Web Services & ShaRef: Applications as Services

Erik Wilde
ETH Zürich

ETH World Explore!, 16.12.2004
`http://dret.net/projects/sharef/`

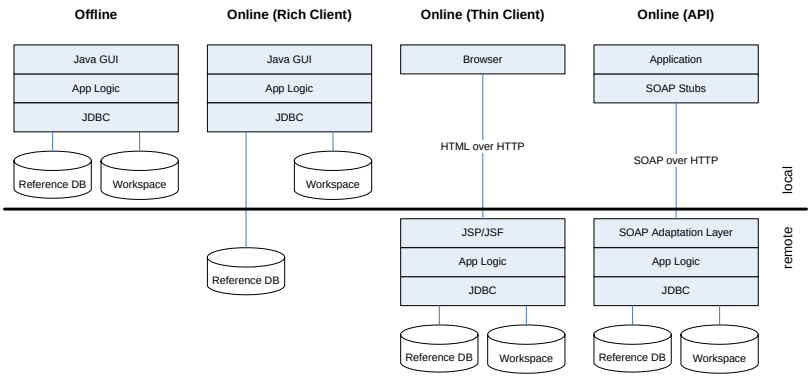
Overview

- ShaRef: Shared References
  - ETH World project
- Web Services and Granularity
  - What is the most appropriate level?
- Web Service Directories
  - UDDI: finding interfaces
  - Open issue: finding services

ShaRef: Shared References

- How to manage bibliographic data
  - in a platform-neutral way (Win, Mac, Linux)
  - in an application-neutral way (LaTeX, Word)
  - collaboratively (research groups, lectures)
  - online and offline (server-based or standalone)

ShaRef Application Model



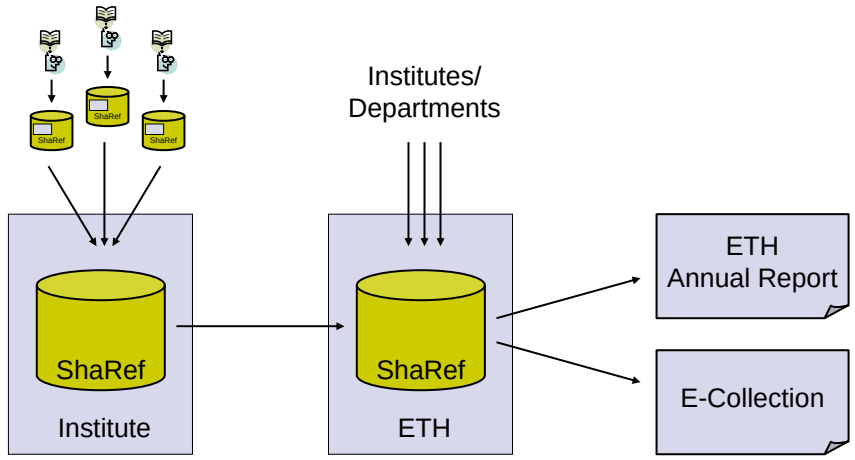
ShaRef Implementation Issues

- online/offline modes
  - local DB vs. remote DB
- online rich client vs. online browser-based access
  - client-side application logic vs. thin client
- online UI vs. online API
  - HTML/HTTP vs. SOAP/HTTP

ShaRef Use Cases

- ShaRef is a program
  - Java-based Client for offline work (local DB)
  - Java-based Client for online work (remote DB)
- ShaRef is service
  - Web-based for online Work (remote DB)
  - Web Service based for application access (remote DB)
- ShaRef users
  - individuals (researchers, students)
  - organizers (group leaders, publication coordinators)
    - using the ShaRef client for managing publications
    - using the ShaRef API through some other application

ETH Annual Report Preparation



A Question of Granularity

- Web Services are published interfaces
  - (almost) anything can be described
  - XML-structured data
  - function signatures (input/output?)
- Web Services can be combined
  - describe an application as message flow
  - numerous Web Service composition languages
  - so far there is no clear winner (maybe BPEL4WS)

Small: Software Components

- High modularity, maximize reuse
  - a sorting Web Service by an algorithm group
  - matrix computations by a scientific computing group
- Pros:
  - maximize reuse, minimize amount of code duplication
- Cons:
  - Web Services are designed for loose coupling
  - too much work for most developers
  - too many implications for the software design process
- → interesting concept, but not realistic today

Large: Application Services

- Low modularity
  - focus is on communicating applications
- Non-goals of this approach
  - robust distributed applications (Web Services may fail)
  - reuse of every possible software component
- Challenges of this approach
  - services may evolve (how to deal with change)
  - services must be found (how to publish services)
- Benefits of this approach
  - avoid duplication of existing services
  - promote Web Services as the way to provide Services

Some Thoughts

- Web Services are a very loose concept
  - basically, Web-style RPC or messaging
- Web Services can be used in many different ways
  - top-level design questions, messaging vs. RPC
  - implementation questions (Schema design)
  - application-level questions (authentication)
- Balance between Freedom and Ease-of-Use
  - design rules make Web Services easier to understand
  - design rules constrain Web Services
  - Goal: a good balance for providers and users

ETH Web Service Directory

- ETH Web service design guidelines
  - WSDL design guidelines, Schema design guidelines
  - Web Service Description guidelines
  - harvest Web Service Descriptions
    - compile a ETH Web Service Directory
    - accumulate the critical mass
- Web Service Directories:
  - UDDI: not successful, no use case, little use
  - "automated service discovery": nice demo idea
  - Reality: find a useful service, read documentation, use it

Compiling the Directory

- Provide a template for Web Service Descriptions
  - a template how to create Semantic Web content
    1. a simple and easy to use XML Schema
    2. a Web site with a form that generates the description
- Provide a site where to submit descriptions
  - accepts only ETH Web Service Descriptions
- Implement a Harvesting engine
  - retrieves all ETH Web Service Descriptions
  - extracts all machine-readable information
  - generates a directory of all ETH Web Services
- Different Publication Channels: HTML, UDDI, RDF

Thanks!

- `http://dret.net/projects/sharef/`

- Erik Wilde and Willy Müller, "Organizing Federal E-Government Standards", submitted to *Fourteenth International World Wide Web Conference (WWW2005)*, Chiba, Japan, May 2005. (available on request)
- Erik Wilde, "Semantically Extensible Schemas for Web Service Evolution", *European Conference on Web Services (ECOWS'04)*, pp. 30-45, Erfurt, Germany, September 2004. `http://dret.net/netdret/publications#wil04j`