

Middleware, Services, etc.

CPSC 315 – Programming Studio
Fall 2010

Goal: Reuse and Sharing

- Many times we would like to reuse the same process or data for different purpose
 - Want ways to “reuse” code, tools, etc. for different purposes
 - Want ways to “share” data across different applications
- A variety of technologies, techniques, etc. have been developed to do this

Non-sharing stuff

- Silos
 - Idea is that data is stored in such a way that it cannot be communicated to other applications
- Standalone programs
 - Run locally on local data

Middleware

- “Middle” layer between applications and more basic system (OS) or between distributed applications
- Provides a common interface for distributed applications
 - Allows heterogeneous HW/OS platforms
- Generally refers to software allowing processes running across network to interact

Middleware

- Relies on common interchange formats
 - XML data interchange
 - SOAP (an XML-based message passing framework)
 - Simple HTTP
 - Application-specific formats (e.g. for a company seeking to integrate products)

Web Services

- Idea is to provide a “service” that will have common use
- Service can be used by a wide variety of programs
- Can vary in many ways
 - Access method
 - State maintained, or passed only through message?

RPC Web Services

- Remote Procedure Calls
- Traditional type of web service
- Basically like calling a subroutine, but on another system
- Usually use a particular message passing system (like SOAP)

RESTful Web Services

- RESTful (REpresentational State Transfer)
- Very few and simple commands (GET, PUT), but more complex method of describing *resources*
- Resources are the “objects”
 - In implementation, typically each resource has its own URL
 - E.g. a person in a database is a resource
 - Will have a unique URL for that person
 - Calling that URL will get a “document” about that person
 - Document will have hyperlinks to other related information, each of which is another resource

Service Oriented Architectures

- Idea is to break up programs into a set of services
- All services are accessed over a network
 - Communicate with each other
- Applications are built by combining services

Commentary

- This area has had an explosion of jargon, terminology, etc.
 - Much of it driven by business interests
 - Those offering frameworks for development
 - Those offering software to integrate
 - IT/corporate groups looking to describe what they're doing
- Basic principles aren't that different, or particularly complicated
 - Distribute the computation and/or data