# Auditable Policy Execution in Autonomous and Semi-Autonomous C2 Systems to Support the Next Generation Warfighter

**Tom Keeley**
**President Compsim**

## Executive Summary:

Future Combat Systems and Information Operations continue to integrate more and more information from a continually expanding variety of sources across the echelons of control. Some of these systems will be manual, but the human's ability to comprehend and react to the escalating complexity demands that more automation be deployed. This means that the role of humans will eventually migrate from information analysis, to one where the focus is more on policy definition and performance auditing. This paper focuses on the inability of the human's verbal and written language to describe policies that can be effectively interpreted by automatic systems. It proposes a dynamic graphical source code language and execution environment that can 'explicitly' define and execute policies that describe how information is to be interpreted. By providing policies documented with this graphical language, automated equipment can interpret policies as a continuous data set rather than static rules. This means that automated systems or embedded devices can interpret information much like human experts. This approach effectively provides the added advantage that the automated systems and devices can be audited and policies can be extended, when appropriate, with relative ease. Using the graphic language to develop explicit policies allows them to be developed much faster than text based solutions, thus allowing faster deployment to tactical operations.

## Discussion:

### Decomposition of 'Expertise':

To effectively transfer experience or expertise from the human domain to the domain of intelligent systems and intelligent devices, one must first define expertise. Dreyfus described a five stage model of skill acquisition to describe the differences between individuals with different levels of competence[1]. In the first stage, the novice understands a series of rules to determine actions. The advanced beginner has applied the rules to real situations and can recognize some relevant cues on his/her own. With more experience, the advanced beginner is overwhelmed with the number of relevant cues and migrates to the level of competence where a plan is available to determine which cues are relevant and which can be ignored. Success at this level cannot be guaranteed. The next stage, proficiency, requires the performer to incorporate experience into the process, replacing all the rules and principles with situational discrimination and associated responses. Behavior shifts from reasoning to intuition. The highest level, expertise, is achieved when he/she intuitively sees what goals need to be accomplished, while at the same time intuitively see what actions should be taken to attain the goals. Enough expertise in a wide variety of situations allows he/she to form classes of situations that share the same decisions, actions, or tactics. In addition, even the highest level of expertise may lead to inappropriate/incorrect decisions when an intuitive versus rational decision process is employed.[2][3][4]

[1] Dreyfus H., *Intuitive, Deliberative, and Calculative Models of Expert Performance* in Zsambock, C. & Klein, G., *Naturalistic Decision Making*, Lawrence Erlbaum, 1997.
[2] Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty, Journal of Risk and Uncertainty, 5, 297–323.
[3] Kahneman, D., & Tversky, A. (Eds.). (2000). Choices, values, and frames. New York: Cambridge University Press.
[4] Kahneman, D. (2003). A perspective on judgment and choice: Mapping bounded rationality. American Psychologist. Vol. 58, No. 9, 697-720.

**The English Language Fails to Describe Expertise for Intelligent Systems:**

Because our intention is to transfer “expertise” from the human domain to the intelligent system domain, we need to have a way to document intuition. We suggest that the human verbal and written languages are ineffective in defining intuition. While human verbal / written languages (including conventional computer programming languages) are effective in describing lower level ‘rules’, they fail to effectively describe judgmental decisions. If we expect our next generation cognitive systems to accept more responsibility then we must “allow” them to make these judgmental decisions on their own, just like we expect our human experts to perform today. We suggest that judgmental decisions have more to do with how information is interpreted (valued) and how information items inter-relate. If there are fixed absolute rules, they just feed the entire process of balancing all of the available information to determine the course of action.

**Examples of Human Systems:**

Today we expect our fighter pilots to use their judgment when attacking a target. They evaluate the target value, the risks associated with an attack, potential collateral damage, the weapons at their disposal, an understanding of the performance of their plane, etc. They know they are flying an expensive piece of hardware and have a responsibility of returning it for future operations. Based on their training, they know what risks to take and how best to achieve their goals.

In a command and control application, a goal or series of goals may be defined. The commander will allocate resources and define schedules to achieve those goals. Lower levels in the echelons of control will decompose the higher level goals into goals under their command and allocate their own resources. This allocation of resources will include judgmental decisions about risks and rewards. Commanders at all levels are trained to evaluate the information available to them in order to determine the best course of action.

**Debriefing Human Systems:**

When humans make judgmental decisions in their pursuit of their assigned goals, they often go through a debriefing process where they describe what they did, how they did it, what worked and what didn’t, etc. This dialog is granular because of the nature of the human language. It does not describe “exactly” how each maneuver was made, it does not describe “exactly” why any decision was made by explicitly describing how each piece of information was interpreted and balanced while performing an action.

**Expectations for Intelligent Systems and Devices:**

While we train our human warfighters to perform tasks, we are often only able to measure performance based on a high level success rate. Because we are dealing with humans, we accept that they are human and accept that the only way of achieving better performance is to provide more training and more technology to assist them perform more effectively. When we transfer some of this decision making to intelligent systems, we do not have the same options. We demand much more from intelligent machines than we do from humans. The reason for this is that we “assume” that humans will not do something really wrong. At the same time we accept that they are somewhat subject to human failure. But because we know that every human is different, we are confident that all humans will not make the same major mistake.

Intelligent Systems, whether they are enterprise software applications, intelligent weapons, or intelligent sensors, have the potential of being mass produced. This means that their actions must be completely auditable and explainable. Because many of these systems will be making judgmental decisions or making recommendations about strategy or tactics, it is mandatory that a way is provided to explicitly define the policies that define their behavior. If we accept that the human language is ineffective in describing policies that define behavior under all circumstances, then an alternative must be provided.

**A Dynamic Graphical Language for Describing Policy:**

KEEL® Technology (Knowledge Enhanced Electronic Logic) incorporates a dynamic graphical language that shows information items as normalized relative values. Information items are inter-related by "wiring" them together. The graphical language is dynamic, in that the policy engine executes as soon as the graphical items are dropped on the screen. The designer gets immediate feedback while the policy is being defined. The designer can simulate external inputs and "see" the system think as it is being designed. The reasoning is "explicit" because the user can "see" how each piece of information is being valued and how each value interacts with any other decision or action. Unlike conventional programming languages that process information sequentially, KEEL Technology processes all of its inputs during a cognitive cycle, somewhat like COTS Programmable Controllers that read inputs at the beginning of a program scan and produce outputs at the end. Unlike the PLC example, however, KEEL Technology "balances" the information during the cognitive cycle, much like a human "balances" the impact of different information sources to determine a set of actions.

When integrated into an enterprise software application or device, the policy execution is deployed as a function or class with associated methods and a series of tables that define the policy. This collection of function and tables is called a KEEL Engine. Each KEEL Engine consumes a very small memory footprint (approximately 3K words depending on the output language) and can be deployed in any localized or distributed architectures. From a systems engineering standpoint, KEEL Engines can be deployed just like organizations deploy human decision makers. Each KEEL engine operates autonomously, while at the same time each can receive outside directives from above that advise them of new information or of new ways to interpret information.

**An Explicit Language:**

The KEEL Graphical Language only uses text to identify graphical items to the human designer. Data items appear as vertical bars, whose height indicates the importance of information at any point in time. When the information indicates an action (or output), the relative support is shown. Information items can drive external outputs to control a level of response or trigger events. Techniques to address time and space based problems are incorporated into the language. Individual data items can impact other decisions and actions, when they are "wired" together. By viewing the importance of information and the relative support for that information one can trace the impact of any piece of information throughout the system by tracing the wires and reviewing the support for various components. During the design phase, one can graph relationships in order to plan the impact of information as it adapts to change. Should there be a need to audit judgmental decisions, an aircraft "black box" approach can be used. Any or all actions can be recorded for off-line review.

For example, if an unmanned aerial vehicle returns from a mission where it made a decision NOT to attack a target, that decision can be evaluated. The auditor can "see" how the UAV interpreted the situation it saw based on the policy described with the KEEL graphical language. Should there be a need to change the policy, it can easily be changed.

**Hybrid Systems:**

A variety of solutions exist in the cognitive space, such as ACT-R, Soar, COGNET, JESS, OMAR, Micro Saint, Neural Networks, and Fuzzy Logic. The systems engineer has the responsibility of choosing the right set of technologies to fit the problem domain. KEEL Technology has a very simple API that allows it to be integrated with almost any system that has access to low level language subroutines or class definitions. In this manner it can be used to augment existing systems or be easily integrated into new systems composed of multiple technologies.

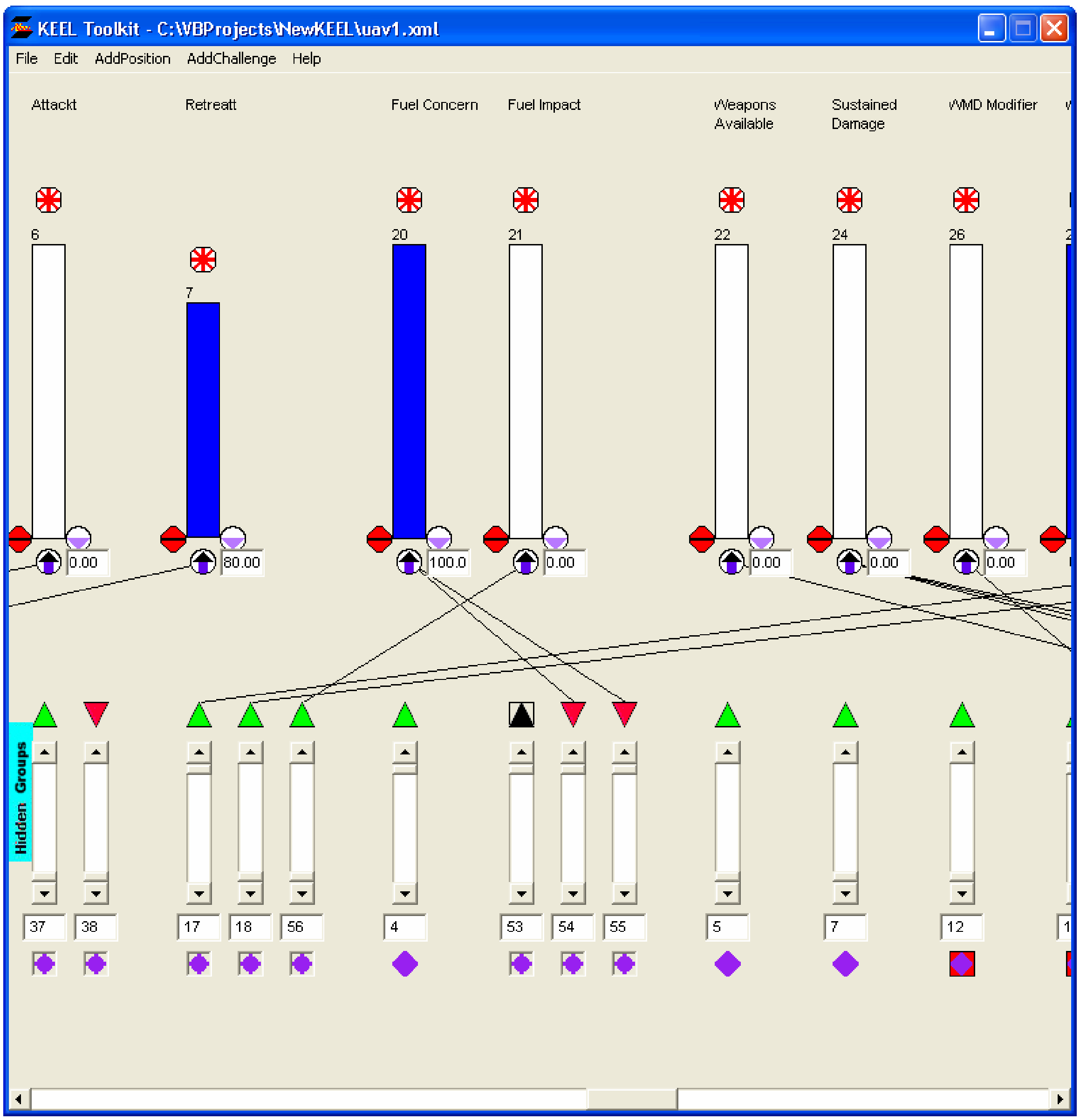
**Summary:**

KEEL Technology provides a new paradigm for explicitly documenting policies that can be executed by intelligent systems and be deployed in the smallest of form factors. The ability to rapidly deploy fully auditable systems will be a requirement as humans move to fill the administrative roles and allow intelligent devices to take on more responsibility in the tactical warfighter space.

**Contact:**

Thomas Keeley
President Compsim LLC
(262) 797-0418
tmkeeley@compsim.com
http://www.compsim.com

**Screen Capture Showing the KEEL Graphical Language**




This image shows a segment of the policy describing actions of an Unmanned Aerial Vehicle with a hunt and destroy objective. Inputs come from target information (value / distance), risk assessment, threat evaluation, fuel supply, weapon supply, damage assessment, external intelligence, etc.