

Methodik der OOSE für Fachinformatiker nach dem Lernfeldansatz unter Einbeziehung der Lehrerfortbildung

Dietmar Johlen

Studienseminar für berufliche Schulen in Kassel
Holländische Straße 141
34127 Kassel
djohlen@web.de

Abstract: Der Rahmenlehrplan für Fachinformatiker stellt die Handlungsorientierung im Berufsschulunterricht in den Vordergrund. Diese Schwerpunktsetzung wird durch den Lernfeldansatz curricular abgesichert. Die Arbeit stellt die didaktische Transformation des Handlungsfeldes der objektorientierten Softwareentwicklung (OOSE) aus der Praxis in das Lernfeld „Entwickeln und Bereitstellen von Anwendungssystemen“ für Fachinformatiker an der Berufsschule vor. Es wird ein auf die Bedürfnisse der Berufsschule angepasstes Vorgehensmodell entwickelt, das die klassische fachsystematische Strukturierung ersetzt. Erste Ergebnisse einer Pilotunterrichtsreihe werden vorgestellt. Die Arbeit verdeutlicht die besondere Bedeutung der Metasprache UML (Unified Modeling Language) für die Übersichtlichkeit der OOSE und diskutiert den Einsatz von Entwicklungswerkzeugen. Da dieser Ansatz eine grundlegend neue Didaktik und Methodik verlangt, beinhaltet die vorgestellte Konzeption auch die Lehrerfortbildung. Konkret werden die Kursmodule zur OOSE der IT Akademie Hessen vorgestellt.

1 Einleitung

Über den Erfolg einer Softwareentwicklung am Markt entscheidet heute wesentlich die Einhaltung eines definierten Softwareentwicklungsprozesses, der sich über den gesamten Lebenszyklus eines Softwareproduktes, d.h. vom ersten Kundengespräch bis hin zur Demissionierung, erstreckt. Stärken im Softwareentwicklungsprozess sind direkt in Wettbewerbsvorteile, wie geringere Kosten, kürzere Entwicklungszeiten und größere Kundenzufriedenheit, umsetzbar. Deshalb liegt die Vertrautheit der Auszubildenden und späteren Mitarbeiter mit dem Softwareentwicklungsprozess im besonderen Interesse der Unternehmensleitung. Dieser Prozess ist mehr als lediglich die Befolgung einiger Verhaltensregeln, der Prozess verlangt es, „gelebt“ zu werden. Die Nachfrage nach guter Überschaubarkeit, hoher Wiederverwendbarkeit und leichter Wartbarkeit von Software führte zur Durchsetzung der objektorientierten Programmiertechnik (OOP).

Der objektorientierte Softwareentwicklungsprozess stellt daher ein wichtiges Handlungsfeld dar, das im Rahmen des Lernfeldansatzes im Lernfeld „Entwickeln und Bereitstellen

von Anwendungssystemen“ für Fachinformatiker auf die schulischen Gegebenheiten der Berufsschule umgesetzt wird.

Kurse zur OOP gehen vorwiegend nach einer Methodik vor, die auf die strukturierte Programmiertechnik ausgelegt war. Einer fachsystematischen Strukturierung folgend, werden verfügbare Datentypen vorgestellt, gefolgt von Kontrollstrukturen, Schleifen und weiteren Sprachdetails. Der Schwerpunkt wird nach wie vor auf den Algorithmus gelegt. Die Konzepte Datenkapselung, Objektinteraktion, Polymorphismus und Dynamische Bindung werden außer Acht gelassen. Ebenso wird die Metasprache UML (Unified Modeling Language) nicht zur Verdeutlichung der OOP als Unterstützung eingesetzt. So entsteht bei den Schülern beim Übergang zur OOP der Eindruck, dass einfache Probleme kompliziert (weil objektorientiert) gelöst werden, denn Beispiele und Konzeptionen sind auf die strukturierte Programmiertechnik ausgelegt. Es besteht nun die Gefahr, dass auch die Metasprache UML in dieser Form Eingang in den Unterricht findet. Einem solchen Ansatz folgend, könnten losgelöst von den Anforderungen des Entwicklungsprozesses Klassendiagramme mit Assoziationen und Vererbung eingeführt werden, auf deren Grundlage z.B. dynamische Vorgänge an Sequenz- oder Kollaborationsdiagrammen demonstriert werden. Eine Strukturierung dieser Art ohne handlungsorientierte Einbettung, hier in den Kontext des Softwareentwicklungsprozesses, wird im Folgenden als fachsystematische Strukturierung bezeichnet. Ziel der Arbeit ist es, an Stelle der fachsystematischen Strukturierung eine handlungsorientierte Strukturierung in Anlehnung an das Vorgehensmodell des Softwareentwicklungsprozesses zu erarbeiten.

Lehrer und Ausbilder stehen dem Softwareentwicklungsprozess oftmals ablehnend gegenüber, weil man den Einsatz teurer und unübersichtlicher Entwicklungswerkzeuge fürchtet, die im Unterricht oder in der betrieblichen Ausbildung nur schwer handhabbar erscheinen.

Aus dieser Situation heraus sollen einige Thesen für einen zeitgemäßen Unterricht im genannten Lernfeld aufgestellt werden:

- Das Vorgehensmodell löst die Fachsystematik bei der Strukturierung des Lernfelds „Entwickeln und Bereitstellen von Anwendungssystemen“ ab.
- Die Verwendung eines Vorgehensmodells hilft den Schülern, eigene Erfahrungen aus den Betrieben einzubringen.
- Das Verständnis des Softwareentwicklungsprozesses wird durch Herstellung eines gemeinsamen Dokumentationsstands für alle Schüler zu Beginn jeder Phase des Vorgehensmodells gefördert.
- Bis zur Phase des objektorientierten Entwurfs (OOD) kann ohne Entwicklungswerkzeuge unterrichtet werden.
- Die objektorientierte Programmiertechnik fördert vernetztes Denken im Gegensatz zu linear-analytischem Denken in der strukturierten Programmiertechnik.
- Die objektorientierte Programmiertechnik wird durch die Metasprache UML (Unified Modeling Language) übersichtlicher.

Hiermit verbunden ist eine neue Didaktik für die objektorientierte Programmiertechnik, für die im Rahmen dieser Arbeit ein Zugang über das Vorgehensmodell entwickelt wird.

Die curricularen Rahmenbedingungen für diese Arbeit sind im Kapitel 2 beschrieben. Insbesondere wird hier auf die Betonung der Handlungsorientierung in den Rahmenlehrplänen eingegangen und der Übergang vom Handlungsfeld zum Lernfeld diskutiert. Die handlungsorientierte Strukturierung orientiert sich am Vorgehensmodell der OOSE. In Kapitel 3 wird ein auf die Bedürfnisse der Schule angepasstes Vorgehensmodell vorgestellt. Konkrete Erfahrungen mit diesem Vorgehensmodell im Rahmen einer Unterrichtsreihe zu den Phasen Anforderungsbeschreibung und Analyse werden in Kapitel 4 diskutiert. Als Beispiel für den Unterricht wird die Entwicklung eines Netzkalendariums durchgeführt. Die Hauptfunktion des Netzkalendariums liegt in der Verwaltung von Termineinträgen und der rechtzeitigen Benachrichtigung der zugehörigen Teilnehmer. Beim Netzkalendarium handelt es sich um ein mehrbenutzerfähiges Programm, das im Internet verteilt auf unterschiedlichen Hardwareplattformen einsatzfähig ist ([HK99], S. 13ff). Das Vorgehensmodell wurde nicht speziell auf dieses Beispiel angepasst. Es ist für ähnlich umfassende Beispiele hinreichend allgemein. Das Netzkalendarium liegt vollständig modelliert und kodiert vor [HK99]. Die Lehrerfortbildung wurde von Anfang an mit in die Konzeption eingeschlossen. Kapitel 5 stellt die konkrete Planung eines Kursmoduls der IT Akademie Hessen vor. Die Arbeit schließt mit einer Zusammenfassung und gibt einen Ausblick auf die weitere Gestaltung des Lernfelds „Entwickeln und Bereitstellen von Anwendungssystemen“.

2 Lernfeldansatz

Der Lernfeldansatz ist als Reaktion auf das Transferproblem, das bei der Übertragung von fachsystematisch erarbeiteten Inhalten auf praxisrelevante Probleme auftritt, anzusehen. Der Lernfeldansatz setzt sich das Ziel, die Handlungskompetenz in der beruflichen Bildung zu fördern und curricular abzusichern. Hierbei werden Handlungsfelder aus der Praxis, die von beruflicher, gesellschaftlicher und persönlicher Relevanz sind, aufgegriffen. Dabei ergeben sich die Lernfelder nicht durch direkte Ableitung aus den Handlungsfeldern. Vielmehr entstehen erst durch eine didaktische Aufbereitung und Reflexion wohlbegründete Lernfelder für die Schule. Hierbei ist der Bildungsauftrag der Berufsschule zu berücksichtigen. Er verlangt die Förderung der Persönlichkeitsentwicklung hin zu sozialer Verantwortung in Verbindung mit der ganz konkreten Qualifizierung für eine berufliche Tätigkeit, die auf dem Arbeitsmarkt nachgefragt wird (Qualifizierungsauftrag der Berufsschule). Die Verbindung dieser beiden Ziele der beruflichen Bildung schafft die „Herausbildung beruflicher Identität als Komponente von sozialer Integration“ [Ba00]. Damit wirkt der Bildungsauftrag als Brücke vom Handlungsfeld zum Lernfeld. Mit Blick auf den Bildungsauftrag der Berufsschule kann hier auch auf Arbeiten zur Didaktik der Informatik an allgemeinbildenden Schulen zurückgegriffen werden [Hu01]. Diese Ansätze müssen jedoch bezüglich des Qualifizierungsauftrags der Berufsschule modifiziert werden.

Es ist Aufgabe der Schulen, Lernfelder in Form von Lernsituationen weiter zu konkretisieren [BS98]. So soll die Umsetzung von erlernten Inhalten in praktischen Anwendungssituationen, in anderen Worten „Vom Wissen zum Können“ [SI00], erleichtert werden.

Mit der im Mai 1996 verabschiedeten Handreichung zur Erstellung von Rahmenlehrplänen werden Lernfelder zur Grundlage für die Gestaltung von Rahmenlehrplänen [KMK96]. In diesem Zuge ist auch der Rahmenlehrplan für den Ausbildungsberuf Fachinformatiker in Lernfeldern organisiert [RLP97].

Die Verlagerung von Teilen der curricularen Gestaltung auf die Schulen bei der Ausgestaltung der Lernfelder hat tiefgreifende Konsequenzen:

- Die Absage an eine Organisation des Unterrichts nach der Fachsystematik und die Auflösung des Fächerkanons hinterlässt eine Lücke, die erst noch durch eine zu erarbeitende handlungssystematische Strukturierung gefüllt werden muss. Neue fachwissenschaftliche Inhalte müssen integrativ im handlungsorientierten Kontext übernommen werden.
- Lehrer müssen sich mit den Handlungsfeldern der Praxis vertraut machen. Hierfür schafft eine enge Lernortkooperation zwischen der Schule und den Ausbildungsbetrieben die Grundlage.
- Angebote der Lehrerfortbildung müssen handlungsorientiert ausgelegt sein und Ansätze zu lernfeldübergreifendem Unterricht aufzeigen (Vernetzung der Kursmodule).
- Der Unterricht im Lernfeld erfolgt in Lehrerteams. Die Schulen müssen hierzu verstärkte Anstrengungen zur Teambildung unternehmen. Die Stundenpläne müssen genügend Flexibilität für die Zuordnung von Lehrerteams auf die Lerngruppen aufweisen.
- Fortwährende Evaluation der herausgearbeiteten Lernsituationen. Zur didaktischen Analyse eignen sich Erschließungsfragen (z.B. [BS98] und [S100]).
- Die Vorbereitung von Unterricht nimmt, zumindest in der Anfangsphase, mehr Zeit in Anspruch.

Der Rahmenlehrplan für den Ausbildungsberuf Fachinformatiker enthält keine methodischen Vorgaben für den Unterricht. Gleichwohl legt er nahe, dass handlungsorientierte Unterrichtsformen „angemessen berücksichtigt“ werden sollen [RLP97, S. 2]. Für das Lernfeld „Entwickeln und Bereitstellen von Anwendungssystemen“ fordert der Rahmenlehrplan [RLP97, S. 10]:

Die Schülerinnen und Schüler können komplexe Anwendungssysteme in Projekten analysieren, entwerfen, realisieren und bereitstellen. [...] Sie wenden für das Entwickeln von Anwendungssystemen eine Programmentwicklungsmethode an und erstellen die (Anwendungs-)Programme auf der Grundlage bekannter Algorithmen und Datenstrukturen unter Nutzung von Softwareentwicklungsumgebungen.

3 Auf die Situation der Schule angepasstes Vorgehensmodell

Das Vorgehensmodell muss auf die Situation in der Schule angepasst werden (siehe Abb. 1), um die Anforderungen des Bildungsauftrags zu berücksichtigen. Dazu wird das Grundgerüst der Phasen und der zu erstellenden Dokumentationen aus dem Vorgehens-

modell der Praxis übernommen. Die Unterschiede aufgrund der didaktischen Aufarbeitung treten in der Ausgestaltung dieser Grundlage hervor, die im Folgenden beschrieben wird. Die Lerngruppe wird zu Beginn der Unterrichtsreihe in Gruppen aufgeteilt. Die Teambildung ist notwendig, um die vorgenommenen Modellierungsergebnisse zu bewerten. Die Gruppen bearbeiten zu Beginn die Anforderungsbeschreibung des Netzkalendariums in Textform. Diese Phase verfolgt das Ziel, die Anforderungen möglichst gut zu verstehen und Unklarheiten zu erkennen und zu beheben.

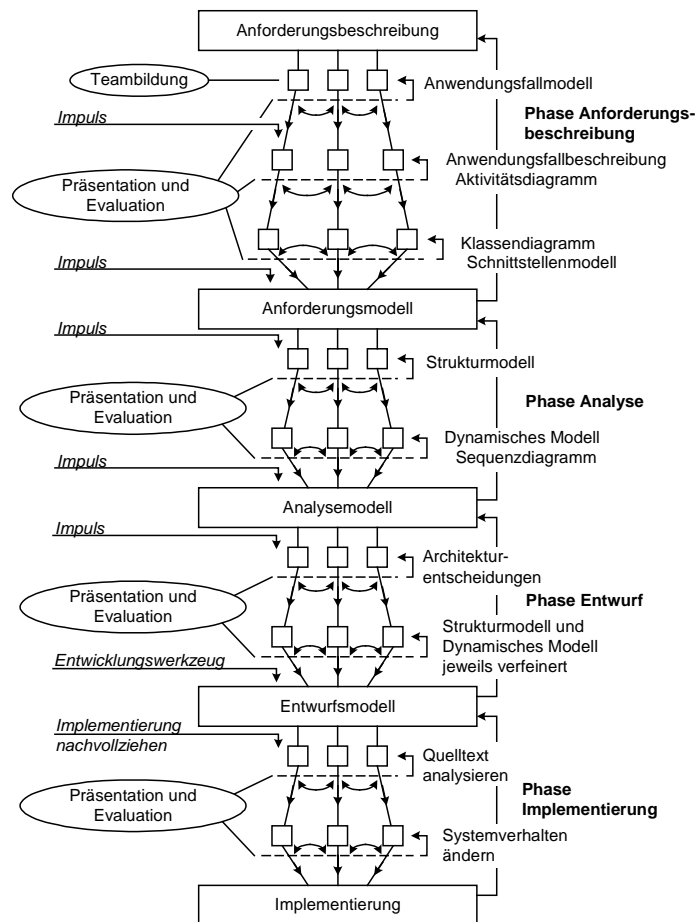


Abb. 1: Auf die Situation in der Schule angepasstes Vorgehensmodell. Die Quadrate symbolisieren die Modellierungsergebnisse der einzelnen Schülergruppen, die am Ende jeder Entwicklungsphase zu einem einheitlichen Dokumentationsstand zusammengefasst werden.

Hierzu ermitteln die Schüler die Anwendungsfälle des Netzkalendariums und fassen sie in einem Anwendungsfalldiagramm zusammen. Das Anwendungsfalldiagramm dient auch als Grundlage, um mögliche Schnittstellen zu erkennen. In den Gruppen werden die

einzelnen Anwendungsfälle ausformuliert. Bei Anwendungsfällen mit vielen alternativen Abläufen, können sie zusätzlich ein Aktivitätsdiagramm anfertigen. Aus den so abgeleiteten Beschreibungen filtern die Schüler Kandidaten für Klassen heraus und bestimmen die Abhängigkeiten der Klassen untereinander. Das sich hieraus ergebende Klassendiagramm fasst die statische Struktur zusammen. Die Schüler sollen anhand des Klassendiagramms exemplarisch demonstrieren, dass sie Kundenwünsche umgesetzt haben. Das Schnittstellenmodell beschreibt alle Übergänge vom Akteur zu den Anwendungsfällen im Anwendungsfalldiagramm. Diese Modellierungsergebnisse der Schüler werden im Anforderungsmodell zusammengefasst.

Die sich anschließende Analysephase hat das Ziel, dem Modell Klassen hinzuzufügen, die nicht unmittelbar aus der Anforderungsbeschreibung hervorgehen. Die Schüler steigen mit einem einheitlichen Dokumentationsstand ein. In den Gruppen werden die Klassen des Problembereichsmodells nach ihrer Funktion als Informations-, Kontroll- und Schnittstellenklassen gekennzeichnet. Mit Hilfe des Analysemodells der EBC-Kollaboration (entity-boundary-control-Kollaboration) ergänzen sie fehlende Klassen und erhalten so das Strukturmodell der Analysephase. Auf diesem Modell aufbauend, werden die Anwendungsfälle dynamisch z.B. mit Sequenzdiagrammen abgebildet. Diese Ergebnisse führen zum Analysemodell.

In der Entwurfsphase verschiebt sich der Schwerpunkt von der Frage „Was“ Bestandteil des Funktionsumfangs ist, hin zum „Wie“ der Umsetzung. Hierunter fallen zuerst die Architekturentscheidungen z.B. Wahl der Programmiersprache, der Entwicklungsumgebung, Realisierung der Datenpersistenz und Umsetzung der Nebenläufigkeit. Weiter ist zu klären, wie die Programmkomponenten auf unterschiedliche Rechner aufgeteilt werden und wie die Kommunikation zwischen diesen realisiert wird (Verteilungsdiagramm). Mit diesen Entscheidungen kann nun das Strukturmodell des Analysemodells verfeinert werden. Dies beinhaltet die Anpassung von Modellteilen, die in der Zielsprache nicht implementierbar sind (z.B. Auflösung der Mehrfachvererbung bei der Zielsprache Java) und den Einsatz von Entwurfsmustern. Die Umsetzung von UML Elementen (z.B. 1:n Assoziation) wird in der Zielsprache exemplarisch vorgenommen. Erst zu diesem Zeitpunkt werden Entwicklungswerkzeuge eingesetzt. Das Netzkalendarium ist damit im Rahmen der Unterrichtsreihe vollständig modelliert.

In der Implementierungsphase wird zunächst das statische Modell kodiert und die Benutzerschnittstellen erzeugt. Daran schließt sich die Programmierung des dynamischen Verhaltens an. Der Weg vom Modell zum lauffähigen Programm kann aus Zeitgründen bei einem umfassenden Beispiel wie dem Netzkalendarium nicht vollständig in der Schule erfolgen. Die beschriebenen Schritte werden daher exemplarisch vorgenommen und mit dem vom Lehrer zur Verfügung gestellten, vollständig implementierten Programm verglichen. Dabei wird der Einsatz von Entwicklungswerkzeugen demonstriert. Hierbei liegt im ersten Schritt der Schwerpunkt auf dem Nachvollziehen der Programmstruktur auf der Grundlage der UML Diagramme des verfeinerten Strukturmodells und des dynamischen Modells. Im zweiten Schritt erhalten die Schüler die Aufgabe, am fertigen Programm definierte Änderungen im Verhalten vorzunehmen. Hierzu identifizieren sie mit den genannten Modellen die dafür verantwortlichen Stellen im Quelltext und ändern sie geeignet ab (z.B. das Verhalten eines Zugriffs auf einen Kalender mit

nicht ausreichenden Rechten beim Anwendungsfall „Termin erfassen“). Ein wichtiges Lernziel ist hierbei, die Änderungslokalität zu bewerten. Die eingesetzten Entwicklungswerkzeuge sollten die Modifikationen am Quelltext im Modell übernehmen (round trip engineering), um Modell und Quelltext konsistent zu halten. Das Konzept setzt klar auf die Verwendung von Quelltext (Qualifizierungsauftrag der Berufsschule), weil eine rein auf UML basierende Programmierung [SN02] in der Praxis noch nicht einsetzbar ist.

Das Vorgehensmodell beinhaltet den iterativen Ansatz zur schrittweisen Verfeinerung der Modelle, der in Abb. 1 durch kleine nach oben zurückgreifende Pfeile symbolisiert ist. Diese Iterationen können auch auf mehrere Stufen zurückgreifen, wenn anfänglich noch nicht bekannte Anforderungen aufgedeckt werden. Das Vorgehensmodell verlangt, dass die Dokumentation zu jedem Zeitpunkt in sich konsistent ist.

In der Praxis treten in jeder Entwicklung unvorhersehbare Ereignisse auf. Diese werden im Vorgehensmodell über Impulse von außen einbezogen (siehe Abb. 1). Hierbei handelt es sich z.B. um zusätzliche Anforderungen an das Netzkalendarium oder Anfragen aus anderen Entwicklerteams. Impulse können auch von Schülern selbst eingebracht werden, indem sie eigene Erfahrungen mit Vorgehensmodellen und Anforderungen ihrer Ausbildungsbetriebe erläutern. Die Modellierungsergebnisse werden auf diese Anforderungen hin diskutiert.

Die in Abb. 1 gezeigten Präsentations- und Evaluationsphasen sind wesentlicher Bestandteil des Vorgehensmodells. Sie werden zur Bewertung der Arbeitsergebnisse der Gruppen benutzt. Die gestalterische Komponente der Modellierung [Sc01] verlangt mit ihrem Dreiklang Planen-Urteilen-Entscheiden von den Schülern Verantwortung zu übernehmen, wodurch in besonderem Maße die vom Bildungsauftrag unterstrichene Sozialkompetenz gefördert wird [Ka02].

Die Anwendungsfälle sind für das Vorgehensmodell von grundlegender Bedeutung und wichtiger Bezugspunkt. Aus ihnen werden zunächst die Klassen abgeleitet, einem Kunden wird mit ihnen das Problembereichsmodell erläutert und dynamische Modelle folgen ihren Abläufen. In der Implementierungsphase setzen die Änderungen im Verhalten bei den Anwendungsfallbeschreibungen an.

Die im Vorgehensmodell beinhalteten Phasen sind kein Selbstzweck. Ihre zeitliche Ausdehnung und ihr Detaillierungsgrad in der Ausarbeitung sind von Projekt zu Projekt verschieden. Die „Modellierung und Strukturierung“ muss dabei so umgesetzt werden, dass sie einerseits nicht zum „philosophischen Exkurs“ überhöht wird oder im anderen Extrem „spezifische Eigenheiten der verwendeten Programmiersprache in den Mittelpunkt des Unterrichts rücken“ ([Hu99], S. 24f).

4 Reflexion der Pilotunterrichtsreihe

Das beschriebene Vorgehensmodell wurde mit Fachinformatikern, Fachrichtung Anwendungsentwicklung im 3. Ausbildungsjahr bis zur Anfertigung des Analysemodells (siehe Abb. 1) am Beispiel des Netzkalendariums mit einem Umfang von 40 Schulstunden als Pilotkurs an der Oskar-von-Miller-Schule Kassel durchgeführt. Die Schüler hatten in diesem Lernfeld zusätzlich Unterricht zu Datenbanken und zur OOP mit C++.

Das Auffinden der Anwendungsfälle und die Ableitung der wichtigsten Klassen mit ihren Beziehungen hieraus gelang den Schülern ohne größere Probleme. Dies ist bemerkenswert, weil ihnen aus den eingangs genannten Gründen die objektorientierte Programmierung nicht ausreichend geläufig war. Die Schüler erkannten eine Ähnlichkeit des Klassendiagramms zum ER-Modell, das ihnen aus dem Unterricht zu Datenbanken vertraut war. In der Analysephase wurde auf den Unterschied zwischen beiden Diagrammen eingegangen.

Die Erfahrung aus der Unterrichtsreihe zeigt, dass sich Schüler schnell mit dem erreichten Detaillierungsgrad zufrieden geben. Eine iterative Verfeinerung musste daher mehrfach durch Impulse angestoßen werden.

Beim Übergang zur Analysephase griff ein Arbeitsauftrag das allgemein akzeptierte Klassendiagramm einer Schülergruppe auf und verlangte, den dynamischen Ablauf eines Anwendungsfalls mit einem Sequenzdiagramm zu modellieren. Hierbei erkannten die Schüler, dass das Klassendiagramm nicht alle notwendigen Objekte bereithält, nämlich vorwiegend Informationsobjekte. Es fehlten Objekte mit Schnittstellen- und Kontrollfunktion, die im Rahmen der EBC-Kollaboration nachgetragen werden.

Das Aufgreifen von Schülerlösungen in Verbindung mit Arbeitsaufträgen, die Ziele der anstehenden Modellierungsphase thematisieren, hat sich als sehr motivierend erwiesen. Die Schüler erkennen intuitiv, dass die Modelle noch nicht hinreichend vollständig sind. Das Vorgehensmodell erschließt sich ihnen aufgrund dieser „eigenen“ Erfahrungen. In diesem Zusammenhang brachten die Schüler Bezüge zum Vorgehensmodell ihrer Ausbildungsbetriebe ein. Die Bündelung der Schülerergebnisse zu Beginn der Entwicklungsphasen auf einen gemeinsamen Dokumentationsstand erleichterte die Unterstützung der Gruppen durch den Lehrer. Sie schaffte die Grundlage für die Vergleichbarkeit der Schülerlösungen in den Präsentationsrunden. Die Schüler erkannten, dass Modellierungsentscheidungen nicht „richtig“ oder „falsch“ sind, sondern mit den Komparativen „besser“ und „schlechter“ mit Blick auf die Kundenanforderung zu bewerten sind. Die Präsentationsrunden gaben ihnen Gelegenheit, diese Entscheidungen zu hinterfragen und zu begründen.

Der Ausbildungsberuf Fachinformatiker gliedert sich in die Fachrichtungen Anwendungsentwicklung und Systemintegration. Das Vorgehensmodell ist als Maximalmodell für Auszubildende der Fachrichtung Anwendungsentwicklung anzusehen. Für die Fachrichtung Systemintegration ist es bei einer Schwerpunktsetzung auf das Nachvollziehen der gegebenen Diagramme und in der Umsetzung von Verteilungsdiagrammen in reduzierter Form anwendbar.

5 Einbeziehung der Lehrerfortbildung

Da es sich um einen neuen Zugang zur OOP handelt und den meisten Lehrern Erfahrungen mit dem Softwareentwicklungsprozess aus der eigenen beruflichen Praxis fehlen, ist das beschriebene Konzept nur in Verbindung mit einer hierauf abgestimmten Lehrerfortbildung realisierbar.

Um den Lehrern den handlungsorientierten Zugang zu erleichtern, orientiert sich die Fortbildung am vorgestellten Vorgehensmodell. Als Beispiel wird auch hier das Netzk-

alendarium aufgegriffen. Die konkrete Fortbildung an der IT Akademie Hessen¹ ist in 3 Blöcke mit jeweils 4 Tagen unterteilt. Der erste Block greift an kleinen Beispielen die Phasen Anforderungsbeschreibung und Analyse auf. Der zweite Block schließt die Diskussion der Entwurfs- und Implementierungsphase an. Voraussetzung für die Teilnahme sind Kenntnisse einer objektorientierten Programmiersprache. Im dritten Block wird das Vorgehensmodell durchgehend am Beispiel des Netzkalendariums behandelt.

Die Fortbildung legt den Schwerpunkt auf die Erarbeitung der Modellierungsergebnisse in Gruppenarbeit durch die Lehrer. Der Referent leitet lediglich durch kurze Eingangsaussagen in die jeweiligen Entwicklungsphasen ein und tritt ansonsten als Coach auf, der bei Fragen Hilfestellung anbietet. Mit diesem Vorgehen soll bei den Teilnehmern Verständnis für die Notwendigkeiten des OOSE-Prozesses erzeugt werden. Zudem sollen die Lehrer hierdurch Materialien für den Unterricht erarbeiten. Es ist angedacht, einen 4. Fortbildungsblock für die vollständige Ausarbeitung eines Unterrichtsskriptes anzuschließen. Zur Vorbereitung stehen den Lehrern einführende Materialien zu den UML Diagrammtypen zur Verfügung.

6 Zusammenfassung und Ausblick

Die durchgeführte Unterrichtsreihe zeigte, dass sich das Vorgehensmodell als handlungsorientierte Strukturierung für das Lernfeld „Entwickeln und Bereitstellen von Anwendungssystemen“ eignet. Es erleichtert Schülern, eigene Erfahrungen aus den Betrieben einzubringen. Gerade die vielfach interdependente Struktur des Vorgehensmodells mit einer Betonung der Integration von Kundenanforderungen hat vernetztes Denken gefördert. Die phasenweise Zusammenführung der Schülerergebnisse auf einen einheitlichen Stand, führte zu einer guten Vergleichbarkeit in den Präsentationsrunden. Es half dem Lehrer im Verlauf der Unterrichtsreihe, den Überblick über die Schülerergebnisse zu behalten. Die Entscheidung, auf ein Entwicklungswerkzeug zu verzichten, ließ mehr Zeit für die methodische Arbeit im Unterricht. Die Erfahrungen mit dem OOSE-Prozess sollte die schnelle Einarbeitung in eines der mehreren Dutzend Entwicklungswerkzeuge erleichtern, weil die Schüler die zugrundeliegende Entwicklungsmethodik in der Unterrichtsreihe kennen gelernt haben. Die Metasprache UML erweist sich als Schlüssel zur programmiersprachenunabhängigen Softwareentwicklung. Alternativ kann auch ohne OOP Kenntnisse bis zur Analysephase gearbeitet werden. Über die Entwurfsphase hinaus sind sie dann zwingend notwendig. Die automatische Quelltexterzeugung aus den UML Diagrammen schafft dem Lehrer zudem Raum, sich auf die Unterrichtsmethodik zu konzentrieren, statt der sich beschleunigenden technischen Entwicklung einer Vielzahl von Programmiersprachen hinterhereilen zu müssen.

Das gewählte umfassende Beispiel ließ den Mehraufwand für die Modellierung und Dokumentation bei den Schülern als gerechtfertigt erscheinen. Ob sich eine dann nur teilweise Kodierung als problematisch erweist, muss noch untersucht werden.

Das beschriebene Vorgehensmodell wird nach dieser Pilotphase an der Oskar-von-Miller-Schule Kassel im zweiten Ausbildungsjahr im ganzen Umfang durchgeführt werden. Anpassungen des Vorgehensmodells auf die Schulformen Berufliches Gymnasium

¹ Programm der IT Akademie Hessen für den Herbst 2002 (www.ita.hessen.de).

und Fachschule für Technik sind geplant. Auf der Grundlage der Lehrerfortbildung soll dem Vorgehensmodell eine schulübergreifende Komponente hinzugefügt werden, bei der die Schüler Softwareteillösungen verteilt entwickeln.

Literaturverzeichnis

- [Ba00] Bader, R.: Stand der wissenschaftlichen Forschung zum Lernfeld-Konzept. (<http://www.uni-magdeburg.de/ibbp/bp/downloads/Lernfeld-Konzept.pdf>, 12.7.2002), 2000.
- [BS98] Bader, R.; Schäfer, B.: Lernfelder gestalten – Vom komplexen Handlungsfeld zur didaktisch strukturierten Lernsituation. In: Die berufsbildende Schule (BbSch), 50 (1998), 7-8.
- [HK99] Hitz, M.; Kappel, G.: UML@Work: Von der Analyse zur Realisierung. dpunkt-Verlag, Heidelberg, 1999.
- [Hu99] Hubwieser, P.: Modellieren in der Schulinformatik. In: Log In, Vol. 19, Nr.1, 1999, S. 24-29.
- [Hu01] Hubwieser, P.: Didaktik der Informatik: Grundlagen, Konzepte, Beispiele. 1. korrigierter Nachdruck. Springer, Berlin, 2001.
- [Ka02] Kath, F.M.: Paradigmenwechsel auch in der Fachdidaktik – Wunsch oder Realität? In: Die berufsbildende Schule (BbSch), 54 (2002), 4.
- [KMK96] KMK – Sekretariat der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland: Handreichungen für die Erarbeitung von Rahmenlehrplänen der Kultusministerkonferenz für den berufsbezogenen Unterricht in der Berufsschule und ihre Abstimmung mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe. Bonn, 1996.
- [RLP97] KMK – Sekretariat der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland: Rahmenlehrplan für den Ausbildungsberuf Fachinformatiker/Fachinformatikerin. (Beschluss der Kultusministerkonferenz vom 25. April 1997), 1997.
- [Sc01] Schulte, C.: Vom Modellieren zum Gestalten – Objektorientierung als Impuls für einen neuen Informatikunterricht. In: *informatica didactica* 3, 2001 (<http://ddi.cs.uni-potsdam.de/InformaticaDidactica/Issue3>, 9.7.2002).
- [SI00] Sloane, P.F.E.: Lernfelder und Unterrichtsgestaltung. Die berufsbildende Schule (BbSch), 52 (2000), 3.
- [SN02] Schulte, C.; Niere, J.: Thinking in Object Structures: Teaching Modelling in Secondary Schools. Sixth Workshop on Pedagogies and Tools for Learning Object Oriented Concepts. ECOOP 2002, Malaga, Spain, June 11, 2002.