Safeguarding Your Plant Automation Programs with Change Management

By Gary Gillespie  
M.S. Mechanical Engineering  
B.S. Mining Engineering
The increased use of plant floor automation to achieve production goals has created a dependency on PLCs, PC control systems and programmable automation. These devices and their logic programs are costly to develop but vital to the running of the plant, and are viewed by most companies as corporate assets. It is incumbent on plant and corporate management to insure that proper safeguards are in place to protect and manage change in these assets. This paper examines the sources and types of changes that take place in plant automation environments, and the considerations and approaches necessary to safeguard your automation systems through the effective use of a Change Management System (CMS).

Georgetown University defines change management as the complete set of processes employed to ensure that changes are implemented in a visible, controlled and orderly fashion. Focus should be placed on several key elements:

1. Process: Must be a structured method that stays the same
2. Completeness: Must include all objects (person, place or thing)
3. Visibility: Without visibility, Change Management doesn’t exist
4. Control: Who, what and where
5. Order: Changes may need a particular sequence to be effective

Change Management as it applies to plant automation focuses on the control systems that operate the production equipment. Change Management System (CMS) applications have matured over the past 20 years along with the increases in sophistication and capability of the automation devices and control software developed by automation vendors.

There are many elements of a CMS that make its function distinct from single-file-based source code control systems, with the most distinct differences being a suite of tools to manage a group of files (often referred to as a “project”) as a single entity, and processes for detecting source code changes outside of source code control. In most automation devices the entire project file is necessary when managing revisions and identifying changes.

An automation Change Management System is a centralized system that manages changes to program logic for controls programs and devices such as PLCs, CNCs, HMIs, PC control systems, robots, drives and general automation programs. A typical small plant will have a few hundred programs that should be managed, while large plants will have several thousand. Over the life of a facility the investment in program logic alone represents a significant expenditure that should be preserved and optimized. In order to do this a CMS should have the following features:

- An archive of prior revisions of programs
- The ability to detect changes
- Tools for documenting changes and making them visible to users
- A historical record of who made the change, when, and from where it was made
- Secured user and workstation access
- Features for controlling editor operations mapped to user permissions
- Procedures for recovering from hardware failures
- Change notification
As automation devices have grown more complex and have incorporated more plant data in their operation, there is an increase in the need to make adjustments to variables and logic to continue smooth operation. These adjustments may be minor individually, but are directly linked to machine throughput and uptime. If the current device program and configuration are lost, and an old version of the device program must be used, the result is decreased machine performance, decreased quality and/or downtime. While this situation is costly enough, consider the ramifications to plant operation if there are no older versions of a lost program available and the program must be completely rewritten. This can and does happen, and the effects can significantly impact safety and plant throughput for months. These impacts added to the cost to re-rewrite, test and commission a single program are often greater than the cost to implement a plant-wide CMS product.

Consider the ramifications to plant operation if there are no older versions of a lost program available and the program must be completely rewritten.

### Impact of Plant Activities on Versions of Program Logic

Each facility has a unique set of change types and frequencies that can affect a CMS strategy. A selected set of activities is outlined here to prompt further thought and highlight the need for a proper implementation of a CMS in order to achieve optimum results.

- **Nature and Frequency of Changes**: It is important to ensure that an adequate number of program copies are available (including the “as delivered” copy from the vendor/integrator) to ensure that changes can be classified and reviewed. Some changes represent true improvements, while others highlight a process problem or training issue that should be addressed by other means.

- **Process Enhancements**: If changes are made in the process that make prior versions of the program obsolete, these enhancements should be clearly identified so that users do not revert to an older version of a program to fix a new issue. Plant operating guidelines should identify when the deletion of prior programs is warranted, and which users will have this permission.

- **Unmanaged Changes**: Without a CMS the controls engineer would use the editor software on a workstation or laptop to make changes in a device. If multiple people make changes from multiple computers (as is especially common in multi-shift operations) the documentation of changes is often lost. Using a CMS to compare the program running in the device with the last recorded version, a facility can identify changes that were made outside of the CMS. Once the CMS is implemented and sufficient device networking is in place, edits outside the CMS should be discouraged. These unmanaged edits are often prohibited in more regulated facilities using a CMS.

- **Temporary Changes**: It is common to make a “temporary” change to a program to resume operation while a maintenance task is performed on a failed component. It is also common for these temporary bypasses to be forgotten, which can result in serious safety issues. A CMS is used to note these temporary changes and provide a means of easily restoring a prior version of the program once maintenance is complete.

- **Multi-Process or Recipe Operations**: In facilities that run different processes or recipes it is important to manage which version of a program is being updated. The creation of specialized copies of programs to use as “master versions” for each of these processes can aid in managing them efficiently.
There are many individuals in a facility with the need to make automation changes. Each of their roles and the unique characteristics of the changes they make to the automation layer should be considered. This enables roles to be established in the CMS to manage who can make certain types of changes. For example, some users will have the ability to go online and monitor a device operation, another group will be able to make changes to the program, while others can delete older programs, change the program in use to an alternate version or compare changes between versions of programs. A representative list of user classes is outlined below, along with typical types of changes and plant considerations.

- **Engineering**: Often an engineer is concerned with process improvement, and will make enhancements to the automation systems to increase throughput, visibility, reliability, etc. Engineers will need the ability to modify the running program, change to an alternate version of a program and compare versions of programs for differences.

- **Maintenance Technicians**: One of the primary reasons for maintenance technicians to make program changes is to deal with a failed or degraded component that is impacting production. This could be a sensor that is reading incorrectly or a contact that has malfunctioned. Technicians will often need to “bypass” an input to the program while repairs are carried out, and the need to identify those temporary bypasses to ensure they are removed is a key benefit of a CMS.

- **Contractors**: Many organizations will utilize contractors to perform functions that require specialized skills that are not retained in-house. A CMS will provide the facility with the ability to manage the permissions granted to contractors and detect changes made by them.

- **OEMs**: When the Original Equipment Manufacturer is in the facility installing or updating their equipment it is vitally important to capture the “as delivered” program copy and save that as a reference version in the CMS. This reference can be invaluable when identifying problems with the machinery and working with the OEM on corrective actions. The reference copy can be compared to other versions of the program to identify not only the specific changes, but also enable experts to spot trends in the changes which often point to training or raw material issues.

- **Operators**: The typical change that an operator would make to an automation program would be to switch between versions of the stored program when production objectives change. Having a log showing the history of these changes can often be useful.

- **Administrators**: A CMS administrator will be involved in implementing plant policy within the CMS to assign roles, configure users, workstations, etc. and perform sensitive system maintenance that most users should not have the permission to perform. The administrator should also ensure that the database and files associated with the CMS are properly backed up per plant IT guidelines.

Technicians will often need to “bypass” an input to the program while repairs are carried out, and the need to identify those temporary bypasses to ensure they are removed is a key benefit of a CMS.
There are many events that can have a negative affect on plant performance, and some that represent serious safety hazards. Reliable automation control logic can be compromised by the following events:

- Human Error
- Equipment failure
- Sabotage
- Power surges / interruptions
- Fire

A CMS offers the following protections from these events:

- **Human Error:** If someone makes changes to a program that result in undesired performance, or corrupts the program due to inadvertent changes, the prior version of the program is readily available.

- **Equipment failure:** Equipment can and does fail. If the hardware fails and the only good copy of the program logic was in that hardware, the plant has a problem. With a CMS, the hardware is replaced and maintenance personnel download the latest version of the program to the processor resulting in only a few minutes of downtime.

- **Sabotage:** As unfortunate as this threat is, someone can connect directly to many devices (especially those in remote, unsecured locations) and modify the program with harmful results. A CMS is designed to store processor passwords (in the case of some processors) so these are not available without going through the CMS. Also, the CMS will periodically upload the logic from the processor for comparison with a copy on file. Changes can be identified in graphical detail, and immediate notification can be sent to responsible individuals.

- **Power surges / interruptions:** Power issues can cause equipment to lock up or go off-line. If these situations result in a loss of the program, it can be downloaded from the CMS after the hardware is reset.

- **Fire:** Any fire will be a major disruption. Whether a single device or an entire facility is lost, having all program logic stored in a central, organized CMS repository accelerates the time and decreases the cost associated with resuming production. Insurance underwriters are beginning to factor in the use of a Change Management System in assessing the risk profile of facilities.

Without proper system safeguards these events can lead to increased downtime and an increase in “mean time to repair” (MTTR). Recovering from these events quickly requires adequate planning on the hardware and maintenance strategy, and a reliable and recent backup of the automation control program logic. Current and complete backup copies of the program logic require the features of a CMS. While a manual backup approach may appear adequate at first glance, experience has shown that plant personnel have too many tasks that compete for the time to manually back up programs on a consistent basis. Also the increased visibility of changes through better reporting and the potential for process improvement brought about by the effective use of a CMS application can quickly pay for the CMS.
What types of device programs should be put under the management of a CMS? The types of devices to be maintained by a CMS will vary from manufacturing facility to manufacturing facility, and each of these device types has a wide variety of manufacturers, network configurations, models, support software, and versions of support software. To properly manage these programs and devices, all these factors have to be understood. The most common devices are defined below, along with any unique considerations from a CMS perspective.

- **PLCs** – Programmable Logic Controllers. This is the most common type of device supported by a CMS. There are a significant number of PLC vendors and each has a proprietary software package for PLC program editing. A CMS application should be able to interact with all of the major brands of PLC editor software to capture and detect changes, and provide an alternate method of capturing the project files on the editor workstation in the case of legacy or non-standard devices. A newer variant of the PLC, called the Programmable Automation Controller (PAC) performs the same tasks along with some expanded capacity for managing multiple applications simultaneously.

- **CNCs** - Computer Numerical Controllers. The CMS will need to access the control ladder, part programs, and process parameters as part of the backup process. These can be located in standard locations as defined by the manufacturer, or they can be located in custom locations as developed by the OEM. Secondly, file locations can be different based on types of CNCs within a model. For example, a single or dual axis CNC within the same model can have different file locations. Prior to the installation of a CMS these factors should be defined and documented. Company standards for OEM code design can greatly standardize the logic designs, and reduce CMS costs for supporting CNC devices.

- **Robots** – Most robots that have FTP communications are straightforward for change management. The robot manufacturer or plant controls group should provide the file list for each type of robot.

- **HMI** – Human Machine Interface. CMS applications typically provide integrated support for commonly used HMI packages. If an HMI is used that the CMS does not have a unique driver for, a generic module that backs up the PC workstation development environment files can be used.

- **PC Controls** – These types of devices are typically used in machining operations. These are PCs that are running a control program that mimics the operation of a PLC. Some of these differ slightly from the PLC programming software they are emulating; others are custom applications. In all cases careful consideration to device communications and understanding the PC Control file structure is required for a successful CMS installation. Typically a unique driver or a generic module that backs up the development environment files can be used.

- **Welders, Drives, and Misc. PC Control Applications** – These can be supported by a generic module that provides support for general PC applications and files.

- **Word, Excel, AutoCAD, and general Windows files** - These can be supported by a generic module that provides support for general PC applications and files or specialized drivers for these applications that address document management.
The selection of the best CMS application to meet a facility’s needs requires a careful assessment of the features provided by a CMS and the plant’s requirements. The features outlined earlier are expanded here to explore various approaches used by CMS users.

- **Backup Archive**: Many facilities will set retention parameters to maintain a representative number of prior versions of program files. These parameters will include the number of copies to maintain and a minimum age of a copy to be deleted. The age requirement is very useful when multiple unsuccessful attempts are made to correct a program issue and it is determined that reverting to an older copy of the program would be a better starting point than recent edited versions.

- **Change detection**: Optimal CMS benefit is achieved when users go through the CMS to make all program changes. This ensures a complete history of changes. When this is not achievable for various technical or political reasons, the CMS should have the ability to interrogate devices and compare the program running in the device to a reference copy in the CMS. If changes are detected, appropriate identification and notification should occur.

- **Change documentation**: As program editor software packages vary in their capability to identify changes, the CMS yields tremendous benefit by providing a consistent, intuitive tool to compare changes between any two versions of a program. This could be between a master copy, prior version or the current version in the processor.

- **Historical tracking**: It is of significant use when assessing areas for process improvement to be able to identify the frequency and type of changes to each device against its device type, production line and those individuals making the changes. When norms are identified it is straightforward to identify devices that are experiencing excessive changes and aid in identifying root causes.

- **Secured user and workstation access**: Each user should be authenticated by the CMS. In addition, some facilities have line-of-sight restrictions on which workstations can be used to edit certain device programs. This is often utilized when improper program changes could have safety impacts.

- **Controlled editor operations**: Users should be assigned to groups with permission profiles that map to the user’s authority within the plant. Careful consideration of these roles during implementation is encouraged. It is also advisable to keep the role structure as simple as possible.

- **Disaster recovery**: If the device hardware fails a replacement device will need to be obtained and connected to the network. A CMS user will then download the latest copy of the program to the device to resume operation.

- **Change notification**: Notification of change is an important benefit of the CMS, though users will often establish filters on the types of changes they wish to be alerted to.

- **Non-networked device management**: Disconnected devices can be supported by a checkout and check-in procedure, or if provided by the CMS application, non-networked tools can be used to download copies of programs from the central repository in order to provide the controls engineer access to the programs remotely. It is also helpful if the CMS can compare the program in the device with the one downloaded from the repository, capture user comments regarding program changes, assist in creating new programs in the field, and synchronize all changes back to repository.
The remainder of this White Paper is available by request to Michelle.Meyer@MDT-Software.com. We hope you find it to be of use in managing change and safeguarding your automation assets!