Jini overview

- Framework and services on top of Java RMI
  - finding services
    - finding registries
    - searching registries
    - keeping registries up-to-date
  - Distributed events
  - Transactions
  - JavaSpaces
    - Shared store

Providing Services

Finding providers

Using Services

Proxies

- Java classes that implement standard interfaces
- Classes get downloaded as needed
  - From http server?
- Proxies can use any mechanism to communicate with service
  - RMI stubs, local computation and any communication medium
<table>
<thead>
<tr>
<th>Lego Mindstorm Proxies</th>
<th>Lookup services</th>
</tr>
</thead>
<tbody>
<tr>
<td>• An example: Proxies for Lego Mindstorm robots</td>
<td>• Broadcast looking for lookup services messages</td>
</tr>
<tr>
<td>• Processor on Mindstorm not powerful enough to host a JVM</td>
<td>• Each lookup service on network responds with a proxy</td>
</tr>
<tr>
<td>• Proxy can keep state about robot, communicate through infrared link</td>
<td>• Queries to lookup service are done through proxy</td>
</tr>
<tr>
<td></td>
<td>– actual communication with lookup service is up to proxy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Using a lookup service</th>
<th>Service ID’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can specify a list of classes/interfaces</td>
<td>• When services first register, they are assigned a service ID</td>
</tr>
<tr>
<td>– only interested in results that are subtypes of all</td>
<td>– 128 GUID</td>
</tr>
<tr>
<td>• Can specify attributes</td>
<td>• a particular service using the same service ID on all lookup services</td>
</tr>
<tr>
<td>– Location {</td>
<td>– if you get the same service back from multiple lookup services, can detect duplicates</td>
</tr>
<tr>
<td>String building;</td>
<td></td>
</tr>
<tr>
<td>String floor;</td>
<td></td>
</tr>
<tr>
<td>String wing;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>– must match all attributes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Leasing</th>
<th>Finding lookup services, revisited</th>
</tr>
</thead>
<tbody>
<tr>
<td>• When a service registers with a lookup service, it gets a limited duration lease</td>
<td>• When a service is activated, broadcasts request to find lookup services</td>
</tr>
<tr>
<td>– if lease expires without being renewed, service is de-listed</td>
<td>– using multicast over local net</td>
</tr>
<tr>
<td>• Service must renew lease</td>
<td>• When lookup service is activated, and at periodic intervals, broadcasts “I’m a lookup service”</td>
</tr>
<tr>
<td>– makes sure service is still up and still reachable from lookup service</td>
<td>– services that learn about new lookup services register with them</td>
</tr>
</tbody>
</table>
Example

- Jini network among computers at home
- Jini wireless network among my cell phone, and PDA
- Don’t have to push an “I’m home” button
- Next time lookup services announce “I’m here”, networks will join

Attribute lookup

- Attributes have public references for fields
  - no primitive types
- To match
  - Lookup attribute must be subtype of stored attribute
  - Each non-null field must be equal

TO COME

- Distributed events
- Transactions
- JavaSpaces

Transactions

- Create a transaction from a transaction manager
  - requires maintaining a lease
- All actions performed as part of a transaction should only be visible to that transaction
  - until the transaction commits

Committing a transaction

- All participants are asked to prepare the transaction
  - log it to persistent storage
  - so that it will survive crashes
- Then commit
  - or abort

JavaSpaces

- Derived from Linda TupleSpace
- Store that supports transactions
- Contains entry objects
  - i.e., records
- Stored objects are leased
  - if lease expires, entry removed
### JavaSpace operations

- write - add an entry
- read - find a matching entry
  - Can wait for matching entry (with timeout)
- notify - add listening for matching entries
- take - read and delete atomically

### JavaSpace matching

- Entry’s match just like Jini lookup attributes
- Can provide don’t cares
- Can use subtyping