# Building Explainable Artificial Intelligence Systems

**Mark G. Core, H. Chad Lane, Michael van Lent, Dave Gomboc, Steve Solomon and Milton Rosenberg**
The Institute for Creative Technologies, The University of Southern California
13274 Fiji Way, Marina del Rey, CA 90292 USA
core,lane,vanlent,gomboc,solomon,rosenberg@ict.usc.edu

## Abstract

As artificial intelligence (AI) systems and behavior models in military simulations become increasingly complex, it has been difficult for users to understand the activities of computer-controlled entities. Prototype explanation systems have been added to simulators, but designers have not heeded the lessons learned from work in explaining expert system behavior. These new explanation systems are not modular and not portable; they are tied to a particular AI system. In this paper, we present a modular and generic architecture for explaining the behavior of simulated entities. We describe its application to the Virtual Humans, a simulation designed to teach soft skills such as negotiation and cultural awareness.

## Introduction

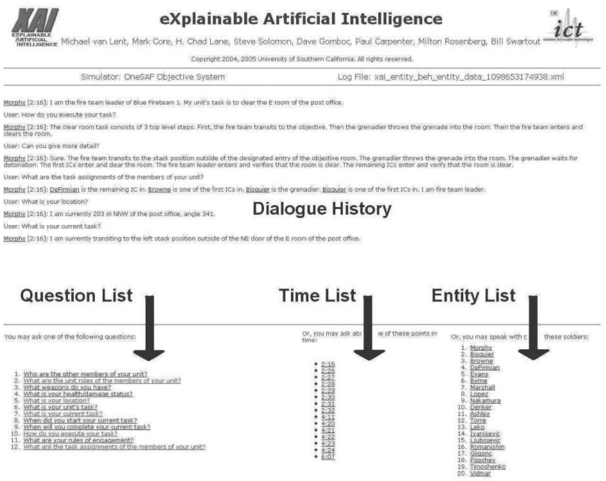
The complexity of artificial intelligence (AI) systems in simulations and games has made it difficult for users to understand the activities of computer-controlled entities. In simulations and games used for training, the explainability of behaviors is essential. Students must understand the rationale behind actions of simulated entities and how their actions affect the simulated entities.

The military has traditionally used live exercises for training; the principal tool for learning from these exercises has been the after-action review (AAR). US Army Field Manual 25-101, "Battle Focused Training", gives recommendations on conducting an AAR, and states that "The OPFOR [opposing forces] can provide valuable feedback on the training based on observations from their perspectives...the OPFOR can provide healthy insights on OPFOR doctrine and plans, the unit's action, OPFOR reactions to what the unit did." (Army 1990)[Appendix G] OPFOR are often played by instructors and participate in the AAR giving their viewpoints not only as opponents in the exercise but also as teachers assessing trainee performance. Friendly forces also participate in the AAR, and provide insight on how trainee orders translate into the behavior of units and individual soldiers. Despite the importance of AARs, military simulators currently do not provide this type of experience. Such simulators can present mission statistics (e.g., casualties, tactical objectives achieved) and replay events from the simulation, but neglect the causes behind these statistics and simulation events. What is missing in the AAR are the simulated entities (both OPFOR and friendly forces) to present their viewpoint on what happened and why, and an AAR leader to ensure students are asking the right questions and learning how to improve their performance.

Military simulations are also used as predictive tools; as such, their AI systems must be validated as behaving realistically and according to military doctrine. Detailed specifications are drafted for such AI systems; the resulting behaviors are put under heavy scrutiny. In most cases, because observers have no way to question AI-controlled entities, the observer's only recourse is to watch numerous simulation runs, looking for cases where faulty reasoning results in an incorrect action. For example, to detect an error where a rifleman entity fails to check its rules of engagement before firing its weapon, the observer must watch simulation runs until a situation occurs where the rifleman fires its weapon but its rules of engagement forbid this action. A better approach would be to question the entity about a single weapon fire event, and see whether checking the rules of engagement is part of its reasoning.

Figure 1 shows a screenshot of the user interface of our explanation system for the One Semi-Automated Forces Objective System, a military simulation (Courtemanche & Wittman 2002), and introduces the concept of an explanation system for simulated entities. Users select a time point to discuss, an entity to be questioned, and the question itself. Some of the questions are specific to the particular entity (e.g., what is your health?) while others concern a larger group (e.g., what is your unit's task?).

Two of the first explanation systems for simulation were Debrief (Johnson 1994) and Explainable Artificial Intelligence for Full Spectrum Command, FSC (van Lent, Fisher, & Mancuso 2004). In Debrief, Johnson built explanation capabilities into Soar (Laird, Newell, & Rosenbloom 1987) agents that were part of battlefield simulations. van Lent et al. added explanation capabilities to a training aid developed by commercial game developers and academic researchers. van Lent et al. coined the term "explainable artificial intelligence" (XAI) to describe the ability of their system to summarize the events of the game/simulation, flag key events, and explain the behavior of computer controlled entities.



eXplainable Artificial Intelligence

Michael van Lent, Mark Core, H. Chad Lane, Steve Solomon, Dave Gomboc, Paul Carpenter, Milton Rosenberg, Bill Swartout

Copyright 2004, 2005 University of Southern California. All rights reserved.

Simulator: OneSAF Objective System

Log File: xai_entity_beh_entity_data_1098653174938.xml

Morphy [2:16]: I am the fire team leader of Blue Fireteam 1. My unit's task is to clear the E room of the post office.

User: How do you execute your task?

Morphy [2:16]: The clear room task consists of 3 top level steps. First, the fire team transits to the objective. Then the grenadier throws the grenade into the room. Then the fire team enters and clears the room.

User: Can you give more detail?

Morphy [2:16]: Sure. The fire team transits to the stack position outside of the designated entry of the objective room. The grenadier throws the grenade into the room. The grenadier waits for detonation. The first ICs enter and clear the room. The fire team leader enters and verifies that the room is clear. The remaining ICs enter and verify that the room is clear.

User: What are the task assignments of the members of your unit?

Morphy [2:16]: DeFirmian is the remaining IC in. Browne is one of the first ICs in. Bisquier is the grenadier. Bisquier is one of the first ICs in. I am fire team leader.

User: What is your location?

Morphy [2:16]: I am currently 203 m NNW of the post office, angle 341.

User: What is your current task?

Morphy [2:16]: I am currently transiting to the left stack position outside of the NE door of the E room of the post office.

You may ask one of the following questions:

1. Who are the other members of your unit?
2. What are the unit roles of the members of your unit?
3. What weapons do you have?
4. What is your health/damage status?
5. What is your location?
6. What is your unit's task?
7. What is your current task?
8. When did you start your current task?
9. When will you complete your current task?
10. How do you execute your task?
11. What are your rules of engagement?
12. What are the task assignments of the members of your unit?

Or, you may ask ab e of these points in time:

- 2:16
- 2:26
- 2:27
- 2:28
- 2:29
- 2:30
- 2:31
- 2:32
- 4:11
- 4:20
- 4:21
- 4:22
- 4:23
- 4:24
- 6:07

Or, you may speak with c these soldiers:

1. Morphy
2. Bisquier
3. Browne
4. DeFirmian
5. Evans
6. Byrne
7. Marshall
8. Lopez
9. Nakamura
10. Denker
11. Ashley
12. Torre
13. Leko
14. Ivanisevic
15. Ljubojevic
16. Romanishin
17. Gligoric
18. Popchev
19. Timoshenko
20. Vidmar

Figure 1: Interface to XAI for OOS

While novel, this research did not address how lessons learned from explaining expert system decisions (Swartout & Moore 1993) applied to explaining the behavior of simulated entities. Swartout and Moore advocated building a high-level knowledge base containing facts about the domain and problem-solving strategies, and using an automatic program writer to build an expert system from this specification. The problem is more complicated for XAI because of technical challenges (i.e., the executable code must interface with an external simulation) and real world constraints (i.e., we have little or no control of the simulated entities - they have already been built or designed).

In the follow-on project to XAI for FSC, we faced these challenges but sought a middle ground between building another simulation-specific explanation system and demanding to build the simulated entities ourself (as Swartout and Moore built their own expert systems). Following this philosophy, we re-engineered our system to support:

- domain independence - supporting reuse and the development of common behavior representations while being flexible enough to deal with simulation-specific idiosyncrasies
- the ability to explain the motivations behind entity actions
- modularity - allowing external components such as GUIs, natural language generators, and tutors to interface with the system

In the related work section, we discuss how the Johnson and van Lent et al. work falls short in addressing these goals; we then present our XAI architecture designed specifically to meet these needs. Our first XAI system using this new architecture works with the One Semi-Automated Forces Objective System (OOS), a tactical military simulator and is described in the papers, (Gomboc *et al.* 2005; Core *et al.* 2005).

The focus of this paper is our second instantiation of the new architecture, the Virtual Humans simulator (Traum *et al.* 2005). This was not designed as a tactical simulation, but rather for the teaching of soft skills such as leadership, teamwork, negotiation and cultural awareness. The current scenario has one character, a virtual doctor who communicates with students through spoken input/output as well as generating animated gestures. The student plays the role of a U.S. Army Captain whose unit is planning an operation against insurgents in the same neighborhood as the clinic run by the doctor. In order to minimize the risk to civilians, the Captain is ordered to move the clinic to a safer location. The doctor is not interested in moving; the Captain's task is to convince the doctor to move the clinic.

Although the core structure of our architecture has not changed since we built XAI for OOS, we are continually making improvements such as the replacement of hard coded questions with logical forms. In this paper, we present an abbreviated case study describing the process of connecting this architecture to the Virtual Humans simulator. It illustrates the steps necessary to connect a simulation to an external explanation system as well as giving an idea of what parts of the system are reusable and what must be authored. We then present a sample dialogue with interactions between a student, automated tutor, and the XAI system. We do not focus on the details of this tutor, but include it in this presentation to illustrate the usefulness of XAI in a pedagogical context. In the future work section, we discuss taking features specific to XAI for OOS and XAI for Virtual Humans and making them domain-independent functions that can be reused in new target simulations.

## Related Work

In this section, we discuss two pieces of previous work on generating explanations of entity actions during a simulation: Debrief (Johnson 1994) and explainable artificial intelligence (XAI) for Full Spectrum Command (FSC) (van Lent, Fisher, & Mancuso 2004). Debrief works with entities controlled by Soar (Laird, Newell, & Rosenbloom 1987) agents in a tactical air combat domain. FSC was a collaboration between game developers and academic researchers who developed their own artificial intelligence system to control simulated entities in this training tool for commanding a light infantry company.

Debrief uses Soar's learning mechanism to save the agents' states during the simulation. After the simulation, an agent gives a textual description of what happened and the user is allowed to ask questions. To answer these questions, Debrief performs reflection by rerunning the agent and selectively changing elements of its state to determine the cause of its actions (including actions such as inserting a belief into memory).

FSC has logging facilities that save each entity's state: simple values such as ammo status and a representation of the task being executed (including its subtasks and how the platoon is broken into squads). After the simulation is over, FSC presents mission statistics and flags "interesting" time points (an entity is wounded or killed, the first contact with the enemy, task start and end times). The user selects a time and entity and is able to ask questions about the entity's state.

Both systems were important first steps in adding explanation capabilities to simulated entities, but neither approach was suitable for our goals of domain independence and modularity. Debrief and XAI for FSC are specific to the AI systems controlling the simulated entities in those applications, and not directly applicable to other AI systems.

(Gomboc *et al.* 2005) was the first description of our domain-independent XAI architecture; it described our efforts to connect this architecture to the tactical simulations, the One Semi-Automated Forces Objective System, OOS and Full Spectrum Command. We noted that simulations differ as to their "explanation-friendliness", and one of the key issues is how the simulation represents behaviors. Simulations may encode behaviors directly in a programming language, use planning operators, use a declarative rule format, or combinations of the three. We argued against a purely procedural representation because it provides no representation of the motivations underlying the behavior. Consider the example of sending a fire team to clear a room. Once the fire team is in position outside the room, the grenadier throws a grenade before the team enters the room. This could be encoded as a procedure (the step before entering the room is always throwing the grenade) in which case, the system cannot explain why the grenade was thrown. In the next section, we summarize the XAI architecture presented in (Gomboc *et al.* 2005) as well as discussing what options are available when dealing with an explanation-unfriendly simulation.

## XAI Architecture

Figure 2 shows our domain-independent XAI architecture. The (Gomboc *et al.* 2005) paper focused on the left side of diagram and the problem of importing data from the simulation. We use a relational database to store this information because tactical military simulations typically produce large log files; they record data at a fine granularity (e.g., OOS records the values of variables about once a second). The Virtual Humans currently log a much smaller amount of data, but the full power of the relational database may be useful later if the logging in Virtual Humans becomes more fine-grained. We focus on a subset of the virtual human's behaviors: if we covered the entire range, then we would see log files of comparable size as those in tactical simulations.

Assuming XAI is able to import the necessary information from the simulation, the next step is to inspect the data looking for "interesting" facts. The definition of interesting will change with each application. For OOS, we used a placeholder definition that highlighted when an entity fired its weapon, or when it began, was half-way through, or completed a task. Collaboration with domain experts will help us refine this definition. Because we developed an automated tutor for the Virtual Humans, our definition of interesting referred to events that the tutor should discuss with the student and it is the responsibility of the tutor to identify them. Although we currently hand-annotate these teaching points, we are automating the process using a heuristic-based approach to identifying teachable moments in an exercise.

Once the XAI system is initialized, users select an entity with which to speak and the time point in the simulation they wish to discuss first. Users then query the entity about the current time point by selecting questions from a menu. The dialogue manager orchestrates the system's response; first using the reasoner to retrieve the relevant information, then producing English responses using the natural language generator (NLG). NLG fills slots in natural language templates with information from the database. NLG is coded in XSL templates and takes advantage of XSL features such as iteration and procedure calls (common tasks such as state descriptions are shared among templates).

As noted in (Gomboc *et al.* 2005), this is a best-case scenario where the simulation makes available a rich behavior representation containing, for example, entity goals and action preconditions and effects. However, this does not mean that if the simulation does not have such a representation we cannot attempt to explain entity behavior.

Previous work in explaining expert system behavior (summarized in (Swartout & Moore 1993)) dealt with a similar problem; if the expert system's inference engine contains special features (not represented in the system's knowledge base) then the output of these features can still be explained by hard coded explanation routines. However, if changes are made to the special features without also updating the explanation routines, there is a potential for inaccurate explanations. Thus, explanations can still be made but at a cost to the maintainability and robustness of the system.

In the case of representing simulation behaviors for XAI, there are three options with associated costs and benefits:

1. automatically import the behaviors. *cost:* requires a representation of goals and the preconditions and effects of behaviors. *benefit:* high maintainability and robustness. *target:* plan-based representations
2. semi-automatically import the behaviors. *cost:* must be able to find goals, preconditions, and effects in behavior representation. *benefit:* more maintainable and robust than option 3, and makes fewer assumptions than option 1. *target:* rule-based representations
3. hand-build the XAI representation of the behaviors. *cost:* low maintainability - any change in the behavior must be duplicated in the XAI representation. Need subject matter expert to author missing goals, preconditions, and effects. *benefit:* makes no assumptions about the simulation's behavior representation. *target:* procedural representations

The reason that we target rule-based representations with option 2 is that some elements on the left hand side of rules are preconditions (e.g., you must have ammunition to fire your weapon), but other elements may be abort conditions (e.g., do not fire your weapon when a friendly entity is in the path) or internal bookkeeping (e.g., setting internal variables, making sure a rule does not fire twice). Similarly, not all elements on the right hand side of rules are effects and may instead also be internal bookkeeping. With such a representation, we can hand-annotate the preconditions and effects in these rules, then automatically import the behavior. Although there is no guarantee that annotations will be updated as developers change entity behaviors, at least this meta-data is co-located with the original behavior representation.

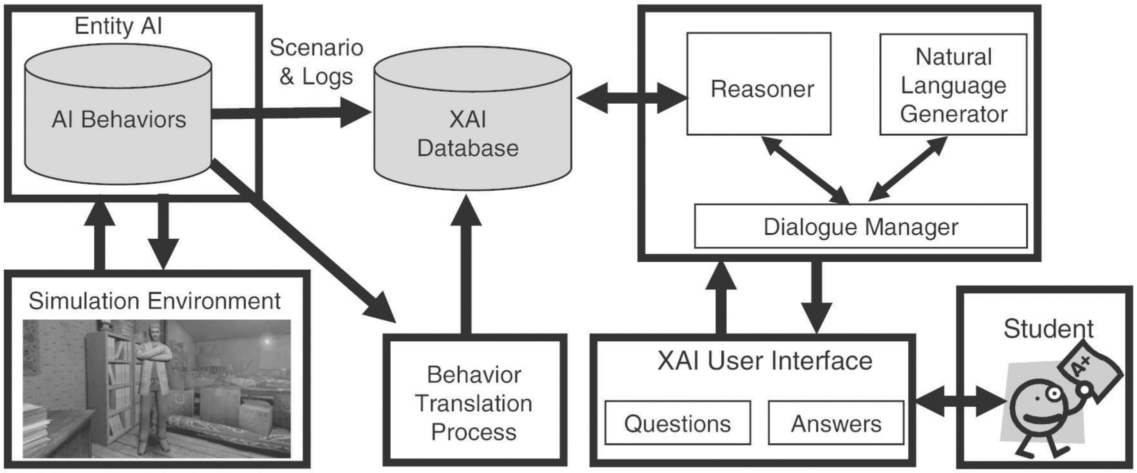


Figure 2: XAI Architecture

Collecting log files and scenario information is not trivial, but here the options are simple: if the simulation does not make information available for export, the XAI system cannot answer questions about it. In our XAI for OOS system, we entered some scenario and log information by hand to build a proof-of-concept system, but this data is specific to a particular simulation run, and some of the hand data-entry must be repeated for each new simulation run. It is more feasible to hand-author behavior representations because these do not change between simulation runs.

## Building a new XAI System

Connecting our current XAI system to a new simulation requires several steps:

1. study the behavior representation and choose one of the three approaches to importing behaviors as discussed in the previous section.
2. implement data import for behaviors and log files
3. specify the question list for the domain
   - 3a. write the logical form (LF) of each question
   - 3b. write the query LF
4. augment the natural language generator to support the new questions and their potential answers
5. create GUI

Specifying the question list for a new simulation requires two steps. The first step is writing the logical form of the question which is used to generate the English form of the question. For the Virtual Humans, we had 110 distinct questions so by using the natural language generator to produce the questions we could change how they were phrased without rewriting all 110 questions. The second step is writing the query to retrieve the answer from the database; we use an abstract language called the query logical form to encode queries (see below for more details).

The last step in connecting XAI to a new simulation is building a new GUI or reusing a GUI from a previous XAI system. Although every XAI system will have the same basic GUI components (ways to select entities, times, and questions, and displays of dialogue between user, XAI, and tutor), to support replay of the simulation requires support from the target simulation, and if XAI is a feature integrated into the simulation, it will share the simulation's GUI. Because of these constraints, we designed the GUIs of XAI for OOS and XAI for Virtual Humans as separate components that communicate with the rest of the system through XML messages. Our abstract message format facilitates this play-and-plug functionality. The messages convey the content of menus such as the question list and list of time points as well as user selections from these menus. The messages also update the state of the dialogue between the student and tutor and the dialogue between the student and XAI. The GUI can display these menu choices and text in whatever widgets (e.g., radio buttons, drop-down menus) it chooses.

## XAI for Virtual Humans

Following the steps enumerated in the previous section, the first task in connecting XAI to the Virtual Humans was to study the behavior representation. In this case, the simulation developers not only had to model physical behaviors such as treating patients but also the behaviors underlying the utterances produced by the student and doctor (e.g., committing, insisting), and the doctor's mental reasoning (e.g., making the decision to help the Captain). The model of physical actions contained preconditions and effects explaining the relationships between the actions (e.g., you need supplies to treat the patients). In importing this model we found some bugs in the model, so it is more accurate to say that we semi-automatically imported the physical behaviors. Now that the bugs are fixed, we should be able to fully automate this process.

The non-physical behaviors were implemented with hundreds of rules developed in the Soar cognitive architecture (Laird, Newell, & Rosenbloom 1987). Given enough time, it should be possible to hand-annotate the goals, precondi-

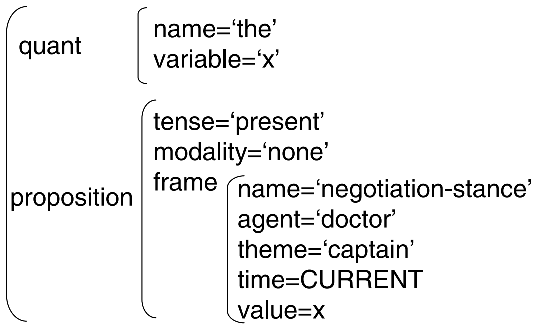


Figure 3: LF for "What is your negotiation stance?"

tions and effects in all these rules. As an initial step, our implementation focused on the rules governing trust since teaching trust building is one of the pedagogical goals of the 2005 Virtual Humans system. As described in (Traum *et al.* 2005), the Virtual Humans model trust as influenced by three factors: familiarity, credibility, and solidarity. All three have direct positive relationships to trust; the more the doctor feels he knows you, feels you speak the truth, and feels that you share common goals, the more he trusts you (and vice versa). In our current prototype, we model single steps of the doctor's reasoning by linking rules to English paraphrases (e.g., "the negotiation failed because you lost the trust of the doctor").

Once we designed the database format for the target simulation and wrote the code to import the data, the next step involved encoding the questions to be answered by XAI in the target domain (i.e., specifying the question itself, encoding the relevant database queries, and necessary changes to the natural language generation templates). Questions are encoded in a logical form (LF), an abstract representation of their content. A simplified graphical version of our XML representation for the question, "What is your negotiation stance?" is shown in figure 3. The logical form was designed to support future plans to generate syntactic features of the character's language such as tense and modality rather than hard coding them in templates. The other feature to note is the variable, CURRENT which is substituted at runtime with the current line being discussed. It is obvious that we would not author separate questions such as "what is the negotiation stance at line 1" and "what is the negotiation stance at line 2". However, this same mechanism also allows us to have one logical form for the questions, "why did the negotiation fail?" and "why are you avoiding the negotiation?". Here, the runtime variable is the negotiation stance.

The logical form of the question is accompanied by an abstract representation of the query (we call it the query LF) to retrieve the answer. It also uses runtime variables so that authors only have to write one query LF for the questions "why did the negotiation fail?" and "why are you avoiding the negotiation?". The XAI reasoner translates the query LF into the SQL query which is sent to the database. An area of future work is to derive the query LF automatically from the LF of the question.

The next step is modifying the set of XSL templates so that the English form of the question can be generated as well as the range of potential answers. Templates can be reused to support new questions and their potential answers. For example, there is one set of templates that generates English descriptions of states and tasks. These are used to describe states and tasks in questions as well as answers. In future work, we intend to make our natural language generation more domain-independent by hard coding less English, and adding templates encoding domain-independent aspects of language such as syntax and morphology.

## Sample Dialogue with XAI for Virtual Humans

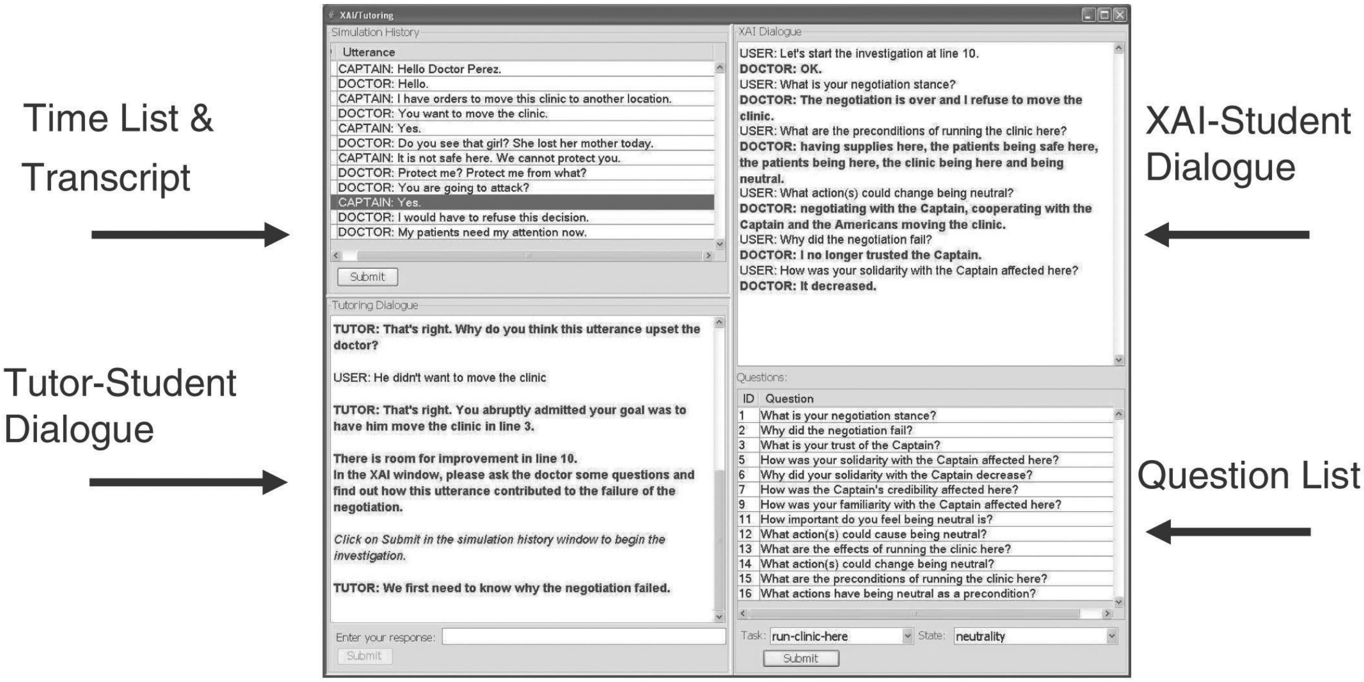
As mentioned in the introduction, after action reviews (AARs) are the Army's primary tool for learning from live training exercises, and our goal is building AAR experiences for simulated exercises. Our AAR GUI for the Virtual Humans is shown in figure 4. We currently lack a replay of the simulation and instead show a transcript of the interaction between the student and the virtual doctor that took place during the simulation (upper left of figure 4). Currently our prototype works with the following sample dialogue (C=Captain and D=Doctor):

C: Hello Doctor Perez
D: Hello
C: I have orders to move this clinic to another location
D: You want to move the clinic
C: Yes
D: Do you see that girl? She lost her mother today.
C: It is not safe here. We cannot protect you.
D: Protect me? Protect me from what?
D: You are going to attack?
C: Yes
D: I would have to refuse this decision
D: My patients need my attention now

The dialogue between the student and tutor appears in the lower left of the AAR screen. In this session, the tutor asks the student to use XAI to investigate what happened at line 10. The student then asks questions of the doctor using the question list on the bottom right of the screen, and the dialogue between student and doctor appears in the upper right of the screen. This is a prototype interface and we are currently experimenting with merging the two dialogue windows, adding playback capabilities, and developing a more usable question list.

Before the AAR begins, the tutor must analyze the dialogue searching for teaching points. Intuitively we can see that the dialogue is unsuccessful. In line 10, the student revealed secret information, and triggered the doctor's ending of the conversation. Examining the log files showing the doctor's mental state confirms that the dialogue was unsuccessful and that line 5 (where the Captain asked the doctor to move the clinic) and line 10 decreased trust. Our knowledge of the rules governing the doctor helps us see opportunities the student missed to gain trust in lines 3 and 10.

After loading the log files and behavior data into our XAI system, we hand-annotate these teaching points. Because



| Utterance |
|---|
| CAPTAIN: Hello Doctor Perez. |
| DOCTOR: Hello. |
| CAPTAIN: I have orders to move this clinic to another location. |
| DOCTOR: You want to move the clinic. |
| CAPTAIN: Yes. |
| DOCTOR: Do you see that girl? She lost her mother today. |
| CAPTAIN: It is not safe here. We cannot protect you. |
| DOCTOR: Protect me? Protect me from what? |
| DOCTOR: You are going to attack? |
| CAPTAIN: Yes. |
| DOCTOR: I would have to refuse this decision. |
| DOCTOR: My patients need my attention now. |

**TUTOR: That's right. Why do you think this utterance upset the doctor?**

USER: He didn't want to move the clinic

**TUTOR: That's right. You abruptly admitted your goal was to have him move the clinic in line 3.**

**There is room for improvement in line 10.**
**In the XAI window, please ask the doctor some questions and find out how this utterance contributed to the failure of the negotiation.**

*Click on Submit in the simulation history window to begin the investigation.*

**TUTOR: We first need to know why the negotiation failed.**

Enter your response:

USER: Let's start the investigation at line 10.
**DOCTOR: OK.**
USER: What is your negotiation stance?
**DOCTOR: The negotiation is over and I refuse to move the clinic.**
USER: What are the preconditions of running the clinic here?
**DOCTOR: having supplies here, the patients being safe here, the patients being here, the clinic being here and being neutral.**
USER: What action(s) could change being neutral?
**DOCTOR: negotiating with the Captain, cooperating with the Captain and the Americans moving the clinic.**
USER: Why did the negotiation fail?
**DOCTOR: I no longer trusted the Captain.**
USER: How was your solidarity with the Captain affected here?
**DOCTOR: It decreased.**

| ID | Question |
|---|---|
| 1 | What is your negotiation stance? |
| 2 | Why did the negotiation fail? |
| 3 | What is your trust of the Captain? |
| 5 | How was your solidarity with the Captain affected here? |
| 6 | Why did your solidarity with the Captain decrease? |
| 7 | How was the Captain's credibility affected here? |
| 9 | How was your familiarity with the Captain affected here? |
| 11 | How important do you feel being neutral is? |
| 12 | What action(s) could cause being neutral? |
| 13 | What are the effects of running the clinic here? |
| 14 | What action(s) could change being neutral? |
| 15 | What are the preconditions of running the clinic here? |
| 16 | What actions have being neutral as a precondition? |

Task: run-clinic-here State: neutrality

Figure 4: Interface to XAI for Virtual Humans

these are not subjective judgments, we anticipate that it will be fairly simple to encode these intuitions, and annotate teaching points automatically. The losses of trust and failure of the negotiation appear directly in the log file. Determining what the student should have done requires recognizing rules that increase trust and checking their preconditions (e.g., uttering a pleasantry such as "nice to meet you" is possible at line 3 and would have increased trust).

Our automated tutor is a prototype and we experimented with a hierarchical task network planner, JSHOP2 (a Java derivative of SHOP (Nau *et al.* 1999)) as its control system. In ongoing work, we are implementing a more reactive approach because relying solely on a planner for control meant that the planner had to be rerun after each student input (we could not predict the student's action), resulting in an inefficient system. In this section, we will focus on using XAI as teaching aid; the tutor also has the ability to question students directly and evaluate their answers using key-word spotting. Below are lines of a sample dialogue between tutor and student (T=tutor and S=student); the student is given the opportunity to identify his own errors and the student picks line 10. The tutor determines that line 10 is best taught by having the student investigate the problem using XAI.

1. T: In the simulation history window, can you select a line where you think you could have done better?
2. S: Line 10
3. T: That's right. There is room for improvement in line 10.
   In the XAI window, please ask the doctor some questions and find out how this utterance contributed to the failure of the negotiation.

While using XAI for the Virtual Humans, students are presented with a question list like the one in figure 5. The questions reflect information logged and our model of the doctor's behavior. The missing questions (q2, q4, etc.) are "why" questions associated with q1 through q9. We illustrate how "why" questions work in the continuation of the tutoring session below. The user asks the doctor about his negotiation stance, and now a new question, q2, is available (i.e., "why did the negotiation fail?"). The dialogue manager keeps track of these even numbered questions and every time we ask about line 10 this new question will be available.

4. S: Let's start the investigation at line 10.
5. D: OK
6. S: What is your negotiation stance?
7. D: The negotiation is over and I refuse to move the clinic
8. S: Why did the negotiation fail?
9. D: I no longer trusted the Captain

When the tutor instructs the student to use XAI, it activates a component called the investigation model tracer to track the student's progress with XAI. Currently, we define an ideal investigation as a series of questions whose answers provide the lessons to be learned. For this example the ideal investigation is very simple and consists of q2 (which was just asked) and q6. q6, "Why did your solidarity with the Captain decrease?", is made available after the student asks q5.

One reason the ideal investigation is so simple is that the student did not get very far in his negotiation. If the student had done better, he would have needed to know about the preconditions and effects of actions that concerned the

q1: What is your negotiation stance?
q3: What is your trust of the Captain?
q5: How was your solidarity with the Captain affected here?
q7: How was the Captain's credibility affected here?
q9: How was your familiarity with the Captain affected here?
q11: How important do you feel being neutral is?[a]
q12: What action(s) could cause being neutral?
q13: What are the effects of running the clinic here?
q14: What action(s) could change being neutral?
q15: What are the preconditions of running the clinic here?
q16: What actions have being neutral as a precondition?

[a] These virtual humans use utility theory to evaluate actions and states of the world. This question allows the user to query the relative importance of different states of the world from the doctor's perspective.

Figure 5: Sample question list for XAI for Virtual Humans

doctor such as running the clinic. Questions 11-16 allow the student to learn about the doctor's model of physical actions. In the current version of the system, there are 16 actions and 17 states of the world that the user can ask about. Users select actions and states from menus, and questions 11-16 change to match the selection.

The question list in figure 5 was generated when the current task was "running the clinic in its current location", and the current state was "the doctor's neutrality". Running the clinic in its current location, and running it in a new location are the two most important tasks in the doctor's world. Neutrality is important as we see in the dialogue continuation below, because it is a prerequisite for running the clinic. If the initial dialogue with the doctor had been more successful, the student would need to consider hiring locals to move the clinic instead of using U.S. troops.

10 S: What are the preconditions of running the clinic here?
11 D: having supplies here, the patients being safe here, the patients being here, the clinic being here and being neutral
12 S: What action(s) could change being neutral?
13 D: negotiating with the Captain, cooperating with the Captain and the Americans moving the clinic
14 S: What action(s) could cause the clinic being [moved] there?
15 D: the Americans moving the clinic, locals moving the clinic, and me moving the clinic

While the student was asking questions, the investigation model tracer watched his progress. After several turns have passed and the student has not asked q6, the tutor gives a hint as shown below (actually the hint appears in a different window). We use a standard model of hinting where the hints get more specific until the tutor gives away the answer. In this case, because we have not authored any hints, the tutor gives away the answer. Our mechanism for "unlocking" questions does not encode the relationships between the content of questions. So the tutor does not know that in order to know why solidarity decreased, the student must know that solidarity did decrease. We are working on fixing this problem in our next version of the system. Let's assume that the student realizes he must unlock q6 and asks q5 as shown below. Once the student has asked q6, the tutor recognizes that the investigation is complete and reengages the student in dialogue (in the tutor-student dialogue window).[1]

16 T: Please ask question #6
17 S: How was your solidarity with the Captain affected here?
18 D: It decreased.
19 S: Why did your solidarity with the Captain decrease?
20 D: The Captain is committing to performing an undesired act.
21 T: Good job. You found out that admitting to the planned attack decreased the doctor's trust of you, which caused the negotiation to fail...

## Future Work

In this section, we focus on the issues of domain independence and modularity, comparing our two systems, XAI for OOS and XAI for Virtual Humans. We discuss plans to take domain-dependent aspects of these systems and make them available as general features that can be activated or deactivated as appropriate. For example, XAI for OOS has the feature of removing non-applicable questions from its question list; if an entity did not fire its weapon at the current time point then the question, "what are you shooting at?", will be removed from the question list. However, there are pedagogical ramifications to this "feature". Consider a question from the Virtual Humans domain, "How was your solidarity with the Captain affected here?". We may want to display this question even if solidarity did not change because we want the student to realize that solidarity could have changed. Thus, we will allow the feature to be deactivated on a question-by-question bias.

Another feature present in XAI for OOS is the ability to use HTML formatting to include links to entities and times. Because entities and times are often the topic of conversation, we added the ability for users to click on mentions of entities and times to change the entity or time as opposed to selecting them from the entity or time menu. The XAI-for-OOS GUI interprets tags in the natural language generator's output in order to make this linkage. We will formalize this convention in our XML message format.

Other features are associated with dialogue context; in the sample dialogue, we saw that asking certain questions of the virtual doctor made new questions available (i.e., question unlocking). In XAI for OOS, entities "introduce" themselves when you first select them (e.g., "I am the fire team leader of Blue Fireteam 1..."), and in some contexts, users can ask the question, "can you give me more detail?". Currently XAI for Virtual Humans and XAI for OOS use ad-

[1] Although not shown in these excerpts, the goal is to encourage the student to be honest but vague (e.g., "I cannot discuss our operational plans").

hoc models of context limited to the information needed to support features such as question unlocking. We will build a general model of context storing each line of the student-XAI-tutor dialogue and who produced it. To enable question unlocking, we will store the logical form of questions asked and answers produced by the system. Questions will reference the dialogue context by listing applicability conditions rather than relying on a domain-specific context checking mechanism. This model of context will also be useful for the natural language generator, allowing it to tailor its output based on the context and produce more natural and readable text.

## Conclusion

Rather than simply writing an explanation system that only worked for its target AI system and simulator, we used our generic and modular architecture for explainable artificial intelligence (XAI) systems in building XAI for the One Semi-Automated Forces Objective System, a tactical military simulation (Courtemanche & Wittman 2002). This architecture continued to evolve as we worked on XAI for Virtual Humans, a simulation designed to teach soft skills such as leadership, teamwork, negotiation, and cultural awareness. In this paper, we presented an abbreviated case study on how to connect an explanation system to a target simulation and in particular, model behaviors and add support for new questions. We also showed how our prototype tutor uses XAI as a teaching tool, getting the student to understand his simulated negotiation partner's reasoning and mental state.

The key point of the XAI architecture is its domain independence and modularity. Every XAI system will have the basic components in figure 2 but their exact implementation will vary based on project requirements. For example, our reliance on a relational database and SQL queries was based on the requirement of handling a large dataset size and our short development times. The creation of the query logical form was a step toward a more declarative representation and future work may include more powerful reasoning and storage components.

We are currently continuing to work in the domain of negotiation with cultural awareness and are collaborating with a project called ELECT (Enhanced Learning Environments with Creative Technologies) here at the Institute for Creative Technologies. The goal of ELECT is to build a system including explanation and tutoring to be used directly in Army training and involves several external collaborators contributing subject matter expertise. We plan to work with the Army Research Institute to evaluate the effectiveness of our tutor and the XAI system.

## Acknowledgments

We thank William Swartout and David Traum for their guidance and support. The project described herein has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the U.S. Government; no official endorsement should be inferred.

## References

1990. FM 25-101: Battle Focused Training. Headquarters Department of the Army. Washington D.C.

Core, M. G.; Lane, H. C.; van Lent, M.; Solomon, S.; Gomboc, D.; and Carpenter, P. 2005. Toward question answering for simulations. In *Proc. of the IJCAI 2005 Workshop on Knowledge and Reasoning for Answering Questions (KRAQ05).*

Courtemanche, A., and Wittman, R. 2002. OneSAF: A product-line approach for a next-generation CGF. In *Proc. of the Eleventh SIW Conference on Computer-Generated Forces and Behavioral Representations*, 349–361.

Gomboc, D.; Solomon, S.; Core, M. G.; Lane, H. C.; and van Lent, M. 2005. Design recommendations to support automated explanation and tutoring. In *Proc. of the Fourteenth Conference on Behavior Representation in Modeling and Simulation.*

Johnson, W. L. 1994. Agents that explain their own actions. In *Proc. of the Fourth Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL.*

Laird, J. E.; Newell, A.; and Rosenbloom, P. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence* 33:1–64.

Nau, D. S.; Cao, Y.; Lotem, A.; and noz Avila, H. M. 1999. Shop: Simple hierarchical ordered planner. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99)*, 968–973.

Swartout, W. R., and Moore, J. D. 1993. Explanation in second generation expert systems. In David, J.; Krivine, J. P.; and Simmons, R., eds., *Second Generation Expert Systems*. Springer-Verlag.

Traum, D.; Swartout, W.; Marsella, S.; and Gratch, J. 2005. Fight, flight or negotiate: Believable strategies for conversing under crisis. In *Proc. of the 5th International Working Conference on Intelligent Virtual Agents.*

van Lent, M.; Fisher, W.; and Mancuso, M. 2004. An explainable artificial intelligence system for small-unit tactical behavior. In *Proc. of the Sixteenth Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press.