Track-Finder Test Plans

Approximate schedule

- ➡ **Jan.'03: SP2002 prototype completed, initial tests begin**
- ➡ **Feb.'03: MPC→SP optical link tests**
- ➡ **Mar.'03: SP trigger logic tests**
- ➡ **Apr.'03: CSC system tests with cosmic rays**
- ➡ **May'03: beam tests with CSC chambers at CERN**

Time is very tight, and still have a lot of firmware and software to write

- ➡ **Software is being written using XDAQ, hopefully in a way that is relevant for future slice tests of muon system**

People involved with software at UF

Song Ming Wang

➡ **Post-doc, 50% time, XDAQ interfaces and communication**

Holger Stoeck

➡ **Post-doc, 50% time, VME test software, code manager**

Bobby Scurlock

➡ **Student, full-time, VME test software, simulation code**

Alex Madorsky

➡ **HW and SW engineer, author of some original SP code and our "technical consultant" for new development**

➡ **100% time during testing period in 2003**

Longer term:

➡ **Barashko and Drozdetski available to help with interface to EMU FAST-DAQ software (starting ~ March 2003)**

CVS Repository and Web Page

http://www.phys.ufl.edu/~tfcvs

Old Test Software

This summer we resurrected the test software used in the Florida tests of 2000 and ported it to Linux

- ➡ **Verified that SP2000 prototype still works, so it can be used for tests with new software while we wait for 2002 prototype**

Original code has the following packages and sub-packages:

- ➡ **ORCA**
  - ❑ **A standalone version of the trigger simulation used in the test to validate the hardware results. This has been fully updated to the latest prototypes.**

- ➡ **LB**
  - ❑ **bedtest: National "SCANEASE" software ported to download Xilinx FPGAs using JTAG via Bit3 VME interface**
  - ❑ **jam: Same thing for Altera**
  - ❑ **LoadLookUps: Downloads SP LUTs**
  - ❑ **svf2evf, datToBin: utility routines to convert file formats**
  - ❑ **LoadChips→CrateLoader: JAVA GUI**

Crate Loader

A summer student from Ohio, Paul Pfeiffer, spent some time generalizing the Java GUI to load more than one card in a crate and more than one crate

➡ **This is to prepare for a real chain test down the road, and to become compatible with XDAQ**

➡ **Keeps track of the FPGA and LUT configuration files, calls low-level downloading programs in LB package**

➡ **Want to update to read/write XML files used by XDAQ**

➡ **Possibly extend it to send SOAP messages to XDAQApplications**
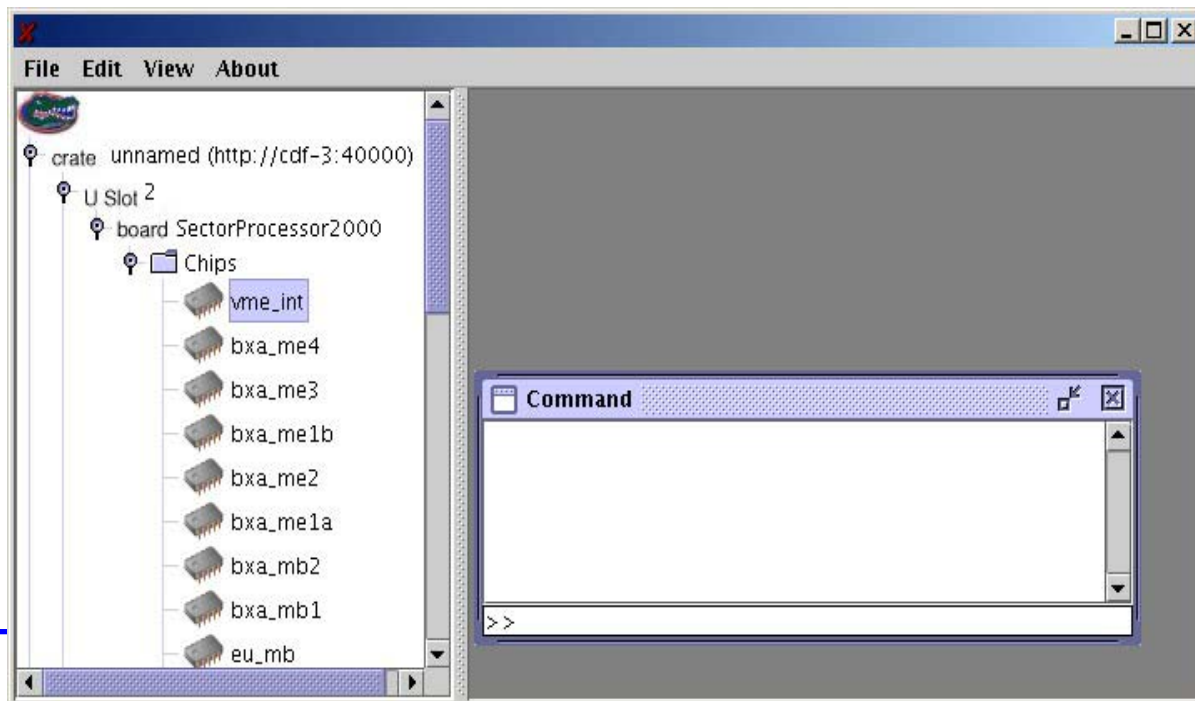
Implementing a XDAQ compliant set of software to run CSC trigger tests

VME control (using HAL library):

➡ **Classes exist for VME configuration of TMB, MPC, CCB, and TTC from Rice University (Greg Pawloski)**

➡ **Developing similar SP classes with additional capability to download LUTs and FPGAs from external files**

  ❐ **Integrating previous code from prototype tests in 2000**

"Front-End Drivers" (XDAQApplications)

➡ **Developing "Driver" classes that instantiate VME classes and execute XDAQWin configuration commands**

Bit3

Track-Finder Crate

Linux
CPU

XDAQApplications:
SPDriver(s)
CCBDriver
(MPCDriver)

DB
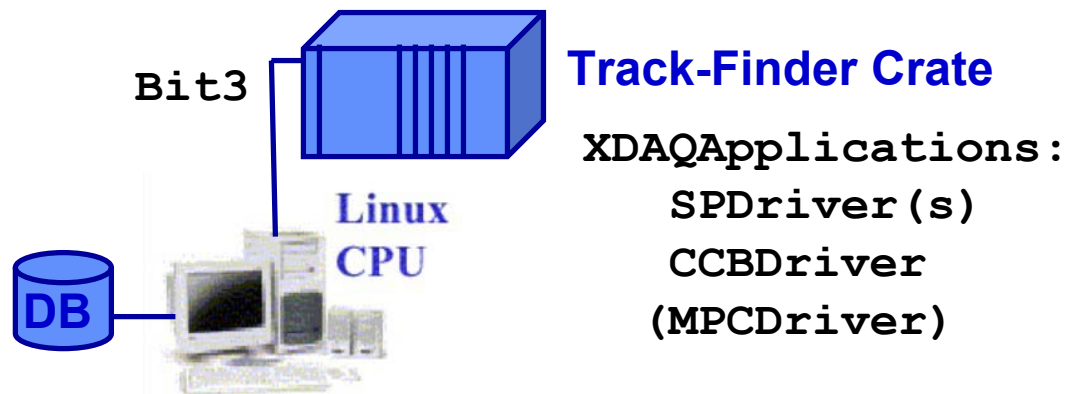
CSC "TrigDAQ" Implementation

Track-Finder Crate   **Peripheral Crate 1**   **Peripheral Crate n**
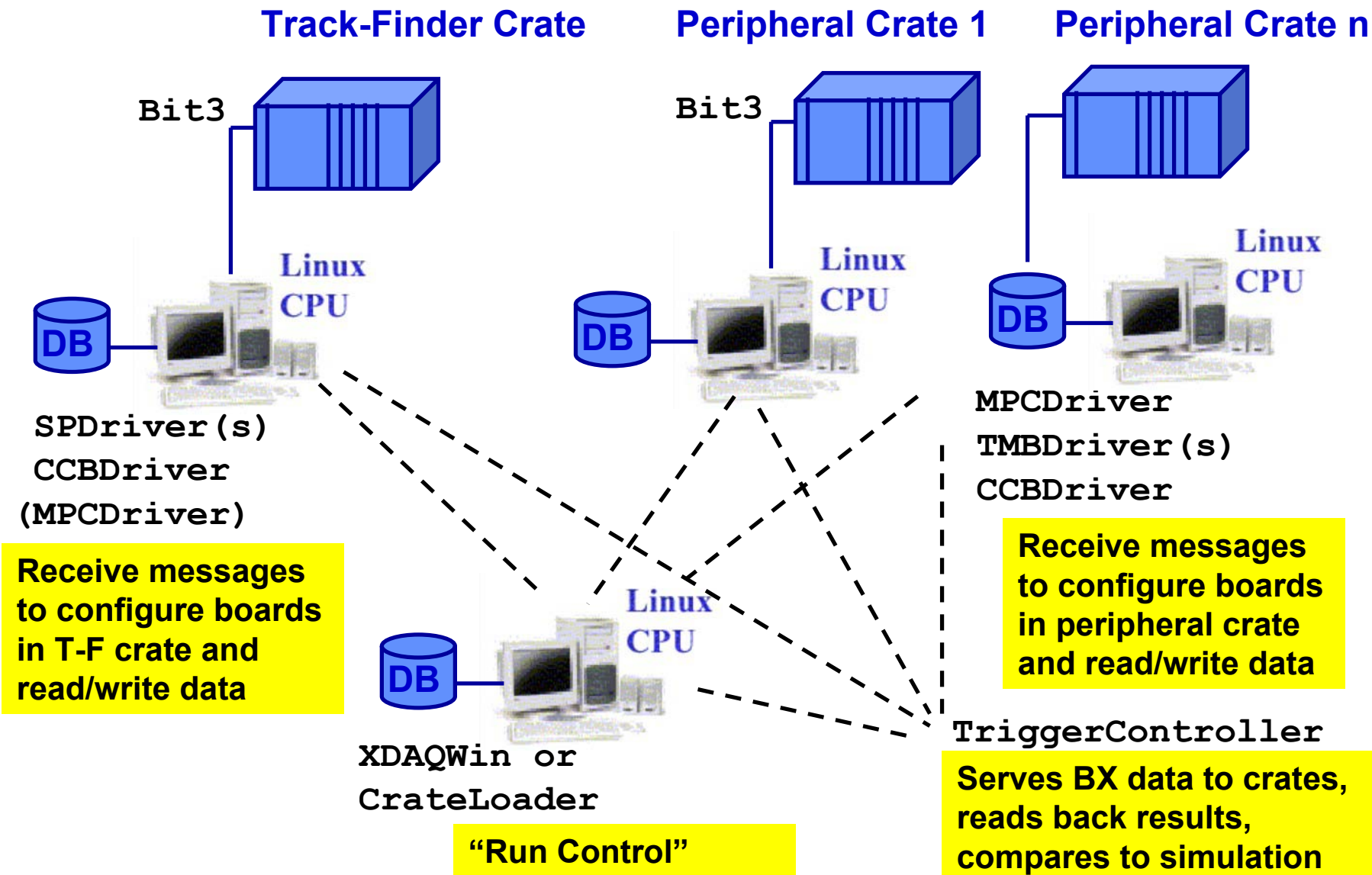
Bit3

Linux CPU

Bit3

Linux CPU

Linux CPU

DB

DB

DB

```
SPDriver(s)
CCBDriver
(MPCDriver)
```

```
MPCDriver
TMBDriver(s)
CCBDriver
```

Receive messages to configure boards in T-F crate and read/write data

Receive messages to configure boards in peripheral crate and read/write data

Linux CPU

DB

```
XDAQWin or
CrateLoader
```

"Run Control"

`TriggerController`

Serves BX data to crates, reads back results, compares to simulation

Package Organization

Presently organizing code in repository into packages that correspond to which XDAQ Executive application it is expected to run in:

➡ **TrackFinderCrate**

    ❏ **Here the SP classes based on HAL reside, and the XDAQApplication that wraps it and receives SOAP messages**

➡ **PeripheralCrate**

    ❏ **Here the TMB, MPC, and CCB classes based on HAL reside, and the XDAQApplications that wrap them and receive SOAP messages**

➡ **TriggerController**

    ❏ **This application talks to our standalone ORCA simulation to serve events, downloads data into boards (above applications), reads back results, and compares to simulation**

Some details…