Remote Invocation: Remote Method Invocation (19-1)

Summer Term 2020 | 24.06.2020
Claudia Müller-Birn, Barry Linnert
Middleware Layers

Applications, services

Remote invocation:
Request-Reply Protocols, Remote Procedure Calls

Underlying inter-process communication

UDP and TCP
Our Topics Today

• Properties of Remote Object Invocation

• Implementation of RMI

• Java RMI
Remote Call Variantes

Remote procedure call (RPC)
- A procedure is called (typically part of a module) for procedural languages (e.g., Modula, C)
- Also for heterogeneous infrastructures, e.g. Distributed Computing Environment (DCE)

Remote object invocation (ROI)
- For object-oriented languages (e.g., Java, C++, C#), where an operation/method of an object is called.
- In Java it is called Remote Method Invocation (RMI) and in C# "-.NET Remoting"
- Also for heterogeneous infrastructures, e.g. CORBA
Commonalities of ROI and RPC

• Support of programming languages with interfaces.

• Both are typically constructed on top of the request-reply protocol and it offers a number of call semantics (exactly once, at most once, and at least once).

• Offer a similar level of transparency, means local and remote calls employ the same syntax, but remote interfaces expose the distributed nature for example by supporting remote exceptions.
Remote invocation

REMOTE OBJECT INVOCATION
Remote and Local Method Invocation

Remote object reference: Other objects can invoke the methods of a remote object if they have access to its remote object reference.
Remote Object Reference

The remote object reference is an identifier for a remote object that is valid throughout the distributed system. It is passed in the invocation message to specify which object is to be invoked.

<table>
<thead>
<tr>
<th>32 bits</th>
<th>32 bits</th>
<th>32 bits</th>
<th>32 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet address</td>
<td>port number</td>
<td>time</td>
<td>object number</td>
</tr>
<tr>
<td>interface of remote object</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Remote object reference: Other objects can invoke the methods of a remote object if they have access to its remote object reference.

Remote interface: Every remote object has a remote interface that specifies which of its methods can be invoked remotely.
A Remote Object and its Remote Interface
Instantiation of Remote Objects
Instantiation of Remote Objects

- Remote invocation from A to B
- Local invocation within B
- Remote invocation from C to D
- Local invocation within C
- Remote invocation from E to F
- Remote invocation from G to H
- Local invocation within G
- Remote invocation from I to J
- Local invocation within I
- Remote invocation from K to L
- Local invocation within K

Diagram shows the flow of remote and local invocations among various objects.
Exceptions

• Any remote invocation may fail for reasons related to the invoked object being in a different process or computer from the invoker.

• Remote method invocation should be able to raise exceptions such as timeouts that are due to distribution as well as those raised during the execution of the method invoked.

• Exceptions provide a clean way to deal with error conditions without complicating the code. In addition, each method heading explicitly lists as exceptions the error conditions it might encounter, allowing users of the method to deal with them.