An Intelligent Control Architecture
For Expert Process Control

Jianguo Chen
Ronald D. Bonnell
Electrical and Computer Engineering
University of South Carolina
Columbia, South Carolina

Abstract
An intelligent control system architecture is proposed that employs multiple knowledge sources including procedural algorithms, rule-based methods, reasoning mechanisms and process knowledge for complex process control. The architecture is based on the blackboard control model which enables expert systems to flexibly select their own control strategies at different levels of abstraction for multi-task problem-solving in process control. Control knowledge sources and a control blackboard are included so that the control plan can be modified dynamically according to the state of the process. Expert control systems can improve their behavior dynamically by adapting to an environment with large uncertainty or contingencies.

Introduction
There is a growing interest in applying artificial intelligent methods to process control and the term "intelligent control" frequently occurs in the present control literature. The reason for this interest is that there still exist a large class of practical problems for which a satisfactory solution can not be found in conventional control system theory. Some processes are difficult to model because of poorly defined process knowledge. A large number of processes have complex models which include time delays, time-varying and non-linear coefficients, etc., so that no single numerical algorithm can solve the problem. As a result, integration of multiple algorithms is required. Even for the widely used PID regulator, a safety jacket is necessary to deal with large amounts of heuristic logic [Astrom86]. More over, a control system needs the ability to monitor, interpret, diagnose and handle urgent events. In other words, the system design should be concerned more with the "intelligent" performance [Hayes89b] of the system than just about the direct mapping from input signals to output control as specified by conventional control theory.

One of the early works concerned with the concept of "intelligent control" [Fu71] is the fuzzy or rule-based control approach [Mamdani74][Kickert76]. In this research the controller consists of sets of if-then rules, which influence the plant directly by providing the plant with manipulating signals. The software is implemented more or less in standard programming techniques. Advances in artificial intelligence have addressed the research interest in "Expert Control" [Astrom86] which is the attempt to apply an expert system as an intelligent supervisory for extending the functionality of a conventional control system.

Moore describes the intelligent system PICON [Moore84] where an expert system is used to give real-time alarm, diagnosis and control action advice to the process operator. Another real-time expert system ESCORT[Sachs86] deals with large amounts of changing data and reasoning about the behavior of a process plant to ease the cognitive overload on operators. G2[Gensym89] is a new AI tool for real-time applications where hundreds of variables may be continuously monitored. The main capability of G2 is its rapid response to dynamic data changes in process control. The knowledge in G2 is represented in schematics, graphical flowcharts, dynamic models and rules.

A more recent application of expert systems is the implementation of "smart" tuners. Porter describes an expert system tuner for the PI controller. The parameters in the PI controller are adjusted by the heuristic tuning rules in the knowledge base for different classes of transient responses [Porter87]. Astrom and Arzen implement an auto tuner by combining the numerical algorithms (control algorithms, identification algorithms and monitor algorithms) with a knowledge-based system which handles large amounts of heuristic logic in the control process [Astrom86]. Trankle and Markovian examine a real-time knowledge-based adaptive control system, which consists of three expert modules: the system identifier, control system designer and control implementation supervisor [Trankle85].

The Architecture Requirement
From an AI point-of-view, process control systems perform a series of problem-solving actions to achieve a final objective. For example, if parameters of the plant vary with time the control system has to recognize the deterioration of the output response and make a decision on tuning. On-line tuning should be done when the system works in a stable status. During tuning a load change may occur and must be identified. Then tuning must stop quickly and the system must switch smoothly back to closed-loop control to correct the error caused by the load change. This is in fact a multi-task planning problem. To solve problems of this type a appropriate way is to use partitioned knowledge sources which can be operated at different level of priority rather than a uniform knowledge base.

The next consideration is the system control problem. Conventional rule-based systems use implicit control knowledge. This increases the difficulty of selecting the control strategies for domain-specific knowledge. This difficulty is particular true for complex process control. Intelligent systems need to explicitly solve the control problem (explicitly decide what pieces of knowledge should be used, and when and how to use them). A blackboard control architecture facilitates the implementation of explicit control scheme for intelligent control systems, which is the main concept addressed in this.
A generic blackboard framework [Nii86] contains three components: a common data region called the blackboard (BB); a set of independent knowledge sources (KS); and a control mechanism. The blackboard keeps a set of hypotheses of the partial and full solutions. It has a hierarchical structure with each level corresponding to a different level of abstraction in problem solving. Each knowledge source contains its own domain knowledge, e.g., a control algorithm for particular control phase, which will be used to confirm or reject the current hypothesis for the problem solutions. The control mechanism chooses and schedules the execution of a knowledge source. By using opportunistic reasoning, knowledge source is accessed with either forward or backward chaining. The final solution is determined by selecting the hypothesis which is highly supported by various knowledge sources. Some applications of blackboard architecture in process control have emerged [D'Ambrosio87] [Rubin881 [Shadbolt89].

One implementation of the control mechanism includes a BB monitor, a scheduling queue and a scheduler. The BB monitor examines which KS is executable from evidence of events on the BB. The scheduler calculates a priority rating for KS instantiation on the schedule queue, selecting for executing the one with highest rating. A new approach achieves scheduling flexibility by incorporating a control BB and control knowledge sources [Hayes85][Hayes89a]. The control BB records the desired actions, the feasible actions and the action executed. This information is used by the scheduler to decide which KS to execute from a set of executable knowledge sources. Thus, the control mechanism provides a great deal of flexibility in realizing a focus-of-attention mechanism for expert control systems.

**An Intelligent Control Architecture**

Expert control systems are required to make intelligent decisions supported by multiple knowledge sources and inference methods. To accomplish this, the selection of the control mechanism is very significant. Many of the present expert control systems use conventional expert system architectures, where the control strategies are implicit or fixed. These approaches to control are appropriate for relatively simple environments but not satisfactory for complex process control.

This research presents a study of an intelligent control architecture based on the blackboard control framework [Hayes85] for complicated control processes systems showed in Fig. 1. The framework contains two parts:

1. a domain blackboard (DBB) with domain knowledge sources (DKSs) which integrate multiple numerical algorithms and rule-based methods, and provides the communication mechanism for the domain knowledge sources. The first part serves as the execution level of the intelligent control hierarchy.
2. a control mechanism which includes a control blackboard (CBB), control knowledge sources (CKSs), basic control modules including a scheduler, executor and two BB monitors (agenda manager), an agenda, a DBB event-DKS table and CBB event-CKS table. The second part of the model serves as the organization level to perform high-level operations like planning and reasoning about the behavior of the execution level.

The design is expected to be capable of flexibly combining multiple numerical algorithms with the knowledge-based system for multitask-oriented behavior in expert control systems such as monitoring, diagnosing, tuning, PID control algorithm, rule-based control, etc. Furthermore, the mechanism of dynamically constructing, modifying and executing control plans provides the intelligent control system with the ability to react promptly, to re-focus its attention in response to the occurrence of critical events, and to improve its behavior by adapting to environments with large uncertainties.

**Control Mechanism**

The DBB is a global database which contains current events and hypotheses about the problem domain. An event is either a segment of input data or a change of the information on the DBB. Some examples of events are the current system output, the setpoint, the current control mode, the control output, parameters of the controller, etc. A hypothesis is an organized collection of the derived understandings of the process conditions. This organized collection forms different levels of abstraction on the DBB. The low-level concern deals with tropistic reasoning with the input data, e.g., the hypothesis about status of system errors is easily deduced from sensor data. At a higher-level more complicated hypotheses are formed as abstractions of the low-level hypotheses. An example of this type of hypothesis can be an evaluation of current level of stability of the process. The knowledge sources are a set of programs that can be either algorithms, rule sets, an expert system, or another whole blackboard system.

Some of the basic DKSs are:

- **Process monitor** - The process monitor inspects the system's status, detecting the control error and extracting the features of the error regarding a set of attributes (e.g., overshoots, oscillation, and steady-state error, etc.). It can also be viewed as a feature extractor which works in the forward chaining mode.

**Figure 1 Blackboard Control Architecture for an Expert Control System**

- Executive
- Agenda (KS/AR)
- DBB Monitor
- CBB Monitor
- DBB Event-DKS Table
- CBB Event-CKS Table
- Control plan
- Poissy CKS
- Strategy CKS
- Problem CKS
- CB
- PID Controller
- Fuzzy Controller
- Tuner
- Diagnostician
- Process Monitor
- Operator Information
- M/S
- Disturbances
- Process Outputs
- Process Output Control Inputs

The design is expected to be capable of flexibly combining multiple numerical algorithms with the knowledge-based system for multitask-oriented behavior in expert control systems such as monitoring, diagnosing, tuning, PID control algorithms, rule-based control, etc. Furthermore, the mechanism of dynamically constructing, modifying and executing control plans provides the intelligent control system with the ability to react promptly, to re-focus its attention in response to the occurrence of critical events, and to improve its behavior by adapting to environments with large uncertainties.

**Control Mechanism**

The DBB is a global database which contains current events and hypotheses about the problem domain. An event is either a segment of input data or a change of the information on the DBB. Some examples of events are the current system output, the setpoint, the current control mode, the control output, parameters of the controller, etc. A hypothesis is an organized collection of the derived understandings of the process conditions. This organized collection forms different levels of abstraction on the DBB. The low-level concern deals with tropistic reasoning with the input data, e.g., the hypothesis about status of system errors is easily deduced from sensor data. At a higher-level more complicated hypotheses are formed as abstractions of the low-level hypotheses. An example of this type of hypothesis can be an evaluation of current level of stability of the process. The knowledge sources are a set of programs that can be either algorithms, rule sets, an expert system, or another whole blackboard system.

Some of the basic DKSs are:

- **Process monitor** - The process monitor inspects the system's status, detecting the control error and extracting the features of the error regarding a set of attributes (e.g., overshoots, oscillation, and steady-state error, etc.). It can also be viewed as a feature extractor which works in the forward chaining mode.
associated with a certain level of the CBB. The kind of problem should be solved next. This activity is also
execute next. Most of the CKSs are domain-dependent; the hierarchical control structure. Each of the CKSs is
knowledge for decision-making at the higher level of supervisory control. The CKSs are organized according to
Focus levels to create the control plan on the CBB which is on the agenda.

Focus levels to create the control plan on the CBB which is on the agenda.

Policy level includes strategy objects each of which is used by the scheduler to decide which DKS or CKS to
plan.

Problem level includes the objects of control planning for general description of problem solving behavior. Each control plan specifies a series of strategies used for completing the plan.  
Strategy level includes strategy objects each of which is divided into a set of subobjects needed to complete the strategy.

Policy level includes focus objects which provide the policy decision for rating and scheduling KSARs (Knowledge Source Activation Records) which are recorded on the agenda.

Figure 2 The Levels of the Control Blackboard

Figure 2 The Levels of the Control Blackboard

CKSs determine which of the problem-solving action are desirable. CKSs use heuristics at the Problem, Strategy and Focus levels to create the control plan on the CBB which is used by the scheduler to decide which DKS or CKS to execute next. Most of the CKSs are domain-dependent; however, instead of detail algorithms they contain the knowledge for decision-making at the higher level of supervisory control. The CKSs are organized according to the hierarchical control structure. Each of the CKSs is associated with a certain level of the CBB. The problem CKS analyze the current solution state and determine what kind of problem should be solved next. This activity is also
known as focusing-of-attention because the system will concentrate on the context assigned by problem CKSs. The strategy CKS determine several control strategies related to the focused problem. The policy CKS generate policy decisions which are used to establish global criteria for KSARs in response to particular problem-solving states and control strategies. For example, after examining the system's state, a CKS associated with the problem level of CBB may generate a hypothesis of tuning, which may trigger some CKSs to put hypotheses about steps of tuning on the CBB at the strategy level. Furthermore, these hypotheses together with the current events (state of the system) will generate policy decision on the policy level about the criteria of priority for each detail algorithm of tuning. This procedure forms a control plan of tuning.

The basic control modules are usually domain-independent. They identify which action is feasible and executable on the basis of focus objects on the BBs. Their functions are described as follows:

- **Problem Monitors** checks its blackboard and its event-KS table. It creates a new KSAR on the agenda for each pair of knowledge source and blackboard event that is appropriate for the current status of its blackboard.
- **Scheduler** interprets the control plan on the CBB, computes each KSAR's priority against the decision at the policy level, chooses an executable KSAR that best satisfies the current control plan and send it to executor.
- **Executor** executes the action of the selected KSAR, and produce events (the changes to BB objects)

The Event-KS tables list all of the possible matching pairs of each knowledge source and blackboard event which are referenced by the monitors to update the agenda.

The Agenda is a database which contains knowledge sources that can be executed, and their triggering and action context, i.e., KSARs.

The system's basic operating steps are:
1. update the agenda to create new KSARs, and then
2. schedule the executable KSAR, and then
3. execute the selected KSAR, update BBs, and then repeat to (1).

These three steps in a loop form the blackboard control architecture's basic control loop.

The proposed knowledge-based system will be implemented in GBB [Gallagher88], one of the well-known blackboard tools. GBB contains two distinct subsystems: 1) a blackboard database compiler which includes specification languages for defining a blackboard and blackboard objects, and for specifying the insertion / retrieval storage structure; 2) a control shell defines the knowledge sources, specifies the conditions for their activation, and strategies for their scheduling. The knowledge sources are implemented as frames with data and method slots.

**Conclusion**

A prototype of the proposed intelligent control architecture for process control is under development. The prototype will serve as a test bed where control paradigms for an expert control system can be studied. Our main research interest is concerned with the control architecture, control knowledge, control plans and the link between different levels of abstraction on the control blackboard in a real-time situation.

To accomplish this task, a simulation model of an actual process is selected as the process of the proposed expert control system, and different groups of the domain-specific knowledge can be collected as the execution level of the expert control. The system supports flexible, critical-event
driven control. It is designed for dynamically selecting the most appropriate of multiple, alternative problem-solving strategies in response to the conditions in the system's task environment.

References


