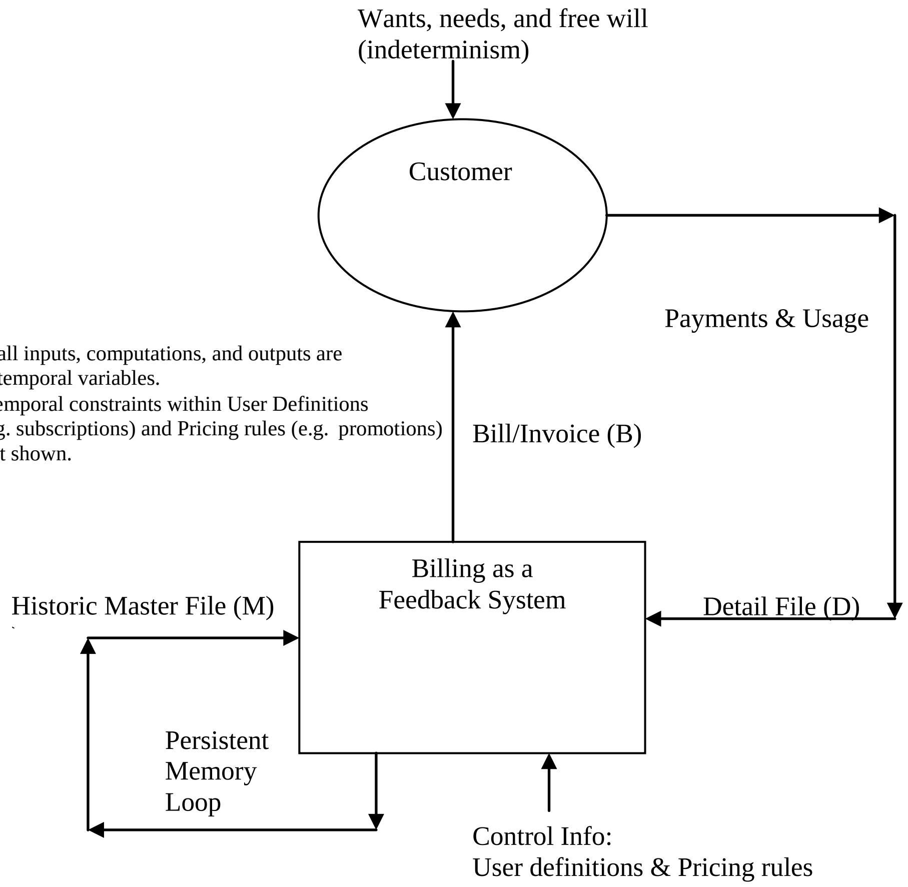
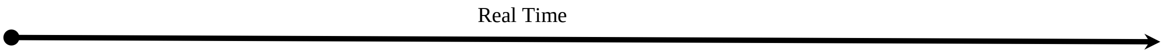
Data Processing as Process Control

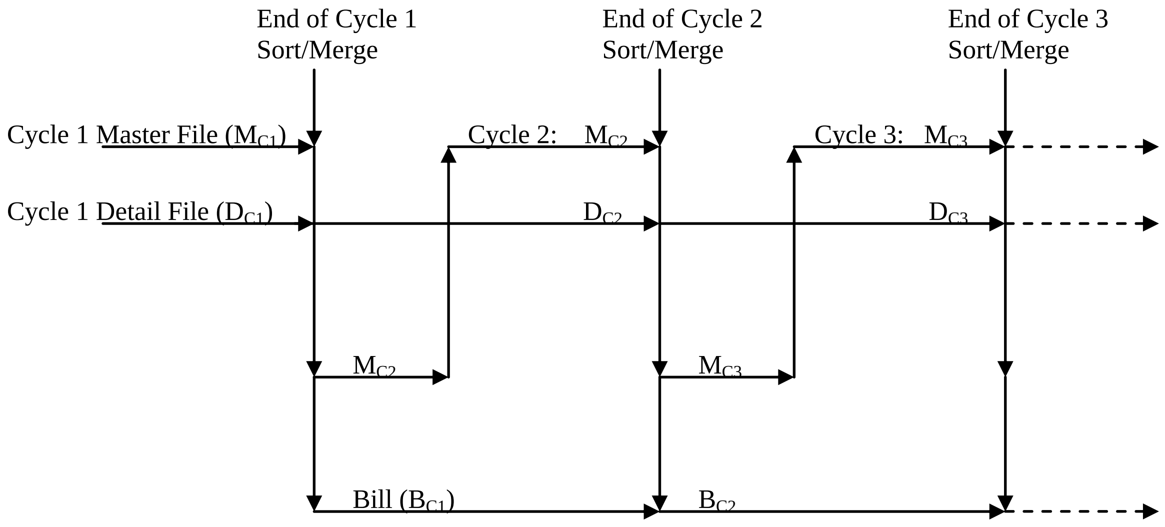


Note:
- all inputs, computations, and outputs are temporal variables.
- temporal constraints within User Definitions (e.g. subscriptions) and Pricing rules (e.g. promotions) not shown.

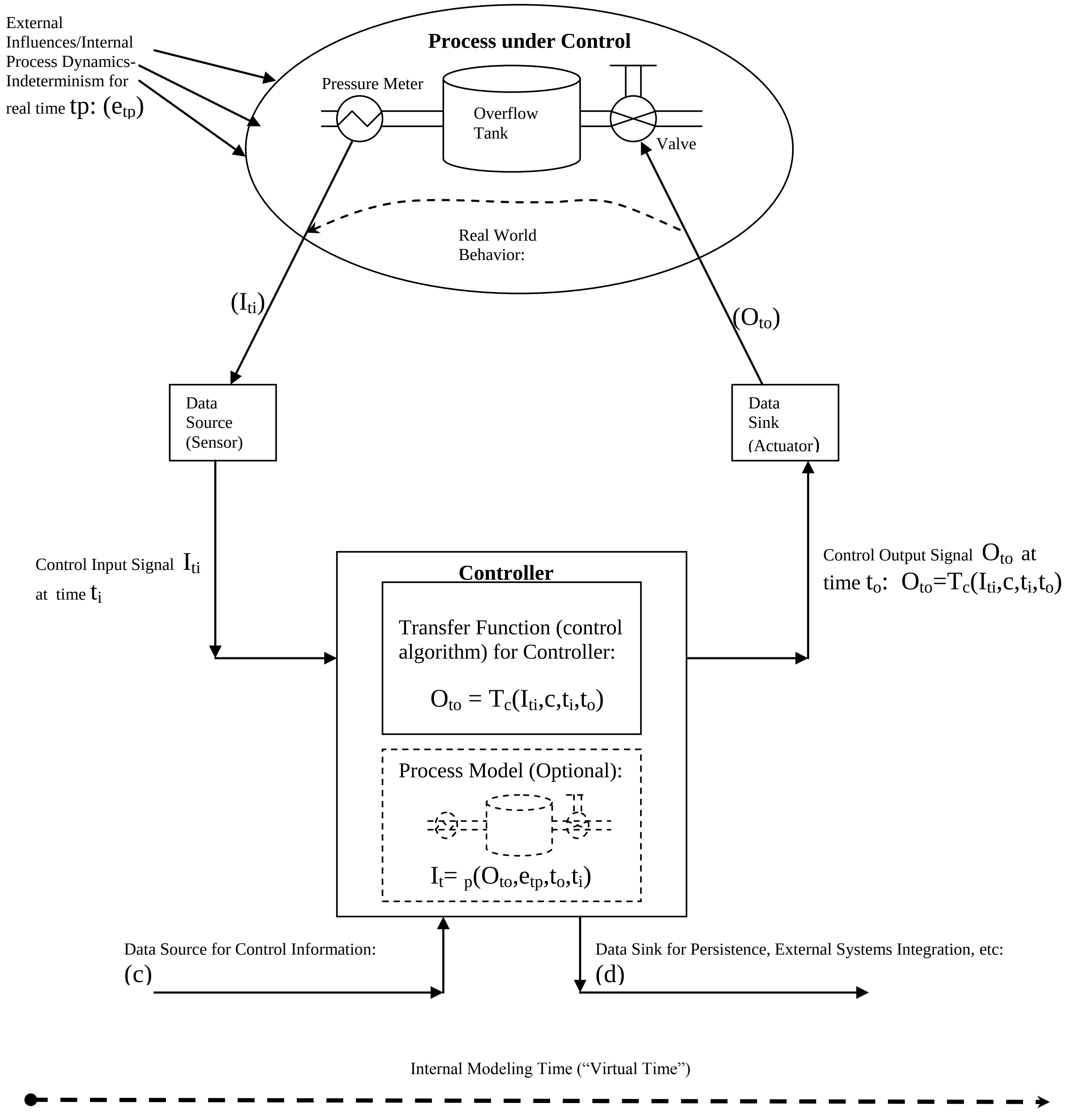
The Classic Data Processing Pattern: Temporal Sort/Merge



Sort/Merge at End of Billing Cycle uses Virtual Time to recreate previous real time interval and sets up the next:


End of Cycle 1 Sort/Merge
End of Cycle 2 Sort/Merge
End of Cycle 3 Sort/Merge
Cycle 1 Master File (M_{C1})
Cycle 2: M_{C2}
Cycle 3: M_{C3}
Cycle 1 Detail File (D_{C1})
D_{C2}
D_{C3}
M_{C2}
M_{C3}
Bill (B_{C1})
B_{C2}

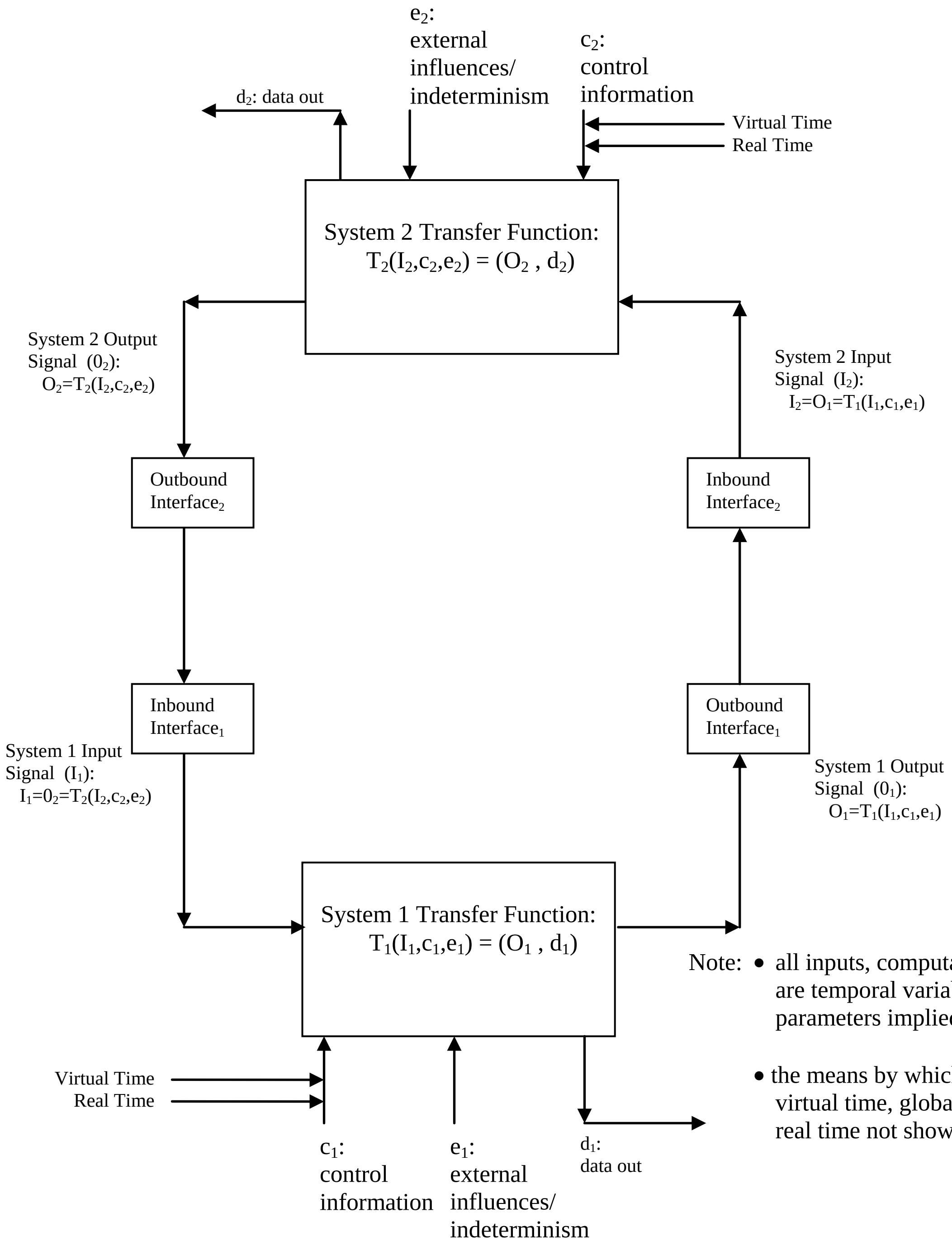
Basic Servo Control Model -Continuous Process Control (Closed Loop Example)


External Influences/Internal Process Dynamics-Indeterminism for real time tp: (e_{tp})
Process under Control
Pressure Meter
Overflow Tank
Valve
Real World Behavior:
(I_{ti})
(O_{to})
Data Source (Sensor)
Data Sink (Actuator)
Control Input Signal I_{ti} at time t_i
Control Output Signal O_{to} at time t_o: $O_{to}=T_c(I_{ti},c,t_i,t_o)$
Controller
Transfer Function (control algorithm) for Controller:
$O_{to} = T_c(I_{ti},c,t_i,t_o)$
Process Model (Optional):
$I_t= {}_p(O_{to},e_{tp},t_o,t_i)$
Data Source for Control Information:
(c)
Data Sink for Persistence, External Systems Integration, etc:
(d)
Internal Modeling Time ("Virtual Time")

Note:

- the Transfer Function $T_c()$ relates control loop inputs to outputs (and optional Transfer Function $T_p()$ models the physical process).
- all inputs, computations, and outputs are temporal variables, i.e. time places a crucial role in the application processing and data semantics.
- the relationship between real time and modeled virtual time is governed by “real time constraints” of the application:
    - generally in continuos process control, real time constraints fall out from the laws of physics;
    - where as in data processing applications, real time constraints generally come from SLAs and financial rules.

Dynamic Feedback: Two (or more) Interconnected Systems


e_2: external influences/ indeterminism
c_2: control information
d_2: data out
Virtual Time
Real Time
System 2 Transfer Function: $T_2(I_2,c_2,e_2) = (O_2 , d_2)$
System 2 Output Signal (0_2): $O_2=T_2(I_2,c_2,e_2)$
System 2 Input Signal (I_2): $I_2=O_1=T_1(I_1,c_1,e_1)$
Outbound Interface$_2$
Inbound Interface$_2$
Inbound Interface$_1$
Outbound Interface$_1$
System 1 Input Signal (I_1): $I_1=0_2=T_2(I_2,c_2,e_2)$
System 1 Output Signal (0_1): $O_1=T_1(I_1,c_1,e_1)$
System 1 Transfer Function: $T_1(I_1,c_1,e_1) = (O_1 , d_1)$
Virtual Time
Real Time
c_1: control information
e_1: external influences/ indeterminism
d_1: data out

Note:
- all inputs, computations, and outputs are temporal variables (temporal parameters implied, but not shown).
- the means by which to coordinate local virtual time, global virtual time, and real time not shown.

Each Input/Output variable is non-linearly defined to refer (feedback) to itself:

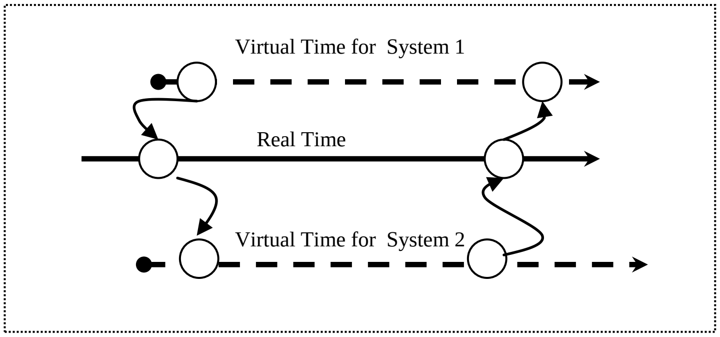
$\mathbf{O_1}= T_1(I_1,c_1,e_1)$
$= T_1(O_2,c_1,e_1)$
$= T_1(T_2(I_2,c_2,e_2),c_1,e_1)$
$= T_1(T_2(\underline{\mathbf{O_1}},c_2,e_2),c_1,e_1)$

$\mathbf{O_2}= T_2(I_2,c_2,e_2)$
$= T_2(O_1,c_2,e_2)$
$= T_2(T_1(I_1,c_1,e_1),c_2,e2)$
$= T_2(T_1(\underline{\mathbf{O_2}},c_1,e_1),c_2,e2)$

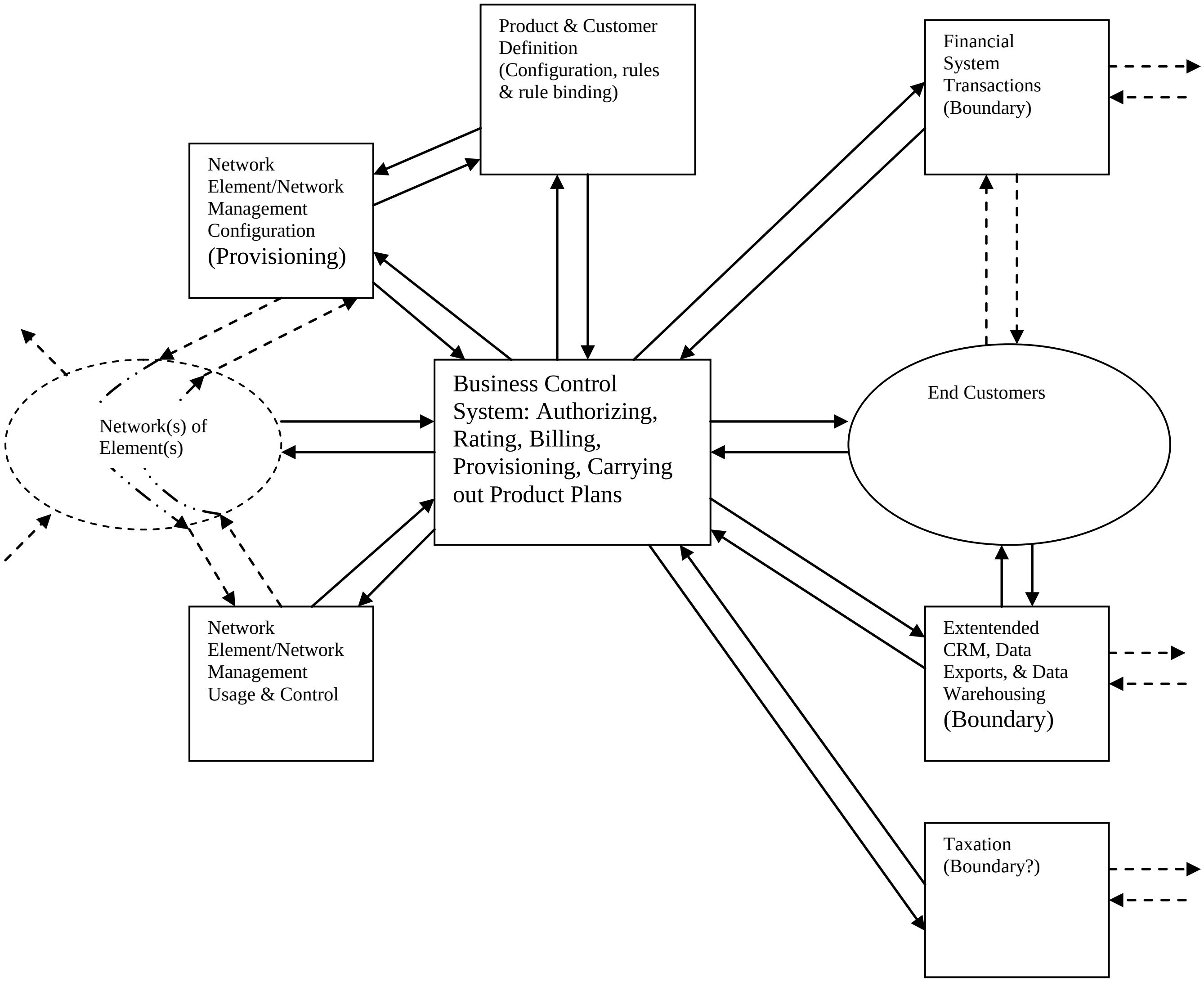
$\mathbf{I_1} = T_2(I_2,c_2,e_2)$
$= T_2(O_1,c_2,e_2)$
$= T_2(T_1(\underline{\mathbf{I_1}},c_1,e_1),c_2,e_2)$

$\mathbf{I_2} = T_1(I_1,c_1,e_1)$
$= T_1(O_2,c_1,e_1)$
$= T_1(T_2(\underline{\mathbf{I_2}},c_2,e_2),c_1,e_1)$

rjd 10/5/2007


Virtual Time for System 1
Real Time
Virtual Time for System 2

A Major Architectural Pattern:
Billing and Rating as a Control System for Business Processes –
Extending The Process Control View of Billing and Rating Further into Feedback Loops of the Business Domain


Product & Customer Definition (Configuration, rules & rule binding)
Financial System Transactions (Boundary)
Network Element/Network Management Configuration (Provisioning)
Network(s) of Element(s)
Business Control System: Authorizing, Rating, Billing, Provisioning, Carrying out Product Plans
End Customers
Network Element/Network Management Usage & Control
Extentended CRM, Data Exports, & Data Warehousing (Boundary)
Taxation (Boundary?)

1.1 Analysis Questions For each Major Functional Element (F.E.) in the Feedback Loops

i) What are its inputs: sensed values from its environment (control), and commands and responses from partner F.E.s (feedback)

ii) What are its outputs: control values and commands to entities in its environment, further commands and responses to partner F.E.s

iii) What is it’s Transfer Function: what does it do, what are it’s control objectives, what are it’s rules?

iv) In which extended flows and feedback loops of data, commands, and responses does it participate?

Note: The above questions i) thru iv) ask what application specific protocol(s) does the entity speak (or what portion of a greater multinodal protocol)?

v) How does each Functional Element determine the temporal aspects of its data, rules, commands, and responses:

- What is its unique (internal) sense of computation time (Virtual Time)?
- What is the relationship of internal Virtual Time(VTime) to Real Time (RTime)?
- How does it synchronize its sense of VTime with the RTime time of incoming and outgoing data, commands, and responses?
- How does it synchonize its sense of collective/global RTime with others (e.g. NTP)?
- How does it synchronize its sense of collective/global VTime in extended flows and feedback loops?

vi) What are the temporal constraints of its data, rules, commands and responses, including extended flows and feedback loops?

- RTime to RTime: temporal constrains related to RTime, e.g.
    - ➢ Compute and respond within x Rtime units (e.g. seconds) from the RTime of arrival (aka “hard Real Time”).
    - ➢ Collect all data upon arrival (“latch”) and transmit in a single batch once an hour on the (Rtime) hour.

- Vtime to Vtime: temporal constraints relate to Vtime, e.g.

    - Bill only those portions of the call within this billing cycle (Vtime period), and split the allocation between adjacent billing cycles if need be.

    - Rate the first three minutes of each call at $x/minute, the next 57 minuts at $y/minute, and any more minutes at $y/minute when month end loyalty credits exceed 25 units or YTD billed amount exceeds $2,000.

- Rtime to Vtime: temporal constraints which define or restrict Rtime to Vtime mapping e.g.

    - Record clock time (Rtime) of this call; or convert Eastern Standard Time clock time and record as New Zealand Daylight Time; or convert actual Mars Prime Meridian Time to recorded GMT; or translate actual wall clock Metric Time to internal American Time format…

    - Adjust all arrival times by the computed network clock scew.

    - After 12 hours (Rtime) drop all event records (Vtime) of type X.

- Vtime to Rtime: temporal constraints which define or restrict Vtime to Rtime mapping e.g.

    - Compute pre-paid allowed minutes (Vtime) against the current balance at the rate of 1 minute/$x and send to the Network Call Setup Control Element (which will use the computed Vtime value as a Rtime constraint on the actual call, should one occur).

      Note: In this case Vtime is ahead of Rtime! The allowable duration is in the future.

      Note also: likely the constraints in this example would also include “real time hard” restrictions (Rtime to Rtime): Compute allowed minutes (Vtime) and respond within 5 seconds (Rtime) of request arrival. The network will discard unused authorizations after 30 seconds (further Rtime to Rtime).

    - If at any time (Rtime) during rating calculation (Vtime), total minutes used since last payment exceeds 10,000 minutes, block further access to this account (Rtime), but do not tear down ongoing (Rtime) calls.

    - The estimated billing usage (Vtime) will be no greater than 15 minutes behind arrival (Rtime) of network event reports (Vtime)

- Other classes of temporal constraints: temporal relations / operations / calculations (temporal algebra/calculus), interaction with non-temporal elements, e.g. account structure???

vii) Which temporal constraints are likely to lead to separation of strategies for persistence:

- What are the implication for employing different memory mechanims and memory hierarchies, information sharing through persistence (e.g. common memory, shared files, shared databases),etc.?

- How will the external persistence mechanisms also honor the FE's related temporal constraints including synchornization of VTime in processing rules and the FE's contraints on the relationship between VTime and RTime?

viii) Which temporal constraints permit parallelization and which constrict it? Hence these constraints contribute to determining the relative ability in principle to distribute the processing load, e.g.

- Treat calls made during the same billing cycle (Vtime) but attached (guided) to different liable accounts of unrelated customers as subject to the same set of active pricing policies in effect (Vtime), but fully independent for the purposes of computing (Rtime) actual rating and billing amounts, i.e. in principle, each account could be computed in a separate computational stream (SIMD architecture).

- Stepped prices are to be applied in daily network arrival order (Rtime to Vtime) within 24 hours of mid-night closing (Rtime to Rtime) for Business Control System A. Basically the event to event data dependency introducted by the stepped pricing function restricts the rating calculation to be sequential (unless done by some form of "optimistic computing" with potential roll back or aggragate adjustment). Temporay persistence (buffering) may allow a little bit of further parallization outside of the process of rating each call.

- But for Business Control System B, apply unstepped prices at arrival time (Rtime) of network events (hence in raw network arrival order). Each rating activity could be farmed out in parallel (SIMD), although aggragation (if any for the purposes of rating) would still require eventual reconcilation into a single value.