In Behavior Representation in Modeling and Simulation (BRIMS) 2006

Explaining Agent Behavior

Glenn Taylor
Keith Knudsen
Lisa Scott Holt
Soar Technology, Inc.
3600 Green Court Suite 600
Ann Arbor, MI 48105
734-327-8000
glenn@soartech.com, knudsen@soartech.com, lholt@soartech.com

Keywords:
Behavior Explanation, Intelligent Agents, Intelligent Synthetic Forces, Intelligent User Interfaces

ABSTRACT: *As intelligent synthetic forces (ISFs) become more complex, the ability to develop and employ them becomes more costly, in part because their behavior is inscrutable to non-developer users. This paper summarizes research aimed at reducing the cost of developing ISFs by addressing the lack of transparency of ISF behavior. Our goal is to produce systems that exhibit transparency of behavior, allowing users to interrogate an ISF about what it is doing, and why it is performing that behavior and not another behavior. Our approach is to develop a generic framework for automatic generation of multimodal explanation for ISFs. This framework (a) makes few assumptions about the underlying behavior architecture; (b) takes the form of an external observer of the behavior; (c) generates explanations based on a reconstruction of that behavior; and (d) incorporates a number of sources of knowledge to elaborate the explanations.*

1. Introduction

As intelligent synthetic forces (ISFs) have become more complex, covering with more breadth and depth the range of human behavior in simulation environments, the cost of developing and using them has increased. This is not surprising: complex systems are often opaque, offering little insight into their behavior, or else offering all their information in their own terms, thereby making users' interactions with them even more complex.

Our goal is to develop the capability for ISFs to explain what they are doing and why they are doing it, in terms familiar to the user. Our approach is to develop a generic framework for building explanation capabilities that can be connected to a wide range of agent systems in a wide range of domains. This approach extends the existing VISTA toolkit (Taylor, Jones et al. 2002) to provide support for explanation, including interfaces for querying about agent behavior and interfaces for providing explanations of that behavior.

2. Related Work

This work is a confluence of a few lines of research:

- Explaining the behavior of expert systems (Swartout, Paris, & Moore, 1991) (Wick & Thompson, 1992)
- Automatic generation of multimedia presentations (Feiner & McKeown, 1993) (Andre et al., 1993)

We are specifically interested in explaining the behavior of ISFs, which offer a rather unique set of issues for explanation. Unlike expert systems, ISFs continuously generate relevant behavior over the course of their lifespan—typically, a mission or multiple related tasks. Behavior in ISFs can span planning and execution during a mission, the maintenance of situational awareness, fusion of data sources, and decisions to act in the environment. This breadth of activity adds complexity when attempting to explain the behavior of the system. As ISFs typically are meant to act as humans would similar situations, explaining an ISF is analogous to explaining the behavior of a human performing a similar task.

There are a few other recent efforts that have examined the issue of explaining the behavior of ISFs:

- Debrief (Johnson, 1994) constructed "episodes" that captured the relevant information for an explanation. It also allowed for mental simulation to answer questions about hypothetical situations.
- Explainable AI (van Lent, Fisher, & Mancuso, 2004) uses low-level behavior traces and domain-specific information for answering a fixed set of questions about an entity's behavior.
- TRACE (Young & Harper, 2005) allowed for low-level investigation of beliefs, goals, and percepts to answer why a particular decision was made. Domain-independent causal graphs are used to explain behavior.

The approach taken in VISTA can be distinguished by the following:

- A general framework for building explanation systems across agent systems and application domains
- Use of multiple modalities for explanation (interlinked text and graphics)
- Use of multiple distinct knowledge sources from which to construct explanations—design knowledge, domain knowledge, behavior trace knowledge, and display knowledge
- A reconstructive approach to explanation

3. Key Technical and Scientific Challenges

In the work, we identified several challenging questions: 1) What kinds of explanations are useful? 2) What do users want to know? 3) What knowledge is available to generate explanations, and to integrate this knowledge into coherent explanations?

3.1 What Kinds of Explanations Are Useful?

(Gregor & Benbaset, 1999) offer a good survey of empirical studies of knowledge-based system explanation. They draw conclusions about explanations based on the type of user (novice versus expert) and type of task (learning task or problem-solving task). They also consider differences between context-specific versus generic explanation. Their conclusions are that (a) explanations are useful when focused on a task, and (b) different *kinds* and *forms* of explanation are useful for different users and tasks. In our work, we have further distinguished tasks specific to various user roles in the same ISF system—typically, one might be a *subject matter expert*, a *developer*, or *tester/QA* user. We explore these areas later in the discussion of the study we performed.

3.2 What Do Users Want to Know?

A previous VISTA study (Avraamides & Ritter, 2002) established a baseline for the kinds of questions asked of an ISF system. (Haynes, Ritter, Councill, & Cohen, (in process)) produced a deeper analysis of that same study, focusing on the types of questions users asked of the system, and relating these to prior work in system explanation.

Earlier work by (Lehnert, 1978), later refined by (Graesser, McMahen, & Johnson, 1994), identified types of questions asked when people are seeking answers generally (not just in explanation-seeking). We use these question categories to organize the explanation knowledge, and to determine the presentation of information to a user. We have focused on the following subset of Lehnert's question categories:

- **Causal Antecedent:** Asks about states or events that have caused the current state, e.g., "Why did you fire your missile?"
- **Goal Orientation:** Asks about motives or goals behind an action, e.g., "What is the purpose of the get-missile-lar goal?"
- **Enablement:** Specifies a causal relationship between an activity and physical or social enablers, e.g., "How were you able to fire the missile?"
- **Expectational:** Asks about the causal antecedent of an event that did not occur, e.g., "Why didn't you fire your missile?"
- **Concept Completion:** Who, what, where, and when questions, asking for the completion of a concept, e.g., "Where is the enemy contact?"
- **Quantification:** Asks for an amount, e.g., "How many missiles do you have?"
- **Feature Specification:** Asks about some property of a subject, e.g., "What is the range of the missile?"

3.3 What Knowledge Is Available to Use in Constructing Explanations?

Behavior trace data is the most common data available, and is used by many systems related to this effort (e.g., van Lent et al., 2004; Young & Harper, 2005). Because behavior trace data is a primary source from the behavior system, it is critical to use it in developing explanations. However, measured against the kinds of questions users have asked in prior studies, and the generic question categories developed by Lehnert and Graesser, it clearly is not enough information. Other necessary information includes the rationale behind a decision (e.g., why a particular goal was selected) or a justification for how some information is known.

In this system, we explore five sources of knowledge:

1) *Behavior Trace Data*: Data drawn directly from the agent system
2) *Agent Design Rationale*: The rationale that went into the design of the agent performing the task, including decisions regarding the organization of knowledge and processes that make up an agent
3) *Domain Knowledge*: Background knowledge about the types of objects and relationships in the domain, used for decision making
4) *Display Ontology:* Knowledge about the display and the information currently visible to the user
5) *Explanation Knowledge*: Knowledge about how to develop and present explanations

In large part, these classes of knowledge are developed as data for a particular application, independent of the VISTA framework. VISTA makes some assumptions about the form of the data, but not the content. The next sections describe how different knowledge sources are integrated to create explanations.

4 Explanation-enabled VISTA

4.1 VISTA and the SAP—Prior Work

The Situational Awareness Panel (SAP) (Taylor, Jones, Goldstein, Frederiksen, & Wray, 2002) is a tool developed to expose the awareness and decision-making of agents working in the tactical air combat domain. It is meant to help a user understand an agent's outward behavior and internal decision-making, given the agent's knowledge of its situation. The SAP was developed using VISTA, which provides a few features for developing agent visualization tools:

- Infrastructure for communicating with agents
- Default data and display components for classes of knowledge common to many agent architectures: goals, perceptions, other agents, etc.
- The ability to easily extend or create new data and display components for domain-specific applications
- Logging behavior trace data to allow for later replay and after action review

The SAP is a particular instantiation of the VISTA toolkit within a domain-specific application, tactical air combat, and extends VISTA to include domain-specific elements including enemy/friendly contacts, weapons, and a radar display.

4.2 Extending VISTA for Explanations

In our recent work, the objective was to extend the system in two dimensions: 1) increase the user interactivity of the system, so that the user takes an active role in asking questions; 2) improve the depth of information provided to the user beyond simply the behavior trace of the agent. With these extensions, a VISTA-based system can be thought of as a whiteboard for user-system explanation interactions, registering queries from the user and answers from the system. Users can query about objects on the display, invoked by clicking on the object (essentially, "tell me about this thing"), or by selecting hyperlinks in text explanations, which generate more explanation content ("answer this question" or "tell me more about this thing"). We chose to use hypertext and clickable objects in part to avoid the inherent problems of natural language input.

With each user request, the explanation system determines the relevant information, generates an abstract response plan, and then frames the response using the appropriate output formatting. This roughly follows the process espoused by the Intelligent Multi-Media Presentation Systems (IMMPS) Reference Model (Bordegoni et al., 1997).

The explanations provided by the system currently describe two major aspects of agent behavior or knowledge, which take advantage of the text and graphical displays native to VISTA:

- *Situation Summary*: an explanation of what the agent is currently doing and what it knows, in terms of its knowledge of its environment (percepts), its goals, and knowledge related to those goals
- *Object Summary:* explanation of a particular knowledge object in the agent and how it relates to the agent's goals. A special case of a knowledge object is a goal, and the system can explain elements of the goal, such as why the agent is pursuing that goal, what alternatives it might have pursued, and why it is not currently pursuing those alternatives.

The decision to present this particular content derives from prior work in explaining intelligent systems (Buchanan & Shortliffe, 1984; Swartout & Moore, 1993; Swartout et al., 1991; Wolverton, 1995), general question-answering (Graesser et al., 1994; Lehnert, 1978), and a study of users' experiences with the Situational Awareness Panel (SAP) (Avraamides & Ritter, 2002). In some cases, explanation types were added to directly support user questions that arose during the 2002 SAP study. Those questions were later expanded to account for the broader question categories described by the Lehnert/Graesser frameworks. Note that the system only allows the user to ask questions about the current situation in the present tense, not past or future. However, when replaying a logged behavior trace, the user can move forward or backward throughout the log to select (and store) points of interest for explanation.

Using this extended explanation-enabled VISTA framework, we extended the prototype Situational Awareness Panel to include example knowledge bases relevant to the tactical air combat domain. **Figure 1** below illustrates the Explanation SAP with combined graphical and textual modes of information display, and hyperlinks between them.

5. Explanation System Design

This section describes the architectural design and components of the VISTA explanation system. One of the first design questions we faced was the question of who should be responsible for providing explanations. Other approaches, such as Debrief (Johnson, 1994), placed the burden of providing explanations on the agent system itself. One advantage of this approach is

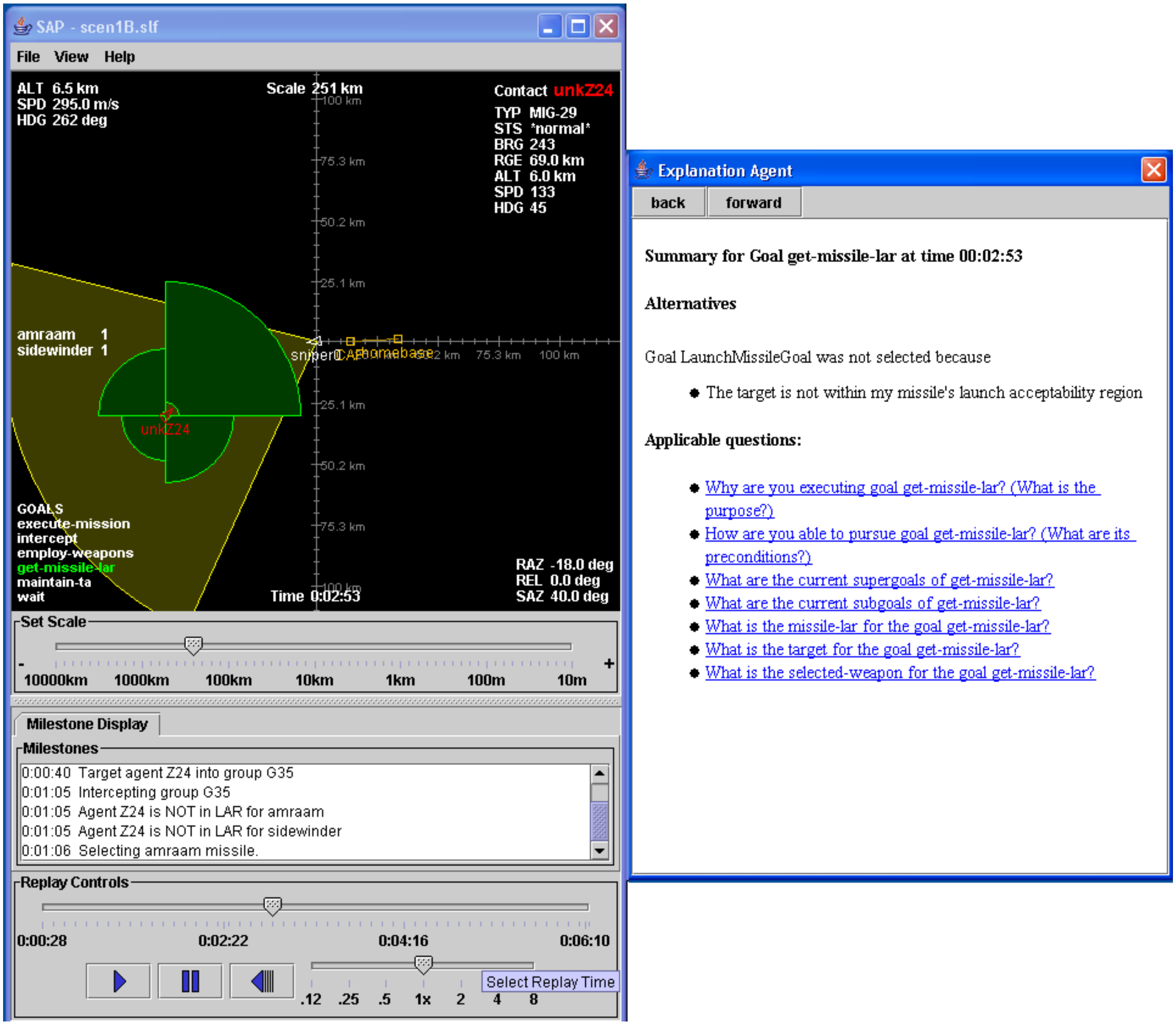

Figure 1: The Explanation-Enabled SAP and a Goal Object Summary with hyperlinked explanation

that the reasoning processes are directly accessible for explanation, but the solution was tightly coupled to the agent system and hindered the agent's performance. Alternatively, systems like REX (Wick & Thompson, 1992) use an external system to construct an explanation after the fact, allowing the agent system to perform its task unburdened.

Considering that generality is a major driving requirement of this work, we could not assume capabilities on the part of an agent system without tying ourselves to one kind of implementation. Our solution was to separate the explanation capability from the performance of the task at hand (e.g., tactical air combat as in TacAir-Soar). Our system distinguishes between two agents: first, the *performing agent (PA)* whose behavior the user is attempting to understand; and second, the *explanation agent (EA)*, which helps the user understand the behavior of the performing agent by entering into a dialogue with the user and providing answers in multimodal displays. The deliberate separation of the PA and the EA keeps the EA from having direct access into the decisions and knowledge of the performing agent. To provide explanation, the EA must reconstruct the situation to answer questions. This design decision was made to generalize and encapsulate the explanation capability independent of the behavior system, at the recognized price of being a step removed from the internals of the PA. **Figure 2** below illustrates the component-level architecture of VISTA, distinguishing the PA and EA, and indicating knowledge sources and data flows. The remainder of this section presents more detail of the

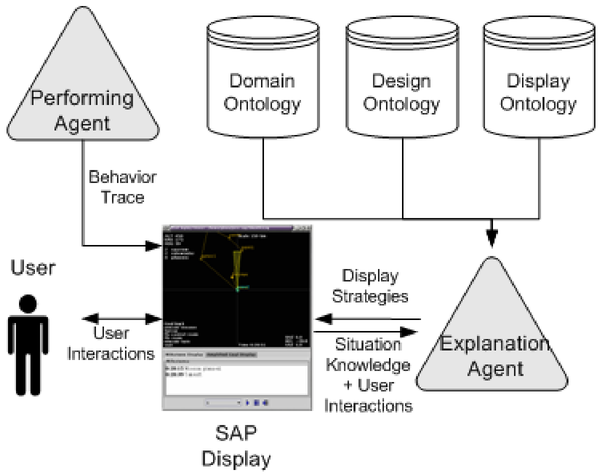

Figure 2: Component Interaction View of VISTA with Explanation Agent

knowledge sources, user interactions, and explanation processes.

5.1 Knowledge Sources

In order to determine what the user can ask about, and to construct answers to those questions, the system uses a few different knowledge bases in the form of ontologies. What follows is a detailed breakdown of the different knowledge sources and how they're used by the system.

5.2.1 Domain Ontology

The domain ontology describes knowledge about the domain in which the PA operates, including objects and relationships in that domain, independent of the design of the particular agent. In the TacAir-Soar example discussed later, the domain is tactical air combat, and includes, for example, terminological and class hierarchy information such as ***FA-18 is-a FixedWingAircraft***, ***Sparrow is-a Air2AirWeapon,*** and ***Sparrow has-a MaxRange of 5km.***

5.1.2 Design Ontology

The design ontology describes the actual design of the performing agent, including concepts and relationships that form the basis of the agent's design, such as specifics about goal hierarchies, preconditions, agent processes, etc. This knowledge is derived from the agent design process, including the rationale behind the organization of the knowledge in the knowledge engineering effort. Examples include ***InterceptGoal has-a InterceptGoalPreconditions*** and ***InterceptGoal has-supergoal ExecuteMissionGoal.***

5.1.3 Display Ontology

The Display Ontology consists of the objects and relationships that are used to construct the visual data and display elements. There are also mappings from the display components to the domain and display ontologies, in order to cross-query between them. The Display Ontology contains concepts such as ***Goal, Goal Stack, Radar, Contact, etc***.

5.2 User Input

We simplify user interaction by allowing only relevant questions to be asked in the current timeframe (wherever the user pauses or moves the time sliders). The different ontologies present a wide range of knowledge that can be queried, and each is mined to find information that could be sought in a given context. This information is then presented to the user as questions that he or she might ask of the system in the current context. As the context changes, the available questions change to match.

For example, suppose the user clicks on the icon for a contact—what are the questions that can be asked about this contact? The contact that was selected is a visual representation of an underlying data object, both of which have a reference in VISTA and in the explanation agent (EA). The EA representation of the contact has properties (bearing, range, altitude) that can be used to generate immediate questions about the object (e.g., "What is <u>the altitude</u> of this contact?" or, more generally, "What is <property> of <object type>?"). In this simple case, the answer is the value of the desired property. In other cases such as goal preconditions, more complex answers may be constructed. The contact object in the EA is also linked to other VISTA objects, including the active and inactive goals that might be related to contacts, which are used to construct other aspects of the object explanation.

5.3 Explanation Generation

The process of explanation used in VISTA roughly follows the steps described in the Reference Architecture for Intelligent Multi-Modal Presentation Systems (Bordegoni et al., 1997), which are described in the subsections that follow.

5.3.1 Managing the Communications

VISTA's job is to manage the user's input to determine what kind of explanation the user is seeking, and to define communicative goals to produce that explanation. Each situation (i.e., a current timestep in a behavior sequence) is a separate context for interaction, and the user drives the process for determining what information is presented by clicking on graphical elements or hyperlinks in the display.

5.3.2 Selecting Content

Selecting content for explanation is, in part, a process of querying the knowledge bases for information relevant to the explanation. This depends on the particular artifact selected by the user and the context in which that selection has occurred. Selecting a new situation produces a Situation Summary, which is meant to be the entry point to the general question "What is happening now?" Selecting content here is a matter of combining the current PA behavior trace information with design knowledge and knowledge about what is current displayed to develop a hypertext summary of the situation, an example of which is shown in **Figure 3**. Selecting the hyperlinks on this display can, depending on the subject of the link, highlight the object on the graphical display in some way, or possibly generate more explanation content.

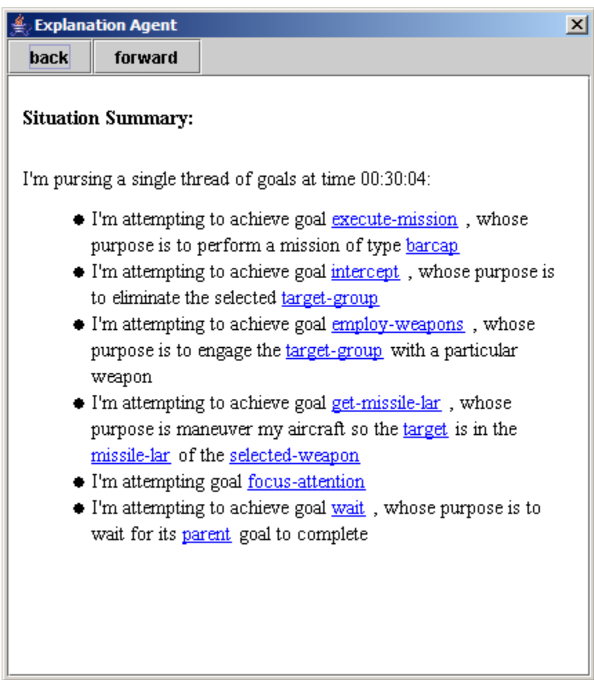

Figure 3: A Situation Summary

Selecting objects on the screen produces an Object Summary, which typically consists of questions matching Lehnert's *Feature Specification*, *Quantification* and *Concept Completion*, in addition to references to goals for which this object is relevant. Figure 4 illustrates an example Object Summary for a particular airborne contact in the display. Here, questions are shown as links, and answers are given upon selection, all of which is derived from domain and design knowledge about this object. This might result in showing an answer inline, or in generating a new page to display more detailed content. An example Situation Summary is shown in **Figure 4**.

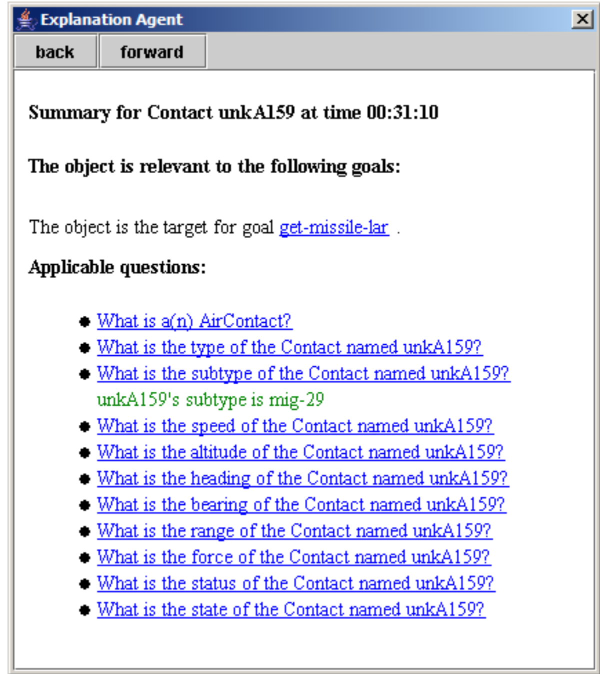

Figure 4: An Object Summary for an object of type AirContact, with an inline answer given

In a special case, if the object selected is a Goal, the Object Summary also contains answers to the following questions: *Enablement* (what pre-conditions made the agent perform this goal?), *Goal Orientation* (what is the purpose of this behavior?) and *Expectational* (what alternative behaviors might the agent be performing, and why isn't it doing them?). A Goal Object Summary is presented in the right-most panel of **Figure 1** given earlier.

5.3.3 Designing Presentation/Allocation of Media

An explanation is internally represented as a plan for presenting content to the user. This process currently follows some straightforward heuristics regarding presentation format, based on type of content to be display (e.g., Object Summary content is displayed as a text document with links). An explanation engineer can easily modify or extend these heuristics for different uses or users.

The output of this step is *presentation plan* composed of *display strategies*. Display strategies range from a description of an entire text document, to a command to display an object on the graphical display. Display strategies can be embedded and triggered by user action, or defined to execute immediately. Display strategies are often used in combination to form a more complete communicative act. Examples of VISTA/SAP display strategies include:

- **Display Object:** Ensure that the specified object is visible on the screen. May involve drawing the object and/or zooming the display to ensure visibility.
- **Highlight Object:** Highlight object on screen.
- **Display Document:** Display hypertext document constructed based on context (Situation Summary, Object Summary, answer to a question, etc.).
- **ShowLAR:** Display missile Launch Acceptability Region (LAR) icon.

Note that the last strategy here, ShowLAR, is domain-dependent, and is shown as an extension that can be made to the base generic strategies to highlight or explain certain domain-specific features. The LAR icon in this extension is shown above in **Figure 1** as a green multi-faceted halo around the target, based on features of the target (its orientation and icon) and features of the selected weapon. This information is known to the Explanation Agent through a combination of behavior trace information and domain knowledge.

5.3.4. Generating Media/Laying Out Media

In VISTA, generating media and layout is a combined step that consists of producing the actual text or graphics used to convey content. Layout is described precisely in the presentation plan, no further reasoning must be done in this step. Rendering of the presentation plan is handled internally to the drawing components of VISTA, rather than through direct manipulation by the explanation agent. The output of this step is the final rendered explanation, in graphical and textual modes.

6. Usability Study Summary

To evaluate the effectiveness of providing explanations to a user, we conducted a limited study using the SAP to address the following research questions: 1) How does the information presented in the SAP interface influence different populations performing behavior verification tasks, and 2) Do different populations seek or require different kinds of information when performing behavior verification tasks?

6.1 Behavior Verification Task

The user task in this study was to verify the behavior of a TacAir-Soar agent (Jones, Laird, & Nielsen, 1998) flying an air-to-air intercept mission, using a description of the correct behavior. In most military simulation exercises, systems like JSAF (Ceranowicz, Nielsen, & Koss, 2000) are the environment in which testers observe agent behavior and compare to pre-defined requirements. This process, called *face validation*, remains one of the most widely used methods of behavior verification in these environments despite its known limitations. JSAF's display natively provides information such as the position of the pilot agent on a map, its altitude, speed, heading, weapons, etc. However, it does not provide insight into the agent's cognitive behaviors (e.g., its perceptions, goals, or decisions). Along with the limited information provided by JSAF, the overall simulation behavior is non-deterministic, making it very difficult to reproduce the behavior from one run to the next (due to subtle timing issues, agent preferences, etc.). To reduce the complexity of the task and eliminate many of the variables inherent to using JSAF directly, we used a pre-recorded log of agent behavior played through a reduced-capability version of the SAP to provide information similar to that available to a JSAF user (henceforth called the Reduced SAP). **Table 1** summarizes the differences between the Reduced SAP and the SAP extended to provide detailed multi-modal explanations (henceforth called the Explanation SAP).

Table 1: Comparison of Reduced SAP and Explanation SAP

System Capability	Reduced SAP	Explanation SAP
Agent Altitude, Speed, Heading	+	+
Enemy Altitude, Speed, Heading	+	+
Weapon Inventory	+	+
Rewind/Fast Forward	+	+
Waypoints/Routes	+	+
Enemy Range and Bearing		+
Agent Goals		+
Situation Summary		+
Object Summary		+
Mission Milestones		+

6.2 Participants

The populations that typically perform agent behavior verification tasks vary greatly in their expertise.

- *System Testers* may have some software engineering experience but have not developed intelligent agents and have no knowledge of the target behavior. They use a behavior specification to verify system performance.
- *Knowledge Engineers (KEs)* have extensive experience programming intelligent agents in the target domain and have explicit knowledge of the rules governing the agents' behavior, but typically no experiential knowledge of the target domain.
- *Subject Matter Experts (SMEs)* typically have no software engineering or agent programming expertise. They are, however, experts in the domain of the target behavior.

For this study, twelve participants were recruited—six System Testers and six KEs. (SMEs were unavailable for this study.) Three participants from each population were randomly assigned to use the Explanation SAP interface. The remaining three participants in each population used the Reduced SAP. Human-subjects approval was secured from the Pennsylvania State University Institutional Review Board.

6.3 Target Behavior and Procedures

In this task, a TacAir-Soar agent performs an air-to-air intercept against an enemy aircraft. For this task, users were asked to study a description of the correct behavior of an intercept, with details of the decisions and other activities performed throughout the intercept task.

Three scenarios were created based on the intercept task:

- Scenario A: the agent performs the intercept mission correctly
- Scenario B: the agent selects the wrong missile to engage the target
- Scenario C: the agent begins the intercept too early according to its rules of engagement

The user's task was to observe each of these scenarios and answer questions about the correctness or incorrectness of the agent's behavior. Again, one test group (both System Testers and Knowledge Engineers) used the Reduced SAP, and the other group (again, both types of users) used the Explanation SAP. Questions were asked immediately after reading the correct behavior description, then while interacting with the SAPs, then after all the scenario quizzes were complete. The order of scenarios was randomized across subjects.

6.4 Results Summary

Overall, the System Testers and Knowledge Engineers did very well on the behavior verification tasks. In cases where the agent's decision making processes were directly observable through outward behaviors, most participants were able to answer quiz questions correctly. The users of the Explanation SAP performed better than those of the Reduced SAP in the following ways: 1) They were better able to describe the agent's cognitive behaviors, which were not directly observable, 2) They were more accurate in identifying specific times or distances, and 3) They demonstrated greater consistency between scenarios. We found that in certain scenarios, an error in the agent's decision making became highly visible through outward behaviors, but at other times (such as when the agent took the right action or nearly the right action, but for the wrong reasons) errors were much more subtle or completely imperceptible. Most users of the Explanation SAP were much more consistent in their ability to detect errors. There were surprisingly few differences in performance between the testers and the KEs. This was most likely an artifact of the simplicity of the study task, and relatively similar background among participants. The interface instructions and correct behavior description seemed to be comprehensive enough to bring the testers up to par with the KEs for the purposes of the tasks.

7. Conclusions

We have described a general framework for developing explanation capabilities for intelligent synthetic forces. This framework, extending the VISTA framework discussed in (Taylor et al., 2002), adopts a reconstructive explanation approach (e.g., REX (Wick & Thompson, 1992)) in which an *explanation agent* observes the behavior trace of a *performing agent*, and provides an interface through which a user can explore the rationale behind that behavior. We created a VISTA-based prototype as an interface for user exploration and displaying explanation content. We also developed an example explanation system for a subset of the TacAir-Soar system, in particular for air-to-air intercepts. We used this prototype as the basis for an efficacy study that explored the usefulness of the particular explanations provided and the types of information users sought in a behavior validation task.

Future work includes expanding the categories of knowledge used for constructing explanations, including user models, and developing more effective means of presenting information to a user.

8. Acknowledgements

This work was performed under Office of Naval Research Contract # N0001403C0095. We would like to thank Dr. Harold Hawkins for his support of this work, and Penn State University for their help in developing the study.

9. References

Andre, E., Finkler, W., Graf, W., Rist, T., Schauder, A., & Wahlster, W. (1993). WIP: The Automatic Synthesis of Multimodal Presentations. In M. T. Maybury (Ed.), *Intelligent Multimedia Interfaces*: MIT Press.

Avraamides, M. N., & Ritter, F. E. (2002). *Using Multidisciplinary Expert Evaluations to Test and Improve Cognitive Model Interfaces.* Paper presented at the Eleventh Conference on Computer-Generated Forces and Behavior Representation, Orlando, FL.

Bordegoni, M., Faconti, G., Maybury, M. T., Rist, T., Ruggieri, S., Trahanias, P., et al. (1997). A Standard Reference Model for intelligent Multimedia Presentations Systems. *Computer Standards and Interfaces, 18*, 477-496.

Buchanan, B. G., & Shortliffe, E. H. (Eds.). (1984). *Rule-Based Expert Systems*. Reading, MA: Addison-Wesley.

Ceranowicz, A., Nielsen, P. E., & Koss, F. (2000). *Behavioral Representation in JSAF*. Paper presented at the Ninth Annual Computer Generated Forces and Behavior Representation Conference, Orlando, FL.

Feiner, S. K., & McKeown, K. R. (1993). Automating the Generation of Coordinated Multimedia Explanations. In M. T. Maybury (Ed.), *Intelligent Multimedia Interfaces*: MIT Press.

Graesser, A. C., McMahen, C. L., & Johnson, B. K. (1994). Question Asking and Answering. In M. A. Gernsbacher (Ed.), *Handbook of Psycholinguistics* (pp. pp. 517 - 538). New York, NY: Academic Press.

Gregor, S., & Benbaset, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly, 23*(4), 497-530.

Haynes, S., Ritter, F. E., Councill, I. G., & Cohen, M. A. ((in process)). Explaining Intelligent Agents. *Journal of Autonomous Agents and Multi-Agent Systems (submitted)*.

Johnson, W. L. (1994). *Agents that Learn to Explain Themselves.* Paper presented at the Proceedings of the Twelfth National Conference on Artificial Intelligence.

Jones, R. M., Laird, J. E., & Nielsen, P. E. (1998). *Automated intelligent pilots for combat flight simulation.* Paper presented at the Tenth Annual Conference on Innovative Applications of Artificial Intelligence.

Lehnert, W. G. (1978). *The Process of Question Answering*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Swartout, W., & Moore, J. (1993). Explanation in Second Generation Expert Systems. In J. M. David, J.-P. Krivine & R. Simmons (Eds.), *Second Generation Expert Systems* (pp. 543-585). Berlin: Springer.

Swartout, W., Paris, C., & Moore, J. (1991, June). Design for Explainable Expert Systems. *IEEE Expert*, 58-64.

Taylor, G., Jones, R. M., Goldstein, M., Frederiksen, R., & Wray, R. E. (2002). *VISTA: A Generic Toolkit for Visualizing Agent Behavior.* Paper presented at the Eleventh Conference on Computer Generated Forces and Behavior Representation, Orlando, FL.

van Lent, M., Fisher, W., & Mancuso, M. (2004). *An explainable artificial intelligence system for small-unit tactical behavior.* Paper presented at the Sixteenth Conference on Innovative Applications of Artificial Intelligence Conference, San Jose, CA.

Wick, M. R., & Thompson, W. B. (1992). Reconstructive Expert System Explanation. *Artificial Intelligence, 54*(1-2), 33-70.

Wolverton, M. (1995). *Presenting Significant Information in Expert System Explanation.* Paper presented at the Seventh Portuguese Conference on Artificial Intelligence, Portugal.

Young, A., & Harper, K. (2005). *TRACE: An Ontology-based Approach to Generic Traceability Tools for Human Behavior Models.* Paper presented at the Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation, Universal City, CA.

Author Biographies

GLENN TAYLOR is a Research Scientist at Soar Technology, Inc. His R&D activities include agent-based systems, human behavior modeling, and natural language interfaces. He received his BS in computer science in 1994, and his MS in computer science and engineering in 1996, both from the University of Michigan.

KEITH KNUDSEN is a Research Scientist at Soar Technology, Inc. focusing on modeling human behavior and knowledge for simulation and systems evaluation. He has a background in cognitive psychology, computer science and cognitive modeling and has earned a Masters of Human Computer Interaction from Carnegie Mellon University's School of Computer Science.

LISA SCOTT HOLT, PhD, is a Research Scientist at Soar Technology, Inc. responsible for the instructional design, development, and evaluation of training technologies using intelligent SoarTech systems. She received a BS (1991) in Physics & Mathematics from Allegheny College, and a PhD (2001) in Cognitive Studies in Education from the University of Pittsburgh.