PAUL SCHERRER INSTITUT

Elke Zimoch :: Paul Scherrer Institut

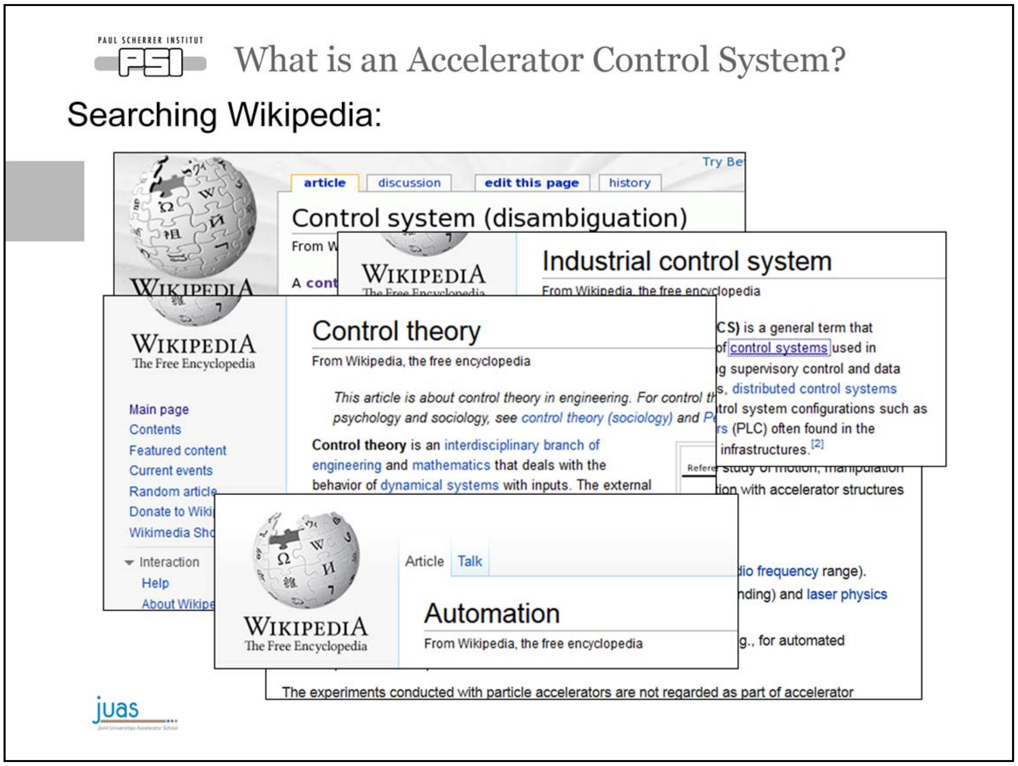
Accelerator Controls

JUAS 2016

Dr. Elke Zimoch
Section Controls
Paul Scherrer Institut
5232 Villigen PSI
Switzerland

Tel.: +41 (0)56 310 5121
E-mail: elke.zimoch@psi.ch
www.psi.ch

Table of Content



From Wikipedia (http://en.wikipedia.org/wiki/Main_Page in Feb/2016):

Control System: A control system is a device or set of devices to manage, command, direct or regulate the behavior of other devices or systems.

Accelerator Physics: Accelerator physics is a branch of applied physics, […]. It is also related to other fields: […] Computer technology with an emphasis on digital signal processing; e.g., for automated manipulation of the particle beam.

Industrial Control System: Industrial control system (ICS) is a general term that encompasses several types of control systems used in industrial production, […].

Control Theory: Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems with inputs.

Automation: Automation or automatic control, is the use of various control systems for operating equipment such as machinery, processes in factories, […] and other applications with minimal or reduced human intervention.

Unfortunately the „Controls Theory" does not cover highly complex and diverse systems like accelerators. So, there is no theory to learn.

Conclusion: This is all related to Accelerator Control Systems, but does not hit the point.

PAUL SCHERRER INSTITUT

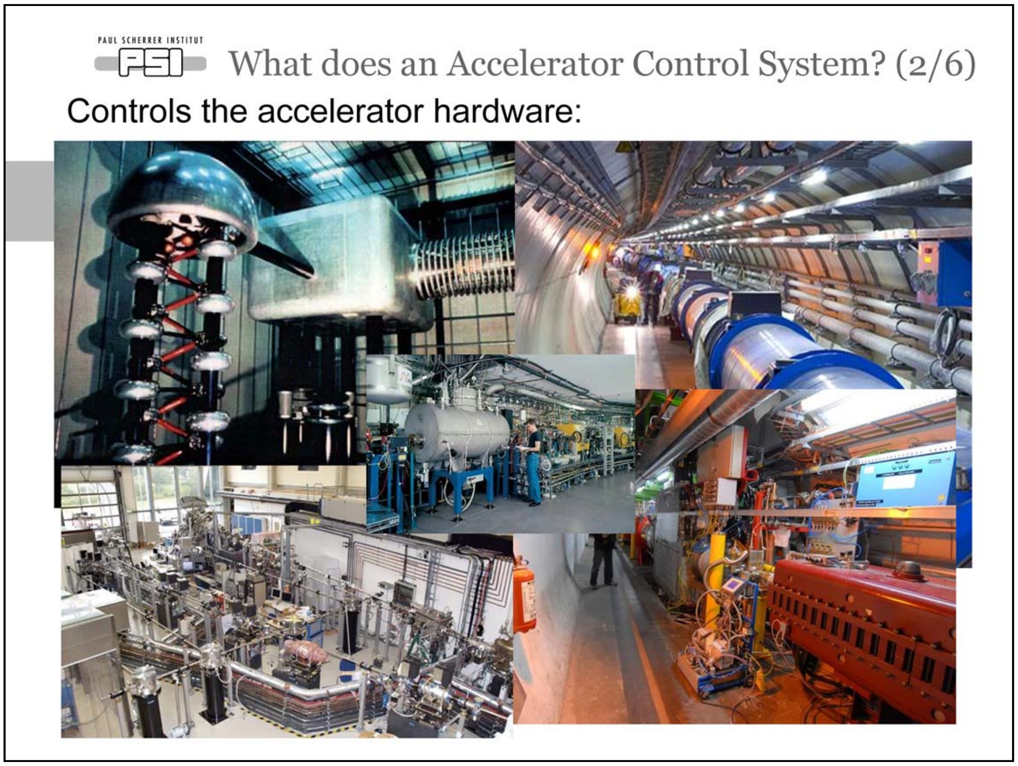
What ~~is~~ does an Accelerator Controls System (1/6)

- Controls the accelerator (Source, Magnets, RF)
- Provides diagnostics information (BPMs, Cameras)
- Monitors environment (Vacuum, Temperature)
- Feedback programs for beam parameters (orbit feedback)
- Makes “the machine” running and controllable …

… reliable, with good performance, flexible … economical safe (without producing black holes and destroying the world)

juas

As there is no easy answer to what an Accelerator Control System is, lets try to find out what it does.



What does an Accelerator Control System actually control?

For example the number of components needed to run an accelerator:

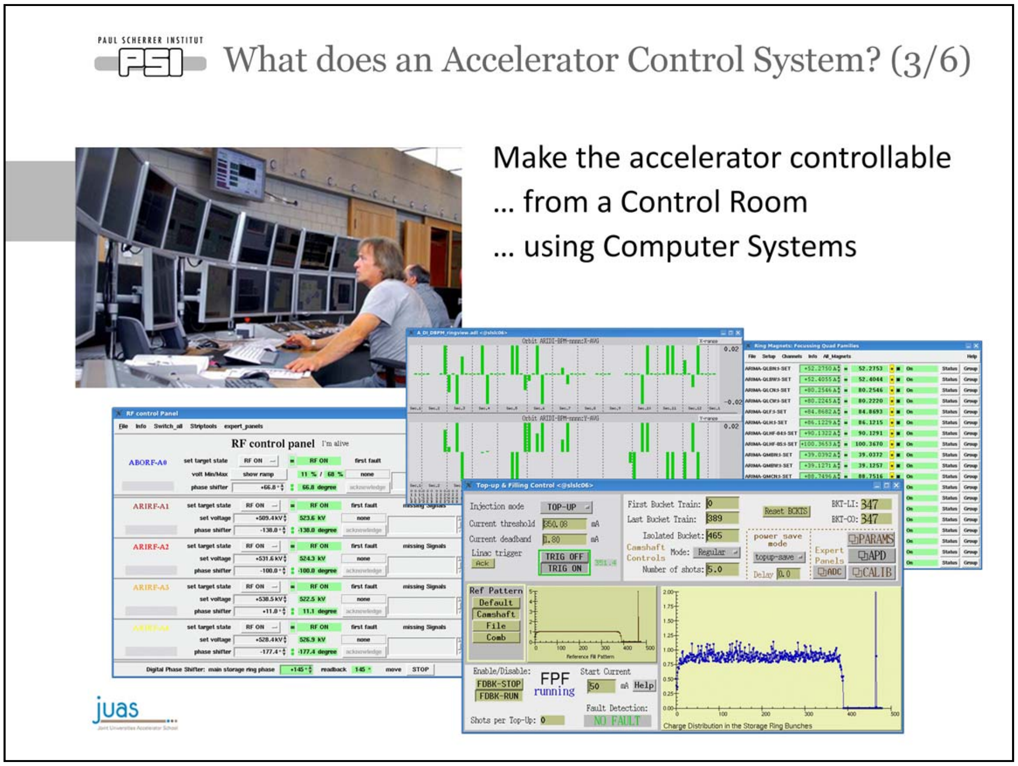
Swiss Light Source (SLS at PSI) Accelerator components to control:

- ca. 200 computers
- ca. 600 magnets (+power supplies)
- 300 vacuum pumps
- 9 cavity structures
- ca. 150 beam position monitors

21 beamlines (together)

- ca. 300 computers
- 10 undulator magnets
- more than 1200 motors

Distances between components (stroage ring with 130 m diameter): 50 km power cable and more than 500 km signal cable



For what do you need an Accelerator Control System (continued)?

Some examples of applications and programs needed to run an accelerator:

-Device Control Panels

-Feedback Systems (e.g. orbit feedback, filling pattern feedback)

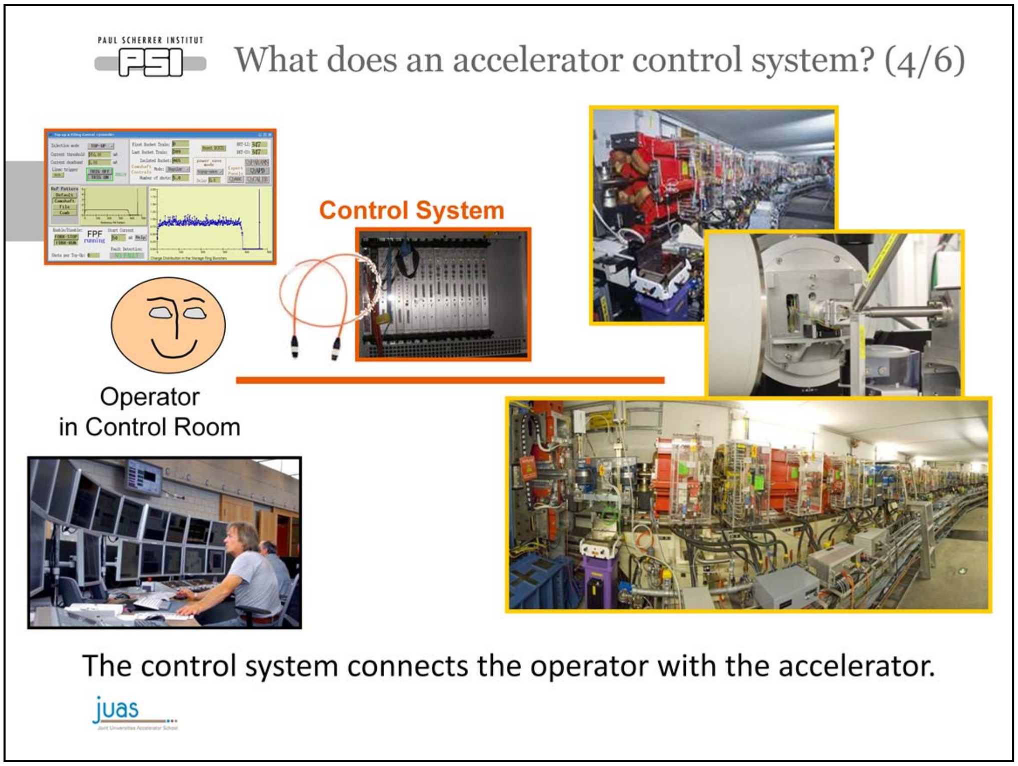
-Archiving and Data Acquisition Systems (Storage)

-Scan tools for experiments and measurements

-Conditioning tools for RF components

-Simulations and comparisons with real accelerator

-Alarm handling and machine protection



The Accelerator Control System connects the Operator in the control room with the accelerator hardware. The control room might not be near the accelerator:

-SLS is 200m away from PSI main control room

-For SwissFEL the control room will be a kilometer away

-Some experiments are controlled remote from all over the world

What does an Accelerator Control System? (5/6)

Inputs → **Accelerator Control System** → **Outputs**

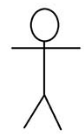
Inputs

- Values from all diagnostic devices (BPMs, screens, ICT/PCT, etc.)
- Information from the experiment
- Read back values from devices (Magnets, RF, etc.)
- Read back values from environment (Vacuum, Temperature, etc.)

Accelerator Control System

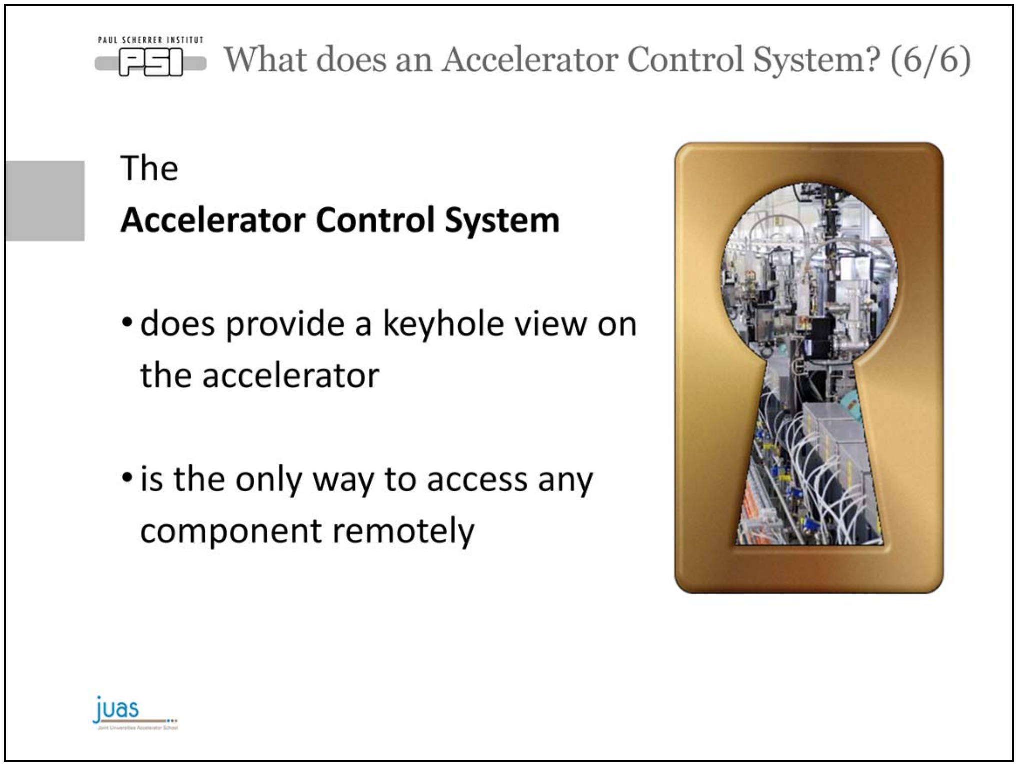
↑

Operator



Outputs

- Set values from devices (Magnets, RF, etc.)
- Actuators of the environment (vacuum pumps, heaters, chillers, etc.)
- Triggers
- Alerts (SMS, emails, alarms, etc.)

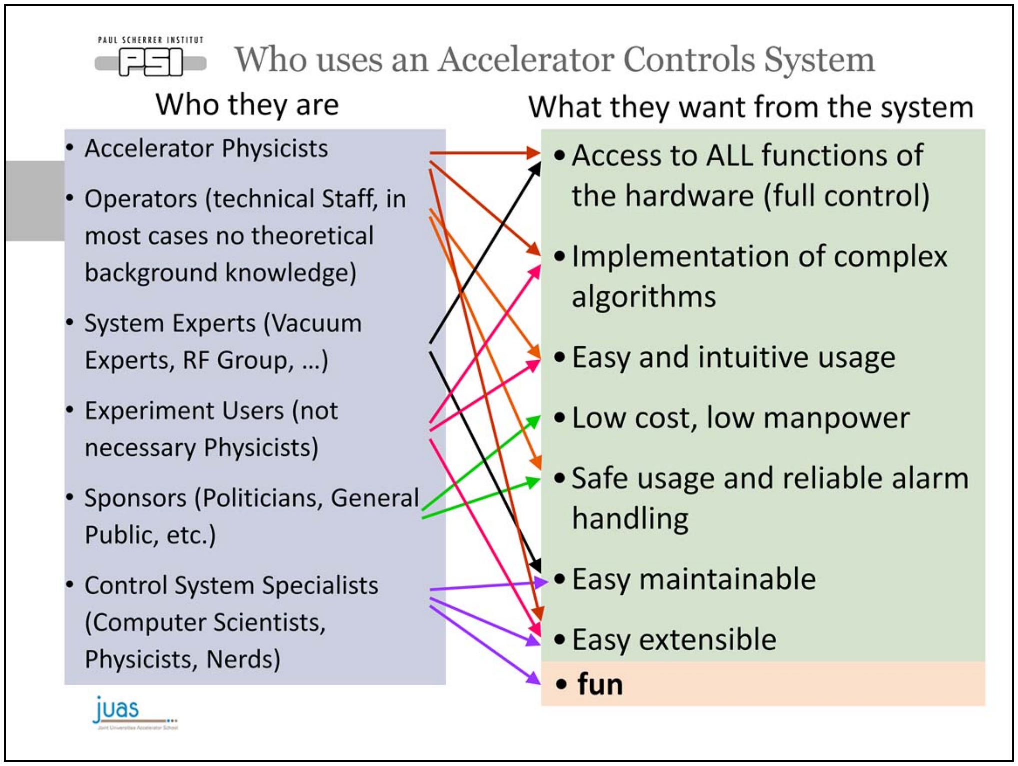


Conclusion:

The Accelerator Control System is the only connection between the Control Room (Operator) and the real hardware.

Everything that is not shown by the control system can not be seen.

If the control system is the only way to access hardware, all systems that are needed for this remote access belong to the control system.



The accelerator control system is used by many different groups for many different purposes – and has to support them all.

What is the Technical Environment?

Control Systems (one way or another) have to deal with …

- **Distributed** end points and processes
- **Data Acquisition** (front end hardware)
- **Real-time** needs (where necessary)
- **Process control** (automation, feedback, PID controller)
- **Central Services** (Archive, Databases, Name Resolution)
- **Data transport** (control system protocol, network)
- **Security** (who's allowed to do what from where?)
- **Time synchronization** (time stamps, cycle ids, etc.)

that is:
Computers (in different flavors) and
Computer Environment

What is an Accelerator Controls System

Definition:
An **Accelerator Control System** is a computer environment that allows remote access to the accelerator hardware with a lot of different functionality to satisfy the requirements of several different user groups.

In addition a modern
Acclelerator Control System:
tries to unify the access to different hardware
(one way to rule them all)

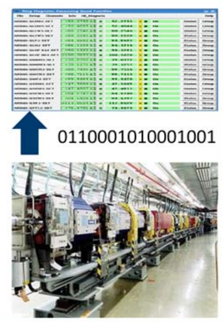
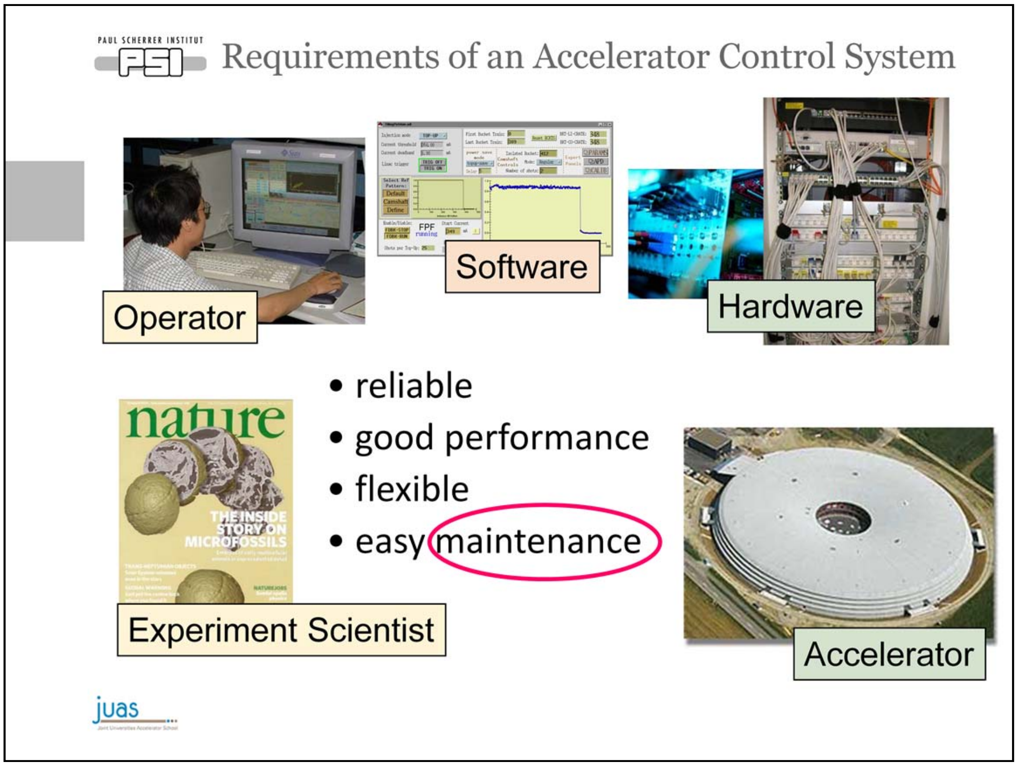


Table of Content



The requirements from different user groups might be different.

The most common requirements are:

•Reliability – if I have beam time scheduled, I want it to happen

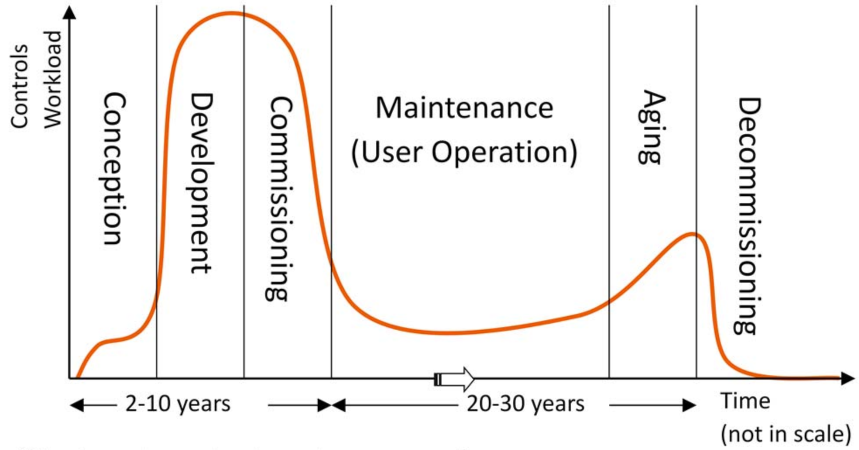
•Good Performance – speed, responsiveness, modern possibilities

•Flexibility – „hey, I have a good idea, lets try …“

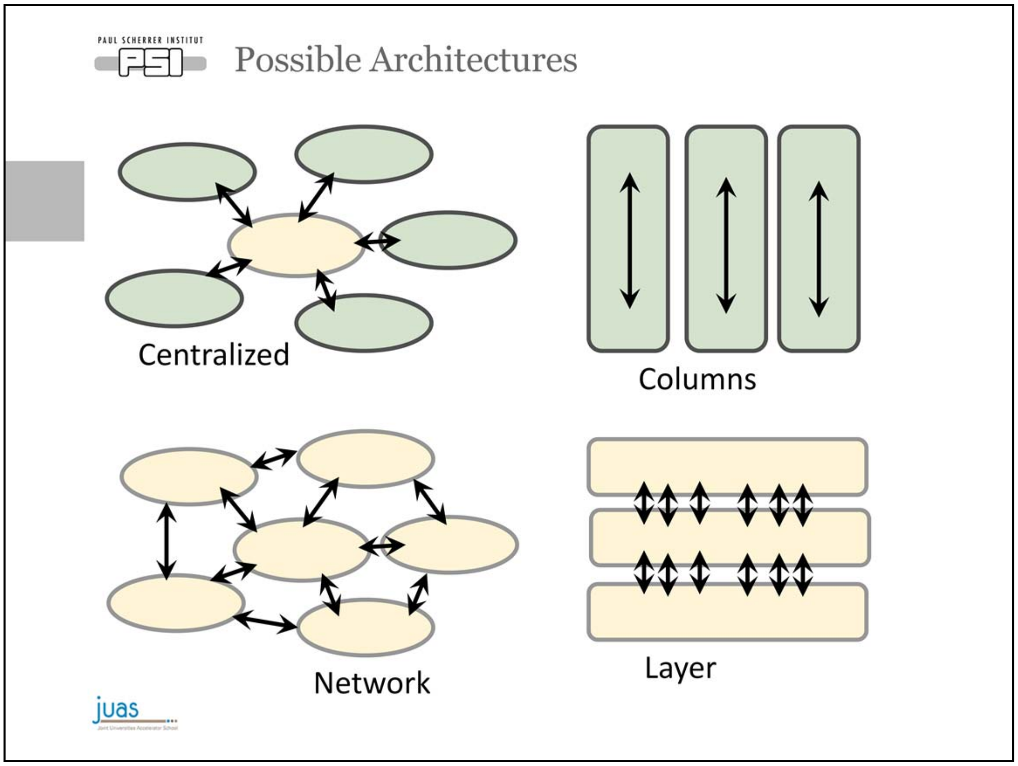
•Easy maintenance – this comes with limited resources

Why is easy Maintenance important?

Controls System Lifecycle:



- *"You have to run to stay where you are"*
- Workload never got to zero during accelerator lifetime
- Normal accelerator lifetime ~ 30 to 40 years



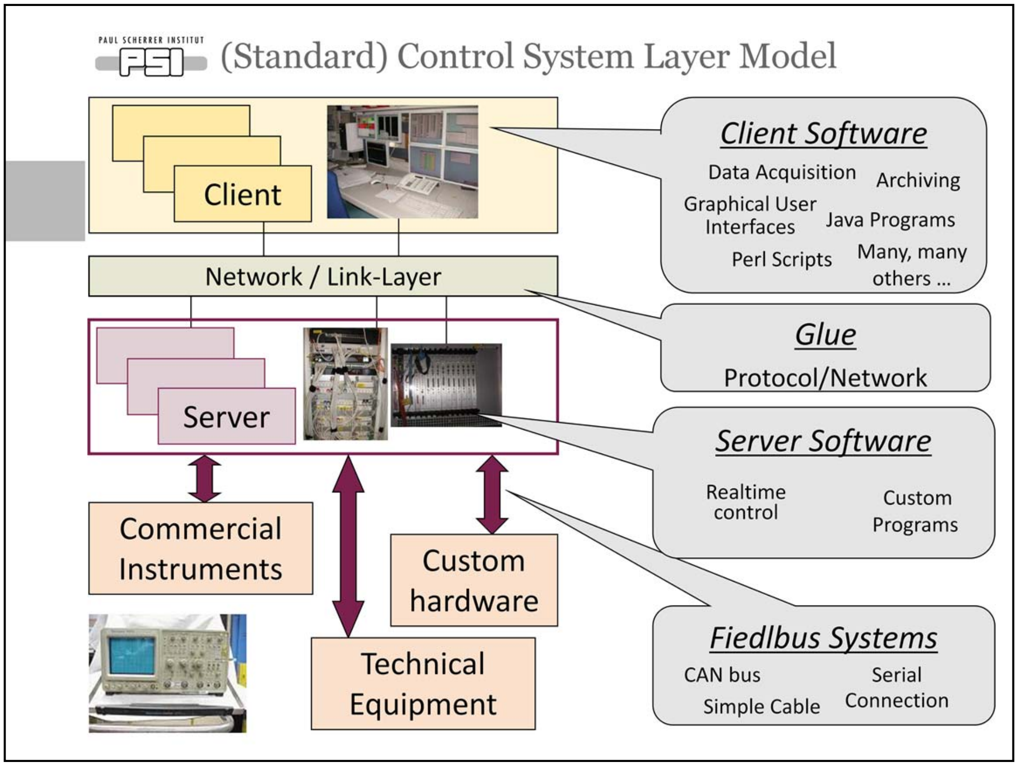
Centralized:
Single point of failure.

Columns:
No exchange of information between different systems.

Network (Peer-to-peer):
Can be difficult to maintain if an addition has to be added to all other systems.

Layer:
Can be difficult to find out where a information came from.

Practical solution:
Mix between some or all of those architectures.



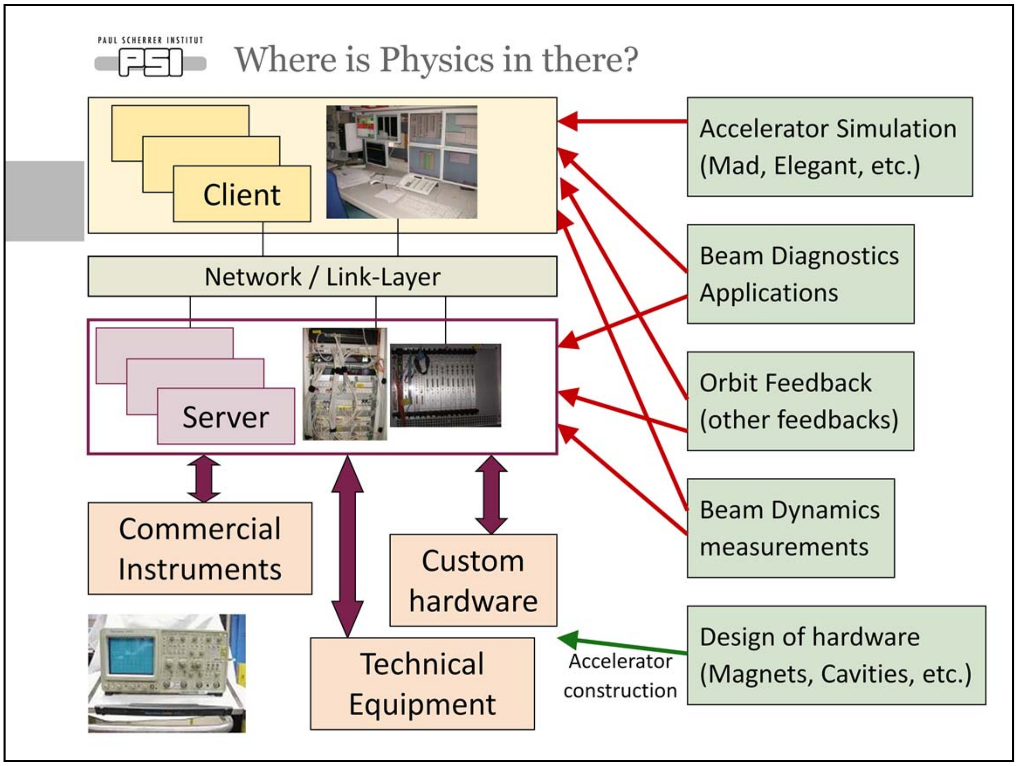
Three-tier standard model for distributed control systems.

The hardware layer was considered dumb when the terminology was developed.

Meanwhile „hardware“ can be as complicated as a robot (with its own controller) or an oszilloscope (with an operating system installed on a specialized computer).

More and more logic is moved to the hardware layer.

Therefore, I would like to call it now a 4-tier model.

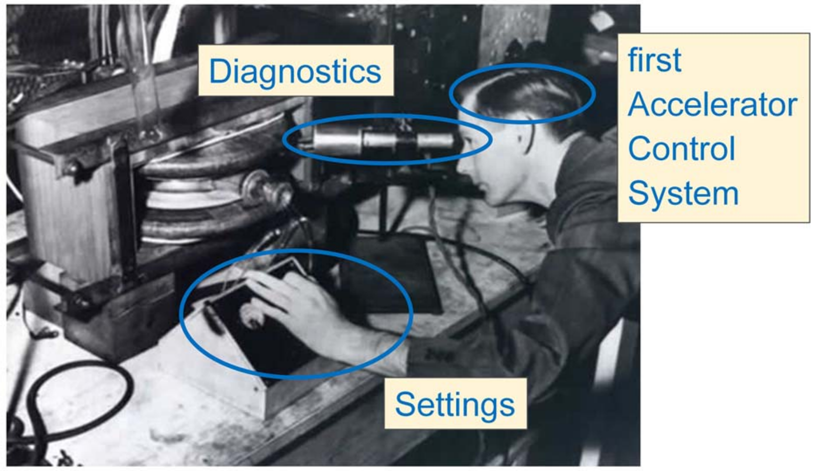

PAUL SCHERRER INSTITUT
PSI
Where is Physics in there?
Client
Network / Link-Layer
Server
Commercial Instruments
Custom hardware
Technical Equipment
Accelerator Simulation (Mad, Elegant, etc.)
Beam Diagnostics Applications
Orbit Feedback (other feedbacks)
Beam Dynamics measurements
Design of hardware (Magnets, Cavities, etc.)
Accelerator construction

PAUL SCHERRER INSTITUT

Table of Content

juas

History of Accelerator Controls (1/3)



Donald Kerst with the first betatron, invented at the University of Illinois (USA) in 1940

juas

http://physics.illinois.edu/history/Betatron.asp

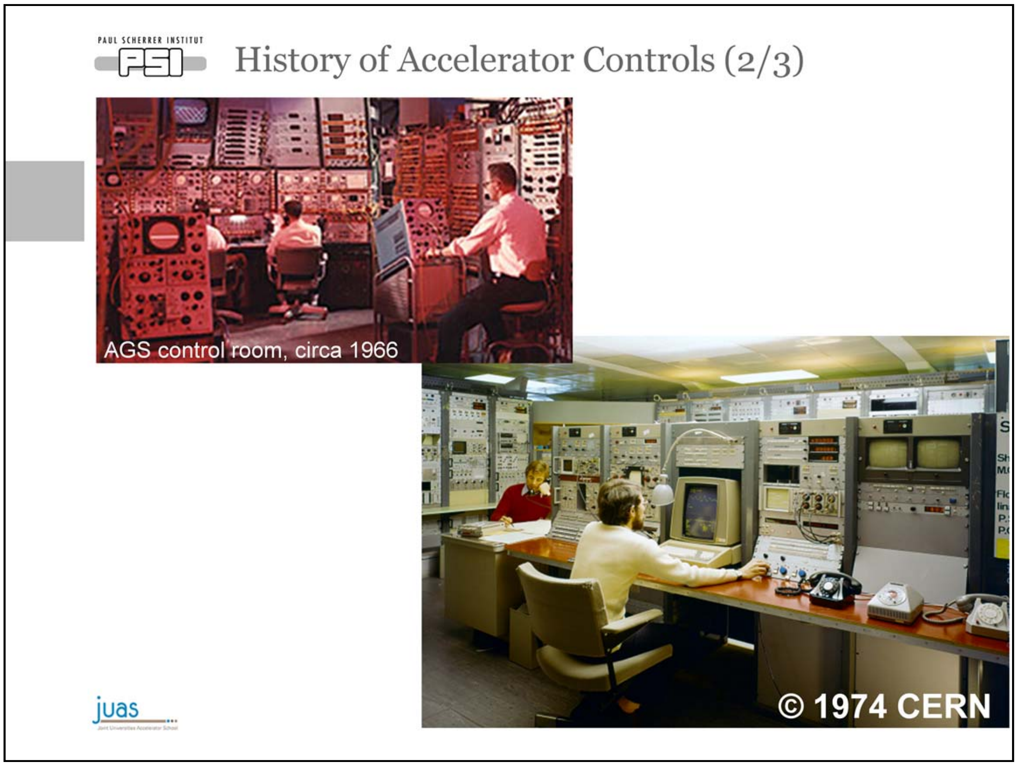
No Control System needed because:

-Direct tuning with knobs

-Direct diagnostics with own eyes

-(Very) limited number of devices involved

-> so the brain is the Accelerator Control System in this case



High Left: AGS control room, circa 1966

Alternating Gradient Synchrotron (AGS) is a Proton Accelerator at Brookhaven National Labs (BLN) in Long Island (USA)

Lower Right: Cern Control Room 1974

Jean-Pierre Potier (turning buttons) and Bertran Frammery (telephoning) on shift. The 26 GeV Synchrotron and later also its related machines (Linacs 1,2,3; PS-Booster; LEP-Injector Linacs and Electron-Positron Accumulator; Antiproton Accumulator, Antiproton Collector, Low Energy Antiproton Ring and more recently Antiproton Decelerator) were all controlled from the PS control room situated on the Meyrin site. The SPS and LEP were controlled from a separat control centre on the Prevessin site. In 2005 all controls were transferred to the Prevessin centre.

Date: Feb 1974
Original ref.: CERN-PHOTO-7402124X

http://www.icalepcs.org/ - Next ICALEPCS will be held 8-13, October, 2017 in Barcelona, Spain

Exerpt from the dinner speech of Roland Müller on ICALEPCS 2013 (http://www.icalepcs.org/uploads/documents/LAA-speeches-2013.pdf):

[…]

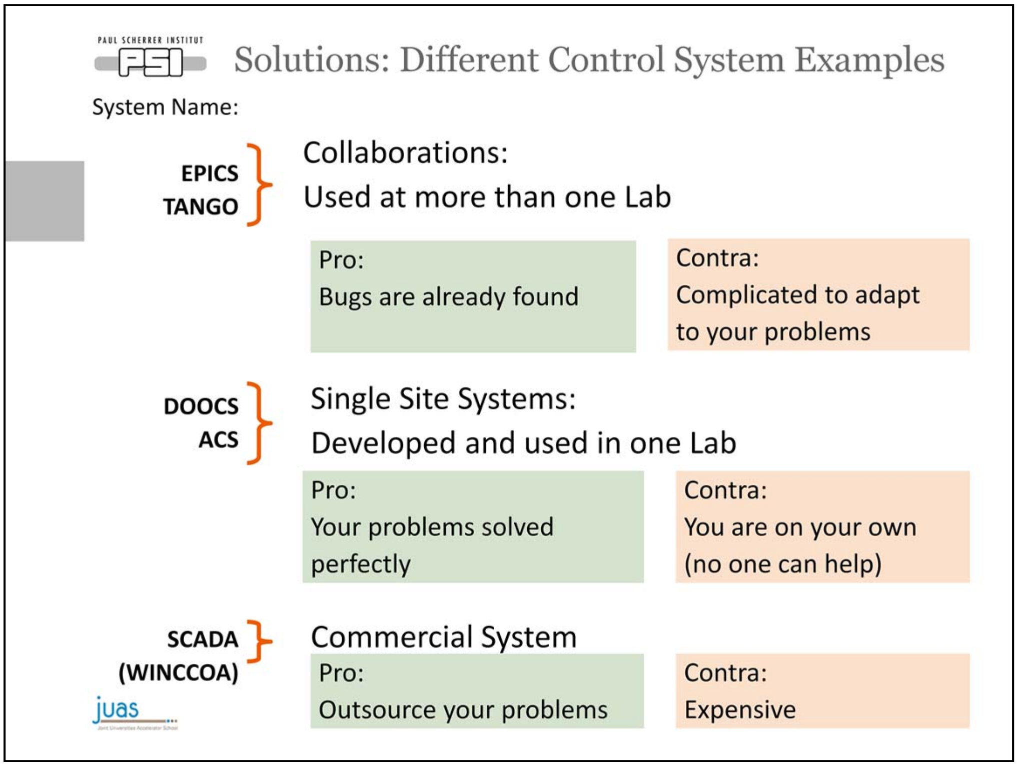
In the 1980´s control system has not been seen as an essential part of accelerator projects. No forum existed to share expertise. On the other hand control systems have been frequently blamed for project delays and cost overruns.

Following 2 precursor events in Berlin and Brookhaven 1985 Peter Clout initiated a workshop in Los Alamos, the first devoted entirely to accelerator control systems. 130 participants presented 50 papers eventually published in Nuclear Instruments and Methods.

2 years later the follow up event the Europhysics conference COCONF in Villars-sur-Ollon already extended the scope to Large Experimental Physics Control Systems.

As the very first representative of an Asian institute Shin-Ichi Kurokawa from KEK participated in both events. On his initiative and with strong support from Axel Daneels from CERN it was already then decided to involve Asia as well as Europe and America in all follow-up events.

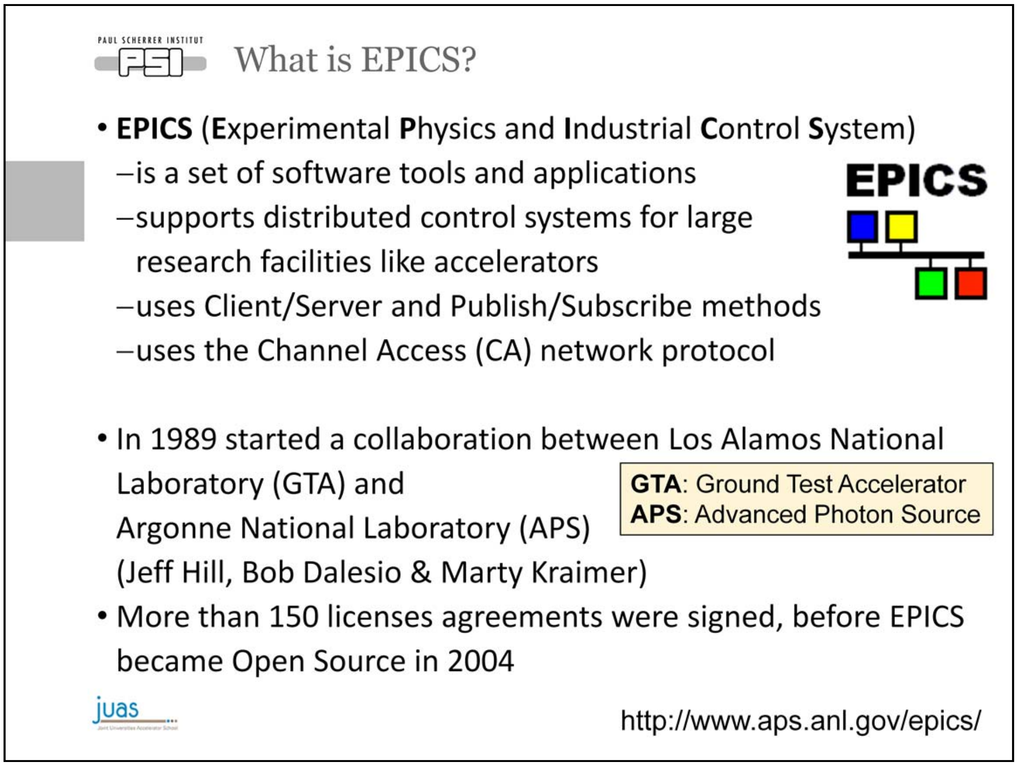
The second conference was held at Triumf in Vancouver in Canada in 1989 where the name has been coined. The third followed 1991 at KEK, Tsukuba, Japan, attended by 240 participants. […]



SCADA = Supervisory control and data acquisition

WINCCOA is the successor of PVSS II from Simens.

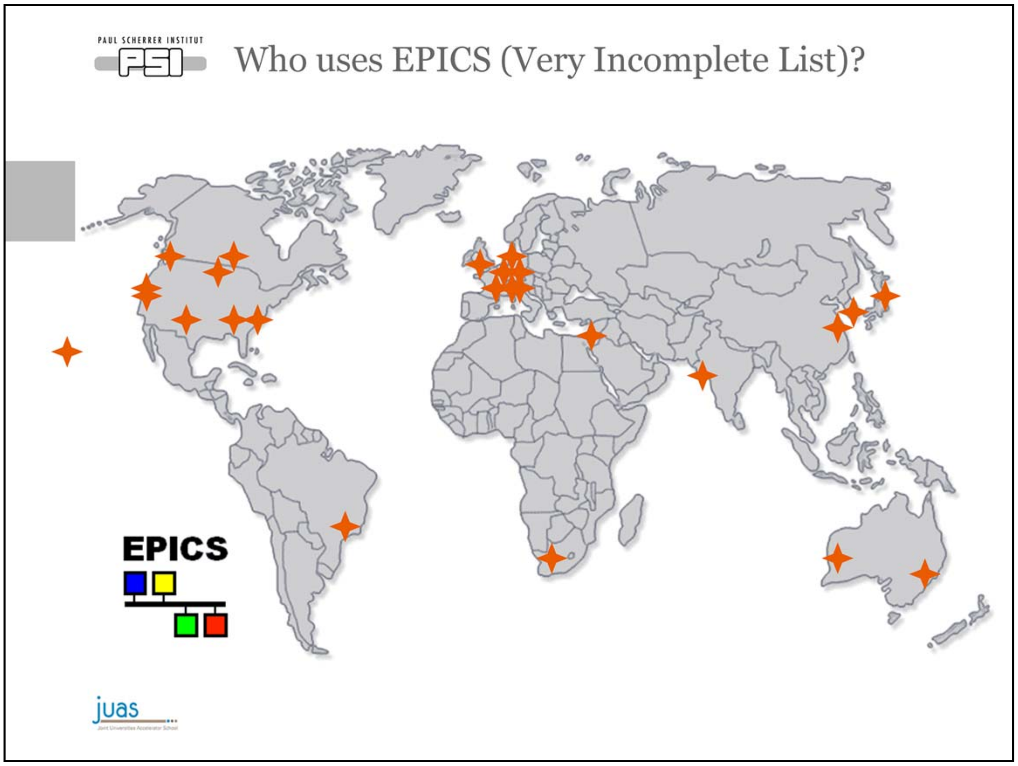
Best solution: all systems are used somewhere to control accelerators successful. All systems support good experiments and produce excellent science. Therefore, the choice of system is political not technical.



Quote from the EPICS web page (http://www.aps.anl.gov/epics/about.php):

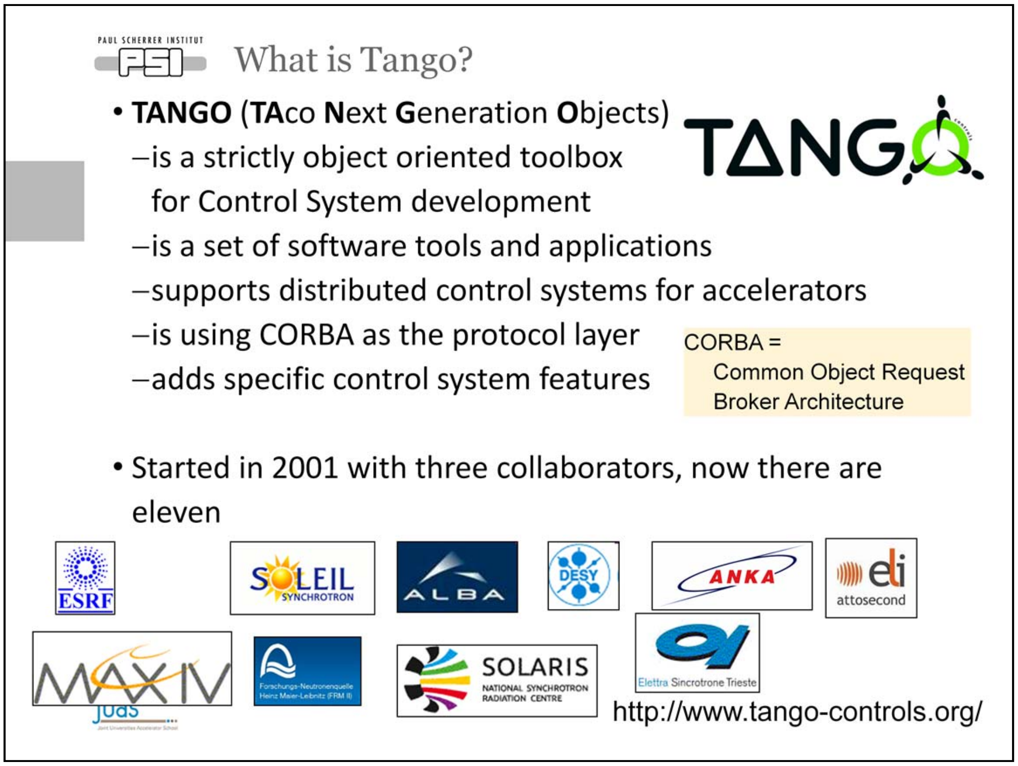
EPICS is a set of software tools and applications which provide a software infrastructure for use in building distributed control systems to operate devices such as Particle Accelerators, Large Experiments and major Telescopes. Such distributed control systems typically comprise tens or even hundreds of computers, networked together to allow communication between them and to provide control and feedback of the various parts of the device from a central control room, or even remotely over the internet.

EPICS uses Client/Server and Publish/Subscribe techniques to communicate between the various computers. Most servers (called Input/Output Controllers or IOCs) perform real-world I/O and local control tasks, and publish this information to clients using the Channel Access (CA) network protocol. CA is specially designed for the kind of high bandwidth, soft real-time networking applications that EPICS is used for, and is one reason why it can be used to build a control system comprising hundreds of computers.



Some EPICS users (very incomplete list):

The Advanced Photon Source at Argonne National Laboratory
Australian Synchrotron
Berlin Electron Synchrotron (BESSY II)
Brazilian Synchrotron Light Source (LNLS)
Deutches Elektronen Synchrotron (DESY)
Diamond Light Source
Fermilab (FNAL)
Jefferson Laboratory (JLAB)
Keck Observatory
KEK B-Factory
Laboratori Nazionali di Legnaro (INFN-LNL)
Lawrence Berkeley National Laboratory (LBL)
Los Alamos National Laboratory (LANL)
Swiss Light Source (SLS/PSI)
Spallation Neutron Source (SNS)
Stanford Linear Accellerator Center (SLAC)



Quotation from the MAX IV web page (https://www.maxlab.lu.se/node/1395):

Tango Scheme

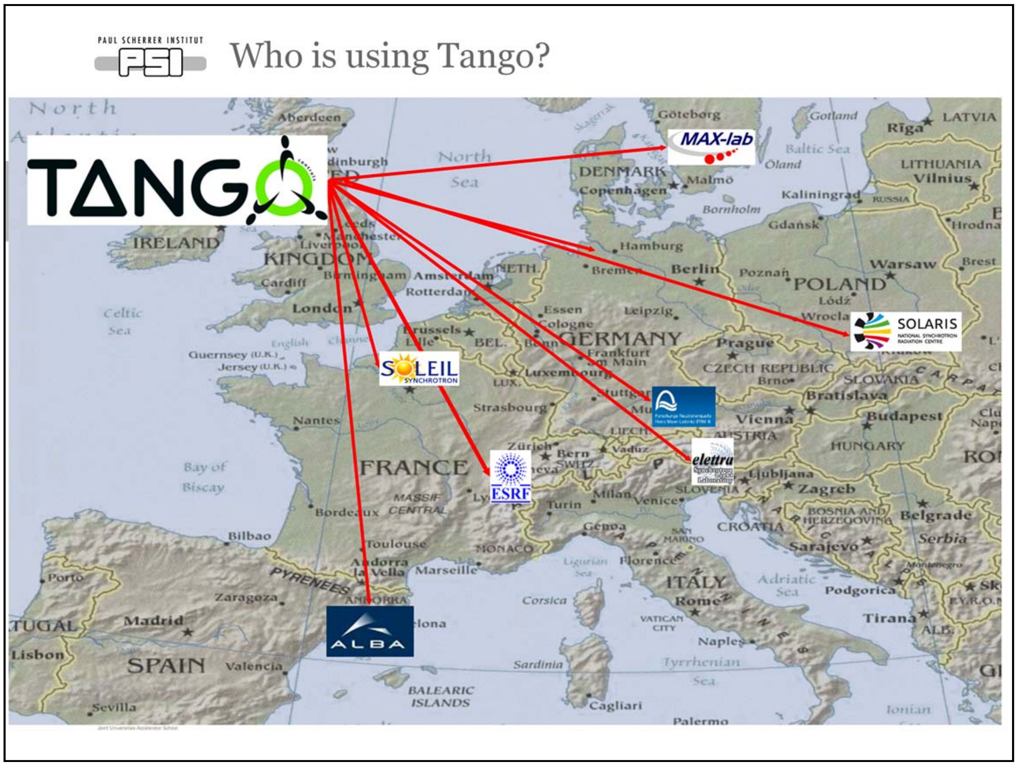
•Tango is based on the Client/Server scheme

•Tango device runs on the server side

•Tango client like Jive, AtkPanel, Matlab binding runs on the client side

•The client communicates with the Tango device through the network [using CORBA]

•Tango Database as a device directory



Quotation from Tango home page (http://www.tango-controls.org/)

“TANGO is an object oriented distributed control system using CORBA and is being actively developed as a collaborative effort between the Alba, Desy, Elettra, ESRF, FRM II, MAX-IV, Solaris and Soleil institutes.”

Alba – Synchrotron radiation facility (Barcelona, Spain)

Desy – Deutsches Elektronen-Synchrotron (Hamburg, Germany)

Elettra – Synchrotron Light Laboratory (Trieste, Italy)

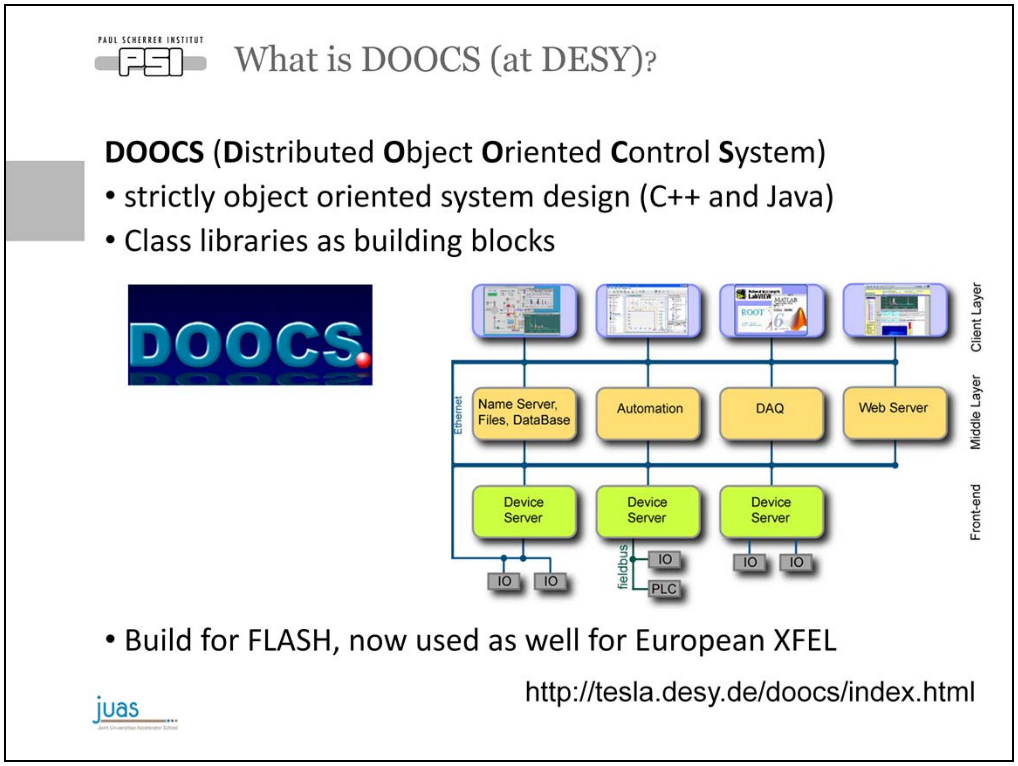
ESRF – European Synchrotron Radiation Facility (Grenoble, France)

FRM II – Forschungsreaktor München II (research neutron source in Garching, Germany)

MAX-IV – Synchrotron radiation facility (Lund, Sweden)

Solaris – Polish National Synchrotron Radiation Center (Krakow, Poland)

Soleil – Synchrotron radiation facility (Paris, France)

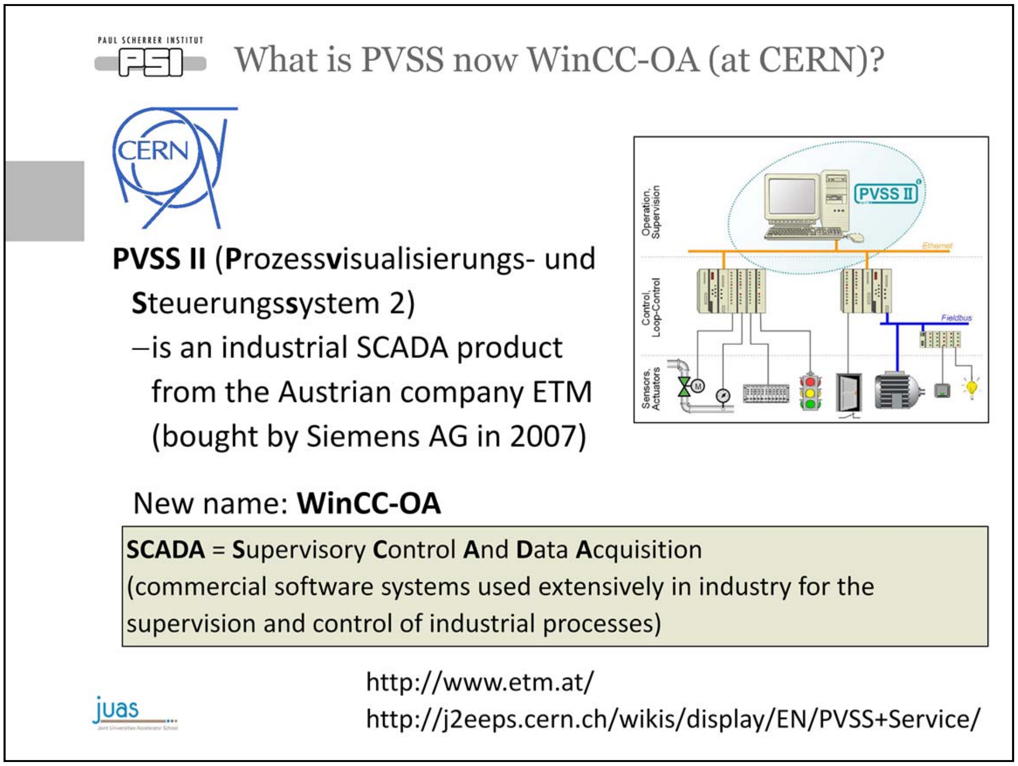


Quotations from the Desy web page (http://tesla.desy.de/doocs/doocs.html):

The Distributed Object Oriented Control System (DOOCS) was designed for the TTF linac (now FLASH). Currently it is extended to control the European XFEL accelerator. It is completely written in C++ language and follows the object oriented programming paradigm [2] since the design idea from the device level, including the device servers, up to the user display is based on objects. Recent developments for the client side applications are written in JAVA to allow them to be used on many computer platforms.

…

The front-end servers represent the lowest layer or tier in the control system. These servers are directly connected to the hardware by different kind of busses: µTCA, VME, SEDAC (a DESY-developed field bus), CAN, PROFIBus, GPIB, Ethernet-based busses like MODBUS etc.). Simple input/output controllers as well as complex ones like PLCs are connected to the device servers within this layer. The middle layer or service tier is used for global services: the equipment name server (ENS), services for automation (e.g. finite state machines), the DAQ, databases and web services. A design goal was to implement complex functions and automation in this tier and not in the client layer. The client layer is used to present the data of the control system to the user. Client programs do not require complex logic or processing and do not hold states of the control system.



From 0407PVSS_II_GettingStarted_Basics_en.pdf found on http://j2eeps.cern.ch/wikis/display/EN/PVSS+Service

What is PVSS II?

"PVSS II" is the German abbreviation for "Process visualization and control system II", a software package

designed for the field of automation engineering. Its main application is in the operation and supervision

of technical installations using VDU workstations with full-graphics capability.

WinCC-OA

SIMATIC WinCC Open Architecture is part of the SIMATIC HMI family and designed for applications of large scale and high complexity as well as projects with special requirements on system prerequisites and customized functionality

(*http://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/simatic-wincc-open-architecture/pages/default.aspx*)

PAUL SCHERRER INSTITUT PSI

Mixed Systems

- At DESY:
  Tango, EPICS, and DOOCS mixed
- At PSI:
  ACS – EPICS migration
- At PSI (former SLS beamline):
  Tango beamline at EPICS accelerator

- There are gateways between the systems

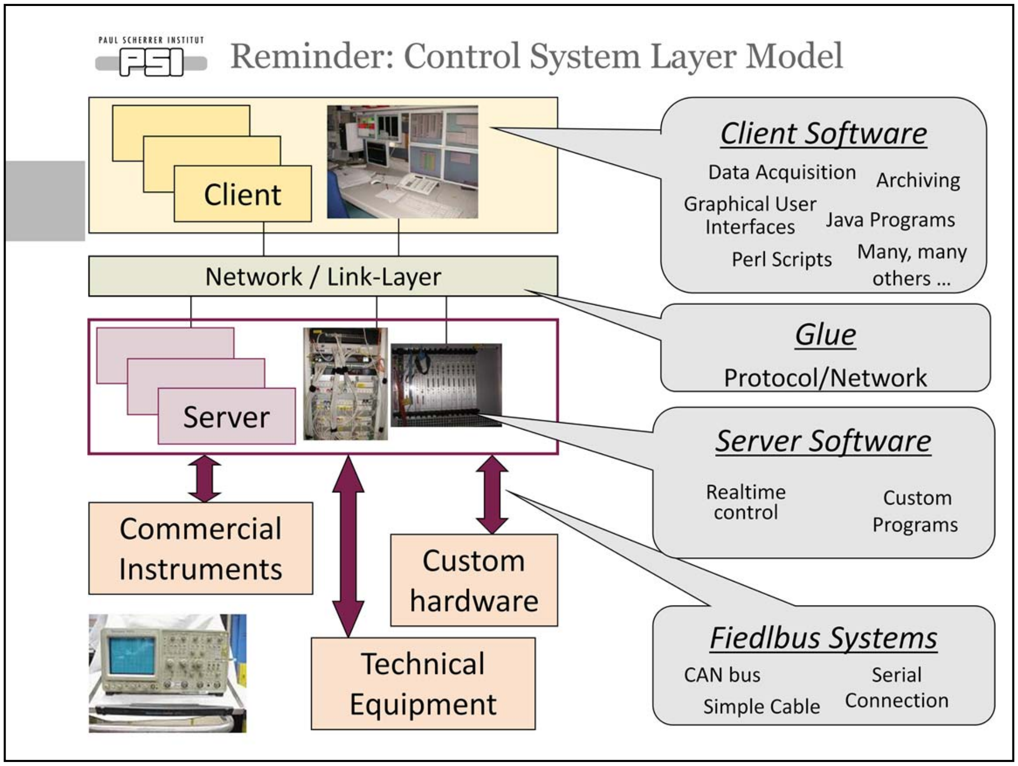
The choice for one system is not exclusive

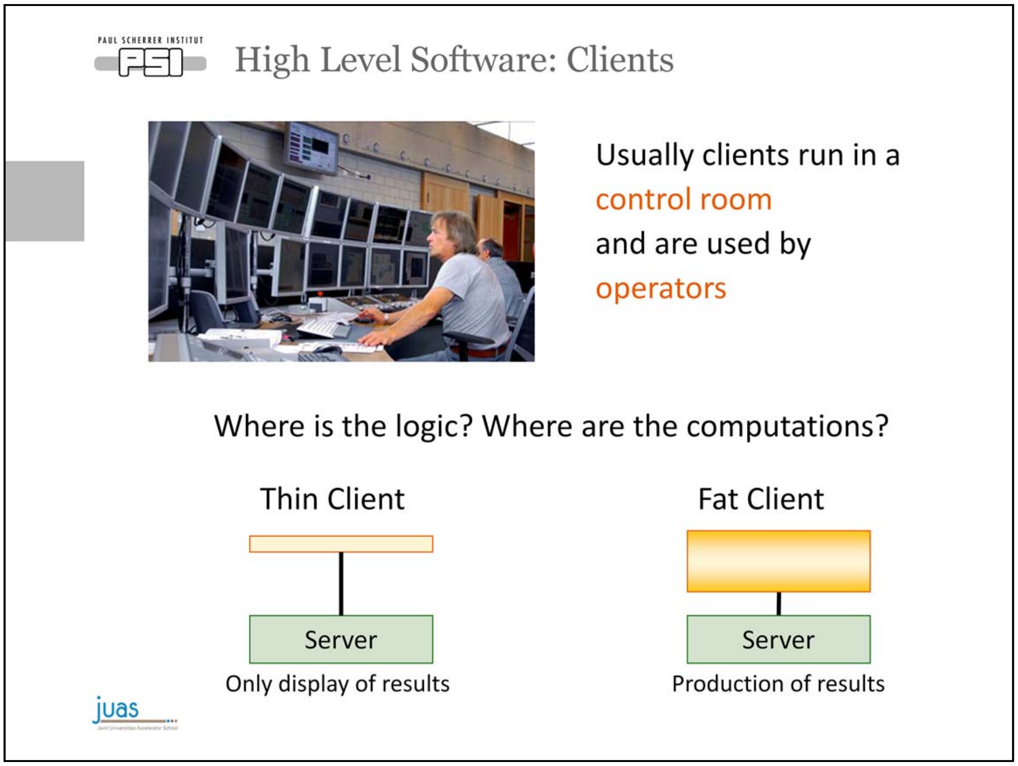
juas
Joint Universities Accelerator School

Table of Content

Technical Requirements

Use open source firmware/software.	• You can change things and you have control of further developments
Use commercial solutions based on open standards developed and sold by a large number of companies	• Don't become dependent on single companies with proprietary solutions
Use standards with a long life-time (20 years+)	• Keep long lifecycles of accelerators in mind


PAUL SCHERRER INSTITUT
PSI
Reminder: Control System Layer Model
Client
Network / Link-Layer
Server
Commercial Instruments
Custom hardware
Technical Equipment
Client Software
Data Acquisition
Archiving
Graphical User Interfaces
Java Programs
Perl Scripts
Many, many others ...
Glue
Protocol/Network
Server Software
Realtime control
Custom Programs
Fiedlbus Systems
CAN bus
Serial Connection
Simple Cable



Control Room

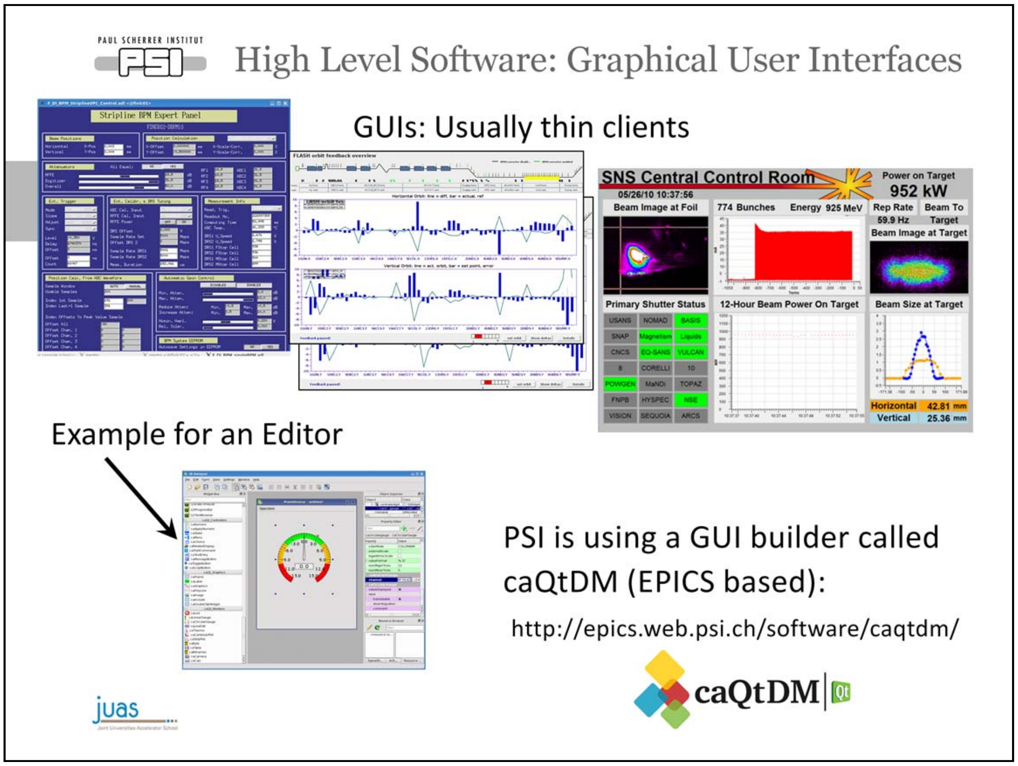
There might be several control rooms: one for the accelerator and one for each beamline or experiment. Or there can be several control rooms around the world in a huge collaboration.

Operators

There might be dedicated personel for running an accelerator or experiment. Or the scientists and system experts can be on control room duty for some part of their working time.

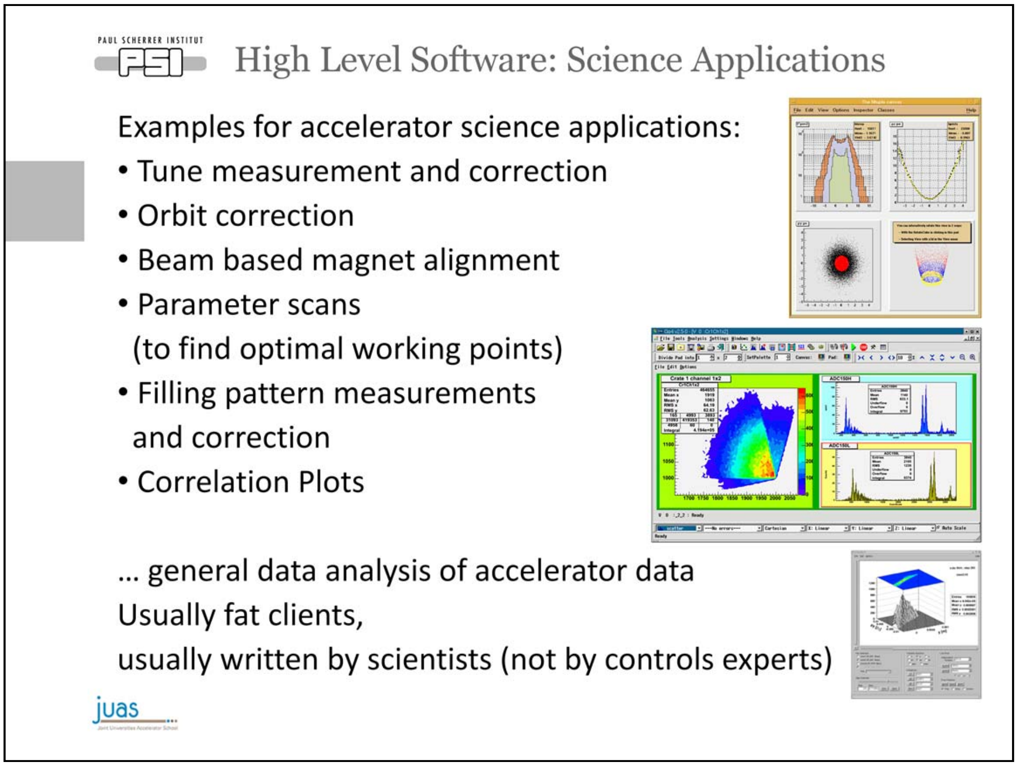
Thin Clients: Only displays results that are produced in some server elsewhere. Therefore, the display is decoupled from the calculations or logic. Both can be changed independently (for example for updates or new features). Mostly used for expert screens that display the complete interface of the device.

Fat Clients: The results are produced inside the client software. The display is part of the same software. Everything is kept together but no other client can access the results. Usually done by scientists to cover their complete measurement, for example in MATLAB.



Home page of JDDD (Java DOOCS Data Display): http://jddd.desy.de/

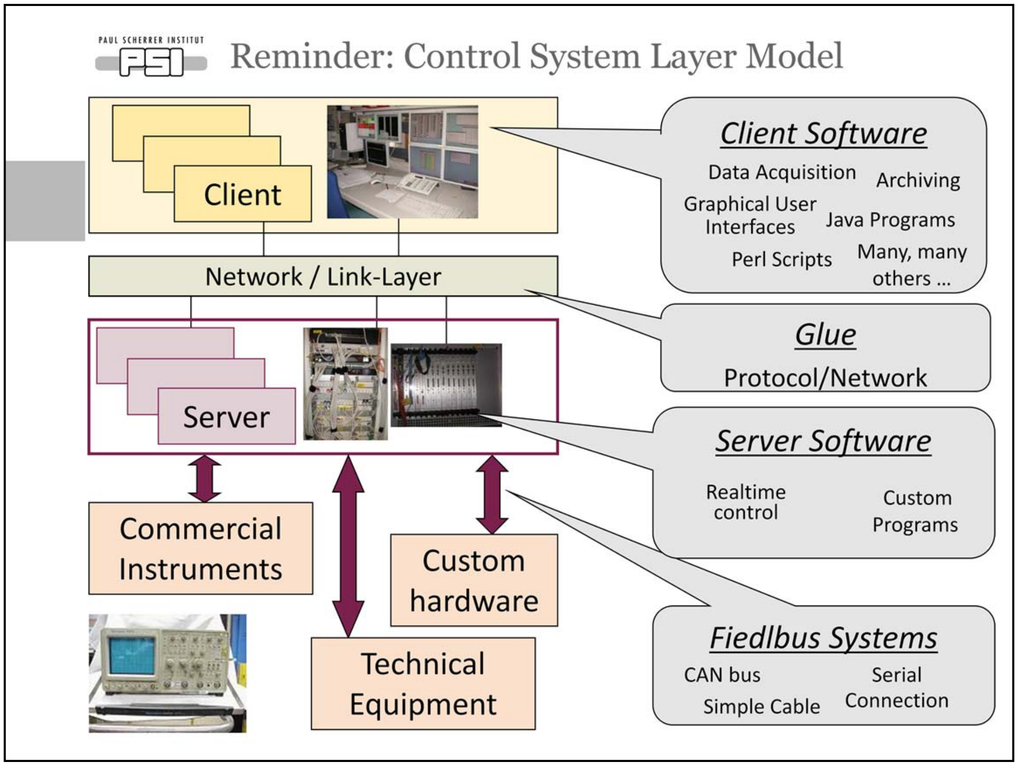
BOY (Best OPI, Yet) Operator Interface for Control System Studio (EPICS based):
http://sourceforge.net/apps/trac/cs-studio/wiki/BOY

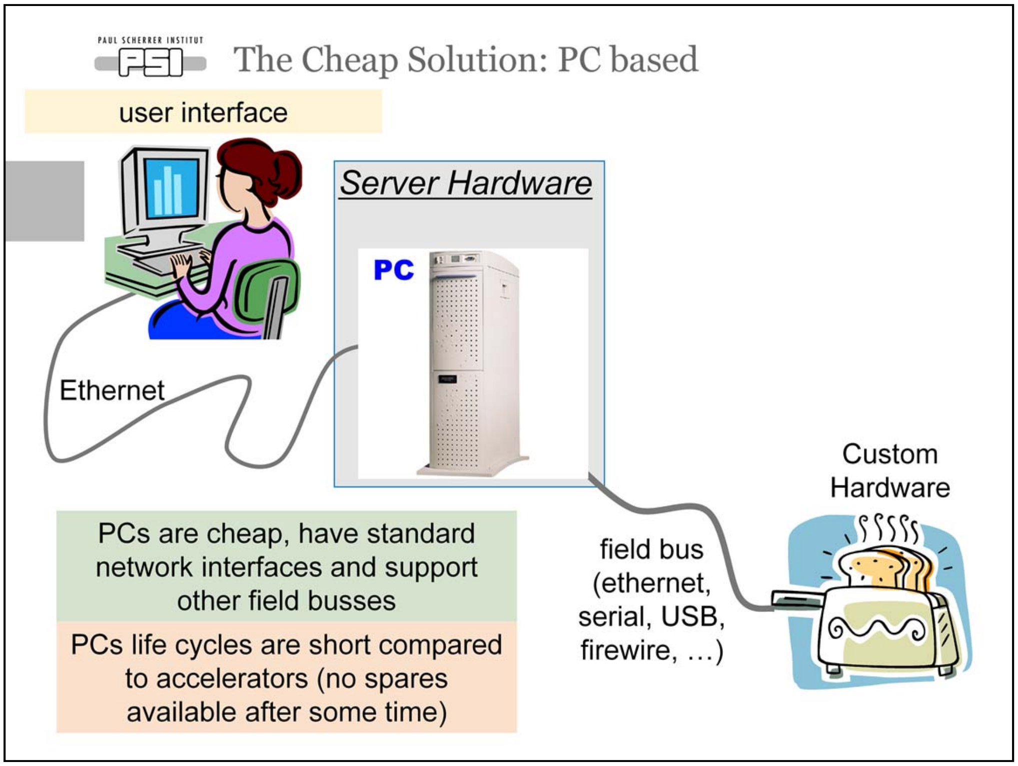


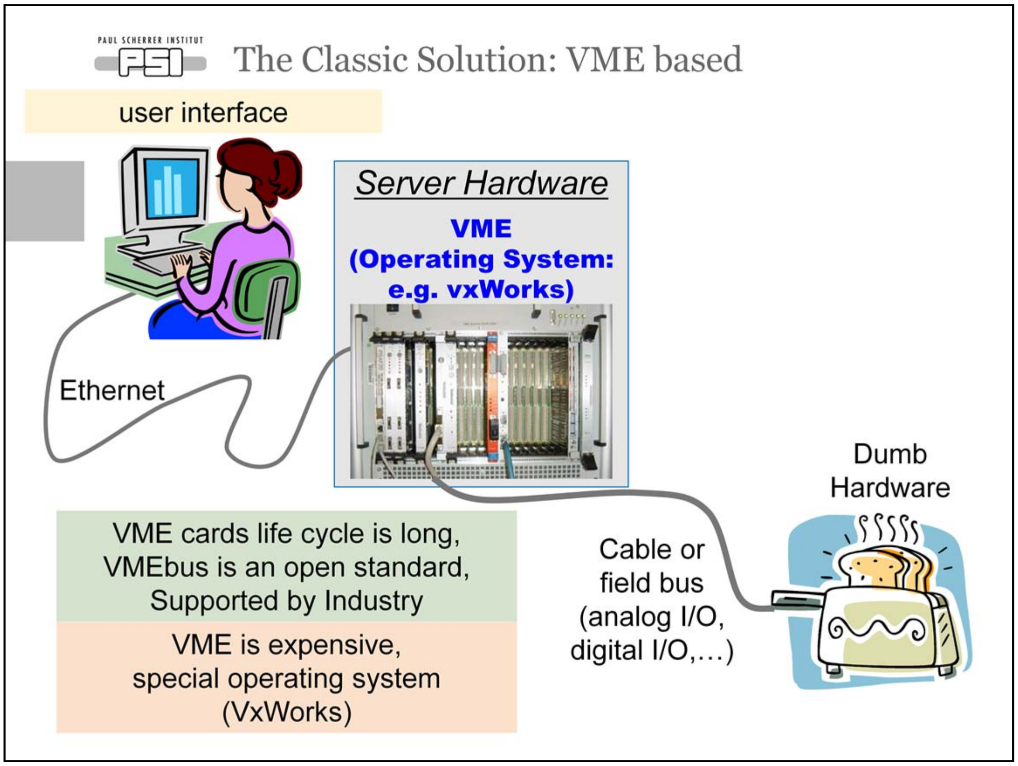
Science Application often use or result in software libraries for specific purposes.

For example ROOT is an object-oriented program and library developed by CERN (http://root.cern.ch/drupal/)

Other labs use MATLAB from Mathworks for science applications.


Reminder: Control System Layer Model
Client
Network / Link-Layer
Server
Commercial Instruments
Custom hardware
Technical Equipment
Client Software
Data Acquisition
Archiving
Graphical User Interfaces
Java Programs
Perl Scripts
Many, many others ...
Glue
Protocol/Network
Server Software
Realtime control
Custom Programs
Fiedlbus Systems
CAN bus
Serial Connection
Simple Cable


The Cheap Solution: PC based
user interface
Server Hardware
PC
Ethernet
Custom Hardware
field bus (ethernet, serial, USB, firewire, …)
PCs are cheap, have standard network interfaces and support other field busses
PCs life cycles are short compared to accelerators (no spares available after some time)


The Classic Solution: VME based
user interface
Server Hardware
VME
(Operating System:
e.g. vxWorks)
Ethernet
Dumb
Hardware
Cable or
field bus
(analog I/O,
digital I/O,…)
VME cards life cycle is long,
VMEbus is an open standard,
Supported by Industry
VME is expensive,
special operating system
(VxWorks)

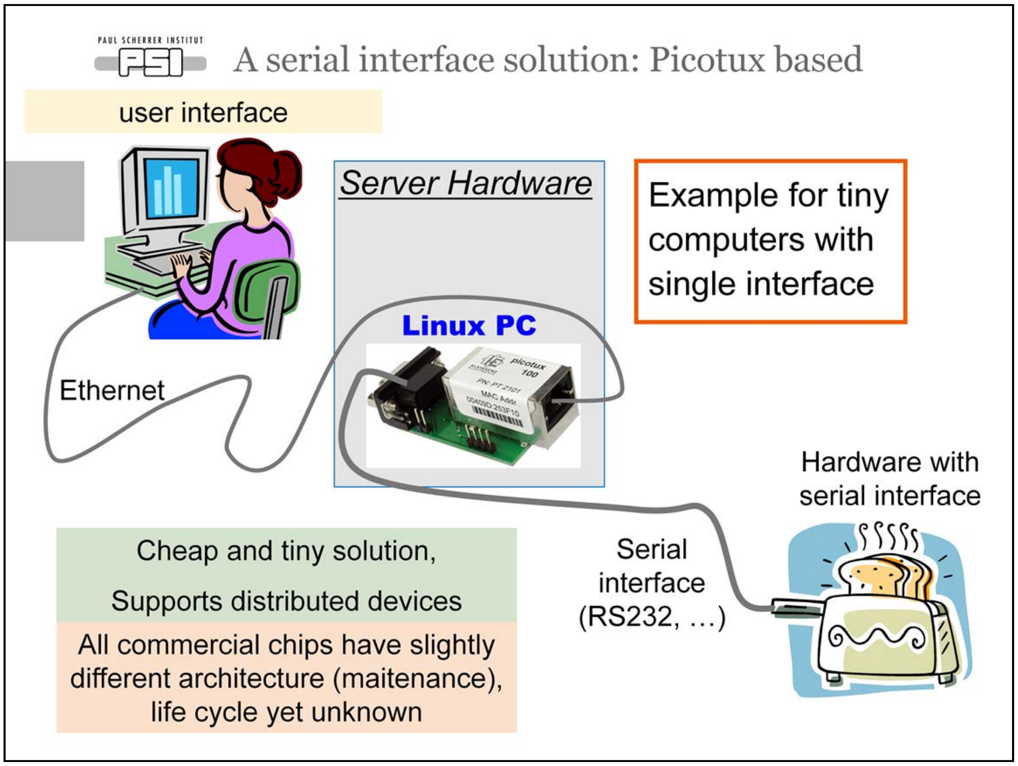
What is a VME Computer?

- VME is an abbreviation for **V**ERSA**m**odule **E**urocard
- Industry Computer based on VMEbus
- Developed since 1980
- It is not a PC
- Real-time capable (i.e. delays are calculable)
- Common used operating system is VxWorks from Wind River company (open source alternative: RTEMS)
- Expensive (~800 Euro per interface card)

VME Crate

VME Card:
Eurocard size
VMEbus interface

http://en.wikipedia.org/wiki/VMEbus


PAUL SCHERRER INSTITUT
PSI
A serial interface solution: Picotux based
user interface
Server Hardware
Example for tiny computers with single interface
Linux PC
picotux 100
Ethernet
Hardware with serial interface
Serial interface (RS232, …)
Cheap and tiny solution,
Supports distributed devices
All commercial chips have slightly different architecture (maitenance), life cycle yet unknown

The Embedded Solution: Device Integrated CPU

user interface

Ethernet

Embedded Hardware

=

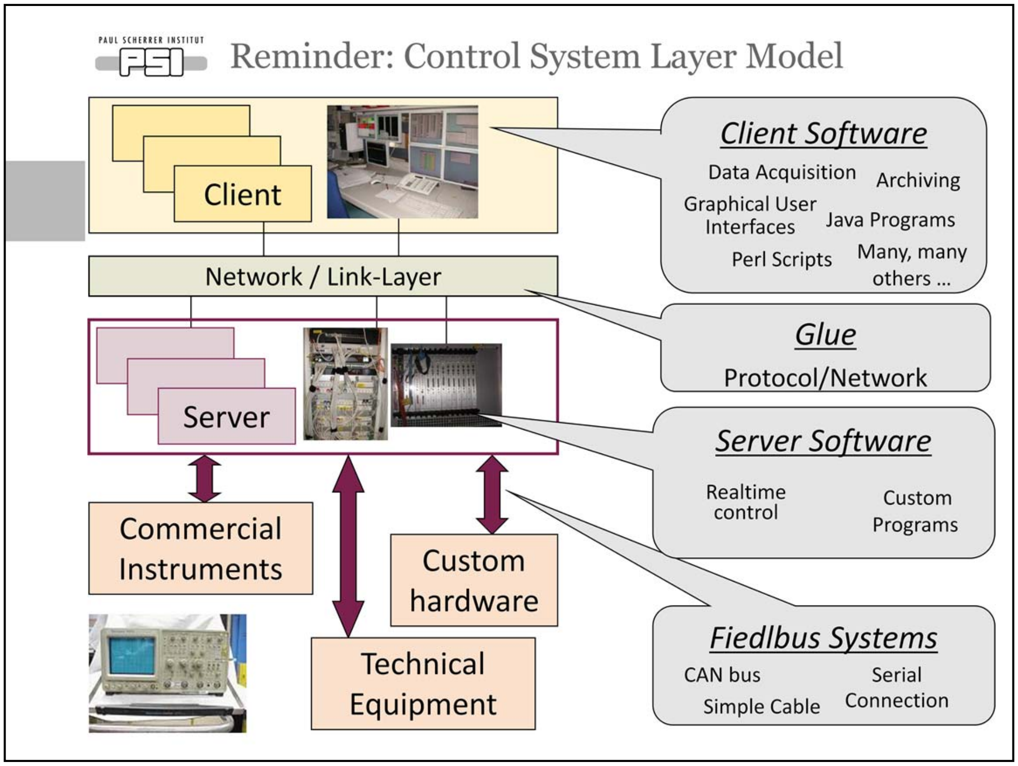
Server Hardware

+

Instrument

Low cost, have standard network interfaces and support distributed devices

All commercial chips have slightly different architecture (maitenance), life cycle yet unknown


PAUL SCHERRER INSTITUT
PSI
Reminder: Control System Layer Model
Client
Network / Link-Layer
Server
Commercial Instruments
Custom hardware
Technical Equipment
Client Software
Data Acquisition
Archiving
Graphical User Interfaces
Java Programs
Perl Scripts
Many, many others …
Glue
Protocol/Network
Server Software
Realtime control
Custom Programs
Fiedlbus Systems
CAN bus
Serial Connection
Simple Cable

PAUL SCHERRER INSTITUT PSI

What are PLCs?

- **PLC** (**P**rogrammable **L**ogic **C**ontroller)
  - is a digital computer used to connect “dumb” devices
- the PLC is designed
  - for multiple inputs and outputs
  - extended temperature ranges
  - immunity to electrical noise
  - resistance to vibration and impact
  - as a real time system
- Programs are typically stored in battery-backed or non-volatile memory
- Products from different providers can **NOT** be mixed!

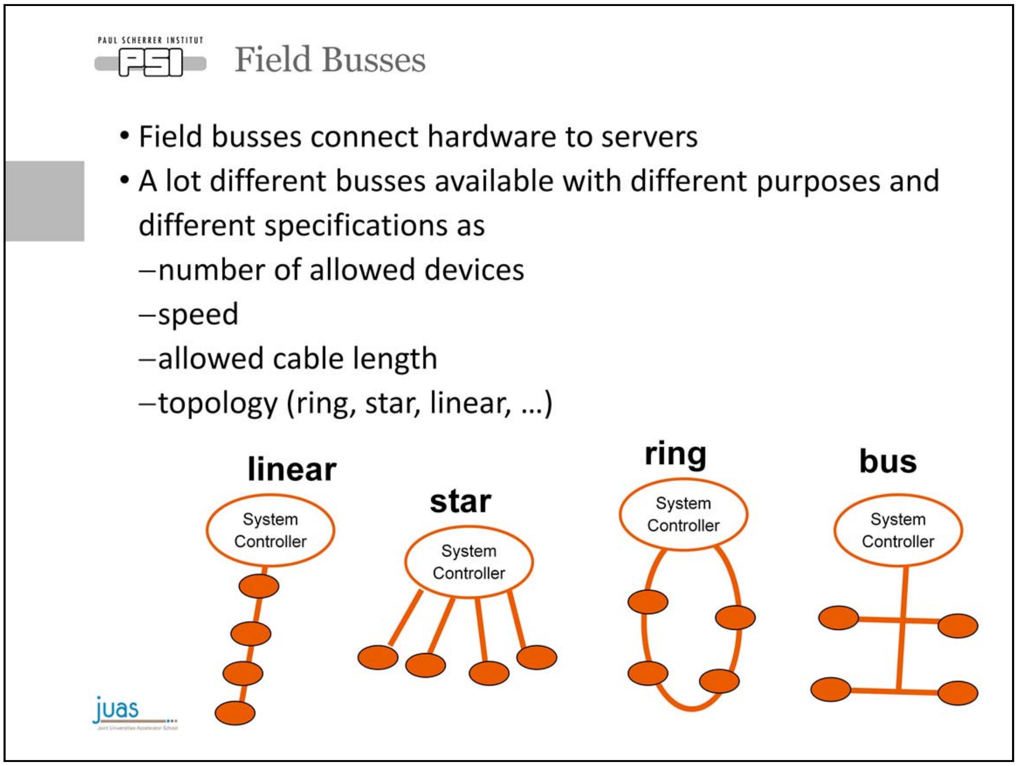
juas

Examples for companies:

•Siemens S7
(http://www.automation.siemens.com/mcms/programmable-logic-controller/en/Pages/Default.aspx)

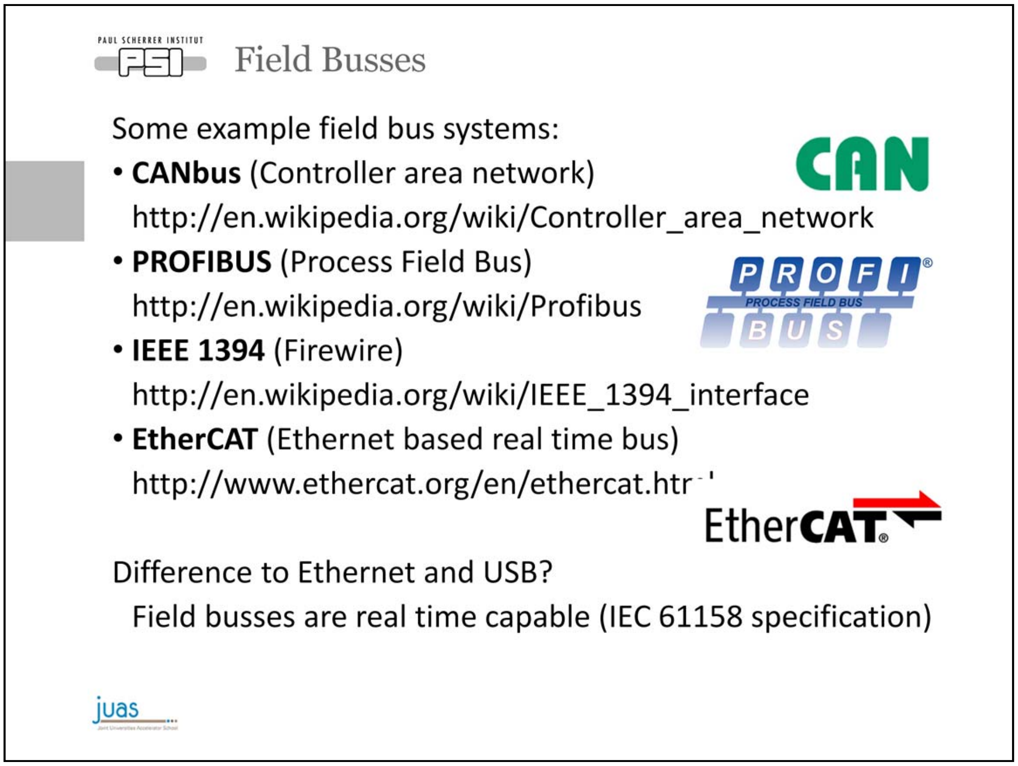
•Allen-Bradley
http://www.ab.com/programmablecontrol/plc/

•Beckhoff
http://www.beckhoff.ch/



From Wikipedia:

Fieldbus is the name of a family of industrial computer network protocols used for real-time distributed control, now standardized as **IEC 61158**.



Definition of Real-time:

The worst case for the reaction time from signal in to reaction out can be determined. Real time is not necessarily fast (but often).

Ethernet and **USB** are used in place of field busses in accelerators where no real time capability is needed. For example when

Table of Content

Borderlands of Control Systems

Accelerator Control Systems have fussy borders.

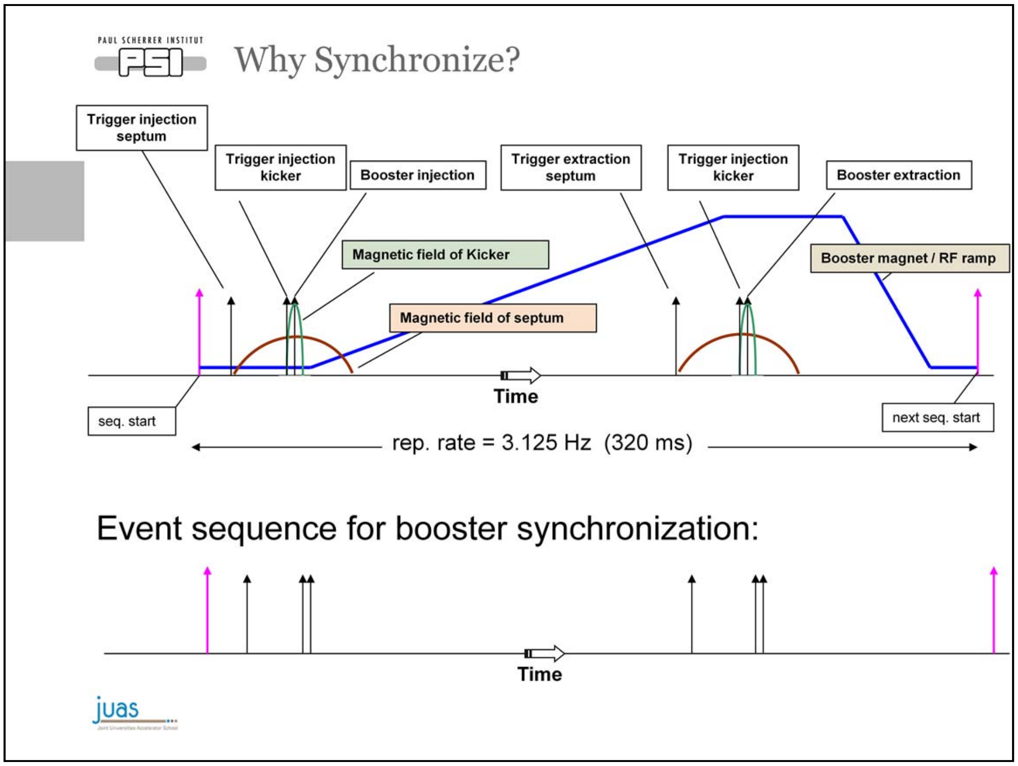
Some example for these borders are:

1. Timing and Synchronisation
2. Feedback Systems
3. Interlock-, Alarm-, and Machine Protection Systems
4. Experiment Data Acquisition
5. Relational Databases
6. Relationship of IT (Information Technology) and Controls

All of these fields are needed in accelerator control. For all of these topics the responsibilities have to be clarified.

For example

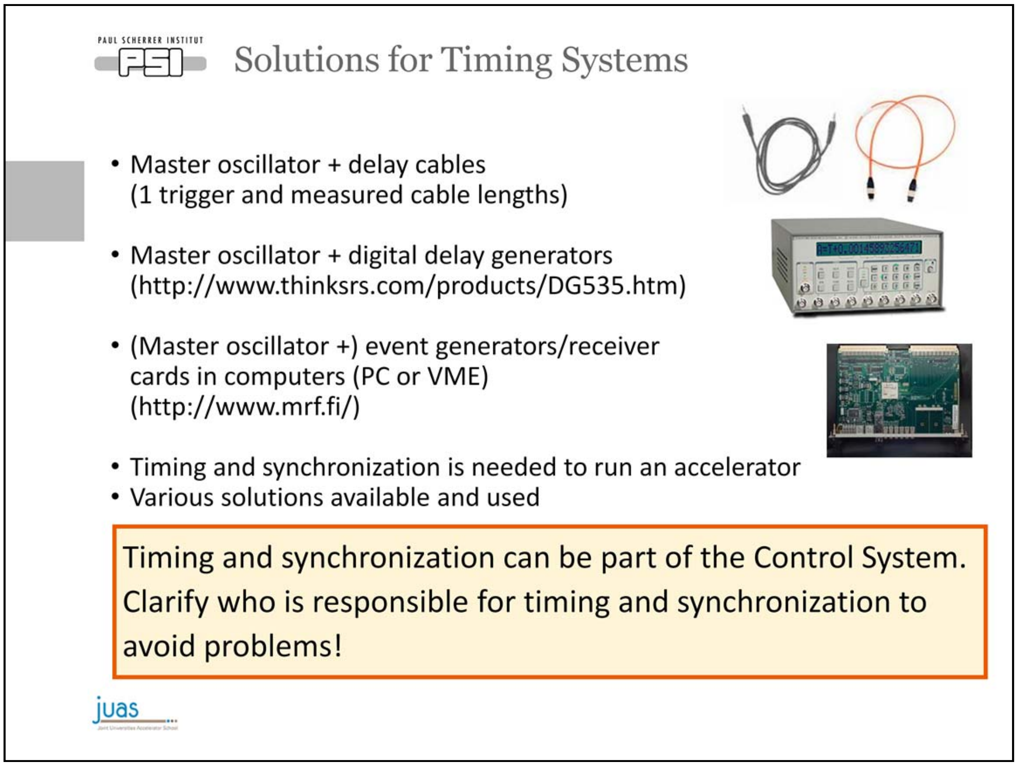
1. Timing and Synchronisation



Triggers might be needed in some advanced time to allow real hardware to ramp up or get the full field.

Especially important timing and synchronisation are for pulsed accelerators like FELs.

The example is taken from the booster synchrotron timing from the Swiss Light Source (PSI).



Accuracy of the timing signal can vary from some pico seconds down to femto seconds – but femto seconds are really hard to get:
http://cbp.lbl.gov/felminiworkshop/pdf/Byrd_LCLSTiming.pdf

Technology depends on needed accuracy and available budget.

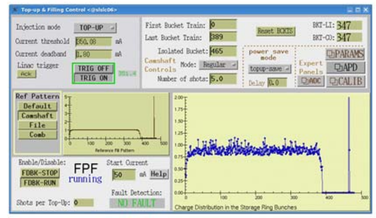
For example

2. Feedback Systems

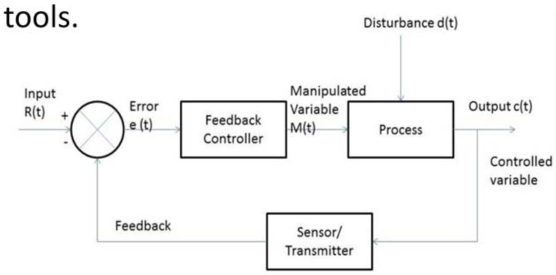
PAUL SCHERRER INSTITUT

Feedback Systems

For example:
•Orbit Feedback (Position)
•Energy Feedback
•Filling pattern Feedback

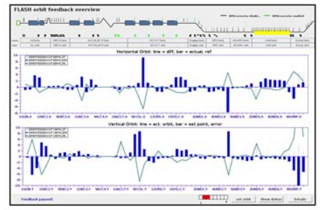


If it needs to be fast, it needs separate cables!
Slow feedbacks can be realised with standard control system tools.



juas
Joint Universities Accelerator School

Example: Orbit Feedback



Needed for beam position stability.

Measurement (once in a time):

- Measure beam response matrix (complete orbit for different corrector magnet settings)
- Invert the matrix (normally not possible analytical) a stable method is singular value decomposition (SVD)

Feedback during runs:

- Measure the beam position and correct it with the appropriate set of correctors

juas

http://en.wikipedia.org/wiki/Singular_value_decomposition

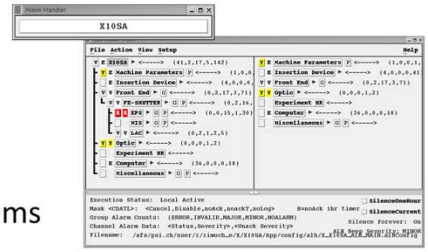
For example

3. Interlock-, Alarm-, and Machine Protection Systems

What are Alarms

Murphy's law:

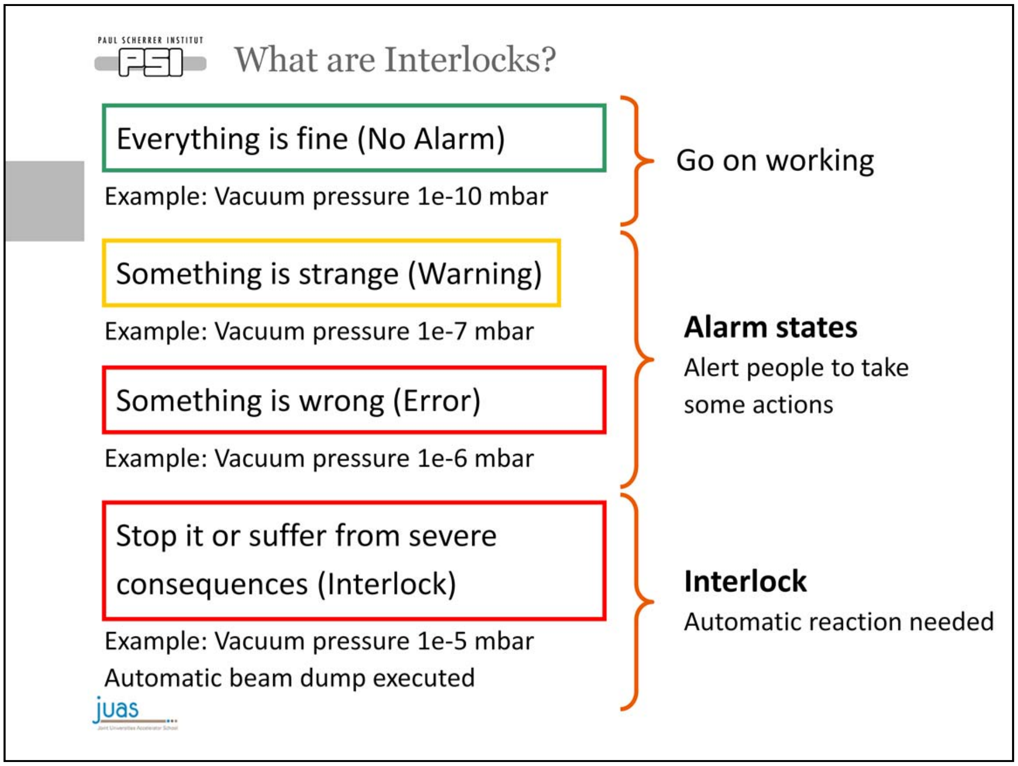
Anything that can go wrong will go wrong.



EPICS Alarmhandler

- Alarms help to avoid Real Problems
- Alarms help to find problems
- Example:
  - Beam position more than 1 mm of from reference
  - Vacuum pressure higher than 1e-6 mbar
  - Orbit Feedback Program not running

- People should react on alarms



All interlock and safety systems need to be hard wired and not involve software (if possible).

Training of the operators need to include alarm handling. As often several alarms occure at the same time (or at the same time as far as a human can see) this can be complicated. For running machines, alarm handling is one of the two main tasks of the operators (the other is setting up and tuning of different machine modes).

The example vacuum pressures are taken from the Swiss Light Source.

Interlock Systems

- Interlock Systems have to be
  - taking automatic actions (no people involved) - fast
  - Reliable (99% might not be enough)
  - as simple as possible (see Murphy's law)
- Avoid computers in Interlock Systems
  Decouple "running" the accelerator (=Control System)
  from "stopping" the accelerator (=Interlock System)
- There can/will be more than one Interlock System in an accelerator (local, global, different goals, etc.), for example:
  - Vacuum Interlock
  - Equipment Protection System
  - local RF Interlock Systems

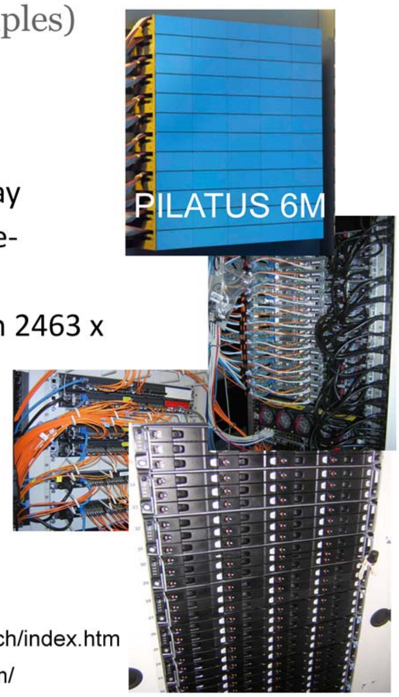
Clarify who is responsible for Interlock Systems to avoid problems!

For example

4. Experiment Data Acquisition

Data Acquisition (Examples)

- **PILATUS 6M Detector**
  (Synchrotron-Beamline at SLS):
  – two-dimensional hybrid pixel array detectors, which operate in single-photon counting mode
  – composed of 5 x 12 modules with 2463 x 2527 pixels
  – Framing rate12 Hz for ca. 6 MByte = 72 MB/s
  – at full speed: 8 hours ≈ 1.6 TeraByte



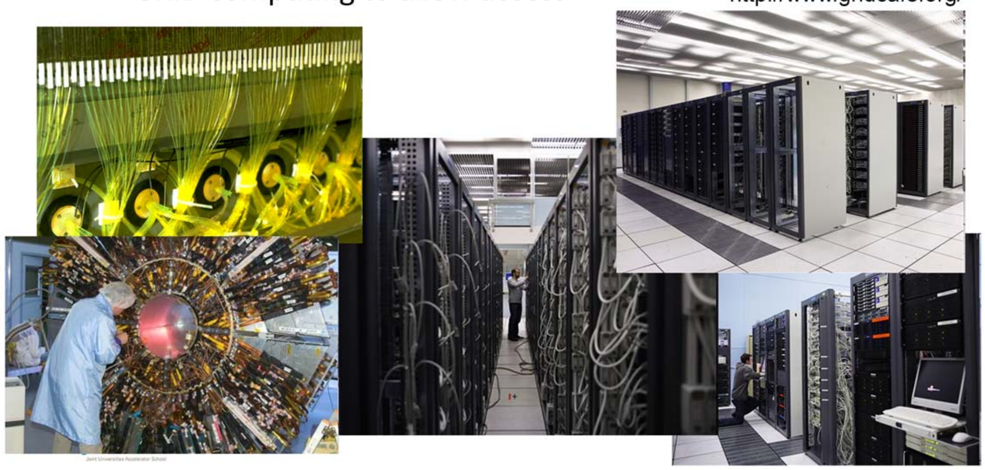
http://pilatus.web.psi.ch/index.htm

http://www.dectris.com/

Data Acquisition (Examples)

- The Large Hadron Collider will produce roughly 15 petabytes (15 million gigabytes) of data annually – enough to fill more than 1.7 million dual-layer DVDs a year!
  - –GRID computing to allow access

http://www.gridcafe.org/



Is Data Acquisition Controls?

- Data Acquisition requires
  - Network infrastructure
  - Computer storage infrastructure
  - Server infrastructure for data access
  - Environment (e.g. Grid) for data access
  - Manpower for setup and maintenance
- Detectors
  - can provide information about accelerator (beam position)
  - need to be adjusted to accelerator setup (connection to control system needed)
- Some detectors (e.g. BPMs) are part of the accelerator anyway

Not necessary

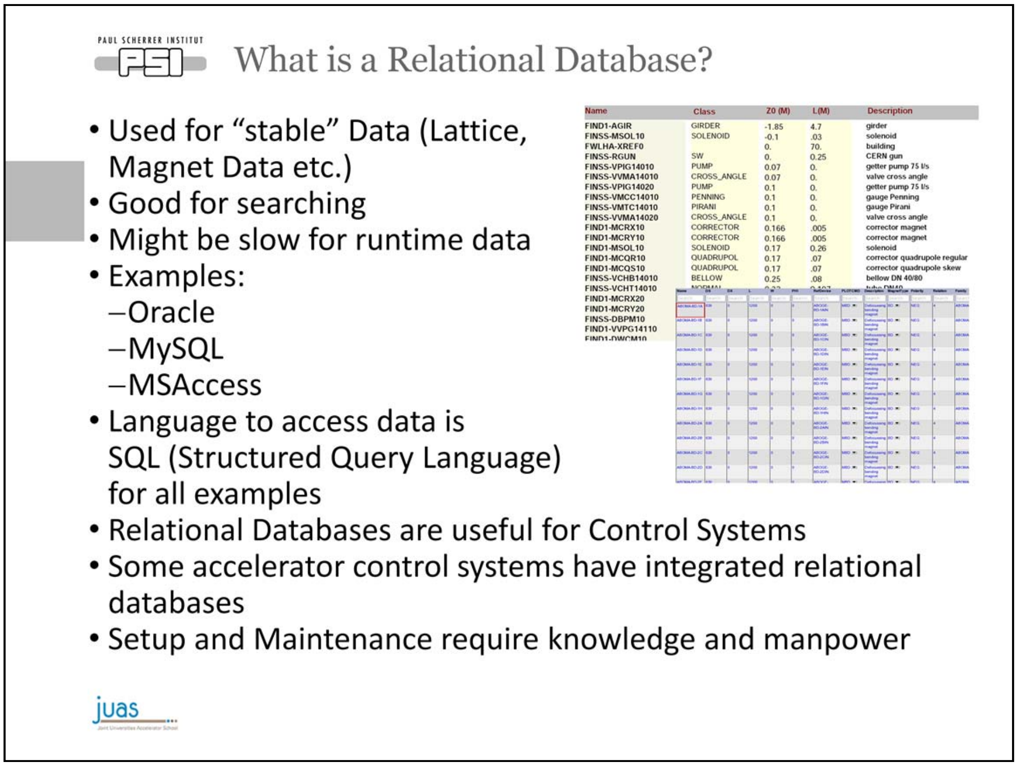
Yes its needed

Has to be discussed to avoid problems!

juas

For example

5. Relational Databases



In most cases the setup of the database framework (i.g. installation of the database) is consideret an IT task. The creation, composition, and fill up with data might be considered a controls task (or even a task for beam dynamics physicists).

For example

6. Relationship of IT (Information Technology) and Controls

Who is Responsible for What?

- Most large research institutes have a Controls Group in addition to a IT Group
- Why separate IT from Controls?

IT	Controls
• Office PC installation • Operating Systems for Office applications • Infrastructure (network cables) • Central Services (Computing Cluster, Server Room …)	• Accelerator computer installation • Integration of accelerator hardware • Control Room applications • Distributed processes

Databases, Timeserver, Network, Security

Controls is dependent on IT.
Responsibilities have to be discussed to avoid problems!

IT = Information Technology (includes Computers but also Telephone, Mobile …)

Table of Content

- What is an Accelerator Control System?
- Accelerator Control Systems Architecture
- Examples of Control Systems
- Control System Parts and Pieces
- Borderlands of Control Systems
- Conclusion

PAUL SCHERRER INSTITUT PSI

Summary: What is Accelerator Controls

- It is hard to define – but every Accelerator has one
- It is organized in layers separating hardware from applications
- It is (has to be) a distributed system, involving some network protocols
- The borders are not clearly defined
  - For example: Where starts the hardware responsibility (PLCs, embedded systems)?

Definition:
An **Accelerator Control System** is a computer environment that allows remote access to the accelerator hardware with a lot of different functionality to satisfy the requirements of several different user groups.

juas

It is not considered science but needs a lot knowledge about science (physics and computer science, sometimes politics)

References

Bad news: There is no book on Accelerator Control Systems
Good news: You can find some things in the Internet

- ICFA Newsletter Number 47 (December 2008) on Control System: http://icfa-usa.jlab.org/archive/newsletter/icfa_bd_nl_47.pdf

- EPICS: http://www.aps.anl.gov/epics/
- Tango: http://www.tango-controls.org/
- CERN Controls Group: https://controls.web.cern.ch/Controls/
- PSI Controls Group: https://controls.web.psi.ch/cgi-bin/twiki/view/Main

…search the institute web pages …

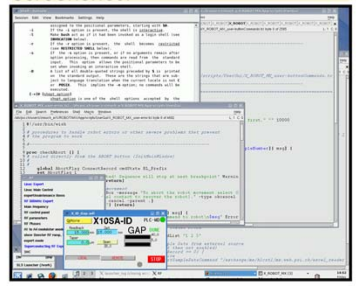
- International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS): http://www.icalepcs.org/

What to Learn as a Controls Guy?

1. Be curious about what your customers do (accelerator physics, experiments, medical treatment, etc.)
2. Enjoy programming
   – Script Language (phython, tcl/tk, etc.)
   – Object Oriented (Java, C++, etc.)
3. Enjoy computer environments

- Useful skills include (non-essential)
  – Basic knowledge in Accelerator Physics or general Physics
  – Database structures/sql commands
  – Linux and/or Windows administration
  – Network administration
  – PLC, FPGA or DSP programming (nearly electronics)
  – Graphical User Interface design

Quick test:
Do you feel comfortable with this screenshot?



The End