A Development Process OOTW/OMO Analytic Toolkit

D.F. Davis

(ddavis@gmu.edu, (703) 993-1703)

The Program on Peacekeeping Policy
The Institute of Public Policy
George Mason University

Background:

Operations Other than War (OOTW), or Other Military Operations (OMO), include those activities of military organizations that are not directly focused on war fighting. The US Army Field Manual on Operations (FM100-5) includes peacekeeping, noncombatant evacuation operations, support to civil authorities, and others in this category. Since 1972 or 1973 at the end of the Vietnam War, the great majority of operations conducted by the US military have been in these categories; the major exception is Desert Storm/Desert Shield. Strong argument may be made that the operations in Grenada and Panama were OMO and not War Fighting Operations. These arguments even when the major efforts were combat driven.

Since the end of the cold war and the decline of the bi-polar threat in the late 1980s, the concern of the US security planners has moved more and more toward the OOTW/OMO areas. Prior to that, even given the arguments of the previous paragraph, the community had focused on a significant strategic war with the former Soviet Union. Combatant Commands and the Military Departments are rapidly gathering lessons learned documents, writing doctrine, and conducting exercises. All of this activity to gain an understanding of OOTW/OMO.

This analysis is currently unfocused and occurring in a very decentralized way. Unlike the combat and warfighting processes which still demand complex analysis by thousands of scientists and engineers using well understood computer simulations and exercises, there is little directed or focused analysis of the new missions. Unfortunately, some of the analysis that is being done is being done with the old combat simulations. Using the old simulations for the new missions is potentially misleading and dangerous.

Process:

This paper describes how a collection of tools may be designed to partially remedy this situation. This design, however, needs to be developed carefully in this new arena. Systems development can proceed along many different paths. Unfortunately, the most common path is the ad-hoc development process. This occurs when a manager or commander states a need: "I think we need a tool kit." This may turn out to be the complete statement of the requirement. It then becomes necessary to 'flesh out' the requirement in order to specify a design. In fact, the initial

statement is usually proffered in the form of a solution. The manager or commander sees a need, and jumps to a potential solution to the problem. This premature statement of a solution has caused many problems. Not the least of which is artificial narrowing of the solution space. Developers are biased by the statement, and self-limit their engineering process.

The systems processes today stem mostly from the software engineering domain. Terms like Waterfall and Spiral are most often heard. These are variations on the top - down and bottom up approaches. In top down approaches, a high level requirement is 'discovered' and then decomposed until components can be identified. Bottom up approaches, as many object oriented approaches devolve to, begin by identifying the lowest level of component and reverse engineering a design.

The conceptual modeling approach begins by recognizing the strength of both of these techniques and adding to them. Figure 1 shows the approach which begins with a high level model, discovered in a process of increasing participatory design. Then the process cycles down to the existing modules to identify constraints and interfaces, as well as possible reusable components (sometimes called 'golden nuggets'). This high/low process continues: Down in the problem space of the domain model, and up in the solution space of existing components. Essentially, this process allows the engineer/developer/manager team to use the lower level requirements to better understand the higher level requirements.

Tool Kit Development Process

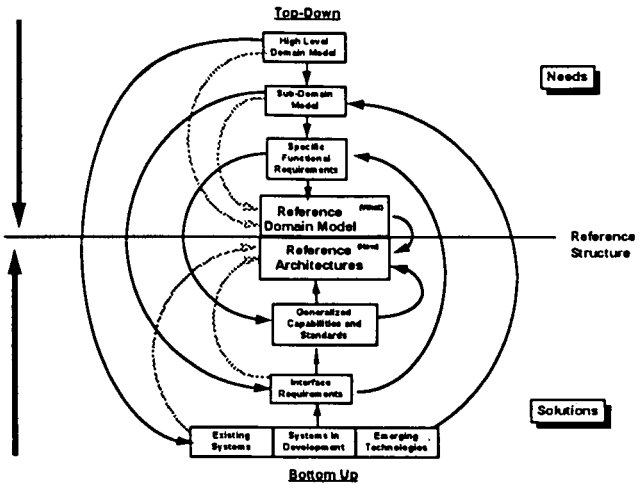

Figure 1
The Whirly-Gig

The top level approach, in what is called the problem domain, is centered on the description of the process to be modeled. It is key during this phase that the developers not try to duplicate existing systems or procedures, but attempt to describe what needs to be done. This enforced independence of existing systems will allow the team to focus on the issue to be solved. Identifying the 'what' of what needs to be done, without contamination from the 'how' of how to do it. The result of the top level model will be a statement of the true system, from which requirements for some sub-domain can be (almost) directly derived.

This derivation depends on the acceptability of the requirements, which depends on the comprehensives of the model. Thus the model must lead the community to some level of consensus. One method of developing and working

toward consensus is that of behavioral modeling.

Example:

As a simple example, we have started the development of a conceptual model of peace operations, as a sub-domain of Operations Other Than War. The behavioral diagram of Figure 2 represents the process of Peace Operations in a loop. The loop can be exited by the outbreak of Peace or War, or a simple Withdrawal of the force. Otherwise, if a resolution is still deemed possible, the process simply continues.

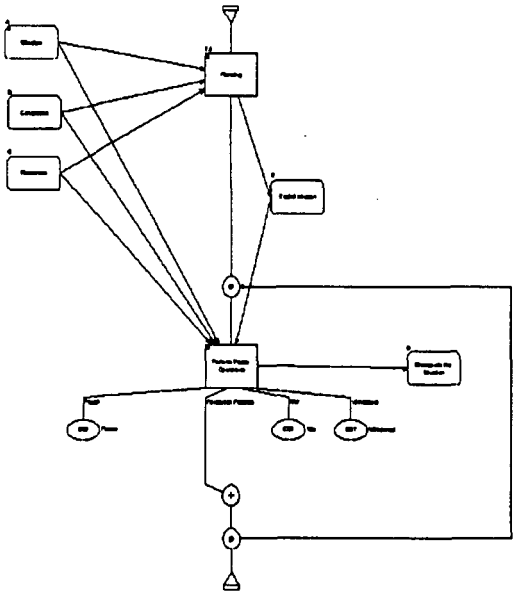

Figure 2
Peace Operations Domain Model

The operation depends upon some sort of planning or command and control process. The efficacy of this planning process may be a point of contention today, but it remains that some planning is done, and a mission is formed.

The internal representation of this C^2 process will be derived from the joint George Mason University/CECOM Software Engineering Directorate's Conceptual Model of Command and Control ($CModC^2$). The $CModC^2$ is a domain model in behavioral form that has seen wide usage in the explanation of C^2 requirements. This model of command and control has undergone continued development and validation effort for several years. The validation has been subjective, command expertise, and objective engineering consistency evaluation.

The operations element of this potential domain model has been provisionally decomposed. One, straw-man, decomposition of the Peace Operations (Figure 3) process shows that some sort of activity is generated based on the inputs of situation, constraints, resources, and mission. Some of these activities will directly effect the parties of interest, some will be coordinated prior to causing the effect.

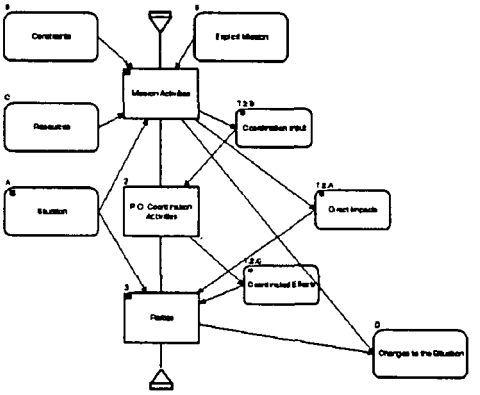

Figure 3
Peace Operations Decomposition

These processes can be further decomposed into other activities. The activities are seen as Military, Political, and Humanitarian. The parties are decomposed into parties to the conflict and external parties. Further decomposition and analysis will lead to the finer definition of what functions are required, and what information is used by and produced by those functions.

Modeling:

These diagrams are developed in an ordered process. Initially, a small group (1- 4) analysts and domain experts jointly develop a straw man that is iteratively improved over a period of time. When sufficient detail (currently a subjective judgment) is modeled, a larger group (10 - 15) of domain experts are gathered for a structured workshop on the model. This workshop is conducted with the goal of providing more detail of decomposition in the processes of the domain model, and of re-validating the current model. The members of the workshop should have previous exposure to the model such as soft copy or Web documents for easy of navigation and browsing.

The resultant model is then made available to the system's developer and to the community at large. The developer may begin to allocate functions to components and start the architecture/design phase. The community at large will comment on the model for update -- but most importantly, to understand what the system is being designed to accomplish.

Once the model has been developed to the point of web exposure, it can be used in the development process of systems. The model should, by this point, incorporate the entire set of domain requirements. The lowest level of model decomposition may then be subjected to an analysis to determine what, if any, current systems exist that provide the needed functionality. Functions that are not provided by current systems, may then be identified, and allocated to a new design effort.

Management:

The usefulness of the model goes beyond the engineering processes of requirements management and design. The original commanders and managers who stated the need, may use the model to refine their understanding and to develop insights into the process. By tracing through the levels of decomposition, and following the information flows, critical processes and functions may be identified for closer surveillance and evaluation.

One of the manager's largest problems is the comprehension of large, complex systems under his or her cognizance. By providing the manager with a browsable, graphical representation of these systems, the manager is given a significant tool.

A Development Process OOTW/OMO Analytic Toolkit

D.F. Davis

(ddavis@gmu.edu, (703) 993-1703)

The Program on Peacekeeping Policy The Institute of Public Policy George Mason University

Background:

Operations Other than War (OOTW), or Other Military Operations (OMO), include those activities of military organizations that are not directly focused on war fighting. The US Army Field Manual on Operations (FM100-5) includes peacekeeping, noncombatant evacuation operations, support to civil authorities, and others in this category. Since 1972 or 1973 at the end of the Vietnam War, the great majority of operations conducted by the US military have been in these categories; the major exception is Desert Storm/Desert Shield. Strong argument may be made that the operations in Grenada and Panama were OMO and not War Fighting Operations. These arguments even when the major efforts were combat driven.

Since the end of the cold war and the decline of the bi-polar threat in the late 1980s, the concern of the US security planners has moved more and more toward the OOTW/OMO areas. Prior to that, even given the arguments of the previous paragraph, the community had focused on a significant strategic war with the former Soviet Union. Combatant Commands and the Military Departments are rapidly gathering lessons learned documents, writing doctrine, and conducting exercises. All of this activity to gain an understanding of OOTW/OMO.

This analysis is currently unfocused and occurring in a very decentralized way. Unlike the combat and warfighting processes which still demand complex analysis by thousands of scientists and engineers using well understood computer simulations and exercises, there is little directed or focused analysis of the new missions. Unfortunately, some of the analysis that is being done is being done with the old combat simulations. Using the old simulations for the new missions is potentially misleading and dangerous.

Process:

This paper describes how a collection of tools may be designed to partially remedy this situation. This design, however, needs to be developed carefully in this new arena. Systems development can proceed along many different paths. Unfortunately, the most common path is the ad-hoc development process. This occurs when a manager or commander states a need: "I think we need a tool kit." This may turn out to be the complete statement of the requirement. It then becomes necessary to 'flesh out' the requirement in order to specify a design. In fact, the initial

statement is usually proffered in the form of a solution. The manager or commander sees a need, and jumps to a potential solution to the problem. This premature statement of a solution has caused many problems. Not the least of which is artificial narrowing of the solution space. Developers are biased by the statement, and self-limit their engineering process.

The systems processes today stem mostly from the software engineering domain. Terms like Waterfall and Spiral are most often heard. These are variations on the top - down and bottom up approaches. In top down approaches, a high level requirement is 'discovered' and then decomposed until components can be identified. Bottom up approaches, as many object oriented approaches devolve to, begin by identifying the lowest level of component and reverse engineering a design.

The conceptual modeling approach begins by recognizing the strength of both of these techniques and adding to them. Figure 1 shows the approach which begins with a high level model, discovered in a process of increasing participatory design. Then the process cycles down to the existing modules to identify constraints and interfaces, as well as possible reusable components (sometimes called 'golden nuggets'). This high/low process continues: Down in the problem space of the domain model, and up in the solution space of existing components. Essentially, this process allows the engineer/developer/manager team to use the lower level requirements to better understand the higher level requirements.

Tool Kit Development Process

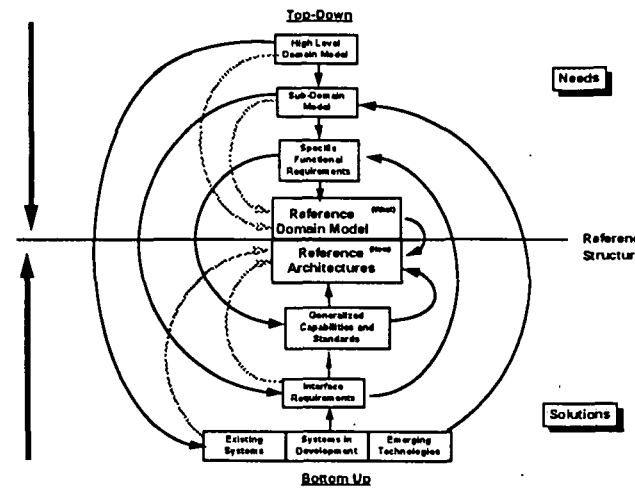

Figure 1
The Whirly-Gig

The top level approach, in what is called the problem domain, is centered on the description of the process to be modeled. It is key during this phase that the developers not try to duplicate existing systems or procedures, but attempt to describe what needs to be done. This enforced independence of existing systems will allow the team to focus on the issue to be solved. Identifying the 'what' of what needs to be done, without contamination from the 'how' of how to do it. The result of the top level model will be a statement of the true system, from which requirements for some sub-domain can be (almost) directly derived.

This derivation depends on the acceptability of the requirements, which depends on the comprehensives of the model. Thus the model must lead the community to some level of consensus. One method of developing and working

toward consensus is that of behavioral modeling.

Example:

As a simple example, we have started the development of a conceptual model of peace operations, as a sub-domain of Operations Other Than War. The behavioral diagram of Figure 2 represents the process of Peace Operations in a loop. The loop can be exited by the outbreak of Peace or War, or a simple Withdrawal of the force. Otherwise, if a resolution is still deemed possible, the process simply continues.

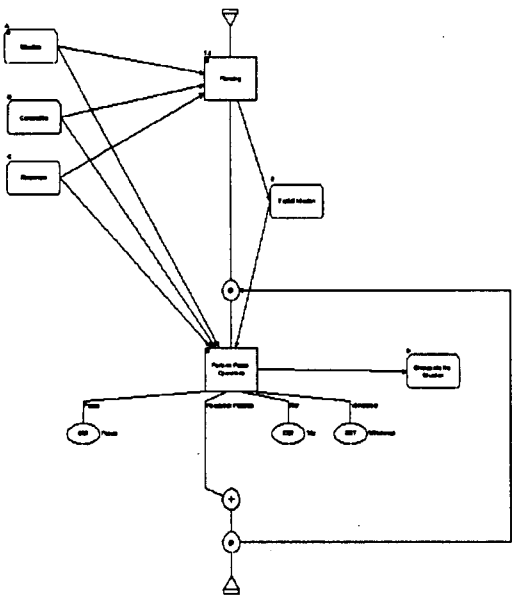

Figure 2
Peace Operations Domain Model

The operation depends upon some sort of planning or command and control process. The efficacy of this planning process may be a point of contention today, but it remains that some planning is done, and a mission is formed.

The internal representation of this C^2 process will be derived from the joint George Mason University/CECOM Software Engineering Directorate's Conceptual Model of Command and Control ($CModC^2$). The $CModC^2$ is a domain model in behavioral form that has seen wide usage in the explanation of C^2 requirements. This model of command and control has undergone continued development and validation effort for several years. The validation has been subjective, command expertise, and objective engineering consistency evaluation.

The operations element of this potential domain model has been provisionally decomposed. One, straw-man, decomposition of the Peace Operations (Figure 3) process shows that some sort of activity is generated based on the inputs of situation, constraints, resources, and mission. Some of these activities will directly effect the parties of interest, some will be coordinated prior to causing the effect.

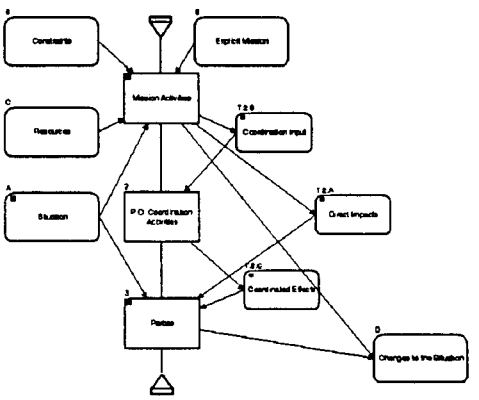

Figure 3
Peace Operations Decomposition

These processes can be further decomposed into other activities. The activities are seen as Military, Political, and Humanitarian. The parties are decomposed into parties to the conflict and external parties. Further decomposition and analysis will lead to the finer definition of what functions are required, and what information is used by and produced by those functions.

Modeling:

These diagrams are developed in an ordered process. Initially, a small group (1- 4) analysts and domain experts jointly develop a straw man that is iteratively improved over a period of time. When sufficient detail (currently a subjective judgment) is modeled, a larger group (10 - 15) of domain experts are gathered for a structured workshop on the model. This workshop is conducted with the goal of providing more detail of decomposition in the processes of the domain model, and of re-validating the current model. The members of the workshop should have previous exposure to the model such as soft copy or Web documents for easy of navigation and browsing.

The resultant model is then made available to the system's developer and to the community at large. The developer may begin to allocate functions to components and start the architecture/design phase. The community at large will comment on the model for update -- but most importantly, to understand what the system is being designed to accomplish.

Once the model has been developed to the point of web exposure, it can be used in the development process of systems. The model should, by this point, incorporate the entire set of domain requirements. The lowest level of model decomposition may then be subjected to an analysis to determine what, if any, current systems exist that provide the needed functionality. Functions that are not provided by current systems, may then be identified, and allocated to a new design effort.

Management:

The usefulness of the model goes beyond the engineering processes of requirements management and design. The original commanders and managers who stated the need, may use the model to refine their understanding and to develop insights into the process. By tracing through the levels of decomposition, and following the information flows, critical processes and functions may be identified for closer surveillance and evaluation.

One of the manager's largest problems is the comprehension of large, complex systems under his or her cognizance. By providing the manager with a browsable, graphical representation of these systems, the manager is given a significant tool.