Service Discovery in Home Environments

KM-/VS-Seminar
Wintersemester 2002/2003
Björn H. Gerth
Betreuer: Muhammad Khan
Structure

- Home Environments, Service Discovery
- Jini
- Universal Plug and Play (UPnP)
- Salutation
- Home Audio/Video interoperability (HAVi)
- Summary & Conclusion
Smart home environment

■ Goals of service discovery networks:
  ■ Network-centric computing instead of disk-centric computing
  ■ Collection of services and clients instead of applications and peripherals
  ■ Easy integration and removal of devices
  ■ Easy communication between devices

■ House of future with smart environment
  ■ Devices help us with daily life
  ■ User interaction for all sorts of devices from any component with display capabilities from anywhere in the house
  ■ Digital technology to enhance entertainment experience
  ■ Remote access of home appliances through the internet
Jini

- Developed by Sun Microsystems
- Open license, but fee for commercial use
- Architecture based on Java
  - Federation of easily pluggable and removable components
  - Low level of administration
  - Avoidance of single points of failure
- Home environments: Useful, but few interfaces for home appliances have been specified
Jini Service Object / Proxy

![Diagram showing Jini client and Jini service with a service proxy object in between.]
Jini Lookup Service

Diagram showing the interaction between a Jini client and a Lookup service, with messages for discovery request, discovery response, lookup request, and lookup response.
Jini Leasing

- Resource allocation and deallocation throughout the network
- Grant of usage of resource for a limited time

Examples:
- Service registration lease
- Use of memory, disk space or display capabilities of other devices
- Event subscription

Advantages:
- Deactivated services
- Recover from temporary breakdown of network
- Cancellation of leases
Universal Plug and Play (UPnP)

- Developed by UPnP Forum, led by Microsoft
- Open source
  - Based on open protocols available almost anywhere
  - OS and language independent
- Targets small to medium size IP networks
  - UPnP included in Windows XP
UPnP protocol stack
UPnP description of services

- Device contains set of services corresponding to functional units
- Description stored in XML file, location given in unicast reply
- List of actions to access service
- List of variables expressing run-time state of service
  - Update messages formatted with GENA
- Description may include Presentation URL
  - Higher level UI at URL
  - Displays status or may allow user to change it
UPnP configuration, AV

- Configuration: DHCP or AutoIP
- Consumer electronics: UPnP AV
  - Set of device and service definitions for devices handling entertainment content
  - Media Server: Has access to entertainment content
  - Media Renderer: Can render such content on local hardware
  - Control Point: Coordinates Servers and Renderers
Developed by Salutation Consortium
Devices available since 1996, but mostly office components
Support of multiple infrastructures
Salutation architecture
Salutation

- Functional unit: One service of a device
- Three ways of communication for components
  - Native mode
  - Emulated mode
  - Salutation mode
- Salutation Lite: Reduced version of architecture for small devices
  - Suitable for devices with limited storage space, low communication bandwidth and little power consumption
Home Audio/Video interoperability (HAVi)

- Focuses on consumer electronics (CE)
- Developed by HAVi organization, founded by eight well-known CE companies (e.g. Sony, Grundig)
- Networking software specifies protocols to be used by components
- Allows multi-directional audio/video streams and share of resources
- Narrow product range: Easy for vendors to develop interfaces and meet demands of audio and video
HAVi architecture
## HAVi configuration

<table>
<thead>
<tr>
<th>Device Class / Element</th>
<th>FAV</th>
<th>IAV</th>
<th>BAV</th>
<th>LAV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Runtime</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Module</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDI Controller</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource Manager</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stream Manager</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCM Manager</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registry</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Manager</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Messaging System</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1394 Communication Media Manager</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDD data</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DCM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Jini</th>
<th>UPnP</th>
<th>Salutation</th>
<th>HAVi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer</td>
<td>Sun Microsystems</td>
<td>Microsoft</td>
<td>Salutation Consortium</td>
<td>HAVi Organization</td>
</tr>
<tr>
<td>License</td>
<td>open license, but fee for commercial use</td>
<td>open source</td>
<td>open source</td>
<td>open source</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
<td>1.0</td>
<td>2.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Network transport</td>
<td>TCP/IP</td>
<td>TCP/IP</td>
<td>independent</td>
<td>IEEE 1394</td>
</tr>
<tr>
<td>Programming language</td>
<td>Java</td>
<td>independent</td>
<td>independent</td>
<td>independent</td>
</tr>
<tr>
<td>OS and platform</td>
<td>independent</td>
<td>independent</td>
<td>independent</td>
<td>independent</td>
</tr>
<tr>
<td>Code mobility</td>
<td>yes (Java RMI)</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Srv attributes searchable</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Central cache repository</td>
<td>optional</td>
<td>no</td>
<td>optional</td>
<td>no</td>
</tr>
<tr>
<td>Operation w/o directory</td>
<td>Lookup Table required</td>
<td>-</td>
<td>yes</td>
<td>Registry required</td>
</tr>
<tr>
<td>Leasing concept</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Security</td>
<td>Java based</td>
<td>IP dependent</td>
<td>authentication</td>
<td>access levels, signatures</td>
</tr>
</tbody>
</table>
Conclusion

- Most service discovery solutions still in developmental stage
  - Product availability
  - Mobile devices
- Interconnecting bridges between architectures

- Thank you for your attention
- Questions & Discussion