

Informatikpraxis mit Open Source im 6. Jahrgang

Ludger Humbert

begonnen: 18. Aug. 2004

Stand: 3. November 2004

Vortrag: 11. Nov. 2004

Um zielgerichtet, erfolgreich und nachhaltig mit Informatiksystemen zu arbeiten, müssen allgemein gültige, »mentale« Modelle dieser Systeme bei den Schülerinnen ausgebildet werden. Diese werden transparent entwickelt, wenn der Unterrichtseinsatz durch eine informatikdidaktisch gestaltete, fachliche Sicht auf die Systeme begleitet wird.

Revision : 1.2

Dieser Erkenntnis folgend wurde die Objektorientierung als fachliche und begriffliche Basis für den Anfangsunterricht im 6. Jahrgang für das Pflichtfach Informatik auf einer ausgewiesenen fachdidaktischen Basis gewählt. Allein, was nützt das Konzept, wenn es nicht auf einer informatisch gestalteten Handlungsebene seine gestaltende Kraft erweisen kann?

Hier kommt das Open-Source-Modell ins Spiel. Unter Nutzung der offenen und dokumentierten Schnittstellen von OpenOffice.org wurde von zwei Studierenden des Lehramts Informatik der Universität Dortmund (Christiane Borchel und Martin Reinertz) 2004 eine didaktisch gestaltete Schnittstelle für Schülerinnen implementiert, mit der eine Realisierung des Fachkonzepts zur Konstruktion von Texten auf einer objektorientierten Basis zur Verfügung gestellt wird. Eine erste unterrichtliche Erprobung dieser Umsetzung findet zur Zeit an der Willy-Brandt-Gesamtschule, Bergkamen statt. Die fachdidaktischen Hintergründe und ein unterrichtliches Szenario werden vorgestellt, um Möglichkeiten einer qualifizierten Auseinandersetzung mit diesem Konzept zu eröffnen. Das Ziel besteht darin, Kolleginnen zu ermuntern, diese Art der Umsetzung selbst im Unterricht zu erproben.

Ideen für Weiterungen für andere Open-Source-Produkte (GIMP – Bildbearbeitung, DIA – vektororientiert Zeichnen, Scribus – Desktop Publishing (DTP)) werden angegeben. Im Ausblick wird angedeutet, wie mit dem vorgestellten Prototypen (Ponto) Unterricht in der Sekundarstufe II gestaltet werden kann (Ideen für eine Unterrichtsreihe).

Inhaltsverzeichnis

1 Basiskonzepte informatischer Bildung	2
2 Informatik ist Ausgangspunkt für die Arbeit mit Informatiksystemen	3
2.1 Ponto kommt ins Spiel – der Anwendungsfall	5
2.2 Algorithmen – erste Vorbereitung	7
3 Seiteneffekte	8
Tabellenverzeichnis	8
Abbildungsverzeichnis	8
Verzeichnis der Algorithmen	8
Literatur	9
Abkürzungsverzeichnis	10

1 Basiskonzepte informatischer Bildung

Die Anzahl der informatischen Basiskonzepte ist – im Gegensatz zu einer häufig geäußerten Vermutung –

1. überschaubar (ca. 10)
2. stabil

Es ist von Seiten der Softwaretechnik notwendig, Sichten auf ein System zu charakterisieren und zu entscheiden, mit welchem konkreten Konzept die jeweilige Sicht modelliert werden kann.

Dabei haben sich die folgenden Sichten als nützlich für die Konstruktion von Informatiksystemen erwiesen: Funktionen, Daten, Dynamik und Benutzungsoberfläche. Damit werden deutlich statische und die dynamische Sichtweisen ausgewiesen und zudem die Interaktion von Benutzerinnen mit Informatiksystemen, die zu einer erheblichen Komplexitätssteigerung bei der Modellierung führen, berücksichtigt.

Basiskonzepte können auf verschiedene Art beschrieben werden. Dies reicht von vollständig formalisierten bis zu informalen Beschreibungen, die in den Ausprägungen textuell bis graphisch ihre jeweilige Darstellung finden.

Für alle Basiskonzepte existieren Vorschläge zur Darstellung in Form von mehr oder weniger formalisierten (grafischen und textuellen) Notationen. Dies wird in Tabelle 2 an Beispielen verdeutlicht.

Darüber hinaus existieren für einige der Basiskonzepte Überlegungen zur Umsetzung im Informatikunterricht (typischerweise für die Sekundarstufe II, vgl. [Humbert 2003]). Seit

Tabelle 1: Basiskonzepte nach [Balzert 1996, S. 98]

Basiskonzept	Sicht
Funktionale Hierarchie Datenfluss	Funktional
Datenstrukturen Entitäten & Beziehungen	Datenorientiert
Klassenstrukturen	Objektorientiert
Kontrollstrukturen	Algorithmisch
wenn-dann-Strukturen	Regelbasiert
Endlicher Automat Nebenläufige Strukturen	Zustandsorientiert
Interaktionsstrukturen	Szenariobasiert

Ende 2003 finden sich erste Veröffentlichungen zu Standards für den Informatikunterricht. Ausgehend von der Überlegung, dass für **P**rogramme for **I**nternational **S**tudent **A**ssessment (PISA) nicht gefordert wird, dass international zu vergleichende inhaltliche Fragestellungen durch konkrete curriculare Inhalte repräsentiert werden müssen, haben sich Fachdidaktiker darauf verständigt, Überlegungen zu Informatikstandards zu diskutieren (vgl. [Puhlmann 2003, Humbert und Puhlmann 2004a, Humbert und Puhlmann 2004b]). Es steht zu hoffen, dass fachdidaktisch reflektierte Überlegungen Eingang in zukünftige internationale Studien finden.

2 Informatik ist Ausgangspunkt für die Arbeit mit Informatiksystemen

Von der Abnahmeseite werden Anforderungen an die von der Schule zu »erbringenden« Basiskompetenzen formuliert. Bereits 1976 wurde von der **G**esellschaft für **I**nformatik e. V. (GI) in den Empfehlungen zum Unterricht im Schulfach Informatik [Brauer u. a. 1976] gefordert, dass Informatik als allgemein bildendes Schulfach verpflichtend für alle Schülerinnen unterrichtet wird.

Dieser Forderung haben inzwischen einige Bundesländer entsprochen: aktuelles Beispiel ist Bayern, das Informatik im Gymnasium ab Klasse 6 soeben als verbindliches Pflichtfach eingeführt hat. In Nordrhein-Westfalen steht dieser Schritt aus, ist aber trotz der Diskussion um **G**ymnasium in **8** Jahren (G8) durchaus nicht als unrealistisch einzuschätzen.

Tabelle 2: Beispiele von Notationsmöglichkeiten für Basiskonzepte

Basiskonzept	grafische Notation	textuelle Notation
Funktionale Hierarchie	Funktionsbaum	
Datenfluß	Datenfluss-Diagramm	
Datenstrukturen und Kontrollstrukturen	Syntax-Diagramm Jackson-Diagramm	Data Dictionary (DD) Jackson Structured Programming (JSP)
Entitäten & Beziehungen	Entity-Relationship (ER)-Diagramm	Structured Query Language (SQL)-Abfrage
Klassenstrukturen	Unified Modelling Language (UML)-Klassendiagramm	Spezifikation
Kontrollstruktur	Struktogramm Programmablaufplan (PAP)	Pseudocode
wenn-dann-Strukturen	Entscheidungsbaum	Regeln
Endlicher Automat	Zustandsgraph (Zustandsautomat)	Zustandsdiagramm Zustandstabelle
Nebenläufige Strukturen	Petrinetz	(textuell)
Interaktionsstrukturen	UML- Interaktionsdiagramm	

nach [Balzert 1996, S. 103]

Exkurs: Warum ist ein Pflichtfach Informatik unabdingbar?

Wie in den Darstellungen der Basiskonzepte (vgl. Tabelle 1) und ihrer Darstellungsformen (vgl. Tabelle 2) deutlich wird, ist die Anzahl der fachlichen Beschreibungsmöglichkeiten der Basiskonzepte überschaubar. Dies darf nicht darüber hinwegtäuschen, dass diese Basiskonzepte von den Schülerinnen nicht ohne fundierten Informatikunterricht »nebenbei« erlernt werden können.

Diese Überlegungen führen zu den Forderungen:

1. Informatik muss als Pflichtfach in den Jahrgängen 5–10 aller Schulen verankert werden.
2. Informatik kann nicht integriert und damit nebenbei unterrichtet werden.

3. Informatik darf nicht fachfremd unterrichtet werden.

Dem steht die Praxis entgegen: fast 100% des Informatikunterrichts in Haupt- und Realschulen wird fachfremd erteilt (Schulstatistik Nordrhein-Westfalen (NW)).

Es ist ein seit 30 Jahren bekanntes Versäumnis [nicht nur] des nordrhein-westfälischen Kultusministeriums, die Ausbildungskapazitäten für die Informatiklehrerbildung dem tatsächlichen Bedarf anzupassen: bis heute kann Informatik für die Lehrämter **Grundschule (GS)**, **Hauptschule (HS)**, **Realschule (RS)** und **Gesamtschule (GE)** (bis Jahrgang 10) – vormals Lehramt für die Sekundarstufe I an **keiner** Universität des Landes NW studiert werden – ich sehe darin einen Skandal.

Die Fachdidaktik hat seit vielen Jahren Vorschläge für den Informatikunterricht in den Jahrgängen 5–10 vorbereitet. Doch erst mit neueren Überlegungen scheint sich eine gewisse Konsolidierung abzuzeichnen, die darin besteht, dass durchaus mit sogenannten Standardanwendungen gearbeitet wird. Lange Zeit wurden diese Informatiksysteme von der Fachdidaktik sehr kritisch gesehen, da es kaum möglich schien, eine Verbindung zwischen den Fachkonzepten der Informatik und dem übermächtigen Funktionsumfang moderner Standardanwendungen zu finden, die dem kognitiven Niveau der Schülerinnen angemessen ist. Durch einen »radikalen« objektorientierten Zugang scheint dieser Konflikt auflösbar (vgl. [Hubwieser und Broy 1997]). Inzwischen wurde Informatik in den Pflichtbereich der bayerischen Gymnasien aufgenommen (trotz G8) und das erste Schulbuch [Frey u. a. 2004] für den Einsatz im 6. Jahrgang genehmigt.

Schülerinnen erfahren, wie produktunabhängig eine fachlich korrekte Sprache eingeführt und durchgängig (für alle Anwendungen) benutzt wird. Diese Sprache orientiert sich an der [üblichen] objektorientierten Notation. D. h. mit Hilfe der syntaktischen Struktur `objekt.Aktion()` bzw. `objekt.attribut` ist es möglich, Objekte zu verändern und auf ihre Attributwerte zuzugreifen (vgl. das soeben genehmigte bayerische Schulbuch *Informatik 1* für den 6. Jahrgang: [Frey u. a. 2004]).

Ein Problem mit diesem konzeptionell bedeutsamen und durchdachten Ansatz besteht in der Abstinenz gegenüber der mit Hilfe der eingeführten (formalen) Begriffe möglichen konstruktiven Nutzung, d. h. die Schülerin ist gezwungen, ihre Lösung der Lehrerin vorzulegen, um herauszufinden, ob die formale Darstellung der Aktionen mit dem Informatiksystem der geforderten Zielsetzung genügen (vgl. [Borchel u. a. 2005]).

2.1 Ponto kommt ins Spiel – der Anwendungsfall

Nachdem den Schülerinnen bekannt ist, dass sich jedes Dokument als Objekt der Klasse `Dokument` erstellen lässt, kann mittels der in Algorithmus 1 zwei Zeilen im Python-Interpreter OpenOffice dazu veranlasst werden, ein neues Dokument zu erstellen.

6a

```

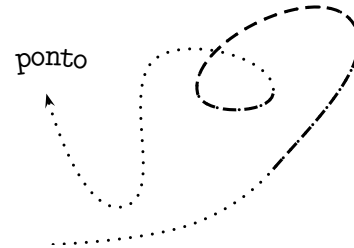
(EinladungGeburtstag.py* 6a)≡
from ponto import Dokument
einladungZumGeburtstag=Dokument()

```

6b▷

Algorithm 1: Instanziierung Dokument

Ist die Anwendung OpenOffice.org nicht aktiv, so wird von *ponto* aus der Programmstart vorgenommen. Bei der Initialisierung ermittelt *ponto* die Plattform und sorgt dafür, dass die auf der jeweiligen Plattform (implementiert sind zur Zeit Linux und Mac OS X) üblichen Konventionen eingehalten werden.



6b

```

(EinladungGeburtstag.py* 6a)+≡
eroeffnung=einladungZumGeburtstag.erzeugeAbsatz("Meine liebe Freundin,")
hauptAbsatz=einladungZumGeburtstag.erzeugeAbsatz(
    """im nächsten Monat werde ich 12 Jahre alt. An meinem Geburtstag
    möchte ich gern' mit dir zusammen feiern."""
)
schluss=EinladungZumGeburtstag.erzeugeAbsatz(
    """Ich wünsche mir, dass du bei dieser Feier dabei sein kannst und
    würde mich sehr freuen, wenn du dabei sein könntest. """
)

```

<6a

Algorithm 2: Dokument wird mit Absätzen versehen

Werden die Zeilen in den (von OpenOffice.org mitgelieferten) Pythoninterpreter eingegeben, so ist das Ergebnis direkt in dem geöffneten neuen Dokument sichtbar. So wird unmittelbar nach der Eingabe

```
einladungZumGeburtstag=Dokument()↔
```

von *ponto* ein neues [Writer-]Dokument in OpenOffice.org geöffnet.

Damit wird das motivierende direkte Erfolgserlebnis direkt nach der Eingabe vermittelt.

Besonders bedeutsam ist die Implementierung der Klasse *Cursor*. In der statischen (Klassen-)Beschreibung taucht *Cursor* nicht auf, da einem fertigen Dokument der Cursor nicht »angesehen« werden kann. Für die dynamische Erstellung und Bearbeitung von Dokumenten hingegen ist die Möglichkeit, auf die Textmarke (== Cursor) Einfluss nehmen zu können, jedoch unabdingbar.

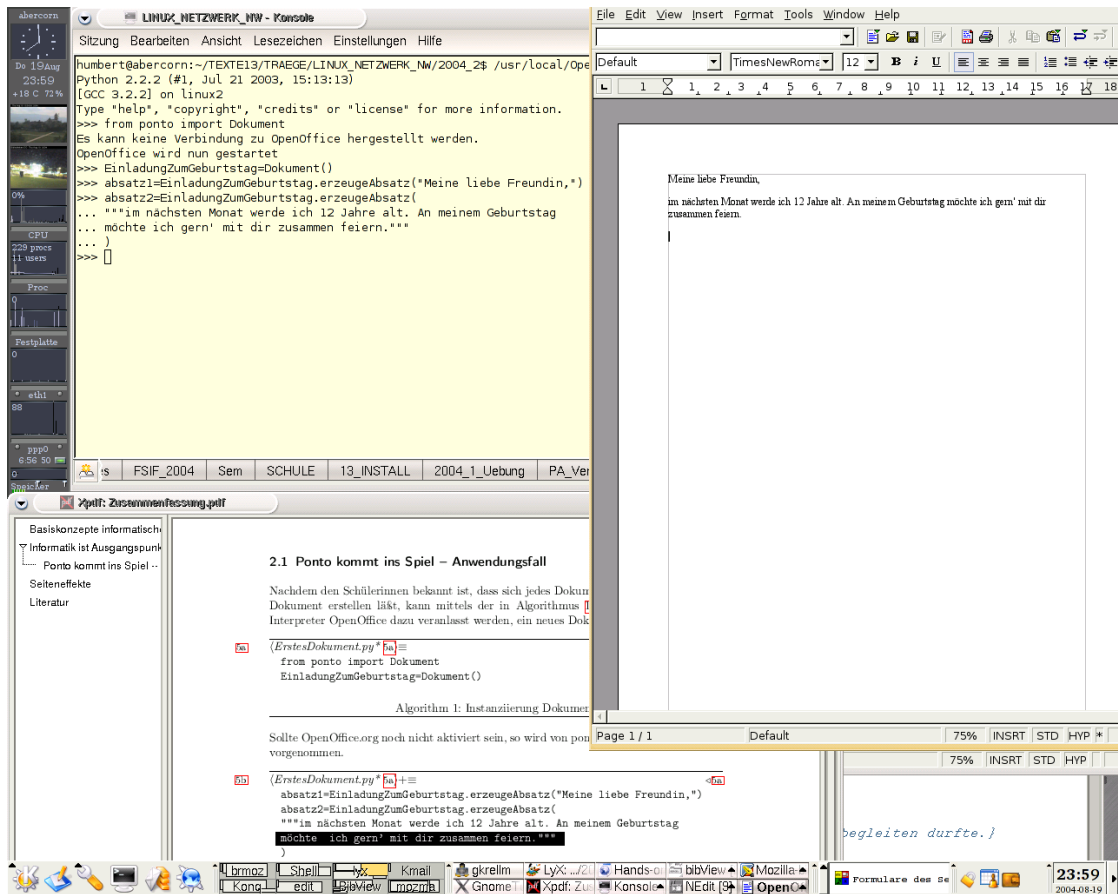


Abbildung 1: Die ersten beiden Absätze werden angezeigt (Linux)

2.2 Algorithmen – erste Vorbereitung

In einem geschriebenen Text navigieren, Attributwerte einzelner Zeichen und Absätze zu ändern, ist allen Nutzerinnen von sogenannten Textverarbeitungssystemen bekannt. Dies geschieht heute üblicherweise durch »Mausschieberei«, wie einige abschätzig bemerken. Allerdings ist hier nicht die Stelle, um über WYSIWYG zu urteilen. Soll in der beschriebenen Weise gearbeitet werden (dies ist durchaus diskussionswürdig!), ergeben sich mit Hilfe von ponto einige Ansätze zur Motivation erster algorithmischer Elemente zur Ablaufsteuerung: in einem Wort sollen die Zeichen abwechselnde Farben erhalten (z. B. in einer Geburtstagseinladung). Ein entsprechender Ausschnitt findet sich in Algorithmus 3.

```

i=0
farbe=Schwarz
while i<19:
    if farbe==Schwarz:
        einZeichen.setzeSchriftfarbe(Gelb)
        farbe=Gelb
    else:
        einZeichen.setzeSchriftfarbe(Schwarz)
        farbe=Schwarz
    i=i+1
    cursor.vor()

```

Algorithm 3: Änderung der Attributwerte *farbe* für eine Reihe von Objekten der Klasse *Zeichen*

3 Seiteneffekte

- Klärung der Lizenzfrage[n]
- Weitergabe der Produktionsmittel?
- Andere Backends (PyTeX)
- Überlegungen für den Einsatz in der Sekundarstufe II

Tabellenverzeichnis

1	Basiskonzepte nach [Balzert 1996, S. 98]	3
2	Beispiele von Notationsmöglichkeiten für Basiskonzepte	4

Abbildungsverzeichnis

1	Die ersten beiden Absätze werden angezeigt (Linux)	7
---	--	---

Verzeichnis der Algorithmen

1	Instanziierung Dokument	6
---	-----------------------------------	---

2	Dokument wird mit Absätzen versehen	6
3	Änderung der Attributwerte <i>farbe</i> für eine Reihe von Objekten der Klasse <i>Zeichen</i>	8

Literatur

- [Balzert 1996] BALZERT, Helmut: *Lehrbuch der Software-Technik: Software-Entwicklung*. Heidelberg : Spektrum Akademischer Verlag, 1996 (Lehrbücher der Informatik)
- [Borchel u. a. 2005] BORCHEL, Christiane ; HUMBERT, Ludger ; REINERTZ, Martin: Design of an Informatics System to Bridge the Gap Between Using and Understanding in Informatics. In: MITTERMEIR, Roland (Hrsg.): *Informatics in Secondary Schools. Evolution & Perspectives Klagenfurt, 30th March to 1st April 2005*. Berlin : Springer Verlag, 2005 (Lecture Notes in Computer Science). – eingereichter Beitrag – Review noch nicht abgeschlossen – Stand Oktober 2004
- [Brauer u. a. 1976] BRAUER, Wilfried ; CLAUS, Volker ; DEUSSEN, Peter ; JÜRGEN EICKEL (FEDERFÜHREND) ; HAACKE, Wolfhart ; HOSSEUS, Winfried ; KOSTER, Cornelis H. A. ; OLLESKY, Dieter ; WEINHART, Karl ; GESELLSCHAFT FÜR INFORMATIK E. V.: Zielsetzungen und Inhalte des Informatikunterrichts. In: *ZDM* 8 (1976), Nr. 1, S. 35–43. – ZDM – Zentralblatt für Didaktik der Mathematik
- [Frey u. a. 2004] FREY, Elke ; HUBWIESER, Peter ; WINHARD, Ferdinand: *Informatik 1. Objekte, Strukturen, Algorithmen (Schülerbuch)*. Stuttgart : Ernst Klett Verlag, Juli 2004. – ISBN 3–12–731468–X
- [Hubwieser und Broy 1997] HUBWIESER, Peter ; BROY, Manfred: Grundlegende Konzepte von Informations- und Kommunikationssystemen für den Informatikunterricht. In: HOPPE, Heinz U. (Hrsg.) ; LUTHER, Wolfram (Hrsg.): *Informatik und Lernen in der Informationsgesellschaft*. Berlin, Heidelberg : Springer, September 1997 (Informatik aktuell), S. 40–50
- [Humbert 2003] HUMBERT, Ludger: *Zur wissenschaftlichen Fundierung der Schulinformatik*. Witten : pad-Verlag, März 2003. – zugl. Dissertation an der Universität Siegen <http://www.ham.nw.schule.de/pub/bscw.cgi/d38820/> – geprüft: 13. August 2003. – ISBN 3–88515–214–2
- [Humbert und Puhlmann 2004a] HUMBERT, Ludger ; PUHLMANN, Hermann: Essential Ingredients of Literacy in Informatics. In: MAGENHEIM, Johannes (Hrsg.) ; SCHUBERT, Sigrid (Hrsg.): *Informatics and Student Assessment. Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics* Bd. 1. Bonn : Köllen Druck+Verlag GmbH, September 2004a, S. 65–76

[Humbert und Puhlmann 2004b] HUMBERT, Ludger ; PUHLMANN, Hermann: Informatische Bildung und PISA-Standards – zur Umsetzung für die informatische Bildung. In: MICHEUZ, Peter (Hrsg.): *Standards im IKT/Informatik-Unterricht? Projekt CDA-Sonderausgabe*. Perg : CDA-Verlag, 2004b. – http://www.ham.nw.schule.de/pub/bscw.cgi/d65451/i_standards16.pdf – geprüft: 3. März 2004, S. 21–24

[Puhlmann 2003] PUHLMANN, Hermann: Informatische Literalität nach dem PISA-Muster. In: HUBWIESER, Peter (Hrsg.): *Informatik und Schule – Informatische Fachkonzepte im Unterricht INFOS 2003 – 10. GI-Fachtagung 17.–19. September 2003, München*. Bonn : Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, September 2003 (GI-Edition – Lecture Notes in Informatics – Proceedings P 32). – ISBN 3-88579-361-X, S. 145–154

Abkürzungsverzeichnis

DD	Data Dictionary	NW	Nordrhein-Westfalen
DTP	Desktop Publishing	PAP	Programmablaufplan
ER	Entity-Relationship	PISA	Programme for International Student Assessment
G8	Gymnasium in 8 Jahren	RS	Realschule
GE	Gesamtschule	SQL	Structured Query Language
GI	Gesellschaft für Informatik e. V.	UML	Unified Modelling Language
GS	Grundschule		
HS	Hauptschule		
JSP	Jackson Structured Programming		