Improved Situation Interpretation Metrics in Context-Based Reasoning Simulations

Roberto C. Sanchez, Avelino J. Gonzalez
Intelligent Systems Lab
University of Central Florida
4000 Central Florida Blvd
Orlando, FL 32816-2450
sanchezr@ieee.org, gonzalez@ucf.edu

Abstract

The current implementation of situation interpretation metrics (SIMs) in context-based reasoning (CxBR) simulations makes it difficult to effectively and efficiently characterize situations in the simulation. While previous work in the field has dealt to a certain extent with SIMs, the primary focus of such work has been in other areas. Thus, while the current implementation of SIMs is functional, there is much room for improvement. This paper describes a series of changes and improvements to the current implementation of SIMs that allow a more structured and modular approach to characterizing situations. By creating a more clearly defined structure for the SIMs, it is possible to separate the SIMs that are pertinent to the simulation at large and all the agents from the SIMs that are pertinent only to individual agents. This clear separation allows agents to set up a dependency tree between their own local SIMs and the global SIMs. By being aware of the dependencies between the SIMs, the agents can make more efficient decisions. Because the situation is only reevaluated when necessitated by a change in a particular SIM, only the affected bits of knowledge represented by the changed SIMs are reevaluated. The implementation of this structure is in progress, thus experimental results are not yet available. But, it is believed that this approach will provide a more efficient means of situation evaluation and decision making in CxBR simulations.

Introduction

Over the past several years context-based reasoning (CxBR) has developed from simple scripts layered over a rule-based system (Gonzalez & Ahlers 1993) to a full-fledged framework (Norlander 1999) capable of supporting a complete simulation environment (Gonzalez, Grejs, & Gonzalez 2000), resolving ambiguous situations (Saeki & Gonzalez 2000b), and allowing very fine-grained control of the simulation environment (Gallagher, Gonzalez, & DeMara 2000). Gonzalez and Ahlers have also shown that the CxBR approach is particularly well-suited to the representation of human tactical behavior (Gonzalez & Ahlers 1998), allowing for the development of simulations that maintain a high level of fidelity because the intelligent agents in the simulation behave as a human is expected to in the same situation.

While the CxBR framework is quite powerful in its current form, the implementation of situation interpretation metrics (SIMs) provides only a limited facility for agents to accurately characterize situations. SIMs are handled exclusively by the simulation-at-large and are only accessed by the individual agents when a specific course of action must be decided on. For example, in a tank simulation, the tank commander reaches a point where he encounters two enemy tanks. He must make a decision by first evaluating the SIMs to characterize the situation and then evaluating his goals and deciding which course of action will most likely lead to accomplishment of his goals. While this model is functional, and provides an excellent representation of human tactical behavior, major improvements are certainly possible.

The best way to remedy the limitations of the current SIM implementation is to restructure the implementation of SIMs to allow for separate global and local SIMs. Global SIMs are better characterized as *quantitative*. They represent facts, and are generally accessible to all agents within the simulation. Local SIMs are better characterized as *qualitative*. They represent intermediate evaluations of the situation as they are made by individual agents, and are generally only accessible to the local agent. Of course, the mechanism is more complex than this simplistic explanation. This article will explain, in detail, a proposed method for implementing a system of separate global and local SIMs.

Overview of CxBR

Assumptions of CxBR

According to (Gonzalez & Ahlers 1998), CxBR is based on the following assumptions:

- "Tactical experts are proficient in their tasks by recognizing and treating only the key features of the situation."
- "There are only a limited number of things that can realistically take place in any situation."
- "The emergence of a new situation will generally require alteration of the present course of action."
- "Tactical knowledge is quite general in nature."

Essentially, CxBR assumes that the expert is, in fact, an expert and that he knows what he is doing with respect to his area of expertise. Thus, an auto mechanic would be expected to troubleshoot a car engine but know little or nothing of performing brain surgery. While someone's level of proficiency is a rather subjective matter, it is reasonable to expect that the individual would not be considered an expert if he or she lacked the fundamental skills to effectively deal with situations which encompass the area of knowledge. The expert is also expected to separate necessary and unnecessary information as part of the decision-making process. For example, when troubleshooting a car engine, the fact that the upholstery is cracked and fading is largely irrelevant—an expert is expected to identify such information as irrelevant.

While it is possible to surmise that an unlimited number of things can occur in any period of time, it is not realistic. This is not to say each situation is limited to a very small number of happenings, but rather, that the situation dictates what should be expected. An auto mechanic troubleshooting an engine would expect only a small number of things to happen, all of which would be at his discretion, such as connecting the engine to a diagnostic computer, starting the engine, and checking the various parts. Likewise, a military commander leading a tank platoon in an engagement against an enemy tank platoon would expect occurrences from a limited range of possibilities. Thus, the situation itself tells the expert what to expect, and the expert is expected to glean this information upon entering the situation.

An expert typically takes a particular course of action based on the current situation and the available information. Had the expert entered into a different situation, it is entirely possible that the expert would have elected a different course of action. So, when the situation itself changes into a different situation or when a different situation comes into play, it is expected that the expert will change his course of action to reflect his evaluation of the new situation.

General Description of CxBR

The CxBR paradigm represents human tactical behavior by defining a hierarchy of contexts that can direct the AIP's actions. Essentially, CxBR is effective and efficient because it uses the currently selected context to exclude irrelevant knowledge from the decision-making process (Gonzalez & Ahlers 1998). A good illustration of excluding irrelevant knowledge is that of an automobile driver on the highway—the driver does not need to consider knowledge related to traffic lights, only those things which can conceivably occur on the freeway need be considered.

The way that CxBR is composed, each agent is controlled through its own set of contexts (Gonzalez & Ahlers 1998). At the highest level, the mission context defines the goals, constraints, and other parameters relevant to the agent. The agent is controlled through a major context and the various sub-contexts. The mission context contains several major contexts that the agent can assume to perform various tasks. For example, a tank may have the following main contexts available: patrol, attack, defend, retreat. The major contexts are exclusive of each other, so that a tank cannot attack and retreat at the same time. Beneath each major context are several sub-contexts, which act as helpers to the major context. The sub-contexts may or may not be exclusive, such as when an attacking tank is maneuvering to avoid enemy fire and reloading the main gun at the same time.

By controlling when and how each agent transitions into and out of each context, it is possible for the agent to behave just as an expert would in the same situation. This system would be ideal if all context transitions were known at development time and if they were small in number. However, as Saeki and Gonzalez (2000a) observed, in tactical situations all the valid context transitions cannot normally be identified in advance and, if they could all be identified, there are far too many to realistically hard code them. To overcome this obstacle Saeki and Gonzalez developed (Saeki & Gonzalez 2000b) and implemented (Saeki & Gonzalez 2000a) the competing context concept. [For a more complete discussion of CxBR please see (Gonzalez & Ahlers 1998), (Norlander 1999), (Gonzalez, Grejs, & Gonzalez 2000), (Saeki & Gonzalez 2000b), and (Gallagher 2000)]

Current Implementation of SIMs

Currently, the CxBR framework implements SIMs in a sort of large pool, as part of the simulation's global fact base. Each agent is allowed access which corresponds to what is visible to the agent. For example, a tank would be allowed access to the SIMs that detail the number, status and location of two nearby enemy tanks. Whereas the simulation would mask the number, status and location of two enemy tanks behind a mountain five miles away. This masking is performed on an agent-by-agent basis. Still, the agent must rely on general simulation for all of its SIMs each time a decision is to be made (Norlander, 1999) and (Saeki & Gonzalez, 2000a).

Situation interpretation metrics provide a way for the agent to evaluate the current situation against its current goals. The SIMs can be anything relevant to the simulation environment and are only limited in number by the available memory. Some examples of SIMs in a military simulation are: current state of the agent, number and location of friendly units, number and location of enemy units, weather, and terrain features.

Current Limitations of SIMs

The common thread among the examples of SIMs is that they are all quantifiable. This is one of the limitations of the current implementation, because whenever a decision is made by an agent, it must inventory the available SIMs and its local fact base. Only then can a decision be made. One way to overcome this limitation, as stated in the introduction, is to also allow the individual agents to maintain their own qualitative SIMs. Thus, an agent can make qualitative evaluations only when the relevant SIMs change, instead of each time a decision must be made.

Improvements to the Implementation of SIMs

The limitations of the current implementation of situation interpretation metrics notwithstanding, it is a proven method. The current CxBR framework allows the individual agents to access the global fact base, a local fact base, and the various SIMs. However, the fact that SIMs are only available globally, and are maintained by the simulation environment, causes some difficulties when implementing the reasoning of the individual agents. The difficulties arise because the individual agent lacks the ability to track the situation on a continual basis. Currently, the situation is reevaluated when a context transition becomes available (Saeki & Gonzalez 2000b), although many times a context transition coincides with a change in SIMs. This is the case regardless of whether the SIMs have changed or whether the context transition occurs at a time when none of the SIMs have changed. The result is that there are times when the situation is reevaluate without necessity, and there are other times when a reevaluation of the situation would be warranted but not take place.

The solution to overcome this limitation requires several major changes to the way SIMs are implemented in CxBR simulations. The changes can be summarized in a short list:

- Institute global SIMs (*quantitative*) and local SIMs (*qualitative*)
- Create and reserve space during simulation initialization in a global *SIM table* for each *quantitative* SIM
- Create a dependency tree (for each agent during initialization) between the various local SIMs and the global SIMs
- Update the global SIMs during each simulation cycle [this is already part of the current implementation, but bears mention here because of changes]
- Check the *changed* state of the global SIMs during each simulation cycle and update any local SIMs whose parent SIMs have changed

Describe Global and Local SIMs

To improve the implementation, the SIMs must be divided into two categories. Global SIMs are maintained and updated by the simulation environment. The global SIMs are quantitative and objective in nature, such as weather conditions, terrain features, number and location of agents, and status of agents (damage sustained, crew status, fuel, and ammunition). The key criterion for identifying a SIM as global is that no matter from whose perspective, the value of the SIM will not change. For example, whether the terrain is viewed by friendly or enemy units, the location of the terrain features on the battlefield will not change.

The local SIMs are particular to each individual agent in the simulation, and are qualitative and subjective in nature. Some examples of qualitative SIMs are battlefield conditions (favorable or unfavorable) and tactical advantage (disadvantaged or advantaged). Unlike the global SIMs, local SIMs are identified by their change in value depending on the perspective from which they are viewed. For example, a lone tank that encounters an enemy ambush would identify its situation as being at a tactical disadvantage, whereas the enemy tanks would perceive their situation as being a tactical advantage.

Generate Global SIM Table

The organization of the global SIMs is best accomplished in a table. This allows the simulation environment to easily organize the SIMs in a way that provides the most flexibility. Each SIM is labeled and associated with the appropriate parameters. For example, *status_of_agent_1* corresponds to the overall state of the first agent in the simulation. This overall state is composed of several other SIMs, including *operational_state*, which for this example is *immobilized*. When the simulation cycle completes, the simulation environment notifies any other agents that depend on the information in the changed SIM (this is explained in more detail in the following section).

Some SIMs are always present, such as weather and terrain. Other SIMs appear and disappear as the agents enter and leave the simulation, such as the various SIMs pertaining to individual agents. Clearly, a dynamic implementation of the global SIMs is called for. This allows the SIM table to grow and shrink as necessary, while remaining uncluttered. While not specifically beneficial in small scale simulations, such as those on which this implementation will be tested, it provides a way to scale up the size of the simulation without changes to the way the SIMs are implemented. This implementation also makes a network-aware simulation a simple extension.

Register Local SIM Dependencies

The local SIMs are particular to each individual agent, so that each agent takes the following steps at instantiation:

- Generate local SIM table, similar to global SIM table
- Create a dependency tree between local and global SIMs
- Register dependencies with the simulation environment

The local SIM table is similar to the global SIM table described above. The key difference between the global SIM table and the local SIM table maintained by each agent, is that the local SIM table is static. This is because once the table of local SIMs has been defined by an agent, the presence of the SIMs does not change. For example, if an agent defines its local SIMs as *tactical_advantage* and *battlefield_condition*, then the existence of those SIMs will not change. The values may change, such as the agent reevaluating *tactical_advantage* and revising its value from *advantaged* to *disadvantaged* after encountering an ambush of 4 enemy tanks, but their existence does not. So, once the agent specifies that *tactical_advantage* is one of its SIMs, it will be a SIM for the agent throughout the simulation. Thus, the agent's reasoning and decision making are based an unchanging set of parameters. The local SIM table also allows the agent to establish which global SIMs each local SIM depends on.

The agent determines, based on the local SIMs and the available global SIMs, the required dependencies by establishing a relationship between the various bits of knowledge. For example, the agent has a local SIM called *tactical_advantage* which identifies whether the agent evaluates the situation to currently be in its favor or against. The *tactical_advantage* local SIM is based on several pieces of information, one being the *status_of_wingman* global SIM. Thus, when the agent is instantiated, it will determine that this dependency (along with many others) exists.

The dependency is then registered with the simulation environment. Then, whenever the *status_of_wingman* global

SIM changes the simulation environment informs the agent, and the agent reevaluates the corresponding local SIM (this is explained in further detail in following sections).

This SIM dependency structure allows two agents to have very different concepts of the same SIM. As in the example above, one agent bases its evaluation of the *tactical_advantage* on the *status_of_wingman*. Another agent may base the evaluation of *tactical_advantage* on *ambient_light_condition* because, for example, it has night vision equipment available.

Update Global SIMs

At the end of each simulation cycle, the table of global SIMs is updated. Each SIM that changes has zero or more dependent local SIMs that correspond to the various agents. The simulation environment then informs each agent that has registered a dependency that the SIM has changed, allowing only those agents with an interest in that particular bit of information the opportunity to reevaluate the situation.

Currently, the CxBR framework performs a similar action. Because the SIMs must be updated each simulation cycle, the simulation environment takes account of such things as positions and actions taken by the agents. But no specific effort is made to notify the agents of which SIMs have changed. In the new implementation, the agents have constructed dependency trees identifying which local SIMs depend on which global SIMs. Thus the simulation environment, which already must check each individual global SIM, is able to pass a message to the agent about which SIMs of interest have changed.

This eliminates the need for each agent to check each and every global SIM after each simulation cycle completes. Instead, each agent receives a message from the simulation environment. This approach reduces the number of accesses to memory, or across the network in the case of a network distributed simulation.

Update Local SIMs

As explained in the previous section, the updating of a local SIM only occurs when a global SIMs on which it depends has been modified. So, the agent awaits the arrival of message from the simulation environment which notifies it of a change in a global SIM. The agent then identifies which of its local SIMs depend upon the knowledge represented by the changed global SIM, and reevaluates the local SIMs as necessary.

By updating the local SIMs, even at times when a specific decision does not have to be made, the agent can always be ready to make a decision. Rather than waiting to reach a decision and then evaluating or reevaluating its SIMs, this method provides a cleaner execution. This method also reduces some of the computational burden associated with the decision making, since decisions are being made incrementally at the end of each simulation cycle, rather than all at once. This has the effect of reducing spikes in processor usage and distributing it more evenly across the simulation.

Current Status of Research

As of this writing, the implementation of the new SIM structure is in progress. It is too early to tell to what extent this will improve simulation execution and realism, but we are reasonably certain that measurable gains will be realized.

References

Gallagher, A.; Gonzalez, A. J.; and DeMara, R. F. 2000. Modeling platform behaviors under degraded states using context-based reasoning. In *Proceedings of the Interservice / Industry Training Systems and Education Conference*.

Gallagher, A. 2000. Modeling platform behaviors under degraded states using context-based reasoning. Master's thesis, University of Central Florida, Orlando, FL.

Gonzalez, A. J., and Ahlers, R. H. 1993. Concise representation of autonomous intelligent platform in a simulation through the use of scripts. In *Proceedings of the Florida Artificial Intelligence Research Society Conference*.

Gonzalez, A. J., and Ahlers, R. H. 1998. Context-based representation of intelligent behavior in training simulations. In *Proceedings of the Naval Air Warfare Center Training Systems Division Conference*.

Gonzalez, F. G.; Grejs, P.; and Gonzalez, A. J. 2000. Autonomous automobile behaviour through context-based reasoning. In *Proceedings of the Florida Artificial Intelligence Research Society Conference*.

Norlander, L. 1999. A framework for efficient implementation of context-based reasoning in intelligent simulations. Master's thesis, University of Central Florida, Orlando, FL.

Saeki, S., and Gonzalez, A. J. 2000a. Competing context concept: Experimental results. In *Proceedings of the Interservice / Industry Training Systems and Education Conference*.

Saeki, S., and Gonzalez, A. J. 2000b. Soft-coding the transitions between contexts in cgf's: The competing context concept. In *Proceedings of the Computer Generated Forces and Behavior Representation Conference*.