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