

Informatische Bildung in Anwenderschulungen

Siglinde Voß

Fachbereich Didaktik der Informatik
Technische Universität München
Boltzmannstraße 3
D-85748 Garching
siglinde.voss@web.de

Abstract: Anwenderorientierte Schulungen, etwa für Tabellenkalkulationssoftware, gewinnen mit zunehmender Komplexität dieser Werkzeuge an Bedeutung, da die Anforderungen an die Fähigkeiten der Anwender mit fortschreitender Entwicklung dieser Werkzeuge steigen.

In der beruflichen Fort- und Weiterbildung sind somit Konzepte für nachhaltige Benutzerschulungen gefordert. Informatische Modelle erweisen sich dabei als geeignet, Dokumentstrukturen zu vermitteln, die im Gegensatz zu den schnelllebigen Benutzeroberflächen längerfristig Bestand haben und helfen, die Arbeitsweise moderner Softwaresysteme zu verinnerlichen.

Der Beitrag stellt Unterrichtskonzepte und –inhalte für Anwenderschulungen für Tabellenkalkulationssysteme in der beruflichen Weiterbildung vor, wie sie im Rahmen einer Mitarbeiterschulung für Büroangestellte bzw. einem Wahlpflichtkurs in der Oberstufe eines Gymnasiums umgesetzt worden sind.

1 Modellieren im Fachunterricht Informatik

Verfolgt man die fachdidaktische Diskussion der letzten Jahre¹, so ist ein großer Teil der Auseinandersetzung mit der Thematik Modellierung im Informatikunterricht zu beobachten. Hubwieser entwickelte ausgehend vom informationszentrierten Ansatz ein Unterrichtskonzept, welches die informatische Modellierung im Informatikunterricht an Gymnasien in den Mittelpunkt des Unterrichts rückt [Hu00].

Für die 6. Jahrgangsstufe schlägt er eine objektorientierte Modellierung bzw. Analyse von Standardsoftware vor, die es „den Schülerinnen und Schülern ermöglicht, sich bei der Manipulation der Dokumente unabhängig von der (...) produktspezifischen Bedienoberfläche des jeweiligen Werkzeugs an einer „inneren“ Struktur der Dokumente zu orientieren.“ [Fr00].

Während an allen bayerischen Gymnasien in der 6. Jahrgangsstufe mit der Analyse und Strukturierung von Textverarbeitungs- bzw. Präsentationsdokumenten die objektorientierte Modellierung im Vordergrund steht [FW04], [Vo04], sieht das Curriculum für

¹ Brinda fasst in seiner Dissertationsschrift gegenwärtige Ergebnisse und den aktuellen Stand der Forschungsdiskussion im Bereich Objektorientierter Modellierung zusammen. [Br04], S. 15 ff

das Pflichtfach Informatik für die 8. Jahrgangsstufe in der naturwissenschaftlich-technologischen Ausbildungsrichtung unter anderem die funktionale Modellierung in den Vordergrund gerückt [St04]. Im Pflichtunterricht werden Tabellenkalkulationssysteme ausschließlich zur Simulation und Umsetzung von funktionalen Modellen genutzt, dargestellt als Datenflussdiagramme [Hu04], [Sc05].

Auch an den bayerischen Realschulen wird derzeit in Schulversuchen getestet, das klassische Fach Textverarbeitung durch ein um informatische Inhalte erweitertes Fach Informationstechnologie abzulösen, welches dem Modellieren von Softwaresystemen einen breiten Rahmen einräumt. Offenbar hat sich die Erkenntnis durchgesetzt, dass insbesondere auch zur Vorbereitung auf Büroberufe fundierte Kenntnisse vermittelt werden müssen, was ein an einem bestimmten Produkt und deren Menüoberfläche orientierter Anwenderunterricht nicht leisten kann.

Ähnliche Ansätze sind in Österreich zu beobachten: Tabellenkalkulationssysteme werden in den Gymnasien zur Umsetzung des Curriculums in großem Umfang eingesetzt. Geeignete Modelle sollen den Schülerinnen und Schülern einen pragmatischen Zugang zu diesen Systemen ermöglichen [Ho05].

2 Tabellenkalkulationssysteme im Unterricht

Bei der Vermittlung fachinformatischer Konzepte im Unterricht können Tabellenkalkulationssysteme auf der einen Seite genutzt werden, informatische Strukturen zu identifizieren und zu analysieren: Die Prinzipien der objektorientierten, zustandsorientierten bzw. funktionalen oder ablauforientierten Modellierung werden am konkreten Anwendersystem entdeckt und verstanden. Andererseits werden mit Hilfe der erlernten Modellierungstechniken elektronische Dokumente konstruiert und entworfen, das Modell wird dann umgesetzt und dadurch am System simuliert.

In der beruflichen Weiterbildung kann man jedoch von den Teilnehmern keine informatische Vorbildung erwarten. Der didaktischen Reduktion kommt daher eine wichtige Bedeutung zu. Graphische Repräsentationen informatischer Modelle werden genutzt, den Anwendern ein anschauliches Hilfsmittel bereitzustellen. Im Vordergrund steht hier eine deskriptive Modellierung, deren Ziel vornehmlich die Illustration der Dokumentstruktur bzw. Funktionsweise der Dokumentbausteine ist. Sie soll den Anwender unterstützen, elektronische Dokumente effizient zu entwerfen und zu gestalten bzw. Mängel zielgerichtet zu beheben und damit Probleme in der täglichen Berufspraxis zu bewältigen.

3 Tabellenkalkulationssysteme² als Rechensysteme

3.1 Klassen- und Objektstruktur in Tabellen

Tabellendokumente dienen in erster Linie zur Berechnung von Daten. Sie enthalten Zellen, deren Zellelemente entweder Zahlen, Zeichenketten oder Formeln sind. Zur Verarbeitung von Daten kann eine Formel mehrere Zellen oder Zellmatrizen referenzieren, dabei wird jeweils ein Referenzrahmen erzeugt, der die entsprechende Zelle oder Zellmatrix umrahmt.

Die relevanten Klassen und entsprechende Beziehungen können durch folgendes Klassendiagramm³ dargestellt werden:

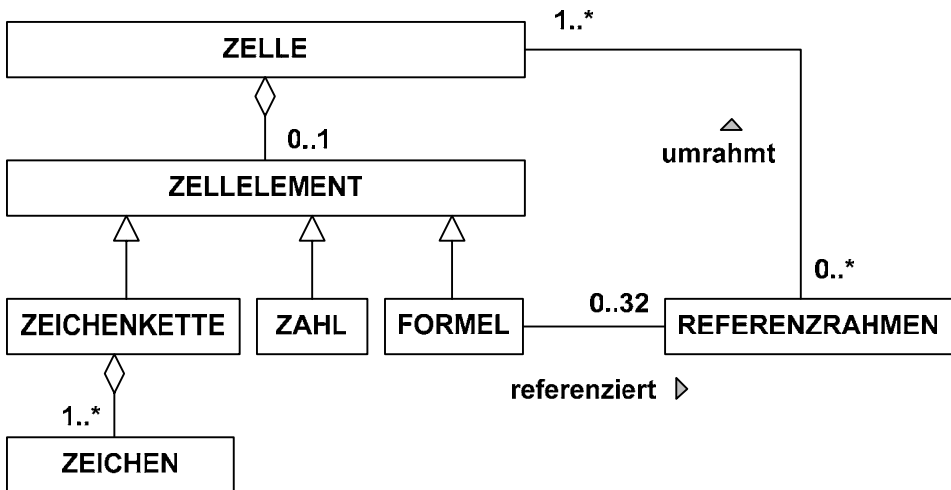


Abbildung 1: Klassendiagramm im Tabellendokument

Die in Abbildung 1 dargestellten Konzepte der objektorientierten Modellierung wie Aggregation, Vererbung usw. werden in Anwenderschulungen nicht thematisiert. Dort hingegen werden ausschließlich Objektdiagramme zur Veranschaulichung einer konkreten Beispielausprägung herangezogen. Abbildung 2 zeigt ein Beispiel, welches alle typischen Zellelemente und deren Beziehungen zueinander vorstellt.

In Anwenderschulungen werden anhand des Objektmodells konkrete Dokumente erzeugt oder typische Verhaltensweisen von Dokumentbausteinen erläutert.

² Alle hier ausgeführten Modelle beziehen sich auf Tabellendokumente des Softwarepakets OpenOffice 1.1.3 bzw. StarOffice 7.0 der Firma *Sun microsystems*.

³ Klassendiagramme sind nach dem UML (Unified Modeling Language)-Standard konstruiert [Oe01]

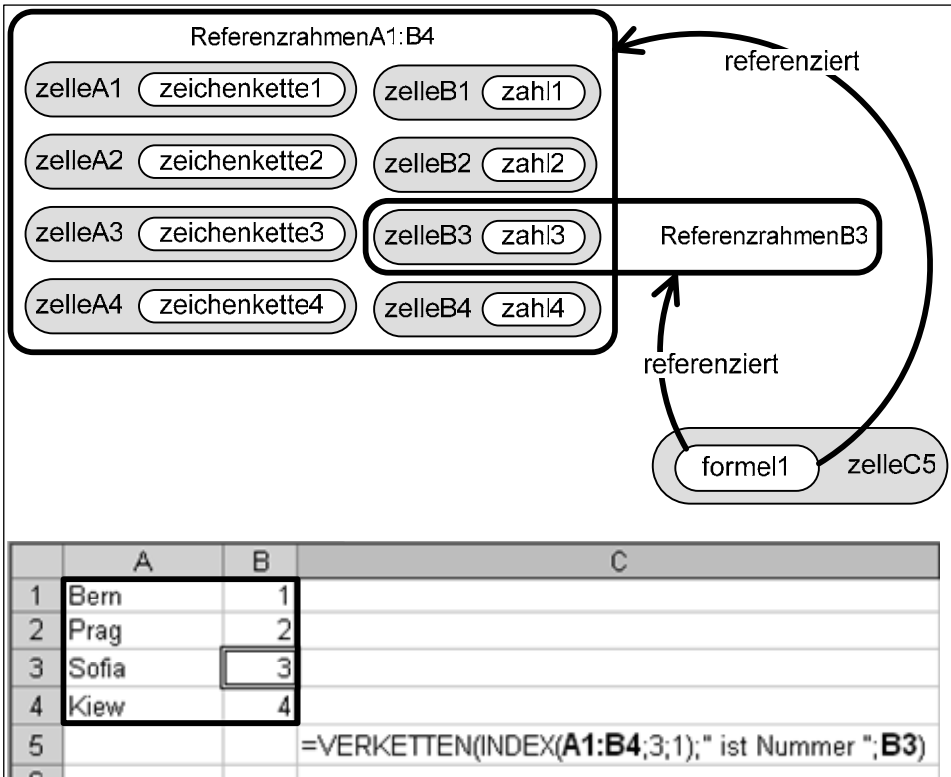


Abbildung 2: Objektdiagramm (oben) und zugehörige Beispielausprägung (unten)

Die Zellen *zelleA1* bis *zelleA4* enthalten Zeichenketten, die Zellen *zelleB1* bis *zelleB4* dagegen Zahlen mit den Zahlenwerten 1 bis 4. Die Zelle *zelleC5* enthält eine Formel, dessen Formelterm unten gerade sichtbar ist. *Formel1* holt sich die Argumente für die Berechnung von den mit den Referenzrahmen umrahmten Zellen. Die entsprechende Referenzbeziehung ist im Objektdiagramm durch Pfeile symbolisiert.

3.2 Datenflussdiagramme zur Beschreibung von Berechnungen

Im funktionalen Modell kommt der Zelle eine zentrale Rolle zu. Sie ist die Schnittstelle zur Umgebung und dient der Eingabe bzw. Ausgabe eines Datenwerts. Eine enthaltene Formel wird als Daten verarbeitender Prozess dargestellt, der evtl. in Teilprozesse zerlegt werden kann. Dort finden die Berechnungen statt, welche die Formel vorgibt [Ba00].

Anschaulich lässt sich die Berechnung durch ein so genanntes Datenflussdiagramm beschreiben [Hu04], [Sc05]: Die Rechtecke symbolisieren die Zellen als Schnittstellen, die Pfeile stellen den Datenfluss dar und die Ovale repräsentieren den Rechenprozess. An den Pfeilen können die Daten einer Beispielimplementierung notiert werden.

Abbildung 3 zeigt das Datenflussdiagramm, welches zur Termdarstellung von *Formell* in Abbildung 2 gehört. Die zu verarbeitenden Daten werden der Zellmatrix mit der Referenzadresse A1:B4 bzw. B3 entnommen, durch die Funktionen *INDEX* und *VERKETTEN* weiterverarbeitet und das Ergebnis in der Zelle *zelleC5* ausgegeben.

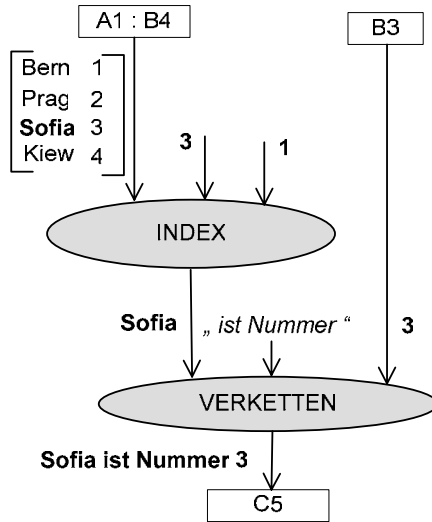


Abbildung 3: Einfaches Datenflussdiagramm

In Anwenderschulungen werden einfache Datenflussdiagramme einerseits dazu genutzt, auch Zwischenergebnisse, nämlich die Ausgabewerte einzelner beteiligter Funktionen sichtbar zu machen und somit die Konstruktion der Termdarstellung einer Formel zu erleichtern. Andererseits kann damit die Funktionalität der im Tabellenkalkulationssystem bereits zur Verfügung gestellten Funktionen an geeigneten Beispielen visualisiert werden, was zum Verständnis der Wirkungsweise komplexer Funktionen (Black-Box-Sicht) völlig ausreicht.

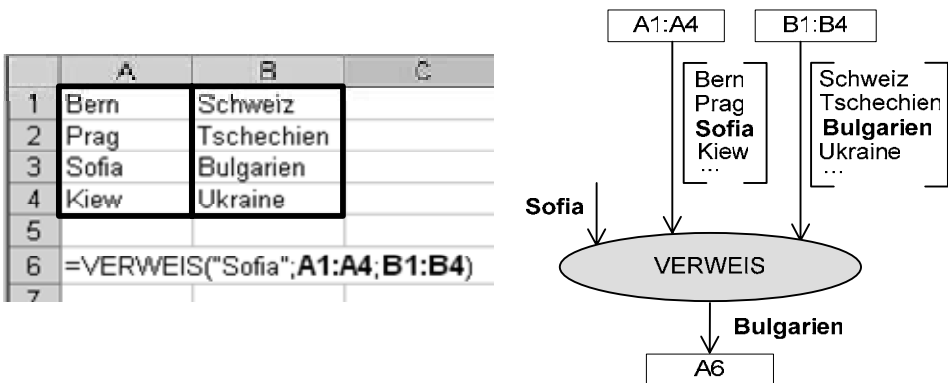


Abbildung 4: Funktionalität der Funktion VERWEIS

Die Darstellung in Abbildung 4 rechts zeigt beispielhaft, dass die Funktion VERWEIS dreistellig ist, ein Skalar und zwei Vektoren gleicher Länge als Eingabeparameter benötigt und den Wert im Ergebnisvektor ausgibt, welcher dieselbe Stelle einnimmt, wie der Suchwert im Suchvektor.

Ist die Funktionalität bekannt, lassen sich einzelne Funktionen als Bausteine für komplexe Anwendungsaufgaben verwenden. Datenflussdiagramme erleichtern das selbständige Konstruieren umfangreicher Terme, da im Modellierungsstadium der schrittweise Fortgang der Berechnung veranschaulicht werden kann.

Ein Ergebnis des Auftrags „Konstruiere ein Datenflussdiagramm zur Umsetzung eines Quizspiels Städte-Länder“ ist Folgendes:

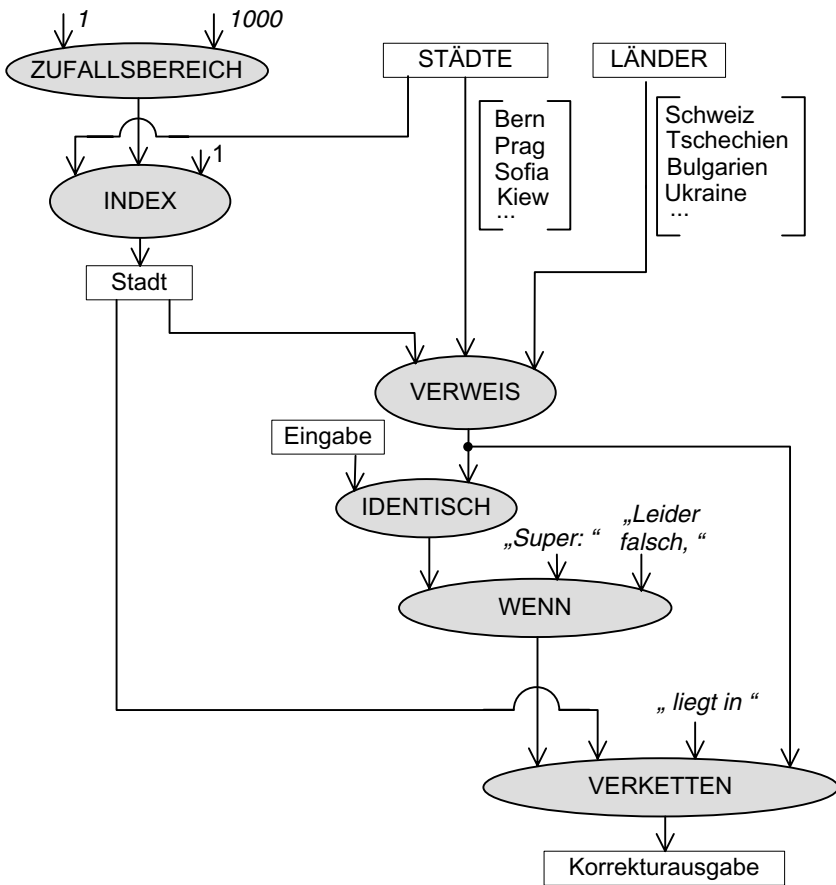


Abbildung 5: Datenflussdiagramm „Städte-Länder-Quiz“

Die Zellvektoren „STÄDTE“ bzw. „LÄNDER“ der Länge 1000 sind an einer für den Spieler nicht einsehbaren Stelle angelegt. Die Zelle „Stadt“ enthält eine Formel mit dem

Formelterm „=INDEX(STÄDTE;ZUFALLSBEREICH(1;1000);1)“ und gibt einen zufällig gewählten Begriff aus „STÄDTE“ aus. Das korrespondierende Wort aus dem Zellvektor mit den Ländernamen wird anschließend mit dem Eingabewort des Spielers in der Zelle „Eingabe“ verglichen und eine entsprechende Korrektur mit der richtigen Lösung in Zelle „Korrekturausgabe“ ausgegeben. Der zur Zelle „Korrekturausgabe“ gehörige Formelterm lässt sich nun relativ leicht bestimmen. Der Term kann mit Hilfe des Datenflussdiagramms von unten nach oben aufgebaut werden und lautet also „=VERKETTEN(WENN(IDENTISCH(Eingabe;VERWEIS(Stadt;STÄDTE;LÄNDER));„Super.“;„Leider falsch.“);Stadt;„liegt in“;VERWEIS(Stadt;STÄDTE;LÄNDER))“.

3.3 Endliche Automaten zur Festlegung des Ausgabeformats

Unabhängig vom eigentlichen Zellwert des Zellelements kann die Zelle durch ihr Ausgabeformat die sichtbare Darstellung des Zellelements festlegen. Die Berechnung der Darstellung wird als zweistellige Funktion aufgefasst. Das Zellelement liefert den genauen Zellwert, die Zelle ihren Formatcode.

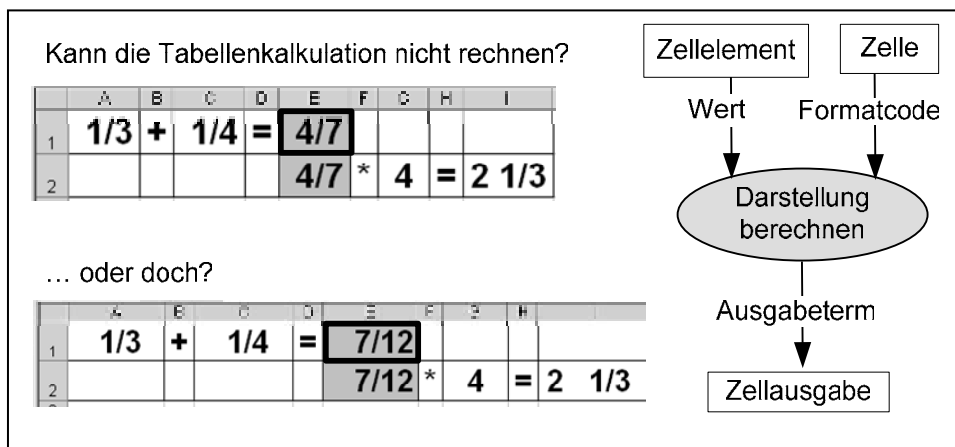


Abbildung 6: Funktionales Modell zur Berechnung des Ausgabeformats

An Beispielen lässt sich der Unterschied zwischen Zahlenwert und Ausgabe eines Zellelements verdeutlichen: Bei gleichem Zahlenwert wird die Zellausgabe bei unterschiedlichem Formatcode auch unterschiedlich ausfallen. Das in Abbildung 6 rechts dargestellte Funktionale Modell liefert ein Erklärungsmuster für die scheinbare falsche Rechnung des Tabellenkalkulationswerkzeugs von Abbildung 6 links.

Den Anwendern wird bewusst, dass sie nicht den tatsächlichen genauen Wert des Zellelements sehen, sondern lediglich die Zellausgabe, dessen Darstellung durch den Formatcode der Zelle beeinflusst ist.

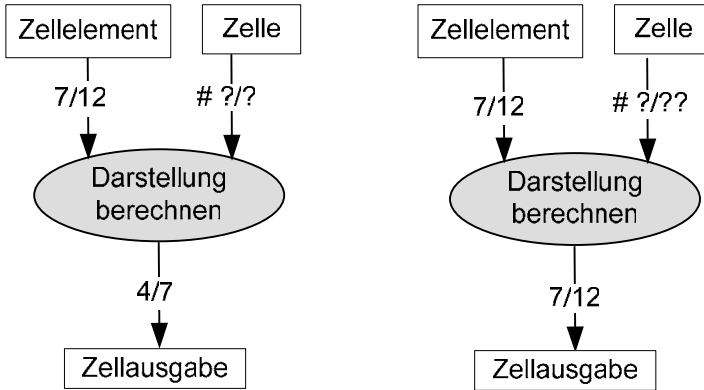


Abbildung 7: Verschiedene Ausgabeformate bei gleichem Zahlenwert des Zellelements

Die in Tabellenkalkulationssystemen gültigen Formatcode-Wörter folgen Regeln, welche anhand des Konzepts der endlichen Automaten formuliert werden können. Auch in diesem Fall wird man sich in Anwenderschulungen auf einfachste Automaten beschränken, welche nur die wichtigsten Anwendungsfälle einbeziehen. So zeigt folgende Abbildung die gängigsten Möglichkeiten für die Erzeugung von Formatcode-Wörtern, für Bruchzahlen mit möglicher nachfolgender Benennung.

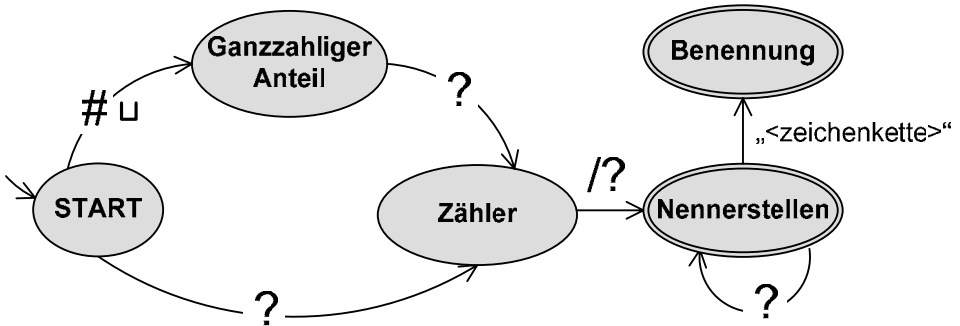


Abbildung 8: Automatenmodell zur Erzeugung gültiger Formatcode-Wörter für Bruchzahlen

Das in Abbildung 8 dargestellte Automatenmodell [Sc99] stellt in diesem Sinne eine Verallgemeinerung des Konzepts dar, welches durch die Datenflussdiagramme von Abbildung 7 beispielhaft dargestellt ist. Formatcodewörter für Gemischte Brüche beginnen mit # und einem Leerzeichen. Anschließend folgt ein ?-Zeichen, gefolgt von / . Die Anzahl der Nennerstellen wird durch die Anzahl der nun folgenden ?-Zeichen festgelegt. Die Einschränkung, dass in der konkreten Umsetzung die Anzahl der ?-Zeichen für die Nennerstellen natürlich nicht beliebig groß sein darf, wie das Automatenmodell suggeriert, wird hier nicht berücksichtigt. Mit einer in „-Zeichen eingeschlossenen Zeichenkette kann das Wort mit einer Benennung abgeschlossen werden.

3.4 Formatvorlagen für Zellen und bedingte Formatierung

Unter Berücksichtigung von Formatvorlagen werden die Objektstrukturen von Tabellendokumenten zunehmend komplexer und vielfältiger. Auch wenn man sich nur auf Zell- und Bedingungsformatvorlagen beschränkt, müssen bereits fünf verschiedene Arten von Beziehungen berücksichtigt werden:

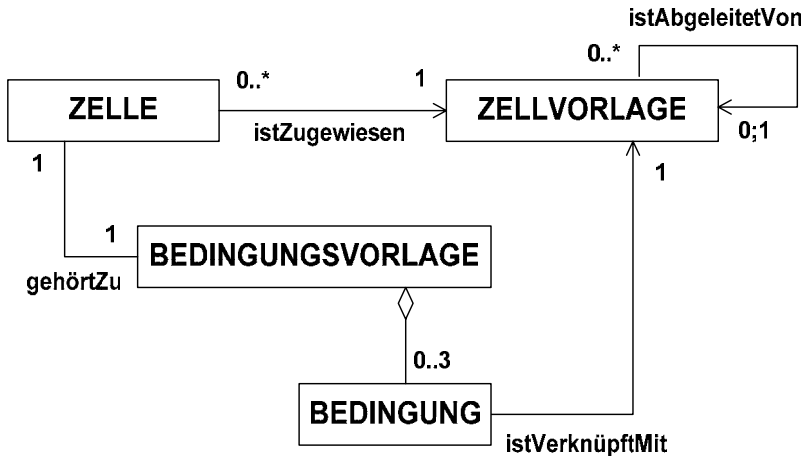


Abbildung 9: Klassendiagramm unter Berücksichtigung von Vorlagen

Zellvorlagen speichern Listen von Attributwerten zur Formatierung der ihnen zugewiesenen Zellen, ähnlich wie Absatzvorlagen in Textdokumenten [Vo05]. Analog ist es möglich, eine gegebene Zellvorlage als Basis für eine neue zu verwenden, wenn sich zugewiesene Zellen nur geringfügig in ihren Attributwerten unterscheiden sollen. Die Beziehung *istAbgeleitetVon* drückt die Möglichkeit aus, Zellvorlagen voneinander ableiten zu können. Ist ein Attributwert einer Formatvorlage nicht explizit festgelegt, der „Wert“ wird dann mit *null* bezeichnet, so liest diese den entsprechenden Wert von der zugehörigen Quellvorlage ab.

Zu jeder Zelle gehört genau eine Bedingungsformatvorlage, welche bis zu drei Bedingungen enthalten kann. Jede dieser Bedingungen ist mit genau einer Zellvorlage verknüpft, welche die Attributwerte dort gesetzter Attribute nur in gewissen Fällen in der Zelle überschreibt.

Als Übungsaufgabe stellt man beispielsweise ein in einer Tabelle gespeichertes Kalenderblatt zur Verfügung. Über bedingte Formatierung soll nun die Hintergrundfarbe aller Zellen gelb gesetzt werden (mittels *ZellvorlageGelb*), deren enthaltener Datumswert einem Samstag oder Sonntag entspricht. Bei einem Monatswechsel soll „automatisch“ eine horizontale Linie (mittels *ZellvorlageLinie*) gezogen werden.

Das zugehörige Objektdiagramm, das je nach Leistungsfähigkeit differenziert, entweder selber zu konstruieren ist bzw. als Anhaltspunkt vorgegeben wird, zeigt Abbildung 10 mit den entsprechenden Objektkarten der Bedingungen von *BedingungsformatvorlageA258*.

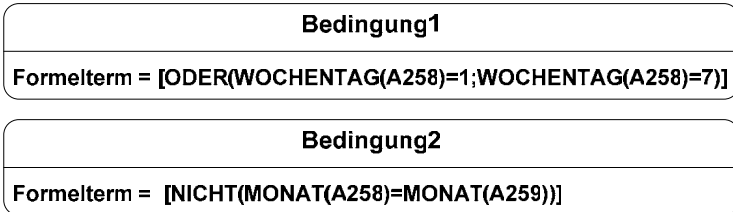
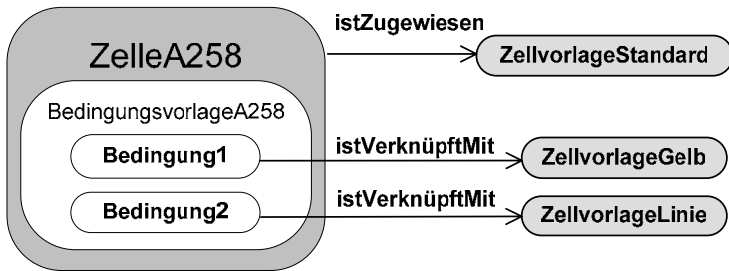


Abbildung 10: Objektdiagramm und Objektkarten der Beispielanwendung

Die Beispielausprägung im Tabellendokument könnte so aussehen:

	A	B	C	D	E
240	09.09.25				
241	10.09.25				
242	11.09.25				
243	12.09.25				
244	13.09.25				
245	14.09.25				
246	15.09.25				
247	16.09.25				
248	17.09.25				
249	18.09.25				
250	19.09.25				
251	20.09.25				
252	21.09.25				
253	22.09.25				
254	23.09.25				
255	24.09.25				
256	25.09.25				
257	26.09.25				
258	27.09.25				

Bedingte Formatierung

Bedingung 1

Formelart:

Zellvorlage:

Bedingung 2

Formelart:

Zellvorlage:

Bedingung 3

Abbildung 11: Beispielausprägung des Lösungsdokuments

Ablauforientierte Modelle, dargestellt als Entscheidungsbäume oder Struktogramme, visualisieren die Kontrollstrukturen, welche beim Formatieren mit Formatvorlagen durchlaufen werden. Wird die der Zelle zugewiesene Zellvorlage (hier: *Zellvorlage-Standard*) der Einfachheit halber nicht berücksichtigt, kann dafür der in Abbildung 12 gezeigte Entscheidungsbaum herangezogen werden. Dieser wird vornehmlich bei Büroanwendern eingesetzt, welche im Interpretieren von Struktogrammen nicht geübt sind.

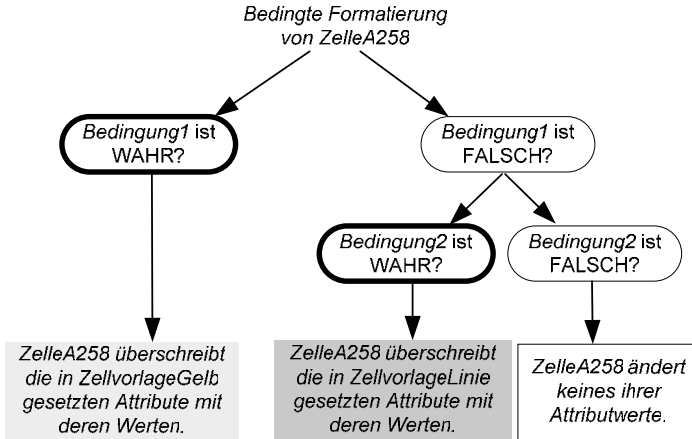


Abbildung 12: Entscheidungsbaum zur bedingten Formatierung von *ZelleA258*

Der Entscheidungsbaum hilft auch zu verstehen, warum beim Jahreswechsel 2005/2006 keine horizontale Linie trotz Monatswechsel gezeichnet wird. Da der 31.12.2005 ein Samstag ist, ist bereits *Bedingung1* erfüllt, weshalb der rechte Entscheidungsteilbaum keine Berücksichtigung mehr findet, auch wenn *Bedingung2* ebenfalls erfüllt ist.

Der Entscheidungsbaum dient evtl. auch als Vorstufe zur Konstruktion des Struktogramms. Das in Abbildung 13 gezeichnete Struktogramm berücksichtigt noch die notwendige Sequenz der Formatierung durch die der Zelle (direkt) zugewiesene Zellvorlage.

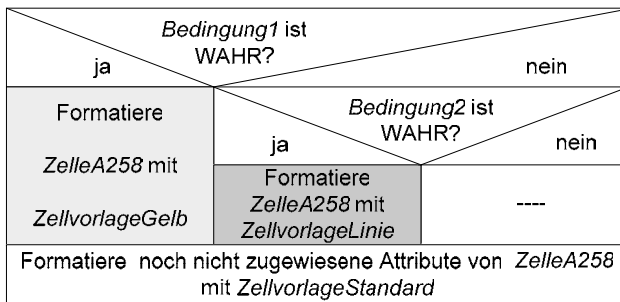


Abbildung 13: Struktogramm zur bedingten Formatierung von *ZelleA258*

3 Zusammenfassung

In der beruflichen Weiterbildung können informatische Modelle genutzt werden, die Komponenten und die Wirkungsweise von Tabellenkalkulationssystemen zu veranschaulichen. Es eröffnet die Möglichkeit, den Spagat zwischen Schulung von Anwendersoftware und Vermittlung informatischer Kompetenzen zu schaffen. Tabellenkalkulationssysteme können eingesetzt werden, verschiedenartige Fachkonzepte zu vermitteln bzw. informatische Strukturen zu identifizieren und zu analysieren.

Literaturverzeichnis

- [Ba00] Balzert H.: Lehrbuch der Software-Technik; Heidelberg, Berlin; Spektrum, Akad. Verl. Bd. 1. Software-Entwicklung, 2000.
- [Br04] Brinda T.: Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II. Dissertation, FB Elektrotechnik und Informatik, Universität Siegen, 2004.
- [FHW04] E. Frey, P. Hubwieser, F. Winhard: Informatik 1. Objekte, Strukturen, Algorithmen. Ernst Klett Verlag, Stuttgart, 2004. (In Bayern zugelassenes Schulbuch für den Informatikunterricht in der 6. und 7. Jahrgangsstufe)
- [Fr00] Frey E., Hubwieser P., Humbert L., Schubert S., Voß S.: Informatik-Anfangsunterricht. In: LOG IN 21 (2001) 1, S. 20-32.
- [Ho05] Hodnigg K.: A Pragmatic Approach to Spreadsheet Training Based Upon the “Projection-Screen” Model. In: [Mi05]
- [Hu00] Hubwieser, P.: Informatik am Gymnasium. Ein Gesamtkonzept für einen zeitgemäßen Informatikunterricht. Habilitationsschrift, Fakultät für Informatik, TU München, 2000.
- [Hu03] Hubwieser, P. (Hrsg.): Informatische Fachkonzepte im Unterricht. Köllen, Bonn, 2003.
- [Hu04] Hubwieser, P.: Functional Modelling in Secondary Schools Using Spreadsheets, Education and Information Technologies vol. 9, no. 2, pp. 175-183, June 2004.
- [Mi05] Mittermeir, R. ed., Informatics for Secondary Schools - Evolution and Perspectives: From Computer Literacy to Informatics Fundamentals; Proceedings of ISSEP 2005, Klagenfurt.
- [Oe01] Oestereich B.: Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified modeling language; München, Wien: Oldenbourg, 2001.
- [Ra04] ZBORNIK RADOVA (Ed.), SAVREMENE INFORMATICKE I OBRAZOVNE TEHNOLOGIJE I NOVI MEDIJI U OBRAZOVANJU, Sombor 2004.
- [Sc99] Uwe Schöning: Theoretische Informatik kurzgefaßt; Spektrum Akademischer Verlag Heidelberg-Berlin, 1999.
- [Sc05] Schneider M.: A Strategy to introduce Functional Data modeling at school informatics. In: [Mi05]
- [St04] Staatsministerium für Unterricht und Kultus des Landes Bayern (Hrsg.): Lehrplan Informatik für die 8. Jahrgangsstufe im G8; <http://isb.contentserv.net> (12.02.2005)
- [Vo03] Voß, S.: Objektorientierte Modellierung von Software zur Textgestaltung. In [Hu03]
- [Vo04] Voß, Siglinde: Informatik in der 6. Jahrgangsstufe. In: [Ra04]
- [Vo05] Voß, S.: Informatic Models in Vocational Training for Teaching Standard Software. In: [Mi05]