Towards Generic and Flexible Web Services for E-Assessment

Mario Amelung  Peter Forbrig  Dietmar Rösner

Otto-von-Guericke-University Magdeburg

ITiCSE 2008
Agenda

Motivation

Implementation

Experience

Summary
**Motivation**

»How can automatic testing of (programming) assignments effectively be used to make CS courses more efficient?«

- For students
  - More interactivity and flexibility
  - More timely feedback
  - More discussion of solutions and problems
  - More programming practice

- For teachers
  - Improved opportunities to track students’ performance
  - Better support during the entire assessment process
  - Liberation from avoidable (administrative) work
Motivation

»How can automatic testing of (programming) assignments effectively be used to make CS courses more efficient?«

► For students
  ► More interactivity and flexibility
  ► More timely feedback
  ► More discussion of solutions and problems
  ► More programming practice

► For teachers
  ► Improved opportunities to track students’ performance
  ► Better support during the entire assessment process
  ► Liberation from avoidable (administrative) work
Motivation

»How can automatic testing of (programming) assignments effectively be used to make CS courses more efficient?«

► For students
  ▶ More interactivity and flexibility
  ▶ More timely feedback
  ▶ More discussion of solutions and problems
  ▶ More programming practice

► For teachers
  ▶ Improved opportunities to track students’ performance
  ▶ Better support during the entire assessment process
  ▶ Liberation from avoidable (administrative) work
Requirements

What we wished was a system or service which provides

- automatic testing and assessment functionality, especially for programming assignments,
- easy integration into already existing e-learning environments, and
- flexibility to adapt and reuse it for specific needs.
Requirements

What we wished was a system or service which provides

- automatic testing and assessment functionality, especially for programming assignments,
- easy integration into already existing e-learning environments, and
- flexibility to adapt and reuse it for specific needs.
Requirements

What we wished was a system or service which provides

- automatic testing and assessment functionality, especially for programming assignments,
- easy integration into already existing e-learning environments, and
- flexibility to adapt and reuse it for specific needs.
Spooler

A Web service which

- manages a submission/result queue and several backends,
- provides information about available backends,
- gets submissions including all necessary information for testing (e.g., students’ and model solution, and test data),
- forwards submissions for testing to the appropriate backend, and
- returns the test results to the frontend.
Spooler

Mario Amelung, Peter Forbrig, Dietmar Rösner

Generic and Flexible Web Services for E-Assessment
Backend

Also a Web service which

- provides syntax checking and testing for a specific programming language,
- usually in conjunction with the corresponding compiler and/or interpreter,
- defines a schema with information about input fields for testing, and
- take security precautions, e.g., by using chroot, jails, or systrace.

With the appropriate backends, the system can also be used to test submissions in other formal notations.
Backend

Also a Web service which

- provides syntax checking and testing for a specific programming language,
- usually in conjunction with the corresponding compiler and/or interpreter,
- defines a schema with information about input fields for testing, and
- take security precautions, e.g., by using chroot, jails, or systrace.

With the appropriate backends, the system can also be used to test submissions in other formal notations.
Backend

Server C

<<Backend>>
Haskell

<<Interpreter>>
runhugs

Mario Amelung, Peter Forbrig, Dietmar Rösner
Generic and Flexible Web Services for E-Assessment
Frontend

- Any general-purpose content management or a learning management system
- In our case we use Plone with **ECAutoAssessmentBox** which features
  - electronic submissions for (programming) assignments,
  - forward submissions to ECSpooler for automatic testing,
  - feedback on whether a submission yields the expected results,
  - support for the subsequent manual grading process, and
  - analysis and statistics about the performance and progress of individual students or the overall class.
Frontend

Server A

CMS/LMS

<<Frontend>>
ECAutoAssessmentBox

<<...>>
...

Mario Amelung, Peter Forbrig, Dietmar Rösner

Generic and Flexible Web Services for E-Assessment
System architecture

- Clearly separates frontends, spooler, and backends
- Offering a high degree of flexibility
- Enabling a variety of frontends and backends to be used
- Allows running the components on disparate operating systems and in different environments over a network
System architecture

- Clearly separates frontends, spooler, and backends
- Offering a high degree of flexibility
- Enabling a variety of frontends and backends to be used
- Allows running the components on disparate operating systems and in different environments over a network
System architecture

- Clearly separates frontends, spooler, and backends
- Offering a high degree of flexibility
- Enabling a variety of frontends and backends to be used
- Allows running the components on disparate operating systems and in different environments over a network
System architecture

- Clearly separates frontends, spooler, and backends
- Offering a high degree of flexibility
- Enabling a variety of frontends and backends to be used
- Allows running the components on disparate operating systems and in different environments over a network
System architecture

Server A

CMS/LMS

<<Frontend>>
ECAutoAssessmentBox

<<…>>
...

Server B

<<Spooler>>
ECSpooler

<<…>>
...

Server C

<<Backend>>
Haskell

<<Interpreter>>
runhugs

<<…>>
...

Server D

<<Backend>>
Scheme

<<Interpreter>>
mzscheme

<<…>>
...

Mario Amelung, Peter Forbrig, Dietmar Rösner

Generic and Flexible Web Services for E-Assessment
Employment Flexibility

Backends are

- available for Haskell, Scheme, CommonLisp, Erlang, Prolog, Python, and Java.
- possible for more programming languages or other formal notations.
- possible for different testing approaches (static vs. dynamic testing, black box vs. white box testing).
- easy to integrate since they are derived from general backend classes.
Employment Flexibility (cont.)

Depending on the available backends teachers can select from a variety of options for their assignments:

- Which programming languages, compiler or interpreter?
- Which testing approach, e.g., model solution, property specification, or unit tests?
- Which comparison function or equality predicate?
Motivation Implementation Experience Summary

Employment Flexibility (cont.)

Depending on the available backends teachers can select from a variety of options for their assignments:

▶ Which programming languages, compiler or interpreter?
▶ Which testing approach, e.g., model solution, property specification, or unit tests?
▶ Which comparison function or equality predicate?
Employment Flexibility (cont.)

Depending on the available backends teachers can select from a variety of options for their assignments:

- Which programming languages, compiler or interpreter?
- Which testing approach, e.g., model solution, property specification, or unit tests?
- Which comparison function or equality predicate?
Defining Test Data

Edit Auto Assessment Box

by Mario Amelung — last modified 2006-12-05 14:42  History

[default] [backend]
Enables the creation, submission and grading of automatically tested online assignments.

Automatically accept assignments
If selected, an assignment which passes all tests will be automatically accepted.

Tests
Select one or more tests.

Simple
Permutation

Test data
Enter one or more function calls. A function call consists of the function name (given in the exercise directives) and test data as parameters of this function. Each function call must be written in a single line.

fib 0
fib 1
fib 8
fib 12
Defining a Model Solution

**Model solution**
Enter a model solution.

```haskell
fib :: Integer -> Integer
fib n = fibGen 0 1 n

fibGen :: Integer -> Integer -> Integer -> Integer
fibGen a b n = case n of
    0 -> a
    n -> fibGen b (a + b) (n - 1)

-- Function fib returns the n'th Fibonacci number
-- It uses algorithm: binary recursion
--fib :: Integer -> Integer
--fib n | n == 0 = 0
--     | n == 1 = 1
```

Mario Amelung, Peter Forbrig, Dietmar Rösner

Generic and Flexible Web Services for E-Assessment
Case Study: Magdeburg

- Exercise sheets and assignments are online available using ECAutoAssessmentBox
- Submission period ends several hours prior to the weekly classroom session
- Students submit their programming solutions, get immediate feedback and if necessary resubmit
- Teachers get an overview of all submission before the classroom session
Case Study: Magdeburg

- Exercise sheets and assignments are online available using ECAutoAssessmentBox
- Submission period ends several hours prior to the weekly classroom session
- Students submit their programming solutions, get immediate feedback and if necessary resubmit
- Teachers get an overview of all submission before the classroom session
Case Study: Magdeburg

- Exercise sheets and assignments are online available using ECAutoAssessmentBox
- Submission period ends several hours prior to the weekly classroom session
- Students submit their programming solutions, get immediate feedback and if necessary resubmit
- Teachers get an overview of all submission before the classroom session
Case Study: Magdeburg

- Exercise sheets and assignments are online available using ECAutoAssessmentBox
- Submission period ends several hours prior to the weekly classroom session
- Students submit their programming solutions, get immediate feedback and if necessary resubmit
- Teachers get an overview of all submission before the classroom session
Exercise Sheet

Functional Programming

Haskell and Erlang

Haskell: fib
The Fibonacci numbers \( f_0, f_1, \ldots \) are defined by the rule that

- \( f_0 = 0 \),
- \( f_1 = 1 \) and
- \( f_{n+2} = f_n + f_{n+1} \)

for all \( n \geq 0 \). Give a definition of the function fib in Haskell that takes an integer \( n \) and returns \( f_n \).

▷ Work on this assignment

Erlang: fac
Define a function fac in Erlang which computes the factorial of an integer.

▷ Work on this assignment

Erlang: reverse
Define a function reverse/1 which reverses the order of the elements of a list.

reverse/1 should be the only function which can be called from outside the module. Therefore check your export declaration for the module twice.

▷ Work on this assignment
Submission

Haskell: fib
Submission period ends: 2007-06-08 16:00
You have 4 attempts left.
▲ Up one level

The Fibonacci numbers $f_0, f_1, \ldots$ are defined by the rule that

- $f_0 = 0$,
- $f_1 = 1$ and
- $f_{n+2} = f_n + f_{n+1}$

for all $n \geq 0$. Give a definition of the function \texttt{fib} in Haskell that takes an integer \texttt{n} and returns $f_n$.

Download answer template

Answer
Enter your answer for this assignment

\texttt{fib :: Integer \rightarrow Integer}

File
or upload a file (existing content will be replaced).

by Mario Amelung — last modified 2007-06-03 07:11
Feedback

Assignment of Milliken, Kate
submitted at 2006-03-23 14:51, state: Submitted

Answer:

```
fib :: Integer -> Integer
fib n
  | n == 0 = 0
  | n == 1 = 1
  | n >= 2 = fib(n-1) + fib(n-1)
```

Auto feedback:

Your submission failed. Test case was: 'fib 8' (simple)

Expected result: 21
Received result: 128

by Kate Milliken — last modified 2006-03-23 14:57
### Overview

**Haskell: fib**

<table>
<thead>
<tr>
<th>date</th>
<th>user</th>
<th>state</th>
<th>grade</th>
<th>actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007-04-10 16:13</td>
<td>Amelung, Mario</td>
<td>Superseded</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-04-18 16:14</td>
<td>Amelung, Mario</td>
<td>Pending</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-04-18 16:14</td>
<td>Amelung, Mario</td>
<td>Superseded</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-06-03 07:32</td>
<td>Milliken, Kate</td>
<td>Superseded</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-06-03 07:32</td>
<td>Milliken, Kate</td>
<td>Superseded</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-06-03 07:33</td>
<td>Milliken, Kate</td>
<td>Accepted</td>
<td>1.7</td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-06-03 07:34</td>
<td>Driscoll, John</td>
<td>Superseded</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-06-03 07:35</td>
<td>Driscoll, John</td>
<td>Superseded</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-06-03 07:36</td>
<td>Driscoll, John</td>
<td>Superseded</td>
<td></td>
<td>[View] [Grade] [Download]</td>
</tr>
<tr>
<td>2007-06-03 07:36</td>
<td>Driscoll, John</td>
<td>Graded</td>
<td>2.0</td>
<td>[View] [Grade] [Download]</td>
</tr>
</tbody>
</table>

**Average Grades**

- **Average Grade**: 2.00
- **Median Grade**: 2.00

Change State
Select the new state for the selected items.
- Grade
- Retract
- Review
- Accept
- Reject

Average and median grades are calculated from all assignments in this assignment box which are in the state graded and which have a grade assigned.
Case Study: Magdeburg (cont.)

- winter 2006: »Programming paradigms« (80+ students)
- summer 2007: »Programming paradigms« (60+ students)
- winter 2008: »Algorithm and data structures« (intended for 300+ students)

- Since 2006 about 12,000 submissions were automatically tested for more than 200 assignments.
Case Study: Magdeburg (cont.)

- winter 2006: »Programming paradigms« (80+ students)
- summer 2007: »Programming paradigms« (60+ students)
- winter 2008: »Algorithm and data structures« (intended for 300+ students)

- Since 2006 about 12,000 submissions were automatically tested for more than 200 assignments.
Case Study: Rostock

Module »Abstract Data Types« with about 200 students:

- 2 hours lecture, 1 hour exercise and 1 hour lab per week
- Students have to perform programming assignments at home and during lab hours
- As assignments students get algebraic specifications
- Students have to provide the Haskell version of the specification which is an executable Haskell program
- ECAutoAssessmentBox is used as frontend for submissions
- Automatic testing is done by ECSpooler and the Haskell backend

Mario Amelung, Peter Forbrig, Dietmar Rösner

Generic and Flexible Web Services for E-Assessment
Case Study: Rostock (cont.)

- Summer 2007: »Algorithm and data structures«
- First year computer science course with about 140 students
- Teaching algorithm and data structures with Java
- Programming assignments are automatically tested with the Java backend
Conclusion

- The implementation as Web services and the separation of frontends, spooler, and backends, offers a high degree of flexibility.
- **ECAutoAssessmentBox**, ECSpooler and several backends are successfully used at two different universities.
- Recent development: backend for *Standard ML (SML)* at LMU Munich.
Conclusion

➢ The implementation as Web services and the separation of frontends, spooler, and backends, offers a high degree of flexibility.

➢ ECAutoAssessmentBox, ECSpooler and several backends are successfully used at two different universities.

➢ Recent development: backend for Standard ML (SML) at LMU Munich
Conclusion

- The implementation as Web services and the separation of frontends, spooler, and backends, offers a high degree of flexibility.
- **ECAutoAssessmentBox**, ECSpooler and several backends are successfully used at two different universities.
- Recent development: backend for *Standard ML (SML)* at LMU Munich
Future work

- More formal evaluation with a questionnaire for the involved students
- More flexibility in the feedback reporting
- Extension module for *OpenCMS* to allow automatic testing using ECSpooler
Further Information

► Web page:
   http://wdok.cs.uni-magdeburg.de/

► Software download:
   http://wdok.cs.uni-magdeburg.de/software/

► Demo server:
   http://wdok.cs.uni-magdeburg.de/demo/