

Yu-Wen Chen

Advised by Prof. Henning Schulzrinne, Suman Srinivasan

CCNxServ Project : **Extending Services with FileType Check**

Goal & Progress History

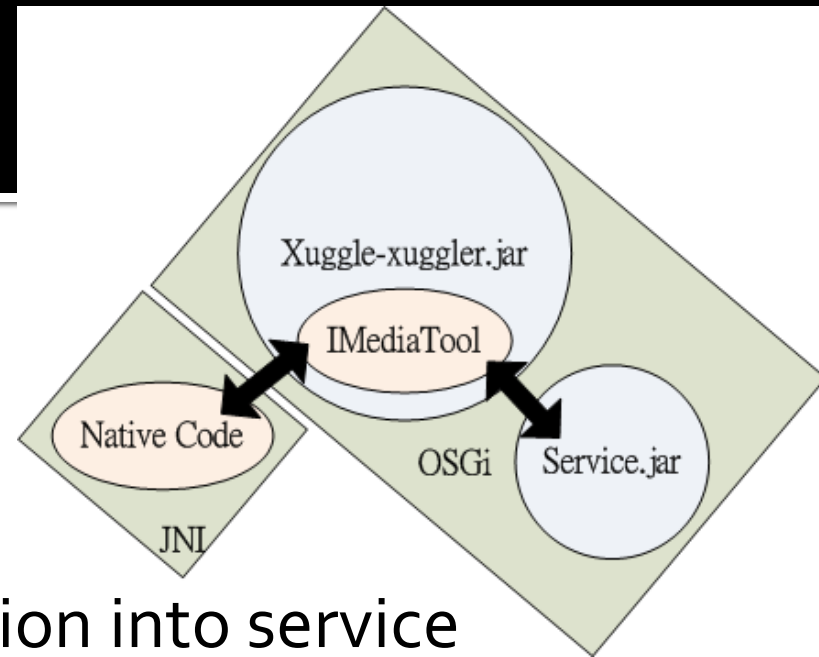
- Goal: Integrate Xuggler applications on CCNService
- Background of Content Centric Network & OSGi (1 week)
 - Read the draft of the paper “CCNServ: Dynamic Service Scalability in Information-Centric Networks” [1]
 - Project: Dynamic Services with Content Centric Networking[2]
- Project CCNxTM [3](1 week)
 - Environment Setting
 - Example Code

Progress History

- CCNServ Project ^[4] (1 week)
 - NetServServiceCore
 - CCNService Examples
 - Linenumber & Nextservice Code - By Amandeep Singh's
- Xuggle ^[5](2 weeks)
 - Java jar files + Native shared libraries: Install the libraries
 - Xuggle applications
 - Know how media files work
 - Example Code & Tool usage
 - Issue1 : "mp4" is more stable for xuggle to work on
 - Issue2 : "[java.lang.UnsatisfiedLinkError](#)" - when we launch applications like Eclipse, it doesn't get the environment variables
 - Solution: DYLD_LIBRARY_PATH login-wide on Mac
 - Solution: Use command line to compile the java file directly in the shell

Progress History

- New service (1 week)
 - Build.xml, Manifest
 - Activator
 - Modify text files successfully.
- Integrate the Xuggle application into service
 - IMediaTool
 - Issue:
 - Fail when calling IMediaTool Reader
 - File permission
 - OSGi and JNI
 - Project - NetServ:ActiveCDN [6] (1 week)



Possible solutions

- Bundle Native Code (2 weeks)
 - Allows developers to notify the OSGi framework of native libraries included in the bundle
 - `<attribute name="Bundle-NativeCode" value="... ..
${lib}/libxuggle-ferry.so;${lib}/libxuggle-xuggler.so;
osname=linux; processor=x86" />`
 - Issue: Cannot find those .so files
- Class path (0.5 week)
 - Set link path
 - Automatically load the Xuggle.jar file when the netserv launched (netserv.core.osgi.Launch)
 - Not solving....

Goal Changed & Approach

- Goal: Extend services to check the file type is supported or not before processing. (Nov.21)
 - Enhance the error-handling capability
- Approach:
 - <Services side>
 - Use Array to store the supported file types
 - Allow services support multi types
 - Get the filePath & Check the type is supported or not.
 - test.txt
 - test.txt%2Blinenumber
 - test.txt%2Bnextservice%2Blinenumber

Approach & Demo cases

- Approach:
 - When the type is not supported by the services:
 - Write Error messages "Error: Filetype is not supported." in the output file.
 - Instead of letting the CCNServer exit (system.exit())
 - Continually make other commands conveniently
- Demo:
 1. Service "linenumber" with supported file type
 2. Service "linenumber" with unsupported file type

References , Q & A

- References:

- [1] Suman Srinivasan, Amandeep Singh, Dhruva Batni, Jae Woo Lee, Henning Schulzrinne and Volker Hilt, "CCNServ: Dynamic Service Scalability in Information-Centric Networks"
- [2] Project - Dynamic Services with Content Centric Networking
- [3] Project - CCNx™, <http://www.ccnx.org>
- [4] Project - CCNService, <https://github.com/amanus/CCNServices>
- [5] Xuggler, <http://www.xuggle.com/xuggler/>
- [6] Project - NetServ:ActiveCDN

Thank you!