PHILIPS

Performance X-ray systems

Jo Peters

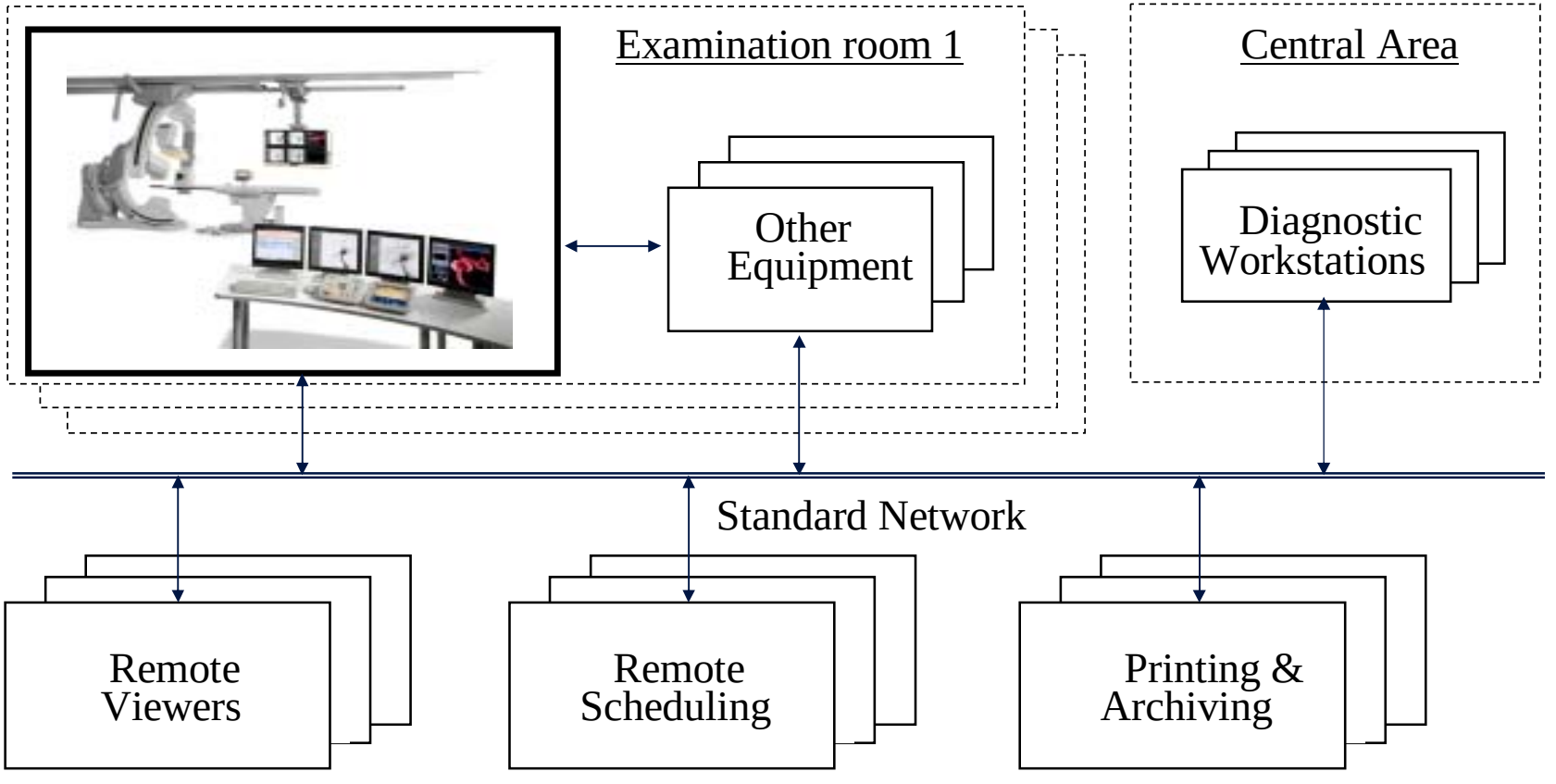
Philips Medical Systems; X-ray C/V
7/2/2006

Content

- System introduction
- Experienced problems
- System architecture background
- Performance requirements
- Case example
- Conclusions, lessons learnt
- Status and Follow-up


PHILIPS

Hospital Environment



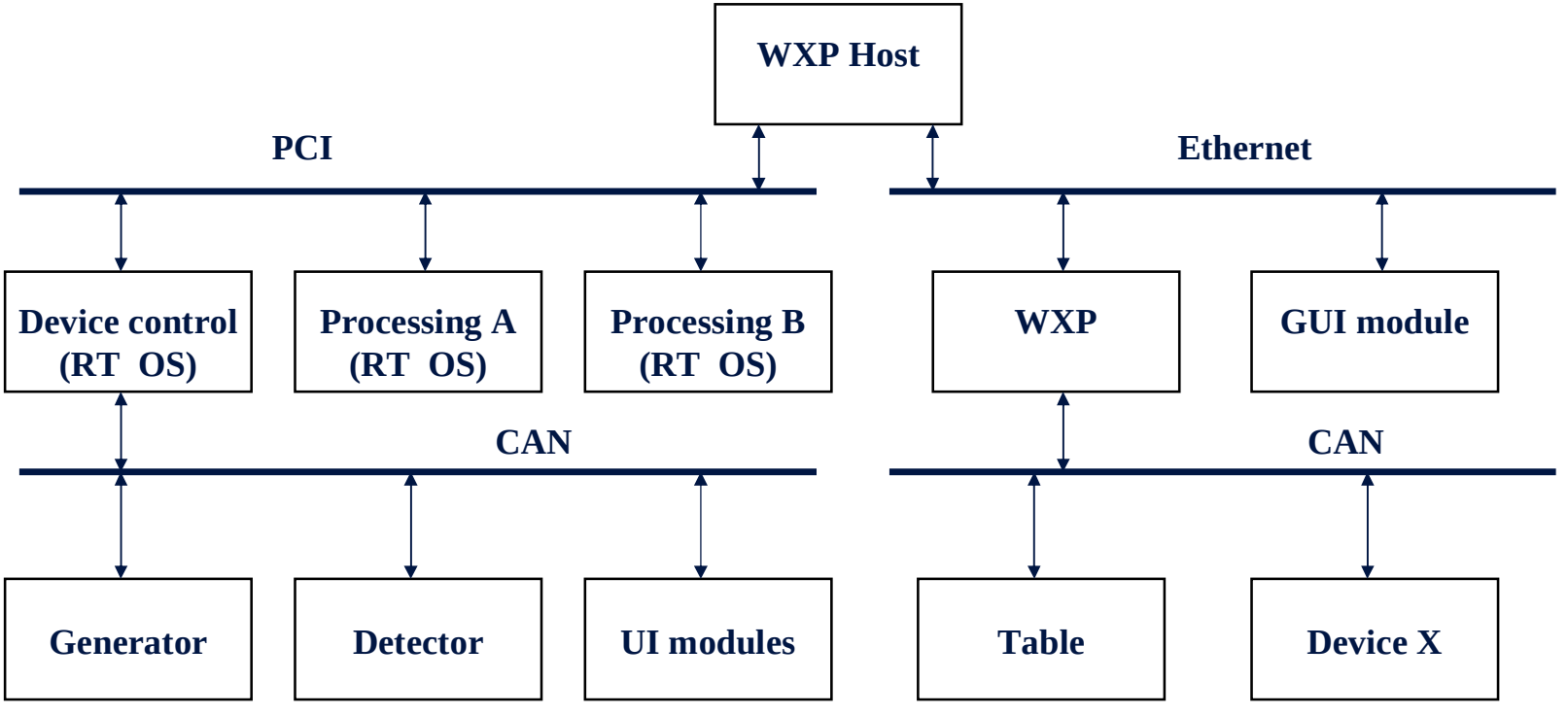
Main characteristics

- Use: diagnostic & interventional
  - High quality: positioning, images, ease of use,...
  - Safety critical: electr., mech., radiation, sterile
  - High demands on throughput & responsiveness
- Main functions:
  - Patient Administration: scheduling, reporting
  - Acquisition: positioning, automation, X-ray
  - Reviewing: navigation, post-processing
  - Finish: distribution, archiving, printing

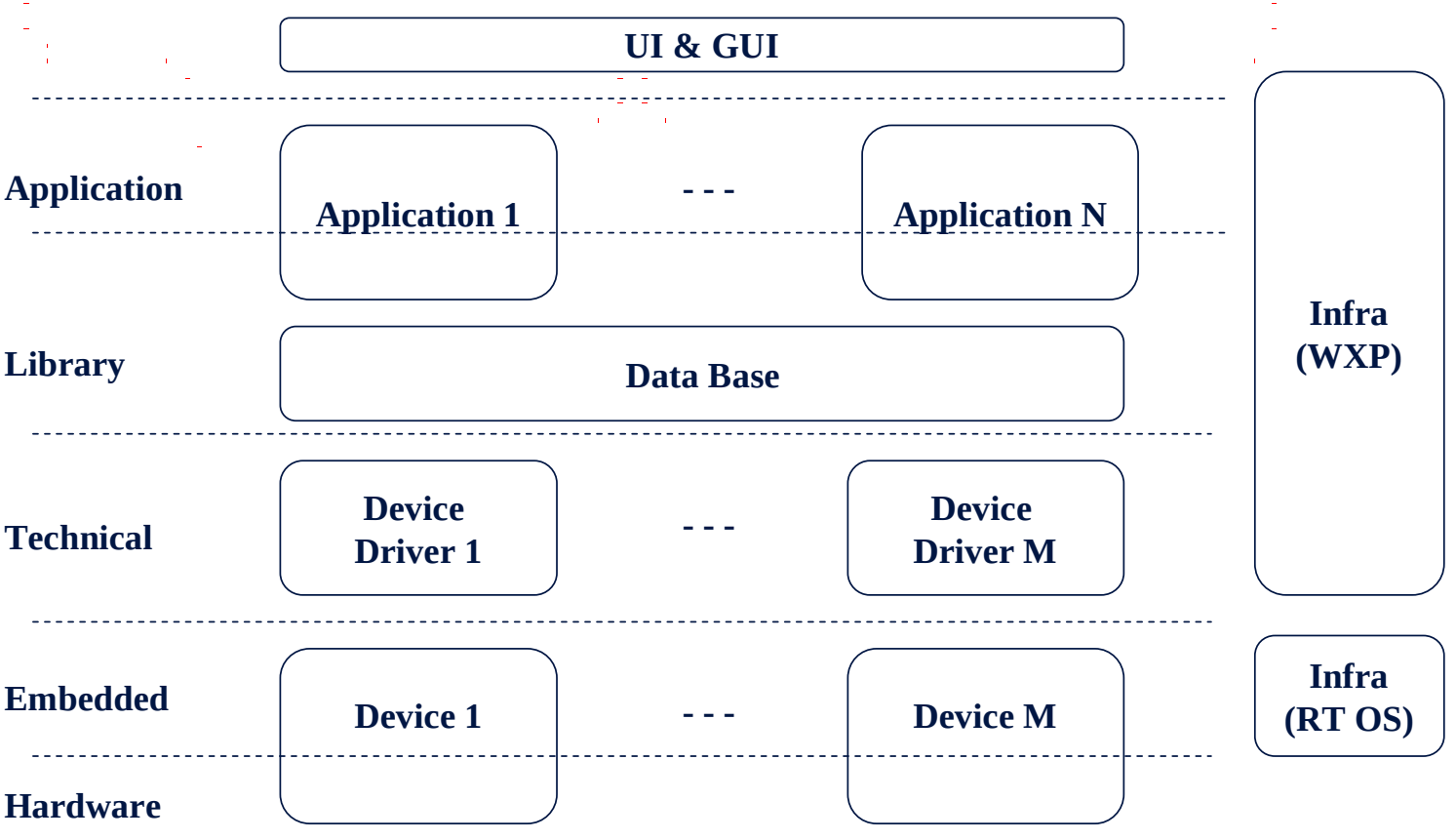
Experienced Problems

- Compromised performance/usability
  - Barely usable slow Application-System
  - Forced acceptance of annoying response times
- Project delay due to ‘surprises’
  - Discovery of discrepancies by coincidence
  - Last-minute system test & verification
  - No design for performance

System Architecture



Host software & layers



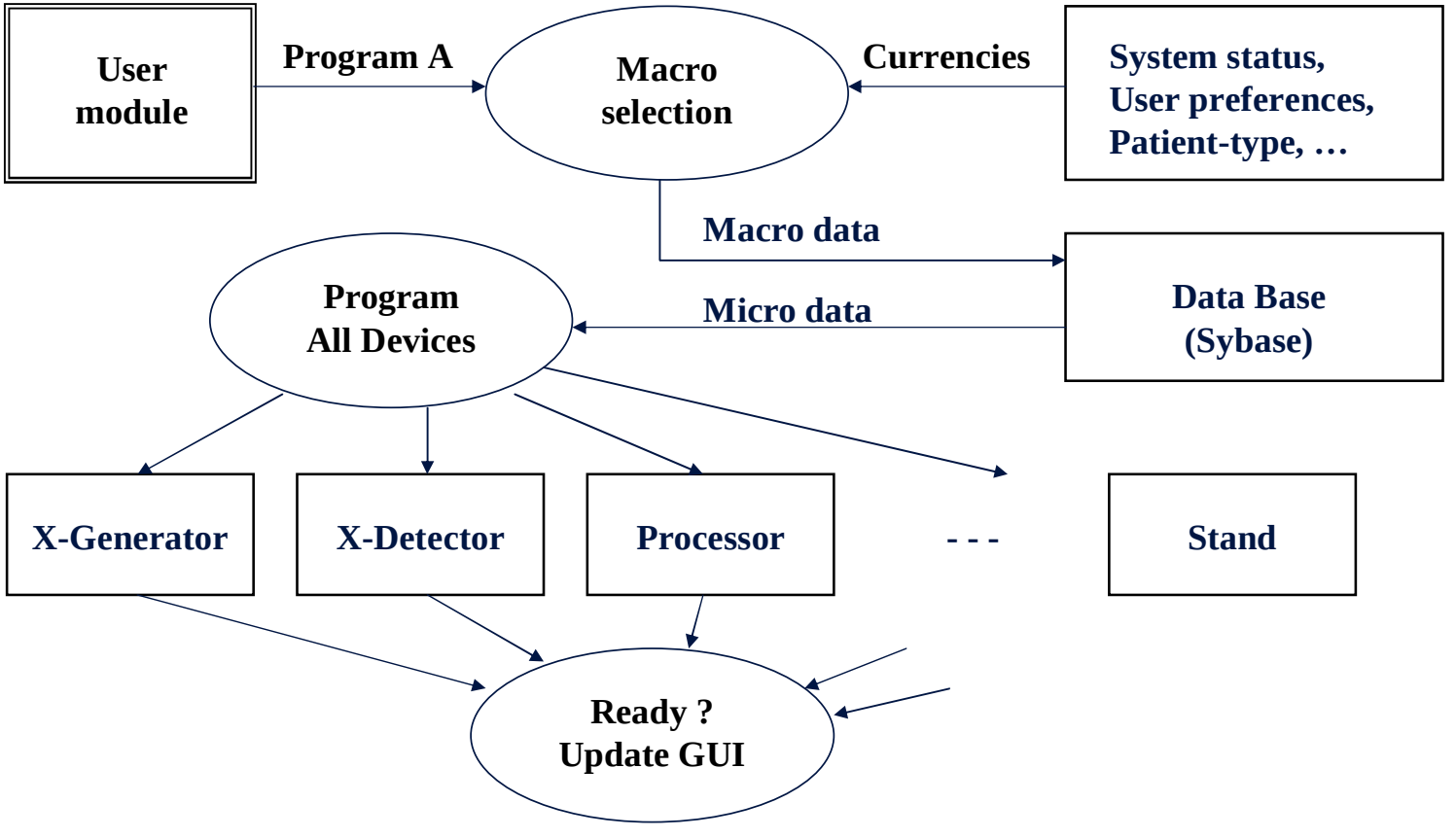
Performance Categories

- Hard Real-time: e.g. 30 images/sec
- Soft Real-time: interactive, 100 msec
- Fast Responsive: sub-second range
- Foreground jobs: seconds range
- Batch jobs: minutes, throughput

Performance Requirements

Functions / use-cases	Must (sec)	Want (sec)	Measured	Action
System power-up	200	50	110	OK
Select Automation Program	1.5	0.5	4.0	PR 12345
Move Stand	0.2	0.1	0.1	OK
Start Radiation (on idlle system)	1.0	0.5	0.4	OK
Start Radiation (while doing ...)	1.0	0.5	1.2	PR .....
Stop Radiation	1.0	0.5	0.3	OK

User selects new Program



Guestimated performance

- Start: User presses button 30 msec
- Get macro settings 10
- Data Query to Sybase 30
- Prepare programming devices 10
- Slowest device 100
- Update GUI 30

Overall: 210 msec; in reality: 4 sec !

Analysis; potential causes

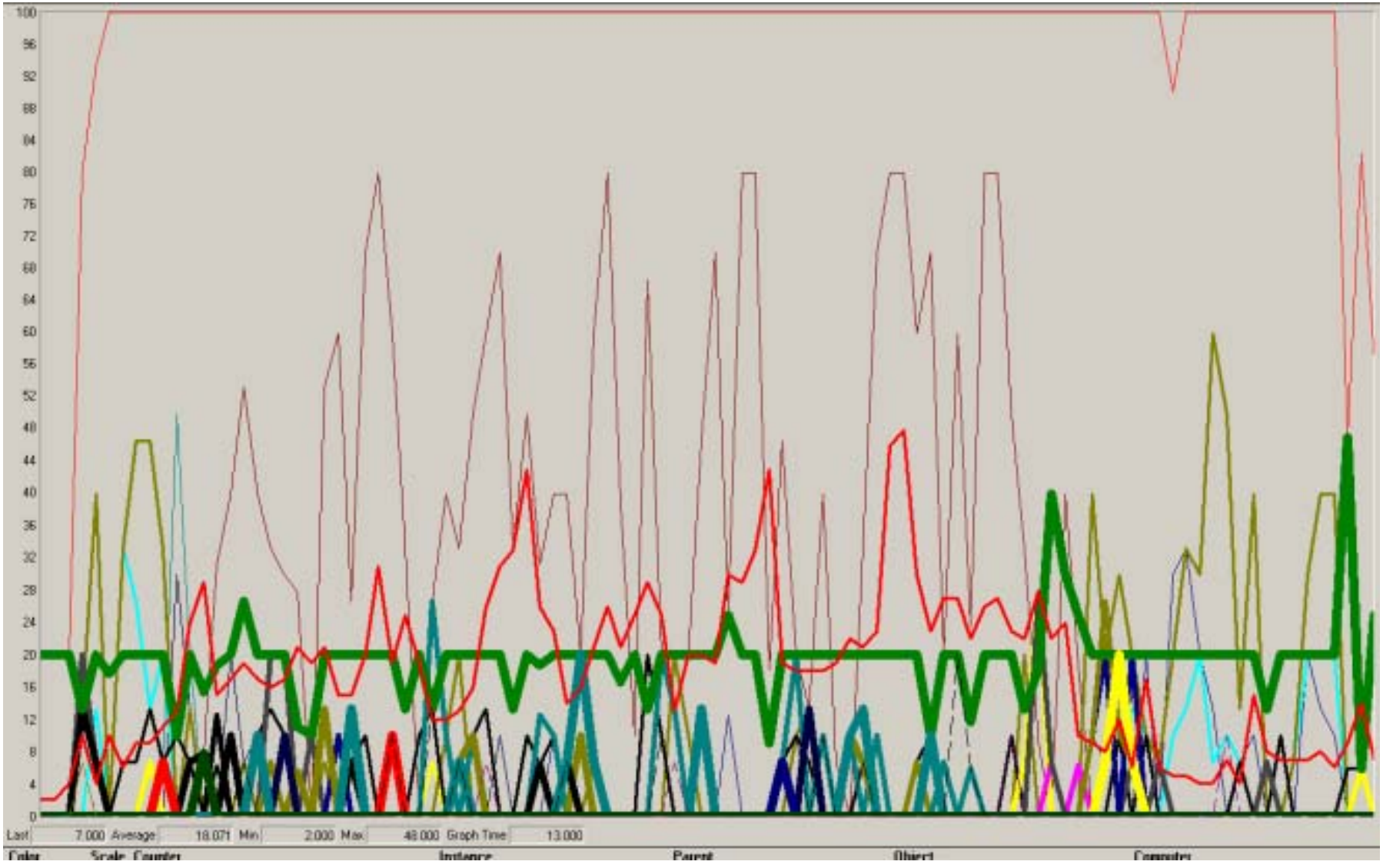
- Explosion due to subscription mechanism
- Thousands of COM Factories & Objects (complete destruction & re-creation)
- Multiple programming of devices
- Overloaded CPU; some characteristics:
  - 100.000 system calls/sec
  - 1700 threats
  - 5000 context switches/sec

Illustration detailed analysis

1333888/286 = ~4650usec

CAcqCICOMParameterControl::CAcqCICOMParameterControl		17		512.65
	CAcqCIParameterControl::CAcqCIParameterControl	286	133888.45	C++ object
	CInfraSubject::CInfraSubject	286	8964.27	Infra
	CInfraComImplClass<CAcqCICOMParameterControlATL,CAcqCICOMParameterControl>::CInfraComImplClass<CAcqCICOMParameterControlATL,CAcqCICOMParameterControl>	286	2785.84	Infra
	CInfraComImplBaseClass::CInfraComImplBaseClass	286	912.03	Infra
	CInfraObserver::CInfraObserver	286	55.08	Infra
	CInfraComImplClassBase::CInfraComImplClassBase	286	2.99	Infra
CInfraComFactory<CAcqCICOMParameterControl>::InitializeObject	286	17373.09		Infra
new	286	5464.76		Infra
CInfraStructuredException::~CInfraStructuredException	286	1342.73		Infra
CInfraStructuredException::CInfraStructuredException	286	1033.27		Infra

Case Example



Lessons learnt

- System not designed for performance
- Implementation over-structured; too general
- Very difficult to understand overall behavior
- Inadequate measuring tools
- Lacking performance decomposition budgets
- No skills; lacking pro-active focus
- SW courses contra-effective

Dedicated Performance Team

Short-term goals:

- Priority 1: no further deterioration
- Preventing last-minute repairs/delays

Means:

- Measure frequently
- Immediate action
- Easy opportunities to be included now

Dedicated Performance Team

Long-term goals:

- Significant improved responsiveness
- Structured performance Analysis & Design
- Embedding performance awareness
- Fire-fighting never needed anymore

Means:

- SW-Principals to Execution-Architecture course
- Performance/resource decomposition; budgets
- Regular/structured verification with Profiler tool
- Determine guidelines: design of (anti-) patterns


PHILIPS