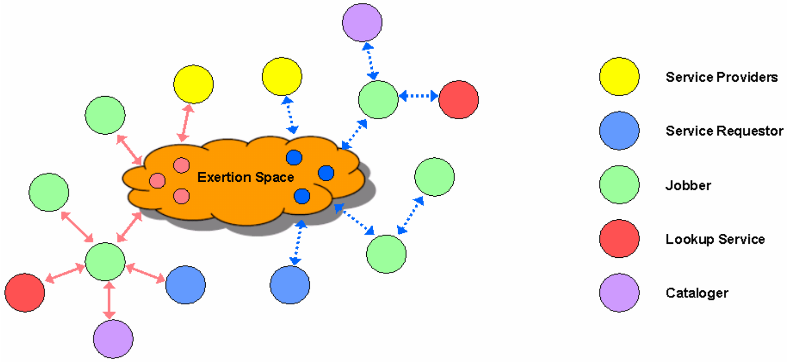
Service-Oriented Programming with Federating Jobbers

Michael Angelo A. Alger

THESIS DESCRIPTION

- SORCER Jobber can coordinate job execution via the Exertion Space in terms of tasks only
- A service requestor cannot submit jobs via the Exertion Space but only through Jobbers
- All inner jobs are coordinated by the same Jobber recursively even if other Jobbers are available and idle



Job execution coordinated via Exertion Space with the help of other underutilized Jobbers is needed .

OBJECTIVE / APPROACH

Objective

- Develop an Enhanced Service-Oriented Programming (ESOP) framework with directly federating and indirectly federating Jobbers via the Exertion Space

Approach

- Define requirements for the ESOP framework in SORCER
- Define algorithmic logic specification for the ESOP framework
- Design the enhanced SORCER Jobber for handling jobs via Exertion Space with other cooperating Jobbers
- Setup and configure the SORCER environment
- Implement the enhanced design of the existing SORCER Jobber
- Implement Admin ServiceUI for Jobber
- Develop validation service providers with respective relevant jobs
- Validate the ESOP framework using the SGrid environment

SCHEDULE

1.	Requirements for Enhanced SO Programming	10/12
2.	Literature Review	10/09
3.	Define Algorithmic Logic for SO Programming	10/13
4.	Enhanced SORCER Jobber Design	10/26
5.	SORCER environment operational	10/19
6.	Feasibility Study in SGrid	10/26
7.	Thesis proposal presentation	10/31
8.	Enhanced SORCER Jobber implementation	11/16
9.	Admin Jobber ServiceUI implementation	11/24
10.	Validation of the Enhanced Jobber and its ServiceUI in SGrid	11/30
11.	ESOP Framework validation with S-Grid	12/07

MISCELLANEOUS SUPPORTING DATA

Benefits

- Flexible job execution with inner jobs via the Exertion Space
- Improved performance and scalability by utilizing other idle Jobbers
- Asynchronous job execution via the Exertion Space that complements synchronous execution by direct access to service providers
- Enhanced job execution control strategy with conditional and iterative tasks
- A basic Admin ServiceUI for listing jobs and tasks