ORNL/TM-2014/363
SMR/ICHMI/ORNL/TR-2014/05

Development of an Automated Decision-Making Tool for Supervisory Control System

Sacit M. Cetiner
Michael D. Muhlheim
George F. Flanagan
David L. Fugate
Roger A. Kisner

September 2014

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2014/363
SMR/ICHMI/ORNL/TR-2014/05

Reactor and Nuclear Systems Division

DEVELOPMENT OF AN AUTOMATED DECISION-MAKING TOOL FOR SUPERVISORY CONTROL SYSTEM

Sacit M. Cetiner
Michael D. Muhlheim
George F. Flanagan
David L. Fugate
Roger A. Kisner

Date Published: September 2014

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

ACRONYMS

AdvSMR	Advanced Small Modular Reactor
AHP	Analytic Hierarchy Process
CDF	Core Damage Frequency
DES	Discrete Event Systems
dll	Dynamic Link Library
DOE	US Department of Energy
ET	Event Tree
ETA	Event Tree Analysis
FT	Fault Tree
FTA	Fault Tree Analysis
HMI	Human Machine
ICHMI	Instrumentation, Controls, and Human-Machine Interface
ICS	Integrated Control System
IE	Initiating Event
LOOP	Loss of Offsite Power
LSCS	Local Supervisory Control System
MSCS	Master Supervisory Control System
O&M	Operation and Maintenance
OLCs	Operational Limits and Conditions
OOS	Out of Service
OP	Operating Procedure
PRA	Probabilistic Risk Assessment
RPS	Reactor Protection System
RTEL	Real-Time Execution Layer
RWB	Reliability Workbench
SCS	Supervisory Control System
SMR	Small Modular Reactor
UHS	Ultimate Heat Sink

ACKNOWLEDGMENTS

This project is funded by US Department of Energy, Office of Nuclear Energy under the Instrumentation, Controls, and Human-Machine Interface (ICHMI) research pathway within the Advanced Small Modular Reactors (AdvSMR) program.

EXECUTIVE SUMMARY

Small modular reactors (SMRs) can provide the United States with a safe, sustainable, and carbon-neutral energy source. Because of their small size and, in many cases, simplified nuclear island configurations, it is expected that the total cost of power generation will be significantly less for SMRs compared to those of large Generation III+ light-water reactors. Advanced SMRs, which use coolants other than water as the primary heat transport medium, can enhance the simplicity gains by introducing several passive safety and control characteristics.

The benefits of SMRs can include reduced financial risk, operational flexibility, modular construction, grid flexibility, and waste reduction. Achieving these benefits can lead to a new paradigm for plant design, construction, and operation to provide for multi-unit, multi-product stream-generating stations and compensate for reduced economy-of-scale savings from their smaller size. However, there are technology needs that must be addressed to resolve challenges to establishing this new paradigm. This condition is particularly true for the unique characteristics and different operating environments associated with advanced SMR concepts. Consequently, the US Department of Energy (DOE) Office of Nuclear Energy (NE) established the Advanced SMR (AdvSMR) Research and Development (R&D) Program.

The economic factor most strongly affected by the loss of economy of scale is the day-to-day cost of plant operations. The controllable day-to-day costs of SMRs are expected to be dominated by operation and maintenance (O&M) costs, which are heavily dependent on staffing size and plant availability. Efficient, effective operational approaches and strategic maintenance can help contain these costs and ensure economic viability.

Instrumentation, Control, and Human-Machine Interface (ICHMI) technologies provide the foundation for what is the equivalent of the central nervous system of a nuclear power plant. Therefore, innovative use of intelligent automation can have a significant impact on reducing plant staffing compared to current plants and controlling O&M costs based on reductions in workload realized with improved plant control systems. Intelligent automation in the control systems can be used to increase plant availability (and thus safety) by maintaining system operational parameters within safety system setting. Essentially, the economy of automation can serve as a compensating factor for the loss of economy of scale while simultaneously increasing plant availability.

Unfortunately, highly automated, intelligent control capabilities have not been demonstrated for nuclear power plant operations, and there is limited experience in other application domains. Improved supervisory control system capabilities provide a means for the integration of control, decision, and diagnostics to support extensive automation. The targets for automation include operational management of highly complex plants, dynamic management and control of multiple product streams from a plant, and coordinated management of multiple modules.

Within the ICHMI technical research area under the AdvSMR R&D program, the Supervisory Control of Multi-Modular SMR Plants project was established to proceed with development and demonstration of the architectural framework and foundational modules that are needed to facilitate the integration of control, decision, and diagnostics to support the necessary level of automation.

This report builds on the architecture and decision-making methods identified and documented in the previous phase of the project, where an advanced automated decision-making process was incorporated into the supervisory control system architectural layers through the introduction of a tiered-plant systems approach.

This technical report documents the findings and progress made during the third phase of research activities for the AdvSMR Supervisory Control project. Specifically, the report introduces a *Generalized Framework for Decision-Making*, which can be used as a template flow sheet for automated or autonomous decision-making in various applications. This general framework is composed of three fundamental steps: (1) identify decision alternatives, (2) evaluate alternative decisions, and (3) generate a single solution or a single trajectory. In step 1, the module identifies possible or available courses of action given a deviation from nominal state. These can be due to failure or degradation of a component. In step 2, these options are assessed based on a predetermined metric. Finally in step 3, a single solution is selected for execution. The objectives and the requirements of this framework can be realized using various methods and analytical tools.

Furthermore, this report illustrates the specific implementation of autonomous decision-making—based on the generalized decision-making framework—for a supervisory control system intended for use in an AdvSMR plant with multiple reactor modules. In this proposed implementation, the autonomous decision-making module includes two functional blocks: (1) decision-options block and (2) decision-analysis block. The options block uses a probabilistic risk assessment (PRA) tool with real-time updates, which is called *real-time PRA* in this report. With this unique implementation, it is possible to represent a wide spectrum of system and component states without having to create decision logics. Fault trees and event trees, common tools in conventional PRA analyses, provide insight about system topology, whereby available options and decision trajectories can be ascertained given an abnormal event. The decision analysis block uses utility theory to perform the deterministic portion of decision-making.

In order to perform autonomous decision-making, the supervisory control system must properly flag a component for failure or degradation in the model. This requires that the decision-making module must reconstruct an associated event tree, map the fault to the appropriate event tree branch, and then deconstruct the event tree to identify available control options.

The present research demonstrates the probabilistic portion of autonomous decision-making through a simple thermal-hydraulic loop example, where changing component's operational status leads to changes in the probabilistic models. The supervisory control system then identifies, evaluates, and implements optimum operational decisions.

In the next phase of the project, new features and functionalities will be implemented to recognize complex operational scenarios, such as change of equipment status, and ability to rank and evaluate multiple valid options.

ABSTRACT

This technical report was generated as a product of the Supervisory Control for Multi-Modular Small Modular Reactor (SMR) Plants project within the Instrumentation, Control and Human-Machine Interface technology area under the Advanced Small Modular Reactor (AdvSMR) Research and Development Program of the US Department of Energy. The report documents the definition of strategies, functional elements, and the structural architecture of a supervisory control system for multi-modular AdvSMR plants. This research activity advances the state of the art by incorporating real-time, probabilistic-based decision-making into the supervisory control system architectural layers through the introduction of a tiered-plant system approach. The report provides background information on the state of the art of automated decision-making, including the description of existing methodologies. It then presents a description of a *generalized decision-making framework*, upon which the supervisory control decision-making algorithm is based. The probabilistic portion of automated decision-making is demonstrated through a simple hydraulic loop example.

1. INTRODUCTION

This report documents an approach to integrating automated decision-making in to the supervisory control system. The reference example used in developing the communication links and computational capabilities of a supervisory control system shows the successful merging of the system layout, structure, and capabilities that were specified in previous milestone reports [1-1, 1-2, 1-3].

The introductory chapter provides information on how key results from previous phases of the project are implemented into the supervisory control system model. Chapter 2 provides background information on analytical tools and methods available for automated decision-making. Chapter 3 introduces a *generalized decision-making framework*, which is used as the foundation for developing the supervisory control system for an Advanced Small Modular Reactor (AdvSMR). Chapter 4 discusses the functional requirements for the supervisory control system, whereas Chapter 5 discusses the functional requirements for automated decision-making, and provides an implementation of the probabilistic portion of the decision-making process. Chapter 6 provides a demonstration of the process with an example. Future work, as discussed in Chapter 7, will build-out the capabilities of the supervisory control system to address more complex problems including components that are out of service, degraded states, prognostics and diagnostics, and of course, multiple reactors in a module.

1.1 TAXONOMY FOR AUTONOMOUS DECISION-MAKING

Automated decision-making methods can be categorized in a number of ways. One possible breakdown deals with treatment of time and temporal variations in system behavior. In this sense, automated decision-making methods can be broken into two major categories: (1) static (or off-line) decision-making methods, and (2) dynamic (real-time or online) decision-making methods. In a sense, the static versus dynamic distinction is more in reference to the type of environment and system for which a decision is being made.

Static methods refer to a single-pass process that is performed only once in automated decision-making—similar to open-loop system configurations. These methods likely require an extensive search wherever outcomes of possible decision alternatives are rigorously analyzed. Typically occurring at the beginning of a complex design or a major investment decision, the decision-making process is not repeated once the action is taken. An obvious example is the decision support tools used for large investments, e.g., to identify an appropriate geological site for building a facility. These support tools have been around and have been used by private investors, corporations, local administrations as well as governments. The nature of the investment does not permit refinement of decisions once the action is taken because of the prohibitive cost associated with alteration of course. Static methods are the most common decision-making methods identified in a survey of methods used in military, government, administration, business, and engineering.

Current state-of-the-art decision-making modules are static in that all possible decisions have been analyzed a priori. In contrast, a real-time decision module must be able to account for component failures and system faults while they occur. Thus, a Supervisory Control System requires a real-time response to evaluate plant conditions and equipment failures/faults while the plant configuration changes because of maintenance and outages, as they occur.

Static decisions are simple, conventional, one-time decisions that calculate the system in equilibrium, and thus are time-invariant. Current state-of-the-art decision-making modules used in Integrated Control Systems are only static in that any possible decision for given inputs have been previously analyzed. However, decisions may be revisited multiple times or perhaps continuously.

The design process itself is a static decision-making tool. That is, the design is frozen when it is analyzed for various metrics of interest such as availability, reliability, and capability.

Dynamic methods work analogous to closed-loop systems, in which the system output is continuously monitored; the system output and internal states are exclusively used for the next decision, or correct the course of action to achieve a specified objective—decisions take place in a time varying environment due either to the effect of previous actions of the decision maker or to exogenous events. Dynamic decisions, or real-time decisions, unlike static decisions, are typically complex and occur in real-time. Thus, dynamic decisions account for time-dependent changes in the state of the system.

The ORNL survey identified limited engineering applications that work in such a recursive manner. An example of dynamic decision-making is vehicular route planning and execution. A common decision-making tool used in this fashion is the expert system.

Capabilities of dynamic decision-making in a supervisory control system include:

- identify multiple failures/faults/outages simultaneously,
- identify failures on a real-time basis,
- identify problems for which a priori patterns have not been constructed, and
- change or modify a decision based on newly evolving conditions.

Alternatively, decision-making systems can be categorized in terms of treatment of uncertainties in the environment, as well as the inherent uncertainties associated with functioning of the system. In this sense, automated decision-making methods and tools are in three major categories: (1) probabilistic (or risk-based) methods, (2) deterministic (or mechanistic) methods, and (3) risk-informed methods, which combine the first and the second tools.

Nuclear power plants cannot operate outside known and understood safety limits, which place restrictions on the creation of new (not previously reviewed) action steps. Neither can plants be allowed to operate outside certified regulatory (NRC) limits. Any decision-making process, then, must recognize that limits for specific plant parameters are clearly set.

All data are known beforehand for realizing a deterministic analysis—such an analysis is prefaced by knowing what is going to happen next with little or no uncertainty. However, for real systems, there is always the possibility of not achieving the design objective, i.e., to ensure that the system performs satisfactorily within a specified time period. Thus, system and equipment designs rely on safety margins to reduce the risk of adverse performance. The weakness of deterministic decision-making is that it cannot inherently account for the stochastic nature of system behavior, or of component failures.

Decision-making based on a probabilistic analysis introduces the element of chance, in which variable states are not described by unique values, but rather by probability distributions. The ensuing risk assessments become essentially a decision-making process, often between competing interests, that provides insight as to whether the risks are, or are not, being adequately controlled.

Risk-based decision-making is a process that organizes information regarding the risk probabilities for one or more unwanted outcomes into a broad, orderly structure that helps decision makers make more informed choices. A risk-based decision-making process uses only probabilities to select the action to be taken. Addressing the practical need of supervisory control is risk-informed decision-making. Risk-informed decision-making uses risk assessments as an input (but not exclusively) to decision-making. Other factors, which may themselves be deterministic, are also parts of the decision-making process.

The supervisory control system under development will use a risk-informed decision-making process wherein the probabilities are coupled to characteristics such as magnitude of the response, rate of recovery, and secondary effects of the action.

1.2 FOUNDATIONS DEVELOPED IN PREVIOUS REPORTS SUCCESSFULLY IMPLEMENTED

Before the development of the actual supervisory control system could begin, the functional requirements, capabilities, and architecture of the system had to be determined. How these requirements could be implemented were reviewed, analyzed, and selected. A brief summary of the foundations or building blocks of the supervisory control system are provided below.

1.2.1 Supervisory Control System Architecture

Previous milestone reports on supervisory control discussed the structure of hierarchy for control. Because this report details the successful implementation of a supervisory control system based on the topology discussed in earlier reports, a summary is provided below. With this architecture, the supervisory control system can evaluate operational alternatives and select the best option at the single reactor level; future efforts will be to evaluate more complex problems, including decisions made at a reactor module level.

The supervisory control system is divided into three layers for control as shown in Fig. 1. The supervisory control at the organization layer (layer 1) provides control for the power blocks in the coordination layer (layer 2) and the reactor modules in the functional layer (layer 3).

The sample problem successfully showed the ability to probabilistically/deterministically evaluate control options and demonstrate the communications between the coordination layer and the functional layer.

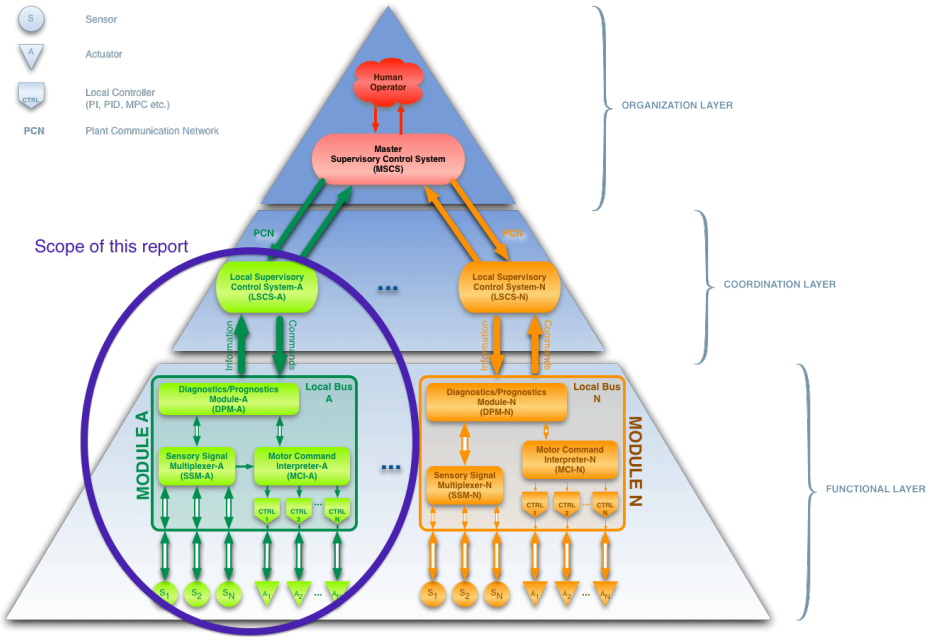

Fig. 1. Scope of this report within the supervisory control system architecture.

1.2.2 System-Level Functional Taxonomy

Previous milestone reports on supervisory control discussed the system-level functional taxonomy for control, which is an essential step to create interface descriptions for the supervisory control system.

The architecture for the supervisory control system divided the plant systems into three tiers based on their functions:

1. Tier-I systems,
2. Tier-II systems, and
3. Tier-III systems.

Tier-I systems are *directly* involved in the heat transport path from the reactor (heat source) to the ultimate heat sink (UHS). The UHS can be a river, lake, sea, or ocean, which is the typical heat sink. It can also be a passive heat dissipation mode that allows heat exchange to the air. Tier-I functions are those performed by Tier-I systems.

The classification of Tier-I system encompasses safety systems, safety-related systems, and non-safety-related systems. In many cases, Tier-I systems may have redundant components to perform their assigned functions to reduce the probability of failure. Some of these functions may be performed by diverse systems to minimize common-mode failures.

Tier-II systems *directly* provide support functions for Tier-I systems. Similarly, Tier-II functions are those performed by Tier-II systems. Tier-II systems and functions have particular significance for the supervisory control system: Systems in this tier provide necessary actuation interfaces for event-based control, such as taking a pump off-line while commencing a start-up sequence for a backup pump, or isolating a main flow pipe using an isolation valve and establishing an auxiliary flow path. They also provide additional sensory information for fault diagnostics to establish a holistic status of plant condition based on the health status of critical components.

Tier-III systems provide common services that supply bulk materials, energy, or data to the Tier-I and Tier-II systems. Tier-III functions are those performed by Tier-III systems. Examples of Tier-III systems include plant electrical, service water, gas supply (argon, helium, nitrogen, compressed air and instrument air), and auxiliary steam supply.

The distinction between Tier-II and Tier-III systems may be obscure for certain systems. The key distinction of a Tier-III system lies in the fact that it does not offer any control options for the operator in the event of loss of availability or reduced performance.

The modular-designed, multi-unit plants have more and stronger dependencies among systems than primarily single-unit plants at a common site. In fact, the design philosophy of the modular multi-unit plants is to form a single power plant station with respect to power generation and control. This philosophy is readily apparent with the single turbine-generator shared among three reactor modules for the ALMR PRISM power block.

Stand-alone units at multi-unit sites commonly share Tier II (support) and Tier III (utility) systems. However, because of the increased sharing of systems between reactor modules, some Tier I (heat removal) systems may be shared at AdvSMRs. This introduces new management and control criteria at Layer 1 and Layer 2 of the supervisory control system.

As shown in previous subsections of this report, the operation and health of the Tiered systems are successfully captured in the fault tree/event tree (FT/ET) models. The supervisory control system recognized the failures in the major plant systems, identifies alternatives for maintaining operation, and selects the best option based on probabilistic/deterministic criteria.

1.2.3 Graded Autonomy

Previous milestone reports on supervisory control discussed the divisions of the integrated control system (ICS), the supervisory control system (SCS), and the reactor protection system (RPS); trip setpoints, and operator involvement. Simply stated, the supervisory control system strives to maintain plant parameters from reaching trip setpoints.

The method is based on a hierarchically structured control system. At the top of the pyramid are the RPS setpoints. Feeding into the RPS setpoints are those conditions or variables that can be controlled to drive the system out of the degraded region back into the homeostatic region. These in turn lead to systems and components that can be controlled via local controllers. For example, a high outlet temperature from the reactor core can be lowered by decreasing power, reducing the coolant inlet temperature, and increasing secondary side flow rate. Each of these can be adjusted using plant controls. Inserting the control rods, increasing coolant flow, etc., are means to reduce core thermal power.

The question to be answered is

"What is the appropriate level of automation for an advanced SMR?"

The exact degree of autonomy is a design decision. The Human Machine Interface (HMI) functions provide the operator with proper interfaces to guide and direct the control system to operate in the proper modes. The HMI will provide key summary information to the operators in a clear manner. Large systems are prone to large quantities of HMI information such as alarms that must be properly organized and managed. Alarm management is significant task for large hierarchal systems. Alarms must be properly classified to their severity and time response requirements to discriminate between long-term maintenance items and critical items demanding immediate attention. Fig. 2 illustrates graphically the relationship of alarm categories. As can be seen, as the system moves away from the nominal state space, importance of status indications increases from *alerts* to *alarms*.[1]

If the system parameters progress into the degraded region of control, operator awareness and involvement increases. The three levels of operator involvement, based on the scale of degrees of automation [1-5], are

1. Nominal operator range: The computer decides everything and acts autonomously, ignoring the operator. That is, no operator intervention; status information provided to operator.

2. Alerts: The computer determines a complete set of action alternatives, selects one, and executes automatically, then necessarily informs the operator

3. Operator alarm: The computer determines a complete set of action alternatives, selects one, and executes that suggestion if the operator approves

[1] An alert is a notification to be watchful and is not to be considered the same priority as an alarm. An alarm indicates if and when the value (or rate of change value) of a measured or initiating variable is out of limits, has changed from a safe to unsafe condition, and/or has changed from a normal to an abnormal operating state or condition.

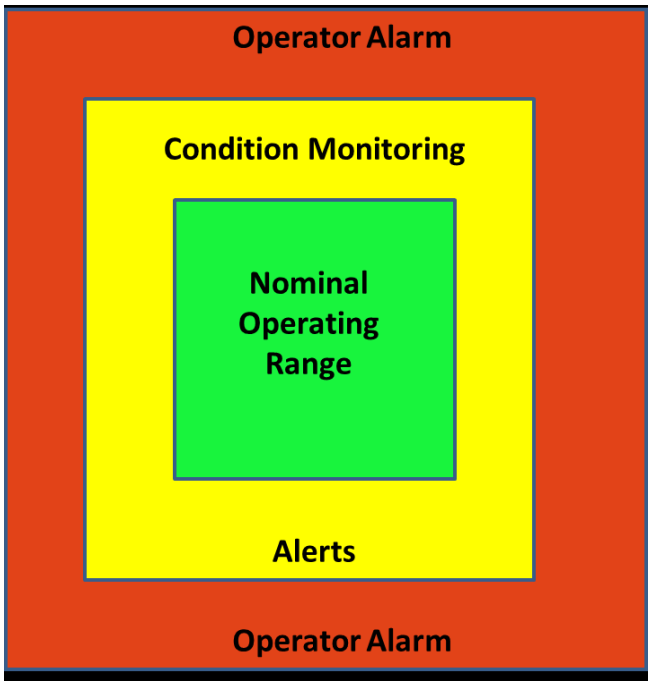

Fig. 2. Graphic description of the relationship of alarm categories. [Adopted from Ref. 1-4]

1.3 FUTURE EFFORTS

Future efforts in the development of the supervisory control system will involve programming the supervisory control system to recognize more complex systems with several control options given a component failure. With multiple options, the programming must allow the control system to select an optimal control decision. In addition, the programming must be expanded to allow the system to differentiate between options if a component is out of service in one of the options.

In addition, the ability to differentiate different options based on power level will be incorporated into the supervisory control system.

1.4 REFERENCES

1-1. S. M. Cetiner et al., *Definition of Architectural Structure for Supervisory Control System of Advanced Small Modular Reactors*, ORNL/TM-2013/320, August 2013.

1-2. S. M. Cetiner et al., *Technical Basis for Automated Decision-making A Survey on the State-of-the-Art of Decision-making and Existing Analytical Tools*, ORNL/LTR-2014/26, February 2014.

1-3. S. M. Cetiner et al., *Revised Functional Description of Supervisory Control for Advanced Small Modular Reactors*, ORNL/LTR-2014/213, June 2014.

1-4. B. Liptak, Instrument Engineers' Handbook, Fourth Edition, Volume Two: Process Control and Optimization, Liptak Associates, Stamford, Connecticut, USA, ISBN: 9780849310812, pp. 59–63.

1-5. T. B. Sheridan, Telerobotics, automation, and human supervisory control, The MIT Press, Cambridge, Massachusetts (1992).

2. BACKGROUND

Decision-making can be defined as a process that results in the selection of a particular course of action among several alternative scenarios [2-1]. A key element of decision-making is incorporation of existing information—generally called a priori information—and the subsequent analysis with the purpose of ascertaining its validity. The analysis specifies the performance measures, which provide the basis for determining how a particular course of action is to be assessed. Higher-level rules can be imposed to constrain the model outputs to avert undesirable or unacceptable course of actions. Finally, a number of criteria are established to select the best option leading to the resolution of the decision process.

The state-of-the-art of autonomous decision-making was surveyed in detail, and the results were published in an earlier milestone report [2-2]. Therefore, this background section is only intended to provide a high-level summary of this field. Further information can be found in Ref. 2-2.

2.1 DECISION-MAKING AS A RATIONAL PROCESS

Decision-making is one of the basic cognitive processes of human behaviors by which a preferred option or a course of actions is chosen from among a set of alternatives based on certain criteria [2-3, 2-4]. Decision theories are widely applied in a number of disciplines encompassing cognitive science, computer science, management science, economics, sociology, psychology, political science, statistics, engineering, business, and governments.

From an engineering standpoint, decision-making is a problem-solving activity to identify and analyze available course of actions, and to determine the most appropriate option given the set of conditions and constraints. The search is essentially terminated if and when a satisfactory solution is reached. The solution space can vastly differ depending on the nature of the problem being solved.

Drawing analogies of decision-making in other psychological, social and engineering fields is important because it helps create a framework by which a robust and consistent process can be developed.

It is necessary to make a distinction between an *automated process* and *autonomous process*. Automated process refers to a predetermined action or set of actions to reach a desired state given a condition or change in condition. Automation is widely used in almost every facet of our lives; but it does not in fact involve decision-making. Automation is merely a convenience that performs certain tasks in the case of a triggering event without human intervention. What is implied in an automated process is that all input states are assumed known. Therefore, uncertainties in monitored processes, unforeseen system states, or deteriorating conditions are not treated directly. However, potential implications of uncertainties can be incorporated into control system design, such as the case in *robust control*. Evidently, automation implies that a limited and relatively small set of actions—typically identified in a *decision table* or *logic table*—is considered given the input states with highest impact on output states.

However, as engineering systems and the processes got increasingly more complex with significantly higher degree of interconnectedness, designing automation systems that address a wide range of operating conditions and equipment availability becomes a challenging task. Furthermore, logic tables are usually constructed for nominal operating conditions, such as for 100%-capacity at steady state; hence, they are limited in terms of covering all possible necessary actions as a function of system status. Therefore, capabilities are needed to (1) diagnose a situation, (2) identify viable course of actions, and (3) determine the best, optimal or at least an acceptable action—or sequence of actions—to transition to a safe state. The process is called *decision-making*.

Autonomy is the ability of a system to determine and perform necessary tasks without human interaction. Decision-making capacity is the fundamental pillar of autonomy.

Decision-making, in a broad sense, is an expansion of automation capability whereby possible system states are either represented as a continuum, or with a highly refined discrete space, rather than a small number of states. Furthermore, uncertainties associated with processes or component statuses are taken into account explicitly as opposed to some bounding assumptions. Clearly, using a decision table becomes impracticable as the number of combinations of input states prohibits a feasible implementation of the logic. Further discussion of this aspect of decision-making is provided in Section 4.1.

2.2 ANALYTICAL METHODS AND TOOLS FOR DECISION-MAKING

Autonomy and decision-making have been a topic of research and development since the arrival of computing machines in the second half of the 20th century. While there were academic groups developing various methods and applications for robotics, the interest and need for autonomous systems gained a more focused direction with deep space missions, where intervention for course correction from the Earth clearly would not be conceivable.

This section provides a summary of methods and tools used in decision-making. A collection of automated decision-making examples from a number of industrial applications is provided in Appendix A.

2.2.1 Statistical Decision Theory

Statistical decision theory is concerned with the making of decisions in the presence of statistical knowledge, which sheds light on some of the uncertainties involved in the decision problem. These uncertainties can be considered to be unknown numerical quantities [2-5].

Classical statistics is directed towards the use of sample information, i.e., the data arising from the statistical investigation, in making inferences about their use. In contrast, decision theory attempts to combine the sample information with other relevant aspects of the problem with the intention of making the best decision.

In addition to sample information, two other types of information are typically relevant. The first is knowledge of possible consequences of decisions. Often this knowledge can be quantified by determining the loss that would be incurred for each possible decision and for various possible values of uncertainties. The incorporation of a loss function into statistical analysis was first studies extensively by Abraham Wald [2-6], which also gives a comprehensive bibliography of leading theorists and practitioners of the field before Wald.

The second source of non-sample information that is useful to consider is called prior information. This is information about uncertainty arising from sources other than statistical investigation. Generally, prior information comes from past experience about similar situations involving similar uncertainties—often called the base rate by economists.

The approach to statistics, which formally seeks to utilize prior information, is called Bayesian analysis, which was named after Bayes [2-7]. Bayesian analysis and decision theory go rather naturally together, partly because their common goal of utilizing non-experimental sources of information, and partly because of deep theoretical ties.

Existing mathematical models of decision-making relies on set theory. The axiom of choice—an axiom of set theory, which states that for every indexed family of nonempty sets, there exists an indexed family of elements. The axiom of choice was formulated by Ernst Zermelo to formalize his proof of the well-ordering theorem [2-8]. It is included in Zermelo-Fraenkel set theory with the axiom of choice extension, which is accepted as the standard form of axiomatic set theory.

Decision

A decision is a selected alternative from a non-empty set of alternatives based on a given set of criteria.

Decision-making

Decision-making is a process of a selecting a decision from available alternatives against chosen criteria for a given decision goal. Alternatively, decision-making can also be described as the process of constructing the choice criteria or choice function and associated strategies, and use them to select a decision from a set of possible alternatives.

In this view, existing decision theories with special mathematical tools provide a method to identify a proper choice function, to come up with an optimal or acceptable decision. Evidently, different decision-making methods and analytical tools generate different choice functions.

2.2.2 Bayesian Decision Theory

Bayesian decision theory can be considered as a subset of statistical decision-making. However, Bayesian approach is one of the most commonly referred mathematical methods that are exclusively used in decision-making processes in a wide range of applications.

In Bayesian decision theory, the choice function is called a decision rule [2-5, 2-6]. A loss function is adopted to evaluate the consequences of an action. Using the loss function for determining possible risks, a choice function is derived for decision-making.

A generic Bayesian decision process is shown in Fig. 3, which can be considered in two phases: inference phase and decision phase. In the inference phase, i.e. steps 1 through 4 in Fig. 3, posterior probabilities are obtained using the prior information (also called evidence) associated with the random processes used in the decision-making process. In the decision phase, possible decision alternatives are identified, and an optimal decision is determined based on the construct of the loss function in that the decision minimizes the expected loss over the posterior probabilities.

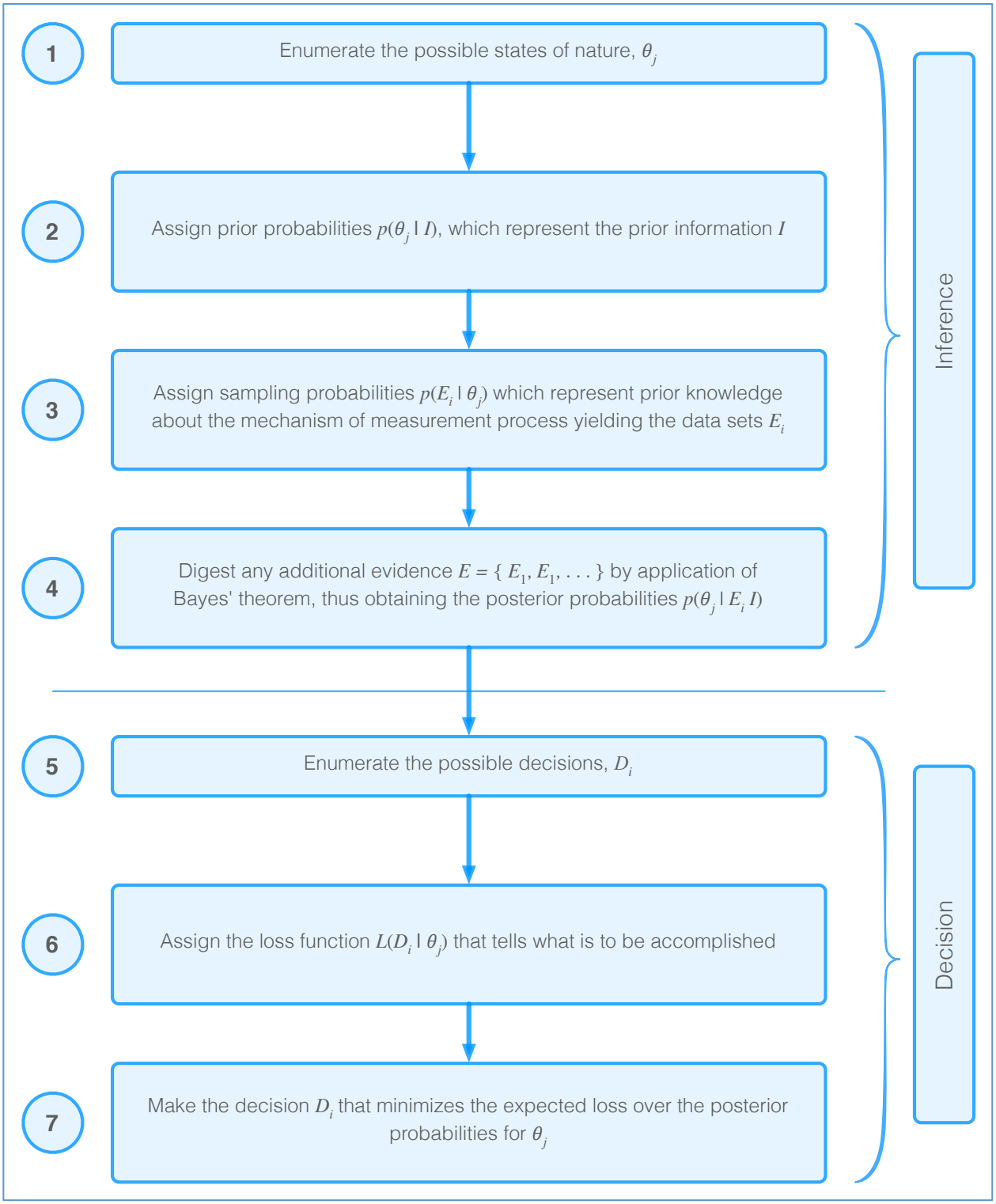

Fig. 3. High-level steps involved in a general decision-making process.

2.2.3 Expert Systems and Rule-Based Decision-Making

Plant operating procedures (OPs) are essentially rule-based decision modules executed by human operators. A rule-based model

- identifies the system state,
- associates the state with a task, and
- accesses stored rules to perform the task.

Operational limits and conditions (OLCs) are developed to ensure that the plant is operated in accordance with plant design assumptions and intent. OLCs also include actions to be taken and limitations to be observed by the operating personnel [2-9].

Operating procedures are developed for normal operation to ensure that the plant is operated within the OLCs and to provide instructions for the safe conduct of all modes of normal operation, such as starting up, power production, shutting down, shutdown, load changes, process monitoring and fuel handling.[2] Either event-based or symptom-based procedures are developed for abnormal conditions and design basis accidents.[3][2-10, 2-11]

Thus, the rule-based actions taken by operators to maintain plant parameters within OLCs are prescribed by the OPs. That is,

- The operator identifies the system state and which parameter is outside operating limits,
- The operator associates the system state with the appropriate OP, and
- The operator modifies the system state based on the rules in that OP.

The limits and conditions for normal operation include limits on operating parameters, stipulations for the minimum amount of operable equipment and staffing levels, prescribed actions to be taken by the operating staff in the event of deviations from the established OLCs and the time allowed to complete these actions. In addition, prescribed margins are used to ensure that normal operating values and the established safety system settings are avoided to prevent the actuation of safety systems.

Any action taken by the Supervisory Control System[4] must not diverge from the established OPs and cannot compromise established OLCs.

A means of automating the plant procedural system could be to implement the rules through decision tables. Decision tables, like flowcharts and if-then-else and switch-case statements, associate conditions with actions to perform. Each decision corresponds to a variable, relation or predicate whose possible

[2] Alarm response procedures are developed in support of the main OPs. The procedures ensure timely and correct response to deviations from the limits of steady state operation and ensure that the plant parameters are maintained within specified limits.

[3] For *event-based procedures*, the decisions and measures to respond to accidents are made on the basis of the state of the plant in relation to predefined events, which are considered in the design and safety analysis report. In using the event-based approach, the operator must identify the specific DBA before the recovery and/or mitigating operator actions have begun. In *symptom-based procedures*, the decisions for measures to respond to events are specified with respect to the symptoms and the state of systems of the plant (such as the values of safety parameters and critical safety functions). This allows the operator to maintain optimal operating characteristics without the need to be concerned with the continuing accident scenario. That is, system-based procedures will use parameters indicating the plant state to identify optimum recovery routes for the operator without the need for accident diagnosis.

[4] As a reminder, the Integrated Control System is used to maintain plant variables within operating limits and to prevent situations that could lead to accidents. The Supervisory Control System is used to maintain plant variables within operating limits given an AOO, component failure/fault, or MWe load change. Mitigating the consequences of an accident is outside the scope of supervisory control.

values are listed among the condition alternatives. Each action is a procedure or operation to be performed, and the entries specify whether (and in what order) the action is to be performed for the set of corresponding condition alternatives. Many decision tables include in their condition alternatives the "don't care" symbol (a hyphen). Using "don't cares" can simplify decision tables, especially when a given condition has little influence on the actions to be performed.

Decision tables vary widely in the way the condition alternatives and action entries are represented. Some decision tables use simple true/false values to represent the alternatives to a condition (akin to if-then-else), other tables may use numbered alternatives (akin to switch-case), and some tables even use fuzzy logic or probabilistic representations for condition alternatives. In a similar way, action entries can simply represent whether an action is to be performed (check the actions to perform), or in more advanced decision tables, the sequencing of actions to perform.

Decision tables can be, and often are, embedded within computer programs and used to *drive* the logic of the program. A simple example might be a *lookup table* containing a range of possible input values and a *function pointer* to the section of code to process that input.

2.2.4 Utility Theory

Utility theory was developed by economists to explain and predict human decision-making under risk and uncertainty. The fundamental assumption underlying utility theory is that the decision maker always chooses the alternative for which the expected value of the utility is maximized. Built into this assumption is a further supposition that a code of rationality is accepted and utilized by human decision-makers—thus making it possible to construct a mathematical representation that allows prediction of human behavior.

In traditional utility theory, a utility function is defined, which represents the sensibility of people to levels of wealth, i.e., the dissatisfaction of loss or satisfaction of gain. Utility functions, which are essentially transfer functions, are separately defined for situations of loss or gain because humans have uniquely different responses to loss and gain. These functions are typically represented using nonlinear relationships as shown in Fig. 4.

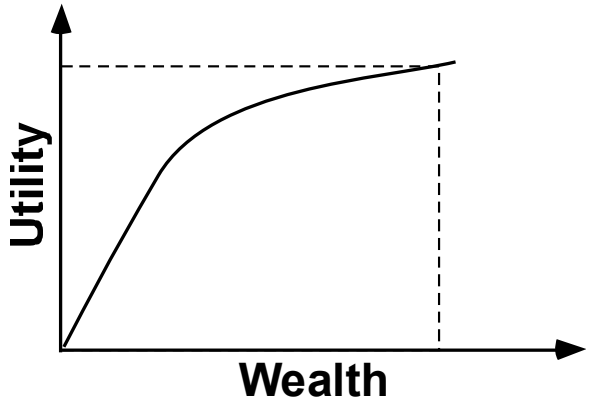

Fig. 4. Utility function for losses.

According to utility theory, the risk R is calculated using a relationship as shown in Eq. (2-1). This relationship, which includes both the utility related to the interested party and a probability of occurrence, effectively becomes the basis for many decision-making methodologies.

$$R = \mu_s(S)\,\mu_p(P) \tag{2-1}$$

where μ_S is the utility function of losses and μ_P is the utility function of possibilities. It becomes possible to compare alternative scenarios on the basis of loss, gain, and probability of occurrence using the risk values calculated from Eq. (2-1).

Concepts such as non-satiation, risk aversion, expected utility maximization, fair bets, certainty equivalents, market elasticity, and risk aversion are incorporated in numerous adaptations of this basic relationship described by utility theory. For example, a variant of the utility theory approach becomes the basis for portfolio optimization used by economists and investors.

The basic approach of utility theory as described above can become a foundational building block for a decision-making system intended for real-time supervisory control. Given a collection of (seemingly) viable alternative solutions, implementation risks determined for each alternative can be compared to find a minimum risk solution. Independent loss and gain (utility) functions as related to plant operating procedures or other decision strategies can be formulated and represented as nonlinear relationships as depicted in Fig. 4. Therefore, utility theory can be adapted as a probability-based decision-making method.

2.2.5 Multi-Attribute Utility Theory and Multi-Criteria Decision-Making

Constructing meaningful utility functions becomes progressively more complex as utility theory is broadened from trivial games of change (e.g., what is the preference ratio related to losing $50 or gaining $100 in a game of chance) to more complex applications such as the siting of nuclear facilities. Although the basic mathematical relationship of utility and probability remains the same (see Eq. (1)), an effective method of identifying complex utility functions and expressing them appropriately is needed. Various researchers have extended utility theory to a form that combines multiple attributes [2-12].

A typical approach to scoring of utility values is to normalize them so that dissimilar measures of performance, cost, and risk can be compared. For most situations, values are normalized to a dimensionless unity scale. Criteria are then weighted according to importance. To identify the preferred alternative, criteria are multiplied by each normalized alternative's utility score.

The typical way of analyzing decisions under uncertainty is to represent options and uncertainties as a decision tree and then select the option with the highest expected value. As an example, consider two mutually exclusive options A_1 and A_2 [2-13]. Their mono-criterion outcomes may vary due to events 1 and 2, respectively (see Fig. 5). If option A_1 were implemented, event 1 could generate either outcome $o_{1,1}$ (with probability $p_{1,1}$) or outcome $o_{1,2}$(with probability $p_{1,2}$). The probabilities of outcomes should sum up to unity. The option with the highest expected value (EV) should be selected.

Usually a multi-attribute utility function is employed to aggregate partial performances for multiple criteria. For example, if there were three criteria (C_1, C_2 and C_3) for assessing the performances of the two options represented in Fig. 5, each k^{th} criterion would have an x_k attribute measuring option performance, an associated u_k partial utility function, and a W_k weight, as illustrated in Fig. 6. If an A_i^{th} option were implemented, there would be three outcomes from each branch of the j^{th} event node $o_{i,j,k}$. Partial utility functions, U_k, would convert partial performances into partial utility and an overall utility function could be calculated. The option with the highest expected utility should be selected.

Table 1 lists a sequence of steps that comprise multi-attribute decision-making. This multi-attribute utility method of comparing alternatives is adaptable and entirely useful as a decision-making engine for supervisory control. Note that this method has close similarities to the weighted method of Kepner-Tregoe except that for the multi-attribute formulation the utility functions may be nonlinearly represented and a probability of occurrence is included.

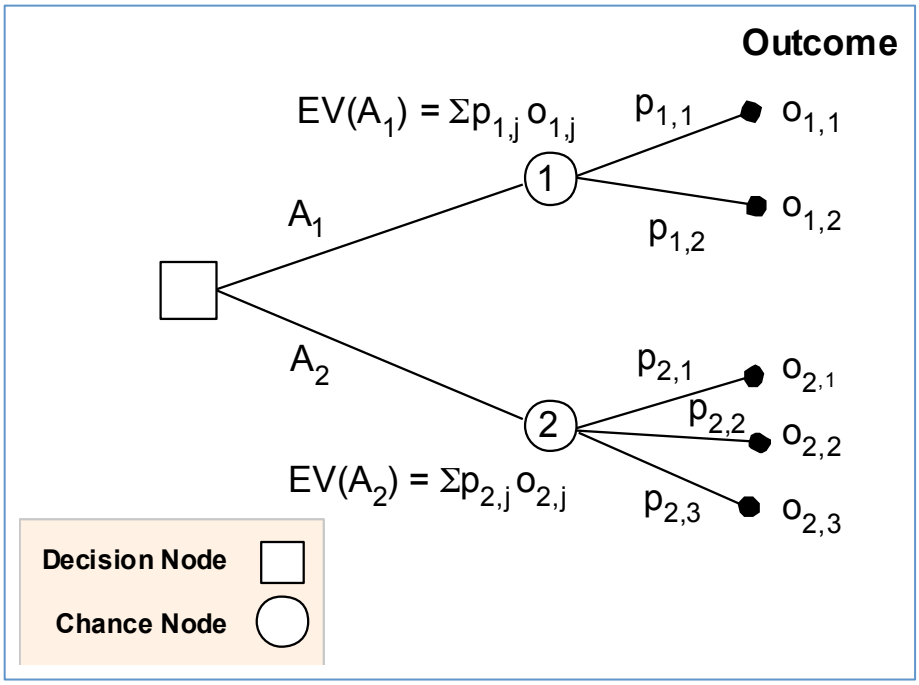

Fig. 5. Traditional decision analysis with decision trees.

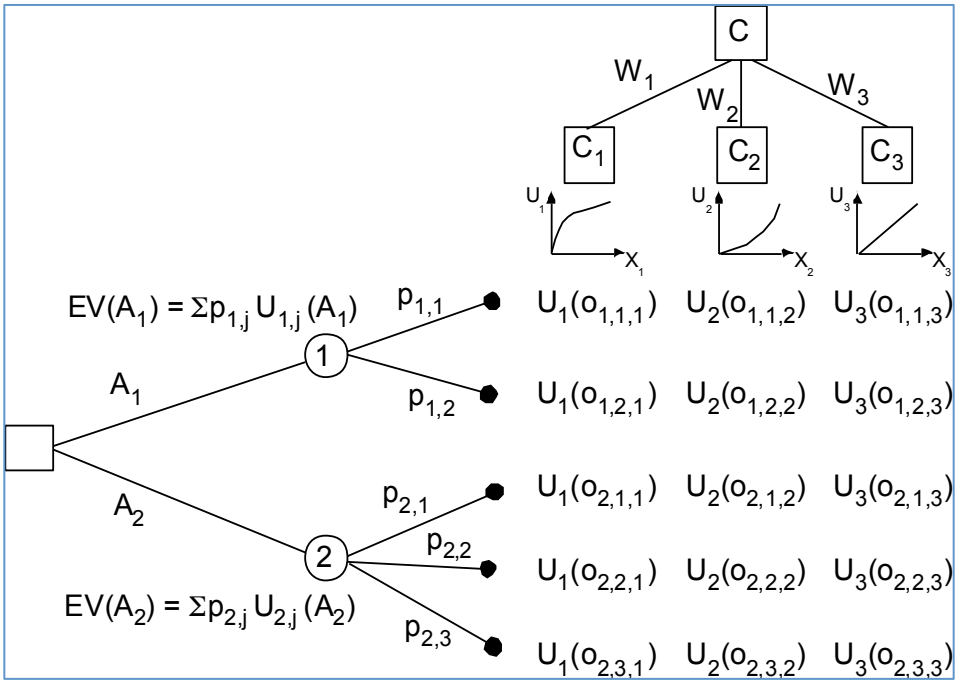

Fig. 6. Traditional decision analysis with a multi-attribute utility function.

Table 1. Logical steps to making a decision

Step	Task
1	Identify options
2	Identify possible outcomes of each option
3	Identify attributes with which to evaluate outcomes
4	Score each outcome on each attribute
5	Weight attributes
6	Aggregate scores and weights into utilities (MAU)
7	Identify events that determine which outcome will follow choice of an option
8	For each event, specify a prior distribution
9	Identify information that might modify the probabilities in Step 8
10	If information is free or cheap, buy it (Max SEU)
11	If information costs, find out how much
12	Determine the conditional gain from information purchase
13	Aggregate cost of information and gain from having it (Max SEU)
14	Decide whether to the buy the information (Max SEU + Bayes)
15	If information is bought, update prior probabilities (Bayes)
16	Back to Step 11: Iterate until no new information is bought (Max SEU)
17	Assemble the numbers output at Steps 6 and 15
18	Calculate expected utilities (Max SEU)
19	Choose the option with the highest expected utility (Max SEU)

MAU: Multi-attribute utility; Max SEU: Maximum subjectively expected utility

2.2.6 Analytical Hierarchy Process

Analytic Hierarchy Process (AHP) is a method to select a preferred alternative by comparing pairs based on performance against criteria [2-14]. The justification for the pair-wise comparison is that humans (and groups of individuals) are better suited at making relative judgments between only a few items (two being optimal) rather than making absolute judgments involving many items. The hierarchical process comprises a systematic procedure that divides a problem into smaller constituent parts until a level is reached that permits pair-wise comparison judgments. Results of the judgments are converted to quantitative scores that drive a weighted comparison matrix. Each normalized alternative score is multiplied by a corresponding normalized criterion weight; the results are summed for all alternatives to identify the preferred alternative, which will have the highest total score.

Typically, the pair-wise comparisons are made using a nine-point scale (see Table 2). The AHP steps are organized as shown in Table 3 [2-15]. The structure can be predetermined and made implementable by a computation system rather than a human.

Table 2. Example pair-wise scaling for Analytic Hierarchy Process

Numerical Value	Qualitative Value
1	Equal importance or preference
3	Moderate importance or preference of one over another
5	Strong or essential importance or preference
7	Very strong or demonstrated importance or preference
9	Extreme importance or preference

Table 3. Analytic Hierarchy Process procedure

Step	Procedure
1	Define problem and determine type of knowledge required
2	Structure the decision hierarchy starting at the top with the decision goal, next structure broad perspective objectives through intermediate levels to the lowest level
3	Construct pairwise comparison matrices. (Each upper level element is used to compare the elements in the level immediately below it.)
4	Priorities obtained from the comparisons are employed to weigh the priorities in the level immediately below—this is performed for every element. For each element in the level below add its weighed values to obtain its overall or global priority. This process of weighing and adding is continued until the final priorities of the alternatives in the bottom most level are obtained.

However, a limitation of the AHP approach is the use of pair-wise comparisons, which is also its benefit at least to human decision-making. The limitation is that extra effort is required to sub-divide the problem space into a hierarchy that leads to comparison pairs at the bottom. For a software-based system, in which the biases and limitations of human decision-making are not pertinent, more efficient methods than pair-wise comparisons can be employed. As an example, a larger collection of alternatives can be simultaneously compared using weighted functions as in the Kepner-Tregoe method or other variants of utility theory. AHP is also less flexible than either Kepner-Tregoe or Multi-Attribute Utility Theory because of the expanding size of AHP matrices especially as newly discovered alternatives or criteria must be considered. A possible selection anomaly arises from the pair-wise method because its *relative* measurement offers no guide to the outcome of manipulations based on combining different measurements from a standard scale (e.g., cost in dollars). For a software based inference engine, the AHP approach may not offer any special benefits over other decision-making approaches.

2.2.7 Probabilistic Risk Analysis

However, rather than evaluating risk as an abstract or independent activity, the Supervisory Control System can employ probabilistic techniques coupled with decision-making modules to determine which control action has the greatest likelihood of averting a challenge to a safety system. Thus, similar to a risk-informed approach, the Supervisory Control System poses the following questions:

- What component failure/fault or plant transient caused a change in plant variables of concern?
- What recovery actions are available to prevent a challenge to a safety system given the current status of the plant, taking into consideration diagnostic/prognostic monitoring (i.e., likely future challenges)?
- Which recovery action is most likely to prevent a challenge to a safety system?

Decision modules, based on this risk-informed approach, can dynamically provide plant control on a real-time basis for actual plant configurations. This benefit is available because the Supervisory Control System estimates the likelihood of challenging a safety system given a component failure/fault or plant transient.

The purpose of probabilistically informed decision-making is to provide information to the Supervisory Control System decision-making module to determine the best response based on unit, module, and plant needs. Probabilistically informed decision-making (like risk-informed decision-making) can add value to almost any situation. The possibility for one or more outcomes distinguishes probabilistically informed decision-making from more traditional decision-making.

Most decisions require more information than solely about risk. Such additional information includes:

- how far is the variable(s) of interest from the preferred setpoint corridor (magnitude of correction), and

- how fast a correction must be made (speed of correction).[5]

Many different probabilistic methods and tools are available. Choosing the appropriate method and using it effectively is important to successful implementation. Several factors are considered in selection of an appropriate tool. First, the Supervisory Control System will manage prevention of incidents through the ICS and its diagnostics/prognostics decision modules. Second, the response-related decision modules for the Supervisory Control System require real-time response to equipment failures and faults.

Licensing of nuclear power plants has been based on a deterministic approach and the principles of defense-in-depth. More recently, risk-informed insights are being used to complement the deterministic evaluations. The PRA methodology used to identify risk-informed insights in the nuclear arena is based on the FT/ET analysis techniques.

A deterministic approach asks the following questions:

- What can go wrong?
- What are the consequences?

A probabilistically-informed approach adds the following question in addition to the two questions listed above:

- How likely is it that something will go wrong?

Event trees (ETs) are used to logically develop the possible outcomes of an initiating event (IE) and use decision trees to create the models. The initiating events for the ETs are the occurrence of a failure with the potential to produce an undesired consequence. For the supervisory control system, the initiating events are plant parameters such as temperature and pressure that exceed allowable values set to ensure that set points for safety-related instrumentation are initially within and remain within the technical specification limits. The consequences of exceeding set-point values can result in a reactor trip, power reduction, or challenges to safety systems. Typical set points of interest for the RPS (and thus the IEs for the ETs), include:

[5] For example, changing pump speed on the secondary side will have a small, slow effect on changing the coolant temperature on the primary side. Similarly, changing the position of the control rods will have a large, rapid effect on changing the coolant temperature on the primary side. Thus, magnitude and speed can be important if the parameter of interest is close to, or moving rapidly toward, a reactor trip setpoint.

- reactor power,
- coolant flow rate,
- power-to-flow ratio,
- reactor outlet temperature,
- coolant level, and
- turbine status.

The branch points in the ET represent (usually) two potential outcomes when a line of assurance is challenged (i.e., a protective system or human action that may respond to the IE). Physical phenomena may also be represented as branch points.

Fault trees (FTs) linked to each branch of an ET model how logical relationships between equipment failures, human errors, and external events can combine to cause specific accidents. Fault Trees within the supervisory control system can reflect real-time plant status by indicating equipment out of service, equipment failures, probability of equipment failing, and human errors.

The accident sequences or scenarios are specific pathways through the ET from the IE to an undesired consequence. For the supervisory control system, the undesired consequences are challenges to safety systems, reactor trip, or power reduction. However, the ETs also show those sequences that, if followed, would lead to continued operation. Thus, the linked ET/FT model helps to identify not only key contributors to the event of interest, but also actions that can be taken to prevent challenges to safety systems.

A “risk-informed” approach represents a philosophy whereby risk insights are considered together with other factors to establish requirements that better focus attention on design and operational issues commensurate with their importance to health and safety. A “risk-informed” approach enhances the traditional approach by: (a) allowing explicit consideration of a broader set of potential challenges to safety, (b) providing a logical means for prioritizing these challenges based on risk significance, operating experience, and/or engineering judgment, (c) facilitating consideration of a broader set of resources to defend against these challenges, (d) explicitly identifying and quantifying sources of uncertainty in the analysis, and (e) leading to better decision-making by providing a means to test the sensitivity of the results to key assumptions.

The use of a probabilistically informed approach in a supervisory control system allows probabilistic insights to be coupled with other factors of concern such as magnitude from nominal set point, speed of parameter adjustment needed, etc. For example, a high outlet temperature from the reactor core can be lowered by decreasing power, reducing the coolant inlet temperature, or increasing secondary side flow rate. Each of these can be adjusted using plant controls. Inserting the control rods and increasing coolant flow are means to reduce core thermal power. Each control option has a different probability of success and can be linked to magnitude, speed, and other metrics of interest. That is, inserting the control rods will have a large, rapid effect on the output temperature while changing pump speed on a feedwater pump will have a small, slow effect.

A decision tree, like an event tree, is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences. Both decision and event trees are tools that are available to the decision maker, however, event trees allow the use of underlying fault trees to capture structure, system, and component failures and human errors.

Decision trees (and event trees) have several benefits such as the following:

- Are simple to understand and interpret

- Have value even with little hard data. Important insights can be generated based on experts describing a situation (its alternatives, probabilities, and costs) and their preferences for outcomes

- Possible scenarios can be added

- Worst, best and expected values can be determined for different scenarios

- Can be combined with other decision techniques.

A disadvantage of decision trees/event trees is that calculations can get very complex, particularly if many values are uncertain and/or if many outcomes are linked.

2.2.8 Discrete Event Systems

Many man-made devices and systems and some natural systems demonstrate only discrete values or outcomes. Man-made systems are governed by operational rules designed by humans. For example, man-made systems are often considered to be either on or off, enabled or disabled, running or stopped and so forth. These types of systems are best described as discrete event systems. Such discrete event systems are not easily analyzed and designed using conventional mathematics and engineering from time-driven processes (as represented by differential equations). Examples include transportation traffic systems, computer systems such as interrupts, communication systems, manufacturing processes, games, queuing systems and many man-made systems.

Discrete event dynamic systems (DEDS) or discrete event systems (DES) satisfy the properties (1) that state-space is a discrete set and (2) the state-transition mechanism is event-driven. Time in such systems is not the appropriate independent variable. Conventional differential equation approaches such as modern control theory do not apply to DES. They are described as [2-16]:

> *A class of dynamic systems characterized as synchronous or asynchronous occurrences of various discrete-valued events. Values are described by discrete values and transitions only occur at discrete points in time. Events are considered to occur instantaneously with some transition of one discrete value to another discrete value. These may be considered as time-driven or synchronous systems or event-drive or asynchronous systems.*

Alternatively, a formal description of this class of systems can be defined as [2-17]:

> *A Discrete Event System (DES) is a discrete-state, event-driven system, that is, its state evolution depends entirely on the occurrence of asynchronous discrete events over time.*

Modeling of DES behavior can include untimed, time or stochastic approaches. Automata and Petri Net formulations are traditional methods used for modeling DES behavior also using a state-transition structure.

Systems that combine DES with other dynamics such as time-driven (continuous-time or discrete-time) are called hybrid systems. Hybrid systems are widely demonstrated in many industries that combine process control with control logic in the hardware and software used to operate processes and machinery. The control logic may interact with a human operator, determine the proper operating mode, determine sequence steps, and also interact with the process control. Specific examples would include processing and containerizing food products. The empty containers go through a queuing process, which is an event-based system, and are filled using a time-driven process control system.

DES can be modeled and studied at three levels of abstraction: languages, timed languages, and stochastic timed languages. The term language is utilized due to ability to describe a set of events as an alphabet and the finite sequences of events as words or as combination of alphabet combinations. The language approach describes the logical behavior and all possible set combinations. Timed languages imply a deterministic behavior with a defined sequence of events with timing information. Stochastic timed languages include timed language information and additional statistical information, which makes it the most detailed language description type.

Modeling of DES is commonly performed using an automata approach or a petri net approach. These approaches use a state-transition structure to describe the possible events in each state of the system. These two approaches differ in how they represent state information. An automaton is a device that is capable of representing a language according to well-defined rules and is commonly represented using a state-transition diagram with a defined set of states, initial states, events, and state-transition functions.

An example untimed sequence or language is $\{e_1,\ e_2,\ e_3,\ e_4\}$ describing a specific sequence based on the system behavior or logic. An example timed sequence or timed language is $\{(e_1, t_1), (e_2, t_2)\ (e_3, t_3)\ (e_4, t_4)\}$ where event e_i occurs at time $t = t_i$.

The choice of one of the three levels of abstraction (languages, timed languages, and stochastic timed languages) depends on the system and the objectives of the analysis. If the analysis is interested in the logical behavior as the precise ordering of events or what states are valid or invalid, etc. the simple language approach is appropriate. In control system applications a set of paths may need to be determined to achieve a desired state or set of stats. The language approach can be used to pre-determine the desired set of paths in the logical behavior to achieve such desired states.

In some applications the timed language approach can be important to understand the timing of events, event transitions, and event paths. This approach can answer questions such as: "How soon can a particular state are reached given the current state?" or "Given a particular state, how soon can an undesirable state be reached?" The timed automata approach requires specific logical and timing information from a timed language description to answer questions about response time or throughput time. In other applications, the stochastic behavior must be included using probabilistic models in the stochastic timed languages abstraction. The language-based approach to discrete event modeling and analysis offers many benefits for understanding DES. A summary of the level of abstraction is shown in Table 4.

Table 4. Summary of DES language abstraction types.

Language Abstraction	Behavioral Aspect
Language	• Logical relationships • Possible states • Conditions or events to cause state transitions • Path to reach a desired or undesired state
Timed Language	• Time duration of a given state • Time to reach a desirable or undesirable state
Stochastic Timed Language	• Probability of a given state • Probability of a path to a desirable or undesirable state

Operations can be performed on these language sets using typical set operations such as union, intersection, difference, and complement. Other operations include concatenation, pre-fix closure, kleene closure, and post-language [2-17]. Projection operations are also performed on language sets.

A state transition automaton with internal states and outputs is called a *Moore automaton* (or Moore machine). A state transition automaton with internal states, inputs, and outputs is called a *Mealy automaton* (or Mealy machine). These automatons, which are well described in the literature, can be represented as an event set, $E = \{a, b, g\}$, and a state set, $X = \{x, y, z\}$.

An automaton that reaches a state, which will not permit any further events to execute is called a deadlock condition. This condition is also describing as a "blocked" condition because the system will enter the deadlock state without completing the task at hand. If a system contains a set of states with a local sequence or cycle but do not have a transition to exit the local sequence that situation is described as a *livelock condition*. In a livelock condition, the system is not deadlocked but is cycling between states and cannot exit the particular cycle. These potential locked conditions lead to the topic of safety properties, which deal with the subject of reachability of undesirable states and means to avoid blocked or livelock conditions.

An automaton can include non-deterministic behavior also. A nondeterministic automaton may demonstrate that for some conditions the state transition may have multiple outcomes. The primary source of non-determinism in a physical DES is limited sensory information, which will result in unobservable events that drive varying state transition outcomes [2-16 and 2-17].

A Petri net is a tool that treats manipulation of events according to specific rules. Often Petri net systems are conveniently described graphically as Petri net graphs (Fig. 7). An automaton can always be represented as a Petri net system. A Petri net system is defined by its graph or structure, the initial state, the set of marked states, and a state transition function. The graph contains places, transitions, and relationships to describe the system behavior. The state transition mechanism in Petri nets is provided when a transition condition is enabled and results in changing the state of the Petri net [2-17].

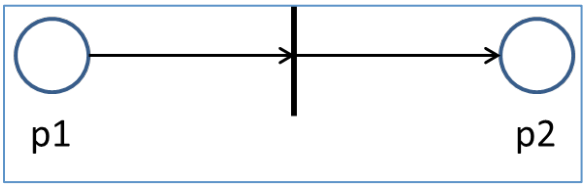

Fig. 7. An example Petri Net state transition diagram.

2.3 REFERENCES

2-1. S. Eilon, "What is a Decision," *Management Science*, **16**(4), pp. B-172–189 (1969).

2-2. S. M. Cetiner, D. L. Fugate, R. A. Kisner, M. D. Muhlheim, R. T. Wood, "Technical Basis for Automated Decision-Making: A Survey on the State of the Art of Decision-Making and Existing Analytical Tools," ORNL/LTR-2014/26 (SMR/ICHMI/ORNL/TR-2014/01), Oak Ridge National Laboratory, Oak Ridge, TN (2014).

2-3. Y. Wang, D. Liu, G. Ruhe, "Formal Description of the Cognitive Process of Decision-making," *Proc. of the Third IEEE Conference on Cognitive Informatics* (ICCI'04), August 16–17, 2004.

2-4. Y. Wang, G. Ruhe, "The Cognitive Process of Decision-making," *Int'l Journal of Cognitive Informatics and Natural Intelligence*, **1**(2), pp. 73–85 (2007).

2-5. J. O. Berger, "Statistical Decision Theory and Bayesian Analysis," Second Edition, *Springer Series in Statistics*, Springer-Verlag, New York (1985).

2-6. A. Wald, "Basic Ideas of a General Theory of Statistical Decision Rules," *Proc. of the International Congress of Mathematicians*, **1**, pp. 308–325 (1950).

2-7. T. Bayes, "An Essay Towards Solving a Problem in the Doctrine of Chances," *Philosophical Transactions of the Royal Society*, **53**, pp. 370–418 (1783).

2-8. E. Zermelo, "Beweis, dass jede Menge wohlgeordnet werden kann," *Mathematische Annalen*, **59** (4), pp. 514–16 (1904).

2-9. M. Drouin et al., *Glossary of Risk-Related Terms in Support of Risk-Informed Decisionmaking*, NUREG-2122, US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, D.C., November 2013.

2-10. International Atomic Energy Agency, "Safety of Nuclear Power Plants: Operation, Safety," Standards Series No. NS-R-2, Vienna, Austria, 2000.

2-11. International Atomic Energy Agency, "Operational Limits and Conditions and Operating Procedures for Nuclear Power Plants Safety Guide," Standards Series No. NS-G-2.2, Vienna, Austria, 2000.

2-12. R. L. Keeney, "The Art of Assessing Multiattribute Utility Functions," *Organizational Behavior and Human Performance*, **19**, pp. 267–310 (1977).

2-13. G. Montibeller and A. Franco, "Multi-Criteria Decision Analysis for Strategic Decision-making," C. Zopounidis and P.M. Pardalos (eds.), *Handbook of Multicriteria Analysis, Applied Optimization*, Springer-Verlag, Berlin Heidelberg (2010).

2-14. D. Baker, D. Bridges, R. Hunter, G. Johnson, J. Krupa, J. Murphy, and Ken Sorenson, *Guidebook to Decision-Making Methods*, DOE WSRC-IM-2002-00002, Dec. 2001.

2-15. T. L. Saaty, "Decision-making with the Analytic Hierarchy Process," *Int. J. Services Sciences*, **1**(1) (2008).

2-16. C. G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems* (Second Edition), Springer (2008].

2-17. R. David, H. Alla, *Discrete, Continuous, and Hybrid Petri Nets*, Springer (2010).

3. AUTONOMOUS DECISION-MAKING FRAMEWORK FOR SUPERVISORY CONTROL

Automation refers to the use of computing resources to make decisions, and implement a structured decision-making process without limited or no human intervention. The overriding goal of automation is to replace or supplement human *decision-makers* with reconfigurable *decision-making* modules that can perform a given set of tasks reliably.

The concept of automated decision-making is deceptively simple and intriguingly complex. From a narrow perspective, a decision is a choice among defined alternative courses of action. From a broader perspective, a decision involves the process of gathering and evaluating information about a situation; identifying a need for a decision; identifying or defining relevant alternative courses of action; choosing the *best*, the *most appropriate* or the *optimum* action; and then applying the solution and choice in the situation [3-1].

Generation of consistent *decisions* requires that a structured, coherent process be defined, which immediately leads to a *decision-making framework*.

This section introduces a generalized framework for autonomous decision-making that can be adopted and tailored to specific requirements for various applications. A specific implementation of this general framework is then provided for the proposed supervisory control system. This implementation is consistent with the overall system architecture defined in Ref. 3-2. Furthermore, this section introduces key performance indicators to assess the status or condition of the supervisory control system.

3.1 A GENERALIZED FRAMEWORK FOR AUTONOMOUS DECISION-MAKING

Decision-making is the process of identifying and choosing alternatives based on an agreed-upon set of metrics and preferences of the decision maker. Indirectly implied in *decision-making* is that there are *alternative options* to be considered. Each option offers a different approach or *path* to move from a given state or condition to a desired state or condition.

Ultimately, the objective of a *decision-making process* is to consider uncertainties, evaluate options, and finally assess potential consequences of a particular decision. Hence, it is quite possible that evaluation and assessment steps require consideration of multiple attributes of a system, components or elements of a system, or their future states, especially for large-scale complex systems, such as a nuclear power plant.

Baker et al. [3-3] suggested that a decision process involves eight logical steps:

1. Define problem,
2. Determine requirements,
3. Establish goals,
4. Identify alternatives,
5. Develop evaluation criteria,
6. Select a decision-making tool,
7. Select a preferred alternative, and
8. Validate solution.

While there are minor differences in the literature about the *necessary and sufficient steps* for decision-making, the *decision-making process* for the supervisory control system is based on *three fundamental elements*:

1. *Identification*—Identify decision alternatives

2. *Evaluation*—Evaluate alternative decisions

3. *Resolution*—Generate a *single solution* or a *single trajectory*, i.e., a collection of steps to finalize an action.

These elements, as illustrated in Fig. 8, define the *generalized autonomous decision-making framework*.

Contrasting with the steps identified in Ref. 3-3, the latter assumes that key steps, such as defining the problem, determining the requirements or developing evaluation criteria, are accomplished a priori, and are known parameters to the decision-making process.

Fig. 8. Elements of decision-making considered within the *Generalized Framework for Autonomous Decision-Making*.

The steps shown in Fig. 8 offers a generalized framework within which various decision-making methods can be implemented, and which can be applied for a variety of engineering problems.

3.2 PROPOSED AUTONOMOUS DECISION-MAKING FRAMEWORK FOR SUPERVISORY CONTROL

The generalized framework provides a conceptual structure that only includes abstract rules, elements, and relationships between them. Adoption of this framework for application to a supervisory control system requires that a specific implementation be created that defines *how* the individual objectives will be accomplished. This section provides a functional definition and some generic specifications for the proposed autonomous decision-making framework for a supervisory control system. Details of this implementation are given Chapter 5. Functionality of the architecture and its partial implementation are demonstrated in Chapter 6.

It should be noted that a fully specified control system is not within the scope of this work, nor would it be possible without detailed specification of entire plant systems. However, this study intends to provide clear guidance as to how autonomous decision-making can be accomplished—with consideration of applicable rules, regulations, guidance and operating experience.

3.2.1 High-Level Description of the Supervisory Control System

The supervisory control system will comply with the following high-level requirements:

1. The supervisory control system shall be implemented as a non-safety-related system.

2. The supervisory control system shall follow all the applicable rules and regulations regarding the separation and isolation of safety- and non-safety-related systems.

3. The supervisory control system shall not perform any safety-related function.

4. The supervisory control system shall not interfere with the functionality and operation of any safety system.

5. The supervisory control system shall not override operator directives.

These requirements are enforced to define the domain of operation of the supervisory control system. Implementing the supervisory control system as a non-safety-related system avoids undue regulatory burden on the vendor and the owner—especially considering the complexity of the system.

The fundamental assumption that goes into the design of the supervisory control system is that, if the supervisory control system fails to act during a transient, the safety system will eventually and independently initiate and bring the plant to a nominal or acceptable shutdown state.

3.2.2 Definition of Terms

The following terms are used throughout the report. A brief terminology is provided below to avoid misinterpretation, and maintain consistency.

Risk

In safety analysis, risk is defined as the product of frequency and consequence. However, in the context of the proposed supervisory control architecture and the autonomous decision-making framework, risk is defined as the probability of challenging a safety system, or probability of safety actuation.

Controllable Domain

A supervisory control system is required to support human decision-making under normal operating conditions, and make autonomous decisions. All of the possible states that the plant can assume constitute the controllable domain. The boundary of the controllable domain is primarily defined by the trip setpoints of the reactor protection system or the engineered safeguards features actuation system.

This domain is illustrated in light blue and orange colors in Fig. 9.

Challenge Surface

The surface of the controllable domain is called the challenge surface, beyond which a safety system actuation is warranted by the design of the plant.

The challenge surface is illustrated with the red line in Fig. 9.

Uncontrollable Domain

This is the domain outside the challenge surface of the plant state space.

The uncontrollable domain is illustrated in fading purple color in Fig. 9.

Probability of Departure from Controllable Domain

This metric is an indication of proximity of the plant state to the challenge surface. While there might be numerous ways to define this probability metric, it can be simply defined as a distance function between the current plant state and the closest point on the challenge surface. The closer the plant gets to the surface, the higher the probability of protection system actuation. Higher order moments of the states can also be considered, such as the rate of approach.

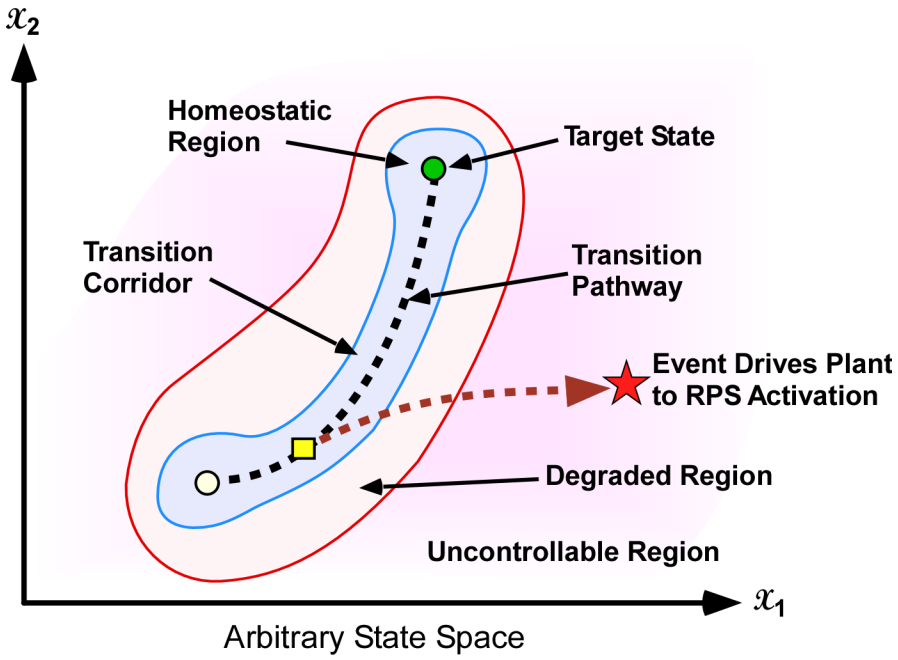

Fig. 9. Illustration of a conceptual state space formed by arbitrary state variables x_1 and x_2 for supervisory control.

3.2.3 Proposed Methods and Tools for Autonomous Decision-Making in Supervisory Control System

The proposed architecture for autonomous decision-making implements the general framework using two methods: (1) the probabilistic portion is implemented using the probabilistic risk analysis to identify decision options, and (2) the deterministic portion is implemented using utility theory to evaluate the alternatives identified by the probabilistic portion and to generate a single solution—i.e., the resolution of the autonomous decision-making process. This is shown in Fig. 10. The cost function for finding the optimal or desired decision is determined by the evaluation metric. Additional constraints, such as regulatory rules and operating guidelines, can be enforced in the deterministic evaluation phase.

Details of the implementation are presented in Chapter 5.

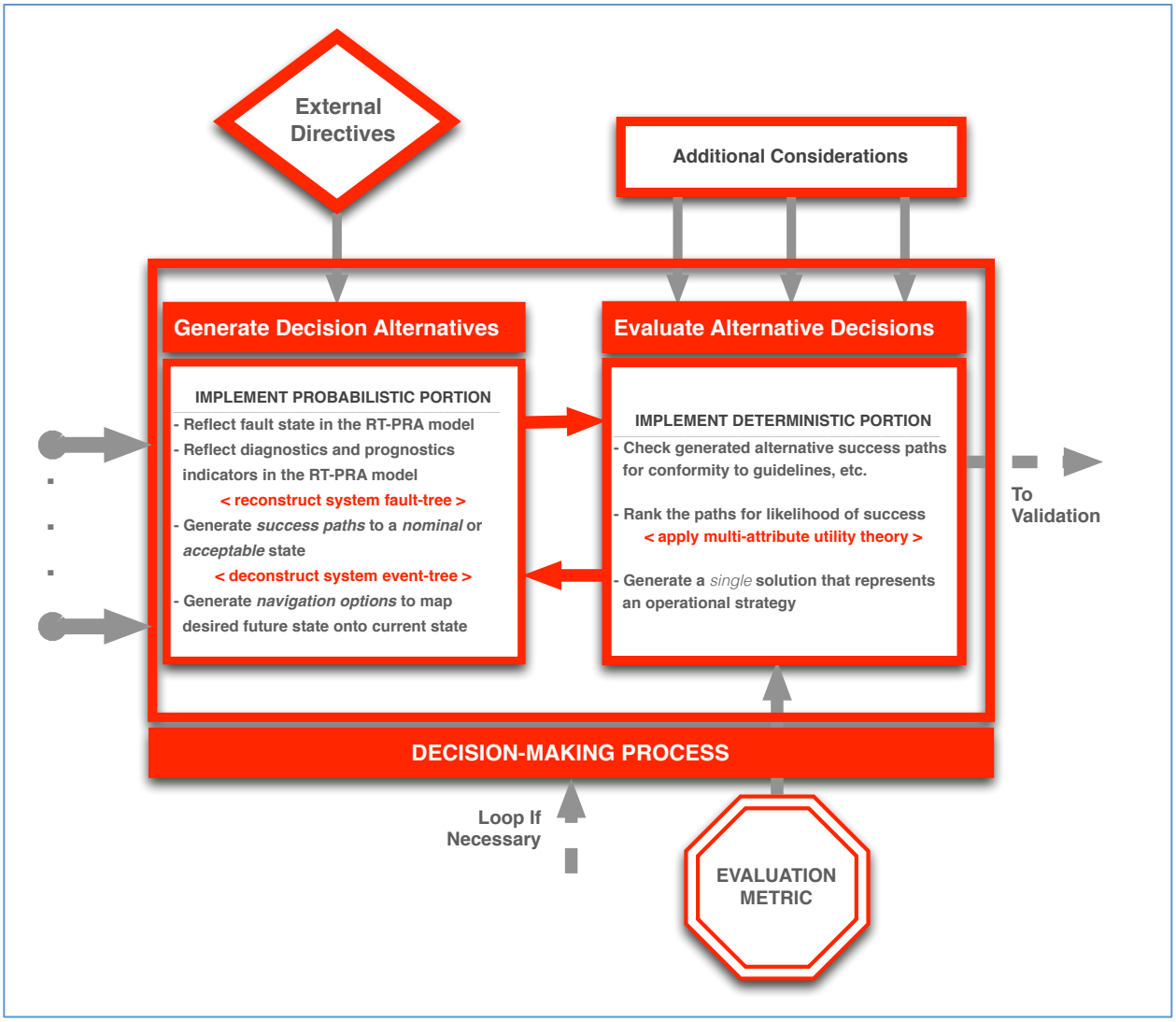

Fig. 10. The proposed framework for autonomous decision-making adopted for the supervisory control system.

3.3 REFERENCES

3-1. http://www.decisionautomation.com/

3-2. S. M. Cetiner et al., "Definition of Architectural Structure for Supervisory Control System of Advanced Small Modular Reactors," ORNL/TM-2013/32, SMR/ICHMI/ ORNL/TR-2013/04, August 2013.

3-3. D. Baker, D. Bridges, R. Hunter, G. Johnson, J. Krupa, J. Murphy, and Ken Sorenson, *Guidebook to Decision-Making Methods*, DOE WSRC-IM-2002-00002, Dec. 2001.

4. FUNCTIONAL DESCRIPTION OF SUPERVISORY CONTROL SYSTEM

An important motivation underlying the need for a supervisory control system for AdvSMRs is to increase plant automation level to reduce operator workload. For plants with multiple nuclear reactors comprising a single power generation system, the predisposition is to staff the plant at levels based on reactor module quantity versus total power output. Staffing at levels based on reactor module quantities results in prohibitive operating costs for AdvSMR concepts and does not necessarily improve safety. It is conceivable that operator workload could become overwhelming. Therefore, supervisory control research for AdvSMR concepts can offer increased levels of automation with improved desired reliability and availability, and reduced operating costs.

The supervisory control system is implemented as a non-safety-related system. It will have minimal mono-directional interactions with the reactor protection system. All safety systems, including the reactor protection system and the interlock systems, will be completely independent and isolated from the regular control and the supervisory control systems.

4.1 OBJECTIVES

The main objective of the supervisory control system is to increase the level of automation and to reduce the cognitive load on reactor operators by performing routine operator actions executed primarily during normal operations, and some actions performed during startup and shutdown. In addition to routine operator actions, the supervisory control system will intervene during off-normal conditions such as component failures or unexpected transients. The supervisory control system is not intended to replace the operator as the key decision node for safety-related actions, nor is it to support or complement protective actions performed by reactor protection or engineered safety features actuation systems. This objective and how it applies to future nuclear power plant concepts such as the AdvSMR is further defined in detail in Ref. 4-1.

4.2 SUPERVISORY CONTROL SYSTEM ARCHITECTURE

The proposed supervisory control system architecture from Ref. 4-1 is provided in Fig. 11. The figure consists of a hierarchical structure with three layers of abstraction going from organization to coordination and execution layer where low-level actions are performed based on the commands or directions from higher layers.

The master supervisory control system (MSCS) is responsible for coordination of system-level functions, that is, power and load allocation between reactor, power conversion, and process heat plants (coordination layer). Each local supervisory control system (LSCS) is responsible for functions within its assigned system (local supervision layer). LSCS monitors and analyzes processes and events within the system, and transmits module-level status information to the MSCS. It also analyzes fault indications from diagnostics and prognostics modules, and relays that information to the MSCS as module health status.

In the supervisory control architecture, the level of decision-making and supervision increases as one goes up the hierarchy shown in Fig. 11. In contrast, the level of activity and time urgency increases as one goes down the hierarchy as illustrated in Fig. 12.

In this report, *real time* is defined as being contemporaneous with the process and event control.

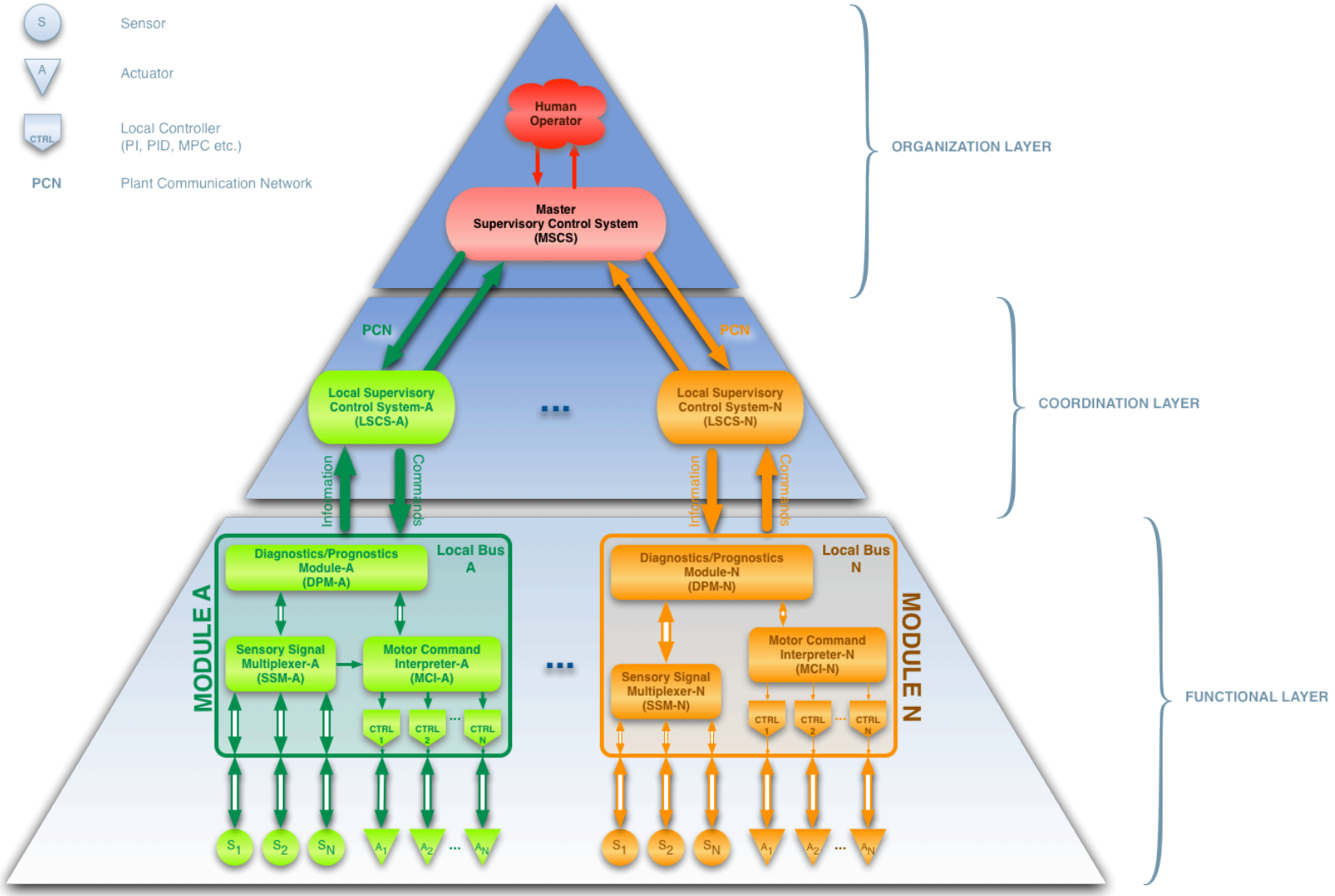

Fig. 11. Top-level system architecture for the supervisory control system.

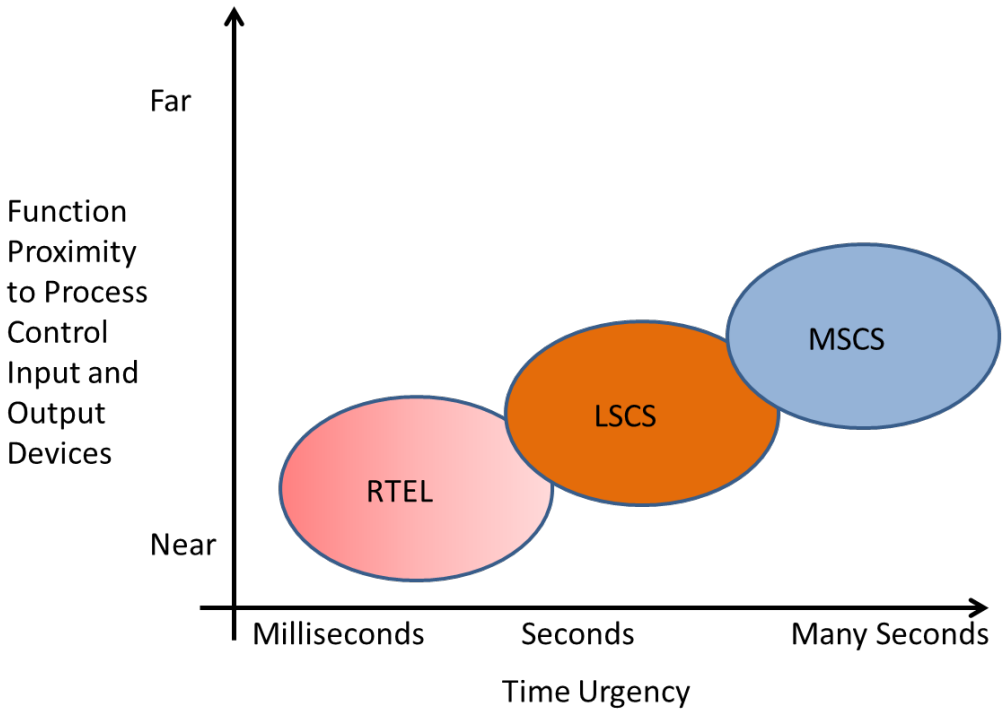

Fig. 12. Time urgency and functional proximity to process devices for RTEL, LSCS and MSCS.

Sensors, actuators, and controllers are at the real-time execution layer (RTEL) within their functional domain. Control elements may be as simple as proportional-integral-derivative (PID) controllers or may employ more sophisticated control algorithms such as model-predictive control, which is a design decision. The controllers are governed by the LSCS, including online adjustment of performance parameters.

To be useful, the supervisory control system must have options to be considered and evaluated on a real-time basis. Current practice is to have decision choices to be determined a priori and the decisions made using a "look-up" table. As the number of components increases, the use of such decision tables becomes intractable. For example, a system with 1000 components would require that all 1000 state changes be evaluated; combinations of two state changes, which provide that ability to track maintenance issues, require that 449,500 combinations of state changes be evaluated. The supervisory control system overcomes this insurmountable problem of a priori decision-making tables by using probabilistic risk assessment (PRA) techniques that can be updated to reflect actual operating conditions on a real-time basis. It should be noted that "risk" in this project scope is not the same as in the conventional PRA applications, where risk implies a potential release of radioactive material. For the supervisory control system application, the analogous risk metric is safety system actuation, where it is assumed that once actuated it will bring the plant to a safe state. While the underlying mathematics is the same for both applications, implications are significantly different.

The functional architecture of the supervisory control system and the proposed implementation of the autonomous decision-making framework are shown in Fig. 13.

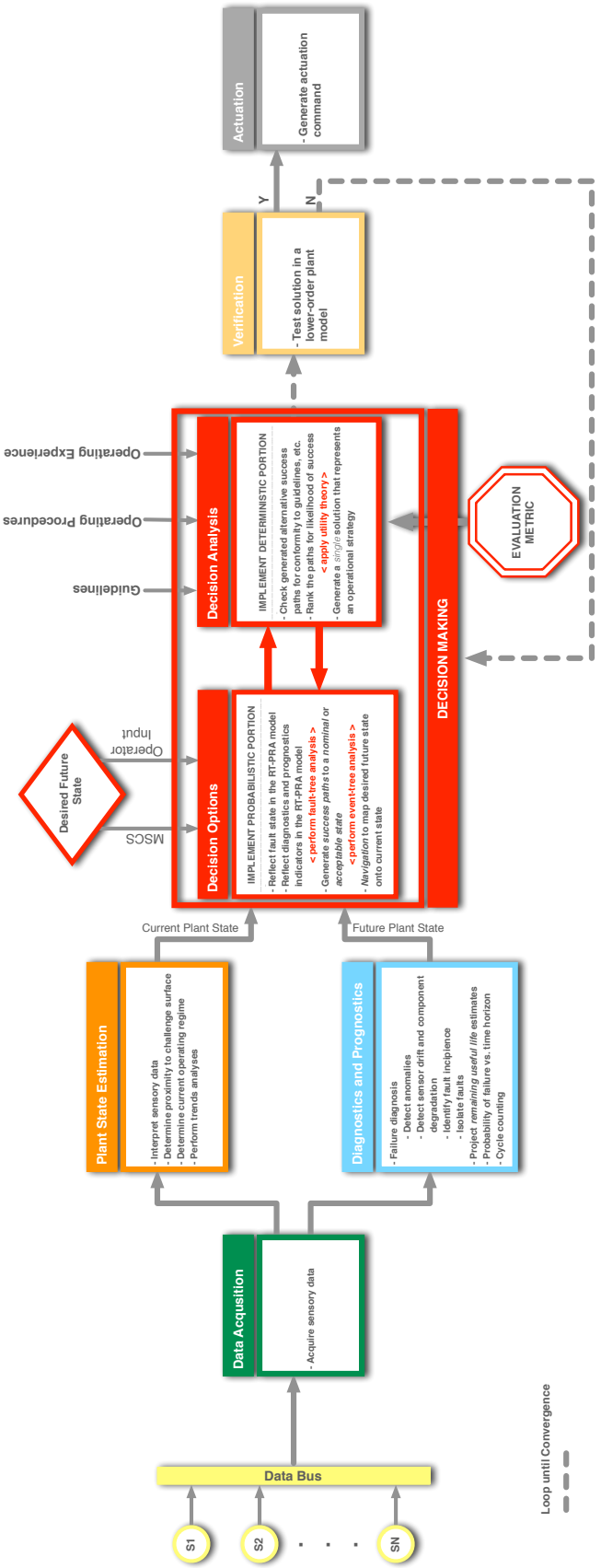

Fig. 13. Functional architecture of the supervisory control system with a specific implementation of the generalized decision-making framework.

4.3 FUNCTIONAL REQUIREMENTS FOR SUPERVISORY CONTROL

Functional requirements of a supervisory controller for multi-modular AdvSMRs are derived from a number of high-level rules and objectives to fulfill a multitude of requirements including safety, reliability, availability, maintainability, and performance with varying priority and weight. In a nuclear reactor and plant design, safety requirements always take precedence. The list of requirements can be expanded based on input from stakeholders. As stated earlier, the principal objective of incorporating a supervisory controller is to render the SMR business plan economically viable.

The set of functional requirements for the supervisory control system (high-level) as well as its individual modules was presented in Ref. 4-2. A summary list of requirements is provided in Appendix A.

4.4 REFERENCES

4-1. S. Cetiner, D. Cole, D. Fugate, R. Kisner, M. Kristufek, A. Melin, M. Muhlheim, N. Rao, R. Wood, *Definition of Architectural Structure for Supervisory Control System of Advanced Small Modular Reactors*, SMR/ICHMI/ORNL/TR-2013/04, August 2013.

4-2. S. Cetiner, D. Fugate, R. Kisner, M. Muhlheim, *Revised Functional Description of Supervisory Control for Advanced Small Modular Reactor*, ORNL/LTR-2014/213 (SMR/ICHMI/ORNL/TR-2014/04), Oak Ridge National Laboratory, Oak Ridge, TN, June 2014.

5. IMPLEMENTATION OF AUTOMATED DECISION-MAKING

Decision-making is as a process that generates a resulting *decision* based on collecting information, evaluating the available alternatives, and selecting the preferred alternative. Every decision-making process produces a final choice. The output of the decision-making process is generally an instruction that will be executed and turned into an action.

Based on plant operating status, component health, and equipment failures, the decision-making capabilities for the supervisory control system will use probabilistic analyses to identify a set of control options that, if taken, should prevent the actuation of the protection system. To determine the preferred option to be taken, the supervisory control system assesses each of the probabilistically determined operational alternatives against a set of deterministic criteria. Once the control option is selected, the supervisory control system transmits the necessary information to the controller of the component of interest for actuation and informs the operator of action taken or requests permission to take action.

5.1 PROBABILISTIC PORTION

There are many different methods and tools that can be used to perform probabilistic assessments; however, choosing the appropriate method is key to any successful program. The first step is to identify the functional requirements of the probabilistic method. After the requirements are identified, the analytical method and tools can be selected and any necessary automation tools developed. The completion of this stage signifies the transition from a theoretical problem to an application of the technology developed.

5.1.1 Functionality

To meet the objectives for the supervisory control system, the following requirements of the probabilistic tools will allow winnowing the selection of probabilistic techniques to be considered. Specifically, the probabilistic techniques must be able to

- address all component states (i.e., failed, out of service, degraded, operating),
- recognize changes in status for one or more components (up to all components) simultaneously,
- recognize changes in component status on a real-time basis (e.g., working to failed),
- recognize a change in probability of failure (e.g., $p = \lambda t$ to $p = 1.0$), and
- calculate different metrics of interest [i.e., measure the appropriate metric for the type of analysis being performed, such as core damage frequency (CDF), challenge to safety system setting, etc.].

Section 2 provides a review of the methods used in different applications of decision-making modules. Because linked FT/ET probabilistic analysis techniques can be used to evaluate the change of state for a component to be assessed (e.g., working to failed) but also allow combinations of component states to be evaluated simultaneously (e.g., component A fails, component B out of service or OOS), this technique was chosen for the decision module to be implemented in the supervisory control system.

Fault tree analysis (FTA) is an analysis technique that allows an analyst to systematically examine combinations of failures that are required to achieve an event defined as the "top event." (It appears at the top of the Fault Tree.) A top event may be any event to be investigated, such as loss of power or system failure. An FTA provides all the combinations of conditions that can bring about the top event and models

logical relationships between combinations of equipment failures and human errors. When incorporated within the supervisory control system, the FT models must be able to account for normal operation, component failures, degraded components, and components being out of service. Thus, the current status of components and their failure probabilities ($p = \lambda t$ or $p = 1.0$)[6] must be captured by the supervisory control system and must be able to be modified to reflect their operating status (i.e., from working to failed) on a real-time basis. The FT must also be able to reflect the degraded status of components, aging effects, or uncertainties in passive systems ($p = \lambda' t$).[7]

Event tree analysis (ETA) is a technique that logically develops the possible outcomes of an IE and provides a systematic framework to identify and qualitatively or quantitatively evaluate accident sequences. ETA is particularly useful for quantifying the frequencies of accident sequences where many events can affect the potential outcome of an accident. An ET begins with an initiating event followed by success or failure for important events that determine the accident sequence. Each path through the ET is an accident sequence. The accident sequences appear at the right side of the ET. The ET is quantified by a rate for the IE and success/failure probabilities for the various branches. In many probabilistic analyses, FTs are linked to ETs. Fault Tree Analysis can be used to quantify the probabilities for the ET branch points. Multiplying the initiator rates and branch probabilities together results in a rate for each accident sequence [5-1].

In most ETs, the success path is upward and the failure path is downward at each ET branch point. Although modeled the same in conventional ET models, the supervisory control system is focused on the success paths of the ETs. Contributors to the path (or sequence) of avoiding an accident include elements such as the successful implementation of changing the status of a component (e.g., pump started, valve opened) such that, in terms of the supervisory control system, operation continues.

For example, for a licensing basis event, the IE could be LOOP and following the failure paths in the last ET branch with CDF as the risk metric. For a supervisory control system, the IE could be "temperature too high" and following the success paths in the last ET branch with "trip avoidance" as the risk metric.

Within the control space, after an alarm or alert occurs, the operators acknowledge the alarm/alert and, based on procedures, take some action. The timescale for action by the supervisory control system must be comparable to that for operators. This timescale is referred to here as a "real-time" requirement.[8] Because of this real-time requirement, decision tables based on FT/ET analyses were evaluated as a method to probabilistically inform the supervisory control system.

Incorporating probabilistic analyses into a control system through the use of decision tables[9] is not new or unique. However, the use of decision tables requires that the probabilistic analyses be performed a priori

---

[6] λ is a component's failure rate (1/hr).

[7] λ' is a modified failure rate based on a components degraded status, aging effects, etc. (1/hr).

[8] The timescale for operator actions is less than the transit time of coolant in a primary cooling system.

[9] Decision tables require the actions for each of the conditions identified to be determined a priori. For example, the decision table for a printer troubleshooter shown below shows that if the printer does not print AND a red light is not flashing AND the printer is recognized (column 4 under Rules), the printer should be checked for a paper jam. Note that each condition must be evaluated individually to identify the actions to be taken. One benefit of using decision tables is that once the table is completed, any actions to be taken by the system can be implemented in real time.

and the results input into a decision table (aka "look-up" table). For example, a new EDG control system used risk-informed decision-making during the design process [5-2]. Probabilistic assessments were used to identify potentially critical situations and access the control system's response, allowing adjustments to the control logic to be made. These analyses were run repeatedly as the system and software was developed. Through the use of decision tables, the results from the probabilistic assessments for the EDG control system were accessed on a real-time basis; however, the probabilistic assessments themselves are fixed (i.e., static).

Decision tables are commonly used for real-time applications; however, for the use of decision tables to be practical, the number of possible combinations of component states must be realistic. Problematic in the use of decision tables is that all possible conditions (i.e., component failures) must be determined a priori. If only single-event failures are considered, the typically small number of actions (i.e., effects) can be identified and evaluated. However, combinations of equipment conditions, such as out of service for maintenance, failed state, degraded state, etc., should be included to more accurately reflect system/component status and operating conditions. The problem of more accurately accounting for the operating status of equipment is that the number of component combinations increases exponentially, as shown by the formula

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

where n is the number of components and k is the number of possible combinations of those components.

As an example of the difficulty in using a decision table, consider that the AP600 PRA modeled 1341component failures. This translates into 1341 conditions with each condition requiring an action to be predetermined. Modeling components out of service, in degraded states, etc., and then assessing the actions to be taken given a component failure makes the use of a decision table impractical. For the AP600 PRA, a single component failure with one component out of service for maintenance would require ~900,000 rules in the decision table, each of which must be individually evaluated [i.e., $\binom{1341}{2}$ possible combinations]. Incorporating the plant status (e.g., shutdown, refueling, low power to full power and selected power levels in between), which must also be assessed a priori, into a decision table is a direct multiplier for the number of $\binom{n}{k}$combinations. Thus, although decision tables are typically used in control systems, the accurate knowledge of the system configuration precludes their use.

The linked FT/ET models can provide the accurate knowledge of the system configuration. However, because of the real-time requirement of the supervisory control system, the probabilistic analysis using coupled FT/ET must be able to be properly reconfigured to reflect changes in component states

---

Printer troubleshooter

		Rules							
Conditions	Printer does not print	Y	Y	Y	Y	N	N	N	N
	A red light is flashing	Y	Y	N	N	Y	Y	N	N
	Printer is unrecognized	Y	N	Y	N	Y	N	Y	N
Actions	Check the power cable			X					
	Check the printer-computer cable	X		X					
	Ensure printer software is installed	X		X		X		X	
	Check/replace ink	X	X			X	X		
	Check for paper jam		X		X				

Decision table, from Wikipedia, http://en.wikipedia.org/wiki/Decision_table

automatically and in real time. That is, the functionality of implementing a probabilistic analysis tool into a supervisory control system is that all input, calculations, and output must be automatic and autonomous. Thus, the application of the probabilistic models into the supervisory control system requires a direct link between the supervisory control system and the probabilistic models. If this communication pathway can be established, linked FT/ET models will meet the functional requirements of the supervisory control system. Thus, although probabilistic assessments are typically static evaluations used to provide a priori results, a real-time assessment capability would provide analysis capabilities and insights not previously available.

The supervisory control system as configured using FT/ET models uses the power derived from the probabilistic tools to determine the likelihood of success for various control options for the current plant configuration. The results from the probabilistic evaluation are populated in a relational database and the supervisory control system reads the results to identify the control options.

5.1.2 Application

After determining that the linked FT/ET methodology with the results available though a relational database was the appropriate tool for meeting the functional requirements, the next step was to select a software package that would present the ability to access the code and probabilistic models through its dynamic link library (dll). Reliability Workbench (RWB) is a suite of reliability, safety, and maintainability software developed by Isograph that was chosen as the probabilistic tool to be used [5-3]. The use of the dll allows all of the model parameters, FT and ET topologies, and PRA-related analysis functions to be available without having to open the RWB application. That is, all modeling data can be accessed and changed through a relational database. This results in reducing the overhead associated with the amount of time required for the supervisory control system to identify and choose operational alternatives.

The RWB software package is verified and validated and is compliant with ISO 61508, “Safety Instrumented Systems” standard. This aspect of the software package gives confidence and assurance about the results of the reliability analysis, as it is a tool commonly used by industries that deal with safety-critical systems.

By communication to the PRA model through the dll, the supervisory control system will be able to account for potentially rapidly changing plant conditions during transients or accidents. Specifically, the requirements of the Supervisory Control System, as shown in Fig. 14, are to

1. recognize the change in state of a component (e.g., working or failed), (ITEM 1)
2. transmit the change of state to the RWB model (ITEMS 2, 3),
3. automatically adjust and execute the RWB models to reflect these changes, and
4. receive the analysis results from the RWB model (ITEM 4).

After the updated analysis results are transmitted back to the supervisory control system, the supervisory control system, through the use of a relational database, will

- automatically and autonomously identify operational alternatives ranked by the probability of successfully avoiding the actuation of a safety system setpoint,
- if more than one option is identified, select the preferred option based on deterministic criteria,
- transmit an actuation signal to the component of interest, and
- inform the operator of action taken or request permission to take action.

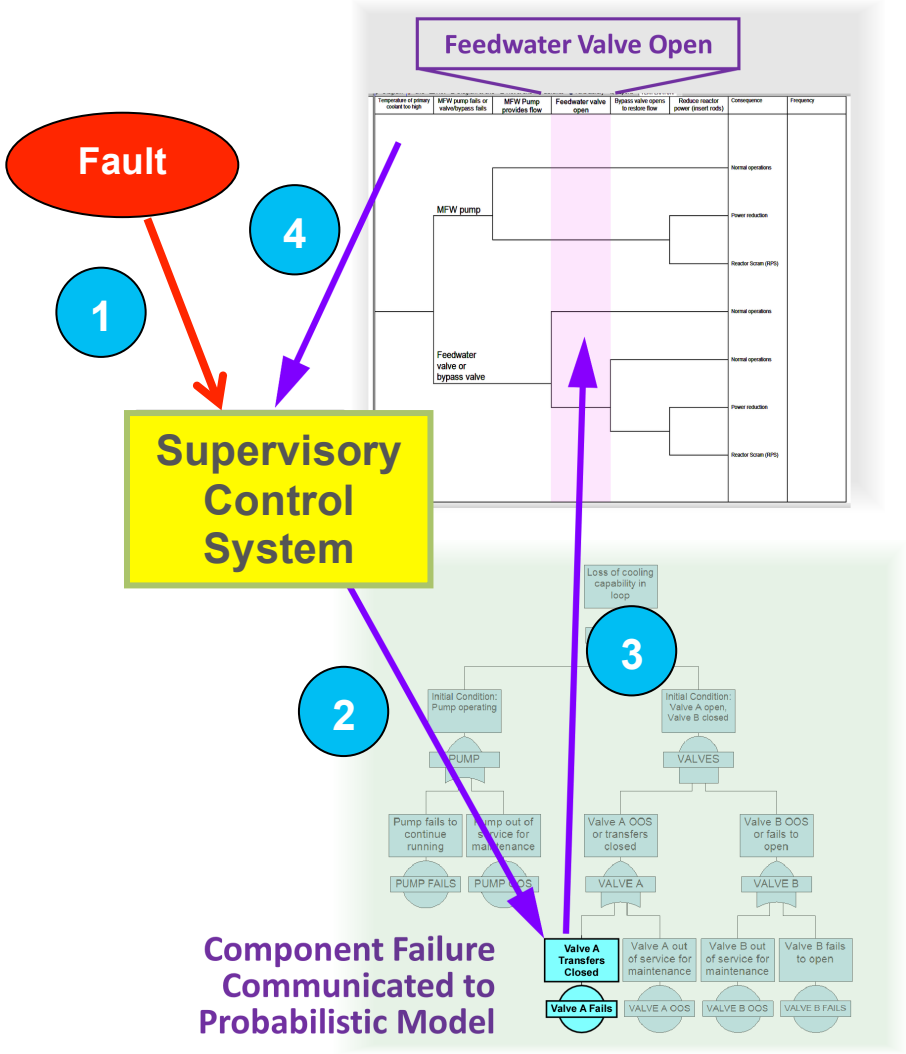

Fig. 14. Graphical representation of the probabilistically informed decision-making process.

Because the linked FT/ET calculates the metrics of interest after a (any) component fails, the number of degraded components or components out of service has no effect on the ability to calculate the metrics of interest. This is because the linked FT/ET avoids the use of a priori decision tables and the limitations associated with them. A component failure (or injected fault) is recognized by the control application program. This failure information is transmitted to the probabilistic model from the supervisory control system. Once the failure occurs, no user interface is required and the results from the probabilistic assessment are transmitted back to the supervisory control system.

Because the results from the FT/ET models are written to a relational database, the supervisory control system reads a self-generated results table that is created in real time. Thus, any change in system configuration or operating status is accurately modeled.

In summary, capabilities of real-time decision-making using linked FT/ET methodology with the results available through a relational database include the ability to

- identify multiple component failures/faults/outages simultaneously,

- identify component failures on a real-time basis,
- identify problems for which *a priori* patterns (or models) have not been constructed, and
- change or modify a decision based on newly evolving conditions.

5.2 DETERMINISTIC PORTION

Plant operating procedures (OPs) are essentially rule-based decision modules executed by human operators. A rule-based model

- identifies the system state,
- associates the state with a task, and
- accesses stored rules to perform the task.

Operational limits and conditions (OLCs) are developed to ensure that the plant is operated in accordance with plant design assumptions and intent. OLCs also include actions to be taken and limitations to be observed by the operating personnel [2-9], or in this case, the supervisory control system.

Operating procedures are developed for normal operation to ensure that the plant is operated within the OLCs and to provide instructions for the safe conduct of all modes of normal operation, such as starting up, power production, shutting down, shutdown, load changes, process monitoring, and fuel handling.

Any action taken by the Supervisory Control System must not diverge from the established OPs and cannot compromise established OLCs.

Other deterministic criteria may be beneficial when coupled to decision-making options identified probabilistically. That is, a probabilistic selection (risk based) may not be the optimal or desired choice (risk informed). For example, a high outlet temperature from the reactor core can be lowered by decreasing power, reducing the coolant inlet temperature, or increasing secondary side flow rate. Each of these can be adjusted using plant controls. Inserting the control rods and increasing coolant flow are a means to reduce core thermal power. Each control option has a different probability of success and can be linked to magnitude, speed, and other metrics of interest. That is, inserting the control rods will have a large, rapid effect on the output temperature while changing pump speed on a feedwater pump will have a small, slow effect.

5.3 REFERENCES

5-1. Nuclear Engineering Department, Rocky Flats Environmental Technology Site, *Nuclear Safety Technical Report, Safety Analysis and Risk Assessment Handbook*, RFP-5098, April 22, 1997.

5-2. American Nuclear Society, "Updating Plant EDGs with Intelligent Digital Control Systems," *Nuclear News*, LaGrange Park, IL, July 2012.

5-3. Isograph, Reliability Workbench, http://www.isograph.com/software/reliability-workbench/

6. DEMONSTRATION OF AUTOMATED DECISION-MAKING

The requirements for the supervisory control system are to use probabilistic information directly during operations on a faster than real-time basis. Translating the requirements into a working model requires a communication pathway between the supervisory control system and the probabilistic models. The sample problem provided in this chapter demonstrates the ability to develop the real-time probabilistic decision-making portion of the supervisory control system.

If more than one option is cited by the probabilistic models, the supervisory control system ranks the operating based on the likelihood of successfully avoiding a trip setpoint. Deterministic criteria applied to the operational alternatives based on the probabilistic results determine the option to be given by, and thus commanded by, the supervisory control system.

The task control and data exchange protocols developed for the supervisory control system allow the probabilistic models to

- reflect the change of state in any component,
- reconfigure the probabilistic models to reflect the change,
- execute the probabilistic tools, and
- transmit the results to the supervisory control system.

After the updated analysis results are transmitted back to the supervisory control system, the supervisory control system will

- automatically and autonomously identify operational alternatives ranked by probability of successfully avoiding the actuation of a safety system setpoint,
- if more than one option is identified, select the preferred option based on deterministic criteria,
- transmit an actuation signal to the component of interest, and
- inform the operator of action taken or request permission to take action.

The program recognizes and implements any change of state and executes the models both automatically and autonomously without any operator input.

This report documents the development of the basic communication capability to exchange data with the probabilistic model using RWB. The difficulty in developing the communication pathways is that the models need to recognize the change in state of a component, transmit that change to the models, automatically adjust and execute the models with the change of state, and transmit the updated results to a user interface. All of these operations must occur without operator interface or direction. That is, the programming has to autonomously recognize and implement any change of state and execute the probabilistic models.

To create a probabilistic tool that would recognize failures and evaluate the consequences of those failures in real time, a simple PT/ET model was created to develop and demonstrate the software code to implement the communication capabilities between the dll and RWB. A program, written in C#, successfully communicates faults to the probabilistic model through the dll. The dll provides access to the internal data, model structure, and the built-in methods and functions that the RWB application uses

internally to perform its model modifications and reliability calculations. The code is able to create new gates in the fault trees, create new failure events, and is able to link the events to the gates.

The communication pathway for injecting a fault (i.e., failing a component) to the probabilistic models was successfully completed; a program, written in C#, successfully communicates faults to the probabilistic model through the dll. Just as important, the communication pathway transmitting the results of the probabilistic models that reflect the failure back to the supervisory control system was also successful.

6.1 PROBABILISTIC PORTION OF DECISION-MAKING

The communication pathways for injecting a fault, instructing RWB to recalculate the metrics of interest, and transmitting the results back to the supervisory control system were successfully completed. This meets the "automatic" requirement for the supervisory control system. To meet the "autonomous" requirement, the supervisory control system must be capable of making a decision based on current plant configuration coupled with a system or component failure. Besides being faster than real time, the corrective action options must be determined automatically and autonomously. That is, once a fault or failure is detected, the supervisory control system must determine what has failed and identify the control options to maintain the plant within the control boundaries. Because the supervisory control system is not based on a priori decisions and it is not executing the reliability software, the supervisory control system must "reconstruct" the event tree, map the failure to the appropriate event tree branch, then "deconstruct" the event tree to identify the control options at the component level (Fig. 15).

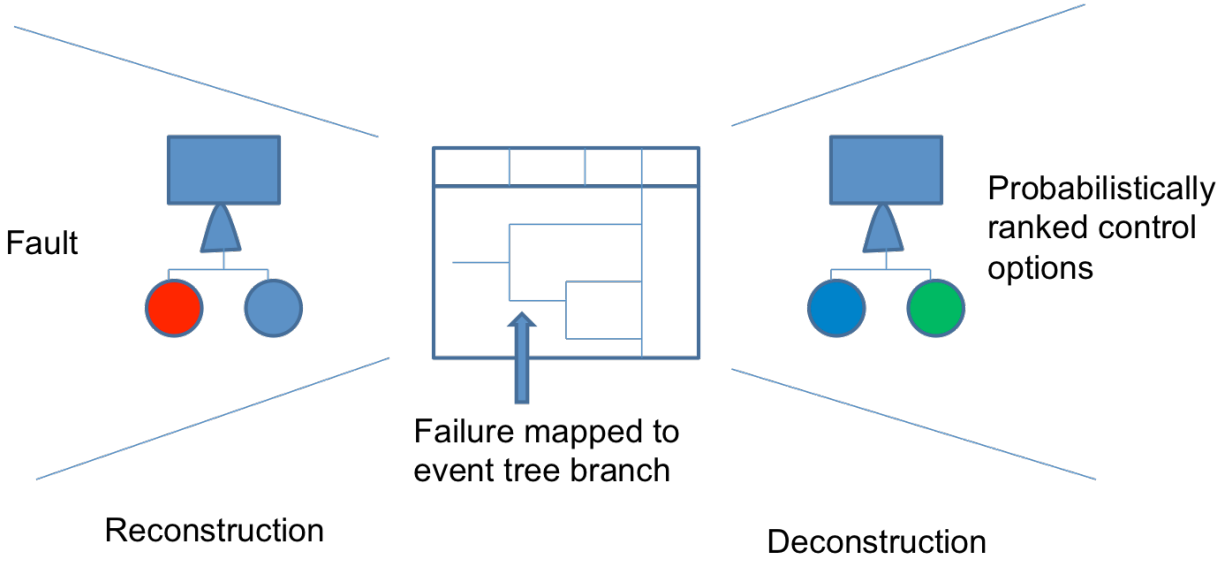

Fig. 15. Sequence to identify probabilistically ranked control options.

The proof-of-concept example models a simple system with a bypass valve arrangement, as shown in Fig. 16. With the system in operation, Valve A is open and bypass Valve B is closed and in the bypass position.[10]

[10] Because actual system conditions can be accounted for, Valve B could be out of service for maintenance. The probabilistic model would recognize this and identify a different control option.

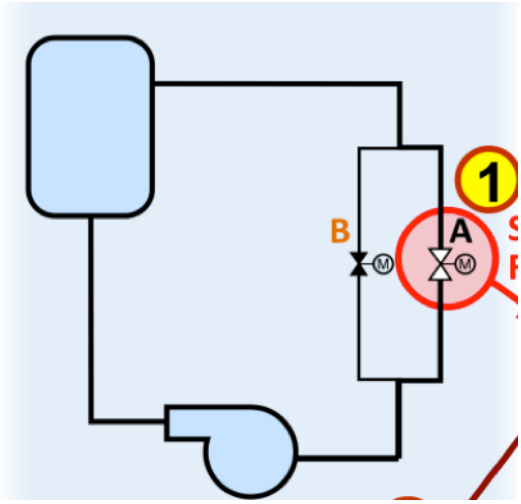

Fig. 16. Proof-of-concept system model.

The flow of information is for the fault to be recognized, rank control options and transmit them to the supervisory control system, which then evaluates the probabilistic options coupled with a set of deterministic criteria and takes action. To select options, the control system must know what failed, where this failure maps to the ET, and then, based on this, identify possible success paths. The supervisory control system must be able to automatically and autonomously identify these success paths for any possible component failure. This process is shown in Fig. 17.

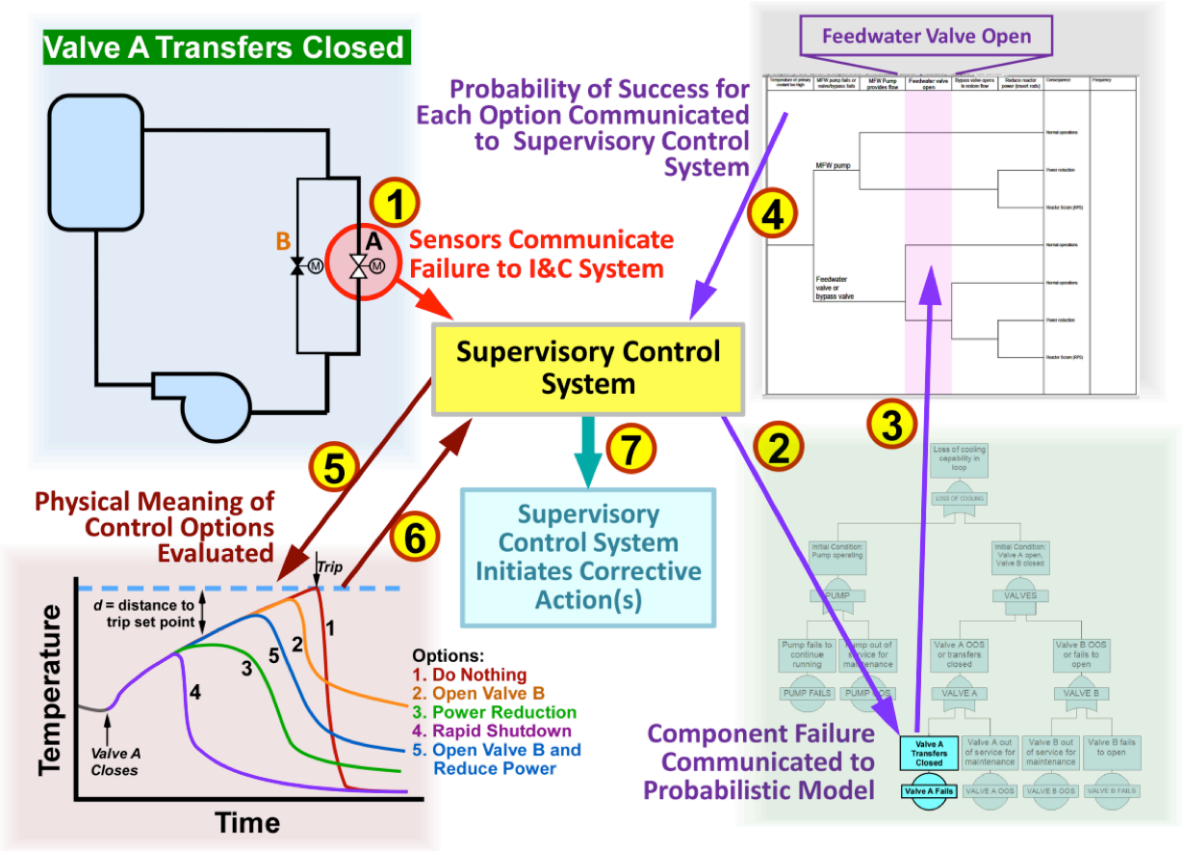

Fig. 17. Communication flow path for supervisory control system.

6.1.1.1 Reconfigure and Execute Probabilistic Models

For the example problem, Valve A fails. The first step in identifying control options is for the supervisory control system to recognize Valve A failed and to modify the probabilistic model to reflect the failure (item 1 in Fig. 17). In this example, a fault is injected to simulate the failure of Valve A. The supervisory control system recognizes that Valve A is no longer operable and is in the failed state. The supervisory control system changes the status of VALVE A in the FT model from operating (VALVE A FAILS, $\lambda = 3.0\times10^{-5}$) to failed (VALVE A FAILED STATE, $\lambda = 1.0$), as illustrated in Fig. 18. The supervisory control system executes the probabilistic analysis with the current plant configuration models and stores the results in a relational database.

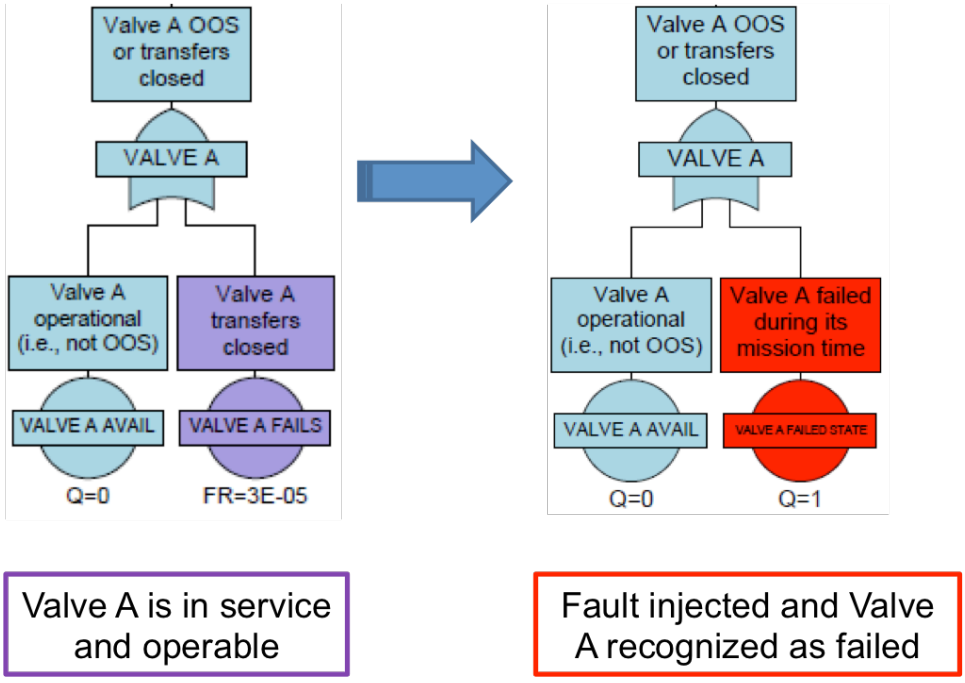

Fig. 18. Component failure is communicated to the probabilistic model.

6.1.2 Reconstruction of ET from Component Failure

In reconstructing the probabilistic model from the data, the supervisory control system must recognize that the fault VALVE A FAILED STATE is input into Gate "VALVE A" in the FT (Fig. 18). That is, the supervisory control system maps the basic event to the gate.[11] The other basic event into Gate VALVE A shows that the valve is not out of service (OOS) for maintenance (i.e., VALVE A AVAIL).

After the fault has been properly mapped to the FT, the FT must be mapped to the ET. In this example, the supervisory control system recognizes that the Gate "VALVE A" is in ET Branch 3 (Fig. 19).[12] Thus the supervisory control system "knows" that the component VALVE A failed and it is in the failure part of ET Branch 3, "Feedwater valve open."

[11] PRACoupling_GateInputs maps the "ObjectType" (i.e., basic event) to "Gate."
[12] PRACoupling_ETColumns maps the "gate" to "SubIndex" or ET Branch. The SubIndex column also tells the supervisory control system that there are five ET branches after the initiating event.

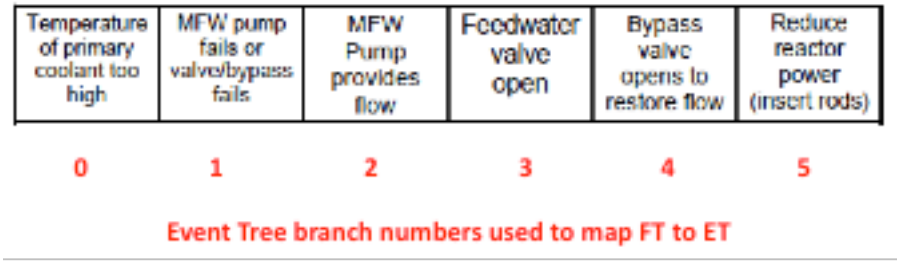

Fig. 19. Mapping of FT to ET.

The next step in the reconstruction process is to actually reconstruct the ET in order to identify and quantify the success paths. Beginning with the IE, the supervisory control system reconstructs the ET.[13] For our example, the branch ID of interest is EB44, which is the failure branch of ET Branch 3 (Fig. 20).

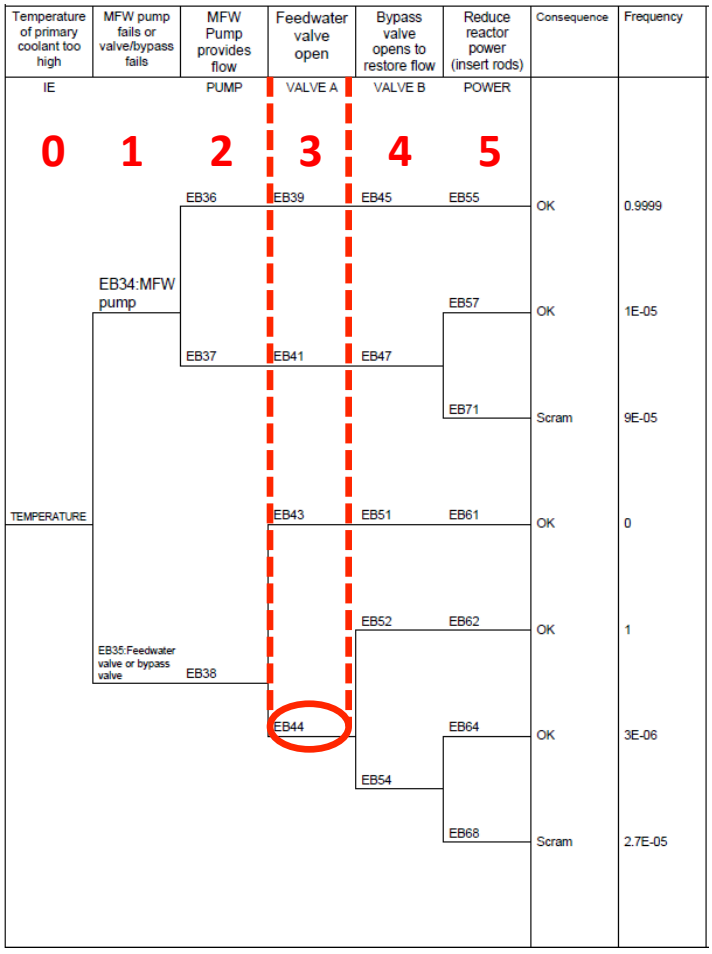

Fig. 20. Reconfigured ET.

[13] The supervisory control system, using InputBranch identifiers, Id, Column, and Type (i.e., success, failure, or null branch), in PRACoupling_ETBranches.

6.1.3 Deconstruction of ET to Corrective Action

Similar to how the supervisory control system reconstructed the ET with the fault properly accounted for in the FT, the supervisory control system must now deconstruct the ET to identify options for system control.

To determine the options associated with the failure of Valve A, the supervisory control system moves its pointer from ET Branch 3 to ET Branch 4 (Fig. 20).[14] ET Branch 4 maps to FT gate "VALVE B" and basic events "VALVE B FAILS" and "VALVE B AVAIL." That is, the supervisory control system maps basic events VALVE B FAILS ($\lambda = 3.0\times10^{-5}$) and VALVE B AVAIL ($\lambda = 0.0$; i.e., not OOS) as inputs to gate VALVE B (Fig. 21).

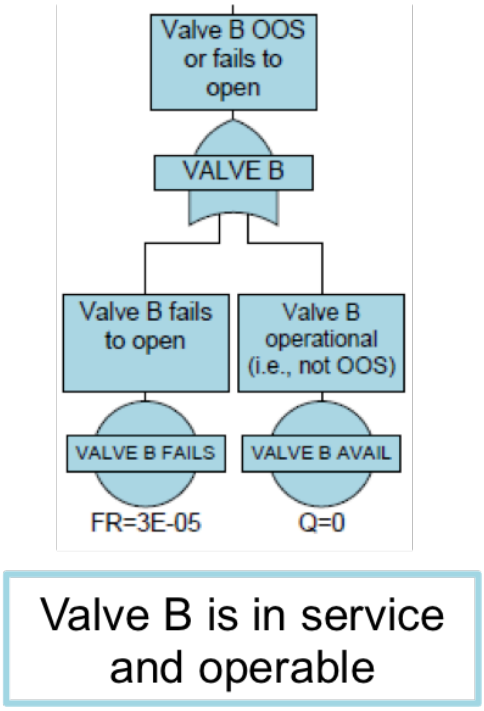

Fig. 21. Deconstruction of ET Branch 4 to Gate VALVE B.

Because the FT models failures and the supervisory control system needs to identify success paths, the supervisory control must convert the failure space into success space and interpret VALVE B FAILS as the option "Open Valve B." This is success path EB52 in ET branch 4 on Fig. 20.

The supervisory control system has automatically and autonomously determined ET branch sequence EB44-EB52-EB62 (Fig. 20) is the chosen control option with essentially a 1.0 probability of success. Thus, the supervisory control system has successfully recognized both automatically and autonomously the existence of a fault, evaluated the operational alternatives available, and probabilistically ranked those alternatives.

[14] Deconstructing the ET using PRACoupling_ETColumns SubIndex, ET Branch 4 is associated with Gate "VALVE B". From PRACoupling_GateInputs, ObjectType maps pseudo events 4 and 5 as inputs to gate VALVE B. PRACoupling_FTETPseudoEvents maps OriginObjectIndex 4 and 5 to VALVE B FAILS and VALVE B AVAIL, respectively. The Frequency column maps the failure rates to the components.

6.2 DETERMINISTIC PORTION OF DECISION-MAKING

Nuclear power plants cannot operate outside known and understood safety limits, which places restrictions on the creation of new (not previously reviewed) action steps, nor can plants be allowed to operate outside certified regulatory limits. Any decision-making process, then, must recognize that limits for specific plant parameters are clearly set and should include operational limits that affect economy and availability.

All data are known beforehand for a deterministic analysis—such an analysis is prefaced by knowing what is going to happen next with little or no uncertainty. However, for real systems, there is always the possibility of not achieving the design objective, that is, to ensure that the system performs satisfactorily within a specified time period. Thus, system and equipment designs rely on safety margins to reduce the risk of adverse performance. The weakness of deterministic decision-making is that it cannot inherently account for the stochastic nature of system behavior, or of component failures.

The decision-module is probabilistically informed rather than probabilistically based decision because the supervisory control system cannot violate the licensing basis or exceed protection system settings. These are rule-based decisions. However, within this rule-based construct, the supervisory control system can use both probabilistic and deterministic decisions. For example, a probabilistic decision can be coupled to the magnitude and speed of actions to be taken.

For an example of probabilistically informed vs. probabilistically based decision, consider an example of lowering the outlet temperature of the reactor core by

- decreasing reactor power,
- reducing the coolant inlet temperature, or
- increasing secondary side flow rate.

Each of these can be adjusted using plant controls. Inserting the control rods, increasing coolant flow, etc., are means to reduce core thermal power. Each control option has a different probability of successfully maintaining the reactor coolant outlet temperature below the trip setpoint. Each control option is based on current plant conditions and/or deterministically identified parameters or criteria that are dependent on reactor power level, distance from the trip setpoint (magnitude), response time necessary to stop the parameter of interest from reaching the trip setpoint (speed), etc. For example, inserting the control rods may have a large, rapid effect on the output temperature while changing pump speed on a feedwater pump may have a small, slow effect. A strictly probabilistically based approach may select “inserting the control rods” as the most likely option for avoiding the trip setpoint limits. A probabilistically informed approach may have “changing pump speed” and “inserting the control rods” as both options for success but select the “changing pump speed” as the best option because it has a deterministic rule of “maintaining MWe to the grid” if possible.

As discussed previously, because the example used to develop the capabilities of the supervisory control system did not have control options, deterministic criteria were not incorporated. At present, utility theory is the method of choice to implement the weighting of the probabilistic options. This option is under development and has not been incorporated into the supervisory control system at present.

7. CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSIONS

The communication pathways to the supervisory control system from the component (i.e., fault) and the probabilistic model were successfully developed and tested. These communication pathways are independent of what component failed and the probabilistic model and are applicable for complicated models as well as the example problem used in the previous section. That is, the communication pathways work for any failure in any model.

The supervisory control system, after successfully accessing the data for the revised and updated probabilistic model, was able to automatically and autonomously reconstruct/deconstruct the FTs and ETs to identify the appropriate action to maintain operations through the use of self-generating lookup tables. This demonstration problem shows that the methodology is viable for recognizing any component failure, even with a changing system configuration. More importantly, the demonstration shows that actual system configurations can be reflected in the probabilistic models because combinations of failures are no longer a limiting factor in being able to solve the problem.

The supervisory control system is a system that combines the computational power of probabilistic analyses and quick-access results tables stored in a relational database. The supervisory control system uses power derived from the probabilistic tools to determine the likelihood of success for various control options for the current plant configuration. By populating a relational database with the results, the supervisory control system uses look-up tables to manipulate the data to identify the control options.

Another feature incorporated into the supervisory control system is the real-time probabilistic generation of control system options. Any change in system configuration or operating status is accurately modeled, with any number of combinations of equipment out of service or in test mode reflected.

The sample problem used in developing the communication links and computational capabilities of the supervisory control system shows the successful merging of the system layout, structure, and capabilities that were specified in previous milestone reports.

7.2 FUTURE EFFORTS

Future work will build-out the capabilities to address more complex problems including components that were out of service, degraded states, prognostics and diagnostics, and of course, multiple reactors in a module.

Future efforts in the development of the supervisory control system will involve programming the supervisory control system to recognize more complex systems with several control options given a component failure. With multiple options, the programming must allow the control system to select an optimal control decision. In addition, the programming must be expanded to allow the system to differentiate between options if a component is out of service in one of the options or placed back into service with no operational change required.

As part of increasing the capabilities of the supervisory control system, the following tasks need to be completed:

- Evaluate the options for the supervisory control system to recognize equipment out of service, especially if this occurs in a success path.

- Develop the ability of the supervisory control system to “learn” when a component is placed back in to service, which does not generate a fault and may or may not require control options to be identified.
- Develop and test a problem with multiple control options identified through the probabilistic models.
- Incorporate the deterministic capabilities into the supervisory control system.
- Incorporate the use of multi-physics models to provide a time-dependent assessment of the approach to trip setpoints that can then be used as input to the deterministic models.
- Develop a problem with multiple ETs and test the supervisory control systems ability to properly reconstruct/deconstruct the FT/ET models.
- With the use of multi-physics models, evaluate the ability to differentiate different options base on power level.
- Assess the computational time for performing the probabilistic analyses and the time associated with data manipulation for the reconstruction/deconstruction process

APPENDIX A. FUNCTIONAL REQUIREMENTS FOR SUPERVISORY CONTROL

APPENDIX A. FUNCTIONAL REQUIREMENTS FOR SUPERVISORY CONTROL

The functional layer structure was based on the IEEE/ANS 830, "Guide for Software Requirements Descriptions," with extensions that ORNL identified based on physical system descriptions. The functional layers are as follows:

1. Objective
2. Functions
3. Variables
4. Interfaces
5. Inputs
6. Outputs
7. Performance Measures
8. Limiting Conditions
9. Trigger Conditions

The proposed supervisory control system concept expands the functional responsibility and level of automation for the control system, and in no way is it intended to support or augment the functions to be performed by the protection system.

Because a conceptual design baseline has not yet been identified, it is not possible to finalize the in-depth requirements for the supervisory control system. Requirements provided in this chapter are generic in nature and are expected remain valid as the requirements matrix evolves.

A.1 FUNCTIONAL REQUIREMENTS FOR MASTER SUPERVISORY CONTROL SYSTEM

The high-level functional requirements of the master supervisory control system (MSCS) are given in Table A.1.

Table A.1. List of high-level functional requirements for master supervisory control system

Objective	Supervise and coordinate operations of individual reactor modules, associated power conversion systems, and allocation of steam (or another fluid medium) for customers.
Functions	1. Supervise and coordinate operation of individual reactor modules, associated power conversion trains, and the system interfaces to the process heat plant. 2. Monitor, process, and analyze the health status of critical SSCs through associated diagnostic and prognostic calculation modules of local control systems. 3. Keep limits and maintain plant stability by mitigating propagation of unexpected transients between reactor modules. 4. Develop operations, maintenance, and refueling strategies, or modify existing strategies, based on information related to the health status of critical SSCs.
Variables	The MSCS shall (if necessary) have access to information (i.e., processed data) regarding the condition and status of all sensory signals.
Interfaces	1. Provide input capabilities for operator directives (operator terminals). 2. Provide bi-directional communications link with local supervisory control systems at each reactor modules. 3. Provide bi-directional communications link with balance-of-plant systems. 4. Provide bi-directional communications link with the interface control system for the process heat plant. 5. Provide direct communications link with diagnostics and prognostics modules for random inquiry of the health status of critical SSCs. 6. Provide capability to bypass the LSCS for direct access to the local control system under such circumstances that any of the SSCs are not responsive.
Inputs	1. Accept plant control directives from reactor operators. 2. Accept load demand input from the grid central dispatch. 3. Accept load demand input from the process heat plant.
Outputs	1. Generate high-level instructions for LSCSs. 2. Generate permission requests for actions that require operator concurrence. 3. Generate a detailed report for supervisory actions.
Performance Measures	These metrics will be determined later.
Limiting Conditions	These metrics will be determined later.
Trigger Conditions	These metrics will be determined later.

A.2 FUNCTIONAL REQUIREMENTS FOR LOCAL SUPERVISORY CONTROL SYSTEM

This section provides the high-level functional requirements of the local supervisory control system (LSCS) and the definition and the requirements for the functional modules.

A.2.1 Top-Level Requirements

The high-level functional requirements of the LSCS are given in Table A.2.

Table A.2. List of high-level functional requirements for master supervisory control system

Objective	Monitor and analyze processes in the system, transmit module-level status information to the MSCS, and direct the RTEL that performs the real-time control functions. The LSCS uses various signal processing and control theory approaches to process input measurements and information from the MSCS and to determine viable options for the next decision-making step in the control of the system. To achieve this objective, clear requirements for the LSCS must be articulated.
Functions	1. Process the input data to enable decision-making: • State estimation • Diagnostics • Prognostics 2. Utilize methods to generate sets of decision options for consideration: • State analysis and determine options 3. Select a prime candidate solution from the generated sets of solutions: • Decision analysis 4. Validate and verify that prime candidate solution with analytical tools: • Validation and verification 5. Generate control actions from the prime candidate solution for the RTEL: • Control actions
Variables	To be determined
Interfaces	• MSCS coordination, direction, and guidance to the LSCS for reactor modules, power conversion systems, process heat, and other systems • Subordinate RTEL modules
Inputs	• Local system measurement inputs and control outputs • Information from the MSCS for desired future state based on the MSCS coordination functions • Information from the human operator for the specific systems
Outputs	The LSCS performs decision-making tasks to direct the system real-time controllers in the execution layer. The outcome of the decision-making tasks is updated guidance for control action to the RTEL modules.
Performance Measures	• Information that is provided to the MSCS will facilitate plant-level evaluation of meeting the desired performance and outcome. • The LSCS will examine the current state and the desired state to determine steady state and transient metrics such as an error in the system performance versus the desired values.
Limiting Conditions	The decision-making process includes a physics-based assessment component that provides insight into the limit margin for the current state and future decision state options. This limit margin includes operating limits, engineering limits, and component limits.
Trigger Conditions	Component failures or degradation will trigger various responses from the RTEL and the LSCS layer. The LSCS will examine the failure or degradation event and determine the proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

A.2.2 Plant State Estimation Module

Table A.3. List of functional requirements of the plant state estimation module

Objective	Process input information from the LSCS and the MSCS to generate estimates of the system state including non-measured values. This includes declaration of subsystem failures by basic failure detection information. Failed sensors may be substituted by synthesized values using state estimation, models, and correlation techniques.
Functions	1. Generate a state estimate of the system based input information, local input measurements, and local output commands. 2. Examine measurements (inputs, outputs) and estimated states to determine if a component, subsystem, or system has failed (failure declaration). 3. Examine measurements (inputs, outputs) and estimated states to determine if a sensor or measurement device is experiencing drift (shifting error behavior) or a change in its noise characteristic. 4. Examine input and output measurements and provide an estimate of a measured state if the measurement experiences measurement failure (synthesis).
Variables	To be determined
Interfaces	• The plant state estimator interfaces with the LSCS input data and the LSCS status analysis and determines options functions.
Inputs	1. Local system measurement inputs and control outputs 2. Information from the MSCS for desired future state based on the MSCS coordination functions
Outputs	1. Status of subsystems and components based on current state and conditions 2. Failure declaration of a subsystem or component 3. Additional algorithm to determine sensor drift and noise (process and sensor) 4. Diagnostic information for diagnostics and prognostics functions
Performance Measures	The effectiveness of applying Kalman filtering state estimation is based on how well the process noise characteristic is understood. The filter output residuals can provide an indication of the effectives of the filtering (filter gain, noise description, etc.) and the system model that is the basis for the filter design.
Limiting Conditions	Some operating modes and conditions may not be modeled at a sufficient level of detail to facilitate state estimation functions. These operating modes and conditions will rely on traditional monitoring and control approaches.
Trigger Conditions	Component failures or degradation will trigger various responses from the RTEL and the LSCS layer. The LSCS will examine the failure or degradation event and determine proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

A.2.3 Diagnostics and Prognostics Module

The *Diagnostics and Prognostics Module* (DPM) includes two functional blocks: diagnostics block and prognostics block.

This diagnostic block consists of on-line monitoring (OLM) of measurements, states, and synthesized parameters to determine if the system behavior does not comply with the expected behavior. If the noncompliance is identified, then algorithms and models will provide estimates about the potential root causes of noncompliance (Table A.4).

Table A.4. List of functional requirements for the diagnostics module

Objective	The diagnostics block processes input information from the MSCS and LSCS state estimation information to determine the nature and cause of a current event or condition, to detect current faults, and to determine the current state of health. This diagnostic information can provide guidance for short-term and long-term decision-making.
Functions	1. Use dynamic models of components and subsystems with input data to identify unexpected behavior and to identify possible root cases. 2. Use look-up tables to examine combinations of events, and to determine the likely root cause and associated diagnostic information. 3. Use methods to properly characterize and summarize the current continuous and discrete states versus the desired state combinations in a manner that indicates potential degradation modes.
Variables	To be determined
Interfaces	The diagnostic block interfaces with the Data Acquisition Module, Plant State Estimation Module, and the Decision-Making Module.
Inputs	The diagnostic block receives input measurements, feedback measurements, control output data, etc., to estimate the current system state of health.
Outputs	The diagnostic block provides information regarding the indication or failure declaration of a subsystem or component with some degree of confidence.
Performance Measures	The effectiveness of the diagnostic block is the degree to which the generated information is accurate, timely, and useful for the other systems. Non-detected failures and improperly diagnosed events are to be minimized.
Limiting Conditions	Some operating modes and conditions may not be modeled at a sufficient level of detail to facilitate diagnostic functions. These operating modes and conditions will rely on traditional monitoring and control approaches.
Trigger Conditions	Component failures or degradation will trigger various responses from the real-time executive layer and the LSCS layer. The LSCS will examine the failure or degradation event and determine proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

The prognostics block consists of usage and duty cycle tracking functions, life prediction algorithms, and long-term monitoring of key parameters that are indicative of component life. This information provides an estimate of the future state of health of the system over various time horizons.

The functional requirements of the prognostics block is given in Table A.5.

Table A.5. List of functional requirements for the prognostics module

Objective	The prognostics block processes input information from the MSCS, LSCS state estimation, and LSCS diagnostic information to generate a prediction of the time at which a system or a component will no longer perform its intended function with certainty, which is often described as Remaining Useful Life (RUL). A description of the RUL and associated probabilities of failure over various time horizons provides component life and performance time horizon information for the LSCS decision-making processes.
Functions	1. Usage and cycle tracking process various power plant events to determine the proper usage and cycle tracking values for the system and subsystems 2. Life prediction algorithms include various approaches to estimate component life for their intended operating specifications and conditions 3. Determination of RUL consists of a method to combine the usage and cycle tracking information and the life prediction results to generate component, system, and subsystem RUL and probability of failure over various time horizons
Variables	To be determined
Interfaces	The prognostics block interfaces with the LSCS input data, state estimation, diagnostic functions, and the LSCS *Decision-Making Module*.
Inputs	The prognostics block receives input data, state estimation information, and diagnostic information.
Outputs	The prognostics block generates output data describing the system and subsystem RUL and probability of failure for various time horizons, such as $P_{immediate}(t)$, $P_{100\,h}(t)$, $P_{1000\,h}(t)$.
Performance Measures	The effectiveness of the prognostics block is the degree to which the generated information is accurate, timely, and useful for the other systems. Improper RUL and probability of failure occurrences are to be minimized.
Limiting Conditions	Some limited operating modes and conditions may not fall within valid regions or ranges for the life prediction algorithms and the manufacturer data. These limited modes and conditions must occupy a small fraction of the operating duty cycle to avoid polluting the RUL and probability of failure results.
Trigger Conditions	Component failures or degradation will trigger various responses from the real-time executive layer and the LSCS layer. The LSCS will examine the failure or degradation event and determine the proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

A.2.4 Decision-Making Module

The decision-making module includes two functional blocks: (1) decision-options block, which implements the probabilistic portion of decision-making using real-time PRA (Table A.6), and (2) decision-analysis block, which implements the deterministic portion of decision-making (Table A.7).

Table A.6. List of functional requirements for the decision-options block

Objective	The decision options block processes input information from the MSCS, LSCS state estimation, diagnostic, and prognostic information to generate a set of decision options to transition the system from the current state to the desired future state. The different features used to generate the set of decision options provide a risk-informed, physics-constrained, and regulatory-compliant outcome. This is accomplished with four main functions used to generate decision options: dynamic probabilistic risk assessment (RT-PRA), physics-based assessment, procedure-based assessment, and a navigation function.
Functions	1. Input data processing is necessary to map the input data from the other functions into the proper format for the key decision generation functions. 2. The RT-PRA function will process an input set that describes the current failure status of systems and subsystems, health status, RUL estimations, and probabilities of failures. Then the RT-PRA function will update the appropriate fault-tree probabilities and generate an updated event tree. The updated event tree is used to identify trajectories with acceptable likelihood of success and low probability of success. 3. A navigation function is used to map the desired future state onto the current state and to identify decision options to reach the future state based on the plant design.
Variables	To be determined
Interfaces	The decision options block interfaces with the LSCS Data Acquisition, Plant State Estimation, and Diagnostic and Prognostic modules, and the LSCS decision analysis functions.
Inputs	The decision options block receives MSCS input data (desired future state), state estimation information, and diagnostic and prognostics information.
Outputs	The decision options block generates an output data set of possible decision options for the LSCS decision analysis function to process.
Performance Measures	The effectiveness of the decision options block is the identification of several decision options under all operating conditions.
Limiting Conditions	Some limited operating modes and conditions may not fall within valid regions or ranges for generating automated decision options. These limited modes and conditions must occupy a small fraction of the operating duty cycle and will rely on human operator or legacy control approaches. The effectiveness of the RT-PRA function is related to the degree to which the systems and subsystems are accurately described with event probabilities, event combinations, and the event tree.
Trigger Conditions	Component failures or degradation will trigger various responses from the real-time executive layer and the LSCS layer. The LSCS will examine the failure or degradation event and determine proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

Table A.7. List of functional requirements for the decision-analysis block

Objective	The decision analysis block examines the set of decision options from the decision options block and determines a prime candidate option that provides the best balance of a desired likelihood of success, limit margin, and procedure compliance. This is accomplished using the utility functions.
Functions	1. The physics-based assessment function processes an input set that describes the current state of the system. This input set is compared to various engineering, component, and stability limits in a manner that results in a data set that describes the distance or margin from the current state to these limits. 2. The procedure-based assessment function processes an input set that describes the current state of the system and compares this input set to plant procedures. This results in a data set that describes the compliance of the current state to the intended procedural operation of the plan.
Variables	To be determined
Interfaces	The decision analysis block interfaces with the decision options block and the verification module.
Inputs	The decision analysis block receives input data from the decision options block.
Outputs	The decision analysis block generates an output dataset prime candidate solution for the LSCS Verification Module to process.
Performance Measures	The effectiveness of the decision analysis block is the selection of a prime candidate option for all conditions.
Limiting Conditions	Some limited operating modes and conditions may not fall within valid regions or ranges for generating automated decision options. These limited modes and conditions must occupy a small fraction of the operating duty cycle and will rely on human operator or legacy control approaches.
Trigger Conditions	Component failures or degradation will trigger various responses from the real-time executive layer and the LSCS layer. The LSCS will examine the failure or degradation event and determine proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

A.2.5 Verification Module

Table A.8. List of functional requirements for the Verification Module

Objective	The *Verification Module* examines the prime candidate solution from the Decision Making Module with dynamic plant model simulation and analysis tools that examine the simulation results to verify the desired outcome. Feedback from the results is used to improve the DMM function.
Functions	The verification dynamic simulation includes continuous state dynamics and discrete state dynamics. The simulation is intended to estimate stability, limit margin, and the effect of varying conditions. The time duration of the dynamic model simulation and results analysis is based on a time horizon that is appropriate for the desired future state time horizon.
Variables	To be determined
Interfaces	The Verification Module interfaces with the LSCS Decision Making Module and the Actuation Module.
Inputs	The VM receives input data from the Decision Making Module.
Outputs	The VM generates evaluation results of the prime candidate solution for the control action function to process. The evaluation results are also used to provide feedback guidance to the DMM function.
Performance Measures	The effectiveness of the VM is the ability to evaluate the prime candidate solution with the near real-time update timing requirements.
Limiting Conditions	Some limited operating modes and conditions may not fall within valid regions or ranges for generating automated decision options. These limited modes and conditions must occupy a small fraction of the operating duty cycle and will rely on human operator or legacy control approaches.
Trigger Conditions	Component failures or degradation will trigger various responses from the real-time executive layer and the LSCS layer. The LSCS will examine the failure or degradation event and determine proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

A.2.6 Actuation Module

Table A.9. List of functional requirements for the Actuation Module

Objective	The *Actuation Module* performs a transformation of the prime candidate solution into control action commands and guidance for the real-time executive layer.
Functions	The Actuation Module transformation includes continuous state dynamics and discrete states. The AM transformation algorithm must be sufficient to update the real-time executive layer in a near real-time manner.
Variables	To be determined
Interfaces	The AM interfaces with the LSCS VM and the real-time executive layer.
Inputs	The AM receives input data from the VM function.
Outputs	The AM provides continuous and discrete state commands and guidance to the real-time executive layer.
Performance Measures	The effectiveness of a control action function is the ability to update the real-time executive layer with the near real-time update timing requirements.
Limiting Conditions	Some limited operating modes and conditions may not fall within valid regions or ranges for generating automated decision options. These limited modes and conditions must occupy a small fraction of the operating duty cycle and will rely on human operator or legacy control approaches.
Trigger Conditions	Component failures or degradation will trigger various responses from the RTEL and the LSCS layer. The LSCS will examine the failure or degradation event and determine proper next steps by the robust decision-making process. These steps are considered near real-time to complement and guide the RTEL.

A.3 EXAMPLES OF DECISION-MAKING

The following applications of automated decision-making processes in industry were reviewed:

- Railway,
- Engineering, business, and finance,
- Aerospace industry,
- Unmanned aerial vehicles,
- Nuclear power, and
- Highly autonomous driving.

A.3.1 Railway

The control of railways is in many ways analogous to the negotiating a pathway through a state-space region during system transition, as illustrated in Fig. 38 in Ref A-1. Hence, there is interest in examining railroad supervisory decision-making. The degrees of freedom are more limited for the case of railroads (i.e., the tracks and stations are fixed) as contrasted with the parameter space of a nuclear power plant. Nevertheless, developments in supervisory decision-making emerging from the rail industry have applicability.

Dispatching large areas in a railway network is a challenging task because of the numerous constraints that must be accounted for during the decision-making process. Rail transport differs from road transport in that vehicles move over a very restricted topology, which results in strong interaction between vehicles. In some ways, railroad operation is similar to nuclear power plant control because there are specific

predetermined pathways that must be followed. Two examples of restricted pathways are (1) the adherence to detailed and approved procedures, which are administrative pathways, and (2) the fixed piping and electrical pathways with their attendant pumps, valves, and circuit breakers.

Railway dispatching has been historically accomplished by human operators. Algorithmic approaches attempted over the last three decades have not completely solved this task—computational complexity and simulation accuracy has been inadequate for practical application. Over the last few years, however, new dispatching methods including predictive control have emerged that brings a new algorithmic approach with practical application.

Other methods reviewed include Petri nets [A-2], train-timetabling problem (TTP) [A-3], and the use of fuzzy logic [A-4], [A-5], [A-6].

A.3.2 Engineering, Business and Finance

Systems engineering is a methodological approach to developing and realizing products, processes, and services such that critical considerations in the corresponding project and system lifecycles are optimized. Systems engineering uses tools to organize project, product, and service development with a goal to maximize workflow efficiencies and satisfaction of project stakeholders and end users. One family of tools centers on the Kepner-Tregoe decision analysis method [A-7].

The Kepner-Tregoe decision analysis method is typically used by a team of experts to score alternatives numerically based on individual judgments. The method as originally conceived imposes a linear weighting of objective criteria against which the alternatives are assessed. A total score is determined for each alternative by multiplying its score for each criterion by the criterion weight and then summing across all criteria. The method generates a quantitative comparison of alternatives. The preferred alternative will have the highest total score. Several modifications have been made to the original decision analysis method over the years [A-8, A-9].

The Kepner-Tregoe decision analysis process is amenable to automation provided that (1) good objective criteria can be generated for each decision session, (2) a consistent method of scoring alternatives against each objective criterion can be applied, and (3) weightings can be developed that reflect the nature and importance of the decision to be made. These three requirements, although achievable by a software system, need development to implement effectively. Some research is needed to construct a means to make such scoring methods. For example, one significant modification to the Kepner-Tregoe is to change the linear relationship between the objective criteria.

In the area of finance, the need is increasing to automate real-time monitoring to take advantage of time-sensitive business opportunities and detect fraud in near real-time. Nguyen et al. introduced an enhanced business intelligence architecture that covers the complete process to sense, interpret, predict, automate and respond to business environments and thereby aims to decrease the reaction time needed for business decisions [A-10].

A.3.3 Aerospace Industry

Aerospace control systems perform various decision-making tasks related to operating the vehicle and propulsion systems in a desired and safe manner. These tasks must incorporate pilot commands, current conditions, the state of the various systems, and other information to determine the appropriate actions for the vehicle and propulsion systems. Several examples of legacy and new approaches are introduced in the following text.

A.3.3.1 Current Commercial Jet Engine Control

Commercial jet engine control systems maintain fan speed, engine pressure ratio, and control fuel flow to regulate aerodynamic thrust. Aerodynamic thrust is not directly measured but can be estimated by the measured system states. The control is based on a series of selection logic functions (minimum and maximum) that are used to select the appropriate fuel flow and actuation [A-11]. The arrangement of the selection logic is chosen for the desired performance, reliability, and safety. The inputs to the selection logic are the current state, the desired future state, and the appropriate limits of the system based on current conditions and state. The fuel flow selection logic outputs a change in fuel flow, which is integrated to produce the actual fuel flow value. This is often described as the "divide and conquer" approach which partitions various control functions and operating regimes and addresses each one with limited consideration for the complete control system [A-12]. Each of these regimes may be approximated with a linearized dynamic model, which enables linear control theory applications. These various controllers are "stitched together" with gain schedules and selection logic [A-12]. This is quite successful with lower performance aircraft but has serious limitations for higher performance military aircraft applications.

A.3.3.2 Current Commercial Jet Engine Health Monitoring

Commercial jet engine control systems perform various monitoring functions for detection and diagnosis of faults, prediction and prognostics for future component faults and determination of engine health. These functions and methods are determined by the results of a failure modes effects and criticality analysis (FMECA) for the various systems. These results determine what failure modes must be detected and diagnosed.

Sensor and actuator validation is performed using two approaches. The first approach is monitoring the electrical integrity of the various subsystems. This monitoring is performed continuously by the electronic controls and is often described as continuous built-in-test (CBIT). This includes detection of open circuits, short circuits, and improper electrical current. If improper operation has been discovered the control system will perform appropriate actions such as selecting redundant systems or altering the control laws in consider the failure. This is described as fault detection and isolation/accommodation (FDI or FDA).

A second approach to sensor and actuator validation is performed by using measurements, algorithms, and logic to determine if the subsystem is operating nominally. This validation is performed using simple logic, real-time models, and first-order approximations for responses to perform limit checks, rate of change checks, and response checks for sensors and actuators. These can direction immediate control actions and maintenance actions. In some cases on-board models are used to aid in sensor validation and in redundant system voting.

In addition to subsystem validation, other health related parameters such as engine vibration and the lubrication system are monitored to drive maintenance actions. Life cycle counting is also performed to direct maintenance inspections and actions for various components.

A.3.3.3 Jet Engine Control Allocation

In the field of aerospace vehicle and propulsion control, the concept of control allocation is used to perform real-time decision-making and determination of proper solutions based on the state of the system, the health of the components, the current conditions, and the desired future states [A-13]. The control allocation includes consideration of system constraints, limits, and component failures. This control allocation is achieved with optimization techniques (constrained optimization, matrix inversion) and

decision-making logic (logic based on system requirements and FMECA studies). This approach requires an over-actuated system with redundant effectors and actuators that enables multiple solutions for a given desired state [A-14].

The control allocator [A-13] computes control actuation to produce the desired aerodynamic moments in roll, pitch, and yaw from the flight controller (Eq. 1). This allows the separation of the design of the flight control laws and the design of the control allocator. Selects desired optimized solution such as performance, efficiency, stability, etc. In the case of an identified failure, a different solution can be chosen to provide the desired response to the flight control laws.

The control allocation on-line decision-making is determined by techniques such as constrained optimization, matrix inversion, or explicit control laws, decision tree logic. The logic is based on system requirements and FMECA considerations. The optimization techniques may include direct control allocation, daisy chaining, linear programming, etc. The matrix inversion technique does not always produce an optimal solution but it usually computationally faster.

The control allocator solves a system of constrained equations, which is considered a mapping in the controlled system. After linearization, the mapping can be rewritten in the standard formulation for a constrained linear control allocation problem.

The control algorithm hierarchy of motion control for over-actuated mechanical systems with a redundant set of effectors and actuators commonly includes three levels [A-14]. First, a high-level motion control algorithm (linear or nonlinear matrix-based) commands a vector of virtual control efforts (i.e. forces and moments) in order to meet the overall motion control objectives. Second, a control allocation algorithm (linear or nonlinear matrix-based) coordinates the different effectors such that they together produce the desired virtual control efforts, if possible. Third, low-level control algorithms (minor loop closed-loop control) may be used to control each individual effector via its actuators. Control allocation offers the advantage of a modular design where the high-level motion control algorithm can be designed without detailed knowledge about the effectors and actuators.

Important issues such as input saturation and rate constraints, actuator and effector fault tolerance, and meeting secondary objectives such as power efficiency and tear-and-wear minimization are handled within the control allocation algorithm. Control allocation is demonstrated in a rapidly growing range of applications that have expanded from the aerospace and maritime industries, where control allocation has its roots, to automotive, mechatronics, and other industries. Applications consist of two main classes based on the use of linear or nonlinear models, respectively. The presence of physical constraints (e.g. input saturation and rate constraints), operational constraints and secondary objectives makes optimization-based design a powerful approach. The simplest formulations allow explicit solutions to be computed using numerical linear algebra in combination with some logic and engineering solutions, while the more challenging formulations with nonlinear models or complex constraints and objectives call for iterative numerical optimization procedures.

There is interaction and division of responsibility for the flight controller, control allocator, fault detection & isolation and model-based state estimation, supervisor, and the actuation and aircraft.

The supervisor function provides the actuator constraint information and the appropriate actuator behavior mode to the control allocator. The supervisor also provides actuator excitation signals that are used for fault detection. The supervisor receives fault detection information and state estimation from the FDI/MBSE function and determines the health status of the aircraft, what fault conditions are present, and the appropriate actuation behavior mode (control mode) to pursue. Combinational logic, state machines, and numerical maps provide the desired decision-making.

The fault detection & isolation and model-based state estimation function examines measurements from the aircraft, compares the measurements to actuation commands, and then performs fault detection and isolation and model-based state estimation. The estimated state is provided to the flight controller. The fault detection and isolation information is provided to the supervisor function. Kalman filtering is performed with linear or non-linear approaches to estimate the actuation fault detection state. Real-time models provide state estimation of the aircraft based on measurements and control commands.

The control allocation approach does not rely heavily on the gain scheduling and selection logic used in legacy approaches, which enables performance and stability that is appropriate for high performance military aircraft [A-12]. This approach has been demonstrated to provide disturbance rejection, compliance with system constraints, limits, and component failures. Control allocation has been applied to the F-35 Joint Strike Fighter program to achieve proper flight handling and stability for three aircraft and propulsion system variants: conventional takeoff and landing (CTOL), short takeoff and vertical landing (STOVL), and carrier variant (CV) [A-15].

A.3.3.4 Intelligent Control of Space Shuttle Main Engine (SSME)

The SSME control system was updated in the 1990s to reflect advanced performance and reliability in military jet engine controls. This intelligent control of the SSME is a hierarchy of various control and diagnostic functions including life-extending control, real-time identification, and sensor/actuator fault tolerance [A-11]. Artificial intelligence, If-Then logic and rule functions based on requirements, and onboard real-time models are used for the Engine Level Coordinator function. The intelligent control system increased the autonomy of the engine controls by becoming self-diagnostics, self-prognostics, self-optimizing, and mission adaptable. The intelligent control hierarchy structure consist of lower levels operating in a fast real-time manner with the subsystems which consists of algorithmic tasks with less intelligence and upper levels operating on a longer real-time scale with more intelligence. In other words, the lower level provides the closed-loop control and basic diagnostics. The upper level evaluates the ability to carry out the mission. The upper level communicates status and health to the Propulsion Level Control.

The life-extending control function provides the desired steady state and transient performance with reductions in component fatigue due to mechanical, thermal, and other effects [A-11]. This is primarily done by adjusting the engine acceleration schedule and control, which accelerates the fan and core to provide the desired thrust within the required time. The adjustment of these schedules balances transient performance with minimizing component damage. These results in a reduction in the accelerations and velocities of actuators, reductions in peak overshoot temperatures in the core. Another aspect of life-extending control is active clearance control. As the engine components degrade, the control can track the degradation and adjust the control action to provide proper engine performance and reliability.

A.3.4 Unmanned Aerial Vehicles

As the tasks and roles of unmanned aerial vehicles (UAVs) increases so does the requirements to increase their level of autonomy and intelligence. Today's UAVs are employed for intelligence gathering, surveillance, reconnaissance missions, fighting wildfires, traffic reports, and border security. UAVs are often asked to perform a task such as detecting and tracking a target of interest in a dynamic and uncertain environment. This often includes the processing large quantities of sensory and communicated information. In some circumstances, coordination of multiple vehicles requires task management to avoid conflict or collisions. Autonomous functions and capabilities for UAVs are typically categorized as sensor fusion, communications, path planning, task allocation and scheduling, and cooperation with other resources.

Decision-making functions for UAVs consists of centralized and decentralized approaches. Centralized decision-making consists of a central mission control approach that manages large-scale activities in a locale including specific activities for each UAV. This is a traditional remotely directed approach that may require a human operator directing the UAV remotely.

Decentralized decision-making can reduce data communication between a central control and the UAV, which can reduce the system communication bandwidth requirements [A-16]. Decentralized approaches must provide local UAV situational awareness and task awareness to support the UAV mission. The UAV onboard decision-making subsystems consist of sensory processing, path planning, and the autopilot. The sensory processing utilizes various sensory inputs to determine state estimates of desired targets or parameters of interest. The path planning utilizes information such as GPS locations, communication data about other UAVs, and the state estimate of the desired target to determine a desired path [A-16, A-17]. The desired path is then executed but the auto pilot control and the low level vehicle control system. In [A-18] various classifications of autonomous control levels are presented. Architecture was chosen with a task-level model and a decision-process model. The task-level model provides autonomy with respect to external commands and sensory data and the decision-process model provides lower level autonomy and decision-making.

Target estimation and tracking is performed using probability maps, Kalman filtering, and rule-based intelligence [A-16]. Sensor processing is performed using Kalman filtering and particle filtering techniques [A-16, A-17]. The autopilot is the interface between the higher-level decision-making capabilities and the air vehicle. The autopilot utilizes models of the vehicle dynamics, state estimates, and measurements to properly follow the desired flight path.

Current research is investigating the potential for UAVs to operate autonomously as independent entities or collaboratively with other UAVs. In some military applications it is desirable for a UAV to detect a target, determine the value of the target, and decide if a strike on the target is appropriate. In [A-17] a rule-based fuzzy reasoning approach is used to process data from various information sources to produce appropriate decisions in the presence of uncertainty and measurement noise for target tracking decision-making. Fuzzy petri nets can be used to construct or design a rule-base fuzzy reasoning approach. The use of petri nets provides the ability to visualize the structure of the rule-base and provide a mathematical form to express the behavior of the rule-base. These features enable validation of a rule-based system.

In Ref. A-19, an architecture for decision-making for cooperation of multiple UAVs is presented. In this example, some UAVs are directly controlled by a remote human operator, some have operational autonomy, and some have decisional autonomy. This architecture includes algorithms for decision-making. A contract-net protocol handles task allocation for multiple UAVs. A planning scheme is based on Hierarchical Task Networks planning. The contract-net uses a market-based approach where a UAV member will auction a desired task for all UAVs to bid on. The highest bid UAV receives the tasks and must integrate this task into its task planning. Optimization constraints are used to limit bids based on minimizing the actual distance of travel and other cost factors. This results in reducing the total travel distance resultant for a set of UAVs for a set of tasks.

The WITAS project investigated approaches for fully autonomous helicopter vehicles including basic research of artificial intelligence, data and knowledge representation, sensor fusion, interaction with ground control and operations, and validation strategies [A-20]. In Ref. A-21, a genetic algorithm approach is used to process information for target selection in a multiple UAV environment. This approach allocates various targets to various UAVs using the target value, target distance, UAV fuel consumption, weapons payload, and other information.

A.3.5 Nuclear Power

Since the mid-1980s, risk-informed decision-making has formed the basis for licensee requests to change a plant's licensing basis.[15] In implementing risk-informed decision-making, changes to the licensing basis (including changes to technical specifications and maintenance activities) are expected to meet the following set of key principles:

1. The proposed change *meets the current regulations* unless it is explicitly related to a requested exemption.

2. The proposed change is consistent with the defense-in-depth philosophy (i.e., adequate *defense-in-depth is maintained*).

3. The proposed change *maintains sufficient safety margins*.

4. When proposed changes result in an *increase in core damage frequency (CDF) or risk, the increases should be small* and consistent with the intent of the NRC Commission's Safety Goal Policy Statement.

5. The impact of the proposed change *should be monitored* using performance measurement strategies.

The acceptability of proposed changes is based on the results of traditional engineering evaluations, supported by insights (derived from the use of PRA methods) about the risk significance of the proposed changes. Any risk-informed decisions and proposed changes to the licensing basis or technical specifications must meet current regulations, orders, and license conditions.

The risk-informed decision-making process in nuclear power plants is static in that all decisions are made prior to any changes to the plant status or licensing basis. It is also important to note that these evaluations using PRA methods pertain to safety and safety-related structures, systems, and components and measure the increase in CDF. However, several key insights—meet current regulations, maintain safety margins, use PRA methods to support traditional engineering evaluations—are insights that should be carried forward in the development of a supervisory control system.

With respect to automated decision-making at nuclear power plants, two examples are provided below.

Automated decision-making is already in use at nuclear power plants. For example, the integrated control system (ICS) at a B&W-designed plant has as its basic requirement the matching of generated electrical megawatts with demanded electrical megawatts. The ICS accomplishes this requirement through four subassemblies [A-22]:

- the unit load demand functions as a megawatt electric setpoint generator for the ICS, and can be used to adjust reactor power between 15-100%.

[15] Risk-informed guidance to support changes to a licensee's approved licensing basis, including operational programs, includes RG 1.174, "An Approach for Using Probabilistic Risk Assessment in Risk-Informed Decisions on Plant-Specific Changes to the Licensing Basis," [A-23] and RG 1.177, "An Approach for Plant-Specific, Risk-Informed Decision-making: Technical Specifications") [A-24]. In addition, before performing maintenance activities, 10 CFR 50.65(a)(4) requires licensee to assess and manage the increase in risk that may result from the proposed maintenance activities [A-25].

- the integrated master receives the megawatt setpoint from the unit load demand to control the electrical output of the turbine generator. In addition, the integrated master translates the megawatt demand into signals for feedwater and reactor control.

- the feedwater demand converts the megawatt demand signal to a feedwater demand in the integrated master, controls the amount of feedwater supplied to the once-through steam generators, and

- the reactor demand moves the reactor's control rods in or out in response to the megawatt demand signal, and also controls the average reactor coolant system temperature.

The subsystems operate based on differences between actual and demanded parameters.

Automated decision-making is also used in other control systems at a nuclear power plant. For example, when updating an analog control system with a digital control system [A-26], the new design for the EDG I&C system at the Kola nuclear plant incorporated probabilistic analyses and factor in available operating experience, research and technological developments into the design.

The risk-informed decision-making for the new control system was made during the design process—(static) probabilistic assessments were used to identify potentially critical situations and access the control system's response allowing Diakont engineers to make iterative adjustments to the logic and redefining responses to various sensors and operating scenarios. These analyses were run repeatedly as the system and software was developed. Following production, all failure scenarios were replicated using a test fixture to validate performance.

Similar to a supervisory control system, the new control system for the EDGs used probabilistic assessments to access the control system's responses to potentially critical situations. Unlike the EDG control system at Kola however, a supervisory control system will use probabilistic information directly during operations on a real-time basis by capturing the information in a rule-based decision module, a weighted decision module, or a probabilistic module. Thus, the supervisory control system will use probabilistic information to evaluate the operating alternatives based on actual plant conditions (e.g., equipment failures, equipment out of service, etc.).

In addition, the control system at Kola shows the value of continuous diagnostics. According to Rosenergoatom, the implementation of continuous diagnostics proved to be the most significant contributor to the design. This collective functionality presented a paradigm shift in operating principles for an EDG control system—the evolution from a sensitive, reactive system prone to initiating EDG shutdown, to a proactive system that maximized EDG uptime and kept EDGs operational. This is an important goal of the supervisory control system.

A.3.6 Highly Autonomous Driving

The agent-based decision-making paradigm is emerging for driverless vehicles. Agents of many kinds are being used more frequently in a variety of fields. Agent-based modeling has developed as a modeling algorithm for complex systems composed of interacting and independent units. Agents have behaviors that are described by simple rules; agents interact with each other, which produce a system-wide behavior. An agent's intelligence may range from pre-determined roles and responsibilities to a learning capability. Agent-based applications are appearing in numerous disciplines—the stock market, molecular self-assembly, and biological science.

The behavior of drivers is often modeled by a two-layered agent architecture: tactical layer and strategic layer. The tactical layer orients to short time scale driving. The strategic layer addresses complex

problems, such as route choice and decision-making. A particularly difficult task in driving is recognizing when it is safe to make a left turn. An incorrect decision can be disastrous.

Some of the most advanced and exciting applications in automated decision-making appears in autonomous driving as recently attempted by several major universities and corporations, such as Stanford University, Google Col, and BMW Group.

A.3.3.5 Stanford University

Stanley, an autonomous car created by Stanford University's Stanford Racing Team in cooperation with the Volkswagen Electronics Research Laboratory (ERL).

Both vehicles are equipped with custom-built systems to enable direct actuation of throttle, brakes, transmission and steering. Vehicle data is accessed by computer control system through the vehicle's controller area network (CAN) bus interface.

The autonomous control system is comprised of three top-level functional elements: (1) perception, (2) planning, and (3) control. The processing unit used in Stanley, consists of approximately thirty modules executed in parallel.

Both implementations, i.e. Stanley and Junior, the software architecture is modular. The modules run asynchronously and transmit data from sensors to actuators in a pipeline fashion, i.e., first-in first-out (FIFO). The modular architecture reduces the system reaction time, which is roughly 300 ms. The system architecture is broken into six layers [A-27]:

1. sensor interface,
2. perception,
3. control,
4. vehicle interface,
5. user interface, and
6. global services.

The *sensor interface layer* comprises a number of software modules concerned with receiving and time-stamping all sensor data. The layer receives data from each laser sensor at 75 Hz, from the camera at approximately 12 Hz, the GPS and GPS compass at 10 Hz, and the vehicle controller area network (CAN) bus at 100 Hz. This layer also contains a database server with the course coordinates (RDDF file).

The *perception layer* maps sensor data into internal modules. The primary module in this layer is the UKF vehicle state estimator, which determines the vehicle's coordinates, orientation and velocities. Three different mapping modules build 2-D environment maps based on lasers, the camera, and the radar system. A road finding modules uses the laser-derived maps to find the boundary of a road so that the vehicle can center itself laterally. Finally, a surface assessment module extracts parameters of the current road for the purpose of determining safe vehicle speeds.

The *control layer* is responsible for regulating the steering, throttle, and brake response of the vehicle. A key module is the path planner, which sets the trajectory of the vehicle in steering- and velocity-space. This trajectory is passed to two closed-loop trajectory tracking controllers, one for the steering control and one for brake and throttle control. Both controllers send low-level commands to the actuators that faithfully execute the trajectory emitted by the planner. The control layer also features a top-level control module, implemented as a simple finite state machine. This level determines the general vehicle mode in

response to user commands received through the in-vehicle touch screen or the wireless E-stop, and maintains gear state in case backwards motion is required.

The *vehicle interface layer* serves as the interface to the robot's drive-by-wire system. It contains all interfaces to the vehicle's brakes, throttle, and steering wheel. It also features the interface to the vehicle's server, a circuit that regulates the physical power to many of the system components.

The *user interface layer* comprises the remote E-stop and a touch-screen module for starting up the software.

The *global services layer* provides a number of basic services for all software modules. Naming and communication services are provided through a special inter-process communication (IPC) toolkit. A centralized parameter server maintains a database of all vehicle parameters and updates them in a consistent manner. The power server regulates the physical power of individual system components. Another module monitors the health of all systems components and restarts individual components when necessary.

Estimation of vehicle's state is essential precision driving. Inaccuracies in pose estimation can cause the vehicle to drive outside the corridor, or build terrain maps that do not reflect the state of the robot's environment, leading to poor driving decisions. Stanley's *vehicle state* comprises a total of 15 variables.

Driving decisions are made using path-planning methods, which generate multiple local trajectory options. These options are then weighed against a number of criteria, such as minimization of the risk of collision as well as favoring the road centers over paths closer to the periphery.

For global path planning, a dynamic programming algorithm—called A*—is employed to search for shortest path, which minimizes the expected drive time to target location. The global search typically takes about one second to execute and generate an optimal solution. However, unexpected changes in the terrain (for Stanley) or traffic complications (for Junior), such as lane changes, require local but discrete refinements to the global solution.

Furthermore, for *unstructured navigation*, such as driving in parking lots or for parking, Junior utilizes a modified version of the A* algorithm, which searches for shortest path relative to the vehicle's map using *search trees*.

Junior employs a decision module to minimize risk of getting stuck in unpredictable environments, such as urban driving conditions. The decision module is implemented as a finite state machine. While the path planner, which can be considered as the global optimizer, works best under normal driving conditions, the finite state machine allows for taking into account driving surprises. Following an impasse, the finite state machine gradually transitions to increasingly unconstrained driving.

A.3.3.6 Google Self-Driving Car

Google's *self-driving car* technology, or occasionally referred to as the *Google driverless car*, is a demonstration concept car for autonomous driving. Google's autonomous vehicle is an improvement on Stanley and Junior.

Originally implemented on a Toyota Prius, Google's concept vehicle includes a light detection and ranging (LIDAR) system, which uses a 64-beam laser. The laser allows the vehicle to generate a detailed 3D map of the environments. The processor then combines the imagery with high-resolution maps,

producing different types of models that allow it to drive itself while avoiding obstacles and respecting traffic laws.

The vehicle also carries other sensors, which include four radars mounted on the front and rear bumpers that allow the car to detect obstacles and other vehicles in close proximity to deal with fast traffic on freeways; a camera positioned near the rear-view mirror that detects traffic lights; a GPS; an accelerometer for inertial measurements; and a wheel encoder that determines the vehicle's location and keep track of its movements. Since details on Google's technology are not publicly available, we believe a review of Stanley's autonomous control technology gives insight into the specifics.

Before a test drive, the engineers drive along the route several times to gather data about the environment. During the autonomous driving session, the computer compares the environmental data being acquired currently to the data previously recorded—an approach proven to be useful to differentiate pedestrians from stationary objects like poles and mailboxes.

Google's fleet of robotic cars are reported to have driven in excess of 300,000 km, including driving in city traffic, busy highways and mountainous roads with only occasional human intervention.

A.3.3.7 BMW Highly Autonomous Driving

The BMW Group Research and Technology is currently developing highly automated assistance and active safety systems for future car generations, and investigating their market potentials [A-28]. An example of this is the *Emergency Stop Assistant* (ESA), which takes over vehicle control, safely steers the vehicle to the side of the road, and stops if the driver suffers a health irregularity, such as an acute problem with the cardiovascular system, or perhaps, even a heart attack [A-29, A-30]. In a freeway scenario with right-hand traffic, this active safety system is reported to conduct secured *automated lane change* maneuvers to the right to reach the breakdown lane to stop the vehicle.

Highly Autonomous Driving (HAD) technology advances existing vehicular automation by providing additional driving assistance. The information concerning the host vehicle's environment (road, lanes, and objects) is provided online by the vehicle's sensors and by a high-precision digital map. The raw sensor and map data are processed within the subsequent *Perception* unit. The *Object Tracking* module fuses the data of multiple sensors and generates a global object list with the objects' attributes [A-31, A-32]. The *Localization* module determines the location of the host vehicle within the digital map. All relevant information is forwarded to the *Functionality* unit. The *Driving Strategy* module makes decisions regarding driving maneuvers. These maneuvers are derived from the general objectives, e.g., ESA or HAD, and the traffic situation determined based on the digital image of the host vehicle's environment. The respective maneuvers are realized by the *Trajectories and Control* module. These maneuvers are finally executed by the steering, acceleration, and deceleration actuators. Furthermore, the system provides information, such as the current system states, to the driver and occupants via the human–machine interface (HMI), and further controls host vehicle functionalities, such as the indicators, via the vehicle bus.

Based on traffic conditions, decisions should be made to identify suitable driving maneuvers for the vehicle. BMW approach uses a hierarchical hybrid decision-making process, which has a limited number of discrete system states that classify various driving maneuvers. This approach combines a finite state machine, which handles the *deterministic* portion, with decision trees, which take into account *probabilistic* aspects of decision-making.

A *lane change* (LC) request is executed if the maneuver is desired and feasible. If an LC is desired, but not feasible, the *lane change gap approach* (LCGA) strategy is applied. This approach helps avoid some

of the potential problems with probabilistic approaches with direct influence on driving maneuvers, which were shown to lead to nondeterministic behavior, or sometimes infringement of traffic rules [A-33, A-34].

This combined approach increases robustness of the decision-making process by adding an additional feasibility examination of driving requests. While the probabilistic portion takes into account environmental and systematic uncertainties and generates a desired driving behavior, the rule-based deterministic portion of the decision-making module considers the worst-case conditions, and eliminates or avoids unfeasible driving requests.

The probabilistic portion of the decision-making module is implemented using a modified version of utility theory, which evaluates the utility of each lane, i.e. suitability, for the vehicle, and generates LC requests on the basis of lane utilities. This approach allows for incorporating uncertainties associated with the sensor data as well as the uncertainties in the processed information from the *Perception* module.

The utility function consists of multiple factors that evaluate the utility of a lane based on various comfort and safety criteria.

Weights for the utility function are determined based on a number of factors, including but not limited to, general traffic characteristics, such as the average longitudinal gap size between objects and the average velocity on a lane, or the specific velocities and distances of single objects. For instance, a low average gap size decreases the utility of a lane due to disadvantages regarding safety.

The uncertainties from the sensor data are taken into account in calculation of individual utilities by using a normal distribution. At each evaluation cycle, utilities of current and two adjacent lanes—if applicable—are calculated, which yields three normally distributed utilities.

A.4 REFERENCES

A-1. S. M. Cetiner et al., "Definition of Architectural Structure for Supervisory Control System of Advanced Small Modular Reactors," ORNL/TM-2013/32, SMR/ICHMI/ ORNL/TR-2013/04, August 2013.

A-2. Automatic Train Control in Rail Rapid Transit, NTIS order #PB-254738, May 1976.

A-3. S. Wegele, R. Slovák, E. Schnieder, "Automatic Dispatching of Train Operations Using a Hybrid Optimization Method," 8th World Congress on Railway Research, May 18-22, 2008, Seoul, Korea.

A-4. A. Giua and Carla Seatzu, "Modeling and Supervisory Control of Railway Networks Using Petri Nets," IEEE Transactions on Automation Science and Engineering, Vol. 5, No. 3, July 2008.

A-5. T. Schlechte, "Railway Track Allocation: Models and Algorithms," Ph.D Dissertation, Technical University of Berlin, Department of Optimization (2012).

A-6. A. Siahvashi and Bijan Moaveni, "Automatic Train Control based on the Multi-Agent Control of Cooperative Systems," Journal of Mathematics and Computer Science, 1(4), pp. 247–257 (2010).

A-7. W-H. Lai, K-Z. Hung, "Optimizing New Chain Retail Store Area by Using Voronoi Diagram Technique," Technology Management for Global Economic Growth (PICMET), 2010 Proceedings of PICMET '10, 18–22 July 2010.

A-8. Z. Tang, C. Guo, P. Hou, and Y. Fan, "Optimal Siting of Electric Vehicle Charging Stations Based on Voronoi Diagram and FAHP Method," Energy and Power Engineering, 5, pp. 1404–1409 (2013).

A-9. C. H. Kepner and B. B. Tregoe, The New Rational Manager (2013)-An Updated Edition for the New World, January 1, 2013.

A-10. T. Nagashima, K. Nakamura, K. Shirakawa, and S. Komiya, "A Proposal of Risk Identification Based on the Improved Kepner-Tregoe Program and its Evaluation," Int. Journal of Systems Applications, Engineering & Development, 2(4), (2008).

A-11. J. S. Parker and J. D. Moseley, "Kepner-Tregoe Decision Analysis as a Tool to Aid Route Selection," Part 1 of a three-part article, Organic Process Research & Development, 12, pp. 1041–1043 (2008).

A-12. T. M. Nguyen, J. Schiefer, and A. M. Tjoa, "Sense & Response Service Architecture (SARESA): An Approach towards a Real-time Business Intelligence Solution and its use for a Fraud Detection Application," DOLAP'05, November 4–5, 2005, Bremen, Germany (2005).

A-13. J. S. Litt, et al., "A Survey of Intelligent Control and Health Management Technologies for Aircraft Propulsion Systems", NASA/TM—2005-213622, ARL–TR–3413, May 2005.

A-14. D. Enns, D. Bugajski, R. Hendrick, G. Stein, "Dynamic Inversion: An Evolving Methodology for Flight Control Design," Advisory Group for Aerospace Research and Development, AGARD Conference Proceedings 560, AGARD-CP-560, January 1995.

A-15. G. Ducard et al., Fault-tolerant Flight Control and Guidance Systems, Springer 2009.

A-16. T. A. Johansen, T. I. Fossen, Control allocation—A survey, Automatica, 49(5), pp. 1087–1103, May 2013.

A-17. J. Bosworth, D. Enns, "Nonlinear Multivariable Flight Control," Success Stories for Control, The Impact of Control Technology, IEEE Control Systems Society, February 2011.

A-18. How, J.P., Fraser, C., Kulling, K.C., Bertuccelli, L.F., Toupet, O., Brunet, L., Bachrach, A., Roy, N., Increasing Autonomy of UAVs, Decentralized CSAT Mission Management Algorithm, IEEE Robotics & Automation Magazine, June 2009.

A-19. Lundell, M., Tang, J., Nygard, K., Fuzzy Petri net for UAV decision-making, Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, 2005.

A-20. Wu, L., Niu, Y., Zhu, H., Shen, L., Modeling and characterizing of unmanned aerial vehicles autonomy, 2010 8th World Congress on Intelligent Control and Automation (WCICA), July 2010

A-21. Lacroix, S., Alami, R., Lemaire, T., Hattenberger, G., Gancet, J., Decision-making in multi-UAVs systems: Architecture and Algorithms, Multiple Heterogeneous Unmanned Aerial Vehicles, Springer Tracts in Advanced Robotics, Volume 37, 2007, pp 15-48

A-22. Doherty, P., Granlund, G., Kuchinski, K., Nordbert, K., Skarman, E., Wiklund, J., The WITAS Unmanned Aerial Vehicle Project (2000), Proceedings of the 14th European Conference on Artificial Intelligence, 2000.

A-23. Zuo, Y., Peng, Z., Liu, X., Task Allocation of Multiple UAVs and Targets Using Improved Genetic Algorithm, 2011 2nd International Conference on Intelligent Control and Information Processing (ICICIP), (Volume: 2), July 2011.

A-24. RG 1.174, Rev. 2, "An Approach for Using Probabilistic Risk Assessment in Risk-Informed Decisions on Plant-Specific Changes to the Licensing Basis," US Nuclear Regulatory Commission, May 2011.

A-25. RG 1.177, Rev. 1, "An Approach for Plant-Specific, Risk-Informed Decision-making: Technical Specifications"), US Nuclear Regulatory Commission, May 2011.

A-26. Title 10 of the Code of Federal Regulations, Part 50, Domestic Licensing of Production and Utilization Facilities, §50.65, "Requirements for monitoring the effectiveness of maintenance at nuclear power plants."

A-27. US Nuclear Regulatory Commission, "Pressurized Water Reactor B&W Technology Cross-training Course R-326C, Manual Chapter 9.0, Integrated Control System."

A-28. American Nuclear Society, "Updating Plant EDGs with Intelligent Digital Control Systems," Nuclear News, LaGrange Park, IL, July 2012.

A-29. S. Thrun, et al., "Stanley: The Robot That Won the DARPA Grand Challenge," The 2005 DARPA Grand Challenge, Springer Tracts in Advanced Robotics, 36, p. 1 (2007).

A-30. M. Ardelt, C. Coester, N. Kaempchen, "Highly Automated Driving on Freeways in Real Traffic Using a Probabilistic Framework," IEEE Transactions on Intelligent Transportation Systems, 13(4), pp. 1576–1585 (2012).

A-31. M. Ardelt, P. Waldmann, N. Kämpchen, F. Homm, "Strategic descision-making process in advanced driver assistance systems," Proc. IFAC Symp. AAC, Munich, Germany (2010).

A-32. BMW Group PressClub Global, "Stopping Safely in an Emergency," [Available Online at http://www.press.bmwgroup.com/] (2010).

A-33. R. Schubert, "Evaluating the utility of driving: Toward automated decision-making under uncertainty," IEEE Transactions on Intelligent Transportation Systems, 13(1), pp. 354–364 (2011).

A-34. M. Aeberhard, S. Paul, N. Kämpchen, T. Bertram, "Object existence probability fusion using Dempster-Shafer theory in a high-level sensor data fusion architecture," Proc. IEEE Intelligent Vehicle Symposium, Baden-Baden, Germany, pp. 770–775, June 2011.

A-35. J. M. Wille, F. Saust, M. Maurer, "Stadtpilot: Driving autonomously on Braunschweig's inner ring road," in Proc. IEEE Intelligent Vehicle Symposium, pp. 506–511 (2010).

A-36. J. M. Wille, F. Saust, M. Maurer, "Comprehensive treated sections in a trajectory planner for realizing autonomous driving in Braunschweig's urban traffic," Proc. 13th ITSC, pp. 647–652 (2010).