Towards Generic and Middleware-independent Support for Replicated, Distributed Objects

Jörg Domaschka, Hans P. Reiser, Franz J. Hauck

Distributed Systems Lab,
Faculty of Computer Science
Ulm University
Germany

Aspectix Research Group
http://www.aspectix.org/

20 March 2007
Motivation

Why replication?

- Object-based distributed systems have single point of failure
- Replication provides fault-tolerance
Motivation

Why replication?

- Object-based distributed systems have single point of failure
- Replication provides fault-tolerance

What are the drawbacks?

- Implementing replication support is non-trivial
- High development costs
- Bigger middleware core
Motivation

Why replication?

- Object-based distributed systems have single point of failure
- Replication provides fault-tolerance

What are the drawbacks?

- Implementing replication support is non-trivial
- High development costs
- Bigger middleware core

State of the art

- Many middleware systems provide replication support
- All of them support group communication, replica management
- Barely re-use of existing solutions
Goal

Provide a single replication framework for multiple middleware systems!

Problems to solve

- What interface does the framework have to provide?
- How to conceptually integrate the framework into the middleware systems?
Outline

1. Architecture for Replication Support
2. Separation in Practice
3. Evaluation
4. Conclusion
Background
Middleware for Distributed Objects

Client accesses a single remote object
Client-server interaction scheme
Request followed by response

⇒ Rather simple communication

J. Domaschka (Ulm University)

Generic Replication Support
http://www.aspectix.org/
Client accesses a single remote object
Client-server interaction scheme
Request followed by response
⇒ Rather simple communication
Background
Replication Support in Middleware

Client accesses a group of identical remote objects

Replica group may change

Determinism by totally ordered multicast

⇒ Complex communication logic
Background
Replication Support in Middleware

- Client accesses a group of identical remote objects
- Replica group may change
- Determinism by totally ordered multicast

⇒ Complex communication logic
Background
Replication support: Separation of concerns

Responsibility of a replication framework
- Ensure determinism and consistency
- Follow changes of the replica group
- Ensure replication transparency

J. Domaschka (Ulm University)
Generic Replication Support
http://www.aspectix.org/
Responsibility of a replication framework

- Ensure determinism and consistency
- Follow changes of the replica group
- Ensure replication transparency
An Architecture for Replication Support

Initial step

Generic replication framework:
- Transports any data
- Handles data in an opaque way
  ⇒ Interface supports only byte[]

Integration into middleware
- Not all stubs support byte[]
- Redirecting calls to replication framework necessary

J. Domaschka (Ulm University)
Generic replication framework:
- Transports any data
- Handles data in an opaque way
Generic replication framework:

- Transports any data
- Handles data in an opaque way

⇒ Interface supports only byte[]
Generic replication framework:
- Transports any data
- Handles data in an opaque way
⇒ Interface supports only `byte[]`

Integration into middleware:
- Not all stubs support `byte[]`
- Redirecting calls to replication framework necessary
An Architecture for Replication Support
Step II: The Adapter

- Application
- Stub
- Adapter
- Replication Client
- Replication Server
- Adapter
- Obj Adapter Dispatcher Skeleton
- Object Impl.
An Architecture for Replication Support

Step II: The Adapter

Adapter characteristics

- Conceptually part of the middleware
- Initialises replication framework

Redirects calls to replication framework

J. Domaschka (Ulm University)
An Architecture for Replication Support

Step II: The Adapter

Adapter characteristics
- Conceptually part of the middleware
- Initialises replication framework

Redirects calls to replication framework
- Does not solve all problems
An Architecture for Replication Support

The binding problem
An Architecture for Replication Support

The binding problem

- Application
- Middleware
- Runtime
- Remote
- Ref (A,B)
- Stub
- Adapter
- Replica A
- Replica B

J. Domaschka (Ulm University)
An Architecture for Replication Support

The binding problem
An Architecture for Replication Support

The binding problem
An Architecture for Replication Support

The binding problem

- Application
- Middleware
- Runtime
- Remote
- Ref (A,B)
- Stub
- Adapter
- Replica Client
- Contact Info (C,D)
- Replica A
- Replica B
- Replica C
- Replica D

J. Domaschka (Ulm University)
Solving the Binding Problem

Remote Reference should reflect lastest known state
Solving the Binding Problem

Remote Reference should reflect latest known state
  - Middleware system has to get the state
Remote Reference should reflect latest known state
- Middleware system has to get the state
  - Middleware pulls information from adapter/stub
    ⇒ Not supported by all middleware systems
Solving the Binding Problem

Remote Reference should reflect lastest known state

- Middleware system has to get the state
  - Middleware pulls information from adapter/stub
    ⇒ Not supported by all middleware systems
  - Replication framework pushes state into middleware
    ⇒ No generic middleware interface
Solving the Binding Problem

Remote Reference should reflect lastest known state

- Middleware system has to get the state
  - Middleware pulls information from adapter/stub
    ⇒ Not supported by all middleware systems
  - Replication framework pushes state into middleware
    ⇒ No generic middleware interface

- Conversion step required
  - Converter as callback handler
An Architecture for Replication Support

Step III: The Converter

- Application
- Middleware
- Runtime
- Remote Ref (A,B)
- Stub
- Adapter
- Replication Client
- Contact Info (B,C)
- Replica A
- Replica B
- Replica C

updateReference

J. Domaschka (Ulm University)
An Architecture for Replication Support

Step III: The Converter
An Architecture for Replication Support

Step III: The Converter

Application
Middleware
Runtime
Remote Ref (B,C)
Stub
Converter
Adapter
Replication Client
Contact Info (B,C)
Replica A
Replica B
Replica C
Starting Point:
FTflex replication framework integrated into Aspectix middleware
Starting Point:
\textit{FTflex} replication framework integrated into Aspectix middleware

Coupling:
- IOR
  - Aspectix: Remote reference, binding
  - \textit{FTflex}: Store state of replica group
- Coupling code: Code in framework unrelated to replication
Separation in Practice

Starting Point:
FTflex replication framework integrated into Aspectix middleware

Coupling:
- IOR
  - Aspectix: Remote reference, binding
  - FTflex: Store state of replica group

- Coupling code: Code in framework unrelated to replication

Separation
- IOR
  - Used for binding only
  - Extended FTflex with a replication state management facility
  - Converter translates between both systems

- Coupling Code: Moved to adapter
Support for Arbitrary Middleware Architectures

- Adapter and converter heavily depend on middleware
- In general four approaches to include the adapter
  - Integration: modify the middleware source code
  - Interception: intercept messages at OS level
  - Service approach: application calls replication services
  - Customised stub: All further stubs are copies of the first one

Existing adapters
- Aspectix-FTflex adapter
- Java RMI-FTflex
Evaluation

Replication:

<table>
<thead>
<tr>
<th>Method</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java RMI</td>
<td>81.7 µs</td>
</tr>
<tr>
<td>RMI-FT\text{flex}</td>
<td>126.7 µs</td>
</tr>
</tbody>
</table>

Costs: 55%

Separation:

<table>
<thead>
<tr>
<th>Method</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT\text{flex} integrated</td>
<td>124.9 µs</td>
</tr>
<tr>
<td>FT\text{flex} adapter</td>
<td>130.4 µs</td>
</tr>
</tbody>
</table>

Costs: 4%

Set-up:
- One client, one replica
- On a single physical machine
- Different virtual machines

J. Domaschka (Ulm University)

Generic Replication Support
http://www.aspectix.org/
Conclusion:

- Middleware-independent replication support is feasible
  - Genericity by the use of byte[] at interface
  - Handle data in an opaque way
- Light-weight Integration into middleware
  - Adapter and Converter
- Overhead introduced by replication logic

Ongoing work:

- Adapter for JacORB
- Comparison with FT-CORBA implementation
Separation of Concerns

Object reference, Interface definition, Object adapter, Binding, Code generation, Consistency management, Replication management
Separation of Concerns

Interface definition, Object adapter, Binding, Code generation, Consistency management, Replication management

Object reference

Middleware concerns  Replication concerns
Separation of Concerns

Interface definition, Object adapter, Binding, Code generation, Consistency management, Replication management

Middleware concerns
- Object reference

Replication concerns
Separation of Concerns

Object adapter, Binding, Code generation, Consistency management, Replication management

**Interface definition**

Middleware concerns

- Object reference

Replication concerns
Separation of Concerns

Object adapter, Binding, Code generation, Consistency management, Replication management

Middleware concerns
- Object reference
- Interface definition

Replication concerns
- Object adapter
- Consistency management
- Message ordering
- Determinism
- Replication Management
- Monitoring of replicas
- Start new replicas
- Migrate replicas

J. Domaschka (Ulma University)
Generic Replication Support http://www.aspectix.org/
Separation of Concerns

Binding, Code generation, Consistency management, Replication management

Object adapter

Middleware concerns
- Object reference
- Interface definition

Replication concerns
Separation of Concerns

Binding, Code generation, Consistency management, Replication management

Middleware concerns
- Object reference
- Interface definition
- Object adapter

Replication concerns
- Object adapter
Separation of Concerns

Code generation, Consistency management, Replication management

Binding

Middleware concerns
- Object reference
- Interface definition
- Object adapter

Replication concerns
- Object adapter
Separation of Concerns

Code generation, Consistency management, Replication management

Middleware concerns
- Object reference
- Interface definition
- Object adapter
- Binding

Replication concerns
- Object adapter
Separation of Concerns

Consistency management, Replication management

Code generation

Middleware concerns
- Object reference
- Interface definition
- Object adapter
- Binding

Replication concerns
- Object adapter
Separation of Concerns

Consistency management, Replication management

Middleware concerns
- Object reference
- Interface definition
- Object adapter
- Binding
- Code generation
  - Dispatching
  - Marshalling in Stub and Skeleton

Replication concerns
- Object adapter
Separation of Concerns

Replication management

Consistency management

Middleware concerns
- Object reference
- Interface definition
- Object adapter
- Binding
- Code generation
  - Dispatching
  - Marshalling in Stub and Skeleton

Replication concerns
- Object adapter

Calling semantics

Monotivity of replicas

Start new replicas

Migrate replicas
Seperation of Concerns

Replication management

Middleware concerns
- Object reference
- Interface definition
- Object adapter
- Binding
- Code generation
  - Dispatching
  - Marshalling in Stub and Skeleton
- Consistency management
  - Calling semantics

Replication concerns
- Object adapter
- Consistency management
  - Message ordering
  - Determinism

J. Domaschka (Ulm University)

Generic Replication Support

http://www.aspectix.org/
Separation of Concerns

Replication management

Middleware concerns
- Object reference
- Interface definition
- Object adapter
- Binding
- Code generation
  - Dispatching
  - Marshalling in Stub and Skeleton
- Consistency management
  - Calling semantics

Replication concerns
- Object adapter
- Consistency management
  - Message ordering
  - Determinism

J. Domaschka (Ulm University)
Generic Replication Support
http://www.aspectix.org/
Separation of Concerns

Middleware concerns
- Object reference
- Interface definition
- Object adapter
- Binding
- Code generation
  - Dispatching
  - Marshalling in Stub and Skeleton
- Consistency management
  - Calling semantics

Replication concerns
- Object adapter
- Consistency management
  - Message ordering
  - Determinism
- Replication Management
  - Monitoring of replicas
  - Start new replicas
  - Migrate replicas
- CORBA-based
- Supports fragmented objects
  - An object consists of multiple fragments
  - Fragments are arbitrarily distributed over multiple address spaces
  - Distribution of state and functionality is arbitrary
  - Communication is an object-internal issue
CORBA-based

Supports fragmented objects
- An object consists of multiple fragments
- Fragments are arbitrarily distributed over multiple address spaces
- Distribution of state and functionality is arbitrary
- Communication is an object-internal issue

Features
- Implicit binding
- Interface Definition
- Code generation

but arbitrary distribution of state and functionality disables generation of marshalling and dispatching
Replication infrastructure based on Aspectix
- Fragmented objects
- Has a more concrete object structure
  - Access fragments
  - Replica fragments
  - Code generation results in marshalling and dispatching code
Replication infrastructure based on Aspectix
  - Fragmented objects

Has a more concrete object structure
  - Access fragments
  - Replica fragments
  - Code generation results in marshalling and dispatching code

Object adapter part of replica fragment
  - Message Management
  - Dispatching
  - Error handling

Own group communication system