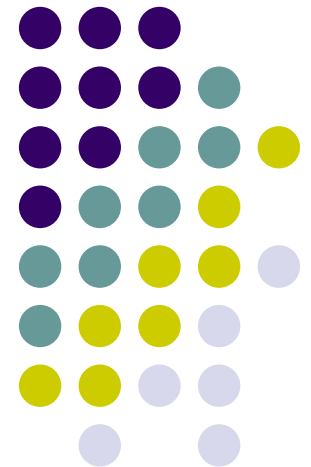
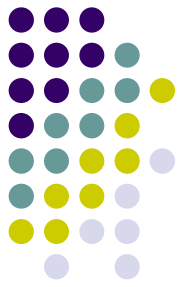


Web Services with JWIG and Xact

Christian Kirkegaard
BRICS, University of Aarhus

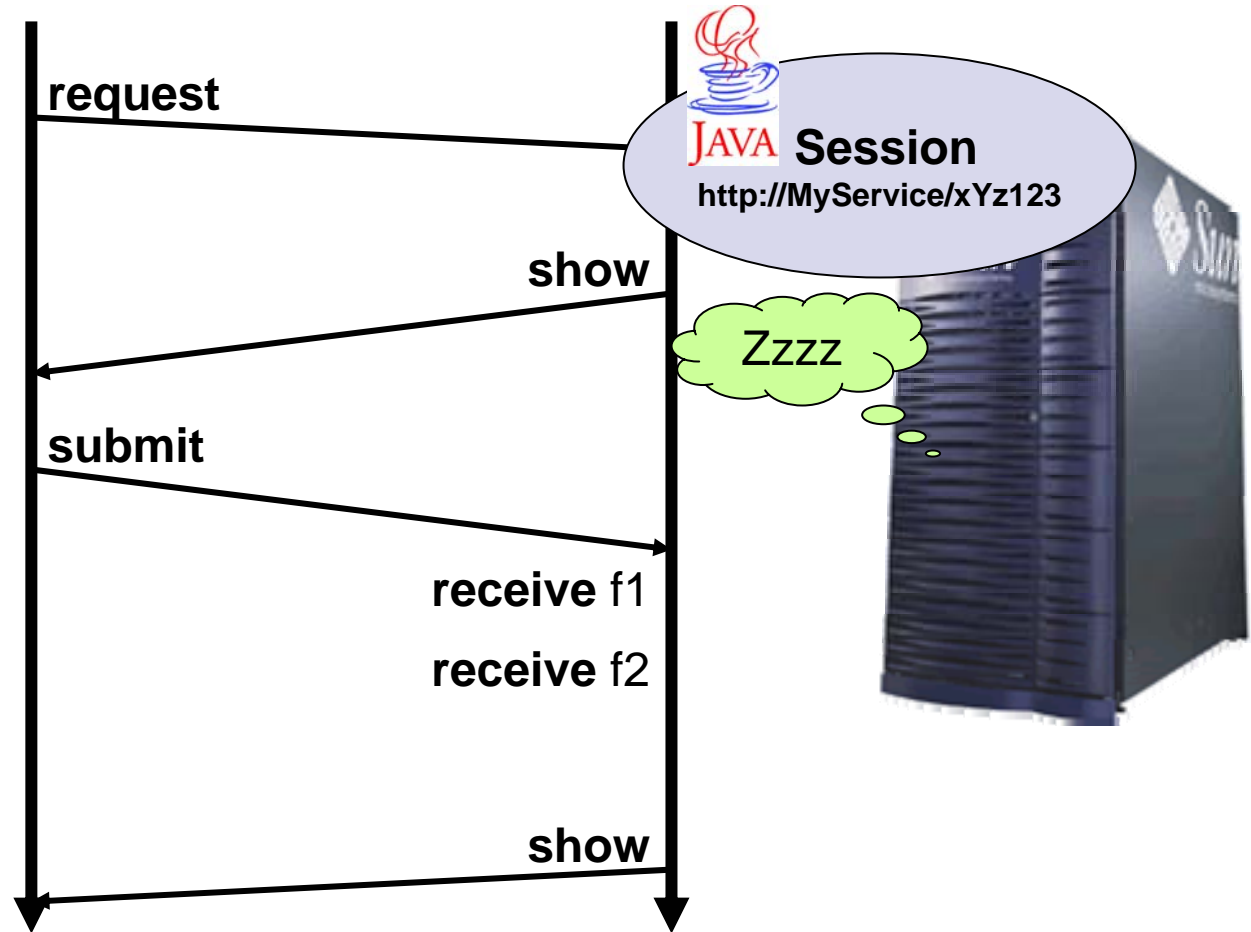




JWIG and Xact

- Extensions of Java for building Web Services
- Key features
 - Session-based execution model
 - High-level operations for XML manipulation
 - Static type analysis
 - **Checkable design contracts**

Session Threads



```
public class MyService extends Service {
    public class Main extends Session {

        DBConnection db = new DBConnection("mydatabase");

        public void main() {
            show [[ <html> ... </html> ]];      /* login page */

            String userid = receive userid;
            String password = receive password;

            if (db.isUser(userid,password))
                show [[ <html> ... </html> ]]; /* menu page */
            else
                exit [[ <html> ... </html> ]]; /* error page */

            ...

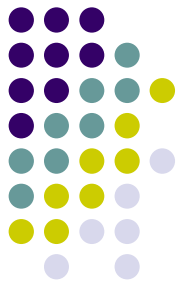
        }
    }
}
```



XML Templates

- Immutable well-formed XML fragments with named gaps

```
[[
  <html>
    <head>
      <title><[TITLE]></title>
    </head>
    <body bgcolor=[COLOR]>
      <[BODY]>
    </body>
  </html>
]]
```



XML Construction

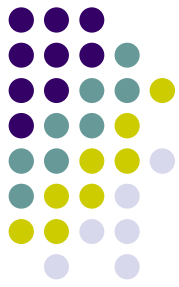
- The “plug” operation binds values to gaps

```
[[
  <html>
    <head>
      <title>My Links</title> <[ TITLE="My Links" ]
                                <[ COLOR="pink" ]
    </head>
    <body bgcolor="pink"> <[ BODY=
      [[
        <h1><[TITLE]></h1>
        <ul>
          <[ITEMS]>
        </ul>
      </body>
    </html>
  ]]
```



XML Deconstruction

- Use XPath to address sub-templates
- The “select” operation returns an array of all addressed sub-templates
- The “gapify” operation replaces all addressed sub-templates with new gaps



Static Analyses

- Show-receive analysis
 - Verify that all expected fields were shown to client
- Plug analysis
 - Verify that all addressed gaps exist
- XML validity analysis
 - Verify that all shown templates are valid XHTML documents

Checkable Design Contracts



- Separation of concerns
 - HTML designer and Web service programmer should cooperate but work independently
- Work scenario
 - Design basic template structure
 - Formalize in a simple design contract
 - HTML designer writes the templates
 - Programmer writes the code
- Automatically verify requirements!

Checkable Design Contracts



```
package org.jwig.examples.login

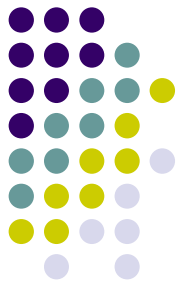
template Wrapper      { gap TITLE: <AnyString>,
                       gap BODY  }

template Login        { form { field user,
                              field password } }

template Menu         { gap ITEM: MenuItem }

template MenuItem     { gap ITEM: MenuItem }

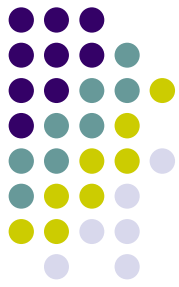
page Main: Wrapper   { gap Wrapper.BODY: Login }
```



Take-home points 1

Sessions and session-based execution

- ✓ Natural abstraction for many Web services
- ✓ Precise analysis and validation of form fields
- ✓ Simple to implement (in Java) using threads
- ✗ Page-like structure sometimes preferable
- ✗ Back-button navigation preferable (when possible)



Take-home points 2

XML manipulation using templates with gaps

- ✓ Flexible and natural abstraction
- ✓ Reuse of common XML fragments
- ✓ Precise static type analysis
- ✓ Separation of concerns using contracts
- ✗ Whole-program analysis
- ✗ Set of operations might be extended

JWIG and Xact



- More information online
 - <http://www.jwig.org/>
 - <http://www.brics.dk/Xact>
- People
 - Michael I. Schwartzbach
 - Anders Møller
 - Aske Simon Christensen
 - **Christian Kirkegaard**
 - Henning Böttger