EPICS-based Control System for a Radiation Therapy Machine

Jonathan Jacky

University of Washington Medical Center
Department of Radiation Oncology
jon@uw.edu, http://staff.washington.edu/jon/

Acknowledgments

UWMC Cyclotron Engineering Group

- Stefani Banerian
- Eric Dorman
- Robert Emery
- Steven Steininger
- Tobin Weber

UWMC Clinical Neutron Therapy System (1)

Hospital-based cyclotron and neutron radiation therapy

UWMC Clinical Neutron Therapy System (2)

Thirty year history:

- NCI funded
- Began treating 1984
- Neutron clinical trials in the 1980s
- Currently the only neutron treatment center in the USA
- *Hospital-based* engineering group (no vendor)

Exotic particles (for medicine):

- Cyclotron
- Neutron beam

But otherwise like a typical hospital radiation therapy machine:

- Isocentric gantry
- Multileaf collimator
- Computer control

CNTS Control System: History

Thirty years, three generations of control systems, including two developed by us:

1984 Scanditronix PDP11/RSX/Fortran
with UWMC custom RTP via DECNET
Ahead of its time

1999 UWMC VME(68k)/VxWorks/C
with UWMC custom Prism RTP via NFS
new functional spec, workflow, UI, hw, sw
State of the art

2013 UWMC X86/Linux/EPICS
with commercial Pinnacle RTP via DICOM
new hw, sw *but* very similar functional spec etc.
Time tested

We are using EPICS for the latest version.

EPICS

EPICS: Experimental Physics and Industrial Control System

Libraries, utility programs, and tools for developing control systems

20+ years of success at large accelerators etc.

Features:

- Input/Output Controllers (IOCs) and clients on a network
- Communicate using Channel Access (CA) protocol on UDP and TCP
- Program each IOC by defining a "database":
  Process Variables (PVs) and data flow program
- Many convenient development tools

EPICS Pros and Cons

Advantages (for us):

- Successful at large accelerators etc. for 20+ years
- Successful in our hands recently for non-therapy functions
- Functionality is a good fit to our application
- More rapid and convenient development than just C
- Standardize on a single collection of components and tools

Disadvantages – ?

- Larger, more complex toolchain and runtime
  (than our previous platform, just C on VxWorks)

Less reliable? Behavior harder to understand and predict?

EPICS Caveats

EPICS developers counsel *against* using it for safety-critical applications:

2008 : "(EPICS) code is not rigorously audited to the standards . . . that would be needed (for medical applications). Here at (laboratory) the general rule is that EPICS software plays no part in implementing personnel safety. . . ."

`epics/tech-talk/2008/msg00803.php`

2012 : "EPICS should never be relied on for safety-critical operations, although it is used at many sites to monitor such equipment . . ."

`epics/tech-talk/2012/msg01836.php`

CNTS Control System: Components and Functions

Includes nonprogrammable, programmable, and human elements

Use the simplest, most mature technology that can do each function

Relay	:	Hardwired Safety Interlock System (HSIS)
PLC	:	Interlocks, combinational logic, simple sequencing
uC ROM	:	Low-level device control via PIO, DAC, ADC
HLL Software	:	Prescription-specific, data-intensive computation
People	:	Select prescription, position patient, avoid collisions
More people	:	Ongoing QA of treatment process including prescriptions, control system

Each layer can independently achieve and maintain a safe state of the components under its direct control.

Dose Terminator Example (1): Components and Functions

Dose termination: turn off the beam when the accumulated dose reaches the prescribed value.

Relay : Open a switch in the hard-wired safety trace to turn off the beam

TTL : Compare accumulating dose register A to prescribed dose register P, when A equals or exceeds P, de-energize relay

DMC (uC ROM) : Load register P with scaled integer prescribed dose

HLL program : Retrieve prescribed dose from prescription database, scale by calibration factors, send to DMC

People : Select appropriate prescription from database, command beam to turn on (may be vetoed by any lower layer)

More people : Measure and enter calibration factors

Dose Terminator Example (2): Redundancy etc.

We use *redundancy*, *crash protection*, and *consensus* so the system remains safe if components halt, delay, or perform erratically.

Redundancy : Two independent hardware dose terminators, two relays

Also, backup timer turns off beam is estimated treatment time is exceeded

Crash protection : Hardware dose terminators ensure correct prescribed dose is delivered even if DMC or HLL programs crash or stall during a treatment.

Electromechanical dose counters record delivered dose even if power lost during a treatment

Consensus : Initiating or continuing potentially hazardous activites (like beam on) requires ongoing active cooperation of several programmable and nonprogrammable components.

Dose Terminator Example (3): Signal Path

We do use HLL programs on safety-critical signal paths, for example the dose terminators:

Radiation Therapy Planning (RTP) System $\rightarrow$

$\rightarrow$ DICOM (Common Lisp) $\rightarrow$ Prescriptions (PostgreSQL) $\rightarrow$

$\rightarrow$ psycopg2 + pyepics EPICS CA (Python) $\rightarrow$

$\rightarrow$ EPICS IOC ai, bi, ... $\rightarrow$ IOC calc, ... $\rightarrow$ IOC ao, bo, ... $\rightarrow$

$\rightarrow$ IOC asyn $\rightarrow$ IOC streamDevice, dmc.proto $\rightarrow$

$\rightarrow$ DMC (Dose Monitor Controller, uC with ROM)

$\rightarrow$ dose terminator hardware

HLL programs include EPICS but also others. All – EPICS or not — require intensive analysis, testing, and monitoring.

Assurance activities

Development

- Requirements, functional specification
  - Identify all inputs, outputs, state variables, state transitions
  - Explicit safety requirements (invariants)
  - Analysis checks each state transition preserves safety invariants
- Design, code
  - Use minimal set of EPICS constructs
- Static analysis (inspection)
  - Tool generates data flow diagrams from EPICS databases
- Dynamic analysis (testing)
  - Python unittest, use EPICS CA to set and read PVs
  - Load test, simulate extended operations in short time
  - System test end-to-end in production configuration

Operations

- Logs and treatment records
- Run-time checking of PVs

EPICS components and coding style

Use simple configuration and minimal set of EPICS constructs:

- Therapy control program runs on one soft IOC
- IOC is the only application running on its computer
- IOC does not require any clients to maintain safe state
- Only database records, StreamDevice .proto files, st.cmd
- No custom device support, no subroutine records
- Database DB links only, no CA links, so no SNL
- Data flow is all "push": SCAN PASSIVE, OUT PP, FLNK

Only these record types: acalcout, ai, ao, asyn, bi, bo, calc, calcout, fanout, longin, longout, mbbo, scalcout, stringin, stringout, seq

Status and Conclusions

Status (Oct 2013)

- Most functionality working
- Not yet complete
- Not yet in clinical use

Enough is working to make some provisional conclusions:

- EPICS database records and .proto files are sufficient
- Sole therapy control IOC uses about 1 percent of processor
- Development straightforward, no unexpected problems
- No apparent obstacles to clinical use

EPICS-based Control System for a Radiation Therapy Machine

Jonathan Jacky

University of Washington Medical Center
Department of Radiation Oncology
jon@uw.edu, http://staff.washington.edu/jon/