
JavaOne℠
Sun's Worldwide Java Developer Conference

JavaOne℠

Sun's Worldwide Java Developer Conference

# Distributed Computing and Persistence

*Jim Waldo*
*Sr. Staff Engineer*
*JavaSoft*

# Distribution and Persistence

- Objects existing over space and time
  - Objects over space = distribution
  - Objects over time = persistence
- Outside the language environment
- Introduces new failure modes
- Usually treated separately

# Preserving and Reconstituting Objects

- Simplest form possible
- Allows reconstituting a copy
- System includes default implementation
- Custom implementations possible

# ObjectStreams

- Self-identifying byte streams
- Utilize Java™ `Stream` abstraction
- Preserve object by writing to a stream
- Reconstitute by reading from a stream

# Some Technical Details

- Identification by structure
- All objects in a graph serialized
- Multiple references and cycles preserved

# Example

- Write out an object

```
 FileOutputStream f = new
   FileOutputStream(“tmp”);
ObjectOutput s = new
   ObjectOutputStream(f);
s.writeObject(“Today”);
s.writeObject(new Date());
s.flush();
```

# Example

- Now read it back

```
FileInputStream in = new
   FileInputStream(“tmp”);
ObjectInputStream s = new
   ObjectInputStream(in);
String today =
   (String)s.readObject();
Date date =
   (Date)s.readObject();
```

# Extending for "Special" Objects

- `writeObject` and `readObject` methods
- Objects can refuse to be serialized
  - Mark fields as `transient`
  - Throw exception
  - Default is NOT to serialize

# Distributed Objects

- Method calls across VM boundaries
- Remote procedure call extended to Java™
- Leverage the universality of the Java™ virtual language

# Preserving the Java Object Model

- Distributed objects are Java™-based objects
- Distributed interfaces are Java™-based interfaces
- Object references carry their true type
- Method invocation may fail in new ways

# Taking Discontinuity Seriously

- No global knowledge
- Partial failure
- No central administration
- No temporal ordering

# Unify and Simplify: JavaSpaces

- Network repository for grouped objects
- Continuous over time and space
- Unifies persistence and communication
- Wormholes between VMs

# Object Sequences

- Ordered sets of Java™-based objects
- Basic entity for persistence and communication
- Can contain any object extending `Object`

# Methods on Object Sequences

- `put()` into a JavaSpace
- `get()` from a JavaSpace
- `remove()` from a JavaSpace

# Templates

- Object sequences with `Null` values
- Used in `get()` and `remove()` methods
- Associative lookup based on value

# Transactions

- Discontinuities made atomic
- Required for reliability
- Span multiple operations in a JavaSpace
- Span multiple JavaSpaces

# Two Phase Commit

- JavaSpaces act as managers
- Participants defined by an Interface

```
join()
prepare()
commit()
abort()
ping()
```

# Replication

- Avoid a single failure point
- Policies via consistency guarantees
- Unexplored design space
  - Clear at the edges
  - Fuzzy everywhere else

# Schedules

- Currently an AD project
- Early access sometime this summer
- Products within the next 12 months

# Further Information

- Watch

  http://chatsubo.javasoft.com/