

Bartelsen, Jan; Brauer, Johannes; Rehder, Heiko

**Working Paper**

## Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen

Arbeitspapiere der Nordakademie, No. 2010-03

**Provided in Cooperation with:**

Nordakademie - Hochschule der Wirtschaft, Elmshorn

*Suggested Citation:* Bartelsen, Jan; Brauer, Johannes; Rehder, Heiko (2010) : Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen, Arbeitspapiere der Nordakademie, No. 2010-03, Nordakademie - Hochschule der Wirtschaft, Elmshorn

This Version is available at:

<https://hdl.handle.net/10419/38612>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



**NORDAKADEMIE**  
HOCHSCHULE DER WIRTSCHAFT

**ARBEITSPAPIERE DER NORDAKADEMIE**  
ISSN 1860-0360

Nr. 2010-03

## **Ideen für die Neugestaltung der Programmier- ausbildung durch Entwicklung von Webanwen- dungen**

Jan Bartelsen  
Johannes Brauer  
Heiko Rehder

Juni 2010

Dieses Arbeitspapier ist als PDF verfügbar: <http://www.nordakademie.de/arbeitspapier.html>



**NORDAKADEMIE**  
HOCHSCHULE DER WIRTSCHAFT



Köllner Chaussee 11  
25337 Elmshorn  
<http://www.nordakademie.de>



# Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen

Jan Bartelsen, Johannes Brauer, Heiko Rehder

17. Juni 2010

## Inhaltsverzeichnis

<b>1</b>	<b>Ausgangssituation</b>	<b>1</b>
<b>2</b>	<b>Weiterentwicklung der Programmiergrundausbildung</b>	<b>4</b>
2.1	Integration von Lern- und Entwicklungsumgebung . . .	5
2.2	Objektorientierte Entwicklung von Web-Anwendungen von Anfang an . . . . .	7
<b>3</b>	<b>Die nächsten Schritte</b>	<b>8</b>
3.1	Projektschritte, Untersuchungsgegenstände . . . . .	8
3.2	Technische Aspekte . . . . .	9
3.3	Didaktische Aspekte . . . . .	10
<b>4</b>	<b>Zusammenfassung</b>	<b>10</b>

Getragen von dem beständigen Bemühen, die Programmiergrundausbildung für die Studierenden attraktiver zu gestalten und gleichzeitig ihren Lernerfolg zu verbessern, wird ein Ansatz beschrieben, den im Studiengang Wirtschaftsinformatik der NORDAKADEMIE seit ca. 10 Jahren verfolgten Objects-first-Ansatz weiter zu entwickeln. Immer mehr Studienanfänger bringen bereits Grundkenntnisse in Web-Techniken mit. Die Entwicklung von Web-Anwendung erhält immer größeres Gewicht. Diesen Tatsachen Rechnung tragend, werden erste Überlegungen angestellt, einen innovativen Programmierkurs für Anfänger unter dem Motto „Objektorientierte Entwicklung von Web-Anwendungen“ zu entwickeln .

## 1 Ausgangssituation

Seit dem Wintersemester 1999/2000 steht die Programmiergrundausbildung im Studiengang Wirtschaftsinformatik der NORDAKADEMIE unter dem Motto „Objektorientierung von Anfang an“<sup>1</sup> und wird unter Verwendung der Programmiersprache Smalltalk durchgeführt. Die Hochschule folgte damals dem Trend, die in den neunziger Jahren noch überwiegend „prozedural“ geprägte Ausbildung stärker auf die Objektorientierung hin auszurichten. Vollerorts setzte sich die Ansicht durch, dass sich die objektorientierte Denkweise in den Köpfen der Studierenden am Besten verankern lässt, wenn man objektorientierte Programmierung von Beginn an unterrichtet. Dieser Ansatz wurde und wird unter dem Stichwort „Objects first“ auf vielen Konferenzen, die sich mit der Didaktik der Programmierausbildung beschäftigen, diskutiert. Ohne dass das hier mit Zahlen untermauert werden soll, drängt sich der Eindruck auf, dass der Objects-first-Ansatz<sup>2</sup> inzwischen vorherrschend ist.

<sup>1</sup> BRAUER, J.: *Grundkurs Smalltalk - Objektorientierung von Anfang an*. Vieweg+Teubner Verlag, 3. Aufl., 2008

Zum Beispiel:

- SIGCSE Technical Symposium on Computer Science Education
- ITiCSE Innovation and Technology in Computer Science Education

<sup>2</sup> KÖLLING, M., B. QUIG, A. PATTERSON und J. ROSENBERG: *The BlueJ system and its pedagogy*. Journal of Computer Science Education, 13(4), December 2003

Dabei kommt als Programmiersprache häufig Java zum Einsatz, wobei aber immer wieder beklagt wird, dass diese Sprache für Anfänger eher nicht geeignet ist. Um Studierenden den Einstieg über Java in die objektorientierte Programmierung zu erleichtern, wurde z. B. mit *BlueJ*<sup>3</sup> sogar eine Java-Entwicklungsumgebung speziell für Lehrzwecke entwickelt.

IN JÜNGSTER ZEIT kann allerdings beobachtet werden, dass der Einstieg in die Programmierausbildung über die Objektorientierung schon wieder infrage gestellt wird.<sup>4</sup> Es wird eine Rückkehr zur prozeduralen Programmierung in der Grundausbildung gefordert unter Beibehaltung der Programmiersprache Java. Auch diese Wendung hat inzwischen in neuen Lehrbüchern<sup>5</sup> ihren Niederschlag gefunden. Gelegentlich wird davon gesprochen, dass die Informatik heute mit der Einführung der Objektorientierung in die Programmiergrundausbildung ein ähnliches Fiasko erlebt wie in den 1960er/1970er Jahren die Mathematik mit der Einführung der Mengenlehre in der Grundschule.

Aufgrund der in den vergangenen Jahren mit dem Objects-first-Ansatz in der Programmierausbildung an der NORDAKADEMIE gemachten positiven Erfahrungen sehen wir keinen Anlass zur prozeduralen Programmierung zurück zu kehren. Auch die Wahl der Programmiersprache soll an dieser Stelle nicht weiter diskutiert werden.

IM WINTERSEMESTER 2006/07 wurde damit begonnen, die programmierpraktischen Übungen in den beiden ersten Semestern von Einzelaufgaben auf Semesteraufgaben umzustellen, da die Konzepte der Objektorientierung anhand kleiner Übungsaufgaben nur schwer motiviert und vermittelt werden können. Um die Studierenden in der Selbständigkeit des Wissenserwerbs zu stärken und zu motivieren, wurde gleichzeitig die „klassische“ Vorlesung durch einen web-gestützten Kurs auf der Grundlage der Lernplattform *moodle*<sup>6</sup> ersetzt. Diese Umstellung wurde auch durch die Tatsache „beflügelt“, dass der gleichzeitige Wechsel von Diplom- zu Bachelor-Studiengängen eine Reduzierung der zur Verfügung stehenden Unterrichtsstunden mit sich brachte und eine Überarbeitung des Konzepts der Lehrveranstaltung ohnehin notwendig war.

DIE UMSTELLUNG AUF *moodle* war zu Beginn eine „Ad-hoc-Aktivität“, die anfangs im Wesentlichen darin bestand, dort die Programmieraufgabe in Teilschritten zu beschreiben und Verknüpfungen auf Teile des existierenden Vorlesungsskripts (PDF-Dokumente) zu hinterlegen.

Den Einsatz der Lernplattform zu systematisieren, war Gegenstand der Master-Thesis von Jan Bartelsen<sup>7</sup>. Die dort entwickelten Ansätze wurden danach im Projekt NORDPOL<sup>8</sup> fortgeführt, so dass nun ein Blended-learning-Kurs „Programmierung 1“ zur Ver-

<sup>3</sup> BARNES, D. J. und M. KÖLLING: *Objects First with Java*. Pearson Education, 2008

<sup>4</sup> REGES, S.: *Back to basics in CS1 and CS2*. SIGSE Bull., 38(1):293–297, 2006

<sup>5</sup> REGES, S. und M. STEPP: *Building Java Programs: A Back to Basics Approach*. Addison-Wesley, Mai 2007

In der Literatur auch als *Neue Mathematik* bekannt

Der Vollständigkeit halber sollte erwähnt werden, dass mancherorts nach wie vor die funktionale Programmierung als der geeignete Einstieg in die Programmierung angesehen wird:

FELLEISEN, M., R. B. FINDLER, M. FLATT und S. KRISHNAMURTHI: *How to Design Programs*. The MIT Press, September 2003

oder:

KLAEREN, H. und M. SPERBER: *Die Macht der Abstraktion*. B. G. Teubner, 1. Aufl., 2007

<sup>6</sup> <http://www.moodle.org>

<sup>7</sup> BARTELTSEN, J.-I.: *Entwurf einer kooperativen Lehrveranstaltung in moodle für die Grundausbildung in der Programmierung*. Master-Thesis an der FernUniversität Hagen, 2008

<sup>8</sup> BARTELTSEN, J. und J. BRAUER: *Kooperatives Lernen mit moodle in der Programmierausbildung*. Arbeitspapier 2008-6, NORDAKADEMIE Hochschule der Wirtschaft, Oktober 2008

fügung steht, der von den Studierenden gut akzeptiert ist.<sup>9</sup>

ALS ENTWICKLUNGSUMGEBUNG wurde von Beginn an *VisualWorks*<sup>10</sup> von *Cincom* verwendet. Dabei handelt es sich um ein professionelles Werkzeug, das in keiner Weise auf die Belange des Lehrens und Lernens von Smalltalk ausgerichtet ist. So hat sich durch wiederholte Befragungen von Studierenden gezeigt, dass die „Reibungsverluste“ durch die Komplexität von *VisualWorks* doch recht hoch sind. Eine mit *BlueJ* (s. o) vergleichbare, auf die Bedürfnisse von Anfängern zugeschnittene Entwicklungsumgebung steht für Smalltalk leider nicht zur Verfügung.

Der Einsatz von *Squeak*<sup>11</sup> ist ebenfalls untersucht worden, aber zum damaligen Zeitpunkt verworfen worden. Der für den Einsatz im Unterricht notwendige „Reifegrad“ schien 1999/2000 noch nicht erreicht, Dokumentation war nicht vorhanden und die „bonbonartige“ Oberfläche machte einen wenig professionellen Eindruck und schien daher kaum geeignet, die beim Einsatz von Smalltalk ohnehin existierenden Akzeptanzprobleme abzumildern. Inzwischen ist die Entwicklung von *Squeak* so weit voran geschritten, dass ein Einsatz in der Lehre durchaus infrage kommt. Das Gleiche gilt für das *Squeak*-Derivat *Pharo*<sup>12</sup>. Allerdings unterscheiden sich diese Entwicklungsumgebungen in ihrer Komplexität kaum von *VisualWorks*, so dass hinsichtlich des Einsatzes in der Lehre kaum Vorteile zu erwarten sind.

PARALLEL ZUR LEHRVERANSTALTUNG *Programmierung 1* findet im ersten und zweiten Semester des Studiengangs Wirtschaftsinformatik die Lehrveranstaltung *Internet-Grundlagen* statt, in der die Studierenden grundlegende Inhalte zu HTML, XML, CSS und JavaScript sowie zur Gestaltung von Web-Seiten erlernen und als Prüfungsleistung (Hausarbeit) einen einfachen Web-Auftritt gestalten. Diese Lehrveranstaltung leidet seit geraumer Zeit darunter, dass der Umfang der vorhandenen Vorkenntnisse sehr heterogen ist. So erscheinen auf der einen Seite Studierende, für die der Themenkomplex HTML vollkommen neu ist und die noch nie eine Webseite erstellt haben. Auf der anderen Seite gibt es Studierende, die bereits seit mehreren Jahren, teils sogar professionell, mit den Themen der Lehrveranstaltung *Internet-Grundlagen* arbeiten. Für den Dozenten ist es daher oft schwer, einen Unterricht zu gestalten, der niemanden über-, aber auch niemanden unterfordert.

Insbesondere bei den Grundlagen (HTML) war es zunehmend schwieriger, den Studierenden noch etwas neues zu vermitteln.

Seit dem WS 2007/08 ist man daher dazu übergegangen, HTML-Kenntnisse im Rahmen der Vorlesung *Internet-Grundlagen* nicht mehr zu vermitteln, sondern diese Inhalte vorauszusetzen und entsprechend mit CSS und XML anzufangen, wobei es derzeit noch eine Schnittmenge zwischen Seminar und Unterricht gibt, die aber mit dem kommenden Jahrgang (also im WS 2010/11) reduziert werden soll. Ganz herausgenommen werden soll HTML jedoch

<sup>9</sup> BARTELTSEN, J., J. BRAUER und H. REHDER: *Abschlussbericht für das Projekt NORDPOL*. Arbeitspapier 2010-02, NORDAKADEMIE Hochschule der Wirtschaft, Juni 2010

<sup>10</sup> <http://www.cincomsmalltalk.com>

Kurzzeitig wurde auch die auf *VisualWorks* aufbauende Lernumgebung *LearningWorks* benutzt. Dies wurde aber wieder beendet, da *LearningWorks* nicht mehr weiter entwickelt wird.

GOLDBERG, A., S. T. ABELL und D. LEIBS: *The LearningWorks development and delivery frameworks*. Commun. ACM, 40(10):78–81, 1997

<sup>11</sup> <http://www.squeak.org>

<sup>12</sup> <http://www.pharo-project.org>

nicht, da die Erfahrung gemacht wurde, dass HTML zwar oft bekannt ist, aber bei vielen kein Bewusstsein für validen (X)HTML-Code vorhanden ist, so dass dies weiterhin Bestandteil der Vorlesung ist. Für diejenigen Studierenden, die komplett ohne HTML-Vorkenntnisse ihr Studium beginnen, wird seit dem WS 2007/08 ein (erst zweitägiger, dann eintägiger) Kompaktkurs am Wochenende zu Beginn des 1. Semesters angeboten, wo HTML vertief gelehrt wird<sup>13</sup>. Aber auch die verbliebene „Rumpf-Veranstaltung“ ist nicht ohne Probleme. Die großen Unterschiede im Bereich der Vorkenntnisse werden auch hier deutlich und führen dazu, dass Studierende sich von der Vorlesung befreien lassen oder während der Übungen den Unterrichtsraum verlassen. Auch hier stellt sich die Frage, wie man den Studierenden mit Vorkenntnissen noch etwas neues vermitteln kann, ohne die „Anfänger“ zu überfordern.

<sup>13</sup> Da hier nur durchschnittlich 19% der Studienanfänger des Studiengangs *Wirtschaftsinformatik* teilgenommen haben, ist davon auszugehen, dass die Entscheidung, diese Inhalte aus dem Unterrichtsstoff herauszunehmen, richtig war, da offenbar ein Großteil der Studienanfänger hier Vorkenntnisse besitzt.

## 2 Weiterentwicklung der Programmiergrundausbildung

In dem fortwährenden Bemühen, die Qualität der Lehre im Studiengang Wirtschaftsinformatik der NORDAKADEMIE zu verbessern, kommt unseres Erachtens den Informatik-Veranstaltungen des ersten und zweiten Semesters eine besondere Bedeutung zu. An dieser Stelle soll ein besonderes Augenmerk auf die bereits erwähnten Lehrveranstaltungen *Programmierung 1* und *Internet-Grundlagen* gelegt werden.

Das Verbesserungspotential im Modul *Programmierung 1* liegt insbesondere auf den folgenden Gebieten:

1. Bisher wurden in diesem Modul von den Studierenden nur „dunkle“ Programme (ohne graphische Benutzungsoberfläche) geschrieben. Das wird von den Studierenden teilweise als unbefriedigend empfunden. Zwar lenkt aus didaktischer Sicht die Beschäftigung mit graphischen Benutzungsoberflächen vom eigentlichen Lernziel eher ab und erhöht auch die „Umgebungscomplexität“, da der Umgang mit entsprechenden Rahmenwerken zusätzlich erlernt werden muss. Aber möglicherweise kann die Motivation der Studierenden für ihre Programmierprojekte durch eine sichtbare Bedienoberfläche gesteigert werden, was wiederum dem Lernerfolg zugute kommen könnte.
2. Regelmäßig durchgeführte Befragung der Studierenden haben ergeben, dass der Einstieg in die objektorientierte Programmierung weiter vereinfacht werden sollte.
3. Das „Zuviele-offene-Fenster-Syndrom“ scheint nach Aussagen von Studierenden das Hauptproblem beim Umgang mit *VisualWorks* zu sein. Das Problem wird sicherlich noch dadurch verschärft, dass neben der Entwicklungsumgebung in einer Blended-learning-Umgebung mindestens ein Browser-Fenster geöffnet sein muss, um die aktuelle Kurslektion bzw. Programmieraufgabe lesen zu können.

Eine Programmierausbildung mit grafischen Oberflächen kann jedoch auch Probleme bereiten. So hat einer der Autoren in einem anderen Kurs die Erfahrung gemacht, dass Studierende nach der Bewertung einer Semesterarbeit die Kritik äußerten „Wir haben uns doch so viel Mühe mit der Oberfläche gegeben. Warum haben wir trotzdem eine schlechte Note erhalten?“ Es wurde so viel Zeit mit der Oberflächengestaltung verwendet, so dass für die eigentliche Programmierung wenig Zeit bleibt, was die entsprechend „schlechte“ Note zur Folge hatte.

Vielen Studierenden bereitet z. B. die Unterscheidung von Klassen- und Exemplarmethoden zu Beginn große Probleme.

## 2.1 Integration von Lern- und Entwicklungsumgebung

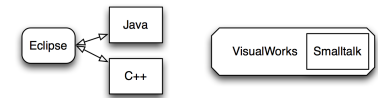
An dieser Stelle soll zunächst auf das dritte der am Ende des vorangegangenen Abschnitts genannten Probleme eingegangen werden.

Eine charakteristische Eigenschaft von Smalltalk besteht in der engen Verzahnung von Programmiersprache und Entwicklungsumgebung. Das mag unter abstrakten software-technischen Gesichtspunkten als ein Mangel an Modularität betrachtet werden, hat aber andererseits den Vorteil, dass die Entwicklungsumgebung optimal auf die Programmiersprache und das Arbeiten mit ihr abgestimmt ist. Andere Entwicklungsumgebungen lassen sich leichter mit verschiedenen Programmiersprachen benutzen (vgl. Abbildung 1).

EINE ENTWICKLUNGSUMGEBUNG dient zur Entwicklung von Anwendungen in einer bestimmten Programmiersprache und ihre enge Verzahnung oder Integration kann also vorteilhaft sein. Übertragen wir diesen Gedanken vom Entwicklungsprozess auf den Lernprozess im Kontext der Programmierausbildung könnte man folgendes formulieren:

Eine Lernumgebung dient zum Erlernen des Programmierens (unter Verwendung von bestimmten Programmiersprachen).

Smalltalk-Anwendungen werden „subtraktiv“ entwickelt.



höhere Modularisierung ← → höhere Integration

Abbildung 1: Integration vs. Modularisierung bei Entwicklungsumgebungen

Ein Beispiel für eine rudimentäre in die Entwicklungsumgebung integrierte Lernumgebung ist der Lesson Browser in VisualWorks, vgl. Abb. 2.

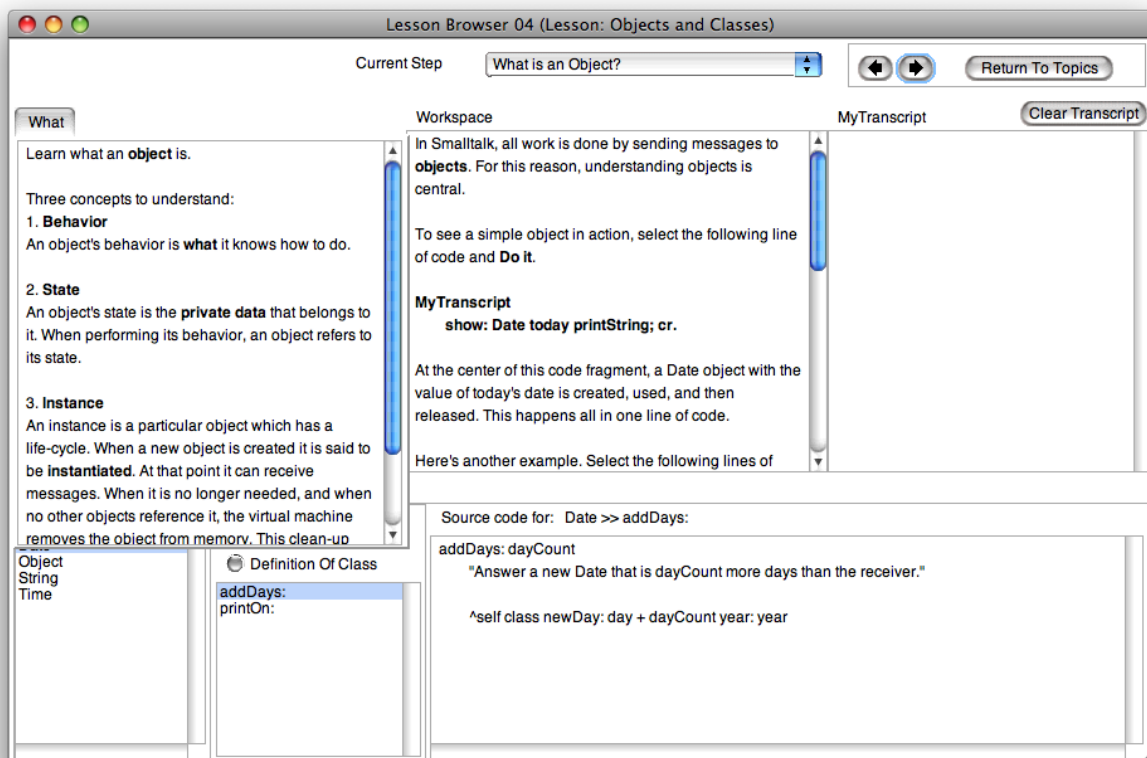


Abbildung 2: Lesson Browser in Cincom Smalltalk



Die Nutzung einer Entwicklungsumgebung kann man an dieser Stelle zwar als notwendiges Übel betrachten, aber wenn sie schon mal da bzw. unvermeidlich ist, kann man sie doch dafür nutzen, die Lern- in die Entwicklungsumgebung zu integrieren, um auf diese Weise den Lernprozess optimal zu unterstützen.

*Moodle* ist eine universell einsetzbare Lernumgebung, die eine Reihe von Lernaktivitäten (Aufgaben, Tests, ...) zur Verfügung stellt und das Einstellen von Arbeitsmaterialien in Form von Textseiten ermöglicht. D. h. aber *moodle* ist in keiner Weise auf das Erlernen des Programmierens zugeschnitten, das war bei der Konzeption von *moodle* natürlich auch gar nicht beabsichtigt.

Unseres Erachtens könnte es nun vorteilhaft sein, eine web-basierte Integration von Lern- und Entwicklungsumgebung zu bauen (vgl. Abbildung 3), um

- die Lernumgebung auf die speziellen Erfordernisse der Programmierausbildung zuzuschneiden,
- die technische Trennung von Lern- und Entwicklungsumgebung aufzuheben und damit
- ein nahtloses Lernen und Entwickeln zu ermöglichen.

HIER STELLT SICH NATÜRLICH SCHON DIE GRUNDSÄTZLICHE FRAGE: Womit ist der Aufwand zu rechtfertigen eine solche integrierte Lern- und Entwicklungsumgebung zu bauen? Warum soll hier das Rad neu erfunden werden?

Zur Beantwortung dieser Frage sei einmal die folgende kleine Analogiebetachtung erlaubt: Eine ähnliches Problem stellt sich in Unternehmen beim Aufbau von betrieblichen Informationssystemen. Warum entwickelt ein Unternehmen eine Individual-Software anstatt „SAP zu nehmen“?

- um Wettbewerbsvorteile erzielen?
- in der Regel nicht, um SAP Konkurrenz zu machen?
- um eine für den Anwendungsfall optimal zugeschnittene Lösung zu bekommen?
- um seine Geschäftsprozesse nicht an SAP anpassen zu müssen?
- um die Funktionsvielfalt von SAP, die gar nicht benötigt wird, auch nicht bezahlen zu müssen?
- um nicht durch „eigenmächtige“ Erweiterungen der Standard-Software in die „Personalisierungsfalle“ zu tappen.

Ersetzt man „SAP“ durch „*moodle*“ und „Geschäftsprozess“ durch „Lernprozess“ bleiben die obigen Sätze nahezu gültig. So wurde z. B. infolge der im Rahmen des Projekts NORDPOL<sup>14</sup> durchgeführten Untersuchungen der Lernprozess an die von *moodle* bereitgestellten Funktionalitäten angepasst. Man könnte aber auch die Frage stellen: Welche Funktionen soll eine Lernumgebung bieten, um den Lernprozess in der Programmierausbildung optimal zu unterstützen?

AUS TECHNISCHER SICHT kommen für die Realisierung einer integrierten Lern- und Entwicklungsumgebung Systeme in Betracht,

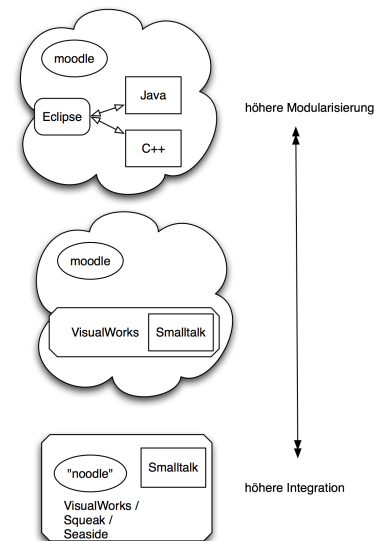


Abbildung 3: Integration von Lern- und Entwicklungsumgebungen

„Make or Buy“

<sup>14</sup> BARTELSSEN, J., J. BRAUER und H. REHDER: Abschlussbericht für das Projekt NORDPOL. Arbeitspapier 2010-02, NORDAKADEMIE Hochschule der Wirtschaft, Juni 2010

die rund um das Rahmenwerk *Seaside*<sup>15</sup> entstanden sind.

Hier wäre z. B. das Content-Management-System *Pier*<sup>16</sup> zu nennen, das erlaubt

- Texte vom Web Browser aus zu erfassen und zu bearbeiten,
- diese Texte z. B. per LaTeX-Export zu publizieren sowie
- beliebige *Seaside*-Applikationen direkt in den Text einzubetten<sup>17</sup>.

Die Firma *Cincom* bietet seit geraumer Zeit unter dem Namen *WebVelocity*<sup>18</sup> eine Entwicklungsumgebung für *Seaside*-Smalltalk-Anwendungen an, die selbst vollständig im Web-Browser läuft und mithilfe von *Seaside* realisiert wurde.

Die Entwicklungsumgebung *seaBreeze*<sup>19</sup> der Firma *Georg Heeg eK*<sup>20</sup> stellt innerhalb eines Web-Browser-Fensters einen „WYSIWYG design editor“ für die Entwicklung von *Seaside*-Applikationen zur Verfügung.

ZU BEGINN DES PROJEKTS wird die Entwicklung einer eigenen Lern- und Entwicklungsumgebung allerdings nicht im Vordergrund stehen. Stattdessen soll zunächst eine Lehrveranstaltung neuen Typs entwickelt werden, bei der die Inhalte der Module *Programmierung 1* und *Internet- Grundlagen* auf die im Abschnitt 2.2 beschriebene Weise zusammen geführt werden. Dieses Vorhaben kann einerseits ohne die aufwändige Entwicklung einer neuen Lernumgebung umgesetzt werden und andererseits können damit bereits einige der mit den existierenden Modulen identifizierten Probleme gelöst werden.

## 2.2 Objektorientierte Entwicklung von Web-Anwendungen von Anfang an

Insbesondere die Diskussionen um die in ihrem jetzigen Zuschnitt nicht mehr sinnvolle Lehrveranstaltung *Internet-Grundlagen* haben den Gedanken aufkommen lassen, die Module *Programmierung 1* und *Internet-Grundlagen* zusammen zu legen und daraus eine neue Lehrveranstaltung unter dem Motto

„Objektorientierte Entwicklung von Web-Anwendungen von  
Anfang an“

zu kreieren. Mit einer derartigen Veranstaltung könnten mehrere Probleme gleichzeitig gelöst werden:

1. Die „ungeliebte“ Lehrveranstaltung *Internet-Grundlagen* wird als eigenständiges Modul obsolet. Die bisher dort vermittelten Inhalte werden – sozusagen nebenbei – im Zuge der Entwicklung einer Web-Anwendung mit gelehrt und gelernt.
2. Die Lehrveranstaltung *Programmierung 1* wird attraktiver:
  - (a) Das Problem der „dunklen“ Programme (ohne graphische Benutzungsoberfläche) wird dadurch beseitigt, dass die Studierenden von Beginn an eine Anwendung mit einer Web-Oberfläche entwickeln.

<sup>15</sup> Rahmenwerk für die Entwicklung von Web-Anwendungen in Smalltalk (<http://seaside.st>)

DUCASSE, S., L. RENGGLI, D. C. SHAFFER und R. ZACCONE: *Dynamic Web Development with Seaside*. Square Bracket Associates, 2010

<sup>16</sup> <http://www.piercms.com/>

<sup>17</sup> vgl. dazu <http://www.seaside.st/about/examples/halos>

<sup>18</sup> <http://www.web-velocity.com>

<sup>19</sup> <http://seabreeze.heeg.de/>

<sup>20</sup> <http://www.heeg.de/>

- (b) Der zunehmenden Bedeutung von Web-Anwendungen wird Rechnung getragen<sup>21</sup>.
3. Durch die Zusammenfassung der Unterrichtsstunden beider Module steht mehr Zeit für die Entwicklung interessanterer Anwendungen zur Verfügung.
  4. Projekt-/problemorientiertes Lernen wird gefördert.  
Für die Entwicklung der Web-Anwendungen das Rahmenwerk *Seaside* einzusetzen, liegt aufgrund der nahtlosen Integration in fast alle gängigen Smalltalk-Entwicklungsumgebungen nahe. Darüber hinaus kann die Verwendung einer web-basierten Entwicklungsumgebung wie *WebVelocity* das o. g. Komplexitätsproblem von *VisualWorks* lösen helfen, da sich die gesamte Entwicklung im Browser abspielt.

Da auch die Lernumgebung *moodle* im Web-Browser läuft, brauchen die Studierenden für ihren Lern- bzw. Entwicklungsprozess den Browser nicht mehr zu verlassen.

Die Überlegungen zu einer solchen Lehrveranstaltung sollen nun im Projekt *WESTPOL* (WEBorientierter SmallTalk Programmierunterricht OnLine) zusammengefasst und weiter ausgebaut werden.

### 3 Die nächsten Schritte

Die Entwicklung der Lehrveranstaltung, wie sie in Abschnitt 2.2 skizziert wird, stellt ein in vielerlei Hinsicht innovatives Vorhaben dar:

1. Den Autoren ist keine vergleichbare Lehrveranstaltung bekannt. Es gibt zahlreiche Lehrbücher zum Thema „Entwicklung von Web-Anwendungen“ und auch der Object-first-Ansatz ist durch die Literatur gut abgedeckt. Aber beides in einer Lehrveranstaltung für Anfänger zusammen zu bringen, ist bisher ohne Beispiel.
2. Es entsteht ein „Großmodul“, für das es bisher kein Beispiel im Bachelor-Studiengang Wirtschaftsinformatik gibt und dessen organisatorische Einbettung in das Curriculum eine Reihe von Fragen aufwirft.<sup>22</sup>

<sup>21</sup> WRIGHT, A.: *Ready for a Web OS?*. Commun. ACM, 52(12):16–17, 2009

vgl. aber z. B.

STAPP, M., J. MILLER und V. KIRST: *A "CS 1.5" introduction to web programming*. In: *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, S. 121–125, New York, NY, USA, 2009. ACM

<sup>22</sup> vgl. Abschnitt 3.1

#### 3.1 Projektschritte, Untersuchungsgegenstände

Um zu vermeiden, das Rad vollständig neu zu erfinden, sollten einschlägige Publikationen darauf hin untersucht werden, ob sich vielleicht doch Beispiele für ähnliche Ansätze für eine Gestaltung der Programmierausbildung finden lassen. Die wissenschaftliche Diskussion auf diesem Gebiet ist sehr umfangreich.

FÜR DIE ENTWICKLUNG EINES BLENDED-LEARNING-KURSES sind die ersten Lektionen von besonderer Bedeutung. Hier sind u.ä. die folgenden inhaltlichen Fragen zu klären:

- Welche Kenntnisse und Fertigkeiten für die Entwicklung von Web-Anwendungen müssen und können an Anfänger vermittelt

werden?

- Welche Aufgabenstellung soll dem von den Studierenden zu realisierenden Programmierprojekt zugrunde gelegt werden?
- Wie viele Projekte sollen die Studierenden in einem zweisemestrigem Modul bearbeiten?
- Wie ist parallel der Einstieg
  - in die objektorientierte Programmierung,
  - in die Entwicklungsumgebung *WebVelocity*,
  - in die Grundlagen der Webseitengestaltung,
  - in die Nutzung von elementaren HTML-Elementen mithilfe von *Seaside*-Smalltalk
 zu schaffen?
- Ist es sinnvoll, Techniken des *Test-driven-learning*<sup>23</sup> zu integrieren?

FÜR DIE NAHTLOSE EINBETTUNG des neuen Moduls in den Studiengang sind eine Reihe von didaktischen und organisatorischen Frage zu klären:

- Sollen die Programmieraufgaben einzeln oder in Kleingruppen bearbeitet werden?
- Wie sehen dann geeignete Prüfungsformen für das neue Modul aus?
- Wie kann ein Modul, das im ersten Semester 7 und im zweiten Semester 5 Semesterwochenstunden umfasst, im Stundenplan angemessen berücksichtigt werden?
- Welche Rolle kommt den Dozenten zu?
- Wie viele Dozenten (Projektgruppenbetreuer) werden benötigt?

### 3.2 Technische Aspekte

Grundsätzlich ist bei einem solchen Vorhaben die Frage offen, welche Programmiersprache in Verbindung mit welcher Entwicklungsumgebung verwendet werden soll. Für die Entwicklung von Web-Anwendungen kommt die Frage hinzu, welches Rahmenwerk benutzt werden soll. Für *WESTPOL* sind diese Fragen pragmatisch beantwortet worden, da die für das Vorhaben zur Verfügung stehenden personellen Ressourcen es nur erlauben, auf den im Rahmen des Projekts *NORDPOL* gemachten Erfahrungen und erzielten Ergebnisse aufzubauen. Eine Abkehr von Smalltalk als Einstiegsprache hätte auch weitreichende Folgen auf die Gestaltung anderer Module des Studiengangs Wirtschaftsinformatik.

Gleichwohl bietet das Vorhaben *WESTPOL* auch Raum für die Untersuchung technischer Fragestellungen. Auf die Thematik der Entwicklung einer integrierten Lern- und Entwicklungsumgebung wurde bereits in Abschnitt 2.1 eingegangen.

Eine weitere, hoch interessante Frage lautet: Kann ein E-Learning-Kurs für den Programmierunterricht so strukturiert werden, dass die zu bearbeitende Projektaufgabe austauschbar wird? Hier wäre wohl eine Kombination von technischen und inhaltlichen Problemen zu lösen.

<sup>23</sup> JANZEN, D. und H. SAIEDIAN: *Test-driven learning in early programming courses*. In: *SIGCSE '08: Proceedings of the 39th SIGCSE technical symposium on Computer science education*, S. 532–536, New York, NY, USA, 2008. ACM

Ist z. B. ein „Projekttag“ pro Woche sinnvoll?

Wird es noch „Vorlesungen“ geben? Ist ein Übergang von Blended-learning zu E-Learning möglich und sinnvoll?

Gesucht ist eine Art „didaktisch/inhaltliches Rahmenwerk“.

### 3.3 Didaktische Aspekte

Ein wichtiges Ziel des Projekts *WESTPOL* besteht in der Steigerung der Attraktivität der Programmierausbildung für Anfänger und in Verbindung damit die Erhöhung der Motivation der Studierenden. Hier ergibt sich natürlich das Problem, ob man den Erfolg diesbezüglich überhaupt messen kann. Die Organisation des Studienbetriebs an der NORDAKADEMIE erlaubt nicht die Durchführung eines „Laborversuchs“, bei dem eine Versuchsgruppe mit dem neuen Kurs konfrontiert wird. Beide Module (*Programmierung 1* und *Internet-Grundlagen*) sind Pflicht-Veranstaltungen. Eine Zusammenlegung beider Module zu einem gemeinsamen Modul erfordert eine Änderung der Prüfungsordnung, die dann für alle neuen Studierenden gilt. Eine „Überlappung“ von zwei unterschiedlichen Prüfungsordnungs-Versionen ist nicht möglich.

Nun bestünde die Möglichkeit, verschiedene Jahrgänge miteinander zu vergleichen und die Durchführung von *WESTPOL* mithilfe einer empirischen Studie medien-, bzw. sozialwissenschaftlich zu begleiten. Hierfür sind zunächst geeignete Methoden zu entwickeln.

Derzeit stellen sich den Autoren in Bezug auf die Programmierausbildung vor allem folgende offene Fragen:

- Was für Probleme und Schwierigkeiten gibt es derzeit in der Programmierausbildung? Sind die eigentlich hinreichend exakt identifiziert?
- Ist projektbasiertes Lernen sinnvoll? Wie können wir das überprüfen?
- Welches sind geeignete Unterrichtsformen? Ist *Blended-learning* wirklich sinnvoll?
- Welches sind geeignete Endgeräte, um Programmieren zu lernen (Desktop-PC, Notebook, weitere mobile Endgeräte) ?
- Wie bildet man studentische Arbeitsgruppen? Welche Art von Gruppen (homogene, heterogene) lassen für wen den größeren Lernerfolg erwarten? Wie kann man das überprüfen?
- Welche Probleme haben Studierende beim Umgang mit einer Entwicklungsumgebung eigentlich genau? Kann hier vielleicht die Durchführung einer Benutzbarkeitsstudie helfen?

eventuell in Kooperation mit dem Usability-Labor der HAW Hamburg

## 4 Zusammenfassung

Die Programmierausbildung an der NORDAKADEMIE soll neu gestaltet werden. Die zu berücksichtigenden Aspekte sind zum einen das Erstellen von Webanwendungen. Hierdurch sollen die bisher erstellten Programme ohne (grafische) Oberfläche abgelöst werden. Zum anderen soll versucht werden, eine einfachere Handhabung der Smalltalk Entwicklungsumgebung zu erreichen und somit den Aufwand für die Einarbeitung in eine komplexe IDE zu verringern. Diese Ansätze könnten zudem die Programmiergrundausbildung für die Studenten mit neuer Attraktivität versehen. Das bisher am

sinnvollsten erscheinende Konzept ist, die Programmierumgebung ebenfalls in den Browser zu verlagern. Allerdings sind in dem Zusammenhang auch didaktische Aspekte zu betrachten. Eine Webanwendung zu entwickeln kann schnell recht umfangreich werden und somit könnte während des Kurses nur an einem großen Projekt gearbeitet werden.

Da die Vielzahl der in den vorangegangenen Abschnitten Problemstellungen nicht alle gleichzeitig angegangen werden können, wird das Hauptaugenmerk des Projekts zunächst auf der Entwicklung eines Konzepts für eine Lehrveranstaltung liegen, die die Module *Internet-Grundlagen* und *Programmierung 1* integriert. Die Abfolge weiterer Aktivitäten bleibt noch zu klären.

### Literatur

- [1] BARNES, D. J. und M. KÖLLING: *Objects First with Java*. Pearson Education, 2008.
- [2] BARTELTSEN, J. und J. BRAUER: *Kooperatives Lernen mit moodle in der Programmierausbildung*. Arbeitspapier 2008-6, NORDAKADEMIE Hochschule der Wirtschaft, Oktober 2008.
- [3] BARTELTSEN, J., J. BRAUER und H. REHDER: *Abschlussbericht für das Projekt NORDPOL*. Arbeitspapier 2010-02, NORDAKADEMIE Hochschule der Wirtschaft, Juni 2010.
- [4] BARTELTSEN, J.-I.: *Entwurf einer kooperativen Lehrveranstaltung in moodle für die Grundausbildung in der Programmierung*. Master-Thesis an der FernUniversität Hagen, 2008.
- [5] BRAUER, J.: *Grundkurs Smalltalk - Objektorientierung von Anfang an*. Vieweg+Teubner Verlag, 3. Aufl., 2008.
- [6] DUCASSE, S., L. RENGGLI, D. C. SHAFFER und R. ZACCONE: *Dynamic Web Development with Seaside*. Square Bracket Associates, 2010.
- [7] FELLEISEN, M., R. B. FINDLER, M. FLATT und S. KRISHNAMURTHI: *How to Design Programs*. The MIT Press, September 2003.
- [8] GOLDBERG, A., S. T. ABELL und D. LEIBS: *The LearningWorks development and delivery frameworks*. Commun. ACM, 40(10):78–81, 1997.
- [9] JANZEN, D. und H. SAIEDIAN: *Test-driven learning in early programming courses*. In: *SIGCSE '08: Proceedings of the 39th SIGCSE technical symposium on Computer science education*, S. 532–536, New York, NY, USA, 2008. ACM.
- [10] KLAEREN, H. und M. SPERBER: *Die Macht der Abstraktion*. B. G. Teubner, 1. Aufl., 2007.

- [11] KÖLLING, M., B. QUIG, A. PATTERSON und J. ROSENBERG: *The BlueJ system and its pedagogy*. Journal of Computer Science Education, 13(4), December 2003.
- [12] REGES, S.: *Back to basics in CS1 and CS2*. SIGCSE Bull., 38(1):293–297, 2006.
- [13] REGES, S. und M. STEPP: *Building Java Programs: A Back to Basics Approach*. Addison-Wesley, Mai 2007.
- [14] STEPP, M., J. MILLER und V. KIRST: *A CS 1.5 introduction to web programming*. In: *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, S. 121–125, New York, NY, USA, 2009. ACM.
- [15] WRIGHT, A.: *Ready for a Web OS?*. Commun. ACM, 52(12):16–17, 2009.