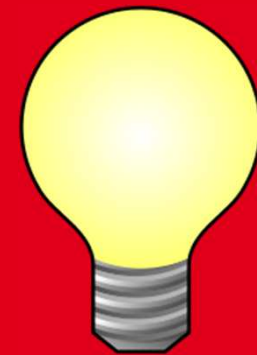


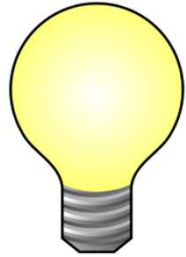


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

20.11.2018 Interaktionsdiagramme

Interaktionsdiagramme nutzen





Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

AGENDA

Einführung ins Thema

Sequenzdiagramme

Sequenzdiagramme – 2 verschiedene Semantiken

Kommunikationsdiagramme

Weitere Interaktionsdiagramme

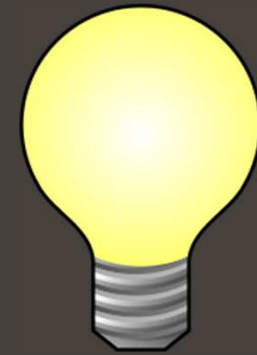
Fazit



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

01 EINFÜHRUNG INS THEMA

Ziel:
Die Eckpunkte des Themas kennenlernen

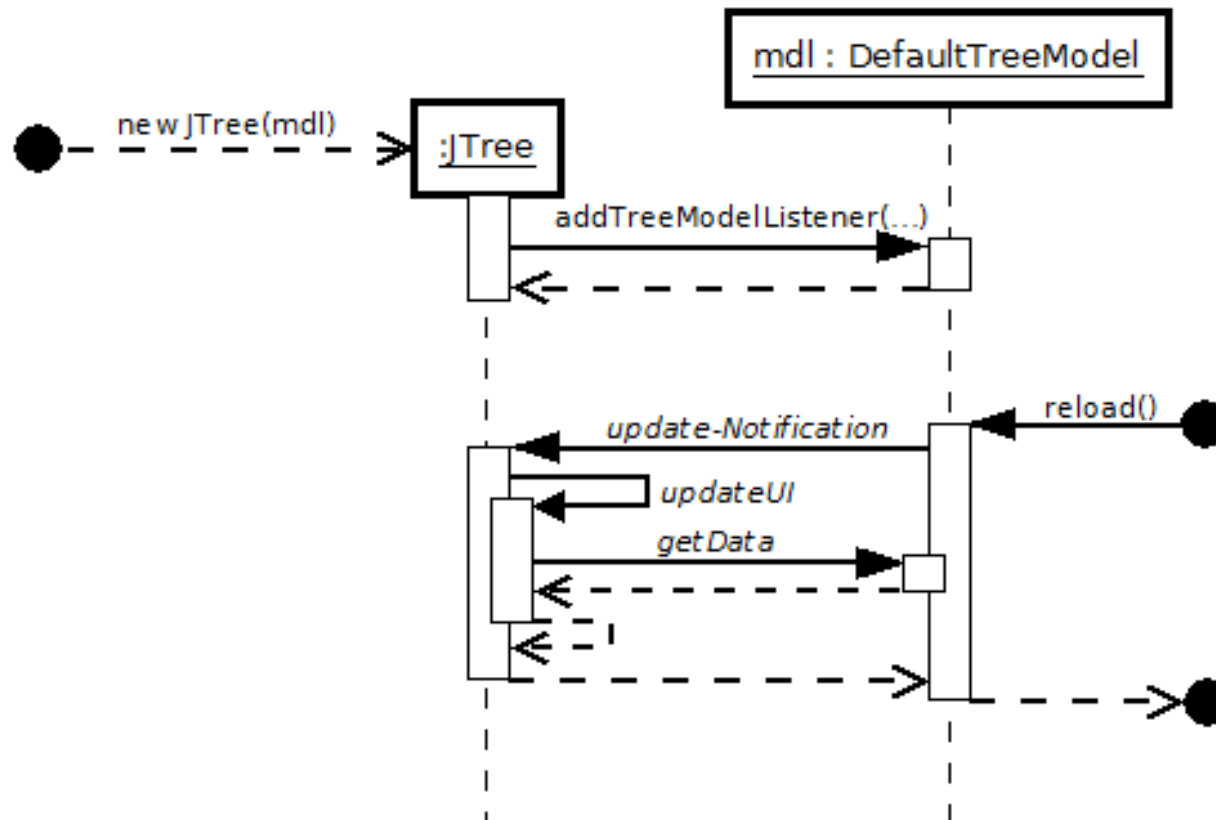


2. SEMESTER, PM: DEN JTREE VERÄNDERN – HINTERGRÜNDE VON RELOAD()



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Sequenz-Diagramm über den groben Ablauf:



Hinweis:

Kursiv gesetzten Ausdrücke *update-Notification*, *updateUI* und *getData* zeigen nur ungefähr den Ablauf, die wahren Namen der Funktionen sind Javainternas!

INTERAKTIONSDIAGRAMME



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Zeigen die Interaktionen zwischen Objekten
 - Interaktion = Abfolge von Nachrichten, die zwischen Objekten ausgetauscht werden
- Grundsätzlich gibt es 4 Interaktionsdiagramme:
 - Sequenzdiagramm (wichtigste)
 - Kommunikationsdiagramm
 - Interaktionsübersichtsdiagramm
 - Metadiagramm, das die Zusammenhänge zw. verschied. Interaktionsdiagrammen zeigt
 - Timing Diagramm:
 - Zeigt Nachrichtenaustausch und Zustandwechsel verschiedener Objekte zu bestimmten Zeitpunkten an.

} Diese besprechen wir
im Detail

BEVORZUGTE EINSATZGEBIETE



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Man hat mehrere Einheiten (Objekte), die Nachrichten austauschen
- Zeitliche Abfolge der Nachrichten sollen dargestellt werden
- Snapshot-Charakter (ähnlich wie Objektdiagramme)
- Entscheidungshilfe:
 - Zeitliche Abfolge wichtig: Eher Sequenzdiagramm
 - Auch Aktivitäten innerhalb der Objekte teilweise darstellbar
 - Eher Struktur wichtig: Eher Kommunikationsdiagramm

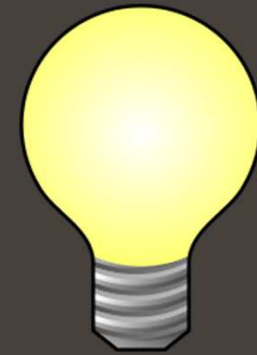


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

02

Sequenzdiagramme - Modellelemente im Überblick

Ziel:
Elemente im Überblick erfassen



CODE-BEISPIEL:

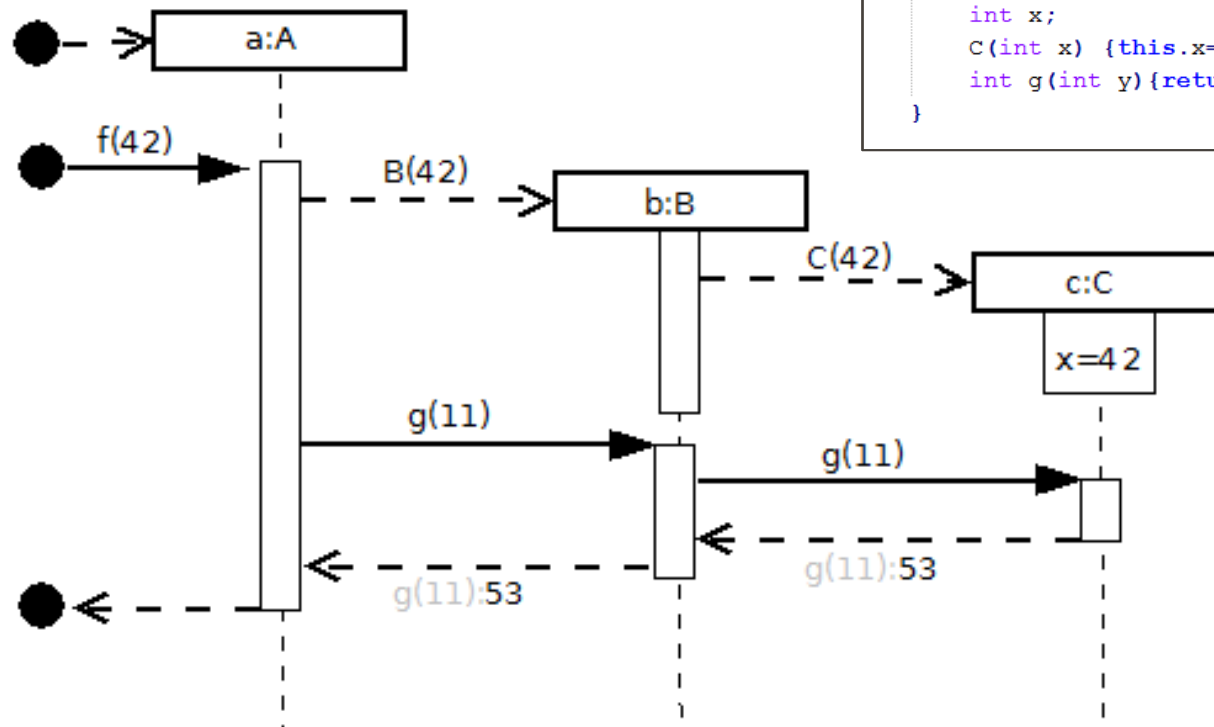


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

```
class A {  
    void f(int x){  
        B b = new B(x);  
        int z;  
        z=b.g(11);  
    }  
  
    public static void main(String [] args){  
        A a=new A();  
        a.f(42);  
    }  
}  
  
class B{  
    C c;  
    B (int x) { c=new C(x);}  
    int g(int y){return c.g(y);}  
}  
  
class C {  
    int x;  
    C(int x) {this.x=x;}  
    int g(int y){return x+y;}  
}
```


DER CODE ALS SEQUENZDIAGRAMM:

```
class A {  
    void f(int x){  
        B b = new B(x);  
        int z;  
        z=b.g(11);  
    }  
  
    public static void main(String [] args){  
        A a=new A();  
        a.f(42);  
    }  
}  
  
class B{  
    C c;  
    B (int x) { c=new C(x);}  
    int g(int y){return c.g(y);}  
}  
  
class C {  
    int x;  
    C(int x) {this.x=x;}  
    int g(int y){return x+y;}  
}
```

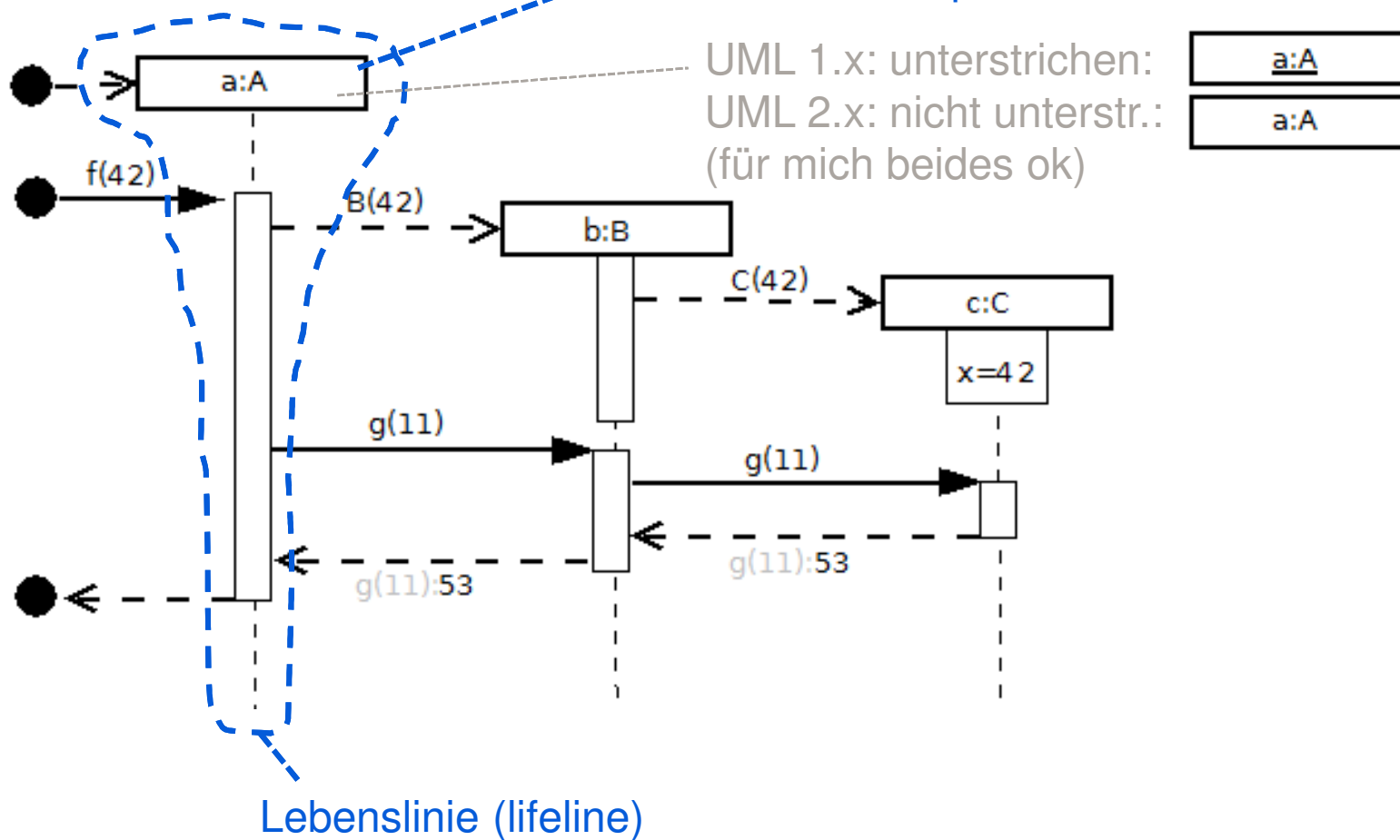


DIE ELEMENTE IM DETAIL:



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

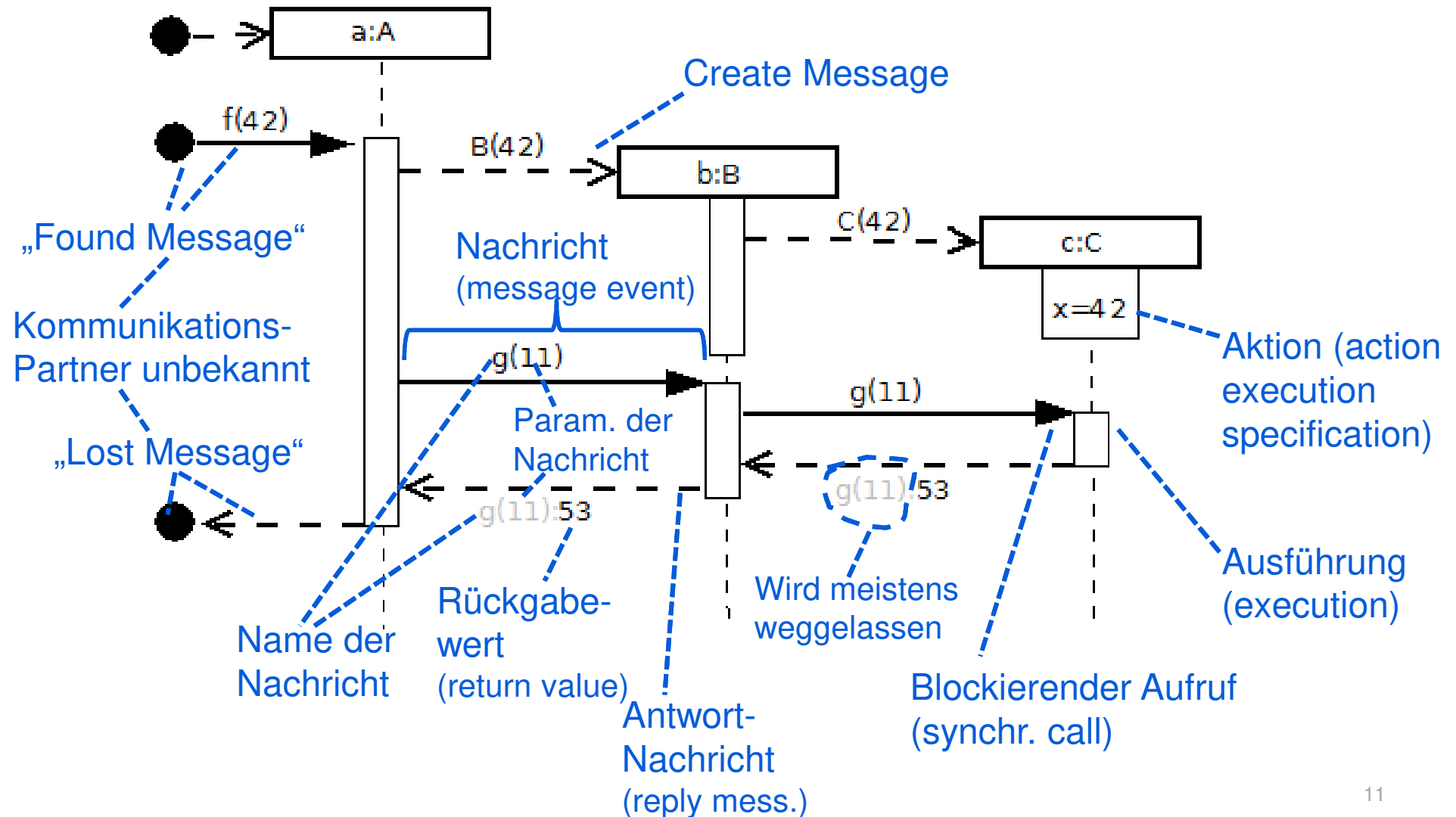
Objekt /
Kommunikationspartner



DIE ELEMENTE IM DETAIL:



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



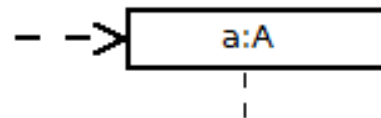
DIE ELEMENTE IM DETAIL:



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Nachrichtentypen bzgl. Aufruf/Rücksprung einer Op.:

1. Erzeugungsnachricht (Create Message):



2. Synchrone Nachricht (blockiert):



3. Asynchrone Nachricht (blockiert nicht):



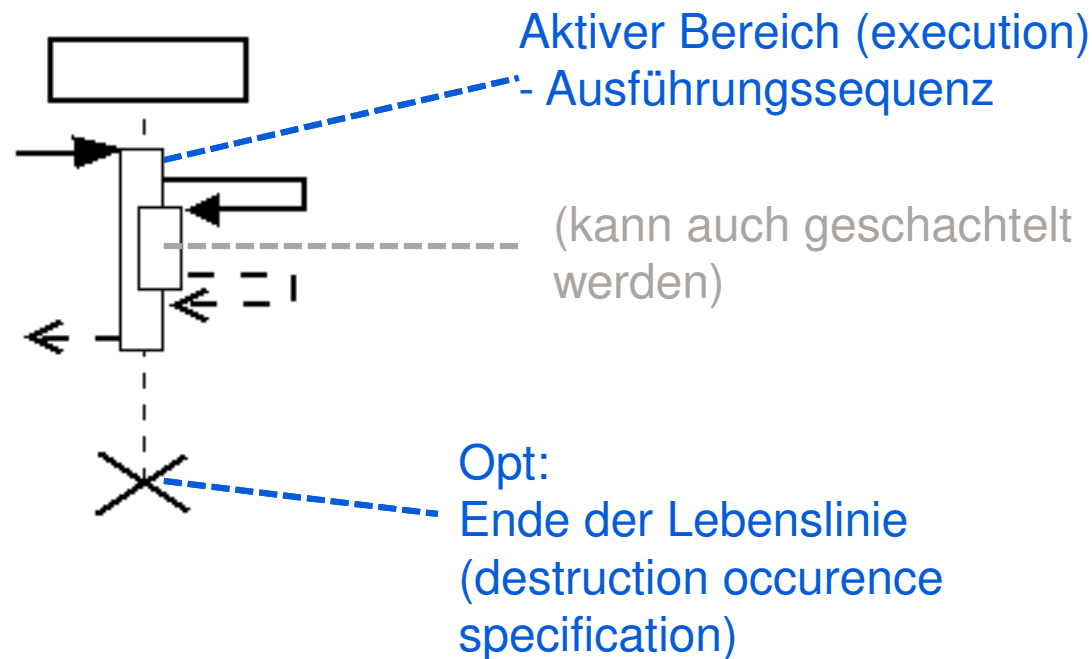
BEM: Exceptions werden auch “nur” als normale Nachrichten behandelt.

DIE ELEMENTE IM DETAIL:



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Die Lebenslinie im Detail:

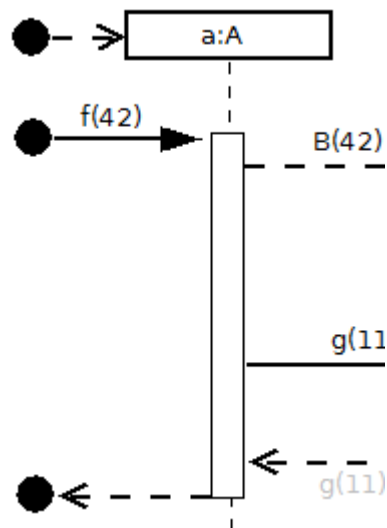


LOST-FOUND MESSAGES VS SOG. GATES



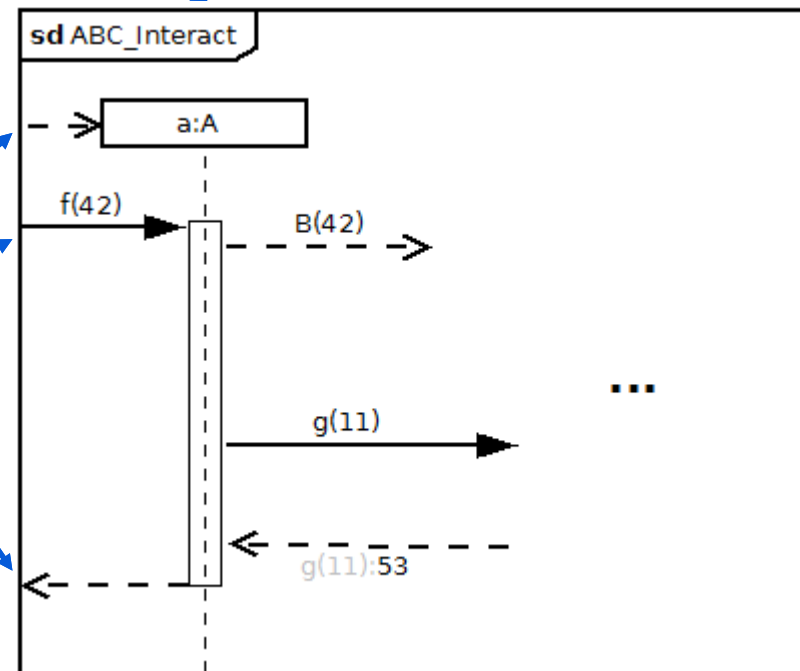
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Lost-Found-Messages:



Gates:

Gates

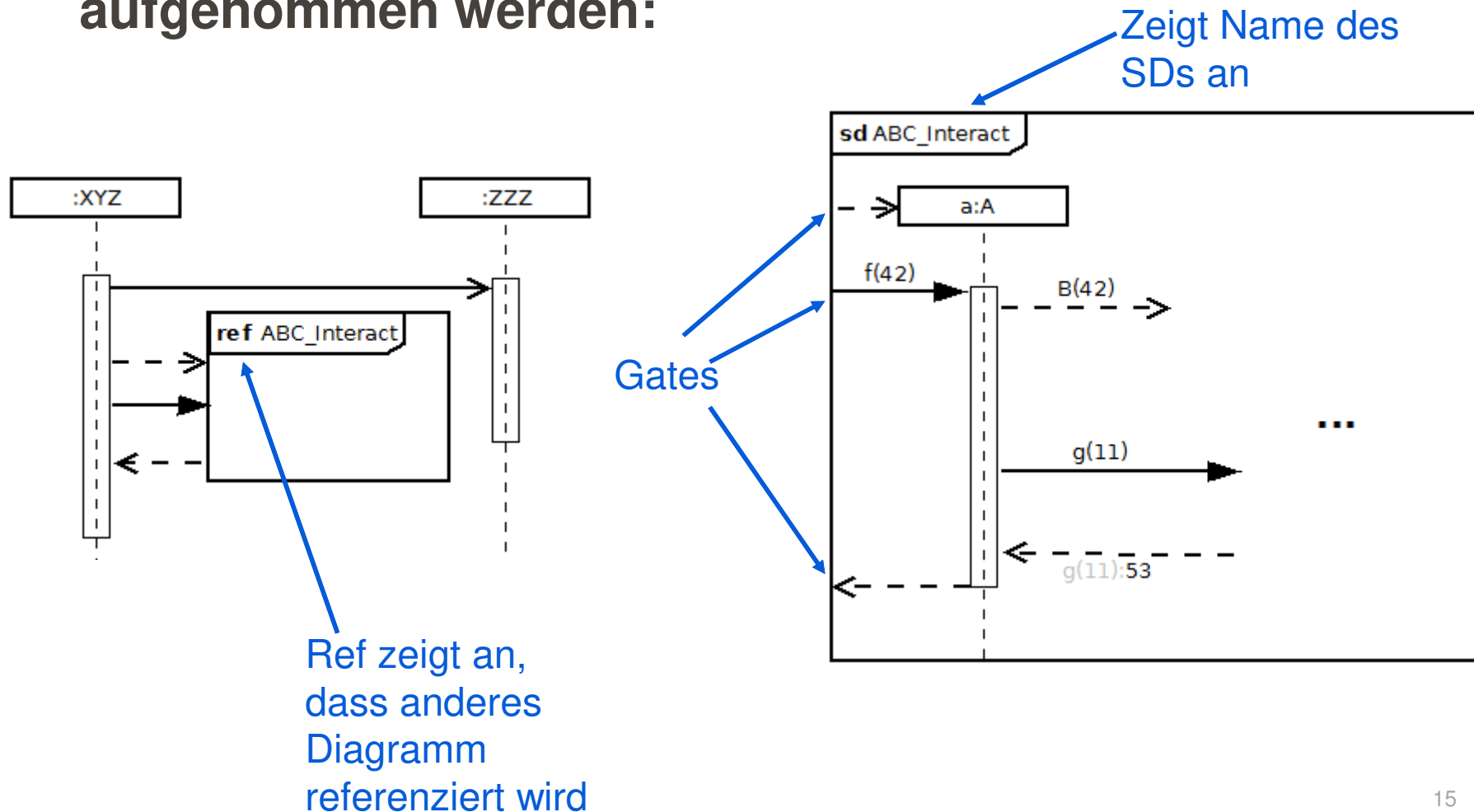


BEM LOST-FOUND MESSAGES UND SOG. GATES



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

→ **Gates sind Anknüpfungspunkte, die bei Referenzen aufgenommen werden:**



GENERELL MÖGLICHE NACHRICHTENTYPEN:



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

1. Erzeugungsnachricht (Create Message): 

2. Synchrone Nachricht (blockiert):



- Sender wartet, bis Empfänger die Nachricht abgearbeitet hat
- Gestrichelter Pfeil für Rücksprung zum Sender nötig!

3. Asynchrone Nachricht (blockiert nicht): 

- Sender wartet nicht auf Empfänger und arbeitet unmittelbar weiter
→ Sender und Empfänger befinden sich in unterschiedlichen Ausführungsprozessen
- Kein gestrichelter Rückgabepfeil!!

BEM: Exceptions werden auch “nur” als normale Nachrichten behandelt.

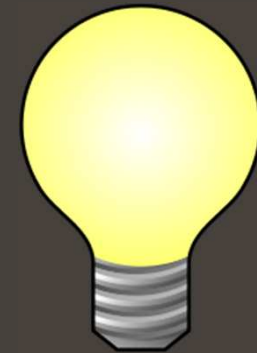


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

03

Sequenzdiagramme – 2 verschiedene Semantiken

Ziel:
Elemente im Überblick erfassen



ZWEI VERSCHIEDENE SEMANTIKEN



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Die wichtigsten Elemente des Sequenzdiagramms kennengelernt
 - Syntax
- Es gibt aber auch zwei Arten der Bedeutung (Semantik):
 - Aufrufsemantik
 - Orientiert sich daran wie in Programmiersprachen Methoden aufgerufen werden
 - Sehr nah an Programmiersprachen
 - Signalsemantik
 - Sagt nur allgemein aus, dass “Signale ausgetauscht” werden
 - Viel allgemeinere Aussage

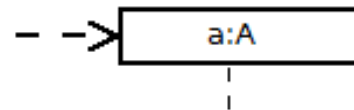
DIE AUFRUF-SEMANTIK:



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Nachrichtentypen wie Aufruf/Rücksprung von Methoden:

1. Erzeugungsnachricht (Create Message):



2. Synchrone Nachricht (blockiert):



3. Asynchrone Nachricht (blockiert nicht):



BEM: Exceptions werden auch “nur” als normale Nachrichten behandelt.

→ Alle Elemente möglich und Bedeutung
wie vorher erklärt

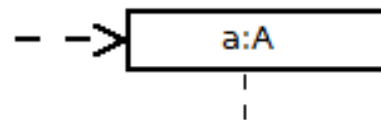
DIE SIGNAL-SEMANTIK



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Übermittlung von 2. Signalen:

- Es wird nur abstrakt von Signalen gesprochen
- Mögliche Nachrichten:
 1. Erzeugungsnachricht (Create Message):



2. Signale:



→ Signale sind als asynchron definiert

- Deshalb wieder der Asynchronpfeil
- Finde ich persönlich etwas ungeschickt, weil für ein Signal vielleicht gar nicht klar ist, ob es asynch oder synch ist.

WARUM BRAUCHEN WIR ZWEI SEMANTIKEN?



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Die Aufrufsemantik ist sehr detailliert
 - Sehr nah an Programmiercode & Programmabläufen
 - Viele Details wie z.B.: wer ruft wen wie genau auf, Rücksprünge müssen definiert sein, ...
 - Schon sehr genaue Festlegung
- Bei der Signalsemantik sind die Signale eher unbestimmt
 - Eignen sich auch besonders, wenn man noch nicht, weiß ob es wirklich ein Aufruf oder etwas anderes ist
 - Eignen sich also gut in den frühen Phasen, wenn man noch nicht festlegen möchte, ob synch. oder asynch. Aufruf, oder was auch immer...
 - siehe dazu auch Dokument mit Bem. zur Robustness Analysis
(lade ich ein paar Wochen später auf die Homepage)

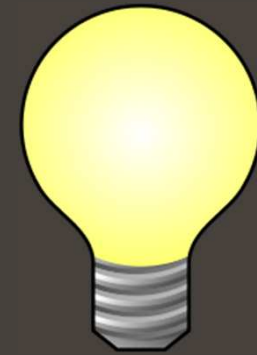


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

04

Kommunikationsdiagramme – Modellelemente im Überblick

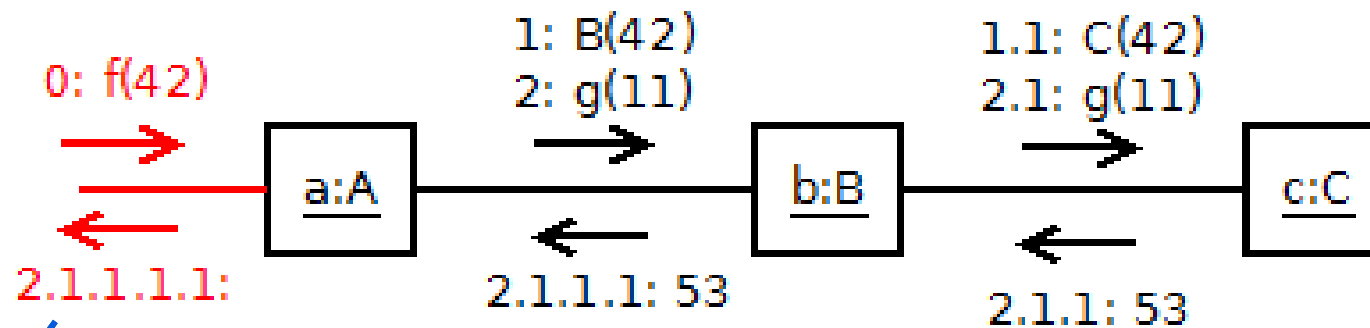
Ziel:
Die Elemente im Überblick erfassen



VORHERIGES BEISPIEL ALS KOMMUNIKATIONSDIAGRAMM:



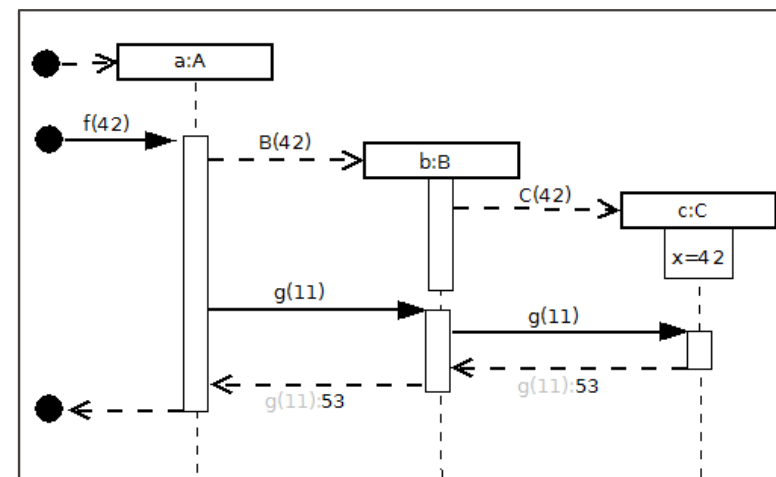
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



BEM zu rot markiertem Bereich:

Vorschlag von Bodo Iglar für lost/found Messages

(gibt es nicht in UML2.x bzw. ist dort nicht geklärt)

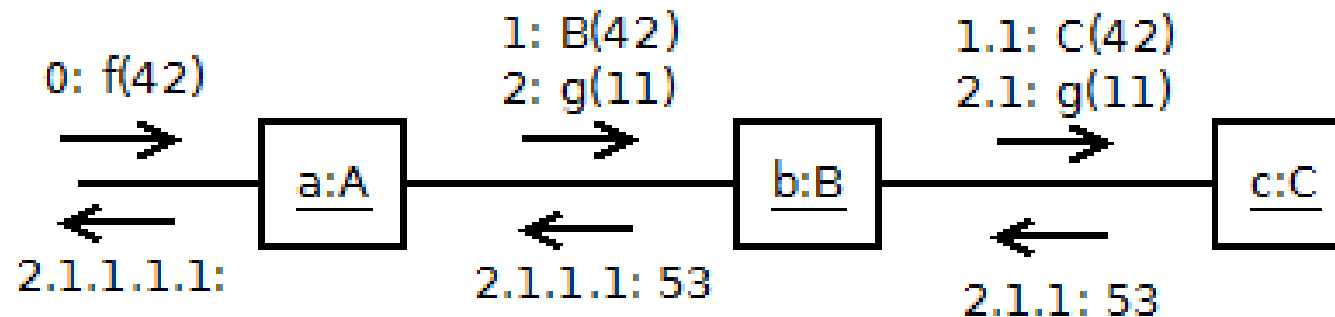


KOMMUNIKATIONS DIAGRAMM

WEITERE BEMERKUNGEN



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



- BEM: Reihenfolge ist nur anhand der Nummerierung ersichtlich
- BEM zu Pfeilen:
 - UML 1.x: Gleiche Pfeile wie in Sequenz-Diagramm
 - UML 2.x: Nur \longrightarrow (==Semantikverlust!)Für mich: Sie können beides verwenden

EIGENSCHAFTEN KOMMUNIKATIONS DIAGRAMME



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Im Prinzip: Kommunikationsdiagramm hat den gleichen Informationsgehalt wie Sequenzdiagramm
- **ABER:** Nicht alles kann dargestellt werden:
 - Fokus auf (Nachrichten-)Verbindungen zwischen den Kommunikationspartnern
 - Information über Erzeugung und Zerstörung geht verloren
 - Zusammenhänge zw. Nachrichten sind schwerer lesbar
- **Stärke des Kommunikationsdiagramms:**
 - Zwitter aus Struktur- und Interaktionsdiagramm
 - Objekte werden oft in selber Anordnung wie in Klassendiagramm angeordnet (== Struktursicht)
 - Nachrichten zeigen dann die Interaktionen



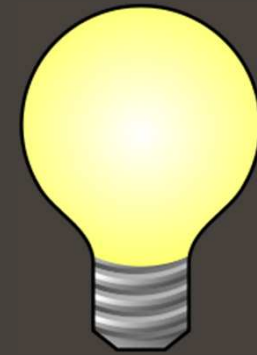
Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

05

Weitere Interaktionsdiagramme

Ziel:

Kurze Vorstellung der anderen Interaktionsdiagramme





TIMING-DIAGRAMM

Zeigt Nachrichtenaustausch und Zustandswechsel verschiedener Objekte zu bestimmten Zeitpunkten an:

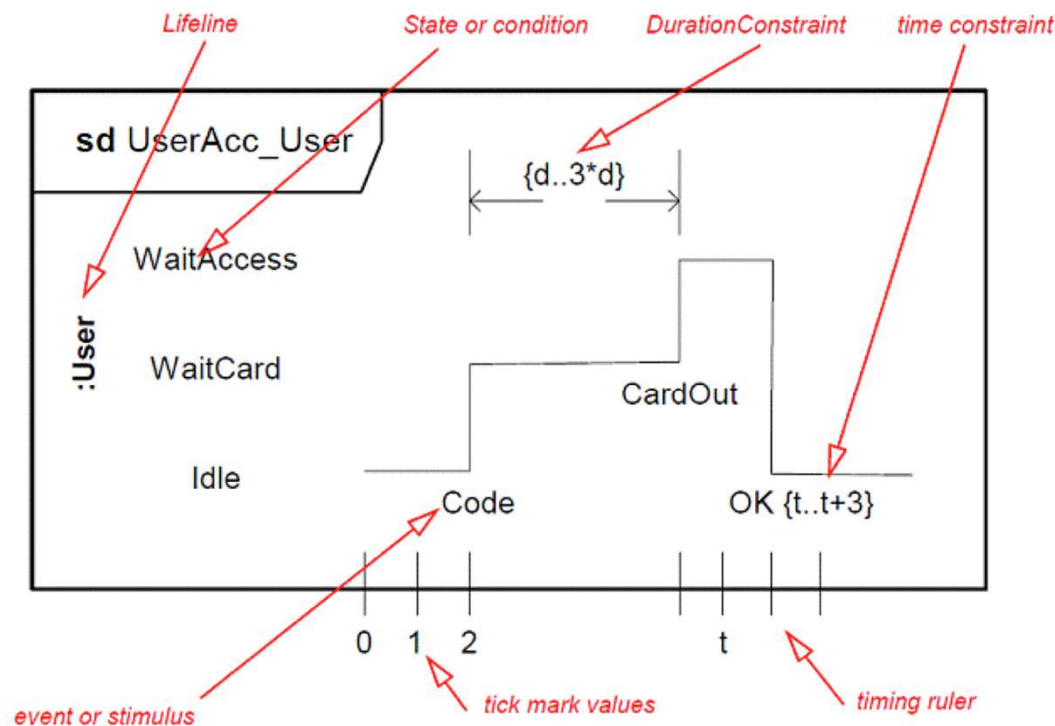


Figure 17.28 A Lifeline for a discrete object

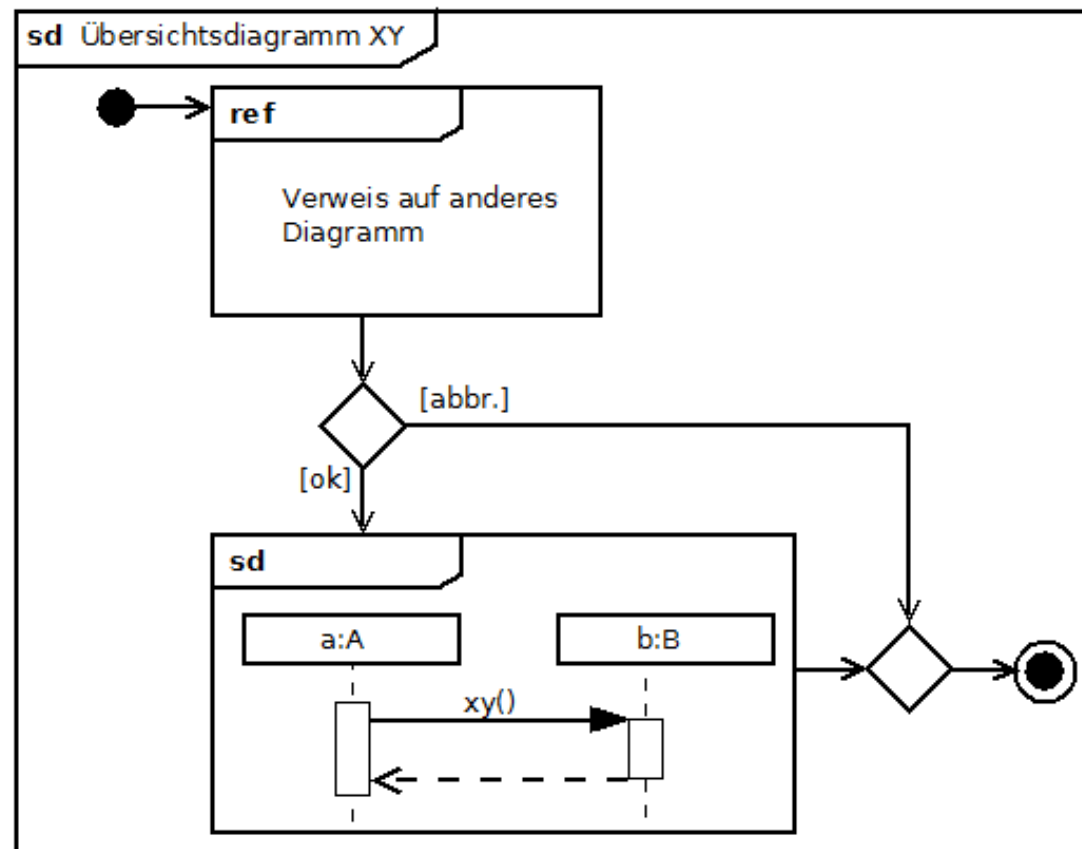
Quelle: UML 2.5-Standard

INTERAKTIONSÜBERSICHT-DIAGRAMM



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

Metadiagramm, das die Zusammenhänge zw. verschied. Interaktionsdiagrammen zeigt:



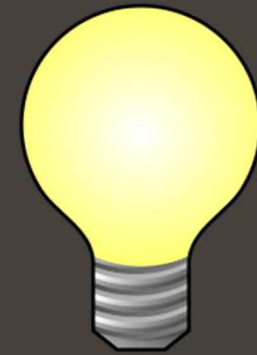
Quelle: UML 2.5-Notationsübersicht auf www.oose.de/uml



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

06 Fazit

Ziel:
Was haben wir damit gewonnen?





WAS HABEN WIR GELERNT?

- Interaktionsdiagramme
 - Sequenzdiagramm (das wichtigste)
 - Kommunikationsdiagramm
 - (Timing-Diagramm)
 - (Interaktionsübersichtsdiagramm)
- Sequenzdiagramm
 - Zeigt Objekte und deren Nachrichtenaustausch im Zeitverlauf



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

AUF GEHT'S!!

SELBER MACHEN UND LERNEN!!

