

Mobile Objekte

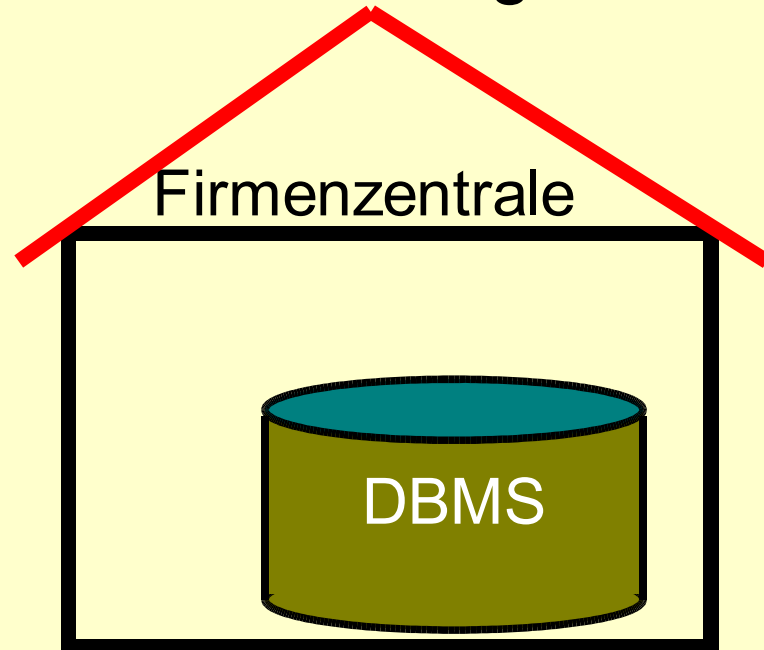
Datenmodelle und Anfragen

„Verteilung und Integration von
Informationen im Verkehrsbereich“

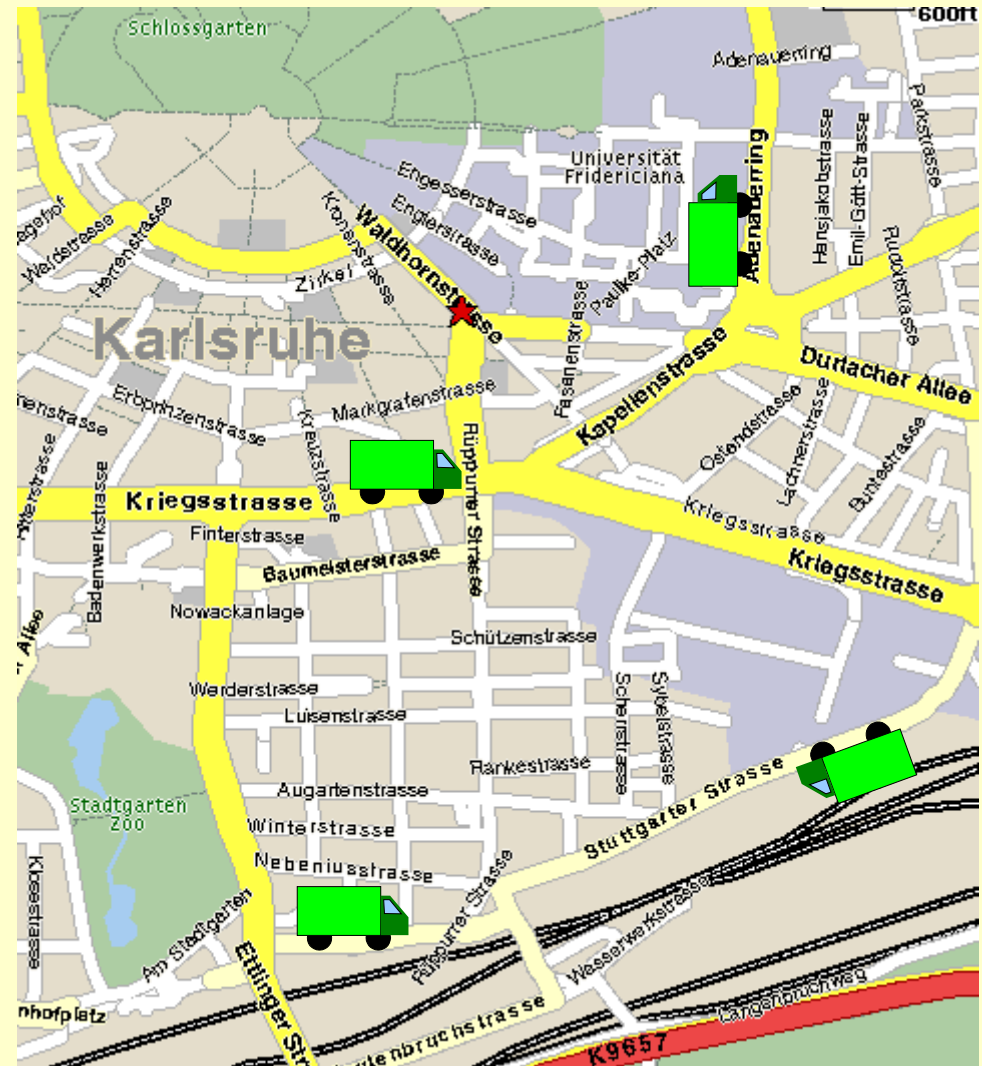
Seminar am IPD im SS 2004

Einführendes Beispiel

Szenario: Kurierdienst mit vielen Lieferwagen

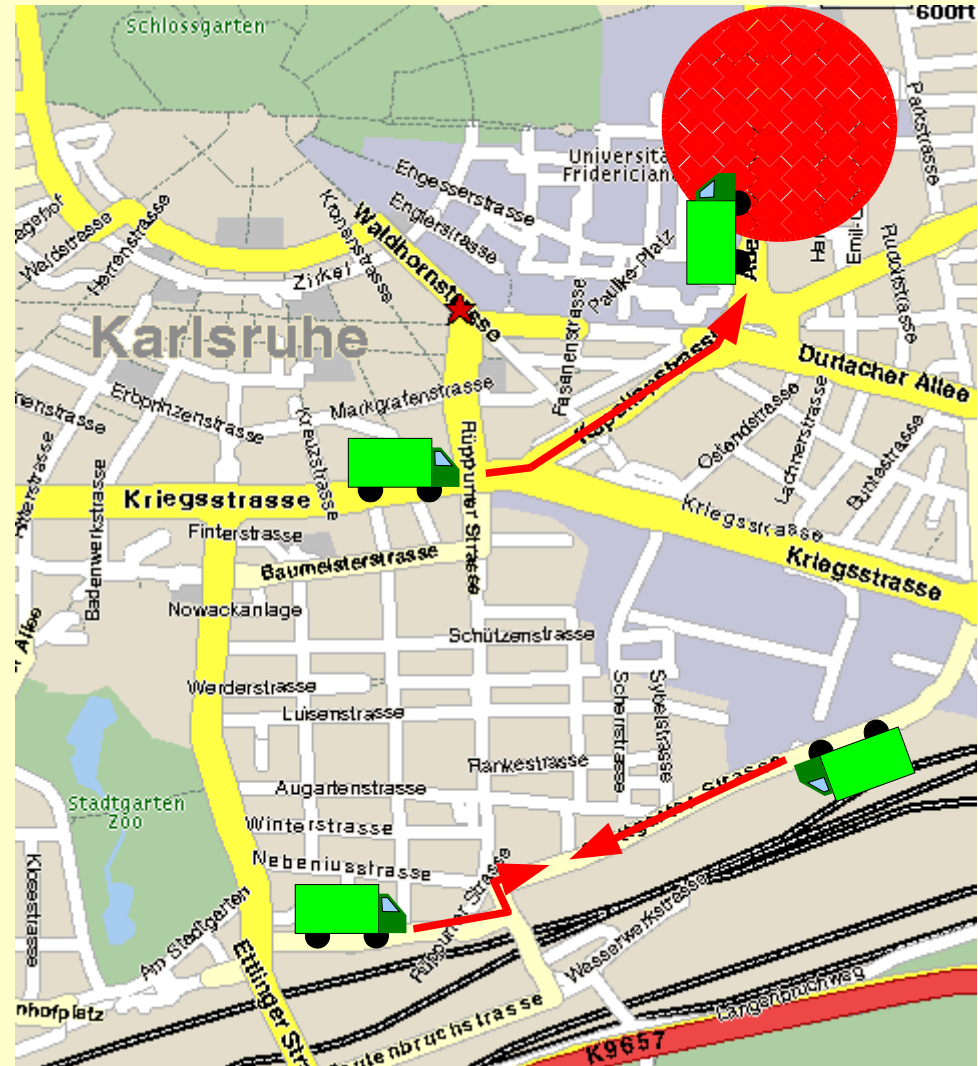


Speicherung der Position aller Lieferwagen



Beispielanfragen

- A1 Welcher Wagen befindet sich (innerhalb der nächsten 5 min) in der Nähe vom Informatikgebäude?
- A2 Welche Lieferwagen werden sich in den nächsten 30 min begegnen?
- A3 Welche Lieferwagen sind unbeladen bis sie den Fasanengarten erreichen?



Klassische DBMS

- Attributwert **konstant**
- Bewegung erfordert ständiges Aktualisieren
 - ◆ **Teuer**
 - Änderungsaufwand
 - hohe Belastung des Netzwerkes
 - ◆ **ungenau** Positionsinformation

Klassische DBMS II

A1 Welcher Lieferwagen ist innerhalb der nächsten 5 min am Informatikgebäude?

- **Räumlicher** Aspekt: Informatikgebäude
- **Zeitliche** Bedingung: innerhalb von 5 min

Traditionelle Anfragesprachen (z.B. SQL) eher **ungeeignet**

- keine zeitlichen und räumlichen Operatoren

Wie geht der Vortrag weiter?

1. Lösungen für die Modellierung von Positionsinformationen
 - (a) **MOST – Datenmodell** von Wolfson u.a.
 - (b) **Datenmodell für Moving Objects Databases** von Forlizzi u.a.
2. Räumliche und zeitliche Anfragesprache - FTL

MOST – Datenmodell

klassische DBMS:

- **Statische Attribute**
(Attributwert konstant,
ändert sich nur bei
einem expliziten Update)

MOST – Datenmodell:

- Statische Attribute
- Einführung von **dynamischen Attributen**
(Attributwert ändert sich
als Funktion der Zeit)

MOST – Datenmodell II

Dynamisches Attribut A wird durch drei Subattributen repräsentiert:

- $A.update\ time$
- $A.update\ value$
- $A.function$

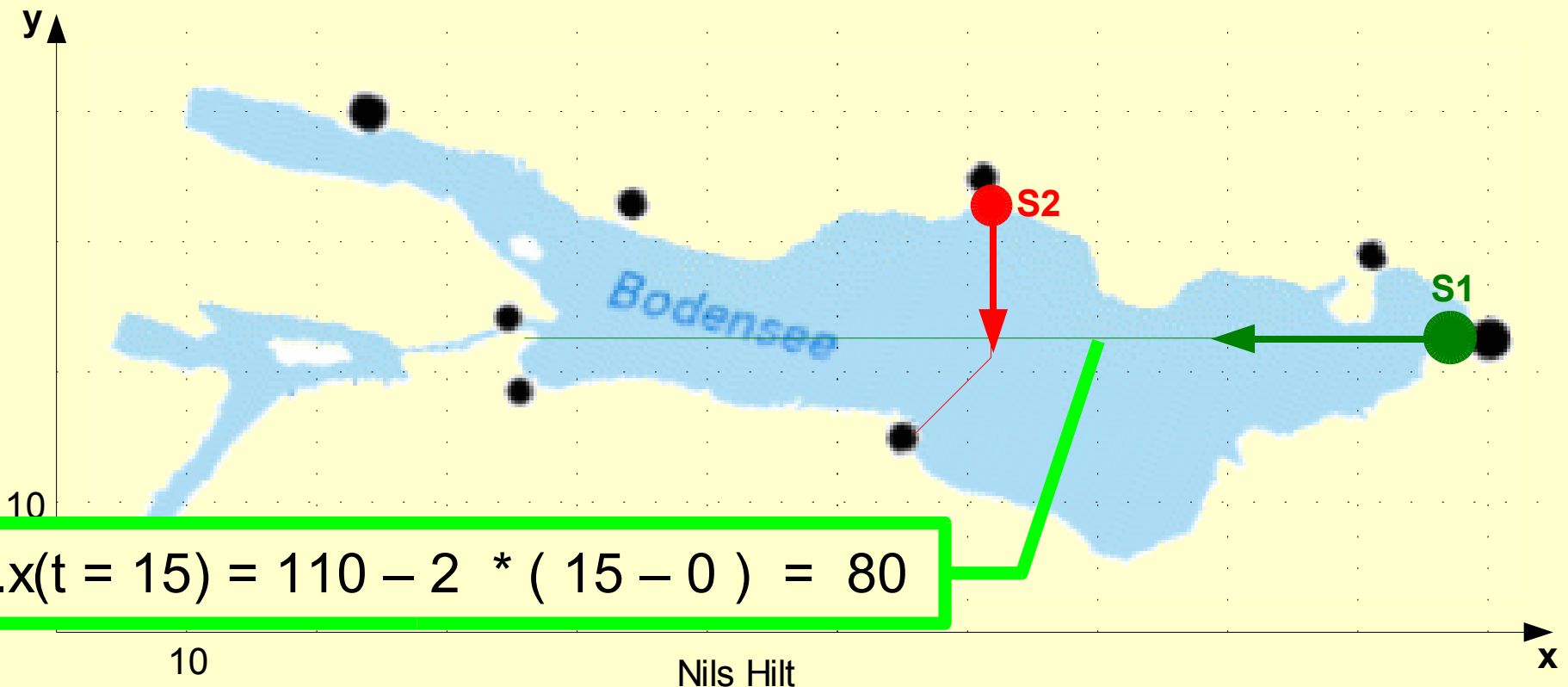
Wert von A zum Zeitpunkt t :

$$A = A.update\ value + A.function(t - A.update\ time)$$

MOST Beispiel

Zeit $t = 10$

ID	X – Koordinate			Y – Koordinate		
	updatetime	updatevalue	function	updatetime	updatevalue	function
S1	0	110	- 2	0	22	0
S2	10	71	- 1	0	35	- 1

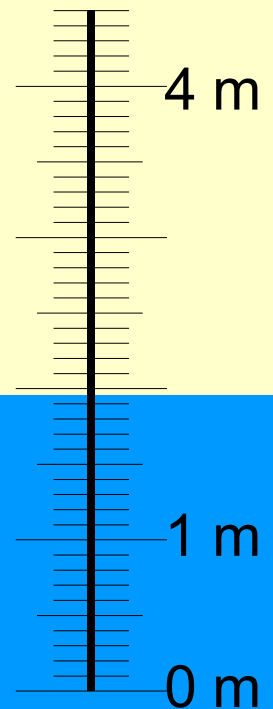
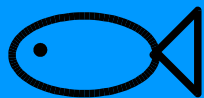


MOST - Bemerkung

- Dynamische Attribute nicht nur für Positionsinformation
- Geeignet für alle sich zeitlich verändernden Zusammenhänge

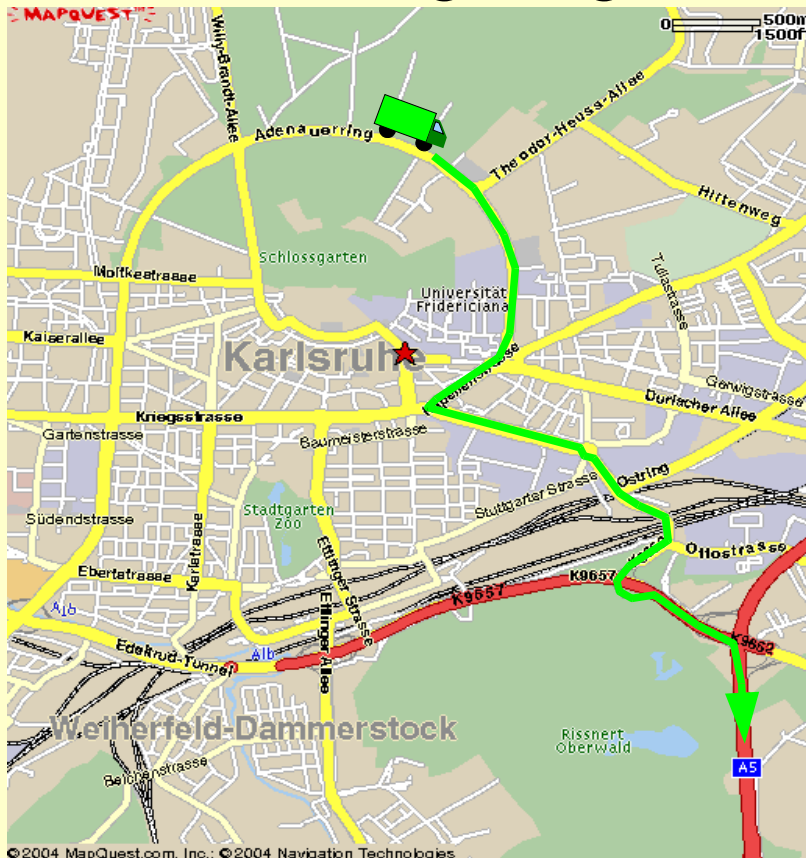
Bsp.: Wasserstand an Nordseeküste

- *updatezeit* = Zeitpunkt Ebbe
- *updatevalue* = Wasserstand bei Ebbe
- *function* = Flutungsgeschwindigkeit



MOST - Nachteil

- MOST gut geeignet für Positionsinformation von Schiffen, Flugzeuge etc.



- Autofahrt erfordert **häufige Änderung** der dynamischen Attribute
 - MOST ungeeignet
- Erweiterung von MOST um das Konzept der **Route**

MOST - Route

Dynamisches Attribut L enthält fünf Subattribute:

- $L.route$
- $L.x.updatevalue$
- $L.y.updatevalue$
- $L.updatetime$
- $L.speed$

Weiterer Modellierungsansatz

Forlizzi u.a. stellen einen anderen Ansatz vor:

- Nicht nur Bewegung von punktförmigen Objekten
- Bewegung von **Punkten, Linien, Regionen**, etc.
- **Veränderung der Ausdehnung und Form** von Linien und Regionen

Abstraktes und diskretes Modell

Modellierung in zwei Schritten:

1. Abstraktes Modell:

- Einfach
- Fokus auf relevante Konzepte
- Nicht für die Implementierung geeignet

2. Diskretes Modell:

- Komplexer
- Legt interne Darstellung fest

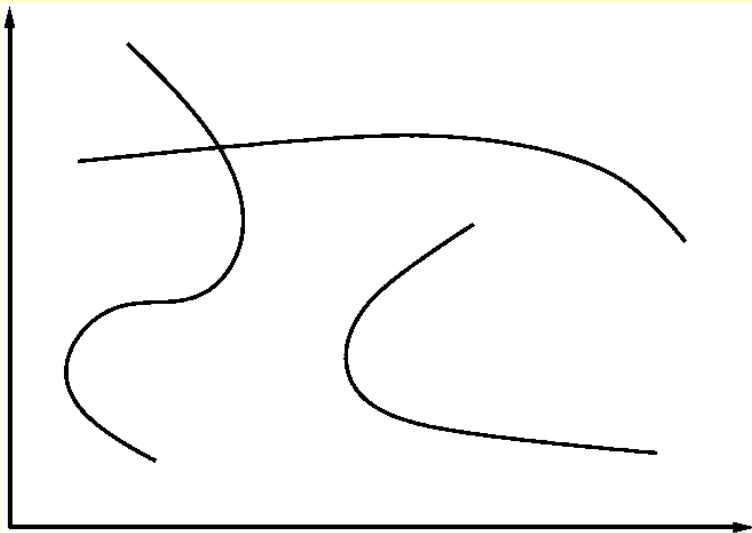
Abstraktes Modell

- Datentypen
 - ♦ *int, real, point, line, region, ...*
- Typkonstruktoren z.B. *moving*
 - ♦ *moving(point), moving(region), ...*
- Operationen:
 - ♦ **trajectory:** *moving(point) → line*
 - ♦ **length:** *line → real*
 - ♦ **distance**
 - ♦ **etc.**

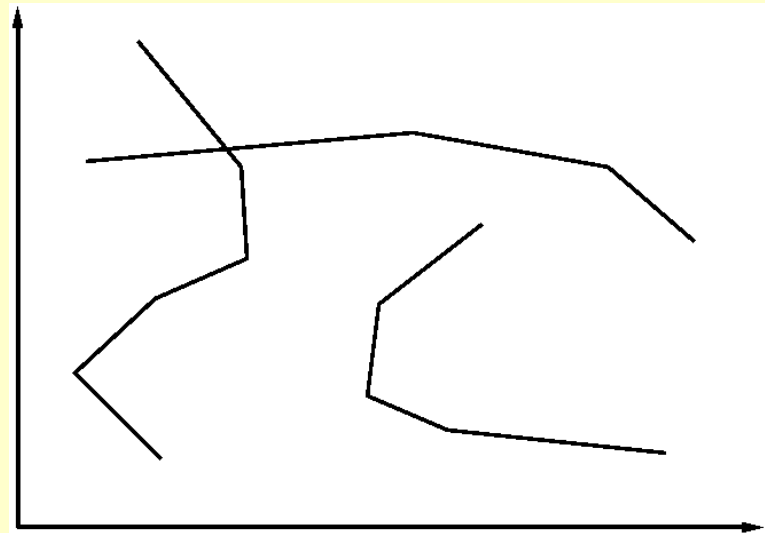
Diskrete Darstellung

Räumliche Objekte

- abstrakte line, Repräsentation als **Polylinie**

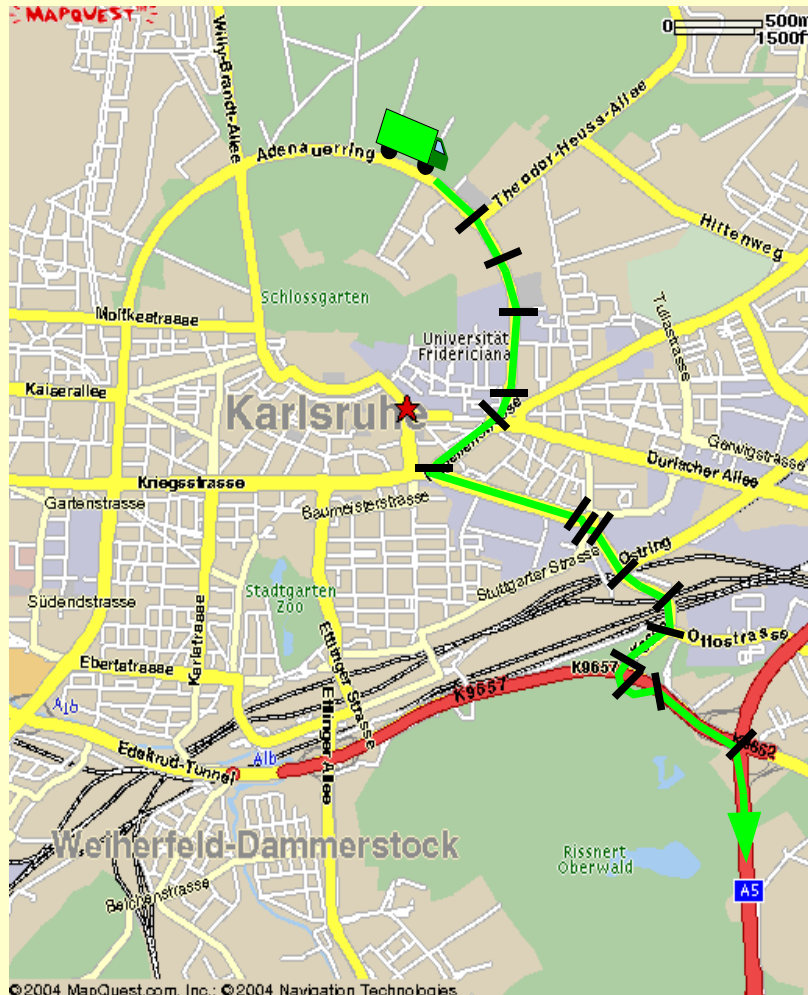


abstrakte line



diskrete line

Sliced Representation



- Aufteilung der Route in Teilstrecken
- Vereinigung mehrerer **Unittype** (i, u)
 - ♦ lineare Funktion u zur Darstellung der Teilstrecken
 - ♦ benachbarte Gültigkeitsintervalle i der Teilstreckenfunktionen

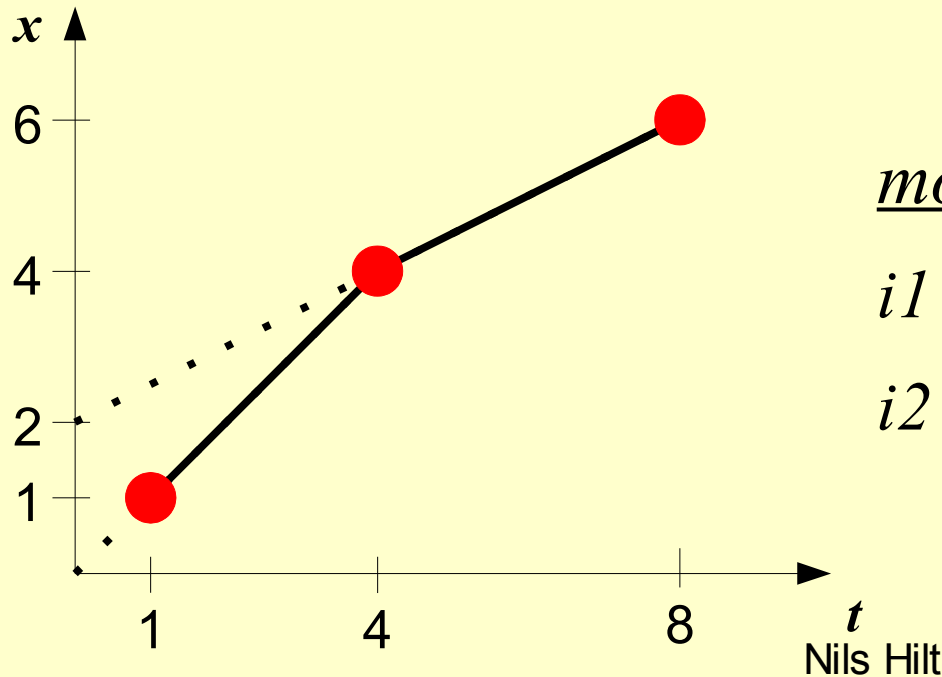
Diskreter moving(point)

- **Unittyp** (i, u) für moving(point) konkret

- ♦ $u = (x0, x1, y0, y1)$

$$x(t) = x0 + t * x1$$

$$y(t) = y0 + t * y1$$



moving(point):

$$i1 = [1, 4] \quad u1 = (0, 1, ?, ?)$$

$$i2 = (4, 8] \quad u2 = (2, \frac{1}{2}, ?, ?)$$

MOST vs. Forlizzi

MOST

- einfaches Modell
- nur Bewegung von Punkten
- eigentlich nur geradlinige Bewegung (Verbesserung durch *Route*)

Forlizzi

- sehr mächtiges abstraktes Modell
- Bewegung von Punkten, Regionen, etc.
- abstrakte Modellierung beliebiger Bewegungskurven
- komplexe Umsetzung in diskrete Darstellung

Was bisher geschah ...

- Zwei Datenmodelle für Moving Objects Databases
 - ◆ Wolfson u.a.
 - ◆ Forlizzi u.a.

Was fehlt noch ?

Räumliche und zeitliche Anfragesprache

Datenbankhistorie

- Traditionelle Anfragesprachen
 - ◆ ein einziger Datenbankzustand
- Jetzt
 - ◆ Datenbankhistorie
 - Unendliche Folge von Datenbankzuständen
 - **Vergangene** und **zukünftige** Datenbankhistorie
- Drei Anfragearten

Anfragearten

- **Momentananfrage**
 - ◆ Anfrage zu festem Zeitpunkt t
 - ◆ Auswertung auf aktueller unendlicher Historie
- **Kontinuierliche Anfrage**
 - ◆ Folge von Momentananfragen
 - ◆ **aber** falls möglich nur einmalige Auswertung
 - Antworttupel enthalten Gültigkeitsintervall
 - Neuauswertung nur bei Änderung der Datenbasis
- **Persistente Anfrage**
 - ◆ Anfrage gegen vorrätig gehaltenen alten Datenbasiszuständen

FTL - Beispiel

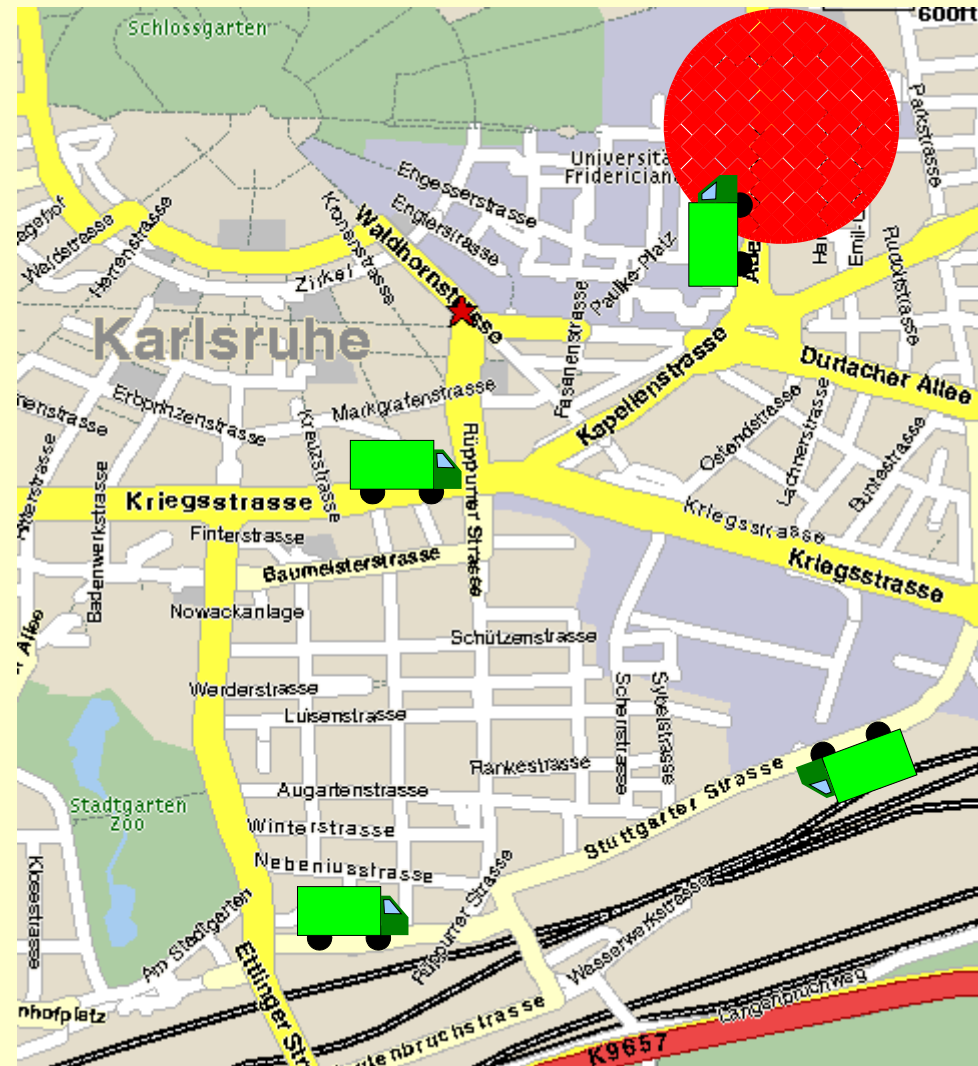
A1 Welcher Wagen befindet sich in der Nähe vom Informatikgebäude?

i = Informatikgebäude

RETRIEVE o

WHERE $o.Type = 'LKW'$

AND $DIST(o, i) \leq 500m$



FTL – Beispiel II

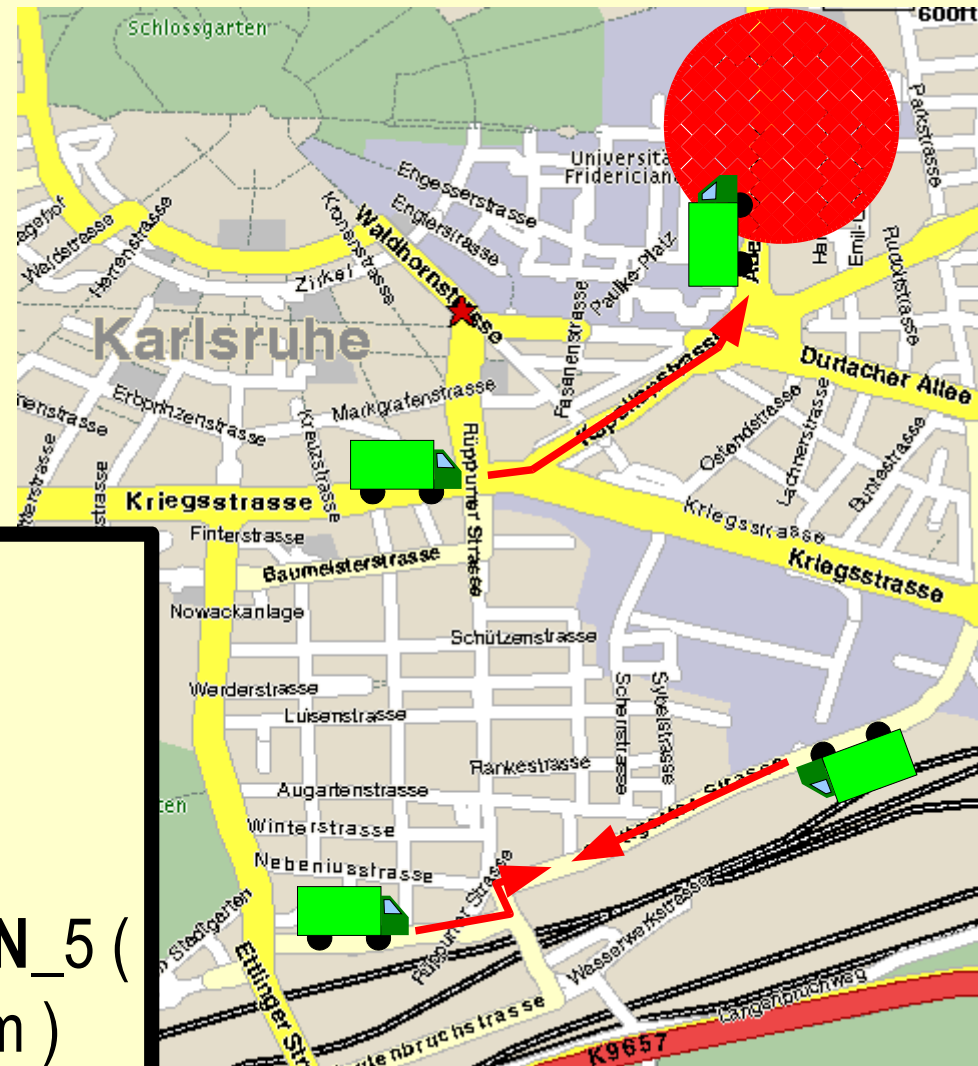
A1 Welcher Wagen befindet sich innerhalb der nächsten 5 min in der Nähe vom Informatikgebäude?

i = Informatikgebäude

RETRIEVE o

WHERE $o.Type = 'LKW'$

AND **EVENTUALLY_WITHIN_5** (
 $DIST(o, i) \leq 500 \text{ m}$)



FTL – Beispiel III

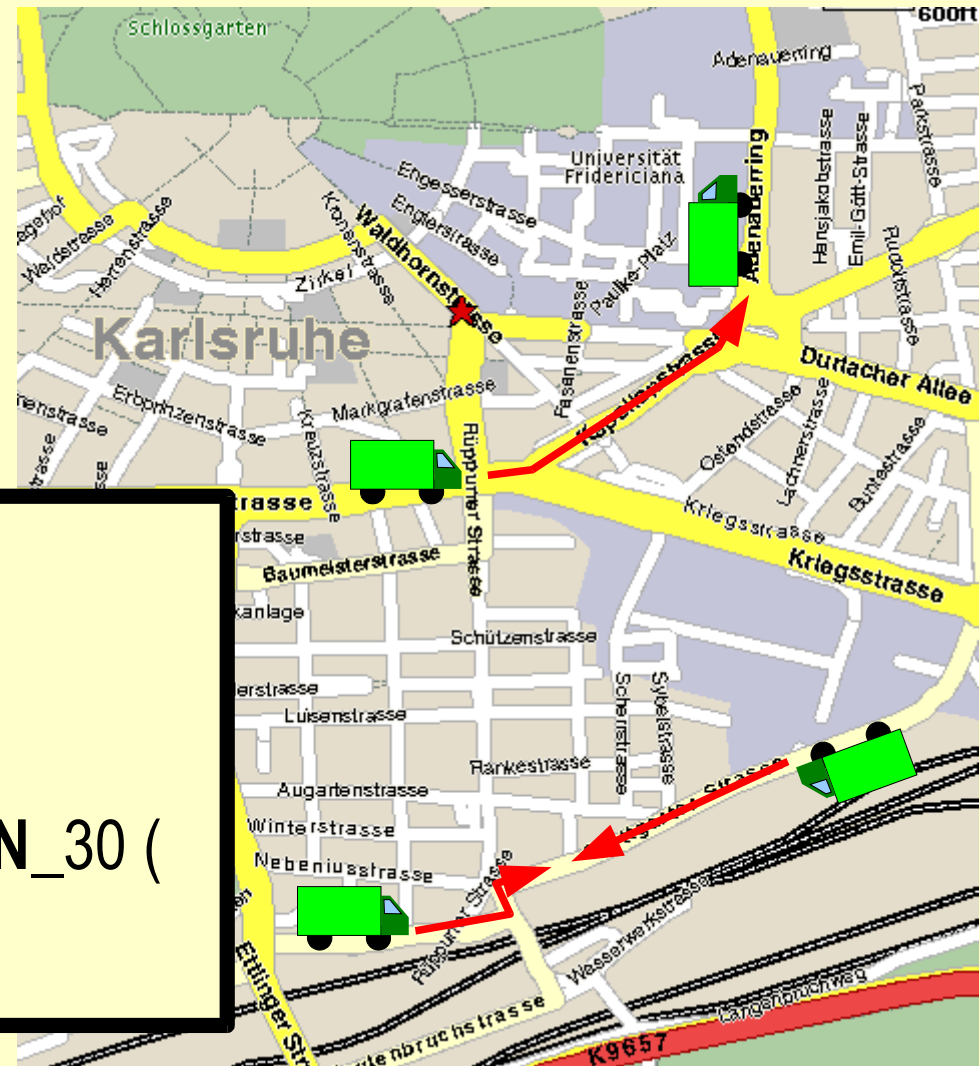
A2 Welche Lieferwagen werden sich in den nächsten 30 min begegnen?

RETRIEVE o, v

WHERE o.Typ = v.Typ = 'LKW'

AND o != v

AND **EVENTUALLY_WITHIN_30** (
DIST (o, v) = 0 m)



FTL – Beispiel IV

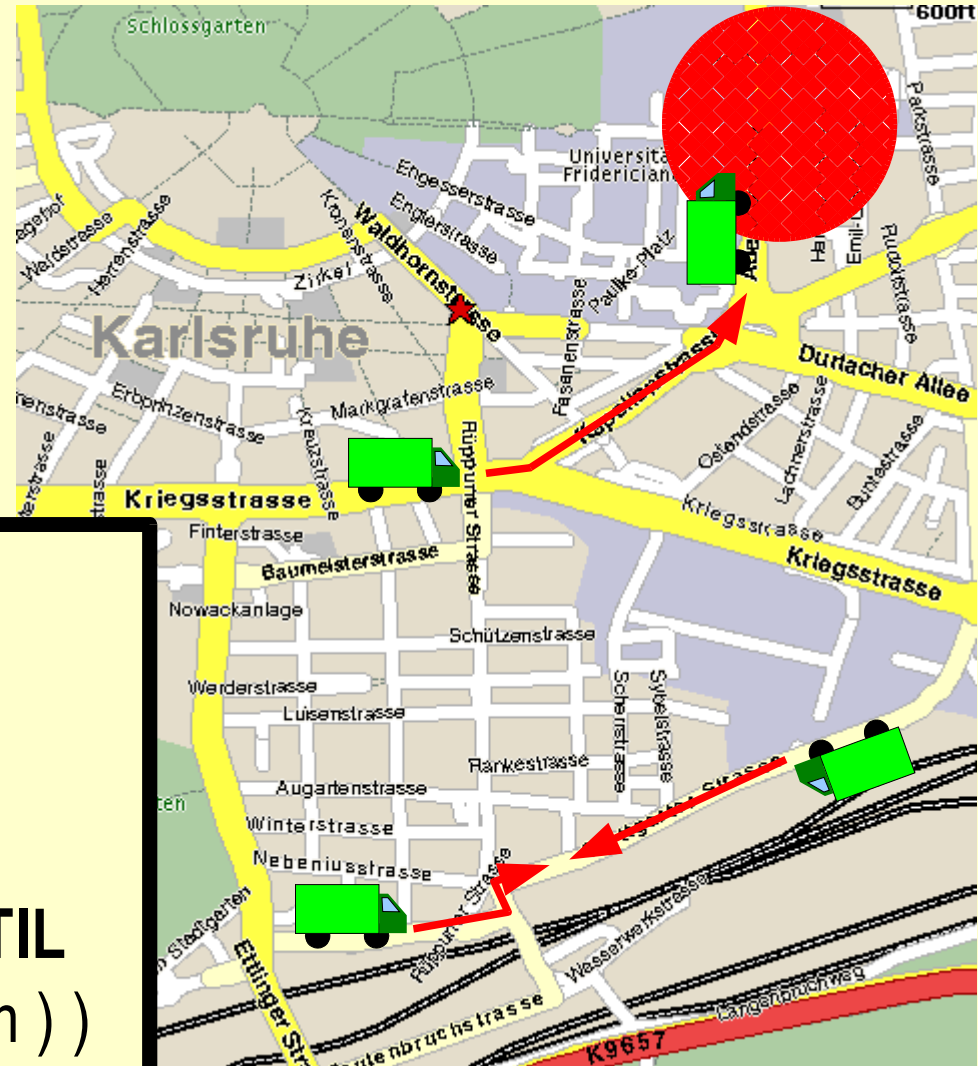
A3 Welche Lieferwagen sind unbeladen bis sie den Fasanengarten erreichen?

i = Informatikgebäude

RETRIEVE o

WHERE $o.Type = 'LKW'$

AND $(o.beladen = false$ **UNTIL**
DIST $(o, i) = 0 \text{ m}))$



Anfragesprache FTL

- Future Temporal Logic
- Anfrageaufbau
 - RETRIEVE o [, v [, ...]]
 - WHERE *Prädikat*(o , v , ...)
- Von Wolfson u.a.
- Räumliche und zeitliche Operatoren

FTL - Operatoren

- Räumliche Operatoren
 - ◆ DIST
 - ◆ INSIDE
- Boolsche Operatoren
 - ◆ AND
 - ◆ OR
 - ◆ etc.

FTL – Operatoren II

- Zeitliche Operatoren
 - ◆ f UNTIL g
 - ◆ NEXTTIME f
 - ◆ davon abgeleitet
 - EVENTUALLY f
 - ALWAYS f

FTL - Zeitliche Operatoren

- EVENTUALLY f

Prädikat f erfüllt

Prädikat f nicht erfüllt



- ◆ EVENTUALLY_WITHIN_t f
- ◆ EVENTUALLY_AFTER_t f

FTL – Zeitliche Operatoren II

- ALWAYS f

Prädikat f erfüllt

Prädikat f nicht erfüllt



- ◆ ALWAYS_FOR_t f

Zusammenfassung

- Probleme der Standard-DBMS
- Datenmodelle für Moving Objects Databases
 - ◆ MOST (Wolfson u.a.)
 - ◆ Forlizzi u.a.
- Räumliche und zeitliche Anfragesprache
 - ◆ FTL – Future Temporal Logic

Ende

**Vielen Dank für Eure
Aufmerksamkeit!**