A Control system design approach for continuous mixer and extruder

Raghu Kodi*, Vijaykumar Nidagalkar

Central Manufacturing Technology Institute, Bengaluru, Karnataka, India

KEYWORDS

Continuous Mixer and Extruder,
Control Systems,
Development Life Cycle,
Process Automation,
DCS,
PLC,
Frameworks.

ABSTRACT

A holistic design and development approach is proposed for process machine Control System of a Continuous Mixer and Extruder (CME). In this approach five frameworks are discussed to improve and standardize development cycle and machine life cycle.

Unlike traditional batch mixers, continuous mixer provides an economical alternation for big capacity mixing processes. CMTI has designed and developed a CME for a chemical processing research laboratory. The CME developed by CMTI is a fully automated system with all control operations automated for all process automation i.e., from silo to bin.

A top-down design approach is followed for control systems design. As recommended by ISA-88 norm, abstraction of equipment units allowed compartmentalization of operations. Further, abstraction enabled mapping operational requirements across life cycle from development to maintenance. The design is carried out considering all possible operations required throughout machine life cycle. This approach allowed for accelerated development, isolated unit testing, seamless integration and system testing.

A holistic approach in design and development of control system enables machine to be operated at variety of phases of development life cycle and machine life cycle. This is achieved with five major frameworks incorporated into the system.

The control system designed and developed consists both hardware and software. Software is developed into multiple Programmable Logic Controllers (PLCs). A network of multiple PLC systems and a central PLC that coordinates all control operations. This architecture forms a Distributed control system (DCS).

1. Introduction

The CME is a twin screw mixer and extruder (TSME). The machine is developed for mixing various solid & liquid ingredients and finally extrude the extrudate (end product) through a die. It consists of multiple sub-systems like refilling systems, loading hoppers, solid feeders, liquid feeders, hydraulic power unit, lubrication unit, vacuum pump, screw driving motor, extrusion cutter, conveying and collecting system. Figure 1 shows systems and sub-systems arrangement of CME developed by CMTI.

The control system discussed in this paper, is designed and developed for the said CME. The control system for the CME is designed and developed with no prior suitable in-house expertise.

The CME requires running process steps parallelly unlike a batch mixer. The steps starting from material conveying, weighing, feeding, mixing, extrusion, cutting, conveying and collection are to be executed simultaneously. For operation life cycle of CME, the system shall have provisions such that all of the steps operated are either independently controlled or interlock controlled, depending on nature of operation like running the process or setting up the process.

The control logic/procedures that need to be executed are programmed into both central PLC as well as respective sub-system controllers. Control logic for process automation of the machine is programmed into 7 different controllers, some are industry standard PLCs and some are vendor specific controllers. This architecture

*Corresponding author E-mail: raghu@cmti.res.in

https://doi.org/10.58368/MTT.23.9-10.2024.20-27

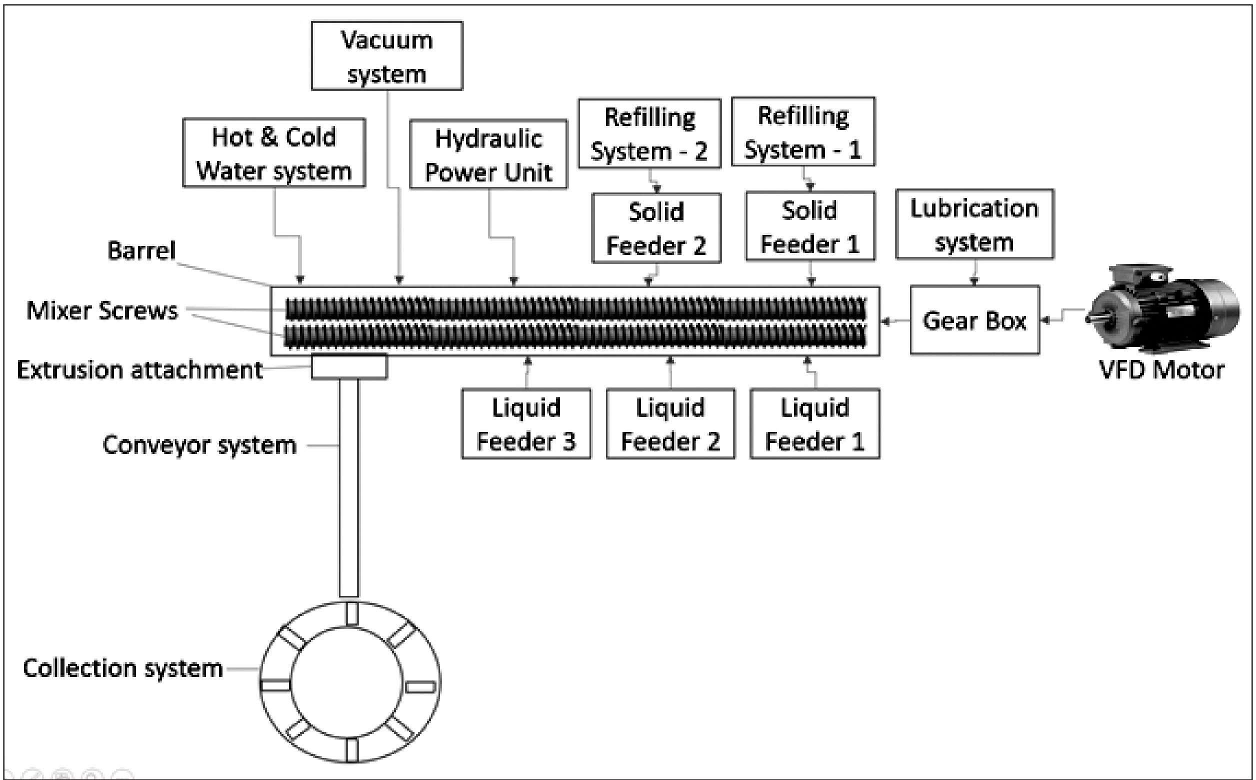

Fig. 1. CME machine and sub-systems.

forms a Distributed control system (DCS). The co-ordination of control is taken care by a central PLC.

The machine and control system for CME are designed to facilitate a chemical research lab with two aspects that are functionally at two different ends of requirements spectrum viz, 1)Experimentation platform for development of various recipes and recipe optimization 2) Fully automated machine with minimal operator interference for production.

2. Need and Scope

The requirements with respect to machine and process are specified by the customer. However, strategies with respect to development life cycle are defined by design team in consultation with customer. Strategies for development are broadly adapted by design team, which are similar to any software development with goals such as modularity, maintainability, reusability and scalability. The development strategy is discussed in Siemens AG (2018), that recommends analysing machine structure, designing interfaces between elements of machine and implementation.

The machine has variety of operational requirements like ease of maintenance, quick process setup, automated process execution, multiple user interfaces for independent operation and recipe access restrictions etc.

The CME machine needs to be designed with features like ease of updating machine safety limits, ease of updating process safety limits, ease of recipe development etc., which reduce modification overhead in case of any sub-system/device/ingredient replacement.

Apart from above requirements, due to first-time machine development following aspects are to be considered for machine realization to achieve synchronized cohesive progress. They are ease of integration/replacement of sub-system and ease of changing sub-system configuration settings.

Frameworks driven holistic approach in design and development of control system is considered to meet complexity in requirements and address uncertainties in development. Frameworks discussed in this paper enable machine control logic to be developed to meet above discussed requirements. Different sections of control logic developed are such that, they can be modified at various phases of machine life cycle. This is achieved with five major frameworks to be incorporated into the system that are

a) Safety and Emergency b) Finite State Machine at the highest level of control logic c) Modes of operation and d) User authorization e) abstraction of equipment.

After adopting the said frameworks development time has reduced significantly. Development phases are documented and are guided by frameworks. Semantics of equipment and devices are dictated by frameworks and shared to every department involved in this prototype machine development process. The frameworks and their scope are shared with various teams like conceptual design, detail design, development, assembly and testing teams. This allowed respective teams to be prepared with their scope of contribution, and be ready with necessary interfaces for implementation.

3. Design and Development of Frameworks

The CME machine consists of multiple sub-systems like vacuum conveyors, loading hoppers, solid feeders, liquid feeders, hydraulic power unit, vacuum pump, screw driving motor, hot & cold-water circulation system, extrusion cutter, conveying and collecting system. During operation, all sub-systems have to communicate with central PLC in real-time for synchronous operation. Each sub-system can operate independently and has its own controller and stand-alone user interface. Apart from stand-alone user interface, CME machine is provided with centralized user interface with necessary SCADA functionalities in an Industrial PC.

To address the previously mentioned life cycle requirements, the control system includes five major frameworks that are

a). Safety and Emergency: hardwired relay logic.

b). Finite State Machine (FSM); designed to notify every sub-system in the machine present state of operation and control. This is a governing control logic for the entire DCS. The logic of overall machine FSM is executed in central PLC.

c). Modes of operation; that will enable necessary control options that otherwise are interlocked or disabled.

d). User authorization; three levels of operational authorizations are provided to demark operational access.

e). Abstraction; of equipment as per ISA-88 (De Minicis et al., 2014; Peter 2024).

a). Safety and emergency

As mentioned earlier the machine has multiple sub-systems with their own control panels, controllers, actuators, sensors and user interfaces for independent operation. Hardwired relay logic is developed to identify and communicate emergency cases across panels. In this relay logic, all cabinets are connected to Master Control Relay (MCR), that can be triggered by any sensor or control panel in emergency relay logic chain. MCR in turn will signal sub-systems that there is an emergency.

The relay logic removes dependency on controllers availability and status of communication systems health. Part of relay logic is that, every sub-system control panel and user interface is provided with emergency push buttons. These panels and user interface stations are positioned across installation site for quick access.

Failure of any sub-system controller or smart device associated with sub-system or central PLC, is identified by central PLC by monitoring communication connection health and respective devices heart beat (Wikipedia contributors, 2024; What Is the Function of a PLC Watchdog Timer?, 2023).

b). Finite State Machine (FSM)

FSM is a control logic designed to make sure that the system execution does not run into any indeterministic state at any given point of time.

FSM developed for this machine is a management control logic to handle and navigate emergency situations, machine boot up, equipment level control logic, machine/sub-system readiness, operator interface, process setup and process execution.

Figure 2 depicts a top-level state machine that represents CME machine state transitions and conditions. As there are multiple controllers in this DCS, machine state determined by the central PLC is communicated to sub-system controllers, so that sub-system controllers take necessary action.

Once central PLC boots, machine enters emergency state, user has to manually turn master key for machine to exit emergency state and enter initialization state. Following, in initialization state readiness of the sub-system

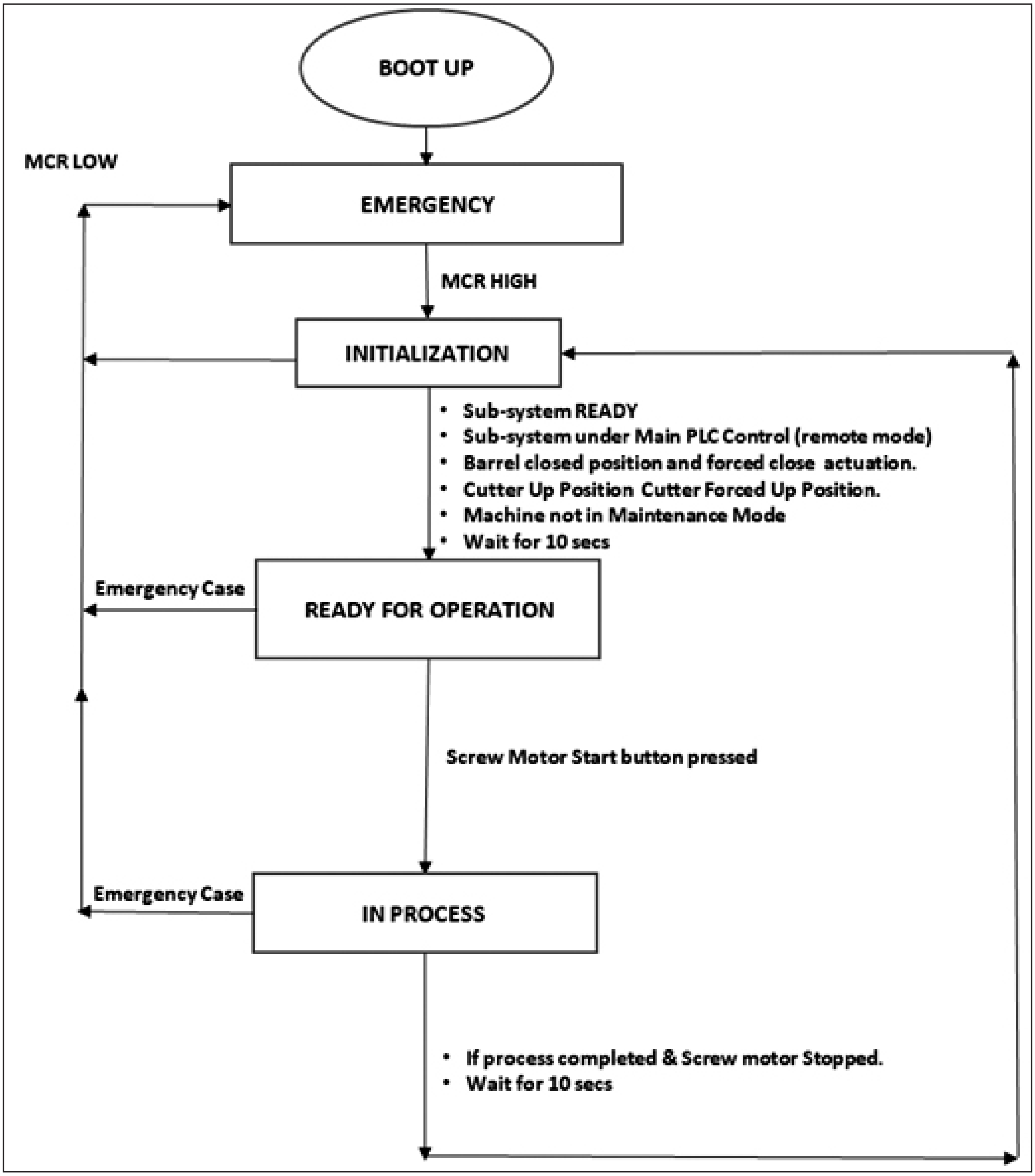

Fig. 2. FSM of CME.

is learned by the FSM logic and other process essential interlocks listed as shown in figure 2. If these conditions are met, machine enters into ready for operation state, where the user is allowed to start process.

In fully automatic mode of operation if the user starts the process by starting screw motor, other feeders are also started automatically with set recipe feed rates, recipe process timings and sequence according to the selected library recipe. Once all bins are filled with products in collection system, it is identified as process completion. In case of process completion machine notifies the user and goes to initialization state.

CME Machine state can enter to Emergency state from any state, this could be caused by trigger of a) sensors positioned in machine to identify emergency state b) sub-systems and their sensors identifying emergency case c) central PLC identifying any process value out of process limits or machine limits.

c). Modes of operation

In this CME machine automation, three modes of operations exist, as discussed below

- Fully automatic/recipe mode: In this mode of operation user selects a recipe from library and everything is taken care by central PLC control logic. User is enabled with minimal interface to start and stop the process, whereas, set points and process timings are defined in recipe. User cannot edit set points and process timings. In this mode control logic takes care of start, ramp, stop sequences of all feeders automatically as defined by library recipe. This mode can be selected by all users of authorization. The mode restricts individual operation of feeders. When this mode is selected at central user interface SCADA, operation is possible from the same central SCADA system, all sub-system user interfaces are disabled except emergency push button.
- Individual unit operation mode: Individual unit can be operated w/o considering status interlocks from other units, this allows for independent operation of sub-systems either from main user interface or sub-systems user interface. In this mode necessary process & machine limits are considered as interlocks. This mode can be selected by users of authorization of either supervisor level or service engineer level.
- Maintenance mode: In this mode individual actuators can be accessed; still necessary machine limits interlocks are considered. In this mode service engineer is enabled to access additional screens that allows him to edit main controller or sub-system configuration settings. Machine never enters 'In process state' or 'ready for operation' if it is set in maintenance mode. This mode can be selected only by a user of authorization of level of service engineer.

d). User authorization

The machine is provided with 3 levels of user authorization levels– operator, supervisor and service engineer

- Operator can only operate the machine at fully automatic mode, where he can run an existing library recipe for production. Operator is restricted to access any recipe development fields on SCADA system.
- Supervisor can operate machine in both fully automatic mode or individual unit mode to create recipes, set process limits and operate units individually for maintenance and calibration. Supervisor will be responsible for recipe creation, sequence feeders start & stop and process timings.

  A validated and proven recipe can be added to recipe library by the supervisor, for production by an operator.
- Service Engineer can operate machine in all modes, he is also provided rights to change machine limits. A service engineer is enabled to access additional screens that allows him to edit main controller or sub-system configuration settings.

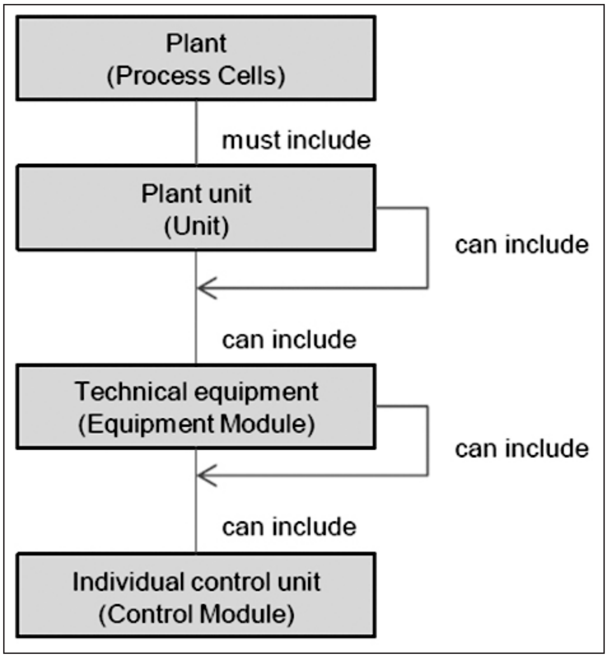

Fig. 3. ISA-88 physical model.

e). Abstraction (De Miniciset al., 2014; Peter, 2024)

The control logic handles various sub-systems, by following abstraction as per recommendations made in ISA-88 norm physical model as shown in the figure 3.

Every sub-system is dealt as a plant unit. For example solid feeder system is handled as a plant unit.

Solid feeder system has two feeders, each with their own control units. Each feeder is dealt as a technical equipment. So, one control logic is developed for one type of technical equipment, and is reused for other.

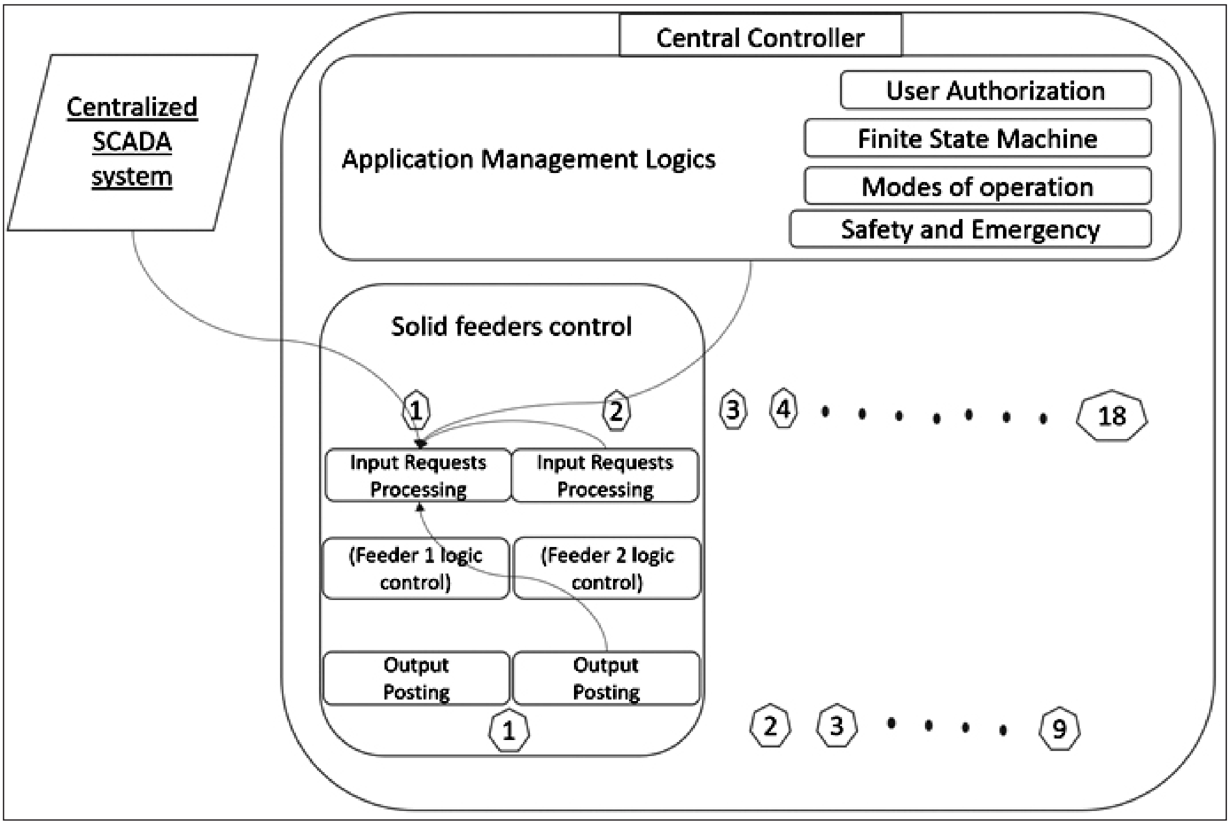

Fig. 4. Control logic functions hierarchy.

Liquid feeder system has 3 feeders, dealt as 3 technical equipment, but the control logic for them is developed once and reused thrice.

There are 9 plant units and 18 technical equipment in the CME.

Interlocks verification that needs to be carried out specific to individual technical equipment are handled in a separate function ('Feeder 1 logic control' as shown in figure 4), allowing its reusability. Interlocks verification to be carried out across technical equipment due to their dependency on readiness or operation status of other technical equipment are handled in separate function ('Input requests processing' as shown in figure 4).

The figure 4 provides an overview of control logic function hierarchy designed for control of a solid feeder (individual technical equipment) separated for modularity and reusability.

From a solid feeder operation point of view, its control logic can receive start, stop and set rate requests either from a) Central PLC under Individual unit operation mode or from b) Central PLC under fully automatic operation mode. In case of individual unit operation, a set rate, start/stop requests are provided in designated fields from centralized SCADA system. In case of fully automatic operation mode, these requests are from library recipe.

A feeder specific run request is processed based on a) application management logic functions b) Field entry in central PLC and c) state of other feeder/sub-system. This processing/verification is carried out at 'Input Requests processing' function of individual feeder logic (technical equipment level).

4. Results and Discussion

CME has various sub-systems which needs to be integrated into a DCS. During development life cycle of the control system of CME, every sub-system is available for integration to central system (PLC) at different point of times. Framework approach reduced dependency on availability of sub-systems. This is achieved due to well defined interfaces between application management logic and individual sub-system control logics. Interfaces definition here makes the automation solution modular, replaceable and scalable.

Any sub-system additions envisaged at a later point of time of development can be accommodated into the system, with minimum affect to existing automation control logic.

Due to modeling of sub-systems as independent units, parallel control logic development by multiple engineers was possible. This resulted in reduction of development time significantly.

CME as mentioned earlier has multiple sub-systems. As and when sub-systems were available for integration, its respective central PLC control logic was developed for it. This development included realization of communication interface between the sub-system controller and central PLC.

FSM provides necessary diagnosis and trouble-shooting means, by which one can trace machine states transition history. During assembly of sub-systems to main system, any unwanted event is traced and triggers for the same can be identified. This provides crucial feedback for first time right development of CME machine and its automation.

One of major improvements experienced during the development process is integration of third-party systems (sub systems). Integration of a sub system like precision loss in weight feeder with its own controller, hopper and refill system could take 4 weeks. The integration includes electrical system, sensors & instrumentation connectivity, communication network set up, I/O data exchange set up etc. Frameworks suggested here allowed the team members to work independently and complete integration in less than 7 days. Well defined interface points by frameworks resulted in less integration time.

5. Conclusion

Frameworks discussed above greatly accelerated development and brought reliability to the development. These frameworks are made to be universal thanks to the abstraction that is masking underlying equipment unit function -alities. This allows them to be practiced across complicated process machinery control system.

Implementation of the recommendations allows to create rich library of equipment module control logics, signal processing logics that can be reused across projects. This accelerates development time as the adapted library module is already tested and qualified.

As the recommendations consider Control systems development life cycle, solution development time will be reduced, ready library/templates allow carrying out developments parallelly with multiple resources. As the library is already proven; adopting it to new developments deliver reliability and consistency.

When developing a novel control system solution, detailed specifications are not spelled out upfront, rather are identified during development process. Frameworks discussed, help in navigating last minute development of control requirements.

The framework covering variety of life cycle aspects for a complicated machine, like CME can be reused for any new prototype machine. As the engineers get familiar with components of the framework, this results in reduced development time, increased reliability and produce repeatable results in new prototype machine automation development operations.

Acknowledgment

The Author would like to thank Mr. B. R. Mohan Raj, Joint Director, CMTI, for providing the opportunity for development of automation solution for variety of special purpose machines. The Author would also like to thank mechanical counter parts in this development Mr. Ananthapadmanabha K. M., Scientist-E, Project Coordinator, CME.

References

De Minicis, M., Giordano, F., Poli, F., & Schiraldi, M. M. (2014). Recipe development process re-design with ANSI/ISA-88 batch control standard in the pharmaceutical industry. *International Journal of Engineering Business Management, 6,* 1-12. doi.10.5772/59025

Peter. (2024, March 19). *ISA-88 (S88) Batch Control explained | PLC Academy.* PLC Academy. https://www.plcacademy.com/isa-88-s88-batch-control-explained/

Siemens AG (2018). *Guide to Standardization,* TIA portal. https://support.industry.siemens.com/cs/document/109756737/guide-to-standardization?dti=0&lc=en-IN

What is the function of a PLC watchdog timer? (2023b, January 23). Industrial Automation, PLC Programming, Scada & Pid Control System. https://forumautomation.com/t/what-is-the-function-of-a-plc-watchdog-timer/10854

Wikipedia contributors. (2024, October 16). Heartbeat (computing). Wikipedia. https://en.wikipedia.org/wiki/Heartbeat_(computing)

Raghu Kodi is a Scientist specializing in the design and development of Control Systems for Special Purpose Machines. He has a degree of Master of Engineering in Electronics and Communication Engineering with specialization of Embedded systems & VLSI design. His skills encompass design of electronic & electrical systems and systems integration for industrial machine control and automation. Over the past 12 years, he has contributed to design and development various technologies, like Machine Tool Thermal Error Compensation using embedded systems, IGBT based AC motor drive, Control systems for twin-screw mixer extruder, Control systems for 10-ton Vertical Mixer, control software for shuttle less rapier loom, etc.

Vijaykumar Nidagalkar is a Scientist specializing in the design and development of Special Purpose Machines. His skills encompass designing mechanical systems for these machines, as well as conducting structural analysis using finite element analysis. Over the past 12 years, he has contributed to the design and development of various technologies, like the twin-screw continuous mixer, twin-screw mixer extruder, reinforced cement concrete machine tool structure as an alternative to traditional machine tool columns in horizontal boring machines, the high-speed shuttle less rapier loom, etc. (E-mail: vijaykumarn@cmti.res.in)