Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/6180--01-8583

Supervisory Control System for Ship Damage Control: Volume 4 — Intelligent Reasoning

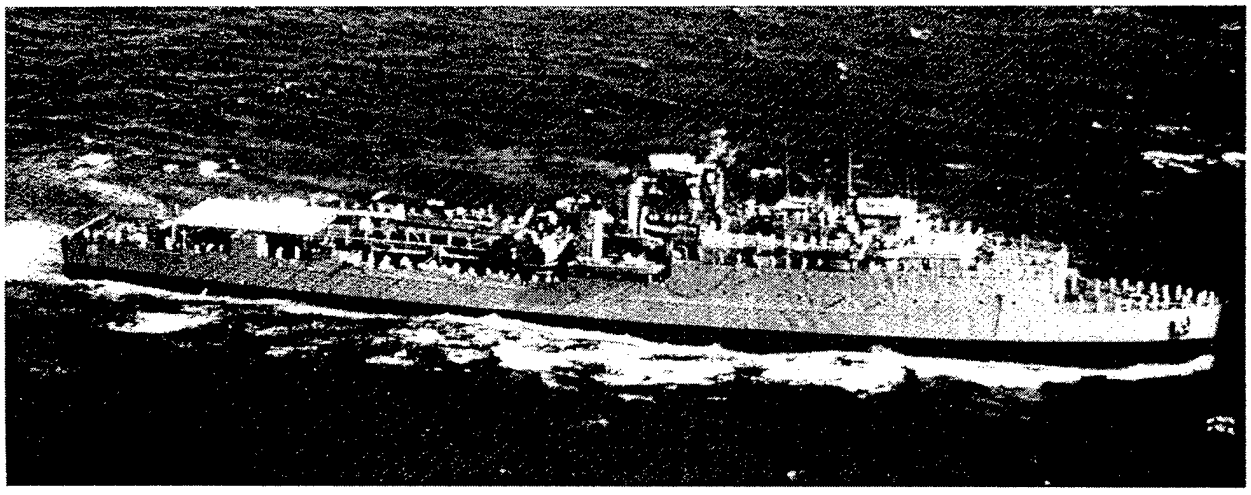
EUGENE GROIS
DAVID C. WILKINS
IAN BEARMAN
MICHAEL BOBAK
AARON BRADY
PAUL HEBBLE
MARK MILLER
SEAN SITTER

Beckman Institute
University of Illinois, Urbana, Illinois

PATRICIA A. TATEM
FREDERICK W. WILLIAMS

Navy Technology Center for Safety and Survivability
Chemistry Division

September 28, 2001



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 28, 2001	Final FY 1999-FY 2001

4. TITLE AND SUBTITLE

Supervisory Control System for Ship Damage Control: Volume 4 — Intelligent Reasoning

5. FUNDING NUMBERS

PE - 63508N

6. AUTHOR(S)

E. Grois,* D.C. Wilkins,* I. Bearman,* M. Bobak,* A. Brady,* P. Hebble,* M. Miller,* S. Sitter,* P.A. Tatem, and F.W. Williams

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory, Code 6180
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6180--01-8583

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

*Beckman Institute, University of Illinois, Urbana IL 61801

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The design for the Damage Control-Supervisory Control System presented in this report conforms to the goals of the DC-ARM project. It provides rapid and reliable situation awareness and generates effective casualty response. A number of design elements remain to be refined, but the system as a whole has been implemented and subjected to limited simulated tests. In these tests, it has proven successful at detecting and controlling casualty events while minimizing the use of resources and nuisance events.

14. SUBJECT TERMS

Flooding
Fire
Damage control
Supervisory control
Automation

15. NUMBER OF PAGES

48

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS

SUPERVISORY CONTROL SYSTEM FOR SHIP DAMAGE CONTROL:

VOLUME 4 – INTELLIGENT REASONING

1. Introduction

This report presents a philosophy and technical approach for automated shipboard damage control within the framework of the Damage Control – Automation for Reduced Manning (DC-ARM) initiative. The DC-ARM program is an Office of Naval Research (ONR) initiative under PE 63508N. In developing a Supervisory Control System that can provide timely and accurate situation awareness, as well as take automated measures to control casualty, we are faced with a number of design and engineering problems. There are three key challenges; the dynamic, real-time nature of the domain, the necessity of making critical decisions under conditions of uncertainty and incomplete information about the system state, and the effective integration of information and actions of personnel with those of automated systems.

We deal with the first two challenges by utilizing efficient Bayesian and rule-based computation techniques with a sound probabilistic bases, providing accurate and robust inference under all conditions of uncertainty. The integration of human and system actions and information is accomplished through a creative design of interfaces for information entry, combined with a powerful and flexible object-based model of the domain.

The present report presents the current design of the Supervisory Control System. However, it must be noted that many aspects of the system are in the process of evolution. In particular, on-going research is aimed at improving the accuracy of casualty identification, and novel techniques are being investigated for the efficient management of limited damage control resources.

2. Background

Time and automation has made it possible to reduce the manning of Navy ships. It is clear that a reduction in the size of a ship's crew will require a high level of technological sophistication in order to facilitate situation awareness for efficient and effective decision making, as well as reliable mechanisms to replace some of the actions presently performed by crew members.

The DC-ARM program addresses these issues in the context of shipboard damage control (DC). Its goal is the integration of effective and reliable technologies necessary for a substantial reduction in DC manning. The automated system ensuing from this work will be expected to effectively control shipboard fires, smoke, and flooding (internal only) situations resulting from survivable casualty conditions.

Manuscript approved August 28, 2001.

3. Objectives

Several objectives have previously been defined for the Supervisory Control System [Bradley, et al., 2000; Peatross, et al., 2000]. The system must

- Enable timely and accurate situation awareness for human supervisors sufficient for them to define relevant damage control objectives.
- Provide efficient management of limited resources.
- Initiate preemptive actions to prevent or mitigate the effects of damage before the damage actually occurs.
- Initiate damage control actions in response to casualty.
- Monitor automated responses to damage and adjust them as needed.

The goals for the present version of the system are largely dictated by the "FY 2000 DC-ARM Demonstration Test Plan" [Peatross, et al., 2000]. There are essentially three high-level requirements that the system must satisfy: (1) It must be capable of acquiring data from casualty detection sensors located throughout the ship; (2) It must provide an accurate and timely casualty classification both in terms of nature of the casualty (e.g., No Fire, Nuisance Event, Fire) and (3) It must generate a correct response to the casualty, taking into account constraints such as availability of resources and personnel, and must do so in a timely manner.

4. Overview of the DC-SCS System

The following is a detailed breakdown by sub-system of the requirements for the intelligent reasoning portion of the DC-SCS System:

- Data Acquisition
  - *Shadwell* Sensors – MASSCOMP interface
    - Thermocouples
    - Optical density meters (ODMs)
    - Air pressure sensors
    - Gas sensors
    - Doors/hatches sensors
    - Firemain pressure sensors
  - Early Warning Fire Detection (EWFD) sensors
  - Commercial Off the Shelf (COTS) Fire Sensors
  - Human personnel reports
- Casualty Detection and Classification
  - Pre-processing of sensor data (filtering)

- Probabilistic identification of casualty type and severity from *Shadwell* sensor data
- Verification of internal casualty estimate with external evidence (i.e., EWFD/COTS data and personnel reports)

- Casualty Response
  - Assessment of damage to resources critical to casualty response (e.g., personnel, sensors, water mist systems, firemain pressure loss)
  - Estimation of available resources and personnel (from acquired data and damage assessment)
  - Clustering of casualty compartments and events for efficient damage control
  - Generation of appropriate casualty response plan
  - Incorporation of supervisory override and generation of new casualty response plan if necessary

The DC-SCS System meets these requirements through a combination of a variety of data processing and artificial intelligence techniques. Figure 1 presents a high-level view of the approach taken in the system for shipboard automated situation awareness and casualty response. The Data Acquisition module accepts inputs from all *Shadwell*, EWFD, and COTS sensors, as well as personnel reports. A filtering procedure is performed on these data to detect bad sensors and/or transmission problems, as well as to remove spurious readings and transients. The Casualty Identification module uses probabilistic techniques to detect and classify casualty conditions from the acquired data. This information is then provided to the Damage Assessment module, which also incorporates raw sensor data to estimate resource loss resulting from the casualty. The Resource Availability module determines which resources are available for damage control by utilizing resource loss information from the Damage Assessment module, and keeping track of resources in use. Finally, the Casualty Response module uses casualty type and severity information from the Casualty Identification module and resource availability data to generate an effective plan for controlling the casualty.

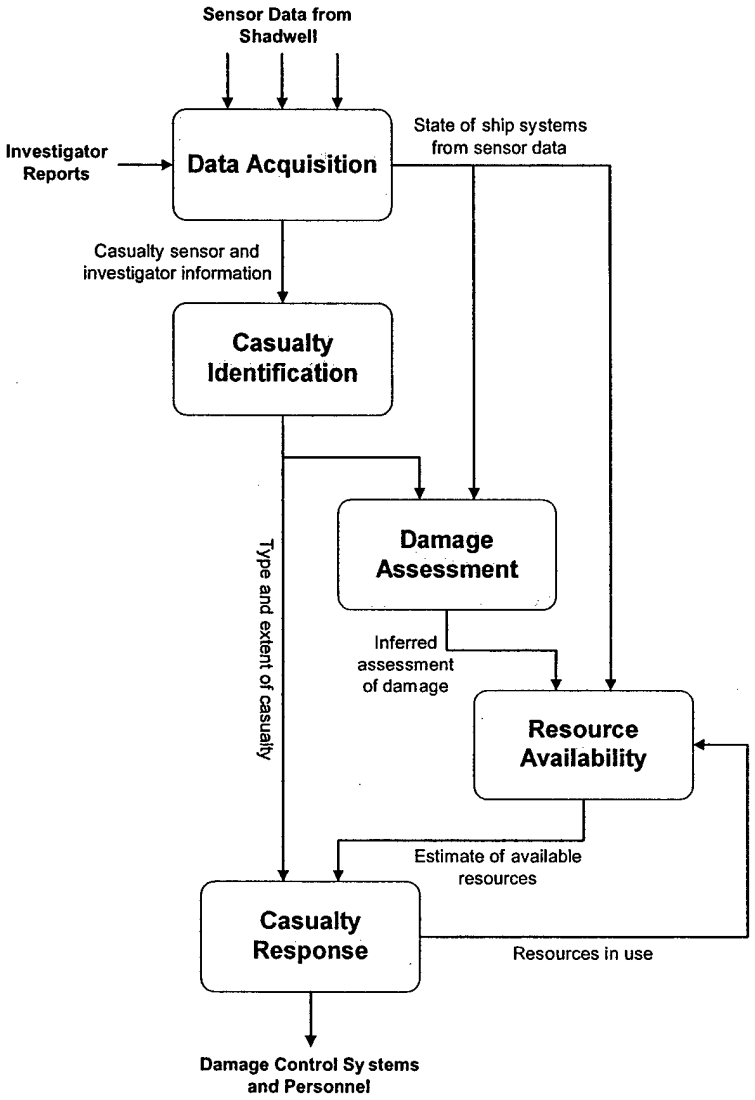


Figure 1. Information and control flow in the DC-SCS System.

5. In-Depth Discussion of Design

This section presents a detailed discussion of the DC-SCS System design.

5.1 Data Acquisition

Due to the differing formats and fluctuating quality of the raw situation awareness data as it arrives from sensors and information entry interfaces, a pre-processing step must be added before these data can be used for analysis and reasoning.

5.1.1 Sensor Data Filtering

While the sensors installed on the ex-USS *Shadwell* are fairly reliable, they are still prone to occasional failure, which may be due to an intrinsic malfunction and/or damage caused by any number of conditions, such as an explosion, fire, or even contact with water. Furthermore, RF interference on the data transmission lines caused by electromechanical equipment or devices, such as welding torches, can lead to spurious sensor readings. The sensor data filter is designed to eliminate the effects of these conditions.

Three types of situations may be defined that can lead to erroneous sensor data. First, there is a small-amplitude oscillatory component, which is inherent in all sensor readings. This phenomenon is caused by minor physical changes in the sensor resulting from such things as small fluctuations in atmospheric temperature and humidity. While not critical, these oscillations do affect the accuracy of sensor data, and the analysis that builds upon them.

Second, transients on the data transmission lines and sizeable atmospheric changes may cause spurious data to be read. Such spurious data are characterized by large incremental gaps in an otherwise continuous stream of values. These effects are temporary (often appearing for only a second or two), but they can lead to erroneous interpretations of the situation if not neutralized.

Finally, a particular sensor may have been disconnected or damaged due to casualty. In this case it will usually generate a constant erroneous reading. As with transient faults, these readings can significantly affect situation assessment and thus must be removed from the data stream.

The small-amplitude oscillatory component may be removed by taking the root-mean-square (RMS) average of the data stream. To this end, a time interval moving window is used. The size of the interval is made variable from 3-15 seconds[1], this parameter usually being set at 5-10 seconds. The ability to modify the interval size is necessary for empirical determination of the optimal size, as well as permitting adjustment to varying ambient conditions. At any point in time for a given sensor, the RMS average of the values in the window (which includes the last actual sensor reading plus several prior readings) is taken as the current reading.

In order to detect and discard spurious values, the notion of rate-of-change is used. As each RMS average is calculated every second, the rate-of-change of the RMS average relative to its previous value is also determined. If the absolute value of this rate-of-change is above a certain threshold, the sensor is marked as "bad" and not used in decision-making until the condition is

---

[1] The assumption is made that data are read from the MASSCOMP interface at 1 Hz.

alleviated and its status is returned to "good". The rate-of-change threshold can be between 5% and 50%, with a typical setting being 10% to 15%.

The rational for doing this rate-of-change determination based upon the RMS average of the entire window and not each successive *actual* sensor reading is that with individual sensor readings we have no way of knowing if a large jump in the reading was caused by, for example, direct application of a torch to a thermocouple (hence legitimate) or is just a spurious value. If, however, the rate-of-change of one reading is truly enormous enough to be unacceptable, it will pull the rate-of-change of the RMS average of the entire window out of bounds and thus force the sensor to be marked as "bad" anyway. This may eliminate a sensor that in fact could have momentarily overranged.

A sensor is also marked "bad" if *within a window* there are two *successive* changes in direction. The threshold amplitude for detecting this type of oscillation may be significantly less than the upper-lower-bound band (typically 1/2 the height of the band).

When a sensor is marked as "bad", it, however, continues to be sampled by the filter since the problem-causing condition may be transient (or the sensor may be replaced at some point) and may become useable again in the future. A given number (typically two to three) of successive "OK" windows is sufficient to mark the sensor as "good".

It should be noted that use of a time interval window adds a "memory component" to sensor readings and tends to delay slightly the indication of any physical changes as seen by the system. However, it removes the effect of any spurious values and transient components and makes the sensor readings much more accurate and reliable.

Another point is that this approach relies on the fact that sensors are sampled once a second. It is important that the sample rate does not change, or if there is a possibility that it might, a mechanism should be put in place that permits the dynamic adjustment of the moving window size with the change in sample rate. In any case, however, the sample rate should really be at least 1 Hz. This is because we want the system to react to changes in sensor reading with a delay of no more than several seconds, and the need to maintain a moving time interval window already introduces a delay equal to the size of the window. Thus, we want to keep this to a minimum by sampling fairly rapidly to include a significant number of samples in the window over a relatively short time interval.

5.1.2 Pre-processing of EWFD/COTS Data and Human Personnel Reports

EWFD, COTS, and personnel reports data are used primarily in the verification of the Casualty Classifier output. In order for such a comparison to be possible, however, the varying formats of the external data must be brought in line with the internal format.

5.2 Casualty Identification

While COTS are quite reliable when it comes to detecting real fires, they are poor at distinguishing between real fires and nuisance events. This is a highly debilitating shortcoming of the COTS sensors, since responding to nuisance events as if they were real fires wastes expensive ship resources, perhaps even at the expense of critical real fires raging elsewhere [Peatross, et al. 2000]. EWFD sensors accurately identify real fires and appear to do better with

nuisance events than COTS sensors, but still do not provide optimal casualty classification [Gottuk, et al. 1998, 1999]. We would like our module to use raw data from the *Shadwell* sensors to both reliably detect real fires, as well as accurately distinguish between from nuisance events. While water mist and halon-type agents will in most cases be automatically activated and are effective for all classes of fire, the knowledge of fire class could be used in the decision process. Depending on the class of the fire, its temperature and rate of propagation vary widely. In the case of limited availability of damage control resources it may be necessary to allow one fire to burn, while directing all available resources to another more dangerous fire. Furthermore, damage control, even with full implementation of DC-ARM, will not always be automatic. If human firefighters are called upon to intervene, they will use techniques that are highly dependent on the class of fire in progress [Williams, et al. 1993, 1997].

The Casualty Identification module uses all available ship and systems state information to detect the type and severity of a casualty condition. Figure 2 shows data and control flow in the module. The Casualty Classifier sub-module is a Bayesian belief network structure that detects and identifies a crisis by reasoning over data collected from sensors mounted throughout the ex-*Shadwell.* The sensors currently utilized are thermocouples, ODMs, and gas sensors.

The Bayesian belief network is a directed acyclic graph (DAG) that encodes causal relationships among events, together with conditional probabilities specifying the degree to which these relationships hold. This paradigm provides optimal accuracy and robustness in the presence of varying level of input fidelity [Pearl, 1988]. Once a topology for the network is defined, the quantification of network variables can be learned very efficiently from statistical data. However, because explicit causal relationships among variables are required, learning the topology of the network from statistical data is difficult. For this reason, the aid of an expert is generally required to delineate the causal relationships among the variables. Alternatively, a generate-and-test methodology may be utilized, where multiple models are generated with the help of a bias (such as some prior knowledge) and evaluated with respect to the degree they satisfy the statistical data [Heckerman, 1995].

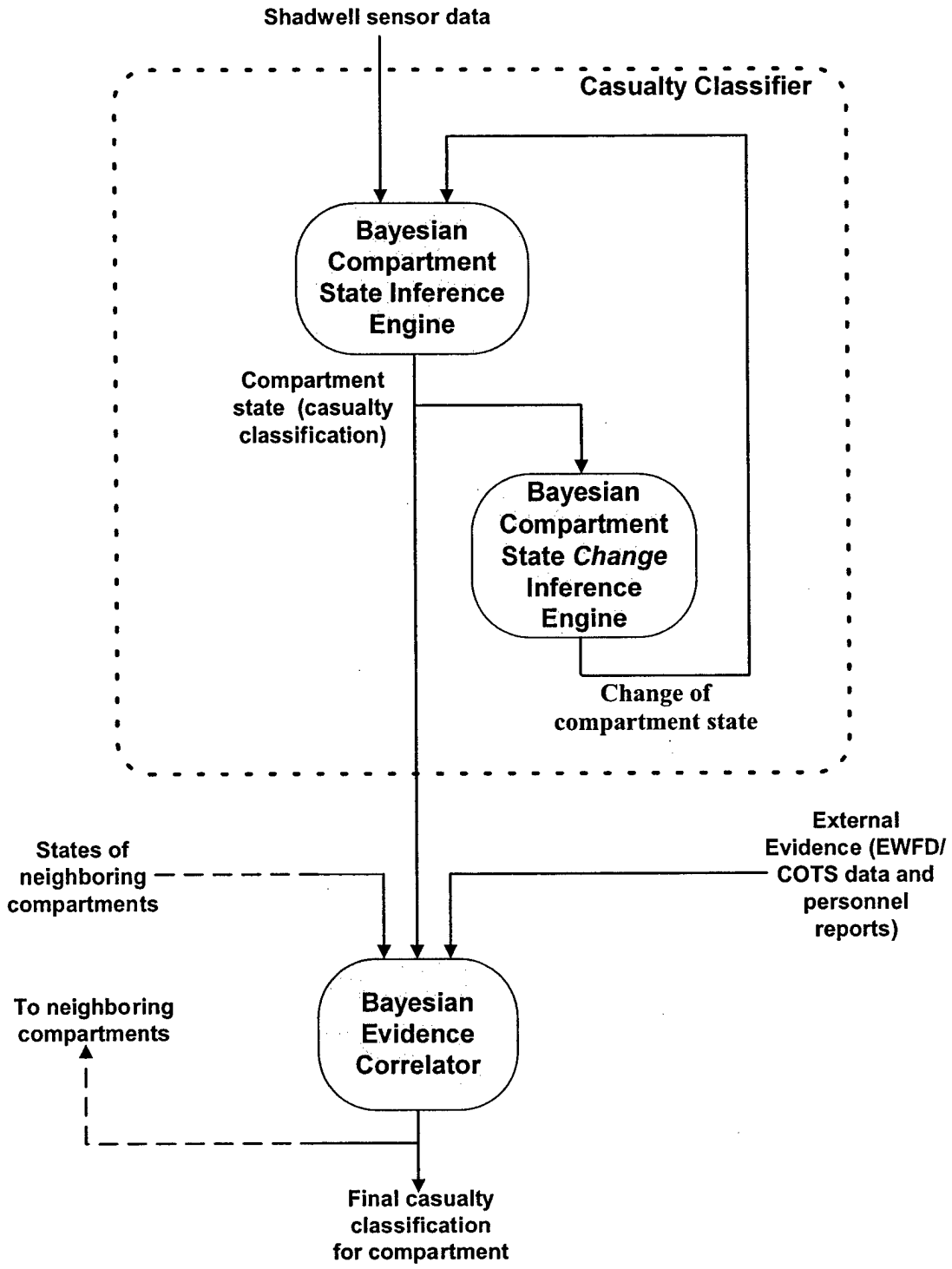


Figure 2. Flow diagram for the Casualty Identification Module, showing control and information flow for one compartment.

In designing the Casualty Classifier we took the second approach. The models of causal relationships for shipboard casualty were constructed based upon information gleaned from the research literature [Gottuk, et al. 1999; Grosshandler, 1995; Gottuk, et al., 1998]. The network continually monitors sensor data to infer the *state* of the compartment, as well as the *vector*

change in the state. The vector change of the state influences the state assessment through a feedback loop. A description of the topology of the Casualty Classifier belief network may be found in Section 6.

As Figure 2 illustrates, once the Casualty Classifier determines the state of a compartment, this evaluation is input to the Bayesian Evidence Correlator. This sub-module combines the evaluation with external information (i.e., EWFD/COTS data and personnel reports) and the states of neighboring compartments to produce a final casualty assessment for the compartment.

5.3 Casualty Response

The overarching objectives for the Casualty Response module are to produce an *effective* casualty response plan at the *lowest possible computational cost*. By an effective plan we mean one that satisfies all constraints, which include the damage control goals and resource constraints[2]. The basic approach is to (1) determine the system state (which includes nature of casualty, as well as available resources), (2) attempt to fit an existing stored DC plan to the requirements of the situation (with minor modifications if necessary), (3) if unable to utilize an existing plan, create a new plan from action-level primitives and parts of existing plans. An important fourth element of casualty response is incorporation of supervisory override.

Figure 3 illustrates the architecture of the Casualty Response module. Operation of this module is triggered by the generation of a casualty alarm by the Casualty Identification module, which is accompanied by the type and severity of the casualty. Inside the Casualty Response module this information is input to the System State Inference Engine, together with resource availability data from the Resource Manager. The System State Inference Engine uses this information to generate a set of high-level system constraints, which carry information about the casualty, DC resources, and DC priorities. These constraints are passed onto the High-level Case-based Planner, which submits them to a High-level DC Goals knowledge base and obtains a set of high-level goals. Armed with the goals and system constraints it attempts to fit an existing plan to the situation. If this is not possible, portions of the plan, together with relevant constraints are submitted to the Action-level Planner for modification. If no suitable modification can be made, then the planning process is turned over entirely to the Action-level Planner and a new plan is constructed from action-level primitives.

---

[2] While this is not a basic design objective (at least for the near term), resources should be optimized whenever possible.

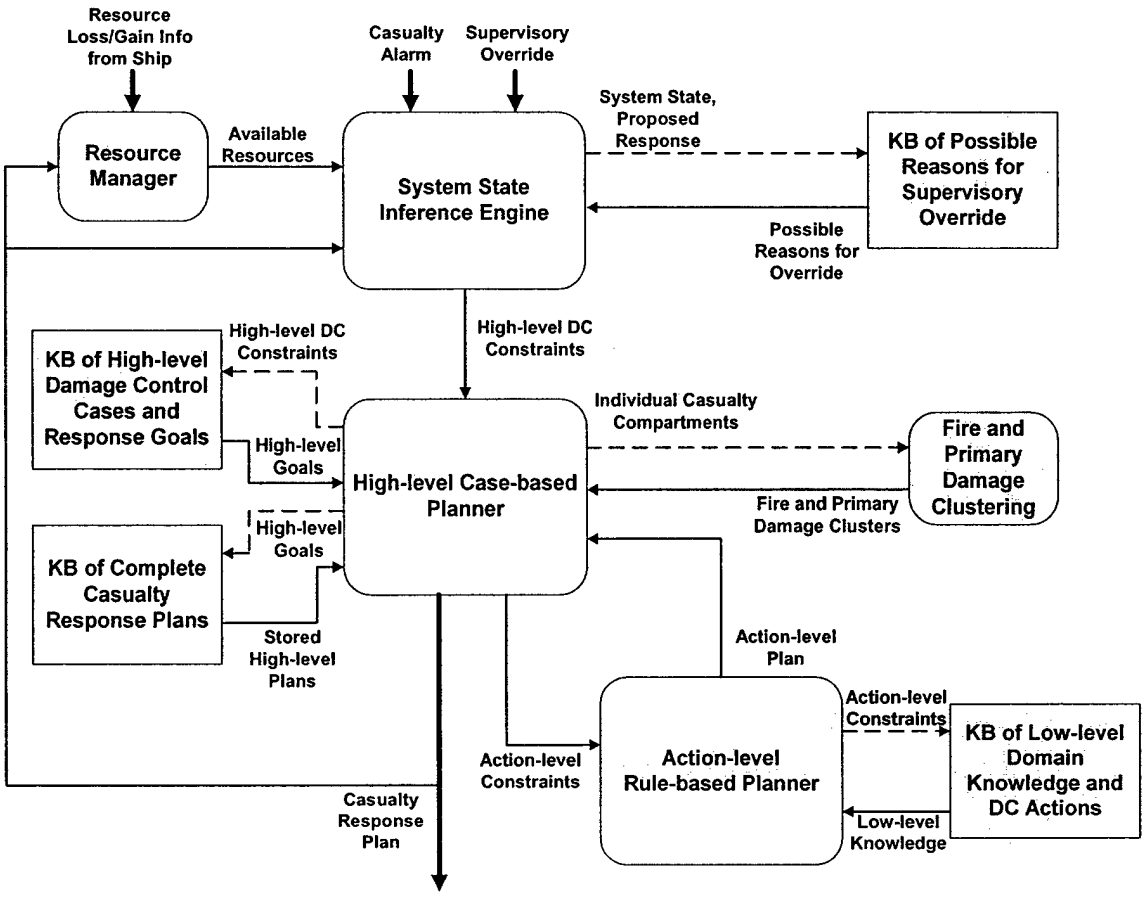


Figure 3. Information and control flow in the Casualty Response Module. ("KB" is Knowledge Base)

5.3.1 The System State Inference Engine and Supervisory Override

The System State Inference Engine uses information about casualty and ship systems to construct a model of the state of the ship and to generate a set of damage control constraints that describe the situation. The module contains a knowledge base of rules, which fire when specific casualty and resource conditions are satisfied, generating the corresponding constraints. When a plan is proposed by the planner, its actions are input to the module, which allows it to estimate the probable resultant changes in the system state.

A potentially problematic situation arises if a supervisory override of the proposed plan is asserted. Since the system must be robust, it should be capable of inferring what caused its plan to be overridden and generate a new plan taking this information into account. We deal with this challenge by having a knowledge base relating possible types of supervisory override (defined by the prevailing system state and proposed casualty response plan) for likely reasons. The System State Inference Engine uses information from this knowledge base to infer a new system state together with the corresponding constraints. The Planner then generates a new casualty response plan based on the new set of constraints.

5.3.2 Fire and Primary Damage Clustering

When several neighboring compartments are on fire, the casualty should be treated as one large fire. Two operations must be performed to make this possible: (1) All the fires on the ship should be assigned to a fire cluster and (2) the dominant fire class of each cluster and the fire severity of each cluster must be determined. Fire clustering should be done by a nearest neighbor-type algorithm. The dominant class for each cluster may be found by comparing the classes of the individual fires, weighted by the areas of the compartments. The fire severity of each cluster can be determined by an average of the severities of the individual fires, weighted by the areas of the corresponding compartments.

Algorithm

Fire Clustering

1. Put all compartments that are classified as being on fire in a set S.
2. Find the compartment with the highest temperature (which is used to estimate a fire center). Call this compartment a fire center. Remove it from S and start a new cluster C_i for it.

   Find all compartments from S that are adjacent to a compartment in C_i. Remove them from S and place them in C_i. Repeat until no more compartments in S are adjacent to a compartment in C_i.

   Repeat 2 until S is empty.

Cluster Fire Class and Severity Determination

3. For each cluster C_i, add up the areas of compartments with each type of fire class. Assign the fire class associated with the largest sum to be the fire class of the cluster.

   Repeat 3 for all clusters C_i.

4. For each cluster C_i, add up the severities of each fire multiplied by the deck area of the corresponding compartment. Divide the sum by the combined deck area of all of the compartments in the cluster.

   Repeat 4 for all clusters C_i.

Inputs

- Compartment state for each compartment:
  - No Fire 0
  - Nuisance Event 1
  - Mild Fire 2
  - Intermediate Fire 3
  - Severe Fire 4

- Fire class – A, B, C, D, X

Output

Set of fire clusters with associated fire classes and fire severities.

Implementation Notes

- As indicated, the compartment state should be encoded by an integer (0 to 4). That way a fire event will be quickly found by searching for a compartment with state ≥ 2.
- The following internal knowledge sources are needed for this algorithm:
  - Adjacency list for all compartments
  - Table of most likely fire classes for each compartment
  - Table of compartment deck areas
- The output of the algorithm should be in the form of a list of clusters, each of which contains a list of included compartments, a fire class, and a fire severity.

Complexity

The complexity of the fire clustering algorithm is $O(n^3)$ – polynomial. This can be shown by induction over the total number of fires and the worst-case number of comparisons.

5.3.3 The Damage Control Case-Based Planner

In case-based planning, previously generated plans are stored as cases in a knowledge base and can be reused to solve similar planning problems in the future, thus potentially saving considerable time over planning from scratch [Hammond, 1986]. Case-based planning can be a very useful paradigm for the DC domain because casualty situations often resemble one another in many ways, and response actions are fairly similar across crises.

One drawback of case-based planning is the need for a highly structured knowledge base organization, which would permit straightforward matching of target goals with the goals of existing plans. We solve this problem by utilizing a two-tiered approach to target goal generation. In the first stage high-level constraints are determined by the System State Inference Engine. These are then used to design the damage control goals, which are presented to the knowledge base of cases.

In the event that a suitable existing plan cannot be found, the Action-level Planner is invoked.

5.3.4 The Action-level Planner

The Action-level Planner utilizes a knowledge base of DC actions, which corresponds to specific low-level goals. This knowledge base is manually constructed from a compilation of damage control information obtained from experts and Naval DC documents. A detailed listing of the goal-action rules together with implementation-level information may be found in Section 7.

5.3.5 The Resource Manager

The resource manager is in essence composed of four sub-modules, the Damage Assessment sub-module, the Repair Assessment sub-module, the Resource Use sub-module and the Resource Availability sub-module. Figure 4 illustrates data and control flow between the two modules, as well as their interaction with other modules in the DC-SCS system.

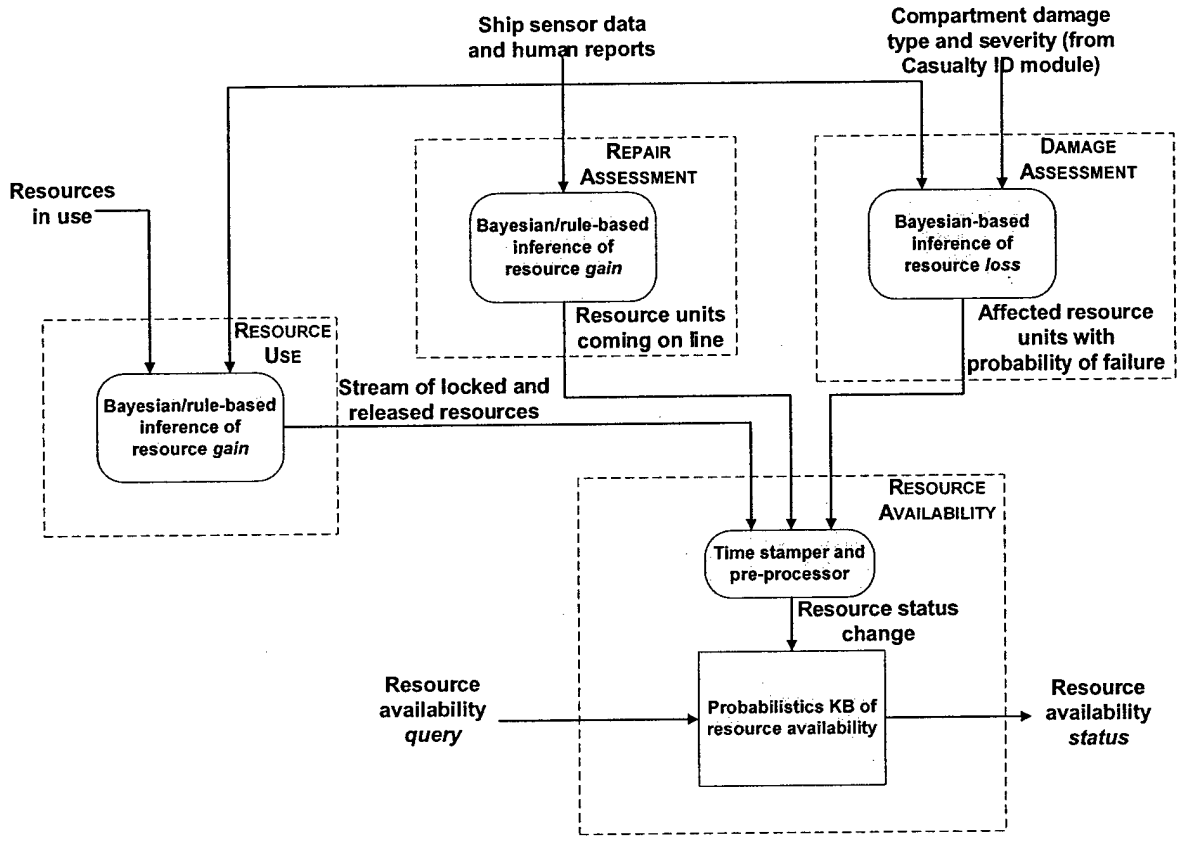


Figure 4. Resource Manager control and data flow diagram.

The Damage Assessment sub-module evaluates resource loss due to casualty. It takes as input the type and severity of the casualty in each compartment affected by the crisis, as well as ship systems and personnel known to be lost from sensor data and human reports. It outputs an estimate of lost resources in the form of a stream of resource units with corresponding probabilities of loss.

The Repair Assessment sub-module (Recovery) evaluates resources gained from either repair of damaged systems or the direct addition of resources into the system (such as the activation of a fire pump). It takes as input ship sensor data and human reports. It outputs an estimate of gained resources in the form of a stream of resource units with corresponding probabilities of the resource being on line.

The Resource Use sub-module determines which resources are in use due to the execution of a plan and which have been released. It takes as input an external stream of resources in use, as well as ship sensor data and human reports (to determine when resources have been released from use). It outputs a stream of resource unit subtractions and additions.

The Resource Availability sub-module takes as input the stream of lost resource units from the Damage Assessment sub-module, the stream of gained resources from the Repair Assessment sub-module, and the stream of resource use information from the Resource Use sub-module. This information is used to update the Resource Knowledge base as to the availability of

resources. The Planner queries this knowledge base whenever it needs to know if a resource is available.

5.3.5.1 Damage Assessment

In order to make judgments about the availability of damage control resources, it is necessary to determine which resources have been lost due to primary and secondary damage. The basic approach is to use casualty type and severity information from the Casualty ID module, combined with systems sensor data and personnel reports to estimate damage to DC resources.

Because it is difficult to obtain crew casualty information, and to estimate from this the reduction in operational effectiveness of the DC teams, it will be assumed at this stage that DC teams do not sustain any damage and remain fully operational throughout the crisis.

The resource damage assessment is made by reasoning over a Bayesian deactivation diagram, which causally relates all ship compartments to the DC systems affected by primary or secondary casualty in these compartments. Information about casualty type and severity is input to the root nodes (compartments), while specific system failure data from sensors and/or reports are input to relevant resource nodes in the network. Evidence propagation is then performed in the network to determine which resources have likely been lost and with what probability.

Damage Control Resources

- Water mist system
  - water mist installation
  - pressure
- Water for manual spraying
  - firemain pressure
- Halon replacement
  - installation
- Investigator team
  - personnel
- Repair party
  - personnel

Inputs

- Compartment state for each compartment (*with probability of state*):
  - No Fire 0
  - Nuisance Event 1
  - Mild Fire 2
  - Intermediate Fire 3
  - Severe Fire 4

- Blast overpressure damage by compartment with severity and probability
  - No damage 0
  - Mild Damage 1
  - Severe Damage 2
- Fire main low pressure information by section (These data can be obtained from smart valve algorithm [Bradley, 2000].)
- Water mist system damage by section (These data can be obtained from water mist concept report [Mawhinney, 2000].)

Output

- Resource units lost with associated probabilities of loss

Implementation Notes

- As indicated, the compartment state and blast overpressure levels should be encoded by integers.
- The following internal knowledge sources are needed for this algorithm:
  - Human casualty information
  - Blast overpressure estimation from reports (and possibly some sensor data)
- The output should probably be in the form of a stream, which will be written to the resource availability knowledge base.
- The evidence propagation algorithm should be re-run each time any new evidence enters the deactivation network.

5.3.5.2 Repair Assessment

The Repair Assessment sub-module inputs *Shadwell* sensor data, as well as personnel reports, and infers from these, if and when resources come on line. This may occur either from the repair of damaged resources or from the addition of previously non-active resources.

5.3.5.3 Resources in Use

The Resources in Use sub-module keeps track of the resources being utilized at any given time. When a DC plan becomes active, the resources needed are locked and thus not available for use by another plan. When a plan finishes executing, the locked resources are released.

5.3.5.4 The Resource Availability Knowledge Base

The Resource Availability Knowledge base contains an entry for each resource unit available in the system. Information about the availability of resources is stored and dynamically modified by the Damage Assessment, Repair Assessment, and Resources in Use sub-modules. The knowledge base accepts queries about resource availability and outputs a probabilistic estimate.

6. Casualty Classification

Figure 5 illustrates the topology used for the Casualty Classifier belief network. Because fire growth and spread behaviors vary to some extent according to compartment size, three such networks have been separately trained on data from experimental fires run in compartments of corresponding sizes: *small*, *medium*, and *large*.

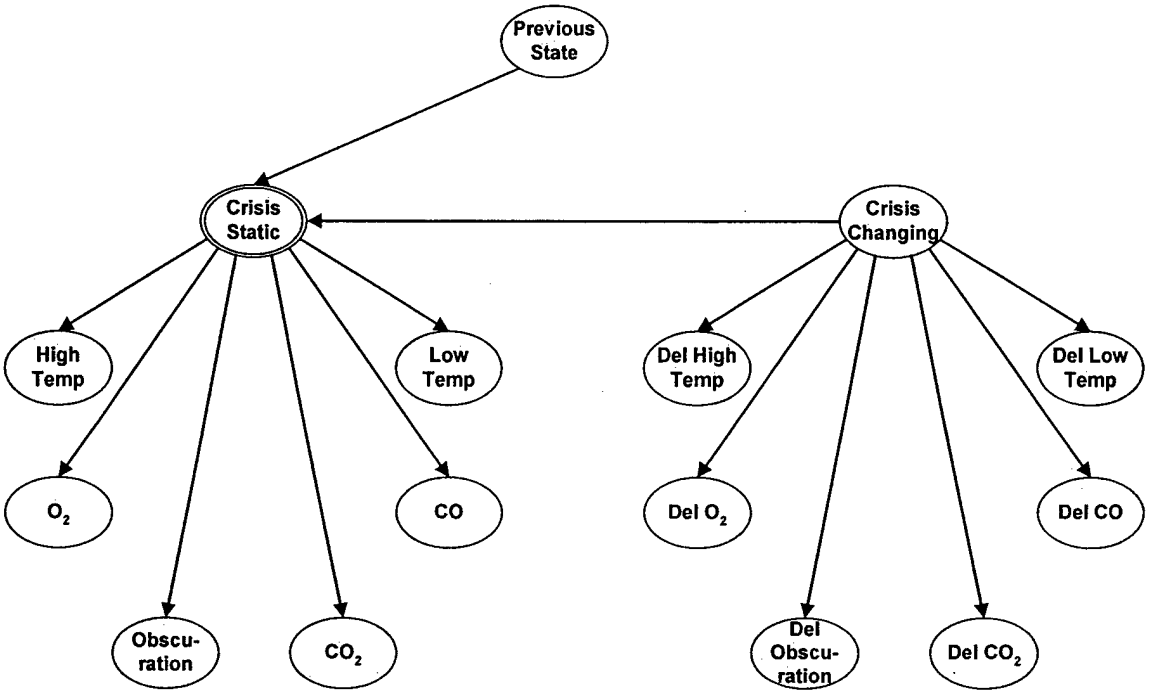


Figure 5. Topology of Bayesian belief network for Casualty Classifier.

This topology represents a qualitative model, relating casualty (in particular fire) to sensor readings. The sub-graph on the left, with vertex at the node labeled **Crisis Static**, infers the *current state* of the casualty in the particular compartment being evaluated. The right sub-graph, with vertex at the node labeled **Crisis Changing**, infers the *vector change* in the casualty for that compartment. The node at the top, labeled **Previous State**, simply adds some degree of memory to the model and helps make it robust to small, non-state-affecting changes in compartment conditions, such as gas concentrations and obscuration. The **Crisis Static** node outputs the state of the compartment at any given time.

Each compartment is processed separately by this network to determine its state. Correlation of evidence from different compartments is carried out later by another network.

7. Casualty Response

The Casualty Response module uses an object oriented model to describe situations on board the ship. Figure 6 shows a snapshot of the current class hierarchy. It contains only those classes that have been fully implemented in code. Documentation follows for descriptions of the current and future classes.

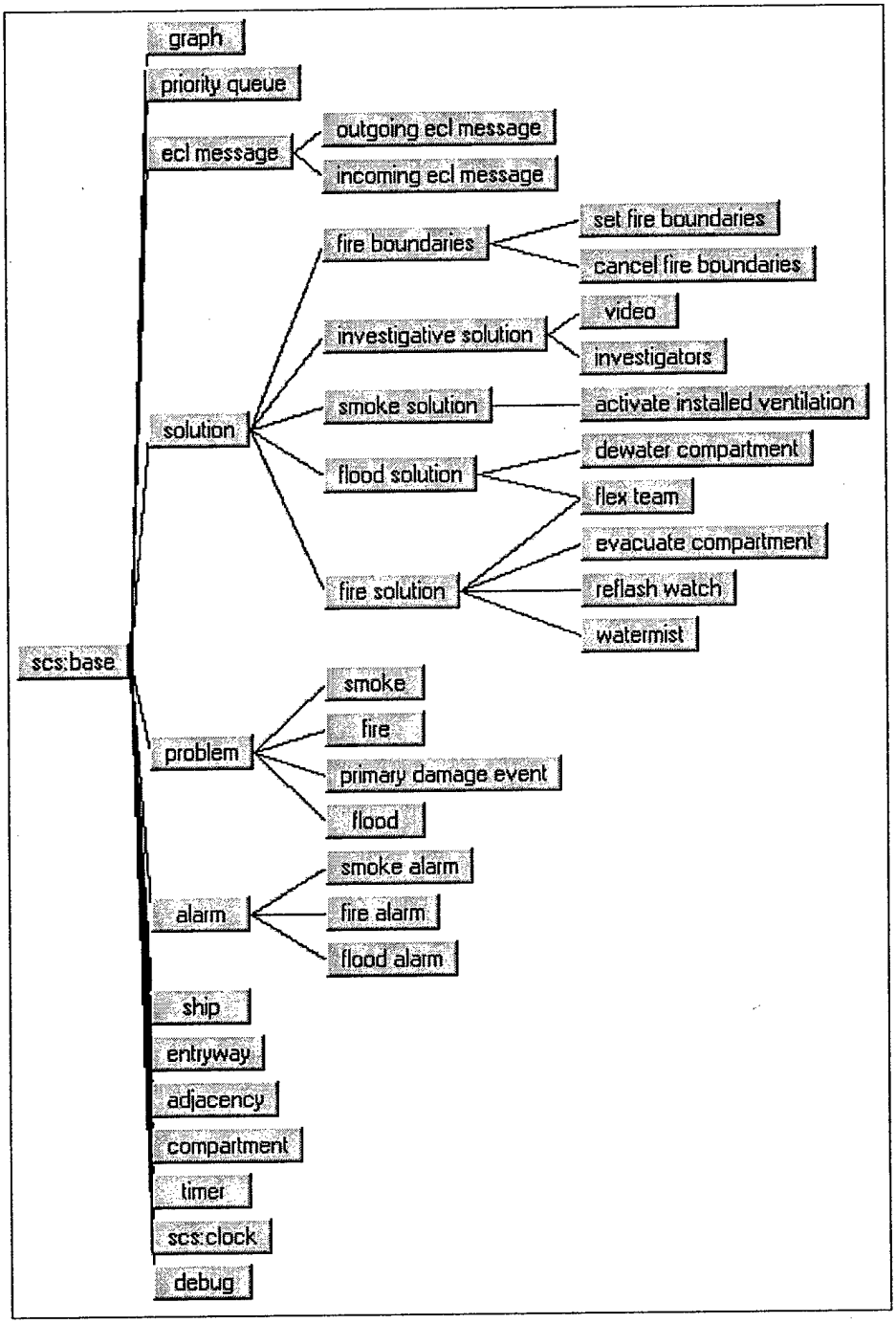


Figure 6. Class hierarchy

The following is a brief description of the object hierarchy presented in Table 6. The Table (although not presented in color), has color significance. Red items have been or will be deprecated. We are in the process of removing these items. Blue items are unimplemented features and fixes. Green items are those that have a conflict with another existing item or items. All but one of each set of green items will be removed. Each group of classes is described below and then a detailed description of each unique class is given.

The parent of all the classes in the system is the SCS:Base class. It contains no attributes or methods of its own but would allow easy addition of data or functionality to all the objects in the hierarchy. It inherits from the nil object (nothing).

SCS: Base, inherits *nil*

Attributes:
<none>

Methods:
<none>

The Debug class allows for easy debugging within the system. It contains functions, values, and database connections needed to properly print out and save debug messages. It derives from SCS:Base and has no children.

Debug, inherits SCS:Base

Attributes:
DB-Level – Threshold to print messages to the database
Display-Level – Threshold to print messages to the screen

Methods:
Out – Outputs a debugging message based on the threshold values
Set-db-level – Sets the value of DB-Level
Set-display-level – Sets the value of Display-level

The classes graph and priority queue are used to determine the distance between compartments. They both descend directly from SCS:Base and have no sub-classes. The priority queue object is a generic implementation of a priority queue using a heap. The graph class is an implementation of the graph data type, creating a graph of adjacency objects. The graph class has the ability to find the distances between any two adjacencies or compartments (using paths which contain doors or hatches). Here is a look at the two objects as they are currently implemented:

Graph, inherits SCS:Base

Attributes:
Adjacencies – a list of the adjacencies that we are graphing

Methods:
get-min-path – Gets the minimum path from one adjacency to another
get-min-path-compartment – Gets the minimum path from one compartment to another
initialize-compartment-source – Initialize one compartment so the start of a path is through the doors of the start compartment
initialize-single-source – Initialize a source, so the start of a path is through that source
min-path – Book algorithm to find the distance between two nodes in a graph

min-path-compartment – Variation on min-path method to find the distance between two compartments

Priority-Queue, inherits SCS:Base

Attributes:

Head-array – the array used to store the heap
Heap-size – size

Methods:

Extract-Min – Gets the minimum value from the queue
Heapify – Heapifies the heap used to represent the queue
Left – gets the left child of a node in the heap
Right – gets the right child of a node in the heap
Parent – gets the parent of a node in the heap
Insert – Add a node to the queue

Time in the system is represented with a clock object called SCS:Clock. SCS:Clock has a single rule that updates the clock whenever we get an ecl-message telling us the current time (we assume that time changes in discrete chunks). SCS:Clock and its closely associated sister object Timer are derived directly from SCS:Base and have no sub-classes. An SCS:Clock object can have timers set. These timers are represented by Timer objects each of which contain a function to call when it goes off. When a clock notices that a timer has expired, the clock notifies the appropriate Timer object that it has gone off. Both of these objects are defined along with the clock updating rule in "Cron.art"

SCS:Clock, inherits SCS:Base

Attributes:

Time – The current time contained in the clock. Each clock value represents 1 second
Timers – A list of timers that the clock has set

Methods:

Update – Updates the clocks time attribute with a new time
Get-Time – Returns the current value of the time attribute
Set-Timer – Creates a new Timer object and adds it to the clock
Add-Timer – Adds an already created Timer object to the list of timers associated with the clock
Kill-Timer – Removes a timer that was set on the clock

Timer, inherits SCS:Base

Attributes:

Clock – The clock the alarm is tied to
Time-To-Fire – Time that the timer will alarm
Action – The function to call when the timer alarms

Methods:

Activate – Sets the timer off, calling the function stored in action
Get-Time-To-Fire – Returns the Time-To-Fire value

Many physical objects on board the ship have object representations within our system. These objects are used to maintain the status of variables associated with those objects. Whether a door

is open or closed, whether the ship is at general quarters, the temperature in some compartment, and other such information are all stored this way. Currently, we have objects for the ship (Ship), hatches and doorways (Entryway), bulkheads (Adjacency), and compartments (Compartment). These four objects all inherit from the SCS:Base class. None of them currently have any sub-classes, but future plans include dividing doorways and hatches into separate sub-classes of Entryway. In the future we also plan to institute a similar structure for things such as personnel and systems. The Entryway, Adjacency, and Compartment objects are all instantiated during start-up from information in the database. One Entryway object is created for each entry in the doorways table and in the hatches table. One Adjacency is created for each object in the bulkheads table. In addition, if a bulkhead has more then one doorway/hatch, then it is split into two arbitrary Adjacency objects. Like Entryways, one Compartment object is instantiated for each entry in the compartments table in the database. Each object's definition is located in its own file with an appropriate name. A brief description of the above objects follows.

Ship, inherits SCS:Base

Attributes:

At-GQ – A Boolean saying whether the ship is at General Quarters or not
Highest-Deck – The upper most deck on the ship
Lowest-Deck – The lowest deck on the ship

Methods:

Call-GQ – Calls General Quarters
Cancel-GQ – Cancels General Quarters
Get-Above-Deck – Returns the deck above the given one, or the Highest-Deck, whichever is lower
Get-Below-Deck - Returns the deck below the given one, or the Lowest-Deck, whichever is higher
Get-Compartments-Between-Frames – Returns all the compartments between two given frames on any given deck on either port side, starboard side, or both sides

Friendly-Rules:

Send-GQ-Message – Creates the appropriate ECL message whenever GQ is called

Entryway, inherits SCS:Base

Attributes:

Adjacency – The wall that the door is in
Coordinates – The relative x,y,z coordinates of the entryway
Status – If it is open, closed or dogged

Methods:

Set-Coordinates – Sets the value of the Coordinates attribute
Get-Coordinates – Returns the value of the Coordinates attribute
Travel Time – Time to travel through the entryway. Large for hatches, small for open doors

Adjacency, inherits SCS:Base

Attributes:

Entryway – A pointer to the door/hatch in a bulkhead, if one exists

Travel Time – Time to travel through the adjacency. Large for hatches, small for open doors,
Neighbors – List of (adjacency, travel-time) pairs for each neighboring adjacency (in the graph)
Predecessor – Used by min-path algorithm to keep track of the path taken though the graph
Distance – Accumulator used by min-path algorithm to track distances in graph
Compartment1 – One of the two compartments connected by the adjacency
Compartment2 – The other compartment connected by the adjacency
Compartments – The two compartments connected by the adjacency

Methods:
Get-Travel-Time – Given two adjacencies it returns the time to travel between them
Has-Entryway – Returns true if Entryway is a valid object
Get-Entryway – Returns the Entryway object
Add-Entryway – Changes the Entryway reference to point to a new Entryway object
Add-Neighbor – Adds a pair to the Neighbor list

Compartment, inherits SCS:Base

Attributes:
KBS-Name – The name of the compartment that KBS uses in ECL grammar
Frame – Frame number of the compartment
Deck –Deck number that the compartment is on (+ for below deck)
Position – Position of compartment from center of ship
Type – The type of the compartment (the last letter in its official name)
Fire-Solutions – A list of solutions available to fight fires
Flood-Solutions – A list of solutions available to fight floods
Smoke-Solutions – A list of solutions available to fight smoke
Personnel – The personnel in the compartment
Alarms – Alarms going off in the compartment
Problems – Problems located in the compartment
Percent-No-Fire – Belief that there is no fire in the compartment
Percent-Nuisance-Fire – Belief that there is a nuisance fire in the compartment
Percent-Critical-Fire – Belief that there is a real fire in the compartment
Obscuration – The current obscuration level in the compartment
High-Zone-Temperature – High zone temperature reading in the compartment
Low-Zone-Temperature – Low zone temperature reading in the compartment
Obscuration-Window – The obscuration range in the compartment
High-Zone-Temperature-Window – High zone temperature window in the compartment
Low-Zone-Temperature-Window – Low zone temperature window in the compartment
Number-Of-Adjacencies – Number of adjacencies that the compartment has
Adjacencies – A list of adjacencies that the compartment has
Has-Camera – Whether a compartment has installed cameras

Methods:
Add-Adjacency – Add an adjacency object to the Adjacencies list

Get-Min-Adjacency – Finds the closest compartment from the door entered
Get-Adjacent-Compartments – Returns a list of all the adjacent compartments
Get-Number-of-Compartments – Returns the number of adjacencies
Get-KBS-Name – Returns the KBS name of the compartment
Add-Alarm – Inserts an alarm object into the compartment's Alarms sequence
Add-Problem – Inserts a problem object into the compartment's Problems sequence
Has-Fire – Returns true if one of the associated problems in the compartment is a fire
Has-Smoke – Returns true if one of the associated problems in the compartment is smoke
Has-Flood – Returns true if one of the associated problems in the compartment is a flood
Get-Percent-No-Fire – Returns appropriate belief value
Get-Percent-Nuisance-Fire – Returns appropriate belief value
Get-Percent-Critical-Fire – Returns appropriate belief value
Set-Confidence-Factor – Sets the three belief values for the compartment
Get-Fire-Problem – Returns the fire problem if one exists
Get-Flood-Problem – Returns the flood problem if one exists
Get-Smoke-Problem – Returns the smoke problem if one exists
Get-Frame – Returns the frame number
Get-Deck – Returns the deck number
Get-Position – Returns the position value

Friendly Functions:

Get-ID – Returns the ID of a compartment from a given KBS name, requires DB access
Get-Object-From-Name – Returns the name of the compartment object from a given KBS name, requires DB access

Personnel

Flex Team
Rapid-Response Team
First-Responder/Investigators

Systems

Sprinklers
CO_2 Extinguishers
AFFF Extinguisher
PKP Extinguishers
AFFF Flooding System
Hose
Watermist System
CO_2 Flooding System
HFP Flooding System
Pumps
Installed ventilation
Portable ventilation
COTS
EWFD
MassComp
Video Cameras

The next type of objects to discuss are the ECL-Messages. The super-classes of this branch of the heierarchy is ECL-Message. This class derives from SCS:Base and represents a row in the ECL Messages table or in the Pending ECL Messages table. It has two direct subclasses that break those two apart. Outgoing ECL Message is a class that represesents the Pending ECL Message table's rows. Incoming ECL Messages are the ECL Message table's rows. Outgoing ECL Messages are destroyed once they have been written to the databases and Incoming ECL Messages are destroyed after all rules have tried matching on them at least once. The attributes of these objects are identical to the columns found in their respective tables so only their methods will be discussed below.

ECL-Message, inherits SCS:Base

Methods:
Full – Not documented

Outgoing-ECL-Message, inherits ECL-Message

Methods:
Write – Writes a given outgoing-ECL-message to the database

Friendly Rules:
Write-Outgoing-ECL-Messages – Calls write on every newly created outgoing ECL-message

Incoming-ECL-Message, inherits ECL-Message

Methods:
<None>

Alarms are one of the three major branches of the class hierarchy. Topped with the abstract Alarm class, the heirachy has one alarm class for each problem of fire, smoke, and flood. More alarm classes should be added to keep the current model symmetrical (one alarm for each problem type).

Alarm, inherits SCS:Base

Attributes:
Time-Triggered – Time the alarm was triggered
Compartment – Compartment that the alarm is reported for
From – Name of person/system/module that triggered the alarm
Adjective – Information about the alarm
Size – Size of the alarm small/medium/large

Methods:
Get-Problem – Returns the fire problem associated with an alarm

Fire-Alarm;Flood-Alarm;Smoke-Alarm;Primary-Damage-Event-Alarm, inherits Alarm

Attributes:
<none>

Methods:
<none>

The next major branch in the object model is the problem hierarchy. From the generic class Problem stem one class for each of four types of problems. Fire, Flood, Smoke, Primary-Damage-Event. Due to an oversight in the original design document currently the problem classes can sometimes deal with being in multiple compartments and some times not. One fix would be to create two hierarchies, one for multi-rooms and one for single-compartments. Another fix would be to simply allow allow all problems to occupy more than one compartment. The choice of implementation has not been finalized.

Problem, inherits SCS:Base

Attributes:

Completed – Whether the problem has been completed
Time-Identified – Time the problem was created at
Compartment – Compartment that the problem is in
Compartments – List of compartments the problem is in
Used-Solutions – List of solutions used on the problem
Solutions – List of solutions that can be used on the problem
Status – Status of problem, overrides the Completed attribute
Ranker – Pointer to the ranker object doing the ranking for this problem

Methods:

Add-Used-Solution – Adds a solution to the Used-Solutions list
Get-Compartment – Returns the value of the Compartment attribute
Get-Compartments – Returns the values of the Compartments attribute

Fire, inherits Problem

Attributes:

Class – The class of the fire (A,B,C,D)

Methods:

<none>

Flood; Smoke, inherits Problem

Attributes:

<none>

Methods:

<none>

Primary-Damage-Event, inherits Problem

Attributes:

Forward-Frame – Most forward frame to be hit by the event
Aft-Frame – Most aft frame to be hit by the event
Deck –Deck the event will be on
Side – Side the event will occur on

Methods:

Add-Used-Solution – Adds a solution to the Used-Solutions list
Get-Compartments – Returns the compartments that will be affected by the event
Get-Forward-Frame – Accessor

Get-Aft-Frame – Accessor
Get-Deck – Accessor
Get-Side – Accessor

The solution classes currently have a very simple hierarchy. This will be drastically changed as the ranker becomes in use. In order for the rankers to correctly order potential solutions, the solutions will be reorganized into a complicated web of solution types. Currently, Fire, Flood, and Smoke-Solution classes are directly below the Solution class. Also at this level are the investigative-Solution class and the Fire-Boundaries class. The Fire-Boundaries class will be replaced with a more gernic Boundary-Solution Class with several appropriate sub-classes. In addition there will be branches for handheld-solutions automated-solutions, manned-solutions, water-solutions. Solutions are turned into Outgoing ECL Messages when they are proposed. The solution class has a unique way of defining methods to do this. Each solution class has methods named get-ecl-<column>-field. Where there is one method for each of the columns that that particular ecl message uses. These special methods are *not* defined below.

Solution, inherits SCS:Base

Attributes:
Time-Proposed – Time the solution was proposed
Time-Activated – Time the solution was activated
Time-Completed – Time the solution was completed
E-T-Completed – Estimated time of completion
Activation-Status – Status of solution (pending, ranking, working and completed)
Compartment – The compartment the solution is being used in
Associated-Problem – The problem the solution is being used on
ECL-Num – The ECL number of the solution from ECL grammar
ECL-Arguments – A list of arguments that the solution needs, from ECL grammar

Methods:
Get-Compartment – Returns the Compartment attribute
Get-Problem – Returns the Associated-Problem

Fire-Solution; Flood-Solution; Smoke-Solution, inherits solution

Automated-Solution; Manned-Solution, inherits solution

Evacuate-Compartment, inherits fire-solution, solution

Investigative-Solution, inherits Fire-Solution, Flood-Solution, Smoke-Solution, Solution

Attributes:
Predicted-Status – Status being investigated
Observed-Status – Status observed in the compartment

Methods:
<none>

Send-Rapid-Response-Team, inherits investigative-solution, manned-solution

Investigate-With-Video, inherits investigative-solution, automated-solution

Send-Flex-Team, inherits fire-solution, flood-solution, smoke-solution, manned-solution

Boundary-Solution, inherits fire-solution, flood-solution, smoke-solution

Set-Fire-Boundaries, inherits boundary-solution, fire-solution

Cancel-Fire-Boundaries, inherits boundary-solution, fire-solution

Set-Reflash-Watch, inherits fire-solution

Dewater-Compartment, inherits flood-solution

Activate-Installed-Ventilation, inherits smoke-solution, automated-solution

Activate-Watermist, inherits automated-solution, fire-solution

Activate-AFFF-System, inherits automated-solution, fire-solution

Activate-CO2-System, inherits automated-solution, fire-solution

The final branch of the class hierarchy includes the ranker classes. These classes have not been implemented as of yet but may have a structure as given below. In the future, a Deactivator of the ventilation is needed.

Ranker

Attributes:
Solutions – A sequence of problem, solution, rank triplets

Methods:
Rank – Ranks the solutions

Local-Ranker, inherits Ranker

Attributes:
Owner – The problem that the ranker is working on

Methods:
<none>

Master-Ranker, inherits Ranker

Attributes:
Locals – A list of all the local ranker objects

Methods:
Add-Local-Ranker – Adds a local ranker to the list
Out – Outputs to the GUI
In – Gets input from the GUI

8. Communication To/From the DC-SCS and DCA

This section lists all the English sentences that can be issued to or from the DCA and the DC-SCS. The Human-Computer Interface and Visualization currently allows communication using these sentences to/from the DCA. With time, more and more of these same sentences can be communicated to/from the automated DC-SCS supervisory control system.

8.1 Commands from the DCA and DC-SCS

This section lists commands and outputs from the DCA and the DC-SCS to ship personnel and automated ship systems.

Readiness menu
- Order GQ
- Report condition set [to CO]
- Order condition set [to scene leaders]
- Request permission to set condition [to CO]
- Query status of condition [to System | team leader]

Personnel menu
- Report
- All stations manned and ready [to CO]
- Request manning of compartment X [to scene leader]
- Request manning of repair locker X [to scene leader]
- Request manning [to RRT X]

Investigate menu
- Order Investigation of Compartment X, optionally for suspected status Y [to First Response team]
- Request report of [change in/all info on] [gases/particular gases] in compartment X [to LiveData device]
- Request report of temperature in compartment X [to LiveData device]
- Request report of COTS output for compartment X

Fire menu
- Request [to CO]
  - Permission to activate magazine sprinklers in compartment X
- Order [to team leader]
  - Fire boundaries with settings s-aft: A p-aft: B p-fore: C s-fore: D above: E below: F type: X
  - Fire boundaries around compartments X
  - Fire boundaries around set of compartments (X1, X2, …, Xn)
  - Smoke boundaries with settings s-aft: A p-aft: B p-fore: C s-fore: D above: E below: F type: X
  - Smoke boundaries around compartments X

Smoke boundaries around set of compartments (X1, X2, …, Xn)
Electrical isolation in compartment X
Mechanical isolation in compartment X
Firefighting team to designated location in compartment X
Overhaul compartment in compartment X
Desmoke using [to team leader]
Installed ventilation in compartment X
Portable ventilation in compartment X
Overpressure in compartment X
Activate [to team leader]
AFFF sprinklers in compartment X
Primary hfp in compartment X
Reserve hfp in compartment X
Magazine sprinklers in compartment X
Water mist in compartment X
Secure [to team leader]
AFFF sprinklers in compartment X
Magazine sprinklers in compartment X

Flood menu [to team leader]
Order
Flood boundaries with settings s-aft:A p-aft: B p-fore: C s-fore: D above: E below: F
Flood boundaries around compartments X
Flood boundaries around set of compartments (X1, X2, …, Xn)
Flooding teams to a designated area in compartment X
Patch
Firemain pipe in compartment X
Plug
Rupture in compartment X
Split Seam/Crack in compartment X
Dewater
Using installed eductors in compartment X
Using portable equipment in compartment X
Open/Close
Firemain valve X

Ventilation menu [to team leader]
Order
Negative ventilation in affected compartment X
Positive ventilation in unaffected compartment X
Normal ventilation in compartment X
Emergency exhaust ventilation in compartment X
CPS zone pressure in compartment X
CPS zone de-pressurized and Set Normal Ventilation in compartment X
Request Permission To
Desmoke main space using installed ventilation in compartment X

Desmoke using
Installed ventilation in compartment X
Portable ventilation in compartment X
Overpressure in compartment X
[Open/close] ventilation ducts in compartment X

Structural menu [to team leader]
Order
Overhaul compartment X
Patch
Firemain pipe in compartment X
AFFF pipe in compartment X
Chilled water pipe in compartment X
LP Air pipe in compartment X
Plug
Rupture in compartment X
Split Seam/Crack in compartment X

Stability menu
Request Permission to [to CO]
Counterflood compartment X
Order [to team leader]
Counterflood compartment X

Electrical menu
Request Permission to [to CO]
Rig casualty power in compartment X
Energize casualty power in compartment X
Order [to team leader]
Electrical isolation of compartment X
Rig casualty power in compartment X
Energize casualty power in compartment X

Access menu
Order
Access of compartment X
Access to compartment X using forcible entry tools
Evacuate
DC Central
Compartment X due to [fire/smoke/flood/toxic gas/hazmat]
Recommend
Abandon Ship
Send Y members of RRL #N to compartment X

Magazines menu
- Request Permission To
  - Activate magazine sprinklers in compartment X
- Activate
  - Magazine sprinklers in compartment X [to team leader T]
- Secure
  - Magazine sprinklers in compartment X [to team leader T]

Systems menu
- Request permission to
  - Start Fire pump X
  - Activate Magazine Sprinklers in compartment X
- Start/Stop Fire pump X
  - Open/Close
  - Firemain valve X
  - AFFF valve X
  - Chilled Water valve X
- Request report of door status in compartment X
- Request sensor presence in compartment X

8.2 Commands to the DCA and DC-SCS

This section lists commands and outputs to the DCA and the DC-SCS from sensors, ship personnel and automated ship systems. The Blue-shaded commands are recommendations and requests from the automated DCA to the human DCA, and as such may not be permanent.

Acknowledgement of any order
Grant or deny any request for permission
Inability to complete any order
Completion of any order

For the orders that require specific responses beyond a simple report of completion, we will use these menus:

Readiness menu
Report status of condition X

Personnel menu
Report station X manned and ready
- Report manning of compartment X
- Report manning of repair locker X
- Report manning of RRT X

Investigate menu

Report results of investigation of Compartment X

[compartment unreachable/no smoke/sparse smoke/dense smoke/no fire/small fire/medium fire/large fire/no water/light water/heavy water]

Report results of communication contact to compartment X

[no connection/no answer/no smoke/sparse smoke/dense smoke/no fire/small fire/medium fire/large fire/no water/light water/heavy water]

Report results of [change in/all info on] [gases/particular gases] in compartment X

Report temperature in compartment X

Report COTS output for compartment X

Report of pressure P in a firemain

The video camera indicates [no/sparse/dense] smoke in compartment X

The video camera indicates [no/small/medium/large] fire in compartment X.

The video camera indicates [no/light/heavy] water in compartment X.

The thermal couplings quantitatively indicate temperature T.

Alarms menu

Alarm A in compartment X

Alarm A on system S

Rupture in bulkhead B in compartment X

Loss of electrical power to system S

Reports of rising water level in a compartment C at rate R

Sounding of GQ

Sounding of abandon ship

Temperature in compartment X is [normal/warm/abnormally high/quantitative].

Smoke level in compartment X is [none/sparse/dense].

Report of [change in/all info on] [gases, particular gases] in a compartment

System menu

Status of pump P: online/offline/broken/unknown

Status of valve V: open/closed/destroyed/unknown

Status of other system S: on/off or open/closed or unknown/destroyed

Status of electrical power to a system S, where status: connected/not-connected

Status of door D given by a report or due to ship status, where Status: open/closed/dogged

Status of ventilation system S where Status: No fan/fan low/fan on/fan damaged

Reasoning menu

BN concludes [nuisance fire/real fire/no fire] in compartment X with [low/medium/high] probability.

Incoming missile heading toward [e.g., starboard/midsection] will hit ship in one minute.

Miscellaneous menu

Result of automated or reflexive system: effective/ineffective

Ship is on [peacetime/wartime] footing.

Compartment X has [COTS/thermal couple/video camera]

Compartment contents: any-systems, fuels, metals
The rate of watermist use is X lbs/min per watermist nozzle.
There are Y watermist nozzles in compartment X.
Flow rate in firemain section F is quantity Q.
Report
CPS (Collective Protection System) zone Z low-pressure alarm
CPS zone Z failure

Medical menu
Report injury of personnel
Casualty reports in a compartment X

8.3 Future Commands from the DCA and DC-SCS

This section lists commands and outputs from the DCA and the DC-SCS to ship personnel and automated ship systems. These commands are currently not part of the Illinois DC-SCS system, but may be added at a future time. The Blue-shaded commands are recommendations and requests from the automated DCA to the human DCA, and as such may not be permanent.

Personnel menu
Report
Repair stations manned and ready except X
Provide additional personnel to station X

Fire menu
Request
Dewater fire-fighting water in compartment X
Order
Smoke Curtains
OBA changeout area (SCBA charging)
OBA reliefs
Gas free testing
Deny permission to
Activate magazine sprinklers

Flood menu
Patch
Open/Close
AFFF valve
Chilled water valve
Fuel valve
JP-5 valve
Structural menu
Order
Sprung fitting secured

- Shore
  - Bulkhead
  - Deck/Overhead
  - Door
  - Hatch

Stability menu
- Request Permission To
  - Jettison Topside Weight
  - Ballast
  - Deballast
- Order
  - Jettison of Topside Weight
  - Ballasting
  - Deballasting
- Report
  - Free Communication of Water present
  - Stability Not considered critical
  - Stability Critical Due to...
  - Exceeding Floodable Length
  - Excessive List
  - Heavy Winds and Sea
  - Negative Metacentric height
- Deny Permission To
  - Counterflood
  - Jettison Topside Weight
  - Ballast
  - Deballast

Electrical menu
- Order
  - Electrical damage repaired
  - Electrical power restored
- Deny permission to
  - Rig Casualty Power
  - Energize Casualty Power

Hazmat menu
- Request permission to
  - Jettison hazardous material
  - Jettison compressed gas cylinder
  - Jettison flammable liquid containers
  - Pump flammable liquid containers
- Contain
  - Hazardous Material
  - Flammable Liquid Spill

Toxic Gas
Remove
Hazardous Material
Compressed Gas Cylinder
Flammable Liquid Containers
Flammable Liquid Spill
Toxic Gas
Jettison
Hazardous Material
Compressed Gas Cylinder
Flammable Liquid Containers
Pump
Flammable Liquid Overboard

Magazines menu
Request Permission To
Move Ammunition
Jettison Ammunition
Order
Ammunition cool down
Movement of Ammunition
Jettison Ammunition
Deny Permission To
Activate Magazine Sprinklers
Move Ammunition
Jettison Ammunition

Chemical menu
Order
Chemical MOPP N Checklist
Report
CPS Zone Low Pressure Alarm
CPS Zone Failure
Deny Permission To
Decon designated equipment using standard decon solution
Decon designated equipment using USMC decon solution

Biological menu
Recommend or Order Biological MOPP N Checklist
Report
CPS Zone Low Pressure Alarm
CPS Zone Failure
Deny Permission To
Decon designated equipment using standard decon solution
Decon designated equipment using USMC decon solution

Radiological menu
- Recommend or Order
  - Radiological MOPP N Checklist
- Report
  - CPS Zone Low Pressure Alarm
  - CPS Zone Failure
- Deny Permission To
  - Decon designated equipment using standard decon solution
  - Decon designated equipment using USMC decon solution

Systems menu
- Request permission to
  - Activate CMWD in compartment X
  - Secure CMWD in compartment X
- Activate
  - CMWD Intermittently in compartment X
  - CMWD Continuously in compartment X
- Secure
  - CMWD in compartment X
- Open/Close
  - Fuel valve
  - JP-5 valve
  - LP air valve
  - MP air valve
  - HP air valve

Maneuvering menu
- Request
  - Downwind hazard prediction information
- Recommend
  - OOD maneuver to keep smoke clear of ventilation systems
  - OOD maneuver to improve stability
  - OOD maneuver to avoid contaminated areas

Medical menu
- Order
  - Buddy aid for nerve agent exposure

Transport
- Injured personnel to designated area
- Chemical casualty to designated area
- Heat stress casualty to designated area

8.4 Parameter Values Associated with Commands

For each of the listed parameters in the active command list above, these are the legal parameters.

There are subsets for different commands.

	Value	Context
To	Dca	DCA
To	repair 2	Repair 2
To	repair 3	Repair 3
To	repair 5	Repair 5
To	aft battledressing station	Aft Battledressing Station
To	Forward battledressing station	Forward Battledressing Station
To	Co	CO
To	Bridge	Bridge
To	Dcco	DCCO
To	Csmc	CSMC
To	Eoow	EOOW
To	engineer	Engineer
To	Cic	CIC
To	entire ship	Entire Ship
From	Dca	DCA
From	repair 2	Repair 2
From	repair 3	Repair 3
From	repair 5	Repair 5
From	aft battledressing station	Aft Battledressing Station
From	forward battledressing station	Forward Battledressing Station
From	co	CO
From	bridge	Bridge
From	dcco	DCCO
From	csmc	CSMC
From	eoow	EOOW
From	engineer	Engineer
From	cic	CIC
Problem	fire	
Problem	flooding	
Problem	smoke	

	Value	Context
Problem	black smoke	
Problem	fire and smoke	
Problem	rupture	
Problem	leak	
Problem	failure	
System	firemain	
System	chillwater	
System	electrical	
System	ac	
System	halon	
System	sprinkler	
Status	online	pump
Status	offline	pump
Status	standby	any
Status	lost	Any
Status	activated	halon, sprinkler, etc.
Status	deactivated	halon, sprinkler, etc.
Status	battleshort	electrical
Status	failed	any
Status	overheating	any
Status	manned and ready in progress	MR/Z
Status	manned and ready and zebra in progress	MR/Z
Status	zebra set	MR/Z
Status	growing hotter	compartment
Status	hot	compartment
Status	intact	compartment
Status	ignited	compartment
Status	engulfed	compartment
Status	extinguished	compartment
Status	destroyed	compartment
Status	flooded	compartment
Status	normal	pressure (NEVER compartment)
Status	low	pressure
Status	high	pressure
Status	mechanical	isolation type
Status	electrical	isolation type
Status	electrical and mechanical	isolation type

	Value	Context
Status	effective	halon, sprinkler, etc.
Status	ineffective	halon, sprinkler, etc.
Alarm	high temperature	
Alarm	low pressure	
Alarm	high pressure	
Alarm	halon	
Alarm	flooding	
Status	open	valve
Status	closed	valve
Alarm	smoke	
Adjective	starboard	
Adjective	port	
Adjective	forward	
Adjective	aft	
Adjective	primary	
Adjective	secondary	
Adjective	nothing	(when the adjective is not needed)
Status	smoky	compartment
Status	inaccessible	compartment
String1	it sank	Reason for loss of ship
String1	it capsized	Reason for loss of ship
String1	a magazine exploded	Reason for loss of ship
String1	both radar arrays were lost	Reason for loss of ship
String1	a compartment was flooded without permission	Reason for loss of ship
System	primary halon	
System	reserve halon	
System	cps	
System	countermeasure washdown	
System	afff	
System	normal ventilation	
System	emergency ventilation	
String1	it was abandoned	Reason for loss of ship
String1	chemical	MOPP category
String1	biological	MOPP category
String2	radiological	MOPP category
To	repair 8	Repair 8
From	repair 8	Repair 8

This section lists all the sentences that can be issued to or from the DCA and the DC-SCS. The Human-Computer Interface and Visualization allows communication using these sentences to/from the DCA to be accomplished today. With time, more and more of these same sentences can be communicated to/from the DC-SCS system.

8.5 Commands from the DCA and DC-SCS

This section provides an explanation of some of the commands that were given in Section 8.2. It is a preliminary version that reflects the Navy's damage control doctrine as it is practiced today, and expounded in *Naval Ships' Technical Manual, Chapter 555*. Future revisions of this document will contain a command structure more in line with the goals of the DC-ARM program.

Readiness menu

Order GQ – General Quarters is called in response to a report of an incoming missile or major casualty. General Quarters is called in peacetime during a crisis when the crisis has not been controlled in a reasonable amount of time. Condition Zebra is set in conjunction with General Quarters. Condition Zebra is a state of readiness in which the ship is secured for an impending missile hit in a wartime scenario.

Report condition set [to CO] – This is an output of the DCA. The DCA reports the condition of readiness that has been set by the Commanding Officer.

Order condition set [to scene leaders] – The DCA orders the Scene Leaders, Repair Party Leaders to set condition X, where X is Zebra (w/General Quarters), Xray and Yoke. Condition Xray provides the least degree of water-tightness but the greatest ease of access throughout the ship. Condition Xray is set when there is little or no threat of attack or weather or when the ship is in port in peacetime. Condition Yoke provides a greater degree of water-tightness. Yoke is normally set at sea or in port during wartime. Condition Zebra provides the greatest degree of water-tightness. This state is set at general quarters or when the ship is entering or leaving port during wartime.

Request permission to set condition [to CO] – If a recommendation to set a state of readiness is issued, the DCA requests permission to set that state of readiness from the Commanding Officer.

Query status of condition [to System | team leader] – The DCA can ask the system, Team Leaders, or Repair Locker Leaders what the current state of readiness is (Xray, Yoke, Zebra, including General Quarters). The readiness condition is returned to the DCA either by voice communications with the repair party leaders, or by text output of the system. Ideally, the text output of the system would go into a Minerva output window separate from the three plans GUI.

Personnel menu

Report

All stations manned and ready [to CO, DC-SCS] – This can be a delayed output of the system (ie, waiting a designated amount of time after GQ is called), but this should be a voice output of a scene/repair locker leader, which is manually entered into the system by

personnel in DC Central. This report should come in as soon as possible after MRZ report returns from a station.

Request manning of compartment X [to scene leader] – The DCA can request the manning of a crisis afflicted compartment from the scene leader. This information is especially important if the information is inconsistent with the SCS suggested personnel deployment. The Illinois system retains this information for advising related to personnel management. This information should be displayed in a Minerva output window.

Request manning of repair locker X [to scene leader] – This is again important for personnel management. The DCA is able to query the system for the number of personnel currently available/amount of people dispatched from a given repair locker. After being informed of the personnel availability in the repair locker, the DCA can reassign personnel to crises, overriding Minerva's recommendation (this would be an input to Minerva). This information should be displayed in a Minerva output window.

Request manning [to RRT X scene leader] – A request for manning can also be sent directly to a repair party. This is information

Investigate menu

Order Investigation of Compartment X, optionally for suspected status Y [to First Response team] – Minerva recommends that the DCA investigate compartment X. The results of this investigation are an input to the system. The options should be nuisance fire, small fire and large fire. Fire investigation with personnel should only be recommended if the compartment has no video cameras or the video cameras in the compartment are not operational. Idea: when you drag an investigation from pending to active, the interface should automatically switch to the input box for the results of the investigation (otherwise dragging the action from pending to active will do nothing).

Use video camera to check out nuisance alarm in compartment X – Minerva recommends using installed video cameras. The installed video report can be input into the system through the interface in the same way that investigator reports are input. Fire severity reports are the same identical to investigator reports (nuisance, small, large). See above command for suggestions on how this can be improved.

Use comm. system to contact compartment X – This doesn't seem necessary. Contact compartment X for what?

Request report of [change in/all info on] [gases/particular gases] compartment X [to LiveData device] – Since the system is not currently connected to the gas analyzer, Minerva can request information from the DCA in the three pane regarding the presence of gases in a given compartment. When the DCA drags the report request to active, the input window should change to the gas analysis window, allowing the DCA to input the results of the analysis. These limited real time data are available from the MassComp.

Request report of temperature in compartment X [to LiveData device] – This might not be necessary as an output of the DCA/DC-SCS, since compartment temperature should automatically be acquired through masscomp. Compartment status will be displayed in a viz window in the future and the compartment temperature will be available in this window.

Request report of COTS output for compartment X – Since current COTS (Simplex) will not currently send live data to DC-SCS, Minerva can recommend the investigation of a COTS sensor. When the recommended investigation is dragged from pending to active in the three pane GUI, the form for giving the results of the investigation should automatically appear in the interface. The output of a COTS sensor is merely fire or no fire, and they may not be reliable for classifying nuisance fires.

Fire menu

Permission to activate magazine sprinklers in compartment X [to CO] – The magazine sprinkler should only be used in a very volatile situation (when fire is threatening the magazine). Activating the magazine sprinklers should only be used as a last resort, and therefore requires the permission of the CO. When the DCA drags this command from pending to active, he should be prompted with a yes/no response. If permission is not given to activate the sprinklers, then sprinkler should never be activated (sprinkler activation would not be recommended).

Fire boundaries with settings s-aft A p-aft: B p-fore: C s-fore: D above: E below: F type: X [to team leader]– A fire zone boundary is a bulkhead or deck designed to limit the passage of flame and smoke. Fire zone boundaries confine a fire within a zone and provide protected staging areas for fire parties. When the crisis is classified as capable of spreading (non-nuisance fire), fire boundaries should be set to prevent the spread of the fire beyond the PDA. Fire boundaries are set as needed. All of the bulkheads of the recommended fire boundaries need not be set if the members of the repair party do not deem it necessary. Horizontal fire boundaries (the ceiling of the affected compartment) should always be seen as a priority and set for any non-nuisance fire since fire spreads upward more quickly than any other direction.

Smoke boundaries with settings s-aft A p-aft: B p-fore: C s-fore: D above: E below: F type: X [to team leader] – Smoke boundaries are important in preventing the spread of smoke into compartments and passages adjacent smoke obscured compartment. Smoke boundaries can be set by closing doors and hatches in the affected area, but setting a boundary curtain is generally a better alternative. DC-SCS should recommend smoke boundaries based on information from obscuration sensors and/or visual reports from cameras or personnel. DC-SCS smoke boundaries are merely recommendations, and the decision about their location depends on whether personnel in the test area and the DCA determine that they are necessary.

Electrical isolation in compartment X [EOOW, flex team leader] – Although it is not clear that this will be necessary for DC-ARM test purposes, electrical isolation is generally necessary before using water based fire fighting agents in quantity to prevent potentially dangerous situations for DC personnel.

Mechanical isolation in compartment X [EOOW, flex team leader] – It is also not clear whether mechanical isolation of a compartment is necessary. Mechanical isolation is generally used to prevent the spread of smoke through ventilation systems.

Firefighting team to designated location in compartment X – This would seem obvious, although it would seem like a good idea to have thermocouple information in the compartment status area to give a firefighting team or investigator advice in locating the fire.

Overhaul compartment in compartment X [flex team leader] – After a crisis affected compartment test area is secured, overhaul of the damaged compartment should be recommended.

Desmoke using [to team leader] – Desmoking will be recommended when compartment obscuration reaches a certain level. Desmoking can be achieved by installed and portable ventilation and overpressure.

9. Alternative Approach to Casualty Response Using COGNET

Two casualty response implementations are proceeding in parallel, one rule based and the other blackboard based. The blackboard approach uses COGNET [Zachary, Le Mentec and Ryder 1996], [Zachary and Ryder 1997], to implement the casualty response module, and this approach will now be described. COGNET is an executable cognitive architecture developed by CHI Systems Inc. While being rooted in psychological theory of modeling human behavior, it is based on a state of the art Blackboard system [Englemore and Morgan, 1988], and represents an alternative to current expert systems.
Unlike expert systems that are mostly based on rules, COGNET relies on bigger chunks of procedural knowledge called Tasks. These Tasks interact indirectly with each other by responding to changes in the blackboard, and by modifying and posting new data in the blackboard. Each Task has a trigger condition and a priority formula that are sensitive to specific patterns in the blackboard. When the trigger condition of a Task is satisfied, it competes for attention with other Tasks. The Task with the highest priority gains the focus of attention and executes to completion or until its suspend itself or is interrupted by another Task.

Each Task is organized as a hierarchy of smaller chunks of procedural knowledge called Goals. One of the main differences over a conventional rule based system is that Tasks can typically be carried over long period of time. For example, a single Task can be assigned to monitor the progress of a flex team. It can generate some commands to send a team and then suspend itself until the team reports back. A time-out feature can also be used to query the team if it did not report after a certain amount of time. Implementing the same mechanism with a rule-based system would require several rules with some implicit synchronization to recreate the same sequence of action.

The blackboard content is organized as a set of panels and levels where each level represents a class of objects. These objects, or blackboard elements, have a set of attributes and links to other objects. The attributes are used to store numerical or symbolic values about the object. The links can be used to create a semantic network to express knowledge about the domain. For example we are using links to express adjacency of compartments where each compartment is an object in the blackboard. Compartments can also be linked to a flex team object to express that a team is currently in the compartment.

A COGNET program, or model, is connected to the external world through a set of specialized procedures called demons. Demons are actually called directly by the external world and can post new entries in the blackboard. These entries can in turn trigger new Tasks or un-suspend

existing ones. A model can act on its external environment by calling action functions that are defined in C++ for the application domain.

One of the strengths of COGNET lies in its ability to handle real-time applications. Beside its inherent connectivity to the outside world, it is designed to reason about time and with potential fast response time. It also has some provisions to implement adaptive reasoning where accuracy is traded-off for speed when the CPU resources available do not permit an optimal response. Unlike some Lisp-based expert system, COGNET is implemented in C++ and is equipped with a very efficient execution engine.

COGNET is commercially available under the name iGEN ^{TM} and is provided with an easy to use graphical authoring tool, targeted at domain experts rather than expert programmers.

The Blackboard system approach of COGNET opens the possibilities to integrate easily other technologies that have already been developed in the KBS group such as Bayesian networks or some existing expertise developed with rule based systems.

COGNET has been used successfully around the world over the past 6 years for a variety of applications. Among the most noticeable was Advanced Embedded Training System (AETS) [Zachary, Ryder and Hicinbothom 1998] for the Navy. More recently COGNET has extended to model human performance accurately in a simulated environment [Le Mentec, Zachary, Glenn and Eilbert 1999].

10. Conclusion

The design for the DC-SCS system presented in this report conforms to the goals of the DC-ARM project. It provides rapid and reliable situation awareness, and generates effective casualty response. A number of design elements remain to be refined, but the system as a whole has been implemented and subjected to limited simulated tests. In these tests it has proven successful at detecting and controlling casualty events, while avoiding the waste of valuable resources on nuisance events.

A number of design issues are still considered open research challenges, and as such are being constantly reviewed. These include casualty detection and identification, the planning and re-planning of DC actions under uncertainty, and resource management.

11. References

Bradley, M., Downs, R., Durkin, A., Lestina, T., Runnerstrom, E. and Williams, F.W., "Evaluation of Reflexive Logic for Shipboard Fire Main," NRL Memorandum Report, NRL/MR/61980—00—8429, January 12, 2000

Bradley, M., Burns, S., Downs, R., Lestina, T., Roberts, M., Runnerstrom, E., Yufik, Y.M., Sheridan, T., Tatem, P.A. and Williams, F.W., "Damage Control Automation for Reduced Manning Supervisory Control System Development/ Phase I" NRL Memorandum Report NRL/MR/6180—00--8468, June 26 2000

Englemore, R.S. and Moran, A.J., editors (1988) "Blackboard Systems," Wokingham England: Addison-Wesley Publishing Company

Gottuk, D., Hill, S., Schemel, C., Strehlen, B., Rose-Pehrsson, S., Schaffer, R., Tatem, P., Williams, F. (1998). "Identification of Fire Signatures for Shipboard Multi-criteria Fire Detection Systems," NRL Memorandum Report NRL/MR/6180-99--8386, June 18, 1999.

Gottuk, D. and Williams, F.W., "Multi-Criteria Fire Detection: A Review of the State of the Art," NRL Letter Report, 6180/0422, 10 September 1998.

Grosshandler, W.L. (1995). *A Review of Measurements and Candidate Signitures for Early Fire Detection*. Building and Fire Research Laboratory/Fire Science Division, Report No. NISTIR 5555, 1995.

Hammond, K.J. (1986). CHEF: A Model of Case-based Planning. *In Conference Proceedings of the AAAI,* 1986.

Heckerman, D. (1995). *A Bayesian approach to learning causal networks*. Technical Report MSR-TR-95-04, Microsoft Research, March, 1995.

LeMentec, J.C., Glenn, F., Zachary, W. and Eilbert, J., "Representing Human Sensory and Motor Action Behavior in a Cognitive Modeling Architecture," ONR/SC-21, S&T Manning Affordability Initiative (1999)

Mawhinney, J.R., DiNenno, P.J. and Williams, F.W., "New Concepts for Design of an Automated Hydraulic Network for a Water Mist Fire Suppression System on Navy Ships," NRL Letter Report, 6180/0292, 18 July 2000

Naval Sea Systems Command, "*Naval Ships' Technical Manual, Chapter 555, Volume 1, Surface Ship Firefighting*," Forth Revision, S9286-S3-STM-010/CH-555V1, 1998.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference.* Morgan-Kaufmann, San Mateo, CA, 1988.

Peatross, M., Scheffey, J., Wong, J., Farley, J., Tatem, P. and Williams, F.W., "FY 2000 DC-ARM Demonstration Test Plan," NRL Letter Report, 6180/0247, 23 June 2000

Williams, F.W., Scheffrey, J.L., Wong, J.T., Toomey, T.A., and Farley, J.P., "1993 Fleet Doctrine Evaluation Workshop: Phase I-Class A Fire/Vertical Attack," NRL/MR/6180--93-7429, December 30, 1993.

Williams, F.W., Farley, J.P., Gottuk, D.T., and Peatross, M.J., "1995 Class B Firefighting Doctrine and Tactics: Final Report," NRL Memorandum Report NRL/MR/6180-67-7909, January 13, 1997.

Zachary, W., LeMentec, J.C. and Ryder, J. "Interface Agents in Complex Systems: Conceptual Principles and Design Practice, Norwell, MA, Kluwer Academic Publishers, 1996, pp 35-52

Zachary, W. and Ryder, J. (1997), "Decision support: integrating and aiding," Handbook of Human Computer Interaction, 2nd Edition, Amsterdam: North Holland, 1996, pp 1235-1258

Zachary, W., Ryder, J. and Hicinbothom, J.. "Cognitive modeling and decision making research," Making Decisions Under Stress, Am Psych Assn, Washington DC, 1998, pp 315-344