Software change control in the Sizewell B ISCO.

A Johnson. B.Sc, MIEE, C.Eng
Nuclear Electric Ltd, United Kingdom

Introduction

The main control and instrumentation system at the Sizewell B nuclear power station in the UK is referred to as the Integrated Systems for Centralised Operation (ISCO). Central to the ISCO is a control, data acquisition and display system based on a distributed network of several hundred microprocessor units located about the plant. This system, which is at the heart of the ISCO, was supplied by Westinghouse Electric Corporation and is referred to as the Westinghouse-ISCO or WISCO.

The WISCO comprises three functionally and technologically distinct systems which are each engineered to meet specific requirements; these are the High Integrity Control system (HICS), the Process Control System (PCS) and the Distributed Computer System (DCS). The three systems are shown diagrammatically in Figure 1 and, from this, the size of the system becomes apparent - some 20,000 I/O points and 200 network drops.

Although distinct in technology and function the three systems have been integrated into a single functional entity and must be maintained in this context. Software modifications affecting one of the systems often have a direct or indirect effect on other systems. Consequently, the change and configuration control practices which are applied to the system have to deal with these interactions.

Dealing with change

The initial generation of the software for the three systems was performed by Westinghouse working to a series of specifications provided by Nuclear Electric. Initial releases of software were, on the whole, major events with large packages of code being released and many drops being updated at once. Such an approach to change was acceptable at the time since the plant was still in the construction and commissioning phases and it was possible to deal with the disruption to the system caused by such large loads and the subsequent testing which was required.

However, it was recognised over the later years of development and the end stages of commissioning that such an approach would not be practical for modifications arising when the plant was near operation or throughout station life when the disruption caused by large software loads could not be tolerated.

In order to optimise and enhance the performance of the plant during the first cycle of operation it was necessary to introduce a series of software modifications arising out of

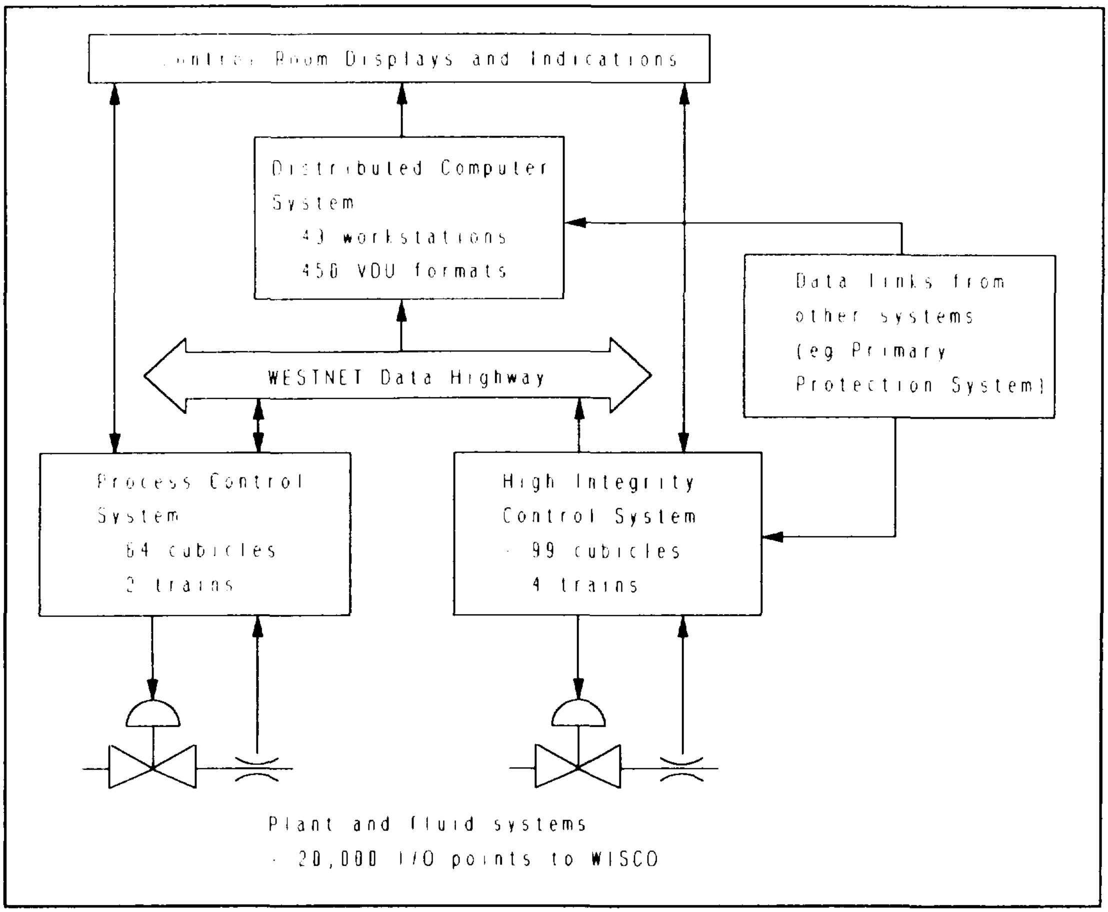
Control Room Displays and Indications
Distributed Computer System
43 workstations
450 VDU formats
Data links from other systems (eg Primary Protection System)
WESTNET Data Highway
Process Control System
64 cubicles
2 trains
High Integrity Control System
99 cubicles
4 trains
Plant and fluid systems
20,000 I/O points to WISCO

Figure 1 WISCO Principal Interfaces

findings late in commissioning or early operation. Such findings are typical to the early operation of any major plant emerging for several reasons such as minor performance discrepancies and improvements to the operator interface. To obtain the benefit from these changes at the earliest opportunity a strategy of introducing change incrementally throughout the first fuel cycle and beyond was required. Furthermore the requirements of the general plant modification procedure, the legislative principles of which are defined within the nuclear site licence, had to be incorporated into the software modification and configuration control procedures for WISCO.

To address these issues procedures were developed based on two principles. Firstly, progressively more use was made of the off-line Maintenance and Test System (MTS) as a platform for thoroughly testing changes before loading onto WISCO. Secondly, the method of specifying changes was reviewed and a new approach adopted which defined changes in smaller functional units which could be tested on the limited scope of the MTS and could be loaded with minimal disruption to the plant operation. Furthermore, the modifications had to be designed in such a way as to facilitate their incremental loading into the on-line system whilst preserving the overall integrity of the system and pedigree of the software. By addressing these considerations a method of software modification and configuration control was developed by evolving and standardising the best practices developed throughout commissioning. This facilitated a smooth transition to the new system.

However, a consequence of specifying smaller changes to be loaded incrementally was that the number of individual changes which needed to be tracked increased and the configuration control of the system was inevitably made more complex. The configuration control and software modification process which derived from the specification of more small changes and extensive off-line testing was initially applied to the HICS which has the most stringent requirements due to it's high integrity status.

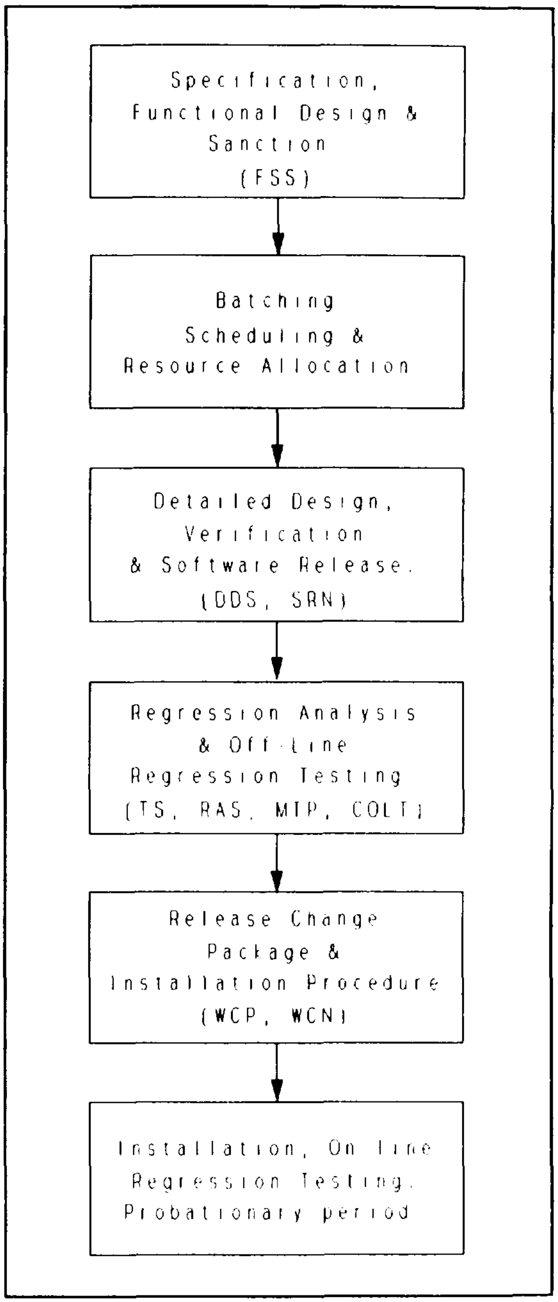
Specification, Functional Design & Sanction (FSS)
Batching Scheduling & Resource Allocation
Detailed Design, Verification & Software Release. (DDS, SRN)
Regression Analysis & Off-Line Regression Testing (TS, RAS, MTP, COLT)
Release Change Package & Installation Procedure (WCP, WCN)
Installation, On line Regression Testing. Probationary period

Figure 2 Software Modification Process

The HICS, a distributed microprocessor based system, is used for a variety of critical applications from the display of safety related actuations to the manual reinforcement of automatic safety actuations. The HICS also provides a number of closed loop automatic control essential to safety or generation. These roles call for high integrity software engineered to the highest standards.

Although the safety requirements of the PCS and DCS are not as stringent as those for HICS, quality software production and configuration control are still required if the plant is to perform reliably and efficiently. Therefore the extension of the principles developed for HICS to the PCS and DCS was prudent. Once established on HICS the same principles were applied to modifications to the DCS and PCS with a minimum of alteration caused by the differing technologies and software structure. The remainder of this paper discusses the change process using the HICS as an example.

Change control process outline

In outline, the modification strategy comprised the following key elements:

- Definition and sanction of modifications through the use of Functional Software Specifications.
- Initial analysis to determine optimal

"batching" and scheduling of modifications to meet the users functional needs, installation requirements and available resources.

- Amendment, documentation and verification of the software to implement the required functionality.

- Regression analysis and regression testing to prove that the change has been fully and correctly implemented.

- Production and implementation of field installation procedures designed to minimise disruption to normal plant operations.

These phases of the process, shown in Figure 2, are discussed in more detail below.

Specification and sanction

The need for software modification is often identified by engineering staff, responsible for the day to day operation of the plant, who experience at first hand minor design shortcomings and can best recommend improvements. The change is specified by means of a Functional Software Specification (FSS) which describes the change fully through the use of marked-up design documents and textual description. The FSS has a variety of users not all of whom would be as familiar with WISCO operation and structure as those writing or implementing the FSS. Typically, the FSS would be required to identify to plant fluid system engineers where change was being introduced and it would often form the basis for a variety of site committees to sanction the change on the basis of it's nuclear safety or other impacts. Given the breadth of potential users it was determined that, so far as possible, the FSS should be based upon a plain English description of the changes so as to ensure that all users could appreciate the scope of the change and not just those who were closely involved in it's specification or implementation. This contrasted with methods of specification earlier in the project which, although providing an efficient and rigorous means of transferring requirements from customer to contractor were, nevertheless, difficult to interpret by "outsiders".

To assist in achieving the above goals, the scope of the FSS is generally limited to a single, functionally related change which, once sanctioned, is released for implementation and tracking. Limiting the scope of the FSS in this way yields benefits in terms of keeping the FSS comprehensible and easy to understand and also provides a means of determining how best to introduce changes and optimise implementation strategies often several months after specification of the changes. However, with this flexibility comes the added load of tracking more small changes rather than fewer large changes.

Batching and scheduling

Once sanctioned by responsible engineers, the FSS is released for implementation to the WISCO software design team. At this point a review of the FSS is undertaken to ensure that all impacts to existing software files and documentation are completely recognised and understood. Omissions at this stage can result in costly rework and later in the process when the impacts are recognised. Only after the review is complete is the change scheduled for implementation.

The scheduling effort takes into account factors such as the urgency and convenience of installing the change in the operating plant and other FSSs which impact the same software modules and could be installed at the same time. In batching several changes together, consideration is given both to the economy of software engineering effort and ease of installation of the final release package. These often conflicting demands necessitate regular dialogue between designers and installers. The basic unit of batching changes throughout this dialogue is the FSS; only in exceptional circumstances are the FSSs split and only partially implemented within a particular software release.

One of the most important considerations of the batching/scheduling process is to identify those modifications which can only be installed during plant outages or those releases which introduce order dependency into the software load process. This is of particular importance since the production of such changes too early can result in software load dependencies which mean that the loading of many FSSs are held up simply because they all build upon an earlier one which has a particular difficulty associated with it's introduction into the WISCO. The early identification of such issues is essential so that one of a number of alternatives can be considered. These range from delaying the implementation of the FSS until closer to the anticipated point at which it can be loaded, revising the FSS so that it's load impact is less severe or undertaking additional analysis of the system impacts so as to be able to make the case for loading onto the online system.

Detailing engineering design

Once the scheduling is complete the implementation phase begins. Detailed Design Specifications (DDS) are prepared which explicitly identify the changes to be made to the files, documents, databases and the revisions which will be changed. The purpose of this stage is to provide a specification of the change in explicit detail as would be understood by the software designer. Attachments to the DDS are intended to identify precisely what needs to be changed in order to implement the FSS. Often, even small FSSs can impact several different skill areas or impact systems other than HICS. Where this is the case then separate DDSs are generated for each specific skill area. In this way each engineer is able to work in parallel on the aspects affecting their area of expertise - the whole effort being integrated by reference back to the single initiating FSS. In addition, the DDS documents second party reviews/verification (where appropriate) of the files as a means of demonstrating that the design has been implemented to the appropriate standards and provide and audit trail. The degree of independent verification of the design is one aspect where the HICS approach differs from that of the DCS or PCS. Because of the HICS is a Safety category 1 system certain changes must be subjected to Independent Verification whereas for DCS/PCS changes a second party "peer" review is generally adequate.

Generation of software releases

Following the individual modifications and reviews, a document is produced to carry the pertinent information related to the software release. This document, the Software Release Note (SRN) states what the release consists of and why it was produced in terms of the FSSs and DDSs from which it was produced. The SRN also provides a statement of the quality of the released package in that it records verification, the completion of the design testing and the promotion of the software as it passes various levels of testing and review necessary to ensure confidence in the software release. The revised software is assigned a status indicative of the degree of successful testing or operational use to which it has been exposed. The status levels extend from "U" (unclassified) through to "A"(fully tested and proven through operational use). Intermediate status also exist as indication below

- U - Unclassified, under development.
- D - Suitable for off-line testing
- C - Released for installation and on-line testing
- B - Probationary on-line period.
- A - Fully tested and proven through operational use.

These status levels are assigned and promoted by the independent librarian after verification of the proper qualifications. At each promotion between status the SRN may be amended to record the life history of the software.

Before release a "Configuration Analysis" is attached to the SRN to assist the installers in determining the impacts of the changes on the system. In particular, this analysis will identify whether the change is part of a larger scheme or whether other system changes need to be made prior to it's implementation.

Regression analysis

Once the software release has been formally generated by the independent librarian and recorded on the SRN a rigorous regression analysis is performed. This analysis determines, through formal processes, the impact of the changes made through the current software release as compared to the last released version used on the on-line system, and confirms that the changes found by analysis match the changes expected. Various tools are available to assist in the performance of the regression and subsequent generation of the Regression Analysis Summary (RAS). This summary details all changes which affect the present release and contains attachments in the form of difference reports, outputs of impact analysis tools etc, which serve to document the changes. By formally applying regression analysis techniques, each change to the software is justified in terms of the initiating FSS and confirmation is gained that only the desired set of changes have been incorporated into the current software release. In this way it is possible to limit the required retesting to those aspects of the software which have been changed - rather than embarking on a blanket retest of the total functionality of each software release.

Regression testing

Following completion and review of the RAS, independent test engineers examine the regression results along with the FSS documents to generate Test Specifications (TS) for testing the changes. The Test Specification aims to ensure that the change is completely tested by a combination of on-line and off-line tests. In general the aim is to maximise off-line testing so as to minimise the disturbance which would result if extensive on-line testing were performed.

Off-line testing is carried out on the Maintenance and Test System (MTS). This system consists examples of most WISCO drop types integrated into a test architecture similar to that of WISCO. In effect, the MTS is a reduced scope WISCO which can be readily reconfigured to a representative WISCO environment to provide testing of software to be loaded onto the full WISCO.

An important feature of the regression testing is that the software which is to be loaded at the plant is tested - rather than a modified version generated to fit the configuration of the test system. This means that the configuration of the MTS must be altered for the tests to emulate the relevant parts(s) of WISCO. The configuration required for the MTS is recorded using the Configuration of Off-Line Test bed (COLT) documents - each COLT relating back to an initiating FSS. The formal regression testing of the software release then occurs on the MTS, using software images exactly as will be loaded into the on-line system. Should any failures occur during the testing then revision and reissue of the incorrect item (software or test documentation) is usually required.

Release for installation

After the successful completion of formal off-line testing the software along with the complete set of design documentation are packaged with an appropriate WISCO Change Notice (WCN) to formal a WISCO Change Package (WCP). The WCP is effectively a complete installation package for the new release consisting of the software and associated design and quality release documentation. The WCN contains the instructions on how and where the new software should be loaded and provides engineers with appropriate check lists to record the software load.

Installation considerations

Although there have been opportunities to install some software changes whilst the plant was shut down the high load factor of the plant has meant that most have had to be installed whilst the plant is on-load. To achieve this installation procedures developed to draw maximum benefit from the inherent redundancy of the equipment design in facilitating software loads. For example, many HICS processing cubicles contain a duty/standby arrangement of redundant subsystems where it is possible to load software into the standby system, failover from duty to standby and then repeat the process in the other subsystem. Close working between the operations and design teams is essential in identifying suitable windows of opportunity to load software, in analysing the functional impact of the load and in achieving plant alignments most conducive to permitting the software load to take place. To this end an extensive and continuing effort has been undertaken within Nuclear Electric to analyse the WISCO design and so create a series of Functional Impact or Software Load Impact Statements which specify, for each cubicle in WISCO precisely the effects of taking that cubicle out of service or loading software whilst the cubicle is in service. These documents are invaluable to operations or maintenance staff in preparing for software loads to take place.

Conclusion

The process which has been described above has been in place for the HICS system since 1995 and a similar process for the PCS and DCS since 1996. During this time several hundred FSSs have been processed. The software modification and configuration control processes for all 3 subsystems are as similar as possible, the only deviations being due to the inherent differences in technology. HICS modifications, for example, always result in the reissue of the entire set of application EPROMS for a target subsystem. Similarly, PCS modifications always result in the release of the entire source code for each subsystem although the release medium is magnetic tape rather than EPROM. By contrast, DCS releases are usually of a partial set of application files for one or more of the drops on the DCS highway but, like the PCS, the release medium is magnetic tape. These differences, taken with others such as a greater use of automated tools and Independent Verification in the software production for HICS mean that there must inevitably be some differences in approach between the subsystems. However, these are kept to a minimum with the result the software modification and configuration control principles may be regarded as common.

The process described in this paper was developed in conjunction with the original supplier of the WISCO, Westinghouse who implemented the FSSs. Since then, Nuclear Electric have established an in-house software maintenance team and have taken responsibility for the whole software production and testing cycle.