Universität Dortmund
Informatik und Gesellschaft
Prof. Dr.-Ing. Thomas Herrmann

Vages Modellieren

Thomas Herrmann, Kai-Uwe Loser

Bericht des Fachgebiets Informatik & Gesellschaft, Universität Dortmund

Mai 1998

1 Einleitung.

In der Informatik werden Modelle in vielfältiger Form verwendet. Unter Modellen sind Repräsentationen von Ausschnitten der Realität zu verstehen, die insbesondere zu einer Komplexitätsreduktion führen. Als grafisch repräsentierte Diagramme dienen sie speziell dem Zweck der Kommunikation zwischen unterschiedlichsten Gruppen: zwischen Softwareentwicklern und Anwendern, in letzter Zeit häufiger auch zwischen Organisatoren, Anwendungsentwicklern und Arbeitsausführenden. Die dafür zum Einsatz kommenden Beschreibungssprachen werden auch als Modellierungsnotationen bezeichnet. Für bestehende Modellierungsnotationen ist der Zweck der Modellierung meist die spätere Umsetzung in technische Systeme. Dadurch liegt ein Schwerpunkt bei der informatischen Modellierung auf einer Formalisierung des modellierten Gegenstandes. Für viele Anwendungsfälle von Modellnotationen ist das Ziel aber nicht ausschließlich in technischen Realisierungen zu sehen.

Der Gegenstand der Modellierung ist dort meist nicht einzig das technische System, vielmehr geht es um die Beschreibung der soziotechnischen Abläufe und Gegebenheiten. Soziotechnische Prozesse beinhalten dabei die wechselseitigen Abhängigkeiten, insbesondere abhängige Aktivitäten von mehreren Personen. Für die Abhängigkeiten sind soziale Aspekte wie organisatorische Strukturen, Kommunikationsstrukturen, Erwartungen und Interessen, Qualifikation usw. zu betrachten. Zusätzlich spielen in soziotechnischen Prozessen Artefakte eine Rolle, wie sie in der Informatik durch Computersysteme gegeben sind. Für die Darstellung von soziotechnischen Prozessen ist es notwendig auch Anteile mit geringerem Formalitätsgrad oder geringerer Determiniertheit, von fehlendem Wissen und unsicherer Erkenntnis etc. abbilden zu können. Zur Unterstützung der Kommunikation und des Erkenntnisprozesses über derartige

Zusammenhänge ist die Erweiterung um angemessene Beschreibungsmittel erforderlich. Der Modellierer soll ein Werkzeug an die Hand bekommen, mit dem er nicht nur das feststehende unerschütterliche (oder nur so wirkende) Wissen abbilden und mitteilen kann. Dazu müssen Attribute „unklar“, „fraglich“ oder „unbekannt“ ebenso darstellbar sein, wie „feststehend“ oder „abgeklärt“.

In diesem Artikel wird dargestellt, wie eine Notation aussehen kann, die Elemente verwendet, die auf übliche Modellierungskonzepte aufbauen und zusätzlich eine flexiblere Darstellung von Zusammenhängen erlauben.

In den Ausführungen zu den vorgeschlagenen Erweiterungen von grafischen Modellierungsmethoden beziehen wir uns auf eine am Fachgebiet Informatik & Gesellschaft entwickelte Methode, auf deren Haupt-Elemente wir im folgenden Abschnitt 2 zunächst kurz eingehen. Anschließend werden in Abschnitt 3 einige grundlegende Überlegungen zur Vagen Modellierung vorgestellt, für die dann in einem ersten Schritt grafische Modellierungselemente eingeführt werden. Anschließend wird in Abschnitt 5 darauf eingegangen, wie diese Elemente in die textuelle Notation für Annotationen integriert werden können. In einem letzten Schritt werden in Abschnitt 6 einige aus der Modellierungspraxis motivierte Spezialfälle mit besonderen Lösungen dargestellt.

2 Haupt-Elemente der diagramm-orientierten Modellierungsnotation

2.1 Haupt-Elemente

Um die für die angestrebten Zwecke sinnvollen und nötigen Modellierungselemente zu identifizieren sind eine Reihe unterschiedlichster Methoden betrachtet worden (Oberquelle 1987, Green & Benyon 1996, Beck et al. 1995, Dearden & Harrison 1997, Sebilotte 1992, Scheer 1991, Harel 1987, Yourdon 1989, Booch 1994). Die Analyse wurde durch die Betrachtung von Anwendungsfeldern vervollständigt (vgl. Crowe 1994, Luczak et al. 1996, Sutcliffe 1997). Stein (1997) enthält einen nützlichen Überblick und Vergleich objektorientierter Methoden und Rosemann (1996) gibt viele Hinweise zur Verbesserung von Modellierungsmethoden. Schließlich sind Konzepte betrachtet worden, die soziale Aspekte im Kontext von I&K-Technologie näher behandeln, z.B. Aktivitäts-Theorie (Kuuti 1992), Mensch-Maschine-Interaktionstheorie

(Gaines 1988), Koordinationstheorie (Malone 1990) und sowohl die Studien von Kraut et al. (1990) über informale Kommunikation als auch der Winograd-Suchman-Diskurs (CSCW-Journal). Dabei haben wir folgende Schwachstellen bestehender Modellierungsmethoden identifiziert:

- Die Möglichkeiten Vagheit zu explizieren sind beschränkt. Im Gegensatz wird Vollständigkeit und Präzisierung gefördert.
- Möglichkeiten zur Darstellung verschiedener Arten von Attributen sind eingeschränkt.
- Darstellung unterschiedlicher Perspektivenund Sichtweisen erfordern meist den Wechsel des Notationssystems.
- Meta-Aspekte sind nicht systematisch einbezogen.
- Soziale Interessen und das Rollen einnehmen können nicht abgebildet werden.
- Nicht vordefinierte, freie und willkürliche Entscheidungen werden nicht behandelt.

Auch die zur Zeit wohl bekannteste neue Modellierungsmethode – die Unified Modeling Language (UML, Rational 1997) – bietet kaum Unterstützung um die genannten Schwächen zu überwinden.

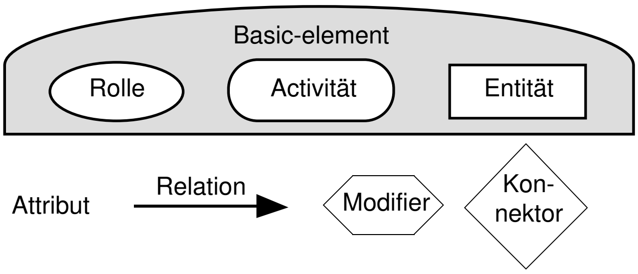

Abbildung 1: Hauptelemente der in den Beispielen verwendeten Modellierungsnotation

Die in diesem Artikel verwendete Modellierungsmethode basiert auf den grundlegenden Konzepten `Rolle`, `Aktivität`, `Entität` und `Relation`. Die zughörigen Elemente sind in Abbildung 1 dargestellt. Durch Rollen werden Mengen von Rechten und Pflichten dargestellt, die einer Person, Gruppe oder Organisationseinheit zugeordnet sind. Rollen repräsentieren „lebendige“ Systeme, deren Verhaltensmöglichkeiten durch Erwartungen geprägt sind, aber nicht definitiv festgelegt sind. Wenn Personen, die eine Rolle einnehmen, aktiv werden, so füh-

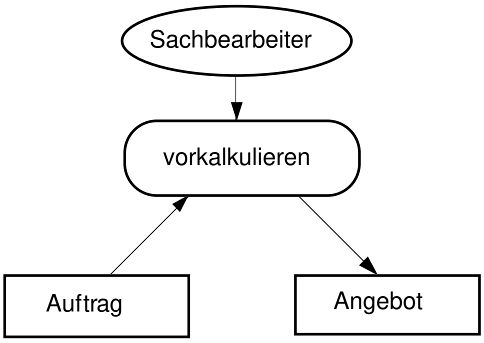

Abbildung 2: Einfaches Beispiel in der eingeführten Modellierungsnotation

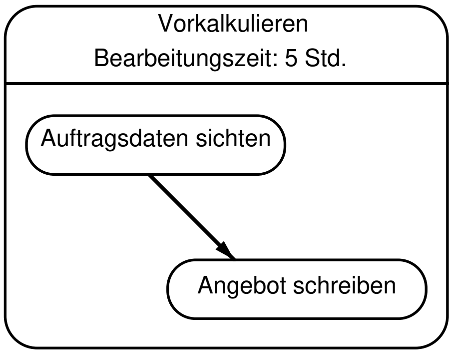

Abbildung 3: Verfeinerte Spezifikation der Aktivität „Vorkalkulieren" aus Abbildung 2.

ren sie Aktivitäten aus. Aktivitätenbenutzen Entitäten als Ressourcen (Dokumente, Werkzeuge, Rechnersysteme etc. sind Instanzen solcher Ressourcen) oder sie wirken auf Entitäten ein. Welche Rollen welche Aktivitäten ausführen und welche Aktivitäten welche Entitäten nutzen bzw. auf sie einwirken kann durch Relationen und deren Richtung ausgedrückt werden. Durch gerichtete Kanten kann man ferner ausdrücken, daß Aktivitäten eine Rolle beeinflussen, daß eine Entität einer Rolle „gehört" oder daß eine Entität die Eigenschaften einer Rolle beschreiben. Kanten werden ferner genutzt um ein Delegationsverhältnis zwischen Rollen auszudrücken, um die Abfolge von Aktivitäten darzustellen und Beziehungen zwischen Entitäten zu verdeutlichen. Abbildung 2 gibt ein einfaches Beispiel. Die Rolle des `Sachbearbeiters` ist (unter anderem) dafür verantwortlich, die Aktivität `vorkalkulieren` auszuführen. Bei dieser Aktivität wird ein Auftrag genutzt, um ein Angebot zu bearbeiten.

2.2 Detaillierung und Attributierung

Alle Basis-Elemente haben mindestens einen Namen. Möglichkeiten, Basis-Elemente näher zu beschreiben bestehen darin, Attribute hinzuzufügen oder sie mittels Sub-Elementen zu detaillieren, wie z.B. in Abbildung 3 für die Aktivität `Vorkalkulieren` durch das Attribut `Bearbeitungszeit` und die Aktivitäten `Auftragsdaten sichten` und `Angebot schreiben`. Durch Attribute werden sowohl Basis-Elemente als auch Modifier oder Relationen textuell näher beschrieben.

2.3 Konnektoren und Modifier

Konnektoren werden verwendet, um Relationen miteinander zu verknüpfen und in Beziehung zueinander zu setzen. In Abbildung 4 sind zwei alternative Abfolgen von Aktivitäten möglich. Das X in dem Konnektorsymbol (auf der Spitze stehendes Quadrat) repräsentiert eine XOR-Verknüpfung zwischen der Abfolge „`vorkalkulieren` -> ...“ und der Abfolge „`vorkalkulieren` -> `überprüfen` -> ...“.

Bestimmte Relationen eines Modelles sind häufig nur unter bestimmten Bedingungen gegeben (In Abbildung 1 z.B. `Auftragswert>10000`), die in unterschiedlichem Ausmaß vorherbestimmbar sind. Dies gilt insbesondere für die Abfolge von Aktivitäten, hier wird dann die Bedingung häufig in Form von Ereignissen spezifiziert, die gegeben sein müssen, damit eine Aktivität stattfindet. In der vorgeschlagenen Notation werden solche Ereignisse nur dann spezifiziert, wenn sie für Entscheidungen bei Verzweigungen relevant sind. Analoges gilt auch für Bedingungen für Relationen, die andere Basis-Elemente miteinander verknüpfen. Ein gedehntes Sechseck dient als Notationssymbol, in das solche Bedingungen eingetragen werden können. Dieses Symbol repräsentiert einen sogenannten Modifier. Modifier können weitere Informationen beinhalten, die die Instanziierung von Elementen modifizieren. Sie werden in Abschnitt 4 näher erläutert.

In dem Beispiel wird neben der Annotation von Bedingungen an Relationen auch die Zuordnung von Bedingungen zu Basis-Elementen verwendet: Die Abteilung (Rolle) `Qualitätssicherung` in Abbildung 4 kann nur unter der Bedingung existieren, daß der Betrieb mehr als 500 Mitarbeiter hat. Die Bedingung bezieht sich also nicht auf die Instantiierung von Relationen sondern auf die Instantiierung eines Basis-Elementes.

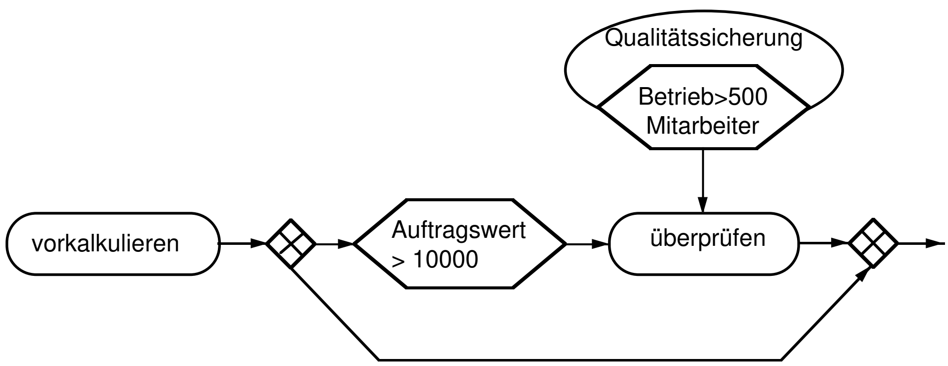

Abbildung 4: Beispiel für Aktivitätenfolgen mit Bedingungen

Diese insoweit sehr grob beschriebene Methode erlaubt es, unter Hinzunahme einiger weiterer, für Modellierungsmethoden typischer Eigenschaften, sozio-technische Prozesse darzustellen. Für den Zweck dieses Papieres wollen wir die beschriebenen Hauptelemente nutzen, um zu zeigen, welche Differenzierungen notwendig sind, um Vagheit auszudrücken.

3 Grafische Notation zur Darstellung von Vagheit

3.1 Basis-Elemente

Es ist einsichtig, daß die Aktivität `Vorkalkulieren` aus Abbildung 3 nicht unbedingt nur aus den Sub-Aktivitäten, `Auftragsdaten sichten` und `Angebot schreiben` besteht, ferner kann man es anzweifeln, ob sich der Modellierer mit dieser Art der Aufteilung überhaupt auf dem richtigen Weg befindet. Derartige Überlegungen sollen im folgenden durch entsprechende Symbole in Modellen darstellbar werden.

Um die verschiedenen Möglichkeiten zu identifizieren, die man zur Beschreibung von Vagheit benötigt, haben wir eine systematische Fallunterscheidung durchgeführt. Generell kann folgende Unterscheidung als Ausgangspunkt dienen:

a) **Bewußte Auslassung:** (Information Hiding) Der Modellierer entscheidet absichtlich, bestimmte Details in der aktuellen Darstellung eines Basis-Elementes nicht zu spezifizieren.
b) **Vagheit ausdrücken:** Dem Modellierer ist die mögliche Unangemessenheit oder Unvollständigkeit aufgefallen.
c) **Vollständigkeit ausdrücken:** Der Modellierer ist davon überzeugt, daß seine Spezifizierung eines Elementes korrekt und vollständig ist.
d) **Unbemerkte Auslassung:** Der Modellierer merkt nicht, daß seine Spezifizierung möglicherweise unangemessen oder unvollständig ist.

Im folgenden konzentrieren wir uns zunächst auf den Fall a) der bewußten Auslassung. Auch hier sind drei Möglichkeiten unterscheidbar, die zusammenfassend in Abbildung 5 mit den Darstellungselementen abgebildet sind:

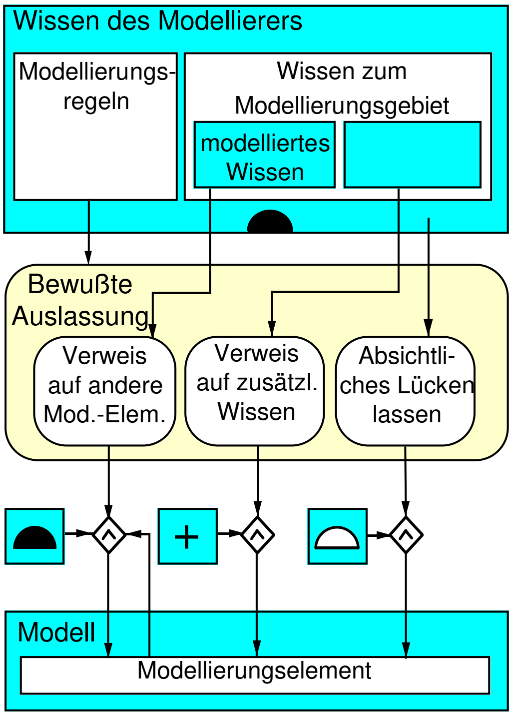

Abbildung 5: Fallunterscheidung zur bewußten Auslassung (Information Hiding)

a1) **Komplexitätsreduktion des Diagrammes:** Der Modellierer hat weiteren Aufwand investiert, um das Basis-Element zu spezifizieren. Er entschließt sich aber aufgrund seines Wissens um Regeln über die Erstellung geeigneter Modelle (`Modellierungsregeln`), dem Betrachter das Ergebnis seiner Bemühungen im aktuellen Kontext des Diagrammes nicht zu zeigen. Es wird aber eine schwarze Fläche (z.B. ◗ oder ♦) angeboten, die in einem Modellierungswerkzeug als anklickbare Schaltfläche verwendet werden kann (`Verweis auf andere Mod.-Elem.`). Wenn sie aktiviert wird, werden die Ergebnisse der Spezifizierungsbemühung des Modellierers gezeigt, die in einem Repository für das Modells bereitgehalten werden. Das können weitere Sub-Elemente oder Attribute sein, aber auch Indikatoren für Zweifel oder Unvollständigkeit, wie sie im Fall b) weiter unten dargestellt werden.

a2) **Komplexitätsreduktion des Modells mit Verweis auf Wissen:** Der Modellierer hat genug `Wissen zum Modellierungsgebiet`, um die Spezifizierung vorzunehmen, unterläßt dies aber, da er dieses Wissen nicht als unmittelbar relevant für den Modellierungszweck erachtet. Er möchte den Betrachtern aber signalisieren, daß man weiteres Wissen bei ihm abrufen kann. Für diesen `Verweis auf zusätzliches Wissen` kann z.B. ein „+“ Zeichen verwendet werden.

a3) **Komplexitätsreduktion durch Lücken lassen:** Während bei a1) und a2) Spezifizierungsarbeit im eigentlichen Sinne geleistet wurde, deren Ergebnis jedoch nicht (sofort) dargestellt wird, beinhaltet a3) zwei Bestimmungsfaktoren: Zum einen hat der Modellierer sehr wohl festgestellt, daß die Spezifizierung unvollständig ist (`Wissen um Nicht-Wissen`), zum anderen entscheidet er sich aber bewußt, daß es für den Zweck des Modells nicht sinnvoll ist, weitere Spezifizierungsarbeit zu leisten (`Wissen zum Modellierungsgebiet`). Die Entscheidung zum `absichtlichen Lücken lassen` kann im Diagramm durch eine leere Fläche dargestellt werden. Dazu wird zum Beispiel ein offener Bogen an

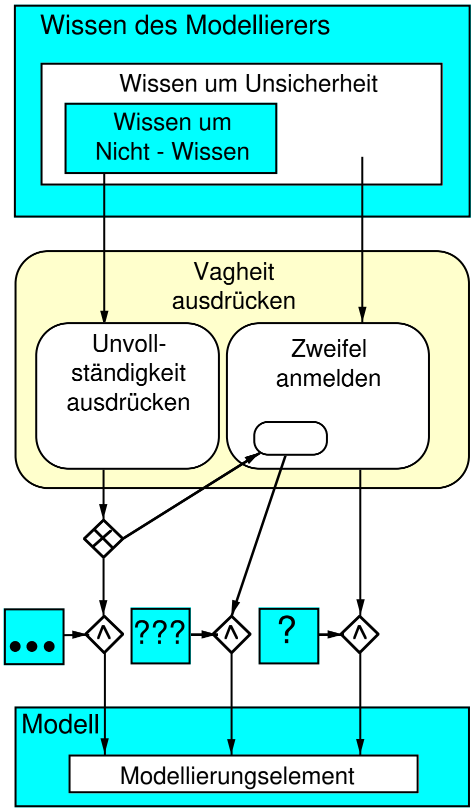

Abbildung 6: Fallunterscheidungzu Vagheit ausdrücken

Elementen dargestellt. Zur Vereinfachung der Darstellungen ist es diesbezüglich sinnvoll, erst dann den leeren Bogen zu verwenden, wenn eine Detaillierung in Form von Sub-Elementen teilweise angegeben wird. Sind keine weiteren Spezifizierungen, dargestellt, so ist die Aussage „Um weitere Details habe ich mich nicht gekümmert“ direkt zu erkennen, da eben keine vorhanden sind.

Für den Fall b) soll Vagheit ausgedrückt werden können, auf die der Modellierer aufmerksam geworden ist. Der Modellierer hat also Wissen bzgl. der Unsicherheit des zu modellierenden Sachverhaltes. Dies mag der Fall sein, weil ihm bei seinen Recherchen bestimmte Fragen nicht beantwortet wurden, weil es zu Widersprüchen kommt oder weil er feststellt, daß zur Spezifizierung bestimmter Details weitere Forschung notwendig wäre, die aber zum Zeitpunkt der Modellerstellung nicht geleistet werden kann. Auch hier lassen sich wiederum Unterfälle unterscheiden, die in Abbildung 6 nachzuvollziehen sind:

b1) **Unvollständigkeit:** Der Modellierer stellt fest, daß die Spezifizierung unvollständig ist bzw. daß er nicht in der Lage ist, sie zu vervollständigen. Um dies auszudrücken, benutzt er drei Punkte „**…**“.

b2) **Bezweiflung der Richtigkeit/Angemessenheit:** Der Modellierer bezweifelt die Richtigkeit einer Spezifizierung oder auch die Angemessenheit der Strukturierung einer Spezifizierung. Dies wird durch ein Fragezeichen ausgedrückt. In Abbildung 7 wird z.B. bezweifelt ob `Auftragsdaten sichten` überhaupt die richtige Bezeichnung ist und ob die Aktivität `Vorkalkulieren` angemessen zerlegt ist; außerdem wird durch den offenen Bogen ausgedrückt, daß man sich um die weitere Zerlegung von `Angebot schreiben` nicht kümmern möchte.

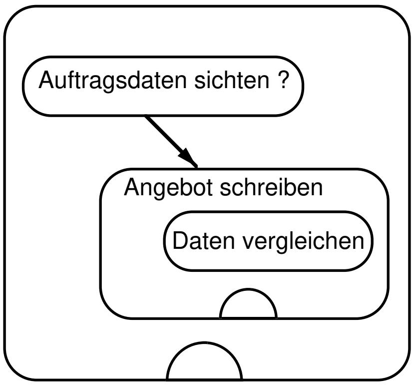

Abbildung 7: Detaillierung der Aktivität „vorkalkulieren"

b3) **Unvollständigkeit mit Zweifeln:** Der Zweifel kann auch auf die Feststellung einer Unvollständigkeit angewendet werden, wenn sich der Modellierer nicht sicher ist, ob seine Spezifizierung vollständig ist oder nicht. Dies kann der Fall sein, wenn ihm etwa unklar ist, ob ein weiteres Element hinzuzunehmen ist. Dieser Fall wird durch drei Fragezeichen angedeutet. Mnemotechnisch kann dies auch so interpretiert werden, daß jeder Punkt mit der Schleife des Fragezeichens versehen wird.

Wenn ein Element mit den beschriebenen Symbolen annotiert sind, so gilt diese Annotation auch für alle Details, die dieses näher spezifizieren. Durch eine abweichende Annotierung (s. unten, Fall c) können Ausnahmen gekennzeichnet werden.

In Abbildung 7 bezieht sich beispielsweise das „?", das die Korrektheit/Angemessenheit von `Vorkalkulieren` in Frage stellt, zunächst auf alle zugehörigen Details dieser Aktivität. Im Einzelnen sind damit die Attribute der Aktivität, die Zerlegung und die Details der weiteren Zerlegung gemeint.

Es ist sinnvoll, den Fall b2 einerseits mit den Fällen b1 und b3 andererseits additiv zu kombinieren, ohne daß sie direkt aufeinander bezogen sein müssen. So kann der Modellierer ausdrücken, daß eine von ihm gewählte Spezifizierung (eventuell) unvollständig ist und daß er sich ferner unsicher ist, ob er die Struktur der Spezifizierung angemessen festgelegt hat. Kombinationen werden durch Semikolon getrennt hintereinander notiert. Für die Kombination „**?;...**" kann „**.?.**" als Abkürzung verwendet werden.

Ferner sind die Kombinationsmöglichkeiten zwischen a) und b) zu beachten. Mit Hinblick auf a1) ist zu betonen, daß das Aktivieren der Schaltfläche nicht nur zusätzliche Spezifikationsdetails zeigen kann, sondern stattdessen (oder auch zusätzlich) Symbole des Zweifelns oder der Unvollständigkeit. Der Verweis auf zusätzliches Wissen des Modellierers gemäß a2) kann ebenfalls mit allen Unterfällen von b) kombiniert werden, um auszudrücken, daß der Modellierer Auskunft zu den Gründen des angemerkten Zweifelns oder der angemerkten Unvollständigkeit geben kann. Die entsprechenden abkürzenden Symbole können wie folgt aussehen: „**?+**", „**.+.**", „**?+?**" oder die zu kombinierenden Symbole werden durch Semikolon getrennt hintereinander geschrieben. Zuletzt ist bzgl. a3) festzuhalten, daß die nicht geleistete Spezifizierungsarbeit auch die Erkenntnis eines Zweifels oder einer Unvollständigkeit zum Ergebnis haben könnte.

Die bisherigen Beispiele zeigen die Verwendung dieser Symbole mit Hinblick auf grafisch repräsentierte Basis-Elemente. In Abschnitt 5 werden die Überlegungen für die Spezifikation mit textuellen Notationen am Beispiel von Attributen erweitert.

Nach der Betrachtung der Fälle a) und b) stellt sich die Frage, wie die Situation zu interpretieren ist, wenn wie in Abbildung 2 keine Symbole vorhanden sind. Es kann dann sowohl Fall c) als auch Fall d) gegeben sein. Wir meinen, daß man Fall c) nur unterstellen darf, wenn eine bewußte Entscheidung des Modellierers gegeben ist, gemäß derer er davon ausgeht, daß die von ihm angegebenen Spezifikationen korrekt und vollständig sind. Dies sollte durch einen Häkchen (✓) ausgedrückt werden. Mit einem „Abhaken“ wird Verantwortung übernommen. Wie auch alle anderen Symbole, so bezieht sich diese Aussage der Korrektheit und Vollständigkeit auf alle Details eines Elements, mit Ausnahme derer, für die explizit eine andere Aussage getroffen wurde. Beispielsweise wenn in Abbildung 7 das untere ? durch ein ✓ ersetzt ist, dann gelten alle Details als korrekt, mit Ausnahme der Wahl des Namens für `Auftragsdaten sichten`.

3.2 Relationen

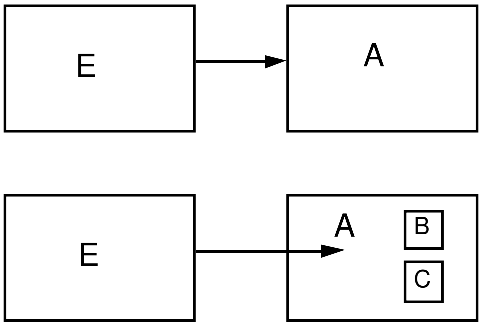

Abbildung 8: Mögliche Verwendung von Relationen

Mit Hinblick auf Vagheit bei Relationen ist zu beachten, daß sich die Vagheit auf drei Teile einer Relation beziehen kann. Einerseits kann sich Vagheit auf die Verbindung zwischen Relation und Basis-Elementen beziehen, also auf die Endpunkte der Relationen. Andererseits bezieht sich Vagheit auf die Relation als Ganzes. Für den ersten Fall können in der vorgeschlagenen Notation dazu Relationen in zweierlei Weisen mit einem Basis-Element verbunden sein kann.

Abbildung 8 zeigt am Beispiel von Entitäten, daß in einem Fall die Relation mit der Entität A als Ganzes verknüpft und im anderen Fall die Relation insofern auf A bezogen ist, als sie mit einigen oder einer ihrer Sub-Entitäten verknüpft ist. Auch weitere Sub-Sub-Entitäten kommen als Endpunkte der Relation in Frage. Diese unspezifische Aussage ist zur Modellierung semi-strukturierter Gegebenheiten von besonderem Interesse. Basis-Elemente, wie Rolle, Entität oder Aktivität können so über unspezifizierte Verknüpfungen verbunden werden. Die unterschiedliche Semantik kann an den Beispielen von Abbildung 9 erläutert werden:

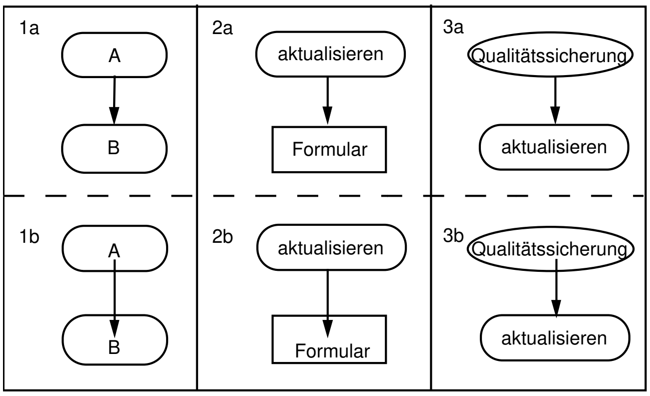

Abbildung 9: Beispiele unspezifischer Relationen

1a) Aktivität A muß vollständig beendet werden, bevor Aktivität B beginnen kann. Aktivität B wird als Ganzes abgearbeitet. Falls es Sub-Aktivitäten für B spezifiziert sind, die in einer vorgegebenen Abfolge mit einander verknüpft sind, so beginnt B mit der Start-Aktivität. Falls die möglichen Sub-Aktivitäten von B nicht miteinander verknüpft sind, werden sie alle parallel abgearbeitet.

1b) Aktivität A muß nicht vollständig beendet werden, sondern B **kann** nach der Beendigung einer Sub-Aktivität von A angestoßen werden; es ist aber nicht ausgeschlossen, daß A als ganzes abgearbeitet wird. B muß nicht als Ganzes ausgeführt, sondern nur ein Teil seiner Sub-Aktivitäten, egal ob sie miteinander verknüpft sind oder nicht. Es ist allerdings auch möglich, daß B als ganzes abgearbeitet wird.

2a) Das Formular wird als ganzes aktualisiert. Das bedeutet, daß jedes Formularfeld auf seine Aktualität hin überprüft werden muß. Das Formular gilt nach Abschluß der Aktualisierung als insgesamt aktualisiert, auch dann, wenn nur in einigen Feldern Veränderungen vorgenommen wurden.

2b) Es ist möglich, daß nur ein (festgelegter) Teil der Felder geprüft und überarbeitet wird. Es darf nicht unterstellt werden, daß das Formular nach Abschluß der Aktivität `Aktualisieren` insgesamt aktuell ist.

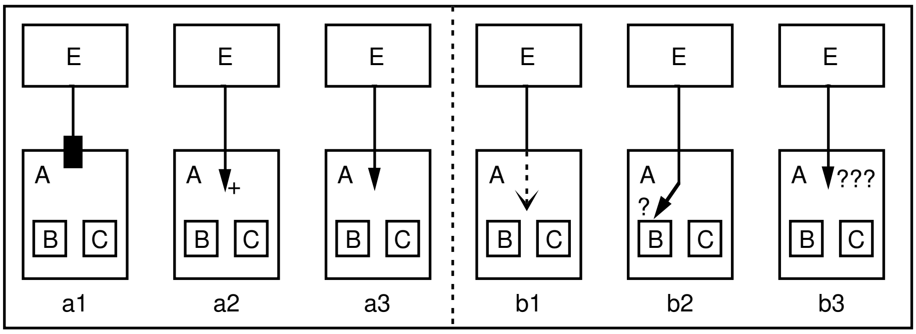

Abbildung 10: Symbole für Vagheit bei Relationen

3a) Die Qualitätssicherung ist insgesamt für die Aktualisierung verantwortlich, auch wenn nur eine ihrer Sub-Abteilungen oder nur ein einzelner Mitarbeiter die Aktualisierung letztlich durchführt.

3b) Es handelt sich um eine Aufgabe, die von einer Untereinheit der Qualitätssicherung autonom und in eigener Verantwortung durchgeführt werden kann; u.U. kann auch die Qualitätssicherung als Ganzes aktiv werden.

Die für Basis-Elemente beschriebenen Arten von Vagheit lassen sich nun ebenfalls auf Relationen anwenden, und zwar auf die Relationsendpunkte, wie auf die Relation als Ganzes. Abbildung 10 verdeutlicht die vage Darstellung der Verknüpfung von Relationen am Beispiel von Endpunkten an Entitäten:

a1) wenn man das verdickte Ende der Relation aktiviert erhält man Informationen, wie dieses Ende mit A, B, oder C verknüpft ist, dabei können auch Konnektoren verwendet werden.

a2) Im Modell wird die Art der Verknüpfung nicht erläutert, der Modellierer kann aber weitere Auskunft geben.

a3) Der Modellierer hält es nicht für zweckmäßig, die Verknüpfung mit A, B oder C genauer zu spezifizieren.

b1) Es wird ausgedrückt, daß das Wissen zur genauen Spezifizierung der Verknüpfung der Relation mit A, B oder C fehlt.

b2) Es wird bezweifelt, ob die Verknüpfung der Relation mit den Sub-Entitäten B korrekt ist oder ob nicht eine Verknüpfung mit A als Ganzem angemessener ist.

b3) Es ist unsicher, ob die Verknüpfung der Relation mit A weiter zu den Sub-Entitäten B und C verfeinert werden muß oder ob es nicht ausreicht, E mit A als Ganzes zu verbinden.

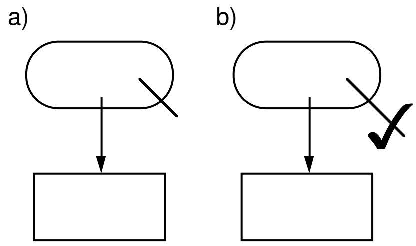

Abbildung 11: Bewußte Auslassung der Darstellung von Relationen (a) und Festlegung der Vollständigkeit dargestellter Relationen (b).

Durch zwei weitere Ergänzungen bezüglich Relationen können Aussagen über die Gesamtheit der Relationen getroffen werden, an der ein Basis-Element beteiligt ist. Falls ein Element, wie in Abbildung 11a) gezeigt, mit einem Relationensymbol versehen ist, das zwei offene Enden zeigt, dann wird damit ausgedrückt, daß es der Modellierer für möglich oder sinnvoll hält, daß dieses Element noch mit anderen Elementen des Modells verknüpft ist, diese Verknüpfungen aber nicht spezifizieren will. In der Fallunterscheidung wird hiermit eine Möglichkeit gegeben den Fall a) (Bewußte Auslassung) darzustellen. Falls, wie in Abbildung 11b) eine solche Relation mit offenen Enden mit einem Häckchen versehen ist, dann wird damit explizit ausgesagt, daß die Möglichkeit weiterer Verknüpfungen ausgeschlossen ist, was in der Fallunterscheidung dem Fall c) (Vollständigkeit ausdrücken) entspricht.

4 Notation von Bedingungen und Wahrscheinlichkeiten mit Modifiern

Bedingung werden in Modellen als Teil von Modifiern repräsentiert. Diese werden grafisch durch Sechsecke dargestellt, während sich im Bereich der Spezifizierung von Attributen hier spitze Klammern anbieten. Die Spezifizierung von Modifiern beinhaltet zwei Seiten. Auf der einen Seite kann man Ereignisse, Zustände oder logische Bedingungen angeben, die gegeben sein müssen. Auf der anderen Seite kann man beschreiben, mit welcher Unsicherheit (NC =

Quantitativ	Qualitativ		
0<=NC <= 1	**Häufigkeit**	**deontologisch**	**Beschreibungsart**
	• oft • häufig • regelmässig • gelegentlich • vorübergehend • ausnahmsweise • ???	• erlaubt • optional • wünschenswert • geboten • verboten • ???	• default • stellvertretend • pars-pro-toto • ???

Abbildung 12: Quantitative vs. qualitativeAussagen

Uncertainty) bzw. Wahrscheinlichkeit das Eintreten dieser Bedingung behaftet ist. Wie Abb. 13 zeigt, kann dies quantitativ durch die Wahl eines Wahrscheinlichkeitswertes zwischen 0 und 1 oder auch qualitativ geschehen, wobei man Häufigkeitsangaben, deontologische Spezifizierung oder auch Festlegungen auf der Ebene der Beschreibungsart treffen kann.

Die beiden unterschiedlichen Spezifizierungsteile sollten durch einen senkrechten Strich getrennt werden. Die Spezifizierung einer Bedingung kann selbst vage sein. Drei Punkte drükken aus, daß man nicht weiß, wie man die Bedingung spezifizieren soll. Ein Fragezeichen in einem Sechseck drückt Unsicherheit aus bzgl. der Frage, ob die Spezifizierung einer Bedingung möglich ist und damit sinnvoll ist. Drei Fragezeichen in einem Sechseck (oder in spitzen Klammern) besagen, daß es prinzipiell fragwürdig ist, ob das mit diesem Symbol annotierte Element im Modell erscheinen soll oder nicht — hier handelt es sich also um eine Art des Meta-Symbols. Ein leeres Sechseck drückt aus, daß man vermutet, daß das annotierte Element nur bedingt instantiiert wird, daß man sich aber um die Spezifizierung dieser Bedingung nicht kümmern will.

Mit Hinblick auf eine nachträglich Spezifizierung ist zu bemerken, daß auch die Symbole $\forall$ und $\neg$ möglich und sinnvoll sind. Das heißt dann im ersten Fall, daß die Bedingung immer erfüllt ist. Damit liegt im eigentlichen formalen Sinn keine bedingte Instantiierung vor. Analog könnte man bei der Bedingung $\neg$ auch ein Weglassen in Betracht ziehen, da es im formalen Sinne niemals instanziiert wird.

Beide Bedingungen sind an sich sehr hilfreich für den Zweck der vagen Modellierung, da man hiermit den Raum des Möglichen beschreiben kann, ohne diesen vollständig spezifizieren zu müssen. Ebenso können mit diesen Bedingungen Vermutungen dargestellt werden (z.B. „∀?“)

5 Textuelle Darstellung von Vagheit bei Attributen

Die in Abschitt 3.1 eingeführten Beispiele zeigen die Verwendung von Symbolen zur Darstellung von Vagheit mit Hinblick auf die grafisch repräsentierten Basis-Elemente. Die verwendete Symbolik läßt sich auch auf die Verwendung in textuellen Notationen erweitern. In der eingeführten Modellnotation werden beispielsweise Attribute mit einer textuellen Notation spezifiziert. Meist können Attribute in grafische Elemente eingetragen werden, wie es in Abbildung 3 für das Attribut `Bearbeitungszeit` dargestellt ist. Vagheit bei der Spezifikation von Attributen beinhaltet auch die vage Spezifikation von Werten. Das ist ein Thema, das bereits im Umfeld von Datenbanken genauer betrachtet worden ist (s. Biskup 1981 & 1982, Brinkmann 1995, Codd 1979 & 1986, Imielinski & Lipski 1981 & 1984 , Imielinski 1989, Lipski 1979).

Um die beschriebenen Arten von Vagheit auszudrücken werden Symbole verwendet, die zur grafischen Repräsentation analog sind. Als Regel läßt sich festhalten, daß die Vagheitssymbole hinter die Symbole geschrieben werden auf die sich die Vagheit bezieht. Als weitere spezielle Symbole sind der Doppelpunkt zu nennen, der den Attributbezeichner und den Attributtyp bzw. den Wert voneinander trennt. Elemente in Listen von Attributen werden durch Semikolon getrennt. Durch die Anordnung der Vagheitssymbole vor bzw. hinter die genannten Symbole wird der Bezug der Vagheit eindeutig.

Für einige Folgen können Abkürzungen verwendet werden, ohne dabei die Eindeutigkeit der Aussage zu verlieren. Beispielsweise bezieht sich „...“ auf den Hinweis auf Unvollständigkeit. Bei atomaren Attributtypen kann sich eine solche Aussage nur auf die Vollständigkeit der Liste beziehen, damit kann das nötige Semikolon auch wegfallen „Größe: 5...“ entspricht damit „Größe: 5;...“. Ferner trägt es zur Vermeidung von Mißverständnissen bei, wenn die Unsicherheitssymbole in Klammern gesetzt werden. Weitere Beispiele der Beschreibungen von Attributen finden sich in Tabelle 1, aus der auch mögliche Abkürzungen ersichtlich sind.

Tabelle 1: Beispiele für vage Beschreibung von Attributen

Fall	Ausdruck	Interpretation
a1)	Größe:♦	Wenn man die schwarze Fläche aktiviert, erhält man nähere Angabe zur Spezifizierung der Größe und/oder Symbole bzgl. Unvollständigkeit oder Zweifel angezeigt.
a2)	Größe:+	Es gibt hier keine näheren Angabe zur Spezifizierung von „Größe", aber der Modellierer weiß mehr dazu.
a3)	Größe:	Hinter dem Doppelpunkt befindet sich eine leere Fläche; dies drückt aus, daß sich der Modellierer mit einer Bestimmung des Wertes für das Attribut „Größe" oder um eine Festlegung von Sub-Attributen absichtlich nicht bemüht hat.
b1)	**ATTR**:(…)	Der Modellierer weiß nicht, wie er das Attribut bezeichnen soll.
b1)	Größe:(…)	Der Modellierer weiß nicht, wie er die „Größe" spezifizieren soll
b1)	Größe:5(…) oder Größe:5;(...)	Der Modellierer weiß nicht, ob neben „5" noch weitere Werte zur Spezifizierung der Größe in Betracht kommen.
b1)	Größe:[min:;(...)]	Der Modellierer weiß nicht, wie er neben „min" noch weitere Sub-Attribute von „Größe" spezifizieren soll.
b2)	**ATTR(**?)	Der Modellierer weiß nicht, ob ein Attribut im Kontext des Modelles der angemessen Typ eines Modellierungselementes ist.
b2)	Größe(?): 5	Der Modellierer zweifelt, ob „Größe" der angemessen Name des Attributes ist. Er weiß aber, daß dieses Attribut den Wert „5" hat.
b2)	Größe: 5(?)	Der Modellierer weiß nicht, ob „5" der angemessene Wert für die Größe ist.
b2)	Größe:[min:;max:](?)	Der Modellierer weiß nicht, ob er die Sub-Attribute von „Größe" angemessen spezifiziert hat.
b3)	Größe:[min:](???)	Der Modellierer weiß nicht, ob er neben „min" noch weitere Sub-Attribute von „Größe" spezifizieren soll; er ist sich bzgl. der vermuteten Unvollständigkeit unsicher
b1) &b2)	Größe: 5(.?.) oder Größe:5(?;...)	Der Modellierer weiß nicht, ob der Wert „5" richtig ist und ob daneben noch weitere Werte in Betracht kommen.

Tabelle 2 gibt einige Beispiele, wie sich Bedingungen kombiniert mit Unsicherheit bei Attributen ausdrücken lassen.

Tabelle 2: Beispiele für Vagheit in Bedingungen

Ausdruck	**Interpretation**
Temperatur< >:	Es existiert eine Bedingung unter der das Attribut Temperatur verwendet wird, die jedoch (bewußt) vom Modellierer offen gelassen wurde.
Temperatur <???>:	Der Modellierer bezweifelt, ob das Attribut Temperatur im Modell überhaupt benötigt wird.
Temperatur:5<…| regelmäßig>	Der Modellierer kann die Bedingung unter der das Attribut Temperatur den Wert „5“ besitzt nicht angeben, es ist aber bekannt, daß der Wert 5 regelmäßig auftritt (Wahrscheinlichkeit)
Temperatur:5<morgens>	Der Modellierer legt fest, daß das Attribut Temperatur unter der Bedingung „morgens“ den Wert „5“ besitzt.
Temperatur: 5 <…| geboten<morgens>>	Der Modellierer kann die Bedingung unter der das Attribut Temperatur den Wert „5“ besitzt nicht angeben. Die Wahrscheinlichkeit des Wertes „5“ ist jedoch „geboten“ und zwar unter der Bedingung „morgens“.
Temperatur:(...<vor-übergehend)	Das Attribut Größe besitzt einen festgelegten Wert, der jedoch nicht angebbar ist. Der Wert ist dabei immer vorübergehend bestimmt.
Temperatur:(<vor-übergehend)	Das Attribut Größe besitzt einen unspezifizierten Wert, was ein vorübergehender Zustand ist.

6 Spezialfälle und besondere Anwendungen

6.1 Unspezifizierte Verzweigung von Relationen

Im Bereich des Workflow-Managements werden Modelle verwendet, die im Falle einer Verzweigung stets unterstellen, daß es spezifizierbare Bedingungen gibt, die festlegen, wie die Verzweigung zu erfolgen hat. Abbildung 13a) gibt hierfür ein Beispiel, das einem Push-Prinzip entspricht: Mit Beendigung der Aktivität A muß klar sein, welche Aktivität als nächstes auszuführen ist (es wird ein XOR-Konnektor verwendet). Abbildung 13c) hingegen präsentiert eine

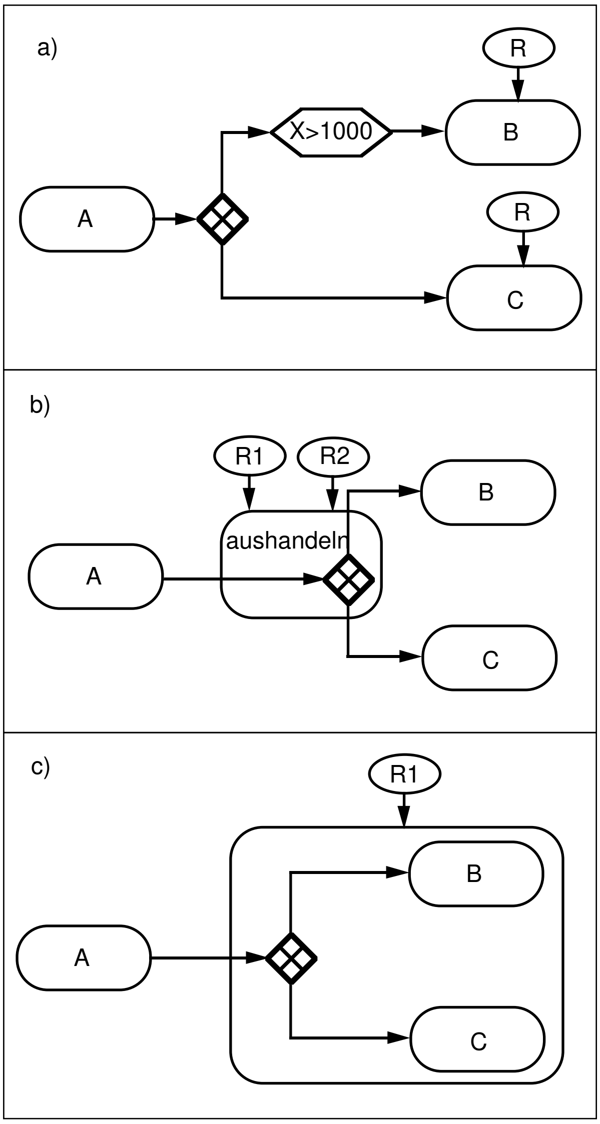

Abbildung 13: Determinierte Entscheidung vs. En tscheidung einer Rolle

Situation, in der die beteiligte Rolle entscheidet, welche Folgeaktivität stattfindet. Es sind hierfür keine Bedingungen spezifiziert. Falls mehrere Rollen beteiligt sind, muß die Entscheidung ausgehandelt werden – auch diese Aushandlung kann wie in Abbildung 13c) aber muß nicht immer spezifiziert werden. Auf diese Weise kann die für viele Situationen typische Abhängigkeit von willkürlichen menschlichen Entscheidungen repräsentiert werden.

Auf Basis der bisher gesammelten Möglichkeiten zur Spezifikation von Vagheit läßt sich nun noch einmal genauer angeben, was eine Relation mit unspezifiziertem Ausgangs- oder Endpunkt genau bedeutet. Abbildung 14 veranschaulicht, welcher komplexere Ausdruck durch die darüber gegebene Notationsform ausgedrückt wird.

Die mit Fragezeichen versehenen Bedingungen drücken aus, daß es unsicher ist, ob das Anfügen von Bedingungen angemessen ist. Wenn einige oder alle Bedingungen sich nicht sinnvoll spezifizieren lassen, dann wird die zu treffende Entscheidungsverzweigung durch eine beteiligte Rolle vorgenommen. Die unspezifizierte Anbindung von Relationen beinhaltet also stets beide Möglichkeiten der bedingungs- oder rollenabhängigen Entscheidung über die endgültige Art der Verknüpfung der Relation.

Die Bedeutung dieser Möglichkeit wird in Abbildung 15 am Stimulus-Response Prinzip erläutert. Abbildung 15a repräsentiert die vereinfachte klassische Sicht des Behaviourismus: Der gesamte Stimulus determiniert die Reaktion als Ganzes. Dieser Determinismus ist bei der Gestaltung technischer Systeme, die ein verantwortbares Leistungsverhalten zeigen sollen, oftmals

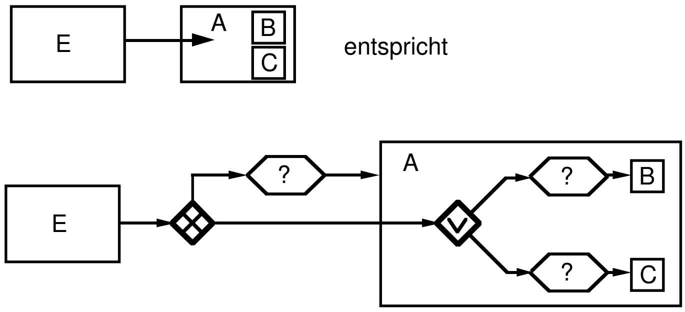

Abbildung 14: Alternative Darstellung unspezifizierter Relationen mittels vager Bedingungen und Konnektoren

erwünscht. Abbildung 15b spiegelt eher die Situation sozialer Systeme wieder, bei denen nur ein Teil der möglichen Reaktion durch einen Teil des Stimulus ausgelöst wird und es zusätzlich potentiell einen Entscheidungsraum für die betroffenen Rolle gibt hinsichtlich der Auswahl der Einflüsse und der Reaktionsmöglichkeit. Der Vorteil der Darstellung in Abbildung 15b besteht darin, daß einerseits die typische system-theoretischen Elemente, die eine Ursache-Wirkungs-Kette repräsentieren, noch erhalten bleiben und andererseits die für soziale Systeme typische Möglichkeit potentiell wahrnehmbarer Entscheidungsspielräume ebenfalls zum Zuge kommt. Abbildung 15c geht noch einen Schritt weiter, indem die Möglichkeit weiterer, nicht näher spezifizierbarer Einflüsse angedeutet wird.

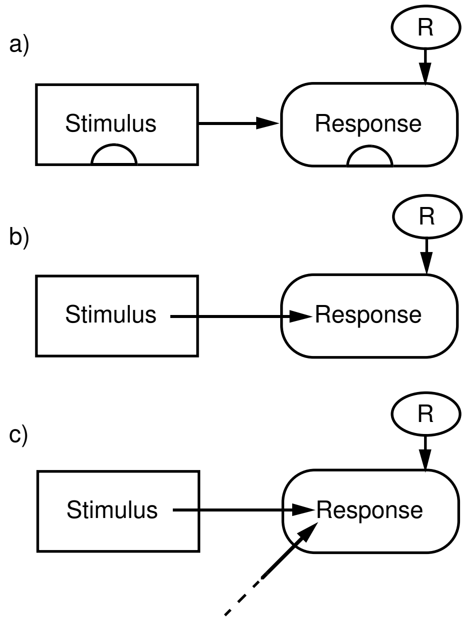

Abbildung 15: Stimulus-Response-Prinzip

6.2 Dekomposition unter mehreren Perspektiven

Für viele Anwendungsfälle kann nicht davon ausgegangen werden, daß es ein einziges Modell gibt, daß einen Sachverhalt adäquat abbildet. Vielfach ist es so, daß mehrere Alternativen gleichwertig nebeneinanderstehen können, oder daß es einen Selektionsprozeß gibt, in dem Vor- und Nachteile gegeneinander abgewogen werden. Für beide Fälle ist es sinnvoll Modellierungsalternativen auch in der Modellnotation zu unterstützen. Möglich ist es dazu alternative Detaillierungen für Elemente zuzulassen. Ein Beispiel dazu ist in Abbildung 16 mit zwei verschiedenen Sichten auf den Begriff der Multimedia Anwendung angegeben. In der Perspektive 1 wird der Begriff über die unterschiedlichen kombinierbaren Medientypen bestimmt, in der Perspektive 2 hingegen liegt das Augenmerk auf der Anwendung solcher Applikationen. Derartige Darstellungen werden benötigt, wenn die Elemente beider (oder mehrerer) Perspektiven untereinander oder mit anderen Elementen in Beziehung gesetzt werden sollen.

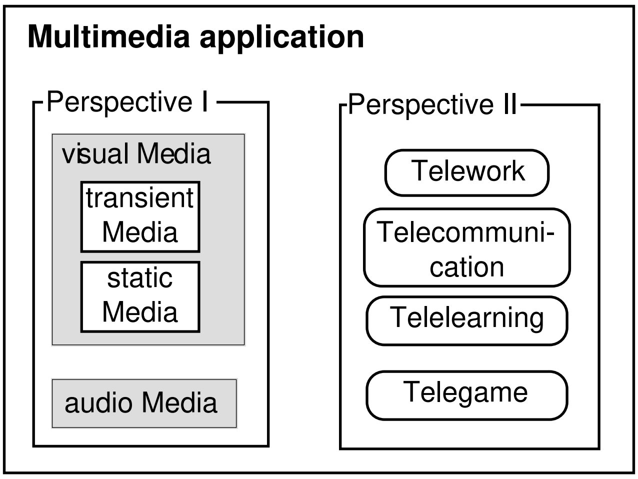

Abbildung 16: Mögliche Zerlegungen von „Multimedia Application“

6.3 Zirkuläre Phänomene

In Prozessen spielen häufig Rückkopplungen eine Rolle. Das heißt, daß die Reaktion selbst einen neuen Stimulus erzeugt. Dies läßt sich nun so verstehen, daß durch eine Aktivität einige Parameter verändert werden, die den Fortgang der Aktivität selbst verändern. Diese Situation soll durch Abbildung 17a) dargestellt werden. Typisch hierfür sind etwa Trainingsprogramme, die die Aufgaben, die sie einem Schüler präsentieren, in Abhängigkeit von dessen vorangehender Aufgabenbearbeitung auswählen. Zu diesem Zweck werden entsprechende Parameter des

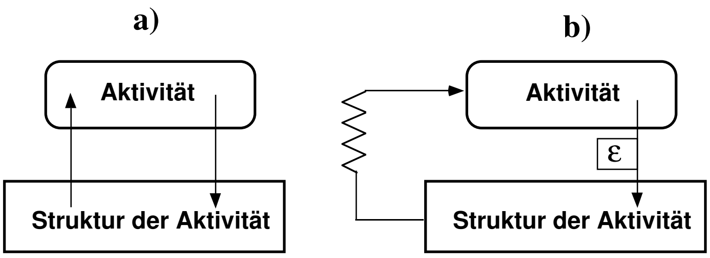

Abbildung 17: Rückkopplung: Selbst-Referentialität und Selbst-Adaption

Trainingsprogrammes nach jeder Aufgabenbearbeitung geändert. Lebendige Systeme sind demgegenüber in der Lage, im Rahmen von Lern- oder Adaptionsprozessen ihre gesamte Struktur zu verändern. Um dies auszudrücken, ist es zunächst notwendig, eine Entität einzuführen, die diese Strukturbeschreibung repräsentiert. Diese Repräsentation einer Strukturbeschreibung läßt sich durch eine Meta-Relation ausdrücken (Zick-Zack-Linie in Abbildung 17b). Diese Strukturbeschreibung — etwa einer Aktivität wie in Abbildung 17b) — kann nun durch diese Aktivität selbst fortentwickelt werden. Dabei geht es nicht nur um die Änderung von Parametern, sondern um eine evolutionäre Fortentwicklung (durch ε ausgedrückt). Dies beinhaltet, daß Attribute oder Sub-Elemente der Strukturbeschreibung gelöscht werden können oder zusätzlich neu eingeführt werden können.

Paetau (1996) beschreibt die Relevanz solcher Selbst-Referentialität für soziale Systeme. Auch andere Beispiele sind von paradigmatischen Stellenwert. Zum Beispiel die Tatsache, daß eine Problemlösung auch stets Einfluß auf die Problembeschreibung hat (s. Rittel & Webber 1984) (s. Abbildung 18.)

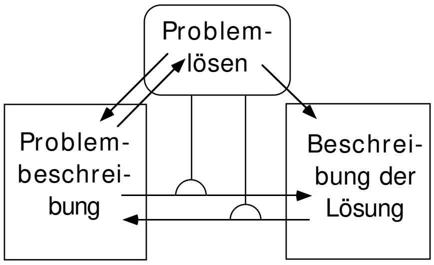

Abbildung 18: Einflüße zwischen Problembeschreibung, Problemlösung und Beschreibung der Problemlösung

Literatur

Beck, A.; Janssen, C.; Weisbecker, A.; Ziegler; J. (1995): Integrating object-oriented analysis and graphical user interface design. In: Software Engeneering. Springer-Verlag, May 1995. p. 15.

Biskup, J. (1981): A Formal Approach to Null Values in Database Relations. In: Gallaire, H.; Minker, J.; Nicolas, J.M. (eds.): Advances in Data Base Theory, Vol. 1. Plenum Press, pp. 299-341.

Biskup, J. (1981): Über Datenbankrelationen mit Nullwerten und Maybe-Tupeln. In: Schriften zur Informatik und Angewandten Mathematik, Bericht (Technical Report) 67, RWTH Aachen.

Biskup, J. (1982): Extending the relational algebra for relations with maybe tupels and existential and universal null values. In: Annales Societates Mathematicae Polonae, Series IV: Fundamenta Informaticae VII.1 (1984), pp. 129-150.

Booch, G. (1994): Object oriented analysis and design with applications. Redwood City, Calif. Benjamin/Cummings Publ. Co. 1994.

Brinkmann, Ralf (1995): Unsicherheit in relationalen Datenbanken. Uni Dortmund. Diplomarbeit.

Codd, E.F. (1979): Extending the database relational model to capture more meaning. In: ACM Transactions on Database Systems, Vol.4, No. 4, Dec. 1979, ppp. 397-434.

Codd, E.F. (1986): Missing Information (Applicable and Inapplicable) in Relational Databases. In: SIGMOD Record, Vol. 15, No. 4, pp. 53-78.

Crowe, M.K. (1994): Cooperative Work with Multimedia. New York et al.: Springer.

Dearden, A.M.; Harrison, M.D. (1997): Abstract models for HCI. In: International Journal of Human-Computer Studies,Vol. 46. pp. 151-177.

Deiters, Wolfgang ; Herrmann, Thomas; Löffeler, Thorsten; Striemer, Rüdiger (1996): Identifikation, Klassifikation und Unterstützung semi-strukturierter Prozesse in prozeßorientierten Telekooperationssystemen. In: Krcmar, H.; Lewe, H.; Schwabe, G. (Hrsg.) (1996): Herausforderungen der Telekooperation. DCSCW'96. Berlin u.a.: Springer. S. 261 - 274.

Fraunhofer Institut für Software- und Systemtechnik ISST (Hrsg.) (1996): Zur Systematisierung von Geschäftsprozeßcharakteristiken, Klassifikation und Erarbeitung von Lösungsmustern für semi-strukturierte Geschäftsprozesse. Dortmund: Fraunhofer ISST.

Gaines, B. (1988): A Conceptual Framework for Person-Computer Interaction in Complex-Systems. In: IEEE-Transactions on Systems, Man and Cybernetics, Vol. 18, No.4, July/August 1988.

Green, T.R.G.; Benyon, D. R. (1996): The skull beneath the skin: entity-relationship models of information artifacts. In: Int. J. Human-Computer Studies (1996) 44,. pp. 801-829.

Harel, D. (1987): Statecharts: A Visual Formalism for Complex Systems. In: Science of Computer Programming Vol. 8. pp. 231-274.

Hofstadter, Douglas R. (1992*2): Gödel, Escher, Bach. Ein endlos geflochtenes Band. Stuttgart: Klett Cotta im Deutschen Taschenbuch Verlag.

Imielinski, T. (1989): Incomplete Information in Logical Databases. In: IEEE Database Engineering Bulletin 12 (2), pp. 29-40.

Imielinski , T.; Lipski, W. (1981): On representing Incomplete Information in a relational Database. In Proceedings of the 7th Int. Conf. on Very Large Databases 1981, Computer Society Press. pp. 388-397.

Imielinski , T.; Lipski, W. (1984):Incomplete Information in Relational Databases. In: J. ACM, Vol. 31, No.4, Oct. 1984, pp. 761-791.

Klir, George; Folger, Tina (1988): Fuzzy Sets, Uncertainty and Information. Prentice Hall, Englewood Cliffs. 1988.

Kraut, R. E.; Fish, R. S.; Root, R. W.; Chalfonte, B. L. (1990): Informal Communication in Organizations: Form, Function, and Technology. In: Baecker (1993): Readings in Groupware and computer- supported Cooperative Work. Morgan Kaufman. pp. 145-199.

Kuuti, K. (1992): Identifying Potential CSCW Applications by Means of Activity Theory Concepts: A Case Example. In: Turner, J.; Kraut, R. (eds.) (1992): CSCW ´92. Sharing Perspectives. Proceedings of the Conference on Computer-Supported Cooperative Work. ACM. pp. 233-240.

Lipski, W. (1979): On Semantic Issues Connected with Incomplete Information Databases. In: ACM Transactions on Database Systems, Vol. 4, No. 3, pp. 262-296.

Luczak, H.; Wimmer, R.; Schumann, R. (1996): Rechnergestütztes Verfahren zur Modellierung und handlungsregulatorischen Bewertung von Arbeitstätigkeiten als Gestaltungshilfsmittel bei der Einführung von Gruppenarbeit. Teil I: Theorie und Konzeption des Verfahrens. In: Zeitschrift für Arbeitswissenschaft Vol. 50 (NF 22), Heft 2. S. 70-79.

Malone, T. W. (1990): What is Coordination Theory and How Can it help design cooperative work Systems? In: Proceedings of the Conference on Computer-Supported Cooperative Work, Oct. 1990, S. 357-370.

Oberquelle, H. (1987): Sprachkonzepte für benutzergerechte Systeme. Berlin: Springer, 1987.

Paetau, Michael (1996): Self-Organization of Social Systems - A new Challenge for Organization Sciences and Systems Design. In: SIGOIS Bulletin, Vol. 17, No.1 (April 1996). pp. 15-17.

Rittel, H.; Webber, M. (1984): Planning Problems are Wicked Problems. In: Cross, N. (ed.): Developments in Design Methodology. Chichester et. al., 1984. pp. 135-144.

Rational Software Corp. (Ed.) (1997): Unified Modeling Language. Documentation Set Version 1.0. 13. January 1997. Santa Clara, CA: Rational Software Cooperation.

Rosemann, M. (1996): Komplexitätsmangement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Wiesbaden: Gabler.

Scheer, A.-W. (1991): Architektur integrierte Informationssysteme. Grundlagen der Unternehmensmodellierung. Berlin: Springer, 1991.

Sebilotte, S. (1992): Task analysis and formalization according to mad: Hierarchical task analysis, method of data gathering and examples of task description. In: Institute National de Recherche en Informatique et Automatique Rocquencourt, BP 105. 78153. Le Chesnay Cedex. France. S. 1-30.

Simon, H. (1984): The structure of Ill-structured Problems. In: Cross, N. (ed.): Developments in Design Methodology. Chichester et. al., 1984. S. 145-166, 1984.

Stein, W. (1993): Objektorientierte Analysemethoden – ein Vergleich, Informatik Spektrum, 16.6, December 1993, S. 317-332.

Sutcliffe, A.G. 1997: Task related information analysis. To appear in International Journal of Human-Computer Studies, 96/037, Academic Press.

Ungeheuer, G. (1982): Vor-Urteile über Sprechen, Mitteilen, Verstehen. In: Ungeheuer, G.: Kommunkationstheoretische Schriften 1. Aachen, 1987, S. 229-338, 1982.

Yourdon, E. (1989): Modern structured analysis. Yourdon Press, Englewood Cliffs, NJ, 1989.

Zaniolo,C. (1984): Database Relations with Null Values. In: Journal of Computer Systems Science, Vol. 28, S. 142-166.