

SMART Platform Independent Design & Application Programming Tool

Machupalli Sudharshan Reddy^{#1}, Chinthu Ramakoti Reddy^{#2}

#1 Team Lead, CYIENT Inc. USA, #2 Team Lead, CYIENT Inc. USA

Abstract - This paper is a proposal related to standardization of Distributed Control System Programming on common platform which minimizes man-hours in the initial phase of software implementation as well as in migration from one DCS platform to another.

Here are some details about the proposed tool. This tool should consist of standard algorithm libraries, along with capability to generate IO database with respect to various DCS vendor specifications. The output of this tool is expected to be compatible with all DCS software properties or attributes, so that any DCS vendor can minimize the engineering efforts required in understanding, designing and implementing complete control software required for process industries.

Adaption above standards provides the suppliers to focus on design of software and conversion of the same into platform independent interoperable software which in turn minimizes the initial design engineering hours and budget for the same. By standardization of software, the End User has "Freedom to Choose" best-in-class, interoperable software implementation techniques, and the "Power to Integrate" control systems, subsystems and devices across the plant enterprise.

It unifies concepts and proposes a common standardized programming tool which interfaces with different DCS vendors with a set of standard function blocks, macros in its own set of libraries and includes an easy way to apply new technologies.

Keywords— Distributed Control System, Standardization, EPC Designer, Consultant, First Time Engineering, Original Equipment Manufacturers (OEMs), End Users.

I. INTRODUCTION

Control system plays crucial part in plant automation to improve efficiency. It communicates with the field parameters to take critical plant operation decisions. This control enables the operators to know the state of the plant and process. Like the body's nervous system, a plant's control system is complex, particularly if it has grown over the years and has changed significantly from its original design.

Currently in all process industries, Software Development/Implementation phase consumes most of the *First Time Engineering* efforts which have significant impact on Project budget & schedule.

The Engineering Procurement Construction (EPC) Designer is well aware of the basic Engineering design concept. At present different EPC designers develop control logic sheets as per the process requirements by using a set of

design inputs received from either consultant or End user. Further the implemented control logic is giving to the DCS vendors as one of the inputs for implementing the same in DCS software. The DCS vendor spends several man-hours to convert the process logic sheets to their respective platform compatible which in turn leads to increase budget of the project.

Consider the situation, if End user wants to go for migration, they will face a lot of dependency on existing DCS vendor software. Because they need to give all the reference documents, control narratives (design inputs) and specifications of existing software. The challenges here are we need to check the compatibility of existing scenario to the upgrading scenario as well as the new DCS vendor has to spend several thousands of hours in understanding and analyzing the existing software, process as well as the design inputs used for implementing the software at that particular time.

In order to optimize the above cases, our proposal is to standardize the software implementation platform in one phase at EPC side, which minimizes the First Time Engineering efforts required in software implementation.

Distributed Control System:

A Distributed Control System (DCS) refers to a control system usually of a manufacturing system, process or any kind of dynamic system, in which the controller elements are not central in location (like the brain) but are distributed throughout the system with each component sub-system controlled by one or more controllers. The entire system of controllers is connected by networks for communication and monitoring.

Distributed Control Systems (DCS)^[1] are dedicated systems used to control manufacturing processes that are continuous or batch-oriented, such as oil refining, petrochemicals, central station power generation, fertilizers, pharmaceuticals, food and beverage manufacturing, cement production, steelmaking, and papermaking. The lists of some of the DCS vendors are ABB, ALSTOM, GE, YOKOGAWA, EMERSON, HONEYWELL, and AZBIL.

In the present challenging scenario, it of importance for process manufacturers to choose a right DCS for their specific application. A successful evaluation should start by developing a clear picture of the requirements of the application and the needs of engineering, maintenance, and operations personnel. Selecting the right technology and the right supplier can help companies. The following need attention:

- Quick responses to changing market conditions and have an edge over the other competitors.
- Focus on Minimize Total Cost of Ownership (TCO) over the life of your plant.
- Establish a system which is easy to maintain/upgrade for the long-term.
- Work towards its future goals and vision.

II. ROLE of KEY PLAYERS in the PROJECT LIFE CYCLE of PROCESS INDUSTRIES

The coordination of key players involved in designing any new control and instrumentation projects is as depicted in Figure 1.

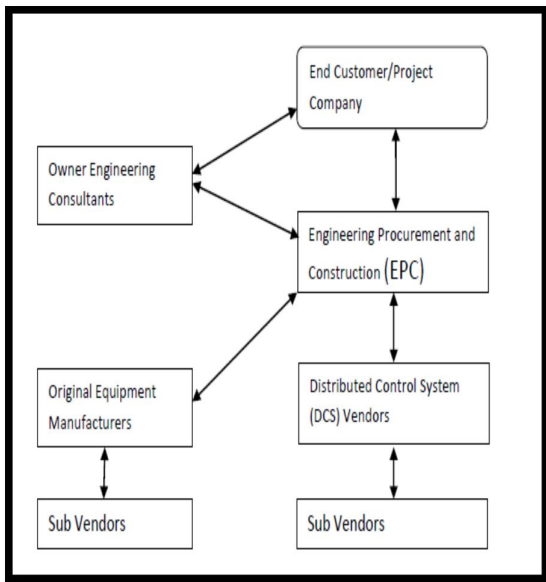


Fig.1 Present Coordination between key players

End Users/Utility:

End users are the owners of the process plant and are mainly responsible for operation, maintenance and develop process plant projects. The owners have the benefit of plant operation when optimum and efficient throughput is produced with optimization activities. They are to be in

constant concern with their EPC effective utilization of the resources.

Original Equipment Manufacturer (OEM):

The OEM brings to the project team not only equipment and systems specifications, but also an entrepreneurial business approach to participate in the sharing of project risk.

Engineering Consultants^[3]:

Consulting engineering is a professional service that provides independent expertise in engineering, for industries, developers and construction firms. Consulting engineering companies can range in size from a partnership of two people to multinational corporations with tens of thousands of employees in offices world-wide.

Role of DCS Vendor in Project Services:

- Full process plant DCS functional design, configuration, testing, commissioning.
- Hardware supply and installation
- Documentation and training
- Verification and validation
- Database development and management
- Interface and cabling design
- Design / Commission management
- Automation level improvements
- Advanced controls implementation
- Loop tuning and performance optimization
- Alarm analysis and reduction

Engineering, Procurement and Construction (EPC):

Engineering, Procurement and Construction (EPC) contract means much more than just putting together three different sources of engineering, procurement and construction for project execution. Combination of engineering, procurement, operation, management, administration, on-time delivery, cost control and risk management is done in project management of EPC contracts, with Planning, controlling and simultaneous activities acceleration regarding project scope of quality.

III. PRESENT CHALLENGES in FIRST TIME ENGINEERING (FTE) of PROCESS INDUSTRIES

Currently in all process industries, Software Development/Implementation phase consumes most of the

First Time Engineering effort in man-hours which have significant impact on Project budget and schedule.

As a first step in the I&C Design, DCS vendor has to get some specific inputs in order to start the software implementation from EPC Designers. The inputs are the *Process Logic sheets, Interface Specifications, System Description documents, Instrument Loop Diagrams, P&ID etc...*

Case I: The Engineering Procurement Construction (EPC) Designer is well aware of the basic engineering design concept. The EPC designer spends time for First Time Engineering efforts to prepare process logic sheets such as SAMA, Control narratives as per the functional requirements or specifications w.r.t the process industry or end user. In the present scenario, the EPC will charge thousands of hours in creating the design inputs and the same is given to the DCS vendors. After receiving the design inputs from EPC, the challenge for DCS vendor is to understand and analyze the design inputs, wherein he has to spend thousands of hours. After analyzing the design inputs, he should check the compatible function block in the DCS library w.r.t the Design input documents for special function blocks like PID, MA Station, Transfer blocks etc.. Then he starts implementing the design inputs in the specific platform. If the above process is considered, it takes lot of man-hours to understand, analyze and implement the design in the software which in turn increases the overall budget of the project. Finally the end user pays a huge amount to EPC and DCS vendors leading to increase in budget.

Case II: Consider the case of big End users like NTPC, NPCIL, Mittal Group, JSW who have many plants operating parallel in different locations. The major challenge faced by such big groups when they build multiple plants with different DCS technologies is the time taken for software implementation. Due to dependency on EPC and DCS vendors who take their own time in executing the project of End-user as per the requirements and specifications. This increases the project life cycle and budget.

Consider the situation if end user wants to go for migration they will face a lot of dependency on specific existing DCS. Because they need to give all the reference documents (design inputs) and specifications of existing software. The challenges here are we need to check the compatibility of existing scenario to the upgrading scenario. The new DCS vendor has to spend thousands of hours in understanding and analyzing the existing software, process as well as the

design inputs used for implementing the software at that particular time. Once you've made the decision to upgrade your control system, the first step is to pick a migration partner and begin the planning and design process. A migration partner could be a *DCS manufacturer*, the manufacturer of the system being upgraded, or system integrator / automation partner. In the present scenario, the end user is totally dependent on the DCS vendor for the software design, implementation. Moreover he is not allowed in going for other DCS platform. Finally the End user is investing lot of time and budgets to accomplish the requirements and specifications. In this case the major challenges we face are:

- Compatibility of one DCS with other.
- The Macros and Elementary Function Blocks (EFB) suitability w.r.t pre-existing algorithm libraries of various DCS vendors.
- Man-Hours in execution of the Project.
- Complexity involved in migration of DCS.
- Adapt to the existing system and software implementation.

At present the EPC/Consultant provides an I/O list which consists of inputs, outputs related to the instrument which gives the status/ value of the process parameters. With this I/O List the DCS vendor will compare the list with his own standard I/O list of specific projects which in turn gives the required amount of signals for the process. By this he can remove the unnecessary signals. For this he has to take the confirmation from EPC/Consultant to move forward. Consider an example, if there is any signal missed/wrong naming philosophy in the database which was freezed, this leads to errors while importing the database into DCS. To debug these errors, we need to analyse the root cause and fix the same in database configuration. As of now this process consumes lot of time and conversation between DCS vendor and EPC/Consultant which increases the cycle time of the project as well as budget also.

Why do we need a Standard Platform Independent Software?

When researching standardization, the advantages consistently cited are:

- Better communication
- Reduced training time and costs
- Lower support and maintenance costs
- Easier budgeting and cost management

IV. PROBABLE and PROPOSED SOLUTION

A common platform for Software implementation which consists of standard algorithm libraries w.r.t various functionalities, able to generate IO database w.r.t various DCS vendor specifications. Prior to this, we need to standardize the function blocks which are frequently used in all DCS library, so that it will be easy during conversion into specific DCS platform. It should be compatible with all DCS software properties, i.e. the tool output should be in readable format of any DCS. With this DCS vendor can reduce the engineering efforts required in understanding, designing and implementation of complete control software required for process industries. We need to standardize the tag name philosophy across the process industries, so that the naming convention will also be the same across different DCS vendors, in which we can reduce the efforts in understanding the logic w.r.t tag names.

The EPC designer implements the software directly on the platform independent tool with reference to process logic diagrams or control narratives and the output of this should be compatible with all DCS software properties. With this approach it will be an added advantage for EPC Designer working for multiple plants to give the software package directly to different end users as per their specification of plants. After the software implementation, the tool output should be given to DCS vendor in order to test and validate the software implementation whether it is as per the user requirements and specifications. In case of any ambiguity in the software, the DCS vendor has to correct and ensure its functionality with the help of his own DCS platform. If we follow this approach, we can reduce the chances of errors in software, which in turn reduce the efforts required in rework for correcting the software.

Case I: In order to avoid the limitation explained in relation to present challenges scenario, we need to implement the software on platform independent tool by using different sets of design inputs. The tool should be implemented in a compatible manner with any DCS vendor software specifications. The EPC can use his resources in developing the software on platform independent tool. After the software implementation, the output which is compatible with DCS vendors can be given as input to the DCS vendor to incorporate the software into their specific DCS platform.

Case II: In the content of migrating, the EPC can give directly the standard tool output whenever he wants to migrate from his existing DCS. By following this approach,

dependency on DCS vendor will come down which in turn drastically decreases the budget and time effort required for the migration.

Example: Platform Independent Tool – The SAMA logic will be implemented here in which the output is .XXX format. The tool should generate the compatible file w.r.t all DCS vendors. The tool should have its own set of libraries for all Functional Block Diagrams (FBD's), so that it is compatible with other DCS Vendor libraries. A Standard also needs to be created for this standard tool.

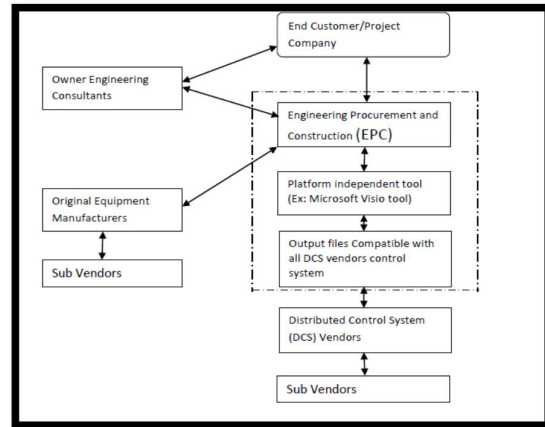


Fig.2 Proposed Coordination between Key Players

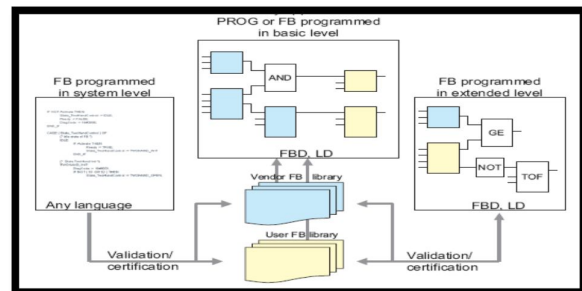


Fig.3 Platform Independent Tool Process

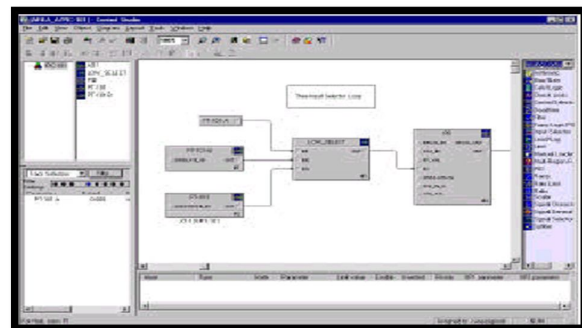


Fig.4 DCS Vendor Specific Platform

Features to be considered before Standardization:

On taking a decision to opt standardization the following need to be considered -

- *Areas to be standardized* - Examine the requirements and standardize where it impacts the most leading to benefits.
- *Users’ needs*–Conduct a detailed usability report of the software, Also list down the anticipated future changes required.
- *Work plan*- Create a timeline for implementation and rollout of the selected changes.
- *Resource Plan*-Budget for the proposed tool and training for resources.

Drawbacks of Standardization

Standardization can have its drawbacks, and it’s worthwhile mentioning a couple here. When you tie yourself to one vendor, you could end up stuck with that vendor and over time, lose your bargaining power because they’re aware you have limited options.

Benefits of Standardizing Your Software (Platform Independent tool)

Standardizing on one vendor for specific software provides organizations with several advantages:

- Project life cycle time reduction.
- Budget savings
- Dependency on DCS vendor comes down.
- Minimizing the logical & human errors in software implementation.
- Faster resolution in correcting the errors during software implementation.
- Easy migration within short span of time.
- Multiple projects can be easily and more quickly implemented by using a common programming tool.
- Avoids a scope battle with vendor

The approximate detailed engineering hours required by different key players in project are as shown below for present as well as proposed probable solution:

TABLE I
Engineering Hours required for Software implementation in present scenario (Single Project)

Present Scenario (Single Project)				
DCS				
Consultants	EPC	Study of Design Inputs	S/W Implementation	Testing & Validation
6000	25000	30000	40000	9000
Total				110000

TABLE II
Engineering Hours required for Software implementation in proposed scenario (Single Project)

Proposed Scenario (Single Project)				
EPC		DCS		
Consultants	Design Inputs	S/W Implementation	Importing, Customization & Testing	Total
6000	25000	30000	10000	71000

TABLE III
Engineering Hours required for Software implementation in present scenario - Multiple projects

Present Scenario (Executing Multiple Projects ~10)				
DCS				
Consultants	EPC	Study of Design Inputs	S/W Implementation	Testing & Validation
60000	250000	300000	400000	90000
Total				1100000

TABLE IV
Engineering Hours required for Software implementation in proposed scenario - Multiple projects

Proposed Scenario (Executing Multiple Projects ~10)				
EPC		DCS		
Consultants	Design Inputs	S/W Implementation	Importing, Customization & Testing	Total
60000	250000	200000	100000	610000

Note: The above values are considered based on some assumptions/approximations and are not precise values.

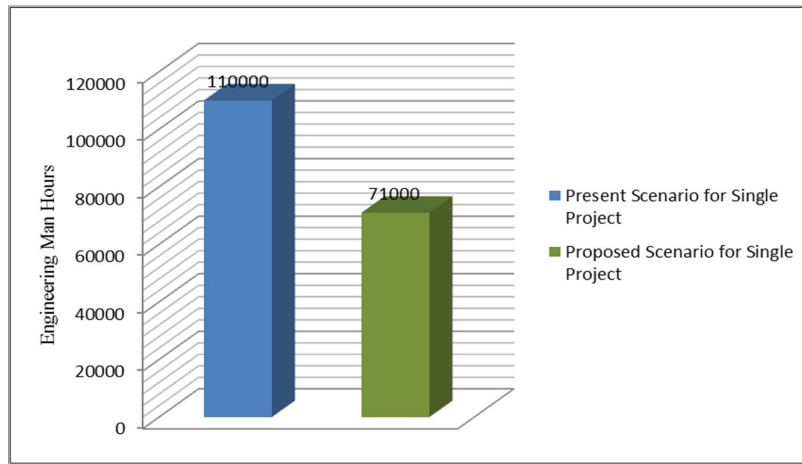


Fig. 5 Present Vs. Proposed Engineering Efforts for Software implementation (Single Project)

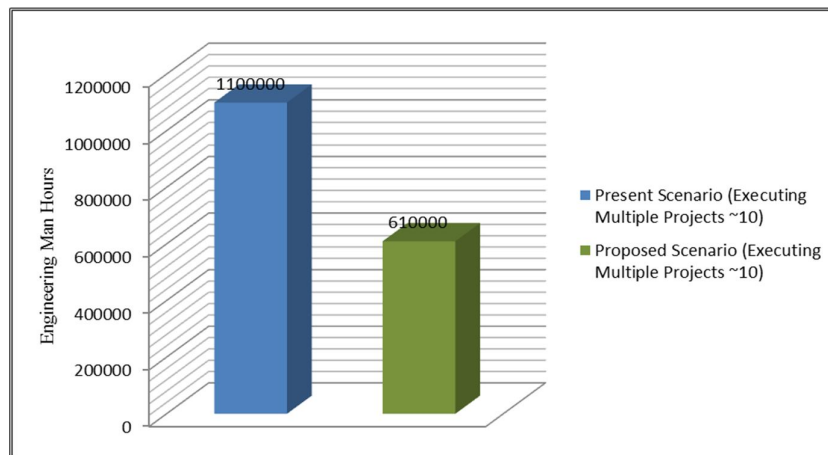


Fig.6 Present Vs. Proposed Engineering Efforts for Software implementation (Multiple Projects ~10)

V. CONCLUSION

By following the above proposed probable solution, i.e. by standardizing the software implementation platform, organizations can take advantage of tremendous cost savings in addition to increase in efficiency and productivity

REFERENCES

1. http://en.wikipedia.org/wiki/Distributed_control_system
2. <http://www.epcengineer.com/definition/132/epc-engineering-procurement-construction>
3. <http://www.engineeringlegacies.com/WhatIs.php>

Definitions/Abbreviations

- DCS – Distributed Control System
- EPC – Engineering Procurement Construction
- OEM – Original Equipment Manufacturer
- FBD – Functional Block Diagrams
- FTE – First Time Engineering
- TCO - Total Cost of Ownership
- SAMA – Scientific Apparatus Makers Association
- I/O – Input and Output
- P&ID – Process and Instrument Diagrams



Machupalli Sudharshan Reddy received his B.Tech degree in **Electronics & Instrumentation Engineering** from JNTU, Hyderabad in 2007 and completed **M.Tech** in **Digital Systems and Computer Electronics** from JNTU, Anantapur. Currently, working in **CYIENT Inc.** as Team Leader in Automation and Field Services for Westinghouse AP1000 Nuclear Project.



Chinthu Ramakoti Reddy received his B.Tech degree in **Electronics & Instrumentation Engineering** from SV University, Tirupathi in 2006. Currently, working in **CYIENT Inc.** as Team Leader in Automation and Field Services for Westinghouse AP1000 Nuclear Project.