Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

IAGO PROJECT AND DEVELOPMENT OF COMPOUND AGENTS

John E. Hiles

MOVES INSTITUTE
U.S. Naval Postgraduate School
Monterey, CA 93940, U.S.A.

ABSTRACT

The IAGO Project explores the question of whether a software model, in the form of a computational model of cognitive behavior, can contribute to better anticipation of asymmetrical threats. The computational model used in IAGO is based on Cognitive Blending, a theoretical model proposed in the Cognitive Sciences to explain fundamental or backstage cognitive operations in the brain. This model was implemented with the use of multiagent systems that coordinated their activity with a bio-inspired operator called a Connector. This operator and several others used in the IAGO project have been incorporated into a programming library, called the CMAS Library. CMAS stands for Compound Multiagent System. Compound refers to multiagent systems, in which at least some of the agents contain embedded multiagent systems. In the case of IAGO these embedded systems implement Cognitive Blending.

1 INTRODUCTION

The IAGO project was conducted during 2002 and 2003 to explore the question of whether or not a computational model of cognitive behavior could contribute to better anticipation of asymmetric threats. This paper describes the IAGO project, and then discusses the project's model of cognitive behavior and the theory from Cognitive Science upon which it was based. The last part of the paper refers to the software library that has been produced to simplify development of the type of multiagent system used in project IAGO.

Current technical solutions that attempt to help in anticipation are often based on rational choice-type models. In the most basic sense, a rational choice model consists of a designer producing a set of alternatives for the subject that may achieve one or more of the subject's goals. Each of the alternatives is then equipped with a utility function. The utility function produces a number that represents the Expected Utility of that option (i.e., the product of the probability that the alternative will work times the value or result if it does work). Rational choice models have two inherent limitations that are particularly important in connection with anticipating asymmetric conflict. The first limitation is that rational choice models assume the subjects' decisions will be at least bounded rational. Bounded Rationality is explored in Arthur (1994) Although this assumption of bounded rational decisions is true most of the time in real-world decision-making, people are idiosyncratic and occasionally, even under conditions of perfect information, make decisions that are clearly not in their best interest. For example, although most of the time a terrorist will conduct an attack on a day that maximizes his benefit and minimizes his costs (such as the risk of being caught), a particular terrorist may wake up one morning, remember that his was the day on which his brother died, and move up his attack schedule without even being consciously aware of why he is doing it. Utility functions of individuals that are not well known are notoriously difficult to estimate at the best of times and such sudden, psychologically driven perturbations in utility are almost impossible to model accurately. The second limitation is actually more important; it involves the important relationship between innovation and asymmetric conflict. In a rational choice model a human designer, such as a subject matter expert, must construct the decision alternatives along with the utility functions of the subjects. The model designer's assumptions limit the options that the subject of the model can choose. But in asymmetric conflict the innovations of the subject are often the most important dimension of the analysis. So, while rational choice-based models are appropriate in certain circumstances (such as long-term strategic directions of a terrorist group), they have limited value in assessing or predicting the how and why of particular actions, especially actions that involve innovation by the subject.

Other types of models take more macro-level, economic approaches, applying generalized grievance functions to critical threshold values to model conflict dynamics, as described in Epstein, Steinbruner, and Parker (2001). These may be useful for assessing macro-level trends in terrorism (such as the effects of wars, recessions etc. on the general level of violence), but cannot give more fine-grain predictions of who exactly is likely to attack where and why.

As we will show later in this paper, the IAGO project proposes a model in which innovation, creative thought and heuristics are an intrinsic part of the behavior of the software subject. The subject model constructs new options as it goes. This is a key aspect of IAGO and an important point of departure from traditional decision-analytic models, such as those based on rational choice.

Our approach focuses on the cognitive foundation of the subject — how the subject processes information and what things mean to the subject. Clinical psychiatry tells us that the understanding of what things mean to an individual is the key to understanding the individual's behavior. This is known as the cognitive context of the subject's actions. Beck (2000) describes the role of this cognitive context: it is clear from clinical experience with individuals who are predisposed to violent activity that events often have a very different meaning to these subjects than they do to us. In most cases, for instance, the person that we see committing the violent action perceives him or herself to be a victim and perceives that the target of the violence is at fault and in many cases is the cause of the problem.

One of the key developments in cognitive psychology over the last decade has been a model that proposes mental spaces and conceptual blending as the mechanisms for conceptual integration or the construction of meaning within the human mind. Fauconnier and Turner (2002) explain how the brain converts experience into knowledge with their theory of Cognitive Blending. With the help of these concepts, it is possible to explain the process by which a subject constructs new knowledge and derives meaning from a stream of events. This is precisely what we require from the software in our model in order to help us with the anticipation of subject behavior.

Our multi-agent work at the Naval Postgraduate School has produced a number of new techniques over the last three years that enable us for the first time to implement this conceptual blending and mental space model. We have defined Tickets to serve as packages that incorporate knowledge inside an agent. Connectors coordinate the activities of multiple agents. We have extended the connector idea so that when two agents form a connection, the connection becomes persistent, resulting in a scale-free network that is based on the coordinated behavior of the agents. This network corresponds to the Integration Networks described in the work of Fauconnier and Turner. We can thereby use our agents to produce "bottom-up" knowledge structures. Then, as the resulting structures grow through the process of conceptual blending, they show how the agent has connected external events to internally generated mental spaces that are unique to the agent, combining past experience and internal conditions. Eventually the agent will make decisions and act upon its external environment. At that stage the knowledge structures will show the connection between a perception of the external world, the internal mental space, and explicit action on the external environment.

A key property of some of the agents in IAGO is their compound nature. These agents interact with the other agents and objects in their environment in a way that is characteristic of multiagent systems. But embedded within these compound agents is another multiagent system – the one that implements the conceptual blending model. We referred to this use of compound agents as a Compound Multi-Agent Systems (CMAS).

IAGO accomplished a limited demonstration of conceptual blending by software. There appears to be high potential in this intersection of techniques for building compound agents and theories from Cognitive Science about how brains produce conceptual blends. The combination could lead to applications in a number of areas including threat anticipation, training, vulnerability analysis, policy analysis, cross-cultural understanding, and virtual characters for simulations.

2 RESEARCH OBJECTIVE

IAGO assumes that a person's context of meaning is the key to understanding and anticipating his or her behavior. In other words, IAGO is interested in shifting the attention from the point of view of an observer to the perspective of the subject. Key events arrive as inputs to the subject, where a perceptual filter applies meaning to the events, which are then incorporated into the subject's mental apparatus. We focus therefore on how events are perceived and how they lead to the construction of meaning for the subject. By using a computational model of blending, IAGO attempts to construct *blended mental spaces* that represent the new meaning (the context of cognitive behavior).

In IAGO mental spaces are represented by agents. Goal orientation (not specific goals) within the software is responsible for the autonomous behavior of individual agents. The multi-agent system within IAGO will continually adjust the relationships between these agents (each representing a mental space) to form structures that fit the context of the subject. These structures are known as integration networks. As the subject changes because of learning and actions, the structure of the multi-agent system will also change. Continuous adjustments to the multi-agent system structure based on goal orientation thus support the production of evolving complex behavior and allow us to explore the cognitive patterns of the subject. In this way, IAGO takes a bottom-up approach to determining a subject's behavior. This depends on what may indeed be the single most important advantage of the multi-agent approach to simulation -- the continuous adaptation of the system's structure based on the subject's goals and intent and the adaptation of the individual agents.

The motivation behind IAGO and the ultimate goal for the project is to achieve a blending multi-agent system driven by tagged input streams produced by subject-matter experts.

3 IAGO'S TECHNOLOGY

The overview illustration of IAGO (Figure 1) explains our approach. The project subject matter expert has created a body of information that was known to the subject in real life. This information becomes the input data to the model. The input data is tagged with type information and is time-stamped according to when the information reached the subject.

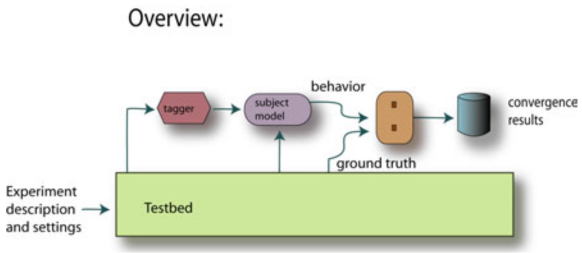

Figure 1: Overview of the IAGO Project

As more and more of the interior of a blended mental space are filled out with subsequent blending, the plan represented by the blended mental space approaches a point where all of its pieces are in place. At this point, the blended mental space is ready to actuate behavior (see Figure 2). The completed blended mental space moves to an output part of the agent where it is emitted into the comparator part of the test bed. The comparator captures the behavior and makes it possible to compare with the ground truth events that describe the subject's behavior based on the subject matter expert's research

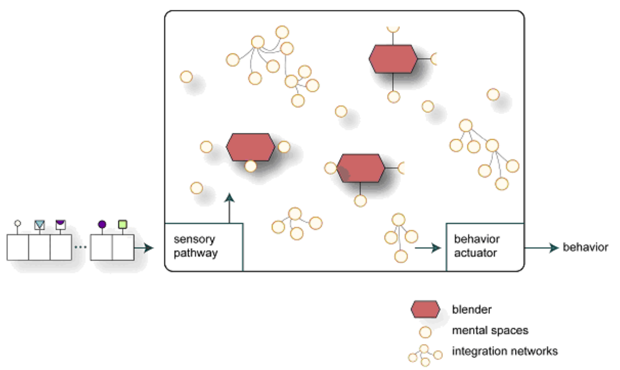

Figure 2: Subject Mode (Compound Agent)

As tagged information reaches the subject model, blenders combine the input mental spaces to form new blends. Our goal is to have these blends formed very fast and very smoothly. Cognitive psychologists studying the conceptual blending model point out that we form these blends effortlessly and in parallel. They also point out that blending goes on continuously, at high speed, and most of the time without our notice. As blends are produced in the subject model, they inhabit the larger mindscape, where they can in turn serve as input mental spaces for new blends and return to the blender to receive further processing.

As more and more of the interior of a blended mental space are filled out with subsequent blending, the plan represented by the blended mental space approaches a point where all of its pieces are in place. At this point, the blended mental space is ready to actuate behavior (see Figure 2).

The completed blended mental space moves to an output part of the agent where it is emitted into the comparator part of the test bed.

The comparator captures the behavior and makes it possible to compare with the ground truth events that describe the subject's behavior based on the subject matter expert's research.

3.1 Integration Networks

Figure 3 depicts the formation of an integration network. In an integration network we have two input mental spaces. These form on either side of the blender. Associated with the two input mental spaces, is the mental space called the generic space. The generic space contains type information, process information and techniques for projecting information elements from the two inputs into the new blended space at the top. It comes up from the bottom and attaches to the blender.

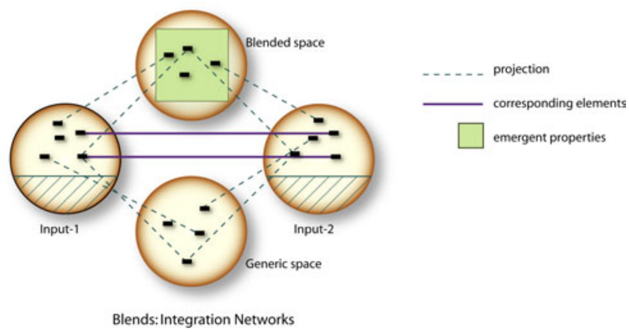

Figure 3: An Integration Network

The generic space and the two input mental spaces then form a blended space. The blended space will contain information found in the two input spaces but also new information not present in either one of the input spaces. This can take the simple form of a combination of input data not found in either one of the input spaces or more complicated forms that involve fusing or compression of information. When the blend is completed as determined by the generic space, the mental spaces leave the blender. But the connections that have held the mental spaces to the blender are converted into persistent links to form a new integration network with the blend at the top, the two input spaces and the generic space all connected as a network. The new blend (or any of the other mental spaces in its Integration Network) can return to a blender and be used in subsequent blends. Upon each return to a blender, new links are formed in the shape of a growing integration network. These networks satisfy the properties of a scale-free network in that the links are formed preferentially and incrementally.

A mental space can be thought of as consisting of frames containing data or instructions. If we look at a closer view of blending, we will see that the frames of a mental space serve as a recipe for that type of mental space. For example, in our project we have one type of generic space representing a terrorist's attack plan. When we examine an attack plan, we can see that the concept is made up of several generalized components. In effect, an attack plan is made up of the following pieces: a target, a harm-agent, an access route, an optional exit route, a trigger and a symbolic value. Each of the 40 to 50 types of generic spaces like the attack plan was constructed with the help of our subject-matter expert, and each of them has this type of highly generalized internal structure. These generic spaces are what guide the formation of new blends.

The central features of blending include the following properties: cross-space mapping, partial projection from inputs, generic space, integration of events, and emergent structure. Cross-space mapping means that information elements in the two input spaces are connected because they involve the same type of information. Partial projection from the inputs is guided by the generic space. Not all of the information in the two inputs is going to make it into the blend. Selected information is pushed up to the blend. Different combinations are tried; feedback is used to decide where the blends are effective or not effective. Generic spaces contain the meta-data that guides the projection and compression operations that result in a new blend. Emergent structure is a key to the success of blends. The information found in the two inputs can be projected selectively, can be fused or compressed by the generic space. The result is that the information that ends up in the blend is not the same as the information in either of the inputs. Something new has emerged in the process of blending.

3.2 The Blending Process

The software blender itself is an agent-based device. Figure 4 shows an illustration of the sequence of steps involved in blending. First, we have an available blender. No mental spaces are connected to the blender and the blender is waiting for first contact. Next we see an input space arriving at the blender. It connects with the blender at the *Input 1* side and causes the blender to extend connectors at the *generic space* and *Input 2* sides that will set up the blender for combination with the right kinds of mental spaces.

The next section shows a generic space and an input space arriving and making a connection with the blender. Now the blender has three mental spaces connected to it. Finally we see the generic space guiding the projection of information from the inputs into the new blend.

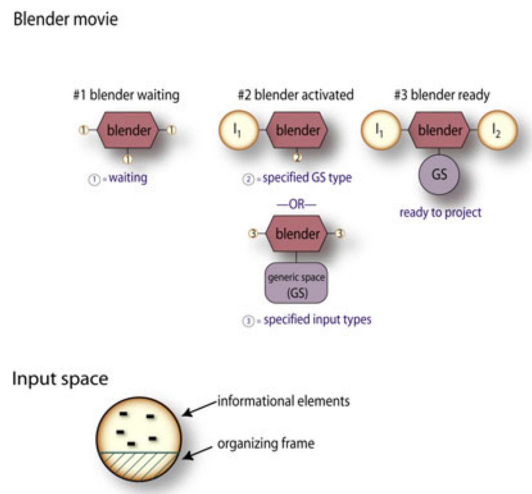

Figure 4: Blender Movie

After completion, the integrated network of these four mental spaces, starting with the blend, the two input spaces and the generic space, move off of the blender leaving it ready to resume blending (as described above). The integrated network of mental spaces can also move back onto the blender either as an input or a generic space and become connected to other mental spaces. The result is a growing scale-free network of mental spaces. The integration network thus becomes more complex - more and more mental spaces are combined with it as it goes through its process. Note that it is possible for an input to connect at the Input 2 side first or even a generic space to connect first in order to activate the blender.

3.3 Connector-Based Agents Make Blends Possible

Agent techniques at NPS have focused on composite agents that have interior sub-agents. The motivation is to create software agents that move in the direction of biological cells in terms of their autonomy and coordination with each other, as explained in Hiles, Osborn, Van Putte, Lewis, and Zyda (*2002).* Our computational model of blending results from a combination of multi-agent system techniques. These include tickets, connectors, composite agents, and networks. A ticket consists of one or more frames. Each frame contains an operation or information item that has a specific type. In IAGO there are two types of tickets – one type that contains data and one that contains information about operations. The data tickets have connectors extended to describe to the outside world – the world outside the ticket – what the frames contain. The active or operational tickets contain the steps or sequences of operations. Connectors are based on an analogy with the way receptors and control work in biological cells. Figure 5 shows two extended connectors that are about to match.

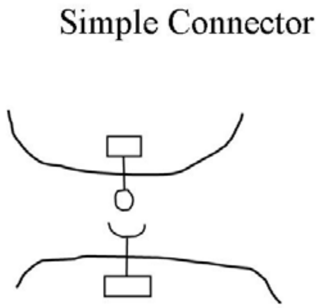

Figure 5: Connectors about to Form a Match

In our software world connectors have the following operations. They can be *extended*, which means that their type information is known outside of the agent, or they can be *retracted*, in which case the type information is pulled back inside the agent. An extended connector is waiting for a complementary or matching connector. When two connectors match, the operation is called a connection. A connector consists of a head part with type information and a tail part of a ticket. When two connectors match, the corresponding tickets inside the two agents execute, so now the operational type tickets can begin to execute. In our blending example a connector match may link an input space in the Input 1 position with the blender. The ticket execution inside the blender causes new connectors to be extended at the generic space and Input 2 positions. (See Figure 4)

A cascading sequence of connectors is extended as the blending process continues. Agents form networks through this connection process. When two agents have successfully matched their connectors, that connection is converted into a persistent link. The result is a growing network of agents. By basing the links on the connection process, our multi-agent system is able to satisfy the two requirements for forming *scale-free networks*: incremental addition of links and preferential attachment of links. We mean preferential in terms of the intent or goals of the agents that have formed the connection so that the application level intent is what is guiding the construction of the network links.

4 RESULTS

The first phase of the IAGO Project succeeded in creating a Compound Multiagent System that can create blends. To our knowledge, this is the first example of a computational model based on software agents to achieve conceptual blending. Detailed software requirements and specifications are described in a design document. Our software consists of approximately 140 classes of Java code, 25 of which are part of the human interface (called a GUI – graphic user interface).

Research has been conducted on a prototype terrorist subject to be used to test the blender. The subject matter expert has constructed a data set of over 300 items that became known to the subject over an extended period. These 300 data points will serve as the initial inputs and ground truth comparisons (and include the subject's resources, actions, skills etc). Each item was tracked so that we know when it became known to the subject. Approximately 25% of this data set has been tagged and packaged by the subject matter expert and is ready to test in the blender. This means that for each item the information has been placed in a formatted mental space with connectors extended that represent the type of information contained in the space. The subject matter expert has also produced approximately 50 kinds of mental spaces appropriate to a terrorist that are supported in the form of generic spaces. For each generic space there is an interior structure that describes in very abstract terms a specific kind of operation or information. The attack generic space described earlier is an example of such a generic space.

At the end of Project IAGO we are able to demonstrate that computational blending is possible and that multi-agent software blending can be applied to support the anticipation of subject behavior, using tagged information streams produced by Subject Matter Experts.

5 THE CMAS LIBRARY

Software operators like Connectors, Tickets, and Membranes facilitated the construction of IAGO. The project team has implemented those operations as services and will make them available to students, investigators, and developers by means of a CMAS Library.

Prior to the development of IAGO the project team referred to Connector-based multiagent systems with the acronym CMAS. But during the project and since, project members have come to realize that a much more important result follows from embedding connectors and agents within a surrounding agent, the Compound Agent. Compound Agents, especially ones that contain cognitive blending components appear to have great potential for building software that possesses adaptive complexity. We refer to the CMAS Library as the Compound Multiagent Systems Library. The CMAS Library will be a standard Java 1.4+ library of services needed to construct compound multiagent systems (or CMAS). The version of CMAS Library that will be released at the end of 2004 will contain support for Connectors, Tickets, and Membranes. These three categories can be used to place an embedded multi-agent blending component inside agents, as this project did with IAGO, by hand, without the services of a library. Preliminary experience with Master's Thesis projects at NPS has shown that development of the CMAS applications is both faster and more precise with the help of the library. We anticipate that a paper about CMAS will appear in 2005 that will document development projects based on the CMAS Library. That paper should also describe new services planned for the library. More information about the CMAS Library and the source and executable versions of the library itself may be found on the MOVES Institute web site at, `<www.movesinstitute.org>` after October 1st 2004.

6 CONCLUSIONS

The first phase of IAGO has shown that our CMAS-based computational model can achieve software blending. This blending process has produced new mental spaces from input streams collected from research into the behavior of a subject involved in asymmetric warfare.

IAGO has extended the Autonomous Software work of the MOVES Institute, here at the Naval Postgraduate School. Software Blending opens new possibilities for Compound Multiagent Systems. One of its key benefits is that Software Blending produces *bi-directional integration networks*, which have three important properties:

- Agents with this capability may go beyond working with remembered facts to produce new knowledge that is connected and linked to their intent and to their past.
- When these agents move forward along their Integration Networks, producing new blends from previous knowledge, they begin to know what they know.
- Conversely, a different process in the agent can move back from the latest blends to earlier mental spaces to produce new blends that contain knowledge about how the agent knows what it knows.

By combining Compound Agents, multiagent systems, Connectors, and Integration Networks, the first phase IAGO project has brought us to the prospect of building cognitive agents that can extend their Conceptual Blends based on new experience, a capability that will allow them to answer the question, "What are you doing?"

ACKNOWLEDGMENTS

This document is based on the work of the Project IAGO team:

- Principle Investigators: Curt Blais, John Hiles (Principle Designer), Ted Lewis
- Programmers: Bruce Allen, Neal Elzenga (Principle Programmer)
- Subject Matter Expert: Gary Ackerman

REFERENCES

Arthur, Brian. 1994. Bounded Rationality and Inductive Behavior (the El Farol Problem), *American Economic Review,* 84,406-411.

Beck, Aaron T. 2000. *Prisoners of Hate: the cognitive basis of anger, hostility, and violence.* New York. HarperCollins.

Epstein, Joshua M., Steinbruner, John D., and Parker, Miles T. 2001.: Modeling Civil Violence: An Agent-Based Computational Approach. `<http://www.brook.edu/es/dynamics/papers/cviolence/cviolence.htm>`

Fauconnier, G and M. Turner, 2002. *The Way We Think: conceptual blending and the mind's hidden complexities.* New York, NY. Basic Books

Hiles, J. E., Osborn, B. A., VanPutte, M., Lewis, T. G., and Zyda, M. "StoryEngine: Dynamic Story Production Using Software Agents That Discover Plans," unpublished technical paper, Naval Postgraduate School, 2002.

AUTHOR BIOGRAPHY

JOHN E. HILES is Research Professor and Director of the Laboratory for Autonomous Software Behavior within the MOVES Institute of the U.S. Naval Postgraduate School. He has been at NPS since 1999. Before that he was involved in software product development for over 27 years. He has led development efforts for nearly one hundred products, covering most major categories of application and system software. His most notable products include Amdahl Corporation's UTS, which successfully brought UNIX to IBM-compatible mainframe computers and Digital Research's GEM, a pioneering work in Graphical User Interface (GUI). In 1992 he began his work with adaptive, agent-based models and simulations at the computer game company, Maxis, creator of ***SimCity***. Mr. Hiles applied early concepts of Artificial Life and multi-agent systems to games designed for government and business. His research now follows the implications of Compound Multiagent Systems both in simulations and for operational systems.