A Refinery Immobot for Abnormal Situation Management

Kurt D. Krebsbach and **David J. Musliner**
Automated Reasoning Group
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418
{krebsbac,musliner}@htc.honeywell.com

Abstract

Oil refineries literally provide the lifeblood for global economic health, and disruptions to their operations have major worldwide impact. We are developing a large-scale semi-autonomous refinery immobot to assist human operators in controlling refineries during abnormal situations. Based primarily on reactive and procedural approaches to intelligent behavior, the AEGIS system will interact with multiple users and thousands of refinery components to diagnose and compensate for unanticipated plant disruptions. Through intelligent autonomous behavior and improved human situation awareness, the AEGIS project is expected to have a multi-billion dollar annual impact on refinery productivity.

Introduction

The largest economic disaster in U.S. history was a $1.6 billion explosion at a petrochemical plant in 1989. This accident represents an extreme case within the spectrum of major industrial process disruptions, collectively referred to as *abnormal situations*. While most abnormal situations do not result in explosions, they can be extremely costly, resulting in poor product quality, schedule delays, equipment damage, reduced occupational safety, and environmental hazards. The inability of automated control systems and plant operations personnel to control abnormal situations has an economic impact of at least $20 billion *annually* in the petrochemical industry alone.

Systems for controlling oil refineries provide an excellent example of what Williams and Nayak have called *immobots* (Williams & Nayak 1997). Immobots are appealing because they provide the richness inherent in interacting with real, physical environments (as with *robots*), while maintaining the ready accessibility of a networked software environment (as with *softbots*). The functions of a refinery immobot are directed inward, focusing on the control of its complex internal functions. These functions include maintaining desired operational conditions (goals), detecting threatened operational goals, estimating (abnormal) states, and diagnosing, isolating, and recovering from malfunctions.

At the Honeywell Technology Center, we are designing and building a large-scale refinery immobot known as AEGIS (Abnormal Event Guidance and Information System). AEGIS is a semi-autonomous agent specifically designed both to assist operations personnel in the plant during an upset (by displaying the correct information, filtering alarm floods, etc.), and to take compensatory and diagnostic actions autonomously via a digital control system. In this paper we describe a portion of the goal-setting, planning, and execution components of AEGIS.

Abnormal Situations

During normal refinery operation, so-called "advanced control" software can be used to automatically adjust parameters of the plant to keep it running at near-peak profitability. Advanced control algorithms are based on detailed mathematical models of the plant's behavior, which are amenable to numerical optimization. However, these models are quite brittle, accurately modeling plant behavior only when the plant is running smoothly. As a result, advanced control is currently an all-or-nothing proposition; if it is on, the operator does not participate in the functions it is controlling, and if it is off, the operator is completely in charge of all aspects of the unit. During many abnormal situations, advanced controls are either turned off automatically or manually, as their fine-grained but narrow techniques become less-fitting models of the real-world plant's behavior. Then, human personnel, including board operators, field operators, and shift supervisors assess the situation as best they can, and begin following general procedures on which they have been trained.

The procedures can be quite long (dozens of pages), and by necessity contain lots of structure and contingencies, since the exact state of the plant is almost never known with certainty. Many of the procedural actions involve sampling data, confirming other readings, and performing diagnostic tests. Some procedures apply to extremely general contexts (e.g., we're losing air pressure from somewhere), while some are less general (e.g., air compressor AC-3 has tripped (shut off)), and some are very specific (e.g., the lube oil pump has

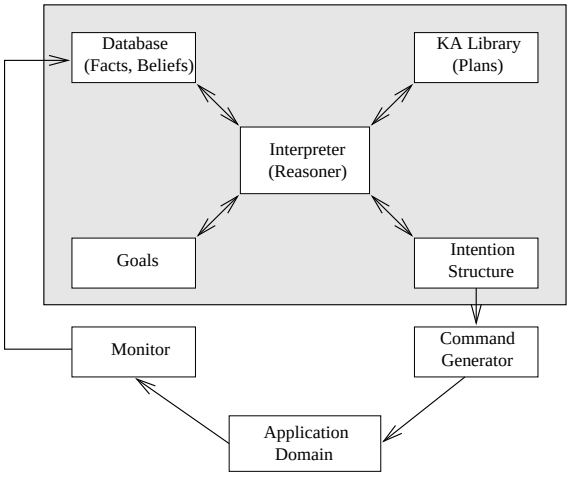

Figure 1: The PRS Architecture.

a broken driveshaft).

The procedures are designed to satisfy several general goals, and the associated actions can fairly consistently be grouped into four sets:

- **Compensatory Actions:** Actions designed to move the plant to a safe state, regardless of whether the root cause(s) of the upset has been determined. Compensatory actions usually have a significant but short-term negative impact on plant profitability which is minor relative to a total shutdown. Typical examples include reducing the feed rate, spilling feed to temporary tanks, lowering temperatures, and adding steam.
- **Diagnostic Actions:** Actions designed to ascertain the root cause or causes of an upset, thus facilitating actions to more specifically address the situation, eventually repairing it. In some cases, many compensatory actions can be taken prior to root cause determination, while in others, a rapid diagnosis is required to avoid automatic subsystem shutdown.
- **Recovery Actions:** Actions aimed to restore the plant to its near-optimal operating conditions, particularly after a root cause has been diagnosed and repair, restoration, and recovery can begin.
- **Communication Actions:** Actions focused primarily on communicating with the user, maintaining accurate user awareness of the ongoing situation and assisting the user in making decisions for which AEGIS has not been authorized.

A Procedural Approach

We have chosen to build the core reasoning engine of AEGIS in C-PRS (Ingrand 1994), a C-based version of the Procedural Reasoning System (Ingrand, Georgeff, & Rao 1992; Georgeff & Lansky 1986). As shown in Figure 1, knowledge in PRS is represented as a declarative set of facts about the world, together with a library of user-defined *knowledge areas* (KAs) that represent procedural knowledge about how to accomplish goals in various situations. Goals represent persistent desires that trigger KAs until they are satisfied or removed. The *intention structure* represents currently-selected KAs that are in the process of executing or awaiting execution, in pursuit of current goals. Finally, the PRS *interpreter* chooses KAs appropriate for current goals, selects one or more to put onto intention structure, and executes one step from the current intention structure.

Our motivation for using a procedural approach comes chiefly from the fact that sufficient models do not exist for abnormal situations. Instead, the expertise for managing abnormal situations is captured in the form of expert-generated standard operating procedures (SOPs), which require detailed elaboration and regular updates to comply with safety laws. Our experience shows that human operators regularly make use of this type of "compiled" procedural knowledge when responding to upsets, largely as a matter of expedience. PRS provides rapid, context-sensitive procedural invocation, so that AEGIS can focus on and respond to new contexts in a matter of milliseconds, rather than, for instance, computing suitable responses from an enormous, first-principles model of the plant.

Other features of PRS which have proven to be extremely useful for this domain include:

- The **hierarchical, subgoaling** nature of its procedural representation, which allows PRS to combine pieces of plans in novel ways, and is important for flexible plan execution and goal refinement.
- Its ability to pursue multiple, **goal-directed** tasks while at the same time being **responsive** to changing patterns of events in bounded time.
- Its ability to construct and act on **partial** (rather than complete) **plans**.
- Its default mechanisms for handling stringent **real-time demands** of its environment, especially crucial for abnormal situation management.
- Its **meta-level** (or reflective) reasoning capabilities, an important feature for controlling the allocation of processing resources, planning attention, and alternative goal achievement strategies.
- Its knowledge representation assumptions, which encourage **incremental refinement** of the plan (procedure) library, an enormous advantage for large-scale applications which undergo constant evolution as regular maintenance, repairs, and improvements modify the plant's operation.

Domain Aspects

In this section, we enumerate several aspects of the domain which, in combination, provide a unique and difficult problem:

1. **Discrete and Continuous:** Control of a refinery involves a combination of discrete choices (e.g., which tank, which valve, which order, which direction, etc.), and continuous settings (how much feed, pressure, temperature, etc.). The PRS binding mechanisms, along with property assignment for resources, priorities, etc., work well for the discrete part of the problem. Small mathematical models for computing appropriate continuous values can be coded directly in PRS procedures or called as external subroutines. We call these "mini-models," to distinguish them from the more all-encompassing model-based approaches employed by advanced control and traditional AI planning systems.

2. **Highly Dynamic:** In general, long-term projective plans are not useful during abnormal situations because the plant is too inherently dynamic and the effects of chosen actions are highly variable and difficult to predict. In most cases, such a plan would become obsolete almost immediately, which is why we have chosen a reactive approach.

3. **Delayed Action Effects:** There exist long, varying gaps between when an action is taken, and when its effects can be confirmed. Timing constants are used to estimate this delay, so that AEGIS can intelligently monitor the progress of its actions.

4. **Delayed Performance Measures:** Due to engineered redundancy in plant design, there are often multiple ways to achieve a given goal. The relative utility of alternative goal achievement methods must often be computed on the fly, as the evaluation can rely heavily on the current context. We use a combination of PRS' native prioritization mechanisms and meta-level control to resolve these conflicts. In addition, we envision using external predictive models to simulate the effects of choices to aid the decision, when time and resources allow.

5. **Hierarchy of Authorization Levels:** The degree of autonomy AEGIS assumes is dictated by the rules of authorization on an action-by-action basis. It is pre-authorized to take some actions autonomously, must request authorization for some, and is never allowed to take others. AEGIS currently seeks authorization for those KAs that require it, and simply fails the KA if authorization is not received. Of course, AEGIS will then seek alternative ways to achieve the goal, which may themselves require further authorization requests.

6. **Mixed Initiative:**
   AEGIS is a semi-autonomous associate system, intended to operate as a partner and assistant to the human operators. Many of the available control actions can be taken either by AEGIS or by the human operator; the problem-solving process between AEGIS and plant personnel is inherently *mixed-initiative*. The human and immobot must agree on who's doing what, when, where, how, and in some cases, why. For example, AEGIS must keep the human informed of its progress and failures, and when it has run out of alternatives for achieving one of its goals. The human must be able to cancel AEGIS-initiated actions, and even goals, in cases where AEGIS does not have sufficient knowledge. We have found ways in PRS to implement a portion of this situational awareness, but it is a difficult problem that requires a great deal more study.

Conclusions

We have briefly described a semi-autonomous, reactive immobot that assists in controlling an oil refinery during abnormal situations. Because sufficient models of the refinery do not generally exist, we have employed a procedural knowledge representation implemented in C-PRS. Preliminary results have been very encouraging, despite the many challenging features of this complex, high-impact domain.

Acknowledgments This effort was in part supported by the NIST Advanced Technology Program, Award 70NANB5H1073 to the Abnormal Situation Management Joint Research and Development Consortium.

References

Georgeff, M., and Lansky, A. 1986. Procedural knowledge. *IEEE Special Issue on Knowledge Representation* 74:1383–1398.

Ingrand, F.; Georgeff, M.; and Rao, A. 1992. An architecture for real-time reasoning and system control. *IEEE Expert* 7:6:34–44.

Ingrand, F. F. 1994. *C-PRS Development Environment (Version 1.4.0).* Labege Cedex, France: ACS Technologies.

Williams, B. C., and Nayak, P. P. 1997. Immobile robots: AI in the new millennium. *AI Magazine* 17(3).