Surveying Software Technology for Accelerator Control Systems

MAX-lab / KTH Stockholm

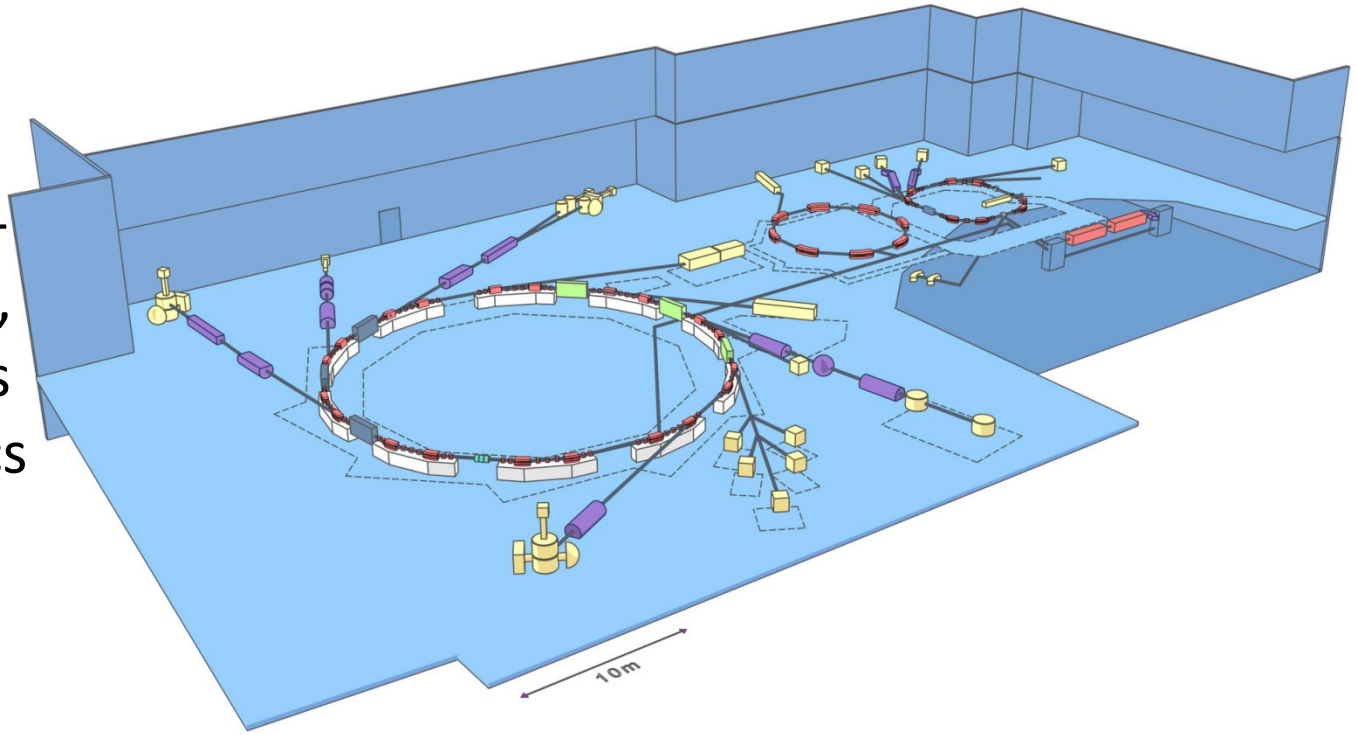
Content

- Introduction - MAX-lab and MAX IV
- Domain-specific frameworks
- Emerging Issues
- Conclusion

MAX-lab

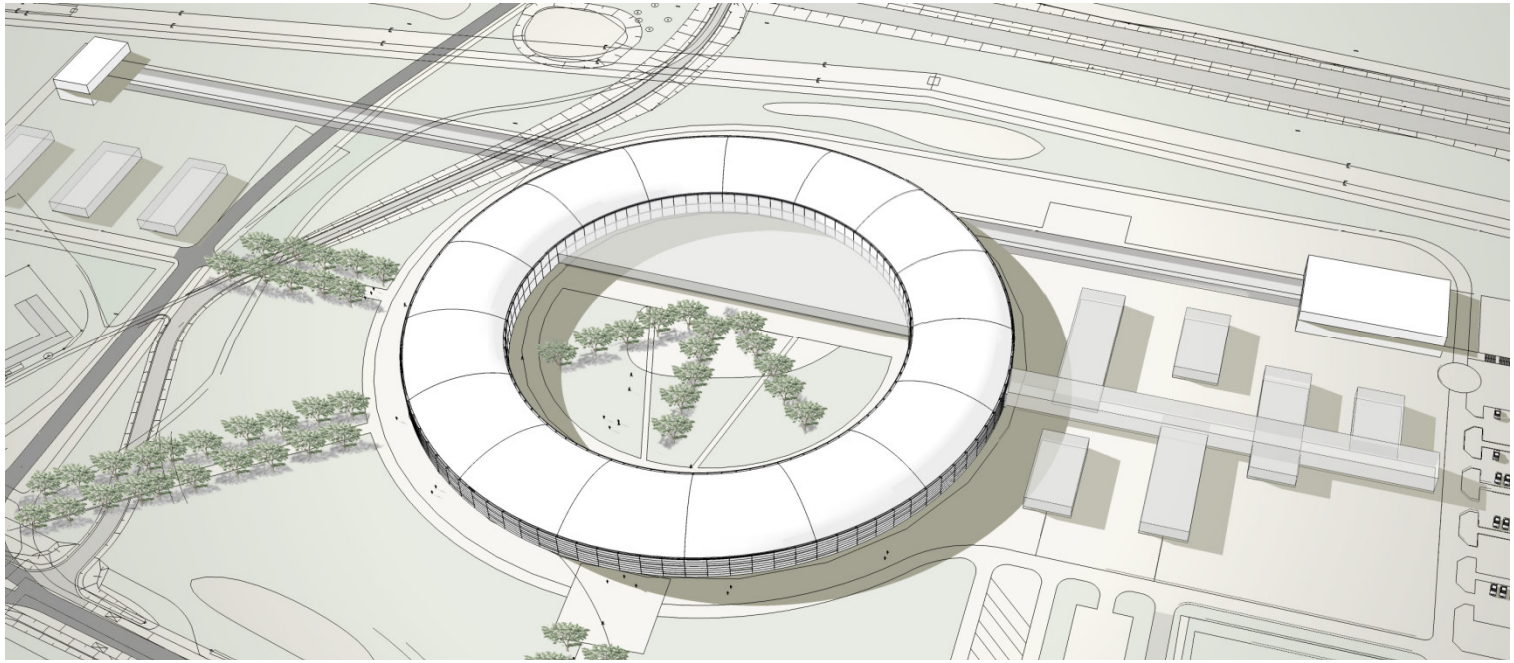
Swedish National Laboratory for Synchrotron Radiation based Science, Accelerator Physics and Nuclear Physics

16-20 beamlines in use



MAX II ring	1.5 GeV	soft x-rays	10 straight sec.
MAX III ring	700 MeV	VUV	8 straight sec.

MAX IV



MAX IV ring	3.0 GeV	hard x-rays	20 straigh sec.
MAX II ring	1.5 GeV (moved)	soft x-rays	10 straight sec.
MAX III ring	700 MeV (moved)	VUV	8 straight sec.
Linac	3 GeV	injection (FEL upgrade)	short photon pulse beamline

Survey motivation

- staff size is small, will remain relatively small
- currently few for computing systems
  - ➢... no integrative control layer

Survey scope:

- MAX IV IT infrastructure
  - – domain-specific frameworks
- development support

Domain specific software frameworks

- domain-specific character (synchrotrons, FELs)
- open-source

- EPICS, TANGO: used at several light sources
- TINE & DOOCS: used in a multi-accelerator and FEL environment

Basic requirements – first conclusion

- integration support
  - desirable hardware platforms, OS, programming languages
- communication system
  - name resolution, meta-data
- development support for applications and control server
- archive system, alarm system
- behavior scripting support
- administrative tools

➢ ... ok, all provide this in some way.

➢ ... worst decision would be to remain independent.

Aspects to consider, notable things

- Application development
  - thin clients, (for end-user development): jddd
  - thin/rich client development: ACOP+COMA
  - workbench approach: Control System Studio (CSS)
- beyond historical archiving:
  - local command archiving (TANGO)
  - local data archives (TINE)
  - event (post-mortem) archive infrastructure (TINE)

Aspects to consider, notable things

- Scalability (number of systems)
  - for us, less a network performance question than about...
  - hierarchy concepts
    - modularization (object/device orient., focus on system-of-interest)
    - namespaces (for name resolution)
    - alternative names (redirection/gateways/aliasing)
    - system administration tools (browsing the system)
- Control Server / hardware integration
  - code generators: VDCT, POGO, Server Wizard, ...
  - hardware interfacing: asyn, CDI, abstract classes
  - using existing drivers, EPICS driver base being the largest

Aspects to consider, notable things

- Performance, high data loads
  - built-in multicasting from all nodes, including acknowledgement (TINE), useful e.g. for ...
- Video system developed for PITZ/XFEL
  - features for beam analysis, DAQ and camera integration
- Security
  - malware, intrusion is a network administration issue
  - access restriction for critical devices is desirable, available (EPICS, TINE) or on the way (TANGO)

Aspects to consider, notable things

- Safety
  - hard real time tasks (e.g. protection systems) are seperate, can be integrated sufficiently for control, archiving, etc.
- Usability of the produced systems
  - mainly application dependent, but toolkit can be limiting
- Dependability
  - availability, robustness, recoverability of existing solutions appear as sufficient for our foreseeable needs

Aspects to consider, notable things

- Beamline control issues
  - beamline frameworks: synApps, GDA, Soleil approach, ...
- FEL specific issues
  - global feedback loops
  - global synchronization
  - data acquisition
  - instrumentation and experiment equipment
  - ➢ addressed by DOOCS, user operation

... there is certainly more out there.

Advanced requirements, second conclusion

- Different frameworks may give us different bonuses
- choosing a framework for technical reasons demands a *sufficiently detailed* requirements analysis with a *finalized validation.*

➢ *out of our reach*

  - (intentionally) vague requirements (FEL upgrade)
  - beamline requirements change

Development Process Support

... in this course, a shift in our scope occurs

- *from the technology properties*
- *to the structures and processes of the development and engineering ...*

- Information management
  - Product Document /Lifecycle Management tool
  - office documents, CAD, other files
  - provides *access*, *consistency* and *change management*
  - for control systems group: central repository for specifications, manuals, published appl.'s, measurement data, etc.

Development Process Support

Requirements and Specifications

- needs suitable information structure
  - different stakeholders' perspectives
  - allow top-down, bottom-up, middle out developments
  - prioritization
- professional requirements engineering software (Borland CaliberRM) software is used
  - consistency, traceability, change history, Word-document export
  - intended for few software and system architects

Development Process Support

Control system simulator:

- real applications and services
- simulated control servers, using a physics models
  - general staff training
  - verification: more realistic test environment (complex communication, services, error injection)
  - validation of an application design, toolkit, complex functionality; finding new requirements
  - building and deployment procedure
- future use: reduce impact on on-going operations
- possible in all frameworks

Beamline oriented standardization

A beamline control package, as generic as feasable.

Area	Goal
Building blocks for IO types (hardware with software interface)	re-use, reduce # of hardware and software diversity
functional libs (calculations, optics, ...)	re-use, reduce # of libs
integrated systems (x-ray mirror, pumps)	re-use, reduce systems diversity
Scan system, integrative BL components	top-up mode, etc., same for all BLs
BL data acquisition and management system	record all, export on demand
GUI builders and scripting tools suitable for non-programmers	shift workload, improve autonomy of experts
services at beamlines: history & snapshot & building and deployment	same on all BLs

Conclusion

Emerging, primary issues:

- managing the new degree of complexity
- standardization
- improving non-expert's user autonomy

➢ guideline development

➢ domain-specific framework not yet decided

Thanks to the community!