Real-Time Course

Transaction based temporal model for Real-time databases
Real-time data

Data used in classical administration system
“bank account”
-> represent status of constant real-world

Data used in real-time systems
“current position” (flight control system)
“outside temperature” (home heating system)
“water level” (mine pumps)
“stock price PHILIPS” (stock market investor)
-> represent status of changing world

Normal consistency criteria do not suffice
-> the database has to “refresh” its content to accurately represent the real world
Temporal data

Distinguish “temporal” and “non-temporal” data

New consistency criteria?

-data items equal status of the world
  - impossible to realize, system is not instantaneous
-data items equal status of the world within error margin
  - impossible without notion of distance
  - impossible without bounded change rate
-data items follow the change of the real-world within bounded time
Temporal model

Temporal item X has an absolute validity duration AVD(X)
-> transactions never read values of X older than AVD(X)

Example

Suppose the mine-pump system has to react to a change in water-level in 10 seconds

Suppose AVD(water-level) = 5

The reaction should take place within 10 –5 = 5 seconds
Temporal model

- Applications issue real-time transactions to database
- Global clock $\tau_{\text{now}}$ used by the database
- Applications specify real-time demands using $\tau_{\text{now}}$
Temporal model

Database contains set of data-items
Data-items can be atomically *read* and *written*
A data-item $X$ is represented with an instance $x$

An instance $x$ is unique and has a value $x.v$ and some control information (defined later)

When $X$ is written a new instance $x$ is created and old instance is removed

**Two types of data-items**
- non-temporal items
- temporal items
Temporal model

Trivial: as time progresses instances grow old. The *time of measurement* $x.\tau$ of instance $x$ defines the moment that $x.v$ accurately reflected the real world.
Age of instance $x$ is $t_{\text{now}} - x.\tau$

Two types of temporal items
- sensor items
  $x.\tau$ is defined by transaction that writes $x$
- derived items
  these items were never measured
Derivation rules

Relation with the time of measurement of input

Semantic information about derivation is available
- X is time critical
- Q is not time critical
**Derivation rules**

No semantic information available

\[ X \ ? \]
\[ Q \ ? \]

Information hiding in derivation trees

\[ X \]
\[ Y \]
\[ W \]
\[ Z \]

June 2004
Temporal sets

$t$ specifies temporal constraints for temporal items in $t$.Spec

$t$.IN set of temporal items that $t$ reads
$t$.OUT set of temporal items that $t$ writes
$t$.Spec set of temporal items, constrained by $t$

t specifies maximal age $t.m(X)$ for each item $X$ in $t$.IN
t specifies maximal age $t.m(Q)$ for each item $Q$ used to derive $X$ in $t$.IN
**Transaction types**

- **User transactions** do not write temporal data $t.\text{OUT}=\emptyset$ and have effects outside the database or write non-temporal data.
- **Refresh transactions** have a supporting role.

---

Diagram showing the relationship between transaction types and age categories.

- **Single writer**
  - Age $< 10$

- **Serialized writer**
  - Age $< 10$
Refresh Transactions

• Sensor transactions $t.IN = \emptyset$

• Derive transactions
Absolute consistency

Database *absolutely temporally consistent* if
- all transactions satisfy all temporal constraints

For all $t$: for all $X$ in $t$.Spec
$t$ reads an instance $x$ with $t_{\text{now}} - x.t \leq t.m(X)$

Requirement holds at start of transaction
Relative consistency

For each $t$, a set $t rtc$ of relative temporal constraints, $rc$ is specified.

Constraint $rc$ consists of:
- a subset $rc.s$ in $t.Spec$ of data-items
- a maximal age difference $rc.m$

Instances read by $t$ must have $t.o.m$ such that
For all $X,Y$ in $rc.s$: $|x.t - y.t| \leq rc.m$

If all $rc$ of all transactions are satisfied, the database is relatively temporally consistent
Database and transactions

Sensor transactions

Derive transactions

Actuator Transaction 1

Actuator Transaction 2
TOM functions

- Sensor transaction: worst case assumption
  \[ tom(t, x) = \text{start}(t) \]
- Derive transaction: worst case assumption
  \[ tom(t, y) = \min_{x \in t.in} x.T \]
- One critical input \( X \)
  \[ tom(t, y) = x.T \]
- Proportional influence
  \[ tom(t, y) = \sum_{x \in t.in} x.T / |t.in| \]
- Prediction model
  transaction \( t \) uses instance \( x \) to create instance \( y \)
  \textit{which} is “measured” in the future: \( y.T > T_{\text{now}} \)
  \[ tom(t, y) = x.T + \delta \]
Design decisions

• **First decision**: all requirements are met at all times
  \[ ma_X = \min( \min_{t,X: X \in t.\text{Spec}} t.\text{ma}(X), \min_{t: rc \in t.\text{rtc and } X \in rc.s} rc.m) \]

• **Second decision**: refresh transactions are periodic

• **Sensor transaction**: if \( t \) writes \( X \), then \( t.p + t.d \leq ma_X \)

  - \( t.p \) is period of \( t \)
  - \( t.d \) is deadline of \( t \)
Derived age calculation

**Derive transactions** read instances that have aged
worst case:
Reaction time system is
\[ ma_x + t.p + t.d \]

Last refresh of X
Transaction reads X

Additional age \( aa_Y \) is maximum of \( t.start - tom(t,y) \)

Example \( t \) reads \( X \) and \( Q \) to write \( Y \)
\[ tom(t,y) = 0.5 x.t + 0.5 q.t \]
If \( ma_x \) in \([0,8]\) and \( ma_Y \) in \([0,6]\) then \( aa_Y = 7 \)

\[ t.p + t.d \leq ma_Y - aa_Y \]
Derived age calculation(2)

This holds for each $Y$ that transaction $t$ writes

$$t.p + t.d \leq \min(Y \text{ in } t.OUT: ma_Y - aa_Y)$$
Example

Show Warning:  SW.p=15 and SW.m(D)=15
Show Count:  SC.p=60 and SC.m(C)=60

tom(Detect, D) = r.τ  
tom(Count, C) = min(r.τ, f.τ) 
tom(Radar image, R) = Radar image.start 
tom(Flight info, F) = Flight info.start
**Example (2)**

**Design choice** requirements on detect and count

Detect\(m(R) = 10\)
\(rc.s = \{R,F\}\) with \(rc.m=30\) and \(rc\) in detect.rtc
Count\(m(R) = 40\)
Count\(m(F) = 40\)
Example (3)

\[\text{ma}_D = \text{SW.m}(D) = 15\]
\[\text{ma}_C = \text{SC.m}(C) = 60\]
\[\text{ma}_R = \min(\text{Count.m}(R), \text{Detect.m}(R), \text{rc.m}) = 10\]
\[\text{ma}_F = \min(\text{Count.m}(F), \text{rc.m}) = 30\]

\[\text{tom}(\text{Detect}, D) = r.\tau, \text{ma}_R = 10\]
\[\text{tom}(\text{Count}, C) = \min(r.\tau, f.\tau), \text{ma}_R = 10, \text{ma}_F = 30\]

\[\text{RC.s} = \{R, F\}\]
\[\text{rc.m} = 30\]
Example (4)

\[ \text{detect.p + detect.d} \leq \text{ma}_D - \text{aa}_D = 5 \]
\[ \text{count.p + count.d} \leq \text{ma}_C - \text{aa}_C = 30 \]
\[ \text{radar image.p + radar image.d} \leq \text{ma}_R = 10 \]
\[ \text{Flight info.p + flight info.d} \leq \text{ma}_F = 30 \]