Sozio-orientierte und semi-strukturierte Modellierung mit SeeMe

Prof. Dr.-Ing. Thomas Herrmann, Dipl.-Inform. Marcel Hoffmann, Dipl.-Inform. Kai-Uwe Loser
Fachgebiet Informatik und Gesellschaft, Universität Dortmund[1]

Zusammenfassung

SeeMe ist eine neue Modellierungsmethode für die Darstellung sozialer und semi-sturkturierter Aspekte von Kommunikations- und Kooperationsbeziehungen. Durch eine Reihe neuartiger Konzepte, die in diesem Beitrag vorgestellt werden, unterstützt SeeMe die Aushandlung von Anforderungen und Diskussion von Gestaltungsalternativen mit Anwendern und Benutzern auf konzeptueller Ebene. Dabei wird besonderer Wert auf die Natürlichkeit der zur Verfügung gestellten Basiskonzepte, die Vermeidung hinderlicher Einschränkungen der Modellierungsfreiheit sowie die Darstellung vager Informationen gelegt.

1 Einleitung

Die Modellierungsmethode SeeMe unterstützt die Darstellung semi-strukturierter und sozialer Aspekte von Kommunikations- und Kooperationsprozessen. Aktuell ist sowohl auf akademischem Gebiet als auch in der Softwarebranche ein großes Interesse an Modellierungsmethoden zu beobachten. Dabei geht es u.a. darum bekannte Modellierungsansätze zu integrieren [Rational], Qualitäten von Modellierungsmethoden und Modellen zu definieren [Rose96] oder die Nutzung von Modellierungsmethoden bei der Anforderungsanalyse und der Softwareentwicklung besser zu organisieren und bessere Modellierungswerkzeuge bereitzustellen [Fisc*96].

Unter einer Modellierungsmethode verstehen wir ein Beschreibungsmittel, das Konzepte zur Darstellung von Phänomenen der Realität enthält. Zum Ausdruck der Konzepte wird eine Notation mit Zeichen oder Symbolen vorgesehen, deren Bedeutung und Kombinierbarkeit beispielsweise in Form eines Metamodells festgelegt wird. Dazu treten Regeln oder Konventionen ggf. sogar eine

[1] e-Mail: {herrmann, hoffmann, loser}@iug.cs.uni-dortmund.de
Address: University of Dortmund, FB Informatik, D-44221 Dortmund, Germany.
Phone: +49 (231) 7 55 - 20 57 Fax: +49 (231) 7 55 - 24 05

methodische Vorgehensweise zum Aufbau von Modellen. Zur Erzeugung, Veränderung und Betrachtung von Modellen dienen computergestützte Modellierungswerkzeuge (Editor). Ein Modell ist eine kommunizierbare Darstellung eines Ausschnitts der Realität, die zu einem bestimmten Zweck konstruiert wird [Herr97].

Wir unterscheiden zwei wesentliche Einsatzgebiete von Modellierungsmethoden (vgl. [PaWa97]). Zum einen dienen Modellierungsmethoden zur formalen Spezifikation bspw. bei der Definition von Schnittstellenprotokollen oder bei der Konfiguration einer Workflow-Engine. Andererseits dienen Modellierungsmethoden aber auch dazu, auf konzeptioneller Ebene Anforderungen an Software auszuhandeln und auszudrücken. In diesem Fall unterstützen Modelle die Diskussion und Begründung von Gestaltungsentscheidungen und fungieren als Kommunikationsmittel bei der Software-Entwicklung und Konfiguration. Hier spielt der Ausdruck „weicher" Faktoren eine wesentliche Rolle und es müssen begründende, vage, unvollständige, unsichere oder semi-strukturierte Informationen dargestellt werden können. Bei der Einführung von Workflow-Management-Systemen aber auch in vielen anderen Anwendungssituationen werden beide Arten von Notationen benötigt.

Viele bekannte Modellierungsmethoden dienten ursprünglich der Spezifikation von Software und sind nachträglich um zusätzliche Elemente erweitert worden, die ihre Ausdrucksmächtigkeit und Kommunizierbarkeit erhöhen sollten. Die meisten dieser Methoden unterstützen den Prozeß der Aushandlung von Anforderungen nicht ausreichend. Nach unserer Erfahrung werden z.B. bei der Erhebung und Sollkonzeption von Geschäftsprozessen Informationen vorgebracht, die mit herkömmlichen Methoden nicht direkt in Modelle eingearbeitet werden können. Beispiele dafür sind unvollständig vorgegebene Vorgehensweisen und vage beschriebene Bedingungen für das Eintreten von Ereignissen.

In diesem Beitrag stellen wir einen Modellierungsansatz vor, der ein besonderes Gewicht auf die Modellierung weicher Faktoren legt und die Aushandlung von Anforderungen sowie partizipative Software-Entwicklung unterstützt. Die SeeMe-Modellierungsmethode enthält eine Reihe speziell für diesen Zweck entwickelter Konzepte und Notationen, die im Hauptteil dieses Beitrags erläutert und anhand von Beispielen veranschaulicht werden. Einige dieser Konzepte können nach unserer Einschätzung vorhandene Modellierungsansätze erweitern (insbesondere Beugung von Relationen, Abschnitt 3.3.4 und Vages Modellieren, Abschnitt 3.5). Zuvor werden in Abschnitt 2 Hintergründe unserer Arbeit ausführlicher erläutert. Abschnitt 4 zeigt mögliche Erweiterungen der Methode auf und gibt einen Ausblick auf spezielle Anwendungen.

2 SeeMe Design Rationale

Der Entwicklung der SeeMe-Methode sind Untersuchungen verschiedener Modellierungsmethoden voraus gegangen ([DeHa97], [GrBe96], [Hare87], [Kung93], [Ober87], [Sche98], [Sebi92], [Rational] etc.). Dabei wurde u.a. geprüft, wie die Methoden für unserer Zwecke erweitert werden könnten. Folgende Schwachpunkte der Methoden haben wir dabei identifiziert (vgl. [Herr*97]).

- Die Möglichkeiten, vage Informationen zu kennzeichnen, sind begrenzt. Vollständigkeit wird erzwungen.
- Künstliche Konzepte, die zwar den Entwurf von Software unterstützen, jedoch Anwendern und Benutzern schwer vermittelbar sind.
- Mangelnde Modularisierung und Anpaßbarkeit der Methoden an unterschiedliche Anwendungsfälle.
- Hinderliche Einschränkungen der Modellierungfreiheit.
- Die Möglichkeiten, Attribute einzuführen und darzustellen, sind begrenzt.
- Wechsel der Sichtweise auf unterschiedliche Aspekte (z.B. Verhalten vs. Struktur) erfordert den Wechsel der Modellierungsnotation.
- Meta-Aspekte werden nicht systematisch erfaßt.
- Soziale Aspekte von Rollen und Rollenspiel können nicht dargestellt werden.
- Entscheidungsspielräume der beteiligten Rollen können nicht ausreichend dargestellt werden.

Die Qualität von Modellen hängt nicht von der Modellierungsmethode allein, sondern gleichzeitig auch von der Einhaltung von Modellierungsregeln und Konventionen und vom Geschick des Modellierers ab. Jedoch sind wir der Auffassung, daß eine Modellierungsmethode die Erzeugung verständlicher, übersichtlicher und deutlicher Modelle fördern kann. Wichtige Kriterien für die in diesem Beitrag vorgestellte Modellierungsmethode SeeMe sind intuitive Verständlichkeit, einfache Erklärbarkeit, Verwendung angemessener Medien und Darstellungsformen und Aussagemächtigkeit [Gogu96, LeLa96, Mood96].

Goguen spricht von der Notwendigkeit, bei der Anforderungsanalyse „nasse Information“ zu erfassen, die ohne Bezugnahme auf Kontext nicht verstanden werden können, weil sie vage und offen sind [Gogu94]. Viele bekannte Methoden stoßen hier an ihre Grenzen und erzwingen scheinbar vollständige und eindeutige Darstellungen. Demgegenüber enthält SeeMe Konzepte zur Kennzeichnung beabsichtigter Auslassung und unsicherer Information.

Ein Modellierer kann über die Notationssymbole seiner Methode nur in dem Maße verfügen, wie er sich der zugrunde liegenden Konzepte sicher ist. Künstliche Modellierungskonzepte, für die

Anwender der Methode keine Beispiele aus ihrer Erfahrungswelt finden, erschweren den Umgang mit einer Modellierungsmethode [Gogu94]. Die Basiskonzepte der SeeMe-Methode (siehe Abschnitt 3.1) differenzieren daher grundlegende Konzepte, die von den Adressaten der Methode geteilt werden, ohne durch ihre Anzahl ungeübte Modellierer zu überfordern.

Viele der von uns untersuchten Modellierungsmethoden unterscheiden nicht zwischen Grundelementen und Erweiterungen, so daß für ihre Anwendung, eine große Zahl von Konzepten und Notationssymbolen erlernt werden. Mit der Auswahl an Modellierungskonzepten und Notationssymbolen wächst die Schwierigkeit, eine Methode zu erlernen. Der Grundbaukasten von SeeMe enthält 3 Basiselemente (Abschnitt 3.1), 1 Relationssymbol mit 9 kontextabhängigen Bedeutungen (Abschnitt 3.3.1), Konnektoren und Bedingungen (Abschnitt 3.4) und kann bei Bedarf mit einer Menge von zusätzlichen Konzepten (Abschnitte 3.3.2 - 3.3.5 und 3.5) erweitert werden kann. Darüber hinaus sieht SeeMe Darstellungen zur Betonung von Sachverhalten und Abkürzungen für Modellkonstrukte vor, die sich als Angebot vor allem an erfahrene Modellierer richten.

Die Modellierungsmethode SeeMe erhöht die Erlernbarkeit, indem sie möglichst wenige syntaktische Einschränkungen bei der Kombination von Symbolen auferlegt. Die Bedeutung der Kombinationen kann intuitiv erschlossen werden oder mittels eines software-unterstützten Editors oder Viewers durch Einblendung von Text erläutert werden.

3 Sozio-orientierte und semi-strukturierte Konzepte von SeeMe

Wir unterschieden Konzepte und Notation als zwei Bestandteile der Modellierungsmethode SeeMe. Abb. 1 gibt eine Übersicht über die enthaltenen Konzepte und ihre Notation.

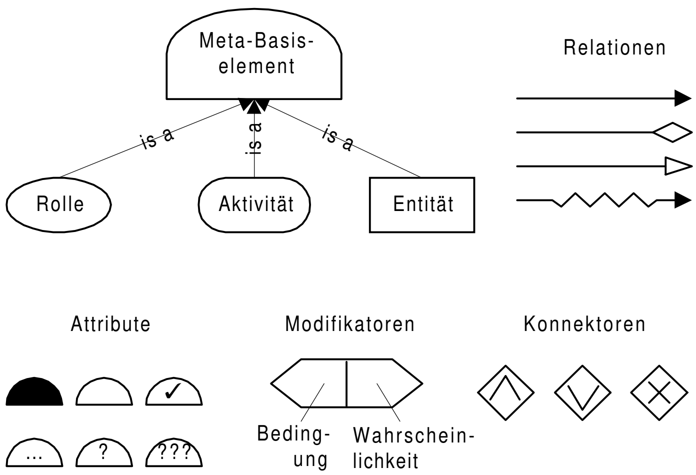

Abb. 1: Elemente der SeeMe Notation

3.1 Basiselemente: Rolle, Aktivität und Entität

SeeMe enthält drei Basiselemente: Rollen, Aktivitäten und Entitäten. Mit der Ausnahme „beabsichtigter Auslassung“ (vgl. Abschnitt 3.5.1) enthält jedes Basiselement als notwendige Attribute einen Namen oder eine Kurzbeschreibung. Basiselemente können mit Hilfe von Relationen (vgl. Abschnitt 3.3) zueinander in Beziehung gesetzt werden und durch Hinzunahme beliebiger weiterer Attribute und Einbettung aller Typen von Subelementen beschrieben werden.

Eine *Rolle* repräsentiert eine Menge von Rechten und Pflichten, die einer Person, einer Abteilung, einer Arbeitsgruppe oder einer anderen organisatorischen Einheit zugeordnet sind. Rechte und Pflichten ergeben sich aus Erwartungen anderer organisatorischer Einheiten an die Rolle. Eine Rolle kann eine Person repräsentieren genauso aber auch Funktionen einer Person. Dementsprechend kann eine Person oder eine Gruppe von Personen mehrere Rollen ausfüllen. Rollen können Aktivitäten ausführen um Entitäten zu verändern. Im Unterschied zu objektorientierten Ansätzen zieht SeeMe eine scharfe Grenze zwischen Personen und Software-Agenten oder Anwendungen, die als Träger von Aufgaben auftreten. Rollen können nicht manipuliert oder bearbeitet, sondern nur beeinflußt werden. Personen verfügen über einzigartige Eigenschaften, wie Interessen oder Ziele und können flexibel andere Rollen spielen.

Aktivitäten beschreiben Verhalten. Aktivitäten haben Start- und Endzeitpunkte und können zueinander zeitlich ins Verhältnis gesetzt werden, sie repräsentieren Operationen an Entitäten, Arbeitsaufgaben, Verrichtungen und Tätigkeiten, die von Rollen ausgeführt werden. Im Gegensatz zu Entitäten können Aktivitäten Veränderungen ihrer Umgebung hervorrufen, indem sie Entitäten manipulieren oder Rollen beeinflussen. Rollen können durch Aktivitäten Veränderungen hervorrufen.

Eine *Entität* ist ein passives Phänomen. Entitäten werden von Aktivitäten verwendet oder verändert. Entitäten sind Ressourcen zur Ausführung von Aktivitäten. Sie repräsentieren Dokumente, Dateien, Nachrichten, Wissen oder Information, die von einer Aktivität an eine andere gegeben werden, und Arbeitsmittel, die Rollen bei der Ausführung ihrer Aktivitäten unterstützen. Software, Hardware und auch Container, die andere Entitäten zeitweise oder bedingt beinhalten (z.B. ein Büroraum), werden als Entitäten aufgefaßt. Zwischen Entitäten bestehen Beziehungen.

Als grafische Repräsentation der Rolle wird ein Kreis oder eine Ellipse vorgeschlagen (siehe Abb. 1). Aktivitäten werden durch Rechtecke mit abgerundeten Ecken dargestellt und Entitäten durch Vierecke. Grundsätzlich können bei der Anwendung der Notation in einem bestimmten Kontext alle grafischen Symbole durch ikonische Darstellungen ersetzt werden. So können beispielsweise Rollen durch Fotos von Personen und Entitäten durch Abbilder der repräsentierten Dokumente oder Bildschirmmasken dargestellt werden. Somit können Symbole verwendet werden, die der Domäne des Adressaten der Modelle entstammen.

Komposition von Basiselementen

Basiselemente können andere Basiselemente enthalten. Zur Notation dieser Beziehung werden zwei Alternativen angeboten (siehe Abb. 5 in Abschnitt 3.3.4). Im Unterschied zu den meisten anderen Notationen sieht SeeMe vor, daß ein Basiselement auch andere Typen von Basiselementen als Subelemente enthalten kann. So kann eine Aktivität auch Sub-Entitäten oder Sub-Rollen enthalten. Auf diese Weise werden dann beispielsweise Entitäten und Rollen erfaßt, die nur für die Ausführung derjenigen Aktivität Bedeutung haben, in der sie eingebettet sind. Die Einbettung eines Basiselements in ein anderes kann oft als Aggregationsbeziehung interpretiert und modelliert werden. Für Relationen zwischen zusammengesetzten Basiselementen ist ein spezielles Konzept vorgesehen, das wir Beugung nennen (siehe Abschnitt 3.3.4).

3.2 Meta-Basiselement

Ein wesentliches Ziel bei der Entwicklung von SeeMe bestand darin, die Freiheit der Modellierer so wenig wie möglich einzuschränken. Um die vielfältigen Kombinationsmöglichkeiten auszudrücken, wird ein Meta-Basiselement eingeführt. Dieses Konzept vereinfacht die ersten Schritte der Modellierungsarbeit, indem Begriffe, Objekte und deren Beziehungen gesammelt werden können ohne direkt ihren Typ festlegen zu müssen. Als Notationssymbol für das Meta-Basiselement wird ein gedehnter Halbkreis verwendet. Abb. 2 zeigt einen Ausschnitt des SeeMe-Metamodells mit dem Meta-Basiselement und den Basiselementen in einer Vererbungshierarchie. Im SeeMe-Metamodell werden vorgegebene Attribute und grafische Repräsentationen eingetragen.

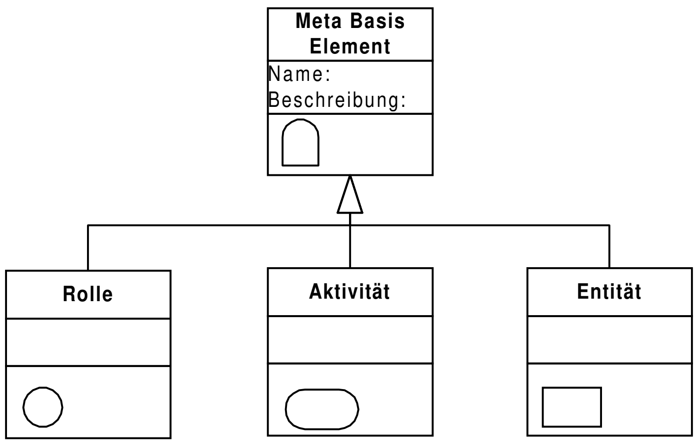

Abb. 2: Meta-Basiselement und Basiselemente von SeeMe (metamodel layer)

3.3 Relationen

Relationen werden durch gerichtete Kanten zwischen Basiselementen dargestellt, wobei mit der Richtung des Pfeilkopfes angegeben wird, welche der 9 vorgegebenen binären Relationen gegeben ist. Im Fall von Aggregation, Vererbung, Meta-Relation und zusätzlich eingeführter Relationen gibt der Pfeilkopf die Leserichtung der Relation an. Relationen können durch Konnektoren verknüpft werden, und es kann durch Modifikatoren bestimmt werden, unter welcher Bedingung und mit welcher Wahrscheinlichkeit eine Relation eintritt. Diese Möglichkeiten werden in Abschnitt 3.4 erläutert. Weiterhin bestehen Möglichkeiten, Relationen als unvollständig, ungenau oder unsicher zu kennzeichnen. Relationen können mit Namen oder Attributen versehen werden.

Im Unterschied zu anderen Modellierungsnotationen kennt SeeMe kein besonderes Konzept für Ereignisse, statt dessen wird das „in Kraft treten“ einer Relation als Ereignis aufgefaßt. Insofern modellieren Relationen die in einem Modell vorgesehenen Ereignisse, ohne daß dafür extra ein Symbol eingesetzt werden muß. Diese Modellierung entspricht der Auffassung von Ereignissen in Zustandsübergangsdiagrammen. Auf Ereignisse kann in Form der Annotation einer Bedingung Bezug genommen werden.

3.3.1 Binäre Relationen

vorgegebene Relation zwischen	Rolle	Aktivität	Entität
Rolle	a hat Erwartungen an → b	a führt aus → m	a wird beschrieben durch → x
Aktivität	m beeinflußt → a	m ist beendet wenn … startet → n	m verändert → x
Entität	x gehört → a	x wird benutzt von → m	x steht in Beziehung zu → y

Tabelle 1: Vorgegebene Bedeutung binärer Relationen

Eine binäre Relation setzt zwei Basiselemente zueinander in Beziehung. Dabei wird die Modellierungsfreiheit so wenig wie möglich eingeschränkt. Das bedeutet, daß jedes Basiselement zu jedem anderen Basiselement in Beziehung gesetzt werden kann und die verwendeten Relationen auch beliebige Namen und Bedeutungen tragen können. Es ergeben sich 9 Kombinationsmöglichkeiten für gerichtete Relationen. Tabelle 1 führt die vorgegebene Bedeutung und den Namen der Relationen auf, die gültig sind, wenn eine binäre Relation nicht explizit anders benannt wird.

3.3.2 Spezifizierte binäre Relationen

Binäre Relationen können durch ein weiteres Basiselement spezifiziert werden. Dabei sind drei Fälle zu unterscheiden, je nachdem ob eine Relation durch eine Rolle, eine Aktivität oder eine Entität spezifiziert wird.

Durch die Spezifikation einer Relation mit einer Rolle wird ausgedrückt, wer an der Relation interessiert ist bzw. vom Eintreten der Relation betroffen ist. Auf diese Weise können einige der wichtigsten sozialen Phänomene ausgedrückt werden: Interessen und Interessenskonflikte. Gerade bei der Reorganisation von Geschäftsprozessen oder bei der Einführung neuer Informationstechnik, wo Interessenkonflikte häufig auftreten und eine große Rolle für die Präferenz und Akzeptanz von neuen Lösungen spielen, sind Methoden von großem Nutzen, mit denen unterschiedliche und widersprüchliche Interessen transparent gemacht werden können. Die meisten zur Verfügung stehenden Methoden sehen die Modellierung von Interessen nicht vor, was die Auflösung von Konflikten und die Suche nach Kompromissen nicht gerade erleichtert.

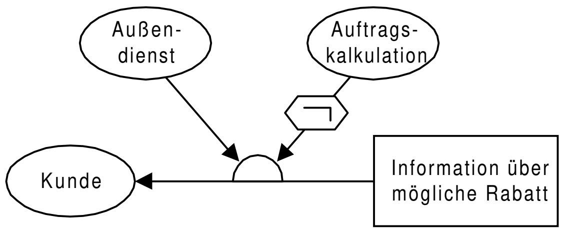

Abb. 3: Darstellung eines Interessen-konflikts bezüglich der Weitergabe von Informationen über Rabatte an Kunden

Mit SeeMe kann beispielsweise ausgedrückt werden, daß Außendienstmitarbeiter und Auftragskalkulation je unterschiedliche Interessen daran haben, daß Kunden über ihren Rabattrahmen informiert werden. Während der Außendienst hofft, daß durch die Weitergabe der Information Auftragsvolumen und Provision steigen, befürchtet der Innendienst weniger profitable Aufträge, wenn Kunden ihren Rabattrahmen voll ausschöpfen. Daher ist die Auftragskalkulation dagegen, daß Kunden Zugang zu Informationen über mögliche Rabatte erhalten. Ein Interesse daran, daß eine Relation nicht in Kraft tritt, wird durch ein Negationssymbol notiert (siehe Abb. 3). Genauso kann in einem Modell notiert werden, daß Mitarbeiter und Management je unterschiedlich Meinungen über die Erfassung, Speicherung und Weitergabe von Leistungsdaten in WMS (Monitoring) haben können. Durch Möglichkeit, Interesse am in Kraft treten von Relationen auszudrücken können die Modelle Aushandlungsprozesse unterstützen.

Mit der Spezifikation einer Relation durch eine Aktivität wird ausgedrückt, wie eine Relation realisiert wird. Die Spezifikation einer Relation durch ein Objekt erlaubt es, einer Relation zusätzliche Attribute zuzufügen oder ihre vorgegebene Bedeutung zu ändern. Auf diese Weise kann z.B. die Kardinalität zugeordnet werden, wenn man sie nicht in Form einer Abkürzung direkt annotiert.

3.3.3 Aggregation und Vererbung

Durch Aggregation wird in SeeMe ausgedrückt, daß ein Basiselement aus Subelementen zusammengesetzt ist. Aggregationsbeziehungen zwischen Basiselementen unterschiedlichen Typs sind ausdrücklich zugelassen.

Mittels Vererbung können Attribute, Relationen und Struktur eines Basiselements an andere Basiselemente weitergegeben werden. Vererbungsbeziehungen werden i.d.R. zwischen Basiselementen gleichen Typs definiert.

3.3.4 Beugung von Relationen

Zwischen zusammengesetzten Basiselementen können unterschiedliche Relationen bestehen. Ob eine Relation alle Unterelemente eines Basiselements mit einschließt oder nur einen Teil der Unterelemente betrifft, kann in der SeeMe-Notation unterschieden werden, indem Relationskanten am Rahmen eines Basiselements verankert werden oder den Rahmen schneiden und in die Fläche des Basiselements zeigen (siehe Abb. 4b, c und d). Als Beispiel zeigt Abb. 4 vier Relationen zwischen zwei Aktivitäten x und y, mit denen unterschiedliche Verhältnisse der Subelemente dargestellt werden können.

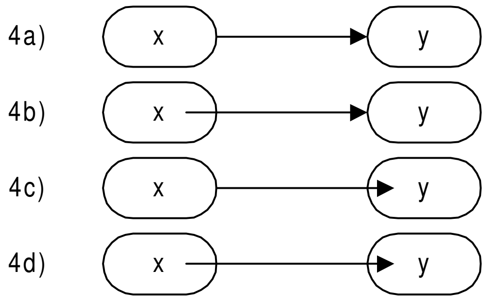

Abb. 4: Relationen zwischen zusammengesetzten Aktivitäten

- Nach Abb. 4a müssen alle Sub-Aktivitäten von x beendet sein, wenn mit der Ausführung der startenden Sub-Aktivität von y begonnen wird.
- Nach Abb. 4b kann mit der Ausführung der startenden Sub-Aktivität von y bereits begonnen werden, wenn ein erst Teil der Sub-Aktivitäten von x beendet ist.
- Nach Abb. 4c kann nach der Beendigung aller Sub-Aktivitäten von x eine beliebige Sub-Aktivität von y aufgenommen werden.
- Nach Abb. 4d ist dem Start eines Teils der Sub-Aktivitäten von y die Beendigung eines Teil der Sub-Aktivitäten von x vorausgesetzt.

Analog zu dem Beispiel aus Abb. 4 können auch alle anderen bisher eingeführten Relationen hinsichtlich ihrer Gültigkeit für Subelemente gebeugt werden. Als Beispiel zeigt Abb. 5 die vier Beugungen einer Aggregationsbeziehung zwischen zwei zusammengesetzten Entitäten. Dabei werden im unteren Teil der Abbildung zwei alternative Notationen verwendet: Einbettung und Modellierung der Aggregationsrelation durch Kanten. Mit Hilfe von Einbettung können in SeeMe Relationen zwischen Sub- und Oberelementen ausgedrückt werden. Einbettung muß nicht notwendig Aggregation bedeuten. Zu beachten ist, daß die Abbildungen 5b – 5d im unteren Teil jeweils nur eine von mehreren Möglichkeiten für die Bedeutung der gebeugten Relation im oberen Teil der Abbildung darstellen. Gebeugte Relationen beinhalten also Vagheit (vgl. Abschnitt 3.5)

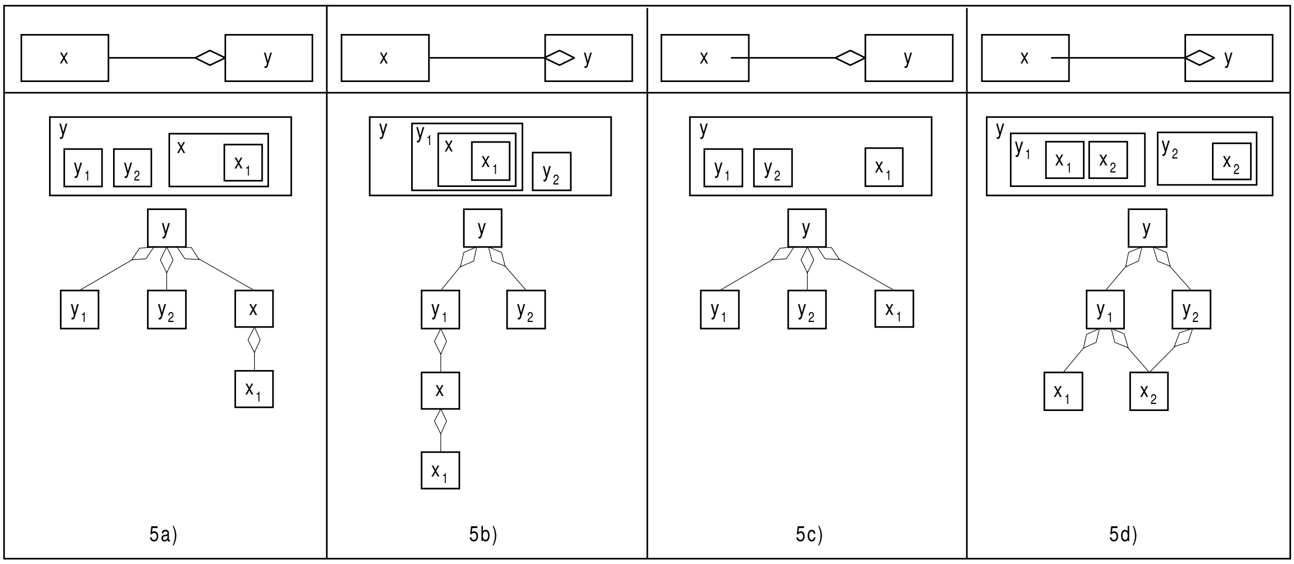

Abb. 5: Aggregation zusammengesetzter Entitäten

3.3.5 Meta-Relation

Die Meta-Relation ist ein nach unserer Kenntnis in anderen Modellierungsmethoden nicht bekanntes Konzept, mit dessen Hilfe der Einfluß eines Elements auf die interne Struktur eines anderen Elements dargestellt werden kann. Abb. 6 zeigt als Beispiel für die Anwendung der Meta-Relation einen Ausschnitt aus einem Geschäftsprozeßmodell. Hier wird mit Hilfe der Meta-Relation ausgedrückt, daß die Vorgehensweise bei der Kalkulation von Aufträgen von Rolle R1 bestimmt wird, und die Ausführung der Aktivität der Rolle R2 zugeordnet wurde.

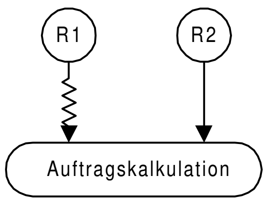

Abb. 6: Modellierung fremdbestimmter Vorgehensweisen mit Hilfe der Meta-Relation

3.4 Logische Verknüpfung und Bedingungen

Häufig treten Relationen nur unter bestimmten Bedingungen in Kraft. Das einfachste Beispiel dafür sind Kontrollflußbeziehungen zwischen Aktivitäten, die in vielen Notationen über Konnektoren und Bedingungen miteinander verknüpft werden können (siehe z.B. eEPK nach [Sche98]). Die in SeeMe enthaltenen Konzepte zur Darstellung von Bedingungen gelten im Unterschied zu anderen Modellierungsmethoden für alle Relationen und Basiselemente. Das heißt, daß beispielsweise auch die Zuordnung von Rollen zu Aktivitäten mit einem Konnektor oder Modifikator versehen werden kann. Genauso kann ausgedrückt werden, daß etwa eine Rolle (z.B. Betriebsrat in einem Unternehmen) nur bedingt existiert.

3.4.1 Konnektoren

Konnektoren werden verwendet, um Relationen logisch miteinander zu verknüpfen und in Beziehung zu setzen. Ähnlich wie bei ereignisgesteuerten Prozeßketten [Sche98] können Konnektoren miteinander verknüpft werden. Dadurch ist es möglich, komplexe logische Beziehungen zwischen Relationen auszudrücken. In Abb. 7 sind zwei alternative Abfolgen von Aktivitäten möglich und es ist eingezeichnet, daß die Aktivität „überprüfen" immer von einem Mitarbeiter der „Qualitätssicherung" und der „Kalkulationsabteilung" gemeinsam durchgeführt werden muß. Das X im Konnektorsymbol repräsentiert eine XOR-Verknüpfung zwischen der Abfolge „vorkalkulieren -> ..." und der Abfolge „vorkalkulieren -> überprüfen -> ...". Das Stattfinden der Überprüfung ist an eine Bedingung geknüpft, was durch die Annotation eines Modifikators (siehe Abschnitt 3.4.2) ausgedrückt wird. Als weitere Konnektoren sieht SeeMe die logischen Operationen UND sowie ODER vor. Werden zwei Relationen miteinander durch einen UND-Konnektor verknüpft, so bedeutet dies, daß beide Relationen gemeinsam in Kraft treten müssen. Bei der Verwendung eines XOR-Konnektors kann nur eine der Relationen gegeben sein. Durch die Verknüpfung mittels Konnektoren können mehrstellige Relationen und beliebige logische Verhältnisse zwischen Relationen hergestellt werden. Relationen können zusammengeführt werden und aufgeteilt werden, wie Abb. 7 zeigt.

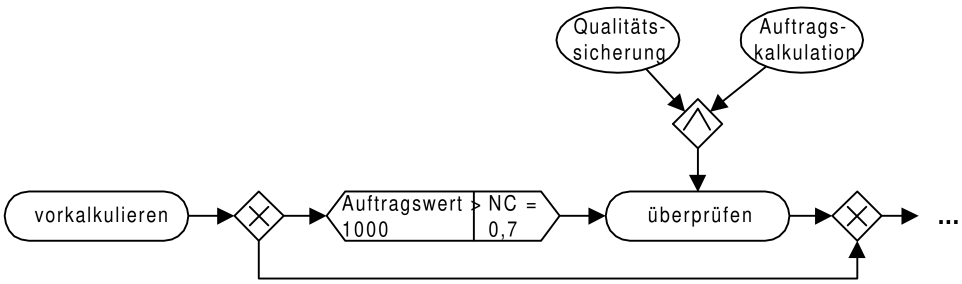

Abb. 7: Beispiel für Aktivitätenfolgen mit Bedingungen

3.4.2 Modifikatoren

Relationen eines Modells treten in der Praxis häufig nur unter bestimmten Bedingungen in Kraft. Dies gilt insbesondere für die Abfolge von Aktivitäten, hier wird die Bedingung häufig in Form von Ereignissen spezifiziert, die gegeben sein müssen, damit eine Aktivität stattfindet. Bedingungen werden in SeeMe als Modifikatoren aufgefaßt. Das SeeMe-Notationssymbol für Modifikatoren, ist das gedehnte Sechseck.

Im Gegensatz zu Ereignisprozeßketten (EPK) werden in SeeMe Relationen nur dann modifiziert, wenn der Modifikator zusätzliche Bedingungen für das in Kraft treten der Relation modelliert, die nicht schon in der Relation selbst enthalten sind. Auf diese Weise sparen wir u.a. die explizite

Modellierung von Ereignissen zwischen sequentiell aufeinander folgenden Aktivitäten, deren Start nur an die Vollendung der vorhergehenden Aktivität geknüpft ist.

Modifikatoren beinhalten zwei Seiten. Auf der einen Seite kann man Ereignisse, Zustände oder logische Bedingungen angeben, die gegeben sein müssen, damit die modifizierte Relation in Kraft tritt, beispielsweise damit auf die Aktivität „vorkalkulieren" die Aktivität „überprüfen" folgt. Auf der anderen Seite kann man beschreiben, mit welcher Unsicherheit (NC = Uncertainty) bzw. Wahrscheinlichkeit die Bedingung eintrifft. Die Wahrscheinlichkeit kann quantitativ durch die Wahl eines Wahrscheinlichkeitswertes zwischen 0 und 1 oder qualitativ durch eine Häufigkeitsangabe (ausnahmsweise, gelegentlich, oft, ...) oder eine deontologische Spezifizierung (erlaubt, wünschenswert, verboten, ...) ausgedrückt werden. Bei der Notation von Modifikatoren werden Bedingung und Wahrscheinlichkeit durch einen senkrechten Strich getrennt (siehe Modifikator in Abb. 7).

3.5 Vages Modellieren

SeeMe enthält zwei Konzepte zur vagen Modellierung, die im folgenden ausschnittsweise dargestellt werden:

1) *Beabsichtigte Auslassung* von Informationen, über die der Modellierer zwar verfügt, die er aber absichtlich im vorliegenden Modell nicht spezifizieren will.
2) *Kennzeichnung unsicherer Information*, wenn der Modellierer Zweifel über die Angemessenheit oder Vollständigkeit seiner Darstellung hegt.

Alle erwähnten Konzepte und Notationssymbole für die Darstellung von beabsichtigter Auslassung sowie die Kennzeichnung von unsicherer und vollständiger Information können in SeeMe auf komplette Modelle, Teile von Modellen, Basiselemente, Relationen, Konnektoren, Modifikatoren und Attribute angewendet und miteinander kombiniert werden. In diesem Beitrag erläutern wir die Anwendung auf Basiselemente in den Abschnitten 3.5.1 und 3.5.2).

3.5.1 Beabsichtigte Auslassung

Eine der schwierigsten Aufgaben beim Modellieren liegt darin, ein Modell auf das wesentliche zu konzentrieren und die Verständlichkeit, Übersichtlichkeit und Deutlichkeit des Modells und durch das Verstecken weniger wichtiger Darstellungen zu erhöhen [Herr97]. Dabei sind die Qualitäten Verständlichkeit, Übersichtlichkeit und Deutlichkeit immer im Hinblick auf den Zweck eines Modells zu messen [Hoff98]. Zwei wesentliche Möglichkeiten sind:

a1) Verweise auf ergänzende Informationen in anderen Modellen werden durch schwarze Flächen („⯊" oder „■") notiert. Bei einer werkzeugunterstützten Modellierung sind diesen Flächen Links zu anderen Modellen hinterlegt.

a2) Bewußt in Kauf genommene Modellierungslücken, deren Schließung nicht als notwendig erachtet wird, werden durch eine leere Fläche in Form eines Halbkreises notiert („⌓").

Als Komplement zu a2) enthält SeeMe noch ein weiteres Konzept, die *Kennzeichnung vollständiger Information*. Dieses Konzept dient dem Modellierer dazu auszudrücken, daß er von der Korrektheit und Vollständigkeit seiner Spezifikation überzeugt ist. Als Symbol schlagen wir ein Häkchen (✓) vor, das an Modelle, Teile von Modellen oder einzelne Basiselemente und Relationen notiert werden kann. Mit „Abhaken" wird Verantwortung übernommen. Als Beispiel für a2) zeigt Abb. 8, daß auf eine weitere Verfeinerung der Aktivität „Angebot schreiben" bewußt verzichtet wurde.

Der Verzicht auf weitere Detaillierung ist existentielle Grundlage des Modellierens. Jedoch ist es unter Umständen nicht sinnvoll, jede beabsichtigte Auslassung explizit zu kennzeichnen. Daher erlaubt es SeeMe als Abkürzung, den leeren Halbkreis wegzulassen, wenn ein Basiselement nur durch seinen Namen spezifiziert ist (vgl. Abb. 8„Auftragsdaten sichten" und „Daten vergleichen").

3.5.2 Kennzeichnung unsicherer Information

Mit der Kennzeichnung unsicherer Information soll dem Modellierer die Möglichkeit gegeben werden, auf Unvollkommenheiten seines Modells aufmerksam zu machen. Der Modellierer hat also Wissen bzgl. der Unsicherheit des zu modellierenden Sachverhaltes, z.B. weil ihm bei seinen Recherchen bestimmte Fragen nicht beantwortet wurden oder weil es zu Widersprüchen kommt. Wir unterscheiden drei Fälle:

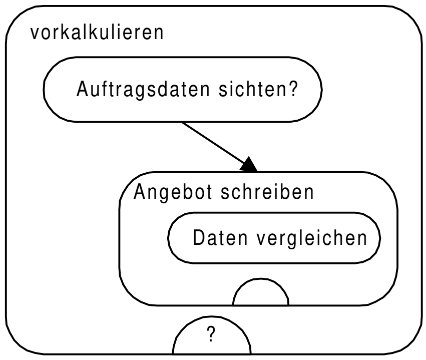

*Abb. 8: Beispiel für vage M*odellierung einer Aktivität

b1) Wenn der Modellierer feststellt, daß eine Spezifikation unvollständig ist und er nicht in der Lage ist, sie zu vervollständigen, kennzeichnet er das unvollständige Basiselement mit drei Punkten („**...**").

b2) Ist sich der Modellierer hinsichtlich der Richtigkeit einer Spezifikation unsicher, so gibt er diesem Zweifel durch die Annotation eines Fragezeichens Ausdruck („?"). Von dieser Möglichkeit kann u.a. Gebrauch gemacht werden, wenn Unklarheit über die Angemessenheit einer Benennung eines Elements oder über die gewählte Zerlegung in Subelemente besteht. So drückt Abb. 8 z.B. Unsicherheit bzgl. der Fragen aus, ob die Aktivität „Auftragsdaten sichten" richtig benannt wurde und ob die Aktivität „vorkalkulieren“ tatsächlich aus den beiden Aktivitäten „Auftragsdaten sichten“ und „Angebot schreiben“ besteht.

b3)Häufig kann sich der Modellierer nicht einmal sicher sein, ob seine Spezifikation unvollständig ist oder nicht. Dieser Fall wird durch drei Fragezeichen („???“) gekennzeichnet.

4 Erweiterungen und Ausblick

Die in den Kapiteln 3.1, 3.3.1 und 3.4 vorgestellten Konzepte: Basislemente, Relationen mit vorgegebener Bedeutung, Konnektoren und Modifikatoren stellen den Grundbaukasten der SeeMe-Methode dar. Die übrigen in diesem Beitrag erläuterten Konzepte haben sich bereits mehrfach als hilfreich erwiesen und werden daher als „Erweitungsset I“ betrachtet. Darüber hinaus sind für SeeMe noch eine Reihe weiterer spezieller Modellkonstrukte entwickelt worden, aus denen im Anwendungsfall eine den Anforderungen entsprechende Modellierungsnotation zusammengestellt wird. Diese Erweiterungen decken folgende Bereiche ab:

- Erzeugung und Löschung von Entitäten
- Mengen als Subelemente
- Zeitliche Ereignisse (time driven events)
- Unterbrechbarkeit von Aktivitäten
- Kooperation nach Pull-Prinzip
- Vergleich unterschiedlicher Zerlegungs- und Strukturierungsprinzipien
- Modellierung von Negationen (verbotene Pfade, Zugriffsbeschränkungen etc.)
- Rollenspiel: Eine Rolle in „schlüpft“ eine andere und übernimmt deren Rechte und Pflichten

Die SeeMe-Methode unterstützt die konzeptuelle Modellierung von Anforderungen und enthält eine Vielzahl von Konzepten, die auch andere Modellierungsmethoden erweitern könnten. SeeMe ist eine noch junge Modellierungsmethode. Aktuell arbeiten wir an der Entwicklung eines webbasierten Editors, der die kooperative Sammlung und Erzeugung von Modellen in Anwendungsprojekten und Lehrveranstaltungen unterstützten soll.

Literatur

[DeHa97] Dearden, A.M.; Harrison, M.D. (1997): Abstract models for HCI. In: International Journal of Human-Computer Studies,Vol. 46. S. 151-177.

[Fisc*96] Fischer, Gerhard; Lemke, Andreas C.; McCall, Raymond; Morch, Anders I. (1996): Making Argumentation Serve Design. In Moran, Thomas P.; Carroll, John M (eds.) (1996): Design Rationale. Concepts, Techniques , and Use. Mahwah, NJ: Lawrence Erlbaum. S. 267-293.

[Gogu94] Goguen, Joseph A. (1994): Requirements Engineering as the Reconcilation of Technical and Social Issues. In: Jirotka, Marina; Goguen, Joseph A. (eds.): Requirements Engineering: Social and Technical Issues. Academic Press. S. 165-200.

[Gogu96] Goguen, Joseph A. (1996): Formality and Informality in Requirements Engineering. In: Proceedings of the Fourth International Conference on Requirements Engineering. IEEE Computer Society Press. S. 102-108.

[GrBe96] 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 Vol. 44. S. 801-829.

[Hare87] Harel, David (1987): Statecharts: A Visual Formalism for Complex Systems. In: Science of Computer Programming Vol. 8. S. 231-274.

[Herr97] Herrmann, Thomas (1997): Communicable Models for Cooperative Processes. In: Salvendy, G. (ed.): HCI International '97. Proc. of the 7th International Conference on Human-Computer Interaction, San Francisco. Amsterdam: Elsevier . S. 185-188.

[Herr*97] Herrmann, Thomas; Hoffmann, Marcel; Loser, Kai-Uwe (1997): Modellierungsnotationen für prospektive, gestaltungsorientierte Technikfolgenforschung. In: Paul, H. (Hrsg.) (1997): Modellierung von Aufbau- und Ablauforganisation: von der Technozentrik zur Anthropozentrik. Projektbericht des Instituts Arbeit und Technik. 02-97. S. 33-46.

[Hoff98] Hoffmann, Marcel (1998): Mitarbeiter-orientierte Erhebung und Modellierung von Geschäftsprozessen bei der Einführung von Workflow-Management. Dortmund: Universität Dortmund. (Forschungsbericht des Fachbereichs Informatik an der Universität Dortmund Nr. 681).

[Kung93] Kung, David Chenho (1993): The Behavior Network Model for Conceptual Information Modelling. In: Information Systems Vol. 18, No. 1. S. 1-21.

[LeLa96] Lee, Jintae; Lai, Kum-Yeq (1996): What´s in Design Ratioale. In: Moran, Thomas P.; Carroll, John M (eds.) (1996): Design Rationale. Concepts, Techniques , and Use. Mahwah, NJ: Lawrence Erlbaum. S. 21-51.

[Mood96] Moody, Daniel (1996): Graphical Entity Relationship Models: Towards a More User Understandable Representation of Data. In: Thalheim, Bernhard (ed.) (1996): Conceptual Modeling, ER96. Proceedings of the 15th International Conference on Conceptual Modeling, Cottbus, Germany, October 1996. Berlin et al. Springer. S. 227-244.

[Ober87] Oberquelle, H. (1987): Sprachkonzepte für benutzergerechte Systeme. Berlin: Springer, 1987.

[PaWa97] Parsons, Jeffrey, Wand, Yair (1997): Using Objects for Systems Analysis. In Communications of the ACM Vol 40. No. 12. S. 104-100.

[Rational] Rational Software Corp. (1997): Unified Modeling Language. Version 1.1. September 1997. UML Summary, UML Notation Guide, UML Semantics. Santa Clara, CA: Rational Software Cooperation. (http://www.rational.com/uml/documentation.html).

[Rose96] Rosemann, Michael (1996): Komplexitätsmangement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Wiesbaden: Gabler.

[Sche98] Scheer, August-Wilhelm (1998): ARIS-Modellierungsmethoden, Metamodelle, Anwendungen. 3. Aufl. Berlin et al.: Springer.

[Sebi92] 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.