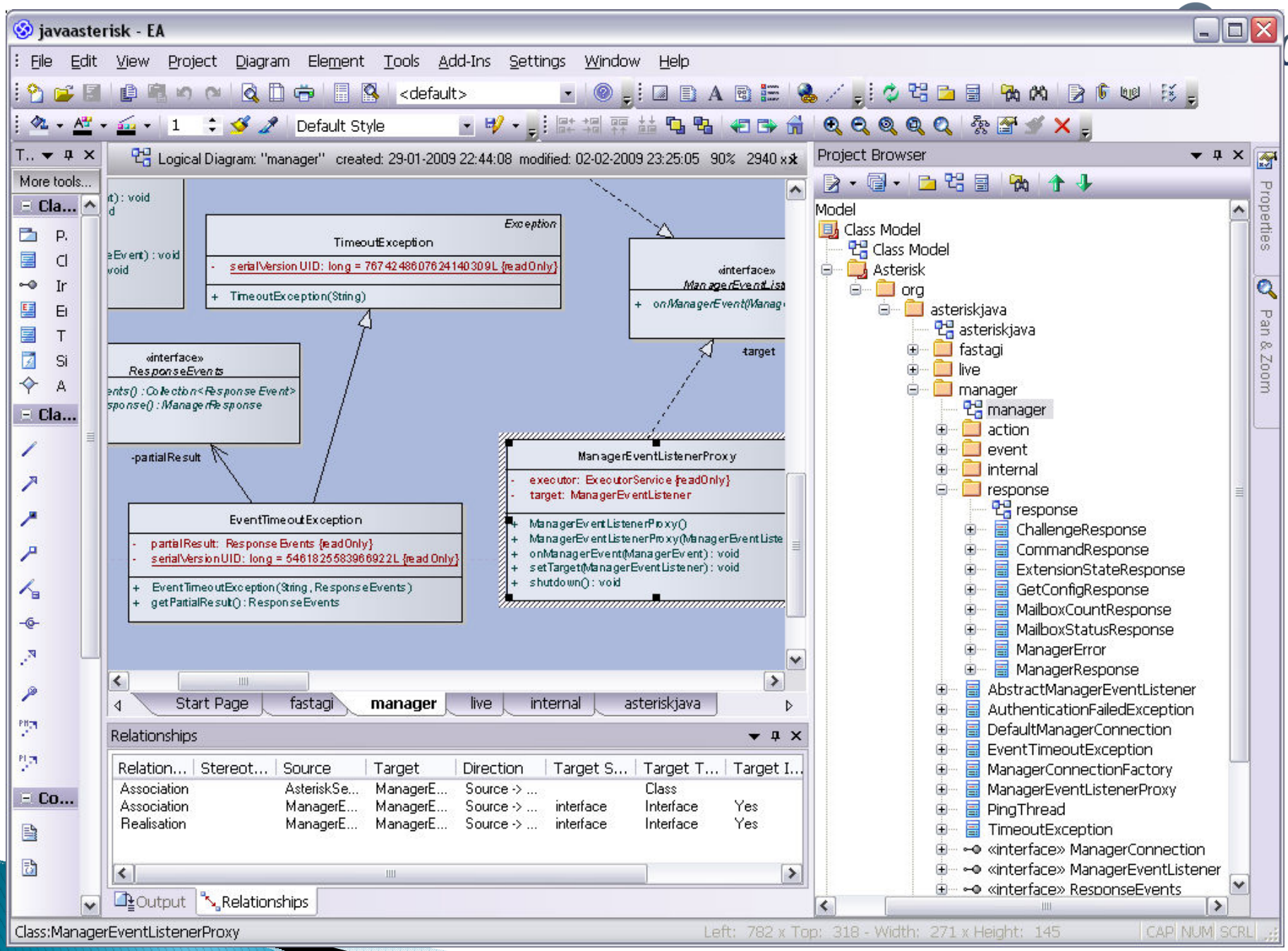
- Advantages
  - Much easier to use (no need to handle complete socket code everywhere)
  - Very easy to change actual implementation of socket operations (e.g., change from TCP to UDP)
- Disadvantages
  - Loses low-level implementation details (e.g., no way to set send buffer's size)

Something cool

Reverse engineering with Enterprise Architect

Reverse Engineering with EA

- UML CASE tool with support for code generation and reverse engineering on multiple languages
- JAVA reverse engineering
  - Retrieves packages, classes and relationships
  - Creates class diagrams for each package


javaasterisk - EA
File Edit View Project Diagram Element Tools Add-Ins Settings Window Help
Logical Diagram: "manager" created: 29-01-2009 22:44:08 modified: 02-02-2009 23:25:05 90% 2940 x
Exception
TimeoutException
- serialVersionUID: long = 7674248607624140309L {readOnly}
+ TimeoutException(String)
«interface»
ResponseEvents
«interface»
-target
-partialResult
EventTimeoutException
- partialResult: ResponseEvents {readOnly}
- serialVersionUID: long = 5461825583966922L {readOnly}
+ EventTimeoutException(String, ResponseEvents)
+ getPartialResult(): ResponseEvents
ManagerEventListenerProxy
- executor: ExecutorService {readOnly}
- target: ManagerEventListener
+ ManagerEventListenerProxy()
+ onManagerEvent(ManagerEvent): void
+ setTarget(ManagerEventListener): void
+ shutdown(): void
Start Page
fastagi
manager
live
internal
asteriskjava
Relationships
Relation... Stereot... Source Target Direction Target S... Target T... Target I...
Association AsteriskSe... ManagerE... Source -> ... Class
Association ManagerE... ManagerE... Source -> ... interface Interface Yes
Realisation ManagerE... ManagerE... Source -> ... interface Interface Yes
Output
Relationships
Project Browser
Model
Class Model
Class Model
Asterisk
org
asteriskjava
asteriskjava
fastagi
live
manager
manager
action
event
internal
response
response
ChallengeResponse
CommandResponse
ExtensionStateResponse
GetConfigResponse
MailboxCountResponse
MailboxStatusResponse
ManagerError
ManagerResponse
AbstractManagerEventListener
AuthenticationFailedException
DefaultManagerConnection
EventTimeoutException
ManagerConnectionFactory
ManagerEventListenerProxy
PingThread
TimeoutException
«interface» ManagerConnection
«interface» ManagerEventListener
«interface» ResponseEvents
Properties
Pan & Zoom
Class:ManagerEventListenerProxy
Left: 782 x Top: 318 - Width: 271 x Height: 145
CAP NUM SCRL

Questions & Answers

