GAMMA (Global Aggregated Model for Military Assessment) – DESIGN AND FUNCTIONALITY

Dr. Uwe K.J. Dompke,
Dr. Wolfgang Nonnenmacher
NC3A ORFS Division
P.O. Box 174
2501 CD Den Haag
The Netherlands

Stephen Yates

Newman & Spurr Consultancy Ltd
1 Pembroke Broadway
Camberley, GU15 3XD
United Kingdom

ABSTRACT

GAMMA is a simulation system developed at the North Atlantic Treaty Organisation (NATO) Consultation, Command and Control Agency (NC3A) in The Hague. It is used as an analysis tool in exercises on the operational level as well as in support of real operations (e.g. in the Balkans).

Using the baseline prototype the simulation was reengineered to follow software engineering best practice using an Object-Oriented Computer Aided Software Engineering (CASE) tool. The key features of GAMMA are its modularity and applicability to a wide range of scenarios from peace keeping and support, crisis response, anti-terrorism through to conventional war gaming. The paper will give an overview of the system design and its major software components including the handling of intelligent agents to autonomously represent both passive and aggressive military and non-military entities.

ABOUT THE AUTHORS

Uwe K.J. Dompke is currently Principal Scientist and Project Leader at the NATO C3 Agency in The Hague, The Netherlands. His main research areas are Human Behaviour Representation and Modelling and Simulation of Crisis Response Operations. He is Director of the Lecture Series on "Modelling of and for Military Decision Making" of the Research and Technology Organization of NATO. He has directed NATO Research and Technology Board Long Term Scientific Studies (LTSS) on Computer Assisted Exercises, Computer Generated Forces and Human Behaviour Representation in the last 10 years. He has been Co-Chairman of the Simulation Interoperability Workshop (SIW) Human Behaviour Forum. He received his Dr. degree in Computer Science from University of Federal Armed Forces Munich in 1992.

Stephen Yates is currently a consultant working for Newman & Spurr Consultancy Ltd. He has provided support to model design and development at NC3A over the past two years and has been involved in development of military simulation software for 7 years. In addition to work on the GAMMA project, he has a particular interest in simulation of military operations in urban terrain.

Wolfgang Nonnenmacher was until recently Principal Scientist and Project Leader at the NATO C3 Agency in The Hague, The Netherlands. His main research areas are Human Behaviour Representation and Modelling and Simulation of Crisis Response Operations. He has developed the GAMMA prototype.

GAMMA (Global Aggregated Model for Military Assessment) – DESIGN AND FUNCTIONALITY

Dr. Uwe K.J. Dompke
Dr. Wolfgang Nonnenmacher
NC3A ORFS Division
P.O. Box 174
2501 CD Den Haag
The Netherlands

Stephen Yates

Newman & Spurr Consultancy Ltd
1 Pembroke Broadway
Camberley, GU15 3XD
United Kingdom

INTRODUCTION

NATO has established Guidelines for Operational Planning (GOP) which are made to support the Joint Multinational environment in which NATO operates. SHAPE (Supreme Headquarter Allied Powers Europe) has set up operational planning courses (individual training) to familiarize staff officers coming from the different nations with the NATO environment. These courses are supported by operational analysts from NATO Consultation Command and Control Agency (NC3A) and NATO and national headquarters. One main part of this Operational Analysis (OA) support is the analysis and evaluation of different courses of action. This is done under severe time constraints with a small team of people. GAMMA is a simulation model, which was designed to fulfil these requirements to support NATO's Guidelines for Operational Planning (GOP). This was first done for high intensity conflict scenarios.

Recent changes in world wide political and military groupings require NATO to maintain a capability for carrying out new types of operations as in the case of Bosnia or Kosovo. Consequently, operational analysis support tools have to keep up with these new challenges and be able to map and analyze the characteristics of the new types of scenarios, the most significant of which are:

- Strong interdependencies of military and non-military issues
- Multinational (including Non-NATO) operations
- Low intensity conflict
- Multi-faction conflicts
- Tasks other than (traditional) military tasks
- Severe political constraints on military (or non-military) operations

The concept of GAMMA is based on an open architecture, which describes all interacting objects such as military units, assets, geographic objects etc. in very general terms, so that new types of entities, for example new military unit types of all services or non-military elements such as refugees, civilian population, or civilian organisations (such as The Red Cross), infrastructure elements such as power plants or cities etc. can be defined and instantiated easily without requiring any program changes.

Unlike cold-war oriented "classical" simulation systems, GAMMA is not restricted to a fixed, two party friend-enemy scheme. Instead, any number of factions may be grouped into any number of coalitions, the mutual relations between them may vary from "allied", "friendly", "neutral", "suspect" to "hostile"; In a war fighting simulation, not only attrition caused on the combatants but also collateral damage on civilian objects can be calculated.

GAMMA is an event driven simulation model, which can be run both in an interactive (man in the loop) or a closed mode. In the closed mode, conditional orders can be given to each simulation entity. The conditional order concept is very helpful in setting up the environment for the comparison of different courses of actions in a given scenario, which is an essential step in NATO's Guidance for Operational Planning.

GAMMA as a prototype system already supports the above mentioned functionality. NC3A has started to develop a new GAMMA system based on the experience gained with the prototype. This system is built using state of the art software engineering practices (open architecture, component based design). The paper describes the design of the system and the agent based module.

DESIGN OF THE SYSTEM

GAMMA was developed as a prototype at the NATO C3 Agency using an object-oriented approach to support operational analysis in the planning phase of military missions. This prototype developed by Wolfgang Nonnenmacher will be in use until the reimplementation is finished (planned for 2nd quarter 2003). The new design of GAMMA is based on the functionality of this prototype and some of the kernel concepts.

Why Redesign?

As with many software projects within the research and academic community, GAMMA was developed from an initial concept to satisfy a recognised capability shortfall – namely that of rapid support of courses of action analysis and evaluation. Whilst GAMMA development was in progress the wider utility of the initial concept and the power of the approach was realised. To support the change in scenarios from high intensity war fighting to asymmetric war fighting and to peace support operations the functionality of GAMMA evolved. As with many projects of this type, it would not have been possible to capture these as functional requirements at the model concept stage as the concept had neither been proven nor explored. This led to a series of short application development cycles during which a wide range of functional enhancements were made to the model. This resulted in a system which, whilst being functionally correct and capable, lacked software structure and documentation. In particular, code was replicated in many units within the application and functionality was not encapsulated into distinct packages therefore often requiring changes to many source files to implement minor functionality changes. The lack of documentation meant that tracing exactly where changes were required to further extend model functionality became more and more complicated.

It was recognised that to produce a robust, future-proof model would require either significant effort to restructure and document the original model or a rather more radical approach. It was decided that the more cost effective solution was to treat the original GAMMA model as a functional prototype and perform a full software redesign.

The objective of the software redesign was to produce a design functionally identical to the prototype but with continual attention paid to the application of software engineering best practice to establish a robust, flexible and future-proof architecture which allows the evolutionary development of new components to be integrated in the system. In this way it was possible to strictly adhere to the initial stages of a waterfall software development cycle without recourse to an iterative approach due to "requirements creep" or a desire to prematurely begin hands-on coding.

Use of Software Prototype

To follow the early stages of a waterfall development cycle, the redesign process was divided into analysis and design tasks. The analysis task involved the capture of user requirements. Since the user requirements had already been implemented in the prototype software, this task became a matter of exploring and documenting the functionality encapsulated in the prototype. As such, user input was restricted to confirmation of functional statements and approaches and vague statements of functional requirement were avoided. Besides the use of the GAMMA prototype for the functional analysis two other prototypes developed at NC3A (CASSANDRA (C3 Army Simulation System using Aggregated Networks for Defence Research and Analysis) for a C2 model and MARAIM (Maritime Artificial Intelligence Model)) were analysed to determine if some of their functionality should be covered in the new model.

Approach

To ensure flexibility and encapsulation of functionality, the design was conducted using Object-Oriented principles and was divided into three phases: abstract design, component design and concrete design. Broadly, the abstract design identified the major classes within the model and explored their interactions; the component design grouped these classes into functional areas to identify functional components and to define the boundaries between the components; the concrete design phase developed the abstract class definitions further to produce a series of class and activity diagrams for each functional component. This also included an element of User Interface and procedural level design.

During the whole analysis and design process Rational Rose© was used as the supporting CASE tool (see table 1).

Task	Rational Rose Environment
User requirements capture	Use Cases diagrams
Functional analysis	Activity diagrams
Abstract design	Class diagrams
Component design separation	Component diagrams
Concrete design	Class diagrams
Interface design	Delphi embedded in Rose
Procedural design	Activity and sequence diagrams

Table 1: Design Environment

Use Cases

Approximately 100 use cases were identified and documented. They are functionally grouped in separate modules:

- System Module

  The system module contains use cases associated with the general running of the system. The system module is the shell for the application, and is independent of the simulation and scenario.

- Display Modules

  The Display Module contains use cases associated with visualisation and manipulation of the map terrain and map display, with reference to how the map is displayed and the editing tools available within the system.

- Object Management Module

  The Object Management Module contains use cases associated with creating and editing objects within the scenario.

- Scenario Module

  The Scenario Module contains use cases associated with setting up the scenario.

  A scenario consists of map objects and elements, and the relationships of those elements – including command and control hierarchies.

- Strategy Module

  The strategy module provides the means to define interactive and closed strategies for acting elements. It also provides the possibility of defining general behaviour rules and supports independently acting elements (agents).

- Simulation Module

  The simulation module provides all functions of a classical event-driven simulation.

- Evaluation Module

  The Evaluation Module contains use cases associated with the evaluation and export of simulation results.

Abstract Design

From the use cases, an abstract view was constructed. The abstract view consists of high-level classes and constructs required to implement the recorded use cases (see figure 1 for the GAMMA object architecture from the prototype). This abstract view was broken down into components which are independent from each other and communicate with each other using well defined interfaces.

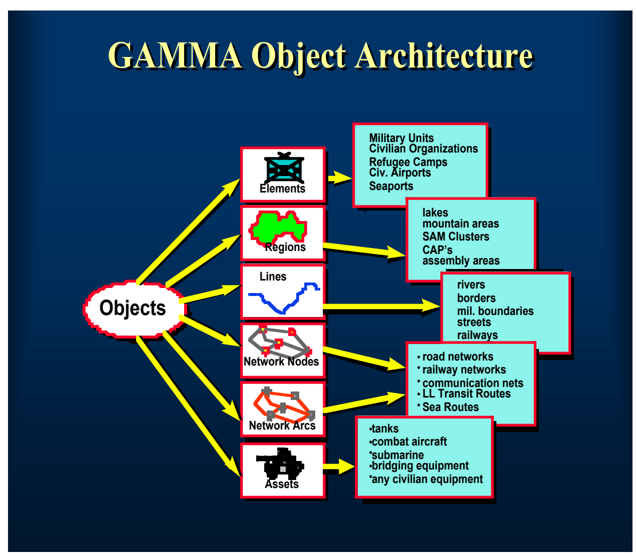

Figure 1: GAMMA Object Architecture

Interoperability

At the abstract design stage it was necessary to make a decision regarding interoperability of GAMMA with other simulations; in particular should GAMMA be designed to be High Level Architecture (HLA) compliant. It was agreed that to achieve the goals the HLA requirements should neither constrain nor force the design process. The abstract design was therefore conducted without direct adherence to the HLA (especially not linking the internal components using HLA). However, to increase flexibility for future use the decision was made to allow for HLA interoperability through the addition of an "HLA Interface" component.

Component Design

It was decided at the same time that the decision was made to perform a software redesign that to maintain flexibility and provide an open architecture for future development, a component based architecture would be adopted. Candidates included the Microsoft Common Object Model (COM), the Common Object Request Broker Architecture (CORBA) and Java-beans – the COM architecture was adopted for cost, performance and simplicity reasons. CORBA would have introduced additional licensing requirements for the broker software and the project team had little experience of using Java-beans for complex event-driven simulation and this approach was therefore considered to be a higher risk (and possibly higher cost) approach.

The component interfaces are defined in a collection of language independent GAMMA type libraries. The choice of a development language to implement GAMMA in becomes less important when component based architecture is used; all the main Object Oriented development languages support the COM framework and therefore each component has been designed language-independent and can be developed in the most appropriate language.

By clearly defining the published interfaces for each component during the development of the component level design it was possible to develop an architecture that will support variable component configurations and will allow for insertion of additional components in future with little or no impact on existing components. For example, it will be possible to operate the GAMMA model with either a Lanchester or stochastic combat resolution model simply by reconfiguring the system to use a different component delivered as a COM Dynamic Link Library (DLL). Historically in other simulations, changes of this nature have required changes to many different parts of the system and, at least, require the software model to be recompiled.

The derived components operate together to form the GAMMA model. They can be grouped in following categories:

- Administration and Application
  - Simulation Manager
  - Application
  - System Utilities
  - Help
- Graphical User Interface
  - Graphical Display System
  - Coordinate Manipulation
- Object Management
  - Object Store
  - Gamma Annotation
  - Gamma Element
  - Gamma Networks
  - Gamma Region
- Statistics
- Data Provision
- Event Handling Models
  - Collateral Damage Model
  - Constraints Model
  - Incident Model
  - Intelligence Model
  - Lanchester based Interaction Model
  - Mission Phases Model
  - Movement Model
- Order System
  - Order Interpreter
  - Order Interface

Concrete Design

The concrete design phase involved taking the abstract design in its component form and producing a series of class diagrams which would form the basis of the implementation classes of that component. The intention was to capture all classes within the component, not just those related to core system functionality this then included classes to manage menu options, logging and user interface issues.

The objective of the concrete design phase was to produce class diagrams for which the Rational Rose© code generation facility could be used to produce the majority of the implementation code. In reality, this has not always been possible and during implementation the concrete level design of some components has varied slightly as implementation specific issues are addressed. The concrete level design has therefore become a high-level design for the individual components, with the component design and interface definitions forming a system architecture definition.

Lessons Learned

Having completed the design stage it is now possible to look back and assess to what extent the process met its objectives. To recap, the objective of the software redesign was to produce a design functionally identical to the prototype but with continual attention paid to the application of software engineering best practice to establish a robust, flexible and future-proof architecture which allows an evolutionary growth. This approach to cover new ideas and possible future extensions already in the design phase proved to be cost-effective.

During implementation, the component interfaces have been modified slightly for implementation reasons, the majority of the design (80-90%) has been carried through to implementation unchanged. This has only been possible through thorough requirements capture and analysis, and for this type of research development the requirements capture process has been significantly enhanced through the use of the functional prototype.

The development of the initial series of COM components within the GAMMA architecture has been a success with only minor changes to the architecture design and this has proved that the architecture will support this particular application. Furthermore, by adopting a COM component based approach the architecture retains flexibility by allowing addition and modification of individual components as required and to meet emerging future needs. The architecture will also therefore be future-proof in as far as the underlying component technology is.

In summary, the redesign process has been a success resulting in a well structured and documented design which is currently being taken to implementation. This has only been possible through the development of the functional prototype software.

INDEPENDENTLY ACTING AGENTS

New multi-sided low intensity conflict scenarios are now used in NATO and in the nations in exercise and training as well as for analysis. These kinds of scenarios need a different simulation support than attrition based high intensity conflicts. An independently acting agents model was integrated in GAMMA to give support for such scenarios. This approach is followed describing an application of GAMMA to support an exercise.

The Scenario

GAMMA was used in exercises which start in a post-conflict situation with a scenario in which a military conflict was ended by intervention of Alliance Forces. A demilitarised zone had been established on both sides of the border. Compliance with the requirements of demilitarisation within a buffer zone had to be monitored by the Alliance Force.

In the buffer zone, terrorist and paramilitary groups were creating a variety of incidents, such as terrorising the civilian population of opposite affiliation, attacking refugee treks, which were crossing the buffer zone, sniper ambushes, robberies and so on. In addition, groups of local inhabitants (in particular peasants) which were dissatisfied with the current situation were organising demonstrations and thus creating confusion and disturbance.

Apart from monitoring the compliance of the demilitarization agreement, the Allied Forces were tasked to control and reduce those incidents as far as possible.

The challenge for the GAMMA model was to find a way to represent the essential constituents of this situation properly. One of the expected outcomes of the simulation was an assessment of the Allied Forces effectiveness.

Representation of the Scenario in GAMMA

In the described scenario, three types of elements are acting in the given environment:

- Allied Forces (e.g. reconnaissance platoons)
- Terrorist /paramilitary groups
- Local civilian population.

There are three prerequisites in order to regard an element as an agent within the simulation:

1. It must have some internal goals (independent of the current environment)

2. It must have a memory function in order to incorporate past events into its strategic reasoning

3. It must be able to react to the environment, e. g. to include environmental influences into its reasoning

To represent the scenario in GAMMA terrorist /paramilitary groups and groups of local civilian population are represented as agents. They have their own goals, a memory function and interact with the environment. The Allied Forces are controlled by conditional rules, which can also use a memory function as parameter of the forces.

Agents in GAMMA are elements with special properties which enable them to act independently (see Figure 2).

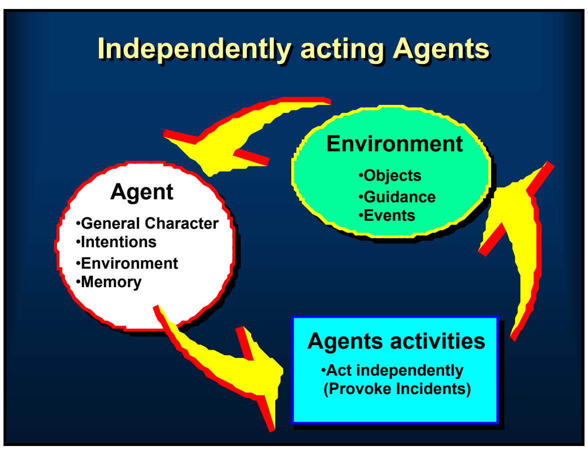

Figure 2: Independently acting agents

Whether or not and how agents act depends upon:

- their general character
- their intentions
- their current state and
- the environment.

The agent's general character determines what kind of activities he is ready to perform in principle. This general character is described by:

- Morality level
  (0 <= moral level <= 1; 0 = criminal, 1 = saint)
- Readiness to risk

Their intentions are described by:

- political interest
- military interest
- economic interest
- ecological interest
- psychological interest
- interest in serving own benefit

The current state is described by the agitation level as readiness to initiate activity. Each activity initiated by an agent presumes that the respective agent has an agitation level that allows him to perform such an activity. The needed agitation level to do a bomb attack is e.g. higher than the one needed to initiate a demonstration. The agitation level of an agent is dependant on his activities he has done in the past (e.g. after doing a bomb attack his agitation level will go down and it takes some time that he will recover to build up his agitation level again) and dependant on events that happened to him in the past like meetings with patrols.

Possible Incidents

Agents can create certain type of incidents, which have a specific type to be defined for a specific scenario. Examples are:

- Demonstration
- Public riot
- Occupation of media or embassy
- Robbery
- Destruction of private property
- Destruction of military infrastructure
- Sniper attack
- Bomb attack

Incidents have a target object which can be:

- Cities
- Civil persons
- Buildings
- Industrial plants
- Military elements

With each incident a moral and agitation level which must be met by an agent to provoke an incident are combined. In addition for each incident an impact on the political, military, economical, ecological, and psychological situation is given which is used to determine the interest of the agent and the outcome of the incident on the situation.

As well as these internal influences the actions of other parties like patrols and checkpoints also have an impact on the memory of the agents and the generation of incidents in a specific situation. The model works on the following basic assumptions and descriptions of the elements in a way that for each step:

- the movement of the agent to a specific location where he can act in regard to his interest is determined and
- the potential generation of incidents when an agent arrives at some location is checked, taking his characteristics and the environment (e.g. patrols or checkpoints) into account

Calculating Priorities for Incident Types

If an agent is in the process of creating an incident, he selects from the list of possible incident types those

- for which suitable victims are available,
- the risk is below his accepted risk threshold,
- the moral quality is equal or higher than his moral level,
- the required agitation level is equal or less than his current one.

From the remaining list of incident types which meet all 4 criteria, he selects the one which best serves his goals, by calculating a priority for each and selecting the one with the highest priority.

The priority for an incident of type **i** is calculated according to this formula:

$$\mathbf{P(i) = relation} * \sum_{j=1}^{j\,=\,\text{Size of goal/impact vector}} \mathbf{goal(j)*impact(j)}$$

relation describes the relation between agent and victim, where

-2	=	**hostile**
-1	=	**suspect**
1	=	**neutral or friendly**
2	=	**allied**

In an example for this process an agent has following properties[1]:

Agent	Bandit
side	Orange
moral level	0.500
readiness to risk	0.300
agitation level	0.950
…	

In the process of determining which kind of action the agent will perform, it is first checked in which environment the agent is and what possible target objects in this situation are available. Then it is checked for which actions his readiness for risk is sufficient.

Table 2 shows as an example certain type of incidents with the respective values for the target object type, the moral level and the agitation level.

[1] The properties used in GAMMA at the moment are based on some basic assumptions which are derived from an information exchange with subject matter experts. The results from simulation runs with GAMMA using this data are accepted by these experts. In the future more work is needed to get more detailed data from the experts (e.g. psychologists). The experience gained shows that GAMMA can serve as a tool to explore ways to model human behavior.

Type of incident	Target object type	Moral level	Agitation level
Kidnapping	Civil population	0,3	0,8
Robbery	Civil population	0,4	0,6
Attacking soldiers	Military	0,3	0,8
Sniper attacks	Military	0,2	0,5
Destruction of private property	Civil population	0,4	0,6
Murder of selected persons	Civil population	0,2	0,8
Murder of any persons	Civil population	0,1	0,8
Demonstration	Civil population	0,9	0,8
Destruction of civilian infrastructure	Civil population	0,5	0,8

Table 2: Incidents

An agent with the properties shown would exclude for moral reasons (his moral level is higher than the moral threshold for the respective action, e.g. for kidnapping moral level should be below 0.4):

- Kidnapping
- Robbery
- Attacking soldiers
- Sniper attacks
- Destruction of private property
- Murder of selected persons
- Murder of any persons

In this example demonstration and destruction of civilian infrastructure would remain as possible actions. Demonstration would have mainly a political impact whereas destruction of civilian infrastructure has also a military and economic impact. For target objects with hostile relation, he would calculate the following priorities based on the specific impact given for the type of incidents

Demonstration	1.98
Destruction of civilian infrastructure	1.20

Thus, he would prefer a demonstration rather than destruction of civilian infrastructure.

Lessons Learned

The approach chosen for the independently acting agents was used several times to support Peace Support Operation (PSO) exercises with analysis support. It was not the aim of the use of GAMMA to generate message traffic on tactical level, GAMMA is used to support operational level exercises. On that level GAMMA could help to identify trends in the generation of incidents by different agents in response to the planning of the Alliance Forces (e.g. checkpoints and patrols).

The model provides at the moment no response to psychological operations, but NC3A is working on a module in that direction.

Besides the use in exercises first steps were made to support requirements reviews with GAMMA. The agents module could be used to analyze issues regarding size and layout of PSO forces.

Verification and Validation (V&V)

The prototype of GAMMA was developed in an evolutionary manner. At each stage in the development reviews were conducted to confirm that the model meets the emerging requirements. The basis for verification issues in the new implementation of GAMMA is the requirements review documented using Rational Rose©.

The GAMMA prototype has been informally validated in a two step process:

1. The simulation results gained by using GAMMA have been compared and adjusted with other studies and models used at NC3A.

2. The exercises (especially with the 1 – 3 star generals and with the operational planners from the Operational Planning Groups (OPG) at the Headquarters) have been used as a forum to gather intensive feedback from the audience based on their profound military judgement.

SUMMARY

Experience in supporting military planning at the operational level showed that existing analysis tools for rapid support of operational analysis in conventional and new tasks like Crisis Response Operations (CRO) are insufficient. The NATO Consultation, Command and Control Agency (NC3A) in The Hague has developed a prototype of the simulation model GAMMA to fill this gap. The experiences gained with this prototype have been included in the design of a new component-based version. The design of the new system followed "best practice" in software engineering.

GAMMA includes a model to represent, among other things, terrorist/paramilitary activities based on an intelligent agent approach. Each agent is described by its intentions, his morality and agitation level and is able to act independently. This model also allows the investigation of the impact on guerrilla activities of military actions such as patrolling and checkpoints.

The prototype of GAMMA is informally validated based on expert knowledge. The new implementation will be based on this experience whereas the new code will be verified against the given requirements.

The prototype is still used in supporting exercises. The new system is now in the implementation phase and will be ready in 2003.

REFERENCES

[1] S.F. Black, S.J. Yates: "GAMMA Analysis and Redesign: Baseline Functional Analysis", NSC-531-009, Camberley, September 2001

[2] S.F. Black, S.J. Yates: "GAMMA Analysis and Redesign: Design Report", NSC-531-036, Camberley, October 2001

[3] Uwe K. J. Dompke et al: "Final Report on NATO Long Term Scientific Study on Human Behaviour Representation", RTO-TR-047, NATO Research and Technology Organization (RTO), Neuilly-sur-Seine, France, July 2001

[4] NATO Bi-SC Document: "Guidelines for Operational Planning (GOP)", December 1999

[5] U. Dompke, W. Nonnenmacher: "GAMMA (Global Aggregated Model for Military Assessment) – Design, Intelligent Agents and Conflict Analysis Module", 11th CGF and Human Behavior Representation Conference, Orlando, FL, May 2002