

MODELING GROUND SOLDIER SITUATIONAL AWARENESS FOR CONSTRUCTIVE SIMULATION WITH RULES

Scott Neal Reilly
James Niehaus
Peter Weyhrauch

Charles River Analytics
625 Mount Auburn Street
Cambridge, MA 02138, USA

ABSTRACT

The behavior models that control simulated warfighters in most modeling and simulation (M&S) efforts are fairly simple, relying predominantly on behavior scripting and simple rules to produce actions. As a result, the simulated entities do not reflect critical situational awareness factors used by Ground Soldiers or allow for the modeling of devices that influence situational awareness, such as user defined operating pictures (UDOPs). This paper describes our approach to this challenge, providing 1) a rule-based method for modeling Ground Soldier situational awareness and devices that influence situational awareness and 2) a user friendly graphical authoring tool for creating these rules. We present a requirements analysis of this modeling task and discuss and provide examples of how our method may be employed for modeling Soldier perception and inferences as well as devices that affect situational awareness.

1 INTRODUCTION

Current military research centers, such as the U.S. Army's Natick Research, Development, and Engineering Center (NSRDEC), develop and apply modeling and simulation (M&S) technologies to evaluate new warfighter systems in terms of their ability to maximize force effectiveness and warfighting capabilities. However, the behavior models that control the warfighters in most M&S efforts are fairly simple, relying predominantly on behavior scripting and simple rules to produce actions. As a result, the simulated entities do not reflect critical situational awareness factors used by Ground Soldiers, nor do they allow for the modeling and assessment of devices that influence situational awareness, such as user defined operating pictures (UDOPs). These limitations degrade both the realism of the models and the applicability of conclusions drawn from the simulation-based studies.

For example, the Infantry Warrior Simulation (IWARS), a constructive simulation of Ground Soldiers, provides pre-defined transition paths between simulated behaviors but does not allow those transitions to happen based on the kinds of perceptions, inferences, and associations that human Ground Soldiers actually use when perceiving and understanding their situation. Thus, a simulated Soldier might be modeled to transition from patrolling to engagement upon seeing an enemy target, but could not be modeled to make that transition based on a more complex analysis of the nature, direction, and severity of the threat. Also, the simulated Soldier's situational awareness cannot be degraded by "fog of war" factors such as fatigue, harsh weather conditions, cognitive overload, or conflicting information—leading to a model of Soldier decision making and performance that may be unrealistic for a given mission. There is a clear need for computational models of the perceptions, inferences, and associations of Ground Soldiers that can be used by simulation environments to improve range, realism, and accuracy in analysis.

This paper describes our approach to this challenge, named Graphical Authoring Tool for Inference Rules (GATIR). GATIR provides 1) a rule-based method for modeling Ground Soldier situational awareness and tools that affect situational awareness and 2) a user friendly graphical authoring tool for creating rules for Ground Soldier perception, inference, and association and devices that influence situational awareness.

The rest of this paper is organized as follows. Section 2 presents our requirements analysis. Section 3 presents related work in situational awareness and perception, inference, and association modeling. Section 4 presents our rule-based approach to meeting the requirements, given the successes of previous systems. Under this approach, Section 4.1 discusses the Rete rule representation, Section 4.2 presents the process flow, Section 4.3 discusses rule-based modeling for perception, inference, and association, Section 4.4 discusses rule-based modeling for situational awareness devices, and Section 4.5 presents the accompanying graphical authoring tool. Section 5 presents conclusions.

2 REQUIREMENTS

Our first task was to analyze the task domain to identify modeling requirements. The goal of this analysis was to determine what features of the modeling infrastructure are needed to best support modeling of situational awareness and devices that influence situational awareness. First, we analyzed a number of military behavior sets to understand the kinds of perception, inference, and association skills that are needed to successfully execute them. We analyzed the following behaviors (Sutherland 2006): lift and shift base of fire, SPOT/SALUTE reporting, reporting commander's critical information requirements (CCIRs), call for fire, reacting to a near ambush, evaluating threats, passing information through a unit, and target assessment. The complete analysis of these behavior sets is available (Neal Reilly and Harper 2006), but the results of the analysis are reflected in the list of requirements below.

Second, manuals describing how Ground Soldiers are supposed to make decisions and act were consulted, as well as two projects that have attempted to understand how they actually do make decisions and act in simulated combat situations. There are multiple generally accepted views of situational awareness—such as Endsley's multi-stage model (Endsley et al. 2000), Boyd's OODA loop (Boyd 1987), and Miller and Shattuck's Dynamic Model of Situated Cognition (Miller and Shattuck 2004). Our modeling technique does not presuppose any one of these, but provides a general framework in which these views might be implemented. The manuals we analyzed include the *Army Field Manual (FM) 7-8: Infantry Rifle Platoon and Squad* (Department of the Army 1992), the *Marine Corps Warfighting Publication 3-35.3: Military Operations on Urbanized Terrain (MOUT)* (Department of the Navy 2006), and the *Army Field Manual (FM) 3-06: Urban Operations* (Department of the Army 2003). We also analyzed the results of the *Urban Canyon Street Fight and BLUFOR in Defense Data Collection Effort*, a live simulation at the McKenna MOUT village at Fort Benning, GA (Woods, Statkus, and Salvi 2006), and a knowledge-elicitation effort involving Ground Soldiers engaged in a close quarters combat room clearing operation (Greenwald 2002).

Third, we incorporated the kinds of perception, inference, and association skills supported by existing modeling tools as discussed in Section 3, including SAMPLE (Harper et al. 2000), Soar (Laird 1987), Jess (Friedman-Hill 2006b), Clips (Giarratano 1998), OPS5 (Forgy 1981), Hap (Loyall and Bates 1993), RAL (Forgy 1994), and Gertie (Loyall et al. 2004). Fourth, we reviewed current UDOP and information display technology to determine what sorts of processing is needed to model a device that is being used to support Ground Soldier situational awareness (SA). The final result of this analysis was a set of functional and design requirements that the perception, inference, and association modeling system should meet, described below.

Constant constraints. Constant constraints are tests of variables against constant values using one of the basic comparison types (=, <, >, <=, >=, <=, >=). For instance, when making an inference about the current effectiveness of opposing forces in a *Lift and Shift Base of Fire* behavior, the casualties are compared to 66% as an indication that opposing forces have suffered sufficient losses to continue moving.

Structured data. Perception, inference, and association skills do not often process single, isolated pieces of data, but structured records of data elements, including pointers among structured data records. For instance, the *Threat Value Determination* behavior set requires evaluating whether the Soldier being modeled is within range of a target's weapon. This requires a representation of a target, including whether the target is already incapacitated, the location and orientation of the target, and the target's weapon. The weapon may, in turn, represent the type of the weapon and where it is being aimed. These two structured data elements are used together to make inferences about the threat posed by this target.

Multiple basic types, new types, and enumerated types. The elements of the environment that are being reasoned about when using perception, inference, and association skills will be most readily represented by a variety of data types (e.g., distances as real-valued numbers, names as strings). The perception, inference, and association models that process these elements of the environment need to be capable of working with a variety of basic data types. In addition, some types are best represented as enumerated data. For instance, the types of weapons that can be reasoned about probably constitute a set of pre-defined (but extensible) types (e.g., AK47, M4). Finally, users will need to be able to create their own types of data. While the simulator will provide concrete sense data about target locations and weapon types, it will not have any built-in data types that correspond to higher-level, inferred entities, such as beliefs and threat assessments. Users must be able to create new enumerated data types and new structured data types.

Cross-condition constraints. Cross-condition constraints represent comparisons across multiple data structures. For instance, in the *Call for Fire* behavior set, one element of the inference about whether to call for fire or not is a comparison of the size of the target group as compared to the size of the friendly forces group. This requires that the size of the target group be compared not to a constant value (as a *constant condition* would) but to a changeable value in a field of the record representing the friendly group. Each of the objects being analyzed (the friendly and opposing groups) constitute conditions that must be satisfied for the inference to occur (i.e., is there an opposing group *and* is there a friendly group). This test of values associated with these two groups is, therefore, called a *cross-condition* constraint.

Conjuncts, disjuncts, and negation. In cross-condition constraints, we have already seen the usefulness of conjoining multiple tests of conditions that must be true for a perception, inference, and association skill to be applied. Disjuncts and negation are similarly powerful. Negation allows for the expression of *not*. For instance, in the *SPOT/SALUTE Report* behavior set, a detected entity is inferred to be a *newly encountered threat* if it is *not* co-located with any of the targets already on the target list. Disjuncts allow for the expression of *or*. For instance, in the *Lift and Shift Base of Fire* behavior set, the inference that it is time to *lift and shift* can occur when the assault force has reached a specific location *or* the assault force is within 75m of the area target. (Note that while *or* is not logically necessary, as it can be expressed through combinations of *and* and *not* or through the writing of multiple rules, it will make writing rules that correspond to human perception, inference, and association skills more natural and straightforward.) Conjuncts, disjuncts, and negation are generally considered extremely useful and powerful in the creation of inference systems, and are supported by all of the inference modeling systems we evaluated.

Temporal reasoning. Some inferences are not made about particular, instantaneous states of the environment, but are made about patterns of temporal events. Recognizing the tactic being employed by an opposing force will typically not be possible by analyzing a static snapshot of the environment, but will rather be possible only by watching the sequence of actions performed over time. Similarly, in the *Call for Fire* behavior set, time is used as an indicator that sufficient potential targets have been revealed and that a *call for fire* can proceed.

Complex functional relationships. Sometimes the test for when a perception, inference, and association skill should be applied is not a simple comparison of two values as is provided by *constant constraints* and *cross-condition constraints*. For instance, in an earlier example we described how a *lift and shift base of fire* should occur when the assault force is within 75m of the target area. Computing distance, however, while not especially complicated, is not a basic test. In this case, we want to know if:

$\sqrt{(\text{blufor}_x - \text{target}_x) * (\text{blufor}_x - \text{target}_x) + (\text{blufor}_y - \text{target}_y) * (\text{blufor}_y - \text{target}_y)} < 75$. Similar functional relationships will also be needed to model various perceptual skills, such as sound localization (Scharine and Letowski 2005). The perception, inference, and association modeling system will need to provide support for this kind of complexity, including the comparison of multiple variable values and the combination of those values with a variety of functions.

Representation of confidence and probabilistic knowledge. Sometimes the knowledge or perceptual data that is used by perception, inference, and association models is not reliable. Furthermore, sometimes the perception, inference, and association processes themselves are known to be good rules of thumb but are not perfectly accurate. For instance, if a Soldier sees a figure through some smoke that looks like an enemy (certainty=0.6) on a building roof and the soldier believes that enemies on roofs tend to be snipers (certainty=0.9), he might infer that he saw an enemy sniper (confidence=0.54). Given the uncertainty, however, he does not want to engage (in case it is not an enemy) and he does not want to expose himself (in case it is), so he might choose to take cover until the smoke clears.

Priority. Sometimes, despite multiple possible conclusions or assessments, one assessment is preferable. For example, if an unknown person can be identified as either a non-combatant or a possible enemy, then it may be preferable to identify the person as a possible enemy and exercise caution in approaching them. Similarly, a UDOP will sometimes want to present some information with a higher priority (such as by making it larger or brighter or bolder) to ensure it is seen first, and the models will need to support this form of reasoning.

Modality. Situational awareness and devices that support situational awareness often use multiple sensory modalities. For example, a UDOP can present information either visually or aurally. We need to support these channels of information both at the UDOP model level and at the Soldier perception level.

Scalability. An issue that can easily undermine a perception, inference, and association modeling system, especially one that provides the sort of expressive power laid out by the previous functional requirements, is that of efficiency. The modeling technology must take efficiency as a serious design constraint in order to enable sufficient scalability.

Ease of use. Somewhat at odds with the desire to provide an extremely rich and powerful perception, inference, and association representation scheme is the requirement that the perception, inference, and association modeling system be easy to use and understand. If the final result is extremely flexible and powerful but so unwieldy as to be difficult to use, the actual modeling advances provided will be minimal. Modelers have modeling expertise, but they should not be required to have expertise in the managing and execution of the implementation technology.

3 RELATED WORK

This section examines how current and past systems meet the requirements of Section 2.

3.1 Perception, Inference, and Association Modeling and Simulation

Inference-modeling systems have been around for many years. Early expert systems, such as MYCIN (Shortliffe 1976), have employed hand-encoded rule sets to draw conclusions about constrained domains. In the case of MYCIN, given patient medical information, the system can make a recommendation for antimicrobial therapy and present the rules it used to make the recommendation to the user. Due to this purpose, the rules that create these inferences are not constructed to imitate the cognition of the experts, but to produce the most correct answer to the query.

Since these early expert systems, a number of general-purpose rule-based computer languages have arisen, including as Jess (Friedman-Hill 2006b), OPS5 (Forgy 1981), Clips (Giarratano 1998), and the Rule Algorithmic Language (RAL) (Forgy 1994). Due to their availability and maturity, many of these systems have been used in artificial intelligence applications to encode the perception, inference, and decision logic of simulated agents. However, simply employing a rule-based language does not enable

modelers to effectively capture the relevant domain logic. We present in this paper a method for using these rule-based languages to model Ground Soldier situational awareness.

Modern human-behavior modeling architectures—such as SAMPLE (Harper et al. 2000), Soar (Laird 1996), and ACT-R (Lovett, Redar, and Lebiere 1999)—have also been used to model military decision making. While these architectures may be effective in the hands of a trained expert, modelers may be unfamiliar with the formalism or modeling methods and either unable to express their model such that it functions as expected within the architecture or unable to take advantage of the key features of such an architecture. Our approach aims to be understandable and easy to use by technical modelers with no other specific training.

3.2 Rule Engines

Executing rules based on the agent's sensory inputs, perceptions, and knowledge can be a costly process, but a number of optimized algorithms have been designed to make it feasible to create large knowledge bases and rulebases with acceptable computational costs. The best-known of these algorithms include Rete (Forgy 1982), Treat (Miranker 1990), Match Box (Perlin and Debaud 1989), and TREE (Bouaud 1993). Of the four, Rete provides the best combination of scalability to large systems and expressiveness of data structures it can work with (Doorenbos 1995). It is likely for this reason that Rete is the most commonly used, including being the matching component for Soar, Jess, Clips, OPS5, RAL, Hap, and Gertie. The Rete algorithm constructs an efficient directed acyclic graph to represent dependencies between working memory elements and rule preconditions and reasons over this graph when new information is processed. A complete description of the Rete algorithm is beyond the scope of the current paper (see Forgy 1982).

3.3 Interfaces for Rule-Based Systems

Because building complex rule-based systems is not a simple task, there have been a number of efforts to simplify the inference-rule creation process, through tools such as Herbal (Cohen, Ritter, and Haynes 2005), Visual Soar (2002), JessTab (a Jess plugin for the Protégé ontology-development tool) (Eriksson 2004), and the Jess plugin for Eclipse (a general-purpose, integrated development environment) (Friedman-Hill 2006a). All of these systems, however, are focused on making it simpler to create text-based inference rules, which still leaves them as being difficult to use by those who are not experts in rule-based systems, such as typical subject matter experts and even many traditional software engineers. None attempt to use drop-down menus and drag-and-drop functionality as a means of making the inference-rule development process simpler and less error-prone.

4 METHOD FOR RULE BASED MODELING OF GROUND SOLDIER PERCEPTIONS, INFERENCES, AND ASSOCIATIONS

Based upon the previous success of rule-based approaches, described in Section 3, in meeting the requirements presented in Section 2, we employ rule-based computing as our modeling paradigm. Because of its expressiveness, its efficiency, and its strong track record, we propose that RETE is a suitable choice for the proposed inference-rule-based perception, inference, and association modeling system. We show how this modeling technique can be applied to simulate Soldier perceptions, inferences, and associations (in Section 4.3) as well as to support design and development of situational awareness devices such as user defined operating pictures (UDOPs) (in Section 4.4). In addition to the rule engine and rule representation, the system features a graphical user interface to address ease of use, a phased rule engine to ease model construction, and a lightweight integration method for constructive simulation environments to hasten integration tasks.

4.1 Rete Rule Representation

Forward-chaining inference rules (also known as production rules) have left-hand side, a *precondition* or *antecedent*, and a right-hand side, also known as the *action*, *consequent*, *result*, or *effect*. The rules may *fire* whenever the conditions specified on the left-hand side are met and the result of the rule firing is specified by the right-hand side. The right-hand side can result in any of a number of actions, including external actions and internal changes to the simulated agent’s knowledge base. The rules are typically annotated with a *priority* that indicates the order in which rules should fire if multiple antecedents are simultaneously true. Figure 1 shows two example Rete rules in the Jess syntax (Friedman-Hill 2006b). The preconditions occur before the “=>” symbol, and the effects occur afterwards.

Rete-based systems provide all of the functionality described in Section 2, with the exception of *ease of use* and explicit support for *temporal reasoning*. Significantly more detail on the Rete algorithm and the various optimizations can be found in Forgy (1982) and Doorenbos (1995). Preconditions of rules can check for *constant constraints* in *structured data* containing *multiple basic types*, *new types*, and *enumerated types*. Preconditions in Rete also allow for *cross-condition constraints* to compare between data structures, *complex functional relationships*, and *conjuncts and negation*. *Disjuncts* are achieved by the implicit “or” between rules. *Temporal reasoning* and *representation of confidence and probabilistic knowledge* can be achieved through representation within the structured data. *Scalability* is a central feature of the Rete algorithm, as Rete is an optimization over earlier production systems (Forgy 1982). We address ease of use through the development of a graphical development tool, which is described in Section 4.5. We discuss implicit support for temporal reasoning and related future work in Section 5.

```
(defrule VulnerabilityThreatDetermination-OrientedTowardsMe
  (declare (salience 10))
  ?v <- (VulnerabilityThreatDetermination (computed FALSE))
  (Agent (force RED) (orientedTowardsMe TRUE) (confidence ?conf))
  =>
  (modify ?v (value 0.4) (computed TRUE) (confidence (* 0.5 ?conf))))

(defrule OverallThreatDetermination
  (VulnerabilityThreatDetermination (computed TRUE) (value ?vt)
                                     (confidence ?c_vt))
  (ClosingRateThreatDetermination (computed TRUE) (value ?crt)
                                   (confidence ?c_crt))
  ?ot <- (OverallThreatValueDetermination (computed FALSE))
  =>
  (modify ?ot (computed TRUE)
             (value (max 0.0
                       (min 1.0 (- 1.0 (* (- 1.0 ?vt) (- 1.0 ?crt))))))
             (confidence (min ?c_vt ?c_crt))))
```

Figure 1: Jess rules representing a portion of a threat determination inference model

4.2 Process Flow

Figure 2 shows the process flow for the integration of the Graphical Authoring Tool for Inference Rules (GATIR) perception, inference, and association with a simulation environment. Section 4.4 describes this process with the addition of a UDOP device model; for presentation purposes, we handle the simpler case first. As described in Section 1, the purpose of GATIR is to augment current Ground Soldier simulations with perception, inference, and association capabilities. These simulation environments often contain decision or action scripts that define how the simulated Ground Soldiers will behave given a set of concrete criteria. However, these simulation environments lack advanced techniques for transforming the ground truth—the exact values of complete information such as locations, headings, and types of equipment—of the simulation into a situational awareness for each Ground Soldier. GATIR performs this task by integrating with the simulation environment and processing the ground truth through perception, inference, and association rules.

At each simulation event, the ground truth information is transmitted to each GATIR agent. The perception rules delete, modify, or add information to the ground truth to create percepts. The percepts are passed to the inference and association rules. The inference and association rules transform the limited perception of the agent into situational awareness, what the agent believes to be true about the environment.

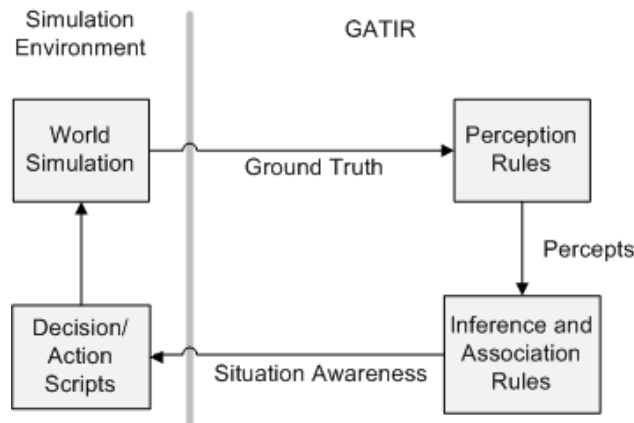


Figure 2: GATIR and simulation environment process flow

For example, while on patrol, a simulated Soldier encounters a civilian walking across the road with a camera in his hands in a crowded marketplace. The civilian’s position, heading, and information about the camera are passed as ground truth to the Soldier’s perception rules (as well as complete information about the rest of the marketplace). The perception rules determine that the Soldier is too distracted by the marketplace to immediately identify the camera and that the civilian is too far away to be identified by recognition. The perception rules create a percept of an unknown person with an unidentified object. The inference and association rules identify this percept as a possible threat to the Soldier and the marketplace (through a rule that raises threat values for unknown persons handling unidentified objects in crowded locations), and this situational awareness is passed to the Action Scripts. The Action Scripts define that the Soldier should investigate the threats above a specified threshold, and the Soldier chooses the action to detain the civilian and investigate the unidentified object. This action is carried out by the world simulation, and the loop repeats. As the Soldier approaches the civilian, his percepts become more clear; he is able to identify the individual and the camera and lower the perceived threat level. The Action Scripts choose to forego the detainment upon further inspection.

4.3 Perception, Inference, and Association Rules

Our basic perception, inference, and association model consists of two corresponding tiers of rules, perception rules and inference and association rules. The rules do not interleave firings; ground truth data is input to the perception rules, which fire to quiescence, then the resulting percept data is transferred to the inference and association rules, which fire to quiescence. The resulting situational awareness data is passed back to the simulation.

The purpose of the perception rules is to prevent the simulated Soldier from being omniscient or overly keen in its senses and cognition, and thus unrealistically effective during simulated missions. For instance, the perception rules can limit visibility to line of sight, make speech and sound information degrade over distance or in the presence of other sound, and lower sensing ability due to time of day (e.g., reduced vision at dusk) or weather conditions (e.g., reduced hearing in heavy wind). Perception rules are also critical in simulation environments with high fidelity models of sensing. Even with high quality sensor input, a Soldier may not be able to integrate all of the sensorial information into a clear cognitive picture of his surrounds. The Soldier may be focused intently on a small number of aspects in his scene,

following a particular individual in a marketplace, for instance, and ignore other information. The Soldier may also be experiencing cognitive overload, unable to process all of the information coming to his senses. For example, if two individuals are talking at the same time, even though the Soldier may be able to hear each one clearly, he may not be able to understand both simultaneously. Lastly, percepts may persist, allowing perception or inference rules to combine data and reconcile perceived inconsistencies over time.

The purpose of inference and association rules is to form the Soldier’s situational awareness from the percepts he is receiving. If the Soldier perceives a non-friendly individual in a defensive position carrying a weapon, he may infer that the individual is an enemy and, furthermore, that there are additional enemies in the area. It is important to note that inference and association rules do not simply rebuild the information altered by the perception rules. They construct a possibly erroneous situational awareness from the percepts the simulated Soldier receives. In the previous example, the non-friendly individual carrying a weapon could be a non-combatant with a radio, misperceived due to the Soldier’s current expectations.

For example, Figure 1 shows rules representing a portion of a threat determination inference model. The first rule computes an aspect of the threat value related to the simulated Soldier’s vulnerability—if the other agent is oriented towards the Soldier. The overall threat determination combines the vulnerability determination with a closing rate determination (not defined here).

4.4 Situational Awareness Device Rules

Figure 3 shows an example rule configuration for modeling the effects of a user defined operating picture (UDOP) device on Ground Soldier situational awareness. A UDOP device plays sounds and picture alerts in response to information streaming live over a wireless network. For example, if a Soldier sees a new enemy unit in the area, the Soldier can enter the enemy’s position into his UDOP device, the information is streamed over the wireless network, and other friendly Soldiers will receive alerts about the new enemy unit. The UDOP profile defines the conditions that cause alerts, and this profile can be configured by the Ground Soldier before or during a mission. An example M&S task for situational awareness devices is to determine the effects of a particular UDOP profile on mission performance.

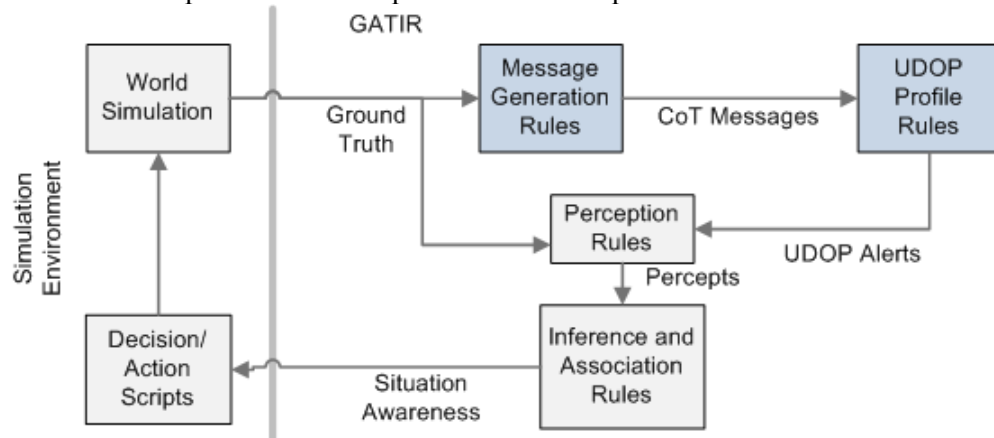


Figure 3: Rule sets for modeling UDOP devices

In the model of Figure 3, the ground truth is passed both to the UDOP message generation rules and the perception rules. The message generation rules generate UDOP traffic on the simulated network, and the UDOP profile rules determine which of this information causes UDOP alerts. The alerts are sent along with the simulation ground truth to the perception rules, where the alerts may interact with other stimuli to form percepts. As an example of this interaction, a sound nearby may prevent the Soldier from hearing a UDOP audio alert or a UDOP audio alert may prevent the Soldier from hearing the sound. These new

percepts are passed to the inference rules to form the Soldier's situational awareness that will affect mission performance.

4.5 Graphical Authoring Tool

To improve the ability for modelers to use GATIR as a modeling method, the GATIR system features a user friendly graphical authoring tool. Though a useful library of common perception, inference, and association rules may be defined by experts, it is impossible to encode all possible perceptual skills and inferences that Ground Soldiers use. However, given the ability to create rules as needed, modelers can provide the types of perception, inference, and association capabilities that are most relevant to their simulation context of interest. Thus, our approach is to create a tool that can be used to create additional perception, inference, and association rules as needed. Based on our initial analysis of the requirements in Section 2, we believe that the two most important features for such a tool are that it provides sufficient richness of expression and that it is easy to use. Firstly, we address richness of expression using inference rules (also called production rules or if-then rules) (Forgy 1982), a powerful and proven technology that is well-suited to the expression of perception, inference, and association models for Ground Soldiers.

Secondly, the perception, inference, and association modeling functionality must be easy to use. Efforts to develop full-scale human-behavior models (HBMs) have resulted in enormously complex agents. Soar's fighter-pilot model, TAC-AIR Soar, consists of over 8,000 rules (Jones et al. 1999) and required 20 man-years to build (Pearson and Laird 2004). The knowledge engineering (KE) necessary to support such a system requires a correspondingly large-scale effort. However, one way to ensure that new perception, inference, and association modeling functionality is maximally usable is to create easy-to-use development tools that support the modeling of complex human inferences. Due to the expertise required by perception, inference, and association modeling, it is unlikely that Ground Soldiers will use a perception, inference, and association modeling tool directly. Our goal is to create a tool for Soldiers to work with knowledge engineers to create new perception, inference, and association models, and to make this process simpler than is possible with current text-based development tools.

Figure 4 shows the GATIR graphical user interface for creating rules, defining data structures for working memory elements, and defining which information is reported to the simulation as the agent's situational awareness. The rule tab shown in Figure 4 allows the modeler to view the complete list of rules in the project in a collapsed list and expand individual rules for further examination. Each rule has an "if" part defining the preconditions, a "then" part defining the effects, a rule control defining the priority of firing (if multiple rules are able to fire at the same time, the priority can be used to choose between them), and documentation defining "What it is," "How it works," and "How you use it."

Figure 5 shows a close-up view of a condition being edited within the interface. The modeler can select the data type of the working memory element of interest, and assign this value a name, "?other", in this case. Next, the conditions can be defined that constrain what working memory elements can be chosen and when the rule fires. In this portion of the rule, the Agent's ID value must equal the "?proc_agentid" value defined elsewhere in the preconditions. Also, the "LocationX" and "LocationY" values for this agent are given names with the "bind-to" operator so that they may be referenced in the other preconditions or in the effects. The drop-down box shows some of the comparisons that can be used in the conditions: equal, not equal, less than, less than or equal, greater than, and greater than or equal. Full expressions containing multiple variables and a library of functions, such as square root, can be defined with an included expression editor (not pictured).

The graphical user interface enforces the syntax and semantics of the preconditions and effects to reduce modeler error and ease the modeling process. As in Figure 5, drop-down boxes are provided where applicable to allow the modeler to select from a limited range of options and avoid invalid input. Where the options are not limited as in the case of general expressions, the interface checks naming conventions syntax and return type to reduce the number of errors due to typos or other input mistakes.

5 CONCLUSIONS AND FUTURE WORK

This paper presents a rule-based method for modeling Ground Soldier situational awareness and devices that influence situational awareness. Our approach provides 1) a rule-based method for modeling Ground Soldier situational awareness and tools that affect situational awareness, and 2) a user friendly graphical authoring tool for creating rules for Ground Soldier perception, inference, and association and devices that influence situational awareness. Future work in this area may investigate the modeling of temporal information, advanced models of uncertainty, improved user interface, and targeted analysis tools. Other future work may combine this modeling of Soldier inferences and associations with other estimates of perception such as mathematical representation of sensor operations.

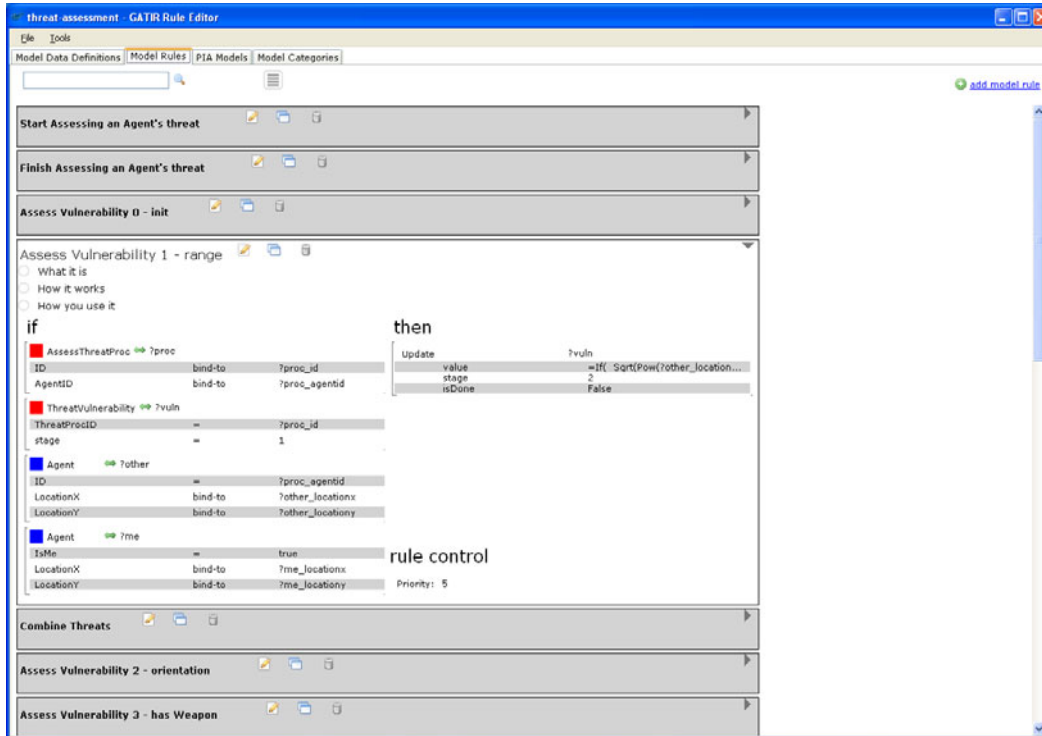


Figure 4: GATIR graphical user interface

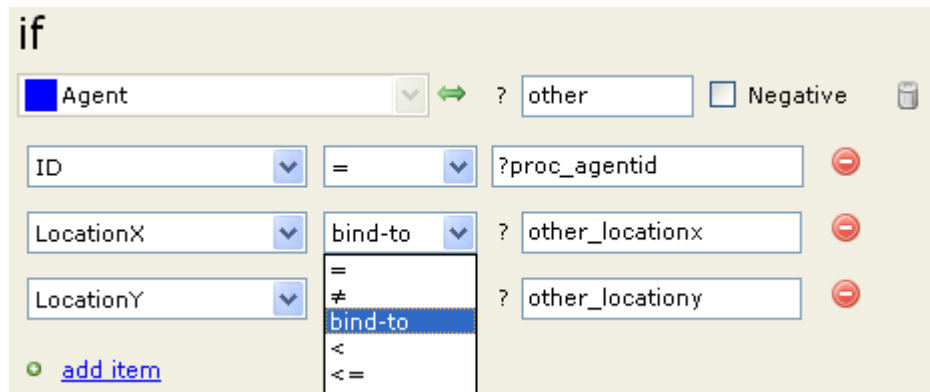


Figure 5: Editing a rule condition

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Army RDECOM Contracting Center, Natick Contracting Division, Natick, MA, under Contracts No. W911QY-09-C-0023 and W911QY-07-C-0096. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the above named agency.

REFERENCES

Visual Soar. Available via

<http://www.eecs.umich.edu/~soar/sitemaker/projects/visualsoar/>
[accessed September 10th, 2010].

Bouaud, J. 1993. TREE: The heuristic driven join strategy of a RETE-like matcher. In *Proceedings of Thirteenth International Joint Conference on Artificial Intelligence*.

Boyd, J. R. 1987. *A discourse on winning and losing*. Briefing slides available at Air University Library, Maxwell AFB.

Cohen, M., F. Ritter, F., and S. Haynes. 2005. Herbal: A high-level language and development environment for developing cognitive models in Soar. In *Proceedings of 14th Conference on Behavior Representation in Modeling and Simulation*.

Department of the Army 2003. *Army Field Manual (FM) 3-06: urban operations*. Washington, DC: Headquarters, Department of the Army.

Department of the Army 1992. *Field Manual No. 7-8 (FM 7-8): infantry rifle platoon and squad*. Washington, DC: Headquarters, Department of the Army.

Department of the Navy 2006. *Marine corps warfighting publication (MCWP) 3-35.3: military operations on urbanized terrain (MOUT)*. Washington, DC: Headquarters United States Marine Corps.

Doorenbos, R. B. 1995. *Production matching for large learning systems*. Ph.D. Thesis. Carnegie Mellon University, Pittsburgh, PA.

Endsley, M. R., L.D. Holder, B.C. Liebrecht, D.J. Garland, R. Wampler, and M.D. Matthews. 2000. *Modeling and measuring situation awareness in the infantry operational environment*. Research Report 1753. Alexandria, VA:U.S. Army Research Institute for the Behavioral and Social Sciences.

Eriksson, H. 2004. *JessTab manual: integration of Protege and Jess*. Linkoping University.

Forgy, C. 1994. RAL/C and RAL/C++: rule-based extensions to C and C++. In *Proceedings of Proceedings of the OOPSLA'94 Workshop on Embedded Production Systems*.

Forgy, C. 1981. *OPS5 user's manual*. Rep. No. CMU-CS-81-135. Carnegie Mellon University.

Forgy, G. L. 1982. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence* 19.

Friedman-Hill, E. *Jess 7.0 Manual*. Available via

<http://www.jessrules.com/jess/docs/70/> [accessed September 10th, 2010].

Friedman-Hill, E. *Jess, The rule engine for the Java platform*. Available via

<http://www.jessrules.com/> [accessed September 10th 2010].

Giarratano, J. C. 1998. *CLIPS user's guide*, Version 6.10.

Greenwald, T. W. 2002. *An analysis of auditory cues for inclusion in a virtual close quarters combat room clearing operation*. M.S. Thesis. Naval Postgraduate School, Monterey, CA.

Harper, K. A., S.S. Mulgund, G.L. Zacharias, and T. Menke. 2000. SAMPLE: situation awareness model for pilot-in-the-loop evaluation. In *Proceedings of 9th Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL:

Jones, R., J. Laird, P.E. Nielsen, K. Coulter, P. Kenny, and F. Koss. 1999. Automated intelligent pilots for combat flight simulation. *AI Magazine* 20:27-42.

Laird, J. 1996. *Soar/IFOR: intelligent synthetic agents*. Washington DC: NRC DMSO HBR Panel.

- Laird, J. E. 1987. Soar: an architecture for general intelligence. *Artificial Intelligence* 33(1):1-64.
- Lovett, M. C., L.M. Redar, and C. Lebiere. 1999. Modeling working memory in a unified architecture: an ACT-R perspective. In *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, ed. A. Miyake & P. Shah. New York: Cambridge University Press.
- Loyall, A. B., and J. Bates. 1993. Real-time control of animated broad agents. In *Proceedings of Fifteenth Annual Conference of the Cognitive Science Society*.
- Loyall, A. B., W.S. Neal Reilly, J. Bates, and P. Weyhrauch. 2004. System for authoring highly interactive, personality-rich interactive characters. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.
- Miller, N. L., and L.G. Shattuck. 2004. A process model of situated cognition in military command and control. In *Proceedings of the 2004 Command and Control Research and Technology Symposium*. San Diego, CA.
- Miranker, D. P. 1990. *TREAT: A new and efficient match algorithm for ai production systems*. San Mateo, CA: Morgan Kaufmann.
- Neal Reilly, W. S. and K. Harper. 2006. *Perceptions, inferences, and associations model for infantry soldiers (PIAMIS)*. Cambridge, MA: Charles River Analytics.
- Pearson, D. and J. Laird. 2004. Redux: example-driven diagrammatic tools for rapid knowledge acquisition. In *Proceedings of Behavior Representation in Modeling and Simulation*. Washington, DC.
- Perlin, M. W., and J.-M. Debaud. 1989. Match box: fine-grained parallelism at the match level. In *Proceedings of IEEE International Workshop on Tools for Artificial Intelligence*.
- Scharine, A. A., and T.R. Letowski. 2005. *Factors affecting auditory localization and situational awareness in the urban battlefield*. Rep. No. ARL-TR-3474. Aberdeen Proving Grounds, MD: U.S. Army Human Research and Engineering Directorate, ARL.
- Shortliffe, E. H. 1976. *Computer-based medical consultations: MYCIN*. New York, NY: American Elsevier.
- Sutherland, D. M. 2006. *Modeling the impact of C4ISR on small-unit command and control*. Rep. No. Natick/TR-06/020/L. U.S. Army RDEC, Natick Soldier Center.
- Woods, R., M.J. Statkus, and L. Salvi. 2006. *Human science/modeling and analysis data project: urban canyon street fight and BLUFOR in defense data collection effort, 21 March Through 1 April 2005*. Rep. No. NATICK/TR-06/009L. U.S. Army Research, Development and Engineering Command, Natick Soldier Center.

AUTHOR BIOGRAPHIES

SCOTT NEAL REILLY is a Principal Scientist at Charles River Analytics. His primary research interest is in building intelligent agents in dynamic, realtime environments. This has led him to work in areas such as realtime agent architectures, intelligent behavior execution, abductive/diagnostic reasoning, belief networks, embedded conversational systems, plan recognition, sensitivity analysis, and hybrid agent architectures. Before joining Charles River Analytics, Dr. Neal Reilly worked as Vice President of Product Development and Senior Scientist at Zoesis, Inc. He has a B.S. in Computer Science from Duke University and a Ph.D. in Computer Science from Carnegie Mellon University. His email address is [<snealreilly@cra.com>](mailto:snealreilly@cra.com).

JAMES NIEHAUS is a Scientist at Charles River Analytics. His primary research interests include interactive narrative, serious games, intelligent user interfaces, and cognitive modeling. His past work has included projects in cognitive models of discourse comprehension, instructional games for learning, and planning approaches to narrative generation. Before joining Charles River Analytics, he was a Postdoctoral Fellow at the Georgia Institute of Technology. Dr. Niehaus has a B.S. in Computer Science

from the College of Charleston and a Ph.D. in Computer Science from North Carolina State University. His email address is [<jniehaus@cra.com>](mailto:jniehaus@cra.com).

PETER WEYHRAUCH is a Principal Scientist at Charles River Analytics. His primary research interests include computational narrative technology simulation-based training, and artificial intelligence. He has worked in areas such as real-time interactive narrative generation and execution, interactive narrative and agent-based training and education, agent architectures, speech processing, and physics and world simulation. Before joining Charles River Analytics, Dr. Weyhrauch worked as Senior Scientist and VP of Product Innovation at Zoesis, Inc. He has a B.S. in Computer Science from the Massachusetts Institute of Technology and an M.S. and a Ph.D. in Computer Science from Carnegie Mellon University. His email address is [<pweyhrauch@cra.com>](mailto:pweyhrauch@cra.com).