The use of interpretation for data acquisition and control; its impact on software development and project management

Otto Vinter
Software Engineering Mentor
Tel/Fax: +45 4399 2662, Mobile: +45 4045 0771
vinter@ottovinter.dk
www.ottovinter.dk

Imagine the time when …

- only one program could run on the computer at a time
- there were no files
- programs were written in assembler language
- programs were kept on paper tape/cards
- the time to add two integers was 2.5 µsec
- online memory was limited to 8-32kB
- and if you were lucky: a backing store of 512 kB

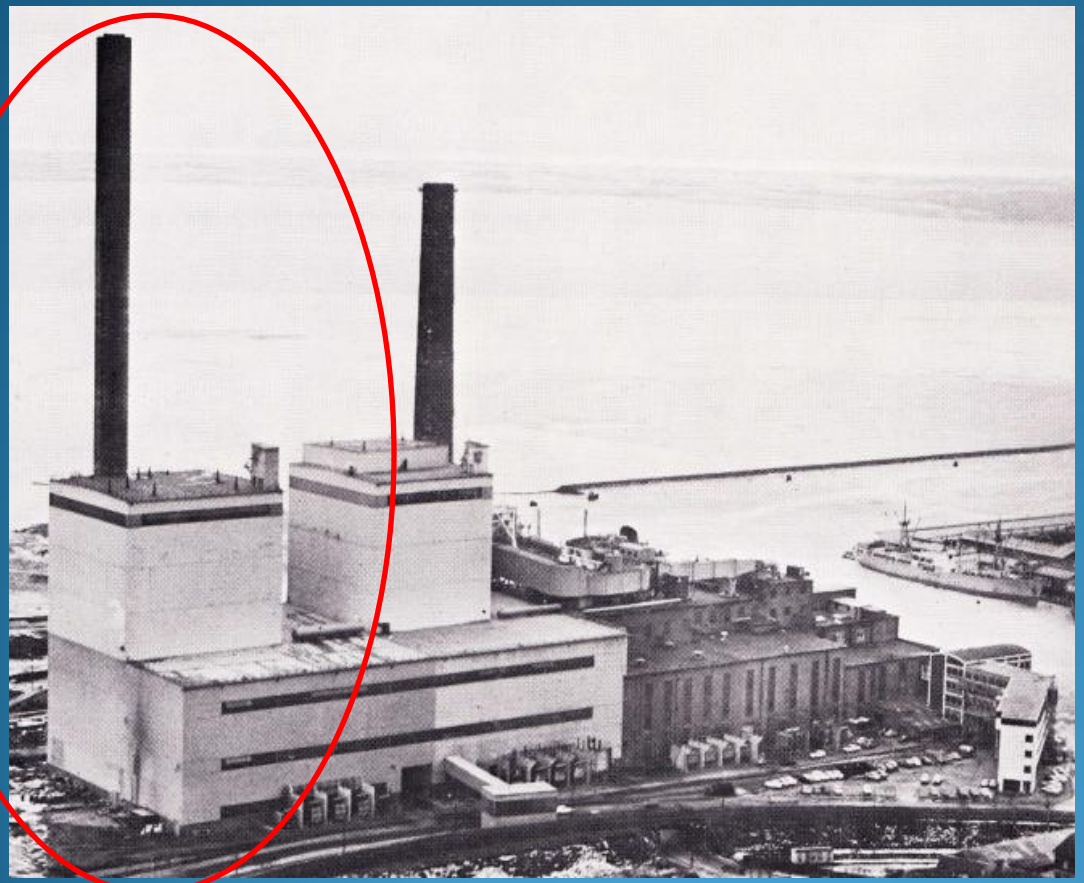
And your project aims to monitor and control a full scale industrial plant !

To-day hardly anyone knows what these things stand for

- so how can we create an appreciation of our working conditions ?
- and how can we tell a story that is meaningful in modern terms ?

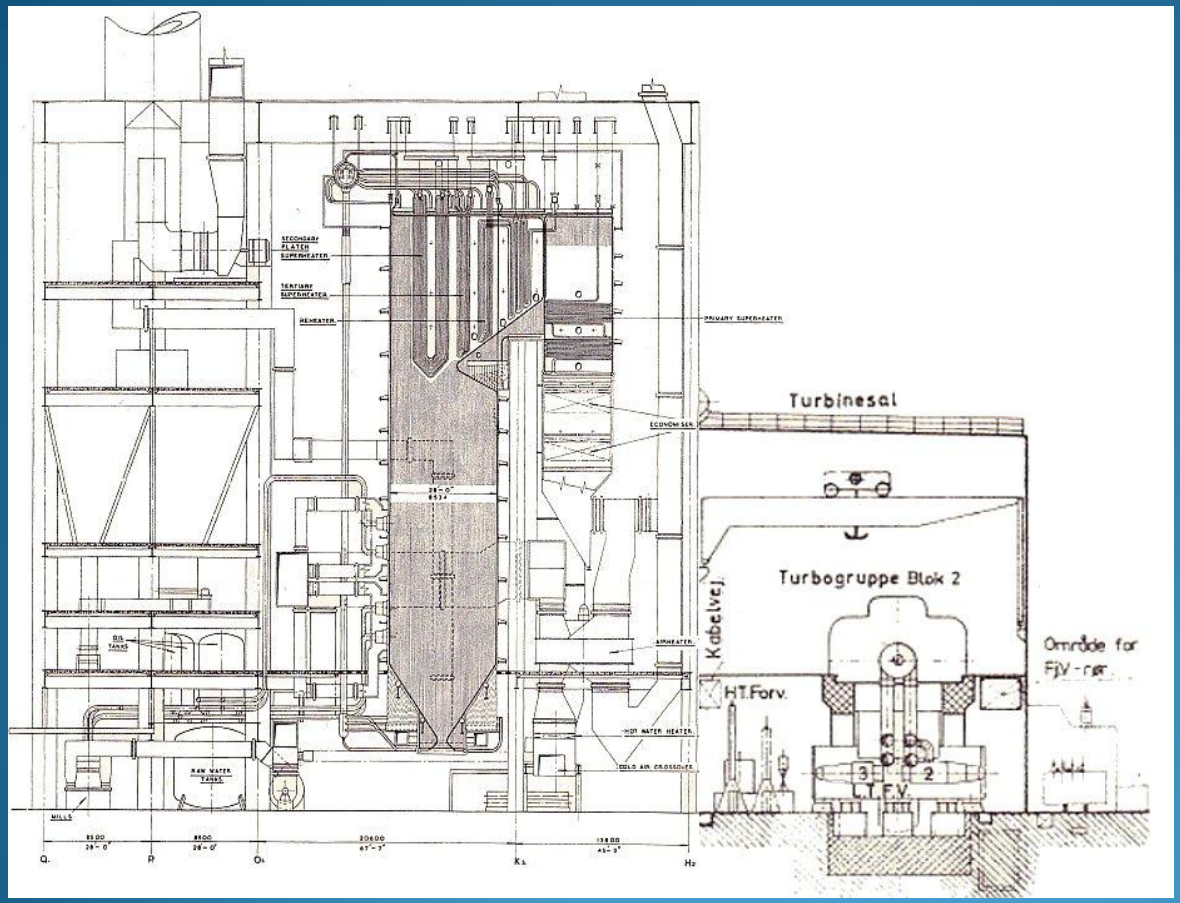
Otto Vinter
Software Engineering Mentor

Vestkraft Blok2



- Built in 1969
- Electric capacity 250 MW
- Heating capacity of 160 Gcal/h
- Turbo-group by BBC
- Boiler unit by Babcock & Wilcox
- All plant controls handled by conventional electronic equipment
- Complete supervision by a digital computer system from A/S Regnecentralen (RC4000)

A view into Vestkraft Blok 2



Computer tasks:

- Every 10 seconds:
  256 analogue measurements
  264 digital inputs and pulses
  48 digital outputs
- Bearing and coil temperatures of major motors, pumps and generators
- Special supervision of boiler drum, oil burners and air pre-heaters
- Correction and compensation
- Alarms

- Every minute:
  256 analogue measurements
- Super-heater pipe temperatures
- Performance and load calculations
- Accumulation and averaging
- Reports

Otto Vinter
Software Engineering Mentor

Compilation vs. Interpretation ?

Problems with compilation

- Lack of speed (execution, compilation)
- Large memory requirements (unpredictable)
- Lack of control over actual code
- Monolithic program structure

Advantages with interpretation

- Dedicated language (plant engineering terms)
- Close to machine code (execution time, memory requirements)
- Model of the plant components in data structures (virtual machine)
- Small subroutines (coroutines vs. embedded procedures)
- Real-time issues separated from model execution
- Flexible development, testing and implementation environment

Processing commands for a temperature variable

```
; Create new value for TFd and add to sum in TFDS10
/802                          ; TFd, steam temperature for HT
 :IWR,      K=802             ; Initialize working registers (variable 802)
 :LSV,      V802              ; Load state and value for TFd (variable 802)
 :ECAV,     R1T25             ; Evaluation control of analog value (range, terminal)
   L1                         ; Skip conversion and checks if compensated by operator
   L2                         ; Skip conversion in case of measurement failure
 :CRE,      K=150             ; Convert resistance element (parameter value)
 :ILCMM,    K=-200,Pih=6000   ; Instrument limit control (min, max)
2:TPC,      V802              ; Test for failure and update status (TFd)
 :TCCV,     V219              ; If compensation use value for TOH (variable 219)
1:PCM,      K=-50,Pah=5650    ; Plant status control (hysteresis, maximum)
 :SSV,      V802              ; Store new state and value (TFd)
 :SUM,      V3301             ; Add to TFdS10 (variable 3301)
```


Impact on testing and quality

The use of the interpretation principle enables:

- Off-line testing (simulation) of individual components
  - Specific configurations (data-model) of components
- Introduction of drivers and stubs
  - Pseudo components and test data
- Logging during tests
  - Comparison to expected values
- Automated regression tests
  - Documented test conditions and test cases

On-site implementation at Vestkraft

- Short on-site implementation period in parallel to running-in of the plant
- Negligible number of bugs (none critical to the running of the plant)

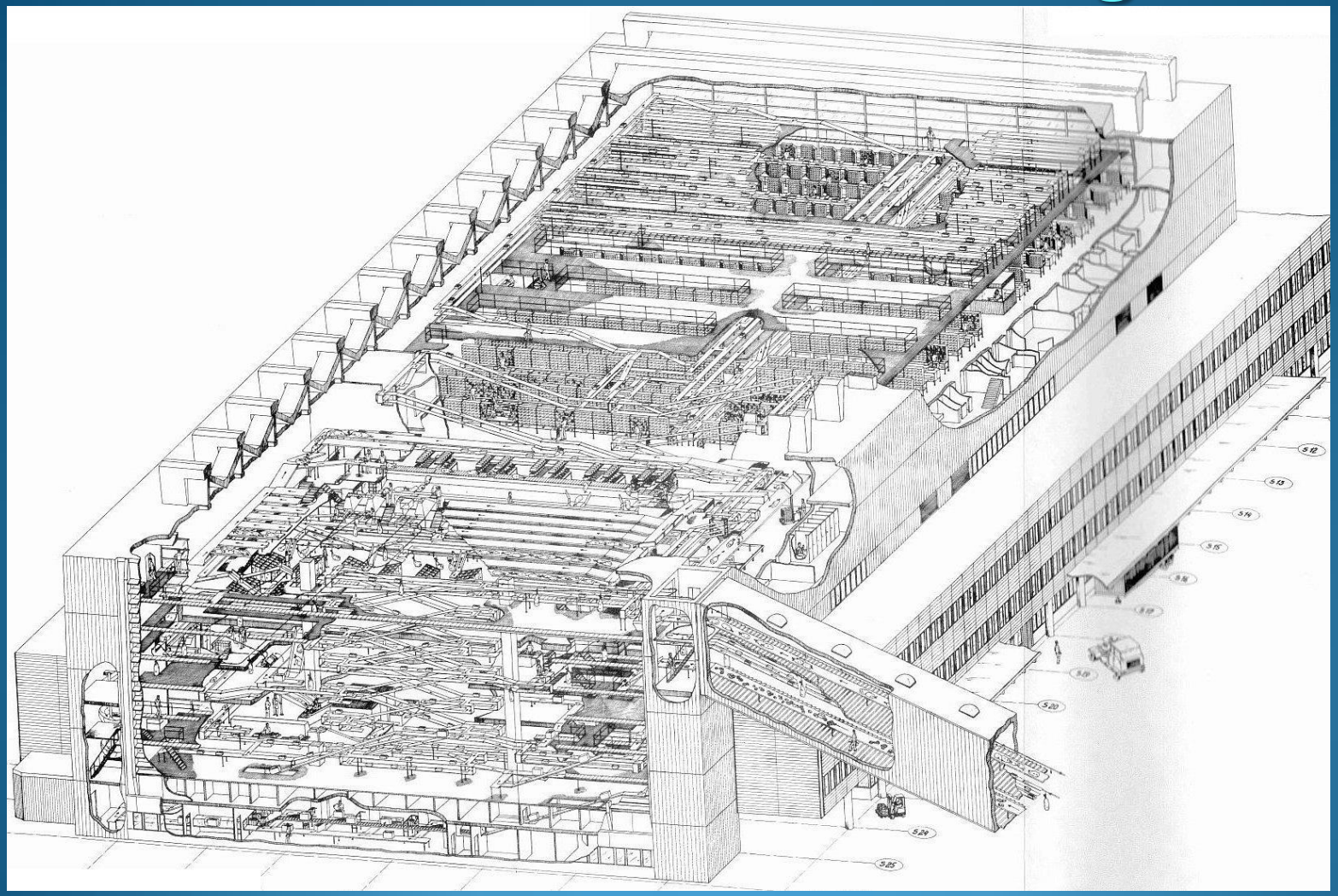
Copenhagen Mail Sorting Center

- Built 1974-1978
- Receiving, sorting and dispatching mail on a tight 24/7 schedule
- 130.000 parcels and 3 million letters per day arriving/departing on trucks/trains
- App. 1000 conveyor belts
- 8 sorting machines
- Hundreds of barcode readers
- Several fixed and power&free chain bag transporters

- 5 pairs of hot stand-by computers (CDC System 18-17, 32-88 kB), a number of microprocessors, and a central management computer (RC8000)
- Data acquisition and control transmission loops with up to 512 local terminal units
- Hundreds of digital input/output

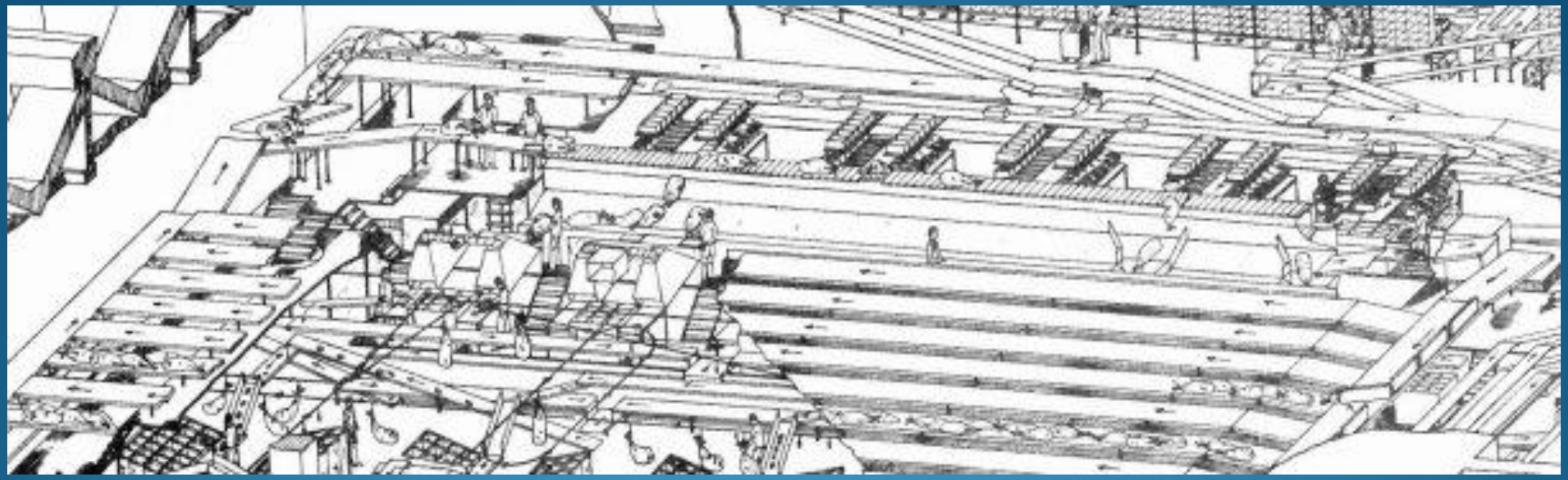
Otto Vinter
Software Engineering Mentor

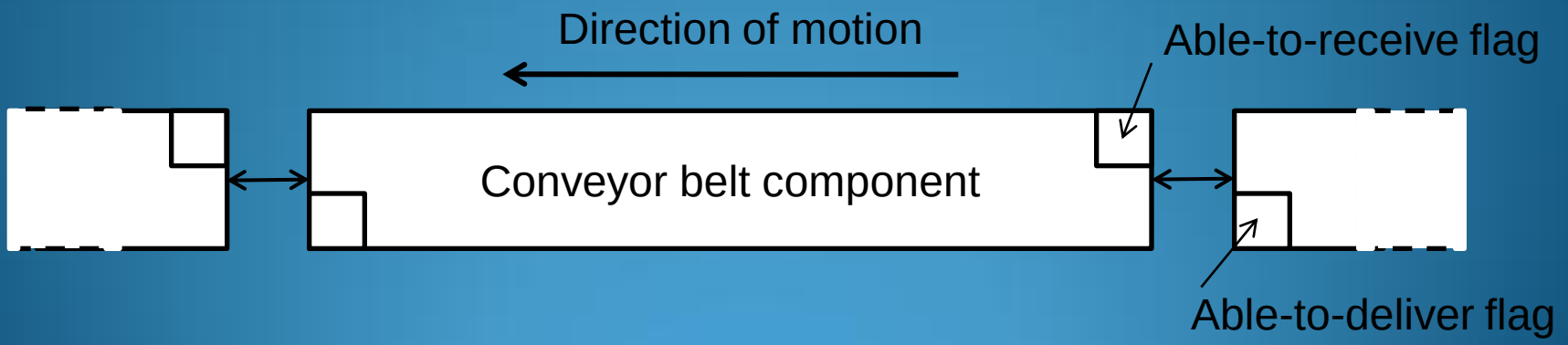
The interior of the Mail Sorting Center



Otto Vinter
Software Engineering Mentor

Conveyor belt connections and their modeling





Otto Vinter
Software Engineering Mentor

Impact on project management

The use of the interpretation principle enabled:

- Efficient planning
  - Simple work break down structure
  - Estimation of primarily small and simple components
  - Quarterly detailed revision of planning and sequencing
  - Flexible assignment of work items and detailed follow-up on progress
- Simple staff hiring and training
- Effective development
  - Independent development and testing (execution of off-line batch jobs)
  - Automated unit and regression testing
  - Promotion in stages (unit, system, production)
  - Focused on-site implementations in parallel to construction work

The software systems for the Mail Sorting Center were delivered within budget (fixed-price)

Implications – why didn't it catch on?

Higher demands on the applications

- Faster response times (interaction)
- Closed loop control over the process (PID)

Advances in computer technology

- Reduced need for comprehensive centralized systems
- Networked dedicated micro-processors and PLC controllers took over
- Less complex tasks for each processing unit
- Cheap standard processor components

Advances in software technology

- Emergence of general purpose languages
- Faster compilers with less memory requirements
- New software concepts (e.g. object orientation)

Contemporary use of interpretation

The interpretation principle is still alive:

- National Instruments' LabView ©
- MathWorks' MatLab ®
- Microsoft Excel ©

When should you use the interpretation principle:

- High complexity of the underlying model
- Need for flexibility in interacting with the user/environment
- Reduced requirements on speed/reaction to stimuli

?

Thank you for listening

Otto Vinter

Software Engineering Mentor

Tel/Fax: 4399 2662, Mobil: 4045 0771

vinter@ottovinter.dk

http://ottovinter.dk

Otto Vinter

Otto Vinter is an independent consultant and mentor advising clients based on his long and comprehensive experience in the software engineering field.

He also collaborates with the Danish software engineering consulting company DELTA on larger assignments and research projects, and implements CMMI compliant processes.

Previously, he was responsible for software process improvements at Brüel & Kjær. He has been the driving force in that company's improvement activities in testing, requirements engineering, development models, and configuration management.

He is appointed external examiner for the M.Sc. education in Computer Science for the Danish Universities. He is an expert evaluator on the IST framework programmes of the European Commission in the areas of: Software Systems and Services, Technology-enhanced Learning, and the Life Long Learning Programme.

He has managed process improvement actions for more than 15 years after having managed software development projects for 25 years.

He received his Masters Degree in Computer Science from the Danish Technical University in 1968.