

# SeeMe -

## Semi-strukturierte, sozio-technische Modellierungsmethode

**SeeMe ist eine Modellierungsnotation.**

**Diese Notation ist Teil einer Methodologie, die benötigt wird, um sozio-technische Informationssysteme zu analysieren und zu gestalten.**

**Die folgende Präsentation gibt eine Einführung in die Modellierungsnotation SeeMe.**

**Mit ihrer Hilfe wird man in die Lage versetzt:**

- SeeMe-Diagramme verstehen zu können
- SeeMe-Diagramme anpassen zu können
- SeeMe-Diagramme selbst entwerfen zu können



Navigationserläuterung

Zur Gliederung



# Sozio-technische Informationssysteme

Sind Systeme, bei denen

- informationstechnische Systeme als Sub-Systeme in soziale Systeme integriert sind,
- die informationstechnischen Systeme mit anderen Sub-Systemen interagieren,
- die informationstechnischen Sub-Systeme von den sozialen Sub-Systemen des umgebenden Systems gesteuert und genutzt werden,
  - um deren Interaktion zu unterstützen und
  - um deren Kontinuität zu unterstützen.



Navigationserläuterung

# Beispiele für Sozio-technische Informationssysteme

Typische Beispiele für sozio-technische Informationssysteme sind:

- Das World Wide Web
- Workflow-Management-Systeme
- Organizational Memories als Teil von Wissensmanagement Konzepten
- Computer-basierte Lehr- und Lernumgebungen
- Groupware-Systeme

Navigationserläuterung

# Überblick über die SeeMe-Notation

**Dieser Überblick zeigt die gewählte Reihenfolge der einzelnen Themen dieser Einführung an. Klicken Sie auf ein gewünschtes Kapitel**

## Erläuterung zu dieser Einführung

1. Die Basis-Elemente
2. Relationen zwischen Basis-Elementen
3. Einbettung von Elementen
4. Spezifizierung von Relationen
5. Modifikatoren (Bedingungen & Ereignisse) und Konnektoren
6. Attribute
7. Vagheit
8. Ein- und Ausblenden
9. Allgemeine Aufgaben
10. Stichworte

**Aufgaben**

**Navigationserläuterung**

# Erläuterung der Navigationsmöglichkeiten I



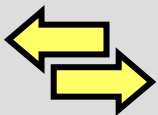
Sprung zurück in der Gliederung – auf die nächst höhere Ebene.



Sprung nach vorn in der Gliederung – auf die nächst höhere Ebene.



Sprung zum Beginn des nächsten (Unter-) Kapitels.



Diese Pfeile führen im Kurzdurchgang durch den SeeMe-Tutor vor oder zurück. Es werden nur Definitionen gezeigt. Wenn die Pfeile schwarz umrandet sind, gehört die Folie zur Kurzpräsentation.



Diese Pfeile führen durch eine Präsentation mittlere Länge. Es werden Definitionen und Beispiele gezeigt.

Pfeile sind schwarz umrandet, falls die Folie einen (Unter-) Kapitelbeginn darstellt



# Erläuterung der Navigationsmöglichkeiten II



Springt zur nächsten Seite und führt durch alle Seiten. Ansonsten schaltet die Präsentation nach 15 bis 30 Sekunden zur nächsten Folie. (kann unter Bildschirmpräsentation auf manuell umgestellt werden)



Springt zur vorangehenden Seite und führt durch alle Seiten.



Springt zu Gliederung am Anfang der SeeMe-Einführung



Springt zu den Stichworten am Ende der SeeMe-Einführung



Springt zur zuletzt gezeigten Folie zurück.




# Erläuterung der Navigationsmöglichkeiten III



**Text**

Pfeile bieten generell eine Unterstützung an, um an andere Stellen in der Präsentation zu springen. Sie enthalten entweder Text, der das Sprungziel beschreibt oder dieses wird aus dem Kontext deutlich



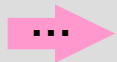
Weißer Flächen, die hinter Symbolen liegen, repräsentieren einen Link zu weiterführenden Informationen, die die unterlegten Symbole erläutern.

Text

Blauer, unterstrichener Text beinhaltet einen Link, mit dem man zu weiteren Informationen springen kann.



# Erläuterung der Navigationsmöglichkeiten IV



Dieses Symbol befindet sich in den Kapiteln, die sich mit der vollständigen Modellierung von Sachverhalten befasst. Mit ihm springt man vorwärts in das Kapitel zu Vagheit und sieht dort, wie man modelliert, wenn sich die Sachverhalte nicht vollständig darstellen lassen.



Dieses Symbol befindet sich in den Kapiteln, die sich mit der vagen Modellierung von Sachverhalten befasst. Mit ihm springt man zurück in das Kapitel, das zeigt, wie die vollständige Modellierung aussehen kann.



Springt vor und zurück auf einer Tour, die nur Aufgaben zeigt.





# Erläuterung zu dieser Einführung in SeeMe - Grundbegriffe

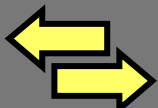
Mit SeeMe werden Klassen von Objekten modelliert. Klassen sind abstrakte Darstellungen, die sich nicht auf ein konkretes Phänomen beziehen, sondern auf eine Menge konkreter Phänomene mit gleichen Eigenschaften.

Sowohl die gezeigten Basis-Elemente als auch die Relationen zwischen ihnen werden als Klasse verstanden.

Man kann dann überlegen, welche konkreten Phänomen (auch Objekte genannt) zu einem aus Klassen bestehenden Modell zugeordnet werden können. Diese Zuordnung nennt man Instantiierung.

Bei der Instantiierung kann man zwei Perspektiven unterscheiden:

- 1) Wenn die zugeordneten Phänomene statisch gesehen werden, also keine Veränderungen im Zeitverlauf berücksichtigt werden, dann wird das Modell als Struktur interpretiert.
- 2) Wenn die zugeordneten Phänomene dynamisch gesehen werden, also zu verschiedenen Zeitpunkten andere Eigenschaften haben, dann wird das Modell als Prozess interpretiert.



# 1) Die Basis-Elemente

Wählen sie eines der  
Elemente aus oder  
lassen sie sich die  
Erläuterungen direkt in  
der vorgeschlagenen  
Reihenfolge anzeigen.

***Rolle***

***Aktivität***

***Entität***

Zum Überblick über die Definitionen

Navigationserläuterung

# 1.1 Die Rolle

## Rolle

**Rollen** werden von sozialen (Sub-)Systemen (Personen, Organisationseinheiten, Institutionen) gespielt und sind durch eine Menge von Rechten und Pflichten gekennzeichnet. Diese Rechte und Pflichten ergeben sich durch die Erwartungen von Rollen an eine Rolle. Die Erwartungen beziehen sich auf die Aufgabe, die eine Rolle innerhalb des übergeordneten sozialen Systems (z. B. der Gesellschaft) wahrnimmt. Soziale (Sub-)Systeme verhalten sich, sie führen **Aktivitäten** aus, sind dabei aber nicht von anderen Systemen steuerbar, sondern höchstens **beeinflussbar**. Technische Systeme (etwa Agenten) können in SeeMe niemals eine Rolle einnehmen.

Zum Überblick über die Definitionen



## 1.1 Die Rolle – Beispiel I



**BSI –  
Bundesamt  
für Sicherheit  
der  
Informationss-  
ysteme**

Das BSI ist eine politische Institution, die innerhalb des Staates eine bestimmte Rolle spielt, in dem sie das Recht hat, Regelungen zur Auswahl und zum Einsatz von Software (insbesondere kryptografische Systeme) vorzuschlagen, Software oder andere Organisationen zu zertifizieren etc. Demgegenüber stehen Pflichten, z.B. die Überwachung der zertifizierten Einrichtungen. Das BSI hat erheblichen Einfluss auf die Möglichkeiten des Verhaltens anderer sozio-technischer Informationssysteme.



# 1.1 Aufgaben zu Rolle

BSI –  
Bundesamt  
für Sicherheit  
der  
Informationen-  
systeme

1. Welche Rolle spielt Microsoft in Bezug auf welches übergeordnete System – was macht den Unterschied, was die Gemeinsamkeit mit dem BSI aus?
2. Nennen Sie fünf weitere Rollen auf der gleichen Stufe.
3. Nennen Sie Beispiele für soziale Phänomene, die man nicht als Rolle bezeichnen kann.
4. Nennen sie weitere Rollen, die bei der Beschreibung der nun folgenden drei Beispiele erwähnt werden.

Aufg.

## 1.1 Die Rolle – Beispiel II




**Rechen-  
zentrum**

Ein Rechenzentrum kann entweder eine eigenständige Rolle innerhalb des Wirtschaftssystems spielen oder es ist eine Organisationsabteilung innerhalb eines Unternehmens oder einer Institution. Die Pflichten bestehen in der Regel darin, das übergeordnete System mit Rechenleistung zu versorgen, Sicherheitsstandards zu gewährleisten, Beratung anzubieten etc. Die Rechte können darin bestehen, dass autonom entschieden werden kann, welche DV-Systeme eingesetzt werden oder ein Budget verwaltet werden kann.



## 1.1 Die Rolle – Beispiel III



**Program-  
mierer**

An Programmierer wird die Erwartung gerichtet, dass sie fehlerfreie, nachvollziehbare Software erzeugen, die den Bedingungen einer vorgegebenen Spezifikation genügen. Sie werden innerhalb einer Abteilung einer Firma oder innerhalb eines Teams tätig. Die Rolle des Programmierers wird von einem Menschen gespielt, der im Unternehmen noch weitere Rollen haben kann – z.B. Kollege zu sein, Auskunftsperson etc.



## 1.1 Die Rolle – Beispiel IV



**Herbert  
Muster-  
mann**

Auch einzelne Personen werden in SeeMe durch Rollen repräsentiert. Hauptsächlichste Erwartungen sind die Sicherung der Existenz, Bewahrung der Identität, Befriedigung von Bedürfnissen.

Erwartungen können also auch von einer Rolle an sich selbst gerichtet werden – sie trägt somit zu ihrer eigenen Konstitution bei – sie ist selbst-referentiell – dies ist auch für die vorangegangenen Beispiele der Fall. Die Identität einer Person wird u.a. durch die verschiedenen Rollen geprägt, die jemand in der Gesellschaft spielt. H. Mustermann kann Projektleiter, Familienvater und Stadtrat sein. Zwischen diesen Rollen kann es zu Konflikten kommen.





## 1.2 Die Aktivität

### **Aktivität**

**Aktivitäten** beschreiben Verhalten und rufen Änderungen in ihrer Umgebung hervor. Durch sie lässt sich die Dynamik eines Systems beschreiben. Aktivitäten können in eine Reihenfolge gebracht werden, sie repräsentieren dann Prozesse. Aktivitäten werden von **Rollen ausgeführt** oder von **Entitäten unterstützt**. Sie erzeugen oder **verändern** Entitäten oder **benutzen** sie.

Eine Aktivität ist durch ihren Verlauf und möglicherweise durch ein Ziel, auf die sie orientiert ist, gekennzeichnet.

Zum Überblick über die Definitionen



## 1.2 Die Aktivität – Beispiel I

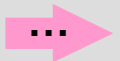
**planen**

**herstellen**

**spezifizieren**

**program-  
mieren**

Aktivitäten können Veränderungen hervorrufen, indem sie etwas erzeugen, das vorher nicht existierte. Es können immaterielle (planen) oder auch materielle Entitäten erzeugt werden.



## 1.2 Aufgaben zu Aktivität

planen

herstellen

spezifizieren

program-  
mieren

1. Berechnen vs. Berechnung, denken vs. Gedanken, Reset vs. neu starten, gesichert vs. Sichern – wo liegen die Unterschiede bei der Darstellung mit SeeMe?
2. Welche Aktivitäten können nicht von einzelnen Menschen, sondern nur von komplexeren Organisationseinheiten durchgeführt werden?
3. Nennen Sie Beispiele für Aktivitäten, die nur von technischen Systemen, (insbesondere Computern) ausgeführt werden können und solche, die nur von Rollen ausgeführt werden – welche Art von Aktivitäten verbinden Rollen mit den rein technischen Aktivitäten.
4. Gibt es Aktivitäten, die keine Spuren hinterlassen?

Aufg.



## 1.2 Die Aktivität– Beispiel II

**aktualisieren**

**löschen**

**korrigieren**

**sortieren**

Viele Arten von Aktivitäten nehmen Veränderungen an bestehenden Entitäten vor. Sie fügen ihnen neue Elemente hinzu, entfernen Elemente, sie verändern die Eigenschaften einer Entität oder sie verändern die Relationen, also den Zusammenhang zwischen ihren Elementen (z.B. sortieren)



## 1.2 Die Aktivität– Beispiel III

**beobachten**

**lesen**

**aussagen**

**bezeugen**

Es gibt Aktivitäten, die keine Wirkung auf Entitäten haben. Allerdings wirken sie auf Rollen. Sie beeinflussen sie. Es gibt Aktivitäten, die nur die ausführende Rolle beeinflussen (lesen, beobachten) und solche, die auch andere Rollen beeinflussen (aussagen, bezeugen, verurteilen)



## 1.2 Die Aktivität – Beispiel IV

**Nach-  
denken**

versus

**Backup  
durch-  
führen**

versus

**regnen**

Es gibt Aktivitäten, die nur von Rollen durchgeführt werden können (z.B. nachdenken, sich verbünden), während es von anderen Aktivitäten erwünscht ist, dass sie völlig automatisch erfolgen (Backup durchführen, Wecksignal auslösen). Letztere werden aber von irgendeiner Rolle verursacht, ausgelöst oder vorab festgelegt.

Daneben gibt es physikalische Abläufe im nicht-technischen Bereich (regnen, fließen) oder auch biologische Vorgänge (brüten), die sich auch als Aktivitäten darstellen lassen.



## 1.3 Die Entität

### ***Entität***

Eine *Entität* ist ein passives Phänomen. Entitäten repräsentieren die statischen Aspekte eines Systems. Entitäten werden von Aktivitäten verwendet und/oder verändert. Entitäten sind Ressourcen für Rollen und Aktivitäten.

Entitäten sind den Entities der Entity-Relationship Modelle ähnlich. Entitäten repräsentieren nicht konkrete Objekte, sondern eine Menge von Objekten mit gleichen Eigenschaften.

Soziale Systeme werden in SeeMe nie als Entitäten repräsentiert.

Zum Überblick über die Definitionen



## 1.3 Die Entität – Beispiel I

**Name**

**Adresse**

**Vertrag**

**Satz**

**Kapitel**

**Buch**

**Dokument**

**Ordner**

**Formular**

Entitäten können Klassen abstrakter Beschreibungen von Phänomen dieser Welt präsentieren, z.B. Name, Adresse, Vertrag, Gebrauchsanleitung.

Sie können auch Einheiten repräsentieren, mit Hilfe derer solche Beschreibungen strukturiert sind (Sätze, Datensätze, Kapitel, Bücher).

Ferner können sie gegenständliche Träger solcher Beschreibungen darstellen (Dokument, Formular, Buch, Vertrag)

Alle diese Entitäten können sowohl Bearbeitungsgegenstände als auch Unterstützung informationstechnischer Prozesse sein.





## 1.3 Aufgaben zu Entitäten

Name

Adresse

Vertrag

Satz

Kapitel

Buch

Dokument

Ordner

Formular

1. Unter welchen Bedingungen würden Sie Phänomene wie Erfahrung, Verantwortung oder Risiko als Entität darstellen? Was wären die Alternativen?
2. Wie viele Möglichkeiten gibt es, „Qualitätssicherung“ darzustellen – warum?
3. Unter welchen Bedingungen kann ein Mensch als Entität dargestellt werden?
4. Geben sie an, nach welchen Gesichtspunkten man die Entitäten im nächsten Beispiel unterteilen und systematisieren kann – warum ist es nicht möglich, eine hierarchische Klassifizierung sämtlicher Entitäten zu geben?

Aufg.



## 1.3 Die Entität – Beispiel II

Fluss

Strasse

Hammer

Verant-  
wortung

Risiko

Vorsatz

Lösungsidee

Größe

Informatik

Lebewesen

Gefühl

Entitäten repräsentieren auch Phänomene dieser Welt, deren **Abbildung** in informationstechnischen und sozio-technischen Prozessen von Bedeutung ist. Solche Phänomene können materieller (künstliche und natürliche) und geistiger Natur sein. Selbst Eigenschaften (wie „Größe“) lassen sich als Entitäten repräsentieren.



## 1.3 Die Entität – Beispiel III

**Maschine**

**Uhr**

**Auto**

**Computer**

**Datenbank**

**Quicksort-  
Algorithmus**

**SMTP**

**SQL**

Es werden auch solche Phänomene als Entitäten dargestellt, die einen dynamischen Aspekt haben, also Veränderungen hervorbringen können, aber auch gleichzeitig einen statischen Aspekt haben, also in einer statischen Form existieren und auch verändert werden können. Während der dynamische Aspekt als Aktivität präsentiert wird, **was als weitere Aufgabe für die hier dargestellten Entitäten versucht werden kann**, drückt die Entität die statische Existenz aus.



## 1.3 Die Entität – Beispiel IV

**Speicher**

**Lager**

**Kaufhaus**

**elektronischer  
Marktplatz**

**Landebahn**

**Büro**

Insbesondere werden auch solche Phänomene als Entitäten dargestellt, die andere Entitäten enthalten können (etwa Daten in einem Speicher) oder in denen sich andere Entitäten befinden oder die den Ort präsentieren, an dem andere Entitäten sein können oder Aktivitäten stattfinden oder sich Rollen aufhalten. Hierzu gehören auch Zeiträume.

**Werktag**



# 1.4 Die Definitionen der Basis-Elemente im Überblick

## Rolle

Rollen werden von sozialen (Sub-)Systemen (Personen, Organisationseinheiten, Institutionen) gespielt und sind durch eine Menge von Rechten und Pflichten gekennzeichnet. Diese Rechte und Pflichten ergeben sich durch die Erwartungen von Rollen an eine Rolle. Die Erwartungen beziehen sich auf die Aufgabe, die eine Rolle innerhalb des übergeordneten sozialen Systems (z. B. der Gesellschaft) wahrnimmt. Soziale (Sub-)Systeme verhalten sich, sie führen Aktivitäten aus, sind dabei aber nicht von anderen Systemen steuerbar, sondern höchstens beeinflussbar. Technische Systeme (etwa Agenten) können niemals eine Rolle einnehmen.

## Aktivität

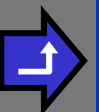
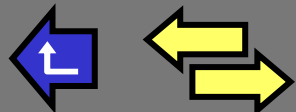
Aktivitäten beschreiben Verhalten und rufen Änderungen in ihrer Umgebung hervor. Durch sie lässt sich die Dynamik eines Systems beschreiben. Aktivitäten können in eine Reihenfolge gebracht werden, sie repräsentieren Prozesse. Aktivitäten werden von Rollen oder Entitäten ausgeführt. Sie erzeugen oder verändern Entitäten oder benutzen sie.

## Entität

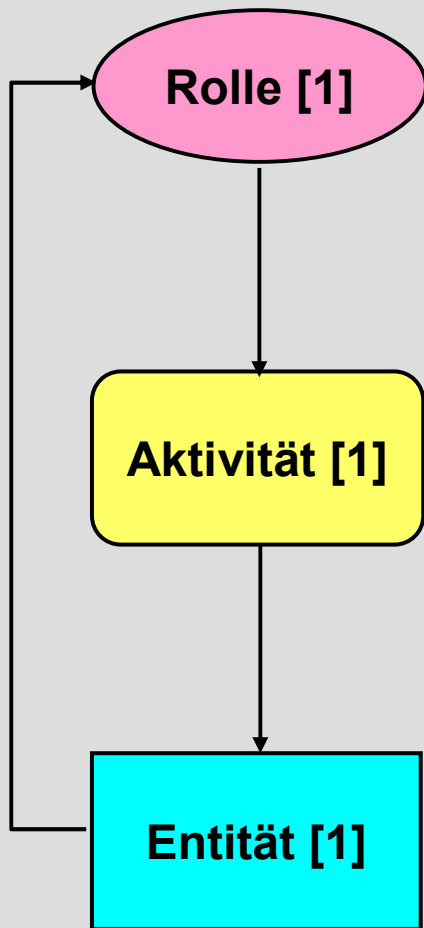
Eine *Entität* ist ein passives Phänomen. Entitäten repräsentieren die statischen Aspekte eines Systems. Entitäten werden von Aktivitäten verwendet und/oder verändert. Entitäten sind Ressourcen für Rollen und Aktivitäten.

Entitäten repräsentieren nicht konkrete Objekte, sondern als Klassen einer Menge von Objekten mit gleichen Eigenschaften.

Soziale Systeme werden in SeeMe nie als Entitäten repräsentiert.



## 2) Einfache Standard-Relationen zwischen Basis-Elementen



Relationen werden in SeeMe durch Pfeile dargestellt.

Wenn keine anderen Angabe gemacht sind, haben die Pfeile eine Standardbedeutung, die von den Elementen abhängt, die durch die Pfeile verbunden sind.

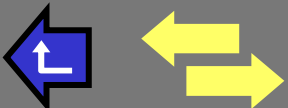
Es gibt neun einfache Standardrelationen, die in diesem Kapitel erläutert werden.

Eine Relation beginnt mit einem Ankerpunkt an ihrem „Start-Element“ und endet mit dem anderen an ihrem „End-Element“.

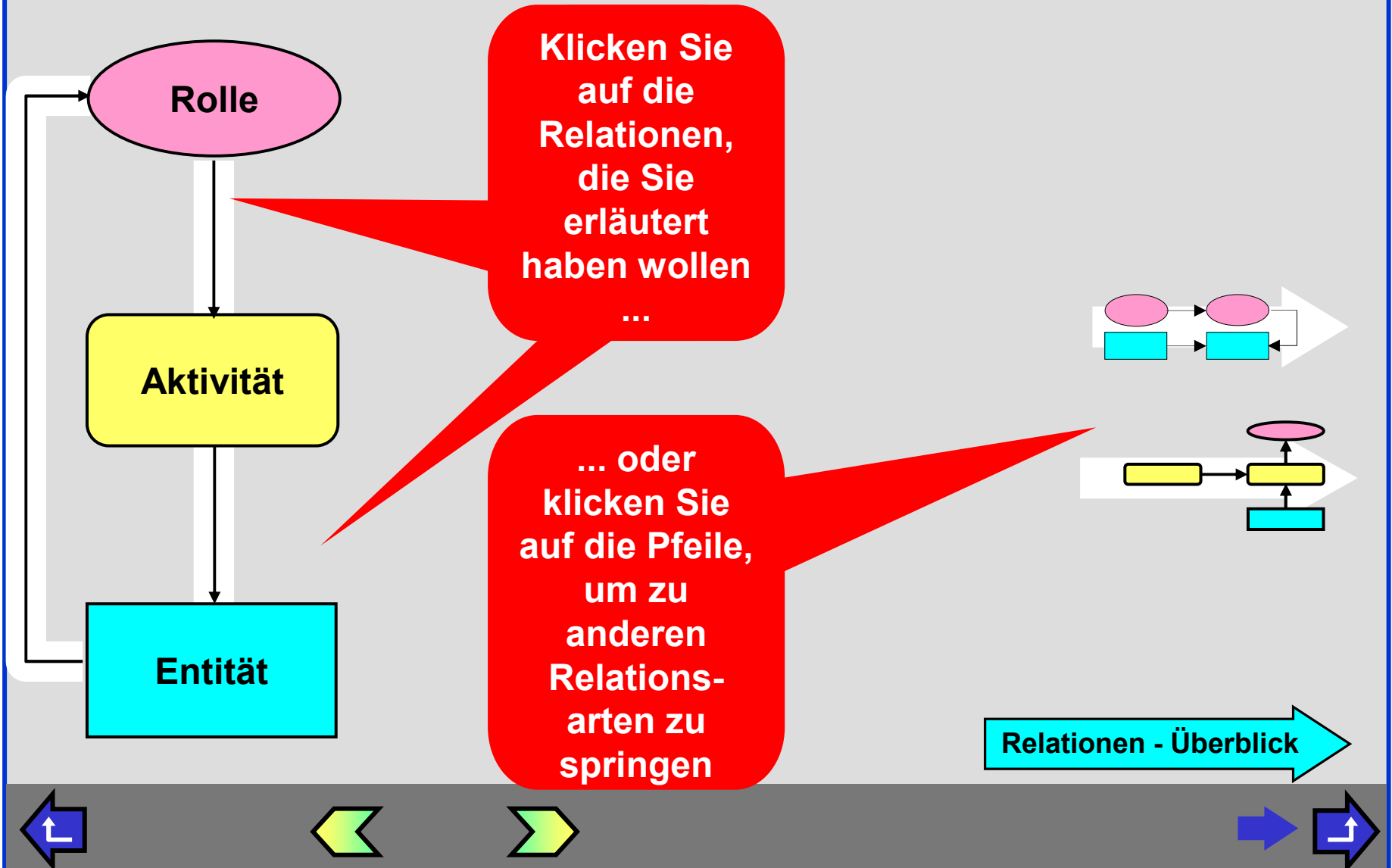


## 2) Gliederungsübersicht zu einfachen Standard-Relationen

- 2.1 9 einfache Standard-Relationen
- 2.2 Abkürzungen
- 2.3 Alle Relationen auf einen Blick
- 2.4 Erweiterte Beispiele
- 2.5 Aufgaben und Verständniskontrolle

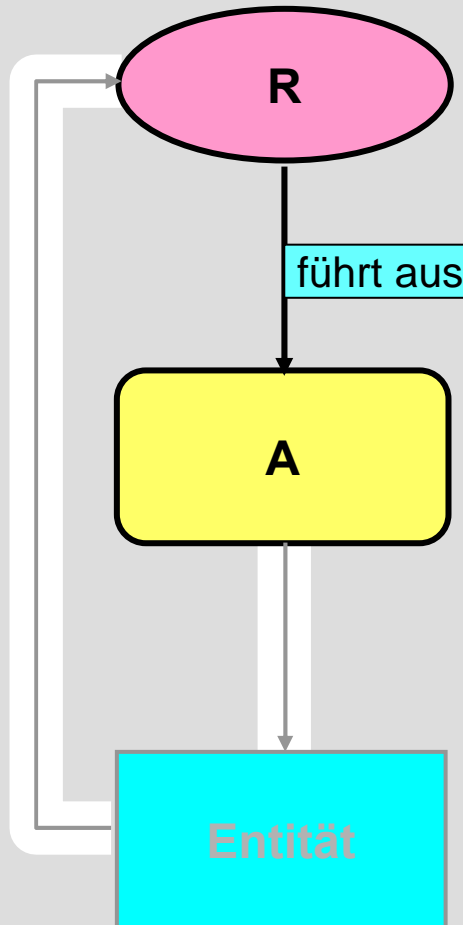


## 2.1 9 einfache Standard-Relationen – Teil I





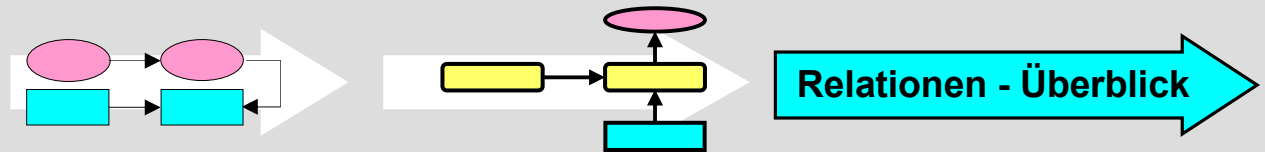
## 2.1.1 Relation: Rolle führt Aktivität aus



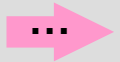
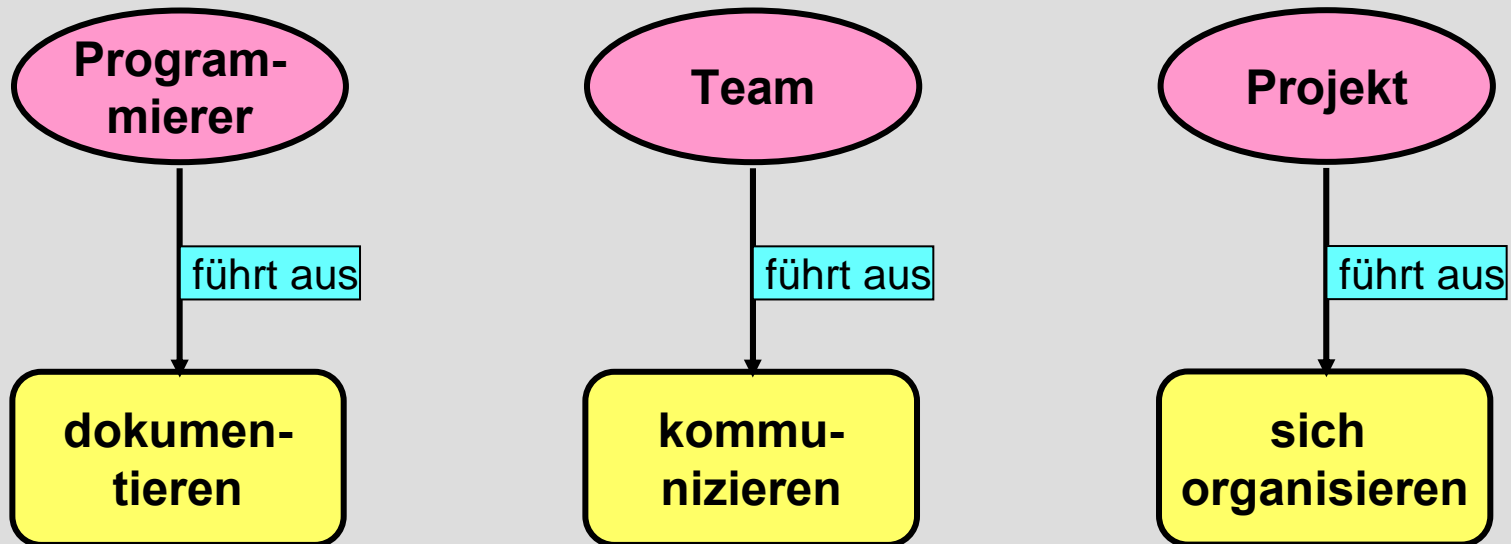
**R führt die Aktivität A vollständig aus.**

**Falls nichts anderes ausgedrückt ist, schließt dies mit ein, dass R die Initiative für die Durchführung von A ergreift und auch für das Ergebnis der Durchführung die Verantwortung trägt.**

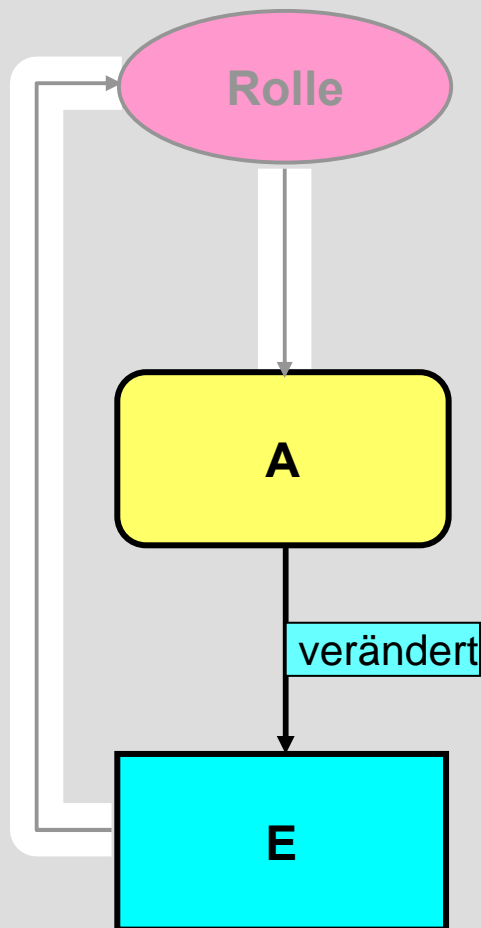
**Die Wirkung, die A auf R hat, muss nicht durch eine zusätzliche Relation ausgedrückt werden.**



## 2.1.1 Beispiele: Rolle führt Aktivität aus



## 2.1.2 Relation: Aktivität verändert eine Entität

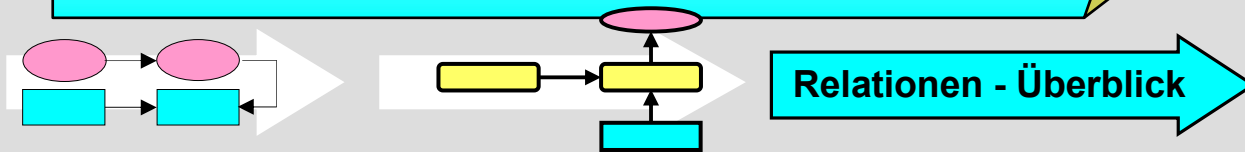


A verändert die Entität E. Diese Veränderung betrifft Eigenschaften von E, die sie als Ganzes kennzeichnen, also nicht nur Teile von E. Die Aktivität *steuern* verändert z.B. ständig solche Eigenschaften. Eigenschaften werden i.d.R. durch Attribute beschrieben.

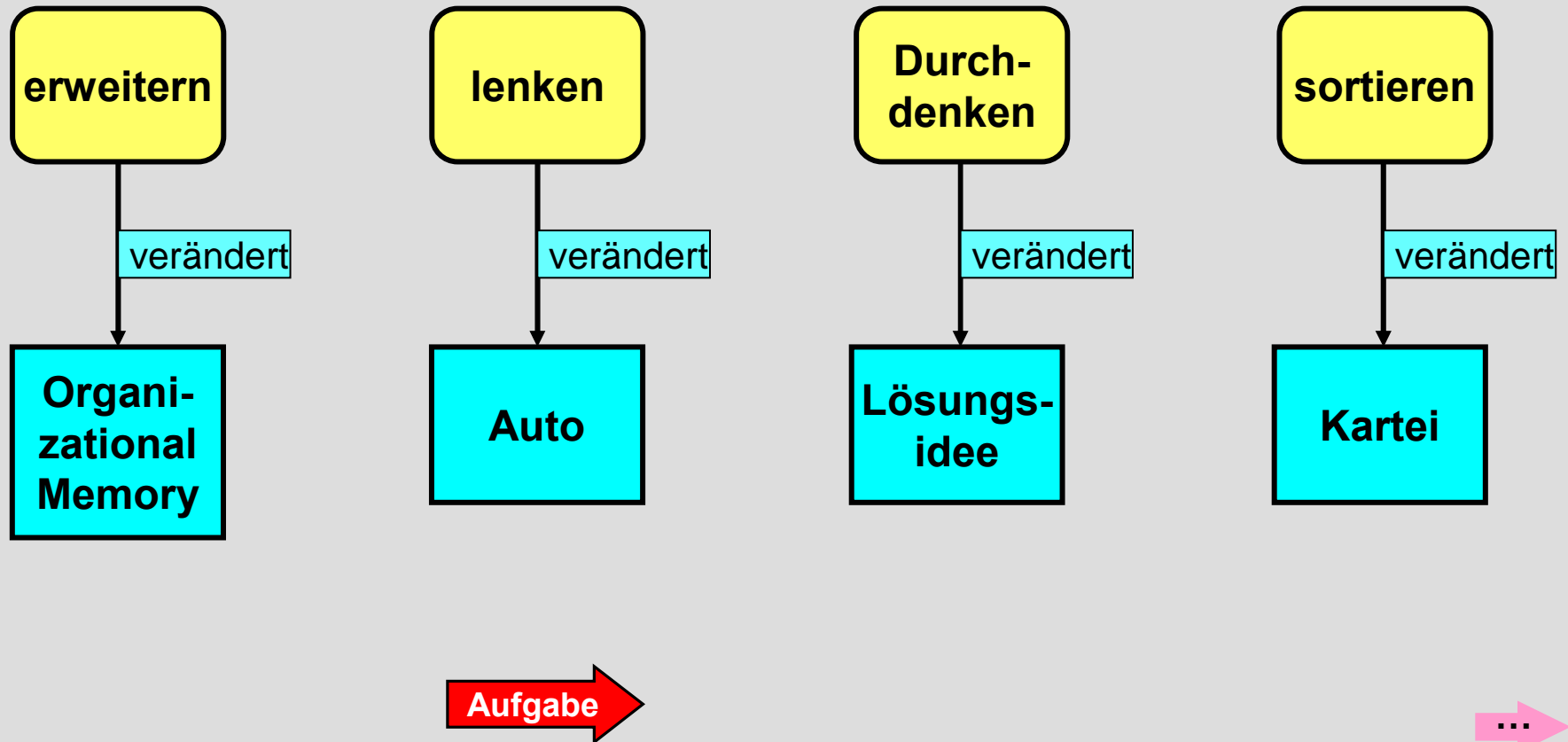
Veränderungen können auch vorgenommen werden, indem Elemente, die zu E gehören, gelöscht oder hinzugefügt werden oder die Beziehungen zwischen ihnen verändert werden. Falls jedoch die Grundstruktur einer Entität verändert wird, ist dies gesondert darzustellen (s. Meta-Relationen)

Die implizite Nutzung der veränderten Entität muss nicht zusätzlich dargestellt werden.

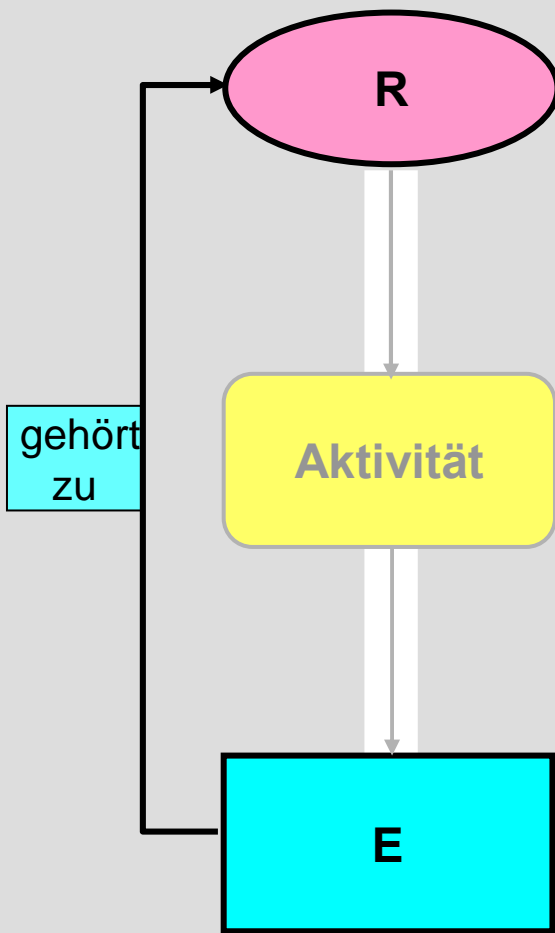
Die Veränderung von E kann erst als abgeschlossen angesehen werden, wenn A vollständig ausgeführt ist.



## 2.1.2 Beispiele: Aktivität verändert eine Entität



## 2.1.3 Relation: Eine Entität gehört zu einer Rolle

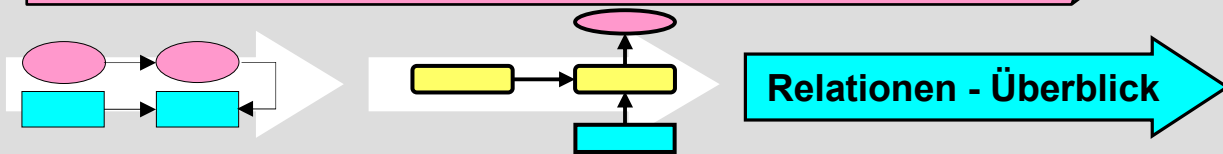


Die *gehört-zu-Relation* bedeutet, dass R über die gesamte Entität E verfügt. Das heißt, dass R mehr Zugriffsrechte oder Nutzungsrechte bzgl. E hat als andere Rollen.

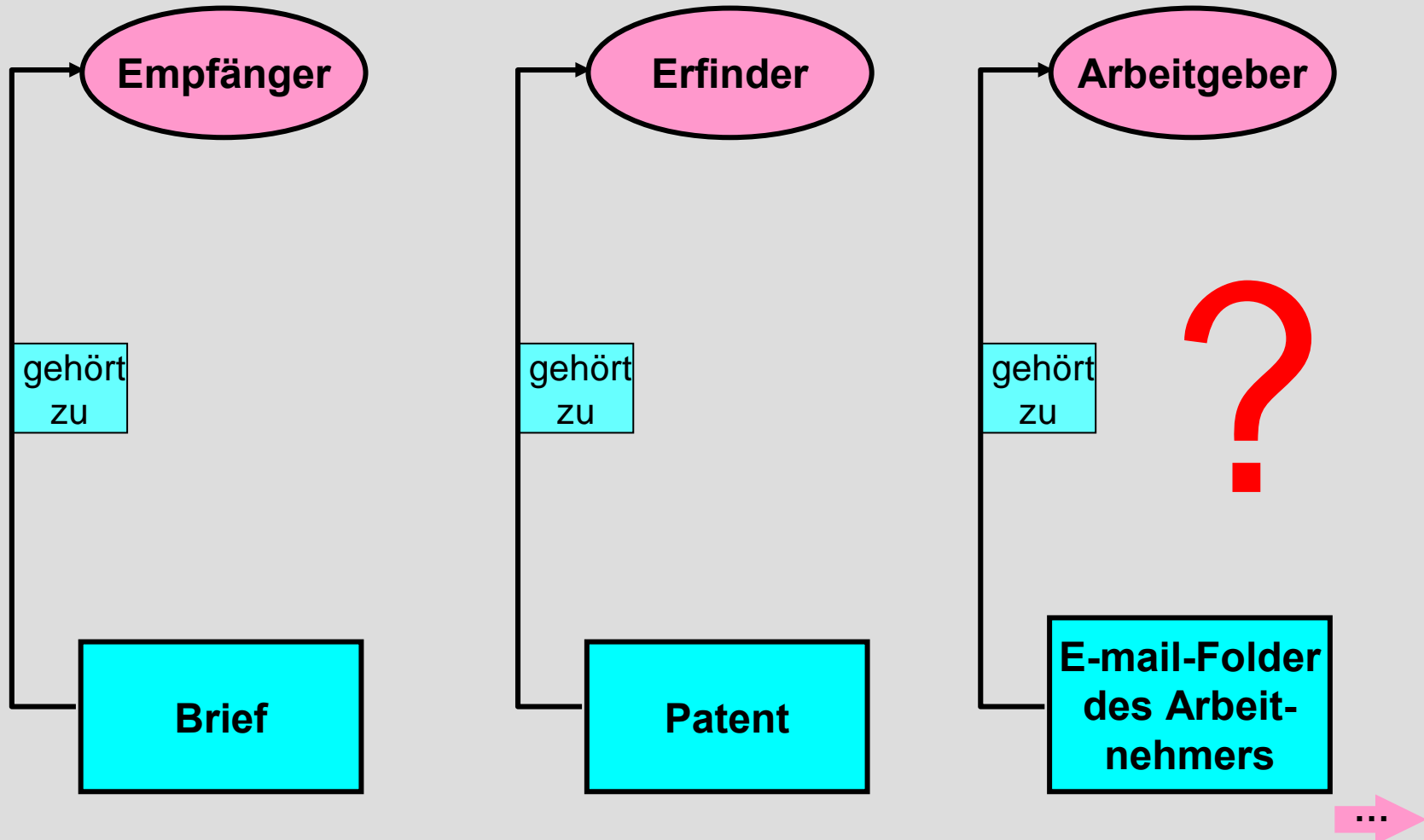
Ferner kann R Zugriffs- oder Nutzungsrechte anderen Rollen zuordnen oder entziehen.

Die *gehört-zu-Relation* repräsentiert einen Sachverhalt, der sich aus sozialen Strukturen und der Eigenarten einer Rolle ableiten. Dies kann zu Konflikten mit technischen Sachverhalten kommen, wenn z.B. ein alleiniges Zugriffsrecht mit einem Systemadministrator geteilt werden muss.

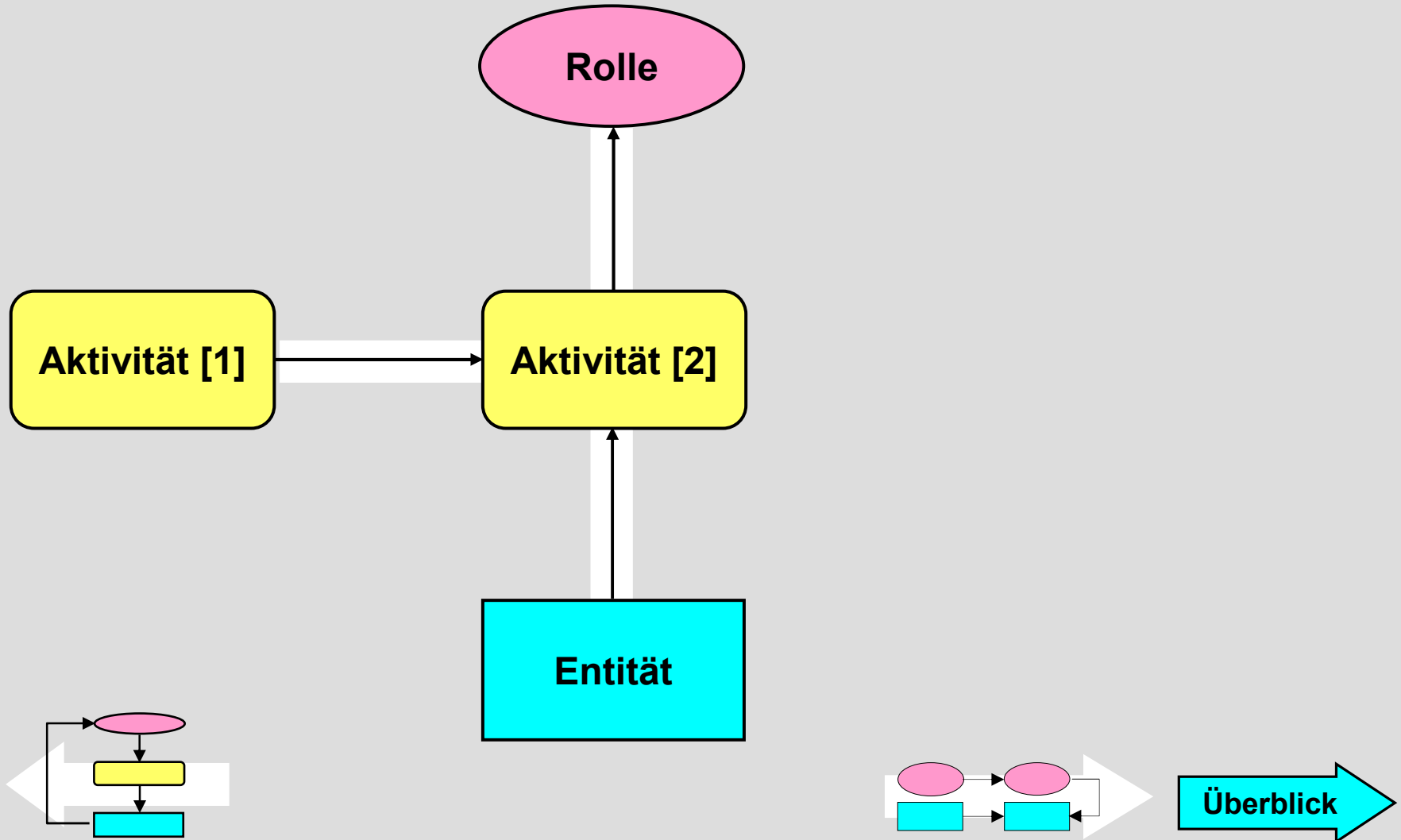
Ein erzeugtes E gehört zunächst der erzeugenden Rolle.



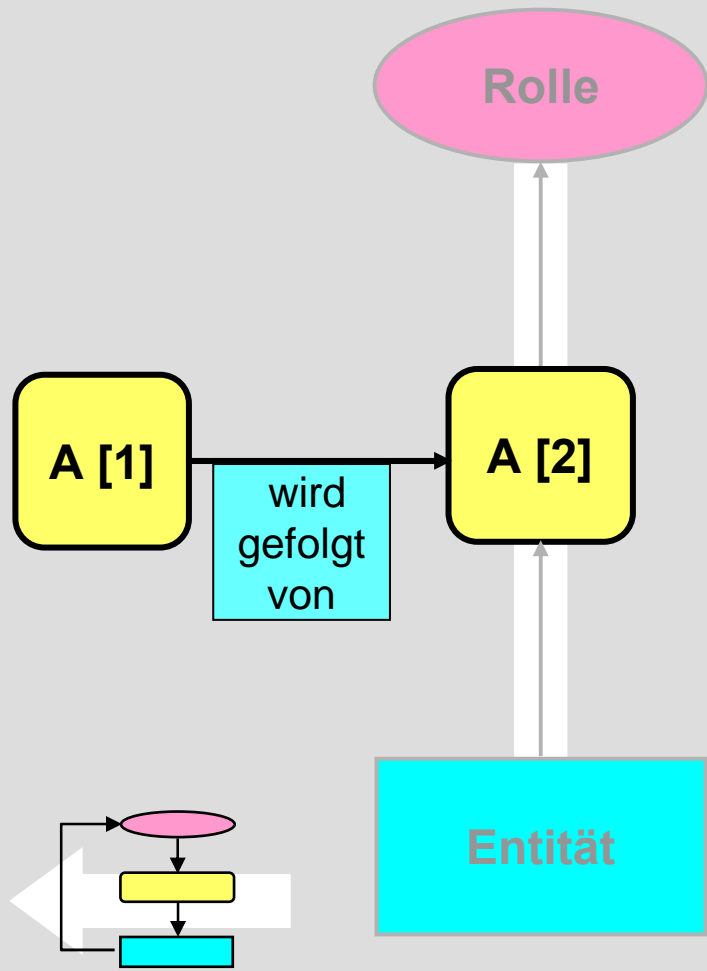
## 2.1.3 Beispiele: Eine Entität gehört zu einer Rolle



# 9 einfache Standard-Relationen – Teil II



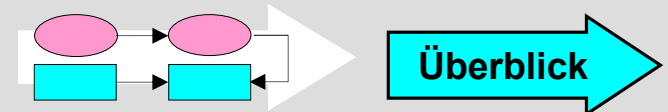
## 2.1.4 Relation: Eine Aktivität wird gefolgt von einer anderen



A [2] folgt auf A [1], wenn A [1] vollständig durchgeführt wurde. Der Übergang von A [1] nach A[2] benötigt keine Zeit.

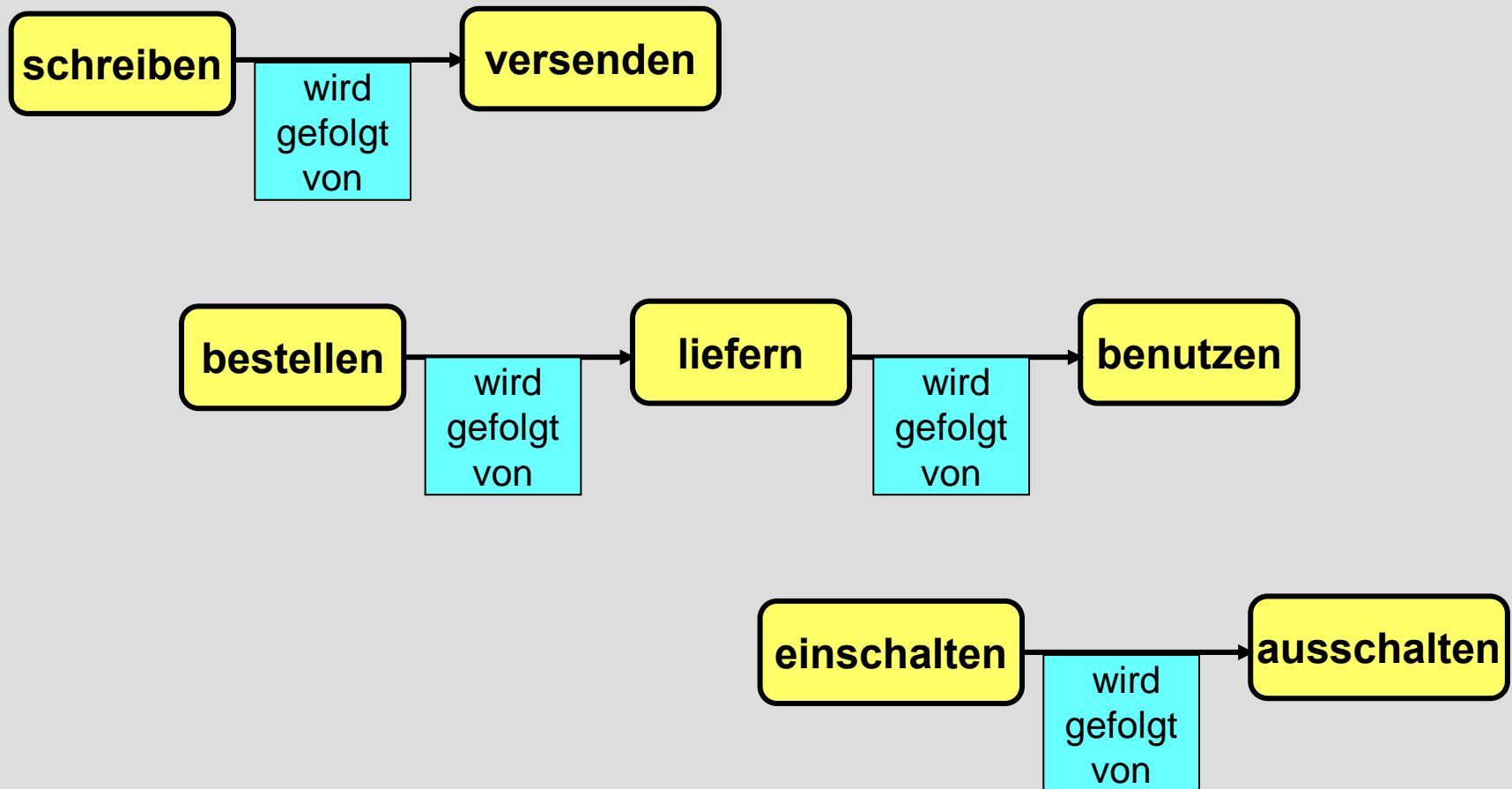
Hier wird deutlich, dass eine Relation eine Klasse von Übergängen repräsentiert, die mehrmals instantiiert werden kann.

Die Instantiierung der Relation *wird-gefolgt-von* drückt aus, dass A[1] und A[2] Teil einer gemeinsamen Instantiierung sind, innerhalb derer A[2] nicht startet, bevor A[1] beendet ist.

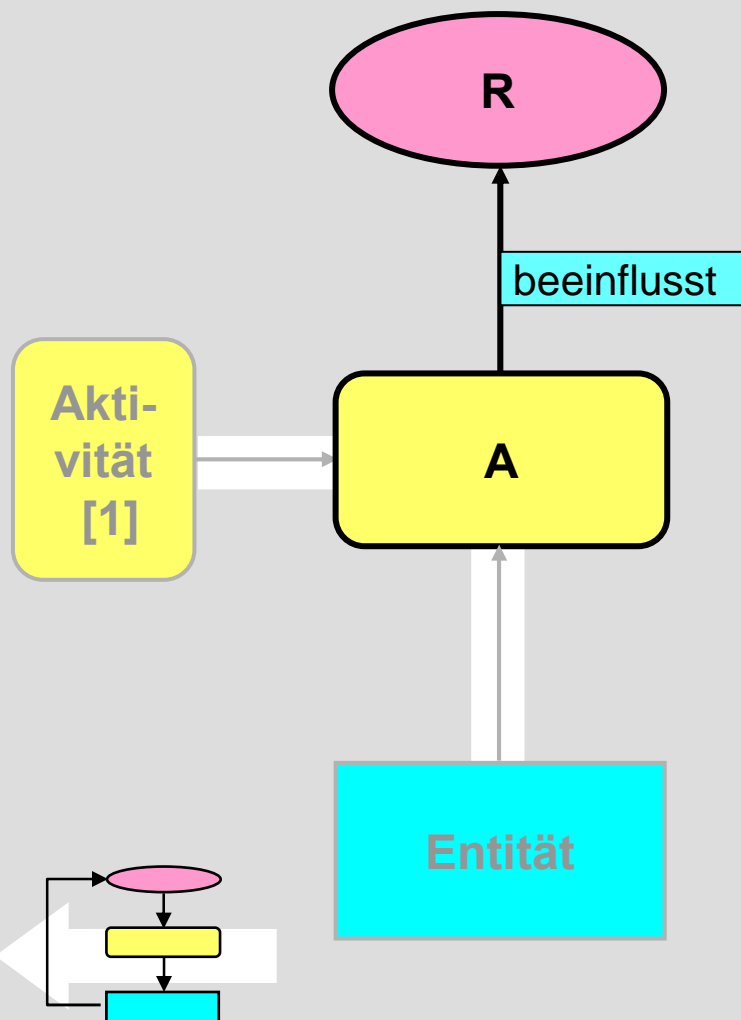




## 2.1.4 Beispiele: Eine Aktivität wird gefolgt von einer anderen



## 2.1.5 Relation: Eine Aktivität beeinflusst eine Rolle



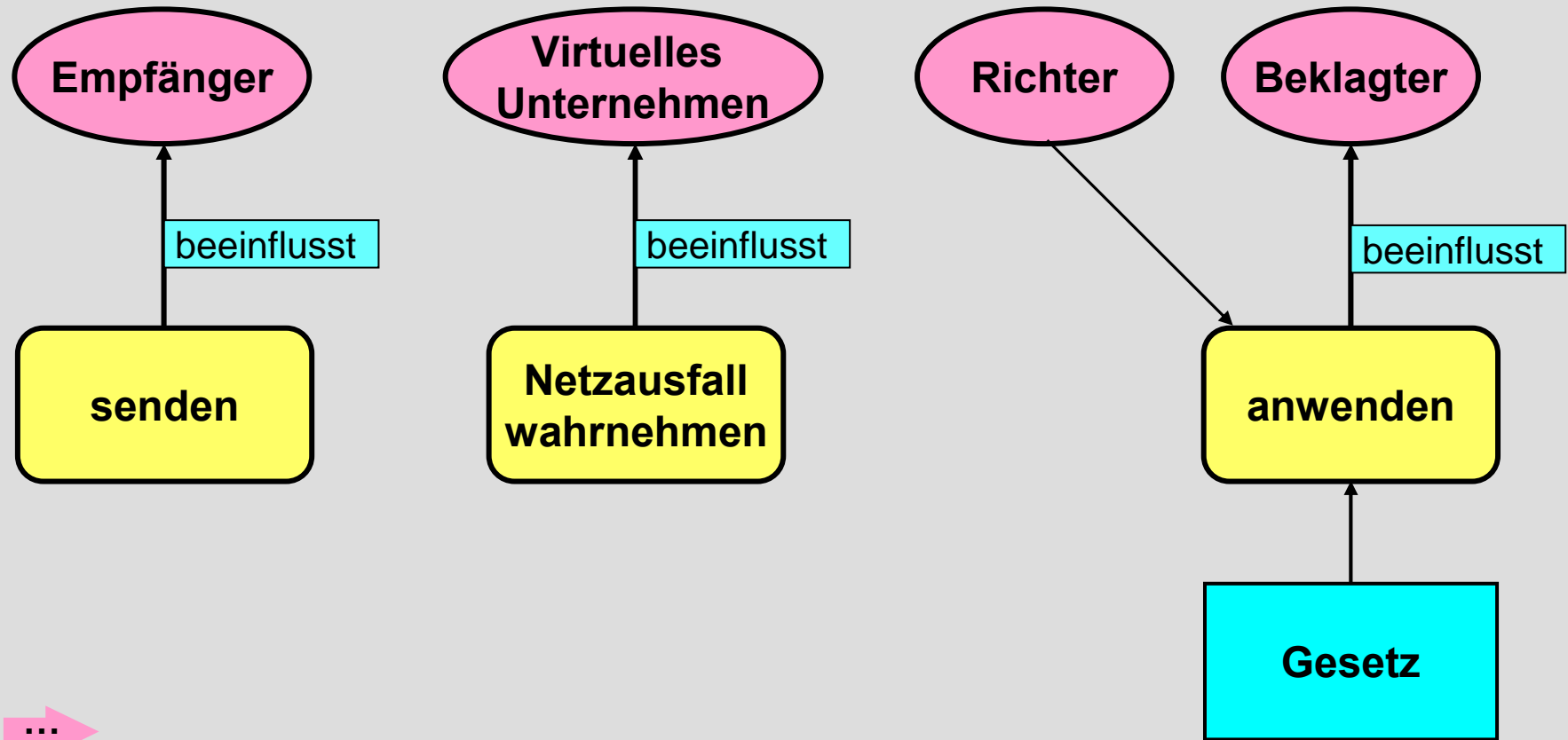
Soziale Sub-Systeme, die von Rollen repräsentiert werden, können nicht in gleicher Weise gesteuert werden, wie technische Systeme. Ihr Verhalten ist von außen nicht determinierbar, sondern nur beeinflussbar.

Die Relation zeigt einen Einfluss, der auf die gesamte Rolle ausgeübt wird und zwar solange, bis A abgeschlossen ist.

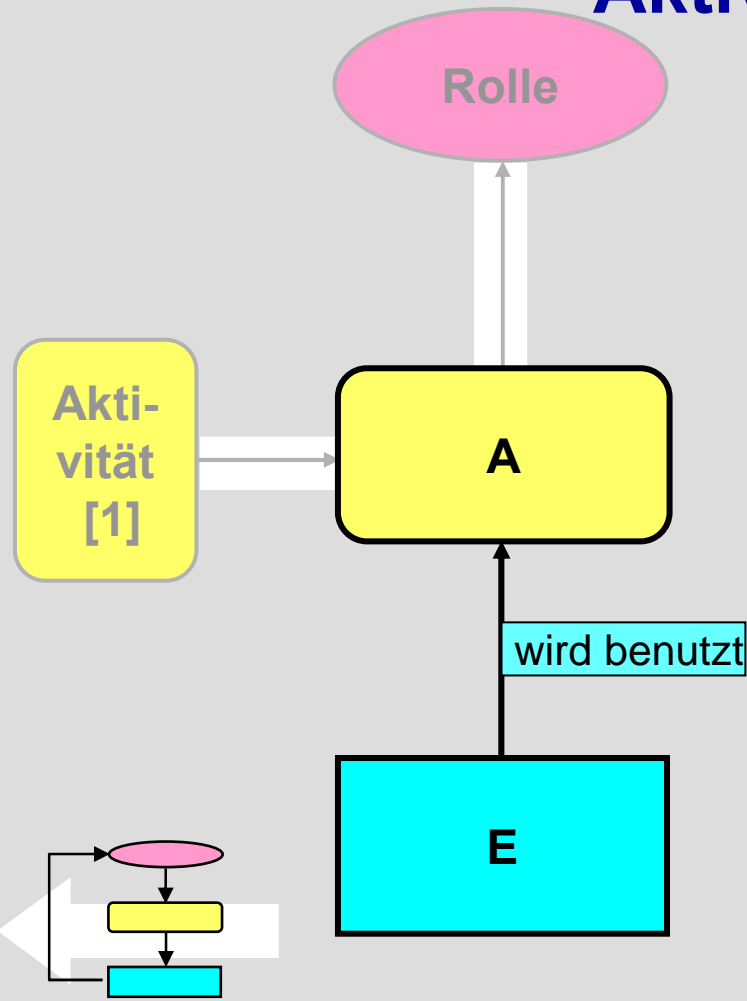
Ereignisse, Entitäten und Rollen können nur mittels einer Aktivität eine Rolle beeinflussen.



## 2.1.5 Beispiele: Eine Aktivität beeinflusst eine Rolle



## 2.1.6 Relation: Eine Entität wird von einer Aktivität benutzt



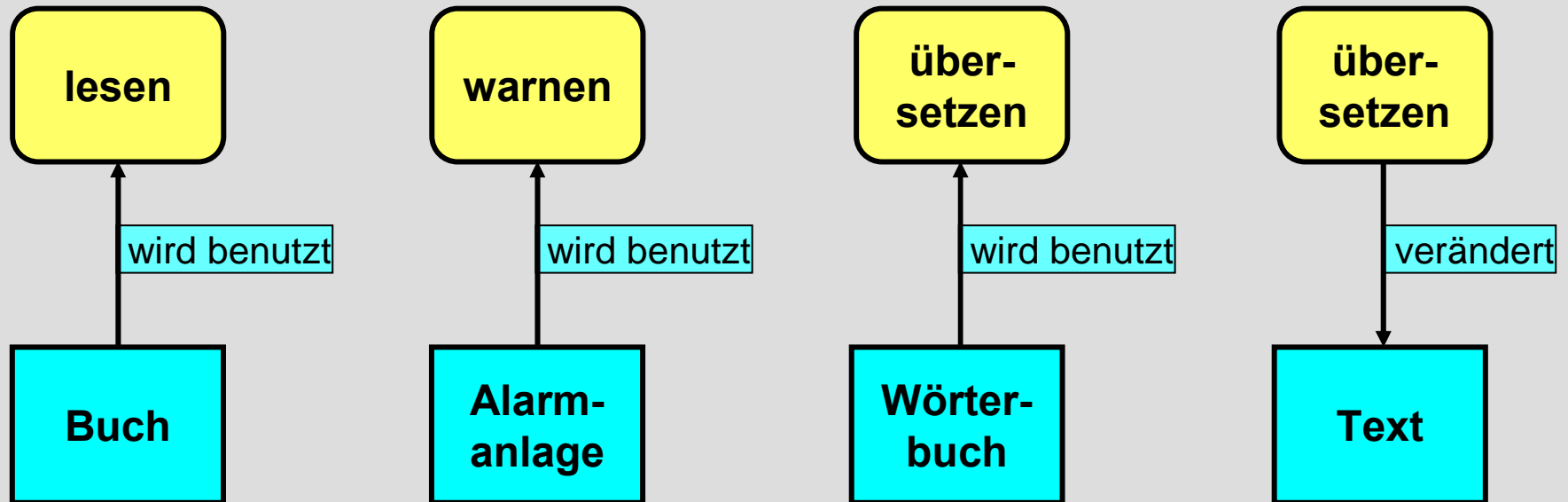
Die *wird-benutzt-von* Relation ordnet alle Arten von Entitäten als Ressourcen einer Aktivität zu, die die Durchführung dieser Aktivität A unterstützen oder erst ermöglichen oder sogar vollständig übernehmen.

Dabei wird die gesamte Aktivität von der gesamten Entität unterstützt oder ermöglicht.

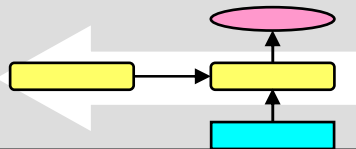
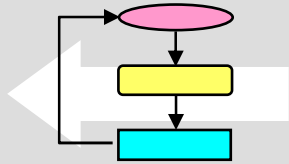
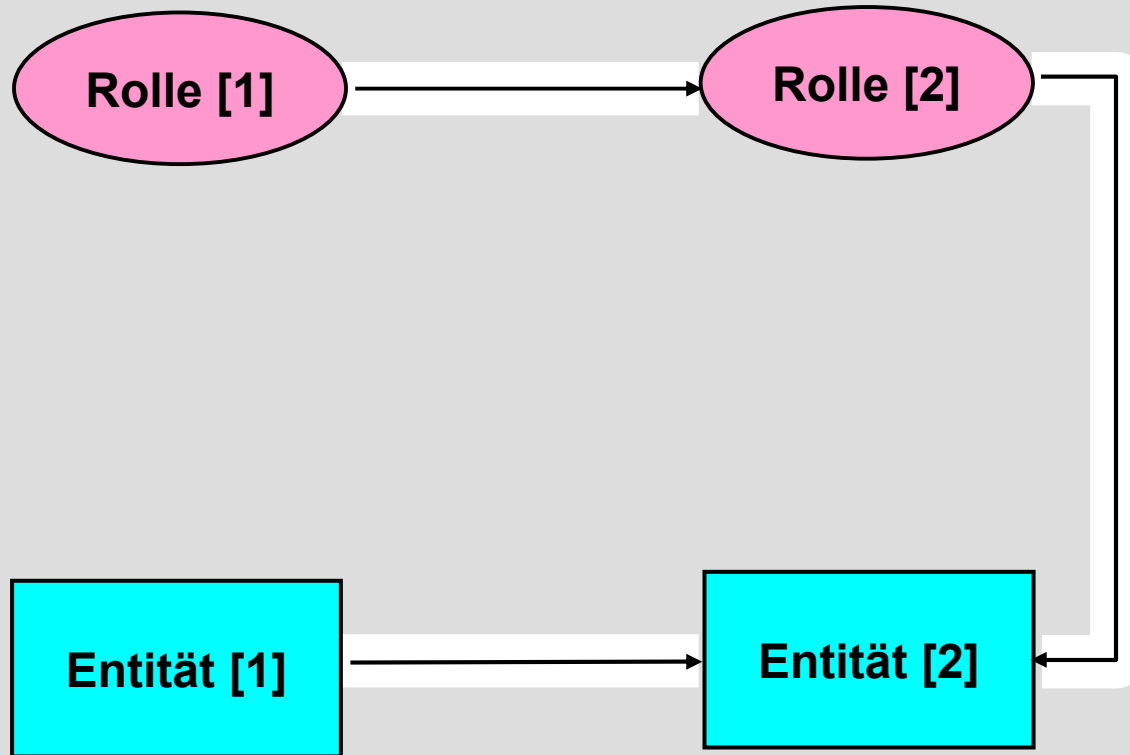


## 2.1.6 Beispiele: Eine Entität wird von einer Aktivität genutzt

aber:

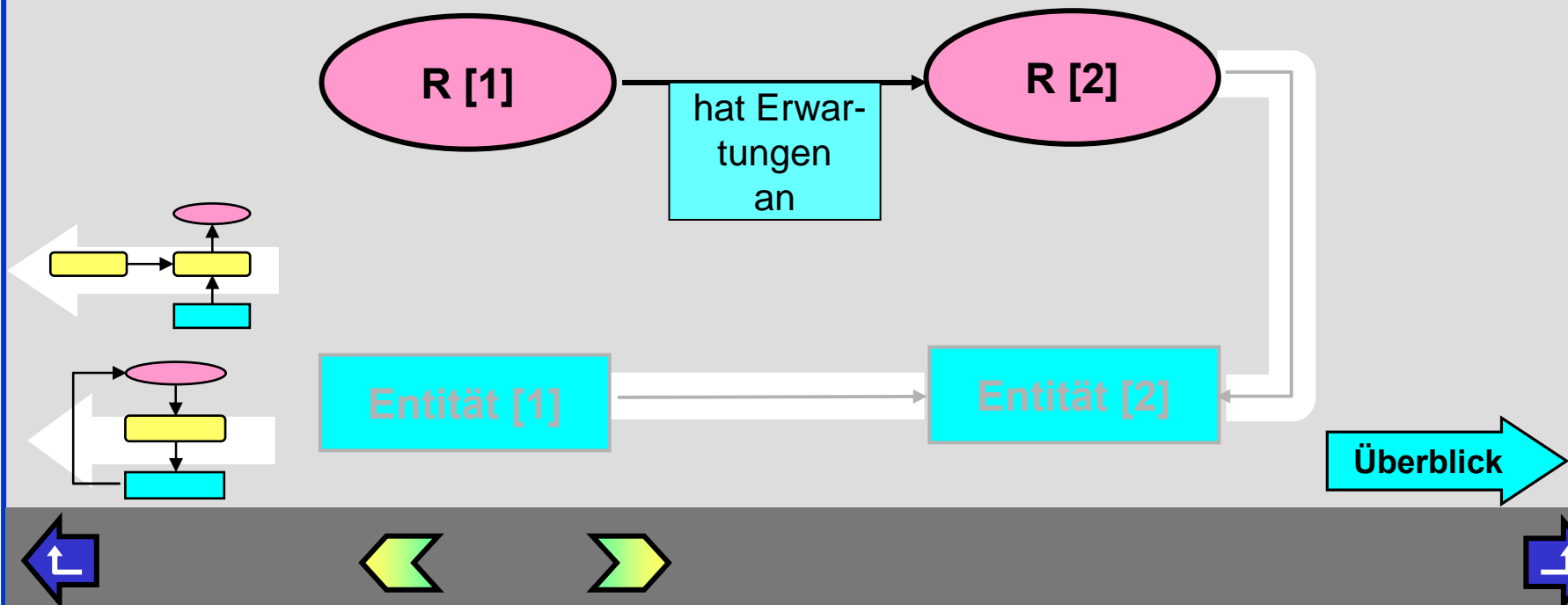


# 9 einfache Standard-Relationen – Teil III

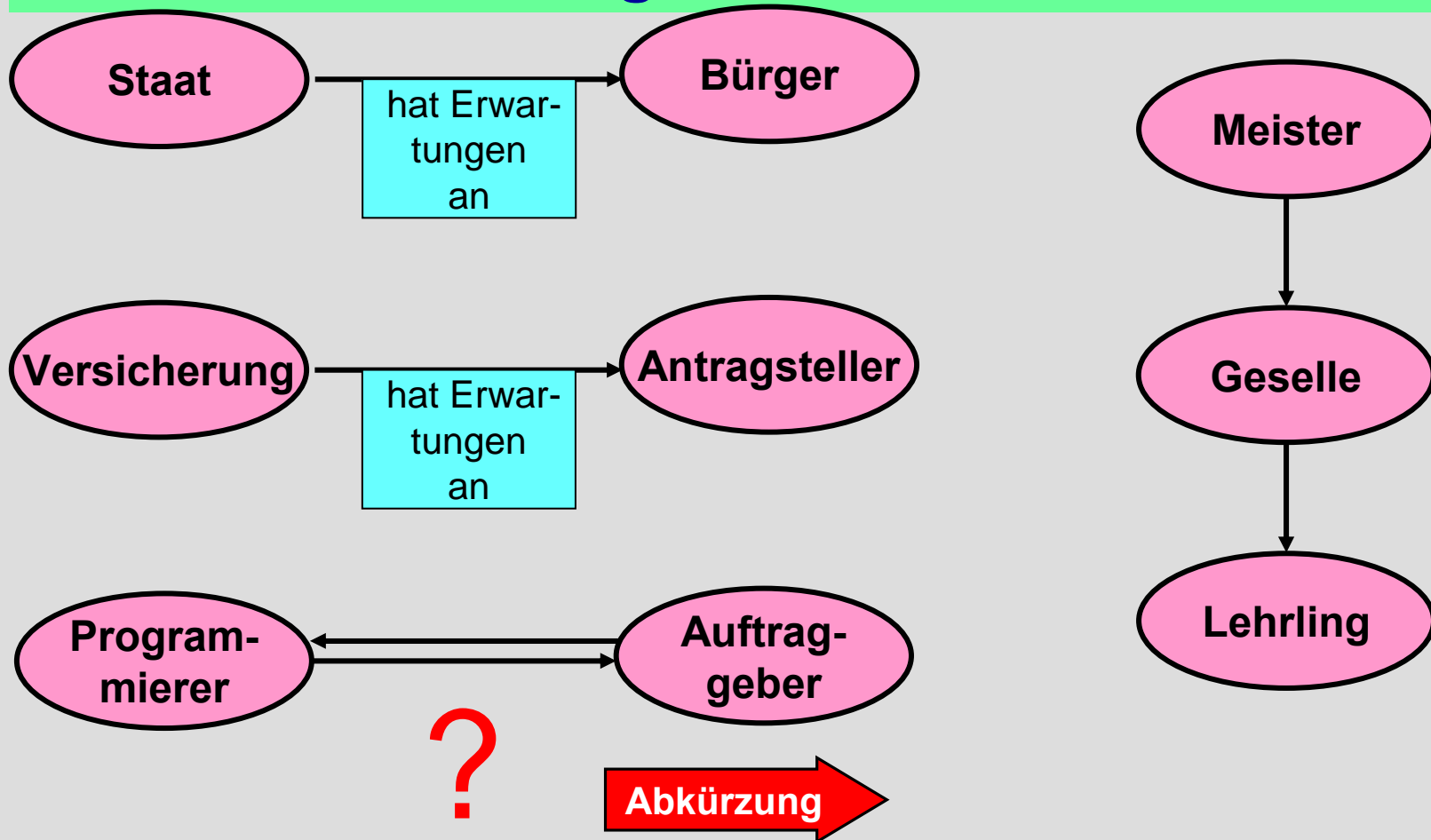
**Überblick**

## 2.1.7 Relation: Eine Rolle hat durchsetzbare Erwartungen an eine andere

Rollen richten Erwartungen an andere Rollen. Diese Relation wird eingetragen, wenn Rollen psychologisch, wirtschaftlich, rechtlich, militärisch oder ähnlich in der Lage sind durchzusetzen, dass die andere Rolle diesen Erwartungen nachkommt. Falls ein symmetrisches Verhältnis vorliegt, werden zwei Pfeile eingetragen, sonst der stärkere Erwartungsdruck; so lassen sich Hierarchien darstellen.



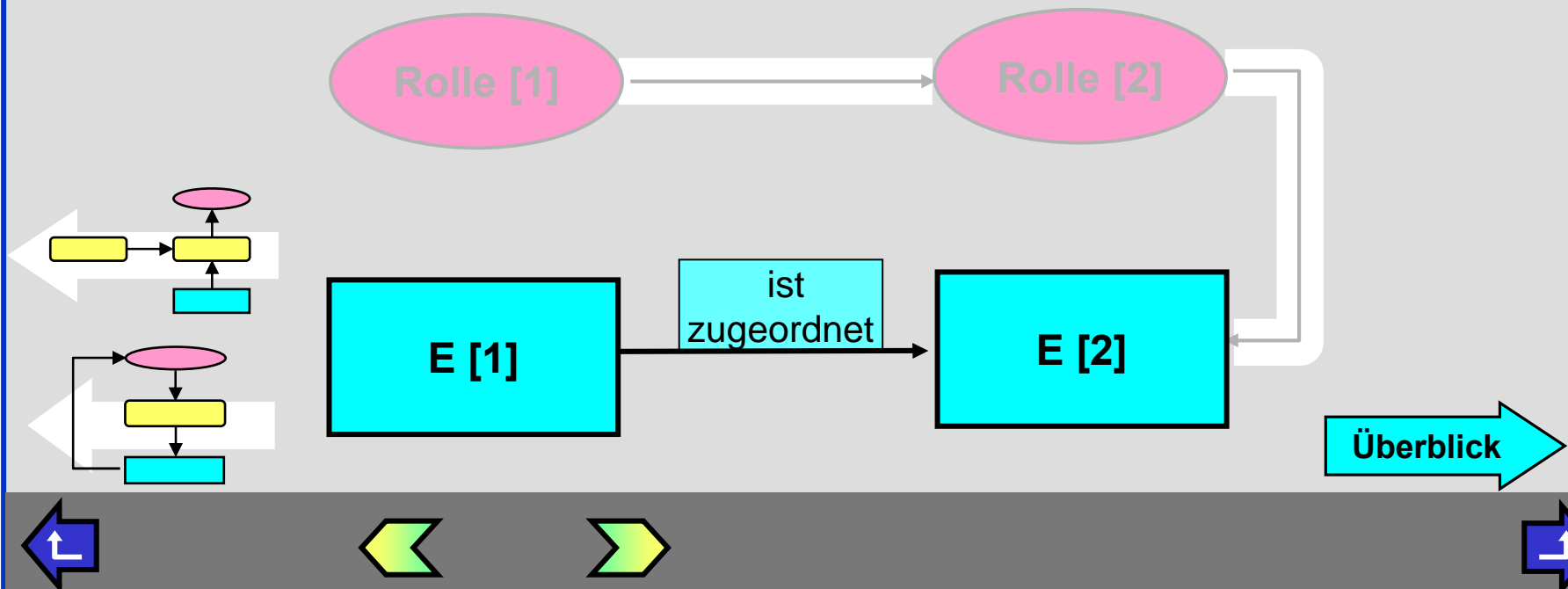
## 2.1.7 Beispiele: Eine Rolle hat durchsetzbare Erwartungen an eine andere



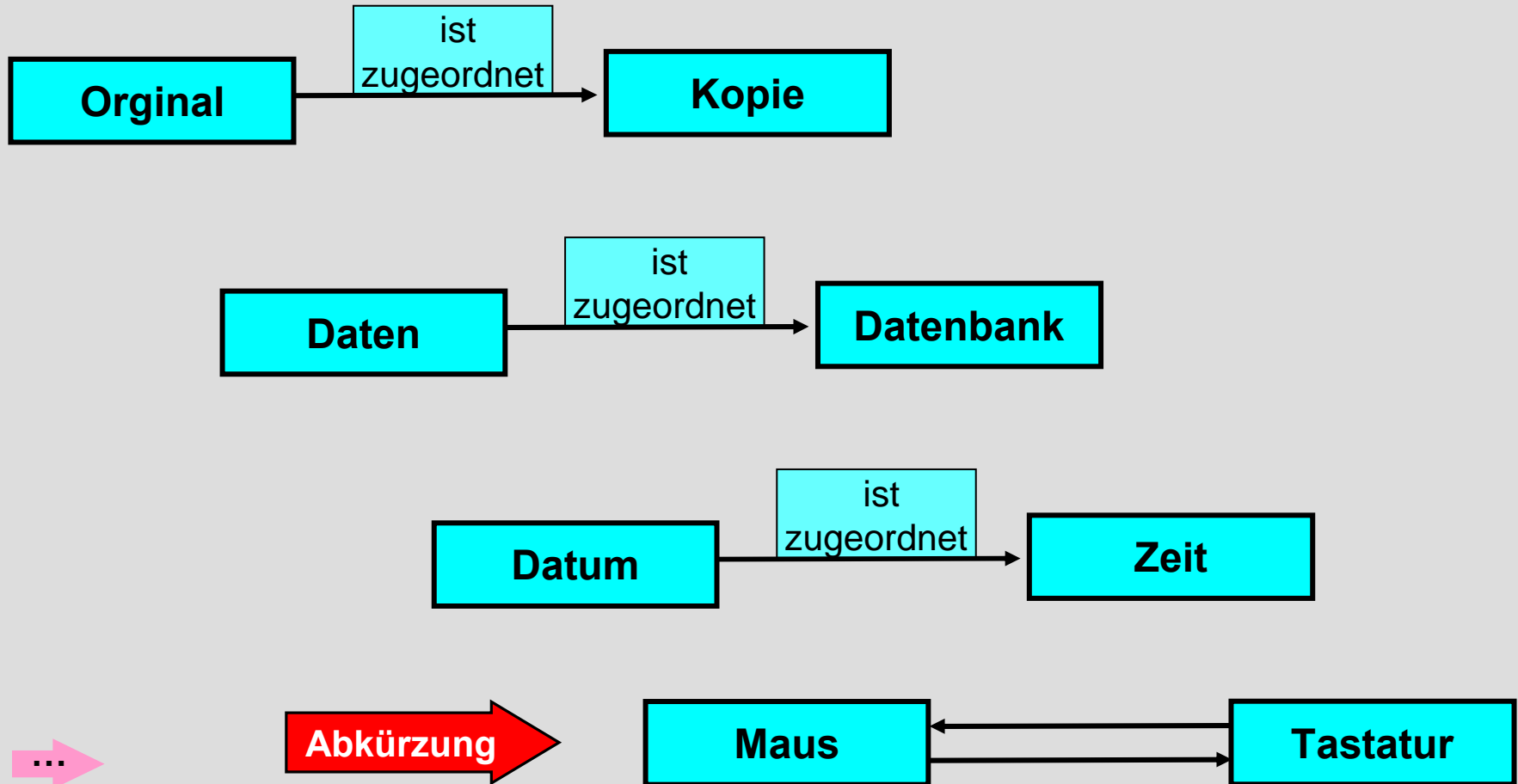


## 2.1.8 Relation: Eine Entität ist einer anderen zugeordnet

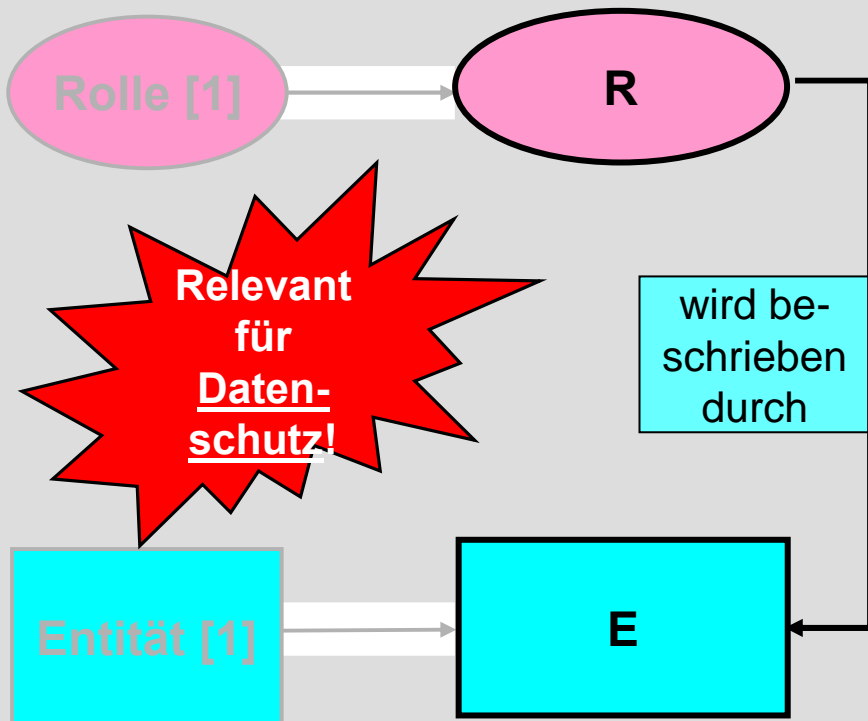
Die Relation ist-zugeordnet-zu wird immer eingetragen, wenn zwei Entitäten als Ganze zueinander in Beziehung stehen. Die Art der Beziehung oder die Perspektive, unter der die Beziehung vom Modellierer betrachtet wird, bestimmt die Richtung der Relation. So kann die Existenz von E[1] der Existenz von E[2] vorausgesetzt sein oder E[2] ist die umfassendere oder allgemeinere Entität.



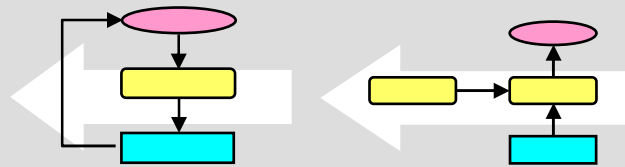
## 2.1.8 Beispiele: Eine Entität ist einer anderen zugeordnet



## 2.1.9 Relation: Eine Rolle wird durch eine Entität beschrieben



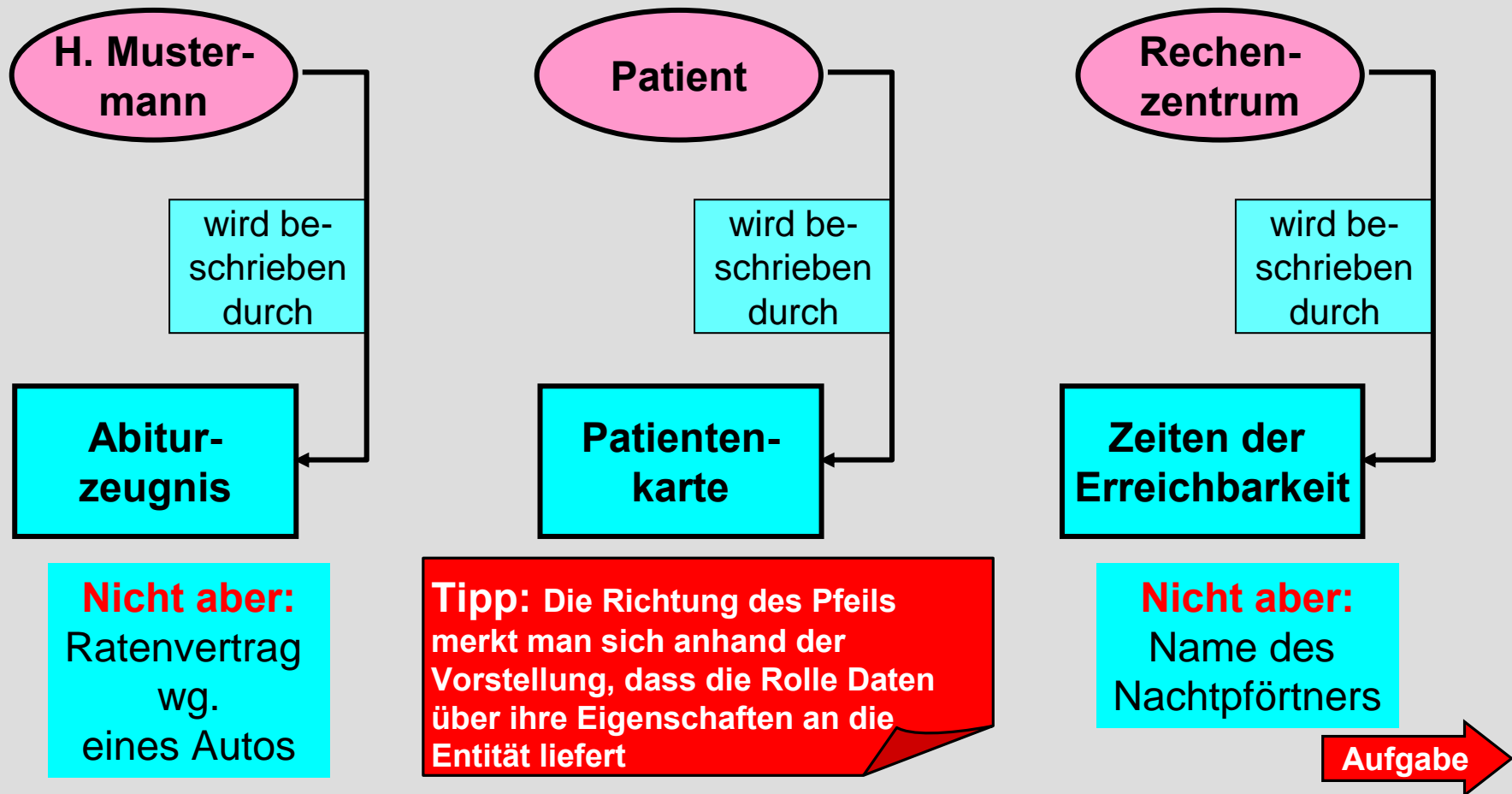
Eine Rolle kann durch eine Entität beschrieben werden. Die Entität beinhaltet Beschreibungen, die für die Rolle als ganzes relevant sind. Die Relation wird nur aufgestellt, wenn die Beschreibung so vorliegt, dass sie anderen Rollen zugänglich sein kann. Auch wenn sich die Beschreibung auf die Rolle als Ganzes bezieht, kann diese niemals vollständig beschrieben werden



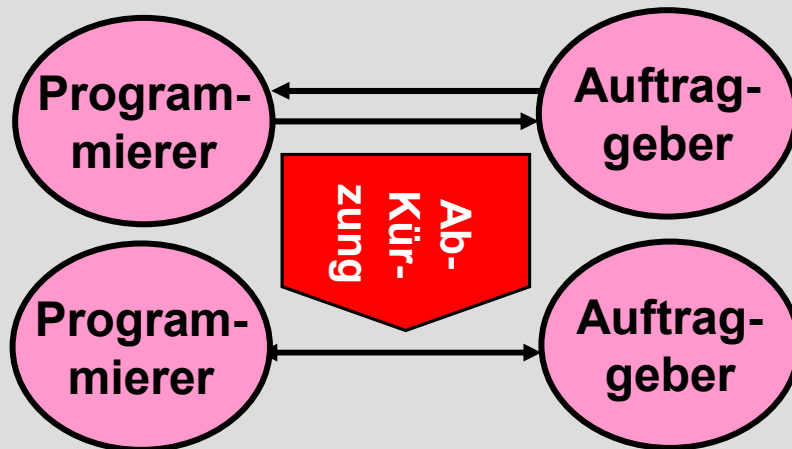
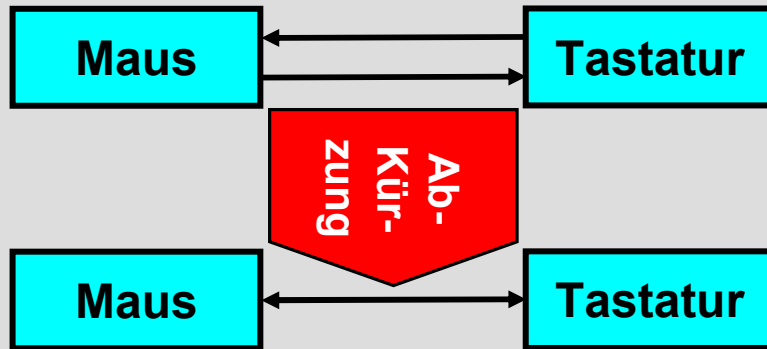
Überblick



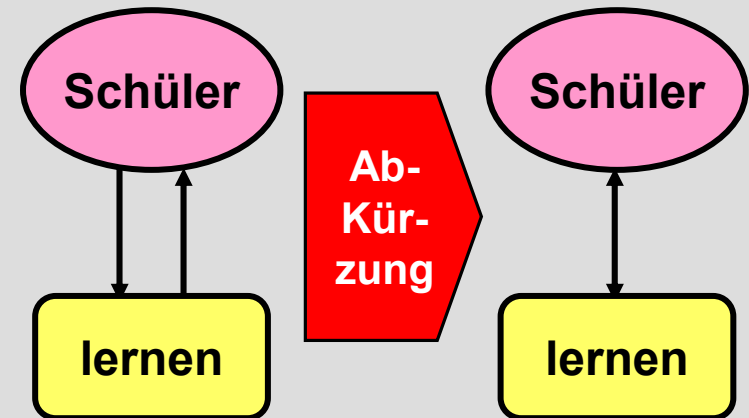
## 2.1.9 Beispiele: Eine Rolle wird durch eine Entität beschrieben



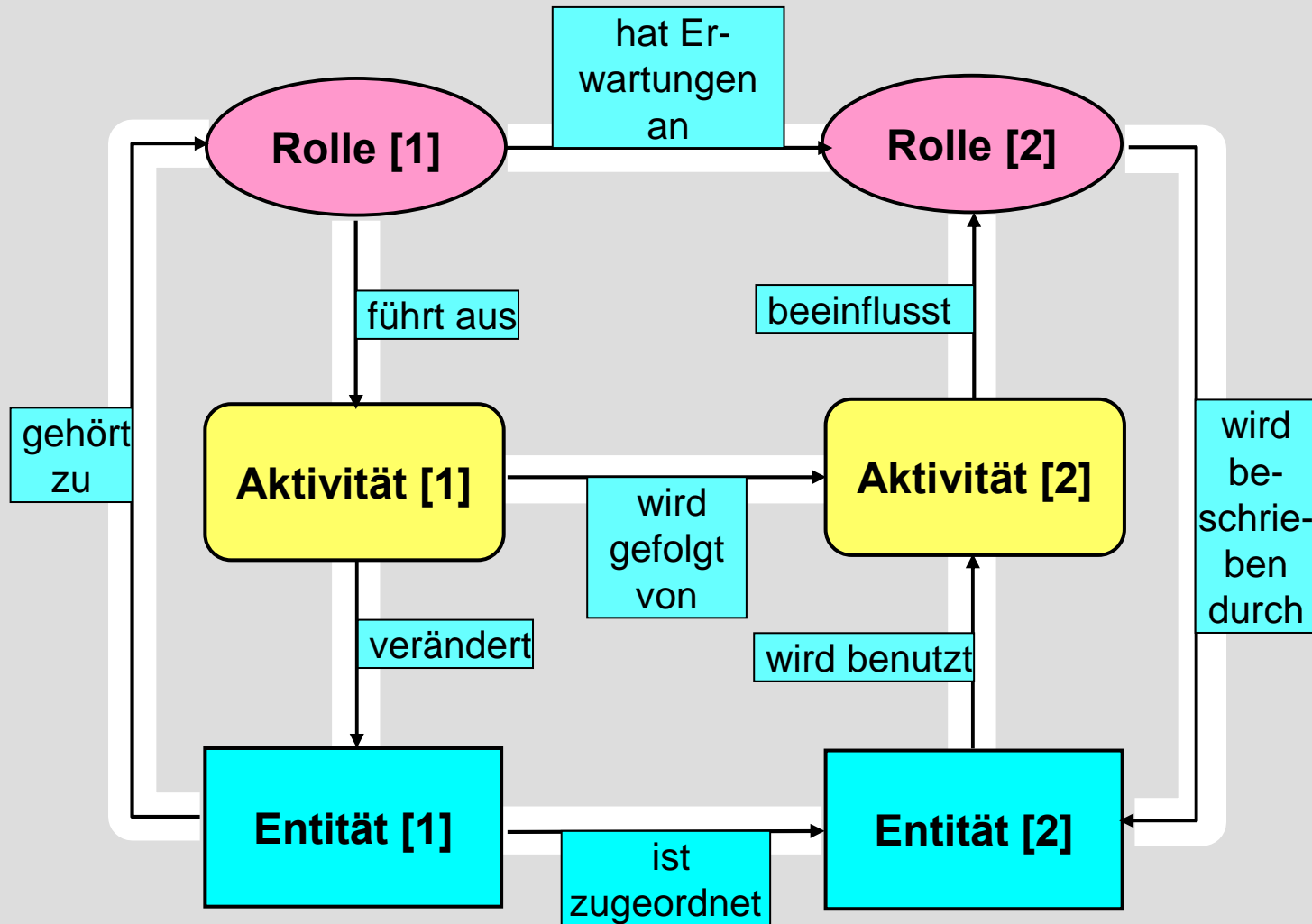
## 2.2 Abkürzungsmöglichkeit: Verwendung von Doppelpfeilen



Auch Relationen unterschiedlichen Typs können zusammengefasst werden:



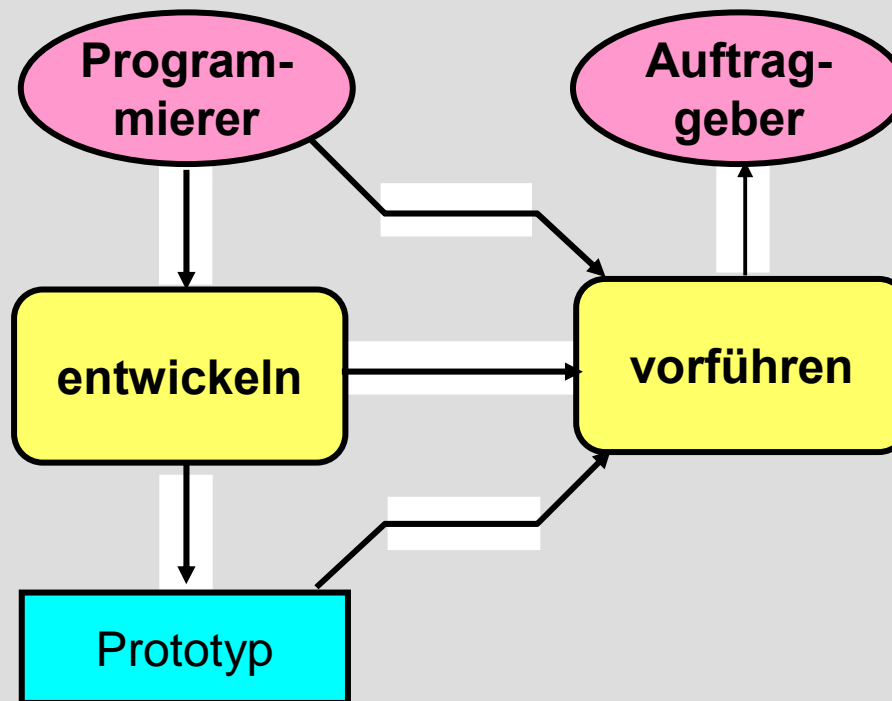
## 2.3 Alle Standard Relationen im Überblick



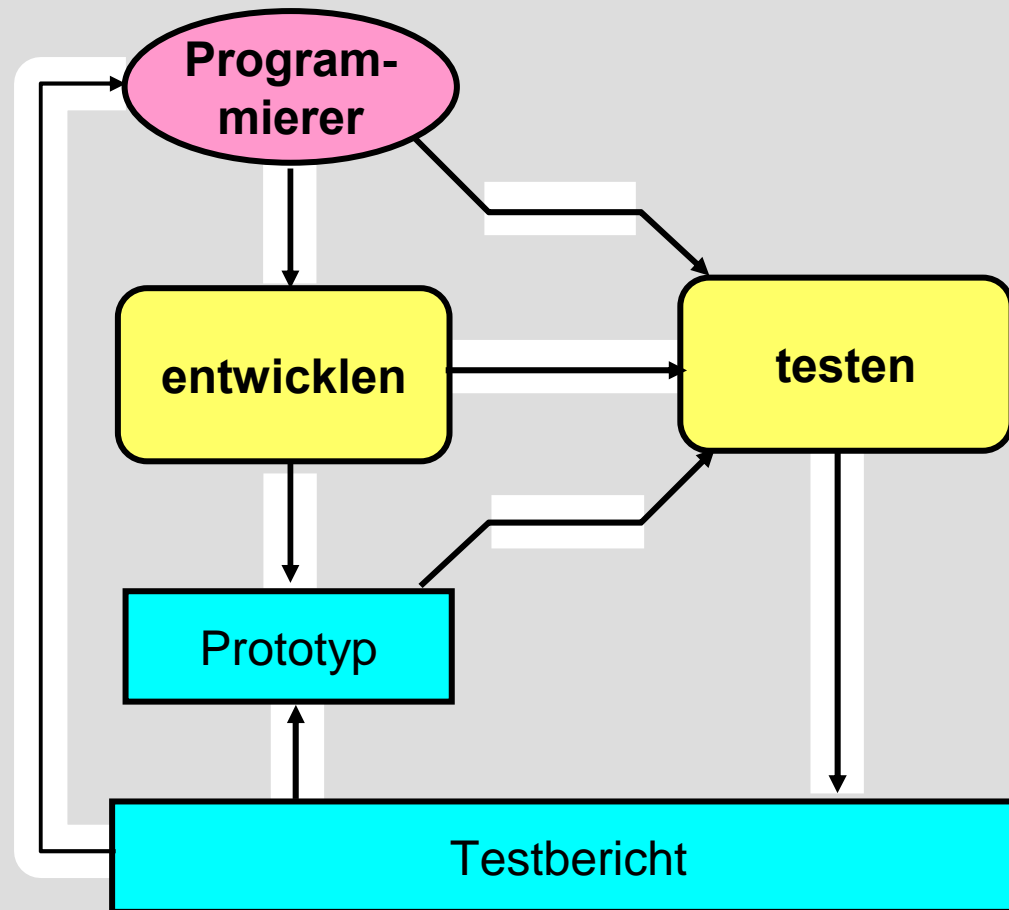
**TIPP:**  
Bei  
Standard-  
relationen  
kann die  
Typ-  
Beschrei-  
bung  
angehängt  
werden,  
muss aber  
nicht.



## 2.4.1 Erweiterte Beispiele I

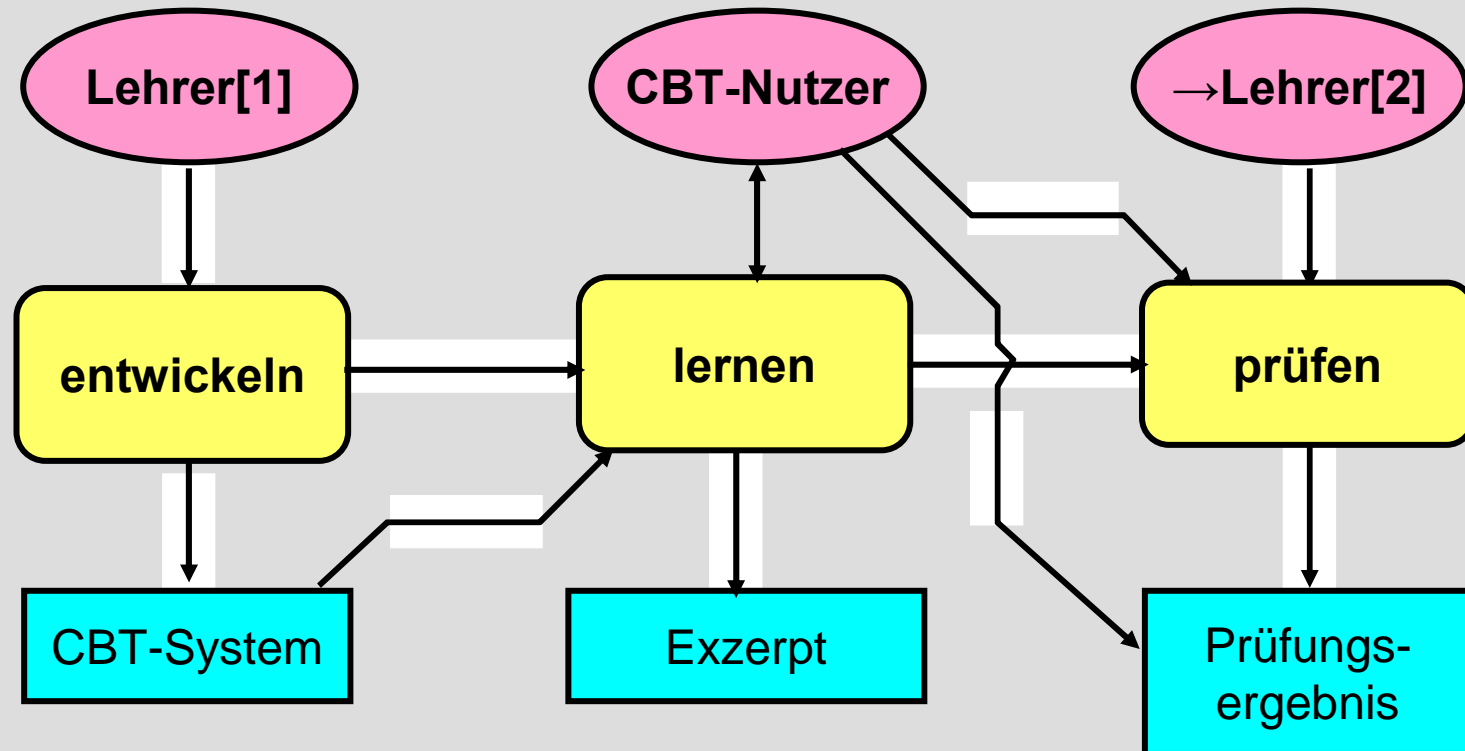


## 2.4.2 Erweiterte Beispiele II





## 2.4.3 Erweiterte Beispiele III

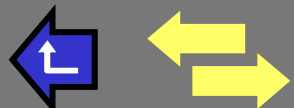


**Tipp:** Das gleiche Basis-Element kann an verschiedenen Stellen des Diagramms wiederholt verwendet werden, wie hier die Rolle Lehrer, sollte dann aber durch einen Pfeil (→) gekennzeichnet werden. Durch die Nummerierung kann man ausdrücken, dass ein anderer Lehrer als der, der das CBT System entwickelt hat, die Prüfung abnimmt.

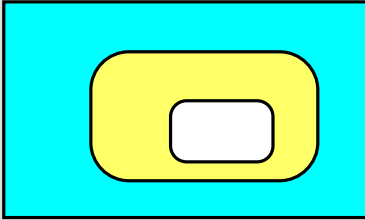


## 2.5 Aufgaben zu einfachen Standard-Relationen

1. Verbinden sie die Aktivitäten löschen, aktualisieren, beobachten, aussagen, bezeugen durch Relationen mit je einer Entität.
2. Erstellen Sie Diagramme, die neben den Entitäten aus [Kapitel 1.3 – Beispiel III](#) jeweils eine Rolle und eine Aktivität beinhalten.
3. Geben Sie ein sinnvolles Beispiel, bei dem zwei Aktivitäten mit einem Doppelpfeil verbunden werden – erläutern Sie es.
4. Geben Sie weitere Beispiele, bei denen sich Rollen selbst beeinflussen – erläutern Sie sie.
5. Entwerfen Sie ein Diagramm zum Thema „Versenden von E-Mail“, bei dem sämtliche Typen von Relationen verwendet werden.
6. Fügen Sie dem [Diagramm mit der Rolle Patient](#) mindestens noch 5 weitere, verschiedene Relationen hinzu.

Aufg.

### 3) Einbettung von Elementen



Gliederung

[3.1](#) 9 Formen der Einbettung

[3.2](#) Einbettung von Attributen

[3.3](#) Segmentierung

[3.4](#) Einbettung und Relationen

[3.5](#) Aufgaben

Elemente können als Sub-Elemente in andere Basis-Elemente beliebigen Typs eingebettet werden – diese repräsentieren dann das Super-Element.

Sub-Elemente können wiederum Sub-Elemente enthalten – sie bilden ein Sub-System. Die Einbettungstiefe ist beliebig. Eine Einbettung ist durch einen Farbwechsel zu kennzeichnen.

Durch Einbettung kann die Struktur einzelner Elemente dargestellt werden. Ein Element sollte nur eingebettet werden, wenn dessen Instantiierung davon abhängt, dass das über-geordnete Element ebenfalls instantiiert wird. Sonst sollte man es separat darstellen.





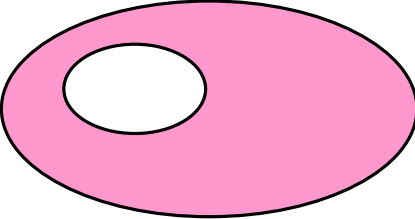
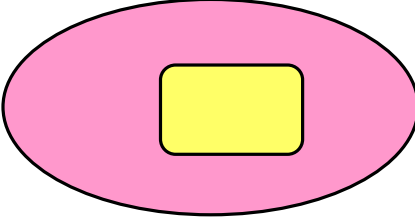
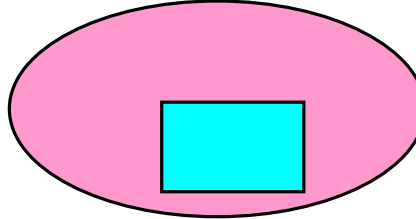

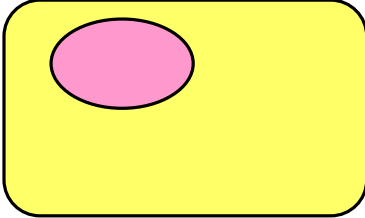
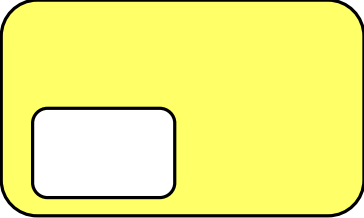
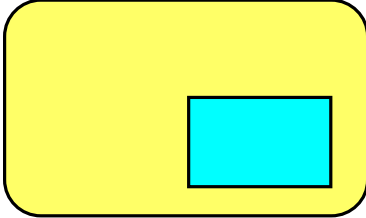

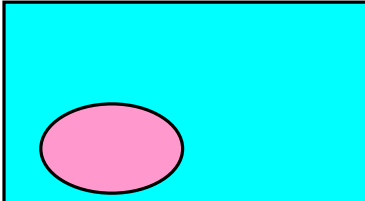
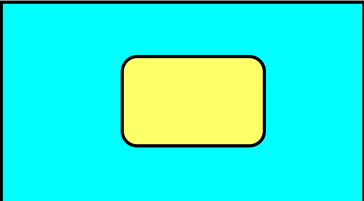
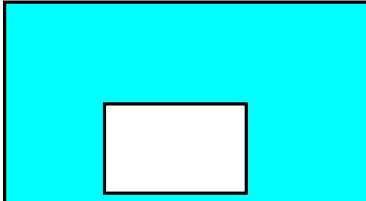
Die Beziehung zwischen eingebettetem und übergeordnetem Element kann verschiedene Bedeutungen haben (s. [→4.4](#))



# 3.1 9 Formen der Einbettung - Überblick

Über-  
Geordne-  
Tes Element

Sub-Ele-  
ment

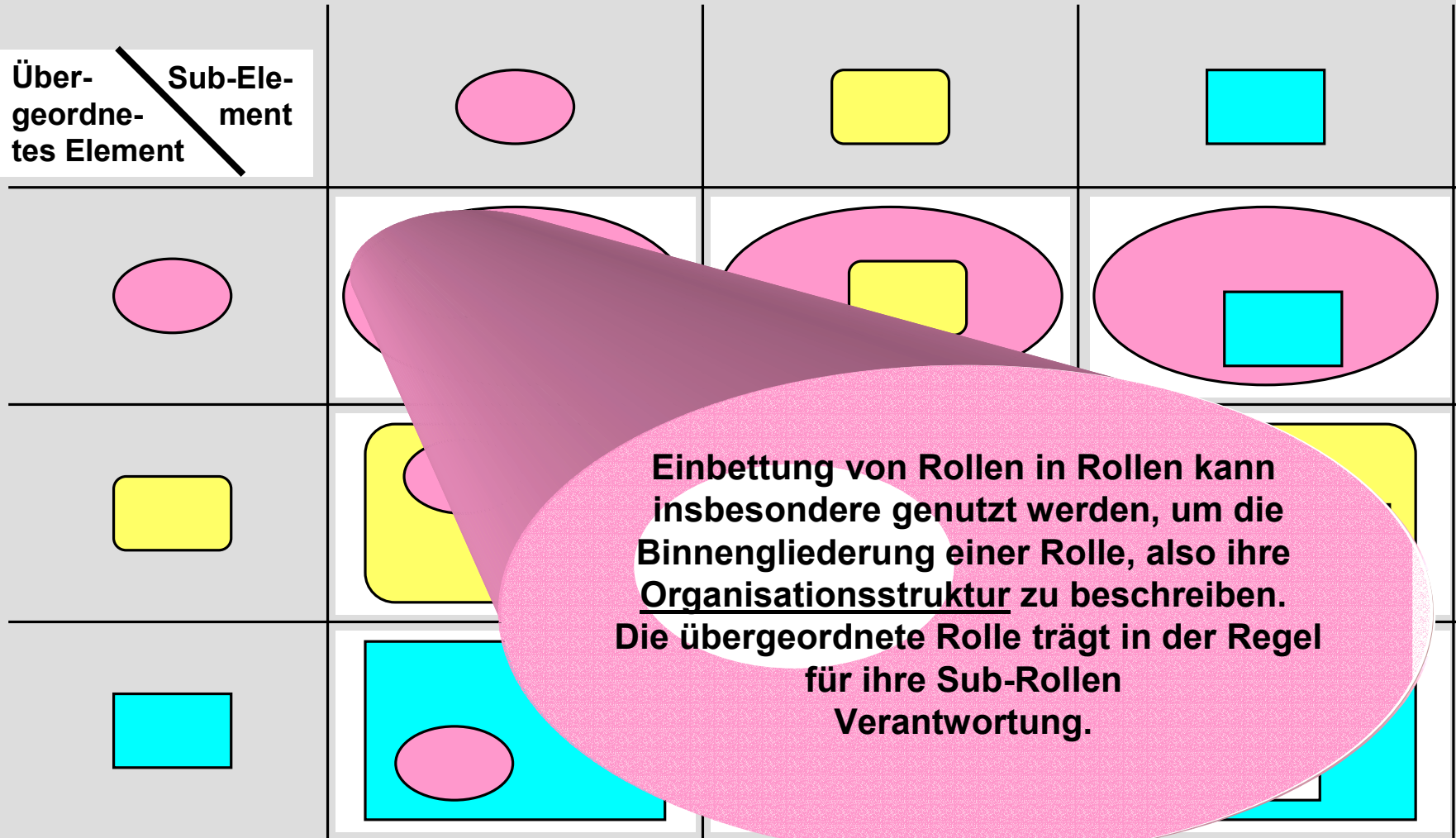
Die Einbettung ist durch einen Farbwechsel hervorzuheben!



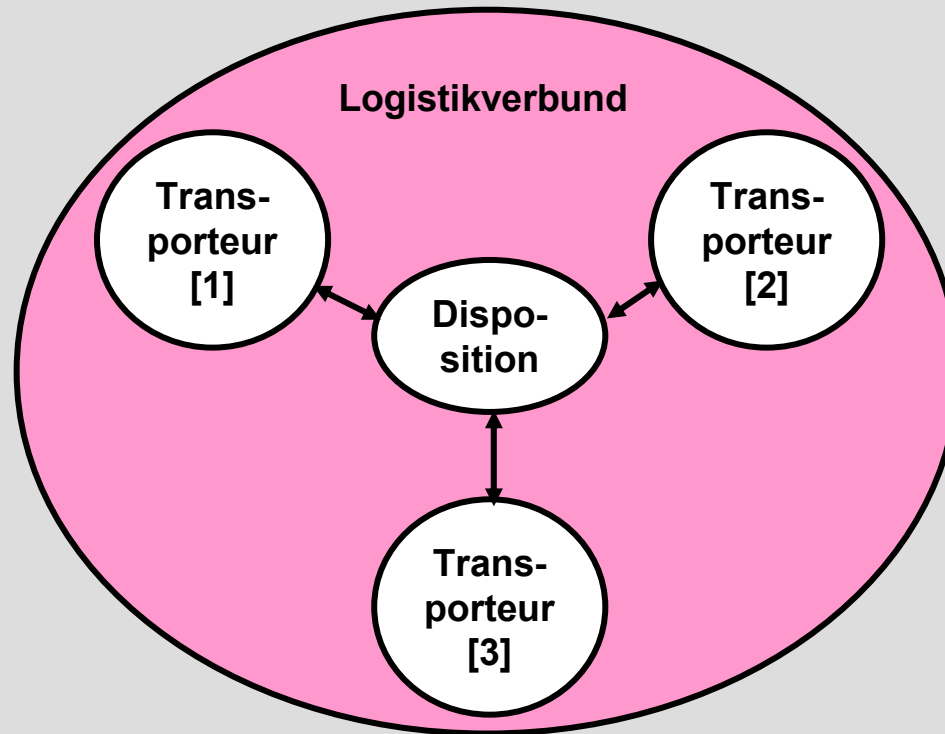
## 3.1.1 Einbettung von Rollen in Rollen

Über-  
geordne-  
tes Element

Sub-Ele-  
ment



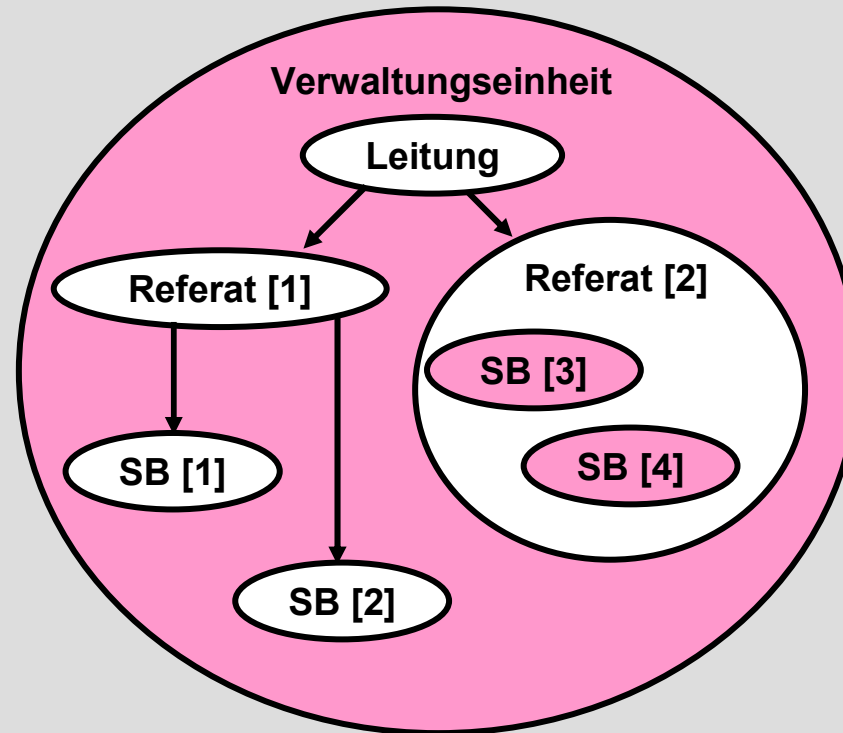
## 3.1.1 Beispiel: Rollen in Rollen



Der Logistikverbund ist sternförmig strukturiert. Die Transporteure regeln ihre Beziehung über einen Disponenten, zu dem sie in gleichrangigem Verhältnis stehen.



## 3.1.1 Beispiel: Rollen in Rollen



Aufg. 7.2.3

Aufg. 7.5.5

Aufg. 8

Die Verwaltungseinheit besteht aus den Hierarchieebenen Leitung, Referat und Sachbearbeiter (SB). Die interne Struktur von Referat [1] ist im Modell nicht näher spezifiziert, es weist SB[1] und SB[2] an, die von ihm unabhängig existieren. SB[3] und SB[4] sind spezielle Rollen, deren Existenz an Referat [2] gebunden ist; sie werden direkt von der Leitung kontrolliert.

...

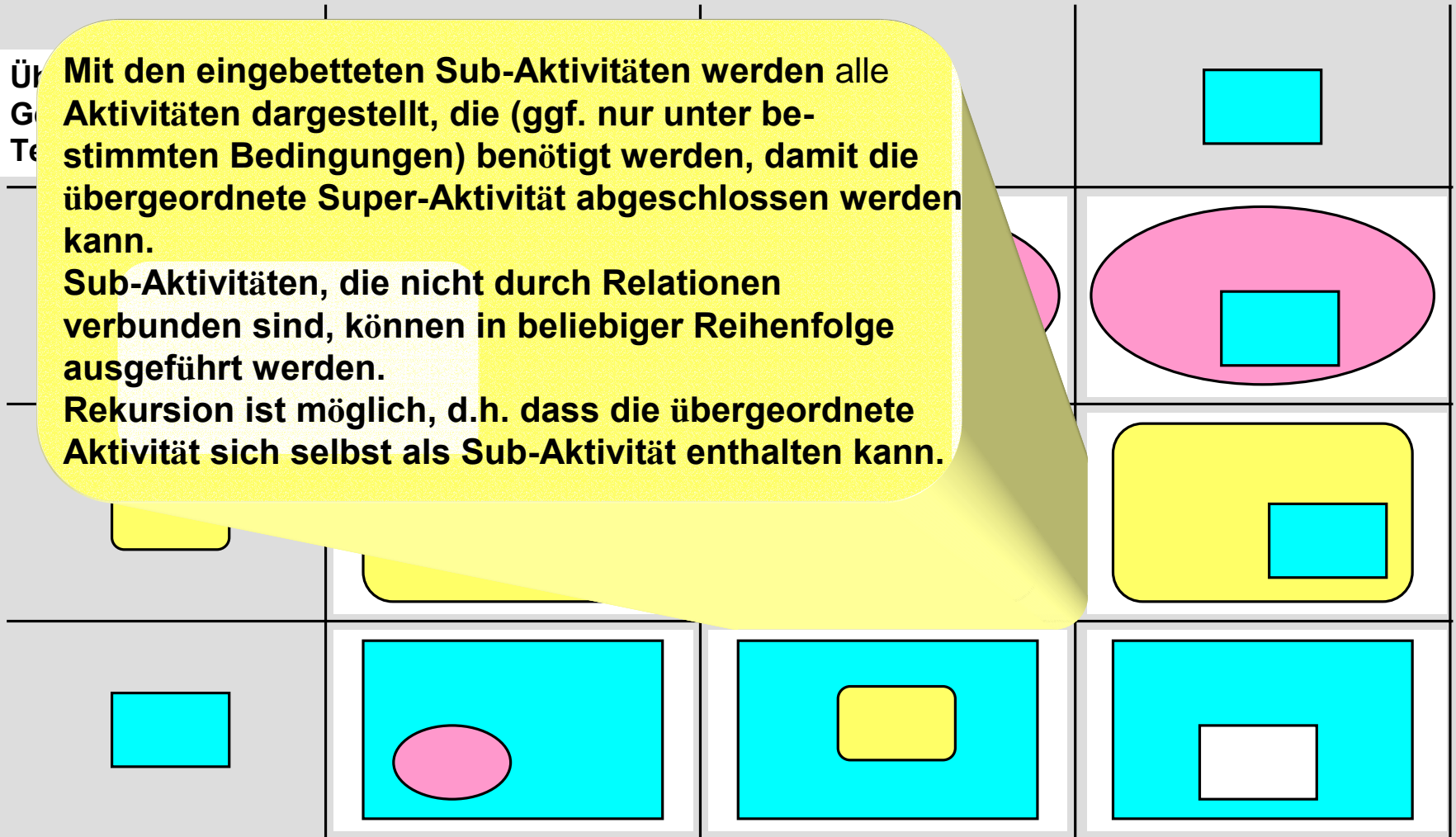


### 3.1.2 Einbettung von Aktivitäten in Aktivitäten

**Mit den eingebetteten Sub-Aktivitäten werden alle Aktivitäten dargestellt, die (ggf. nur unter bestimmten Bedingungen) benötigt werden, damit die übergeordnete Super-Aktivität abgeschlossen werden kann.**

**Sub-Aktivitäten, die nicht durch Relationen verbunden sind, können in beliebiger Reihenfolge ausgeführt werden.**

**Rekursion ist möglich, d.h. dass die übergeordnete Aktivität sich selbst als Sub-Aktivität enthalten kann.**





## 3.1.2 Beispiel: Aktivitäten in Aktivitäten

### Umfassende Nutzung elektronischer Medien

Kommunikation

Kooperation

Koordinieren

Informations-  
verteilung

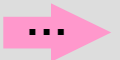
Informationretrieval

Suchen

Finden

Nachvollziehen der Kommunikation und Kooperation

Die Abbildung zeigt die Aktivitäten, die in der Regel zu einer umfassenden Nutzung elektronischer Medien gehören. Die Durchführung dieser Aktivitäten kann verschränkt oder sequenziell erfolgen (dies entscheiden hier die ausführenden Rollen), nur zwischen „Suchen“ und „Finden“ gibt es eine vorgegebene Abfolge.



## 3.1.3 Einbettung von Entitäten in Entitäten

Über-  
geordne-  
tes Element

Sub-Ele-  
ment

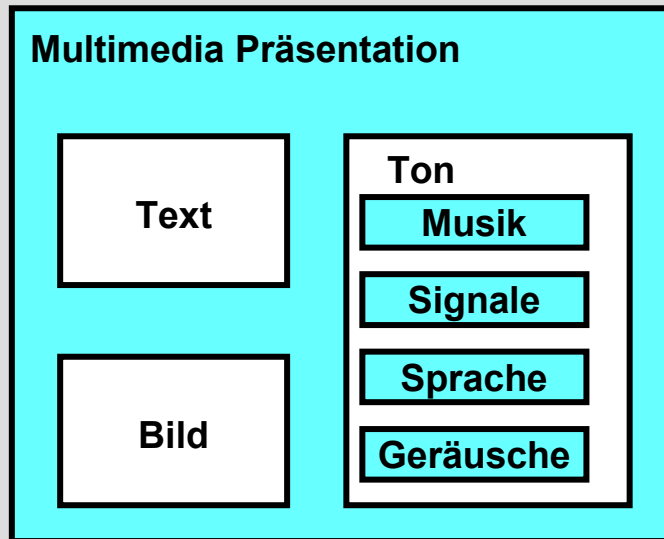
Entitäten bettet man in eine Entität ein, wenn ein enger semantischer oder pragmatischer Zusammenhang zwischen den beiden besteht und die Sub-Entität nicht mit dritten Entitäten im selben Zusammenhang steht.

Die Eigenschaft der übergeordneten Entität sollte die Einbettung der Sub-Entitäten rechtfertigen. Sie kann z.B. aus den Sub-Entitäten zusammengesetzt sein, sie als Container enthalten, in die Sub-Entitäten unterteilt sein. Die Sub-Entitäten können auch Kategorien der übergeordneten Entität repräsentieren.



## 3.1.3 Beispiel: Entitäten in Entitäten

### Multimedia Präsentation

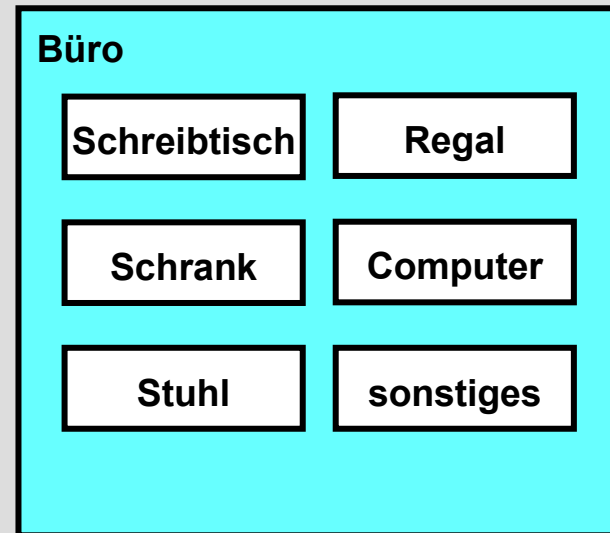


Eine vollständige multimediale Präsentation setzt sich aus den *Darstellungsarten* Text, Bild und Ton zusammen. Den Ton kann man nach den Kategorien Musik, Signale, Sprache und Geräusche differenzieren.

Aufg. 3.6

Aufg. 7.1.6

### Büro



Ein Büro ist ein Ort, an dem sich ein Schreibtisch, ein Regal, ein Schrank, Stuhl und ein Computer befinden sowie sonstige Gegenstände (Pflanzen, Tafel etc.). Das Büro ist somit auch Container, in dem sich etwas befindet. Hier geht es weniger darum, woraus ein Büro besteht.

...



## 3.1.4 Einbettung von Aktivitäten in Rollen

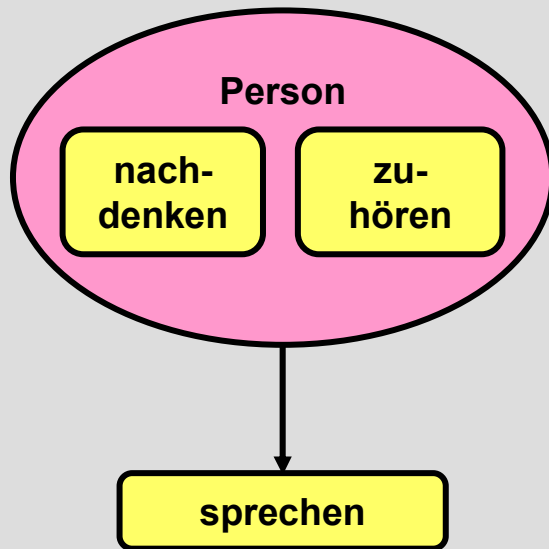
Über-  
geordne-  
tes Element

Sub-Ele-  
ment

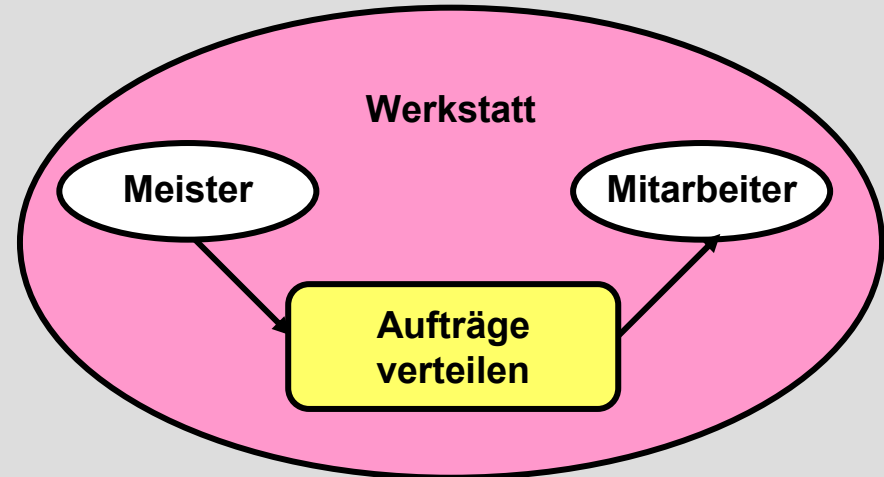
Es ist sinnvoll, eine Aktivität in Rollen einzubetten, wenn diese eine organisatorische Einheit repräsentieren, deren Fortbestand oder Hauptaufgaben durch diese Aktivität gewährleistet wird (z.B. Teambesprechung in Team einbetten).

Aktivitäten werden insbesondere eingebettet, wenn sie nur von dieser Rolle ausgeführt werden und auch nur diese beeinflussen (z.B. Person und Denken).

## 3.1.4 Beispiel: Aktivitäten in Rollen



Während das Sprechen einer Person nach außen wirkt, werden *nachdenken* und *zuhören* von ihr selbst ausgeführt und sie beeinflussen auch nur diese Person selbst – deshalb werden diese beiden Aktivitäten eingebettet.



Wenn die Verteilung der Aufträge in einer Werkstatt nicht von außen beeinflusst wird und auch im Außenverhältnis keine Rolle spielt (one face to the customer), dann ist es sinnvoll diese Aktivität einzubetten.

Aufg. 4.2

Aufg. 7.1.6

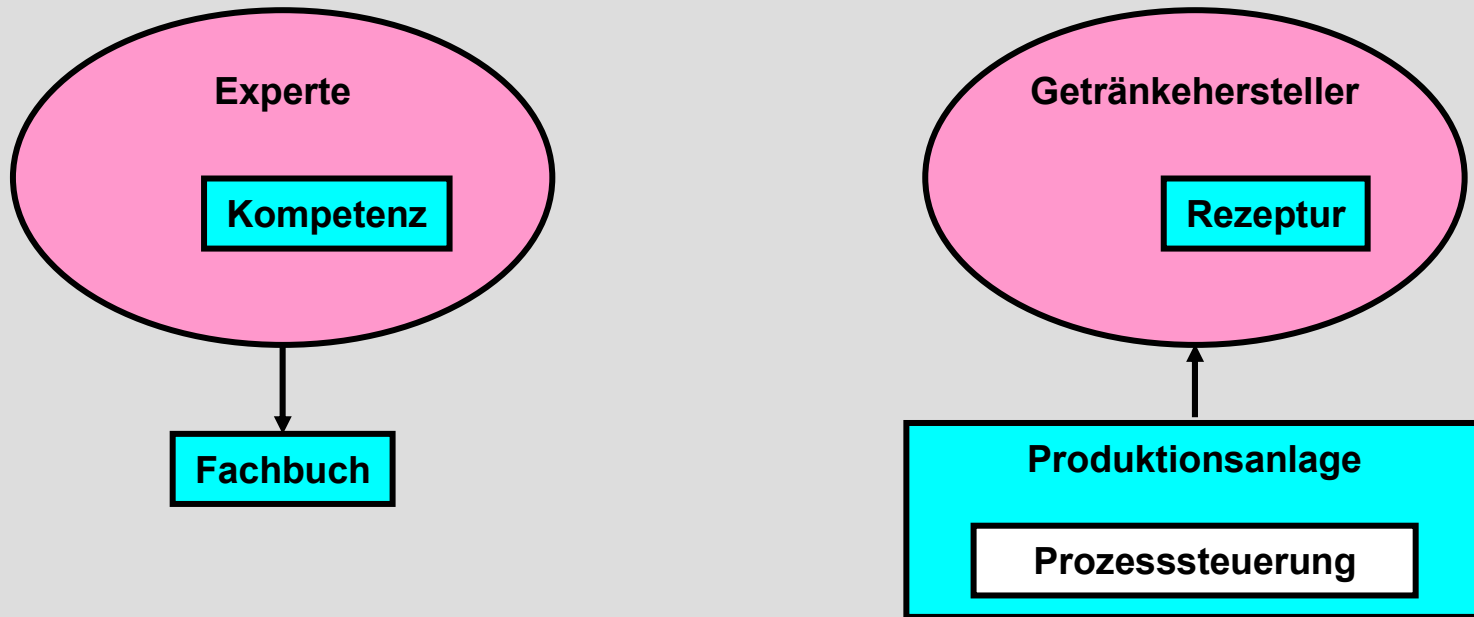
## 3.1.5 Einbettung von Entitäten in Rollen

Über-  
Geordne-  
Tes Element

Sub-Ele-  
ment



## 3.1.5 Beispiel: Entitäten in Rollen

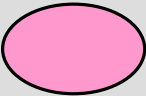




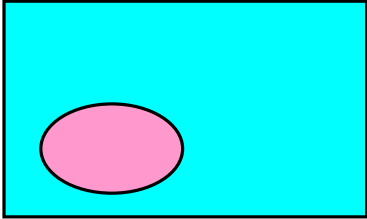
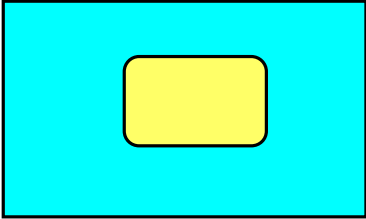
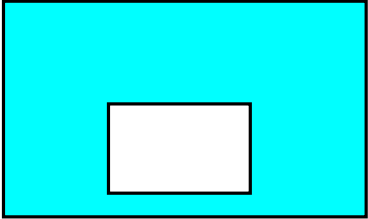


Während ein Fachbuch, das die Kompetenz eines Experten beschreibt, auch von anderen wahrgenommen werden kann, gilt dies für die unmittelbar mit der Person verbundenen Handlungskompetenz des Experten nicht. Sie ist als *interne* Entität eingebettet

Während die Prozesssteuerung der Produktionsanlage eines Getränkeherstellers von außen (etwa für das Wartungspersonal) zugänglich ist, gilt dies für die Rezeptur nicht – sie wird eingebettet dargestellt. Die Anlage ist so konzipiert, dass die Rezeptur im Wartungsfall gelöscht und danach neu eingelesen wird.



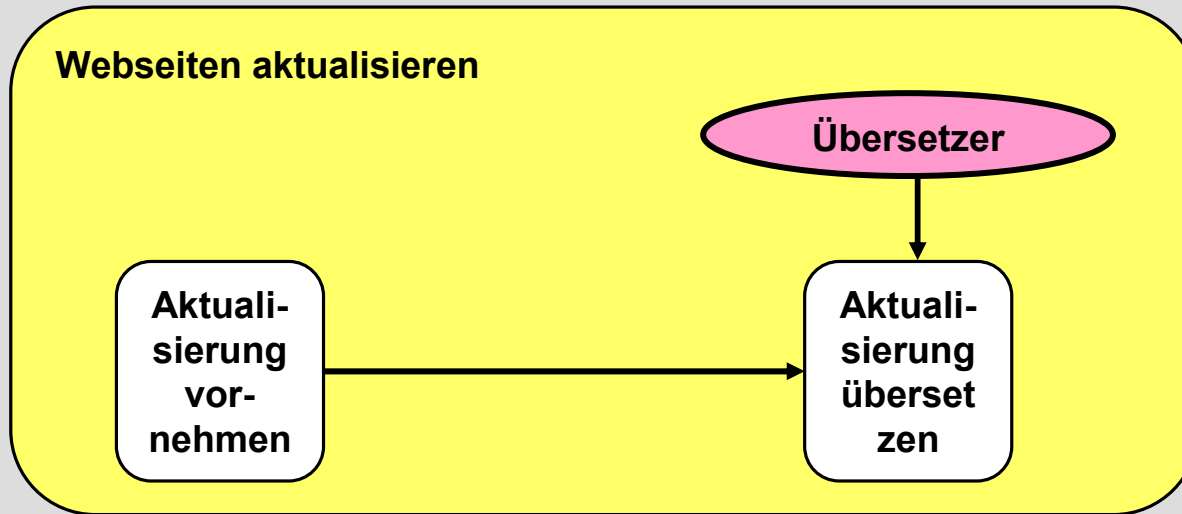
## 3.1.6 Einbettung von Rollen in Aktivitäten

Über-Geordneter Element	Sub-Element
	
	
	  

Eine Rolle sollte in eine Aktivität eingebettet werden, wenn sie nur existiert, um diese Aktivität auszuführen oder in einer sonstigen engen semantischen Verbindung mit dieser Aktivität steht oder wenn sie für alle anderen Aktivitäten eines Modells irrelevant ist.



## 3.1.6 Beispiel: Rollen in Aktivitäten



Wenn für das Übersetzen aktualisierter Webseiten ein Übersetzer aktiv wird, der sonst in einem Diagramm nirgends eine Rolle spielt, dann kann es angemessen sein, diese Rolle in die Aktivität einzubetten. Dies bietet auch die Möglichkeit, die Rolle auszublenden (→ [Kapitel 8](#)) und so ein komplexes Diagramm zu vereinfachen.



## 3.1.7 Einbettung von Entitäten in Aktivitäten

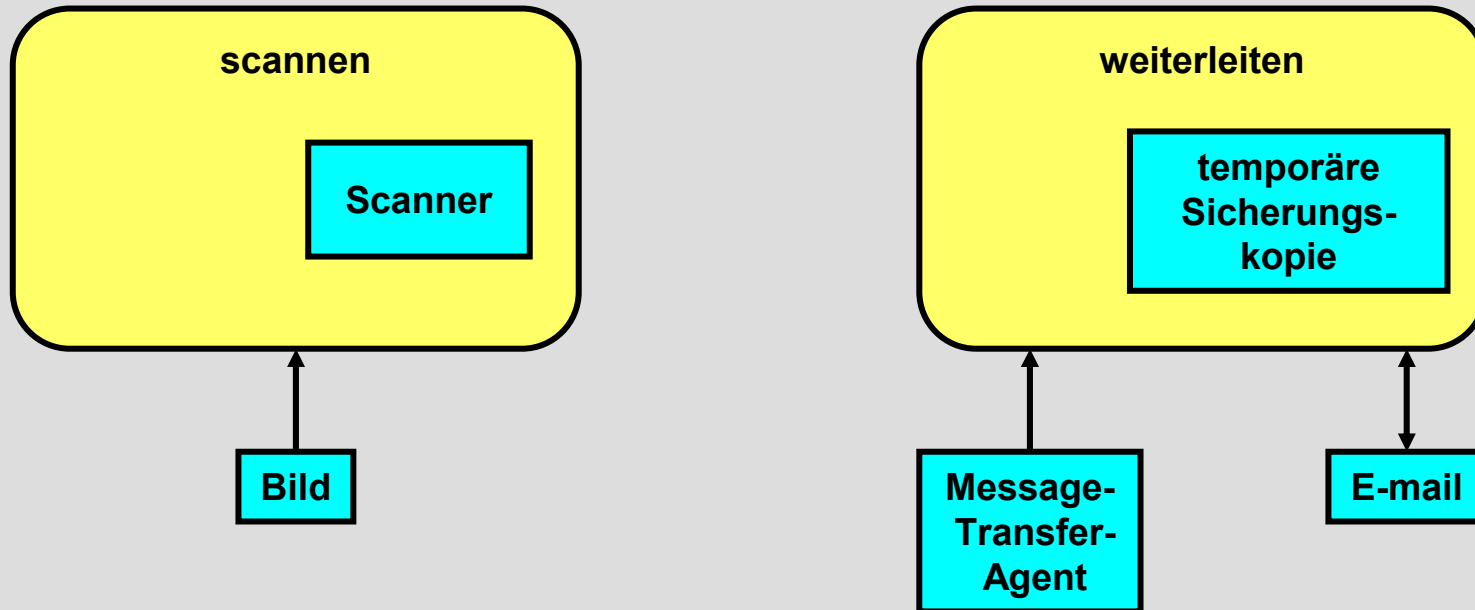
Über-  
Geordne-  
Tes Element

Su

Eine Entität sollte in eine Aktivität eingebettet werden, wenn sie ausschließlich zur Durchführung dieser Aktivität benötigt wird oder ausschließlich von dieser verändert wird und wenn keine anderen Aktivitäten auf sie zugreifen, oder wenn sie in sonstiger enger semantischer Verbindung mit der Aktivität steht (z.B. Schreiben und Schreibmaschine).



## 3.1.7 Beispiel: Entitäten in Aktivitäten



Der Scanner wird nur für die Aktivität des Scannens benötigt und daher in diese eingebettet, während das eingescannte Bild wahrscheinlich auch für andere Aktivitäten eine Rolle spielt.

Der Message-Transfer-Agent erstellt aus Sicherheitsgründen beim Weiterleiten eine temporäre Sicherungskopie, die gelöscht wird, nach dem die erfolgreiche Weiterleitung bestätigt wurde.



## 3.1.8 Einbettung von Rollen in Entitäten

Über-  
Geordne-  
Tes Element

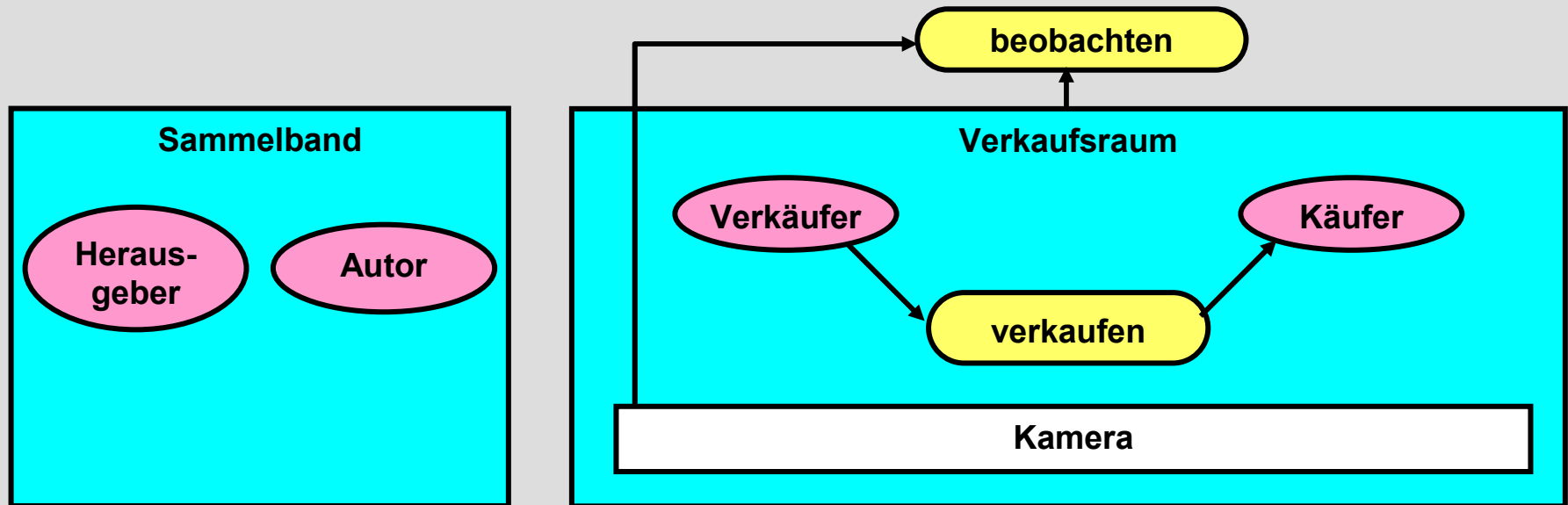
Sub-Ele-  
ment

Eine Rolle bettet man zum einen in eine Entität ein, wenn sie nur mit dieser Entität in Zusammenhang steht und es ist nur die Entität (nicht die Rolle), die für die restlichen Elemente des Models relevant ist.

Zum anderen ist eine solche Einbettung sinnvoll, wenn die Entität einen Raum oder Ort repräsentiert (z.B. Büro, Schiff, Krankenhaus) innerhalb dessen die Rolle gemäß ihrer Definition aktiv ist.



## 3.1.8 Beispiel: Rollen in Entitäten



Ein Sammelband hat Herausgeber und Autoren, wenn diese Rollen nur für die Entität Sammelband relevant sind, sollte man sie einbetten.

Wenn bestimmte Rollen eines Modells (Verkäufer und Käufer) nur innerhalb eines bestimmten Raumes (Verkaufsraum) in ihrer Rolle aktiv werden, so kann der Raum als Entität dargestellt werden, in den die Rollen eingebettet werden. Dies kann auch sinnvoll sein, wenn diese Rollen von außen beeinflusst werden.

**Aufgabe**



## 3.1.9 Einbettung von Aktivitäten in Entitäten

Über-  
Geordne-  
Sub-Ele-  
ment

Das Einbetten von Aktivitäten in eine Entität ist sinnvoll, wenn sie ausschließlich genutzt werden, um diese Entität zu verändern oder wenn die Aktivität ausschließlich von dieser Entität ausgeführt wird. Letzteres ist insbesondere der Fall, wenn Aktivität die Zustandsänderungen eines Computersystems beschreibt. Diese Art der Einbettung macht auch Sinn, wenn ein enger, ausschließlicher Zusammenhang zwischen Entität und Aktivität besteht.



## 3.1.9 Beispiel: Aktivitäten in Entitäten

### Computergestützter Terminkalender

automatisches Erinnern

automatisch Verknüpfungen vorschlagen

Das Einbetten von Aktivitäten ist insbesondere sinnvoll, wenn sie als Leistung eines technischen Systems von diesem automatisch erbracht werden, wie hier die Alarmfunktion eines Terminkalenders oder automatisch erfolgende Vorschläge, die neue Einträge mit bereits vorhandenen Items verknüpfen.

### Pendel

schwingen

Die Aktivität *Schwingen* wirkt nur auf den Zustand des Pendels und sonst auf nichts. Sie ist reflexiv und wird deshalb eingebettet.



## 3.1.10 Aufgaben zu Einbettung

1. Stellen Sie durch Einbettung von Rollen eine Matrix-Organisation eines Unternehmens dar
2. Stellen Sie die Rolle „Werkstatt“ aus [3.1.4 \(Beispiel\)](#) als Entität dar und erläutern Sie das dann entstehende Diagramm.
3. Erstellen Sie zwei Diagramm, in denen einmal ein Krankenhaus als Rolle und beim anderen mal als Entität vorkommt. Ordnen Sie jeweils Ärzte, Pfleger Patienten, Administration etc. ein.
4. Modellieren Sie einen Chatroom mit zugehörigen Aktivitäten und Rollen und entscheiden Sie, welche Sie einbetten und welche Sie außerhalb des Chatrooms einordnen.
5. Geben Sie ein weiteres Beispiel für die Einbettung einer Rolle in eine Aktivität.

**Aufg.**



## 3.2 Einbettung von Attributen

Um Basis-Elemente näher zu beschreiben, kann man →Attribute (s. Kapitel 6) definieren und sie in das Basis-Element einbetten.

Attribute werden als Text dargestellt. Ein Attribut besteht aus einem Attributnamen und dem bzw. den Werten, die das Attribut bezogen auf die Entität hat. Es gibt Standard-Attribute, z.B. NAME.

Man hat die Wahl, ob man die Eigenschaften eines Basis Elementes mittels eines Attributs oder durch Relationen zu anderen Basis-Elementen darstellt.

**Attributname: Wert(e)**



## 3.2 Beispiel: Einbettung von Attributen

**Person;**  
**Alter:** 20-30;  
**Beruf:** Kaufmann  
**Familienstand:**

Hier geht es um die Klasse aller Personen, die zwischen 20 und 30 Jahre alt sind, die von Beruf Kaufmann sind und deren Familienstand beliebig ist.

**schreiben;**  
**Dauer:** >2 Stunden;  
**Schriftart:** {Times, Arial};  
**Schreibinstrument:**  
Computer

Klasse aller Aktivitäten des Schreibens, die länger als 2 Stunden dauern, die Schriftart Times und/oder Arial benutzen und mit Hilfe eines Computers ausgeführt werden.

**Bild;**  
**Art:** {statisch, bewegt};  
**Farbe:** {ja, nein};  
**Entstehungsart:** {Abbild,  
Phantasie}

Klasse aller Bilder, die entweder statisch und/oder dynamisch sind, Farbe beinhalten oder nicht und als Abbild oder Phantasieprodukt entstanden sind.



## 3.2 Abkürzung: Namens-Attribute

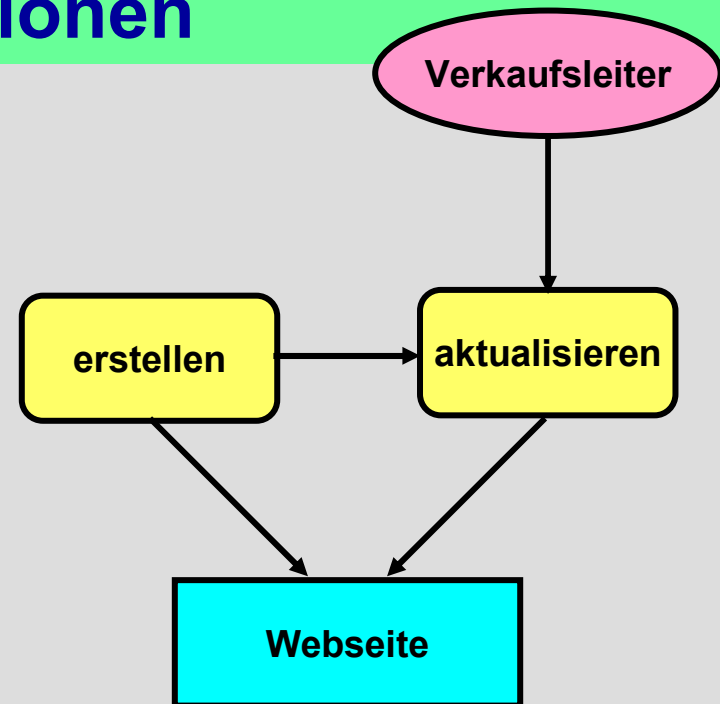


Es gibt Standardattribute, wie NAME, auf grund deren Einbettungsposition (NAME steht immer an erster Stelle), klar ist, um welches Attribut es sich handelt. In diesen Fällen muss nur der Wert des Attributs angegeben werden.



## 3.2 Beispiel: Einbettung von Attributen vs. Basis-Elemente und Relationen

Webseite aktualisieren;  
Vorangehende Aktivität: Seite  
erstellen;  
Zuständiger Bearbeiter:  
Verkaufsleiter



Die linke und die rechte Darstellung sind bzgl. des Aussagegehalts äquivalent. Der Formalisierungsgrad ist auf der rechten Seite höher. Außerdem ist es hier möglich, eine Erweiterung vorzunehmen, z.B. eine Rolle an die Aktivität „erstellen“ anzu-  
binden. Demgegenüber hat die linke Darstellung den Vorteil, Platz zu sparen.



## 3.3 Segmentierung von Elementen

Falls eine große Zahl von Elementen oder auch Attributen in ein Basis-Element eingeordnet werden, kann es sinnvoll sein, diese zu ordnen.

Zu diesem Zweck kann man ein Basis-Element durch waagrechte Striche in Segmente zerlegen.

Der Modellierer ist frei in der Entscheidung, wie er das Ordnungsprinzip wählt, mit dem er die Sub-Elemente auf die Segmente verteilt. Ein typisches Ordnungsprinzip, das bei der Modellierung von Klassen genutzt wird, ist eine Dreiteilung in *Name*, *Attribute*, *Methoden*.

Man kann Segmente auch nutzen, um die Menge der Sub-Rollen, Sub-Aktivitäten und Sub-Entitäten getrennt darzustellen.

Außerdem kann man in verschiedenen Segmenten verschiedene **Perspektiven** bzw. Möglichkeiten darstellen, wie man ein Basis-Element in Sub-Elemente untergliedern kann.

Segmente können einen Namen haben.



## 3.3 Beispiel: Segmentierung von Elementen I



Hier wird die Unterteilung mit Hilfe von Segmentlinien genutzt, um Rollen, Dokumente und Randbedingungen (hier als Attribute dargestellt) aufzulisten, die bei der Aktivität „Wohnung vermieten“ relevant sind. Es kann Relationen geben, muss es aber nicht. Relationen können auch Segmentgrenzen überschreiten (Grundbuchauszug gehört Hausbesitzer).

... **Aufgabe**

## 3.3 Beispiel: Segmentierung von Elementen II



Die drei unteren Segmente stellen drei verschiedene Perspektiven dar, unter denen die Klasse Haus zerlegt werden kann.

Die erste Perspektive befasst sich mit der räumlichen Einteilung,

Die zweite mit dem Material, aus dem ein Haus besteht. Die erste und zweite Perspektive kann man als Aggregationsperspektiven verstehen (aus welchen Teilen besteht das Haus).

Als abstrakte Klasse präsentiert Haus die Menge sämtlicher Häuser. Zur Charakterisierung dieser Menge gehören auch die unterschiedlichen Kategorien von Häusern. Sie sind im dritten Segment dargestellt. Hier sieht man auch, dass Perspektiven nicht immer einen Namen haben müssen.

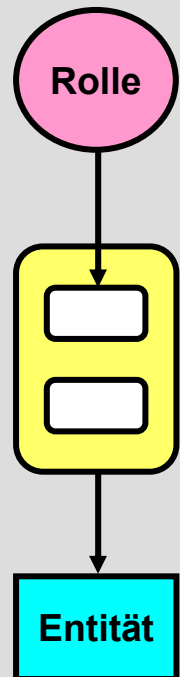
Die Attribute, die unter dem Namen stehen, beziehen sich auf die Entität, als Ganzes und nicht auf die Segmente. Für „Baustoffe“ macht das Attribut „Grundfläche keinen Sinn. Innerhalb eines Segmentes können Attribute spezifiziert werden, die genau zu ihm passen.

## 3.4 Einbettung und Relationen

**Relationen, die mit dem Rand eines Basis-Elementes verbunden sind, beziehen sich in ihrer Bedeutung immer auch auf alle Sub-Elemente dieses Basis-Elementes, sofern sie den gleichen Typ haben.**

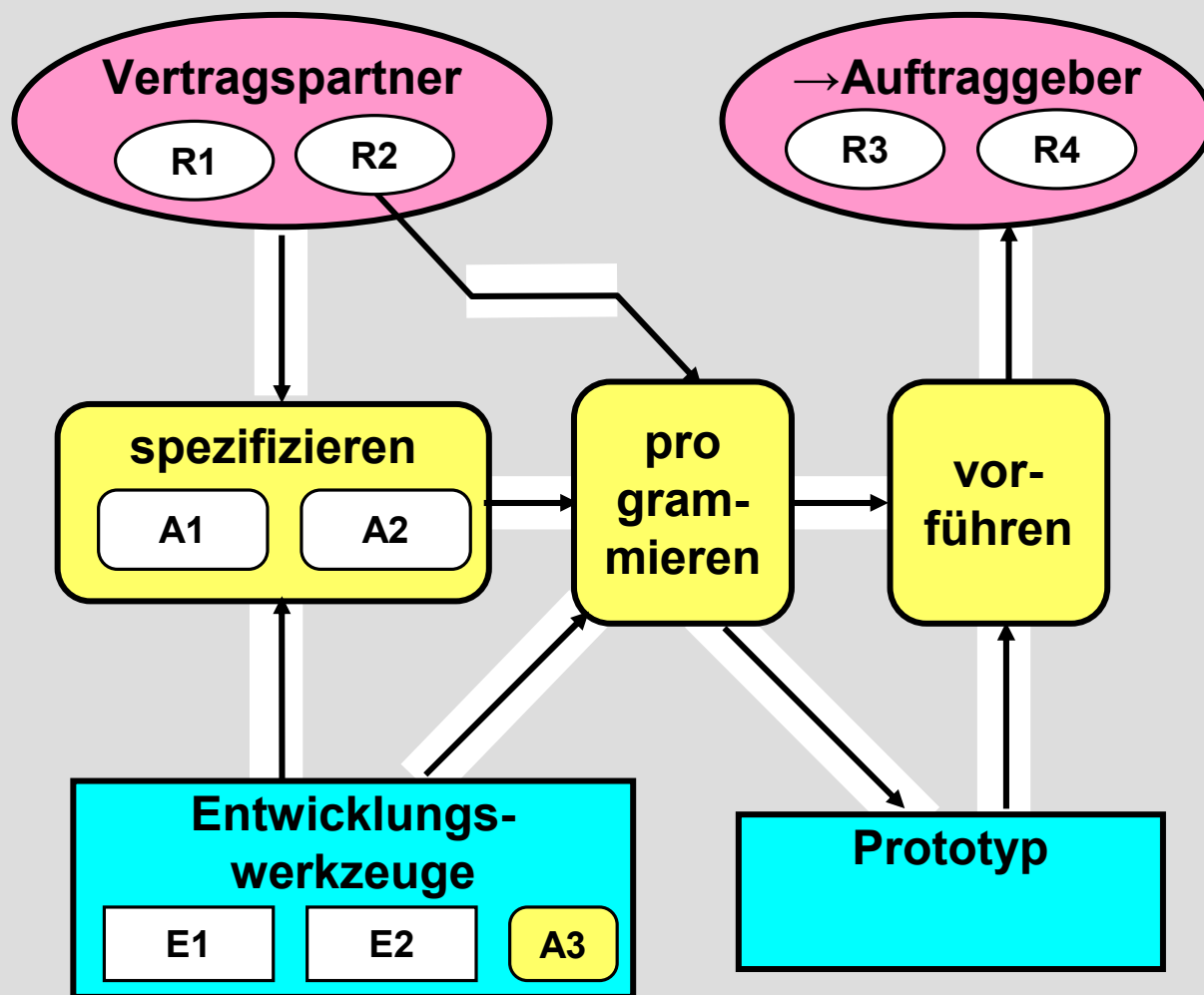
**Eine Relation kann den Rand des Basis-Elementes auch schneiden, um direkt mit einem Sub-Element beliebigen Typs verbunden zu sein. Sie ist zwar dann auch mit dem Basis-Element verbunden, aber nur in dem Sinne, dass sie sich in ihrer Bedeutung auf das ausgewählte Sub-Element dieses Basis-Elements bezieht.**

**Wenn man nicht angeben oder entscheiden will, ob sich eine Relation auf ein gesamtes Basis-Element oder nur auf einen Teil seiner Sub-Elemente bezieht, dann lässt man den Start- oder Endpunkt der Relation innerhalb des Basis-Elementes unspezifiziert ([→s. Kapitel 7.2](#))**





## 3.4 Beispiel: Relationen und Einbettung



Die Anordnung der Relationen drückt aus, dass alle Rollen R1 und R2 an allen Aktivitäten (A1, A2) des Spezifizierens teilnehmen. Alle Entwicklungswerkzeuge (E1, E2) werden genutzt. Alle Sub-Rollen (R3, R4) des Auftraggebers sind von der Vorführung beeinflusst. R2 ist allerdings der einzige Vertragspartner, der programmiert. A1 und A2 – nicht jedoch A3 – müssen beide ausgeführt sein, bevor das Programmieren beginnt.

**Aufgabe**

## 3.5 Erweiterte Aufgaben zu Einbettung

1. Fügen Sie Attribute zu den gegebenen Beispielen für Rolle und Aktivität ein.
2. Stellen Sie die Aktivität „Wohnung vermieten“ in Beispiel 3.3 ausschließlich mit Hilfe von Attributen dar.
3. Spezifizieren Sie verschiedene Perspektiven für die Entität „Multimediale Darstellung“
4. Ersetzen Sie R1, R2, R3, R4, A1, A2, E1 und E2 in Beispiel 3.5 durch konkrete Beispiele



## 4) Spezifizierung von Relationen

Neben den in 2) beschriebenen Standardrelationen kann man bei Bedarf spezielle Typen von Relationen definieren, die man dann entsprechend auszeichnen muss. Ferner kann man Rollen, Aktivitäten oder Entitäten angeben, die die Instantiierung einer Relation unterstützen (4.2). Spezielle Relationen, die immer wieder auftreten, werden besonders dargestellt (4.3). Z. B. ist die Einbettung eine solche spezielle Relation.

### Gliederung

4.1 Besondere Typisierung von Relationen

4.2 Spezifizierung von Relationen durch Rollen, Aktivitäten und Entitäten

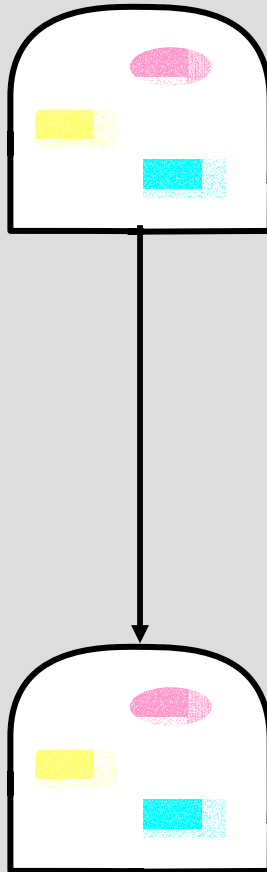
4.3 Spezielle Relationen

- Vererbung
- Aggregation
- Meta-Relation
- Einbettung

4.4 Aufgaben



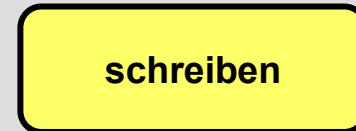
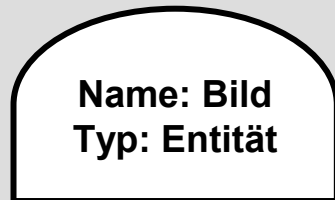
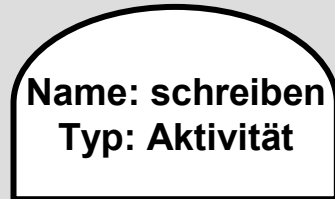
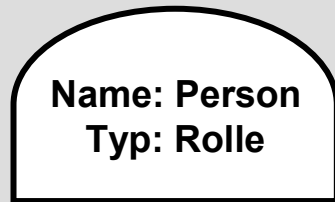
## 4) Spezifizierung von Relationen - Hilfsmittel



Die folgenden Aussagen über Relationen gelten meistens für alle Arten von Relationen unabhängig von der Art der miteinander verbundenen Basis-Elemente. Wir führen daher ein Meta-Element ein, das alle drei Typen von Basis-Elementen gemeinsam repräsentiert.



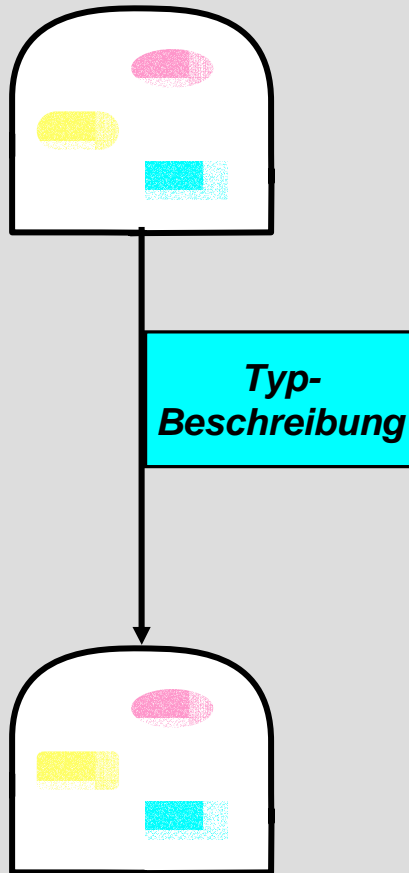
## 4) Abkürzung: Namens-Attribute



Neben dem Standardattribut NAME kann man sich ein weiteres Standardattribut TYP vorstellen, wobei die Typangabe durch Farbe und Form ersetzt werden kann. Im Bedarfsfall können über das Typ-Attribut neue Arten von Elementen definiert werden. Wenn das weiße Halbkreissymbol leer ist, dann bedeutet es, dass die Typangabe unspezifiziert ist, das Symbol repräsentiert also jede mögliche Art von Basis-Elementen.

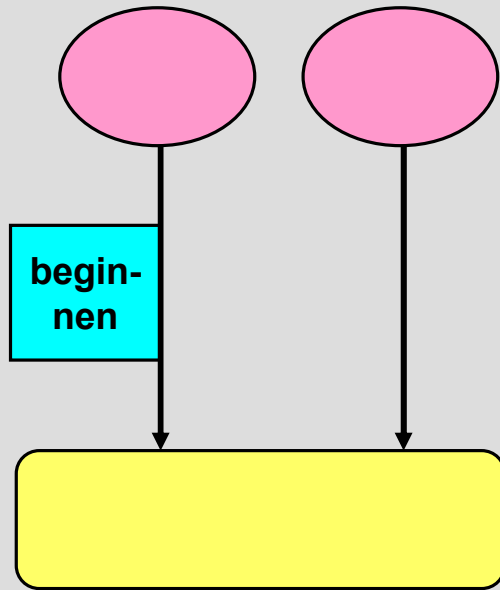


## 4.1 Besondere Typisierung

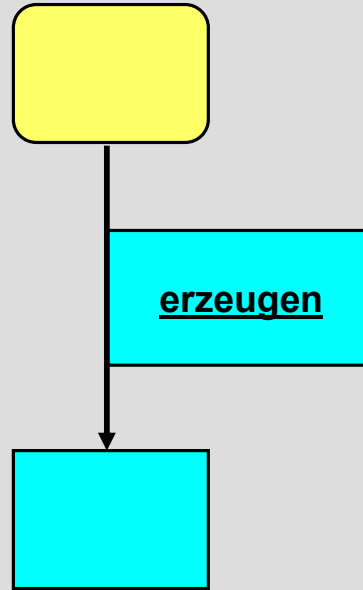


Durch das Anhängen einer Entität kann der Typ der Relation in Abweichung von der Standard-Bedeutung festgelegt werden.

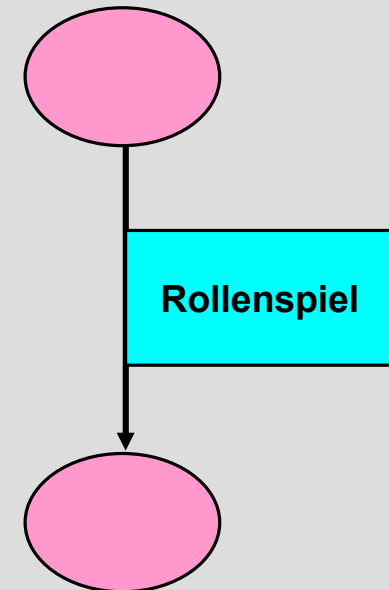
## 4.1 Besondere Typisierung – Beispiele I



Das Beginnen einer Aktivität ist u.U. besonders zu betonen, z.B. wenn sie von anderen Rollen beendet wird.

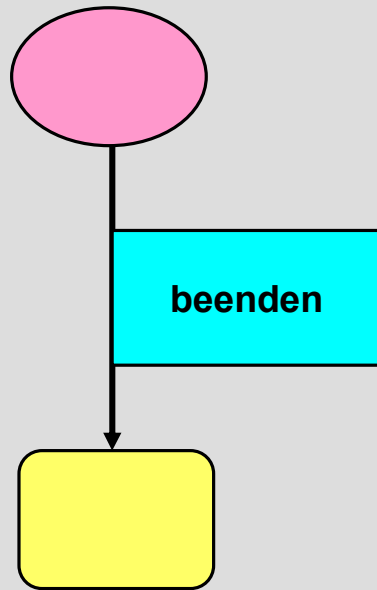


Wenn eine Aktivität eine bislang nicht existierende Entität erzeugt, kann dies durch eine gesonderte Relation ausgedrückt werden.

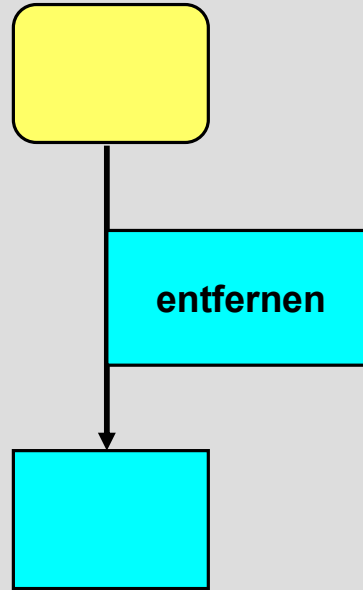


Eine besondere Relation drückt aus, dass z.B. eine Person o.a., die eine Rolle wahrnimmt, auch eine andere spielen kann.

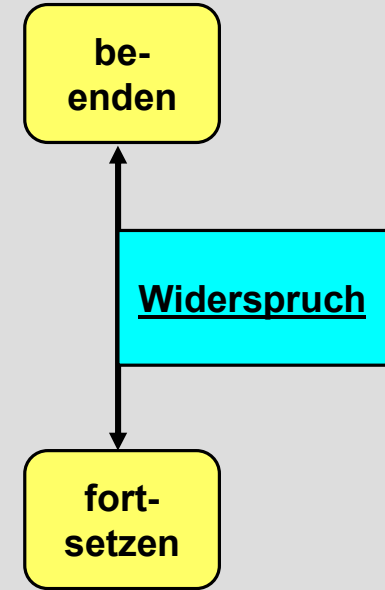
## 4.1 Besondere Typisierung – Beispiele II



Auch das Beenden einer Aktivität kann durch eine besondere Relation ausgedrückt werden.



Es kann auch Aktivitäten geben (z.B. löschen) die eine Entität entfernen.



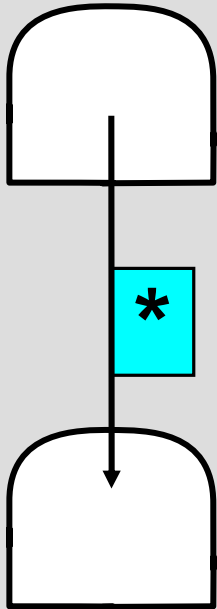
Wenn zwei Elemente in Widerspruch zueinander stehen, kann dies gesondert durch eine Relation ausgedrückt werden.



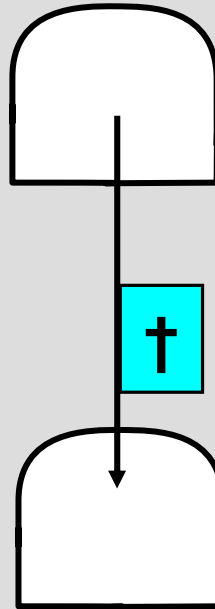


## 4.1 Abkürzungen von Relationstypen

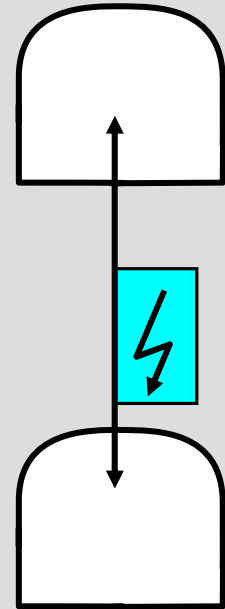
Wenn bestimmte Entitäten zur Spezifizierung von Relationen häufig benötigt werden, kann man den Text durch Symbole ersetzen. Dies ist insbesondere sinnvoll, wenn dadurch Relationen zwischen verschiedenen Typen von Basis-Elementen beschrieben werden.



beginnen,  
erzeugen



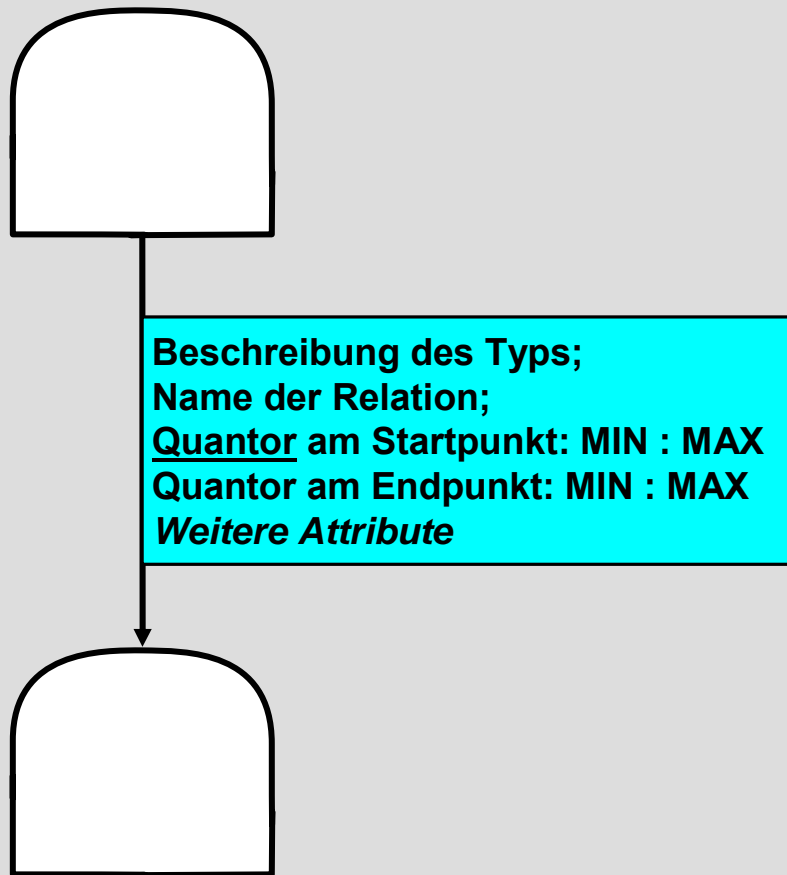
beenden, löschen



Widerspruch



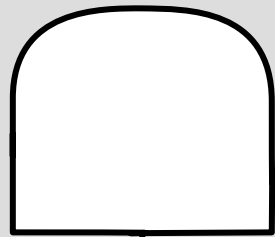
## 4.1 Festlegung von Attributen von Relationen



Die Beschreibung des Typs ist nur ein mögliches von verschiedenen Attributen zur näheren Spezifizierung einer Relation. Wenn die Typbeschreibung nicht festgelegt ist, dann gilt die Standardbedeutung. Weiterhin kann es sinnvoll sein, einen Namen für eine Relation zu spezifizieren (z.B. um Bedingungen festzulegen) oder um die Relation zu quantifizieren. Dabei ist die Angabe eines Minimal- und eines Maximalwertes an beiden Seiten der Relation möglich. Alle Attribute sind optional, sie müssen nicht spezifiziert werden. Es könne noch weitere Attribute nach Bedarf spezifiziert werden.

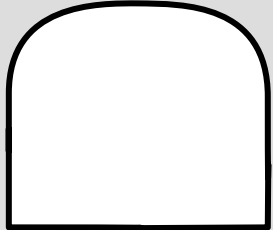
...

## 4.1 Abkürzung von Attributen von Relationen

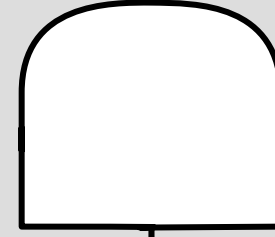


Abkürzung: Die Werte der Quantoren werden direkt an die Enden der Relation geschrieben

Quantor am Startpunkt: 1 : 1  
Quantor am Endpunkt: m : n

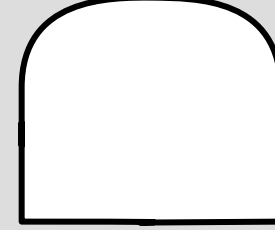


TIPP: SeeMe verwendet als Quantor keine „0“, stattdessen wird das potentiell nicht vorkommende Element mit einem →Modifikator versehen.

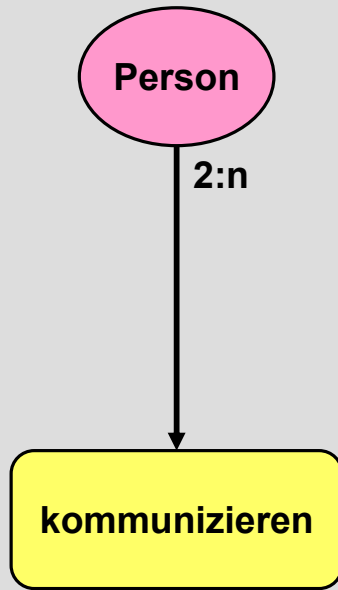


1:1

m:n



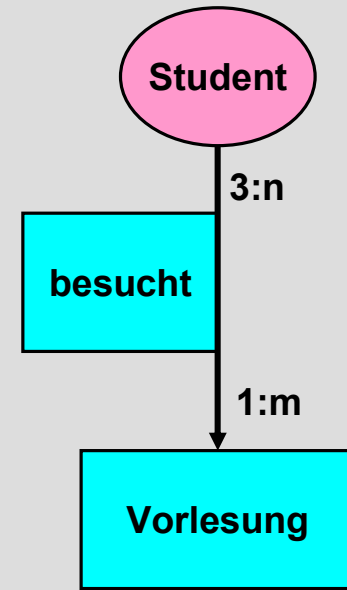
## 4.1 Besondere Typisierung – Beispiele II



Hier wird ausgedrückt, dass es mindestens zweier Personen bedarf, um eine Kommunikation zu ermöglichen.



Zu einem Original kann es eine oder mehrere Kopien geben.

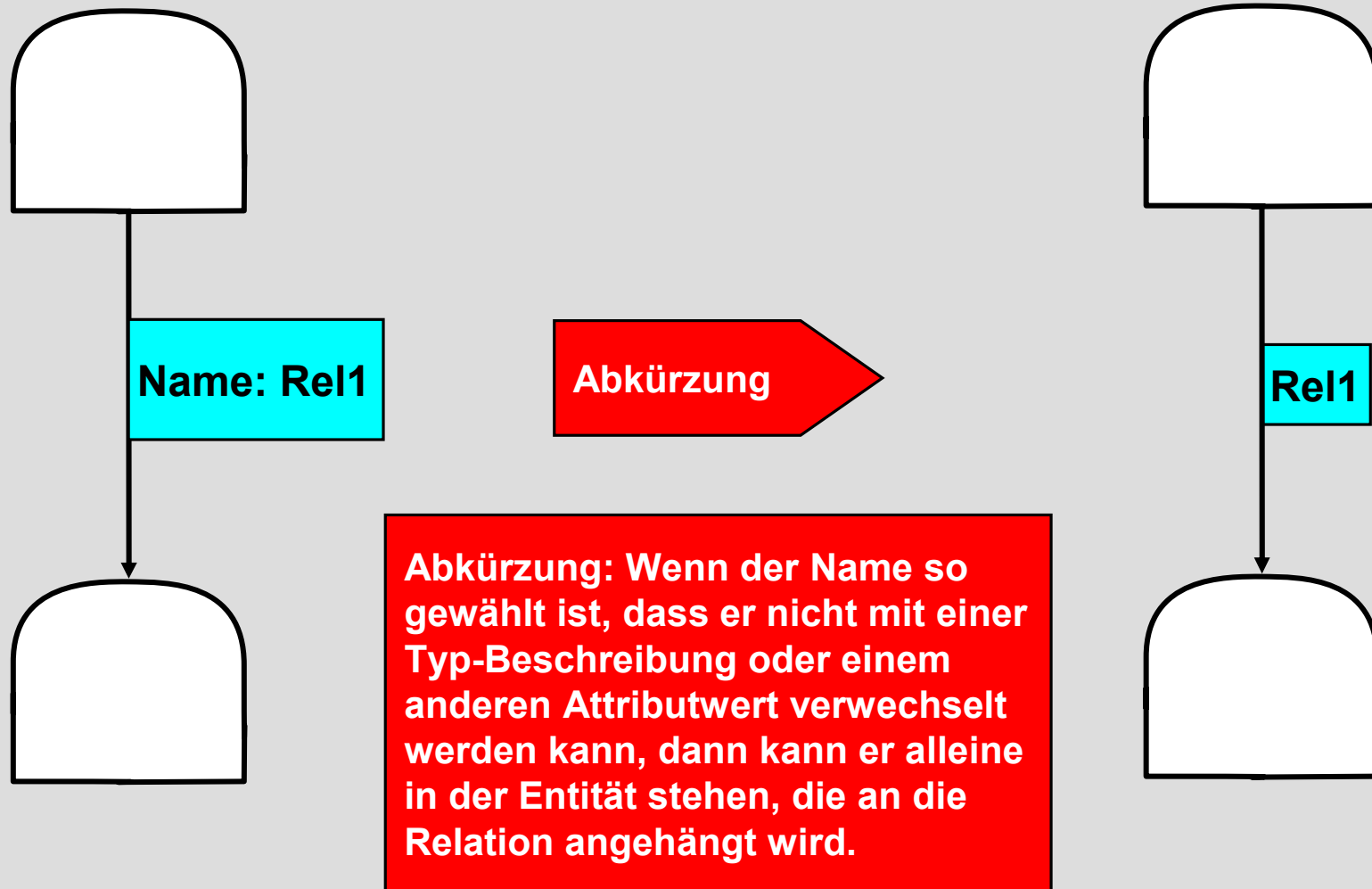


Ein Student kann eine oder bis zu m Vorlesungen besuchen. In einer Vorlesung sind mindestens drei oder bis zu  $n > 3$  Studenten.

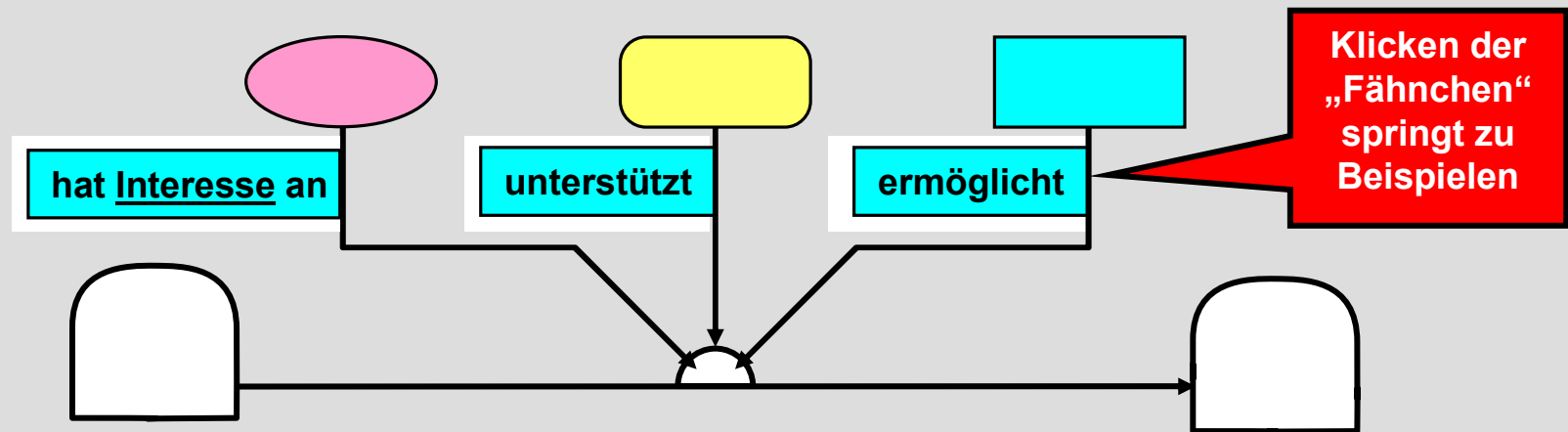
...



## 4.1 Abkürzung von Attributen von Relationen



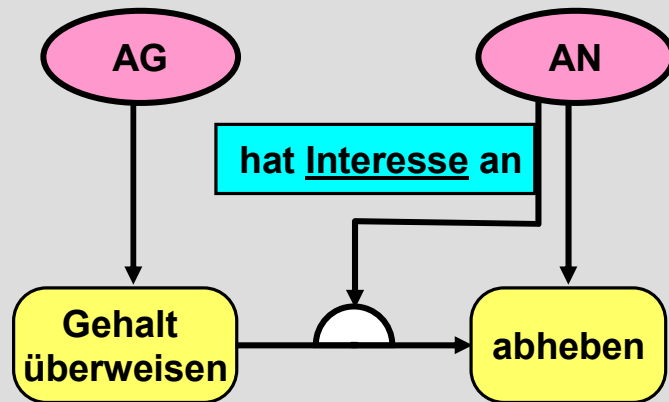
## 4.2 Spezifizierung von Relationen durch Basis-Elemente



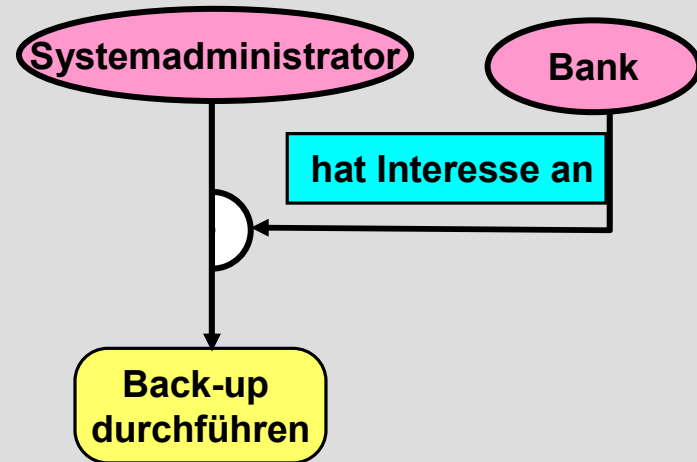
**Basis-Elemente können mit Relationen an Relationen annotiert werden, indem sie auf einen Spezifikator zeigen (Halbkreis). Diese Annotations-Relationen sind von speziellem Typ mit folgender Bedeutung:**

- Die annotierte Rolle hat das Interesse, dass der durch die Relation gekennzeichnete Übergang zu Stande kommt, also die Relation instantiiert wird.
- Die annotierte Aktivität unterstützt die Instantierung der Relation aktiv.
- Die annotierte Entität ermöglichen die Instantierung der Relation, indem sie als Resource oder Werkzeug dient.

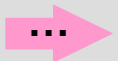
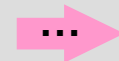
## 4.2 Beispiele zu Spezifizierung von Relationen durch Basis-Elemente



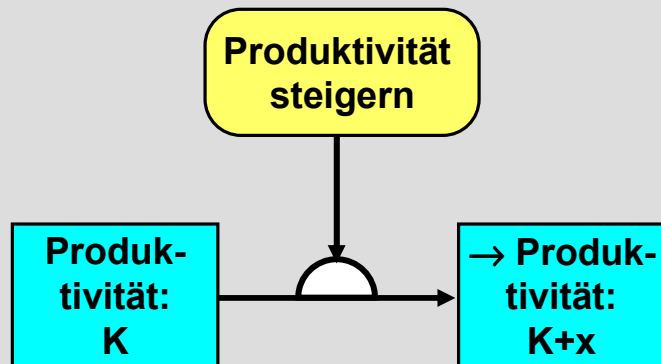
Der Arbeitnehmer (AN) hat das Interesse, dass die Aktivität „Gehalt überweisen“ durch den Arbeitgeber (AG) abgeschlossen wird, bzw. dass die Relation realisiert wird, die letztlich das Abheben des Gehalts ermöglicht.



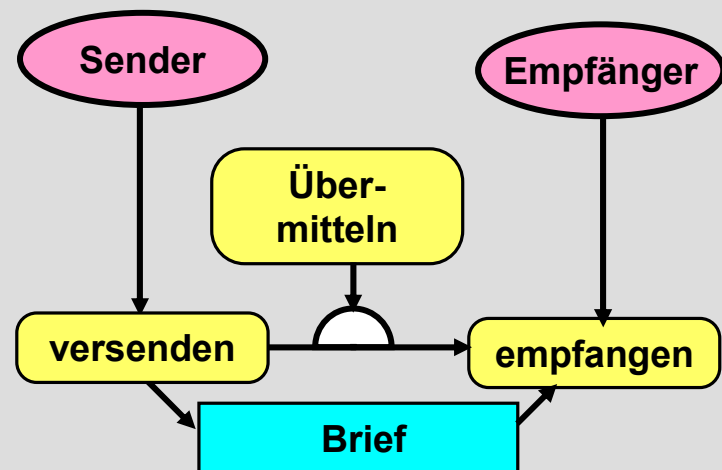
Die Bank hat das Interesse, dass das Back-up der für sie lebenswichtigen Daten durchgeführt wird und zwar speziell durch den Systemadministrator.



## 4.2 Beispiele II zur Spezifizierung von Relationen durch Basis-Elemente



Um den Kennwert K, der den Grad der Produktivität kennzeichnet, zu erhöhen, bedarf es der Aktivität „Produktivität steigern“. Der alte Produktivitätswert K ist in diesem Beispiel dem neue Wert K+x als Vorgänger zugeordnet.

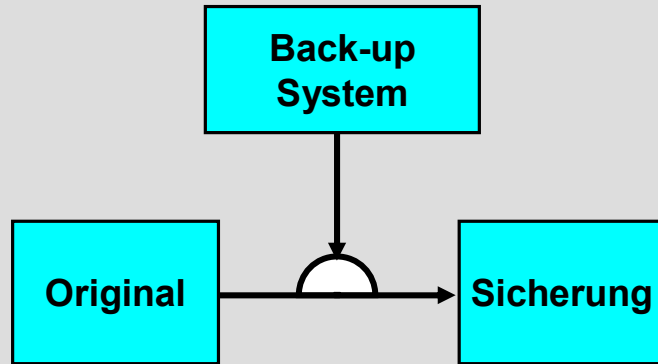


Der Übergang zwischen den Aktivitäten „senden“ und „empfangen“ muss durch eine Aktivität „übermitteln“ unterstützt werden. Dies ist ein häufig auftretendes Muster. Statt der Aktivität könnte auch eine Entität „Transfer-Agent“ eingetragen sein.

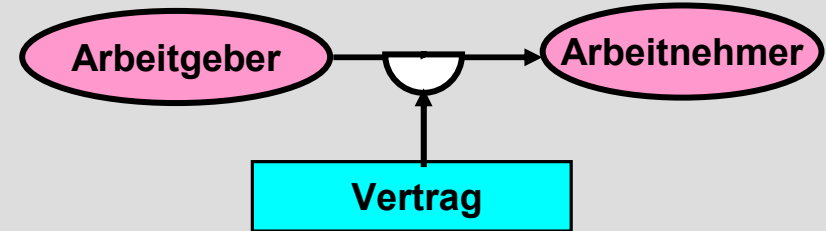




## 4.2 Beispiele III zur Spezifizierung von Relationen durch Basis-Elemente



Das Back-up System wird benötigt, damit die Relation zu einer Sicherungskopie aufgebaut werden kann.



Das Verhältnis zwischen Arbeitgeber und Arbeitnehmer, insbesondere das Weisungsrecht des Arbeitgebers, wird durch einen Vertrag geregelt.



## 4.2 Aufgaben

1. Zerlegen Sie die Aktivität „Aufträge verteilen“ in [3.1.4 – Beispiel](#) in eine Aktivität „Verteilen“ und eine Entität „Auftrag“. Wie sieht das Diagramm dann aus?
2. Modellieren Sie die „Telefonieren via Internet“, wobei unter anderem die Aktivitäten Sprechen und Hören vorkommen sollen.
3. Modellieren Sie das Interesse eines E-Commerce-Kunden an sicherer Übertragung seiner Kreditkartennummer



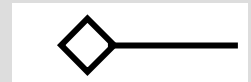
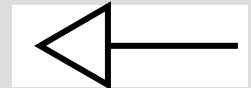
## 4.3 Spezielle Relationen

SeeMe verfügt über einige festgelegte Relationen. Zum einen übernimmt SeeMe Relationen, die sich bei der objekt-orientierten Modellierung als nützlich erwiesen haben:

**Vererbung** und  
**Aggregation**

Zum andern wird eine sogenannte **Meta-Relation** eingeführt. Sie wird benötigt, weil sich die modellierte Struktur von Phänomenen dieser Welt verändern kann. Mit Meta-Relationen kann man die Veränderung modellierter Strukturen durch andere Elemente des Modells ausdrücken.

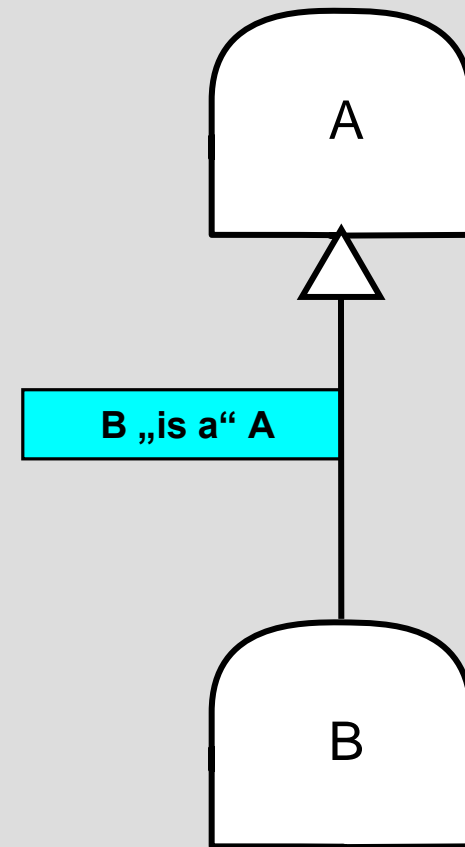
Einbettung lässt sich auch als besondere Relation verstehen, die nicht durch einen Pfeil dargestellt werden muss, aber ggf. durch einen Pfeil dargestellt werden könnte.



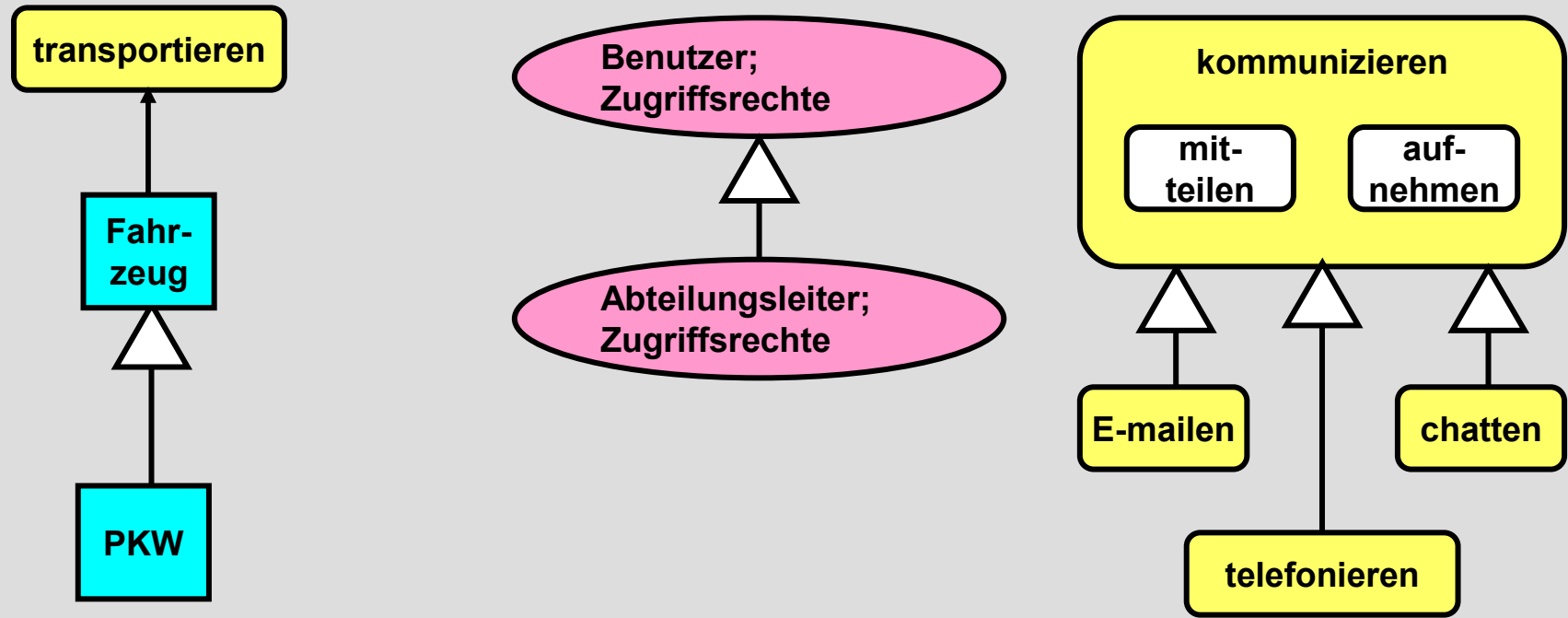
## 4.3 Vererbung

### Vererbung

Der Vererbungsfeil zeigt auf das Element (A), von dessen Art das andere Element (B) ist. Das heißt, dass B die Eigenschaften und die Struktur von A erbt (Sub-Elemente und Attribute), sowie dessen Beziehungen (mittels Relationen dargestellt) zu anderen Elementen.



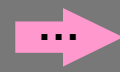
## 4.3 Beispiel zu Vererbung



Der PKW hat die gleichen Eigenschaften, wie die Entität Fahrzeug und wird genauso durch die Aktivität „transportieren“ genutzt.

Jeder Abteilungsleiter ist Benutzer und erbt die Zugriffsrechte eines Benutzers.

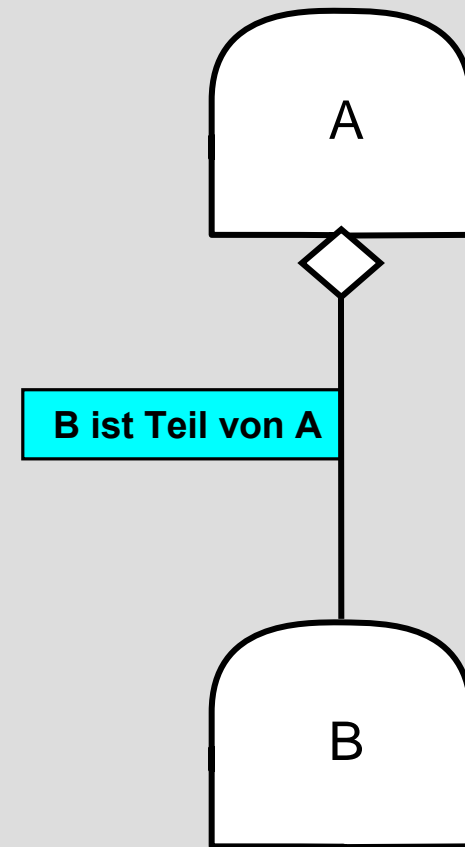
Via Vererbung wird ausgedrückt, dass E-mailen, Chatten und Telefonieren die Sub-Aktivitäten „mitteilen“ und „aufnehmen“ haben.



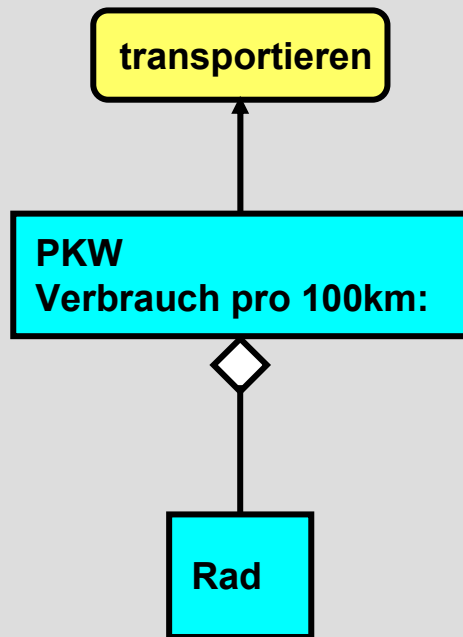
## 4.3 Aggregation

### Aggregation

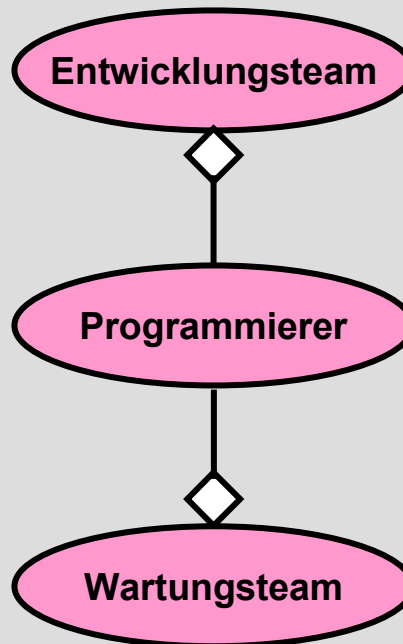
Es wird ausgedrückt, dass B Teil von A ist, also dieselben Eigenschaften hat, als wenn es als Sub-Element von A dargestellt wäre (umgekehrt präsentiert aber nicht jedes Sub-Element eine Aggregation). Die Relationen, die mit A als ganzem verbunden sind, gelten auch für B. Falls A schon Sub-Elemente hat, wird mit der Aggregation von B u.U. eine neue Perspektive der Zerlegung eingeführt.



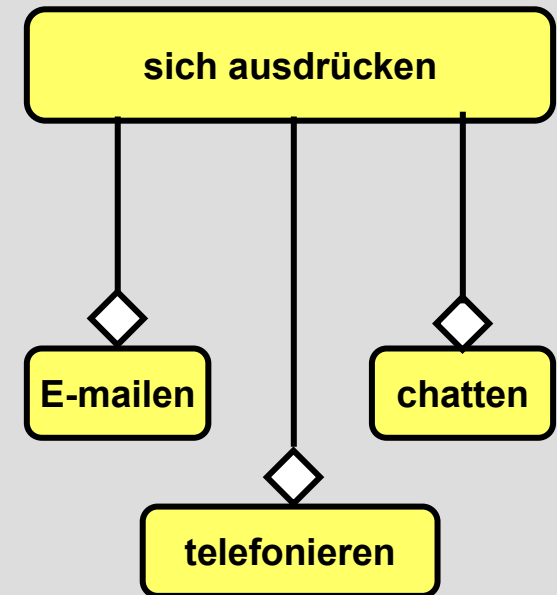
## 4.3 Beispiele zu Aggregation



Das Rad ist ein Teil des PKW, wird auch zum Transportieren genutzt, erbt aber nicht die Eigenschaften des PKW



Programmierer kann Teil eines Entwicklerteams und eines Wartungsteams sein. Das gilt auch für die Person, die letztlich die Rolle des Programmierers spielt.



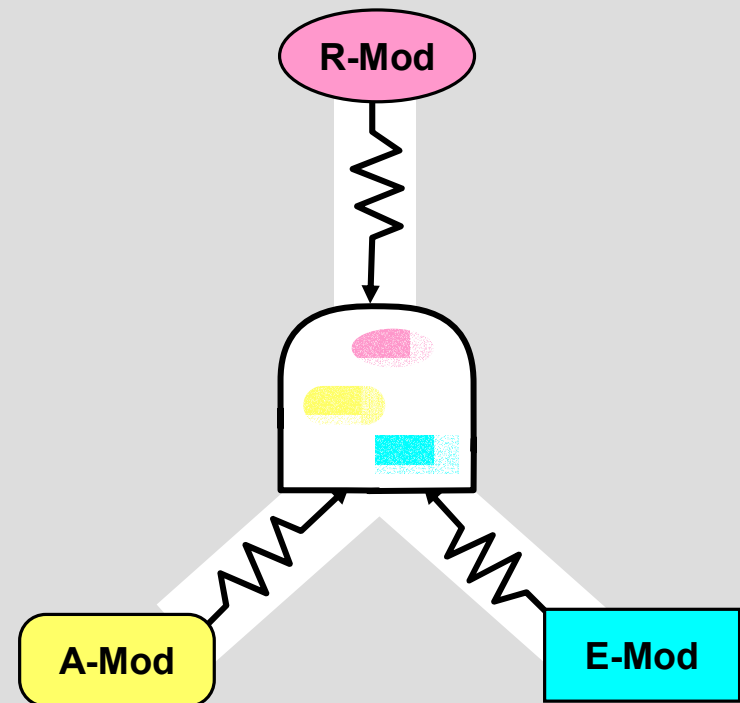
„sich ausdrücken“ ist Teil von E-Mails, Telefonieren, und Chatten. Mittels Aggregation muss es aber nicht 3 mal als Sub-Element dargestellt werden.

## 4.3 Meta-Relationen

Mit Meta-Relationen wird ausgedrückt, dass **R**ollen, **A**ktivitäten oder **E**ntitäten die Struktur einer Rolle, Aktivität oder Entität **modifizieren** können.

Von Meta-Relationen wird gesprochen, weil sie auf Basis-Elemente verweisen, deren Struktur nicht in der dargestellten Weise stabil bleibt, sondern sich unter dem Einfluss anderer Elemente verändern kann. Diese Veränderung kann nicht als Änderung von Attributwerten dargestellt werden.

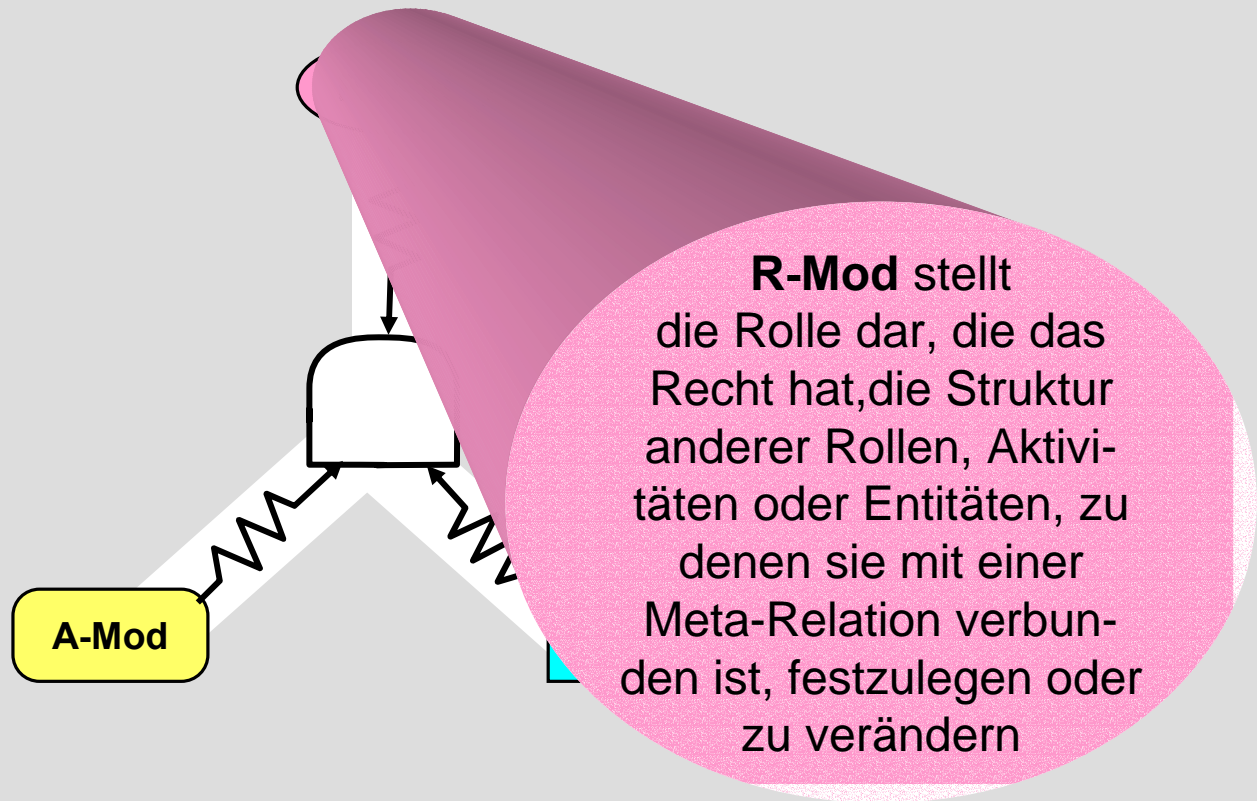
Die Struktur wird wesentlich durch die Einbettung von Sub-Elementen und den Relationen zwischen ihnen bestimmt. Meta-Relationen können Emergenz symbolisieren.





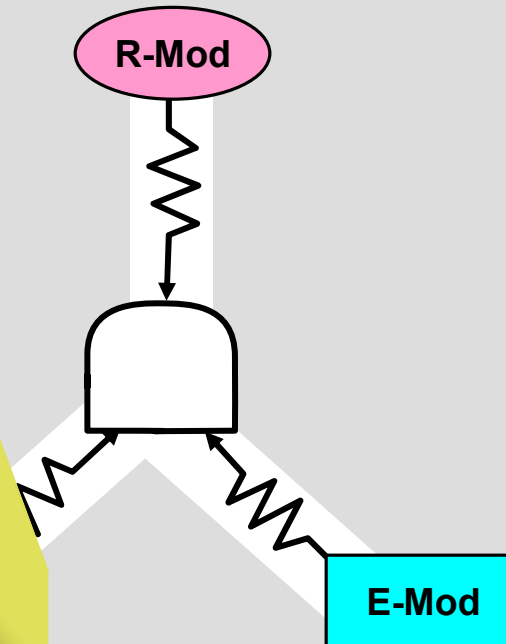
## 4.3 Meta-Relationen

Zum Beispiel kann ein Projektleiter die Rechte und Pflichten der Mitglieder seines Projektteams festlegen. Ein Programmierer kann die Aktivitäten (das Verhalten), eines künstl. Agenten anpassen. Die Struktur der Aktivität lernen kann durch einen Lehrer verändert werden.



## 4.3 Meta-Relationen

**A-Mod** ist die Aktivität, die die Struktur anderer Rollen, Aktivitäten oder Entitäten ändern kann, indem sie Sub-Elemente bzw. Relationen entfernt, erzeugt oder umbettet bzw. umhängt.

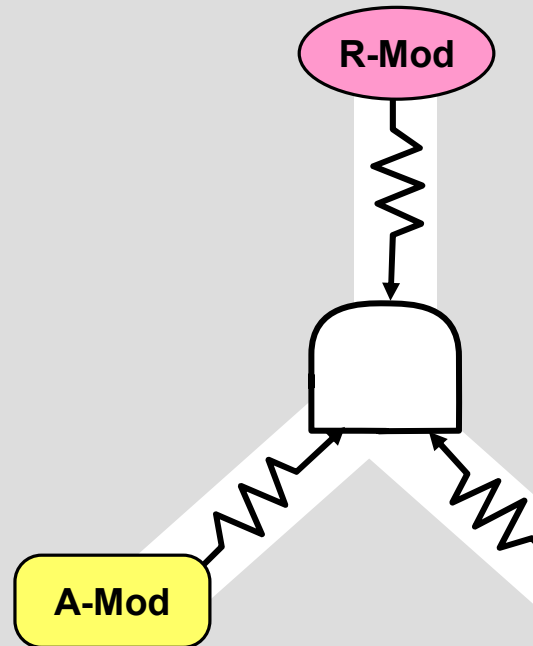


Z.B. sind *organisieren* oder *Vertrag abschließen* typische Aktivitäten, die die Struktur von Rollen festlegen. Das „Lernen“ kann die Struktur der Aktivität „wahrnehmen“ verändern oder „systemisches Rationalisieren“ verändert das Arbeiten. „Gene manipulieren“ verändert die Entität „Organismus“.



## 4.3 Meta-Relationen

Z.B. kann ein Gesetz oder eine Verfahrensvorschrift eine Aktivität (etwa den Handel via E-Commerce) festlegen oder die Struktur von Rollen verändern (etwa die Struktur von „Telekommunikationsdienstleister.“) Die Entität „Information“ kann u.U. die Entität „Kompetenz“ strukturell festlegen.

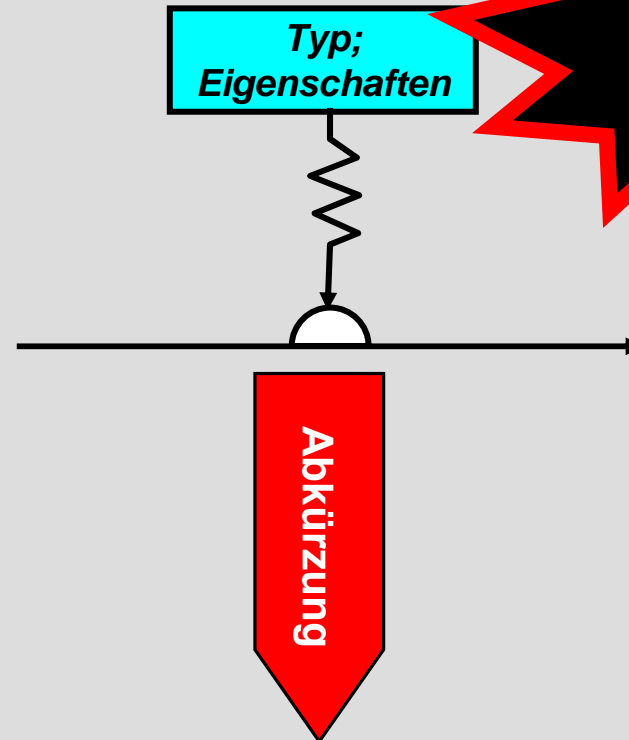
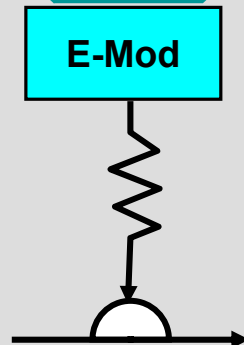


**E-Mod** repräsentiert Ressourcen, die Strukturen festlegen oder beschreiben, sofern eine Rolle, Aktivität oder eine andere Entität so beschaffen ist, dass E-Mod sie determinieren kann.



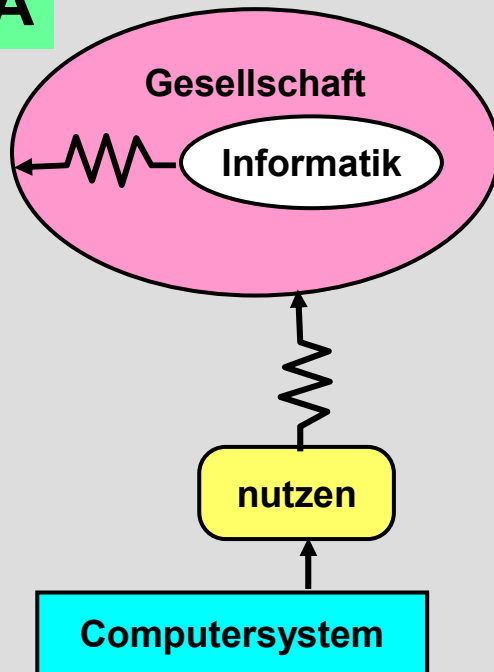
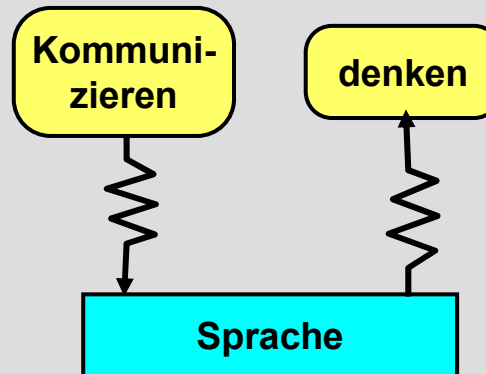
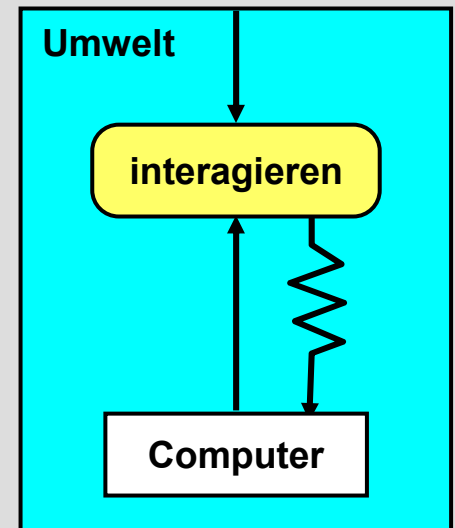
## 4.3 Meta-Relationen

**E-Mod** repräsentiert in diesem Fall eine Beschreibung, die den Typ und die Eigenarten einer Relation näher festlegen.



**Selten  
benötigt**

## 4.3 Weitere Beispiele zu Meta-Relationen

**A****B****C**

Beispiel A) besagt, dass die Informatik und auch die Nutzung von Computern die Gesellschaft strukturell verändert. Computersysteme selbst können diesen Effekt nicht direkt haben – es bedarf der Vermittlung über die Nutzung. Fraglich sind die Beispiele B) und C), in denen ausgedrückt wird, dass das Kommunizieren die Sprachstruktur prägt und ändert und diese wieder unser Denken bestimmt. Es ist noch ungeklärt, ob Computer durch die Interaktion mit der Umwelt strukturelle, emergente Veränderungen durchlaufen können.



## 4.3 Einbettung als Relation

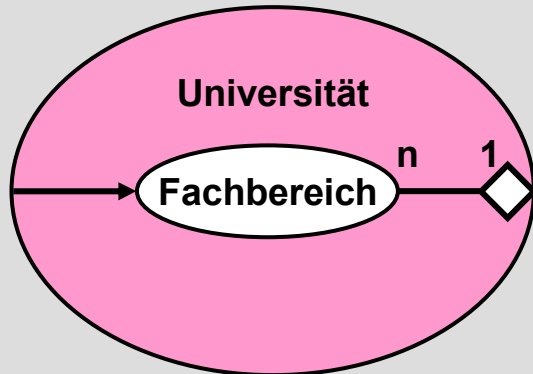
Einbettung kann als Relation verstanden werden, die nicht näher spezifiziert werden muss. Einbettung kann eine Zuordnung, ein Enthaltensein, Aggregation, Vererbung oder auch anderes bedeuten.

Bei Bedarf kann man die Relation zwischen eingebettetem und übergeordnetem Element eintragen, um sie zu betonen oder ihre Eigenschaft zu explizieren, wie bei Beispiel A und C bei den [weiteren Beispielen zu Meta-Relationen \(4.3\)](#).

Hierdurch können insbesondere quantitative Verhältnisse ausgedrückt werden.

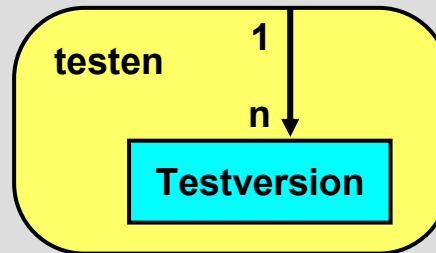


## 4.3 Beispiele zu Einbettung als Relation

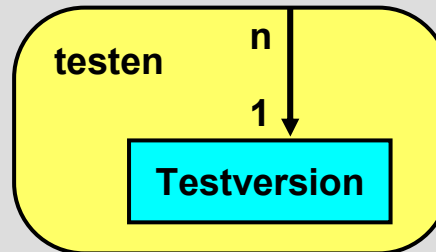


Eine Universität besteht aus mehreren Fachbereichen, jeder Fachbereich gehört zu genau einer Universität. UND Die Universität hat Erwartungen an die Fachbereiche, mit deren Erfüllung sie rechnen kann.

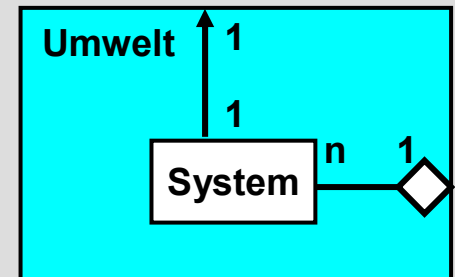
A



B



Die Testversion dient ausschließlich dem Testen.  
A) N verschiedene Versionen werden je 1-mal getestet.  
B) Eine Version wird n-mal getestet.



Die Umwelt besteht aus vielen Systemen, aber jeder Instanziierung eines Systems ist genau eine Umwelt zugeordnet, da jedes System eine andere Umwelt hat.

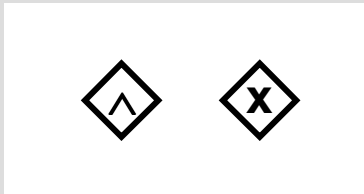
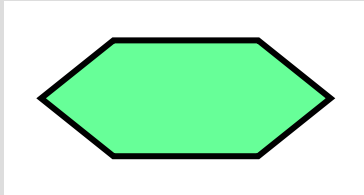
## 4.4 Aufgaben

1. Es gibt 27 verschiedene Möglichkeiten, Standard-Relationen durch Basis-Elemente zu spezifizieren. Davon wurden 5 in den Beispielen des Kapitel 4.2 gezeigt. Ergänzen Sie Beispiele zu drei weiteren unterschiedliche Arten.
2. Stellen Sie mit Hilfe der Aggregationsrelation die Vereinigungsmenge aus den Sub-Elementen zweier Entitäten dar.
3. Erstellen Sie ein Diagramm das zeigt, dass eine Berufungskommission stets aus der gleichen Zahl von Studenten und wiss. Mitarbeitern besteht und dass die Professoren gegenüber der Gesamtzahl dieser beiden Gruppen stets eine Stimme mehr haben.
4. Basis-Elemente können durch Meta-Relationen auf sich selbst verweisen – geben Sie drei sinnvolle Beispiele.

**Aufg.**



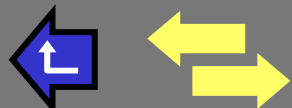
## 5) Modifikatoren (Bedingungen & Ereignisse) und Konnektoren



5.1 Mit Modifikatoren kann man darstellen, dass bestimmte Basis-Elemente oder Relationen nicht immer, sondern nur unter bestimmten Bedingungen und mit einer bestimmten Häufigkeit existieren bzw. instantiiert werden.

5.2 Mit Konnektoren kann man Relationen logisch miteinander verknüpfen. Insbesondere kann man auch Verzweigungen darstellen. Mit Modifikatoren gibt man an, welcher Zweig gewählt wird.

5.3 Propagierung von Konnektoren: Bei der konkreten Umsetzung eines Modells ist zu beachten, wie sich die Bedingungen auf alle Relationen und Basis-Elemente auswirken.



## 5.1 Modifikatoren (Bedingungen & Ereignisse)

Modifikatoren modifizieren die Existenz eines im Diagramm gezeigten Elementes (Basis-Element oder Relation). Das Element existiert in der Realität nicht in jedem Fall, sondern nur in den Fällen, die der Modifikator näher beschreibt.

Die Beschreibung erfolgt, indem Bedingungen oder Ereignisse angegeben werden, die den Fall kennzeichnen.

### Gliederung:

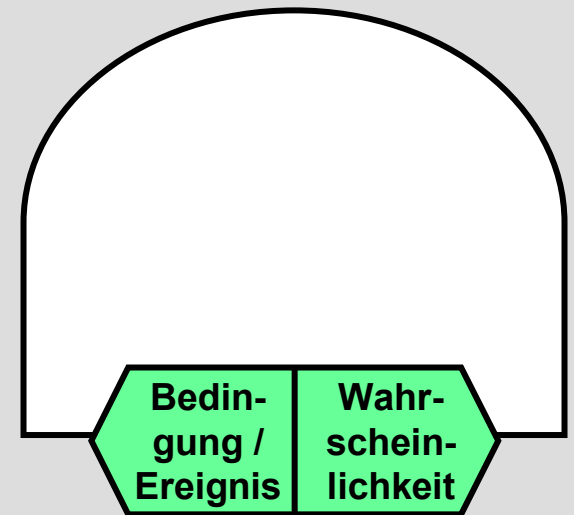
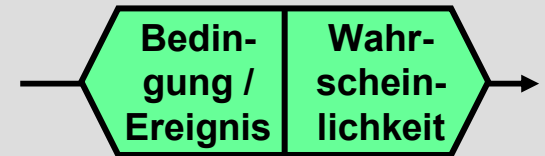
- [5.1.1](#) Annotation von Modifikatoren
- [5.1.2](#) Struktur von Modifikatoren
- [5.1.3](#) Super-Modifikatoren
- [5.1.4](#) Abkürzungen & Standards
- [5.1.5](#) Aufgaben



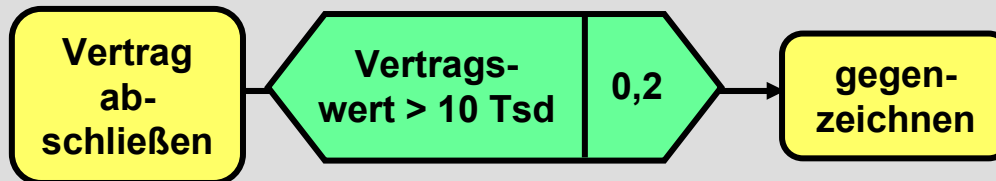
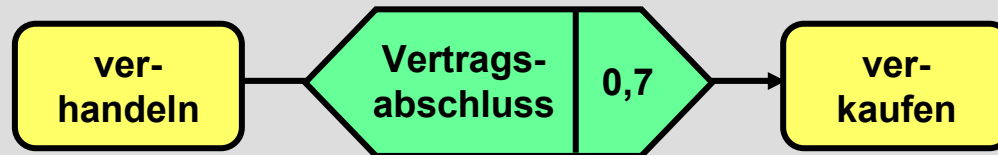
## 5.1.1 Annotation von Modifikatoren

Auf einer Relation kann genau ein Modifikator annotiert werden. Er ist dann ihr fester Bestandteil. Er gibt eine Bedingung oder ein Ereignis an. Diese Bedingung muss erfüllt sein oder das Ereignis muss eintreten, damit die Relation als instantiierbar, das heißt als gegebenes Element eines Diagramms angesehen werden kann. Außerdem wird in der zweiten Hälfte angegeben, wie wahrscheinlich die Bedingung erfüllt ist bzw. das Ereignis eintritt.

Insbesondere kann bei SeeMe je ein Modifikator auch fest mit einem Basis-Element verbunden werden. Er gibt an, unter welchen Bedingungen oder bei welchen Ereignissen dieses Basis-Element existiert und mit welcher Wahrscheinlichkeit. Modifikatoren werden immer am unteren Rand des Basis-Elementes annotiert. Modifikatoren können nicht auf Segmentlinien annotiert werden.



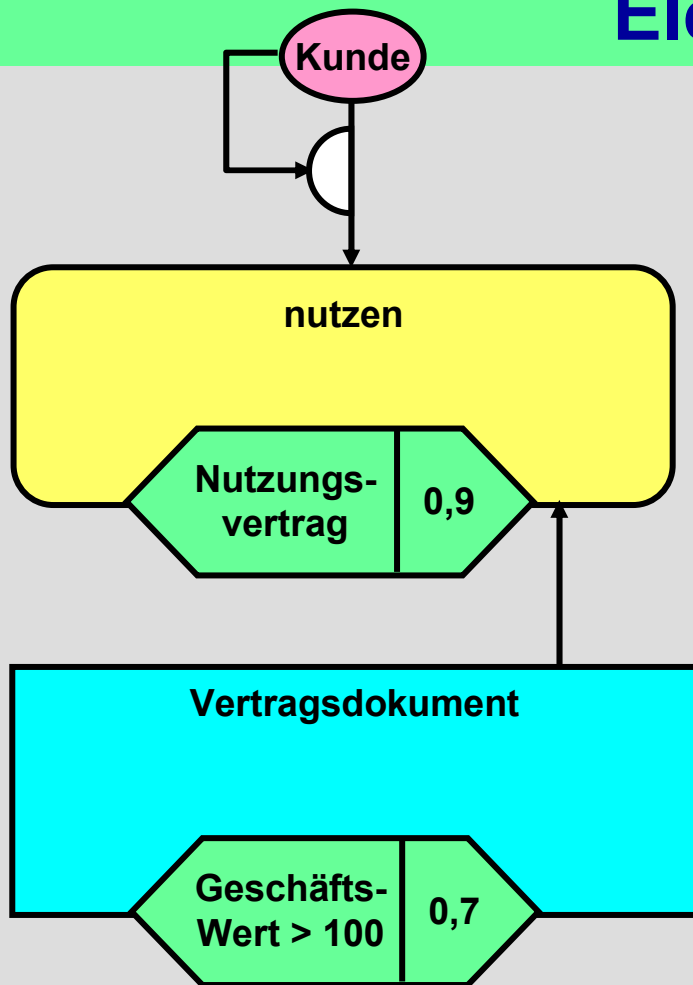
## 5.1.1 Beispiel - Modifikatoren auf Relationen



Das obere Beispiel zeigt ein Ereignis, das eintreten muss, bevor der Verkauf stattfindet. Außerdem wird ausgedrückt, dass dies ihn 70% aller Fälle der Fall ist.

Das untere Beispiel arbeitet mit einer Bedingung: Wenn der Vertragswert höher als 10 Tsd € ist, muss gegengezeichnet werden, was in 20% aller Fälle vorkommt.

## 5.1.1 Beispiel - Modifikatoren auf Basis-Elementen



Hier wird ausgedrückt, dass die Aktivität „nutzen“ nur möglich ist, wenn ein Nutzungsvertrag existiert. Dies ist in 90% aller Fälle, mit denen sich das Diagramm befasst (nämlich dass ein Kunde ein Nutzungsinteresse hat) der Fall. Ein schriftliches Vertragsdokument gibt es nur, wenn der Geschäftswert höher als 100 € ist. Dies ist in 70% der Fälle der Fall. Beide Wahrscheinlichkeitsangaben sind voneinander unabhängig. Auf Basis-Elemente gerichtete Relationspfeile sollen nicht auf Modifikatoren zeigen.



## 5.1.2 Interne Struktur von Modifikatoren

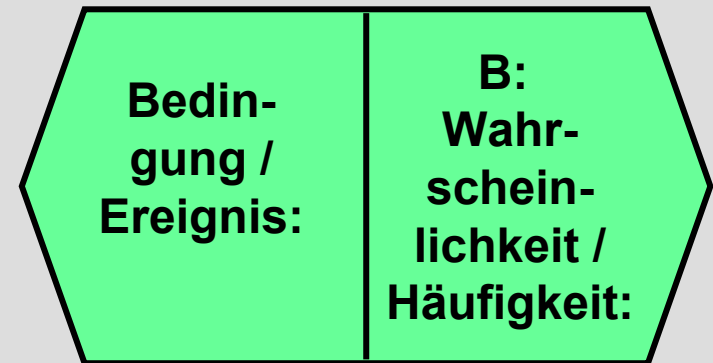
Modifikatoren beinhalten zwei Teile, die als Attribute anzusehen sind.

Die Werte des ersten Attributs spezifizieren Bedingungen oder Ereignisse.

Diese können bei Bedarf eine komplexe logische Struktur haben. Die Kombination mehrerer Modifikatoren wird ersetzt durch einen logisch strukturierten Wertangabe für das erste Attribut.

Das zweite Attribut macht Häufigkeits- oder Wahrscheinlichkeitsangaben. Diese können quantitativ, qualitativ oder deontisch ausgedrückt werden.

Die Attributnamen müssen nicht aufgeschrieben werden, da die Art des Attributes an der Position und am Symbol (grünes Hexagon) erkennbar ist.



## 5.1.2 Interne Struktur von Modifikatoren – Beispiel für verschiedene Ereignisse

nach 5 Uhr  
UND B<C

A  $\Leftrightarrow$  B

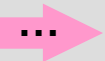
Anfrage  
liegt vor

Anzeige  
ODER Hinweis

Die Beispiele zeigen, dass hier Bedingungen dargestellt und kombiniert werden, die einen quantitativen Hintergrund haben.

Es sind auch logische Ausdrücke möglich.

Qualitativ kann auf Ereignisse Bezug genommen, das heißt, dass bestimmte Phänomene der Fall sind. Hinweise wie „liegt vor“ oder „existiert“ können in der Regel weggelassen werden.



## 5.1.2 Interne Struktur von Modifikatoren – Beispiel für verschiedene Häufigkeitsangaben

0,7seltenkaummanchmalofthäufigfast immer


Häufigkeits- oder Wahrscheinlichkeitsangaben können quantitativ erfolgen, in dem man einen Wert im Intervall zwischen 0 und 1 auswählt.

Außerdem sind qualitative Angaben nach freier Wahl möglich. Dies ist in den meisten Fällen sinnvoll, in denen keine gesicherten statistischen Angaben vorliegen. Man kann die qualitativen Beschreibungen verwenden, die zum Beispiel in Interviews genannt werden.

...



## 5.1.2 Interne Struktur von Modifikatoren – Beispiele für deontische Angaben

 erlaubt verboten geboten geboten UND 0,95 verboten UND 0,05

Eine besondere Art der Angabe, die ausdrückt, dass etwas nicht immer der Fall ist, sind nicht beschreibend, sondern vorschreibend (deontisch).

Etwas kann erlaubt sein, geboten oder verboten.

Wenn etwas verboten ist, heißt dies nicht, dass es nie der Fall ist; genau so, wie das Gebotene nicht immer eintritt.

Vorschreibende Angaben können mit beschreibenden Angaben gekoppelt werden.

 Password ändern Password älter  
als 6 Monaten Geboten  
UND 0,1

Wenn das Password älter als 6 Monate ist, muss es geändert werden und dies ist für 10% aller Passwörter der Fall.

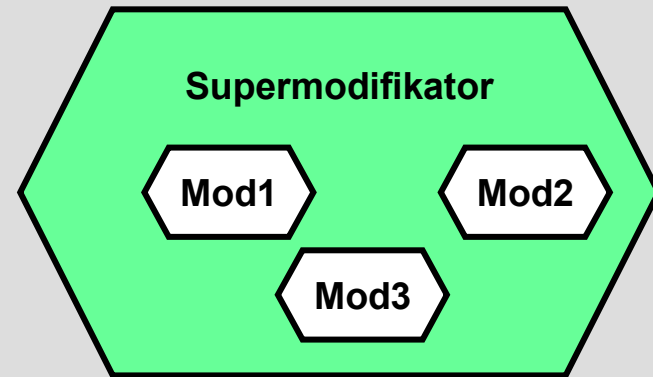
Dies bedeutet nicht, dass es auch in 10% aller Fälle geändert wird.

 Siehe auch

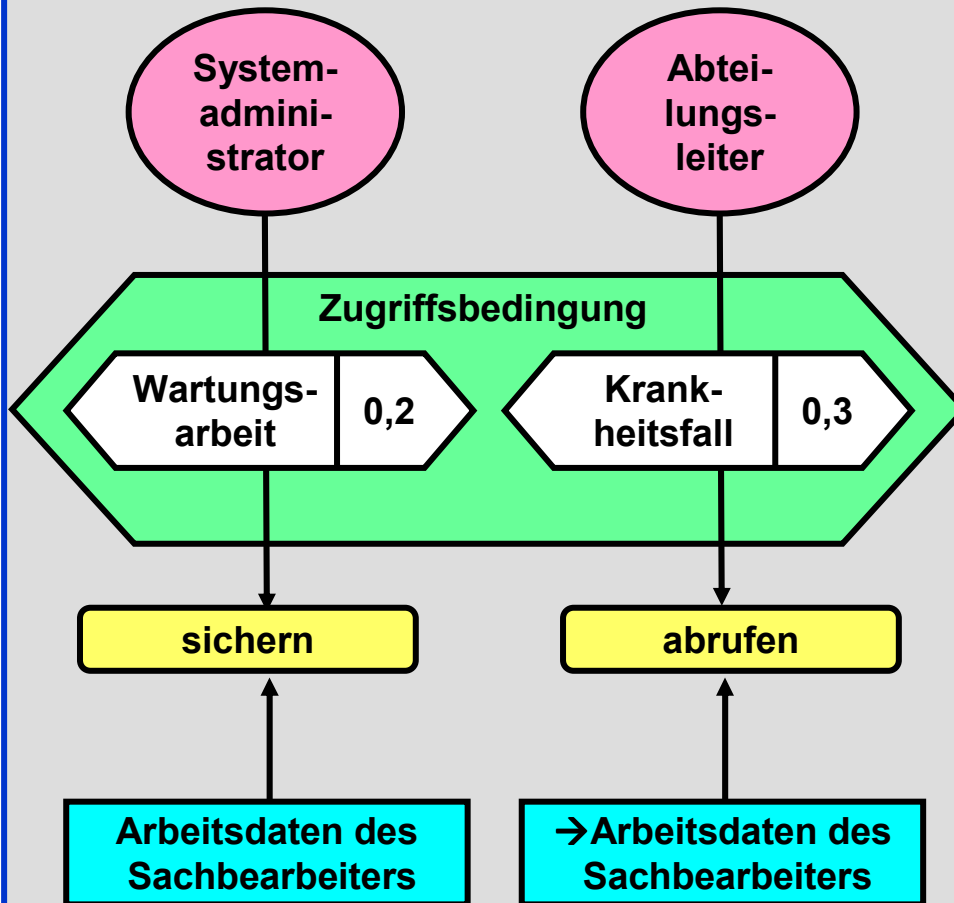
## 5.1.3 Supermodifikator

Supermodifikatoren können genutzt werden, um mehrere Modifikatoren zusammenzufassen. Dies ist insbesondere hilfreich, wenn Bedingungen oder Ereignisse mit gemeinsamen Ereignissen auftreten oder wenn man Ein- und Ausblendemechanismen nutzen möchte.

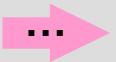
Siehe 8)



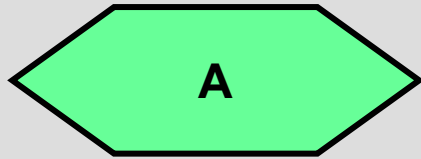
## 5.1.3 Beispiel für Supermodifikator



Der Supermodifikator „Zugriffsbedingung“ sammelt alle Bedingungen, unter denen verschiedene Rollen auf die Daten der Arbeit eines Sachbearbeiters zugreifen darf.

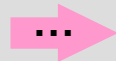


## 5.1.4 Abkürzung und Standardmodifikatoren – nur Ereignisse

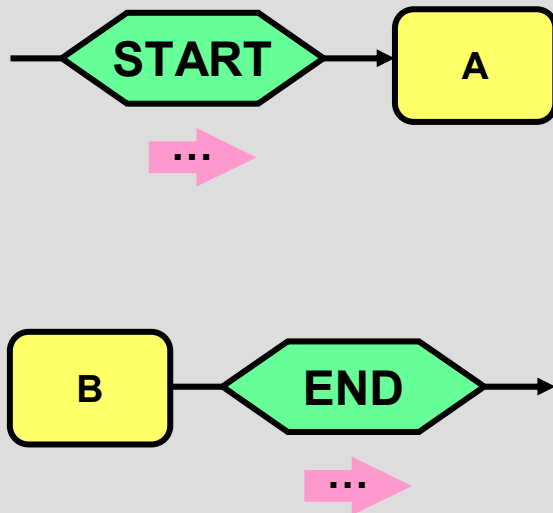


Oftmals kann man die Bedingung oder das Ereignis eines Modifikators nennen, ohne dass es sinnvoll ist, sich um die Häufigkeit oder Wahrscheinlichkeit zu kümmern, mit der sie auftreten. Es gilt folgendes Abkürzungsregel: Wenn nur ein Attribut innerhalb des Modifikators spezifiziert ist, handelt es sich um die Bedingung oder das Ereignis.

Weiter Möglichkeiten zu vagen Angaben werden im Folgenden beschrieben.



## 5.1.4 Abkürzung und Standardmodifikatoren – START und END



Bei der Darstellung von Prozessen ist es manchmal sinnvoll, Start- und Endpunkte zu markieren.

Dies kann mit Hilfe von Modifikatoren geschehen, die auf Pfeile annotiert werden, die jeweils auf die Start- oder die End-Aktivität eines Prozesses zeigen.

Die offenen Enden der Pfeile deuten an, dass vor dem Start oder nach dem Ende noch etwas anderes liegt, das hier nicht näher spezifiziert wird.

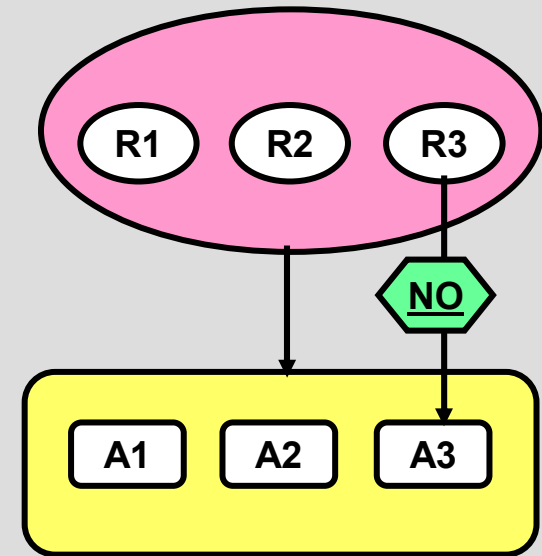
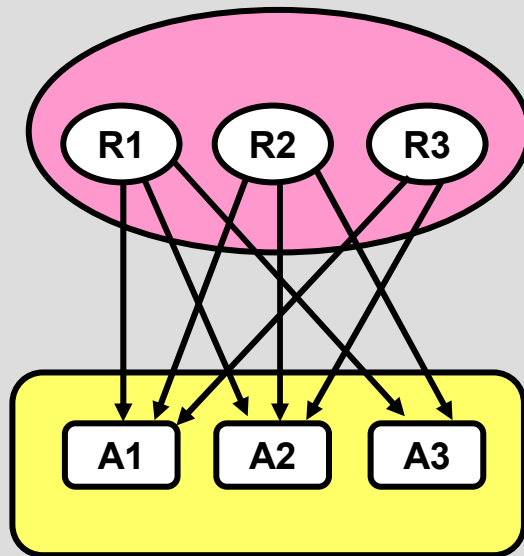
## 5.1.4 Abkürzung – Negation



Es ist hilfreich einen Modifikator zu haben, der ausdrückt, dass ein bestimmtes Element nie instantiiert werden kann. Hierdurch lassen sich kombinatorische Explosionen vermeiden



## 5.1.4 Beispiel für Verwendung von Negation

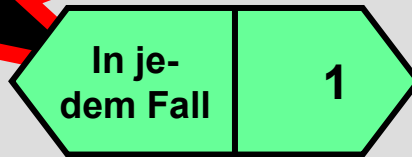


Die linke und die rechte Abbildung sind äquivalent. Durch den NO-Modifikator und das Einbettungsprinzip spart man sich das Eintragen von Relationen.



## 5.1.4 Abkürzung – Immer

Selten  
benötigt



Abkürzung

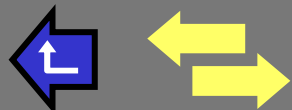


Das Gegenstück zum NO-Modifikator ist ein Modifikator, der ausdrückt, dass das Element in jedem Fall instantiiert wird – seine Existenz wird dadurch betont. Eigentlich gilt dies für alle Relationen oder Elemente, zu denen kein Modifikator annotiert ist. Mit dem ALL-Modifikator drückt man aus, dass man nicht versehentlich vergessen hat, einen Modifikator zu annotieren.



## 5.1.5 Aufgaben zu Modifikatoren

1. Drücken Sie aus, dass es verboten ist, auf fremde personenbezogenen Daten zuzugreifen, wenn die Zugangssperre ausgefallen ist.
2. Wie stellt man dar, dass man in einem Chatroom keine Beleidigungen äußern darf.
3. Drücken Sie aus, dass ein Nutzer auf bestimmte Daten nur während der Kernarbeitszeit von 10 bis 16 Uhr zugreifen darf.
4. Bündeln sie die Ihnen bekannten Bedingungen, unter denen elektronischer Zahlungsverkehr als sicher zertifiziert werden kann.
5. Stellen Sie dar: Eine Qualitätssicherung, die aus drei verschiedenen Tests (T1, T2, T3) besteht, wird ständig wiederholt – allerdings darf der gleiche Test nie zweimal (dreimal) direkt hintereinander ausgeführt werden.



## 5.2 Konnektoren

Relationen können durch Konnektoren logisch miteinander verknüpft werden, um sie zu verzweigen oder zu zusammenzuführen.

Es gibt zwei Basis-Konnektoren: UND und XOR (exklusives ODER). Weitere Konnektoren mit besonderer Bedeutung können eingeführt werden. Hierzu zählen insbesondere der einfache ODER-Konnektor und die OPTIONALE Verknüpfung.

Es können sowohl Relationen gleichen als auch verschiedenen Typs kombiniert werden.

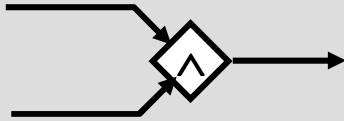
### Gliederung:

- [5.2.1](#) Überblick
- [5.2.2](#) Gemischte Beispiele
- [5.2.3](#) Verknüpfung von Aktivitäten
- [5.2.4](#) Verknüpfung unterschiedlicher Relationstypen
- [5.2.5](#) Konnektoren und eingebettete Elemente
- [5.2.6](#) Aufgaben

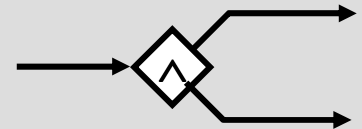


## 5.2.1 Überblick zu Konnektoren

Konnektoren werden als rechtwinklige Rauten dargestellt



Zusammenführung



Verzweigung



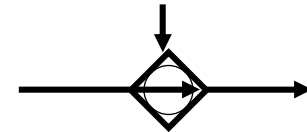
UND



XOR



ODER

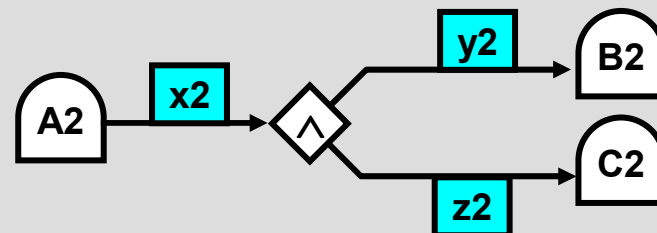
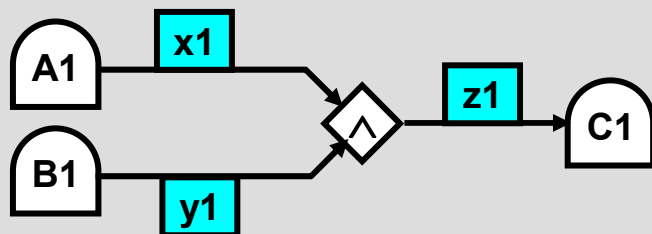


OPTIONAL  
Die von außen auf  
den Rand zeigende  
Relation ist optional

...



## 5.2.1 UND-Konnektor

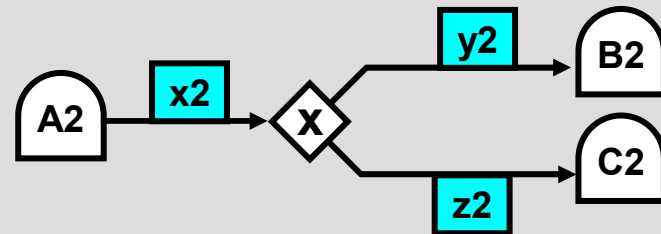
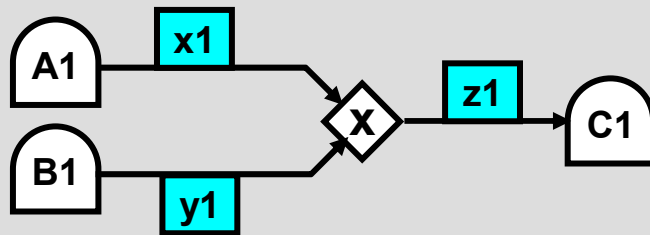


Durch die Zusammenführung zweier Relation (x1 und y1) entsteht eine neue (z1), durch die Verzweigung mit UND entstehen zwei neue (y2 und z2 aus x2). Der Typ der Relationen bestimmt sich aus den beteiligten Elementen und der Richtung; x1 also z.B. aus A1 und C1.

Der UND-Konnektor kennzeichnet gemeinsame Instantiierungen; wenn also x1 und y1 instantiiert sind, dann muss auch z1 instantiiert werden und wenn x2 instantiiert wurde, dann müssen auch y2 und z2 instantiiert werden.

Während die Zusammenführung mit UND ausdrückt, dass A1 und B1 für C1 gemeinsam von Relevanz sind (z.B. als Vorbedingung einer Aktivität), drückt die Verzweigung aus, dass A2 gemeinsam für zwei weitere Elemente B2 und C2 von Relevanz ist.

## 5.2.1 XOR-Konnektor



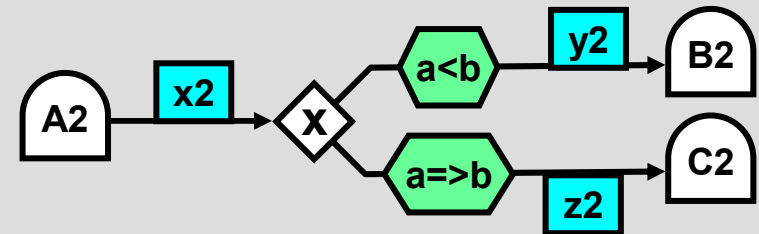
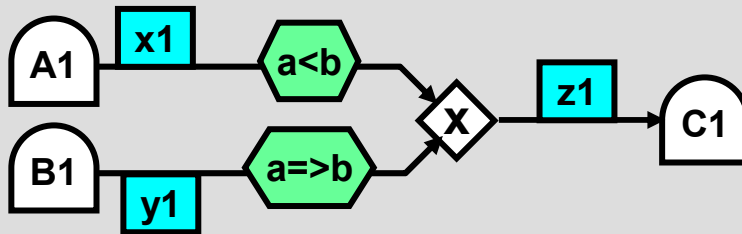
Bei der Zusammenführung zweier Relation (x1 und y1) mit XOR wird eine Entscheidung erzwungen, welche von ihnen die resultierende Relation (z2) repräsentiert und mit einem ausgewählten Basis-Element (C1) verbunden wird.

Bei der Verzweigung muss entschieden werden, mit welchem ausgewählten Basis-Element eine Relation (x2) verbunden wird und von welchem Typ sie letztlich ist.

Der XOR-Konnektor kennzeichnet alternative Instantiierungen; wenn also z1 instantiiert wird, dann darf entweder x1 oder y1 nicht instantiiert werden und wenn x2 instantiiert wurde, dann darf entweder y2 oder z2 nicht instantiiert werden.

A1 und B1 können nicht beide gemeinsam für C1 relevant sein und A2 kann nicht für B2 und C2 gemeinsam relevant sein.

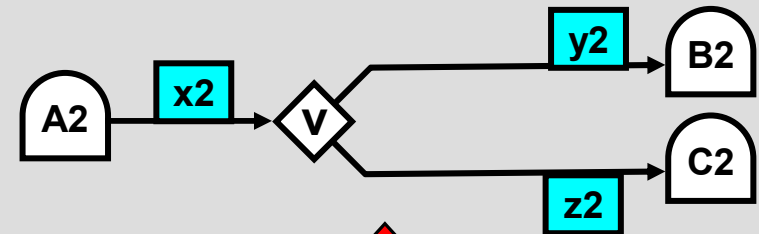
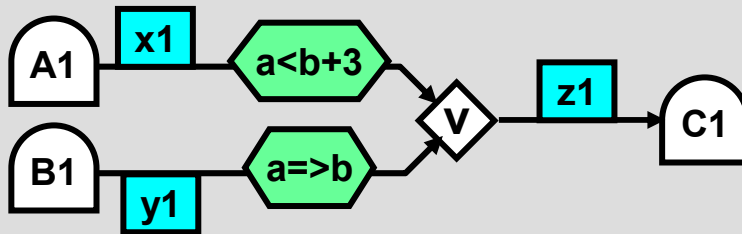
## 5.2.1 XOR und Modifikator



Die bei XOR zu treffende Entscheidung kann durch Modifikatoren vorgegeben werden.

Falls keine Modifikatoren angegeben sind oder diese nicht spezifiziert sind, dann trifft die unmittelbar beteiligte Rolle die Entscheidung ([s. Beispiel I bei 7.7](#))

## 5.2.1 OR-Konnektor

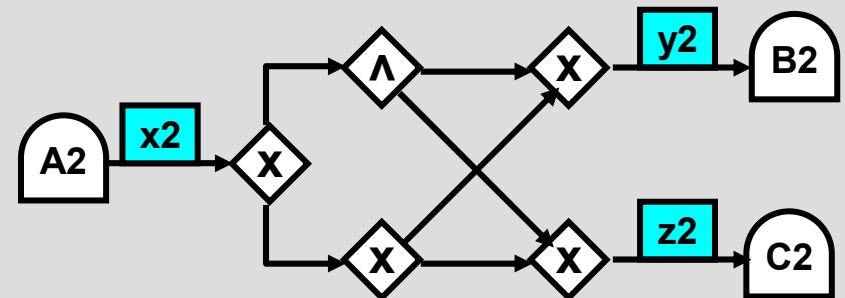


Abkürzung

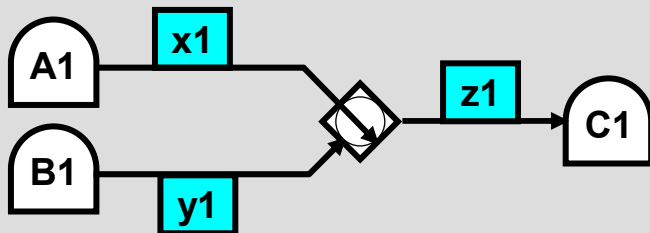
Der OR-Konnektor fordert bei der Zusammenführung, dass mindestens eine Relation (x1 oder y1) oder auch beide realisiert werden, damit z1 realisiert wird.

Bei der Verzweigung bedeutet das OR, dass y2 oder z2 oder beide gemeinsam instantiiert werden.

OR ermöglicht eine Abkürzung der rechts gezeigten Konstruktion.



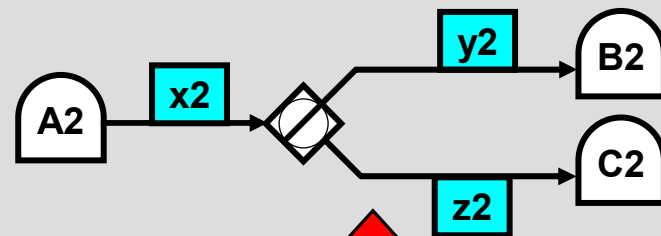
## 5.2.1 OPT-Konnektor



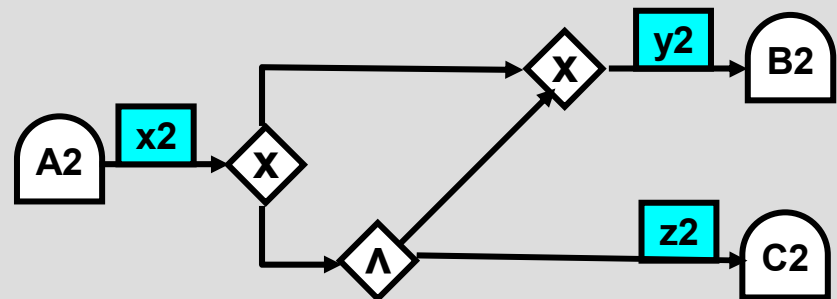
Der OPT-Konnektor verbindet zwei Basis-Elemente in jedem Fall miteinander (hier A1 mit C1 oder A2 mit B2) während die Verbindung für ein weiteres Basis-Element (hier B1 oder C2) nur optional besteht.

Je nach dem repräsentiert z1 (bzw. y2) nur eine oder zwei Relationen.

(Syntaktischer Hinweis: Zu der durch den Konnektor hindurchgezogenen Relation muss auf jeden Fall nur noch eine in die gleiche Richtung zeigende alternative Relation existieren.)



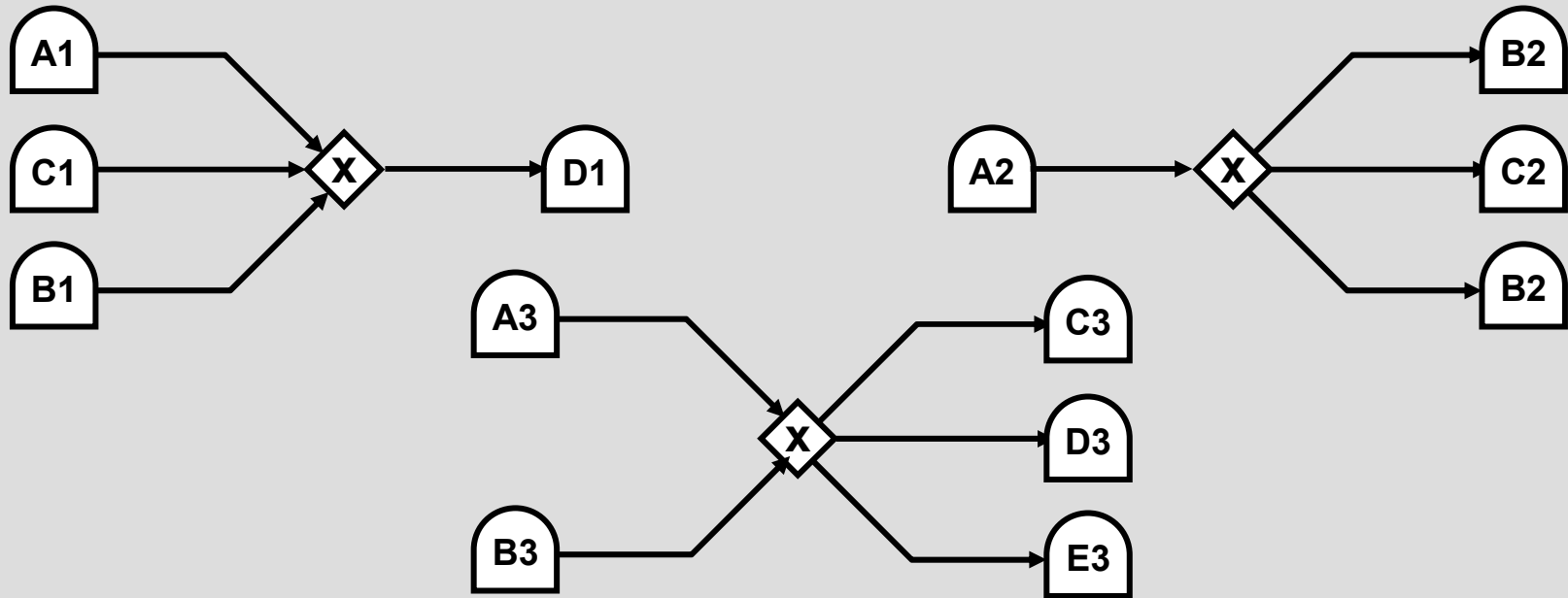
Abkürzung



Aufgabe



## 5.2.1 Verbindung mehrerer Relationen

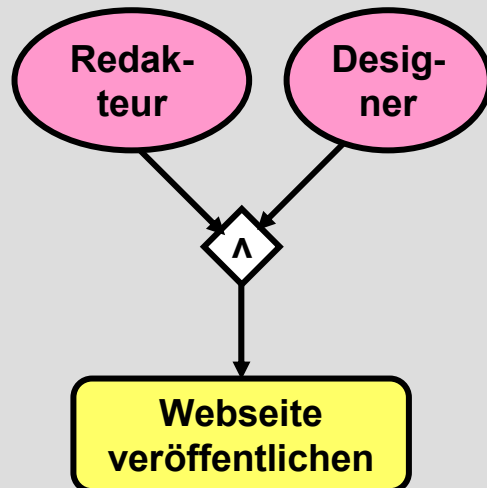


**Konnektoren können auch mehr als drei Relationen miteinander kombinieren. Insbesondere kann das Zusammenführen mehrerer Relationen und das Verzweigen mehrerer Relationen miteinander kombiniert werden (Fall 3).** Diese Mehrfach-Verknüpfung kann auf Konstruktionen zurückgeführt werden, die aus Dreier-Verknüpfungen pro Konnektor bestehen. Die hier dargestellte Mehrfachverknüpfung legt jedoch keine Verknüpfungsreihenfolge nahe.

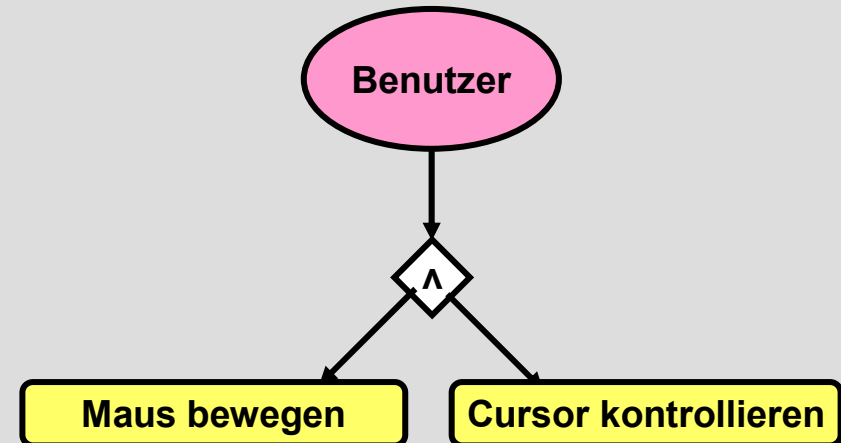


## 5.2.2 UND: Rollen → Konnektor → Aktivität Beispiele

Mit dem UND-Konnektor kann man darstellen, dass zwei Rollen eine Aktivität gemeinsam ausführen oder dass eine Rolle zwei Aktivitäten gemeinsam ausführt.



Der UND-Konnektor drückt Kooperation aus – beide Rollen müssen hier für die Veröffentlichung zusammenwirken.

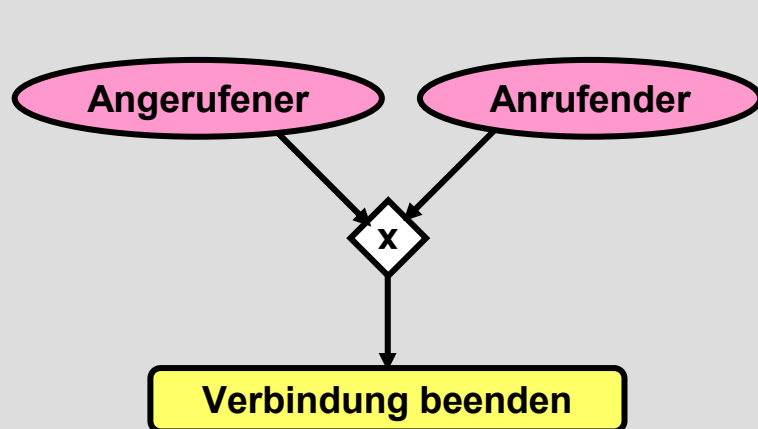


Das Beispiel drückt aus, dass derselbe Benutzer die Maus bewegt und den Cursor kontrolliert.

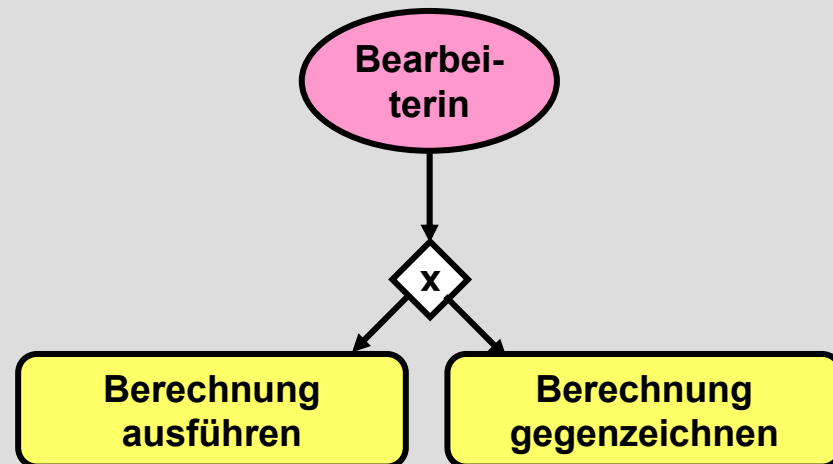


## 5.2.2 XOR: Rollen → Konnektor → Aktivität Beispiele

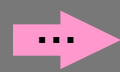
Mit dem XOR-Konnektor kann man darstellen, dass zwei Rollen sich entscheiden müssen, wer eine Aktivität ausführt oder eine Rolle muss alternativ entscheiden, welche Aktivität sie ausführt.



Hier ist ein Telefonnetz so konzipiert, dass eine Verbindung jeweils von einer der beiden Rollen beendet wird. Die jeweils andere Rolle trägt dann zur Beendigung nichts mehr bei.

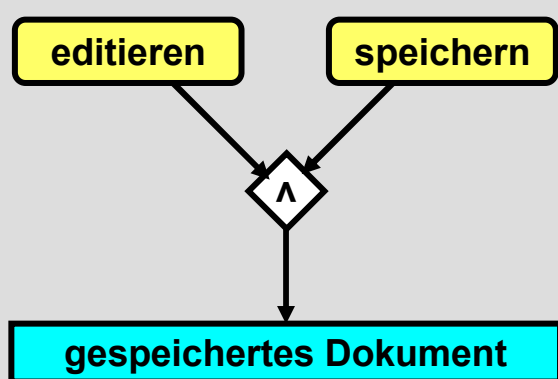


In diesem Prozessmodell kann die Bearbeiterin in jedem einzelnen Fall entweder die Berechnung ausführen oder sie gegenzeichnen, nicht jedoch beides.

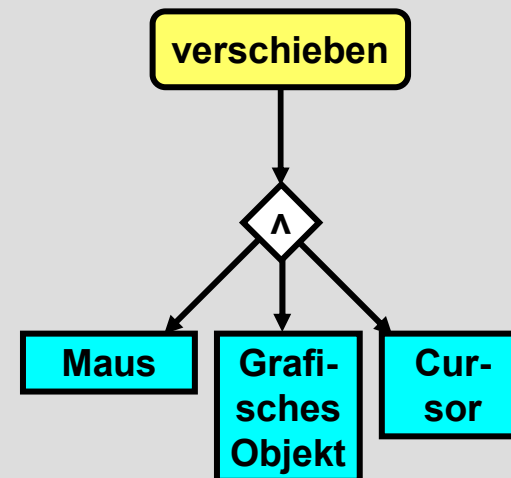


## 5.2.2 UND: Aktivität → Konnektor → Entität Beispiele

Mit dem UND-Konnektor kann man darstellen, dass zwei Aktivitäten gemeinsam notwendig sind, um eine Entität zu verändern oder dass eine Aktivitäten mehrere Entitäten zusammen verändert.



Um ein gespeichertes Dokument zu verändern, muss man es editieren **und** das Ergebnis abspeichern.

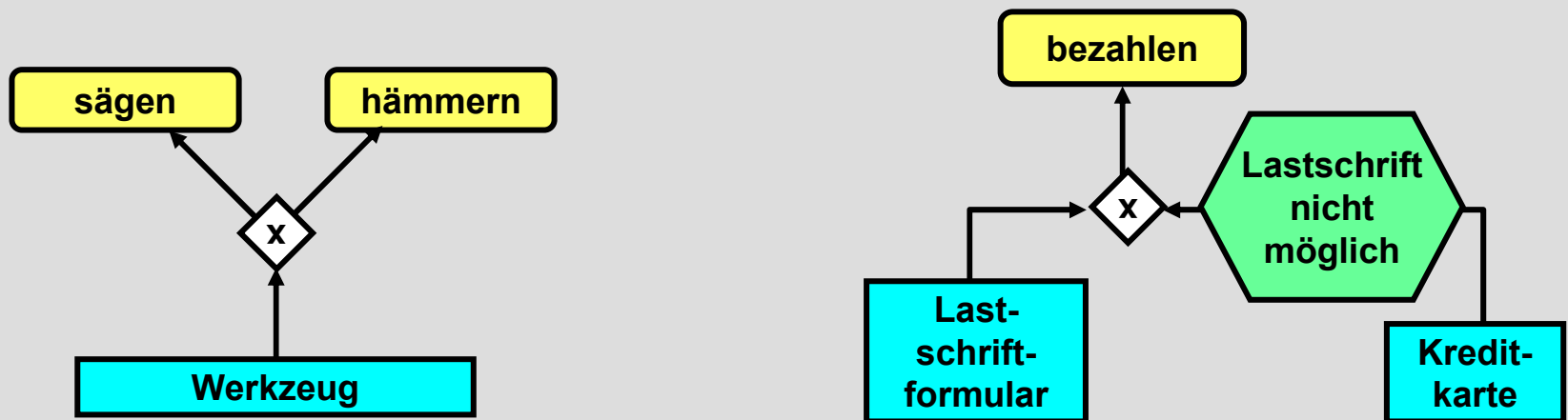


Das Verschieben eines grafischen Objektes wirkt sich auch auf die Position der Maus und des Cursors aus.



## 5.2.2 XOR: Entität → Konnektor → Aktivität Beispiele

Mit dem XOR-Konnektor kann man darstellen, dass eine bestimmte Entität entweder nur für die eine oder die andere Aktivität benutzt werden kann - oder dass eine Aktivität nur die ein oder andere Entität benutzt.

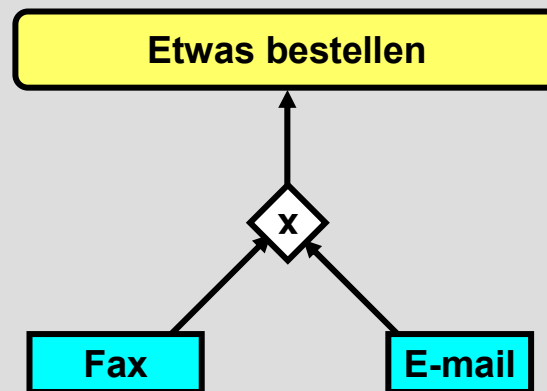
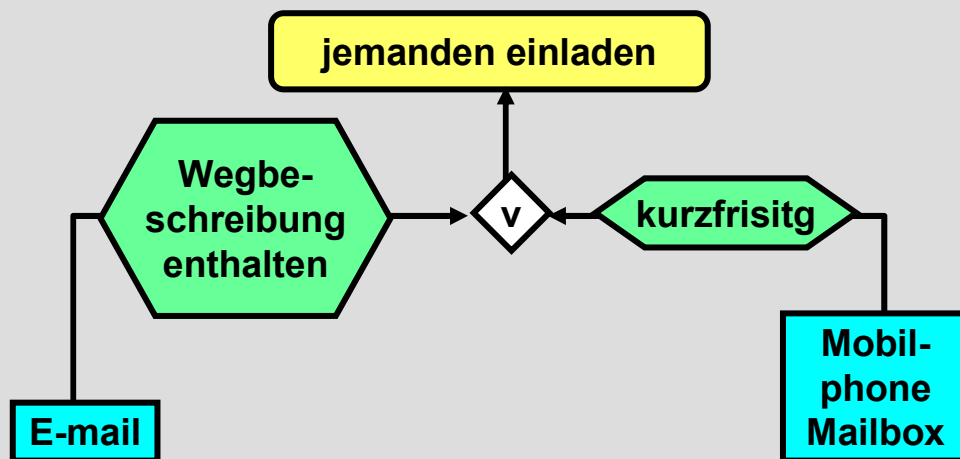


Mit einem konkreten Werkzeug kann man gemäß dieser Abbildung entweder nur hämmern oder nur sägen – ein Werkzeug das beides kann, gibt es nicht.

Das Diagramm zeigt folgendes Verhalten: Man zahlt per Lastschrift und wenn dies nicht möglich ist, per Kreditkarte. In jedem Fall nur über einen dieser Wege.

## 5.2.2 XOR vs. OR → Beispiele

Den OR-Konnektor verwendet man, wenn sich die Bedingungen, unter denen eine bestimmte Alternative gewählt wird, nicht gegenseitig ausschließen, also auch beides möglich ist. XOR symbolisiert dagegen diesen wechselseitigen Ausschluss.



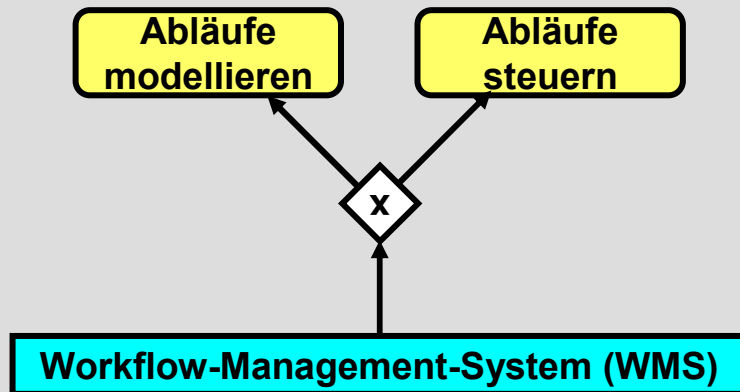
Eine Einladung kann man auch auf zwei Wegen aussprechen, eine Doppelung ist nicht schädlich und kann insbesondere Informationen ergänzen.

Doppelte Übermittlung einer Bestellung kann zu doppelter Lieferung führen und ist daher besser zu vermeiden. Man sollte sich für einen Weg entscheiden.

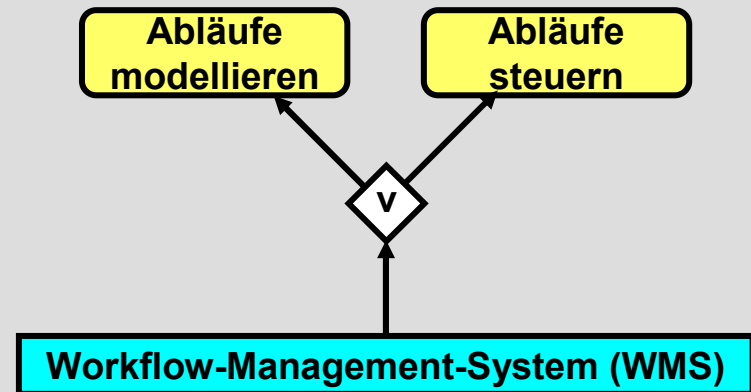
**Aufgabe**

## 5.2.2 Prozess- vs. Strukturmodell

Je nach dem, ob man Diagramme als Prozess- oder als Strukturmodell interpretiert, haben sie unterschiedliche Bedeutung.



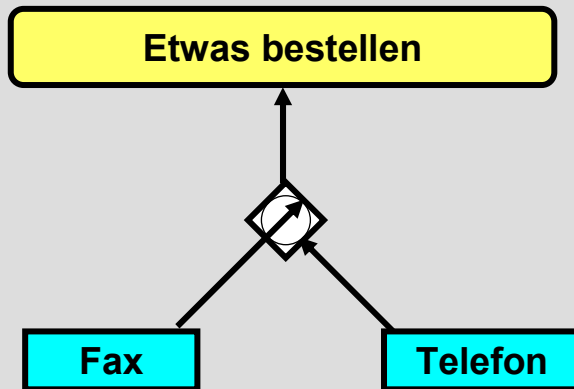
- 1) Als Strukturmodell: Es gibt nur WMS mit denen man entweder nur Abläufe modellieren oder sie steuern kann (unrealistisch).
- 2) Als Prozessmodell: Es gibt WMS, die beides können, aber zu einem konkreten Zeitpunkt nur entweder Abläufe modellieren oder steuern (realistisch)



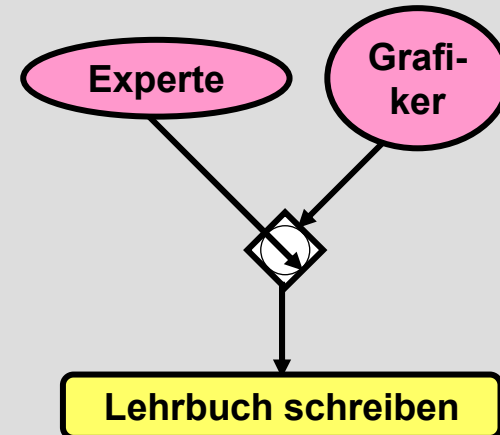
- 1) Als Strukturmodell: Es gibt (auf dem Markt) 3 Arten von WMS: solche, die nur Abläufe modellieren oder sie nur steuern oder die beides können.
- 2) Als Prozessmodell: Ein konkretes WMS kann zu einem konkreten Zeitpunkt entweder Abläufe modellieren oder steuern oder beides.

## 5.2.2 OPT-Konnektor - Beispiele

Häufig treten Fälle auf, in denen ein Element mit Sicherheit mit einem anderen verbunden wird, während dies für ein anderes nur gelegentlich, also bedarfsweise gilt. Dies kann mit dem OPT-Konnektor dargestellt werden



Man nimmt die Bestellung per FAX vor und sichert sich bei Bedarf mit dem Telefon zusätzlich ab.



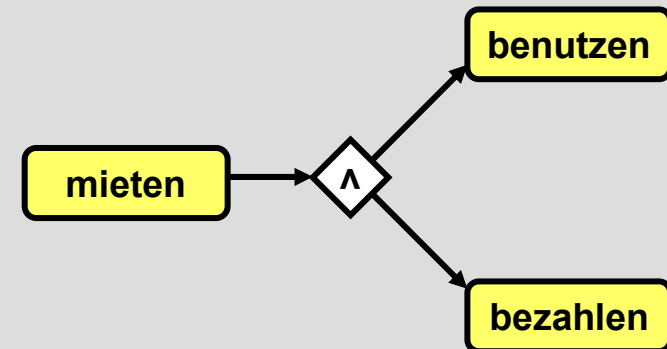
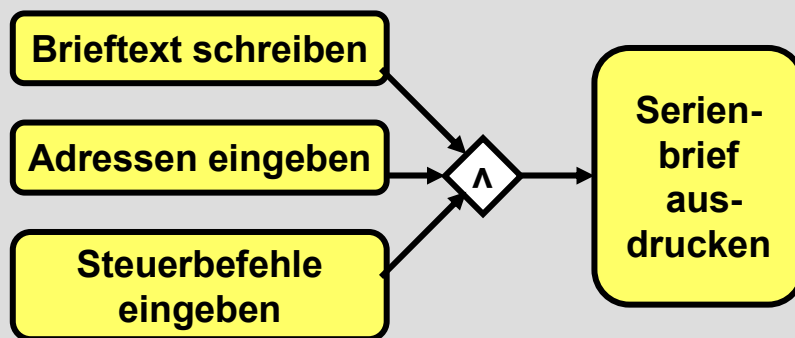
Beim Schreiben des Lehrbuchs wirkt immer ein Experte mit, während ein Grafiker nur gelegentlich benötigt wird.





## 5.2.3 Verknüpfung von Aktivitäten: UND-Beispiele

Mit dem UND-Konnektor kann man ausdrücken, dass mehrere Aktivitäten als Vorbedingung abgeschlossen sein müssen, bevor eine weitere ausgeführt werden kann. Ähnlich kann dargestellt werden, dass auf eine Aktivität mehrere andere folgen müssen.

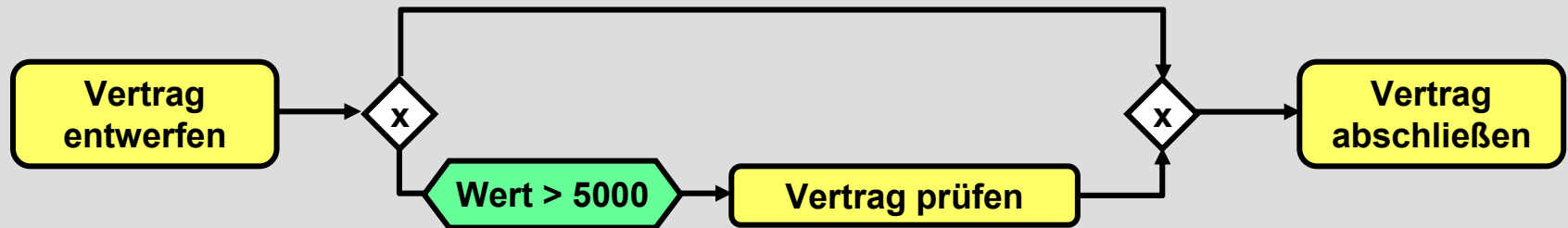


Das Ausdrucken eines Serienbriefes erfordert mehrere Vorbereitungsschritte, die alle ausgeführt sein müssen.

Die Aktivität etwas zu mieten zieht weitere nach sich: Das Gemietete wird bezahlt und benutzt.

## 5.2.3 Verknüpfung von Aktivitäten: XOR-Beispiel

Mit dem XOR-Konnektor kann man darstellen, dass auf eine Aktivität mehrere Alternativen folgen können, die sich wechselseitig ausschließen oder dass zwei sich ausschließende Wege Vorbedingung für eine weitere Aktivität sind.

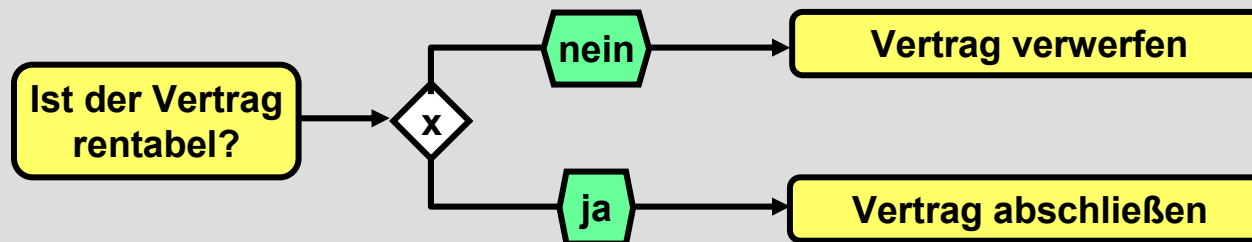


Hier folgt auf das Entwerfen eines Vertrages entweder eine eingehendere Prüfung (wenn der Wert über 5000 DM liegt) und danach der Vertragsabschluss oder der Vertrag wird bei geringerem Wert direkt abgeschlossen.



## 5.2.3 Verknüpfung von Aktivitäten: XOR-Beispiel

Die Aktivität vor einem XOR-Konnektor kann als Frage dargestellt werden, deren Beantwortung die Entscheidung bringt, welche Alternative weiterverfolgt wird.

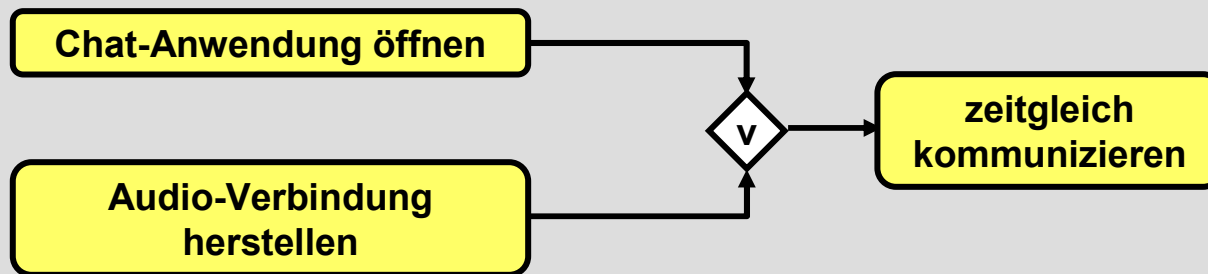


Die Entscheidung wird hier explizit in der Aktivität gefällt, die die Frage beantwortet, ob der Vertrag rentabel ist. Die nachfolgenden Modifikatoren spiegeln das Resultat dieser Entscheidung wider. Je nach dem wird abgeschlossen oder der Vertrag wird verworfen.



## 5.2.3 Verknüpfung von Aktivitäten: OR-Beispiel

Mit OR wird betont, dass mindestens eine aber auch mehrere Aktivitäten als Vor- oder Nachbedingung einer anderen Aktivität instantiiert werden können.



Um zeitgleich zu kommunizieren muss entweder eine Chat-Verbindung oder eine Audio-Verbindung etabliert sein. Beide Möglichkeiten schließen sich jedoch nicht aus, sondern können nebeneinander betrieben werden.

## 5.2.3 Mit UND verknüpfte Aktivitäten – verschiedene Bedeutungen

Wenn zwei Aktivitäten A und B (oder mehrere) miteinander mit UND verknüpft sind, also beide stattfinden müssen, so kann dies verschiedene Bedeutungen haben:

- » A und B sind nebenläufig, sie beeinflussen sich nicht gegenseitig (z.B. Atmen und Schreiben)
- » A und B sind voneinander abhängig:
  - sie müssen gleichzeitig erfolgen (z. B. Sprechen und Ausatmen) oder
  - sie können nicht gleichzeitig, aber in beliebiger Reihenfolge erfolgen
    - sie folgen als ganzes aufeinander ab (z.B. zwei Ausdauerübungen) oder
    - sie sind miteinander verschränkt (z.B. Aktivitäten zu Vorbereitung des Serienbriefes)

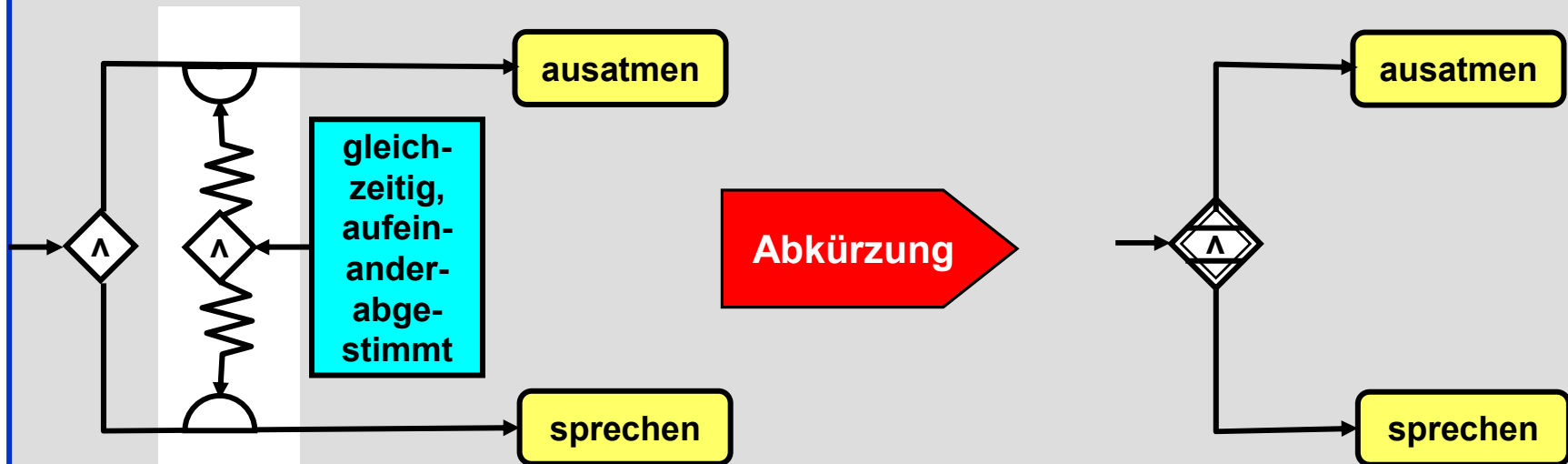


Der normale UND-Konnektor ist bzgl. dieser Bedeutung nicht festgelegt.

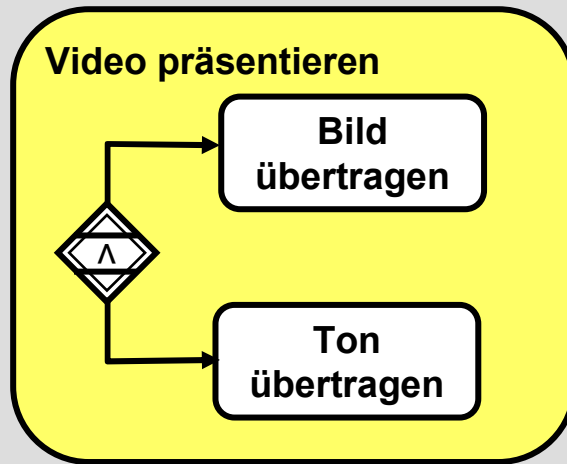


## 5.2.3 Verknüpfte Aktivitäten: synchrones UND

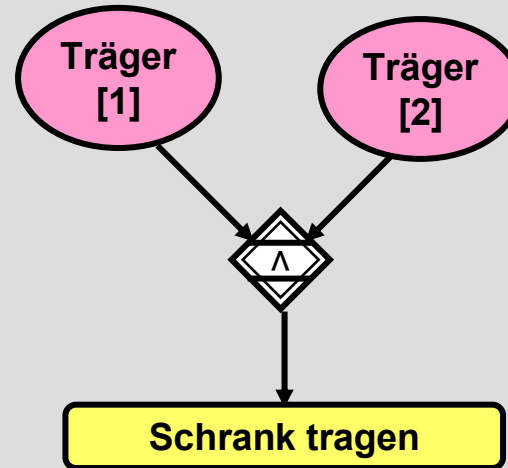
Bei Bedarf kann man spezielle UND-Konnektoren definieren, die das Verhältnis der verknüpften Aktivitäten zueinander festlegen. Hier wird zum Beispiel rechts ein synchrones UND dargestellt. Es stellt eine abgekürzter Darstellung eines Diagramms dar, bei dem man das Verhältnis der verknüpften Relationen durch eine nähere, mit UND-Verknüpfte Spezifikation der Relationen ausdrückt.



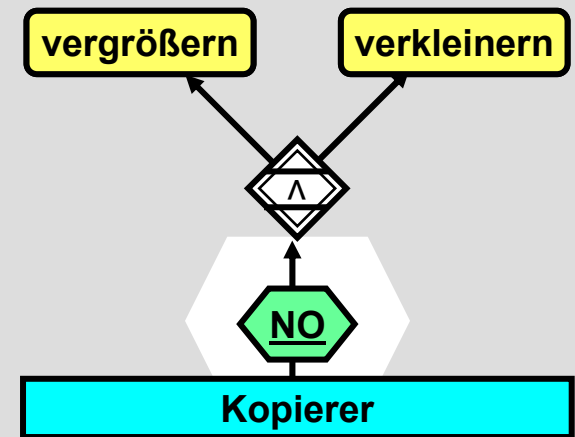
## 5.2.3 Synchrones UND - Beispiele



Bei der Präsentation eines Videos müssen der Ton und das Bild gleichzeitig übertragen werden.



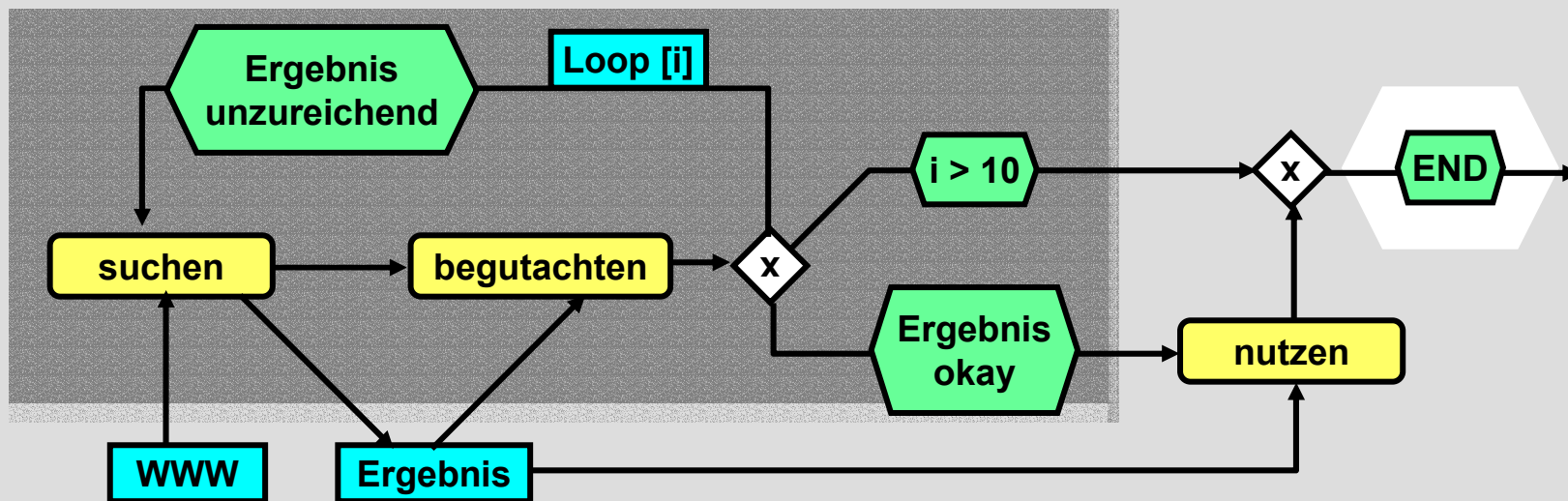
Beim Tragen eines Schrankes müssen beide Träger gleichzeitig aktiv sein.



Hier wird ausgedrückt, dass man mit einem Kopierer nicht gleichzeitig vergrößern und verkleinern kann.

## 5.2.3 Verknüpfung von Aktivitäten: Schleifen

Aktivitäten können durch zurückführende Relationen zu Schleifen gekoppelt werden. Solche Relationen sollte man besonders kennzeichnen. Durch XOR-Verzweigung kann die bedingte Wiederholung von Schleifen ausgedrückt werden.

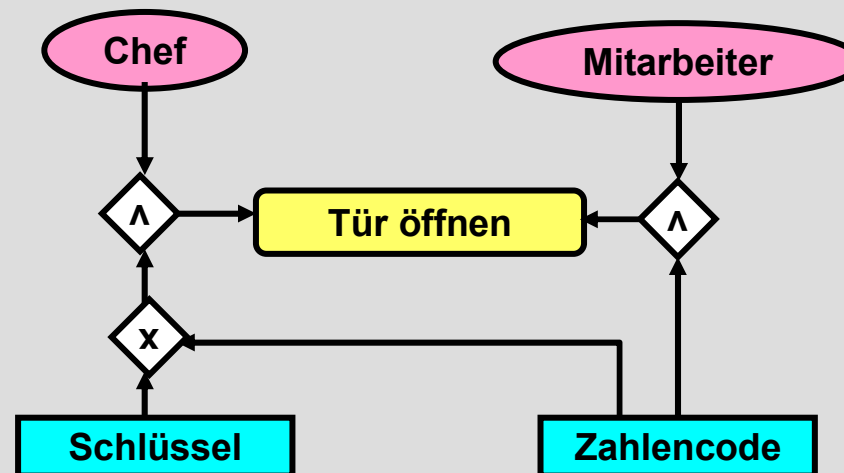


Das hellgraue Feld stellt eine bedingte Schleife dar. Das Suchen wird solange wiederholt, bis man mit dem Ergebnis zufrieden ist oder schon 11 mal gesucht hat. Die Relation, die die Schleife verursacht, wird LOOP genannt – ihr Vorkommen kann indiziert werden – der Index ist wiederverwendbar – z.B. in der Abbruchbedingung.



## 5.2.4 Verknüpfung unterschiedlicher Relationen I

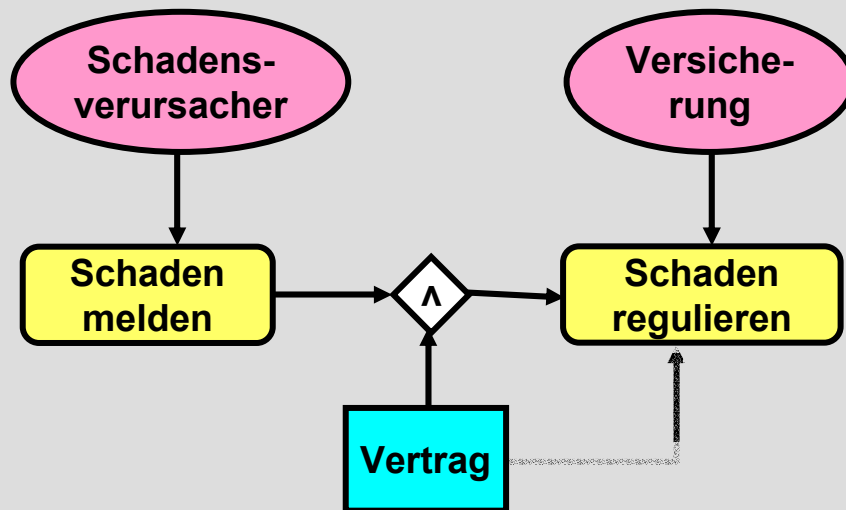
Man kann Relationen unterschiedlichen Typs verknüpfen – z.B. „Rolle führt Aktivität aus“ mit „Entität nutzen“. So kann ausgedrückt werden, dass unterschiedliche Entitäten zum Einsatzkommen, je nach dem wer die Aktivität ausführt.



Hier verfügt nur der Chef über einen physikalischen Türschlüssel, während die Mitarbeiter die Tür mit einem Zahlencode öffnen. Der Chef kann beides benutzen.

## 5.2.4 Verknüpfung unterschiedlicher Relationen II

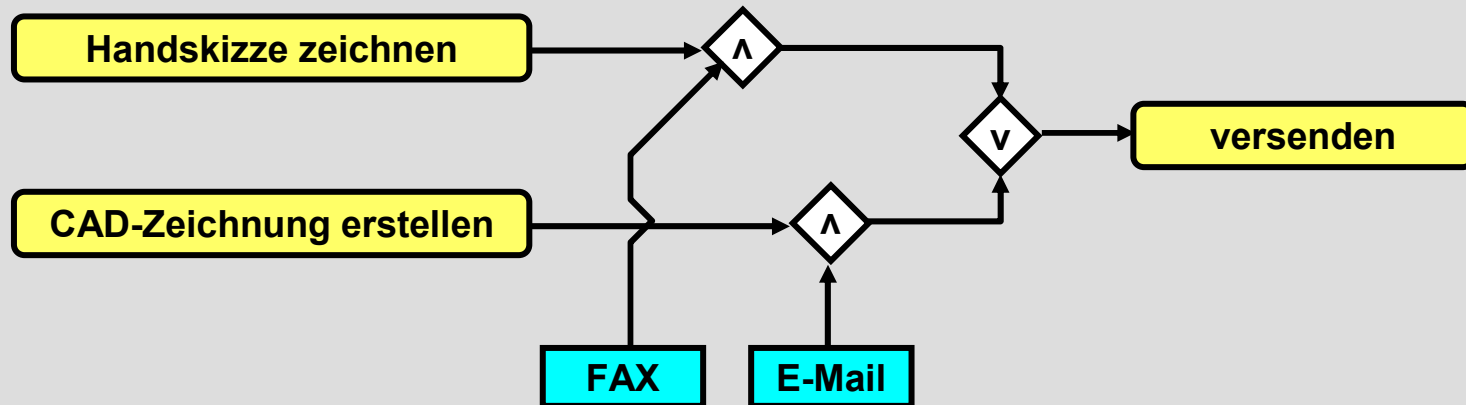
Man kann die Existenz einer Entität als weitere Vorbedingung einer Aktivität (neben der Durchführung einer vorhergehenden Aktivität) mit UND einfügen, bevor diese startet.



Die Versicherung wird nur aktiv, wenn der Schaden gemeldet wurde und ein Vertrag vorliegt. Der Vertrag wird als Grundlage der Schadensregulierung genutzt. Diese durch den grauen Pfeil ausgedrückte Relation kann entfallen, da sie schon implizit in dem Ergebnis der UND-Verknüpfung enthalten ist.

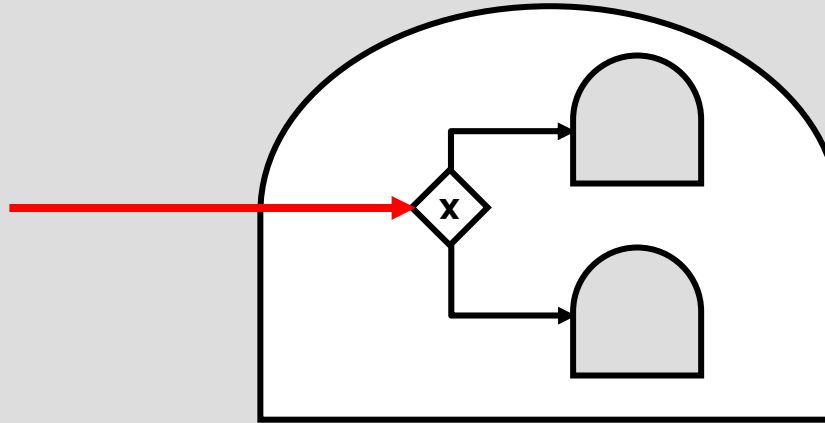
## 5.2.4 Verknüpfung unterschiedlicher Relationen III

Man kann auch ausdrücken, dass abhängig von den vorab stattgefundenen Aktivitäten unterschiedliche Entitäten genutzt werden.



Das Beispiel zeigt, dass man mit Hand verfertigte Skizzen per FAX versendet, während man eine mit CAD (Computer Aided Design) erzeugt Zeichnung besser als Attachment per E-Mail verschickt.

## 5.2.5 Konnektoren zwischen eingebetteten Elementen - Beispiel mit Aktivitäten

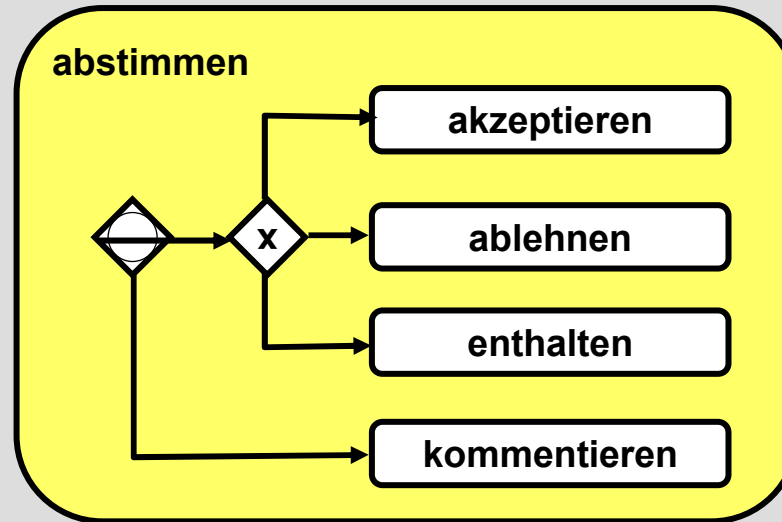


**Konnektoren können auch in Elemente eingebettet werden, um auszudrücken, in welchem logischen Verhältnis eingebettete Sub-Elemente zueinander stehen.**

Der rote Pfeil kann aus Vereinfachungsgründen weggelassen werden. Alle Pfeile, die mit einem Basis-Element verbunden sind, werden zu den Konnektoren hin verlängert, zu denen sie aufgrund ihrer Richtung passen.



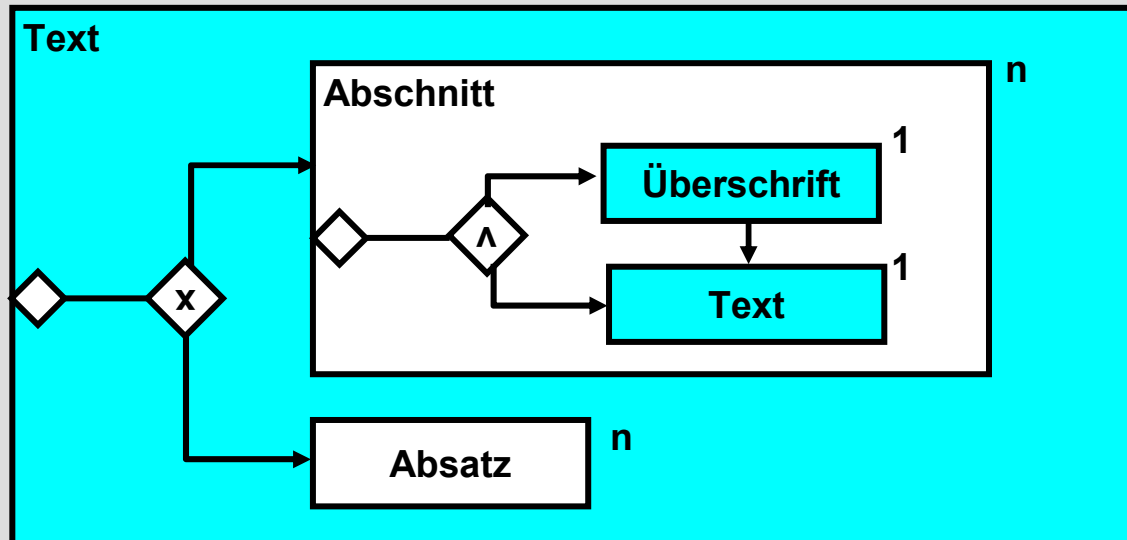
## 5.2.5 Konnektoren zwischen eingebetteten Elementen - Beispiel mit Aktivitäten



Hier werden Konnektoren in ein Element eingebettet, um auszudrücken, dass die Aktivität des Abstimmens alternativ durch die Sub-Aktivitäten akzeptieren, ablehnen oder sich enthalten vollzogen wird. Optional kann ein Kommentar hinzugefügt werden.

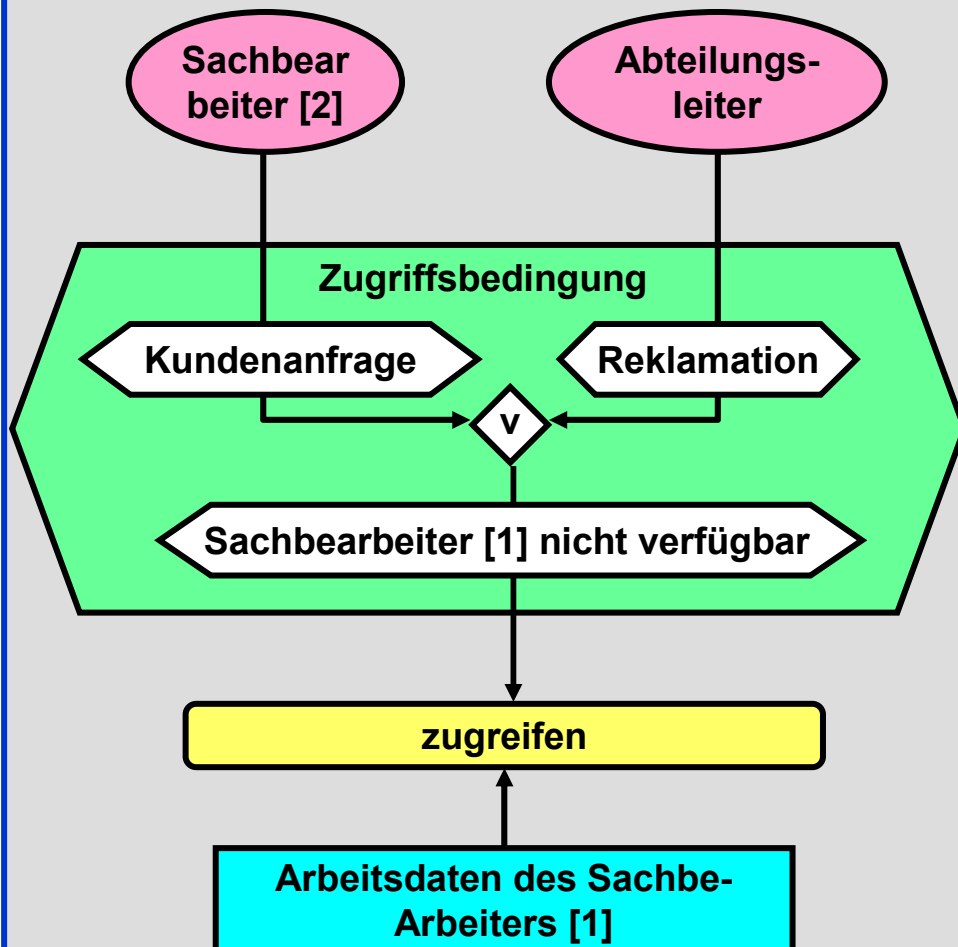


## 5.2.5 Konnektoren zwischen eingebetteten Elementen - Beispiel mit Entitäten



Hier wird der Aufbau eines Textes rekursiv definiert. Der Text besteht entweder aus Absätzen oder aus Abschnitten. Ein Abschnitt besteht sowohl aus einer Überschrift als auch aus einem Text, der der Überschrift zugeordnet ist.

## 5.2.5 Konnektoren zwischen eingebetteten Modifikatoren

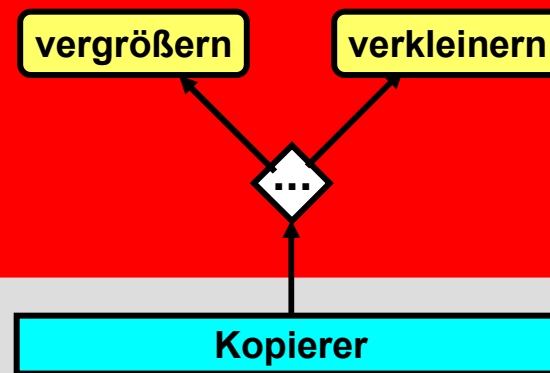


Man kann Sub-Modifikatoren in Super-Modifikatoren logisch miteinander verknüpfen. Dadurch lassen sich komplexe Geflechte von Bedingungen erzeugen.

Hier wird gezeigt, dass mit unterschiedlichen Rollen unterschiedliche Zugriffsbedingungen verknüpft sind. Beide Bedingungen greifen aber nur, wenn der vom Zugriff betroffene Sachbearbeiter nicht verfügbar ist.

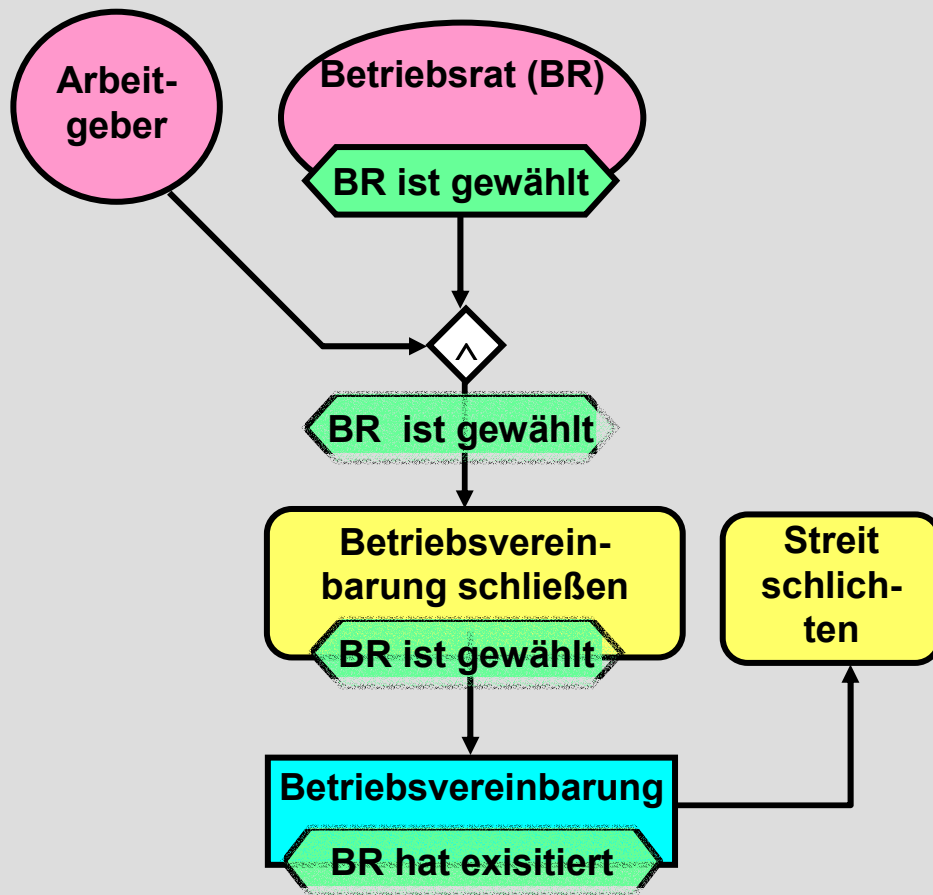
## 5.2.6 Aufgaben zu Konnektoren

1. Drücken Sie aus, dass eine Qualitätskontrolle immer von zwei Sachbearbeitern gemeinsam durch geführt werden muss, die gelegentlich von ihrem Vorgesetzten dabei unterstützt werden.
2. Stellen Sie ein aus XOR- und UND-Konnektoren bestehendes Diagramm dar, das dieselbe Bedeutung wie ein zusammenführendes Diagramm mit OPT-Konnektor (siehe [5.2.1 OPT-Konnektor, oben links](#))
3. Drücken Sie den Fall aus, dass das Bestellen über zwei Kommunikationswege zwar möglich, aber unerwünscht ist ([siehe 5.2.2 XOR vs. OR, links](#)).
4. Stellen Sie Beispiele für folgende Konstellationen dar: „Eine Rolle darf eine Entität nur bearbeiten, wenn diese Daten über sie enthält.“ und „Eine Rolle hat auf bestimmte Entitäten nur dann Zugriff, wenn diese Daten über sie enthalten“
5. Ersetzen Sie im nebenstehenden Diagramm die drei Punkte durch UND oder XOR und interpretieren Sie das Ergebnis aus einer Prozess- und einer Struktursicht.



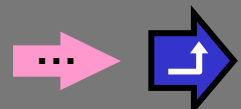
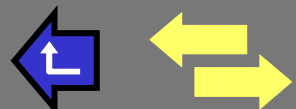


## 5.3 Propagieren von Bedingungen



Wenn man ein Diagramm instantiiert oder realisiert, dann muss man prüfen, wie sich die eingetragenen Bedingungen in Abhängigkeit von der logischen Verzweigung ggf. auf andere Elemente übertragen.

Bei dem Beispiel ist zu beachten, dass auch das gemeinsame Abschließen einer Betriebsvereinbarung ohne gewählten Betriebsrat nicht möglich ist. Die Existenz einer Vereinbarung setzt jedoch nur voraus, dass es einmal eine Betriebsrat gegeben hat. Das Streit-schlichten kann auch ohne Betriebsrat von einer Betriebsvereinbarung profitieren. Die propagierten Bedingungen sind transparent dargestellt.



## 6) Attribute

Attribute werden benötigt um die Eigenschaften von Basis-Elementen bzw. deren Veränderungen durch Werte der Attribute darzustellen. Ein Attribut besteht also immer aus zwei Teilen, die durch Doppelpunkt getrennt sind.

**Attributname: Wert (z.B.: Farbe: Grün)**

Bei Auflistung mehrerer solcher Angaben werden diese durch Semikolon getrennt.

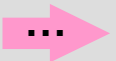
Die Angabe des Wertes kann in sich komplex strukturiert sein. Es kann ein einzelner oder mehrere Werte angegeben werden oder ein Wertebereich. Diese können durch Angabe einer Maßeinheit ergänzt werden. Wertangaben können auch selbst wieder Attribute enthalten.

Attribute können auch als Entitäten dargestellt werden.



## 6) Attribute - Beispiele

Häufigkeit: 5;  
Dauer: 10 min;  
Anzahl: 3-8;  
Zuverlässigkeit: hoch;  
Größe: [Höhe: 1,5 m, Tiefe: 50 cm, Breite: 2 m];  
Verwandte: {Geschwister, Kinder, Eltern, Großeltern}



## 6) Struktur von Attributen

Person;  
Alter: 20-30;  
Beruf: Kaufmann;  
Familienstand

schreiben;  
Dauer: >2 Stunden;  
Schriftart: {Times, Arial};  
Schreibinstrument:  
Computer

Bild;  
Art: {statisch, bewegt};  
Farbe: {ja, nein};  
Entstehungsart: {Abbild,  
Phantasie}

Attributname und Wert(e) werden durch Doppelpunkt getrennt, mehrere Attribute durch Semikolon.

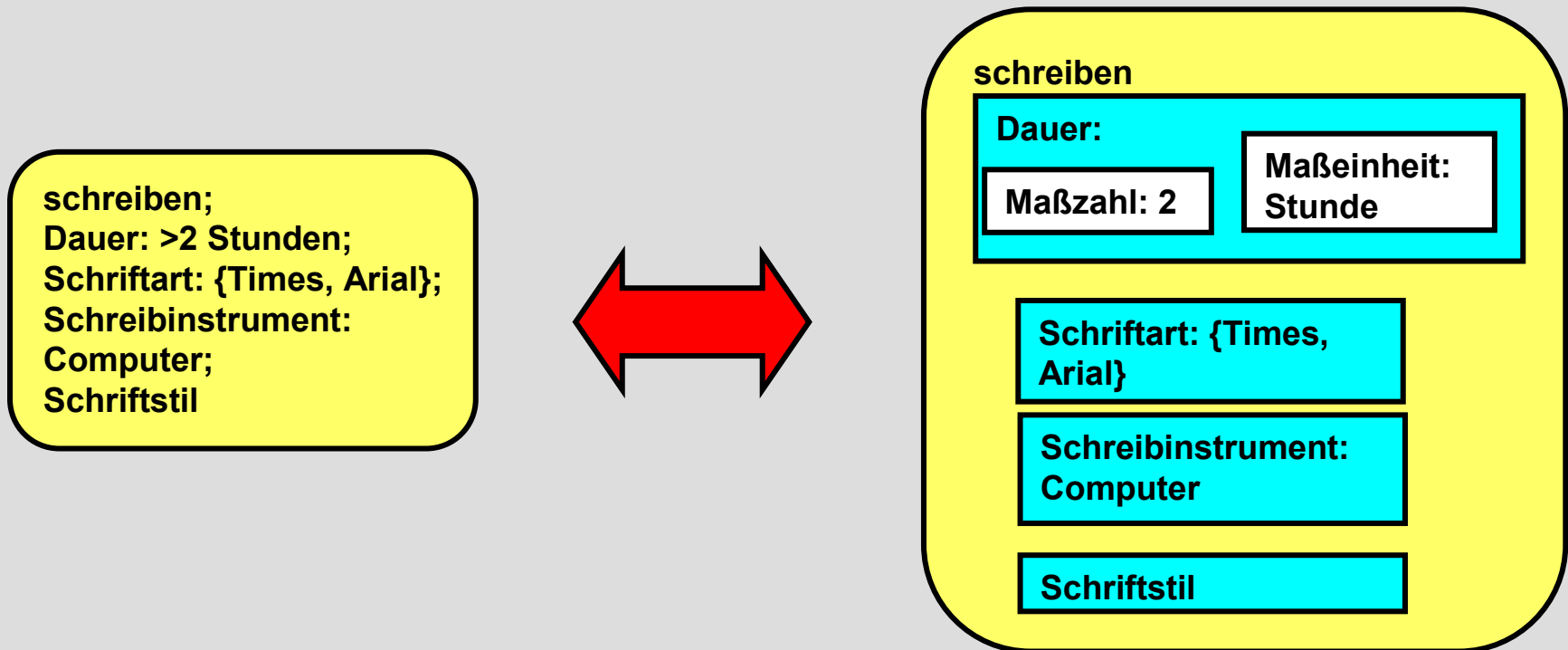
Bei manchen Attributen ist die Maßeinheit des Wertes durch die Art des Attributes klar (z.B. bei Alter), bei anderen muss sie gesondert festgelegt werden (z.B. bei Dauer). Es können Wertemengen (z.B. bei Schriftart) oder Wertebereiche (z.B. bei Alter) angegeben werden. Der Attributname kann schon eine bestimmte Wertbelegung andeuten, die dann durch JA oder NEIN spezifiziert wird (z.B. wird bei Farbe angegeben, ob Farbe vorhanden ist oder nicht).

Unter Umständen gibt man nur das Attribut an, ohne die Werte oder deren Bereich zu spezifizieren, um darauf hinzuweisen, dass hier relevante Unterschiede vorliegen (z.B. Schreibinstrument).

Bei den durch Attribute spezifizierten Basis-Elementen handelt es sich stets um Klassen und nicht um einzelne, konkrete Phänomene. Also werden hier abstrakt alle Person mit dem Beruf Kaufmann zwischen 20 und 30 dargestellt; das Attribut „Familienstand“ wird erst bei der Instantiierung konkretisiert.



## 6) Attribute als Entitäten



Die einzelnen Attribute können samt ihren Werten als Attribute dargestellt werden. Dies ist besonders hilfreich, wenn man darstellen will, dass bestimmte Aktivitäten die Werte von Attributen verändern. Da Werte selbst wieder Attribute enthalten können, kann eine eingebettete Struktur entstehen, die mit Hilfe von Entitäten leichter verdeutlicht werden kann, wie am Beispiel der Dauer gezeigt. Wenn man das Bedingungs/Ereignisattribut eines Modifikators als Entität darstellt, dann kann diese Bedingung durch eine Aktivität bearbeitet werden.

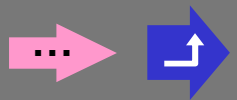
## 6) Abkürzungen bei Attributen

Text;  
Farbe: ja;  
Abbildungen: vorhanden;  
Länge: 10000 Zeichen;  
Erstellungsdatum: 1.1.2000

**Abkürzung**

Text;  
Farbig;  
Mit Abbildungen;  
10000 Zeichen;  
Erstellt am 1.1.2000

Wenn die Wertausprägung eines Attributes so beschrieben werden kann, dass man die Eigenart des Attributes daraus zu erkennen vermag, dann kann auf die Nennung eines Attributnamens verzichtet werden.

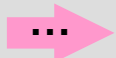


## 6) Attribute und Modifikatoren

Modifikatoren werden mit spitzen Klammern an Attribute angefügt. Bedingungen/Ereignissen einerseits und Häufigkeitsangaben andererseits werden durch einen senkrechten Strich getrennt.

Größe: [Höhe: ; Breite: ; Tiefe<bei dreidimensionalen Gebilden>: ]

Größe: [Höhe: 15 feet <in USA>; Tiefe: 50 cm;  
Breite: 2 <bei Übergröße|selten> m]



## 6) Attribute - Syntax



**Selten  
benötigt**

Die Art und Weise, wie Wertangaben syntaktisch strukturiert werden und wie ihre Umwandlung in Entitäten vorgenommen wird, kann durch eindeutige Regeln beschrieben werden. Da komplexere Attribut-Angaben nur selten benötigt werden, wird auf die Beschreibung dieser Regeln hier nicht näher eingegangen.

Der Modellierer kann meistens eine Form der Darstellung von Werten wählen, aufgrund derer zwischen Attributname, Wertangabe und Maßeinheit unterschieden werden kann.





## 6) Aufgaben zu Attributen

1. Definieren Sie zu der Entität „Auto“ das Attribut Fahrverhalten durch Sub-Attribute und die dazugehörigen Wertebereiche
2. Beschreiben Sie die Klasse aller Aufgaben, die arbeitspsychologisch sinnvoll gestaltet sind, durch eine Aktivität, bei der sich hohe und niedrige Planungsanforderung abwechseln, hohe Variabilität und Entscheidungsfreiheit herrscht, sich hohe und niedrige Belastung die Waage halten und Qualifikation abgefordert werden.
3. Beschreiben Sie die Eigenschaften eines Teams durch Attribute
4. Geben Sie 3 Beispiele, in denen sowohl die Werteangabe als auch die Maßeinheit bedingt ist.



## 7. Vagheit - Einführung

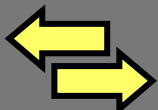
Ein wesentlicher Aspekt sozio-technischer Systeme besteht darin, dass viele Zusammenhänge und Strukturen nicht formal festgelegt sind, sondern nur informal bestimmt sind. Viele Eigenarten von Strukturen oder Prozessen werden von Fall zu Fall oder von Zeitpunkt zu Zeitpunkt unterschiedlich realisiert. Deshalb ist es wichtig, dass man Vagheit (Unvollständigkeit und Unsicherheit) in den Diagrammen darstellen kann.

Unter **Vagheit** verstehen wir **Unvollständigkeit** und/oder **Unsicherheit**.

Unvollständigkeit und Unsicherheit können sich im wesentlichen beziehen auf:

- Sub-Elemente eines Basis Elementes
- Auf die Verankerung und Spezifizierung von Relationen
- Auf die Wertebelegung von Attributen und damit auch auf die Spezifizierung von Modifikatoren
- Auf die Spezifizierung von Konnektoren

Zur Gliederung



## 7. Vagheit - Darstellungsprinzipien

Bei SeeMe kommen immer wieder die gleichen Symbole oder Indikatoren für Unvollständigkeit oder Unsicherheit zum Einsatz. Wir unterscheiden zwischen **gewollter** und **gegebener** Vagheit.

Wir unterscheiden folgende Indikatoren und ihre Bedeutung:

### Gewollte Vagheit:

**Leere Flächen** (leere Halbkreise, leere Modifikatoren oder Konnektoren, im Leeren verankerte Relationen): Der Modellierer will sich nicht um weitere Details kümmern oder hält deren Festlegung nicht für sinnvoll.

**Schwarze Flächen** (schwarze Halbkreise, Modifikatoren etc., verdickte Relationsenden): Hinter diesen aktivierbaren Flächen verbergen sich Verweise auf erweiterte Darstellungen

„+“ : Der Modellierer hat absichtlich nicht mehr dargestellt, weiß aber mehr.

### Gegebene Vagheit:

„...“: Es wäre sinnvoll mehr darzustellen, der Modellierer weiß aber zum Zeitpunkt der Modellierung nicht mehr.

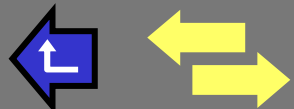
„?“: Das mit Fragezeichen versehene Detail ist eventuell nicht korrekt.

„???“: Es ist unsicher, ob man noch weitere Details festlegen müsste oder nicht.



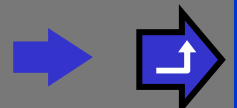
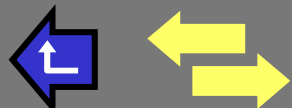
## 7. Vagheit - Gliederung

- 7.1 Vage Spezifizierung von Sub-Elementen eines Elementes
- 7.2 Vage Verankerung des Start- oder Endpunktes einer Relation
- 7.3 Kombinierte Vagheit
- 7.4 Vage Spezifizierung Ankerpunktes einer Relationen
- 7.5 Vage Spezifizierung des Auftretens von Relationen
- 7.6 Vage Spezifizierung von Relationen durch weitere Basis-Elemente
- 7.7 Vage Spezifizierung von Modifikatoren
- 7.8 Vage Spezifizierung von Konnektoren
- 7.9 Vage Spezifizierung von Attributen
- 7.10 Kombination von Vagheitsangaben
- 7.11 Ausdrückliche Vollständigkeit

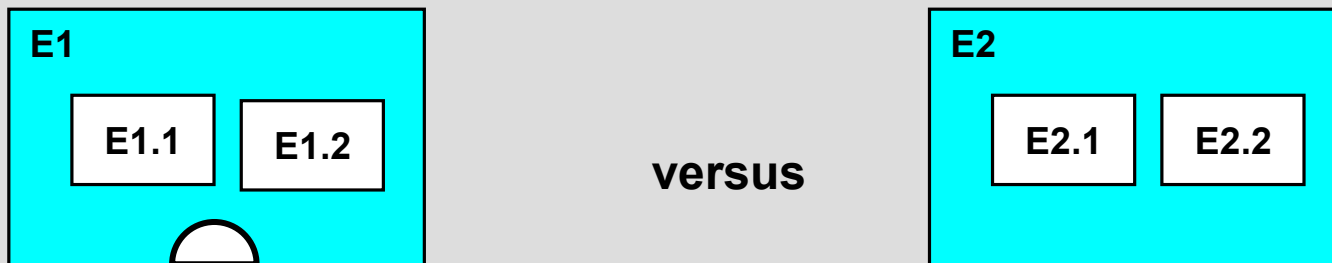


# 7.1 Vage Spezifizierung von Sub-Elementen eines Elementes

- 7.1.1 Gründe und Besonderheiten der vagen Festlegung von einer Struktur von Sub-Elementen
- 7.1.2 Unterschiedliche Arten von Vagheit bei Sub-Elementen
- 7.1.3 Vagheit und Segmentlinien
- 7.1.4 Abkürzung
- 7.1.5 Aufgaben zu Vagheit bei Sub-Elementen



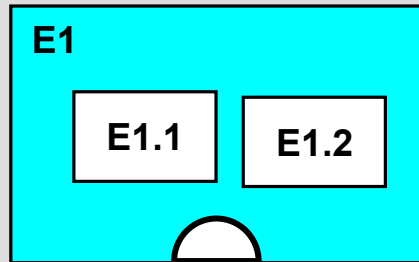
## 7.1.1 Wann braucht man Indikatoren für Vagheit bei Sub-Elementen



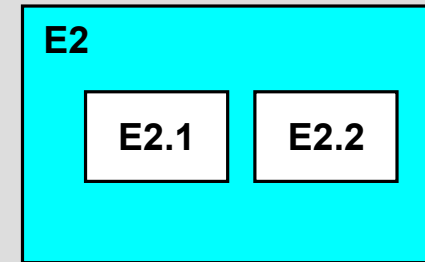
Der leere Halbkreis im linken Diagramm symbolisiert, dass die Festlegung der Sub-Elemente (E1.1 und E1.2) nicht vollständig ist. Die Festlegung weiterer Elemente des gleichen Typs (hier also Entitäten E1.3 etc.) wird nicht als sinnvoll für das Modell erachtet bzw. nicht gewollt.

Die Menge der für E2 festgelegten Sub-Elemente wird dagegen als vollständig angesehen.

## 7.1.1 Gründe für Vagheit bei Sub-Elementen

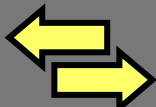


versus

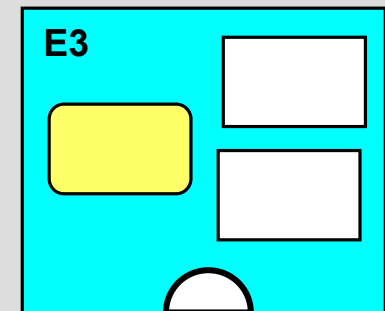
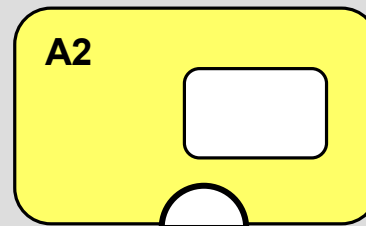
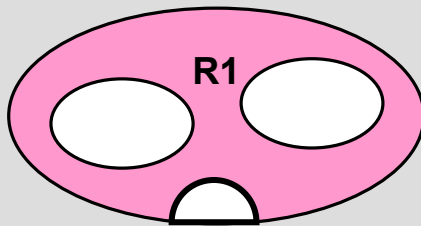


**Es gibt verschiedene Gründe, warum man den leeren Halbkreis benutzt: Zum Beispiel**

- weil man keine weiteren Elemente für die Darstellung von Zusammenhängen im Modell braucht,
- weil weitere Elemente das Modell unnötig überladen,
- weil die Menge der zu ergänzenden Elemente mit jeder Instantiierung anders aussehen könnte,
- weil man nicht den Aufwand betreiben will, zu untersuchen, welche weiteren Sub-Elemente es gibt.



## 7.1.1 Vage Spezifizierung von Sub-Elementen eines Elementes

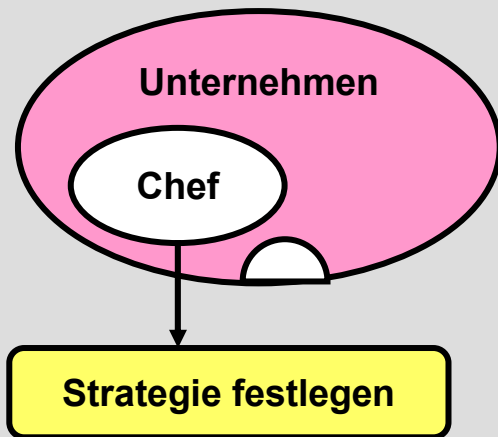


Der Halbkreis bezieht sich – auch bei den weiteren Symbolen – immer nur auf das mögliche Fehlen solcher Sub-Elemente, die vom gleichen Typ sind wie das Super-Element, hier also R1, A2 oder E3. Es wird also nicht über das Fehlen von Elementen eines anderen Typs ausgesagt. Zum Beispiel wird nicht angedeutet, dass etwa bei E3 noch weitere Sub-Aktivitäten fehlen.

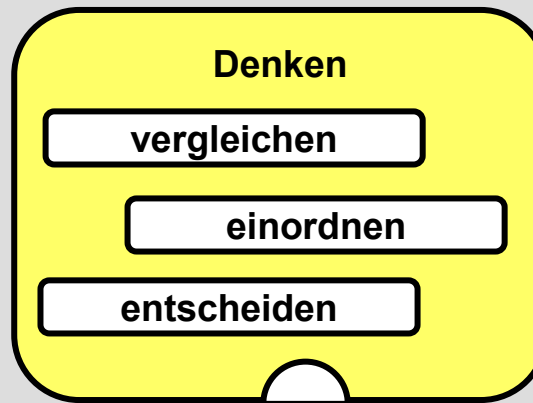
**Achtung:** Die nicht dargestellten Sub-Elemente können potentiell mit Relationen und Konnektoren untereinander und zu den gezeigten Sub-Elementen verbunden sein.



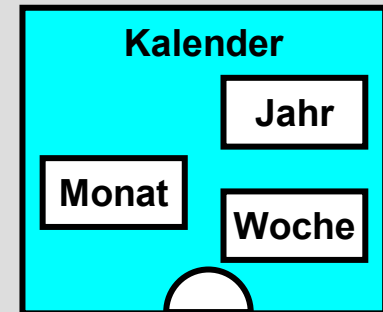
## 7.1.1 Vage Spezifizierung von Sub-Elementen eines Elementes - Beispiele



Hier geht es nur darum zu zeigen, dass es der Chef ist, der die Strategie im Unternehmen festlegt. Andere Rollen im Unternehmen sind daher in diesem Diagramm uninteressant

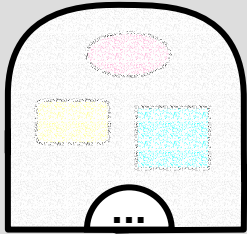


Es ist offensichtlich, dass es noch weitere Sub-Aktivitäten des Denkens gibt und dass es kaum möglich ist, diese abschließend aufzuzählen.

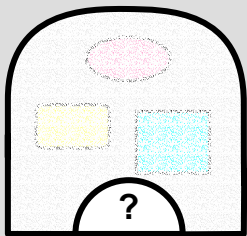


Es gibt viele Möglichkeiten, einen Kalender zu unterteilen. Der Modellierer deutet an, dass er dies weiß und dass er sich absichtlich auf die gezeigte Einteilung beschränkt.

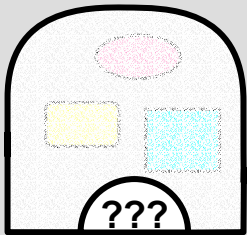
## 7.1.2 Unterschiedliche Arten gegebener Vagheit



Es wurde festgestellt, dass die Menge der festgelegten Sub-Elemente, die vom gleichen Typ wie das Super-Element sind, sicherlich unvollständig ist, man weiß aber nicht, wie man sie vollständig spezifizieren könnte.



Man vermutet, dass die gewählte Menge der Sub-Elemente bzgl. ihrer Struktur, ihres Umfanges oder bzgl. einzelner Sub-Elemente nicht korrekt ist oder nicht für jede Instantiierung angemessen ist (Unsicherheit).



Es kann sein, dass die Menge der festgelegten Sub-Elemente zu erweitern ist, um das Super-Element angemessen zu beschreiben, man ist aber bzgl. dieser Erweiterungsnotwendigkeit nicht sicher.

## 7.1.2 Unterschiedliche Arten gegebener Vagheit - Beispiele

### Besprechung moderieren

fragen

klären

Bewertung  
initiiieren

strukturieren

...

Eine verbindliche Liste aller wesentlichen Sub-Aktivitäten des Moderierens (neben den gezeigten Kernaktivitäten) ist nicht bekannt.

### Software-Ergonomie Kriterien nach DIN

Fehlerrobustheit

Aufgabenangemessenheit

Selbstbeschreibungsfähigkeit

Erwartungskonformität

Steuerbarkeit

???

Seit der Diskussion um eine internationale Norm (ISO) wurde die Vollständigkeit der gezeigten DIN-Kriterien in Frage gestellt.

### Qualifikation

Fähigkeit

Fertigkeit

Kenntnisse

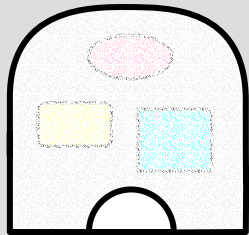
Soziale  
Kompetenz

?

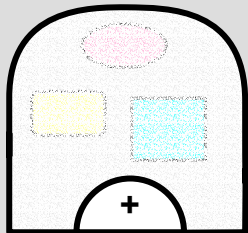
Es gibt so viele Möglichkeiten, das Phänomen Qualifikation zu gliedern, dass man die Angemessenheit der gezeigten Aufteilung bezweifeln kann.



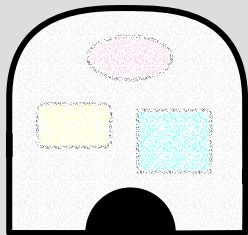
## 7.1.2 Unterschiedliche Arten gewollter Vagheit



Zu der bereits dargestellten Definition des leeren Halbkreises ist zu ergänzen, dass der Modellierer mit diesem Symbol auch absichtlich ungeklärt lässt, dass eine Möglichkeit der dargestellten gegebenen Vagheit zutrifft.



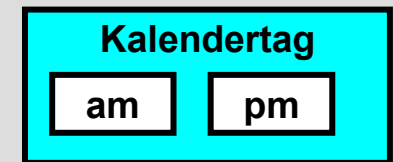
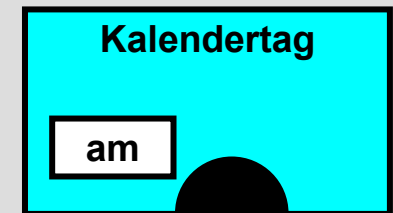
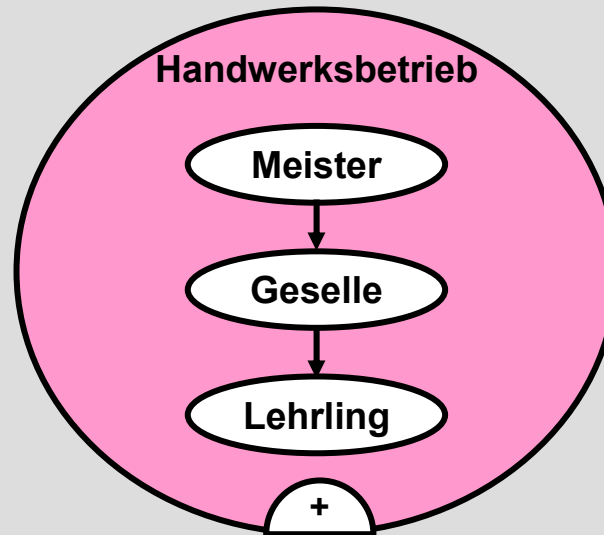
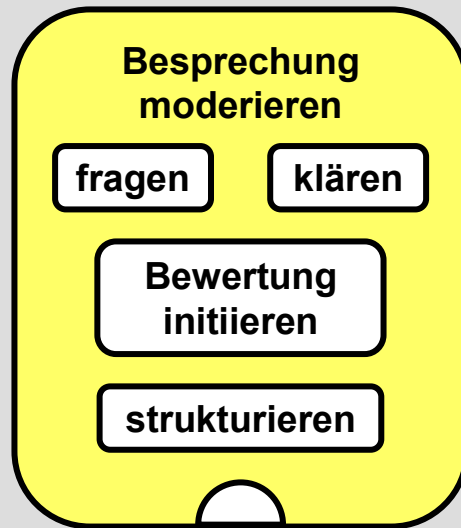
Der Modellierer hat zusätzliche Informationen über Sub-Elemente, die er absichtlich nicht zeigt, aber die er mitteilen kann.



Der schwarze Halbkreis beinhaltet Referenzen bzw. Links zu weiteren Information über das Super-Element. Bei computer-gestützten Präsentationssystemen kann die schwarze Fläche aktiviert werden. ([siehe auch Kap. 8](#))



## 7.1.2 Unterschiedliche Arten gewollter Vagheit - Beispiel

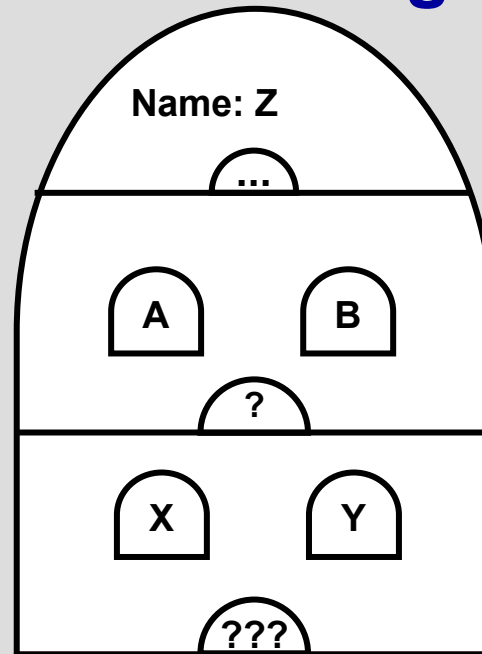


Der Modellierer möchte in dem Modell nicht die Tatsache vertiefen, dass man über die vollständige Liste der moderierenden Sub-Aktivitäten nicht genügend weiß. Der leere Halbkreis beinhaltet jedoch diese Möglichkeit.

Der Modellierer möchte die zentrale hierarchische Struktur darstellen und geht daher auf Rollen wie Buchhaltung oder Fahrer, die er kennt, nicht ein.

Die vollständige Einteilung des Kalendertags wird gezeigt, wenn man die schwarze Fläche aktiviert.

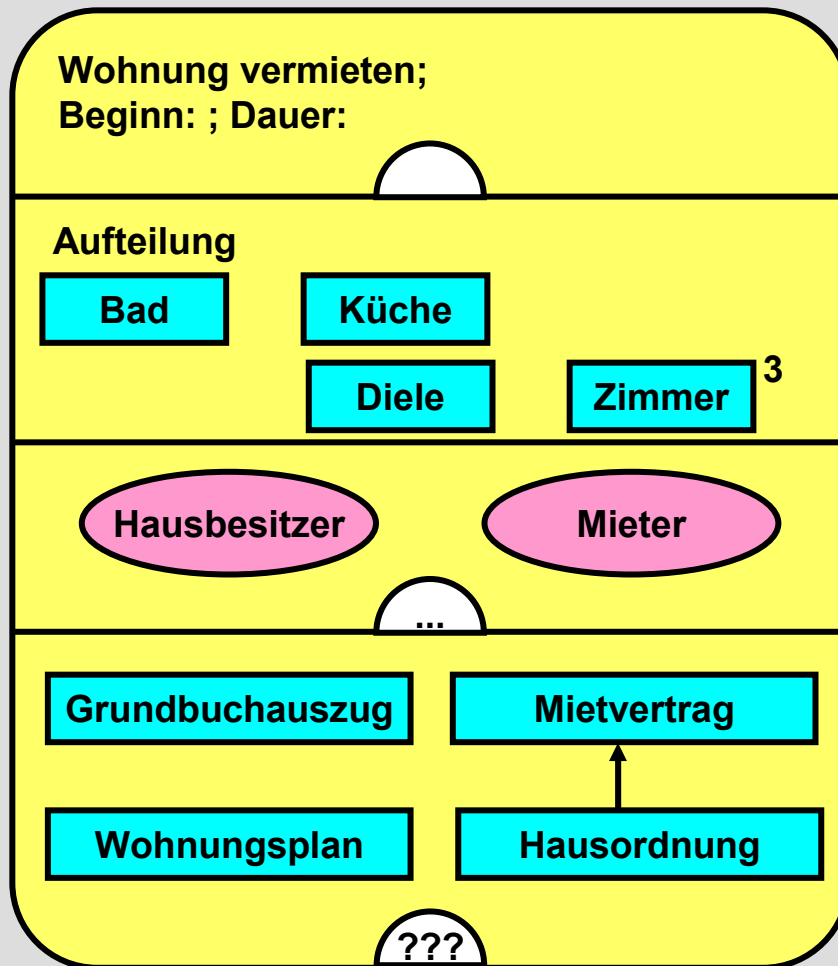
## 7.1.3 Vagheit und Segmentlinien



Man kann unterschiedliche Vagheitssymbole zu unterschiedlichen Gruppen von Sub-Elementen anmerken, indem man Segmentlinien benutzt. Auf diese Weise kann man auch Unvollständigkeit bei Sub-Elementen anmerken, die einen anderen Typ als das Super-Element haben. Zu beachten ist, dass sich die drei Punkte auf Attribute von Z beziehen und die ??? beziehen sich nur auf die Gruppe der Sub-Elemente X und Y.



## 7.1.3 Vagheit und Segmentlinien - Beispiel



Das Diagramm zeigt, dass die Liste der Attribute dieser Aktivität bewusst unvollständig ist.

Nur die Aufteilung der Wohnung ist vollständig dargestellt.

Es ist noch zu erkunden, ob noch weitere Rolle bei der Aktivität vermieten im Spiel sind (etwa Untermieter)

Es ist fraglich, ob die Liste der für die Aktivität relevanten Dokumente vollständig ist.

## 7.1.4 Abkürzung I

Name der Aktivität

Abkürzung

Name der Aktivität

Falls kein einziges Sub-Element gezeigt wird, ist die gewollte Unvollständigkeit offensichtlich und der leere Halbkreis kann entfallen. Damit wird für jedes Basis-Element ohne Sub-Elemente ausgesagt, dass es sich um eine absichtlich reduzierte Darstellung handelt.

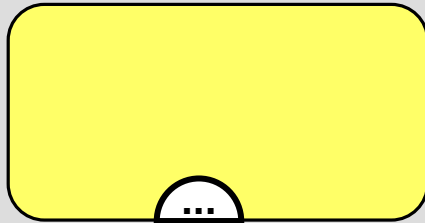
Abkürzung

Man kann auch – ohne Verwendung einer Segmentlinie mit leerem Halbkreis – den Namen eines Basis-Elementes entfallen lassen, um so die völlig unspezifizierte Existenz eines Elementes – hier einer Rolle – anzudeuten.

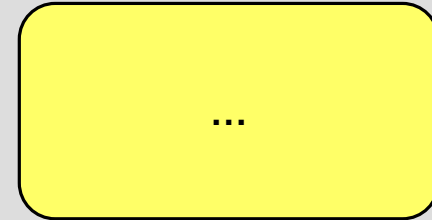




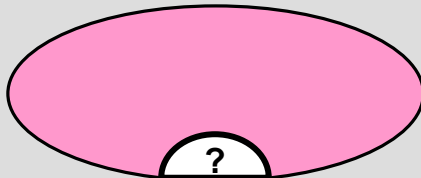
## 7.1.4 Abkürzung II



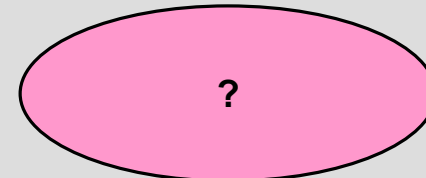
Abkürzung



Man weiß, dass auf jeden Fall eine Aktivität ausgeführt wird, aber man weiß nicht, wie man den Namen oder Sub-Aktivitäten spezifizieren kann. In diesem Fall können die drei Punkte direkt in die Aktivität eingetragen werden.



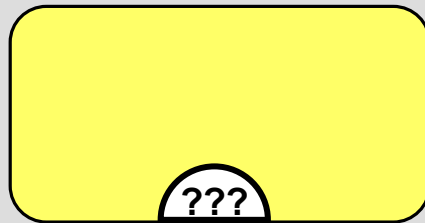
Abkürzung



Der Name und Sub-Rollen einer Rolle werden nicht spezifiziert, weil dies nicht als nicht sinnvoll angesehen wird (z.B. weil eine Aktivität ständig von einer andere Rolle ausgeführt wird). Das Fragezeichen bedeutet, dass man sich nicht sicher ist, ob das Fehlen der Spezifizierung korrekt ist.



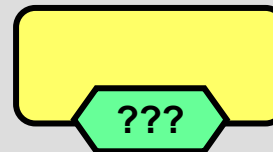
## 7.1.4 Abkürzung III



Abkürzung



Wenn man sich nicht sicher ist, ob die Spezifizierung eines Namens und von Sub-Aktivitäten überhaupt möglich ist, dann können die drei Fragezeichen direkt eingetragen werden.

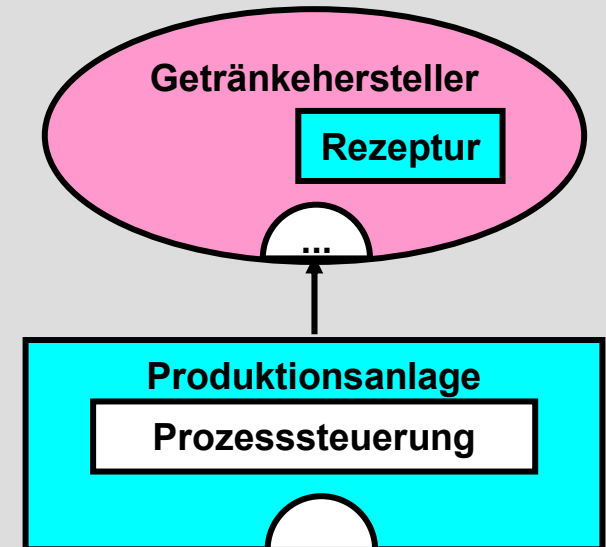
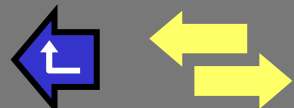


Diese Darstellung bedeutet dagegen, dass man es nicht für sinnvoll hält, die gezeigte Aktivität näher zu spezifizieren UND dass es nicht sicher ist, ob es überhaupt sinnvoll ist, diese Aktivität in das Modell einzutragen, also so ob es sie überhaupt gibt ([siehe Vagheit bei Modifikatoren](#))

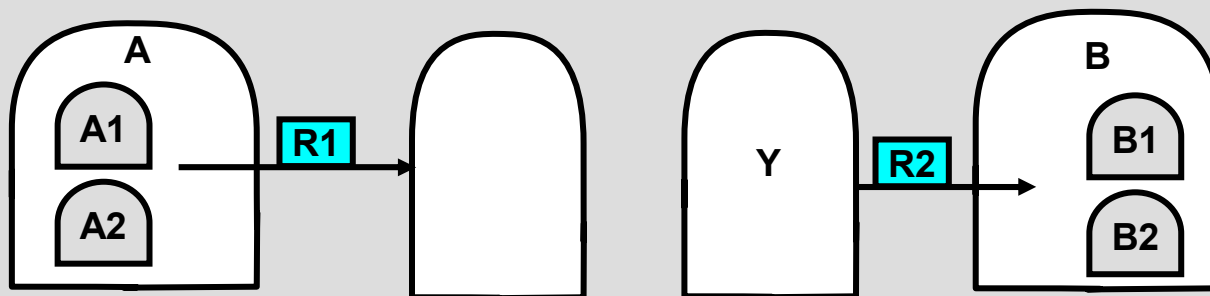


## 7.1.5 Aufgaben zu Vagheit bei Sub-Elementen

1. Erläutern Sie die Bedeutung des gezeigten Diagramms.
2. Ersetzen Sie bei dem Beispiel zur Einbettung von Entitäten in Entitäten die Entität „Sonstiges“ bei „Büro“. Ändern Sie die Entität „Multimedia Präsentation“ so ab, dass zum Ausdruck kommt, dass die Angemessenheit der Aufteilung in Text, Bild und Ton angezweifelt wird und dass die Unterteilung von „Ton“ eventuell unvollständig ist.
3. Drücken Sie bei dem Beispiel zur Einbettung von Aktivitäten in Rollen aus, dass die Menge der Aktivitäten in der Rolle „Person“ eventuell unvollständig ist. Stellen Sie bei Werkstatt dar, dass noch mehr Rollen und Aktivitäten relevant sein können, die man absichtlich nicht darstellt.

**Aufg.**

## 7.2 Vage Verankerung des Start- oder Endpunktes einer Relation



Die Darstellung drückt aus, dass für R1 nicht festgelegt wird wo R1 startet. Der Startpunkt kann bei A insgesamt liegen oder bei A1 und/oder A2. Er kann sogar bei Sub-Elementen von A1 oder A2 liegen. Unter Umständen variieren die Bedingungen für die Verankerung des Startpunktes mit jeder Instantiierung.

Das gleiche gilt für den Endpunkt von R2 bzgl. B, B1 und B2.

A, A1 und A2 müssen vom gleichen Typ sein, ebenso B, B1 und B2.

7.2.1

Definitionen

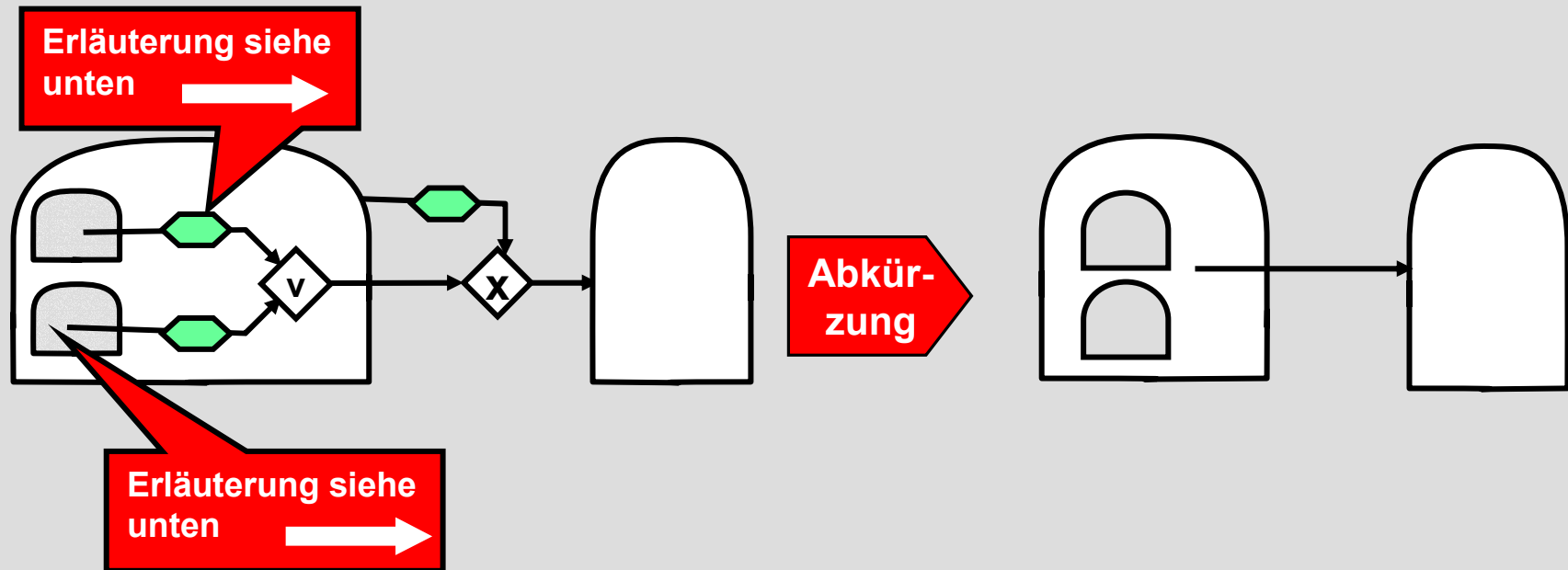
7.2.2

Beispiele

7.2.3

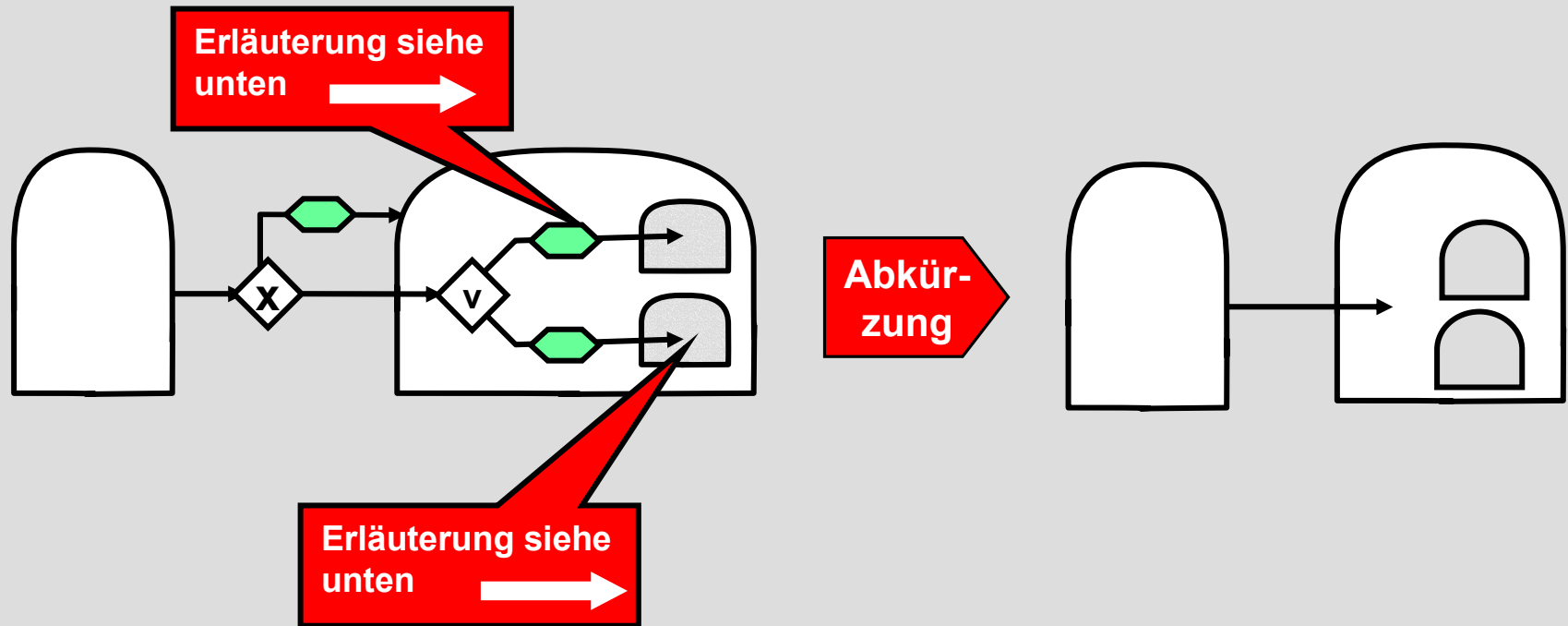
Aufgaben

## 7.2.1 Vage Verankerung des Startpunktes einer Relation als Abkürzung



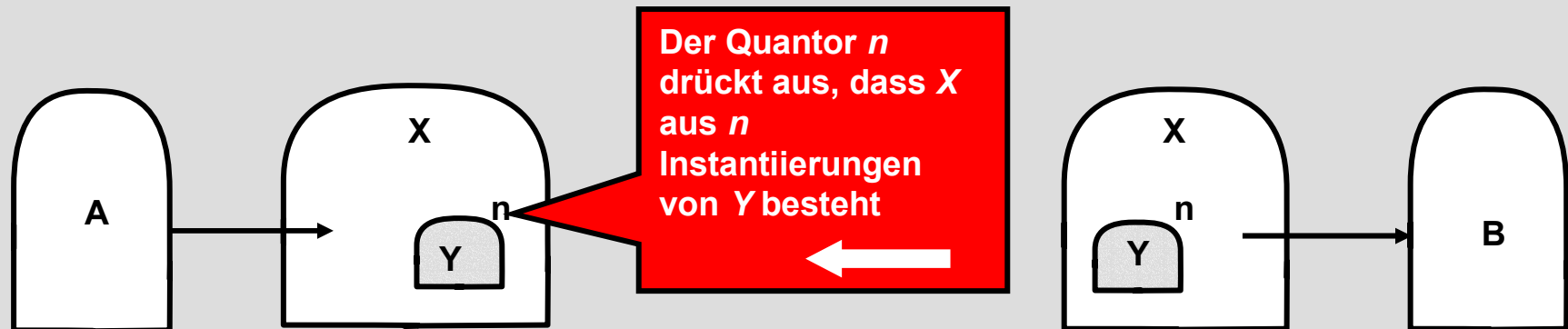
Die vage Verankerung des Startpunktes kann als Abkürzung eine komplexeren Darstellung verstanden werden. Sie basiert auf Konstrukten der kombinierten Unvollständigkeit und des nicht-spezifizierten Modifikators, die weiter unten erläutert werden. Das gleiche gilt für den Endpunkt.

## 7.2.1 Vage Verankerung des Endpunktes einer Relation als Abkürzung



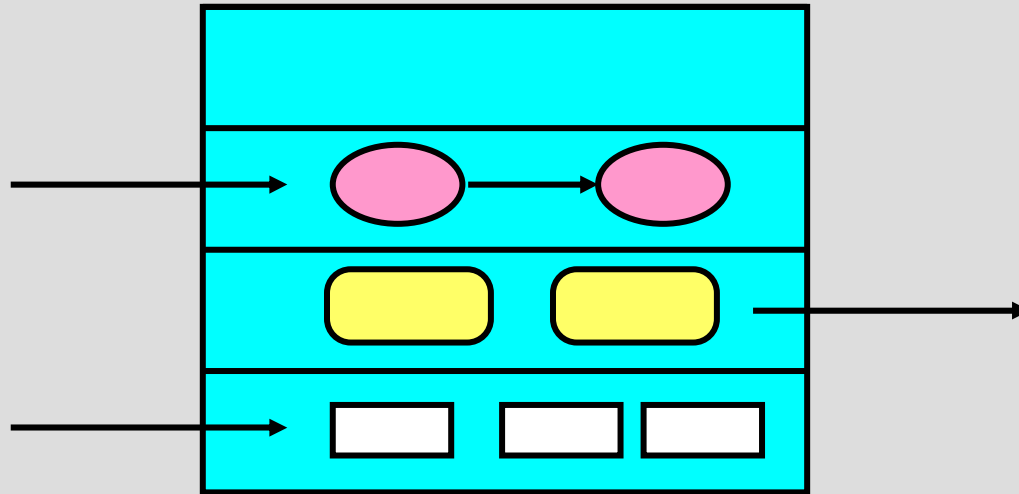
Die vage Verankerung kann auch beim Endpunkt als Abkürzung verstanden werden. Es sei noch ein mal erwähnt, dass ggf. Sub-Elemente der Sub-Elemente als Verankerungsziel dienen.

## 7.2.1 Vage Verankerung des Start- oder Endpunktes einer Relation bzgl. Instanzen



Die vage Verankerung kann sich nicht nur auf verschiedene Sub-Elemente eines Elementes beziehen, sondern auch auf verschiedene Instanziierung desselben Sub-Elementes, sofern ein Element gleichzeitig mehrere solcher Instanziierungen enthält. A ist also mit einer nicht näher spezifizierten Untermenge der  $n$  Instanziierungen des Sub-Elementes Y verbunden, die jede Instanziierung von X enthält. Das gleiche gilt in umgekehrter Richtung für B.

## 7.2.1 Segmente und vage Verankerung einer Relation

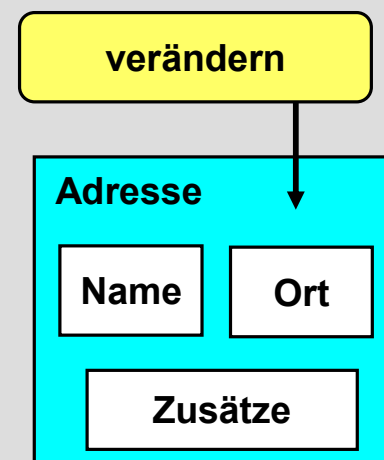
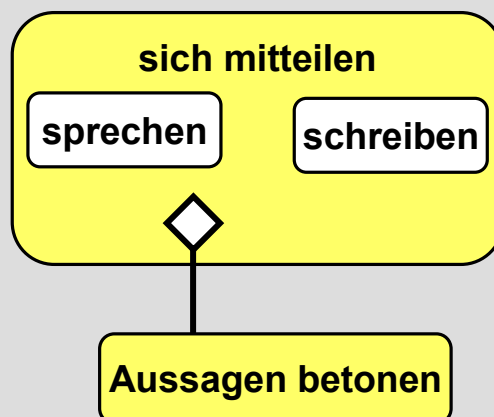
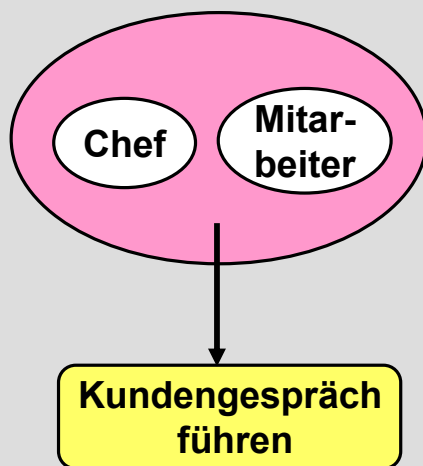


Wenn man Relationen mit unterschiedlichen Gruppen von Sub-Elementen verankern will, dann kann man Segmentlinien verwenden. Auf diese Weise kann man die vage Verankerung auch mit Sub-Elementen vornehmen, die einen anderen Typ als das Super-Element haben.





## 7.2.2 Vage Verankerung des Start- oder Endpunktes einer Relation – Beispiele I



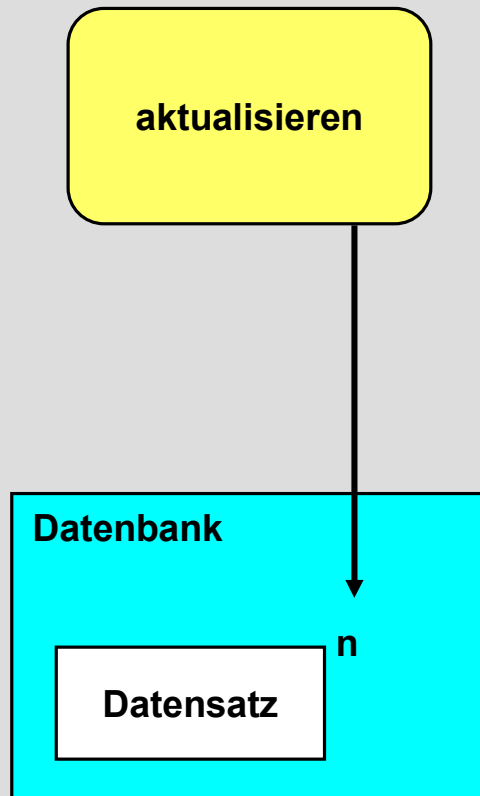
Es ist nicht festgelegt, ob der Chef oder die Mitarbeiter Kundengespräche führen.  
Wie man in [7.1](#) gesehen hat, ist es möglich, den Namen eines Basis-Elementes, hier der Rolle, wegzulassen.

Die Aktivität „Aussage betonen“ kann Bestandteil von „sprechen“ und/oder „schreiben“ oder von „sich mitteilen“ im ganzen sein. Es gibt keine genaue Zuordnung.

Es ist nicht festgelegt, worauf sich „verändern“ bezieht – einzelne Teile einer Adresse können verändert werden, ohne dass die anderen einbezogen sind.



## 7.2.2 Vage Verankerung des Start- oder Endpunktes einer Relation – Beispiele II



Hier wird möglicherweise auch ausgedrückt, dass sich die Aktivität des Aktualisierens nicht auf sämtliche  $n$  Instanzen des Datensatzes der Datenbank bezieht, sondern nur auf eine Untermenge.



## 7.2.3 Aufgaben zu vage verankerten Relationen

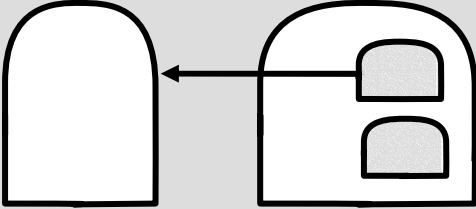
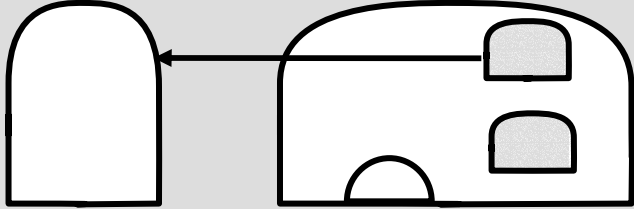
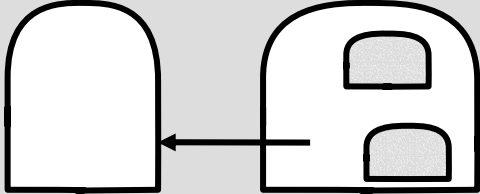
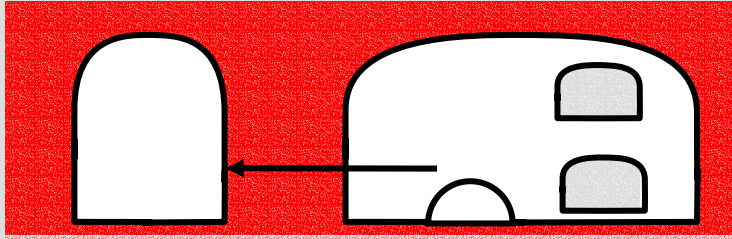
1. Stellen Sie mit Bezug auf das Diagramm bei dem Beispiel der Einbettung von Rollen in Rollen dar, dass Sachbearbeiter 2 mit irgendjemanden aus Referat 2 kooperiert, wobei nicht klar ist, ob die Kooperation nur aus Kommunizieren und/oder aus Koordinieren besteht.
2. Drücken Sie bzgl. desselben Beispiels aus, dass die Leitung eine nicht festgelegte Untermenge aus der Gruppe der aller Sachbearbeiter zu Berichtserteilung auffordern kann.
3. Ergänzen Sie das Diagramm aus Beispiele zur Einbettung von Rollen in Entitäten so, dass nicht festgelegt ist, wie sich ein Autor und der Herausgeber am Korrekturlesen eines Beitrages im Sammelband beteiligen.



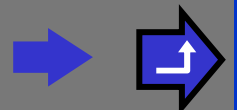
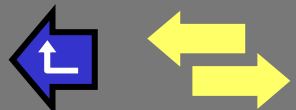
Aufg.



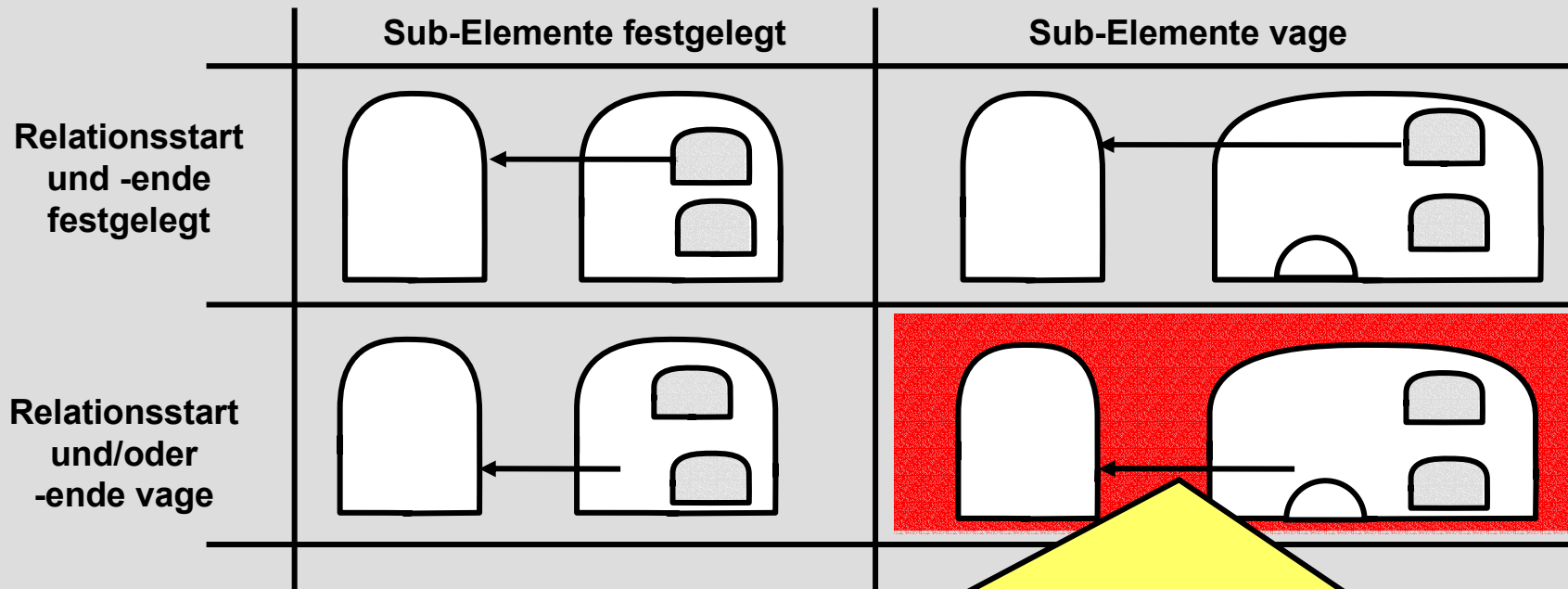
## 7.3 Kombinierte Vagheit

	Sub-Elemente festgelegt	Sub-Elemente vage
Relationsstart und -ende festgelegt		
Relationsstart und/oder -ende vage		

<u>7.3.1</u>	Definitionen
<u>7.3.2</u>	Beispiele
<u>7.3.3</u>	Besonderheiten
<u>7.3.4</u>	Aufgaben



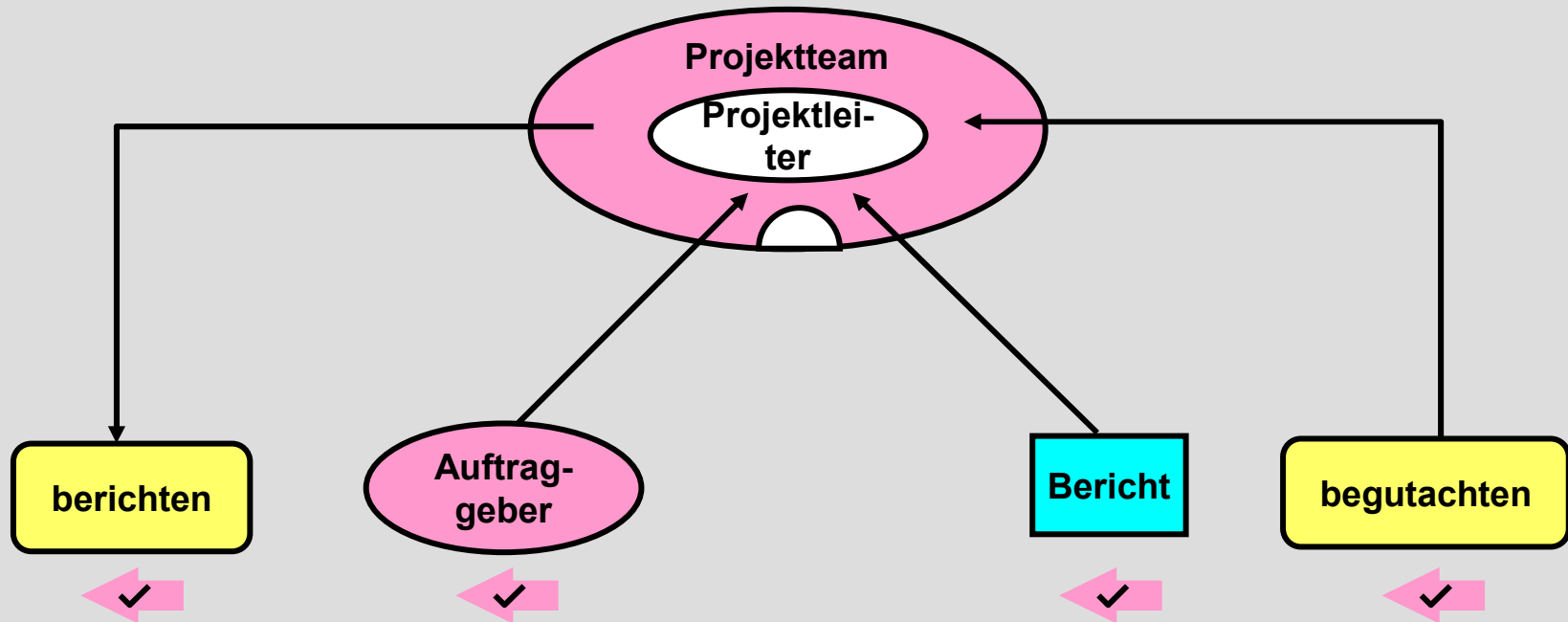
## 7.3.1 Kombinierte Vagheit



Beide Arten der bisher besprochenen Vagheit können kombiniert werden: Für eine Relation, die in ein leeres Element (ohne Sub-Elemente) hinein- oder herauszeigt gilt, dass sie mit potentiellen, nicht explizit festgelegten Sub-Elementen verbunden ist. Falls, wie hier angedeutet, Sub-Elemente festgelegt sind, aber auf die Unvollständigkeit dieser Festlegung hingewiesen wird, dann gilt, dass die Relation entweder mit diesen Sub-Elementen oder mit potentiell anderen verbunden ist. Auch hier ist auf Typ-Gleichheit zu achten.



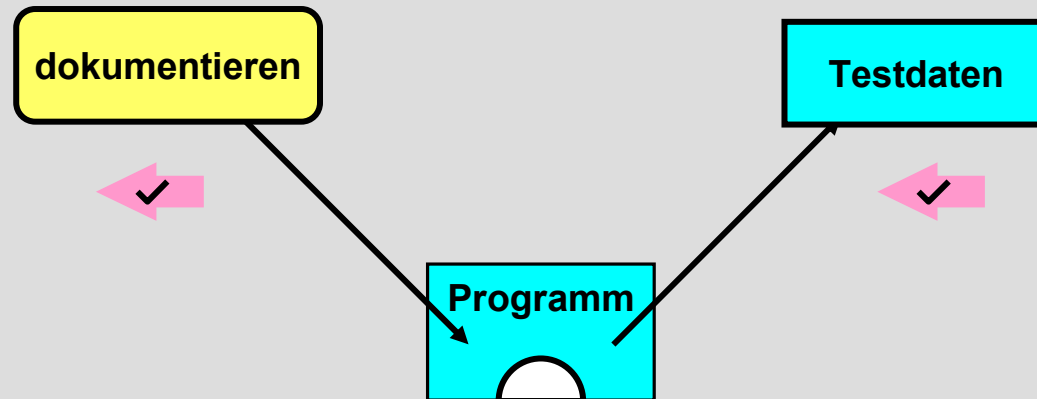
## 7.3.2 Kombinierte Vagheit – Beispiele I



Es ist nicht festgelegt, welche Mitglieder des Projektteams die Aufgabe haben, zu berichten und wer von ihnen direkt den Erwartungen des Auftraggebers ausgesetzt ist.

Es ist nicht festgelegt, welche Mitglieder des Projektteams Zugang zu den Berichten haben und wer von ihnen von einer Begutachtung beeinflusst wird.

## 7.3.2 Kombinierte Vagheit – Beispiele II

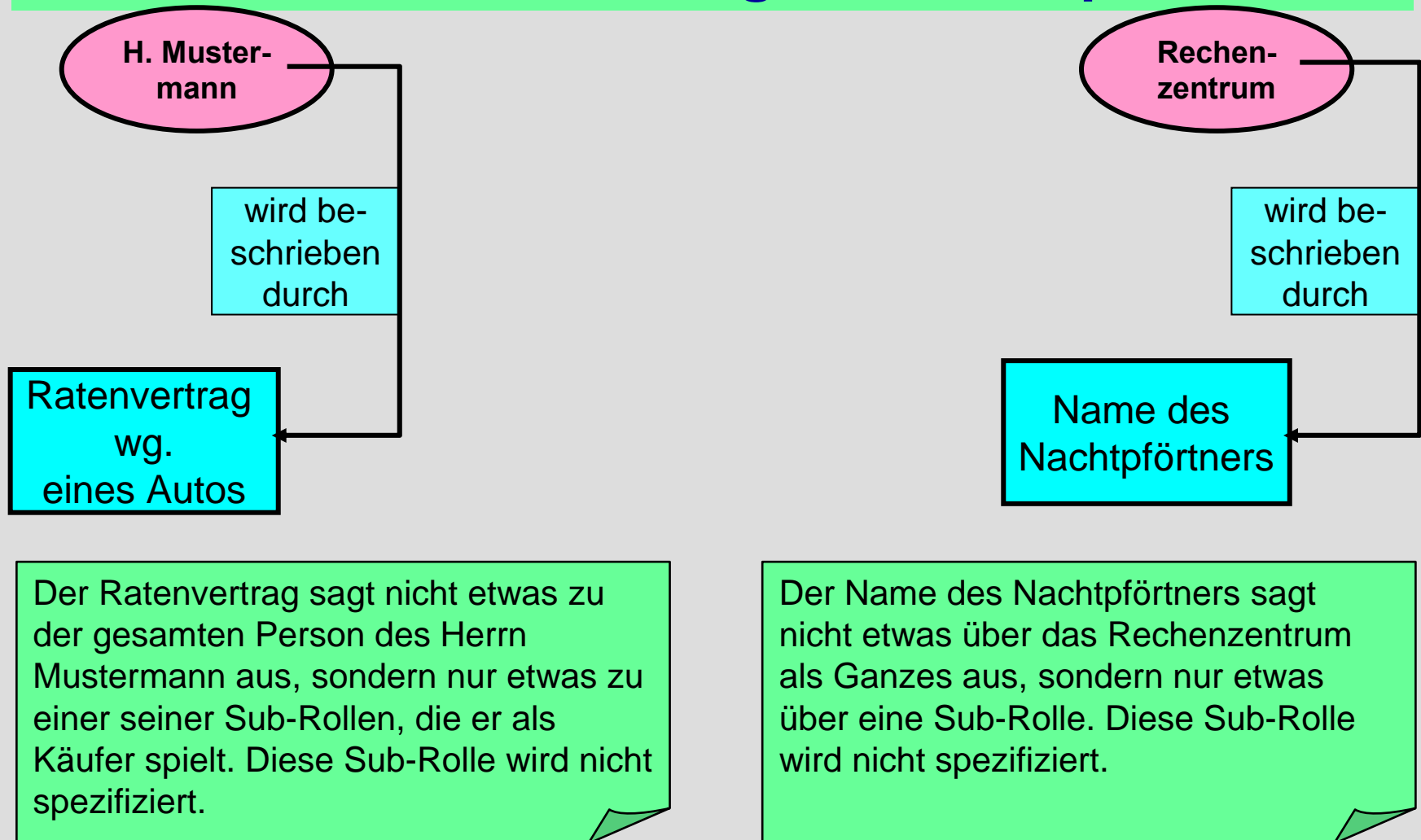


Es ist nicht festgelegt, welche Teile des Programmes dokumentiert werden. Es wird zum Ausdruck gebracht, dass u.U. nur Teile des Programms dokumentiert werden.

Nur Teile des Programmes haben ggf. einen Bezug zu den Testdaten. Es ist nicht spezifiziert, welche das sind.

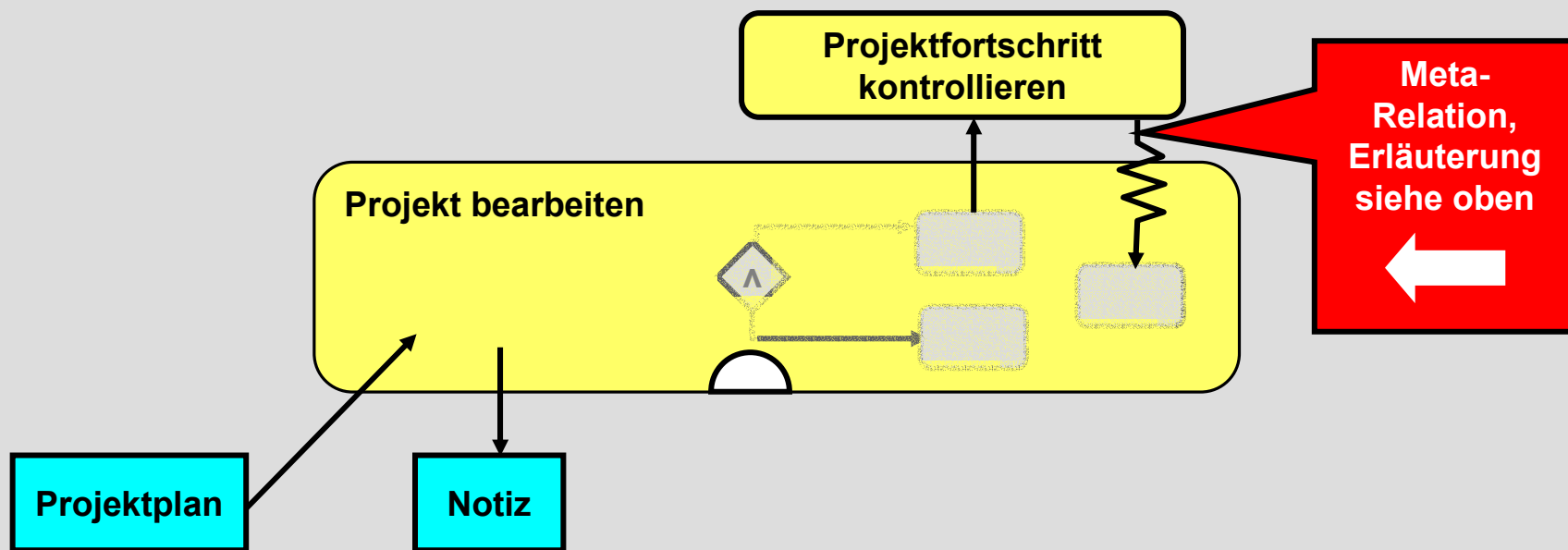


## 7.3.2 Kombinierte Vagheit – Beispiele III





## 7.3.2 Kombinierte Vagheit – Beispiele IV

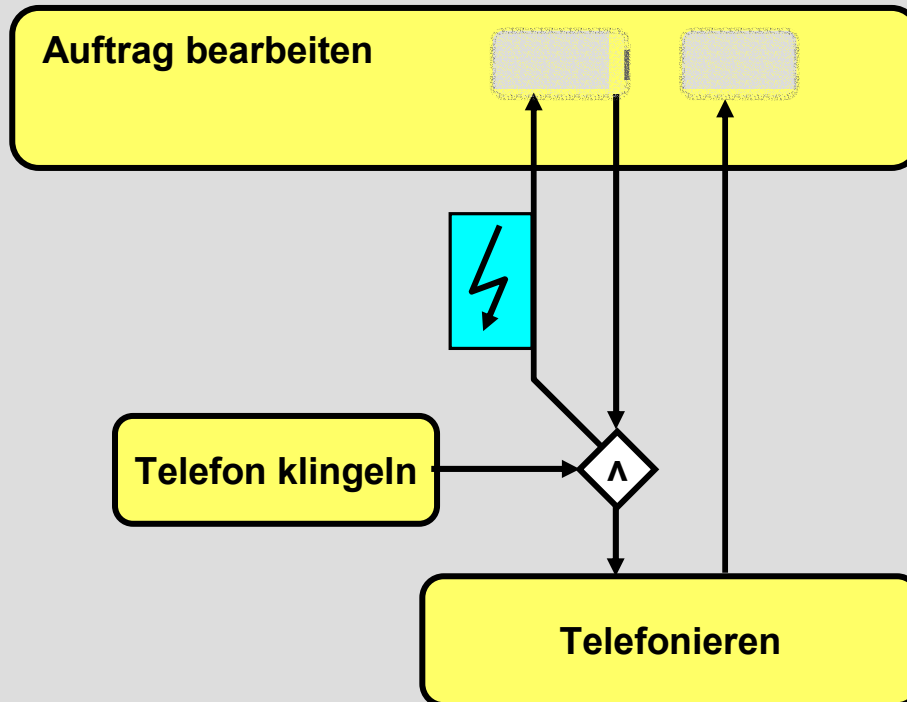


Der Projektplan wird u.U. nur für einige Sub-Aktivitäten der Projektbearbeitung genutzt, die nicht näher spezifizierbar sind. Letztlich entscheiden die Bearbeiter, an welchen Stellen der Projektplan herangezogen wird. Dies gilt auch für das Anfertigen von Notizen.

Die Kontrolle des Projektfortschrittes knüpft an Sub-Aktivitäten an und wirkt auf andere, die nicht vorab festgelegt werden können. Insbesondere kann die Kontrolle parallel zu Sub-Aktivitäten der Bearbeitung verlaufen, wie man an einer möglichen, grauen Struktur sieht.



## 7.3.2 Beispiel Unterbrechung



Eine Unterbrechung besteht aus einer Aktivität, die unterbrochen wird (Auftrag bearbeiten), einer Aktivität, die die Unterbrechung auslöst, und einer Aktivität, die statt der eigentlichen Aktivität durchgeführt wird.

Es ist nicht zu unterstellen, dass nach der Unterbrechung zur gleichen Sub-Aktivität zurückgesprungen wird, von der aus die unterbrechende Aktivität aufgenommen wurde (siehe eine mögliche, grau angedeutete Struktur).

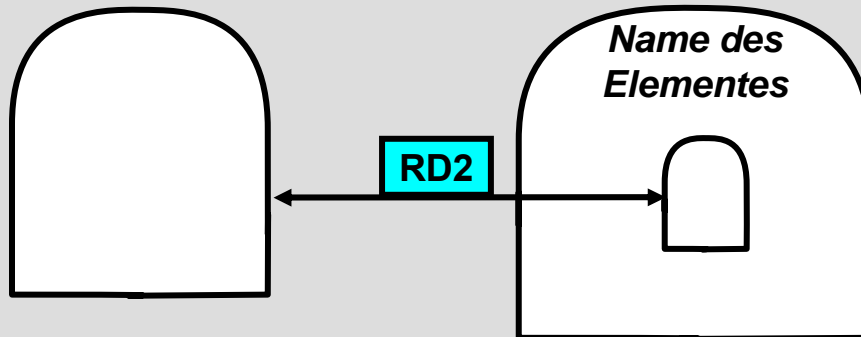
## 7.3.3 Kombinierte Vagheit - Abkürzung



Zwei doppelte Pfeile können auch als Doppelpfeil dargestellt werden, wenn Start- und/oder Endpunkte unspezifiziert sind. Dabei ist aber darauf zu achten, dass RD1 ist nicht auf das gleiche Sub-Element zeigen muss, von dem es auch wegführt!



## 7.3.3 Gleiches End- und Startelement

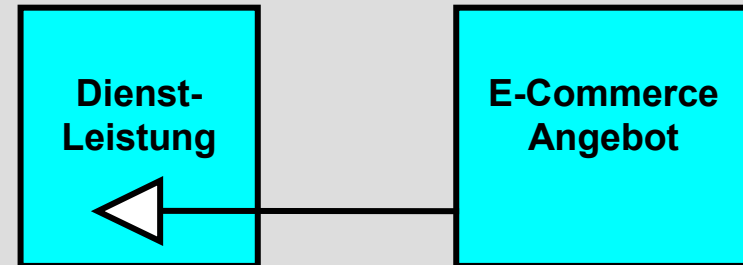


**Selten  
benötigt**

Um auszudrücken, dass der Start- und Endpunkt des Doppelpfeiles in einem Basis-Element an der gleichen Stelle anknüpfen, muss ein nicht näher spezifizierte Sub-Element (vom gleichen Typ wie das Super-Element, ohne Name) eingeführt werden.



## 7.3.3 Kombinierte Vagheit - Vererbung

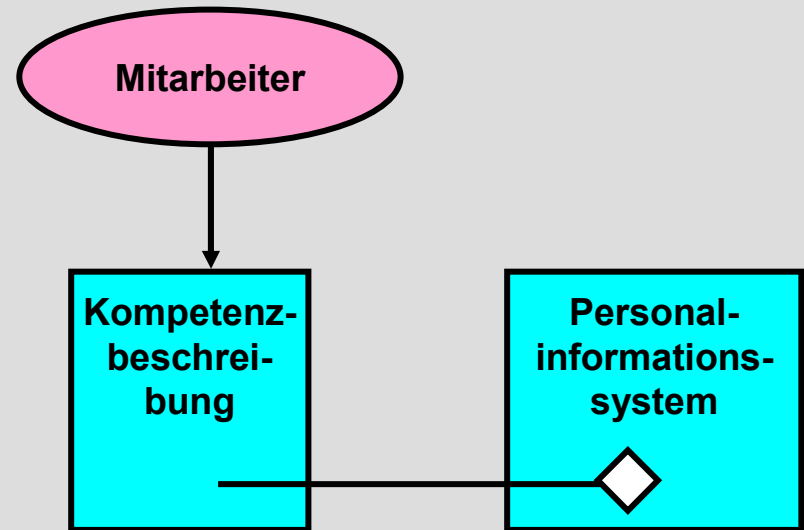
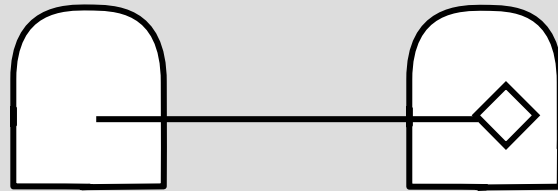


**Auch die Vererbungsrelation kann einen nicht näher festgelegten Start- und/oder Endpunkt haben. Folglich sind potentiell nur Teile der verbundenen Elemente von der Vererbung betroffen.**

Ein E-Commerce-Angebot erbt nur von Teilen der Dienstleistung, auf der sie aufbaut, deren Eigenschaften.



## 7.3.3 Kombinierte Vagheit - Aggregation



Die Aggregation kann einen nicht näher festgelegten Start- und/oder Endpunkt haben. Folglich sind nur Teile des einen Elementes in Sub-Elementen des anderen enthalten.

Nur Teile der Beschreibung der Kompetenz eines Mitarbeiters können ins Personalinformationssystem eingehen. Dort wiederum gibt es Sub-Elemente, die keine Kompetenzbeschreibung beinhalten.



## 7.3.4 Aufgaben zu kombinierter Vagheit

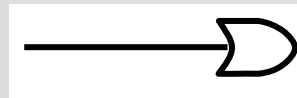
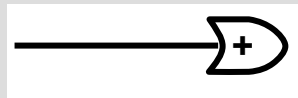
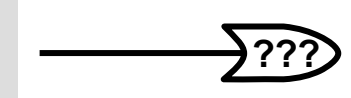
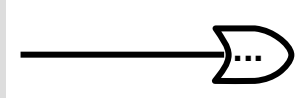
Stellen Sie folgende Sachverhalte dar:

- Ein Prototyp vererbt nur einen Teil seiner Eigenschaften an die Endversion des Programmes.
- A) Die Aktivität „Qualität sichern“ als Teil der Sub-Aktivitäten von „Moderieren“ B) und als Aktivität, die sich mit den Sub-Aktivitäten von Moderieren abwechseln kann.
- Mehrere Programmierer eines Teams sind für das Dokumentieren unterschiedlicher Teile eines Programmes zuständig.
- Das „Lenken“ verändert eigentlich nicht ein Auto als ganzes, sondern nur bestimmte Aspekte desselben.
- Es ist ungewiss, wer außer der Personalabteilung Informationen über Mitarbeiter eines Betriebes abrufen kann.



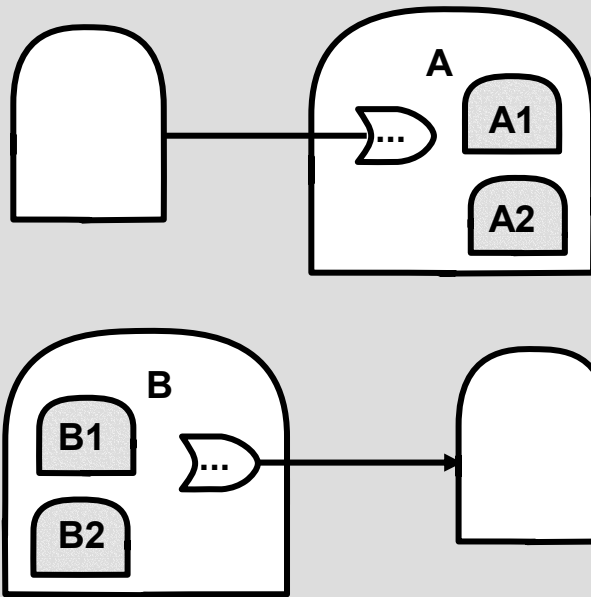
## 7.4 Vage Spezifizierung eines Ankerpunktes einer Relation

Auch die Start und Endpunkte einer Relation können mit den eingeführten Vagheitssymbolen versehen werden.

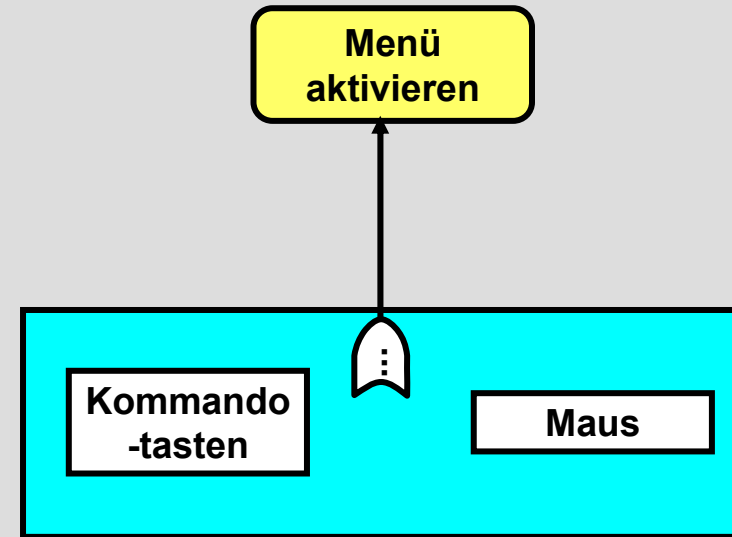




## 7.4 Vage Spezifizierung der Start- oder Endpunkte mit (...)



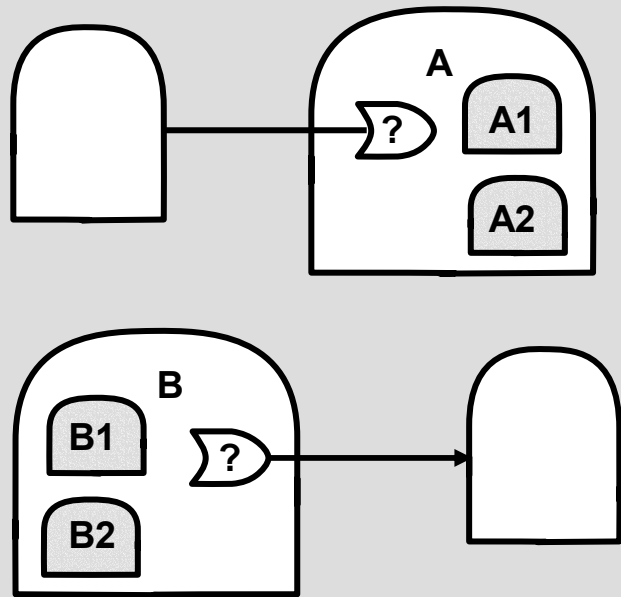
**Der Modellierer weiß nicht, wie er den Start- und/oder Endpunkte der Relation mit A, A1 A2 bzw. B, B1, B2 verbinden soll.**



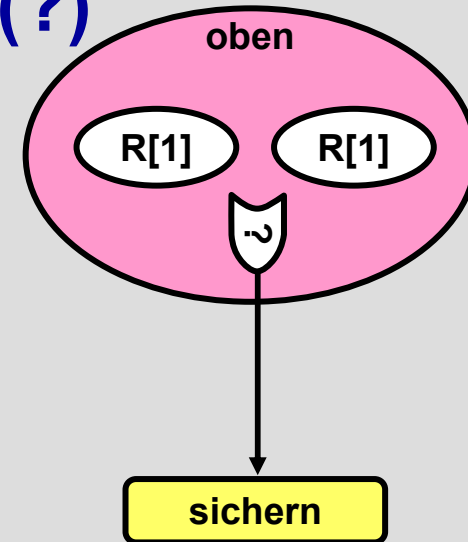
Es ist software-ergonomisch mit dem derzeitigen Wissensstand nicht festlegbar, wann der Einsatz der Maus oder der Kommando-tasten günstiger ist.



## 7.4 Vage Spezifizierung der Start- oder Endpunkte mit (?)

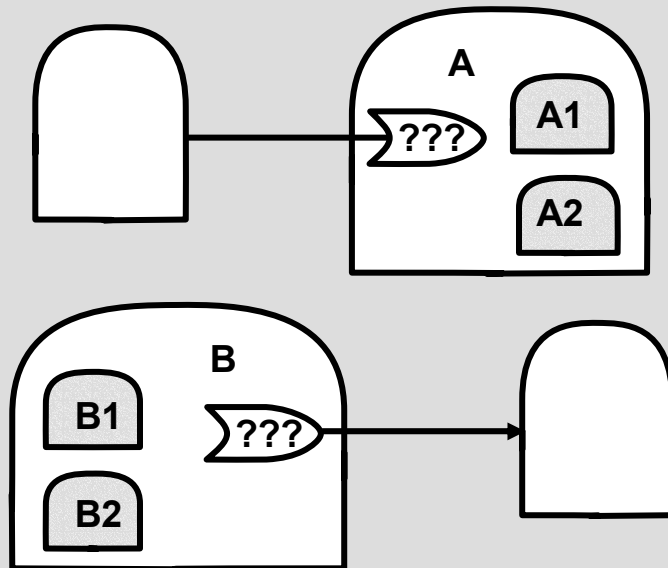


Es ist fraglich, ob es richtig ist, den Start- und/oder Endpunkt nicht genau einem Element (A, A1, A2 bzw. B, B1, B2) zuzuordnen.

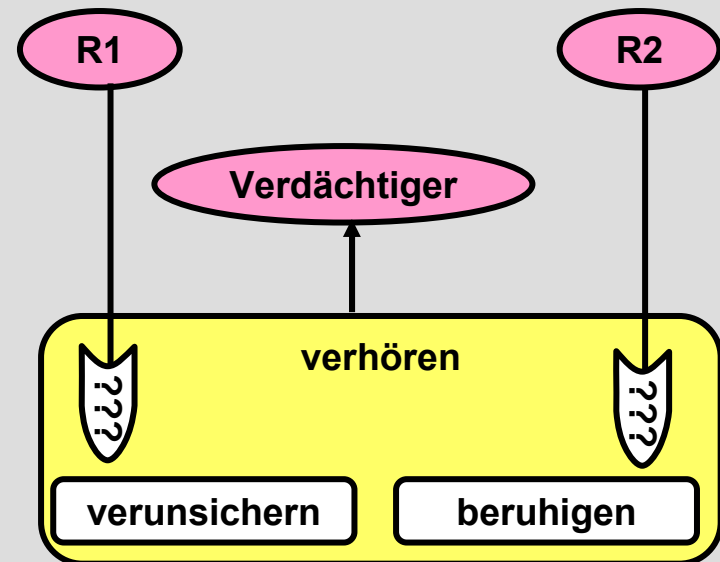


Wenn, wie in diesem Diagramm nicht genau festgelegt ist, wer für das Sichern zuständig ist, kann es zu Missverständnissen kommen. Deshalb wird die Korrektheit bezweifelt.

## 7.4 Vage Spezifizierung der Start- oder Endpunkte mit (???)

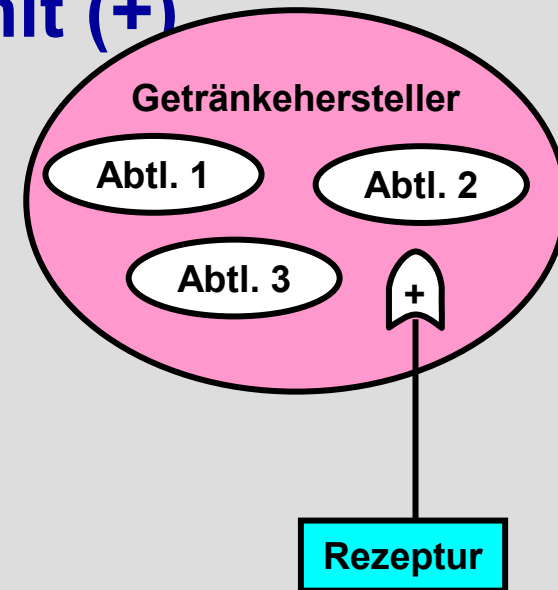
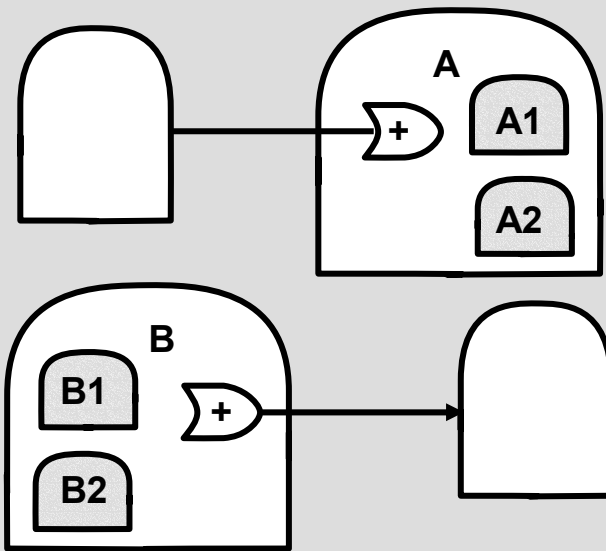


Es ist unklar, ob man Start- und/oder Endpunkt genauer festlegen kann oder sollte.



Wir wissen nicht, ob die Rollenverteilung, wer *verunsichert* oder *beruhigt*, festgelegt oder flexibel angepasst werden sollte.

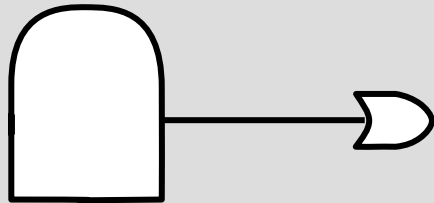
## 7.4 Vage Spezifizierung der Start- oder Endpunkte mit (+)



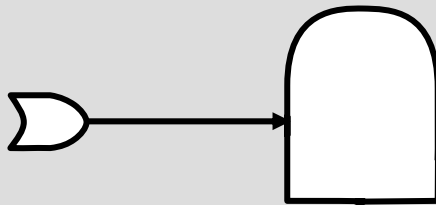
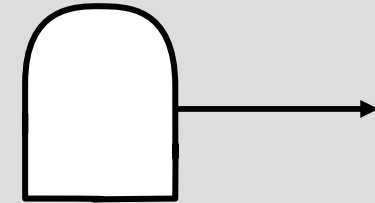
Der Modellierer weiß, wie die Relation zuzuordnen ist, will es aber im Diagramm nicht darstellen.

Der Modellierer will im Diagramm nicht preisgeben, welche Abteilung die Rezeptur kennt, obwohl dies bekannt ist. Dies ermöglicht die Behandlung vertraulicher Mitteilungen.

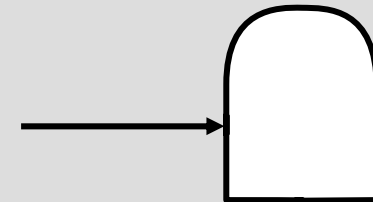
## 7.4 Vage Spezifizierung der Start- oder Endpunkte – Abkürzung I



Abkürzung



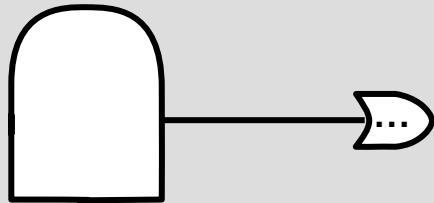
Abkürzung



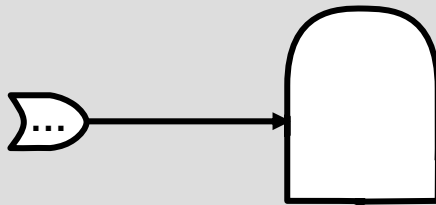
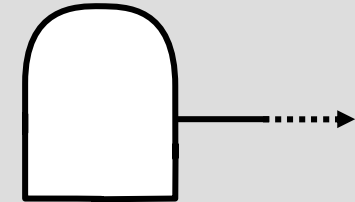
Bei der gewollten Unvollständigkeit, die dem leeren Halbkreis entspricht, kann die am Pfeil verankerte leere Fläche weggelassen werden.



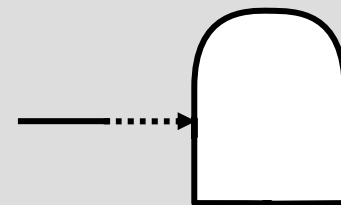
## 7.4 Vage Spezifizierung der Start- oder Endpunkte – Abkürzung II



Abkürzung



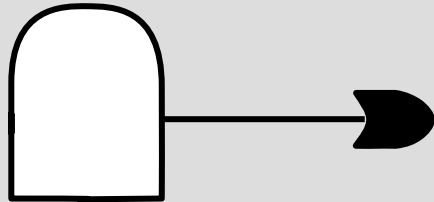
Abkürzung



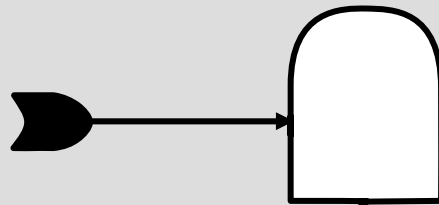
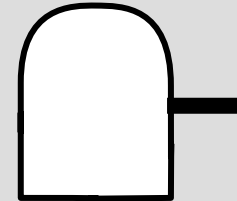
Bei der gegebenen Unvollständigkeit aufgrund fehlender Informationen, die dem Halbkreis mit drei Punkten entspricht, kann die am Pfeil verankerte Fläche weggelassen werden und der Hinweis, dass man nicht weiß, wie die Relation fortzusetzen ist, durch eine gepunktete Linie angedeutet werden.



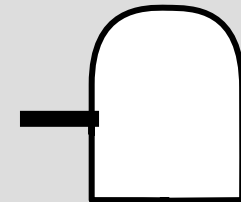
## 7.4 Vage Spezifizierung der Start- oder Endpunkte – Abkürzung III



Abkürzung



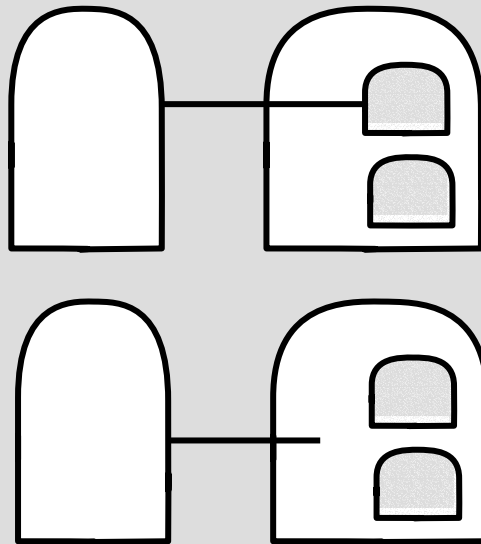
Abkürzung



Bei dem gewollten Verstecken der Verankerung eines Pfeils, das dem schwarzen Halbkreis entspricht, kann die am Pfeil verankerte schwarze Fläche durch eine Verdickung ersetzt werden.



## 7.4 Vage Spezifizierung der Richtung



**Die Richtungsangabe einer Relation kann unter Umständen un spezifiziert bleiben.**

Bei der Zuordnung eines Diagramms zu einer natürlichsprachlichen Beschreibung macht es keinen Sinn, die Richtung der Zuordnung festzulegen.





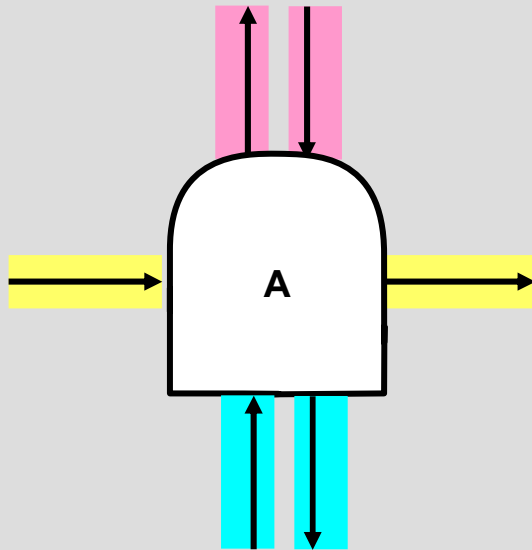
## 7.4 Aufgaben zu vagen Start- oder Endpunkten

Stellen Sie folgende Sachverhalte dar:

1. Man weiß nicht genau, zu welchen Sub-Aktivitäten der Software-Entwicklung (z.B. Spezifizieren, Testen) , die künftigen Benutzer hinzuzuziehen sind.
2. Es wird bezweifelt, ob man genau festlegen sollte, ob Benutzer eines Programmes im Rahmen der Wartung mehr Veränderungen vornehmen dürfen als die Veränderung von Parametern.
3. Es wird bezweifelt, dass man es der Entscheidung des Benutzers überlässt, ob er nach einer Unterbrechung einer Aufgabe den zuletzt durchgeführten Schritt noch einmal wiederholt oder ob er direkt weiter macht.



## 7.5 Vage Spezifizierung des Auftretens von Relationen



- [7.5.1](#) Einführung
- [7.5.2](#) 18 Formen
- [7.5.3](#) Besonderes
- [7.5.4](#) Beispiele
- [7.5.5](#) Aufgaben

Wenn eine Relation an einer Seite mit einem Element A verankert ist und an der anderen Seite weder mit dem Rand eines Elementes verankert ist noch diesen schneidet,

dann hängt ihre Bedeutung davon ab, wie sie geometrisch relativ zu dem Element A angeordnet ist:

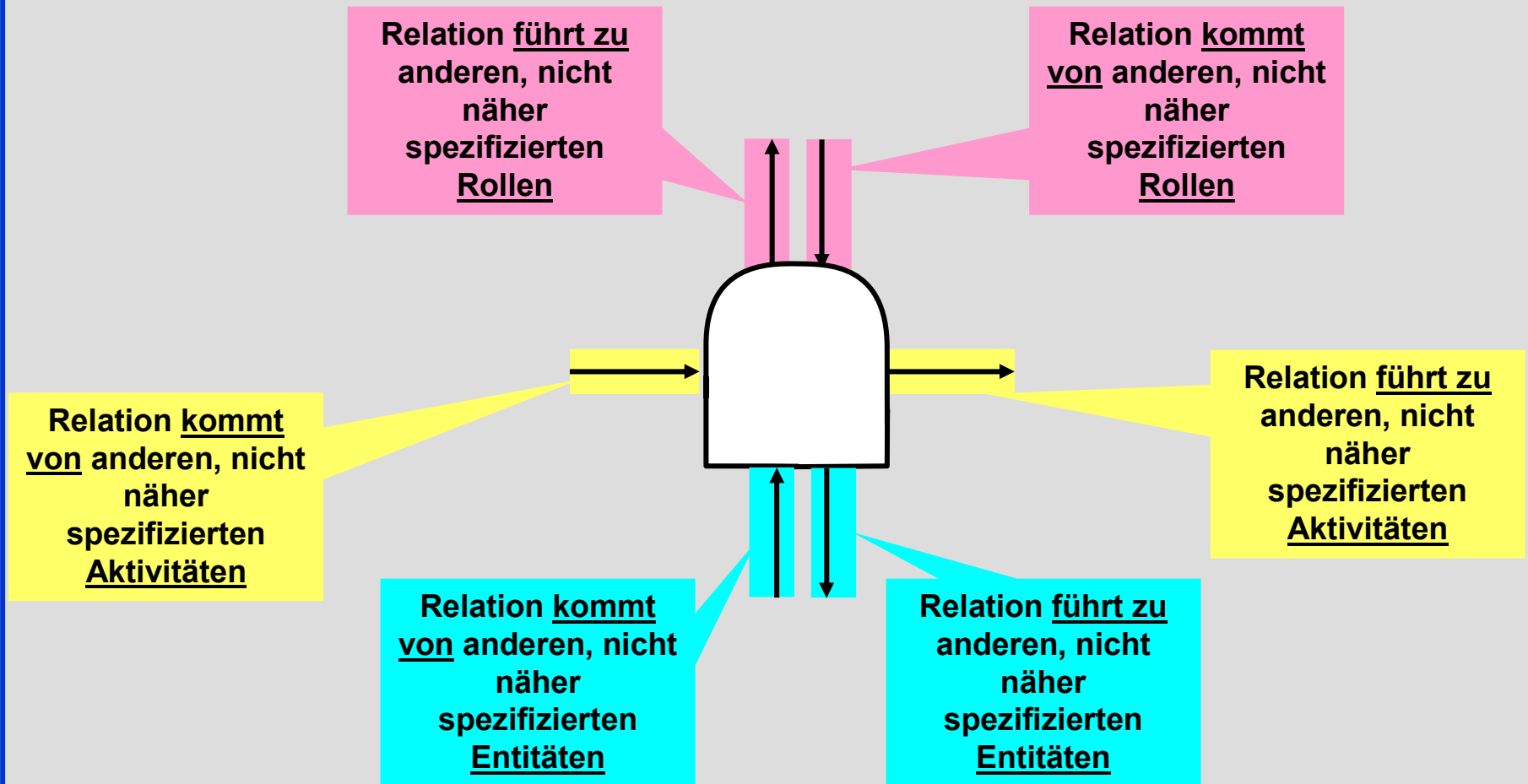
Eine vertikale Orientierung und Anbindung oberhalb von A verweist auf Rollen

Eine horizontale Orientierung verweist auf Aktivitäten.

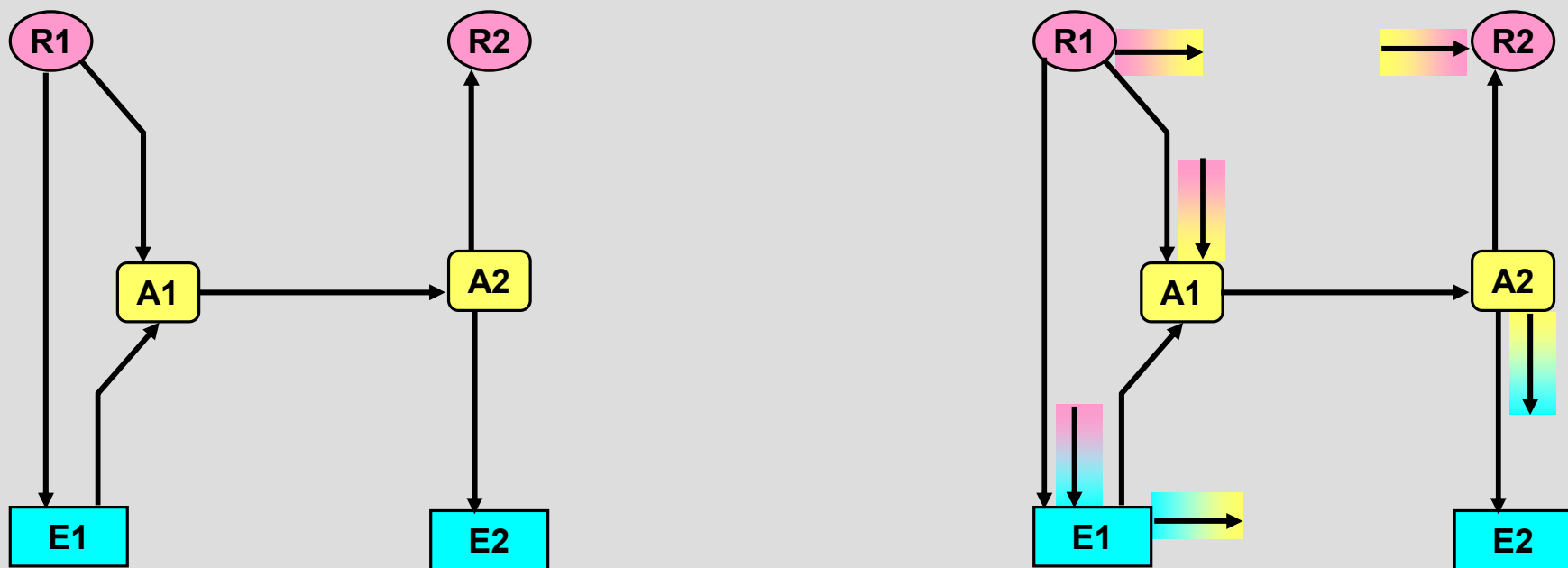
Eine vertikale Orientierung und Anbindung unterhalb von A verweist auf Entitäten



## 7.5.1 Vage Spezifizierung des Auftretens von Relationen



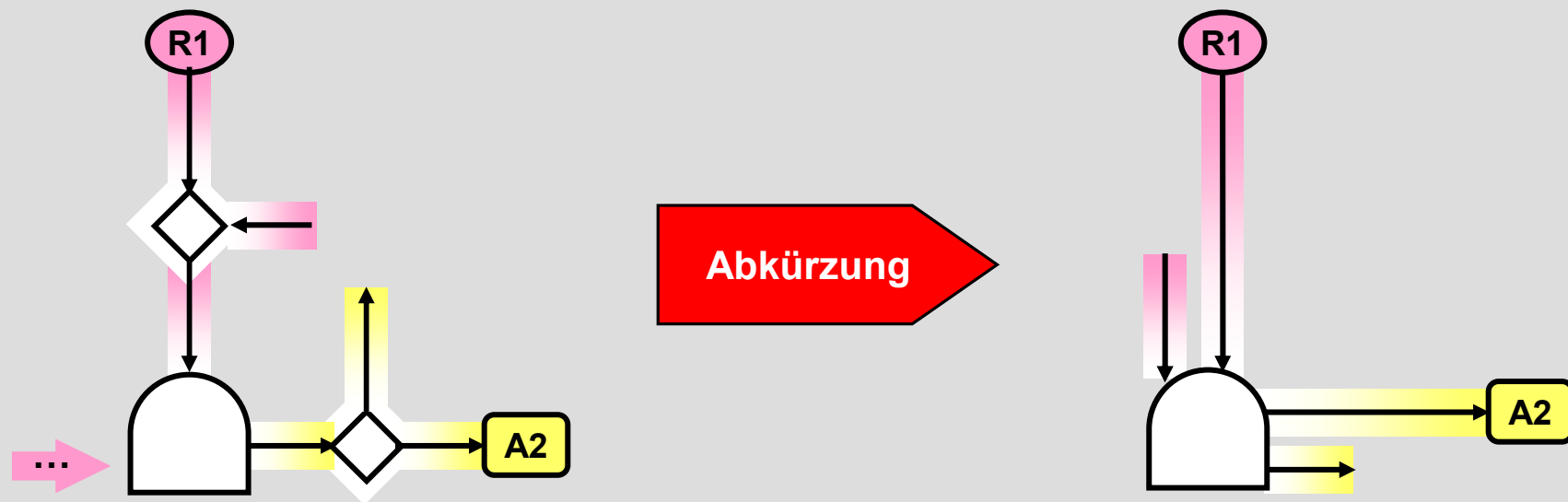
## 7.5.1 Wann braucht man vage Spezifizierung des Auftretens von Relationen ?



Wenn man eine Relation eines bestimmten Typs nicht an ein Element anbindet, dann ist dies als absichtliche Auslassung anzusehen. Wird aber eine Relation dieses Typs (z.B. A2 beeinflusst R2) angebunden, dann stellt sich die Frage, ob es die einzige Relation dieses Art ist, die mit R2 bzw. A2 verbunden ist. Die freien Relationen drücken dann z.B. aus, dass R2 noch von anderen Aktivitäten beeinflusst werden kann (entweder A1 oder nicht gezeigte Aktivitäten), während A2 keine weiteren Rollen mehr beeinflusst.

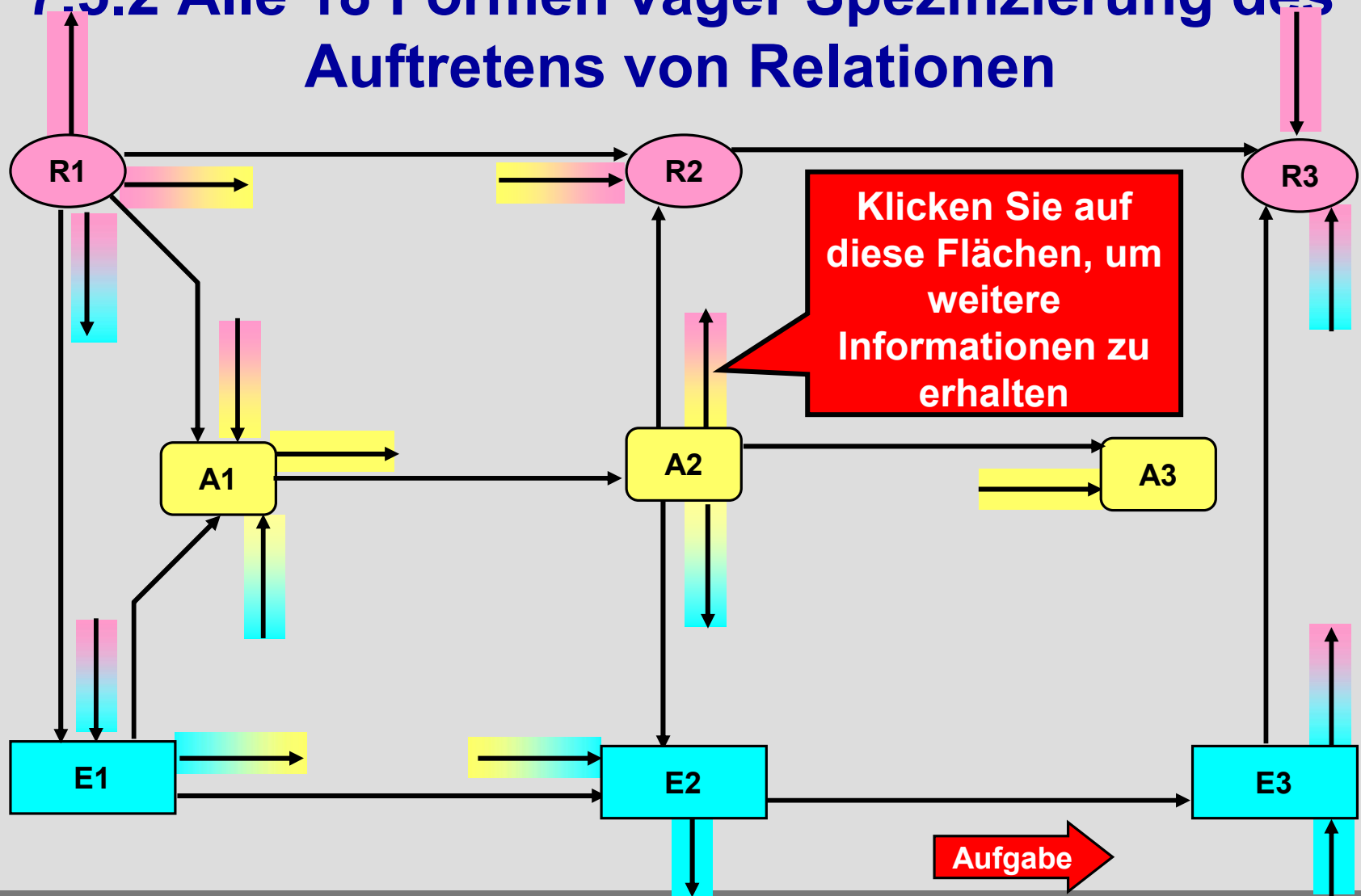
**Aufgabe**

## 7.5.1 Abkürzung

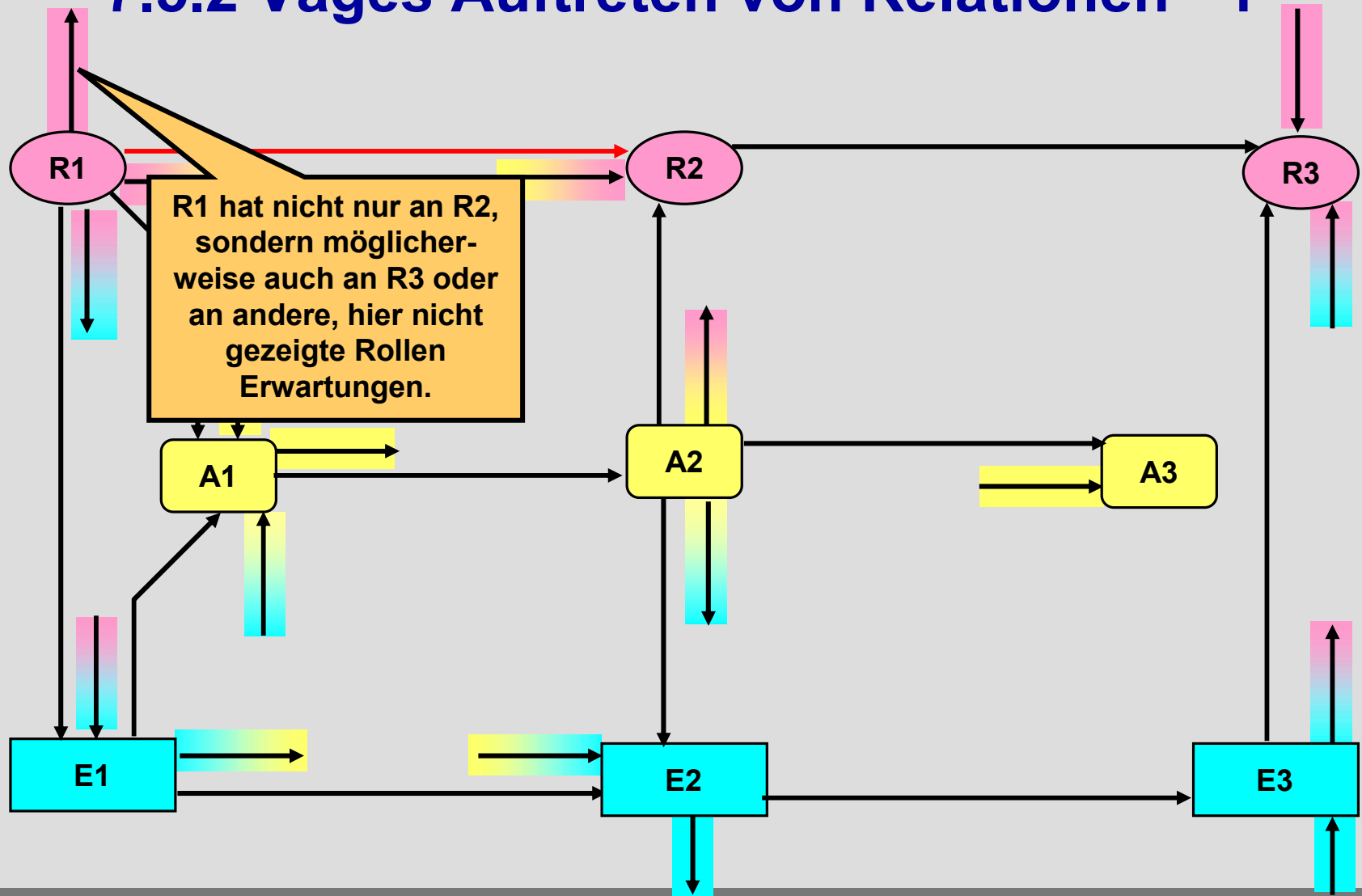


Die rechte Seite stellt eine Abkürzung des links gezeigten Diagramms dar. Die linke Seite verwendet unspezifizierte Konnektoren, die unten erläutert werden. An diese Konnektoren werden die freien Relationen angebunden, wie hier am Beispiel der Verknüpfung mit Rollen und Aktivitäten gezeigt wird. Bei der Verwendung der leeren Konnektoren braucht man die geometrische Anordnung der freien Relationen nicht zu beachten; die freie Relation ist hier immer per Default vom selben Typ, wie die durch den Konnektor sonst miteinander verbundenen Relationen, so lange nichts anderes explizit angegeben wird. Das rechte Diagramm verzichtet dagegen auf die Konnektoren, man muss aber die geometrische Anordnung berücksichtigen.

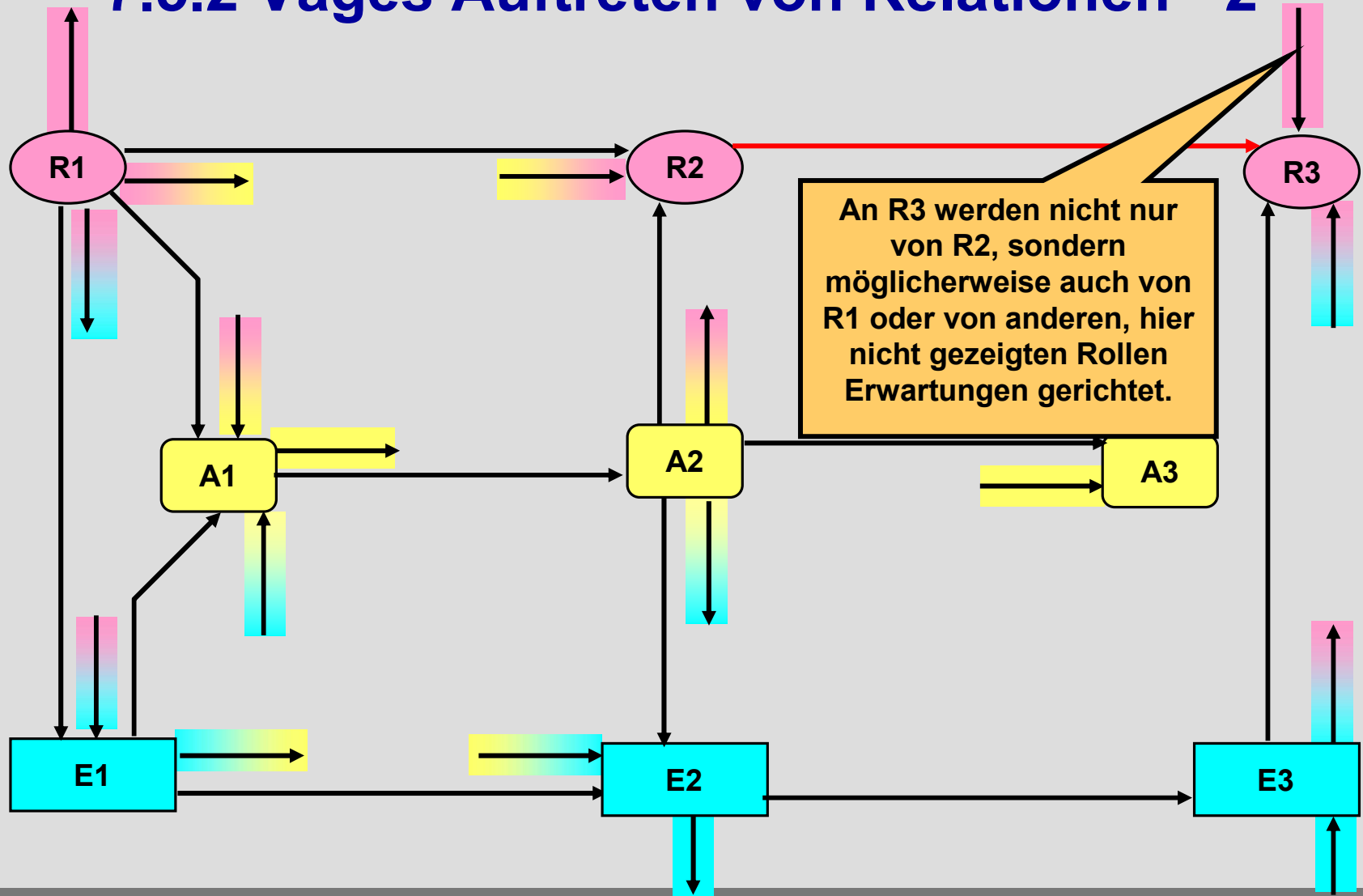
## 7.5.2 Alle 18 Formen vager Spezifizierung des Auftretens von Relationen



## 7.5.2 Vages Auftreten von Relationen - 1

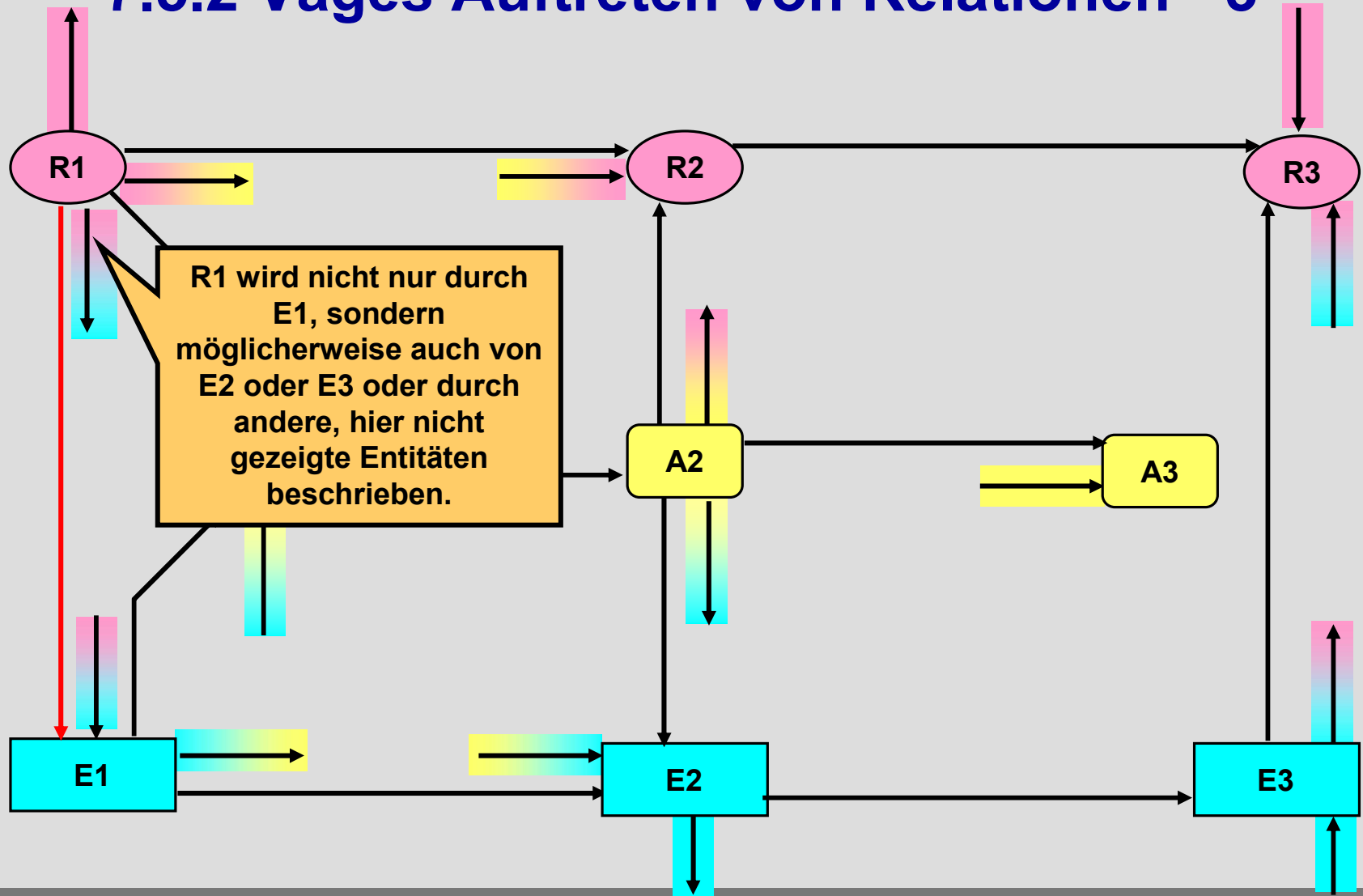


## 7.5.2 Vages Auftreten von Relationen - 2

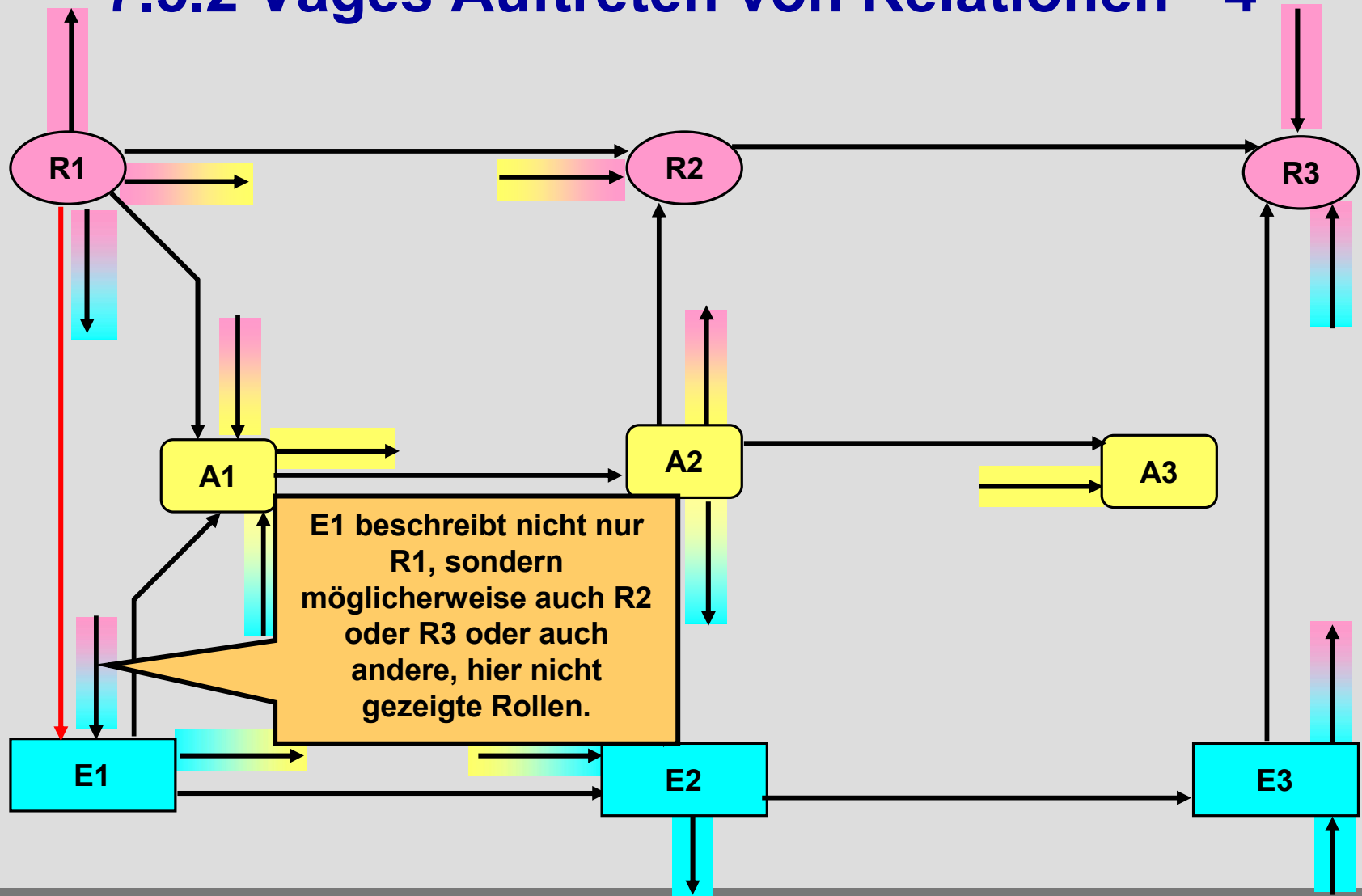




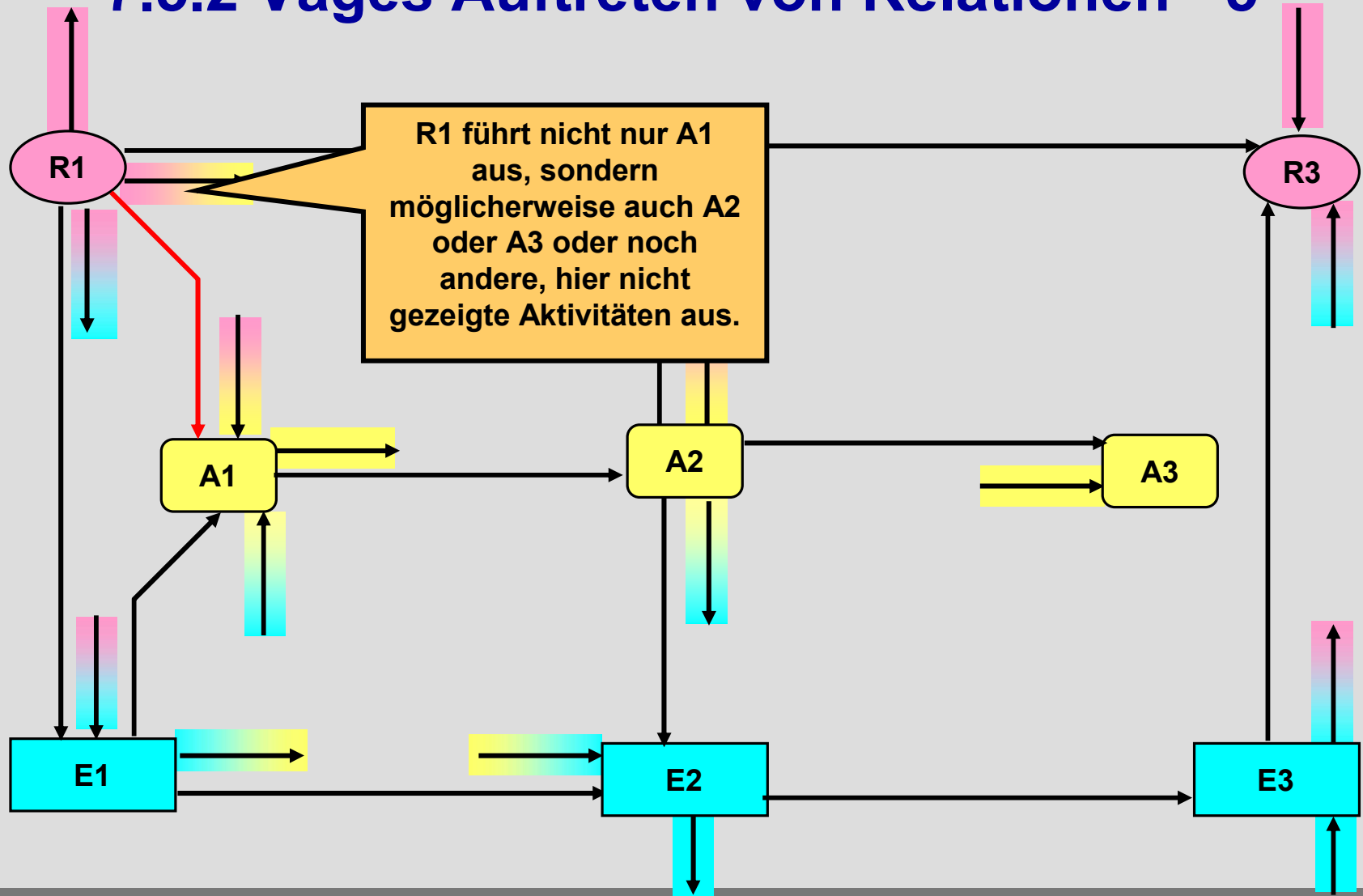
## 7.5.2 Vages Auftreten von Relationen - 3



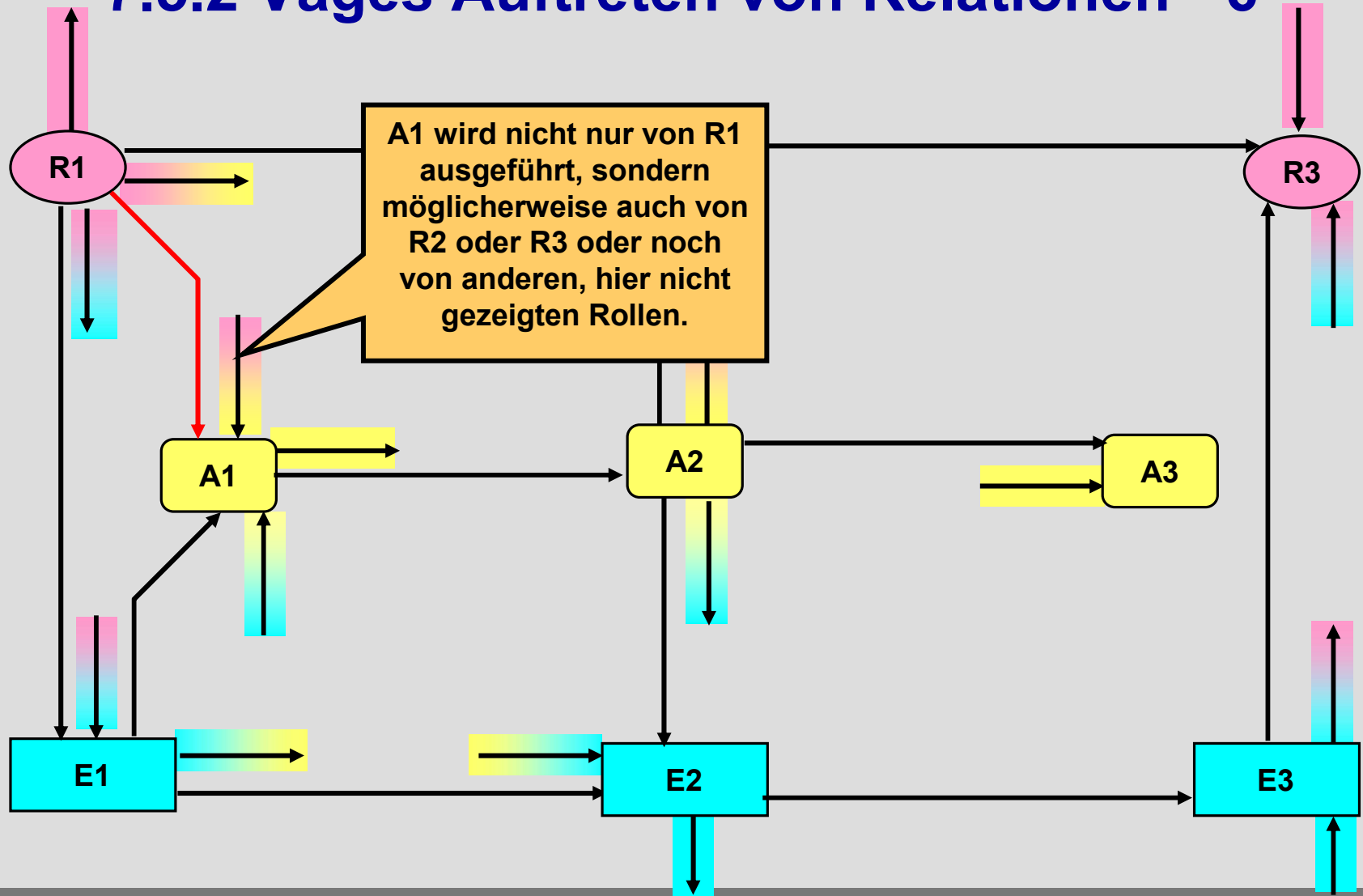
## 7.5.2 Vages Auftreten von Relationen - 4



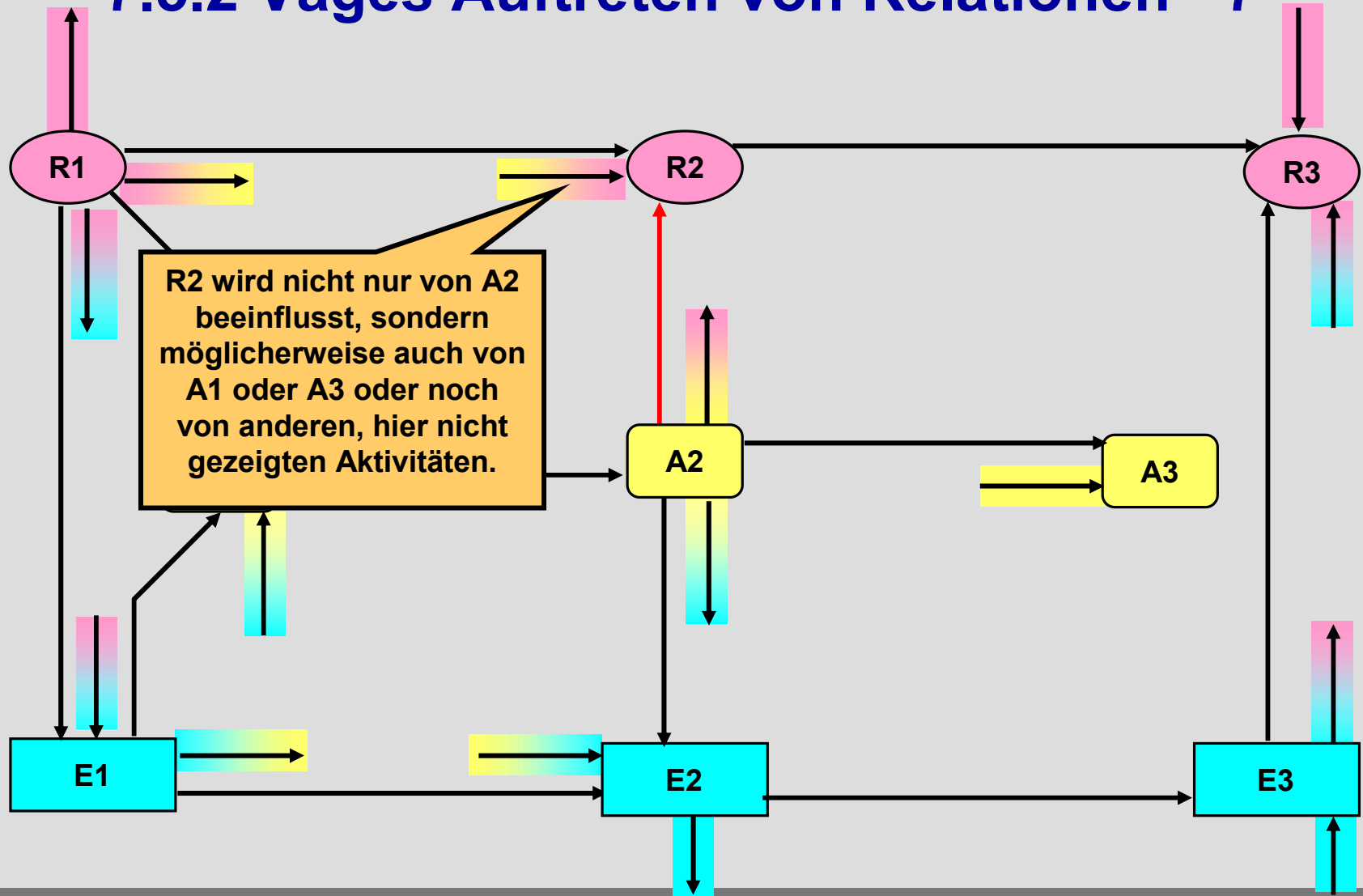
## 7.5.2 Vages Auftreten von Relationen - 5



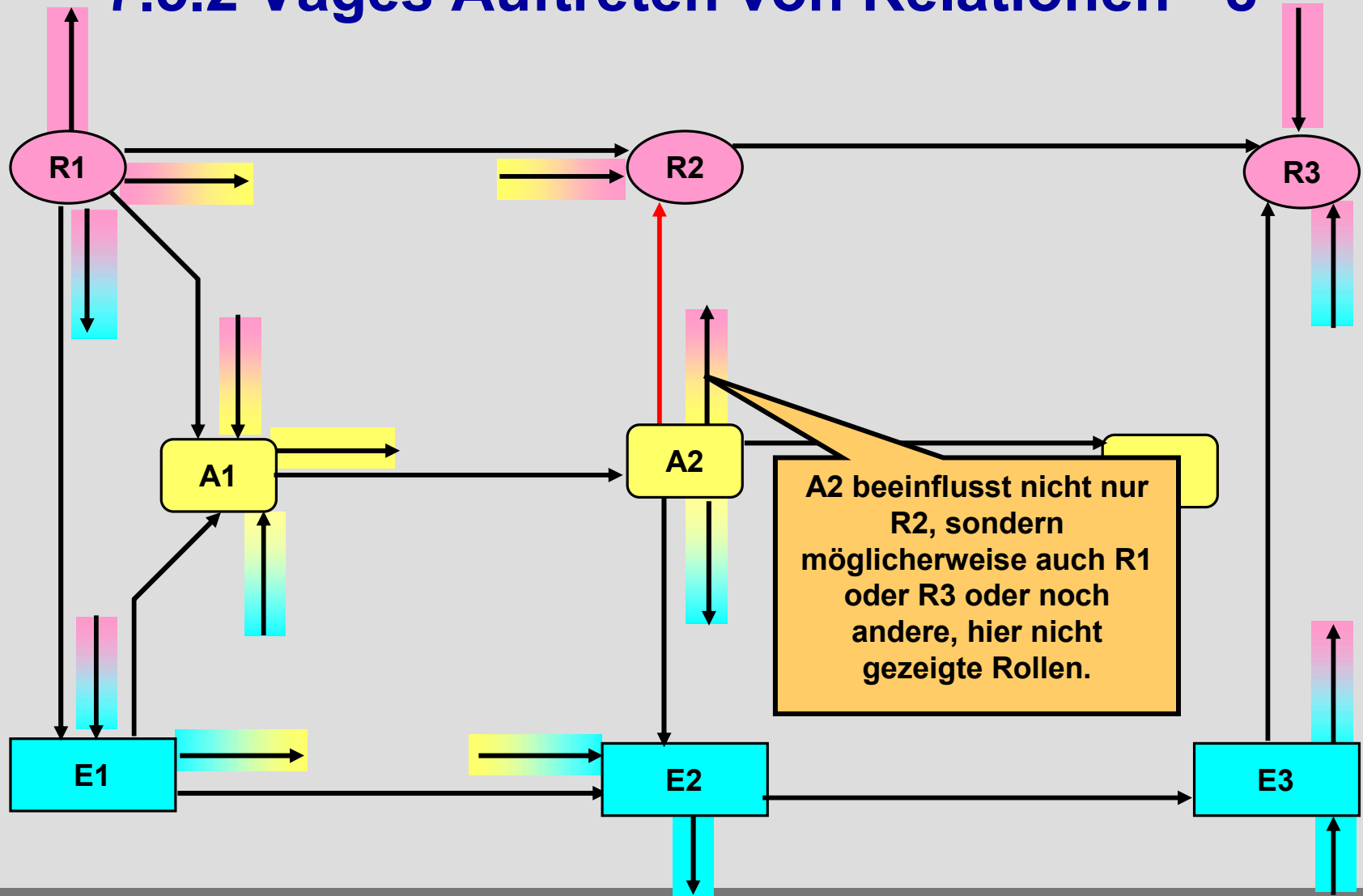
## 7.5.2 Vages Auftreten von Relationen - 6



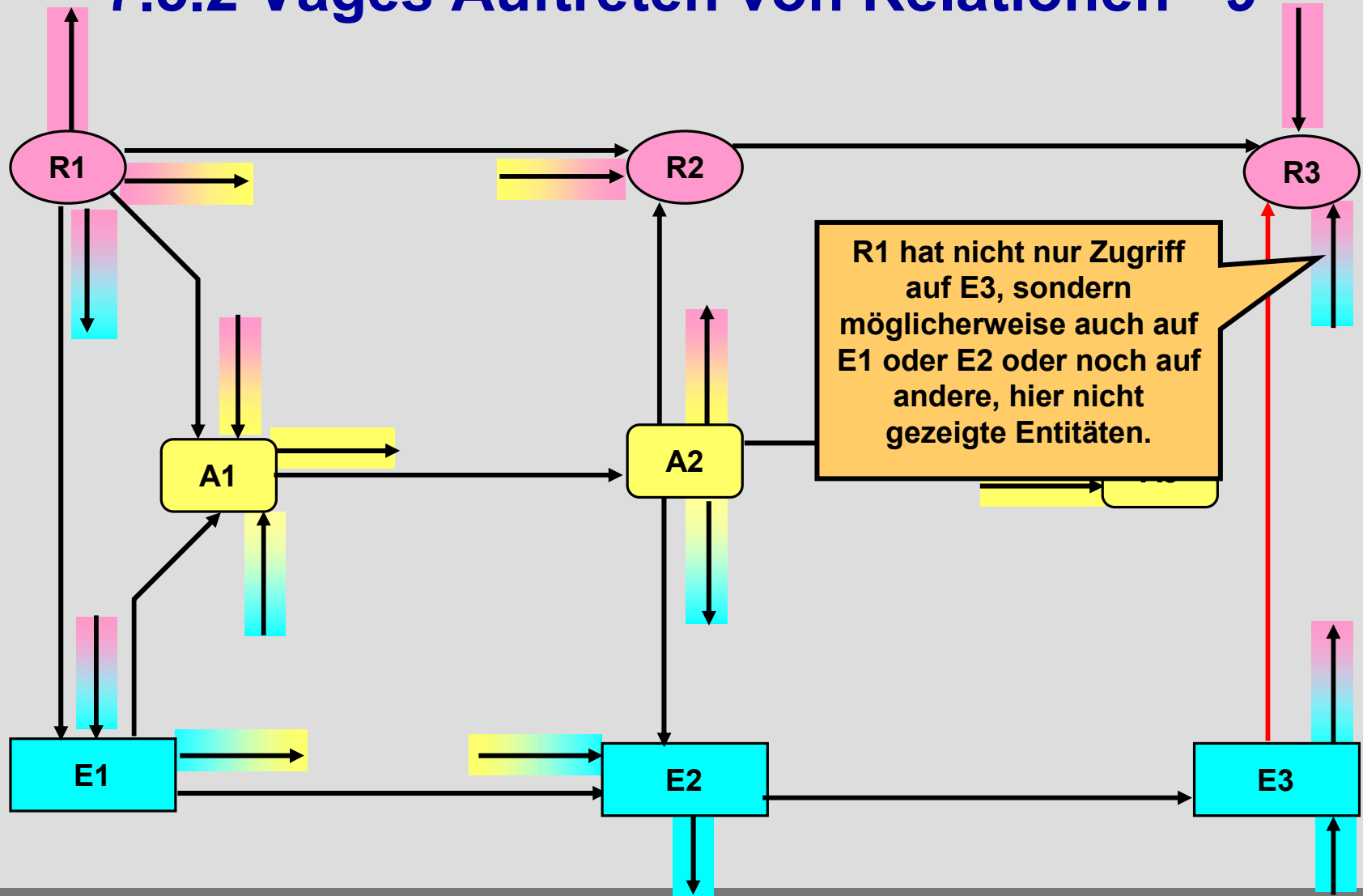
## 7.5.2 Vages Auftreten von Relationen - 7



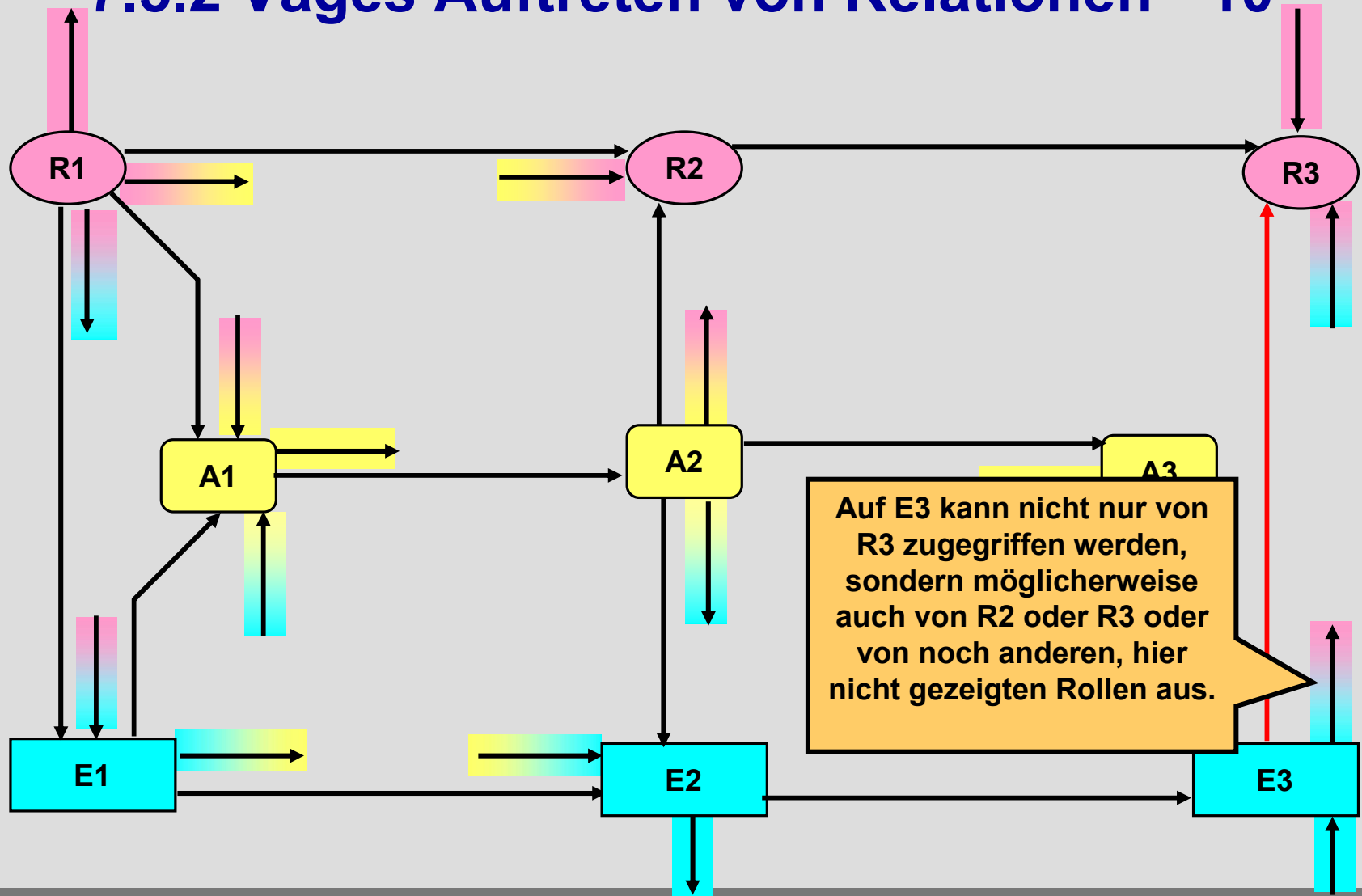
## 7.5.2 Vages Auftreten von Relationen - 8



## 7.5.2 Vages Auftreten von Relationen - 9

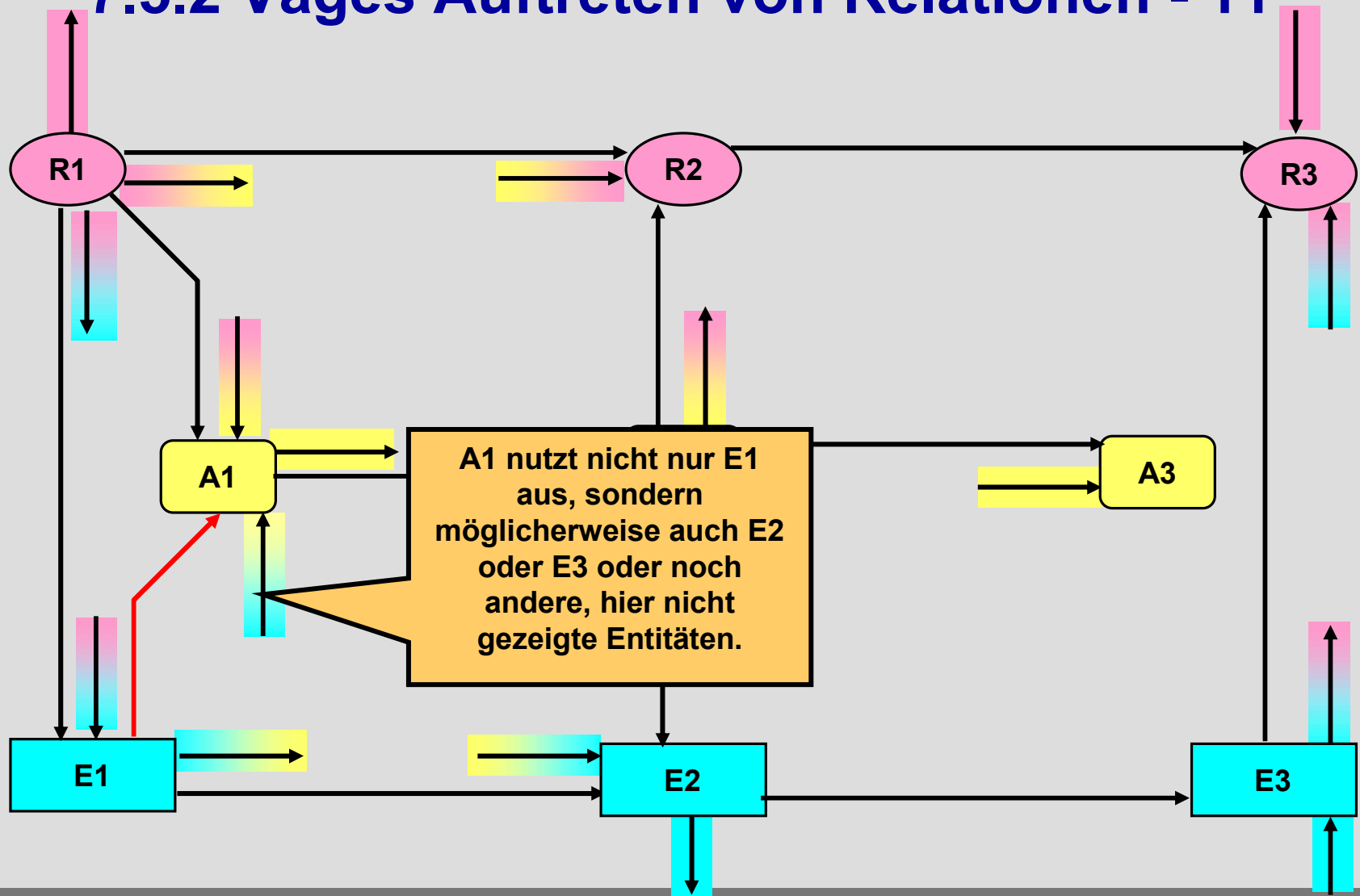


## 7.5.2 Vages Auftreten von Relationen - 10

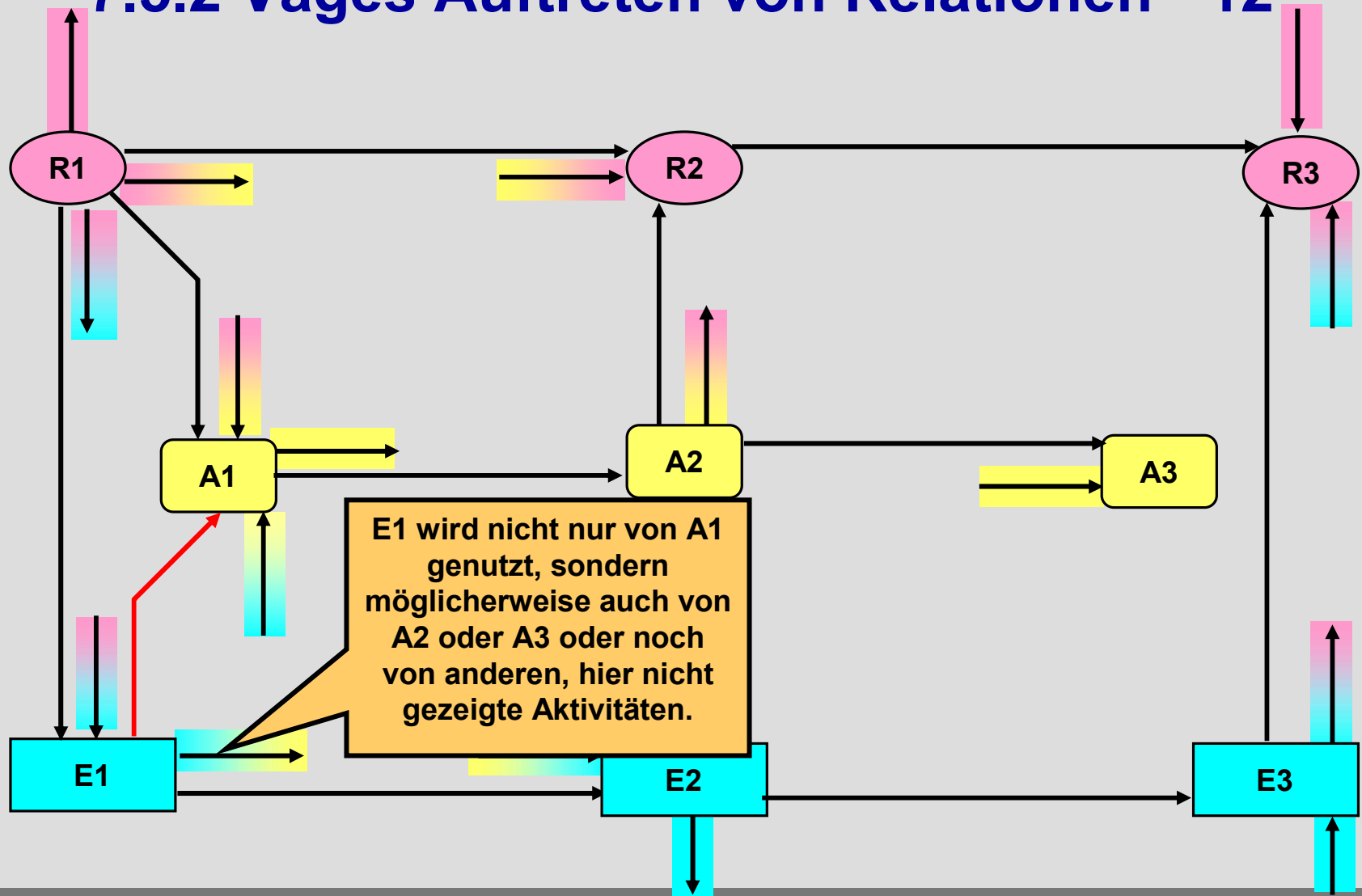




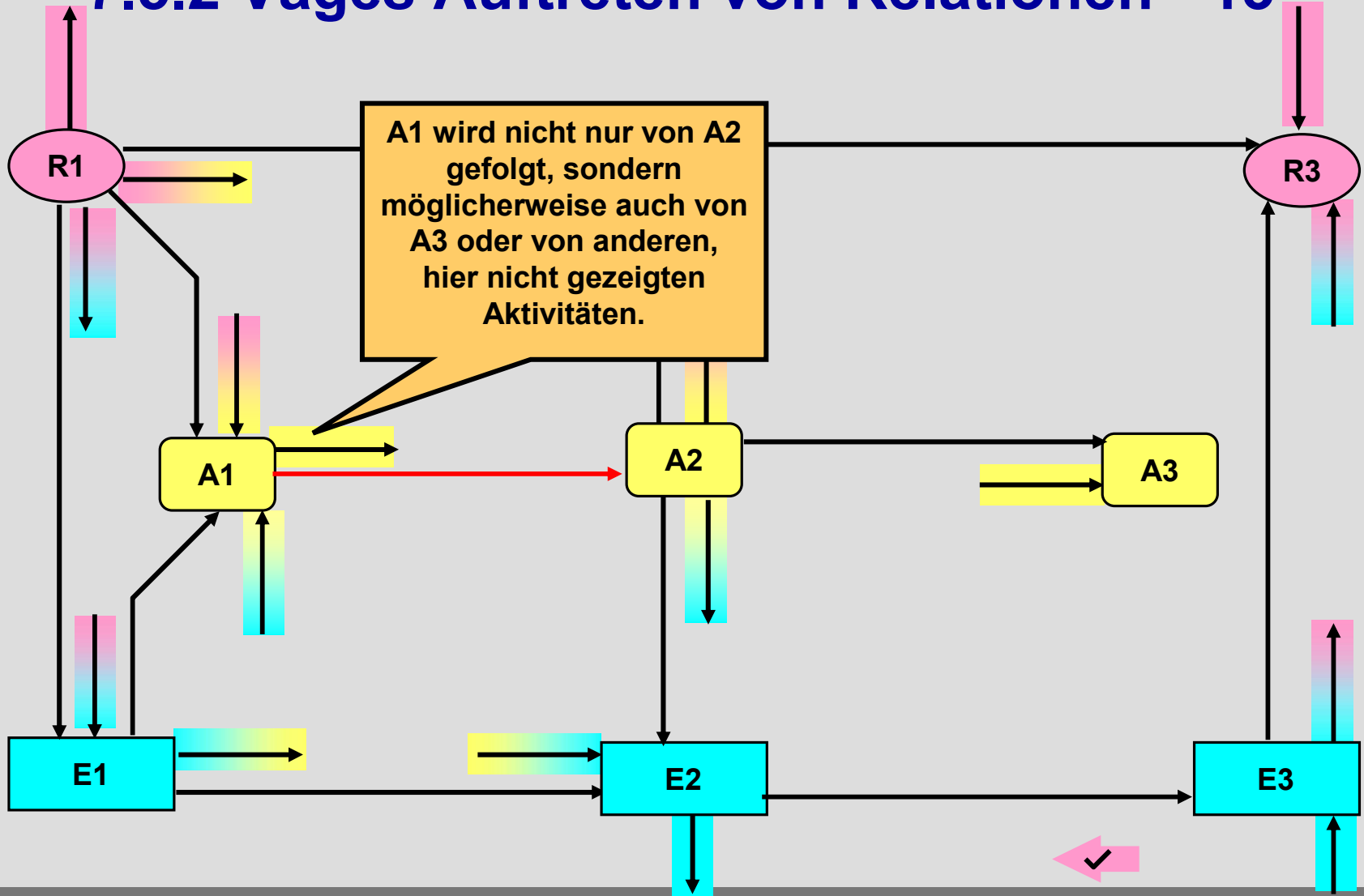
## 7.5.2 Vages Auftreten von Relationen - 11



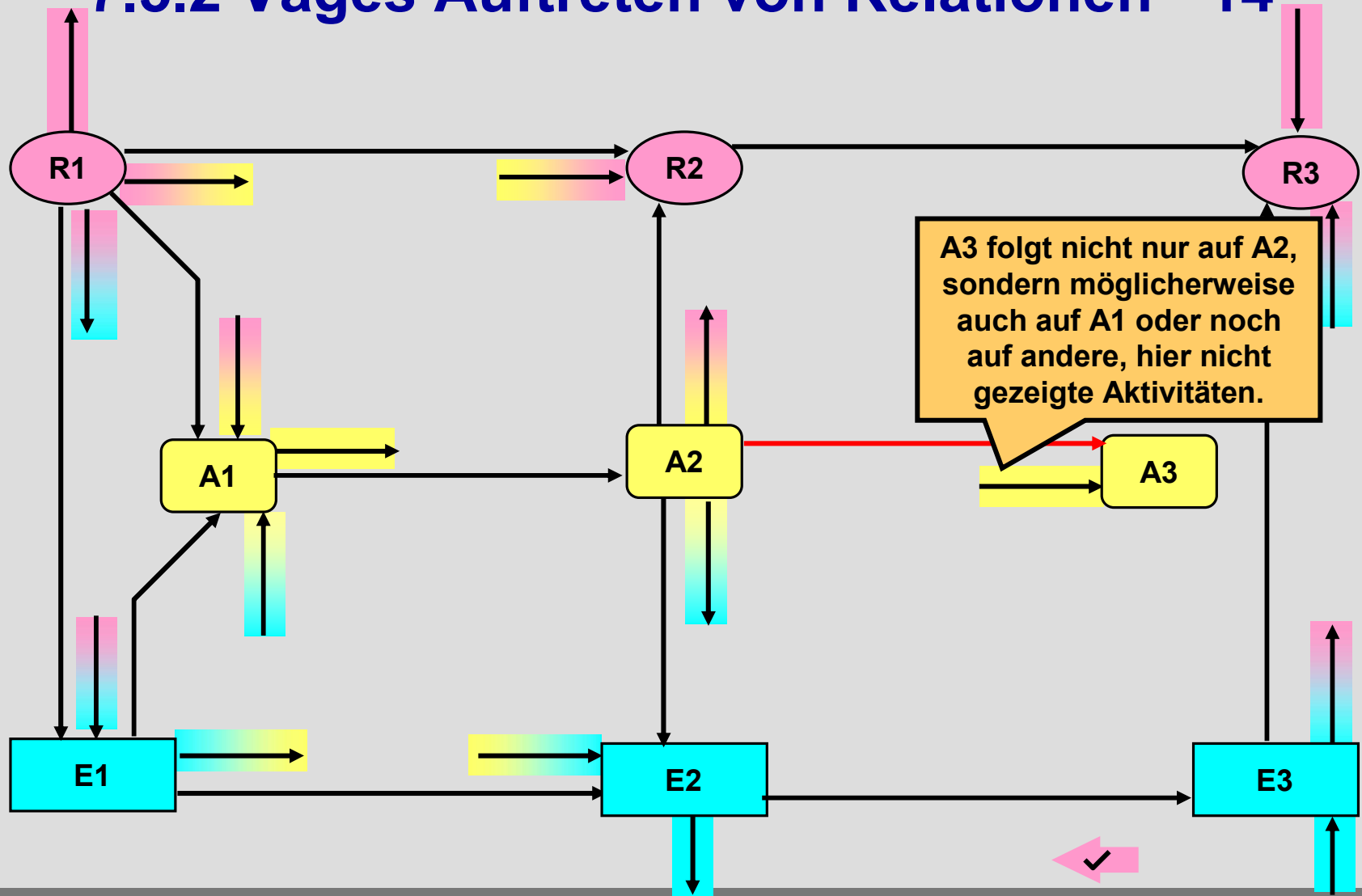
## 7.5.2 Vages Auftreten von Relationen - 12



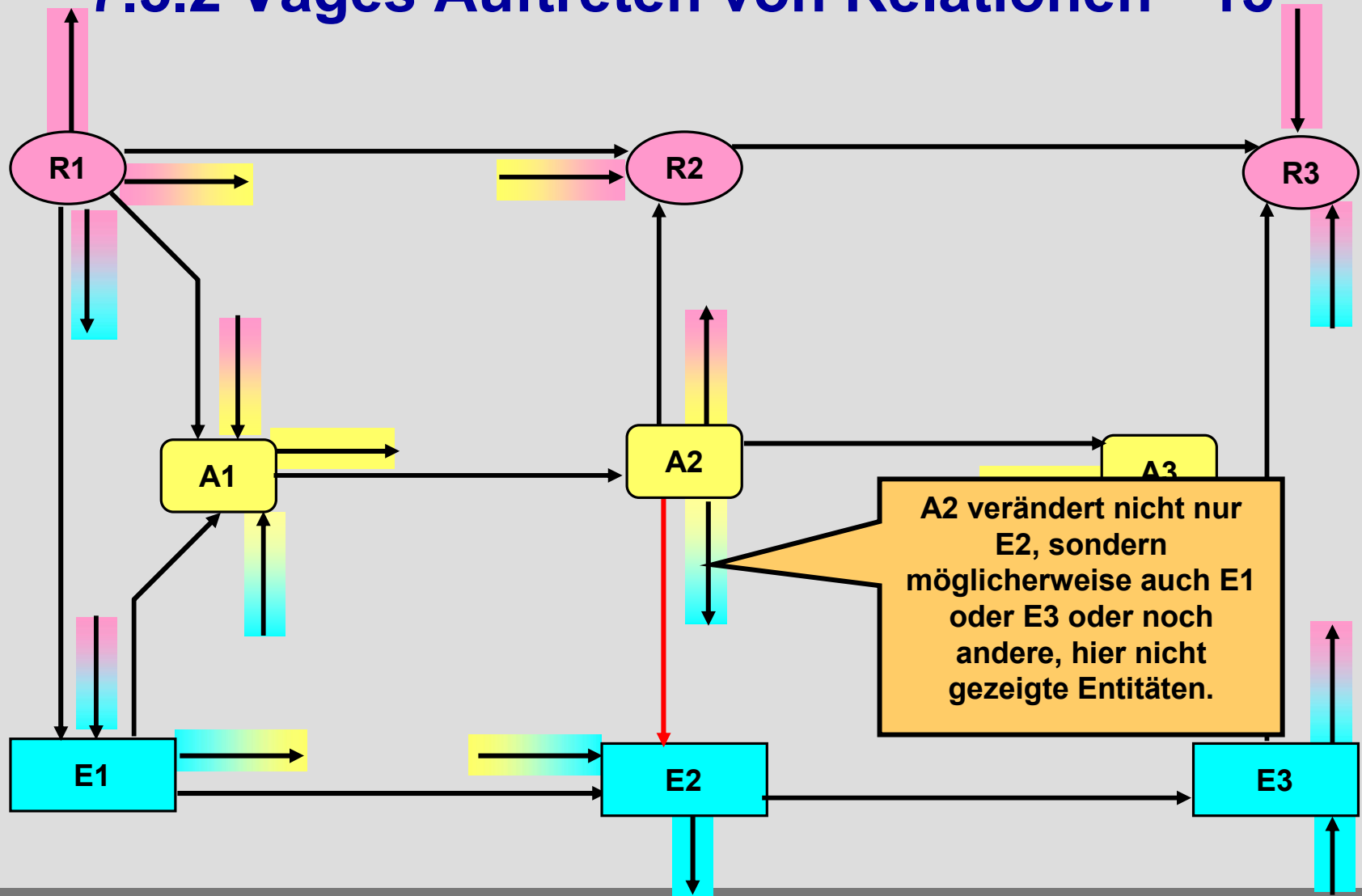
## 7.5.2 Vages Auftreten von Relationen - 13



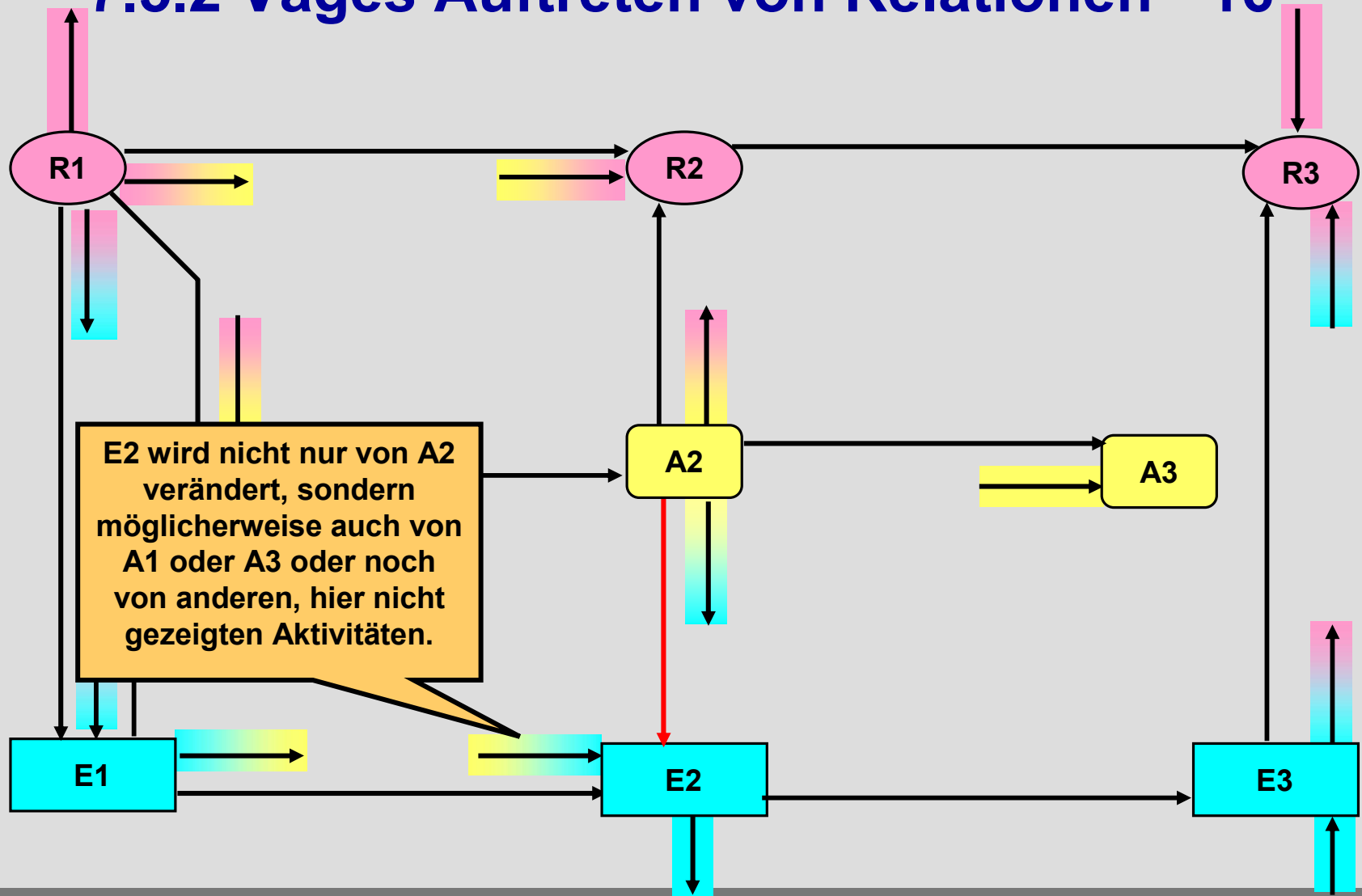
## 7.5.2 Vages Auftreten von Relationen - 14



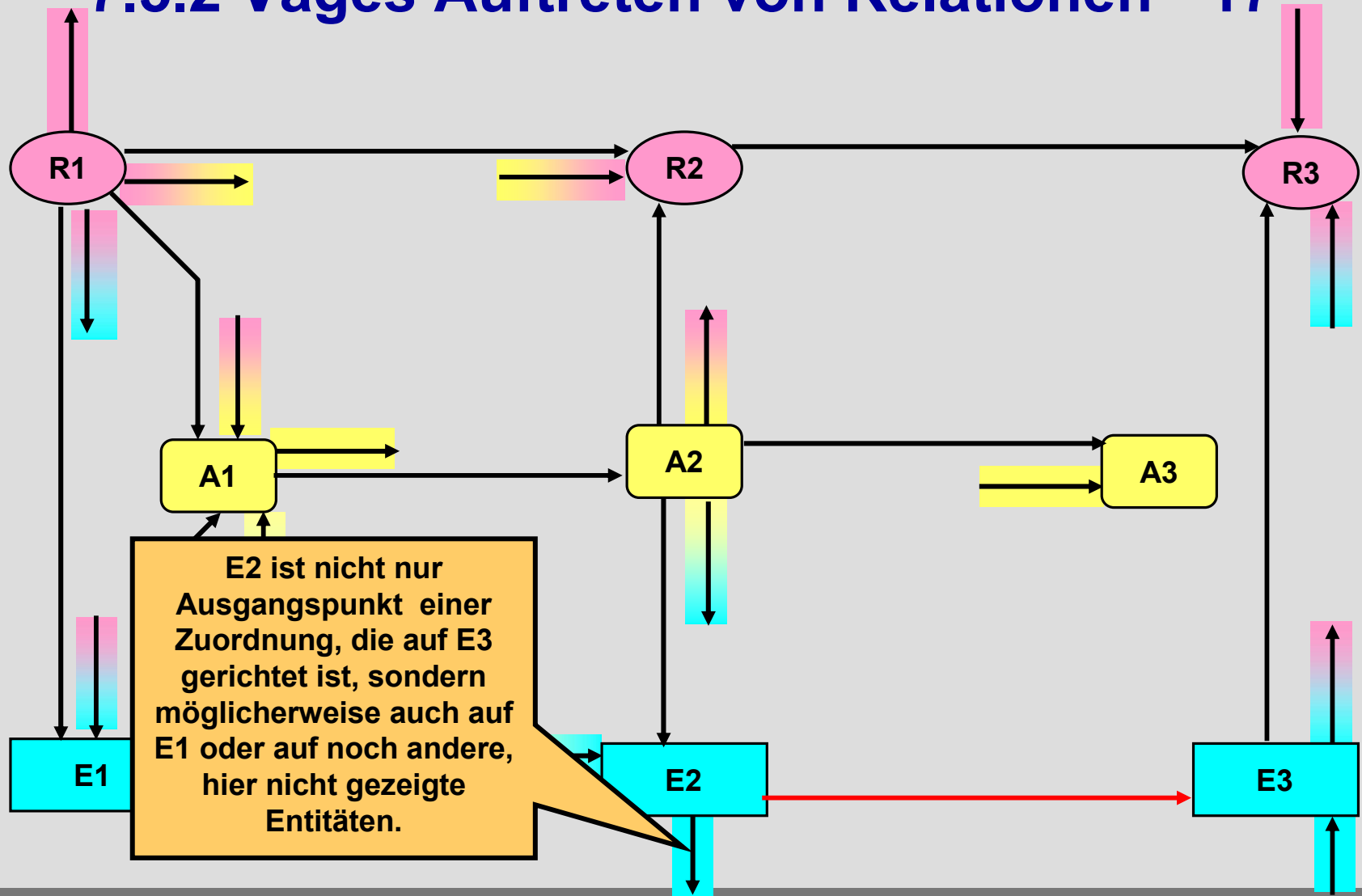
## 7.5.2 Vages Auftreten von Relationen - 15



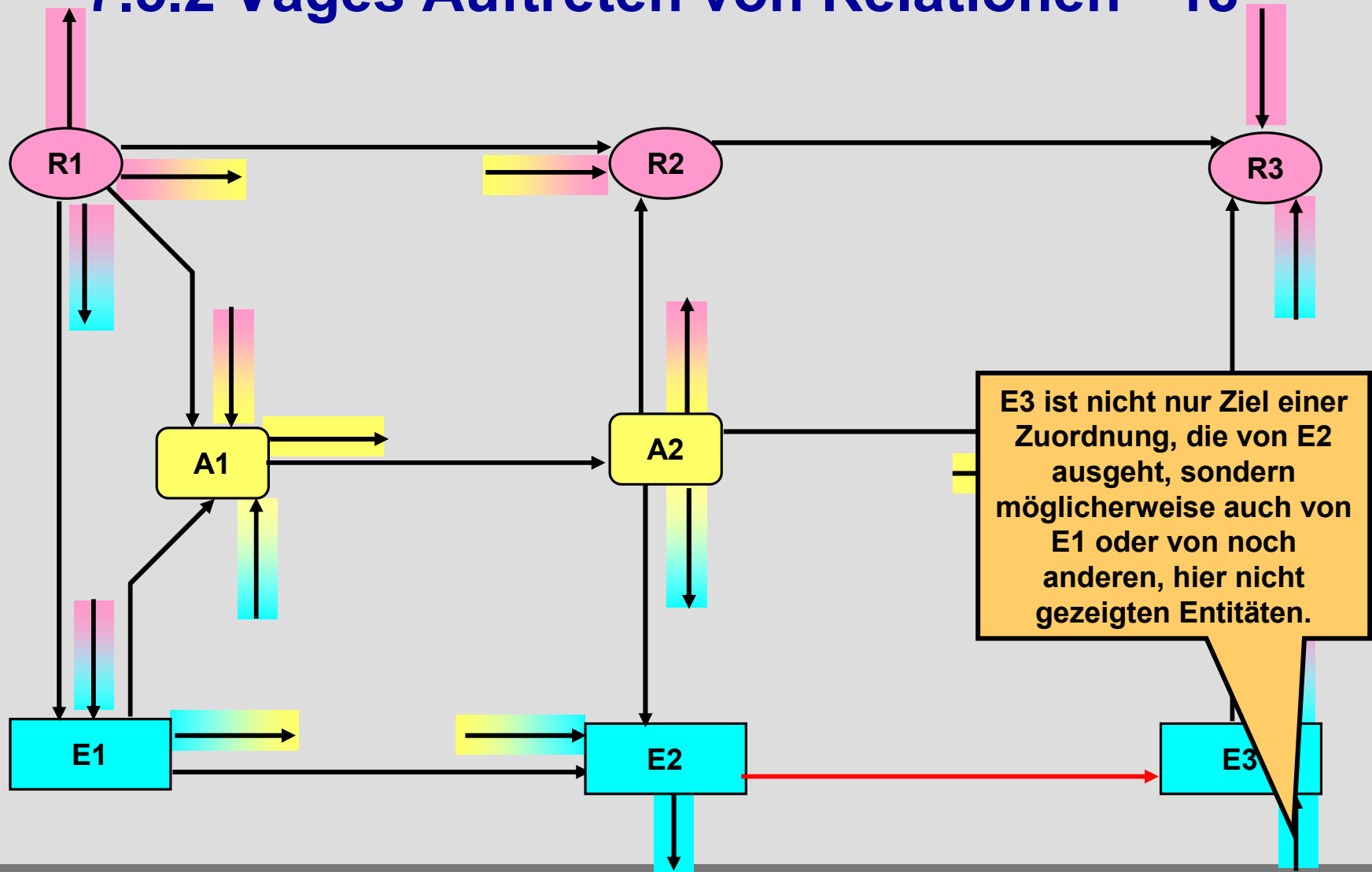
## 7.5.2 Vages Auftreten von Relationen - 16



## 7.5.2 Vages Auftreten von Relationen - 17

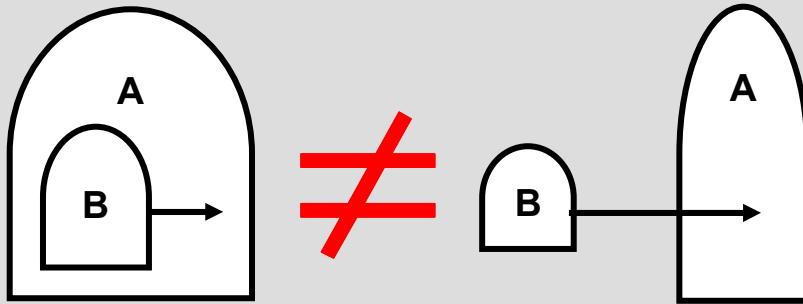


## 7.5.2 Vages Auftreten von Relationen - 18

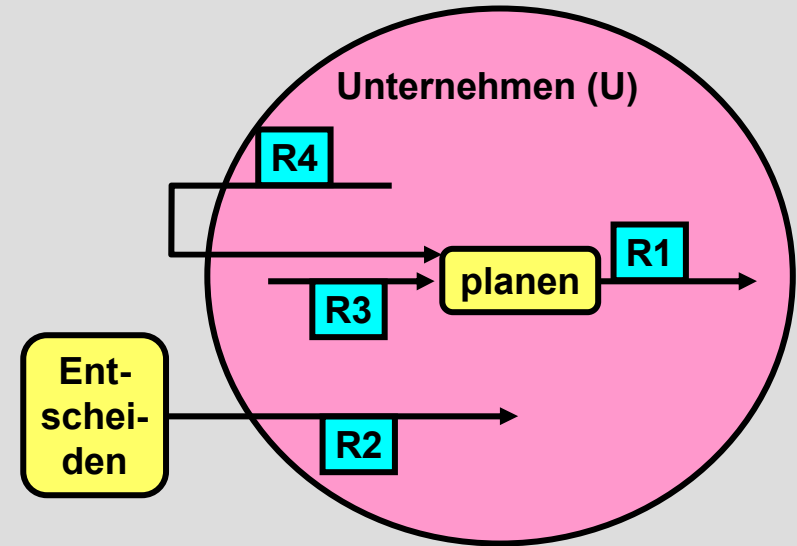




## 7.5.3 Differenzierte Bedeutung



Die Bedeutung der Relation richtet sich im linken Diagramm ausschließlich nach B und ihrer geometrischen Position zu B, während sie sich im rechten Diagramm nach B und A richtet.

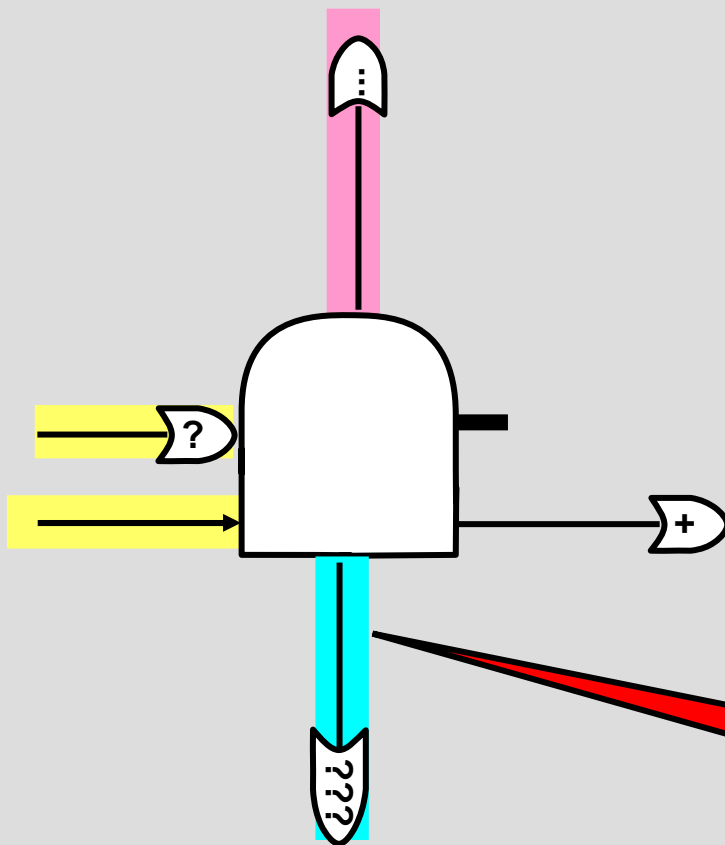


Während R1 auf eine nicht spezifizierte Sub-Aktivität von U zeigt, die auf Planen folgt, zeigt R2 auf eine nicht festgelegte Sub-Rolle von U, die durch das Entscheiden beeinflusst wird. R3 kommt aus einer vorlaufenden Sub-Aktivität von U, R4 von einer Sub-Rolle von U, die das Planen ausführt.

**Aufgabe**



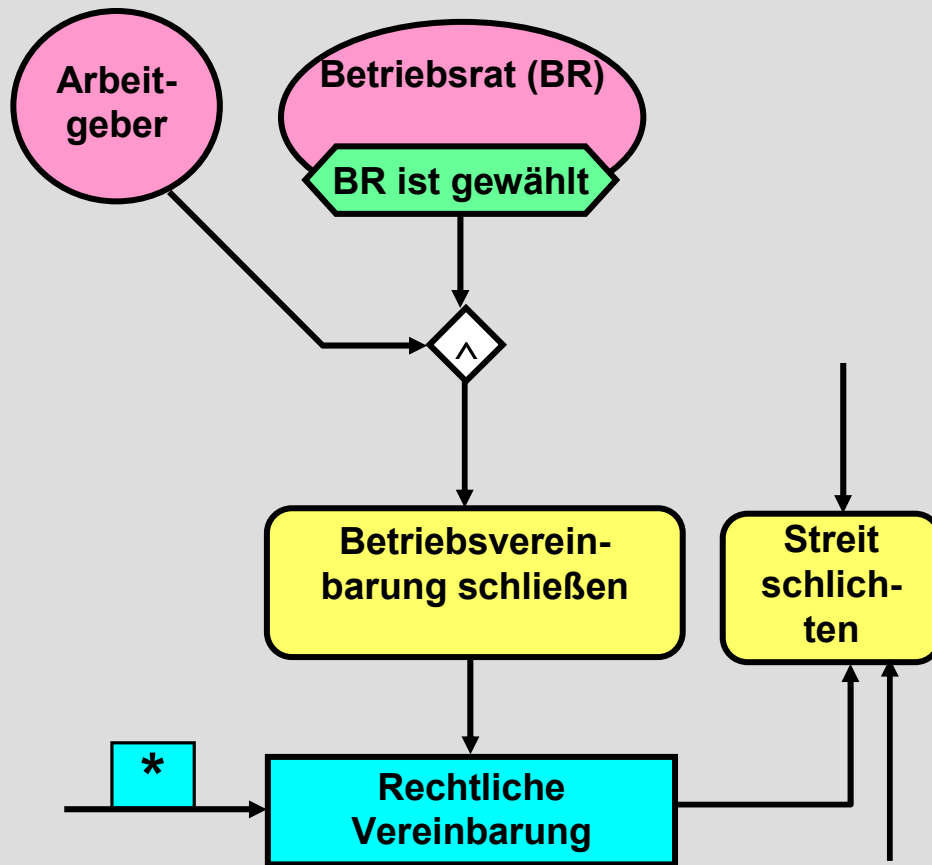
## 7.5.3 Vage Spezifizierung des Auftretens von Relationen



Relationen, die an einer Seite nicht verankert sind, können auch mit Vagheitsymbolen für Relationsstart- und -endpunkte versehen werden. Dies gilt für beide Seiten der Relation, insbesondere auch für die freie. Die Bedeutung dieser Symbole wurde oben definiert.

**Klicken Sie die farbigen Flächen, um die jeweiligen Beispiele zu sehen**

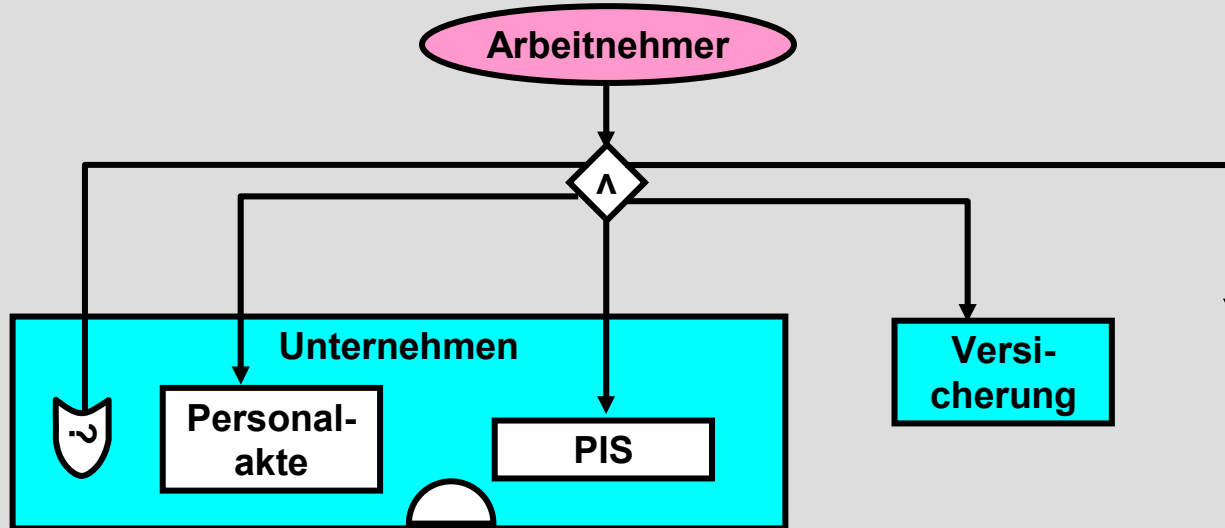
## 7.5.4 Vage Spezifizierung des Auftretens von Relationen – Beispiel I



Es wird ausgedrückt, dass eine rechtliche Vereinbarung auch aus anderen Aktivitäten als der des *Betriebsvereinbarung schließen* resultieren kann, dass das *Streit schlichten* noch andere Dokumente nutzen kann und dass das *Streit schlichten* durch Rollen ausgeführt wird.

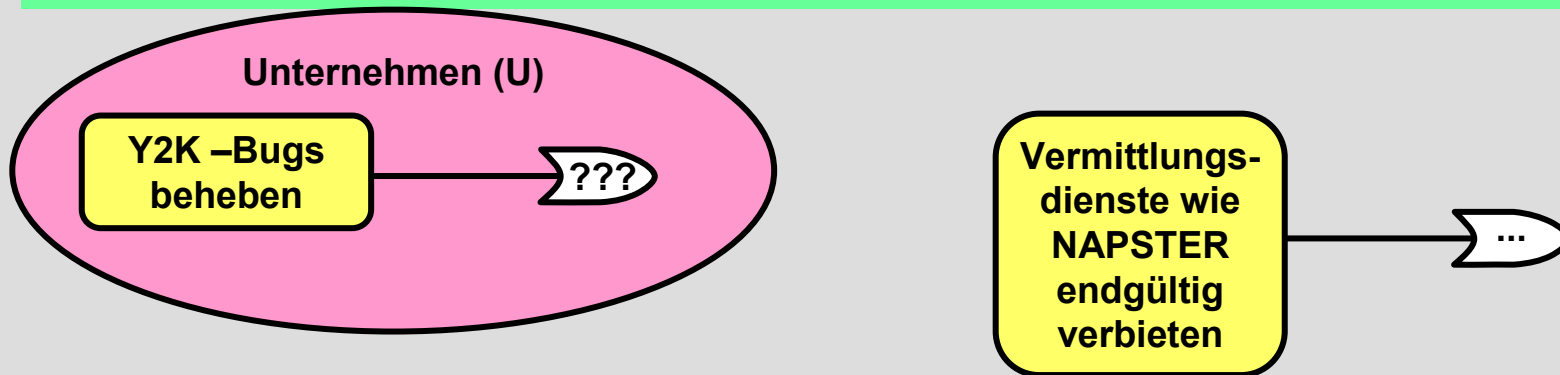
Die letzte Angabe könnte man sich sparen, so lange klar ist, dass die Aktivität nicht ausschließlich von technischen Systemen erledigt werden kann.

## 7.5.4 Vage Spezifizierung des Auftretens von Relationen – Beispiel II



Die Darstellung besagt, dass an mehreren Stellen Beschreibungen über einen Arbeitnehmer liegen und dass es noch weitere Stellen neben dem Unternehmen und den Versicherungen gibt, die hier aber nicht spezifiziert sind. Es wird ferner in Frage gestellt, ob es korrekt ist, dass im Unternehmen noch weitere Beschreibungen existieren können, die nicht spezifiziert sind.

## 7.5.4 Vage Spezifizierung des Auftretens von Relationen – Beispiel III

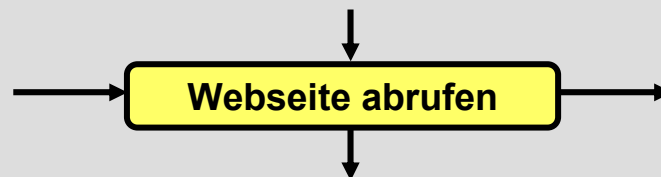


Das Diagramm besagt, dass man sich unsicher ist, ob nach der Behebung der Jahr 2000 Probleme eines Unternehmens noch weitere Aktivitäten folgen sollen, die als Nachfolger der Problembehebung zu betrachten sind.

Die Aussage ist, dass wir nicht wissen, welche Folgeaktivitäten das Verbot solcher Vermittlungsdienste im WWW hervorrufen würde.

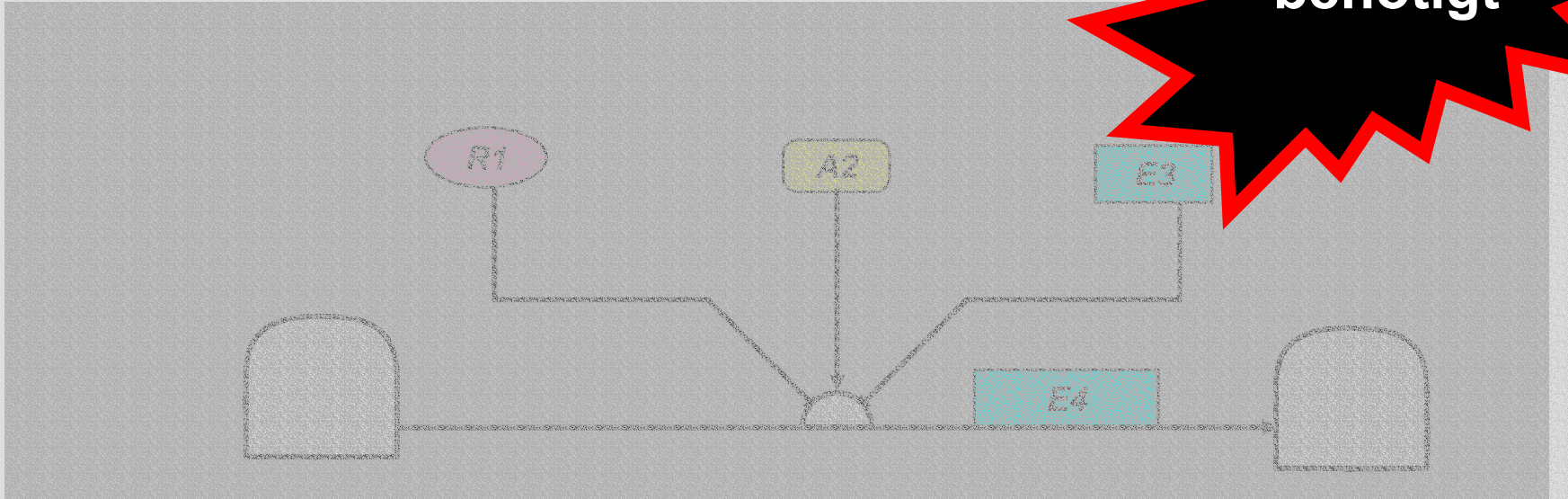
## 7.5.5 Aufgaben zu vager Spezifizierung des Auftretens von Relationen

1. Erläutern Sie die rechte Abbildung bei Wann braucht man vage Spezifizierung des Auftretens von Relationen und ersetzen sie die abstrakten Rollen, Aktivitäten und Entitäten durch ein konkretes Beispiel.
2. Ergänzen Sie die offenen Enden bzw. Startpunkt der unten gezeigten Grafik durch geeignete Beispiele für Basis-Elemente.
3. Vervollständigen Sie die linke Abbildung bei Differenzierte Bedeutung so, dass zum Ausdruck kommt, dass das *Planen* unspezifizierter Sub-Rollen des Unternehmens beeinflusst.
4. Entfernen Sie in der Abbildung zur Einbettung von Rollen in Rollen die mittlere Hierarchieebene (Referate) und deuten Sie dennoch an, dass die Leitung gegenüber anderen Rollen, wie auch den Sachbearbeitern Weisungsbefugnis hat und dass diese auch Weisungsempfänger sind.
5. Stellen Sie dar, dass eine Rolle am Zustandekommen einer bestimmten Relation Interesse hat, dass dies aber auch noch für andere, nicht näher spezifizierte Rollen gilt.



## 7.6. Vage Spezifizierung von Relationen weitere Basis-Elemente

**Selten  
benötigt**



Auch die ergänzende Spezifizierung von Relationen durch Entitäten, Aktivitäten oder interessierte Rollen kann Gegenstand von Vagheit sein.

7.6.1

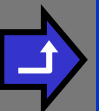
**Definitionen**

7.6.2

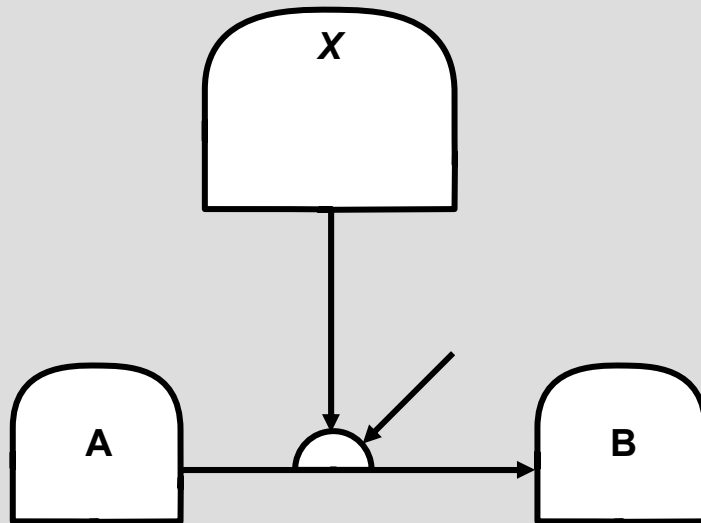
**Beispiele**

7.6.3

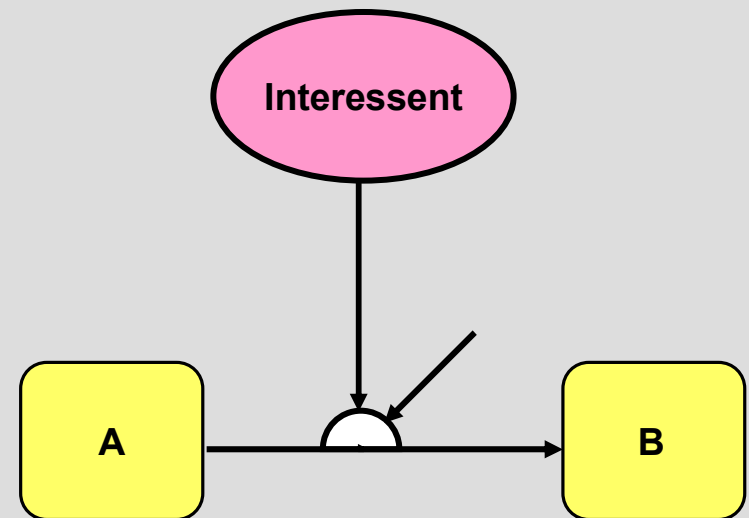
**Aufgaben**



## 7.6.1 Vage Spezifizierung von Relationen durch eine zusätzliche Relation



Neben X können auch noch andere unspezifizierte Basis-Elemente zur Spezifizierung der Relation relevant sein.

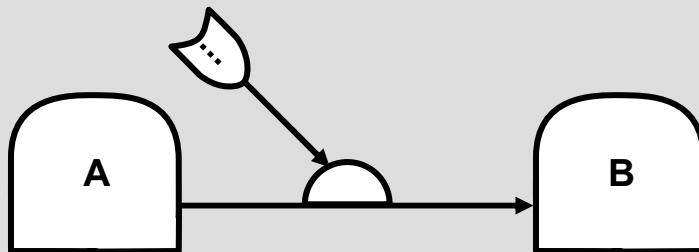


Neben der Rolle *Interessent* kann noch eine Aktivität und/oder eine Entität für die nähere Festlegung der Relation „folgt auf“ relevant sein.

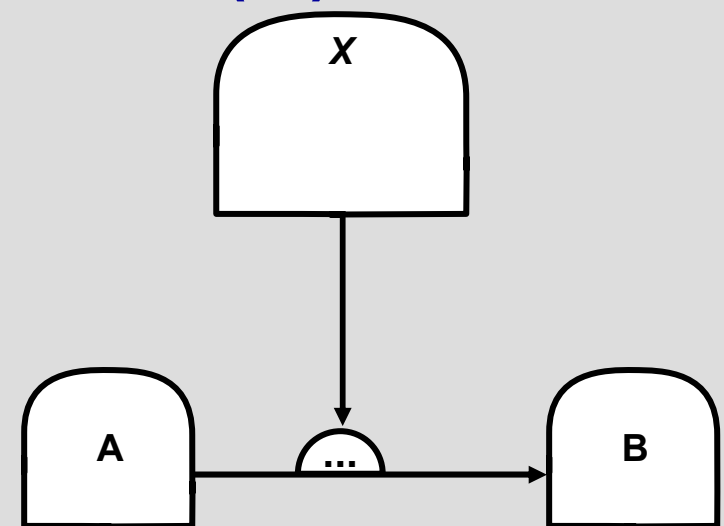
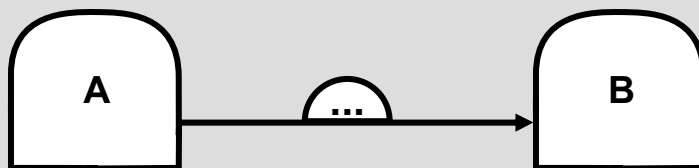




## 7.6.1 Vage Spezifizierung von Relationen durch weitere Basis-Elemente (...)



oder



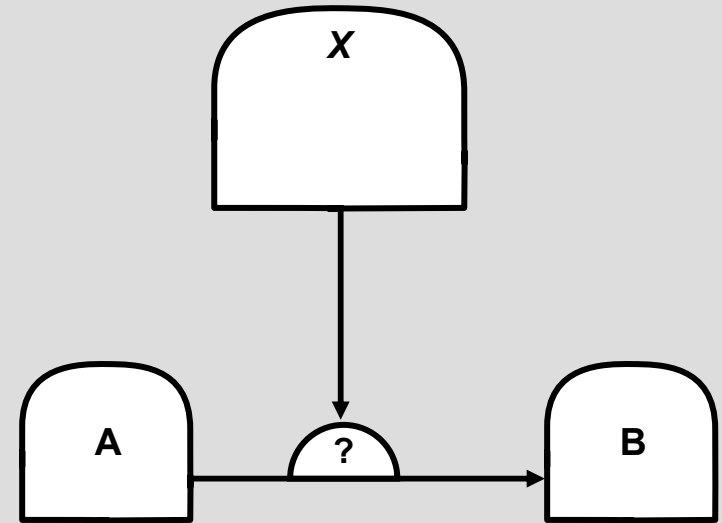
Es gibt Basis-Elemente, die zur Spezifizierung der Relation beitragen, wir wissen aber nicht, wie wir diese Basis-Elemente festlegen können.

Neben X gibt es noch weitere Basis-Elemente, die zur Spezifizierung der Relation beitragen, wir wissen aber nicht, wie wir diese Basis-Elemente festlegen können.

## 7.6.1 Vage Spezifizierung von Relationen durch weitere Basis-Elemente (?)



Wir wissen nicht, ob es korrekt ist, dass die nähere Spezifizierung der Relation offen gelassen wird.

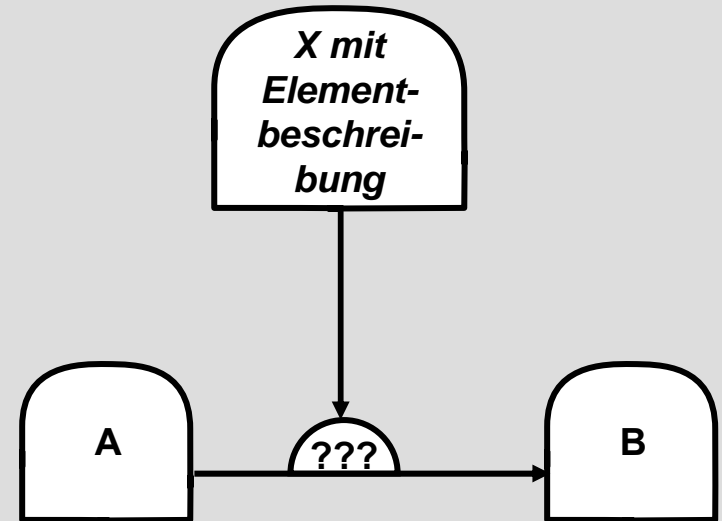


Wir wissen nicht, ob es korrekt ist, dass die Relation mit Hilfe von X näher spezifiziert ist.

## 7.6.1 Vage Spezifizierung von Relationen durch weitere Basis-Elemente (???)

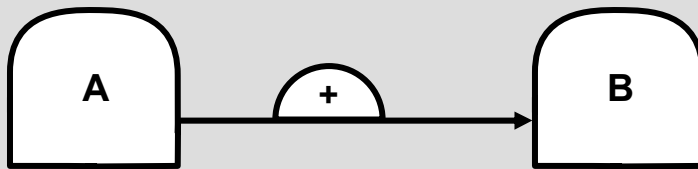


Wir wissen nicht, ob es möglich ist,  
die Relation näher zu spezifizieren.

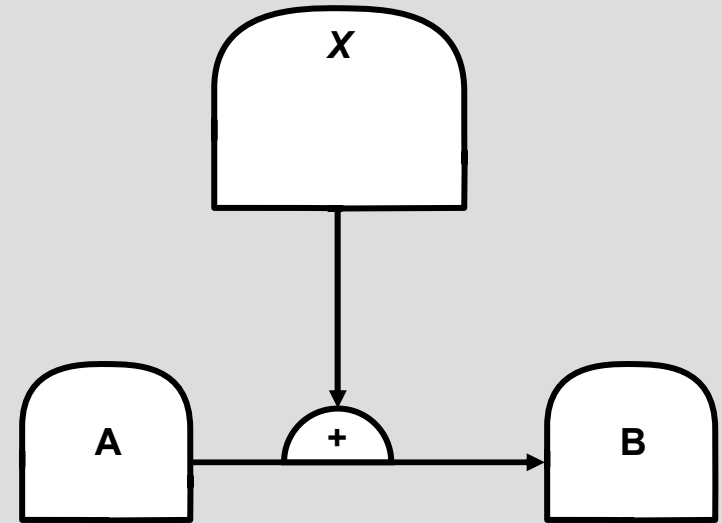


Wir wissen nicht, ob es neben der  
Spezifizierung der Relation mittels X  
noch weitere spezifizierende Basis-  
Elemente gibt.

## 7.6.1 Vage Spezifizierung von Relationen durch weitere Basis-Elemente (+)

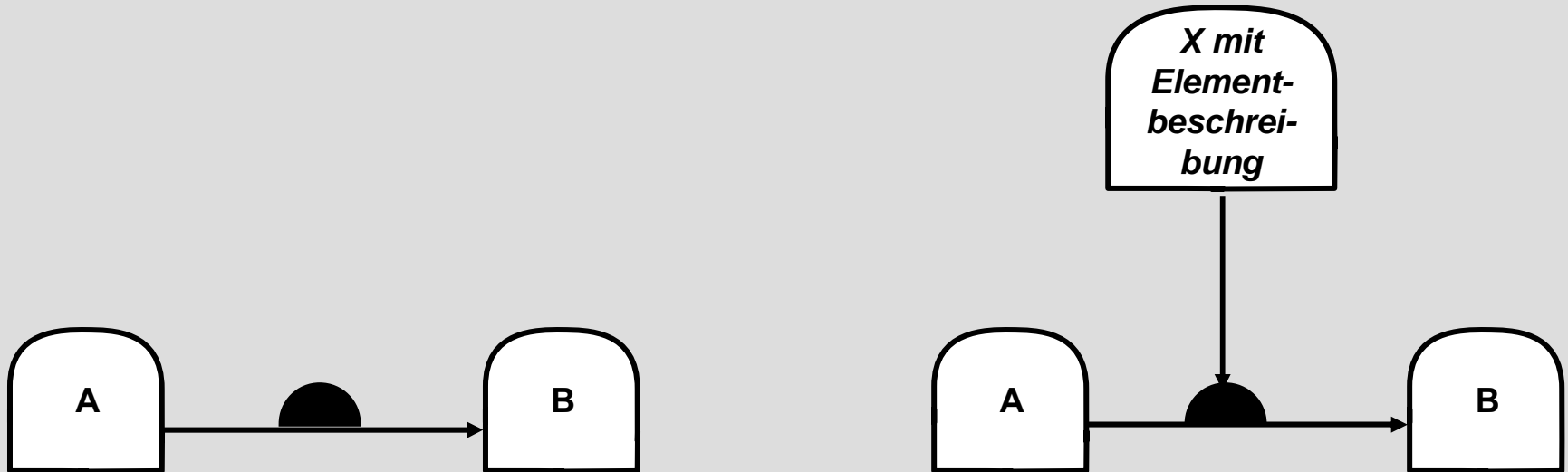


Der Modellierer kennt Informationen zur näheren Spezifizierung der Relation.



Der Modellierer kennt Informationen, die die Relationen neben X ergänzend spezifizieren.

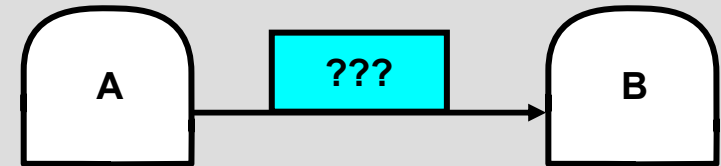
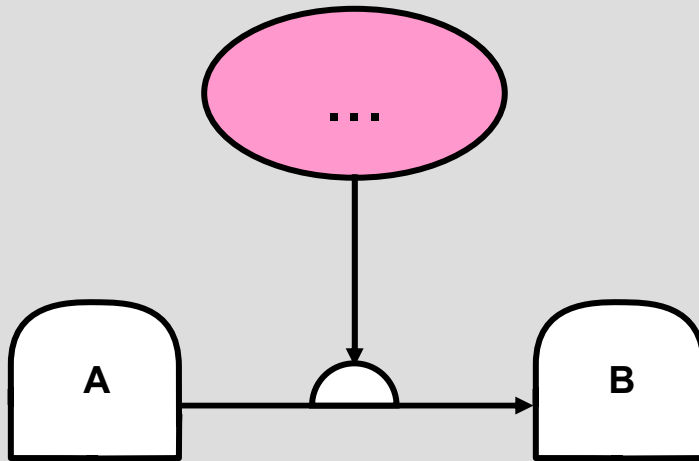
## 7.6.1 Vage Spezifizierung von Relationen durch weitere Basis-Elemente (Ein-Ausblenden)



Durch Klicken auf den schwarzen Halbkreis erhält man weitere Basis-Elemente angezeigt, die die Relation spezifizieren.



## 7.6.2 Beispiel I - Vage Spezifizierung von Relationen durch weitere Basis-Elemente

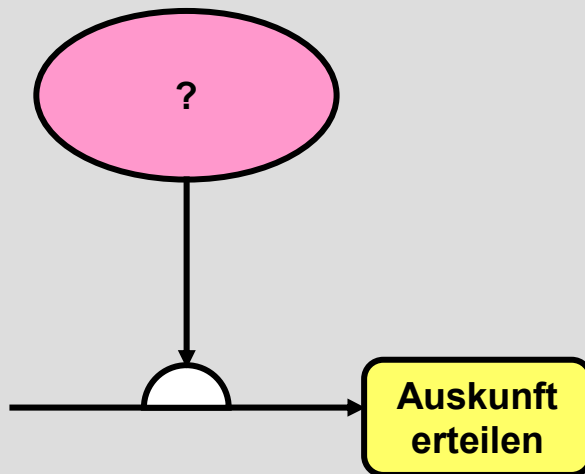


Wir wissen, dass eine Rolle an der Relation interessiert ist, aber wir wissen nicht, wie wir sie genauer festlegen sollen. Da in sozio-technischen Systemen an jeder Relation eine Rolle interessiert sein muss, wird hier auch ausgedrückt, dass wir nicht wissen welche Rolle dies ist.

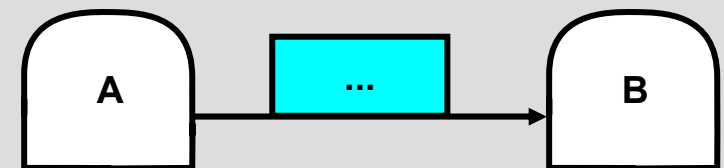
Es ist unsicher, ob die Standarddefinition der Relation ausreicht, oder ob sie ergänzt werden soll bzw. undefiniert werden muss.



## 7.6.2 Vage Spezifizierung von Relationen durch weitere Basis-Elemente - Beispiele



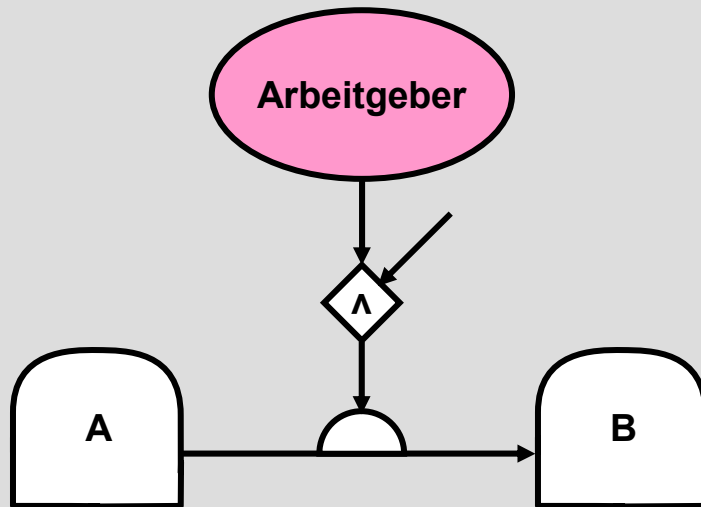
Es ist unter Umständen nicht korrekt, dass man offen lässt, welche Rolle an der Erteilung einer Auskunft Interesse hat.



Es ist sicher, dass die Bedeutung der Relation neu definiert oder ergänzt werden muss, man weiß aber nicht wie.



## 7.6.2 Beispiel II -Vage Spezifizierung von Relationen durch weitere Basis-Elemente



Neben dem Arbeitgeber hat noch eine weitere Rolle Interesse an der Relation, es ist aber nicht sinnvoll oder nicht gewünscht, diese näher zu spezifizieren.

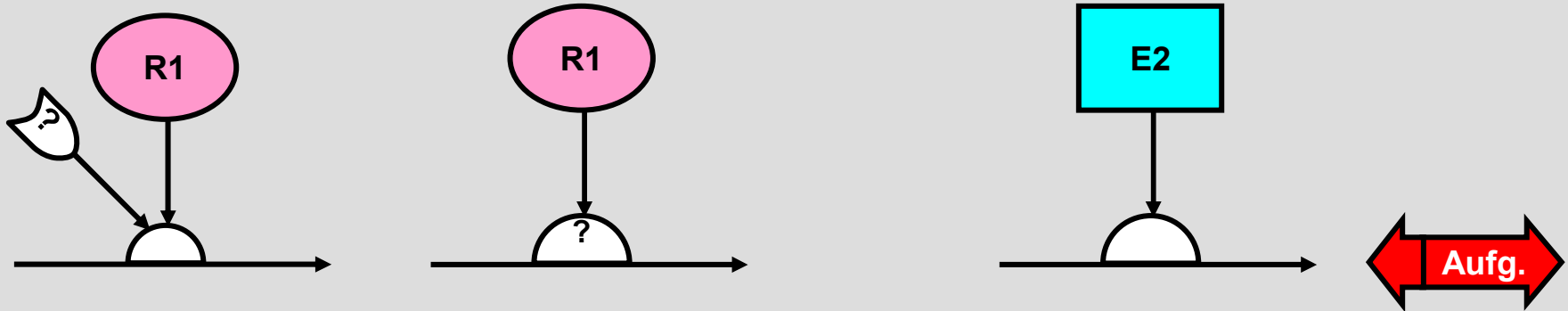


Wenn man die schwarze Fläche anklickt erhält man spezifizierende Details zu der Relation (z.B. Name, Typ, Quantoren).



## 7.6.3 Aufgaben zu vager Spezifizierung von Relationen durch weitere Basis-Elemente

1. Erläutern Sie den Unterschied zwischen den beiden unten gezeigten Grafiken, die die Rolle R1 beinhalten.
2. Wie stellen Sie den Sachverhalt, dass eine Aktivität A das Zustandekommen eine Relation unterstützt und optional noch weitere, nicht näher spezifizierte Aktivitäten zu der Unterstützung beitragen.
3. Ergänzen Sie die Grafik mit der Entität E2 so, dass die Relation noch durch eine weitere Aktivität unterstützt wird, die nicht spezifiziert werden kann. Zeigen Sie zusätzlich ein Diagramm, das aussagt, dass die Relation noch durch weitere Basis-Elemente näher spezifiziert wird, nicht aber durch eine weitere Entität.



## 7.7 Vage Spezifizierung von Modifikatoren

Auch Modifikatoren können mit Hinblick sowohl auf die Bedingungen oder Ereignisse als auch auf die Häufigkeitsangaben vage spezifiziert werden.

Da die Modifikatoren die Bedingungen und Umstände beschreiben, unter denen andere Elemente existieren, ist somit eine weitere Möglichkeit gegeben, Unsicherheit zu beschreiben.

Ereignisse/Bedingungen und Häufigkeitsangaben sind wiederum Attribute, deren Wertangaben mit Modifikatoren versehen werden können und denen auch Vagheitsindikatoren zugeordnet werden können ([s. Kap. 7.9](#)).

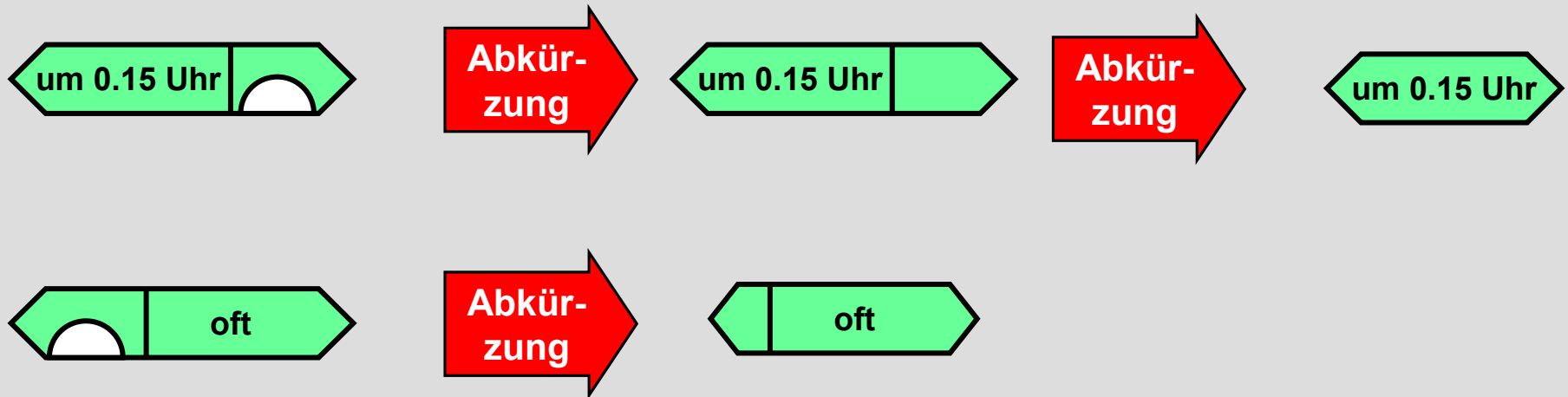
Die Wertangaben selbst können durch ein Wertespektrum oder durch eine Wertemenge präsentiert werden.

Somit ergeben sich mehrere Ebenen, um Vagheit darzustellen.

- [7.7.1](#) Basis Darstellungen
- [7.7.2](#) Kombinationen
- [7.7.3](#) Super-Modifikatoren
- [7.7.4](#) Beispiele
- [7.7.5](#) Aufgaben



## 7.7.1 Abkürzungsregel I



Wenn es nicht sinnvoll oder nicht gewünscht ist, die Häufigkeit zu spezifizieren, dann entfällt der Teilungsstrich des Modifikators. Die dann eingetragenen Spezifikationen beziehen sich nur auf das Ereignis bzw. die Bedingung.

Falls nur die Häufigkeit spezifiziert werden soll, bleibt der senkrechte Strich enthalten, um Verwechslung zu vermeiden.



## 7.7.1 Vage Spezifizierung von Modifikatoren - Ereignisse



Es ist nicht sinnvoll bzw. nicht gewünscht, neben dem Ereignis bzw. der Bedingung X noch weitere zu spezifizieren.



Man weiß nicht, wie man X mit welchen Ereignissen noch logisch verknüpfen kann. Falls die Art der logische Verknüpfung bekannt ist, können die drei Punkte direkt angefügt werden→.

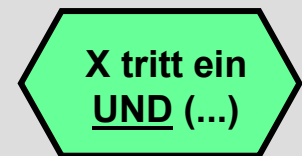


Es ist unsicher, ob Art und Umfang der Spezifizierung des Ereignisses bzw. der Bedingung durch „X tritt ein“ angemessen ist. Falls sich der Zweifel unmittelbar auf X bezieht kann das Fragezeichen dort eingefügt werden→.

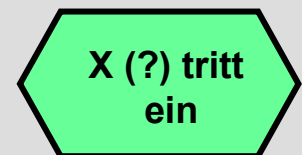


Man weiß nicht, ob neben dem Ereignis bzw. der Bedingung X noch weitere spezifiziert werden können bzw. müssen.

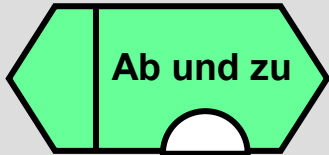
(s. Kap. 7.9):



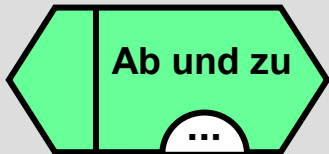
(s. Kap. 7.9):



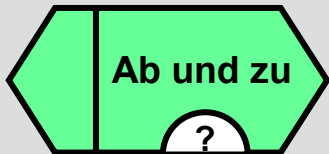
## 7.7.1 Vage Spezifizierung von Modifikatoren - Häufigkeit



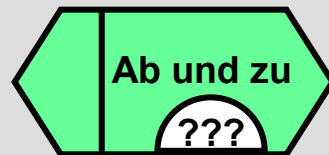
Es ist nicht sinnvoll bzw. nicht gewünscht, die Häufigkeitsangabe noch weiter zu spezifizieren.



Man weiß nicht, wie man die Häufigkeitsangabe noch weiter spezifizieren kann.



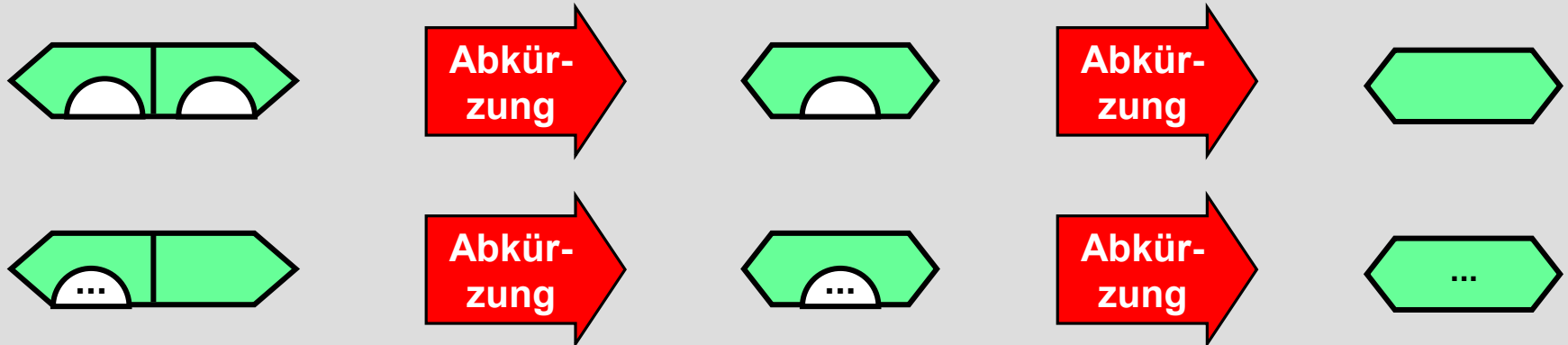
Es ist unsicher, ob die Häufigkeitsangabe angemessen ist.



Man weiß nicht, ob die Häufigkeitsangabe noch weiter spezifiziert werden kann oder muss.



## 7.7.1 Abkürzungsregel II



Der Teilungsstrich des Modifikators entfällt auch, wenn das Ereignis bzw. die Bedingung durch Vagheitssymbole ersetzt werden und es nicht sinnvoll oder nicht gewünscht ist, die Häufigkeit zu spezifizieren.

## 7.7.1 Beispiele - Ereignisse

Daten sichern

ab 0.15 Uhr

Die Datensicherung findet unter der Bedingung statt, dass der Zeitpunkt 0.15 erreicht ist. Dies beinhaltet nicht die Häufigkeitsangabe „täglich“

Daten sichern

Es findet eine Datensicherung statt, aber man findet es nicht sinnvoll oder man will nicht die Bedingungen angeben, unter denen dies der Fall ist.

Daten sichern

...

Man weiß, dass Datensicherung stattfindet, aber nicht die Bedingungen, unter denen dies der Fall ist.

Daten sichern

?

Man weiß, dass Datensicherung stattfindet, ist sich aber unsicher, ob es korrekt ist, dass man die näheren Bedingungen, unter denen sie stattfindet, nicht spezifiziert.

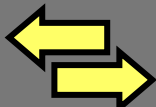
Daten sichern

???

Man weiß nicht, ob es Bedingungen gibt, unter denen Datensicherung stattfindet. Es ist damit insgesamt zweifelhaft, ob die Aktivität „Daten sichern“ an der jeweiligen Stelle modelliert werden sollte

Es ist nicht sinnvoll oder unerwünscht anzugeben, wie häufig Datensicherung stattfindet

+



## 7.7.1 Beispiele - Häufigkeit

Daten sichern

0,3

Ob Datensicherung statt, wird von Rollen entschieden und die Entscheidung fällt in 30% aller Fälle positiv aus.

Daten sichern

Es findet eine Datensicherung statt, aber man findet es nicht sinnvoll oder man will nicht die Häufigkeit angeben, mit der dies der Fall ist.

Daten sichern

...

Man weiß nichts zur Häufigkeit, mit der Datensicherung stattfindet.

Daten sichern

?

Man ist sich unsicher, ob es korrekt ist, dass man die Häufigkeit, mit der Datensicherung stattfindet, nicht spezifiziert.

Daten sichern

???

Man weiß, dass Datensicherung stattfindet, aber nicht, ob es möglich ist, deren Häufigkeit näher zu spezifizieren.

+

Es ist nicht sinnvoll oder unerwünscht anzugeben, unter welcher Bedingung oder bei welchem Ereignis Datensicherung stattfindet, dies wird von einer Rolle entschieden.





## 7.7.2 Vagheit bei Modifikatoren – die wichtigsten Kombinationen

**Abgekürzte  
Darstellung  
der Kombi-  
nationen**

Es ist nicht sinnvoll  
bzw. nicht ge-  
wünscht, Ereig-  
nisse/Bedingungen zu  
spezifizieren

Es ist nicht  
sinnvoll bzw.  
nicht gewünscht,  
die Häufigkeit zu  
spezifizieren

Man weiß nicht  
genug, um die  
Häufigkeit zu  
spezifizieren

Es ist unsicher,  
ob es ange-  
messen ist, die  
Häufigkeit nicht  
zu spezifizieren

Man weiß nicht,  
ob die Häu-  
figkeit spezi-  
fiziert werden  
kann

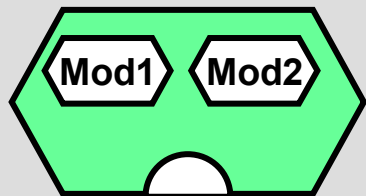
Man weiß nicht  
genug, um Ereig-  
nisse/Bedingungen zu  
spezifizieren

Es ist unsicher, ob es  
angemessen ist, das  
Ereignis nicht zu  
spezifizieren

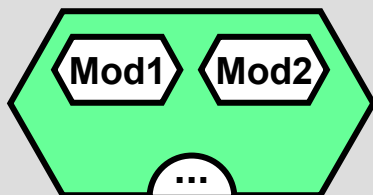
Man weiß nicht, ob  
Ereignisse/Bedin-  
gungen spezifiziert  
werden können



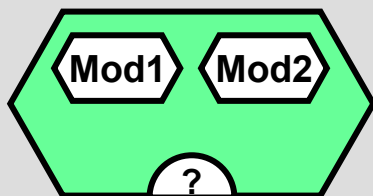

## 7.7.3 Super-Modifikatoren



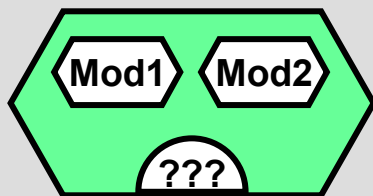
Es ist nicht sinnvoll bzw. nicht gewünscht, neben den gegebenen Sub-Modifikatoren ( hier *Mod1* und *Mod2*) noch weitere zu spezifizieren.



Man weiß nicht, wie man neben den gegebenen Sub-Modifikatoren ( hier *Mod1* und *Mod2*) noch weitere spezifizieren kann.



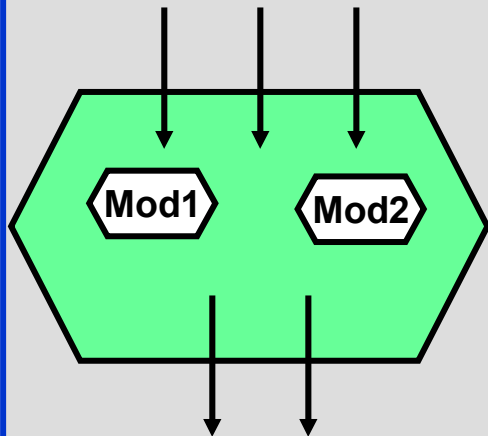
Es ist unsicher, ob die gegebenen Sub-Modifikatoren ( hier *Mod1* und *Mod2*) eine korrekte Struktur der Spezifizierung des Super-Modifikators darstellen.



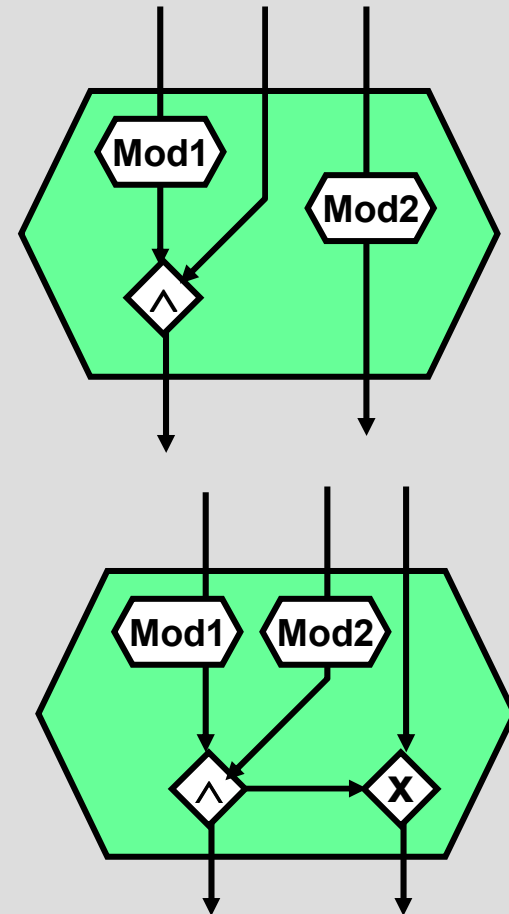
Man weiß nicht, ob neben den gegebenen Sub-Modifikatoren ( hier *Mod1* und *Mod2*) noch weitere spezifiziert werden können bzw. müssen.



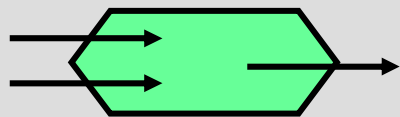
## 7.7.3 Super-Modifikatoren und Relationen mit unspezifizierten Ankerpunkten



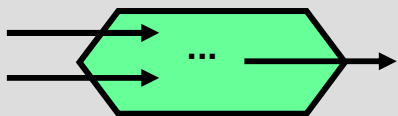
Bei unspezifizierten Ankerpunkten weiß man nicht, wie die Sub-Modifikatoren (hier Mod1 und Mod2) zu den gezeigten Relationen annotiert werden und mit Konnektoren verknüpft werden. Mögliche vollständig spezifizierte Strukturen, die mit der vage spezifizierten Struktur kompatibel sind, sind rechts gezeigt.



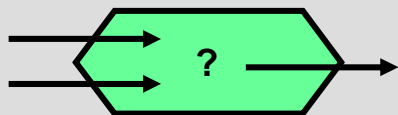
## 7.7.3 Unvollständig spezifizierte Super-Modifikatoren und Relationen mit unspezifizierten Ankerpunkten



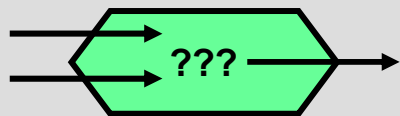
Es ist nicht sinnvoll bzw. nicht gewünscht, Sub-Modifikatoren zu spezifizieren



Man weiß nicht genug, um Sub-Modifikatoren zu spezifizieren



Es ist unsicher, ob es angemessen ist, Sub-Modifikatoren nicht zu spezifizieren

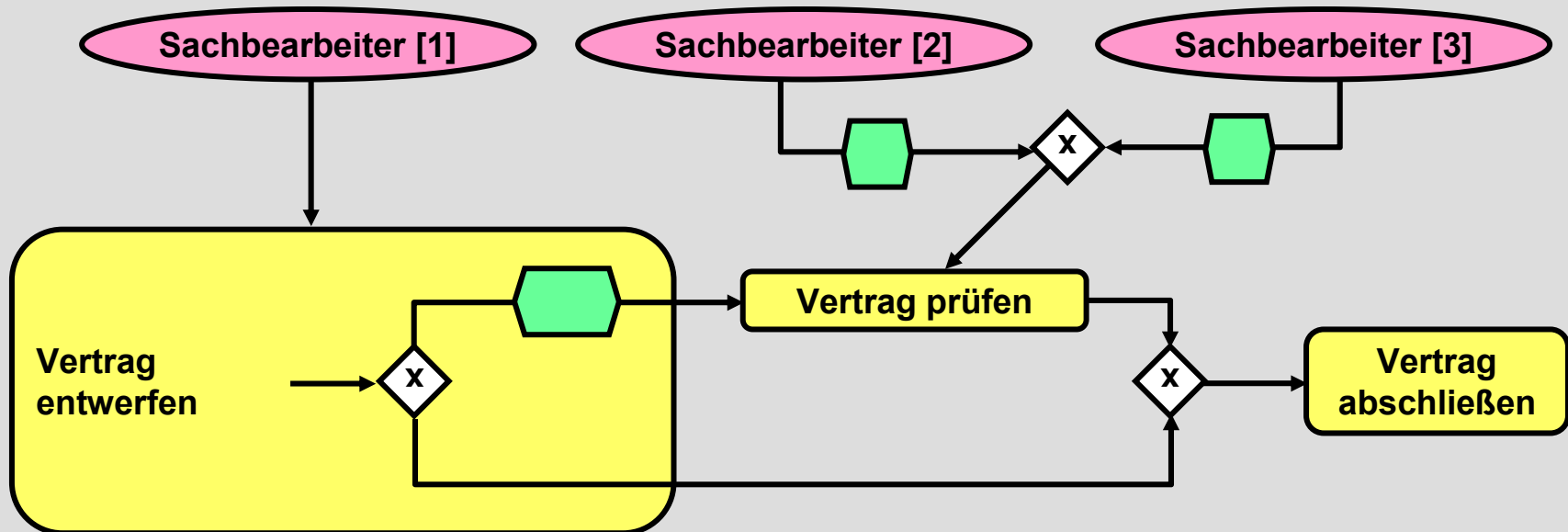


Man weiß nicht, ob Sub-Modifikatoren spezifiziert werden können oder müssen.

Wenn Modifikatoren nicht zu Relationen annotiert sind, sondern diese in den Modifikator hineinzeigen, handelt es sich um einen Super-Modifikator. Die Vagheits-Symbole sind dann anders zu interpretieren.



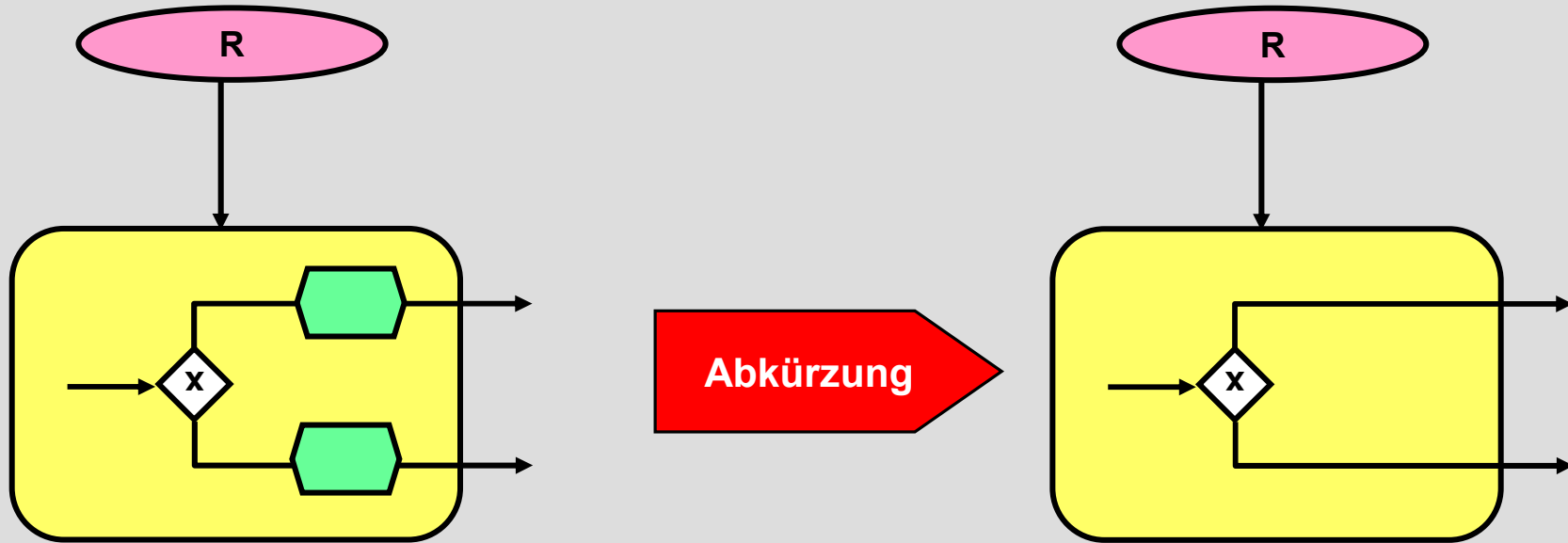
## 7.7.4 Beispiel zu vage spezifizierten Modifikatoren - Entscheidungsfreiheit



Im Unterschied zu dem [Beispiel bei der XOR-Verknüpfung von Aktivitäten](#) entscheidet hier der Sachbearbeiter frei, ob der Vertrag noch von anderen geprüft wird oder nicht. Dies wird durch den unspezifizierte Modifikator ausgedrückt. Die Entscheidung wird im Rahmen der Aktivität „Vertrag entwerfen“ getroffen. Ebenso entscheiden die Sachbearbeiter [2] und [3], letztlich in einem Aushandlungsprozess, wer von Ihnen die Prüfung übernimmt, wobei hier keine Aktivität gezeigt ist, die das Entscheiden beinhaltet – diese Aktivität muss nicht immer gezeigt werden.

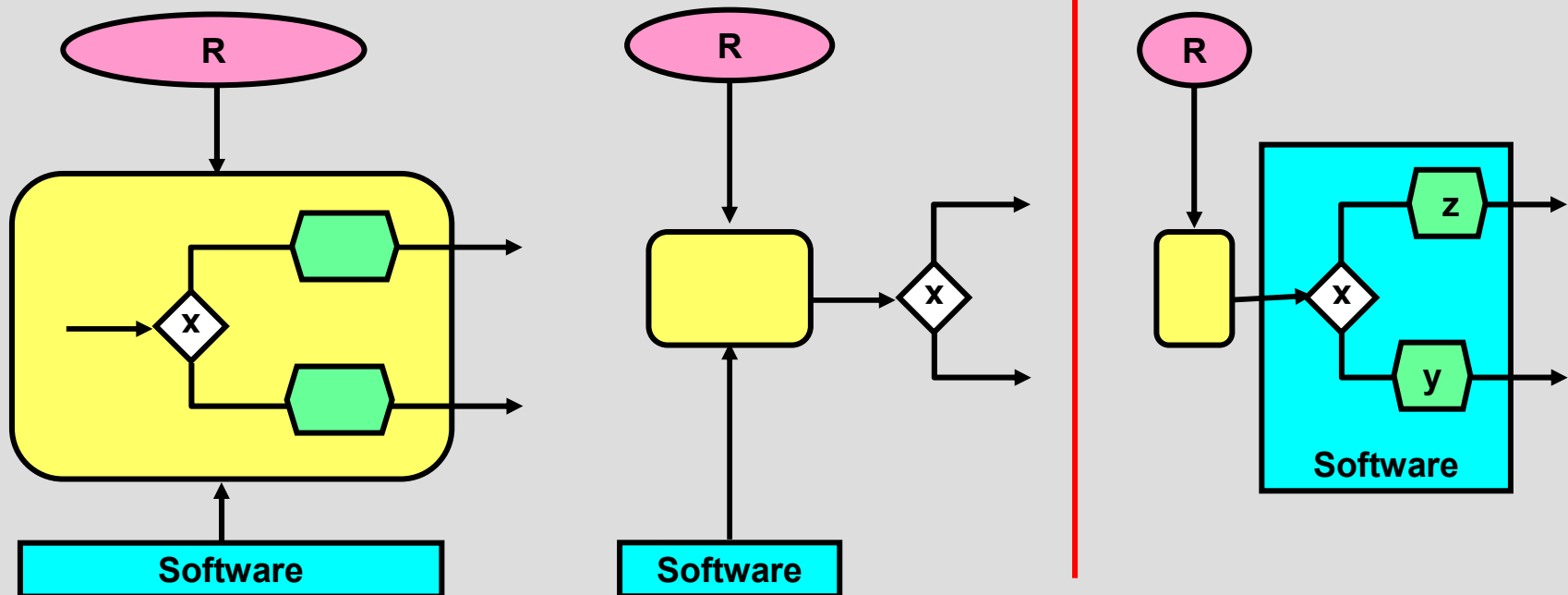


## 7.7.4 Abkürzung zu Entscheidungsfreiheit



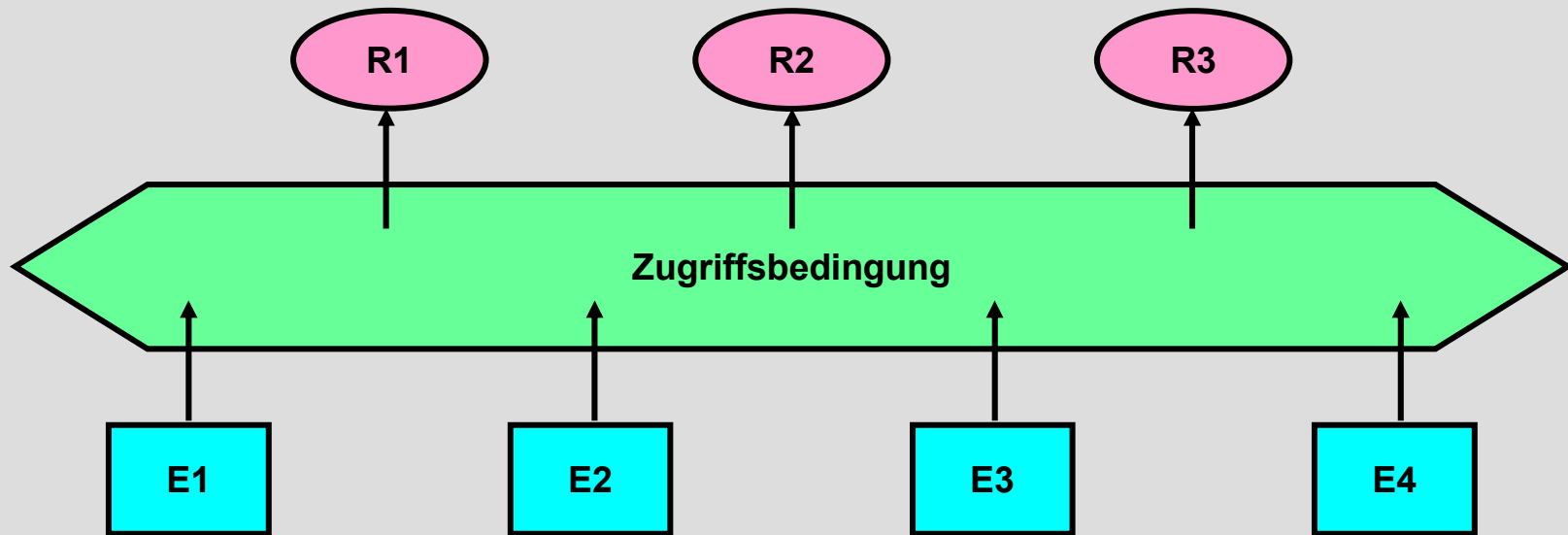
Falls bei einer ODER- oder XOR-Verknüpfung die Modifikatoren fehlen, die die Auswahl der Relationen entscheiden, dann ist das gleichbedeutend mit der Annotation unspezifizierter Modifikatoren, wie es im linken Diagramm gezeigt wird.

## 7.7.4 Erläuterung zu Entscheidungsfreiheit



In beiden linken Fällen wird zum Ausdruck gebracht, dass R die Entscheidung über die Auswahl der XOR-verknüpften Relationen trifft, egal ob die Aktivität, die die Entscheidung beinhaltet, modelliert wird oder nicht. Es ist in dieser Konstellation nie das technische System, das auswählt. Dies müsste durch explizite Modifikatoren, die auf das techn. System bezogen sind, beschrieben werden, wie im rechten Fall gezeigt.

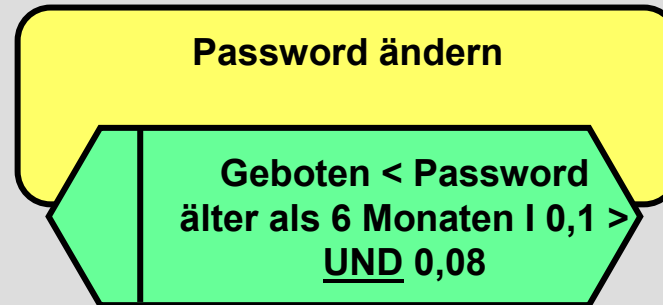
## 7.7.4 Beispiel zu vage spezifizierten Modifikatoren II



Man drückt aus, dass die Zugriffsrechte der Rollen R1 bis R3 auf die Entitäten E1 bis E4 von verschiedenen Bedingungen abhängen und variieren. Man will diese aber nicht genauer angeben. Unter Umständen variieren die Bedingungen mit jeder Instantiierung der gezeigten Elemente.



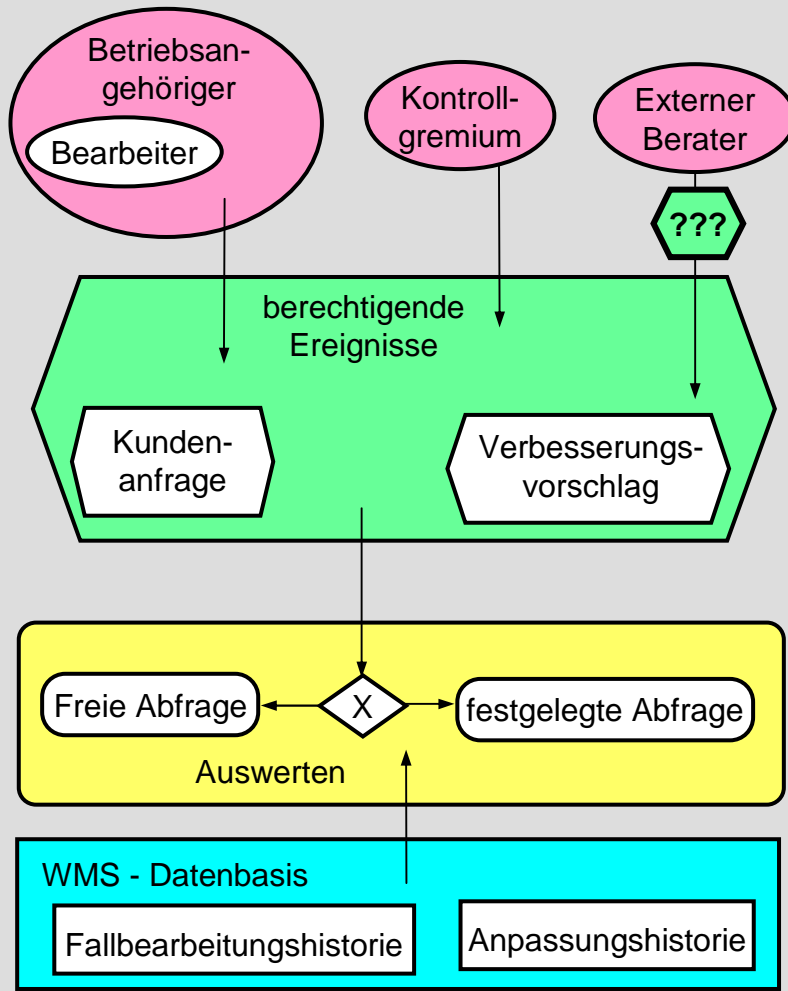
## 7.7.4 Beispiel zu vage spezifizierten Modifikatoren III



Hier wird das Beispiel aus [5.1.](#) umgewandelt. Das Ereignis bleibt unspezifiziert. Es wird ein neuer Modifikator eingeführt und in spitzen Klammern der vorschreibenden Häufigkeitsangabe „geboten“ zugeordnet. So kann ausgedrückt werden, dass die Bedingung für dieses Gebot in 10% aller Fälle gegeben ist, die Änderung aber nur in 8% der Fälle erfolgt. Das heißt, dass das Gebot in 2% aller Fälle missachtet wird.



## 7.7.5 Aufgaben zu Vagheit bei Modifikatoren



1. Interpretieren Sie das links stehende Diagramm

Stellen sie folgende Sachverhalte dar:

2. In eine Büro arbeiten zwei Beschäftigte. Wenn das Telefon läutet entscheiden sie frei, ob man und wer abhebt. Im Durchschnitt wird jeder zweite Anruf angenommen und beide heben gleich viel ab.
3. Vier Arbeitsschritte werden im Rahmen einer Aufgabe hintereinander erledigt. Wenn für einen bereits abgeschlossenen Schritt Verbesserungsmöglichkeiten erkannt werden, können nicht näher spezifizierte Teile dieses Schrittes wiederholt werden.
4. Ein Abteilungsleiter soll auf die Daten seiner Mitarbeiter zugreifen können, wenn diese krank sind oder unter anderen Bedingungen. Diese sind unklar, ebenso wie ein solcher Zugriff erfolgt.

Aufg.

Aufgabe

## 7.8 Vage Spezifizierung von Konnektoren

Vagheit kann auch auf Konnektoren angewendet werden.



Insbesondere muss die Art der logischen Verknüpfung nicht immer festgelegt werden – somit kann sie von Fall zu Fall frei entschieden werden. Zu diesem Zweck wird eine leere Raute verwendet. Auch hier kann wieder die Entscheidung der beteiligten Rollen zum Zuge kommen.

Außerdem kann Unsicherheit bzgl. der Art und der Festlegung eines Konnektors ausgedrückt werden.

7.8.1

**Gewollte Vagheit**

7.8.2

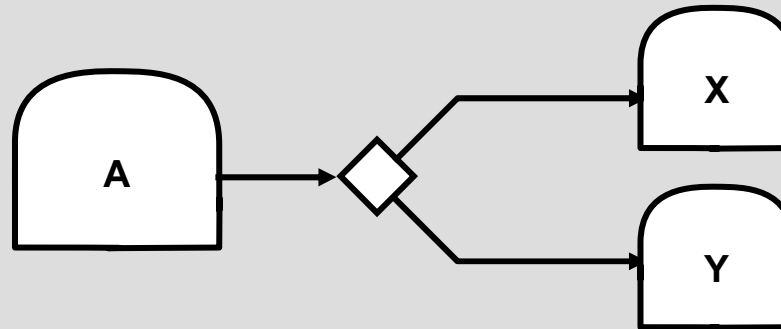
**Gegebene Vagheit**

7.8.3

**Aufgaben**



## 7.8.1 Vage Spezifizierung von Konnektoren



Wenn ein Konnektor leer ist, dann ist dies analog zur Definition des leeren Halbkreises zu verstehen. Das bedeutet, dass der Modellierer nicht festlegen will, wie die logische Verknüpfung der Relationen zwischen A und X sowie zwischen A und Y aussieht, oder er hält diese Festlegung nicht für sinnvoll.

## 7.8.1 Abkürzungsregel für den nicht spezifizierten Konnektor



Wenn zwei Relationen gleichen Typs (hier *ausführen*) mit nur einem Element verbunden sind, fragt es sich, wie diese logisch mit einander verbunden sind. Führt R1 beide Aktivitäten aus oder entweder A1 oder A2 oder mindestens eine von beiden? Will man dies nicht festlegen, dann kann man den unspezifizierten Konnektor verwenden oder - zwecks Ankürzung - die beiden Relationen direkt zu dem Element führen, mit dem sie beide irgendwie in Verbindung stehen – hier also R1. Wo immer eine Struktur auftritt, bei der zwei oder mehrere Relationen gleichen Typs mit genau einem Element verbunden sind, bedeutet dies nicht, dass immer alle diese Relationen gemeinsam instantiiert werden – also dass R1 immer A1 und A2 ausführt, da der Konnektor ja auch potentiell anders spezifizierbar ist.

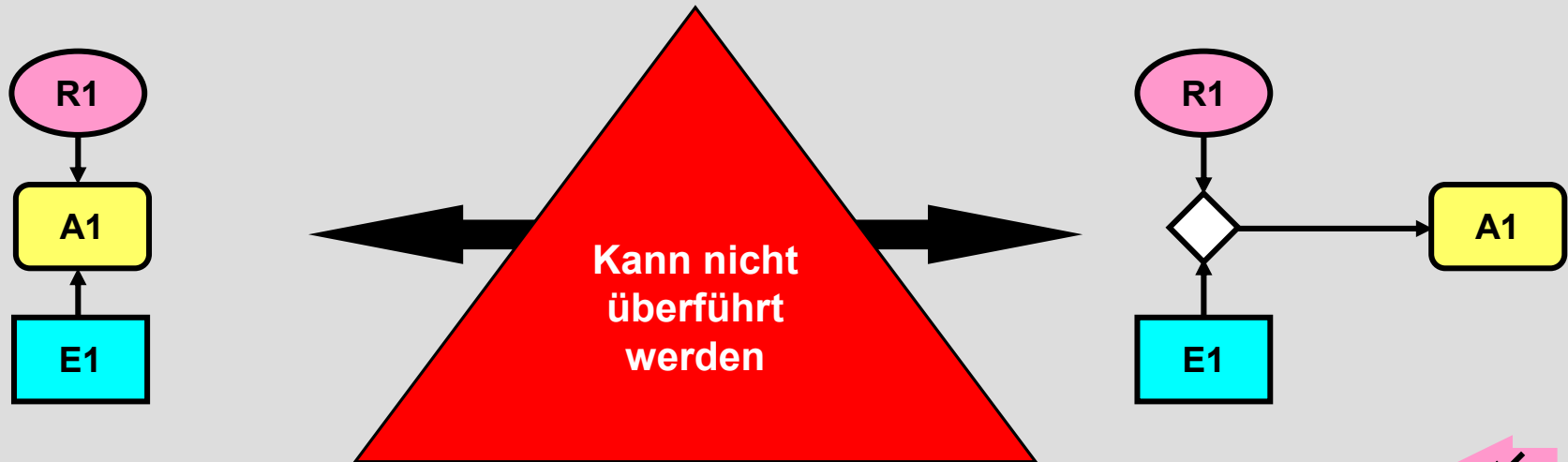
Dies Einschränkung gilt nicht für Relationen unterschiedlichen Typs!

Dies ist insbesondere auch für Relationen mit einem nicht verankerten Start- oder Endpunkte zu beachten (s. [Kap. 7.5.1](#))

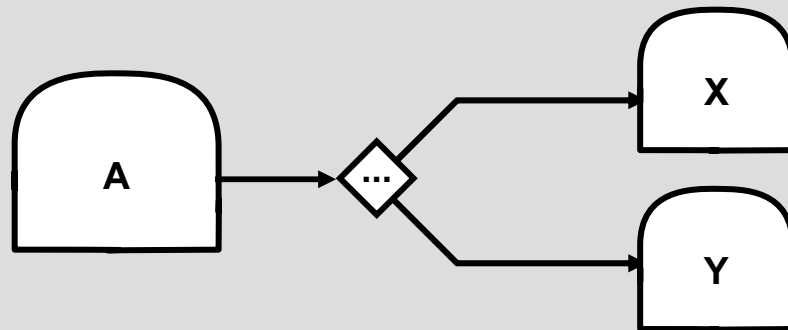


## 7.8.1 Abkürzung bei Relationen unterschiedlichen Typs nicht unterstellbar

Wenn zwei Relationen unterschiedlichen Typs mit einem Element (hier A1) verbunden sind, dann kann nicht die Abkürzungsregel unterstellt werden. Vielmehr bedeutet dies, dass stets beide gemeinsam mit dem Element verbunden sind. A1 wird also in dem linken Diagramm von R1 ausgeführt UND benutzt E1. Bei dem rechten Diagramm liegt dagegen eine unspezifizierte logische Verknüpfung vor, die eventuell auch als XOR spezifizierbar ist. Es kann dann also sein, dass A1 entweder E1 nutzt und nicht durch R1 ausgeführt wird oder R1 A1 ausführt und E1 nicht genutzt wird.



## 7.8.2 Vage Spezifizierung von Konnektoren (...)

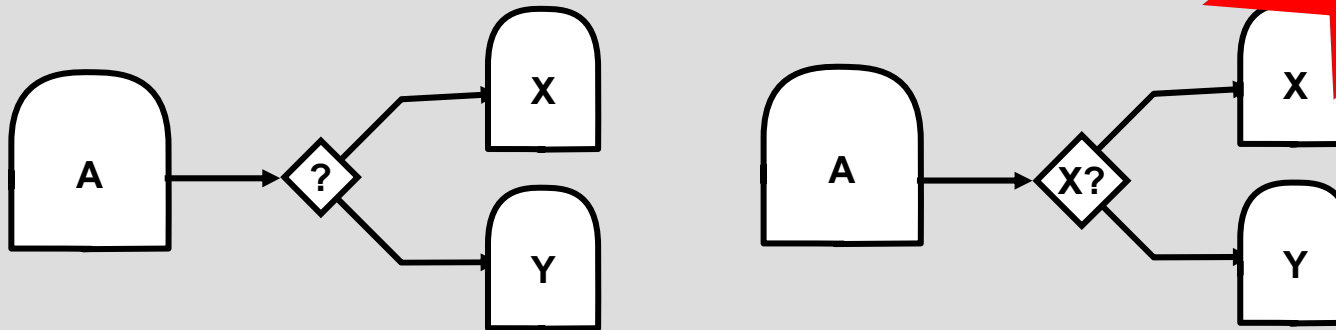


**Selten  
benötigt**

Man weiß nicht, wie man den Konnektor, der die beiden Relationen verbindet, spezifizieren soll.



## 7.8.2 Vage Spezifizierung von Konnektoren (2)



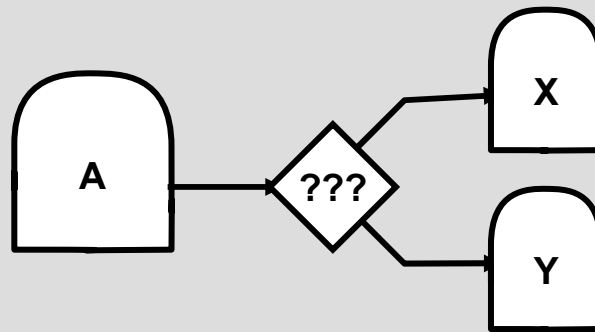
**Selten  
benötigt**

Im linken Fall ist man sich unsicher, ob es angemessen ist, den Konnektor nicht zu spezifizieren. Im rechten Fall wird bezweifelt, ob die spezifizierte logische Verknüpfung (hier XOR) korrekt ist.





## 7.8.2 Vage Spezifizierung von Konnektoren (???)



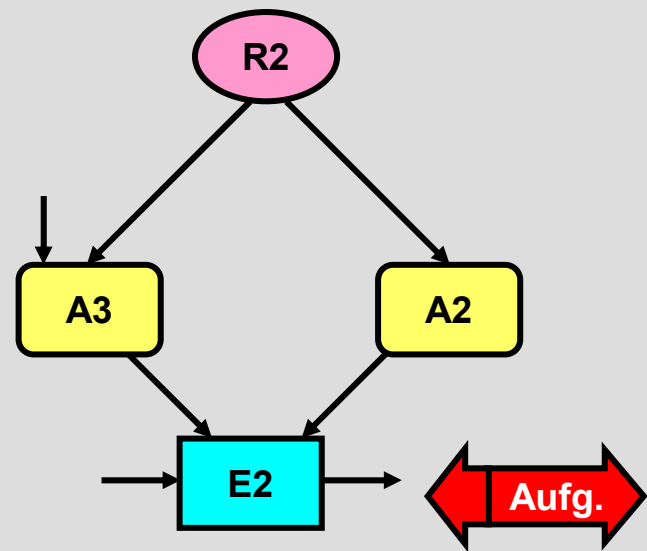
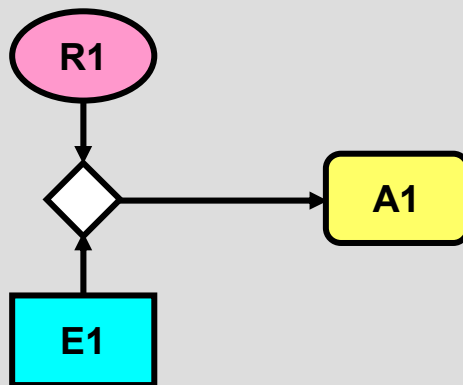
**Selten  
benötigt**

Es ist unsicher, ob es eine Spezifizierung des Konnektors gibt, die für jede Instantiierung gleich ist.



## 7.8.3 Aufgaben zu vagen Konnektoren

1. Finden Sie eine sinnvolle Spezifizierung für R1, A1, E1
2. Stellen Sie die unspezifizierten Konnektoren dar, die im rechten Diagramm als Abkürzungen enthalten sind.
3. Stellen Sie dar, dass A und B eine Aktivität immer zusammen ausführen und dass es fraglich ist, ob dieser Kooperationszwang sinnvoll ist. Stellen Sie im zweiten Schritt eine Lösung dar, wo Bs Mitwirkung optional ist, während A immer an der Aktivität mitwirken muss – außerdem entscheiden beide zusammen, ob B mitmacht oder nicht. Ändern Sie in einem dritten Schritt das Diagramm so ab, dass B entscheiden darf, ob er mitmacht oder nicht und die Korrektheit dieser Entscheidungsfreiheit bezweifelt wird.



## 7.9 Vage Spezifizierung von Attributen

Prinzipiell könne an beliebigen Stellen bei der Angabe von Attributnamen oder Werten Vagheitssymbole annotiert werden.

Sie sollten immer in runde Klammern gesetzt werden.

Entsprechend dem runden, leeren Halbkreis werden runde Klammern verwendet, die ein Blank einklammern.

Es sind Abkürzungen möglich.

Weitere Vagheit wird ausgedrückt, indem Modifikatoren mit vagen Angaben den Attributwerten zugeordnet werden können.

**Zum Beispiel: Löschfrist: 6 (?) Monate< ? l...>**

7.9.1

**Vage Spezifizierung von Attributen**

7.9.2

**Attribute und vage Modifikatoren**

7.9.3

**Aufgaben**



## 7.9.1 Vage Spezifizierung - Beispiele

Häufigkeit: oft (?)

Es ist unsicher, ob das so attributierte Element wirklich „oft“ auftritt

Dauer: 10 min(?)

Es ist unsicher, ob „min“ die korrekte Maßeinheit repräsentieren.

Größe: [Höhe: 1,5 m; Tiefe: 50 cm; Breite(?): 2 m; (???)]

Es ist unsicher, ob „Breite“ der korrekte Attributname ist (z.B. statt Länge) und ob nicht noch andere Angaben zur Größe fehlen (etwa ein Zeitbezug)

Verwandte: {Geschwister, Kinder, Eltern, Großeltern, ( )}

Der Modellierer will weitere Arten von Verwandten nicht nennen.

Betriebliche Kennzahlen: [Produktivität; Durchlaufzeit; (...)]

Man weiß, dass es weitere Arten von Kennzahlen gibt, weiß aber nicht welche.

Zugriffsberechtigte: ( ) (?)

Es ist unsicher, ob es korrekt ist, die Zugriffsberechtigten nicht zu spezifizieren



## 7.9.1 Unsicherheit bei Attributen - Abkürzung

Verwandte: {Geschwister, Kinder,  
Eltern, Großeltern,( )}

Ab-  
Kür-  
zung

Verwandte: {Geschwister, Kinder,  
Eltern, Großeltern,}

Zugriffsberechtigte: ( )(?)

Ab-  
Kür-  
zung

Zugriffsberechtigte: (?)

Adresse;  
Empfänger:{Kunde, Personalabtl., ( )};  
Erlaubte Datenfelder: [ Straße, Ort, ( ) ]  
Speicherdauer: ( ); ( )

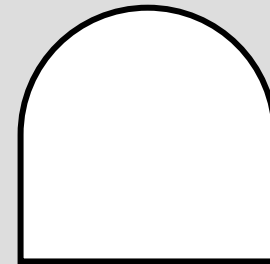
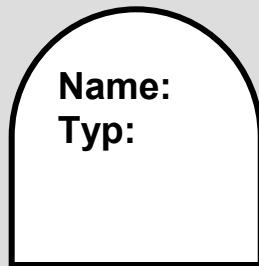
Ab-  
Kür-  
zung

Adresse;  
Empfänger:{Kunde, Personalabtl., }  
Erlaubte Datenfelder: [ Straße, Ort,]  
Speicherdauer: ;

Wenn aufgrund überflüssig erscheinender Satzzeichen deutlich wird, dass eine Angabe nicht spezifiziert ist, dann können die leeren Einklammerungen weggelassen werden.



## 7.9.1 Fehlende Spezifizierung des Attributes „Typ“ - Abkürzung



Wenn es nicht sinnvoll oder gewünscht ist, den Typ eines Basis-Elementes zu spezifizieren, dann kann man das Meta-Element verwenden, ohne das Typ-Attribut darzustellen. Dasselbe gilt für das Namensattribut (auch dann, wenn der Typ spezifiziert wird).



## 7.9.2 Attribute und vage Modifikatoren

Zuverlässigkeit: hoch<...>

Wir wissen nicht, unter welcher Bedingung die Zuverlässigkeit hoch ist.

Zuverlässigkeit<???>:

Wir sind unsicher, ob es Bedingungen gibt, unter denen das Attribut Zuverlässigkeit instantiiert werden sollte, also ob für das betrachtete Element das Attribut „Zuverlässigkeit“ sinnvoll ist.

Löschfrist: 6 < l...> Monate

Wir wissen nicht, wie wahrscheinlich es ist oder wie oft es vorkommt, dass die Löschfrist 6 Monate beträgt.

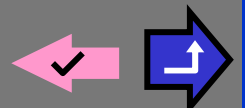
Löschfrist: 6 Monate< l...>

Wir wissen nicht, wie wahrscheinlich es ist oder wie oft es vorkommt, dass die Löschfrist in „Monaten“ spezifiziert ist.

Löschfrist<?>:

Wir sind uns unsicher, ob es korrekt ist, dass wir nicht näher spezifizieren, unter welchen Bedingungen das Attribut „Löschfrist“ zu instantiiieren ist.

**Vergl. dagegen „Löschfrist: <?>“**



## 7.9.3 Aufgaben zu Vagheitsangaben bei Attributen

1. Erläutern Sie den Unterschied zwischen folgenden Angaben und geben Sie Beispiele, unter welchen Umständen solche Vagheitsanmerkungen sinnvoll sind, wo könnte das Attribut eingebettet sein?  
Sicherheitsabstand (?): 10 m  
Sicherheitsabstand: 10 (?) m  
Sicherheitsabstand: 10 m (?)  
Sicherheitsabstand <?>: 10 m  
Sicherheitsabstand: {10 <?> m; (?)}
2. Wie stellt man dar, dass man sich unsicher ist, ob der Typ eines Basis-Elementes richtig spezifiziert wurde?
3. Stellen sie einen Fall dar, bei dem ein zusätzliches Back-up durchgeführt wird, wenn mehr als 10000 lines-of-code neu produziert wurden. Drücken Sie aus, dass man sowohl den Sinn dieses Attributes als auch den Wert 10000 anzweifelt.

**Aufg.**



## 7.10 Kombination von Vagheitsangaben

Selten  
benötigt



Es macht keinen Sinn, schwarze oder leere Flächen mit den untenstehenden Symbolen zu kombinieren. Vielmehr können beim Klicken auf die schwarze Fläche solche Vagheitsangaben erscheinen. Die weiße Fläche signalisiert, dass man sich beim modellieren nicht um die Spezifizierung der Vagheit kümmern will.

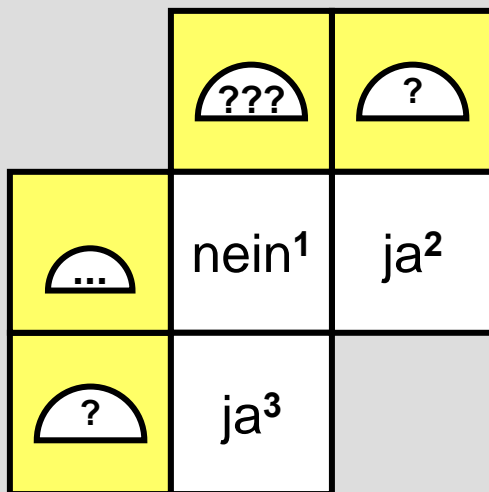
Die aus dem gegebenen Kenntnisstand resultierenden Vagheitssymbole sind eingeschränkt kombinierbar. Hier sind Kombinationsmöglichkeiten zu diskutieren.



Das + Zeichen signalisiert, dass der Modellierer über zusätzliche, im Modell nicht gezeigte Informationen verfügt. Dieses Zeichen ist mit allen Symbolen der aus dem Kenntnisstand resultierenden Vagheit kombinierbar, sowie mit deren Kombinationen.



## 7.10 Kombination von Indikatoren gegebener Vagheit



Da die Reihenfolge der kombinierten Vagheitssymbole sowie deren Doppelung nicht sinnvoll ist, haben inhaltlich gesehen nur drei Kombinationsmöglichkeiten Relevanz:

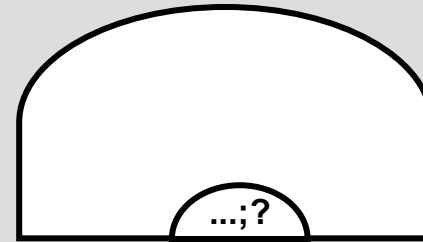
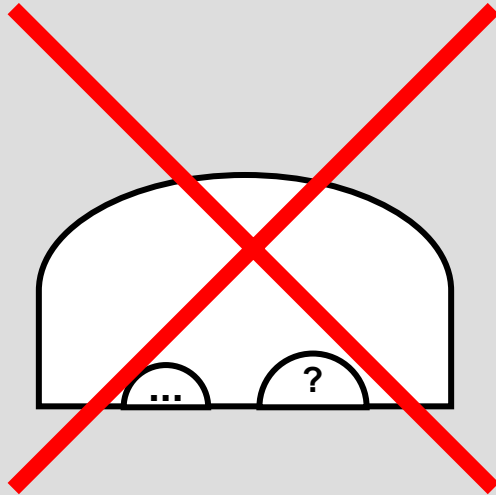
Ad1) Da drei Punkte symbolisieren, dass mit Sicherheit etwas fehlt, schließt sich diese Kombination aus. Daher gibt es auch keine sinnvolle Dreierkombination.

Ad2) Die gezeigte Spezifizierung ist sohl auf Grund von Gegebenheiten unvollständig und eventuell auch nicht angemessen.

Ad3) Die gezeigte Spezifizierung von Sub-Elementen kann bzgl. Korrektheit der vorgenommenen Spezifizierung und bzgl. Ihrer Erweiterbarkeit angezweifelt werden.

Die beiden verbleibenden Kombinationen können zusätzlich mit einem „+“ versehen werden.

## 7.10 Darstellung kombinierter Vagheitsangaben



**Es sollen nie zwei Halbkreis zu einem Basis-Element hinzugefügt werden. Vielmehr sind die Vagheitsangaben in einem Halbkreis zu kombinieren.**



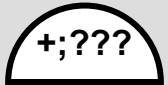
## 7.10 Abkürzung kombinierter Vagheitsangaben



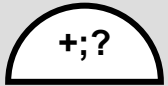
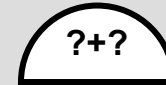
**Abkürzung**



Abkürzung nicht sinnvoll



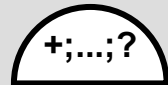
**Abkürzung**



Abkürzung nicht sinnvoll



**Abkürzung**

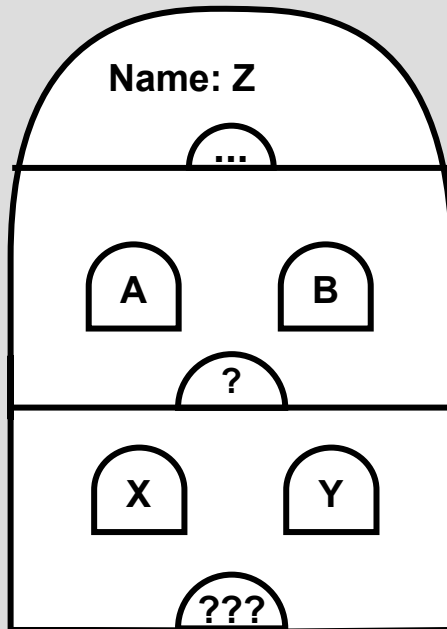


Abkürzung nicht sinnvoll

Zum Teil ist es mnemotechnisch möglich, sinnvolle Abkürzungen einzuführen.



## 7.10 Segmentlinien und Vagheitsangaben



Segmentlinien können Vagheitssymbole tragen, die den jeweils über ihnen dargestellten Sub-Elementen oder Attributen zugeordnet sind. Die „???“ beziehen sich nur noch auf X und Y. Vagheitsangaben, die sich auf das gesamte Super-Element „Z“ beziehen, müssen in jedem Segment wiederholt und entsprechend der dargestellten Regeln kombiniert werden.

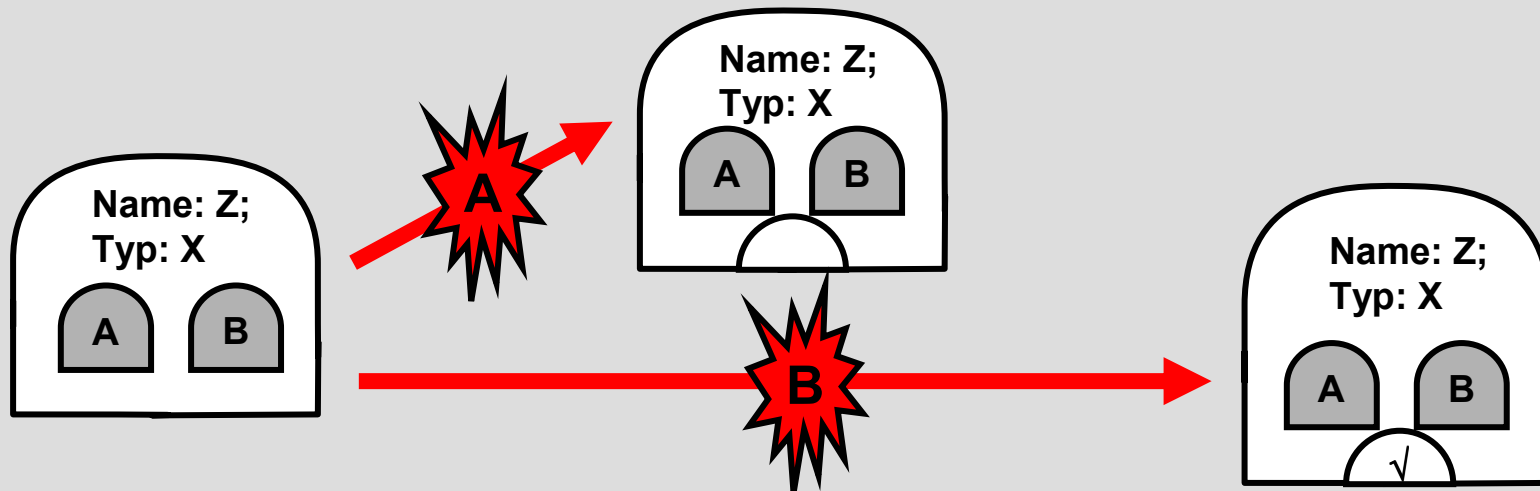
Aufgabe

## 7.10 Aufgaben zur Kombination von Vagheitsangaben

1. Finden Sie Beispiele für die bei den Abkürzungen gezeigten Kombinationen
2. Erläutern Sie an einem Beispiel, warum die Kombination (?,???) bei Elementen, die außer ihrem Namen keine Spezifizierung enthalten, nicht sehr sinnvoll ist.
3. Finden Sie eine sinnvolle Spezifizierung für Z, A, B, X, Y bei den Erläuterungen zu Segmentlinien und Vagheit.



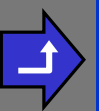
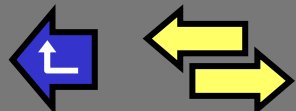
## 7.11 Ausdrückliche Vollständigkeit



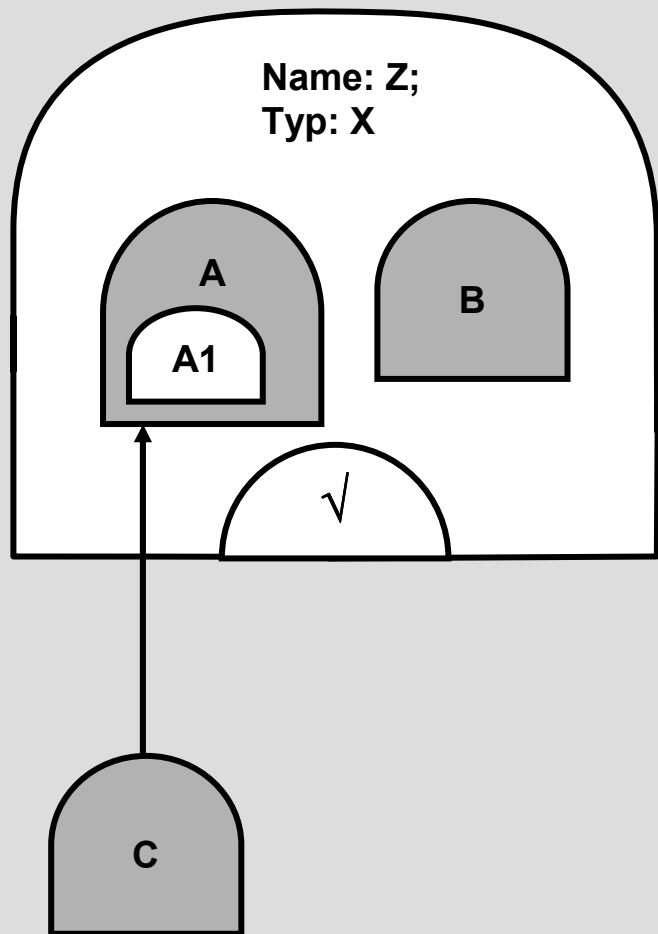
Die links stehende Darstellung kann zu verschiedenen Überlegungen Anlass geben:

Prinzipiell besagt sie, dass Z vollständig spezifiziert ist, da keine Unvollständigkeitsangabe vorliegt. Aber es könnte auch sein, dass diese vergessen wurde und dass die Variante A der Darstellung korrekt ist.

Um solche Missverständnisse zu vermeiden, kann man durch einen Haken explizit machen, dass man die Vollständigkeit der Spezifizierung geprüft hat (Variante B).



## 7.11 Ausdrückliche Vollständigkeit



**Die Bedeutung dieses Symbols besagt:**

**Es ist bzgl. des Zweckes der Modellierung nicht sinnvoll oder gewünscht, mehr als die gezeigten Sub-Elemente zu spezifizieren.**

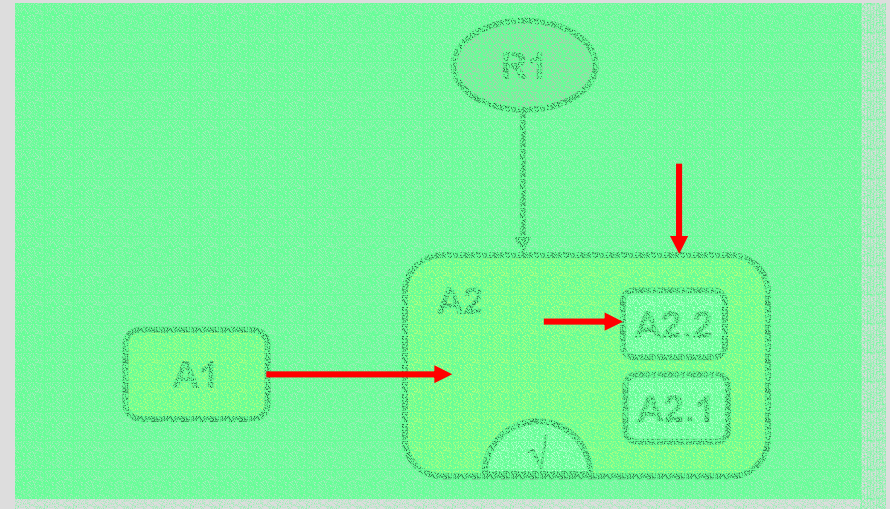
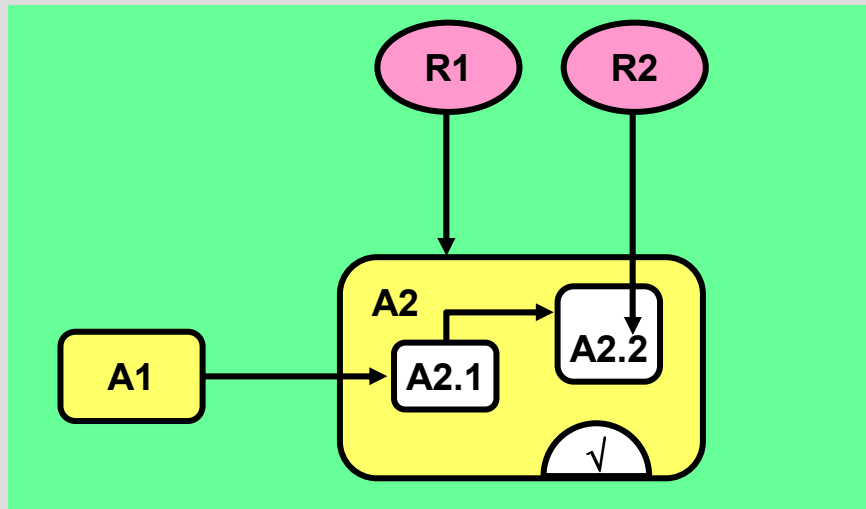
**Es kann nicht mehr als die gezeigten Relationen geben, die einen Bezug zu Z haben, von denen also mindestens ein Ankerpunkt mit dem Rand oder mit einem Sub-Element von Z verbunden ist.**

**Falls es solche Relationen gibt und ihr Typ nicht gesondert spezifiziert ist, dann bedeutet der Haken, dass die Angemessenheit der Standardbedeutung der Relation bewusst geprüft wurde.**

**Der Haken bedeutet nicht, dass für A und B geprüft wurde, ob sie vollständig spezifiziert sind (A könnte also neben A1 noch weitere Sub-Elemente enthalten, falls das Vagheitssymbol bei A vergessen wurde).**



## 7.11 Ausdrückliche Vollständigkeit



Bei Relationen, die mit einem durch „✓“ gekennzeichneten Element (hier A2) in Beziehung stehen (von denen also mindestens ein Ankerpunkt am Rand oder innerhalb von A2 liegt), müssen die Ankerpunkte, die innerhalb des Elementes liegen, zu einem Sub-Element verbunden sein und die Ankerpunkte, die außerhalb des Elementes liegen, mit einem anderen Element verbunden sein .  
Die rechte Beispiel-Abbildung ist also aufgrund der roten Pfeile inkorrekt.



## 7.11 Aufgaben zu ausdrücklicher Vollständigkeit

1. Wie drücken Sie bei einer Aktivität aus, dass die Attribute vollständig spezifiziert sind, nicht aber die Sub-Aktivitäten.
2. Sie wollen ausdrücken, dass alle mit einer Entität in Beziehung stehenden Relationen vollständig dargestellt sind, nicht aber die Sub-Entitäten – welchen Trick wenden sie an.
3. Suchen Sie drei Beispiel-Abbildungen in diesem Tutor, bei denen sie es für sinnvoll erachten, Vollständigkeitssymbole einzutragen.



## 8) Ein- und Ausblenden

**Schwarze Flächen, seien es Halbkreise, verdickte Pfeile, schwarze Kästchen bei Attributen oder ähnliches, verweisen auf Detailinformationen, die ausgeblendet sind oder an anderen Stellen dargestellt sind.**

**Ein eigens für SeeMe konstruierter Editor stellt die durch Ein- und Ausblenden erzielten Effekte automatisch dar. Optimales Ein- bzw. Ausblenden vergrößert bzw. verkleinert die betroffenen Elemente.**

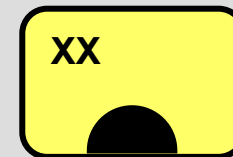
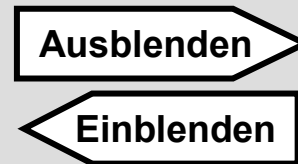
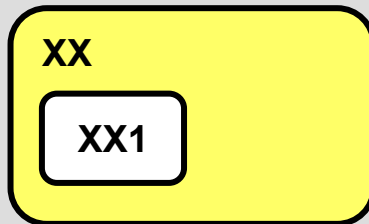
**Man kann das Ein- und Ausblenden bei SeeMe-Diagramme auch mit anderen Software-Systemen, die aktivierbare Hyperlinks anbieten, realisieren.**

**Letztlich lassen sich schwarze Flächen auch bei papier-basierten Diagrammen verwenden – sie sollten dann einen Hinweis enthalten, wo die weitergehende Information zu finden ist.**

**Im folgenden werden Beispiele gegeben, die die Wirkung des Ein- und Ausblendens und die angemessene Darstellungsform verdeutlichen.**



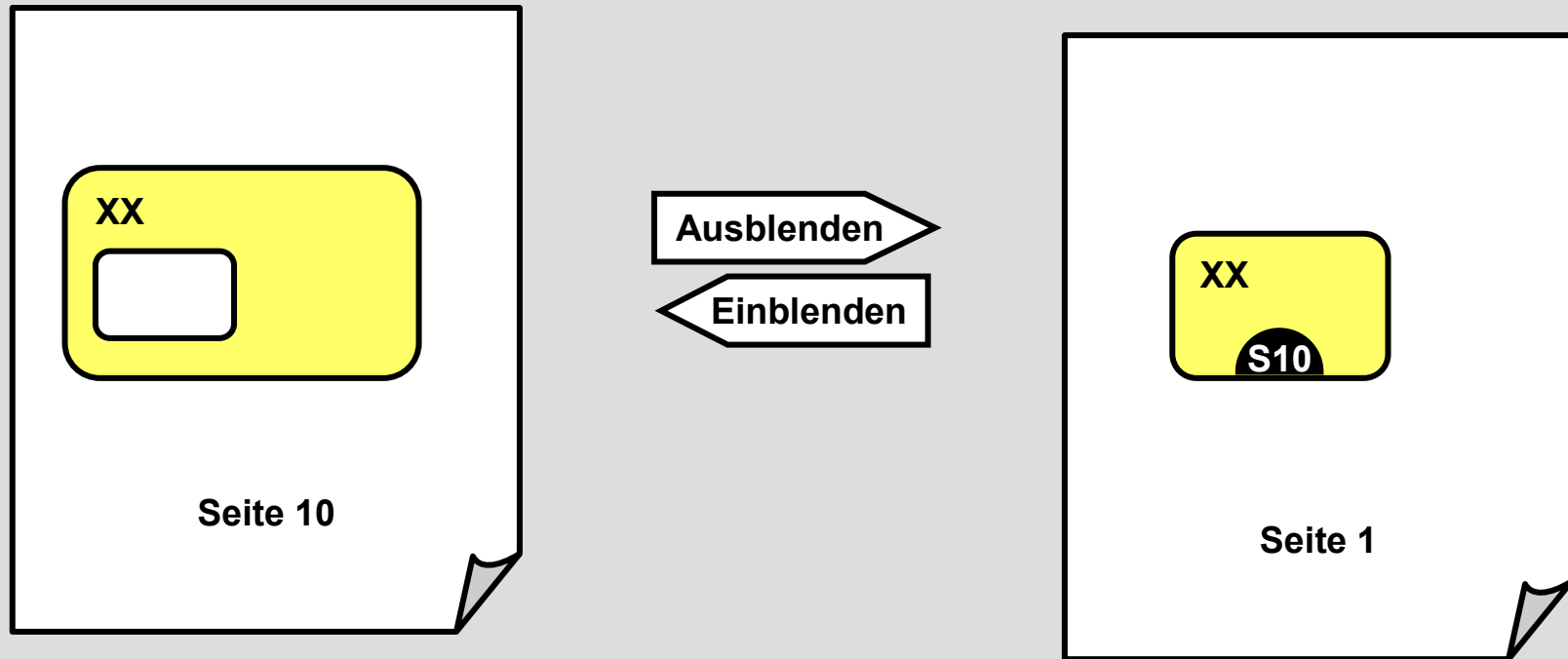
## 8) Ein- und Ausblenden – Sub-Elemente



Wenn ein Sub-Element ausgeblendet wird, hier die Sub-Aktivität XX1, dann wird ein schwarzer Halbkreis gezeigt, über den das Wiedereinblenden aktiviert werden kann.

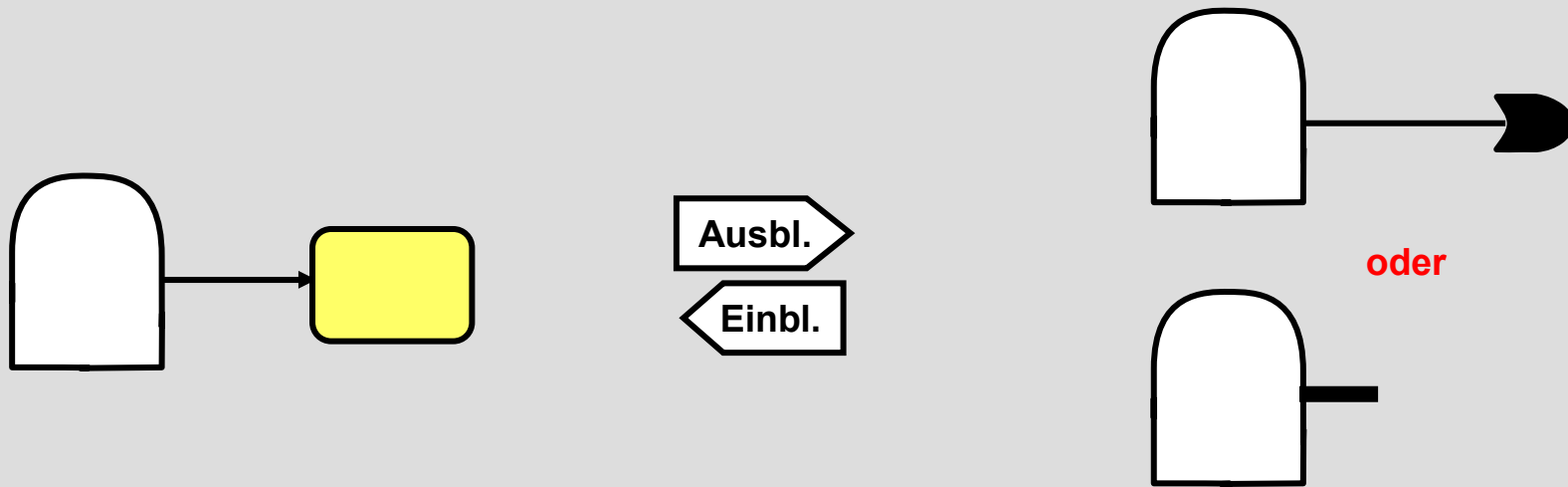


## 8) Ein- und Ausblenden – mit Verweisen



Schwarze Flächen, seien es Halbkreise, verdickte Pfeile, schwarze Kästchen bei Attributen oder ähnliches, können bei papier-basierten Versionen Hinweise enthalten, wo sich mehr Informationen befinden.

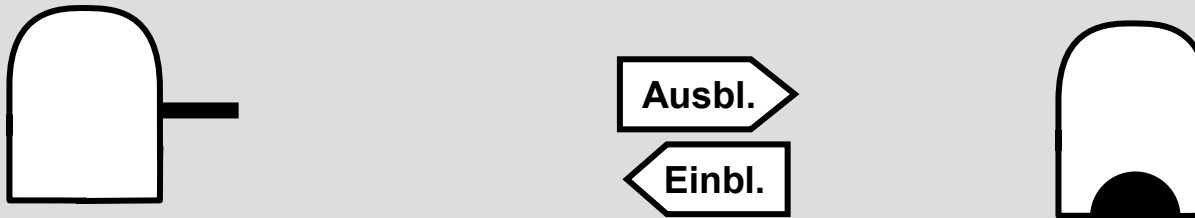
## 8) Ein- und Ausblenden – Relationen auf Basis-Elementen



Hier wird der Effekt gezeigt, der entsteht, wenn die Aktivität ausgeblendet wird. Sie ist durch ein Klicken auf die schwarzen Flächen wieder einblendbar.



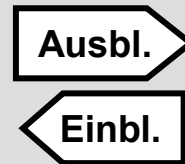
## 8) Ein- und Ausblenden - Relationen



Ein Pfeil kann auch völlig ausgeblendet werden – dies wird durch einen schwarzen Halbkreis angedeutet in den Sub-Elementen angedeutet, mit denen er verbunden ist.. Wenn er aktiviert wird, wird der Pfeil wieder eingeblendet. Prinzipiell werden bei einer automatischen Unterstützung des Ein-Ausblendens immer die Elemente zuerst wieder eingeblendet, die als letztes ausgeblendet wurden – also auch u.U. Sub-Elemente.



## 8) Ein- und Ausblenden – Basis-Elemente

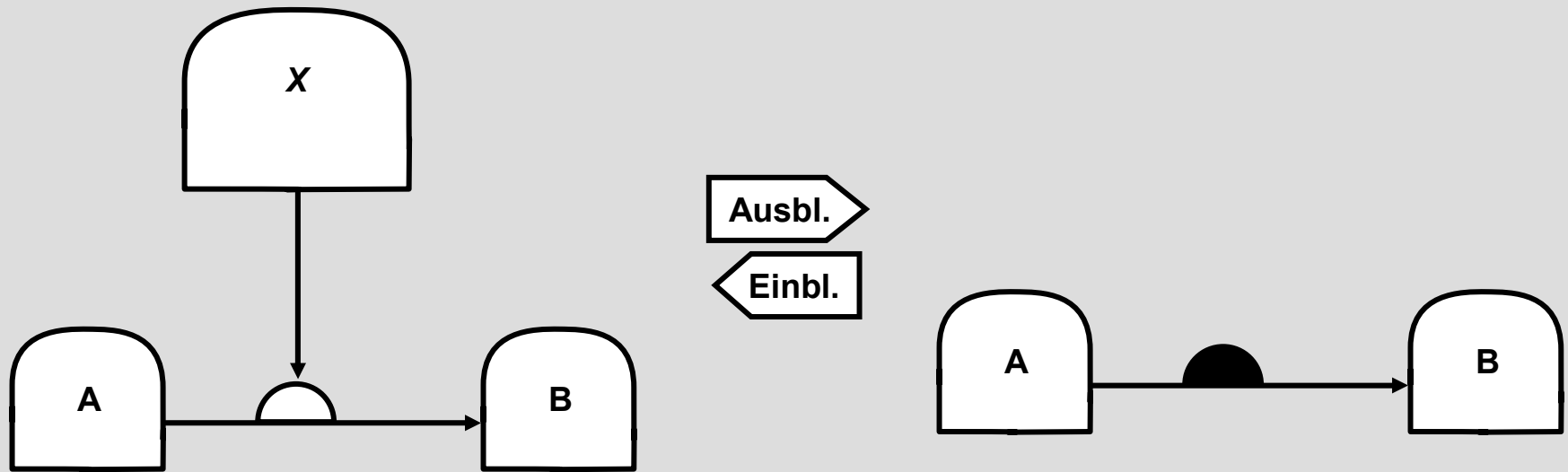


Gegebenenfalls können Basis-Elemente ihren Inhalt ausblenden, indem ihre Größe reduziert wird und sie schwarz dargestellt werden. Beim Klicken auf die schwarze Fläche kehrt der Inhalt zurück.





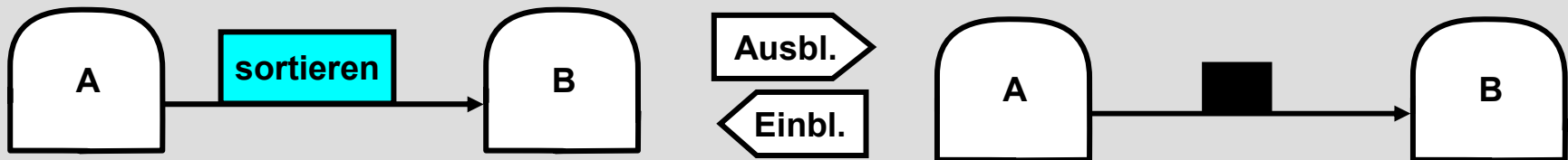
## 8) Ein- und Ausblenden – Spezifizierung von Relationen durch Basis-Elemente



Auch Basis-Elemente, die zur Spezifizierung einer Relation dieser zugeordnet werden, können ausgeblendet werden.



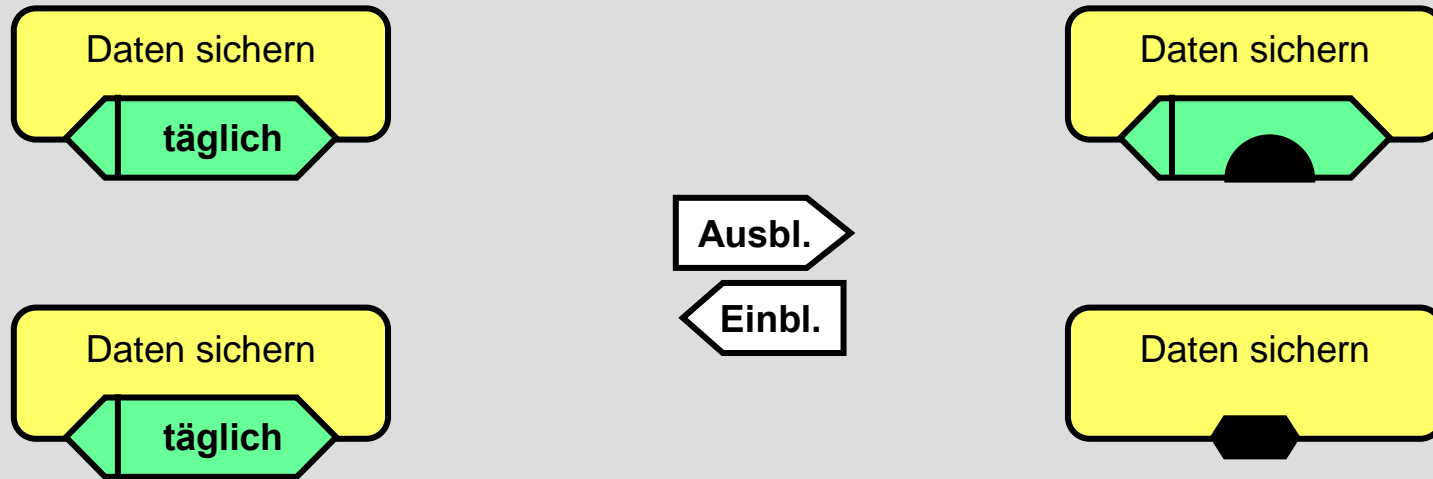
## 8) Ein- und Ausblenden – Definition von Relationen



Auch die Ersetzung einer Standarddefinition einer Relation durch eine andere oder die Erweiterung einer Spezifizierung kann ausgeblendet werden.



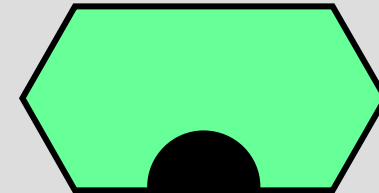
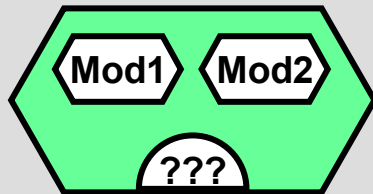
## 8) Ein- und Ausblenden - Modifikatoren



Bei Modifikatoren können entweder die enthaltenen Attribute oder der ganze Modifikator ausgeblendet werden. Letzteres kann mit einer Größenreduzierung einhergehen.



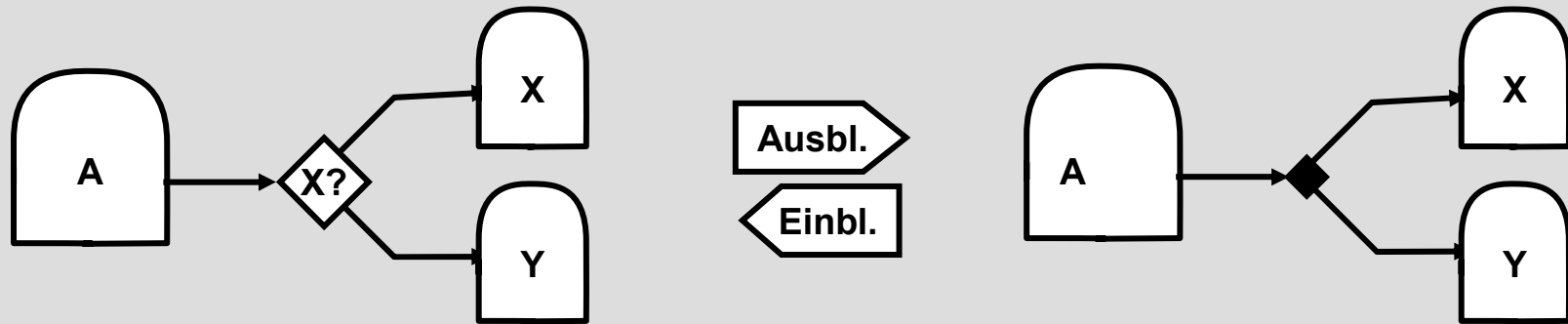
## 8) Ein- und Ausblenden – Sub-Modifikatoren



Bei Super-Modifikatoren können auch die Sub-Modifikatoren aus- und eingeblendet werden.



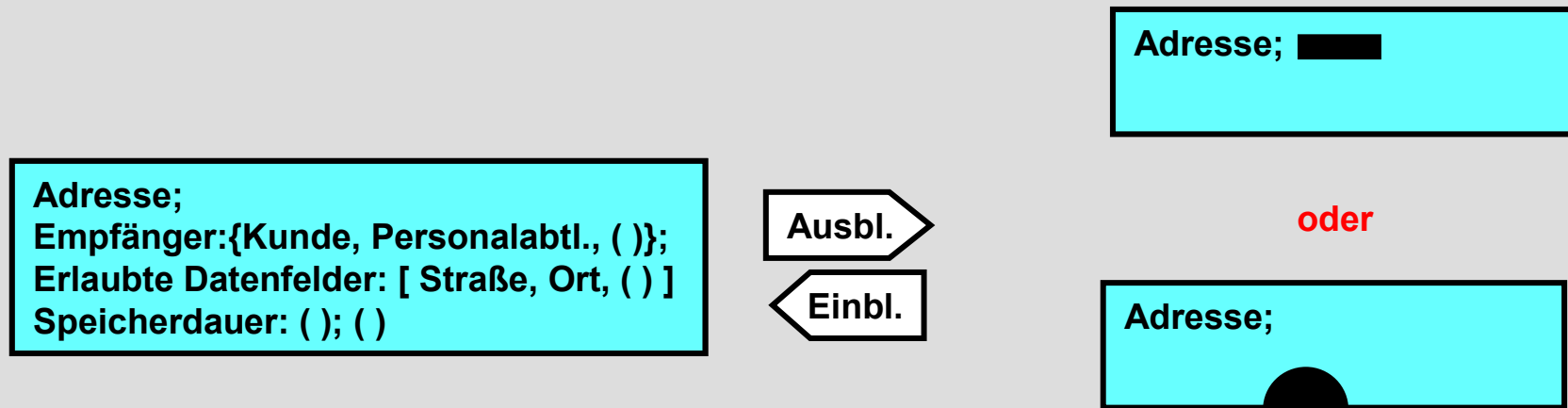
## 8) Ein- und Ausblenden - Konnektoren



Falls die Spezifikation eines Konnektors zu Irritationen führen kann oder bei der Betrachtung des Modells vom wesentlichen ablenkt, kann sie ausgeblendet werden – es ist dann sinnvoll, den Konnektor zu verkleinern.



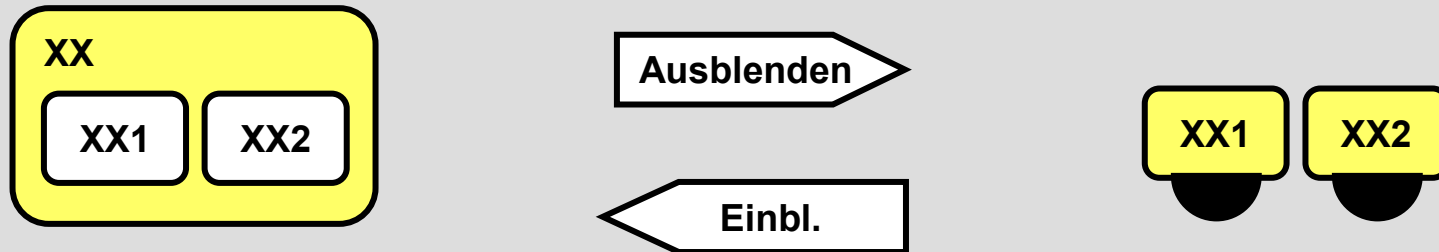
## 8) Ein- und Ausblenden - Attribute



Wenn man Attribute ausblendet, kann entweder ein schwarzer Halbkreis eingesetzt werden oder direkt eine rechteckige Fläche zu der Liste der Attribute hinzugefügt werden, damit keine Verwechslung mit anderen ausgeblendeten Sub-Elementen entsteht und keine Segmentlinie eingeführt werden muss.



## 8) Ein- und Ausblenden – Super-Elemente



Unter Umständen ist es sinnvoll, nicht die Sub-Elemente, sondern die Super-Elemente auszublenden. Dies ist dann angemessen, wenn man aus Gründen der Systematisierung zusammenfassende Super-Elemente eingeführt hat, die manche Betrachter mit ihren konkreten Erfahrungen nicht in Verbindung bringen können. Die Ausblendung des Super-Elementes hat zur Folge, dass der schwarze Halbkreis außen an den Sub-Elementen angebracht wird. Eine automatische Unterstützung sollte einen Farbwechsel herbeiführen.

## 8) Aufgaben zu Ein- und Ausblenden

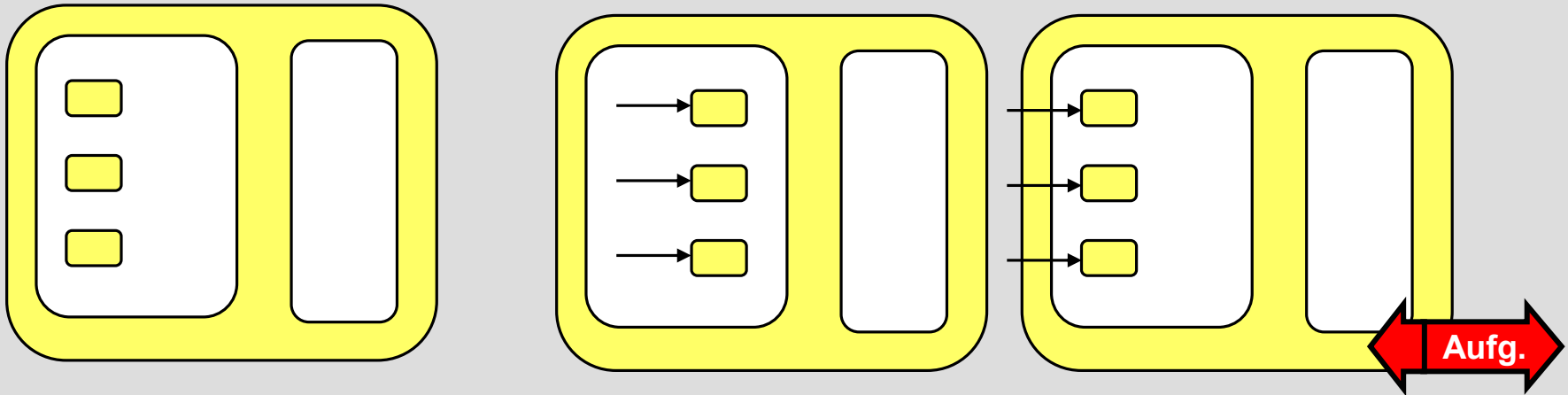
1. Reduzieren das Diagramm zum unspezifizierten Auftreten von Relationen durch Ausblende-Mechanismen auf drei unterschiedliche Basis-Elemente.
2. Entwickeln Sie ein Konzept, um die Grafik bei den Aufgaben zu 7.7 schrittweise durch Einblenden ausgehend von einem Element zu entwickeln.
3. Stellen Sie sich vor, dass Sie die papier-basierte Darstellung eines Prozesses auf einer anderen Seite fortsetzen als der, auf der er beginnt - wie stellen sie die Verbindung zwischen den beiden Prozessteilen dar?
4. Was passiert, wenn Sie bei dem Beispiel Einbettung von Rollen in Rollen genau die *Referate* und sonst nichts ausblenden?





## 9) Aufgaben allgemein

1. Modellieren Sie die Weitergabe eines Zugriffsrechtes.
2. Modellieren Sie die Weitergabe eines Zugriffsrechtes, an einer Kopie eines Originals
3. Modellieren Sie einen Rollenkonflikt
4. Klatschen, Trommeln, gleichzeitig Klopfen – wie stellt man diese der verschiedenen Varianten als Betätigung der Hände dar.
5. Was ist der Unterschied zwischen den drei unten gezeigten Diagrammen

**Aufg.**

## 9) Aufgaben allgemein

3. Modellieren sie eine Unterbrechung einer Aktivität - (einmal eine menschliche Aktivität, beim anderen mal eine maschinelle – diskutieren Sie den Unterschied.
4. Modellieren sie ein Paradoxon: Der Barbier soll alle Leute des Dorfes, in dem auch er wohnt, rasieren, die sich nicht selbst rasieren.



# 10) Schlagworte (A – Q)

Aggregation

Ankerpunkt +

Annotation +

Basis-Element

Bedingung

Beginnen

Container

Datenschutz

Deontisch

Einbettung +

Emergenz

Ende

Entscheidung

Erwartung +

Ereignis

Erzeugen +

Formal +

Gebot

Gleichzeitig

Hierarchie +

Informal +

Instantiierung +

Interesse

Klasse

Kompetenz

Konnektor

Maßeinheit

Meta-Element

Meta-Relation

Modifikator

Nebenläufig

Negation

(be)nutzen

ODER (OR)

Option

Organisationsstruktur

Ort +

Person +

Perspektive

Propagieren

Prozess + + +

Quantor



# 10) Schlagworte (R – Z)

Rationalisierung

Rekursiv

Relation +

Rolle

Rollenspiel

Schleife

Segment

Start

Struktur +

Sub-System

Super-Element +

Typ +

UND

Unsicherheit +

Unterbrechung

Unvollständigkeit

Vagheit

Verbot

Vererbung

Verzweigung

Vollständigkeit

Wert

Widerspruch

Wissen

XOR

Zeit + +

Zugriffsrecht

Zusammenführung

