
NATIONAL
INSTRUMENTS™
ni.com

Session TS1214
"Porting a Legacy Control System to PXI with LabWindows/CVI Real-Time"

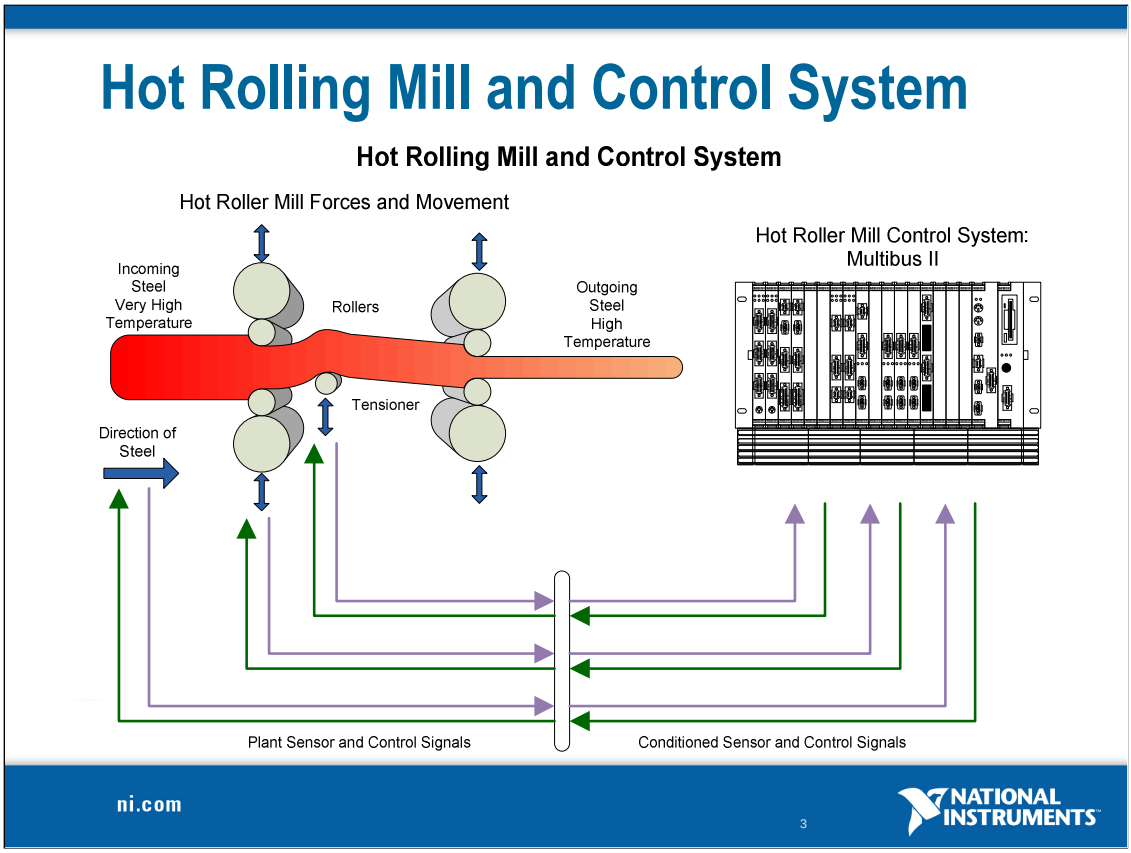
A real-world example of a factory control system was ported from a Multibus architecture to the PXI platform. This session will cover economic benefits of the conversion, why PXI and LabWindows/CVI Real-Time were chosen for this system upgrade, and how existing C software was ported, key lessons learned, and performance characteristics of the final system.

Presenters: David Wheeler and Nate Mackley, Mantaro Networks

About Mantaro

Mantaro provides a full range of product development services to our technology clients. Our technical staff is composed of highly talented professional engineers with a history of successful product development and innovative design experience.

Hot Rolling Mill and Control System



Model of a 4 high hot rolling mill with one Multibus II controller per mill stand

Main components of the mill include

- •Hot steel feeding left to right – note pinch points
- •Work Rollers—parts that actually come in contact with steel
- •Backup rollers---rollers with actual hydraulic and motor forces
- •Tensioner—maintains tension of steel between mill stands

IO of the mill

- •Many forces, temperatures, speeds to measure
- •Many transducers feed back to control the process.
- •Data comprised of analog, digital, counter and other signals
- •IO also passes from Mill to Mill and Configuration systems
- •Plant isolation between the mill and MBII, local at the MBII acquisition cards

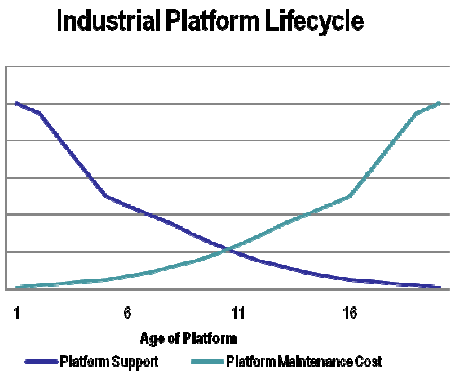
Multibus system

- •Comprised of various IO and communications modules
- •Approximately 6 cards involved in control, configuration and communications processes

Mill Operation

- •Things happen quickly, acquisition occurs on a 1ms time base
- •Total processing of a strip of 2 inch steel can be three minutes
- •A failure can happen within milliseconds, resulting in thousands, or even hundreds of thousands of dollars of lost material and machinery in less than one second.

Business Situation: Critical Equipment Obsolescence



- Multibus II hardware had reached obsolescence
- Hardware was proprietary to original system integrator

ni.com

4

NATIONAL INSTRUMENTS

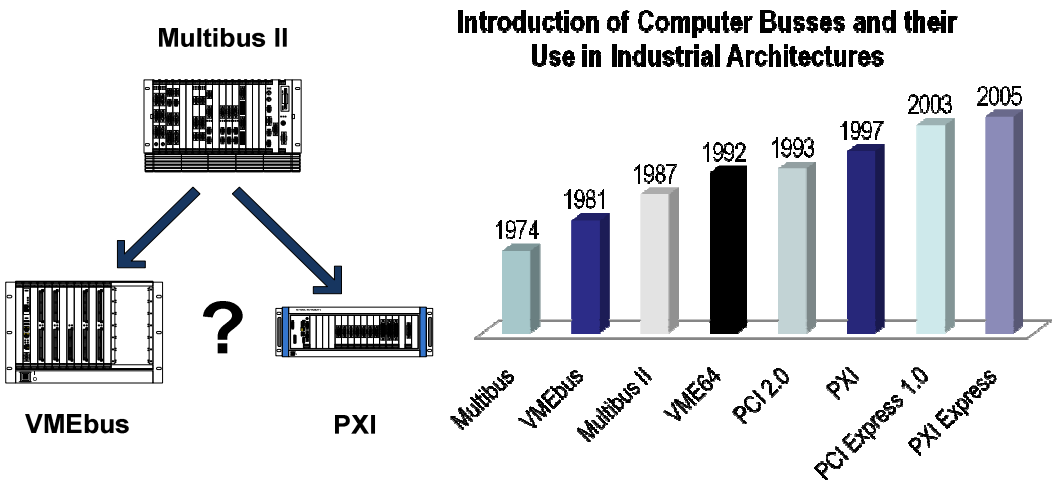
Multibus II hardware had reached obsolescence

- •Components were largely unavailable
- •Components prohibitively expensive

Hardware was proprietary to original system integrator

- •Support for the hardware and software was largely unavailable
- •No directly compatible or comparable Multibus II hardware

Platform Selection: VMEbus vs PXI



- MANTARO engineers evaluated VMEbus and PXI platforms

ni.com 5 NATIONAL INSTRUMENTS

Platform Options:

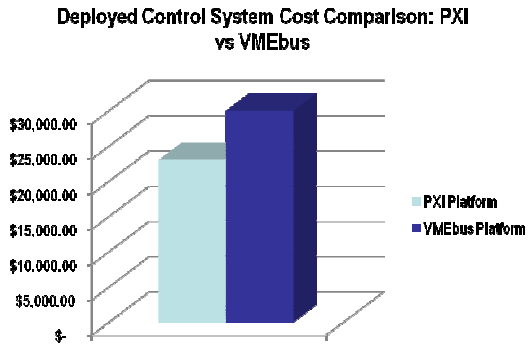
- •Multibus II platform superseded and is effectively obsolete
- •VMEbus has received significant upgrades over the life of the platform
- •PXI platform built upon the newer PCI technology platform
- •PXI Express does not yet have the IO assortment to support the system specifications

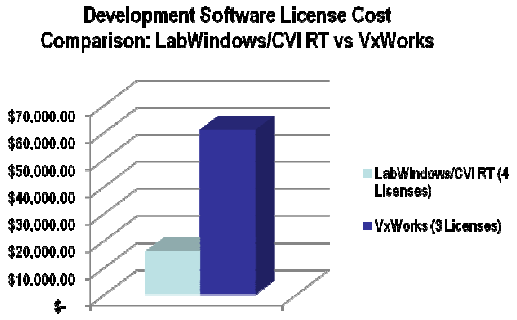
Platform Selection Process:

- •Mantaro engineers evaluated performance and risk of VMEbus with VXWorks versus PXI with LabVIEW Real-Time running LabWindows/CVI Real-Time.
- •Client reviewed cost, features, reliability, previous company wide experience and selected the National Instruments solution

Platform Selection: PXI Advantages

- Single source vendor for all hardware and software tools
- Lower system and development tool costs
- Simplified license portability from client to Mantaro
- Multicore support : PXI-8106 running Intel Core 2 Duo





ni.com 6 NATIONAL INSTRUMENTS

Client viewed single source as ideal.

- •Simplified development,
- •Easier maintenance and
- •Faster, simpler procurement chain

System Cost

- •6k system cost reduction
- •Development software license dramatically cheaper
- •easier to transfer state to state and company to company

Wind River licensing did not readily support cross state transfer

PXI was the ahead of VxWorks 6.6 in Intel Core 2 Duo support

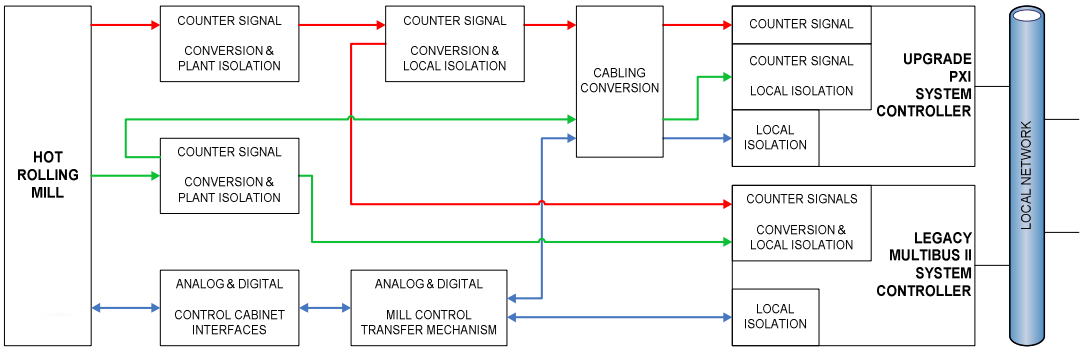
- •A key performance benchmark of the client

The National Instruments DAQmx driver base decisive element of the PXI selection

- •Greatly simplified hardware setup,
- •Reduced development effort
- •Simplified, nearly transparent transition to alternate NI cards

Hardware Architecture

- Utilization of legacy control cabinet interfaces
- Parallel control systems during development



Upgrade system designed as a drop in replacement, no changes required to be made to the legacy system interfaces

Upgrade system comprised of the

- •PXI-1045 18 slot chassis
- •PXI-8106 Controller with an Intel Core 2 Duo processor
- •PXI-6230 and PXI-6232 Multifunction Cards
- •PXI-6511 and PXI-6512 Digital IO Cards
- •PXI-6602 Counter Card
- •National Instruments terminal blocks to provide pinout conversion
- •Signal interface panels to provide parallel inputs to the control system and transfer functionality of the control system outputs

PXI and MBII systems received data in parallel, but outputs could not both drive the mill.

- •Test solution was Shadow Testing, where the PXI would output a control signal based upon input conditions.
- •This output was to be monitored and evaluated against the MBII system actually running the mill.

Development Challenges: Mapping IO to Acquisition Capabilities

- Control system utilized common analog and digital IO signals at relatively low sampling rates
- Dissimilar counter signals required multiple hardware solutions
- Constraint:
  COTS only solutions

ni.com 8 NATIONAL INSTRUMENTS

Signal Conditioning and IO Characteristics:

- Analog and digital IO and low speed counter channel count and performance specifications were well within the capability of common PXI cards.
- High speed counter needed additional signal conditioning

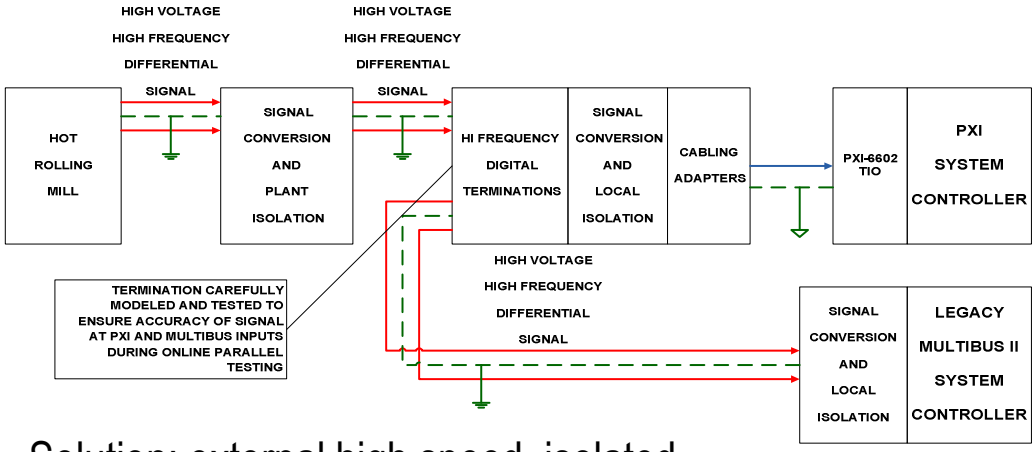
High speed counter performance required careful attention.

- Proper reading of signal necessary for critical functions and state transitions.
- +/-7 volt switching differential
- 2 MHz bandwidth
- Local isolation
- Long cable lengths

TTL level signal available from the signal converter which was between the mill and the control system.

- Use of TTL would have simplified signal input into the PXI-6602.
- Environmental conditions prohibitive to using TTL

Development Solution: Model and Test



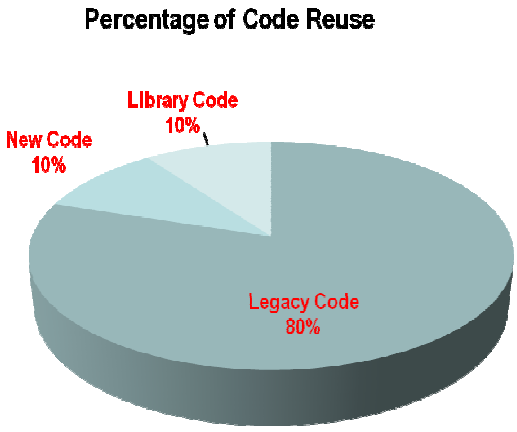
- Solution: external high speed, isolated, differential to TTL signal conditioning

External Signal Conditioning

- In order to maintain an all COTS approach a third party device was utilized to provide the signal conditioning needed to integrate the mill signal to the PXI-6602
- Unmatched impedance between signal conditioning hardware
- Circuit modeled and tested in order to optimize the termination
- Final solution accepted by client and implemented

Software Development

- Reuse legacy source code
- Replace library functions
- Add management features



ni.com

10

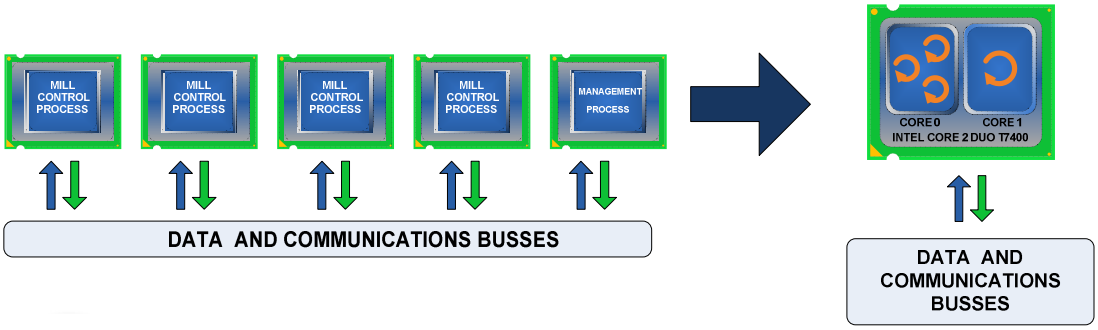
NATIONAL INSTRUMENTS

Existing code base in C.

- Goal: Don't modify behavior of control processes
- Approach: Reuse as much code as possible
- No source code available for many library functions
- New code required to adapt to new platform, new software arch, new features.

Software Architecture

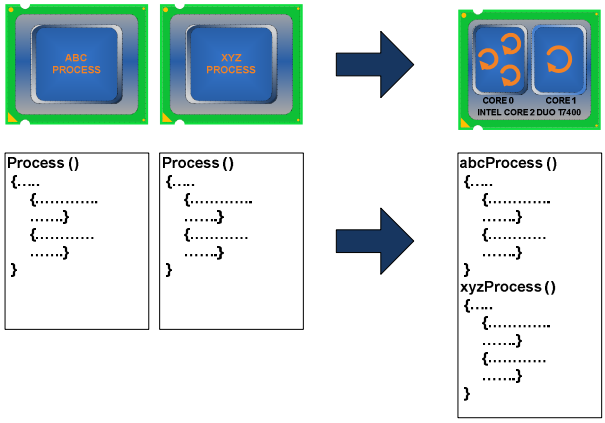
- Multiprocessor Becomes Dual Core with Multithreading



Multiprocessor versus Multicore / Multithreading

- Original system deployed using multiple processors.
- New system uses single processor consisting of two cores.
- New software architecture uses multiple threads in a single process.
- Need to implement new s/w arch.
- Need to implement new communications between processes.

Development Challenge: Name Space Collisions



- Legacy code used common names across processes

ni.com

12

NATIONAL INSTRUMENTS

Early Development Challenge: Namespace Collisions

•An early challenge was a problem with combining code from multiple namespaces.

•Original system consisted of multiple instances of application framework.

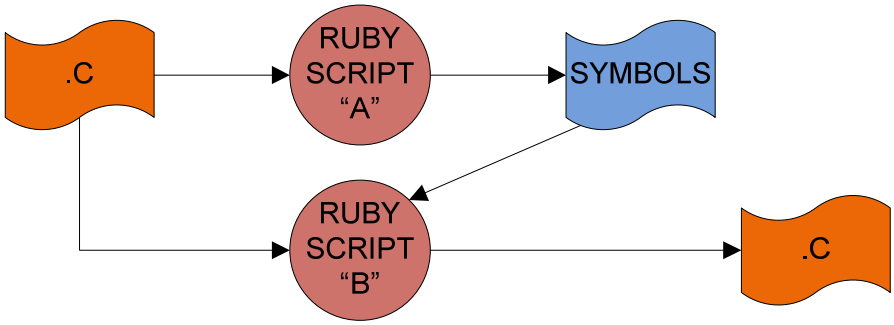
•Each instance of framework was customized for its own requirements.

•Simply combining all processes into one resulted in massive name space collisions.

•Different components with shared names would need to be renamed.

•(Original plan to use C++ was thrown out when LabWindows CVI/RT chosen.)

Development Solution: Ruby Scripts

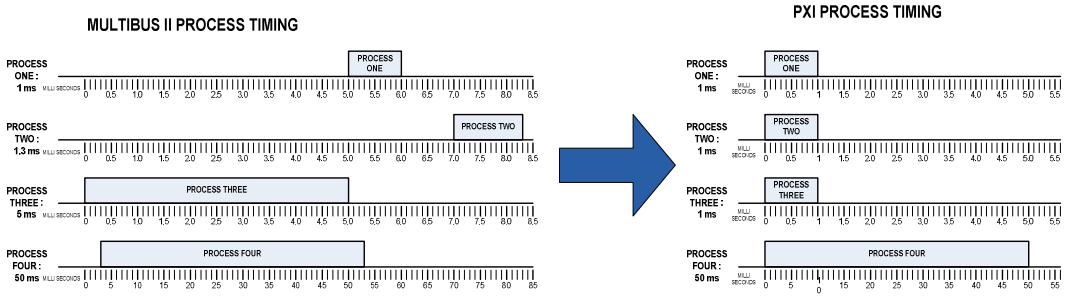


- Process source code to detect name collisions
- Reprocess to replace aliased names
- Cleanup by hand
- 1 week development & discard tools when complete

Solution: Rename the Symbols

- Symbol renaming needed to be automated.
- Ruby scripts used to identify colliding names and replace them.
- Resulting output was cleaned up by hand.
- Ruby, Python, Perl, other scripting languages equally good at this task.
- Throwaway code: once developed, its usefulness ends.

Development Challenge: Timing Changes Affect Behavior

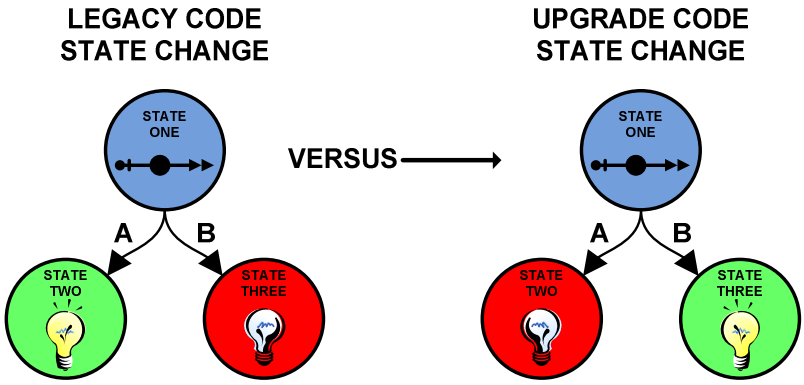


- Up to a 5x reduction in required process time

Timing Changes:

- Original system timing was very asynchronous.
- Processes were truly parallel. Periods and phases were independent.
- New system uses single timing reference.
- Serialized execution of major control processes.
- Periods of some processes have changed.

Development Challenge: Timing Changes Affect Behavior



- Race conditions revealed

ni.com

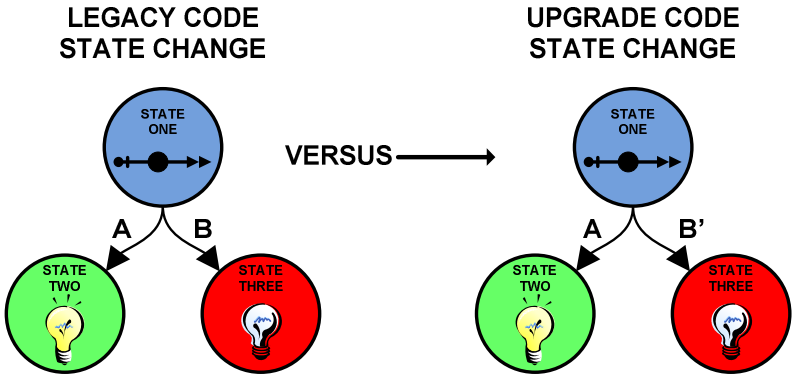
15

NATIONAL INSTRUMENTS

Timing Changes Results in Race Conditions

- Race conditions in original code exposed.
- Original system, race always went to event A due to communication latency.
- New system, race sometimes goes to event B.
- Different behavior is incorrect and needed to be corrected.

Development Solution: Correcting Race Conditions



- Detected during On-line Shadow Testing
- Fixed by qualifying transitions

Modify Original Source Code

- Shadow testing revealed incorrect state transitions.
- Original system transitions to A, new system to B.
- Behavior diverges from this point on.
- Had to modify, qualify transition conditions to eliminate race condition.

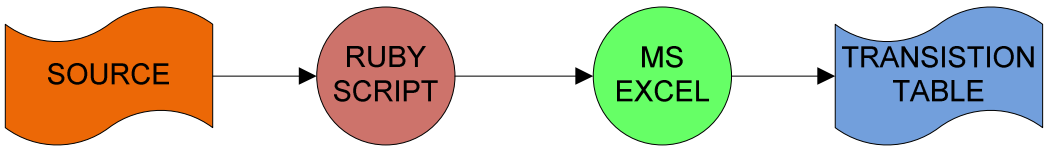
Development Challenge: Reverse Engineering

- Needed to reverse engineer state transition matrix
- Original documentation very old
- Source code maintained by client
- Documentation incomplete and incorrect

Existing Documentation was of Limited Usefulness

- Documentation for original system was out of date.
- Changes to s/w after original docs produced w/o corresponding doc updates.
- For testing, we needed correct documentation of state transition tables.

Development Solution: Machine Generated Documentation

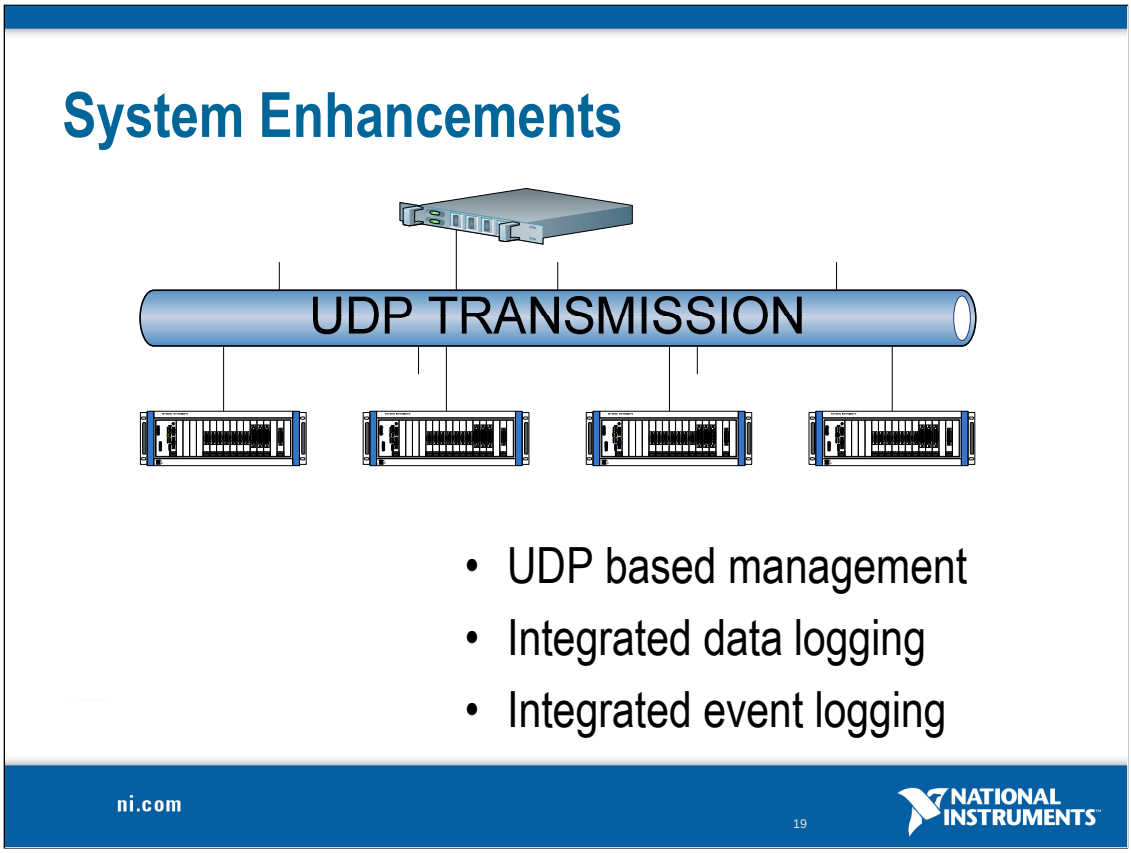


- Accurate documents key to testing and debugging state machines

ni.com 18 NATIONAL INSTRUMENTS

Automate the Generation of New Documentation for Porting and Future Maintenance

- Solution: generate new documentation using scripts.
- Ruby script analyzes source code.
- Ruby script uses COM automation to drive excel.
- New state transition tables generated as Excel spreadsheets.
- Generated documentation guaranteed to be accurate.
- Easy to regenerate when future changes are made.
- Clear, complete documentation essential for debugging.



Communication Enhancements

•Enhanced system functionality: UDP management.

•Communication between controller and WonderWare dictated use of UDP.

•Original system used OSI protocols.

•Original system didn’t have available resources to report performance data.

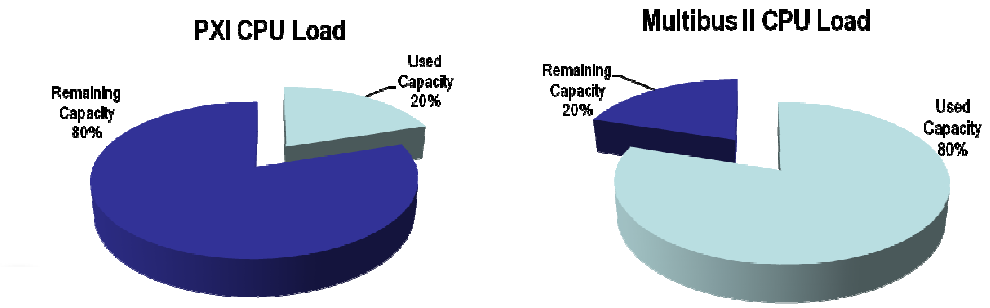
•New system easily handles communication load.

•Easier to manage and monitor system.

•Detailed performance data valuable for process tuning.

Project Results: Performance

- No system hangs during 6 months of development
- Faster response to hot mill changes than legacy system
- Reduced CPU Load



ni.com

20

NATIONAL INSTRUMENTS

Performance Benchmarks:

•During system testing, new system has proven to be reliable.

•Reboot time/performance improvement: 15 minutes vs 30 seconds boot time.

•Sub-millisecond latency on all process control signals.

•CPU load: 20% on each core.

•This means high confidence that real-time deadlines met.

•Also means plenty of headroom for future enhancements.

Project Results: Current Status

- Upgrade system was successfully built and installed
- MANTARO and client continue working through testing phase prior to deployment

Project Results:

- Project is now in testing phase.
- Not online yet.
- Confidence in new system continues to grow.

Project Results: Summary

- Initial technology / architecture evaluation crucial to risk mitigation and providing client with a clear set of criteria to choose the final solution
- Adequately understand the legacy system to design an upgrade, yet avoid the unnecessary and costly effort to recreate the entire code base
- Use of custom, throw away scripts crucial to extracting accurate condition of operational system

Summary:

•Investigation of wide range of technologies led to choice of NI and LabWindows CVI/RT

•Original system behavior maintained by re-use of software.

•Many technological hurdles crossed in porting code to new system.

•System in late stages of integration testing, nearing deployment.