



WIR SCHAFFEN WISSEN – HEUTE FÜR MORGEN

Elke Zimoch :: Section Controls :: Paul Scherrer Institut

Accelerator Controls

JUAS 2019

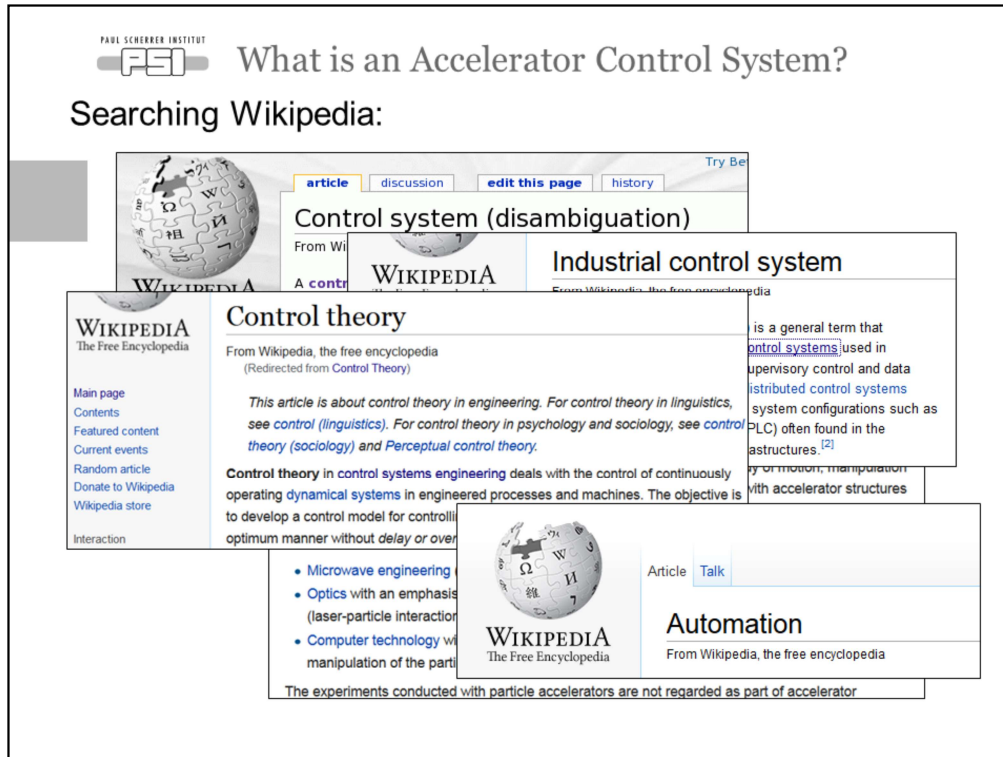


Soon in the future (and once upon a time):
Scientist **Dr. Example Guy** wants to do
VeryImportantMeasurement_OneDotOne
for that he creates some actuators and detectors
Super_Creative_HardwareSolution
puts it into the accelerator
and calls the Controls Group
“Please make it run”.

I want to teach you a
minimum awareness about the control
system that «runs» the accelerator ...



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



- From Wikipedia (https://en.wikipedia.org/wiki/Main_Page):
- **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 in control systems engineering deals with the control of continuously operating dynamical systems in engineered processes and machines. The objective is to develop a control model for controlling such systems using a control action in an optimum manner without delay or overshoot and ensuring control stability.
- **Control Systems Engineering:** Control engineering or control systems engineering is an engineering discipline that applies automatic control theory to design systems with desired behaviors in control environments. The discipline of controls overlaps and is usually taught along with electrical engineering at many institutions around the world
- **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.
- Best match found on Wikipedia is the article about **Distributed Control Systems**
https://en.wikipedia.org/wiki/Distributed_control_system

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

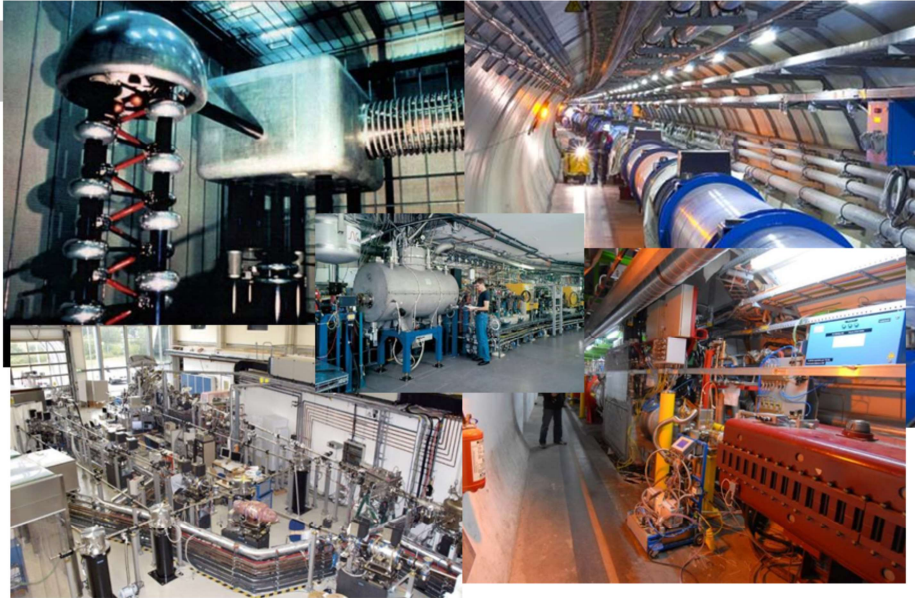
- Controls the accelerator (Source, Magnets, RF)
- Provides diagnostics information (BPMs, Cameras)
- Monitors environment (Vacuum, Temperature)
- Feedback programs for beam parameters (orbit feedback)
- Makes “the machine” running and controllable ...

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

- 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? (2/6)

Controls the accelerator hardware:

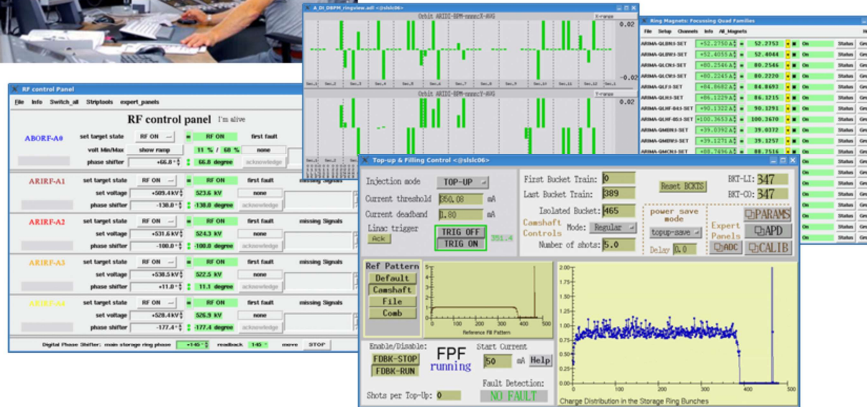


- **What does an Accelerator Control System actually control?**

- 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 (storage ring with 130 m diameter): 50 km power cable and more than 500 km signal cable



Make the accelerator controllable
 ... from a Control Room
 ... using Computer Systems

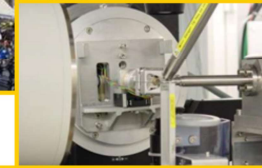


• For what do you need an Accelerator Control System (continued)?

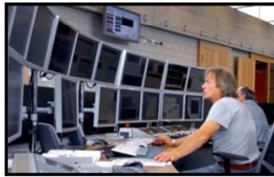
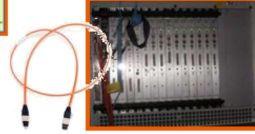
- Some examples of applications and programs needed to run an accelerator:
 - Device Control Panels
 - Feedback Systems (e.g. orbit feedback, filling pattern feedback)
 - Archiving and Data Acquisition Systems (Storage)
 - Scan tools for experiments and measurements
 - Conditioning tools for RF components
 - Simulations and comparisons with real accelerator
 - Alarm handling and machine protection



Control System

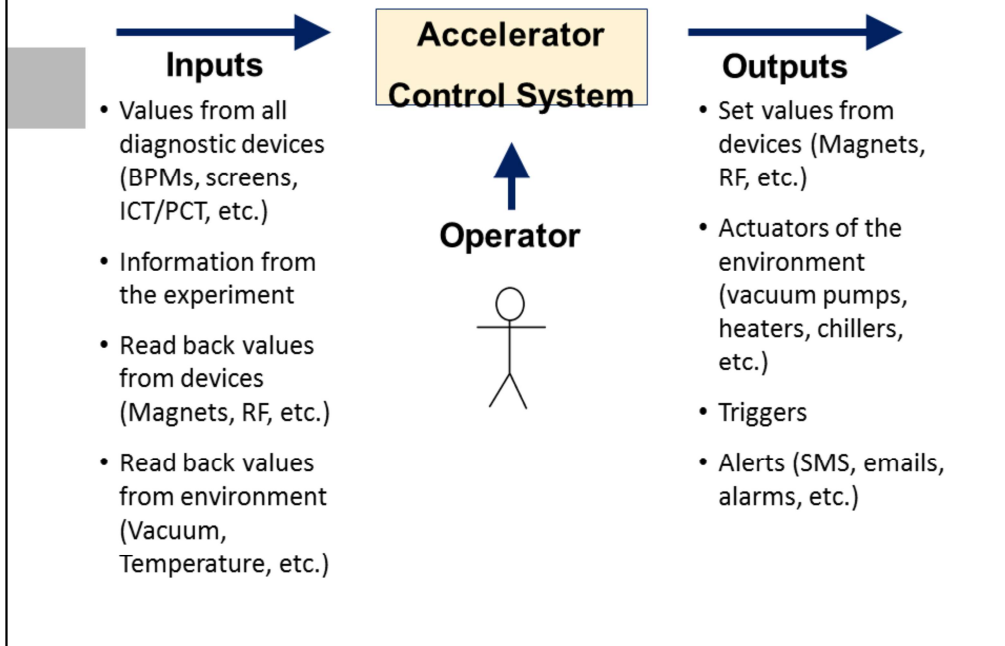


Operator
in Control Room



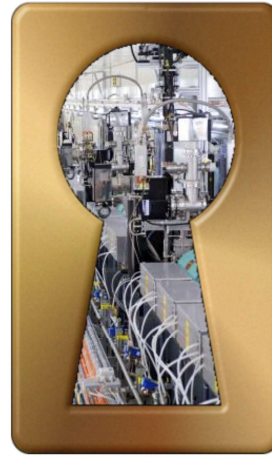
The control system connects the operator with the accelerator.

- 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



The Accelerator Control System

- does provide a keyhole view on the accelerator
- is the only way to access any component remotely



• **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.

Who uses an Accelerator Controls System

Who they are

- Accelerator Physicists
- Operators (technical Staff, in most cases no theoretical background knowledge)
- System Experts (Vacuum Experts, RF Group, ...)
- Experiment Users (not necessary Physicists)
- Sponsors (Politicians, General Public, etc.)
- Control System Specialists (Computer Scientists, Physicists, Nerds)

What they want from the system

- Access to ALL functions of the hardware (full control)
- Implementation of complex algorithms
- Easy and intuitive usage
- Low cost, low manpower
- Safe usage and reliable alarm handling
- Easy maintainable
- Easy extensible
- **fun**

- 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

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
Accelerator Control System:
tries to unify the access to different
hardware
(one way to rule them all)

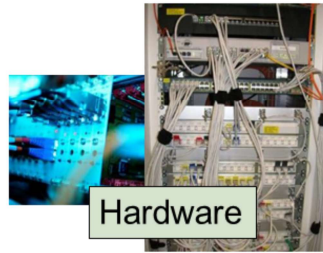
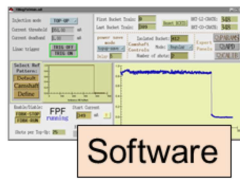


0110001010001001



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

Requirements of an Accelerator Control System



- reliable
- good performance
- flexible
- easy maintenance

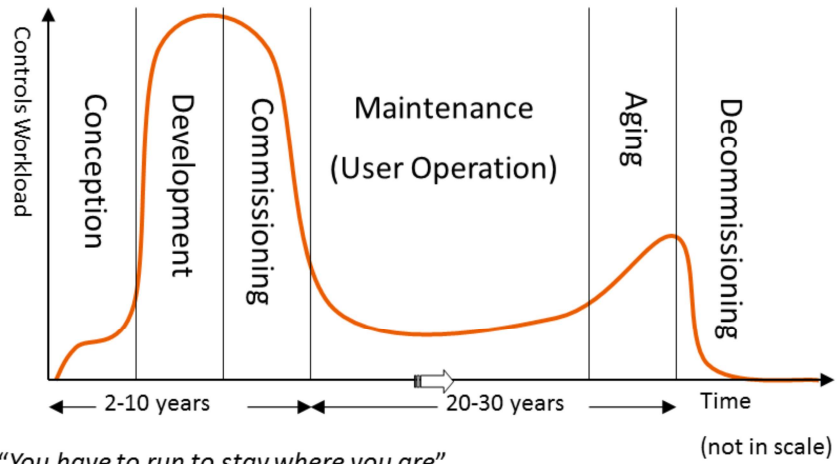


?

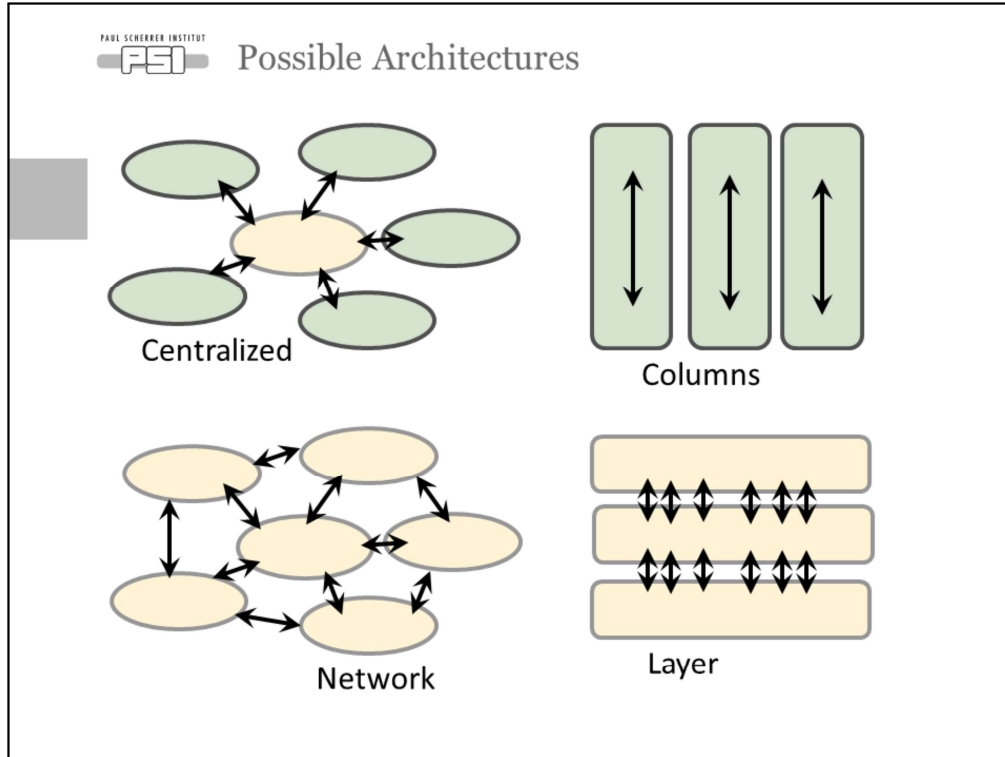
- The requirements from different user groups might be different.
- The most common requirements are:
 - Reliability – if I have beam time scheduled, I want it to happen
 - Good Performance – speed, responsiveness, modern possibilities
 - Flexibility – „hey, I have a good idea, lets try ...“
 - Easy 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):**

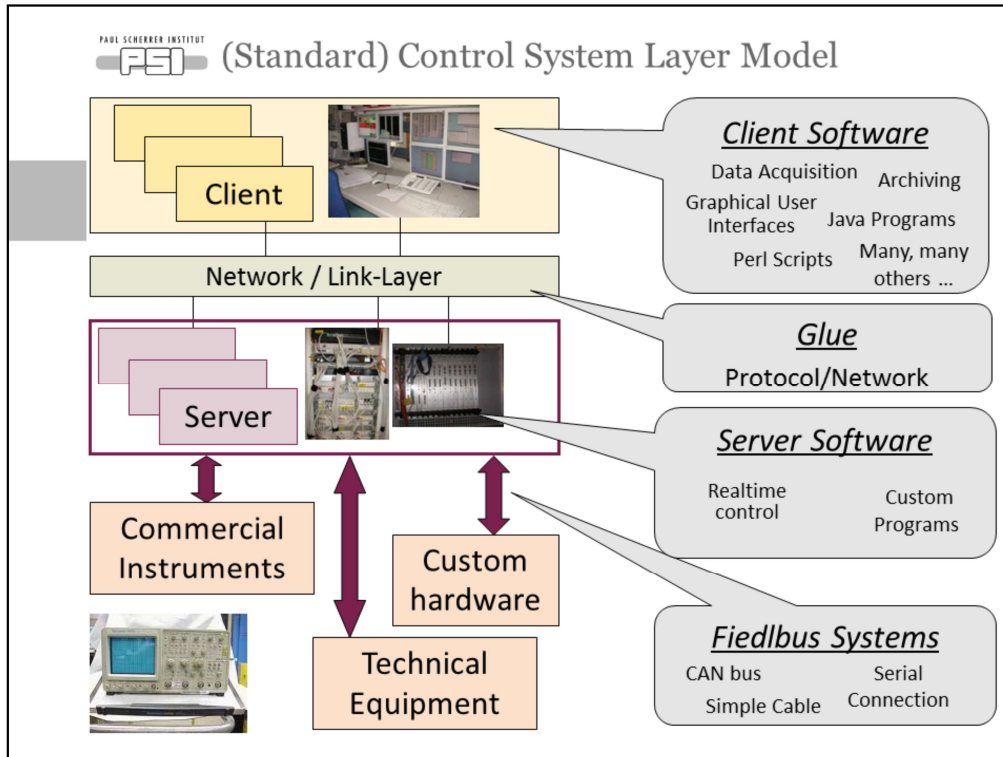
- Can be difficult to maintain if an addition has to be added to all other systems.

- **Layer:**

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

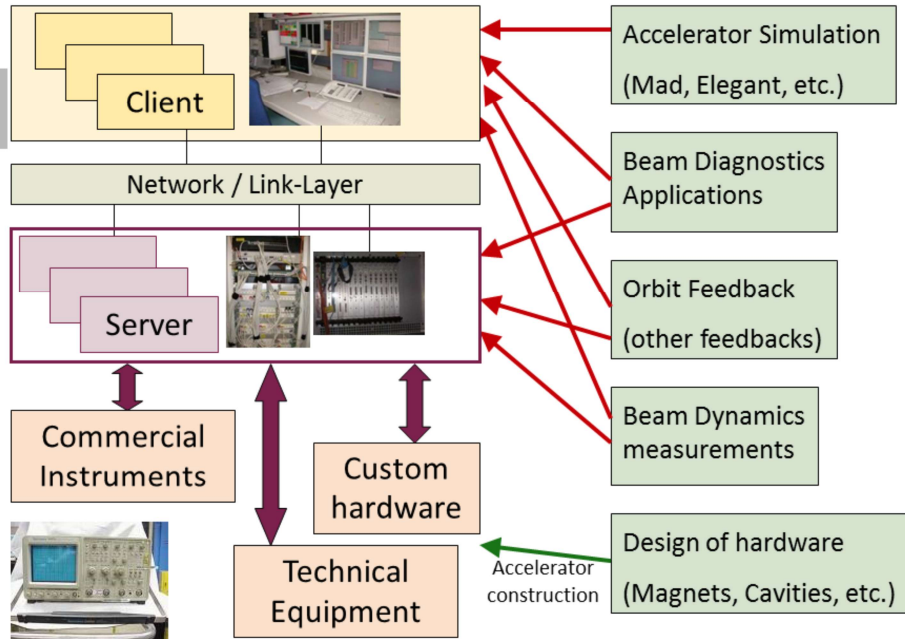
- **Practical solution:**

- Mix between some or all of those architectures.

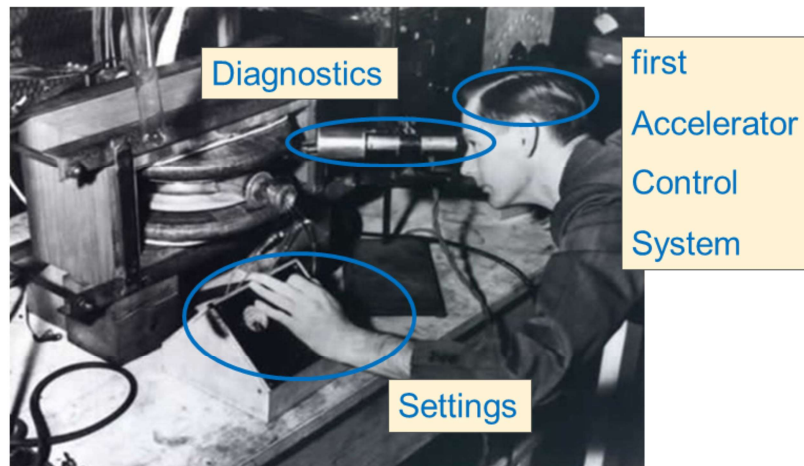


- Three-tier standard model for distributed control systems.
- The hardware layer was considered dumb when the terminology was developed.
- Meanwhile „hardware“ can be as complicated as a robot (with its own controller) or an oszilloscope (with an operating system installed on a specialized computer).
- More and more logic is moved to the hardware layer.
- Therefore, I would like to call it now a 4-tier model.

Where is Physics in there?



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



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

- <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



AGS control room, circa 1966



© 1974 CERN

- 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 Preveessin site. In 2005 all controls were transferred to the Preveessin centre.
- *Date:* Feb 1974
Original ref.: CERN-PHOTO-7402124X

International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)

First held in 1987 in Villars-sur-Ollon (Switzerland), hosted by CERN.

The term "Control Systems" in ICALEPCS is broadly interpreted to include:

- all components or functions, such as processors, interfaces, field-busses, networks, human interfaces, system and application software, algorithms, architectures, databases, etc.
- all aspects of these components, including engineering, execution methodologies, project management, costs, etc.







<https://www.icalepcs.org/> - Next ICALEPCS will be held 5-11, October, 2019 in New York, USA

- Excerpt from the dinner speech of Roland Müller on ICALEPCS 2013 (<https://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. [...]

Solutions: Different Control System Examples

System Name:

EPICS
TANGO

Collaborations:
Used at more than one Lab

Pro:
Bugs are already found

Contra:
Complicated to adapt
to your problems

DOOS
ACS

Single Site Systems:
Developed and used in one Lab

Pro:
Your problems solved
perfectly

Contra:
You are on your own
(no one can help)

SCADA
(WINCCOA)

Commercial System

Pro:
Outsource your problems

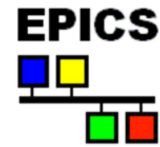
Contra:
Expensive

- 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.

What is EPICS?

- **EPICS (Experimental Physics and Industrial Control System)**

- is a set of software tools and applications
- supports distributed control systems for large research facilities like accelerators
- uses Client/Server and Publish/Subscribe methods
- uses the Channel Access (CA) network protocol



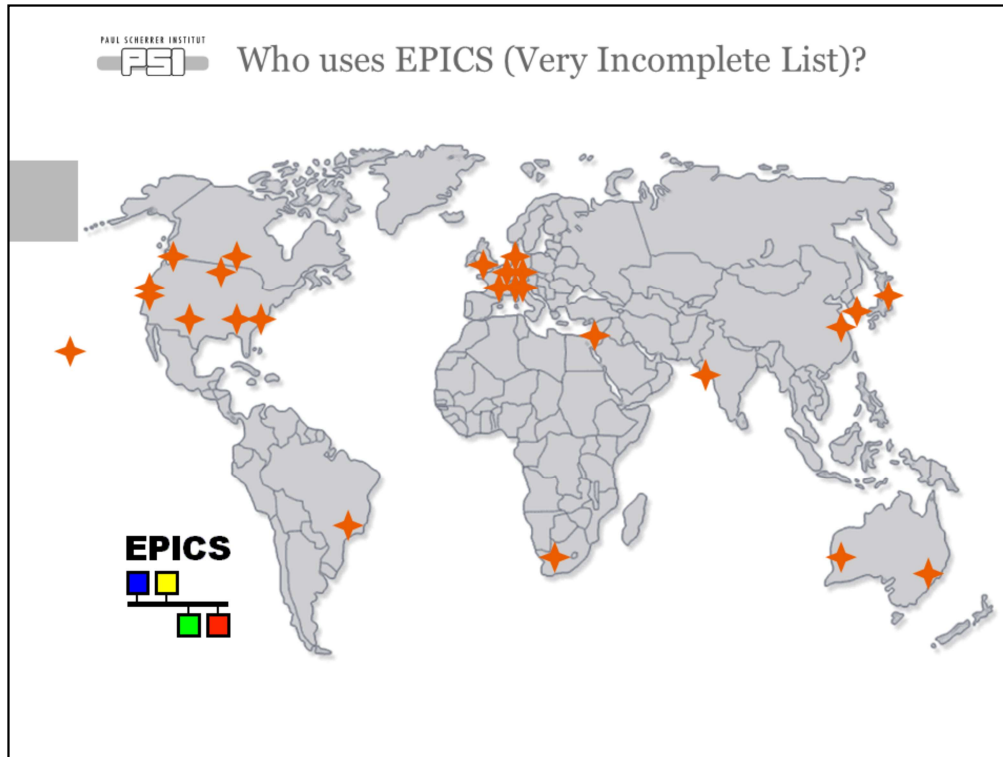
- In 1989 started a collaboration between Los Alamos National Laboratory (GTA) and Argonne National Laboratory (APS) (Jeff Hill, Bob Dalesio & Marty Kraimer)

GTA: Ground Test Accelerator
APS: Advanced Photon Source

- More than 150 licenses agreements were signed, before EPICS became Open Source in 2004

<http://www.aps.anl.gov/epics/>

- 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.

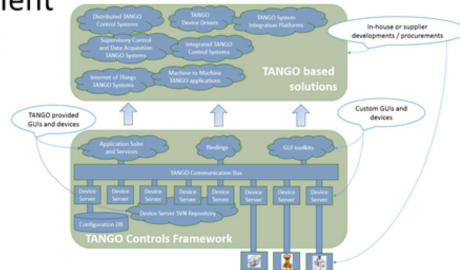


- Some EPICS users (very incomplete list):
- The Advanced Photon Source at Argonne National Laboratory
- Australian Synchrotron
- Berlin Electron Synchrotron (BESSY II)
- Brazilian Synchrotron Light Source (LNLS)
- Deutsches Elektronen Synchrotron (DESY)
- Diamond Light Source
- Fermilab (FNAL)
- Jefferson Laboratory (JLAB)
- Keck Observatory
- KEK B-Factor
- 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 Accelerator Center (SLAC)

What is Tango?



- **TANGO (TAco Next Generation Objects)**
 - is a strictly object oriented toolbox for Control System development
 - is a set of software tools and applications
 - supports distributed control systems for accelerators

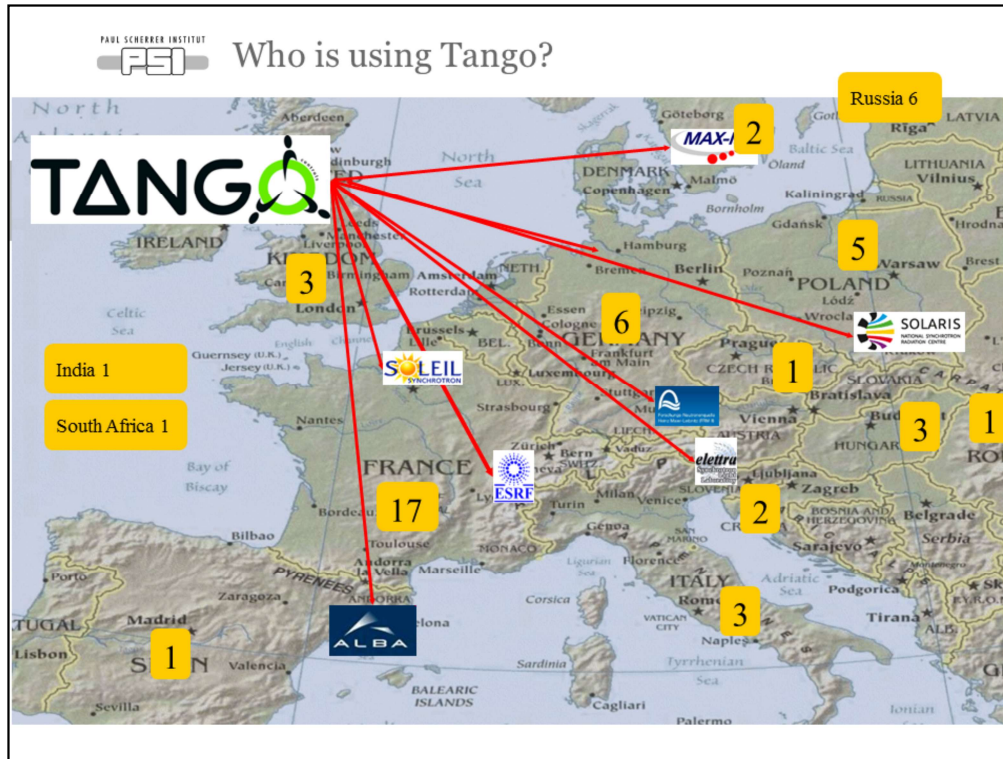


- Started in 2001 with three collaborators, now there are 49



<http://www.tango-controls.org/>

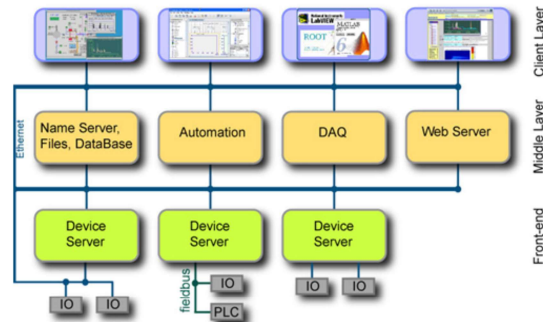
- <http://tango-controls.readthedocs.io/en/latest/overview/overview.html#what-is-tango-controls>
- CORBA and ZMQ to communicate between device server and clients
- C++, Python and Java as reference programming languages
- Linux and Windows as operating systems
- Modern object oriented design patterns
- Naturally implements a microservices architecture
- Unit tested, continuous integration enabled
- Hosted on Github (<https://github.com/tango-controls>)
- Extensive documentation + tools, large community



- 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."
- Find out about the community on <http://www.tango-controls.org/partners/institutions/>

DOOCS (Distributed Object Oriented Control System)

- strictly object oriented system design (C++ and Java)
- Class libraries as building blocks



- Build for FLASH, now used as well for European XFEL

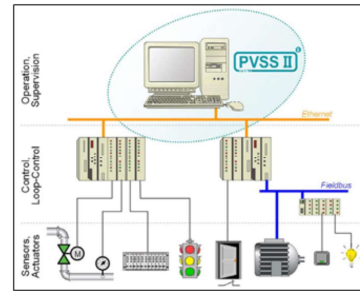
<http://tesla.desy.de/doocs/index.html>

- 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: μ TCA, 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.



PVSS II (Prozessvisualisierungs- und Steuerungssystem 2)

–is an industrial SCADA product from the Austrian company ETM (bought by Siemens AG in 2007)



New name: **WinCC-OA**

SCADA = Supervisory Control And Data Acquisition

(commercial software systems used extensively in industry for the supervision and control of industrial processes)

<http://www.etm.at/>

<https://readthedocs.web.cern.ch/display/ICKB/WinCC-OA+Service>

- From *0407PVSS_II_GettingStarted_Basics_en.pdf* found on [https://readthedocs.web.cern.ch/download/attachments/21178115/ ... 0407PVSS_II_GettingStarted_Basics_en.pdf?version=1&modificationDate=1299054760000&api=v2](https://readthedocs.web.cern.ch/download/attachments/21178115/...0407PVSS_II_GettingStarted_Basics_en.pdf?version=1&modificationDate=1299054760000&api=v2)
- **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>)

- 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



By Evan Swigart

The choice for one system is not exclusive

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

Use open source
firmware/software.

- You can change things and you have control of further developments

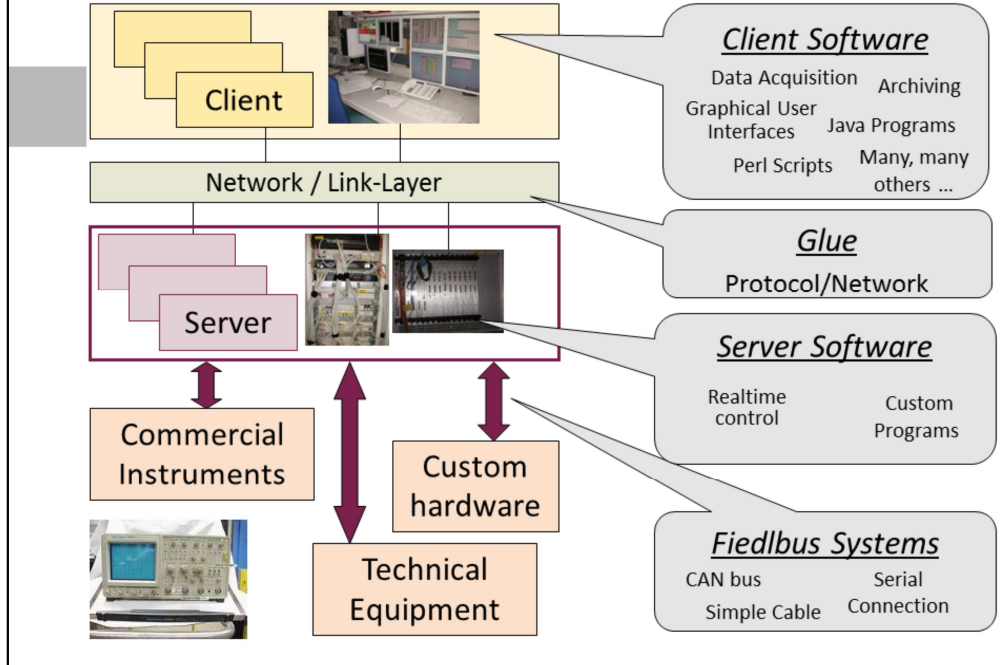
Use commercial solutions based on open standards developed and sold by a large number of companies

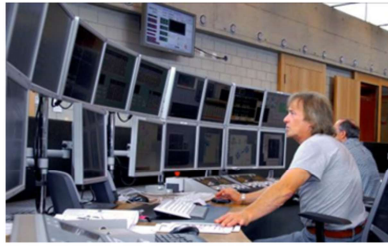
- Don't become dependent on single companies with proprietary solutions

Use standards with a long life-time (20 years+)

- Keep long lifecycles of accelerators in mind

Reminder: Control System Layer Model





Usually clients run in a
control room
and are used by
operators

Where is the logic? Where are the computations?



- **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.

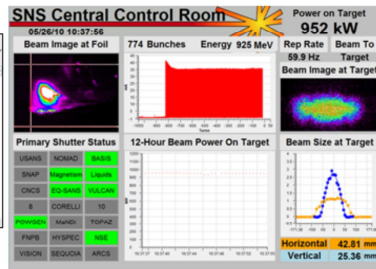
- **Operators**

- There might be dedicated personnel 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.

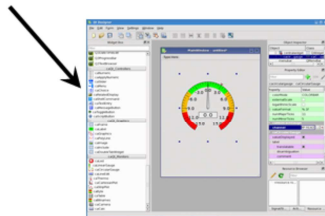
- **Thin Clients:** Only displays results that are produced in some server elsewhere. Therefore, the display is decoupled from the calculations or logic. Both can be changed independently (for example for updates or new features). Mostly used for expert screens that display the complete interface of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

GUIs: Usually thin clients



Example for an Editor



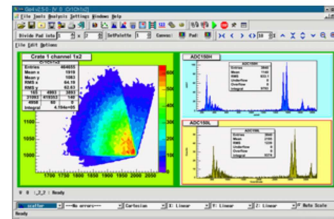
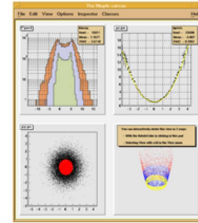
PSI is using a GUI builder called caQtDM (EPICS based):
<http://epics.web.psi.ch/software/caqtmd/>



- Home page of JDDD (Java DOOS Data Display): <https://jddd.desy.de/>
- BOY (Best OPI, Yet) Operator Interface for Control System Studio (EPICS based): <https://github.com/ControlSystemStudio/cs-studio/wiki/BOY>

Examples for accelerator science applications:

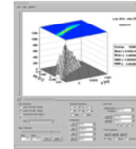
- Tune measurement and correction
- Orbit correction
- Beam based magnet alignment
- Parameter scans
(to find optimal working points)
- Filling pattern measurements
and correction
- Correlation Plots



... general data analysis of accelerator data

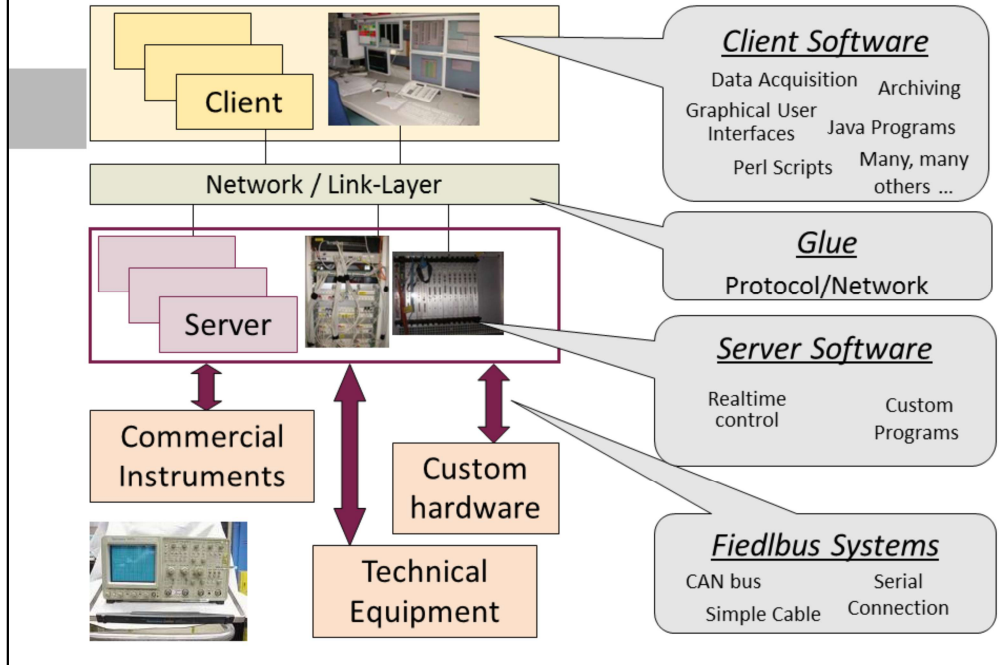
Usually fat clients,

usually written by scientists (not by controls experts)



- 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 (<https://root.cern.ch/>)
- Other labs use MATLAB from Mathworks for science applications.

Reminder: Control System Layer Model



user interface



Server Hardware

PC



Ethernet

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)

field bus
(ethernet,
serial, USB,
firewire, ...)



user interface



Ethernet

Server Hardware

VME
(Operating System:
e.g. vxWorks)



Dumb
Hardware



VME cards life cycle is long,
VMEbus is an open standard,
Supported by Industry

VME is expensive,
special operating system
(VxWorks)

Cable or
field bus
(analog I/O,
digital I/O,...)

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>

user interface



Ethernet

Server Hardware

Linux PC



Example for tiny computers with single interface

Cheap and tiny solution,
Supports distributed devices
All commercial chips have slightly different architecture (maintenance),
life cycle yet unknown

Serial interface
(RS232, ...)



user interface



Ethernet

Low cost, have standard network interfaces and support distributed devices

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

Embedded Hardware

=

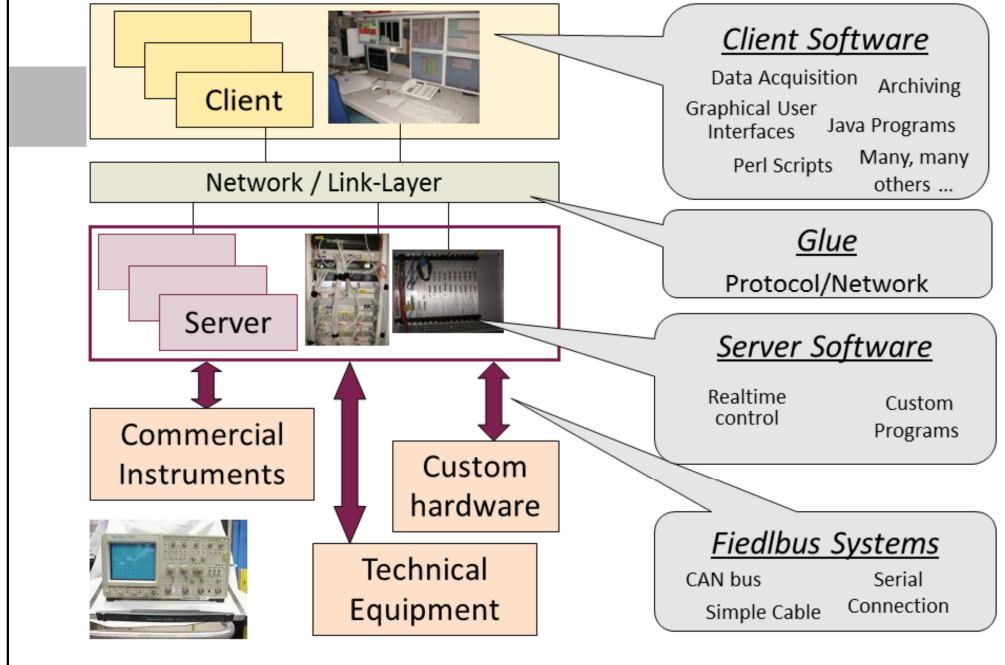
Server Hardware

+

Instrument



Reminder: Control System Layer Model



- **PLC (Programmable Logic Controller)**
 - 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!



• Examples for companies:

- Siemens S7

(<https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc.html>)

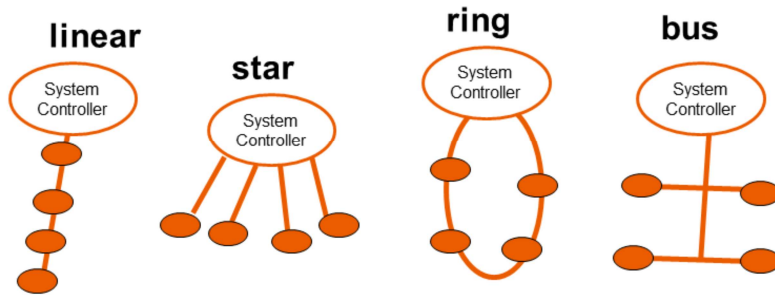
- Allen-Bradley

<http://ab.rockwellautomation.com/Programmable-Controllers/>

- Beckhoff

<https://www.beckhoff.ch/>

- Field busses connect hardware to servers
- A lot different busses available with different purposes and different specifications as
 - number of allowed devices
 - speed
 - allowed cable length
 - topology (ring, star, linear, ...)



- 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**.

Some example field bus systems:

- **CANbus** (Controller area network)
https://en.wikipedia.org/wiki/CAN_bus
- **PROFIBUS** (Process Field Bus)
<https://en.wikipedia.org/wiki/Profibus>
- **IEEE 1394** (Firewire)
https://en.wikipedia.org/wiki/IEEE_1394
- **EtherCAT** (Ethernet based real time bus)
<http://www.ethercat.org/en/ethercat.html>



Difference to Ethernet and USB?

Field busses are real time capable (IEC 61158 specification)

- **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.

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

Accelerator Control Systems have fussy borders.

Some example for these borders are:

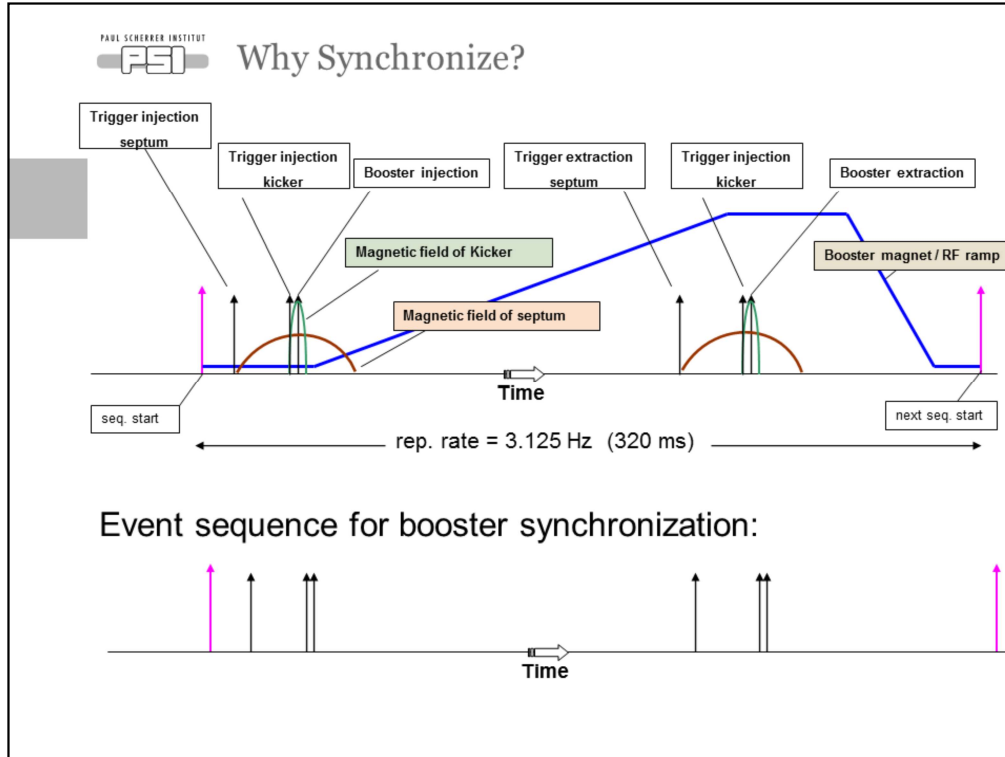
1. Timing and Synchronisation
2. Feedback Systems
3. Interlock-, Alarm-, and Machine Protection Systems
4. Experiment Data Acquisition
5. Relational Databases
6. Relationship of IT (Information Technology) and Controls

- 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).

- Master oscillator + delay cables
(1 trigger and measured cable lengths)
- Master oscillator + digital delay generators
(<http://www.thinksrs.com/products/DG535.htm>)
- (Master oscillator +) event generators/receiver cards in computers (PC , VME, μ TCA)
(<http://www.mrf.fi/>)
- Timing and synchronization is needed to run an accelerator
- Various solutions available and used



Timing and synchronization can be part of the Control System.
Clarify who is responsible for timing and synchronization to avoid problems!

- Accuracy of the timing signal can vary from some pico seconds down to femto seconds – but femto seconds are really hard to get:
<http://accelconf.web.cern.ch/accelconf/ipac2012/papers/weppb006.pdf>
- Technology depends on needed accuracy and available budget.

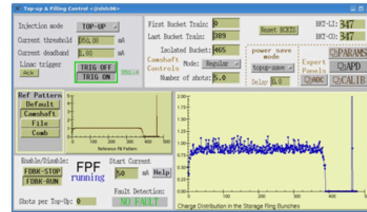
For example

2. Feedback Systems



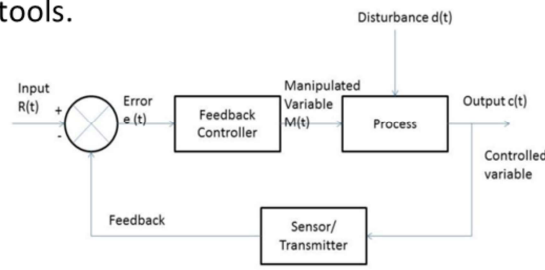
For example:

- Orbit Feedback (Position)
- Energy Feedback
- Filling pattern Feedback



If it needs to be fast, it needs separate cables!

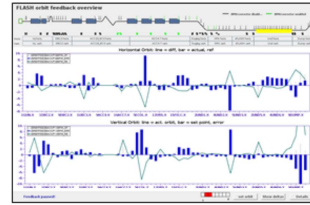
Slow feedbacks can be realised with standard control system tools.



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

- https://en.wikipedia.org/wiki/Singular-value_decomposition

For example

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



What are Interlocks?

Everything is fine (No Alarm)

Example: Vacuum pressure 1e-10 mbar

Something is strange (Warning)

Example: Vacuum pressure 1e-7 mbar

Something is wrong (Error)

Example: Vacuum pressure 1e-6 mbar

Stop it or suffer from severe
consequences (Interlock)

Example: Vacuum pressure 1e-5 mbar

Automatic beam dump executed

Go on working

Alarm states

Alert people to take
some actions

Interlock

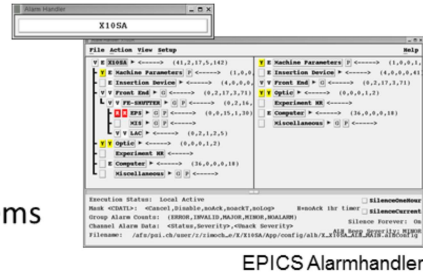
Automatic reaction needed

- 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 occur 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.

Murphy's law:

Anything that can go wrong will go wrong.

- 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



- 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



• **EIGER X 9M Detector**

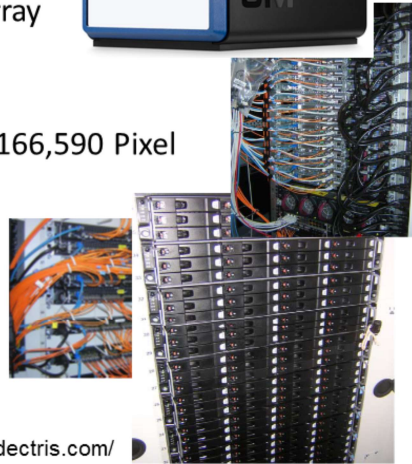
(Synchrotron-Beamline at SLS):

- two-dimensional hybrid pixel array detectors, which operate in single-photon counting mode
- composed of $3110 \times 3269 = 10,166,590$ Pixel
- maximum frame rate 238 Hz
ca. 10 MB \rightarrow 2.3 GB/s
 \rightarrow more than 8 TB per hour

(3 years ago:

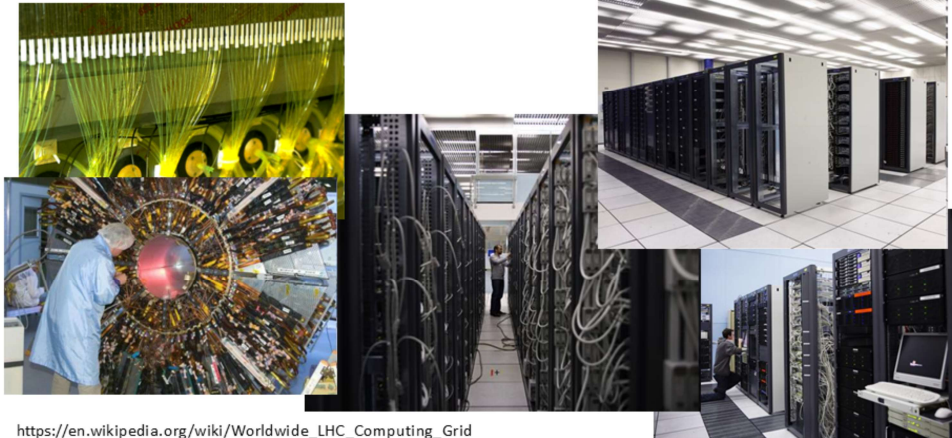
2 TB in 8 hours with Pilatus)

<https://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



https://en.wikipedia.org/wiki/Worldwide_LHC_Computing_Grid

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!

For example

5. Relational Databases



What is a Relational Database?

- Used for “stable” Data (Lattice, Magnet Data etc.)
- Good for searching
- Might be slow for runtime data
- Examples:
 - Oracle
 - MySQL
 - MSAccess
- Language to access data is SQL (Structured Query Language) for all examples
- Relational Databases are useful for Control Systems
- Some accelerator control systems have integrated relational databases
- Setup and Maintenance require knowledge and manpower

Name	Class	Z0 (M)	LM	Description
FIND1-AGR	GRIDER	-1.85	4.7	grider
FINSS-MSOL10	SOLENOID	-0.1	.03	solenoid
FWLHA-XREF0		0.	.70	building
FINSS-RGUN	SW	0.	0.25	CEER gun
FINSS-VPFG14010	PUMP	0.07	0.	getter pump 75 ls
FINSS-VVMA14010	CROSS_ANGLE	0.07	0.	valve cross angle
FINSS-VPFG14020	PUMP	0.1	0.	getter pump 75 ls
FINSS-VVACC14010	PENNING	0.1	0.	gauge Penning
FINSS-VVFC14010	PIRANI	0.1	0.	gauge Pirani
FINSS-VVMA14020	CROSS_ANGLE	0.1	0.	valve cross angle
FIND1-MCRX10	CORRECTOR	0.166	.005	corrector magnet
FIND1-MCRY10	CORRECTOR	0.166	.005	corrector magnet
FIND1-MSOL10	SOLENOID	0.17	0.26	solenoid
FIND1-MCQR10	QUADRUPOLE	0.17	.07	corrector quadrupole regular
FIND1-MCQS10	QUADRUPOLE	0.17	.07	corrector quadrupole skew
FINSS-VCHB14010	BELLOW	0.25	.08	bellow DN 4000
FINSS-VCHT14010	BELLOW	0.25	.08	bellow DN 4000
FIND1-MCQZ0				
FIND1-MCRY20				
FINSS-DIAPR10				
FIND1-VPFG14110				
FIND1-FWVCR10				

- In most cases the setup of the database framework (i.g. installation of the database) is considered 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
<ul style="list-style-type: none"> • Office PC installation • Operating Systems for Office applications • Infrastructure (network cables) • Central Services (Computing Cluster, Server Room ...) 	<ul style="list-style-type: none"> • Accelerator computer installation • Integration of accelerator hardware • Control Room applications • Distributed processes
Databases, Timeserver, Network, Security	
<p>Controls is dependent on IT.</p> <p>Responsibilities have to be discussed to avoid problems!</p>	

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

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

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**.

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

Bad news: There is no book on Accelerator Control Systems

Good news: You can find some things in the Internet

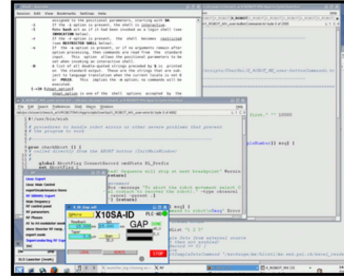
- ICFA Newsletter Number 47 (December 2008) on Control System:
http://icfa-usa.jlab.org/archive/newsletter/icfa_bd_nl_47.pdf
- EPICS: <http://www.aps.anl.gov/epics/>
- Tango: <http://www.tango-controls.org/>
- CERN Controls Group: <https://be-dep-co.web.cern.ch/>
- PSI Controls Group: <http://epics.web.psi.ch/>
- ...search the institute web pages ...
- International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS): <https://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 (python, 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



The
End

