Safety Science 103 (2018) 1–11

Contents lists available at ScienceDirect

Safety Science

journal homepage: www.elsevier.com/locate/safety

Safety engineering of computational cognitive architectures within safety-critical systems

Harry H. Dreany[a,*], Robert Roncace[b], Paul Young[a]

[a] Department of the Navy, Naval Surface Warfare Center, Dahlgren Division, 6096 Tisdale Road, Dahlgren, VA 22408, USA
[b] George Washington University, Department of Engineering Management & Systems Engineering School of Engineering and Applied Science, 1 Old Oyster Point Rd, Suite 200, Newport News, VA 23602, USA

ARTICLE INFO

Keywords:
Safety engineering
Artificial intelligence
Cognitive architecture
Decision support model
Intelligent technologies

ABSTRACT

This paper presents the integration of a cognitive architecture with an intelligent decision support model (IDSM) that is embedded into an autonomous non-deterministic safety critical system. The IDSM will integrate multi-criteria decision making via intelligent technologies like expert systems, fuzzy logic, machine learning and genetic algorithms.

Cognitive technology is currently simulated in safety–critical systems to highlight variables of interest, interface with intelligent technologies, and provide an environment that improves a system's cognitive performance. In this study, the IDSM is being applied to an actual safety–critical system, an unmanned surface vehicle (USV) with embedded artificial intelligence (AI) software. The USV's safety performance is being researched in a simulated and a real world nautical based environment. The objective is to build a dynamically changing model to evaluate a cognitive architecture's ability to ensure safe performance of an intelligent safety–critical system. The IDSM does this by finding a set of key safety performance parameters that can be critiqued via safety measurements, mechanisms and methodologies. The uniqueness of this research will be on bounding the decision making associated with the cognitive architecture's key safety parameters (KSP).

Other real-time applications that could benefit from advancing the safety of cognitive technologies are unmanned platforms, transportation technologies, and service robotics. The results will provide cognitive science researchers a reference for safety engineering artificially intelligent safety–critical systems.

1. Introduction

Safety engineering of a cognitive architecture in safety-critical systems has had few successes over the past three decades owing to a lack of determinism and predictability of the architecture's safety performance (Varadaraju, 2011). To address this, the adaptability of the architecture needs to be constrained through the use of fault-tolerant design as a way to provide safety assurances (Avizienis, 1985). Accomplishing this in a cognitive architecture is done by merging expert systems, fuzzy logic, machine learning and genetic algorithm concepts (Kowalski et al., 2005; Pal et al., 2012).

This study began by integrating a cognitive architecture into an artificially intelligent, safety-critical system, which in this case is an USV. The intelligent system used on the USV is called the autonomous small unit riverine craft (ASURC). The cognitive architecture is based on the SOAR cognitive architecture, an architecture that has controlled robotic platforms prior to this experiment (Laird et al., 1987).

The research observed and evaluated the ASURC's performance of the safety-critical objective of avoiding an unintended collision event with a dynamic obstacle. The system will sense its environment in order to maintain safe distances, appropriate time responses and platform control required by the scenario. When the obstacle was detected, the sensor data was then evaluated at a symbolic level by the AI controller. Performance results led to safety measurements, mechanisms, and methodologies that quantified the ASURC's responses, such as the appropriate turn angle at the appropriate time based on what the variable tolerances are for a given task.

The requirement for determinism and predictability increases the complexity of tasking in any environment. This safety requirement is understood for some unmanned and autonomous platforms. However, the ASURC executed the experiment in a maritime environment, which required USV-specific safety research that is novel to that of an unmanned or autonomous USV without a cognitive architecture. The input provided influenced the logic for the ASURC's safety design and is what makes this research novel.

The prescribed tolerances for this experiment were based on the

* Corresponding author.
E-mail addresses: Harry.Dreany@Navy.mil (H.H. Dreany), Roncace@gwu.edu (R. Roncace), Paul.G.Young@Navy.mil (P. Young).

https://doi.org/10.1016/j.ssci.2017.10.020
Received 1 February 2017; Received in revised form 1 September 2017; Accepted 27 October 2017
Available online 14 November 2017
0925-7535/ Published by Elsevier Ltd.

Table 1
Boat safety levels.

Safety level	1	1	1	2	3	4
Effect on craft and occupants	Normal	Nuisance	Operating limitation	Emergency procedures; significant reduction in safety margins; difficult to crew to cope with adverse condition; passenger injuries	Large reduction in safety margins; serious injury to small number of occupants	Deaths, usually with loss of craft
Effect category	Minor			Major	Hazardous	Catastrophic

International Code of Safety for High-Speed Craft (2000)–2008 Edition. This reference defines a safety failure as, "Any improper operation resulting in a hazardous or catastrophic effect." It classifies maritime vessel safety into four levels (1–4) based on the level of degradation of safety with regard to personnel and equipment (The Maritime and Coastguard Agency, 2008). Table 1 lists the levels of safety.

The experiment was conducted in a real-world nautical environment as well as a simulated environment tuned to represent all the appropriate physics models of a nautical environment. This gave the safety engineer the design specifications to ensure safe performance of the software subcomponents. The participants in the experiment were as follows:

(a) Real and simulated USVs in simulated environment with a human controller and an AI controller
(b) Real USV in real-world environment with human controller and AI controller

The data used to determine system performance were based on the safety levels in Table 1; a human operator executing scenarios; and information garnered through an expert elicitation process prior to, during, and after the experimentation. The AI controller system then executed the same scenarios on the USV in a rea-world environment as well as in simulation to test the variables of interest that affect avoiding an unintended collision event with a dynamic obstacle. The criteria listed below aimed to provide way to determine and predict AI controller behavior.

(a) Determine what a safety failure is for transitioning from scripted synthetic task to an unstructured task based on Table 1.
(b) [Analyze] feedback of what the boundaries should be for the variables of interest to ensure safe task execution.
(c) {Develop} a model to show how boundaries of the variables of interest affect the decisions to safely complete a task.

2. Material and method

2.1. Prior work

Early AI research dealt with creating systems that emulated human actions using programming techniques such as "if-then" statements and basic searches (Geramifard et al., 2013). These AI systems were only "intelligent" with regard to the tasks they were designed to complete. The depth of knowledge was also only as current as the latest program update (Laird et al., 1987).

Cognitive architectures were developed based on how human cognition works; for example—learning, memory, and decision making (Lehman et al., 2006). Examples of intelligent systems that have demonstrated human traits are:

(a) Learning Applied to Ground Vehicles demonstrated navigational algorithms merged by adaptive methods. These methods used information from previous experiences to change the system's behavior accordingly (Jackel et al., 2006).
(b) Control Architecture for Robotic Agent Command and Sensing demonstrated the capability to deterministically react to unanticipated occurrences and re-plan in the face of changing goals, conditions, or resources (Huntsberger and Stoica, 2010).

An issue with many cognitive architectures (not necessarily the ones mentioned above) is that their design does not allow them to effectively deal with unpredictable real-world environments. The development of cognitive architectures and their safety design has mostly occurred in environments like gaming and training simulations because of the comprehensive knowledge and understanding of those environments. This trend has stifled the development of cognitive architectures, while newly developed autonomous systems are created without a compatible architecture. This has led to development occurring mostly by the manipulation of pre-existing platforms and their capabilities. Essentially, implementation of cognitive architectures has mostly been additive to deterministic systems rather than a symbiotic development with an autonomous platform (Lehman et al., 2006).

2.2. Safety engineering process

System-level hazards are situations unsafe to personnel and equipment. These hazards can be adjudicated through a safety engineering process designed to identify, analyze, control, and mitigate them. Safety critical systems, including humans, will never be considered 100 percent safe (Storey, 1996); however, by using safety processes, these safety failures can be identified and mitigated (Bell and Reinert, 1993). Fig. 1 displays the safety engineering process that identified the primary safety concerns associated with the ASURC avoiding an unintended collision with a dynamic obstacle.

(a) The first step was conducting a systems safety analysis of the ASURC that determined and evaluated the various applications and subsystems involved. This was initially conducted at implementation and was continuously reassessed during all stages of development. The purpose was to garner information that ensured safety requirements were feasible (Kurd et al., 2007).
(b) Next, a high-level functional hazard analysis was performed that identified the major operational functions of the USV that provided an understanding of what an incident for each identified hazard could be (Kurd et al., 2007).
(c) Then a preliminary analysis was performed that identified hazards, causal factors, and generic mishaps. This was performed at the start of the software life cycle to explore possible hazards.
(d) Next, safety tasks were postulated based on the above parameters as well as human based "trust" performance criteria, identified by human controlled runs of the scenarios and interviews (Trafton et al., 2006).
(e) This all led to identifying preliminary safety concerns that confirmed the ASURC's designs obeyed and improved the safety requirements by influencing development design (Kurd et al., 2007).
(f) This process iteratively continued to compare hazards, causal factors, and generic mishaps to continuously update the primary safety concerns. The last phase provided a clear and defendable argument that the system performed within the tolerances for being considered safe. The correlation and mapping of the information ensured the following:
(1) Proper wording to properly address the hazards and causal

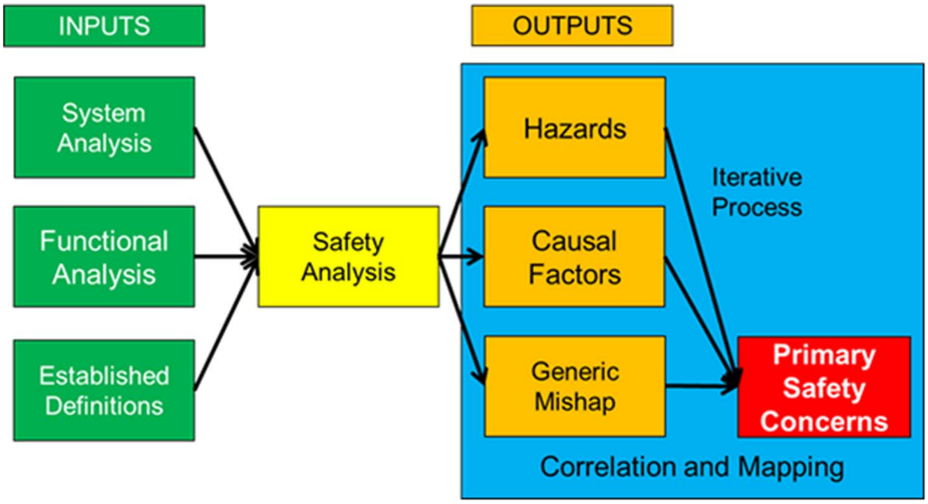

Fig. 1. Safety engineering process.

factors in the thematic language so that the ASURC understands per the language scheme

(2) Primary safety concerns were not too restrictive, which would make the system deterministic
(3) Primary safety concerns provided value-added guidance (Kurd et al., 2007)

Despite the process to ensure safety, there will be inaccuracies in the IDSM which will impact the performance of the ASURC. Learning techniques will allow the adaptation of the model, thus providing better performance (Geramifard et al., 2013). However, the process does require a large number of interactions in order for the learning techniques to reach full potential. It is therefore essential to develop techniques for measuring, predicting, and controlling the performance of the AI system. The IDSM will discover the best way to bind the variables of interest, mitigating the risk. A cognitive system without an IDSM approach does not have the necessary multi-criteria decision-making ability that engenders "trust" (Işıklar et al., 2007).

2.3. Intelligent decision support model (IDSM)

DSMs support the user through data analysis (Klingman et al., 1986). The decision making processes accommodate changes in the environment due so their flexibility and adaptability (Klingman et al., 1987). In this study the DSM has evolved into an IDSM. The IDSM established a basis for designing and developing cognitive architectures for safety-critical applications. This provides focus on identifying, analyzing, controlling, and mitigating performance failure of a safety-critical system (Kurd et al., 2007).

2.3.1. Multiple criteria decision making (MCDM)

MCDM is a decision-making structure that encompasses numerous different criteria and whose purpose is to aid decision makers. Normally, there is not an exclusive optimal result for problems and a decision maker's parameters are required to distinguish between methods (Köksalan et al., 2011). The cognitive architecture is the decision maker in this case. The IDSM will integrate MCDM methods through the intelligent technologies used in the ASURC. This section explains the decision-making task breakdown of the MCDM process.

The goal is to choose and perform the alternative that meets all the criteria in the safest way possible. Each obstacle-avoidance maneuver provides numerous alternatives. For example, passing a dynamic obstacle can be executed at slow or fast speed, in near or distant proximity, and on either side of the obstacle (Furda and Vlacic, 2011). To accomplish this goal, the process is broken out as follows, and is displayed in Fig. 2.

Step 1: Select a set of feasible alternatives. Those maneuvers that cannot be executed by a USV are not considered.
Step 2: Select a subset of feasible alternatives that obey the *International Code of Safety for High-Speed Craft.*
Step 3: Choose the best alternative. Since candidate alternatives are feasible and obey maritime regulations, these alternatives are considered efficient and safe (Chen et al., 2014).

The lists of considered alternatives for this experiment are:

(a) a_1: maneuver to point fast
(b) a_2: maneuver to point slow
(c) a_3: overtake obstacle slow/bear
(d) a_4: overtake obstacle slow/far
(e) a_5: overtake obstacle fast/near
(f) a_6: overtake obstacle fast/far

A hierarchy of objectives was used to choose the best alternatives. The hierarchy began at the most basic and was then broken into more specific objectives that reside on a lower hierarchy level. The base of the hierarchical levels contained only objectives (*obj)* that could be measured with variables. The lower level objectives identified how to accomplish the higher level objectives (Furda and Vlacic, 2011). The following is an example of the objective hierarchy:

(a) Maneuver to objective safely: $obj^{Level\ 1}$
  (1) $obj_1^{\ Level\ 2}$: stay in the water
    (a) *variable* (v_1): keep safe distance from the shore
    (b) v_2: travel in cardinal direction
  (2) $obj_2^{Level\ 2}$: no unintended collisions
    (a) v_3: keep safe distance from obstacles
    (b) v_4: maneuver at safe distances around obstacles
    (c) v_5: avoid unexpected stopping
    (d) v_6: avoid severe direction changes (Furda and Vlacic, 2011)

Weights were assigned to each variable to define various levels of importance. The importance of the objectives changes depending on the environment in which it is performed. For example, at a higher speed, v_5: avoid unexpected stopping is more important than v_1: keep safe distance from the shore. This method allows adjusting the weights according to the environment in which it is being performed (Furda and Vlacic, 2011).

Variables and alternatives are represented as follows:

$$Alternatives(a) = \{a_1,a_2,a_q\},\quad (q = number\ of\ alternatives) \tag{1}$$

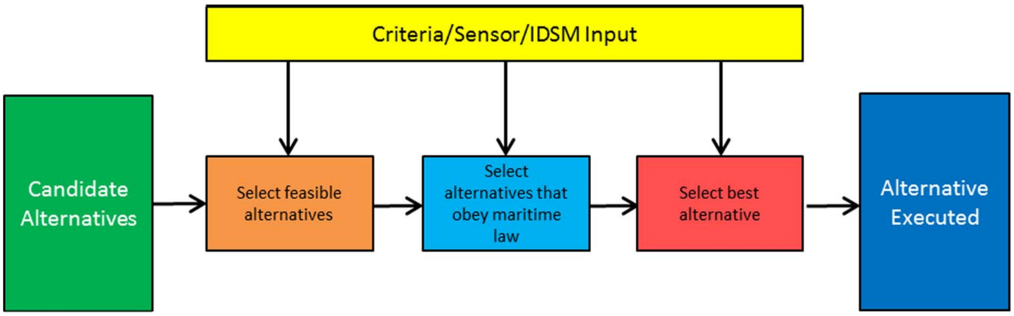

Fig. 2. Decision-making process.

Table 2
Decision matrix.

		Variables (v)						
Alternatives(a)		v1	v2	.	.	.	vy	V(aq)
	a1	*f1(a1)*	*f2(a1)*	.	.	.	*fv(a1)*	
	a2	*f1(a2)*	*f2(a2)*	.	.	.	*fv(a2)*	
	.	.	.	.	.	.	.	
	.	.	.	.	.	.	.	
	.	.	.	.	.	.	.	
	aq	*f1(aq)*	*f2(aq)*	.	.	.	*fv(aq)*	
	V(*Vy*)							

$$Variables\,(v) = \{v_1,v_2,v_y\},\quad (y = number\ of\ variables) \tag{2}$$

Utility functions $f_q(a_q)$ and $f_y(v_y)$ were used to specify the level of performance achieved by an alternative with respect to the variables.

Defining utility functions for all alternatives and all variables resulted in the decision matrix displayed in Table 2:

The Simple Additive Weighting Method was chosen to calculate the best result, the (Yoon and Hwang, 1995). The value of an alternative (a) and variable (v) are defined as follows:

$$V\,(a_q) = \sum_{i=1}^{q} W_i f_i\,(a_q) \tag{3}$$

$$V\,(v_y) = \sum_{i=1}^{y} W_i f_i\,(v_y) \tag{4}$$

2.3.2. *Subsystems*

Fig. 3 depicts the IDSM subsystems. Their functions are discussed throughout this section (Klingman et al., 1988).

2.3.2.1. Data subsystem. This subsystem is designed to coordinate and sustain the data the IDSM requires to properly support decision makers. This subsystem provides the maintenance and distribution of the data between the IDSM and the cognitive architecture. It also facilitates decision making by enabling the cognitive architecture to access all the system's data in addition to data the IDSM chooses to provide (Glover et al., 1992). The data subsystem accomplishes this by using its own modules (refer to Fig. 4). The subsystem contains several modules, each providing a capability: the Data Definition Module specifies and organizes data; the Data Query Module retrieves data from the Data Storage Modules and Data Management System Module updates and changes data items for continuous scenario analysis (Glover et al., 1992).

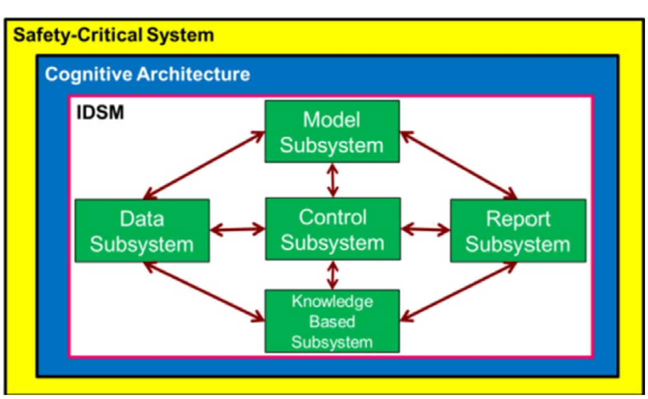

Fig. 3. IDSM.

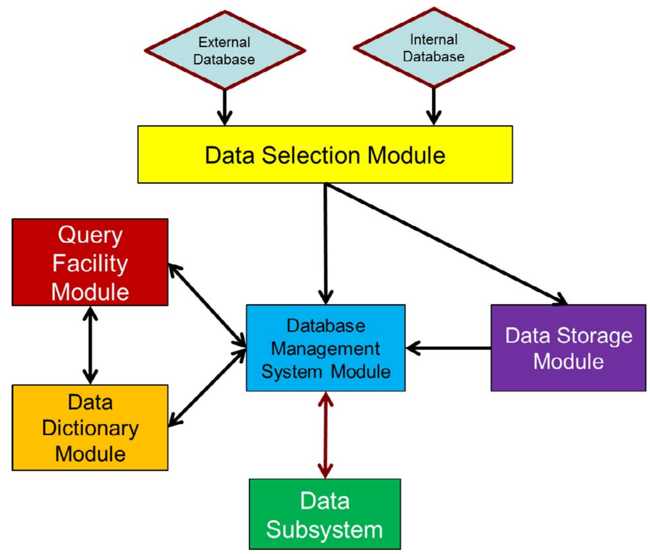

Fig. 4. Data subsystem.

2.3.2.2. Model subsystem. Fig. 5 shows the model subsystem creating and storing models through the IDSM process as an integrated system of systems rather than an individual subsystem (Klingman et al., 1988). This is done by measuring the impact on the planning process, taking into consideration things like resource restrictions and platform capabilities. The role of this subsystem is to perform model generations to identify appropriate solutions for the decision-making process (Dreany et al., 2015).

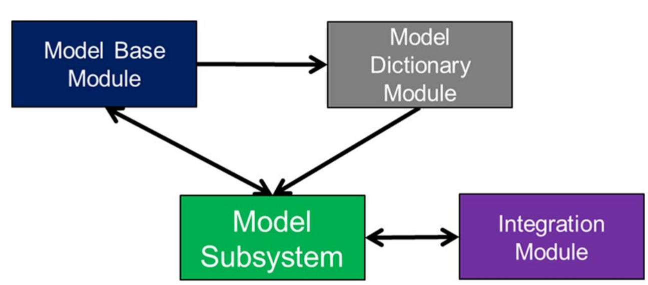

Fig. 5. Model subsystem.

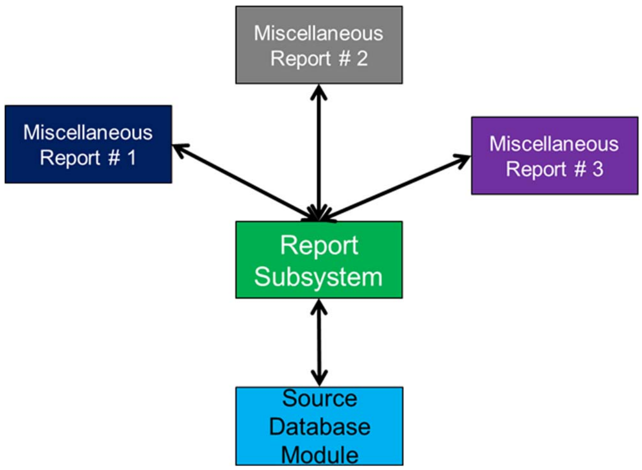

Fig. 6. Report subsystem.

As the situation changes, the model evolves by articulating the description of the data presented to the model as inputs instead of being hard coded into the software (Dreany et al., 2015). This allows input changes to convey the model's evolution without requiring software changes (Stanciu, 2009).

2.3.2.3. Report subsystem. This subsystem is the key to communication between the model subsystem and the cognitive architecture. The report subsystem accesses the data and model solutions from the source database module. Fig. 6 depicts how it generates the reports identified by the architecture (Dreany et al., 2015).

The reports facilitate changes as the architecture allows the system to evolve. The report subsystem is flexible to easily allow inserting new reports or changes to existing reports into the system. This feature is necessary as the system will encounter unscripted scenarios requiring additional informational as it learns and become more sophisticated (Glover et al., 1992).

2.3.2.4. Knowledge base subsystem. This subsystem is what makes the IDSM intelligent through its design to mirror human judgments. This is an expert-assisted, decision-making paradigm used to establish future uncertainty and to select which models are best suited to make safe decisions. This system uses a Bayesian estimation methodology for combining variables of interest to prevent a decision maker from selecting a design that serves one objective more than another (Mostaghimi, 2001). Then adjustments are performed using a database of "if-then" rules. Individual rules in the database can be edited, removed, or new ones can be added. The knowledge base subsystem is made up of rules and data files, the model solutions, and the reports (Dreany et al., 2015). This subsystem employs intelligent technologies (Kowalski et al., 2005; Pal et al., 2012) here, so the system learns from experience through data and observations, which alters its behavior (Glover et al., 1992).

The Language Management Module is associated with the language information that is displayed. The module conducts analysis in terms of symbolic language (Kurd et al., 2007) by transitioning from the symbolic language to a language the IDSM can understand for it to be analyzed. Fig. 7 shows the transfer of the language management module knowledge to the knowledge base subsystem, which is how the data gets into the IDSM.

2.3.2.5. Control subsystem. This subsystem conduits information among all subsystems and is the primary interface for the safety-critical system and the cognitive architecture. This interface, displayed in Fig. 8, facilitates the cognitive architecture providing input to the IDSM (Glover et al., 1992). For example, the AI system solves a basic problem, then solves another more complex problem, and then compares the results. The analysis involves input from numerous modules in each subsystem. The subsystem requests the other subsystems as required to execute these types of commands (Dreany et al., 2015).

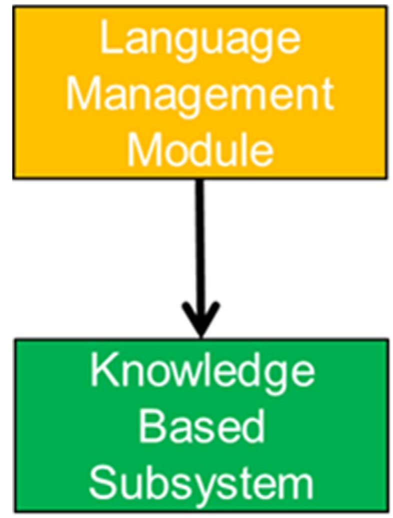

Fig. 7. Knowledge base subsystem.

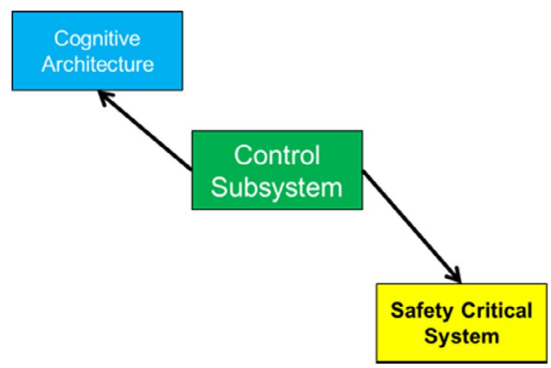

Fig. 8. Control subsystem.

The control subsystem consists of a set of high-level thematic and symbolic control statements custom made to the system's theoretical understanding of a situation. These statements are used to issue commands (Dreany et al., 2015). These instructions are flexible to generate alternative instructions for "what-if" type scenarios. The flexibility fosters a structure and composition of the statements as a natural language to ensure system usability (Glover et al., 1992).

3. Theory/Calculation

3.1. Path planner

The research observed the software subcomponent known as the Path Planner. The experiment identified what the measures of performance were for the key performance parameters of the path planner. Then the ranges of the parameters were adjusted based on that analysis.

The ASURC path planner has a three-level structure:

(a) A Lifelong Planning A∗ algorithm—that combines ideas from the AI to locate the most direct path from a start point to a target location while the environment constantly changes and dynamic obstacles appear—is used for the far field path planning (Koenig et al., 2004).
(b) The mid field path planning is done using a Field D∗ algorithm, which is an incremental planning and re-planning search algorithm

.6eV-p	5V Objective -5V			.6eV-p	.75eV-n	.75eV-n	.75eV-n
.6eV-p				.6eV-p	.75eV-n	Obstacle: .6eV and 3V	
.6eV-p	.6eV-p	.6eV-p	.6eV-p	.6eV-p	.75eV-n		
.75eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n		
.75eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n
Obstacle: .6eV and 3V			.75eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
			.75eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
			.75eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
			.75eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
.75eV-n	.75eV-n	.75eV-n	.75eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.75eV-n	Obstacle: .6eV and 3V		.75eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.75eV-n			.75eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.75eV-n	.75eV-n	.75eV-n	.75eV-n
.8eV-n	.8eV-n	.8eV-n	.8eV-n	.75eV-n	.8eV-n	.8eV-n	.8eV-n
.8eV-n	.8eV-n	Me: Electron	.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
.8eV-n	.8eV-n		.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n
.8eV-n	.8eV-n		.8eV-n	.8eV-n	.8eV-n	.8eV-n	.8eV-n

Fig. 9. Near field path planning methodology.

to produce globally-smooth paths (Ferguson and Stentz, 2007).

(c) The near field path planning is done using a carrier transport model. The model uses an algorithm that analyzes regions of desired traversability, represented as energy states F. Dilatation of objects and approach is controlled by fix potentials that affect USV transport properties. Thematic role representation of the path planning is done by adjusting the approach angle using a noun-verb methodology. An example of this is seen in Fig. 9, with the path planning following the "greater than .7eV" and the carrier are attracted to the lowest potential and the line representing the path selected.

3.2. Expert elicitation

In this study, expert elicitation served as a scientific consensus methodology. Expert judgment will be provided as data via verbal and written communication methods in response to the task (Meyer and Booker, 1991). This study used three basic elicitation techniques:

(a) Individual interviews: experts privately interviewed face to face
(b) Interactive groups: group of experts and a mediator met in a face-to-face situation
(c) Delphi situations: experts were isolated and given anonymous judgments to a moderator (Meyer and Booker, 1991)

Fig. 10 shows the Elicitation Approach Structure for how information was collected from the experts (Avizienis, 1985). The process is a derivation of "Cooke's Method" (Cooke, 1991).

(a) Phase 1 is where the group of experts in the domain is selected
(b) Phase 2 is the calibration of the experts
(c) Phase 3 is where the experts are scored (Cooke, 2008).

This method calculates the distribution of the decision maker's variables as a weighted value of the experts' input. The weights are based on the experts' result with the seed variables (Hammitt and Cohen). The weight of each expert is proportional to the expert's information and calibration scores (The Maritime and Coastguard Agency, 2008).

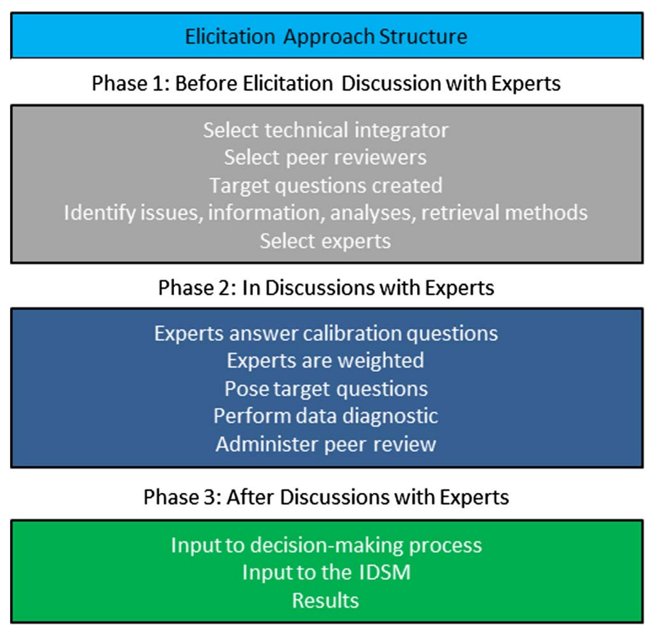

Fig. 10. Elicitation approach structure.

3.2.1. How the experts were chosen

The population for this elicitation was a group of licensed U.S. Naval coxswains that were made available for the experiment by two separate organizations. A total of ten subject matter experts (SME) were used. Three were civilians and seven were U.S. Navy Sailors. The ten SMEs possessed varying degrees of experience with maritime operations. The SMEs had an average of over 11 years of experience and were well-versed in operating different types of sensor systems. Table 3

Table 3
Experts' level of experience.

#	ID	Experience	Status
1	SM	25	CIV
2	PY	18	CIV
3	ER	9	CIV
4	CPO2	18	MIL
5	SE2	2	MIL
6	PO3	5	MIL
7	PO2	9	MIL
8	PO1	12	MIL
9	PO2B	6	MIL
10	PO2C	7	MIL

shows the level of experience each SME had. The chosen participants are considered experts in the navigation of maritime platforms and possessed expert knowledge of the COLREGs.

3.2.2. Validating the experts

Initial vetting of the experts' background in maritime navigation was done by the providing organization. Experts selected were known to the supervisors of the organizations to have experience and knowledge of the issues facing maritime navigation. To validate the experts' knowledge in COLREGs, a set of nine seed questions were included in the elicitation. The experts' individual answers were compared to overall group responses to identify the level of knowledge in the subject area. The responses were plotted to ensure a normal distribution. Questions about the COLREGs were developed with influence from the leadership of the two organizations providing the experts. The questions used are listed below:

(1) How important is the controller's ability to maintain a proper lookout by sight and hearing as well as by all available means appropriate in the prevailing circumstances and conditions so as to make a full appraisal of the situation and of the risk of collision? (COLREG Rule # 5)
(2) How important is the controller's ability to determine what a safe speed is? (COLREG Rule # 6)
(3) How important is the controller's ability to determine if risk of collision exists? (COLREG Rule # 7)
(4) How important is the controller's ability to recognize colors and shapes?
(5) How important is the controller's ability to understand when it is acceptable to overtake another vessel? (COLREG Rule # 13)
(6) How important is the controller's ability to avoid a head-on situation? (COLREG Rule # 14)
(7) How important is the controller's ability to properly handle a crossing situation? (COLREG Rule # 15)
(8) How important is the controller's ability to know the responsibilities between vessels? (COLREG Rule # 18)
(9) How important is the controller's ability to control vessels in restricted visibility environment? (COLREG Rule # 19)

3.2.3. Chi-square test for goodness of fit

A Chi-Square Test for Goodness of Fit test was conducted to assess whether or not the responses from each question came from a normally distributed population or not. The Null and Alternative Hypothesis that were tested are:

- H_o = The Data follows the normal distribution
- H_1 = The data does not follow the normal distribution

There were two observations made to determine if the response data was normally distributed. The first was to compare the P-Value with the alpha. The P-value was more extreme than the alpha for each question, indicating weak evidence against the null hypothesis, thus, failing to reject the null hypothesis. The second observation was to compare the Chi Square value with the Critical Value. The Critical Value is the point on the X-axis that is the Alpha (0.05 in this case). As with the P-Value, if the Critical Value is greater than the Chi Square value, we do not reject the Null Hypothesis. The Critical Value was greater than the Chi Square value for each question's responses.

Table 4
Chi Square Goodness of fit testing results for each question.

Question	ChiSquare	Critical value	P-Value	Alpha	Hypothesis Test
1	5.45	16.92	.79	0.05	Do not reject
2	4.24	16.92	.90	0.05	Do not reject
3	3.33	16.92	.95	0.05	Do not reject
4	1.85	16.92	.99	0.05	Do not reject
5	3.06	16.92	.96	0.05	Do not reject
6	3.00	16.92	.96	0.05	Do not reject
7	3,57	16.92	.94	0.05	Do not reject
8	3.44	16.92	.94	0.05	Do not reject
9	4.88	16.92	.85	0.05	Do not reject

Table 4 presents the results of the Chi Square Goodness of Fit testing on the responses of the nine questions from the ten participating experts. The table represents how the response for each question, with 95% certainty, is normally distributed.

A normal distribution was expected, since the nine seed questions were designed to produce a range of answers. Fig. 11 is a probability plot of the seed questions response with 95% confidence bands displayed.

The data produces a p-value of .17 which is within the 95% confidence interval of 0.05. Each expert has only 9 data points, so a normal distribution with a 95% confidence interval would not be expected in every case. A boxplot of the means, displayed in Fig. 12 indicates a problem with one of the experts.

According to the boxplot, responses of expert # 10 appear to be uniform. A closer look at the responses from expert # 10 confirms that the expert answered at the midpoint for every seed question. While this may be a valid opinion, it could also indicate a lack of understanding of the value of responses or a desire to not participate. For these reasons, expert # 10 was eliminated from the study.

3.2.4. Who did the elicitation?

The administration of the elicitation was in the form of interviews, questionnaires and interactive groups. The questions, interviews and groups were led and administered by a Navy government researcher. The experts were geographically dispersed requiring the elicitation techniques to occur during numerous occasions.

Respondents were not asked to give specific probabilities, but to choose from one of five ranges: less than 10%, 10–40%, 40–60%, 60–90%, and greater than 90%. The elicitation was designed this way to allow the experts a greater freedom in responding in areas where exact probabilities would be very difficult.

3.2.5. Tacit knowledge translated into explicit knowledge

The translation of tacit knowledge to explicit knowledge was critical as it established how the it became part of the Cognitive Architecture's knowledge. The translation of this knowledge was facilitated by knowledge exchange protocols. A knowledge exchange protocol is a method that structures knowledge exchange such that the supplier of the knowledge and the recipient can systematically present and recall knowledge respectively. This research used a variation of the SOAP protocol that is predominantly known in the medical community (see Table 5). The protocol uses a structure that documents situation based obstacle encounters by the platform. The protocol provides a consistent framework for:

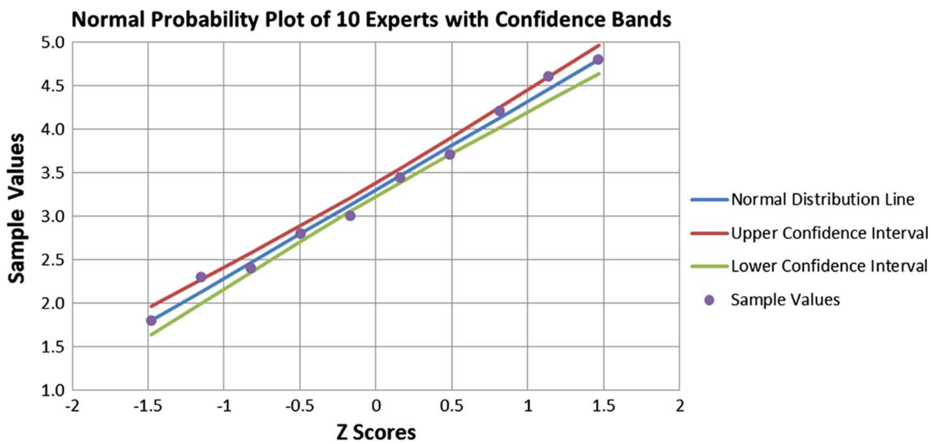

Fig. 11. Probability distribution of average seed data responses for the experts.

- Structuring COLREGs narratives
- Understanding the SME's thinking about how to avoid a dynamic obstacle
- Learning about methods employed by the SME in the knowledge creation process
- Sharing the SMEs' reasons for actions taken to avoid the dynamic obstacle

The protocol provides a consistent instrument for documenting:

- What the SME understands about COLREGs (sense-making activities)
- How the SME closes gaps in their understanding about the avoiding obstacles (knowledge creation)
- What actions the SME takes relative to the situation (decision making).

Use of the protocol allows SMEs to accrue knowledge about obstacle situations over time. Moreover, by structuring the documentation of the SME's knowledge on avoiding obstacles, tacit knowledge is being externalized (Herschel et al., 2001).

3.2.6. Calibration and information

Cooke's Classical Model of quantifying the performance of subject matter experts (SME) was used to provide weights to the participating SMEs. These weights use quantitative measures of performance, calibration, and information (Cooke, 2008). Each SME was regarded as a statistical hypothesis. Calibration measures the probability that the experimental results agree with the SME's assessments (The Maritime and Coastguard Agency, 2008). The information calculation measures the determination of the distribution. Implementation of these measures was done in quantile elicitation layouts. This format presents SMEs an undefined quantity and gives a quantile location of the uncertainty distribution, which is subjective. Each input falls in one of the following quantile intervals with the following probability vector (Herschel et al., 2001):

Table 5
Example of knowledge exchange protocol.

Example: Knowledge Exchange Protocol
Expert ID
SM
Subjective (a brief narrative of the scenario)
Objective (a description of the specific task used to learn the true nature of the situation
Speed
Distance
Direction
Type of Obstacle
Avoidance
Maneuver
Assessment (a decision about what the details are for the obstacle
Plan (a prescribed alternative for the architecture to alleviate the problem(s))

$$p= (0.05;0.45;0.45;0.05) \quad (5)$$

A known result in statistics is that when sampling S acts independently from the distribution with probability vector P, the quantity (Herschel et al., 2001)

$$2NI\,(S,P) = 2N\sum_{i=1}^{n} S_i ln\left(\frac{S_i}{P_i}\right) \quad (6)$$

is asymptotically distributed as a chi-square variable with n degrees of freedom, which is known as the likelihood ratio statistic. I(S, P) is the relative information of distribution S with respect to P. By removing the foremost term of the logarithm we find the chi-square test statistic for goodness of fit (Herschel et al., 2001). Let X_n^2 signify the Chi-square variable with n degrees of freedom (Herschel et al., 2001).

$$C= 1-X_n^2(2NI(S,P)) \quad (7)$$

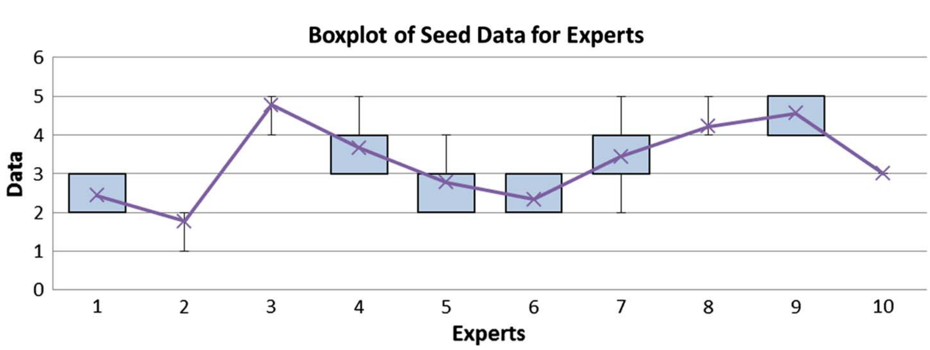

Fig. 12. Boxplot of seed data for experts.

(a) C: calibration score of a SME
(b) N: number of seed variables.
(c) S: sample distribution of a SME, which is a computation of the relative frequencies and the vector of these relative frequencies
(d) P: probability corresponding to the size of the intervals
(e) 2NI(S, P): relative information of S and P (Herschel et al., 2001)

The information score is the level of concentration of the distribution. The information score becomes the average relative information from one SME over all of the variables (Herschel et al., 2001). The relative information between two densities f and g on an intrinsic range I is

$$I(f|g) = \int_L^U f(x) ln\left(\frac{f(x)}{g(x)}\right) dx \tag{8}$$

The background measures need a basic range on which these measures are concentrated. Cooke's model implements the k% overshoot rule, which is that [in] each item we study the smallest interval I = [L, U] that contains the assessed quantiles of the SMEs and the realization. This interval is extended to (Herschel et al., 2001):

$$L_i = L_i - \frac{k}{100}(U_i - L_i); \quad U_i = U_i + \frac{k}{100}(U_i - L_i) \tag{9}$$

The performance weight for each SME is calculated by combining the calibration and information scores. The formula for the performance weight score (e) is Herschel et al., 2001:

$$w_\alpha(e) = l_\alpha(calibration\ score) \times calibration\ score(e) \times information\ score \tag{10}$$

After weighting and after SMEs are individually elicited regarding their uncertainty judgments in relation to questions of interest (Target Items) the "decision maker" is computed (Herschel et al., 2001):

$$DM_\alpha(i) = \frac{\sum_{e=1}^{E} w_\alpha(e) h_e}{\sum_{e=1}^{E} w_\alpha(e)} \tag{11}$$

where

(a) DM: decision maker = cognitive architecture
(b) e: experts selected
(c) E: number of experts participating
(d) W: weight of the expert
(e) Where h_e is expert *e*'s density function where the summations are taken over all experts (Herschel et al., 2001)

The Simple Additive Weighting Method was the scoring method chosen to conduct the calculations (Furda and Vlacic, 2011). The value $V(a_z)$ of an alternative a_z was calculated by multiplying the utility function values with the variable weights and then summing the products of all variables (Yoon and Hwang, 1995). The alternative with the highest scores was chosen to be performed. It also takes into account the additive nature of the compound safety risk of how decisions made early in the scenario can impact decisions made later in the scenario (Furda and Vlacic, 2011).

The utility functions f_i (A) assessed the success level of each variable *i* for each of the 6 alternatives discussed earlier. The utility function values were scaled to the interval between 0 and 1. A value 1 means the best accomplishment of an objective, while 0 means the objective was not accomplished. Each alternative was rated based on how well they fulfilled the objectives at the lowest level of the hierarchy (Furda and Vlacic, 2011). The scale is as follows:

- 1 signifies ideal accomplishment of the objective
- 0.75 signifies good success
- 0.5 signifies adequate accomplishment
- 0.25 signifies poor accomplishment
- 0 signifies unsatisfactory fulfillment

4. Results

The scope of this paper analyzes the results of four tasks the USV negotiated a number of times during the execution of five different scenarios. The following six variables were analyzed during the execution of the four tasks during this research.

(a) Stay in the water: obj_1 $^{Level\ 2}$
  (1) *variable* $(v)_{1:}$ keep distance to the shore
  (2) $v_{2:}$ travel in cardinal direction
(b) No collisions: $obj_2^{Level\ 2}$
  (1) $v_{3:}$ keep minimum distance to obstacles
  (2) $v_{4:}$ maneuver around obstacles
  (3) $v_{5:}$ avoid unexpected stopping
  (4) $v_{6:}$ avoid severe direction changes

The list of the considered alternatives:

(a) a_1: maneuver to point fast
(b) a_2: maneuver to point slow
(c) a_3: overtake obstacle slow/bear
(d) a_4: overtake obstacle slow/far
(e) a_5: overtake obstacle fast/near
(f) a_6: overtake obstacle fast/far

4.1. Data collection

Data for analysis were divided into two datasets: M1 and M2. Dataset M1 was executed by a deterministic AI Controller on a real USV in simulated and real-world nautical environments. Dataset M2 was executed with a non-deterministic AI controller on an actual USV in simulated and real world nautical environments.

4.1.1. Data collection procedures

Each configuration (M1 and M2) execute five scenarios. The results of their execution of the four tasks covered below were analyzed to identify the variables of interest specific to preventing an unintended collision event with a dynamic obstacle.

4.2. Tasks

The experiments were performed with dynamic obstacles to test whether the ASURC could decide how to navigate to a specific location while safely avoiding obstacles.

Task 1: The USV is following the shoreline, relatively far away. In this situation, the architecture has determined the maneuver for going to a waypoint and simply avoiding land has two feasible execution alternatives (a1 and a2).
Task 2: The USV is travelling on a path that will intersect the path of a crossing vessel. In addition to following the shoreline, the architecture determined that intersecting the crossing vessel is feasible by speeding up, slowing down, or maintaining current speed as long as an unintended collision is avoided. This corresponds to all six alternatives being feasible because of the minimum safe distances required when approaching another vessel.
Task 3: The USV is following another vessel, and the objective requires it to overtake the vessel. The architecture determined that all six alternatives are feasible.
Task 4: A stationary dynamic obstacle (a buoy) is in front of the USV. The architecture determined that all six alternatives are feasible.

Table 6
Calculated values $V(a_r)$ of feasible alternatives.

Tasks	Alternatives 1 Det	1 Non-Det	2 Det	2 Non-Det	3 Det	3 Non-Det	4 Det	4 Non-Det	5 Det	5 Non-Det	6 Det	6 Non-Det	P-Value	Stat. Sig. P < .05
1	7.75	7.25	8.5	7.75									0.30	No
2	9.25	6.5	7.5	7.75	8.5	6.75	9	9.5	8.25	9.25	6.75	8.75	0.85	No
3	7.25	9.25	4.75	9.75	6.25	5.5	7	8.25	6.5	8.25	7.75	7.25	0.08	No
4	10.3	7	11	8.75	9.5	8	10	7	7.75	8	9.25	10.25	0.06	No

Utility functions were calculated for each alternative (a_r) and variable (v_z) with each dataset. To check the sensitivity of the decision, attribute weights were set for the variables based on their importance. This method proposed the option to adjust the attribute weights; thus, the decision preferences, based on the situation (Furda and Vlacic, 2011).

Recall that for calculating the best solution, the Simple Additive Weighting Method was chosen (Furda and Vlacic, 2011). The value of an alternative a_r and variable v_z are defined in Eqs. (3) and (4). Tables 6 and 7 show the calculated values $V(a_r)$ for each of the feasible alternatives and the variables for each alternative. The results between each non-deterministic and deterministic value are similar. Two-tailed t-tests were performed on the data from each task to determine if the difference was statistically significant with a significance level of 0.05.

The data show there are no statistical significances between the deterministic and non-deterministic performance of the tasks. There is no credible evidence that deterministic or non-deterministic AI is safer. Although there was no statistical significant difference in the data, the experiment did find differences in the sample.

Despite the data being inconclusive, there are some data points worth mentioning. A total of 20 alternatives were evaluated across the four tasks feasible for execution. Of those alternatives, the deterministic AI controller performed the safest in 50 percent of them; the non-deterministic controller performed the safest in 50 percent.

Another perspective in considering which controller was the safest was taking the aggregate score of the V(ar) and V(vz) utility functions in each task. The deterministic controller had the safest overall score for task 3. The non-deterministic controller had the safest overall score for task 4. Tasks 1 and 2 scored identical between the deterministic controller and non-deterministic controller.

Tables 8 and 9 list the aggregated deterministic and non-deterministic task data. It combines the scores of all the alternatives as well as the variables for each task within the deterministic and non-deterministic systems accordingly.

Table 8
Aggregated alternative data of M1 and M2.

Data Set	Aggregate Score of Alts, Task 1	2	3	4	P-Value	Stat. Sig. P < .05
M1	16.25	49.25	39.50	57.75	.97	No
M2	15.00	48.50	48.25	49.00		

Table 9
Aggregated variable data of M1 and M2.

Data Set	Aggregate Score of Variables, Task 1	2	3	4	P-Value	Stat. Sig. P < .05
M1	16.25	49.25	39.5	57.75	0.97	No
M2	15.00	48.5	48.25	48.99		

5. Discussion

The research herein does not fully present the current state of computational cognitive architectures and their capabilities. However, it does present potential for increasing system safety through the architecture's adaptability. Examples of safety-critical systems that could benefit from this potential are unmanned platforms (DARPA, 1988), transportation technologies (Johnson et al., 2000), and service robotics (Nabney, 2001). At the same time, it requires changes in the techniques used to continue identifying new hazards and causal factors for mishaps.

While the presented evaluation results are inconclusive as to which controller is safer, it does demonstrate that the developed cognitive architecture is appropriate. Additional work is needed to advance development for handling more real-world scenarios, not just obstacle avoidance. Despite the lack of statistical significant differences in the data, there are results that warrant further research to get to an AI capability that safely performs all tasks. For example, the World Model information that is provided needs refinement, the objective hierarchy needs to be extensively expanded, and the set of maneuvers needs to expand (Furda and Vlacic, 2011).

The safety community continues to study the safety assessment of cognitive architecture as an evolving and not yet fully defined requirement. Future research should look to establish an understanding of where existing software system safety analysis and cognitive

Table 7
Calculated values $V(V_y)$ of variables.

Tasks	Variables 1 Det	1 Non-Det	2 Det	2 Non-Det	3 Det	3 Non-Det	4 Det	4 Non-Det	5 Det	5 Non-Det	6 Det	6 Non-Det	P-Value	Stat. Sig. P < .05
1	2.00	1.75	3	2.5	1.25	2	1.75	1.25	6	2.25	2.25	5.25	0.82	No
2	4	4.75	9	10	3	4.25	4.75	2.5	14.25	14.25	14.25	12.75	0.97	No
3	4	4	5	7.5	3.75	4.75	5	4.25	8.25	14.25	13.5	13.5	0.57	No
4	4.5	5	10	9	6	4.5	5.75	4.24	17.25	15	14.25	11.25	0.75	No

development converge.

6. Conclusions

This paper described the development of an IDSM for studying a computational cognitive architecture's ability to safely perform tasks. Numerous sensors have been integrated onto a USV along with the architecture. The ASURC has performed numerous tasks in five scenarios. The research described here denotes capabilities essential for a USV performing safely; however, it does not discover the entirety of the architecture's capabilities.

The use of an MCDM process allows for the possibility of a large number of objectives that would lead to a high-fidelity solution for any scenario and environment (Furda and Vlacic, 2011). The process was broken down into three basic steps. Step one was used to eliminate the possibility of using alternatives beyond the platform's capabilities; step two was used to select a set of feasible alternatives with regard to maritime law; and the third step was tasked with making the optimum decision for the scenario (Chen et al., 2014).

The experiment realized and showed the feasibility of the architecture to dynamically manipulate control algorithms, behavior selection, path-planning algorithms, actuator control parameters, and priority management of the ASURC. The introduction of a safety-critical system computational cognitive architecture into an environment of uncertainty and unpredictability requires rigid and complete analysis of system safety. In future work, the IDSM needs more MCDM techniques to better provide the safety measurements, mechanisms, and methodologies for the computational cognitive architectures to safely perform tasks.

Acknowledgements

I would like to thank everyone who has helped me through this process and contributed to this research. Specifically, I would like to thank my Ph.D. advisors at The George Washington University, Drs. Robert Roncace and Pavel Fomin for their advice, suggestions, and reviews of the ideas, concepts, and approach. They patiently coached me through the process and provided much-needed direction and critique. Additionally, I would like to thank Dr. Paul Young for his mentorship, patience and direction on this research.

References

Avizienis, A., December 1985. The n-version approach to fault tolerant software. IEEE Trans. Softw. Eng. 11 (12).
Bell, R., Reinert, D., 1993. Risk and system integrity concepts for safety-related control systems. Microprocess Microsyst. 17, 13–15.
Chen, Jiajia, Zhao, Pan, Liang, Huawei, Mei, Tao, 2014. A Multiple Attribute-based Decision Making model for autonomous vehicle in urban environment. In: 2014 IEEE Intelligent Vehicles Symposium Proceedings.
Cooke, R.M., 1991. Experts in uncertainty: opinion and subjective probability in science.
Cooke, R.M., 2008. Experience with Expert Judgment: the TU Delft Expert Judgment Data. Wiley Handbook of Science and Technology for Homeland Security, 11/14/2008.
DARPA, 1988. DARPA Neural Network Study. AFCEA International Press.
Dreany, H.H., Young, P.G., Roncace, R., 2015. Safety enginering of computational cognitive architectures within safety-critical systems. In: 33rd International System Safety Conference Proceeding.
Ferguson, D., Stentz, A., 2007. Field D∗: An interpolation-based path planner and replanner. In: Robotics Research. Springer, Berlin Heidelberg, pp. 239–253.
Furda, A., Vlacic, L., 2011. Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making. IEEE Intell. Transp. Syst. Mag. 3 (1), 4–17. http://dx.doi.org/10.1109/MITS.2011.940472.
Geramifard, A., Redding, J., How, J., 2013. Intelligent cooperative control architecture (iCCA): A framework for performance improvement using safe learning. J. Intell. Rob. Syst.
Glover, F., Klingman, D., Phillips, N.V., 1992. Network Models in Optimization and Their Applications in Practice, vol. 36 John Wiley & Sons.
Hammitt, J.K., Cohen, J.T. Combining expert judgments: The classical and copula methods.
Herschel, Richard T., Nemati, Hamid, Steiger, David, 2001. Tacit to explicit knowledge conversion: knowledge exchange protocols. J. Knowledge Manage. 5 (1), 107–116. http://dx.doi.org/10.1108/13673270110384455.
Huntsberger, T., Stoica, A., 2010. Envisioning cognitive robots for future space exploration. In: Proceedings of SPIE, Orlando, FL.
Işıklar, G., Alptekin, E., Büyüközkan, G., 2007. Application of a hybrid intelligent decision support model in logistics outsourcing. Comput. Oper. Res. 34 (12), 3701–3714.
Jackel, L., Krotkov, E., Perschbacher, M., Pippine, J., Sullivan, C., 2006. The DARPA LAGR program: Goals, challenges, methodology, and phase I results". J. Field Robot. 23 (11–12), 945–973.
Johnson, E., Calise, N.A.J., Corban, J.E., 2000. Adaptive Guidance and Control for Autonomous Launch Vehicles. Georgia Institute 0f Technology & Guided Systems Technologies Inc.
Klingman, D., Phillips, N., Steiger, D., Wirth, R., Young, W., 1986. The challenges and success factors in implementing an integrated products planning system for Citgo. Interfaces 16 (3), 1–19.
Klingman, D., Phillips, N., Steiger, D., Wirth, R., Padman, R., Krishnan, R., 1987. An optimization based integrated short-term refined petroleum product planning system. Manage. Sci. 33 (7), 813–830.
Klingman, D., Padman, R., Phillips, N., 1988. Intelligent decision support systems: a unique application in the petroleum industry. Ann. Oper. Res. 12 (1), 277–283.
Klingman, D., Mote, J., Phillips, N.V., 1988. OR practice—a logistics planning system at WR Grace. Oper. Res.arch 36 (6), 811–822.
Koenig, S., Likhachev, M., Furcy, D., 2004. Lifelong planning A∗. Artif. Intell. 155 (1), 93–146.
Köksalan, M., Wallenius, J., Zionts, S., 2011. Multiple Criteria Decision Making: From Early History to the 21st Century. World Scientific, Singapore.
Kowalski, Z., Meler-Kapcia, M., Zieliński, S., Drewka, M., 2005. CBR methodology application in an expert system for aided design ship's engine room automation. Expert Syst. Appl. 29 (2), 256–263.
Kurd, Z., Kelly, T., Austin, J., 2007. Developing artificial neural networks for safety critical systems. Neural Comput. Appl. 16 (1), 11–19.
Laird, J.E., Newell, A., Rosenbloom, P.S., 1987. Soar: an architecture for general intelligence. Artif. Intell. 33 (3), 1–64.
Lehman, J., Laird, J., Rosenbloom, P., 2006. A gentle introduction to soar. An Architecture for Human Cognition.
Meyer, Mary A., Booker, Jane M. (Eds.), 1991. Eliciting and Analyzing Expert Judgment: A Practical Guide. Asa-Siam Series on Statistics and Applied Probability. American Statistical Association and the Society for Industrial and Applied Mathematics, London.
Mostaghimi, M., 2001. Information collection strategic design in experts-assisted decision making paradigm. Group Decis. Negot. 10 (5), 375–388.
Nabney, I., 2001. Validation of neural network medical systems. Workshop on regulatory issues in medical decision support, October 2001.
Pal, S.K., Dillon, T.S., Yeung, D.S. (Eds.), 2012. Soft Computing in Case Based Reasoning. Springer Science & Business Media.
Stanciu, C.O., 2009. Decision support systems architectures. ArXiv Preprint arXiv: 0906. 0863.
Storey, N., 1996. Safety Critical Computer Systems. Addison-Wesley, Harlow.
The Maritime and Coastguard Agency, 2008. International Code of Safety for High Speed Craft (2000), 2008 Edition. The Maritime and Coastguard Agency, London.
Trafton, J.G., Schultz, A.C., Cassimatis, N.L., Hiatt, L.M., Perzanowski, D.P., Brock, D.P., Bugajska, M., Adams, W., 2006. Communicating and collaborating with robotic agents. In: Sun, R. (Ed.), Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation. Dover, NY, pp. 252–278.
Varadaraju, R., 2011. A Survey of Introducing Artificial Intelligence Into the Safety Critical System Software Design Process.
Yoon, K.P., Hwang, C.L., 1995. Multiple Attribute Decision Making. Sage, Newbury Park, CA.