Data Flow
4.1, 2.3

Batch Sequential Pipeline Systems

Tektronix Case Study: Oscilloscope
☞ 1. OO model
☞ 2. Layered model
☞ 3. Pipe & Filter model
☞ 4. Modified Pipe & Filter model
☞ Formalization of Oscilloscope

Lawrence Chung

Batch Sequential Pipeline Systems

"systems where data flows linearly thru a sequence of discrete processing steps"

☞ Early database applications

✦ transactions were collected into large batches

```
Open Acct Acct# 972883
Deposit $100 into Acct# 972883
Withdraw $1 from Acct# 972883
Withdraw $2 from Acct# 972883
Close Acct Acct# 214883
...```

✦ a small number of large stand-alone programs performed sequential updates on flat (unstructured) files.

- accept input
- parse input into transactions
- check validity (e.g., name field: d pha, amount: x xxx x.x)
- filter out invalid transactions
- sort valid transactions
- process each transaction against master files (e.g., CUST, ACCT)
- filter out invalid transactions
- process periodic reports

Lawrence Chung
**Batch Sequential Pipeline Systems**

- **input**
  - **online**
    - accept input
    - parse input into transactions
  - **edit**
    - check validity (e.g., name field: alpha)
    - filter out invalid transactions
  - **sort**
    - sort valid transactions
  - **update**
    - process each transaction against master files
    - filter out invalid transactions
  - **print**
    - produce periodic reports

- **Constraints:**
  - processes run in a fixed sequence; but they do not know each other
  - each runs to completion, producing an output, before the next process begins

---

**Internal Structure of Batch Update Process**

- **batch transaction driver**
  - procedure call
  - access (next) transaction
  - access acct/item
  - basic validation check
  - produce (next) transaction
  - post acct/item

- **Generic**

- **Application-Specific**

- The driver calls a different set of application-specific modules for each transaction type (e.g., no duplicates for Open Acct) (e.g., no male pregnancy)
  - One for each of the master files (e.g., CUST and ACCT) (e.g., CLAIM and TREATMENT)
**Tektronix Case Study**

✧ **Background**

✦ industrial development of oscilloscope SA
✦ 3-year collaborative effort between several Tektronix product divisions and Computer Research Laboratory

✧ **Oscilloscope**

patient monitoring (ECG, EEG, EMG)
automotive diagnostics
radar control, auto pilot, etc.

![Oscilloscope Sw. Sys.](image)

✦ old: simple analog device
modern: rely primarily on digital technology, complex sw
✦ **electrical signal** real-time continuous signal (in voltage)
✦ Oscilloscope sample signals at discrete pts, measure them, display traces huge internal data storage, interface to a network of workstations sophisticated GUI, >>300 user-level commands

**Tektronix Case Study**

✧ **Needs**

✦ accommodate frequent changes in hw capability
✦ new reqs on UI
✦ tailor general-purpose oscilloscope to specialized markets
✦ better reconfigurability (how/when/where signal is acquired & displayed)

✧ **Problem**

“little reuse” -> not “better, cheaper, faster, ...”

![Product Division A](image)

development convention A
PL A
tool set A

![Product Division B](image)

development convention B
PL B
tool set B

![Product Division C](image)

development convention C
PL C
tool set C

✧ **Goal:** develop an architectural framework

✧ **Result:** a domain-specific SA as the basis of the next generation oscilloscopes
### An OO Model

**Object**

- **signal**
- **waveform**
- **trace**
- **measurement**
- **trigger mode**

- **max-min wvfm**
- **x-y wvfm**
- **accumulate wvfm**

- **waveform** a (seq. of) oscillations, in voltages, in a given time interval
- **trace** a (seq. of) oscillations, on screen, in a given time interval
- **measurement** frequency in general
  - #peaks in a given segment
  - max/min/avg peak
  - max/min/avg rise or fall in slope
  - rise/decline time
- **trigger mode** slopes greater than a given threshold value

+ led to a concept recognition & clarification of the data types
- no overall model for explaining how the types fit together

---

### A Layered Model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Interface</strong></td>
<td>interacts with the user and decides which data to show on the screen</td>
</tr>
<tr>
<td><strong>Visualization</strong></td>
<td>manipulates saveform (incl. waveform addition)</td>
</tr>
<tr>
<td></td>
<td>Fourier transformation (converting input signal from time domain to frequency domain)</td>
</tr>
<tr>
<td></td>
<td>maps digitized waveforms (&amp; measurements) to visual rep.</td>
</tr>
<tr>
<td><strong>Digitization</strong></td>
<td>digitize signals and store them internally for later processing waveform acquisition</td>
</tr>
<tr>
<td></td>
<td>(extracting a bounded time slice of a signal)</td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td>manipulates fns that filter signals as they enter the oscilloscope</td>
</tr>
</tbody>
</table>

- **Constraint**
  - communication between neighboring layers only

+ intuitively appealing, due to partitioning of fns into well-defined groupings
- boundaries of abstraction conflicted with the needs for fn interactions users need to affect
  - setting channels in Hardware layer
  - choosing acquisition mode & parameters (e.g., threshold values) in Digitization layer
  - creating derived waveforms (e.g., scaling factor) in Visualization layer
## OSI Reference Model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>email (SMTP, MIME), ftp, telnet, gopher, usenet (NNTP), rlogin, mosaic, netscape, ...</td>
</tr>
<tr>
<td>Presentation</td>
<td>error-free transmission, Data Rep. (ASCII, EBCDIC), Compression, Encryption</td>
</tr>
<tr>
<td>Session</td>
<td>graceful data exchange, dialog mgmt, (re)synchronization</td>
</tr>
<tr>
<td>Transport</td>
<td>Split messages, Quality Of Service (QOS), multiplexing</td>
</tr>
<tr>
<td>Network</td>
<td>fast transmission routing accounting</td>
</tr>
<tr>
<td>Data Link</td>
<td>error-free transmission</td>
</tr>
<tr>
<td>Physical</td>
<td></td>
</tr>
</tbody>
</table>

## Version Management

- incremental development
- modifiability
- portability
- structuring can be difficult
- performance
A Pipe&Filter Model

- **Couple** conditions external signals
  - DC: nothing gets changed
  - AC: the value should be subtracted from the appropriate DC offset
  - Ground: the voltage value = 0
- **Acquire** derive digitized waveforms from the signals, when a trigger event is detected by the Trigger subsystem
- **To-XY** converts waveforms into visual data by scaling it and positioning/translating it
- **Clip** fits the visual data into a display of certain size
  - + intuitively appealing; well-partitioned functions, etc.
  - - interactive control difficult

A Modified Pipe&Filter Model

- Users dynamically control the oscilloscope by setting parameters
  - + intuitively appealing; well-partitioned functions, etc.
  - + interactive control possible
  - A 2-layer architecture
Oscilloscope in Action -> Formalization

Waveform

Volts

AbsTime

scaleH, scaleV

Trace

Vert

Horiz

Clipped Trace

0, maxY

maxX, maxY

0, -maxY

maxX, -maxY

Lawrence Chung