Artificial Intelligence Advancements Applied in Off-The-Shelf Controllers

By: Edward M. Marszal, P.E.
President, Kenexis
Ph: 614-451-7031
Fax: 614-451-2643
3366 Riverside Drive
Columbus, OH 43221
e-mail: edward.marszal@kenexis.com
http://www.kenexis.com

Abstract:

Since the earliest process units were built, CPI engineers have employed Artificial Intelligence to prevent losses. The expanding use of computer-based systems for process control has allowed the amount of intelligence applied in these expert systems to drastically increase. Standard methods for performing Expert System tasks are being formalized by numerous researchers in industry and academia. Work products from these groups include designs for the representation of knowledge of process hazards in a structured, hierarchical, and modular manner.

Advancements in programmable logic controller (PLC) technology have created systems with substantial computing power that are robust and fault tolerant enough to be used in safety critical applications. In addition, IEC 1131-3 standardized the programming languages available in virtually every new controller. The function block language, defined in IEC 1131-3 is particularly well suited to performing modular tasks, which makes it an ideal platform for representing knowledge.

This paper begins by describing some of the advancements in knowledge-based systems in loss prevention applications. The paper then explores how standard IEC 1131-3 programming techniques can be used to build function blocks that represent knowledge of the hazards posed by equipment items. The paper goes on to develop a sample function block that represents the hazards of a pressure vessel, using knowledge developed in the API 14-C standard.

Acknowledgements:

The author would like to thank Henri Tausch and Dale Ward of Honeywell Safety Management Systems for providing the program code used in the examples in this paper.

Introduction:

Since the earliest process units were built, CPI engineers have employed Artificial Intelligence to prevent losses. Specifically, Expert Systems, which diagnose problems and take actions in the same way that a human expert would, are employed to bring the process to a safe state when normal operating conditions are violated. Whether the equipment used to perform these expert actions is an independent computer system, or a series of pneumatic relays, the generic model of what occurs is the same.

Electromechanical and Pneumatic systems are very limited in their abilities to perform these actions. The expanding use of computer-based systems for process control has allowed the amount of intelligence applied in safety-instrumented systems to drastically increase.

Numerous researchers in industry and academia are formalizing standard methods for the application of artificial intelligence. The most prominent group performing research on loss prevention applications of Artificial Intelligence is the Abnormal Situation Management Joint Research and Development Consortium. This group, which was officially established in 1994, contains members from petrochemical operating companies, university research organizations, and control system hardware vendor organizations.

In addition to the advances in the field of Artificial Intelligence and Knowledge Representation, great strides are being made in process control systems. New designs and adherence to standards allowed the development of programmable logic controllers (PLC) that contain large amounts of computing power, and are robust and fault tolerant enough to be used in safety critical applications. In addition, the standard IEC 1131-3 created uniformity in the way that PLCs are programmed. IEC 1131-3 defines standard high-level languages used in PLC programming that include function blocks, ladder diagrams, sequential function charts, and structured text. Almost all new programmable control systems use these languages.

The function block language is particularly well suited to performing tasks that are structured, hierarchical, and modular. Using this approach, for each piece of plant equipment, a function block that represents knowledge of that equipment item’s hazards can be developed. This approach will allow users to program their safety instrumented systems by simply adding function blocks for each equipment item that is contained within the process unit.

Abnormal Situation Management

The Abnormal Situation Management (ASM) Consortium was informally established in 1992 as an outgrowth of effort to improve DCS alarm technology. In 1994, the group was formally established as a joint research and development effort that is partially funded by the National Institute for Standards and

Technology (NIST). ASM Consortium members include Honeywell, BP, Celenese, Chevron, Equilon, ExxonMobil, NOVA Chemicals, Union Carbide, The Ohio State University, and Purdue University.

The goal of ASM work, particularly that portion funded by NIST grants, was to demonstrate the technical feasibility of collaborative decision support technologies for improving the performance of operations personnel when responding to abnormal situations. The consortium proposed a system level solution that it refers to as AEGIS, or Abnormal Event Guidance and Information System. This system provides generic technical innovations in collaborative human-machine interaction, system architecture, and system customization toolkit to achieve this broad-based technological innovation. The final prototype of AEGIS demonstrated technical feasibility in the lab environment.

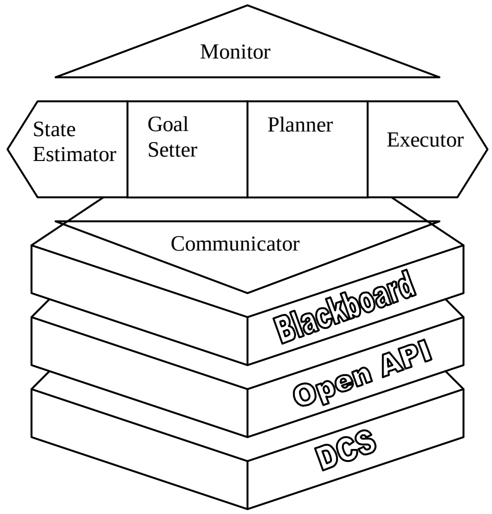
The prototype was developed using a layered architecture and based on open standards so that it can be run on any DCS supporting those standards. The software was developed using an infrastructure that would support multiple applications. All of the applications would support a common Plant and Operations Control Language and Common Blackboard for sharing messages between program modules.

The application modules work together to perform a four-step decision process. The steps in this process are:

- Determine the current state of the plant
- Decide the most appropriate goals to pursue
- Develop plans for pursing these goals
- Execute those plans

The decision process and the layers of hardware and communication required to implement the process are illustrated in *Figure 1*.

Figure 1   AEGIS Collaborative Decision Support Prototype



The original test-bed developed for this system included a wide variety of hardware and software. In addition to the DCS system used for control, several external workstation computers and a communication bridge were required. The hardware included Sun and HP workstations, and PCs. The software for the prototype was developed using packages as diverse as Gensym's G2 expert system shell and Microsoft's Visual Basic.

Advancements in COTS Technology

Advances in controller technology since the ASM team began work allow a significant portion of the functionality developed for the AEGIS system to be employed directly in a Safety PLC. Use of Commercial-Off-The-Shelf (COTS) technology, as opposed to the custom systems that were developed for the project will result in increased deployment of ASM developed technology at reduced costs.

While the ASM team worked diligently to develop a strong foundation of robust control and communication infrastructure where fast and extensive communication was critical, the general market for control systems also placed these requirements on their suppliers. As such, many of the capabilities that were specially developed for the AEGIS system are now standard features of COTS controllers due to their widespread application in any number of control tasks. The new features that are critical to implementing the ASM process include:

- Advanced Communications
- More Powerful and Flexible Processors
- Robust and Fault Tolerant Design

- Standardized Programming

Advancements in communications have been driven by the need to interface control equipment with operator interface. Due to the low-cost and power of personal computer based operator stations, in addition to the popularity of software packages that are deployed on the Windows operating system, Microsoft technologies quickly became de facto standards for data interchange in process controller. The Object Linking and Embedding (OLE) protocol for data transfer was modified by an industry group to suit the purpose of industrial control, yielding the OLE for Process Control (OPC) protocol that is now ubiquitous for controller to computer communications. In addition, communications from operator station PCs to other computers are performed by other de facto data interchange standards for database transactions such as Open Database Connectivity (ODBC) and Structured Query Language (SQL).

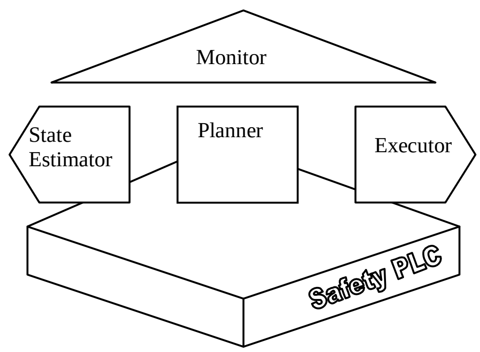
Perhaps the most important advancement is the increase in speed and memory of the processors upon which controllers are built. Controllers are now only a few generations behind personal computers in speed and available memory. COTS controllers are now able to perform functions far exceeding simple PID control. By using new high-level programming languages, virtually any algorithm can now be run directly in the controller instead of an external computer. The benefits of performing the calculations directly in the controller hardware are that the API and Blackboard levels in the AEGIS model are no longer required. The information is accessed directly from memory.

The creation of a variation of PLC that is purpose-built for safety applications will also benefit the application of ASM developments. These controllers are different from conventional PLCs due to their high level of diagnostic coverage. Use of this hardware will allow engineers to focus on diagnosing operational problems of the process without having to be concerned errors due to control system component failures. The extensive built in fault tolerance and diagnostics will ensure that any control system faults are found, and the proper action to move the process to a safe state is taken, all automatically, without additional effort from the process engineer implementing the AEGIS strategy.

The final advancement that will assist the deployment of ASM techniques in COTS equipment is the standardization of programming languages. The International Electrotechnical Commission (IEC) promulgated standard IEC 61131-3, which defines how programming of Programmable Logic Controllers should be performed. This effort is similar to the way that ANSI standardized the C programming language. The importance of this effort is that engineers can write code for one PLC and then port their work over to another PLC, made by a different manufacturer, and still be able to use the code. Control engineers aggressively adopted products that used the standard. At this point in time, virtually all programmable controllers use the IEC 61131-3 standard languages, or something very close, as a programming platform. One of the languages that is described in the 61131-3 standard is Function Block Diagrams which is a visual programming technique that is very modular, flexible and powerful. The combination of modular programming languages and a degree of portability of code between platforms will induce developers to create custom function blocks that implement the procedures that are described by ASM.

Once all of these factors are considered. A COTS Safety PLC is capable of providing a similar functionality to the AEGIS system, with an architecture that has substantially less components, as shown in Figure 2.

Figure 2 AEGIS-like Prototype in a COTS Safety PLC



Functional Knowledge Representation

A key part of performing the four-step process for abnormal situation decision-making is developing a representation of an expert's knowledge of a process. In research that is an indirect part of the ASM consortium's work, Davis, et al, propose that all of the knowledge about a chemical process required to make decisions, including decisions about abnormal situations, can be represented in a single structure.

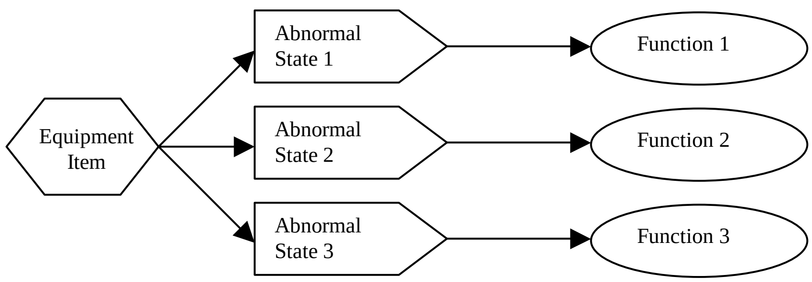
Davis, et al, apply an object-oriented approach that defines each individual equipment item in a process. For each equipment item, the answers to the following questions encapsulate all of the knowledge required for computer-based applications.

- What is it? – This question provides a link between the knowledge base and the physical plant.
- What does it do? – This question provides information on how the equipment item interacts with the process material
- When does it do it? – This question defines the conditions that are required before the actions identified by "what does it do?" are taken.
- How does it do it? – This question provides details on how the desired effect as achieved.

The Functional Representation framework allows each individual equipment item to have its characteristics described in isolation. The behavior of an entire system can then be inferred by combining equipment items into a system.

Functional Representations (FR) of equipment items can be effectively programmed directly into Safety PLCs. The process begins with development of a FR Schematic that describes the equipment item. *Figure 3* is a generic FR Schematic

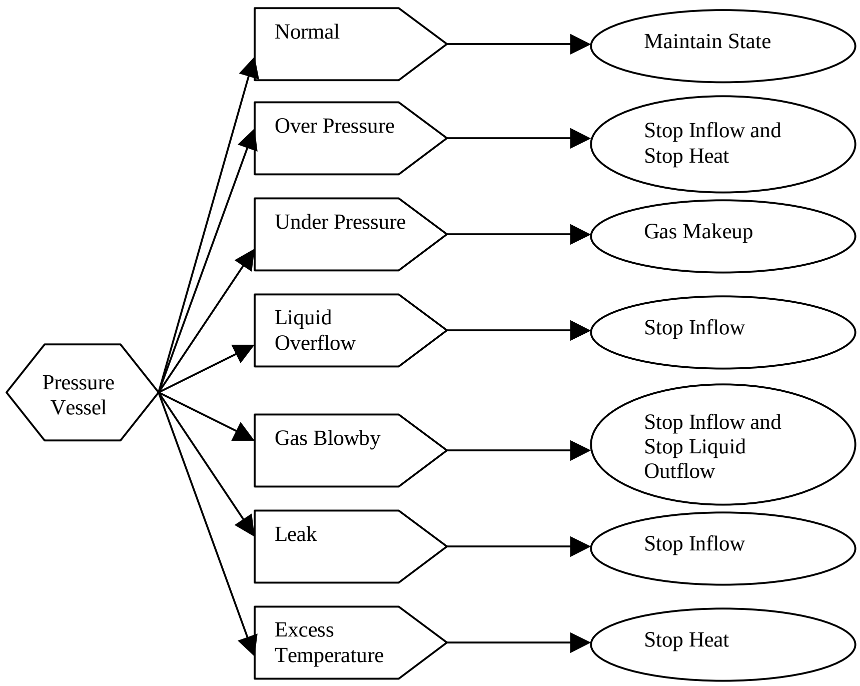
Figure 3 Generic Functional Representation Schematic



A Functional Representation of the knowledge required to protect a pressure vessel was developed using information contained in American Petroleum Institute Standard API-14C – *Analysis, Design, Installation and Testing of Basic Surface Safety Systems for Offshore Production Platforms*. Table A-4.1, *Safety Analysis Table for Pressure Vessels* contains a list of the Undesirable Events that can occur in a pressure vessel, and their associated detectable abnormal conditions. Based on the information in this table, the abnormal states of the FR schematic are the listed undesirable events. The detectable abnormal conditions that are listed in this table will be used in the State Estimator module to determine the state in which the pressure vessel is operating.

Section 4 of the standard describes the protection concepts that should be applied in case abnormal conditions exist. This information will define the function that should be executed upon the detection of abnormal conditions. Based on the information in Section 4 and Table A-4.1 of the standard, the FR Schematic shown in *Figure 4* can be derived.

Figure 4 FR Schematic for a Pressure Vessel



Implementing Functional Representation in COTS Safety PLCs

The knowledge represented the in the FR Schematic shown in *Figure 4* can be implemented directly in a Safety PLC without requiring external computers and communications equipment. By using Function Block Diagram programming, modular program code is created that will allow a single block of code to be used to represent all vessels that are contained in a process. The actual code that is developed in this section is shown at the end of this paper.

A function block diagram has a “top level” where inputs and outputs are shown, and various “interior levels” where the detailed code is executed. Based on Figure A-4 in the API-14C standard, a pressure vessel should have the high and low-pressure indication, high and low-level indication, and high temperature indication. These measurements are used to estimate the state of the process (i.e., detect abnormal conditions). Based on the descriptions of the functions that are required in each abnormal state, the outputs that are required include: gas makeup valve, inflow shutoff valve, liquid outflow shutoff valve, and a heater power interruption relay.

The function block developed for this example has multiple interior levels which are themselves function blocks. These interior function blocks include state estimation, planning, and execution functions. The first function block diagram is the state estimation section. In this section, the inputs to the function block are analyzed to determine the state in which the vessel is currently operating. The goal setting section of the AEGIS model is not required in this instance. The goal is always to return the process to a safe state. The planner section of the code uses the process state as an input and uses that to create a plan for returning the process to a safe state. The final section of the code is the execution section. In this section, the generic plan is converted a set of outputs that will create the desired action.

The benefits of this procedure should be apparent. Development of the software required to monitor and take protective action for a process plant will be enormously simplified, and can be developed by process engineers with little or no expertise in control system programming. The controller configuration process is simplified to listing the process equipment items that are included in the processes, and assigning the proper tag number to the instruments that are located on that vessel. It is entirely possible that by using advanced computer aided design packages for development of Piping and Instrumentation Diagrams (P&ID), that the Safety Instrumented System will be able to program itself, given the digital representation of the process equipment as given in the intelligent P&ID.

Issues that must be resolved prior to application

Although the benefits of using a knowledge-based and object oriented approach to programming safety programmable logic controllers are apparent, there are a number of weaknesses that prevent more widespread application of these techniques. More research in these areas will allow these techniques to be applied in industrial applications.

Knowledge Bases are Incomplete – While the API-14C standard does an excellent job of listing the abnormal states associated with equipment items used in the offshore oil production industry, the information does not contain a comprehensive list of equipment items used in the Chemical Process Industries, nor is the enumeration of the potential failure modes exhaustive for all applications. Development of this information in a standardized format is the key to making knowledge-based techniques useful.

Excessive Program Size – The program size using functional representation techniques is much larger than using traditional programming techniques, and is more difficult to program. In order to make functional representation feasible in industry, Safety PLC memory will have to continue to expand at a rapid rate, and generic function blocks that can be applied to a very large number of applications will need to be developed.

Multiple States and Multiple Modes – The example that was demonstrated here shows a single mode of operation with a set of failure states. In reality, there will likely be a large number of operating modes, especially if the process is operated in batches. In addition, multiple failure modes, or abnormal states, might be present simultaneously. The code developed for the example case is not capable of handling multiple simultaneous abnormal states in its current format.

Restarting after Shutdown – Functional Representation programming techniques will have to consider additional states such as “shutdown” and require manual intervention to move from the “shutdown” state back into an operating mode.

Summary

While the benefits of using functional representation based programming to configure safety systems are great, there are serious impediments to its application. The benefits of the technique include decreased risk analysis effort, and decreased programming effort, while providing a comprehensive, albeit automatic, analysis of the process. The issues that must be addressed prior to satisfactory implementation of this technique is possible stem from the fact that our knowledge of process components is incomplete, and has not been presented in a formalized way.

Eventually, a commercial off-the shelf safety controller will be able to configure itself by only giving it access to a digital representation of a plant’s process equipment by using function representation based methods. The representation of the plant can be gained either through exposure to intelligent piping and instrumentation diagrams or through connection to intelligent field devices. It is possible that by polling the field devices on a process control network, the safety controller will be able to determine the process equipment used in a process in real time, and continuously update and reconfigure itself.

References

American Petroleum Institute, *API Recommended Practice 14-C – Recommended Practice for Analysis, Design, Installation, and Testing of Basic Surface Safety Systems for Offshore Production Platforms*, (American Petroleum Institute: New York, 1998)

Davis, J.F., Piovoso, M.J., Hoo, K.A., Bakshi, B.R., *Process Data Analysis and Interpretation*, (Academic Press: Advances in Chemical Engineering, Vol. 25, 2000)

Honeywell, *The Abnormal Situation Management Joint Research and Development Consortium – Program Overview*, (Honeywell: http://www.iac.honeywell.com/Pub/AbSitMang, as posted 10/30/00)

Miller, D.C., Elsass, M.J., Goodaker, A.W., Davis, J.F., Josephson, J.R., Chandrasekaran, B., *Design of Multipurpose Engineering Knowledge Databases for Process Systems Applications*, (The Ohio State University: Departments of Chemical Engineering and Information Sciences, Columbus, 1999).