CONTROL SYSTEM SEGMENTATION*

K. S. White, M. H. Bickley, Thomas Jefferson National Accelerator Facility, Newport News, VA, USA 23606

Abstract

Over the past seven years, Thomas Jefferson National Accelerator Facility's (Jefferson Lab) control system has grown to include more than two hundred distributed computers running over a complex segmented network, controlling a number of semi-independent operational plants. Several of the plants, including that used for running beam for physics users, operate around the clock with only brief, scheduled interruptions for machine repairs. Because of this, high control system availability is critical. Dividing computing resources into distinct sections, called fiefdoms, improves availability of the control system for each plant while facilitating periodic maintenance. In order to maximize uptime, each fiefdom operates as a completely independent control system consisting of a file server machine with a complete set of control system software and files, a local network, operator consoles and computers to execute high and low level control programs. The fiefdoms are isolated using network hardware, while still allowing limited communication between them. By segmenting the control system in this manner, the effect of a computer failure is minimized and machines can be taken down for periodic maintenance and upgrades without disabling other controls capabilities for the site.

1 INTRODUCTION

Modern control systems are composed of a combination of computers, networks, executable programs and configuration data. The computers generally act as file servers, back-end computers for operational consoles and system and high-level functions, and front-end computers for device control.[1] At Jefferson Laboratory, where the Experimental Physics and Industrial Control System (EPICS) forms the basis of the control system, the front-end computers are referred to as Input/Output Controllers (IOCs). The configuration data is most commonly stored in flat files or a database. For a large project, the control system provides services for a variety of functional areas, called plants. Conceptually, a plant can support a physical area such as a cryogenics facility or an accelerator. Alternatively, a plant can support a logical function such as control system development and testing, which should be isolated from interaction with operational systems. A flexible system like EPICS can be configured in a variety of ways to provide services to the various plants. The specific configuration has a significant impact on the availability and maintainability of the overall system. There are two ways to approach the design of the combination of the control system elements.

1.1 Monolithic Model

First, and most simply, in a monolithic model, control system services are provided to all plants via shared resources. In this simplified configuration, shown in Figure 1, a single file server can provide files for all computers and a single network can provide the means for all data and file transmission. While this model can work well and is the easiest to implement, the effect of a failure of a file server, network switch or client program can prevent the entire system from functioning, a catastrophic effect in an environment where high availability is required.

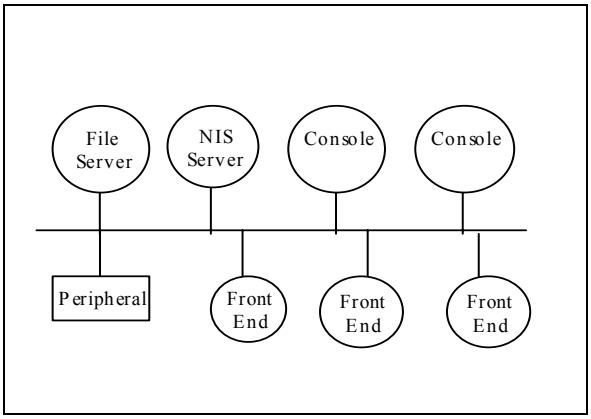

Figure 1: The Monolithic Model

1.2 Segmented Model

Alternatively, to construct a model that minimizes the effect of a file server, network component or program failure, the control system may be segmented in smaller, independent entities called fiefdoms. In this model, each fiefdom hosts its own set of operationally necessary resources and does not depend on resources from any other fiefdom for regular operations. A schematic of a fiefdom is shown in Figure 2. In this model, resources that are not critical for operations, such as the backup system may be shared between fiefdoms.

* This work was supported by the U.S. DOE Contact No DE-AC05-84-ER40150

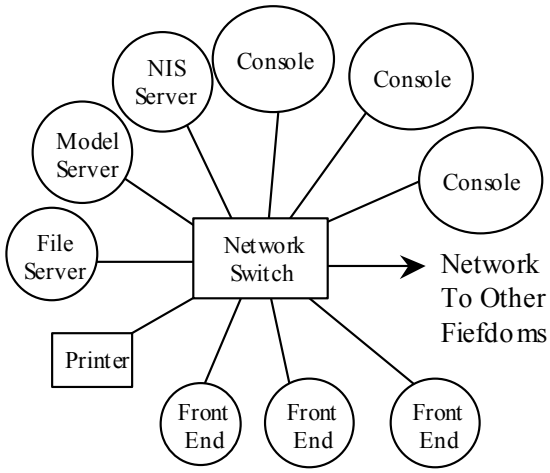

Figure 2: Components of a Typical Fiefdom

2 SEGMENTATION

A fiefdom is comprised of all the same elements present in a control system, but generally supports only a portion of the controls functions for the laboratory. In this model, multiple fiefdoms must then be created to provide the complete set of control system services needed for the site. For example, at Jefferson Lab, the accelerator control system is split into eleven independent fiefdoms, as shown in Figure 3. Some of these fiefdoms are for machine operations and others are for support functions. Each fiefdom has at least one file server, back-end computer, front-end computer and network segment. Additionally, the file server in each fiefdom stores a complete set of program and data files needed to initialize and operate each computer within the fiefdom. This configuration allows each fiefdom to function regardless of the status of the servers, network and programs in other fiefdoms.

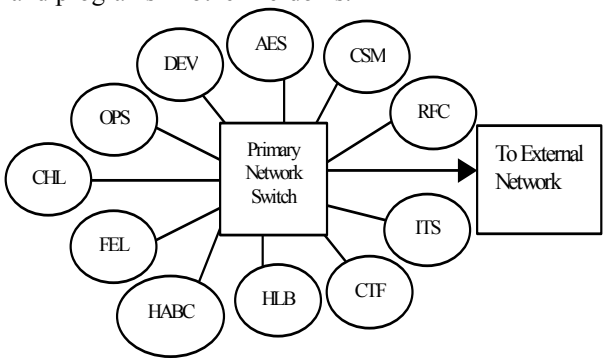

Figure 3: A schematic of the Jefferson Lab Fiefdoms

2.1 Improving Availability

This model has several advantages for machine availability. First, the overall effect of a single server or network component failure is localized. This feature means the cryogenics plant, where the effect of a control system failure can be very expensive due to helium loss, can continue to run smoothly even if the accelerator operations server is down. This would not be the case in the monolithic model. Secondly, the recovery time from a failure can be minimized. By limiting the number of computers impacted by a failure, and therefore the number of machines and services that need to be repaired and restarted, recovery can occur more quickly.

2.2 Improving Maintainability

In addition to increasing availability, control system segmentation provides a more maintainable system. Modern computers require periodic reboots for hardware and software maintenance. It is especially critical to be able to apply security patches in a timely fashion. Additionally, many labs choose to implement projects that require control system upgrades and enhancements, which necessitates changing the computers, the network, or their configuration. The isolated development fiefdom allows programmers to write or modify programs and test in an environment that mimics the accelerator control system without the risk of causing an error on the operating machine.

At a laboratory with several plants, it is often the case that the plants are on different maintenance cycles, so that when one is down the rest are still required to be operational. Independent fiefdoms provide a reliable mechanism to support regular maintenance that can be scheduled on individual plants without impacting the availability of other plants. Without such system isolation, computer reboots and network reconfiguration work would result in extensive controls system outages or require special provisions to keep the necessary systems running during the targeted outages.

3 IMPLEMENTATION

Creating a segmented control system requires more computer and network hardware, and is more labor intensive that the monolithic model. At least one server machine must be provided for each segment, along with a network switch. Once the additional hardware is in place, the more difficult task of separating files and file references, and configuration management must be accomplished. Once the segmented system has been set up, regular maintenance activities are required to ensure the continued integrity of the configuration. We use a combination of system administration tools and configuration management tools and processes to effectively manage Jefferson Lab's eleven fiefdoms.

3.1 Network Configuration

The control system networks must be configured to support the implementation of fiefdoms. The network is completely packet switched, isolating segments of the network from each other. Each fiefdom has its own

dedicated subnet and at least one network switch. The switch supports network communication between all computer nodes within the fiefdom. As long as the fiefdom's switch is working correctly, intra-fiefdom communication continues. The switches of the various fiefdoms are connected with each other, typically with higher-bandwidth connections primarily used to support the centralized operation of computer backups.

3.2 File Management

One of the primary goals of a segmented system is to allow any given fiefdom to continue operations in the absence of any other fiefdom. In order to ensure independent operation, each fiefdom file server must contain a complete set of files needed to initialize and operate the plant. This requirement means a large number of files must be maintained in multiple locations. For example, all the EPICS system tool executable programs must reside on each fiefdom's file server. Further, when a program is upgraded, the new version must be propagated to all the fiefdoms. Similarly, the EPICS database files used to configure EPICS IOCs must be present on the proper fiefdom file servers in order to provide initialization and boot services to the local IOCs. Scripts are provided on the development fiefdom to facilitate file management, and to allow developers to install new versions of software on multiple fiefdoms automatically.

3.3 Use of the Network File System

The automount feature of the Network File System (NFS) is used to maximize fiefdom transparency while minimizing interdependencies. This enables a computer in any fiefdom to access the files on another fiefdom file server, ensuring that only the file systems actually in use are mounted. This mechanism prevents timeout problems that can occur when a file system is mounted and the host server becomes unavailable. This feature is used to allow system administrators to provide each user with a single home directory that can be accessed regardless of fiefdom, as maintaining duplicate user accounts on each fiefdom would require additional, unnecessary work. The pathname used to refer to a file on another fiefdom includes the fiefdom name in the path, preventing ambiguity about the physical location of files. At the same time, each fiefdom is configured with a number of soft links that hide fiefdom-dependent aspects of the file systems. Programs can then be written in a fiefdom independent manner; they will run on any fiefdom without file pathname changes.

3.4 Use of the Network Information System

A second way in which computer system management is used to isolate fiefdoms is through the Network Information System (NIS). To simplify administration, the accelerator control system uses a single NIS master server, which provides a single location for passwords, domain names, groups, and other system administration services. The data in the master server is mirrored to one slave NIS server for each fiefdom. In the event of a failure of the master NIS server, each fiefdom can contact its slave server, which has system configuration information as recent as its last contact with the master.

3.5 Software Configuration and Management

In order for a segmented control system to work properly, all executable programs and scripts must be written with the fiefdom organization in mind. It is particularly important to structure file access in a fiefdom independent manner. In order to prevent cross-fiefdom dependencies, and their associated problems, programs must avoid references to services on other fiefdoms. Ensuring developers follow established programming practices is a key aspect of implementing and maintaining a segmented control system. Once the segmented structure has been established, it is important the fiefdoms remain free of dependencies. At Jefferson Lab, this is accomplished by monitoring of each of the fiefdoms. If this automated check detects persistent inter-fiefdom links, then control system administrators investigate the source of the link, working with software developers and others to eliminate the connection.

4 SUMMARY

The Controls Software Group at Jefferson Laboratory has successfully transitioned the accelerator control system from a monolithic to a segmented model. Additional resources were required to implement the segmented system, due to the need for additional hardware, more complex file management requirements, additional programming complexity and greater system administration needs. However, because of Jefferson Lab's operational upgrade and maintenance requirements and the large size of the controls system, the benefits of segmentation justify the additional cost and effort. The limited time available to maintain operational systems is used more efficiently, allowing the lab to dedicate more time to other maintenance and operational needs. Upgrade work is simplified because the impact of a change is minimized and realistic testing is facilitated. Additionally, on the few occasions that there have been component failures, the isolation has worked effectively to prevent limited disruptions from becoming widespread.

6 REFERENCES

1 M. E. Thout, L. R. Dalesio, "Control System Architecture: The Standard and Non-Standard Models", Proceedings of Particle Accelerator Conference (PAC '93), Washington, D. C., USA, May 1993, pp 1806 – 1810.