Version Control Management for Federated Service-oriented File Sharing

Amaresh Ghosh

Thesis Description

Problem Statement

- Existing version control systems are mostly static client-server systems
- There is no concurrent versioning in existing federated service-oriented systems
- Existing dynamic service-oriented file systems do not support version control

Conclusion

A federated service-oriented file system is needed with concurrent version management capabilities.

Objective/Approach

Objective

 Federated **v**ersioning for service-oriented file **s**ystem (FVS)

Approach
- Review literature on P2P and SOA version control systems
- Elicit the requirements for FVS
- Conduct a feasibility study of FVS
- Develop a methodology to efficiently download and upload versioned files in SILENUS
- Develop required FVS services for SILENUS
- Deploy and validate the FVS framework for exertion-oriented programming

Resources/Schedule

Date	Task
11/04/08	Literature review of research trends and current relevant technologies
11/20/08	Feasibility study of FVS
11/28/08	Design of initial FVS framework
12/10/08	Byte-store, metadata store providers, and FVS façade with version control capabilities
03/10/09	Implementation of FVS in federated service-oriented file system
03/20/09	Final testing of the FVS framework
04/01/09	Validation of requirements end use cases in FVS
04/20/09	Thesis defense

Benefits

- Dedicated, cohesive and decoupled FVS services to maintain file history for FVS service requestors
- Efficient rollback to earlier version of a file based on the retained history of files
- Increased scalability and performance of versioning system by using replicated FVS services
- Increased reliability and bottleneck avoidance by using autonomically replicated FVS Services
- Zero installed user friendly agents attached to the FVS façade
- Zero installed admin agents