A Time-Evolving Data Structure Scalable between Discrete and Continuous Attribute Modifications

Martin Danielsson & Rainer Müller

Tobias Pulls & Lars-Olof Moilanen

Introduction & Motivation

- Commonly systems only know the current state, no way to reconstruct previous states
- Reconstruction of previous states necessary for replay or sift through

Example 1

- A company's ERP system handles employees, data changes over time
- Needs to track for example salaries and positions of the employees

Example 2

- FDR's (Flight Data Recorders) records different operating conditions during a flight

Hastighet, höjd, riktning etc. -> Dataanimerad video av kraschförloppet

Example 3

Example 3

- Presentations with moving objects

Time-evolving data structures

- Martin Danielsson & Rainer Müller presents the *relational approach*

The problem

- Object space, empty from start
- Objects consists of a time-invariant key and several time-variant attributes
- Objects are compound and encapsulated
- Time and time intervals are implicitly defined (through R^+)

R^+ = en eller flera relationer
invariant = oföränderlig
variant = föränderlig
Compound: Sammansatta
Encapsulated: Kan bestå av godtyckliga datatyper

Terminology

- Container (Mandatory object attribute)
- Root container
- Active object
- An object’s properties

Container:	ett sätt att gruppera objekt
Root container:	Container högst upp i hierarkin, krävs minst en
An object’s properties:	Nyckel + attribut
Active:	Ett object som tillhör en container

Object space modifications

- Add
- Delete
- Change

Add: Lägga till det i en container tillhörande objektmängden
Delete: Ta bort ett objekt från en container
Change: Ändra ett attribut (ej nyckel)

Two types of changes

- Discrete change
- Continuous change

Discrete change: Ett attribut skrivs; ex: sätt lön till 70 000 SEK
Continuous change: Ett attribut ändras över ett tidsintervall; öka lön med 10%/månad i ett år

Demands

- Synchronizable
- Random real-time access

Synchronizable: Alla tillstånd skiljer sig från tillståndet innan, ligger i tidsföljd

Random real-time access: Absoluta tiden (i ms) för att återskapa ett tillstånd < $\mu_{a,S}$

Event Queue & Object List

- An event: (timestamp, number of active objects, list of active object identifiers)
- Object List
- Handles continuous changes by making small discrete changes

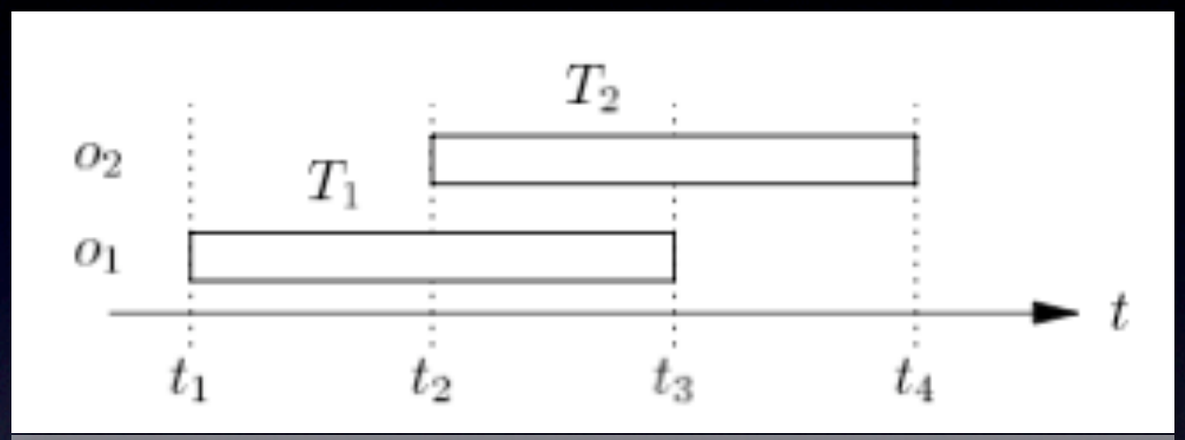
Object List: Alla objekt som existerat i systemet
Changing an object: Gör ett event som lägger till en kopia med det/de ändrade attributen, radera det gamla objektet

The relational approach

- Handles discrete changes by introducing a special continuous change
- A continuous change is a parameterized function on an object that acts like a transition from one state to another over a period of time

Transition recording

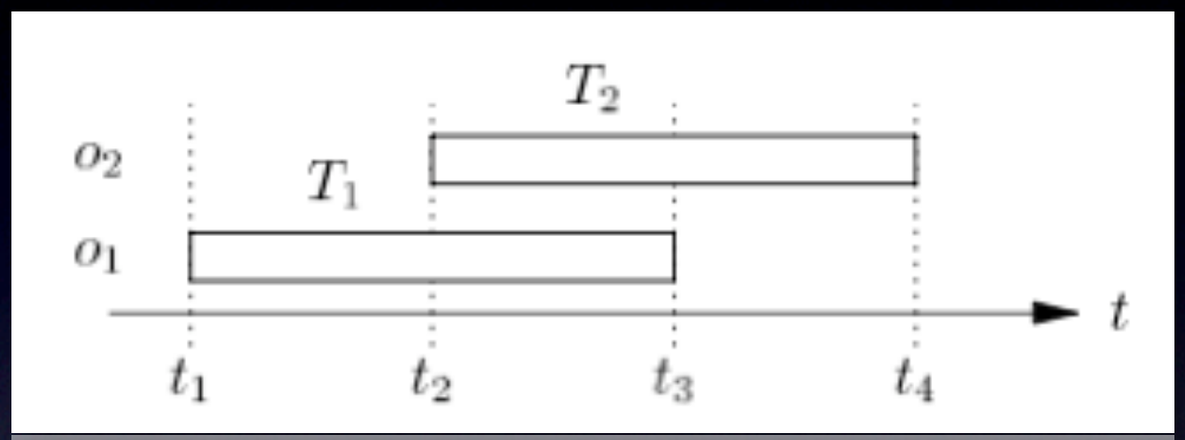
- T-List
- P-List
- TAS
- PAS



T-List: Listan innehållandes alla övergångar
P-List: Listan innehållandes alla objekts attribut
TAS: En ström med ögonblicksbilder av alla aktiva övergångar (trans. access stream)
PAS: En ström med ögonblicksbilder av alla aktiva objekts attribut (properties access...)

Transition recording

- T-List
- P-List
- TAS

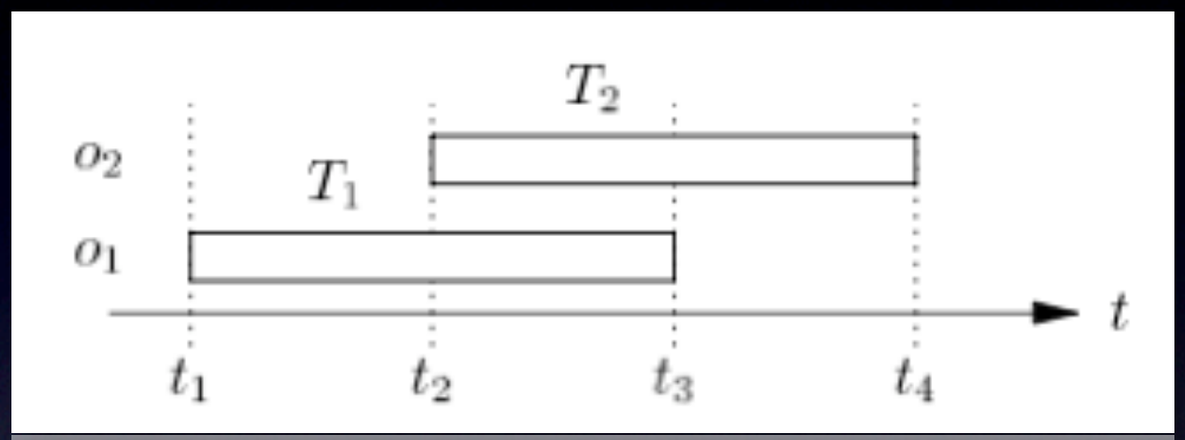


- PAS t_1: TAS={T_1}
  PAS={$p_1^{(0)}$, $p_2^{(0)}$}

T-List: Listan innehållandes alla övergångar
P-List: Listan innehållandes alla objekts attribut
TAS: En ström med ögonblicksbilder av alla aktiva övergångar (trans. access stream)
PAS: En ström med ögonblicksbilder av alla aktiva objekts attribut (properties access...)

Transition recording

- T-List
- P-List
- TAS

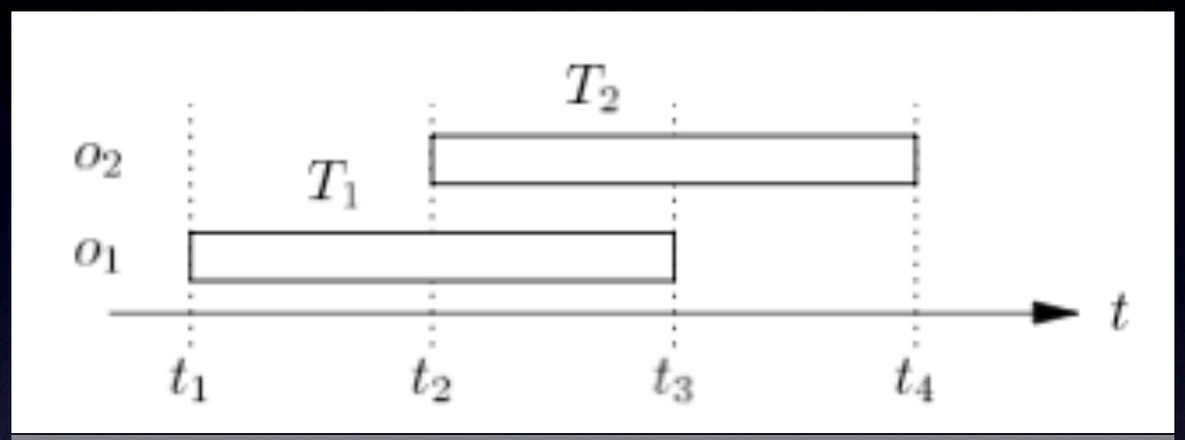


- PAS t_1: TAS={T_1} t_2: TAS={T_1, T_2}
  PAS={$p_1^{(0)}$, $p_2^{(0)}$}

T-List: Listan innehållandes alla övergångar
P-List: Listan innehållandes alla objekts attribut
TAS: En ström med ögonblicksbilder av alla aktiva övergångar (trans. access stream)
PAS: En ström med ögonblicksbilder av alla aktiva objekts attribut (properties access...)

Transition recording

- T-List
- P-List
- TAS



- PAS t_1: TAS={T_1}
  PAS={$p_1^{(0)}$, $p_2^{(0)}$}

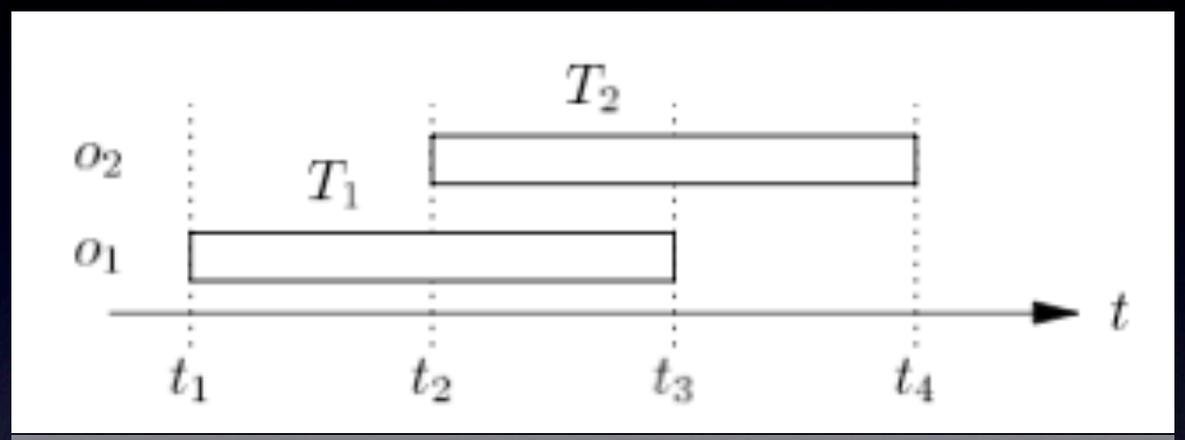
  t_2: TAS={T_1, T_2}

  t_3: TAS={T_2}
  PAS={$p_1^{(1)}$, $p_2^{(0)}$}

T-List: Listan innehållandes alla övergångar
P-List: Listan innehållandes alla objekts attribut
TAS: En ström med ögonblicksbilder av alla aktiva övergångar (trans. access stream)
PAS: En ström med ögonblicksbilder av alla aktiva objekts attribut (properties access...)

Transition recording

- T-List
- P-List
- TAS



- PAS t_1: TAS={T_1} PAS={$p_1^{(0)}$, $p_2^{(0)}$}
  t_2: TAS={T_1, T_2}
  t_3: TAS={T_2} PAS={$p_1^{(1)}$, $p_2^{(0)}$}
  t_4: TAS={ } PAS={$p_1^{(1)}$, $p_2^{(1)}$}

T-List: Listan innehållandes alla övergångar
P-List: Listan innehållandes alla objekts attribut
TAS: En ström med ögonblicksbilder av alla aktiva övergångar (trans. access stream)
PAS: En ström med ögonblicksbilder av alla aktiva objekts attribut (properties access...)

Adding and deleting

- Adding an object ⇒ create PAS entry
- Deleting an object ⇒ create PAS entry

PAS-entry ger objektens attribut om någon ändring sker måste attributen uppdateras

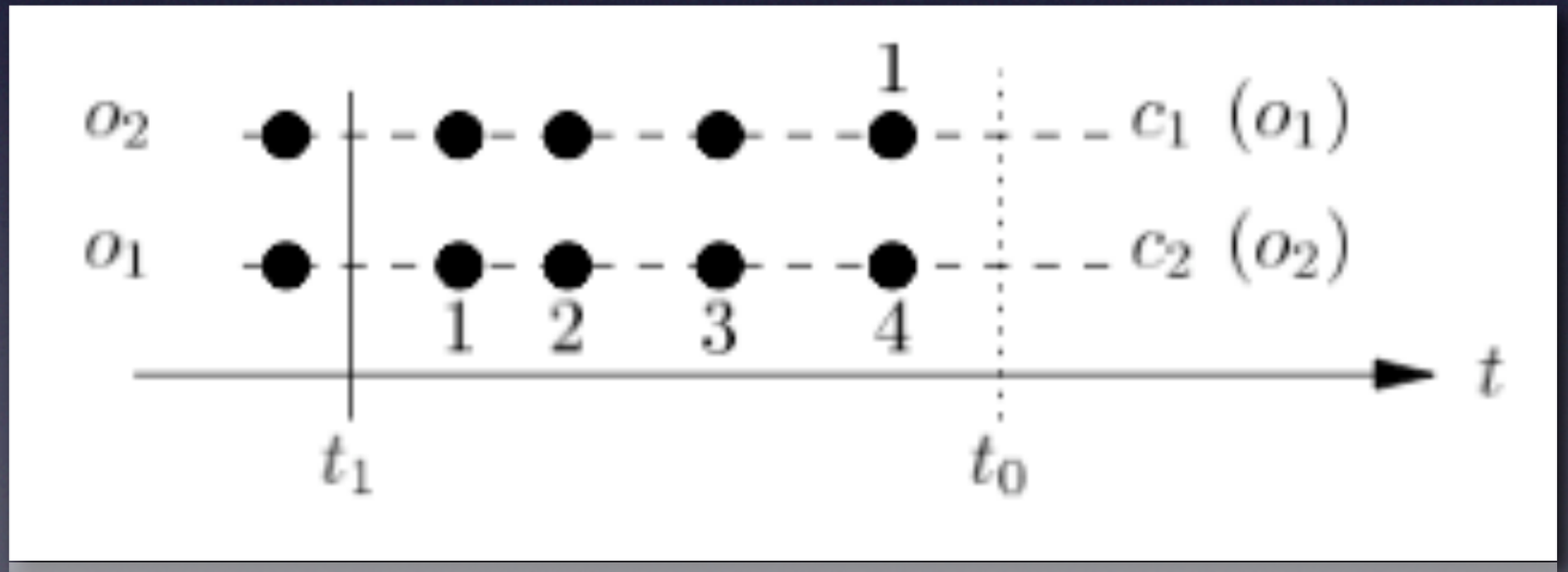
Complete transitions

- A complete transition consists of (time, attribute value) pairs of a specific attribute on a specific object
- Can be overwriting
- Or modifying

Diskret ändring, en komplett övergång per objekt och attribut,
men en komplett övergång kan bestå av flera (tid, attributvärde)-par
Overwriting: sätt ett attributvärde oberoende av dess tidigare värde; ex: sätt lön till...
Modifying: sätt ett attributvärde beroende av dess tidigare värde; ex: öka lön med...

Reconstruction

- Get PAS
- Analyze TAS



Get PAS: Ladda alla attributvärden, vid t_1
Get TAS: Ta hänsyn till aktiva transaktioner

Complexity/Performance

- Finding TAS: O(logN_T)
- Finding PAS: O(logN_P)
- Applying the properties in PAS: O(N_O)
- Reconstructing objects:
  - Continuous transitions: O(M_T)
  - Complete transitions: O(logM_C + M_C)
- Total: O(logN_T + logN_P + N_O + M_T)

sätta ett attribut: O(1) – konstant tid
O(N_O): Linjärt mot maxantalet objekt
O(M_T): Sätta M_T övergångsattribut
O(logM_C + M_C): log: hitta rätt övergång, M_C: sätta M_C attribut; M_C konstant (maxantalet (tid,värde)-par

Benchmarks

Test	**1**			**2**			**3**		
Stream									
Number of objects	25	50	100	25	50	100	25	50	100
Number of transitions	2	2	2	2	2	2	2+23	2+48	2+98
Transition length (ms)	10000	10000	10000	500	500	500	23:300000 2: 500	48: 300000 2: 500	98: 300000 2: 500
Transition offsets (ms)	0	0	0	250	250	250	23: 0 2: 250	48: 0 2: 250	98: 0 2: 250
Access time									
Average (ms)	6,1	11,1	20,1	6,2	11,3	20,0	7,6	14,1	23,5
Maximum (ms)	7,0	12,0	24,0	7,0	13,0	25,0	9,0	17,0	24,0
Bit rate uncompressed (kbit/s)	0,95	1,45	2,10	9,1	12,4	16,8	14,1	23,3	38,4
Bit rate compressed (kbit/s)	0,21	0,28	0,39	1,62	1,73	1,92	1,78	1,90	2,23

Tests run on 200MHz machine

Summary

- Event Queue and Object List vs the Relational Approach
  - EQ/OL makes continuous changes discrete
  - The RA makes discrete changes continuous
  - RA focuses on random real-time access

EQ/OL: nästan optimalt vid t.ex. statiska presentationsslides
RA: klarar att modellera även andra situationer på ett bättre sätt