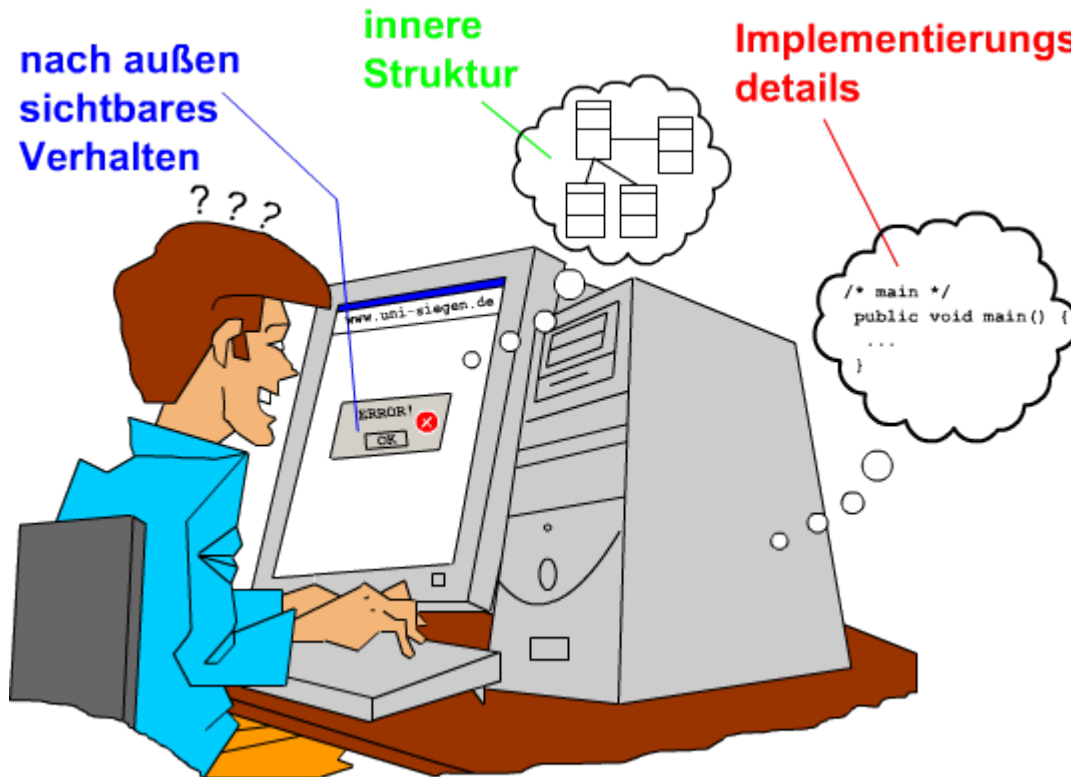


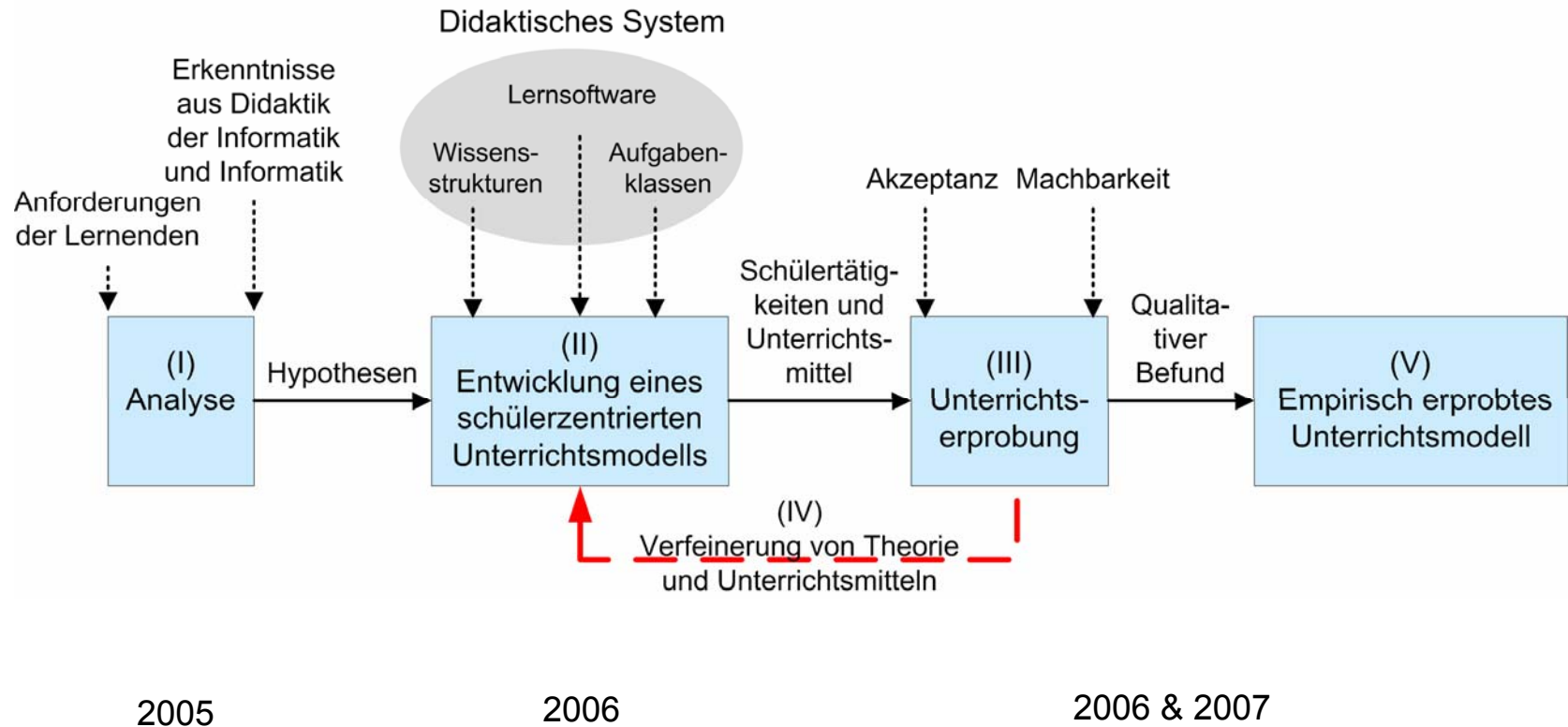
Von vernetzten fundamentalen Ideen zum Verstehen von  
Informatiksystemen – Eine Unterrichtserprobung in der Sekundarstufe II

Peer Stechert  
Didaktik der Informatik und E-Learning  
Universität Siegen

1. Motivation
2. Forschungsmethodik zur Entwicklung des Unterrichtsmodells für das Verstehen von Informatiksystemen
3. Erprobung des Unterrichtsmodells
  - Beispielaufgabe: Informatikexperiment
  - Lernsoftware „Pattern Park“
4. Auswertung der Unterrichtserprobung
5. Zusammenfassung und Ausblick



- vernetzte fundamentale Ideen der Informatik in Informatiksystemen
- Strukturmodelle
  - Von-Neumann-Rechner
  - Schichtenmodelle
  - Entwurfsmuster
- Entwurfsmuster als Wissensrepräsentationen für vernetzte fundamentale Ideen der Informatik [Stechert, 2006]



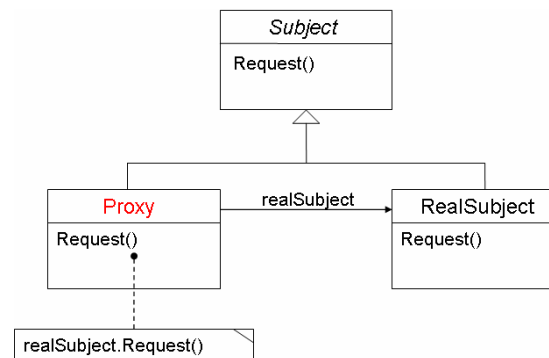
- Erprobung im Herbst 2006 im Rahmen eines Grundkurses Informatik in der Jahrgangsstufe 12 eines Siegener Gymnasiums
- vier Schülerinnen und 19 Schüler
- 12 Unterrichtsstunden zu drei Stunden pro Woche
- Vorkenntnisse: **objektorientiertes Modellieren mit Klassendiagrammen, Vererbung** und **Liste**
- Schülerlösungen wurden eingesammelt
- Evaluation: Lernerfolgskontrolle, schriftliche Akzeptanzbefragung und ein Interview mit der betreuenden Informatiklehrperson

- Startpunkt: **Arztpraxis mit einer Warteschlange** – von den Schülern mittels Klassendiagramm modelliert und mit Objekt-Pascal (Delphi 6) programmiert
- Iteration mit dem Entwurfsmuster **Iterator**
- Zugriffskontrolle mit dem Entwurfsmuster **Proxy**
- Ziel: **Zusammenhang** zwischen **nach außen sichtbarem Verhalten** und Modellen der **inneren Struktur** von Informatiksystemen
- **Informatikexperimente** zur Erkundung von Informatiksystemen

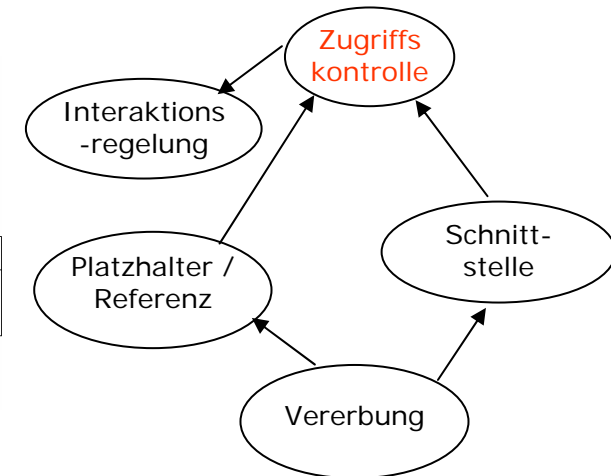
S<sub>A</sub>: nach außen sichtbares Verhalten



S<sub>B</sub>: innere Struktur



vernetzte fundamentale Ideen



1. **Beobachtung** unerwarteten Verhaltens eines Informatiksystems
  2. Formulierung einer klaren Fragestellung durch Lehrer und Schüler (**Hypothesen** zur Erklärung der Beobachtung)
  3. **Konzeption** eines Experiments zur Überprüfung der Hypothese durch Lehrer und / oder Schüler
  4. Durchführung des Experiments und **Dokumentation** der Ergebnisse
  5. Bestätigung oder Widerlegung der Hypothesen durch **Interpretation** der Ergebnisse und des Versuchsverlaufs
  6. **Diskussion** um Kontrollversuche, Varianten, technische und soziale Folgen des Einsatzes von Informatiksystemen [in Anlehnung an Meyer, 2006]
- „Das selbsttätige oder auch vom Lehrer locker geleitete Vorbereiten, Durchführen und Auswerten eines Experiments ist auf jeden Fall **eine der besten denkbaren Möglichkeiten, die methodische Handlungskompetenz der Schüler zu entwickeln**, und zwar sowohl in kognitiver als auch in sozialer und motorisch-manueller Hinsicht“ [Meyer, 2006, S. 318]

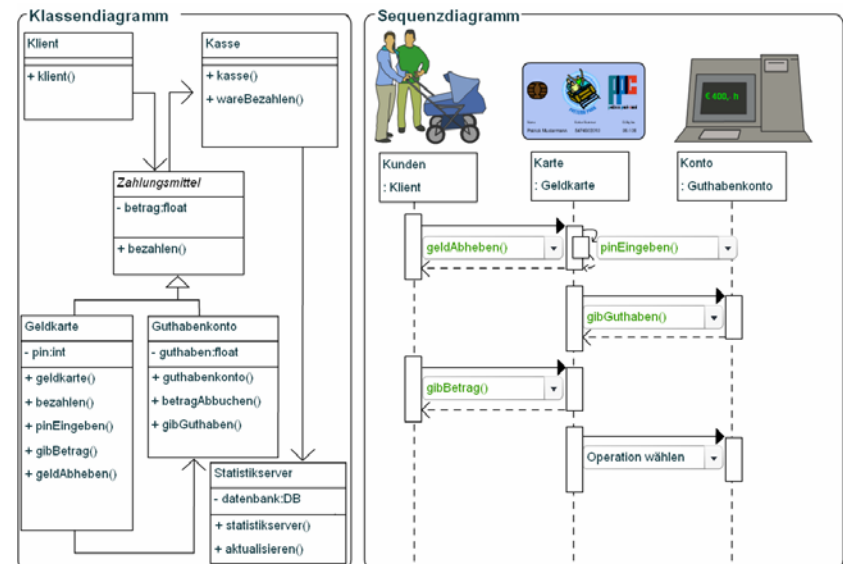
- Arztpraxis
  - Erkundung des nach außen sichtbaren Verhaltens: **Beschreibung des Wartezimmers**
  - Hypothesen über innere Struktur und vernetzte fundamentale Ideen: **Schlange** und **Iteratormuster**
- Operationalisierung der Lernziele mit **Blooms überarbeiteter Taxonomie** (kognitiver Prozess; Wissensdimension) [Anderson & Krathwohl, 2001]

### Schritt 5

- Aufgabe: Nenne Sonderfälle der Schlange  
 „Was geschieht bei einem Notfall?“
- kognitiver Prozess: Evaluieren
- Wissensdimension: konzeptuell



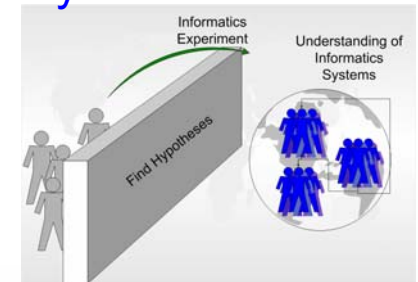
- systematische Erkundung des Systemverhaltens
- Zugriffskontrolle mittels Proxy: **Administrator, Benutzer und Gast** als Rollen mit Rechten
- überraschend war für die Schüler der auf Vererbung basierende Ansatz zur Überprüfung der Zugriffsrechte
- dynamische Aspekte: einfache **Sequenzdiagramme** mit Elementen der Lernsoftware „Pattern Park“\*
  - **Lebensweltbezug**
  - lernförderlicher **Sichtenwechsel**



\* „Pattern Park“ finden Sie unter URL

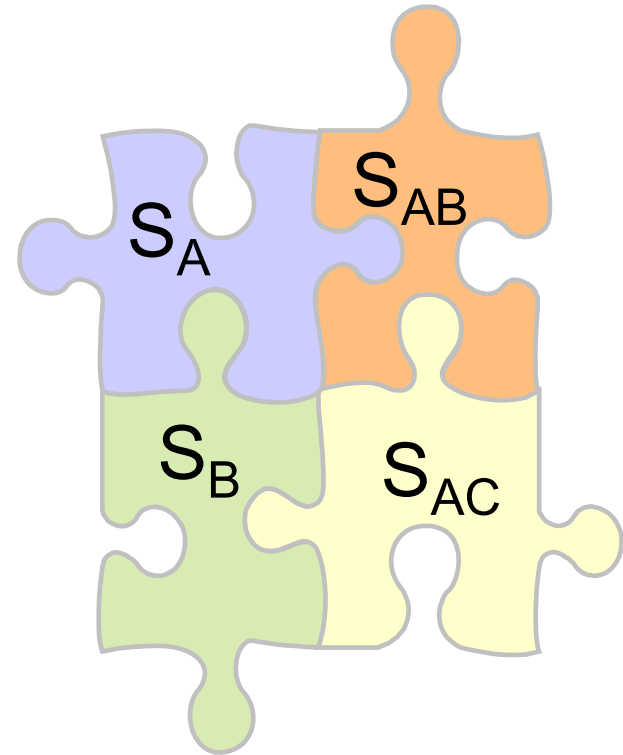
<http://www.die.informatik.uni-siegen.de/projekte/pgpatternpark>

- **Machbarkeit** des Unterrichtsmodells und **Akzeptanz** durch die Schüler
- Schwierigkeit:
  - Schülern fällt es schwer, Hypothesen für das Informatikexperiment zu generieren → **Verfeinerung der Aufgabenstrukturierung** anhand von **Basiskompetenzbeschreibungen**
  - *Aber:* während der Erprobung oft typische kognitive Barrieren der Objektorientierung → exemplarische Analyse mittels „**Laut-Denken**“  
[Stupperich / Warkentin, 2007]
- Auswertung der Akzeptanzbefragung der Lernenden:
  - 16 Schüler gaben an, **nichts oder nicht viel zur Delphi-Programmierung** dazugelernt zu haben
  - 19 Schüler stimmen der Aussage etwas bzw. voll zu, dass die Inhalte des Unterrichtsprojekts **für den Umgang mit Informatiksystemen sehr nützlich** sind



### Basiskompetenzen\*

- $S_A$  nach außen sichtbares Verhalten:  
Beobachtungsaufgaben und Experimente mit vorgegebener Hypothese
- $S_{AB}$  nach außen sichtbares Verhalten und innere Struktur:  
Informatikexperimente
- $S_{AC}$  nach außen sichtbares Verhalten und Implementierungsdetails:  
systematisches Testen
- $S_B$  innere Struktur:  
Analyse der Komponenten (statisch / dynamisch)



\* Kompetenzbegriff nach [Weinert, 2001]

Die Schüler sind in der Lage:

- *Motivation*
  - die Bereitschaft und Fähigkeit zu entwickeln, von Black-Box-Vorstellungen (Computergläubigkeit) zu rationalen Strukturmodellen zu wechseln ( $S_{AB}$ )
  - **Beispiel:** Hinterfragen des Mechanismus zur Zugriffskontrolle
- *Transfer in Verstehen von Fachkonzepten*
  - Erkenntnisse von **einem Strukturmodell zu einem anderen zu übertragen** ( $S_B$ )
  - **Beispiel:** Übertragen von Erkenntnissen aus der Klassenstruktur des Proxymusters auf Sequenzdiagramme

- Ergebnisse der Unterrichtsintervention
  - empirisch erprobte Aufgaben
  - Ausgestaltung und Operationalisierung der Lernziele zu den Basiskompetenzen
  - Schwerpunkt auf Beobachtungsaufgaben ( $S_A$ )
- abgeschlossene Diplomarbeit:  
„Architekturmuster als Beitrag zum Informatiksystemverständnis“ [Ufer, 2007]
- weitere Forschungsarbeit zu typischen Repräsentanten von Informatiksystemen und Strukturmodellen
- nächste Unterrichtsintervention im November 2007: Zustandsautomaten und Zugriffskontrolle mit Entwurfsmustern
- Abschlussdiskussion im Sommer 2008

- [Anderson & Krathwohl, 2001] Anderson L.; Krathwohl D. (Eds.): *A Taxonomy for Learning, Teaching and Assessing – A Revision of Bloom’s Taxonomy of Educational Objectives*. Addison Wesley Longman. 2001
- [ACM, 2003] ACM: *A Model Curriculum for K-12 Computer Science*. 2nd Edition. Final Report of the ACM K-12 Task Force Curriculum Committee. New York. 2003. (<http://csta.acm.org/Curriculum/sub/K-12ModelCurr2ndEd.pdf>, 06.03.2007)
- [Meyer, 2006] Meyer, H.: *Unterrichts-Methoden. II: Praxisband*. Cornelsen Scriptor. 13. Auflage, 2006.
- [Stechert, 2006] Stechert, P.: *Informatics System Comprehension - A learner-centred cognitive approach to networked thinking*. In: *Education and Information Technologies* 11 (2006) 3, Springer Netherlands, Oktober 2006.
- [Stupperich / Warkentin, 2007] Stupperich, P.; Warkentin, S.: „Laut-Denken“ für die Didaktik der Informatik am Beispiel von Vorgehensweisen für das Verstehen von Informatiksystemen. Seminararbeit, FG Didaktik der Informatik und E-Learning, Universität Siegen, März 2007.
- [Ufer, 2007] Ufer, J.: *Architekturmuster als Beitrag zum Informatiksystemverständnis*. Diplomarbeit. Universität Siegen. Mai 2007.
- [UNESCO, 2002] *Information and Communication Technology in Secondary Education – A Curriculum for Schools*. Edited by T. van Weert. Paris: UNESCO, [WWW document, 01.15.2007] URL: <http://unesdoc.unesco.org/images/0012/001295/129538e.pdf>
- [Weinert, 2001] Weinert, F. E.: *Vergleichende Leistungsmessung in Schulen - eine umstrittene Selbstverständlichkeit*. In: Weinert, F. E. (Hrsg.): *Leistungsmessungen in Schulen*. Weinheim und Basel: Beltz Verlag, 2001, S. 17–31.