Pollock: Automatic Generation of Virtual Web Services from Web Sites

Yi-Hsuan Lu
Yoojin Hong
Jinesh Varia
Dongwon Lee
March 14, 2005

Outline

- What is Web Services?
- Motivation
- Main Ideas: Pollock
- Application: OpenDBLP
- Conclusion
Web Services

- XML-based framework for machine interoperability
  - API: **WSDL**
  - Communication: **SOAP**
  - Yellow Page: **UDDI**
- Disguised RPC or CORBA in XML
- Let S/W agents communicate each other w/o human intervention (in theory)

By 2008, 75% of industry applications are expected to have Web Services components [Gartner report]

Tools for Web Services are needed to:
- Generate
- Discover
- Compose
- Analyze
- Optimize
- ...
Web Services Research @ Penn State, USA

- **Atherton** project is to develop tools/methodology for Web Services
  - [http://nike.psu.edu/atherton/](http://nike.psu.edu/atherton/)
- **Sub-projects**
  - Generation: Pollock [ACM SAC05]
  - Discovery and Composition: BF* [IEEE EEE05]
  - Analysis & Optimization: MISQ [IEEE BSN05]

Motivation

- Most commercial companies already have some sort of Web Pages, where users can browse and purchase items
- Most popular interface is **HTML FORM**
- The functionalities currently being provided through FORM are:
  - Common/popular activities
  - When companies decide to build Web Services based interfaces, these functionalities in FORM are likely to be part
Data in the “Hidden” Web

Motivation I: WS Generator

WWW

Human

Internet

FORM

Server

Semantic Web

Machines

Internet

Web Services Layer

Server

Pollock

French

FORM

Internet

Pollock

Machines
Using a given HTML FORM interface, Pollock does:

- Generates Web Services API (WSDL)
- Sits between Web Services API and FORM interface, and translate requests/responses back and forth (SOAP)
- Generates example Web Services client/server
Pollock

- Two technical challenges
- Interacting with FORM
  - Borrow from DB community’ Wrapper/Mediator techniques
  - We used Gatech’s XWrapElite
- Map btw FORM and Web Services
  - Need to identify functionality
  - Need to identify mandatory/optional parameters
  - Need to identify types

Foodgeek Example

- Two Functionalities
  - Search
  - Response
  - Search: <message>
  - Parameters: <part>
  - Type: <types>
  - Response: same…
  - Wire Search and Response via <portType>
  - Generate “foodgeek.wsdl”
Foodgeek Example

HTTP GET("title french OR rating+DE SC&category=9...")

SOAP-ENV:Envelope
  SOAP-ENV:Body
    ns1:FoodGeekSearch
      Search_by_TITLE_or_INGREDIENTS
        type="xsd:string"> title</>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

HTTP GET("search results...")

SOAP-ENV:Envelope
  SOAP-ENV:Body
    ns1:FoodGeekSearchResponse
      return
        resultElements type="soapenc:Array">
        item><title>French Onion Soup</title></>
      </item>...<item>
    </result>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
Applications

- Tried Pollock to 8 real web sites
  - eBay, Bigfoot, Wine, Foodgeeks, Yahoo, …
  - Can handle various FORM constructs

- Tried Pollock to DBLP ⇔ OpenDBLP

OpenDBLP: http://opendblp.psu.edu/
Conclusion

- Pollock as a tool to generate Web Services interfaces for HTML FORM interfaces
  - Providing Real Web Services
- Pollock as a middleware server running between web service client and HTML FORM interface
  - Providing Virtual Web Services
- Further details at:
  - http://nike.psu.edu/atherton/
  - http://opendblp.psu.edu/