

# **Using an Object-Oriented Paradigm to Develop an Insurgency Information Framework**

**LeeRoy Bronner, Ph.D., P.E., Swagata Pramanik, Akeila Richards**

**Morgan State University**

## 1. Background and Problem Definition

For the United States military in Iraq, the insurgent and their insurgent activities are a major cause of injury and death to citizens and the armed forces. Insurgency is a protracted political-military activity directed toward completely or partially controlling the resources of a country through the use of irregular military forces and illegal political organizations. It is an armed rebellion by any irregular armed force that rises up against an established authority, government, or administration. Those carrying out an insurgency are insurgents. Insurgents conduct sabotage and harassment. Insurgents usually are in opposition to a civil authority or government primarily in the hope of improving their condition.

The Army is constantly seeking to understand insurgency as it assumes a peacekeeping role in the Iraq. A large number of factors compound the insurgency problem and make it difficult for the Army to extract useful information in a timely manner to prevent insurgent attacks or to provide an effective response to them. They need to understand the insurgent and their associated population, the culture and behavior of the insurgent. They also need to determine the insurgent's targets and if there is any consistency among these targets. The military has problems in sorting out all the data available on insurgency for strategic purposes. To be effective in their strategic plans, the military needs a way to ask questions of insurgency information and receive immediate responses.

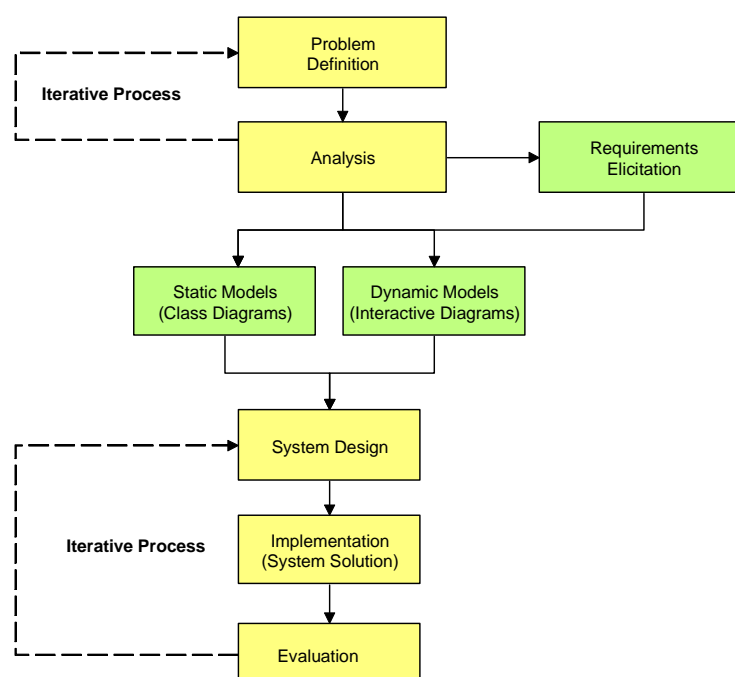
The Engineering and Psychology departments at Morgan State University (MSU) in collaboration have developed an Insurgency Information Framework (IIF). An IIF is a logical structure for organizing, classifying and presenting complex insurgency information for military decision making. The IIF is developed by analyzing and modeling the answers to questions such as: 1) How are insurgencies planned for and what is the process? 2) Does insurgent behavior provide any insight into their future plans or actions (i.e., are insurgencies random events)? 3) Is there a pattern in the locations and surroundings where insurgencies are carried out? The IIF will make it possible for the military to ask such

questions and receive answers for input to their strategic plans as well as support strategic thinking and decision making.

## 2. Theory and Methodology

### 2.1. Object-Oriented Analysis and Design

This research is based on the principles set forth by the object-oriented development life-cycle (figure 1). The development of a lifecycle model has six primary steps: 1) problem definition, 2) analysis, 3) design, 4) implementation, 5) Evaluation, and 6) Maintenance. As shown in the figure, the process is an iterative one providing an ongoing refinement of the initial problem definition model. Modeling is a human construct to help us better understand complex real world problems. Modeling complex systems has several general benefits. Modeling allows the client as well as all stakeholders to better understand the business or engineering problem and craft more effective solutions.



**Figure 1: The Object-Oriented Development Lifecycle**

Each step in the lifecycle process from problem definition to maintenance allows the analyst to work progressively toward a more detailed definition of the problem solution. A discussion of each step in the process is briefly outlined below.

Problem definition: is the phase which describes the functionality expected by the user, the set of requirements to be met, and the deliverables of the new system.

Analysis: is the stage where we begin to understand in depth the requirements of the problem and the need for its solution. It determines what the solution must do. The main components of the analysis phase are Project Scope, Use Case Model, Sequence Diagrams, Activity Diagrams, and Class Diagrams. These artifacts are explained in the following sections with respect to the insurgency model.

Design: the output from the Analysis Phase is the input to the design phase and results in the refinements associated with system specifications and software selection. During this phase, new classes and objects may be added to the model.

Implementation: is the phase where the system is implemented. Databases and source code are installed.

Evaluation: The implemented system is tested and evaluated here. It is an on-going process and is a part of system maintenance.

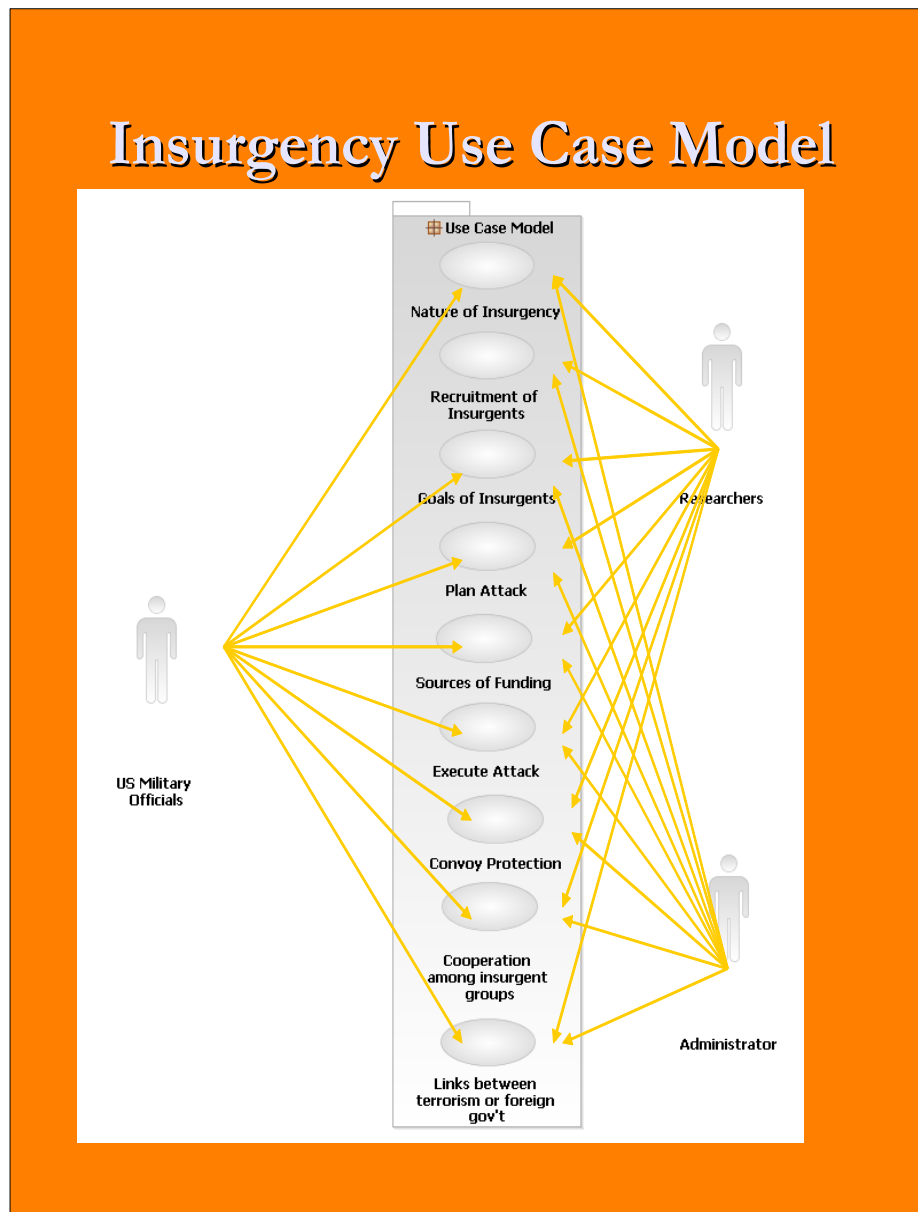
Maintenance: This is the ongoing process of error correction, system enhancements, backup and restore, disaster preparedness and recovery.

## 2.2. Insurgency Artifacts developed during the Object-Oriented Analysis Process

Developing a solution for the Iraqi insurgency problem requires that one must consider who the system stakeholders are and what use they will put the system to. In the case of this research the primary user will be military officers. The first step in the problem solution is defining the system use cases. A use case is a sequence of actions performed on the system under study that produces some measurable result. Whereas a use case model is a diagram or set of diagrams that show what the system is expected to do. For example, one of the use cases defined in figure 2 requires that the system solution be able to answer a question concerning attacks. Questions such as, how do insurgents plan for an insurgency attack?

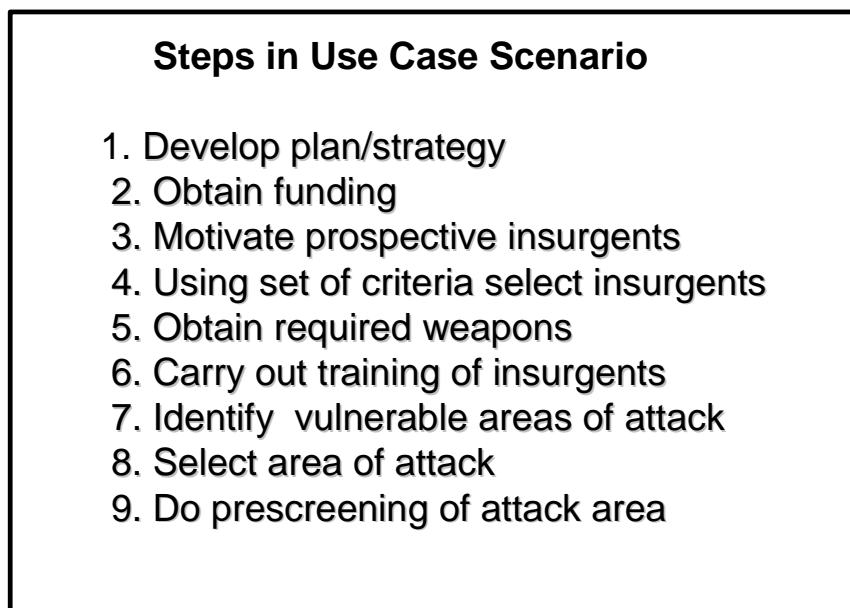
After the use cases are defined, the use case scenarios (figure 3) are developed to understand how the users are planning to use the system. This is a process where the analyst works with the user to conceptual this use process. The user must be interviewed and his responses documented. These use case scenarios will be used to select the key problem classes (figure 5) and their relationships. These classes become input for structuring the class diagrams (figure 4). As shown in figure 4, the class diagram begins to put a structure around the problem where it is possible to begin to see a problem picture. Behind each class defined in figure 4 is an abstraction of a problem entity. For example, figure 6 is a diagram of an object

structure created from the Attack class. Here in this object we begin to see where the insurgency story will be brought to life. We begin to talk about insurgency targets, types of weapons being used by the insurgents, location and occurrence of insurgent activity. This should begin to provide a window into the process being developed to model and understand the activities of the insurgents in Iraq.

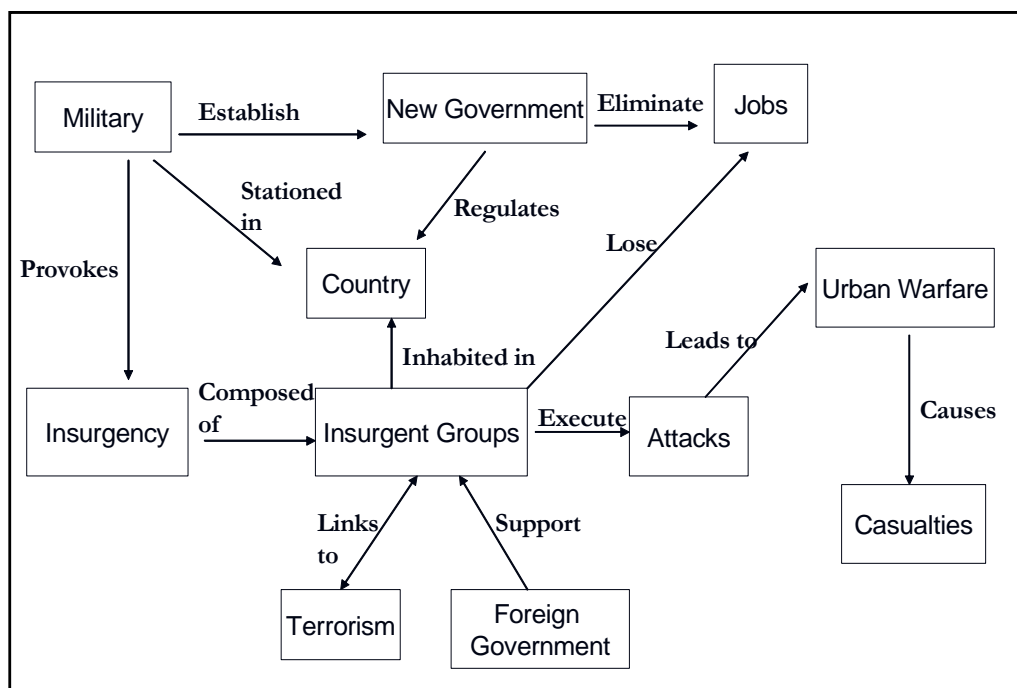


**Figure 2: The Insurgency Use Case Model**

*"Proceedings of the 2006 Mid-Atlantic Section Conference of the American Society for Engineering Education"*

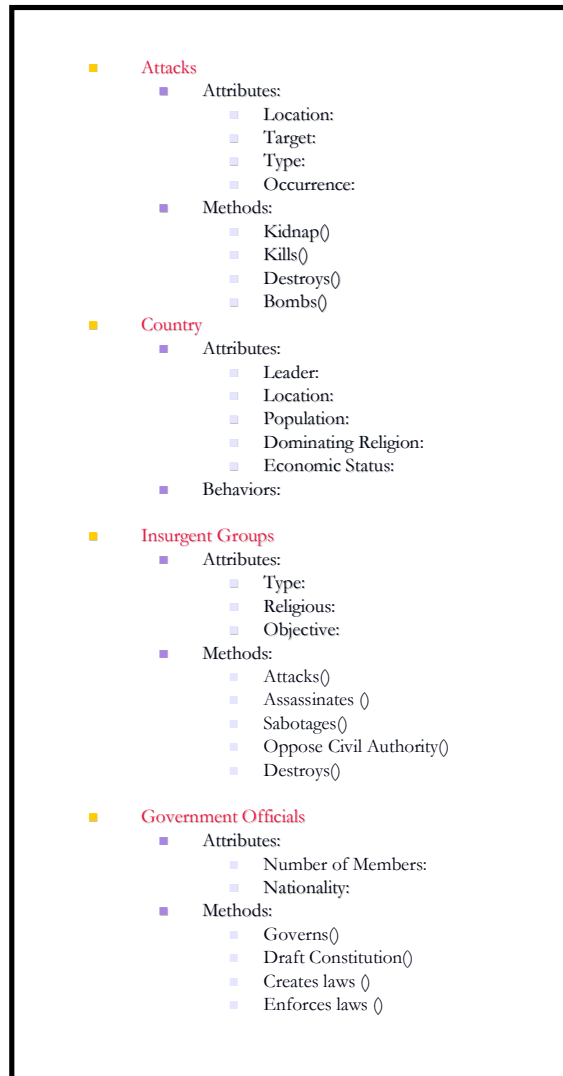


**Figure 3: Use Case Scenario – Attack Use Case**



**Figure 4: Example Insurgency Class Diagram**

*"Proceedings of the 2006 Mid-Atlantic Section Conference of the American Society for Engineering Education"*



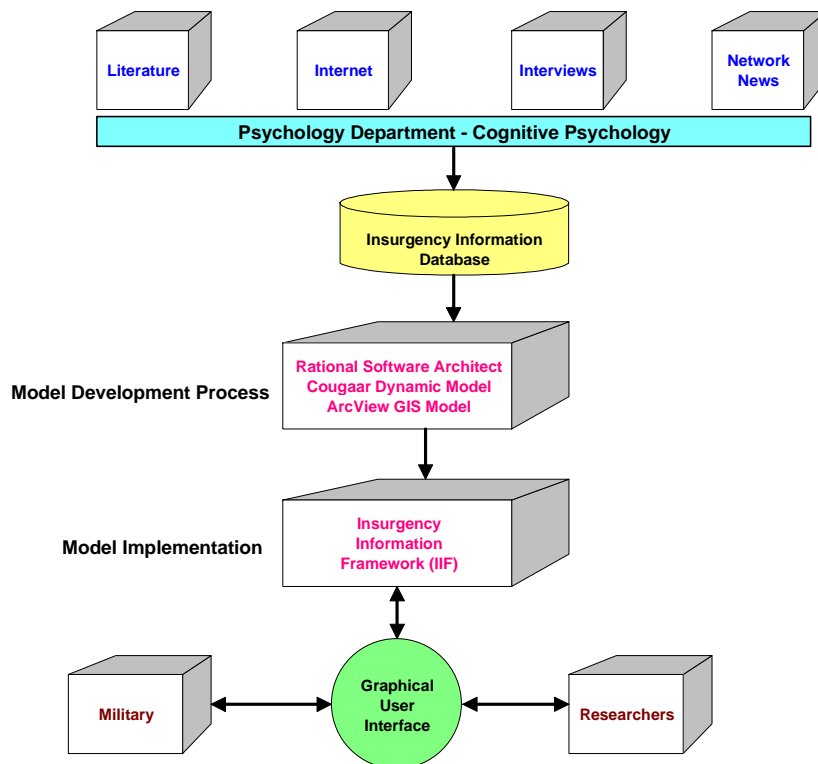
**Figure 5: Example Classes Structures - Nature of Insurgency**

Object: Attacks
<p><b><u>Attributes</u></b></p> <p><b>Targets:</b> Military personnel, civilians, security forces and/or citizens</p> <p><b>Type:</b> Mortar and Rocket Strikes, bombings, ground, weapons</p> <p><b>Location:</b> public facilities, airports</p> <p><b>Occurrence:</b> Random</p>
<p><b><u>Methods</u></b></p> <p>Assassinate( )</p> <p>Kidnap( )</p> <p>Kills()</p> <p>Sabotage ( )</p> <p>Devastates()</p> <p>Destroys()</p>

**Figure 6: Example Object Structure**

### 3. Insurgency Information Framework

The primary purpose of this research is the development of an Insurgency Information Framework (IIF) (figure 7). An IIF is a logical structure for organizing, classifying and presenting complex insurgency information for military decision making. The IIF is developed by analyzing and modeling answers to questions such as: 1) How are insurgencies planned for and what is the process? 2) Does insurgent behavior provide any insight into their future plans or actions (i.e., are these random events)? The IIF will make it possible for the military to ask such questions and receive answers for input to their strategic plans as well as support strategic thinking and decision making.



**Figure 7: Insurgency Information Framework**

Using object-oriented software engineering methodologies, an IIF has been developed driven by psychological research on the culture and behavior of people in Iraq. This research studied the “every-day life” of the Iraqi citizens as well as the behavior of the insurgent. As a part of this research an insurgency database was developed. A major effort in developing the IIF was the development and maintenance of an insurgency information database, which contain information such as Responsible Groups, Insurgent Target, Weapon Type, No-Killed (civilian/military), No-Injured (civilian/military), etc.

#### 4. Cougaar<sup>8</sup>: Dynamic Intelligent Agent Based Model

An agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future. Intelligent agents are agents that are goal-driven and autonomous, and can communicate and interact with each other. Moreover, they can evaluate information obtained from heterogeneous sources and present



information tailored to an individual's needs. So they are dynamic in the sense of the ongoing process of decision making<sup>9</sup>.

Cougaar (for Cognitive Agent Architecture) is the product of a multi-year DARPA research project to develop an open-source agent-based architecture that supports applications ranging from small-scale systems to large-scale highly-survivable distributed systems. It includes advanced core architecture and a variety of components that simplify the development, visualization, and management of complex distributed applications.

Cougaar is an innovative powerful software architecture that enables building agent-based applications in a manner that is powerful, expressive, scalable and maintainable. Cougaar has successfully demonstrated its utility at constructing dynamic, complex, agent-based applications. It is a Java-based architecture for the construction of highly scalable distributed agent-based applications<sup>7</sup>. It is a very powerful tool for the implementation of our human cognitive model.

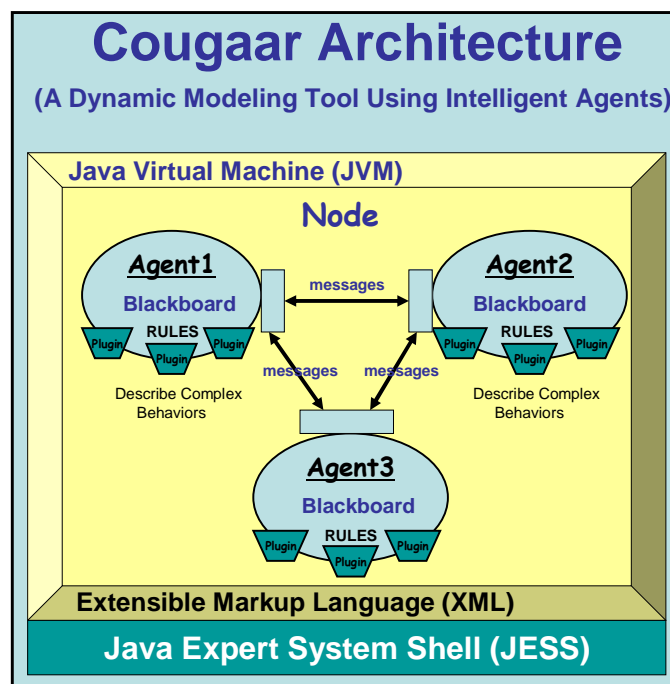


Figure 8: The Cougaar Dynamic Modeling Tool

The definition of Cougaar as a Cognitive Agent Architecture indicates that the Cougaar model solves problems in much the same way as a human would. The human cognitive process for solving a problem is given below. In the solution of a problem, humans tend to invoke several strategies in an iterative fashion which is outlined below:

Decomposing: breaking the problem down into smaller sub-problems.

Delegating: giving some problem to a resource to solve.

Consolidating: taking multiple independent pieces and handling them as a whole.

Monitoring: continually checking to make sure things are proceeding as planned.

Gathering: getting information from the outside world.

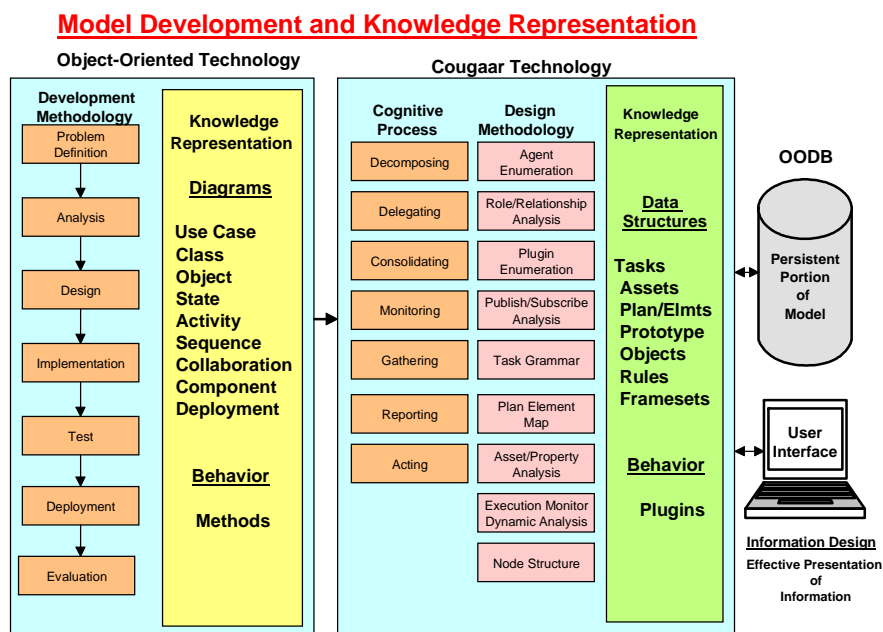
- Reporting: reporting back to the outside world.
- Acting: performing some act that impacts

This cognitive process gives rise to the Cougaar Design Methodology. The purpose of this methodology is to map a set of business processes onto Cougaar concepts. The top-level steps of the cougar methodology are given below:

- Agent Enumeration: define the agent entities of the problem.
- Role/Relationship Analysis: determine the roles or services that they will provide
- Plugin Enumeration: decompose the essential functionality of an agent until all pieces of the functionality play a simple role
- Publish/Subscribe Analysis: define what every Plugin is publishing and subscribing to.
- Task Grammar: declare task verb, preposition , direct/indirect objects for all tasks created by agents
- Plan Element Map: defines the decomposition of tasks into expansions, aggregations, and allocations.
- Asset/Property Analysis: define what assets are owned/managed by each agent.

- Execution Monitor/Dynamic Replanning Analysis: define assessors and triggers that will force replanning based on changes to assets.
- Node Structure: determine how agents should be allocated to nodes.

Since the initial problem analysis and resulting solution is done within the object-oriented paradigm, it is necessary to provide a transformation mapping to the cougaar technology (figure 9). As shown in the figure, artifacts (i.e., classes, objects, sequence diagrams, etc.) generated from the object-oriented analysis must be mapped to cougaar technology (i.e., agents, plugins, plan element maps, etc.).



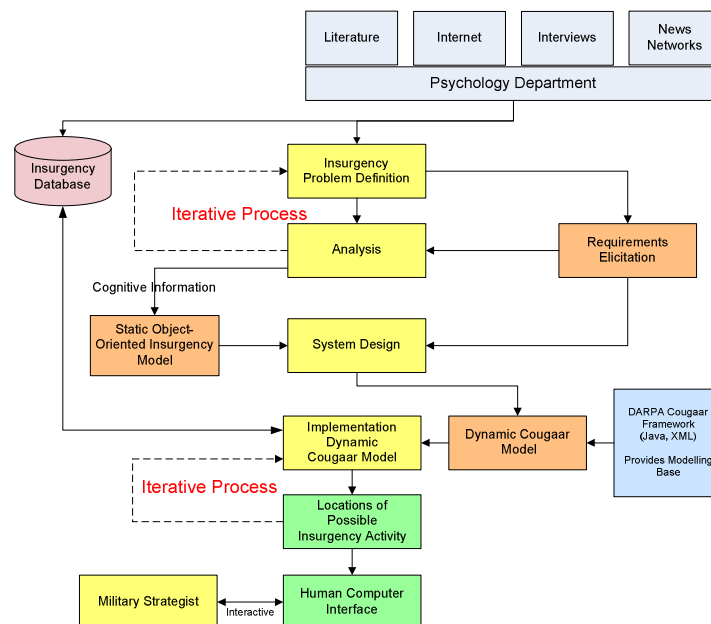
**Figure 9: Object-Oriented Transformation to Cougaar Technology**

## 5. Implementation of the Information Insurgency Framework

The methodology starts with a clear definition of the problem to be addressed (figure 10). To solve any problem, we need to have a clear understanding of the problem. In this research, the problem is to develop an Insurgency Information Framework (IIF). The problem definition is modeled using object-oriented technology using the IBM Rational Software Architect. The Rational Architect

*"Proceedings of the 2006 Mid-Atlantic Section Conference of the American Society for Engineering Education"*

Tool implements the Unified Modeling Language (UML) for modeling problems and their solutions. This tool also allows the analyst to archive and manage the many artifacts developed during problem analysis and design. The JAD session brings together the researchers, military and any other interested stakeholder. After completion of the JAD session, the model defined at problem definition time will continue to be refined and used throughout the object-oriented lifecycle. The object-oriented model will be translated into a form for input to the Cougar model. The following are the stepwise representation of the processes carried out in developing the IIF model.



**Figure 10. Model Development Process**

The problem solution development design process:

- JAD Sessions
- Problem Definition
- Develop Use Case Models
- Develop Supporting Activity Diagrams
- Develop Class Diagrams
- Develop Sequence Diagrams

- Develop Collaboration Diagrams
- Develop Statechart Diagrams
- Develop State Transition Tables
- Perform Transition Process
- Develop Cougaar Model
- Execute Cougaar Model
- Evaluate Results

#### 6. Benefits of the IIF for military use

For planning any strategy to solve a problem and taking action accordingly, such as collecting and organizing information is very important. Joint research between MSU's Engineering and Psychology departments has resulted in collecting data and putting a structure around the information available on Iraq's insurgency problem. The IIF developed is useful for organizing, classifying and presenting complex insurgency information for military decision making. Military can use the IIF to understand the insurgency problem and make strategic plans to react to it. The IIF is beneficial for them in where it is possible to ask several questions in determining the locations of insurgencies involving the largest number of casualties, role of religious holiday in insurgency plans. It may also be asked whether insurgent behavior provide any insight into their future plans or actions or are these very random events. The IIF can also provide answers to queries like whether insurgents are increasing their attacks on High Ranking Officials (Military/Civilian). This insurgent information being processed can be used for military planning purposes as well as support for strategic thinking to improve ongoing decision making.

#### 7. Conclusions and Future Work

This research has shown that modeling and analysis can provide a conceptual framework to aid the military in planning for insurgency activity. By integrating object-oriented modeling techniques with a dynamic intelligent agent-based architecture, the information framework becomes more dynamic and will react or adapt as situations change. This research is ongoing and as we become more skilled at this type of development we hope to take more advantage of the intelligent agent capabilities of Cougaar drawing upon the human cognitive capability. Cougaar uses its human cognitive processes to take dynamic action in an iterative manner. This makes it possible for the model to progressively handle more complex knowledge structures while adapting to change and providing results for improved on-going decision making. The military can make their decisions based on the information obtained from this dynamic model.

## References

- [1] Bradshaw, Jeffrey M., "Software Agents," Cambridge, MA, The MIT Press, 2nd Edition, 1997.
- [2] Hunt, Reed R., Ellis, Henry C., "Fundamentals of Cognitive Psychology," Boston, MA, McGraw Hill, 7th Edition, 2004.
- [3] Jacobson, Robert, "Information Design," Cambridge, MA, The MIT Press, 1999.
- [4] Larman, Craig, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development," Upper Saddle River, N. J., Prentice Hall, 3rd Edition, 2005.
- [5] Tsang, Curtis HK, Lau, Clarence SW, Leung, Ying K, "Object-Oriented Technology," Boston, MA, McGraw Hill, 2005.
- [6] Boggs, Wendy, Boggs, Michael, "Mastering UML with Rational Rose 2002," San Francisco, CA, SYBEX, Inc.
- [7] Cougaar website, <http://cougaar.org>.
- [8] A. Helsinger, R. Lazarus, W. Wright, and J. Zinky, "Tools and Techniques for Performance Measurement of Large Distributed Multiagent Systems," *2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Melbourne, 2003.
- [9] UMBC AgentWeb website, <http://agents.umbc.edu>.
- [10] G. Nguyen, T. Dang, L. Hluchy, Z. Balogh, M. Laclavik, and I. Budinska, "Agent Platform Evaluation and Comparison". Technical report for Pellucid 5FP IST-2001-34519. Jun 2002, Bratislava, Slovakia.
- [11] Daniel D. Corkill, "Blackboard Systems", *AI Expert* 6(9):40-47, September 1991.
- [12] D. D. Corkill, "Collaborating software: Blackboard and multi-agent systems & the future", *Proceedings of the International Lisp Conference*, New York, NY, USA, 2003.

## Biographical Data

### LEEROY BRONNER

Dr. Bronner is a Research Associate Professor at Morgan State University in the department of Industrial Manufacturing and Information Engineering. He has been an instructor at Morgan State University for the past 7 years. Dr. Bronner spent 25 years at the IBM Corporation and brings to academia experience in systems and software engineering, analysis, design, modeling, programming and systems implementation.

### SWAGATA PRAMANIK

Swagata is a Doctoral Candidate at Morgan State University's department of Industrial Manufacturing and Information Engineering. She is doing research in the field of Information System.

### AKEILA RICHARDS

Akeila is senior engineering student in the Industrial Manufacturing and Information Engineering department at Morgan State University. She will graduate in December, 2006.

[Return to Main page](#)