Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

THE MODELLING OF TACTICS AND PROCEDURES USING A COMPONENT BASED SYSTEM

Phillip Martin

CORDA Ltd.
Apex Tower, 7 High Street
New Malden, Surrey, KT3 4LH
UNITED KINGDOM

ABSTRACT

The development of tactics or procedures is a multi-stage, iterative process from the initial concept to their use by operational units. Once developed tactics are modified in response to changes to the targets, sensors or weapons, or modified in response to operational constraints. Modelling is an important tool in this development process being used to quantify performance metrics, to extrapolate from existing conditions or to highlight areas for more detailed investigation in simulators, trials, etc.

This paper describes a methodology for modelling tactics and procedures using a component-based system written in EXTEND ™ from Imagine That, Inc. This methodology allows the analyst or tactical developer to make changes to the tactics without the need to expose the underlying code. Tactics are built up from a library of generic components that are ordered and connected using graphical methods and the components can be customised using dialogue-input parameters. These tactics are used to control the actions of units within a simulation that demonstrates the effectiveness of the tactics and provides the quantitative data for comparison.

1 INTRODUCTION

A common problem of simulations is that they are designed to model the performance of systems of which tactics are an input, and they are not designed to study alternative tactics. The tactics are included in the model code and as the simulations progress the tactics are stepped through, often driven by parameters but essentially the structure of the tactics are fixed. Outside of user defined parameters, changing the tactics means altering the code with the inherent overhead of testing, validation, etc.

This paper describes a concept for a tactical development system that allows the analyst to study the tactics, to vary the order of the steps etc. without having to break open the model each time. The analyst need only be computer literate to alter the structure of the tactics. The analyst does not have to be a computer programmer.

2 WHAT IS A TACTIC?

The term tactics may mean different things to different people. In this paper a tactic is a series of instructions and or processes which are stepped through, driven by external stimuli. The path taken through these steps may vary according to the particular stimuli present or the order of the stimuli, but all the options are defined prior to the simulation. The tactic is the set of instructions or the orders to be followed.

3 THE CONCEPT

The modelling system is developed using the simulation environment EXTEND ™ by Imagine That, Inc. (1997) EXTEND ™ is an environment in which models are built up using pre-defined modules or blocks of code from a series of libraries. These blocks are linked together using drag and click graphical methods. The concept described in this paper uses the basic constructs of EXTEND ™ to develop tactical building blocks which can be connected together to produce the tactical procedures required.

There are effectively two parts to the concept, albeit they work together. The first is an underlying simulation in which the effects of the tactics can be demonstrated by the movement and interaction of real world units. The second is a module contained within the models of each of these real world units where tactics are constructed using specialist tactical blocks. Such a module take inputs from the objects within the unit which act as stimuli for the tactical decisions and output instructions directing the movement and the function of the unit’s systems.

4 THE UNDERLYING SIMULATION

The models have been created in EXTEND ™, using a combination of the standard library blocks and blocks specifically written to facilitate aspects of the simulation. Although the simulation uses discrete event and continuous blocks, it operates as a discrete event simulation. The EXTEND ™ user manual describes the use of continuous and discrete event models.

The simulation is essentially a representation of free space, the world, into which Real World Units are placed and as the simulation progresses the units move about this free space.

Within the design of the model the ownership of methods and attributes has been selected to maximise the flexibility of simulation. To this end the only limitation to the type and number of real world units is the computer memory and the acceptable run time.

4.1 Real World Units

A model of a Real World Unit is a hierarchy or group of objects that define the functionality of that Real World Unit. An example of a Real World Unit is a helicopter, consisting of:

a) A dynamic object, which controls the way the helicopter moves.
b) An active sonar object which represents this particular sensor.
c) A radar object providing the electromagnetic contacts
d) A fuel system object, which monitors the fuel state and fuel usage.
e) A sonobuoy ejector object, which creates and places sonobuoys into the simulation.

These objects are linked using real world parameters (speed, heading) and command messages (turn left, stop) such that, as the simulation progresses they describe the behaviour of the unit.

In addition, each unit has a Tactical Module, which controls the behaviour of the unit and defines how it reacts with the outside world. This structure is illustrated at Figure 1. The structure of the blocks within the Tactical Module is described in Section 5.

The simulation can accommodate any number of these Real World Units; in the air, on the surface, or below the surface. These units are all independent of each other and they are only aware of other units through the use of their sensors.

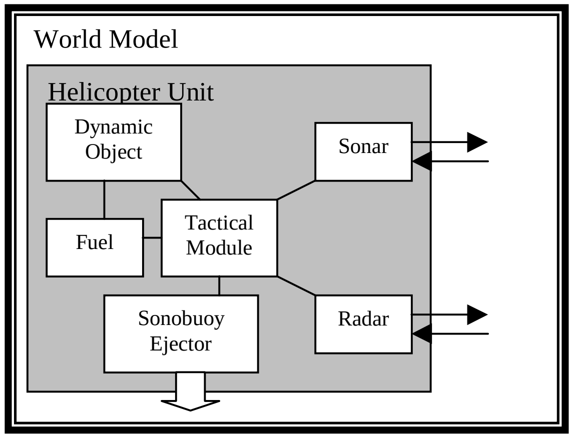

Figure 1: The Unit is a Hierarchy of Self-contained Objects

4.2 Sensor Environments

The simulation contains a number of environments that represent the various media, through which the sensors operate, see Figure 2. For example:

a) Sonar environment in which active and passive sonars operate.
b) Radar environment for the propagation of signals to and from electromagnetic sensors.

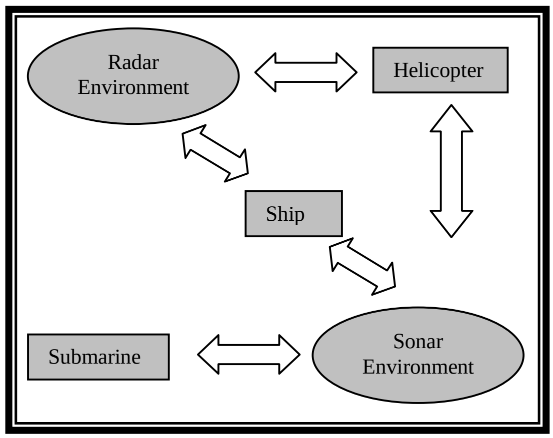

Figure 2: Units Interact Using Their Sensors Through Environments.

The structure of these environments is similar, and is shown at Figure 3, and can deal with:

a) Active sensors, which radiate into the environment and process return reflections
b) Reflectors, that reflect the radiation
c) Modifiers that return a modified signal
d) Passive receivers, that receive the signals
e) Generators, that generate radiation/ noise continuously.

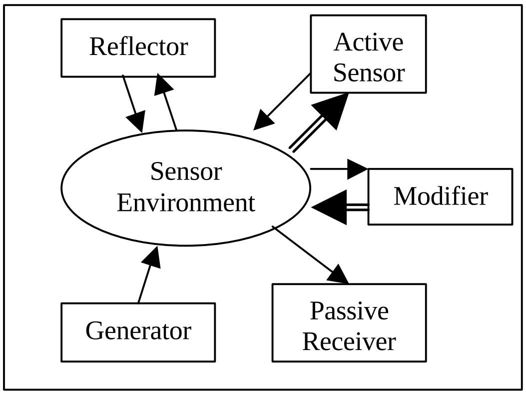

Figure 3: Active and Passive Sensors Link to Sensor Environments

The various unit objects log into and out of these sensor environments as the simulation unfolds. The environment keeps control of these objects, receives any signal inputs and generates the transmission of the signals to the other units, including any propagation effects. Various parameters are calculated and can be passed along with the signal, for example:

a) Range
b) Bearing
c) Doppler velocity
d) Frequency
e) Background noise

This information can then be used by the receiving unit to process the signal.

4.3 Physical Environment

There is another environment, the physical environment, which is not a sensor environment but maintains the true positions of the various units. This environment calculates and records the movement of all the units within the free space. The sensor environments access the physical environment's information during the calculation of their parameters but do not alter the data.

The units can also access the data, Figure 4. These data can also have errors applied appropriate to the equipment being modelled.

5 THE TACTICAL REPRESENTATION

The tactics controlling the behaviour of a real world unit are contained within its Tactical Module. The aim of the development described in this paper was to provide a simple but robust method by which an analyst could investigate and develop tactics.

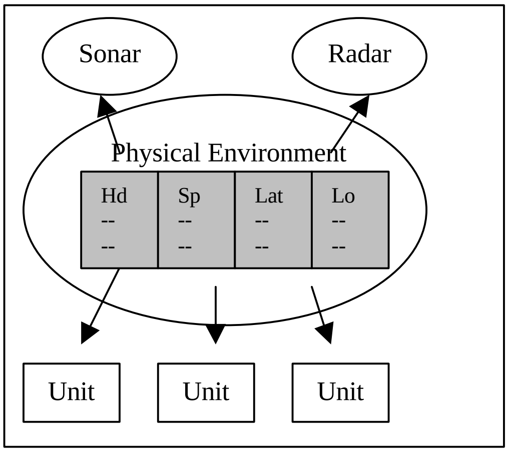

Figure 4: The Physical Environment Maintains the Positions the Real World Units

Tactics are essentially a combination of simple tasks that are linked together and react to external parameters and inputs. Tactics are broken down into component tasks in such a way as to provide a set of generic building blocks from which all tactics can be created. Within EXTEND ™, a library of these generic blocks can be produced which are connected together within the Tactical Module using the graphical linking methods provided.

In EXTEND ™ discrete event simulation mode, "items" are passed between blocks that act upon these items or undertake particular tasks on receipt of the item. The items are created by a Generator Block and are passed from one block to another until they reach the Exit Block where the items are removed from the simulation, Figure 5. There is a distinct path that the items follow, defined by the way the blocks are linked. All the paths are defined before the model is run.

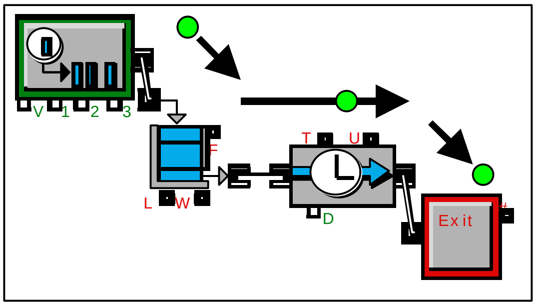

Figure 5: Defined Flow of Items

With this development, the method of generating items and passing them between blocks has been modified to allow more flexibility within the Tactical Module and achieve the development aim.

In the Tactical Module only one item is generated and is passed between the Tactical Blocks like a relay baton. At any one time the block with the item is the part of the tactic which is being performed. This Tactical Item remains within the Tactical Module throughout the simulation. It can go backwards and forwards between blocks as required by the simulation. The order in which the Tactical Item is passed between the blocks is not set before the model is run. The order is defined as the simulation progresses and is controlled by the way the unit reacts to external stimuli. The Tactical Blocks can be grouped into three distinct type:

a) Connector Blocks
b) Performer Blocks
c) Decision Blocks

5.1 Connector Blocks

There is only one block in this block type, which controls the movement of the Tactical Item between the Performer Blocks. It maintains a record of the Performer Block available within the tactic and passes the item messages between them. It acts as the conduit for the tactic in a similar fashion to a computer bus.

This block has a finite number of block connections, but multiple blocks may be connected together to provide as many connections as are required, Figure 6.

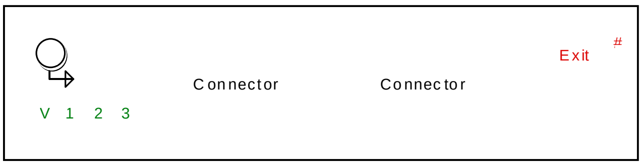

Figure 6: Multiple Connectors Create Sufficient Connections

5.2 Performer Blocks

These blocks are connected to the Connector Block and perform particular parts of the tactic, Figure 7.

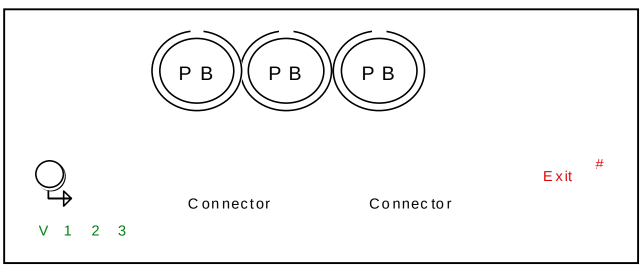

Figure 7: Performer Blocks Controlled by the Connector Block

On receipt of the Tactical Item, these blocks perform their particular task until it is removed by the Connector Block. These blocks are designed to know nothing about the rest of the tactic. They react only to the Tactical Item. However, they may have value inputs that are used within their task, and they may have outputs that could trigger Decision Blocks. These Performer Blocks all have dialogue windows in which the analyst may select options or input parameters, Figure 8.

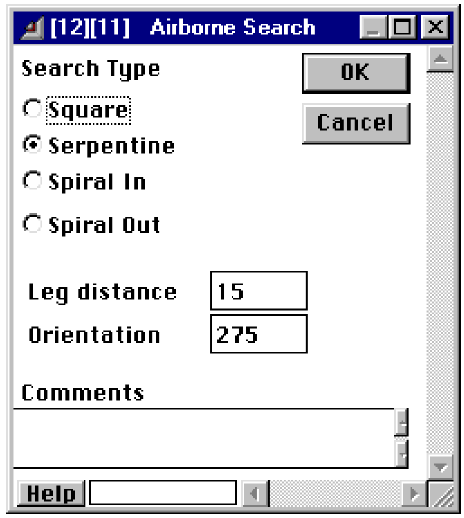

Figure 8: Dialogue Options Modify the Block Performance

In the example above, the Performer Block will undertake particular search patterns. The search pattern is selected in the dialogue using the radio buttons and for each choice particular additional information is requested.

5.3 Decision Blocks

These blocks are the companions to the Performer Blocks. They are unaware of where the Tactical Item is and have no relationship to where in the tactic's logic the simulation may be at any one time. The Decision Blocks only react to particular combinations of stimuli to direct the Connector Block to place the item with a particular Performer Block. The stimuli could be from a change in the Tactical Item parameters or an external / positional stimuli or a stimulus from a Performer Block, Figure 9.

Most Decision Blocks are likely to be instances of a general Decision Block. This would have a structure that allows the analyst to use equation like statements:

On {*1st input*} {>} {*2nd Input or constant*}
Then
Tactical Item Goes to {*Performer No2*}

The analyst will be able to select the variables {in the brackets} from within the block dialogue. The available options will be presented in the dialogue as pop-up menus.

Custom or bespoke decisions will react in the same way directing the connector to move the Tactical Item. Custom decisions may well be tied to specific performer output stimuli. It would be possible for the decisions not to

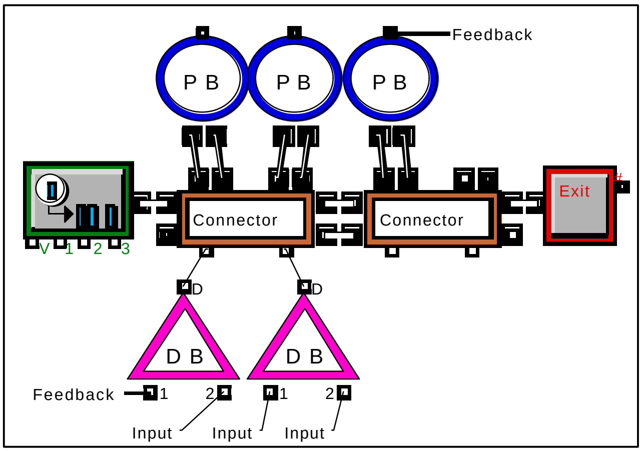

Figure 9: Decision Blocks Control the Movement of the Tactical Item

move the Tactical Item but to provide an output stimulus for processing by another Decision Block. The exact structure of any particular tactic would be at the discretion of the analyst. The concept is to provide as generic a set of blocks as possible to maximise the reusability of the library.

6 RUNNING THE SIMULATION

The simulation is capable of being run in single or batch mode, running as an ordinary EXTEND ™ Programme.

The simulation workspace window is the pictorial representation of free space defined by the model. A scaling factor is applied as a user input to convert the window pixels into linear units (Yards Nmiles etc).

Real World Units are pasted onto this workspace from the specific library, Figure 10, and their positions on the screen define their start positions. The units can be defined as mobile so that they move about the workspace as the simulation progresses, providing a visual representation of the effect of the tactics.

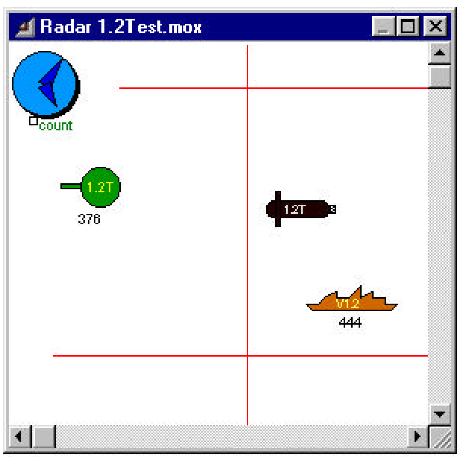

Figure 10: Multiple Units Placed into the Model Window

The units have a number of built in outputs and data gathering devices that can be used to provide more quantifiable data for the run, Figures 11 and 12. Statistical information can be gathered for multiple or batch runs.

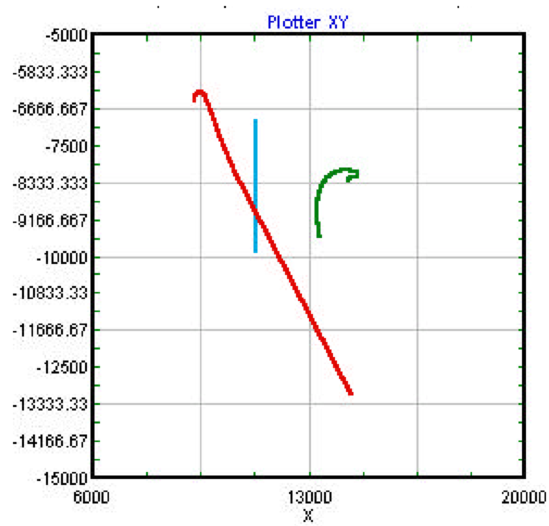

Figure 11: Track Plot of Helicopter Attack

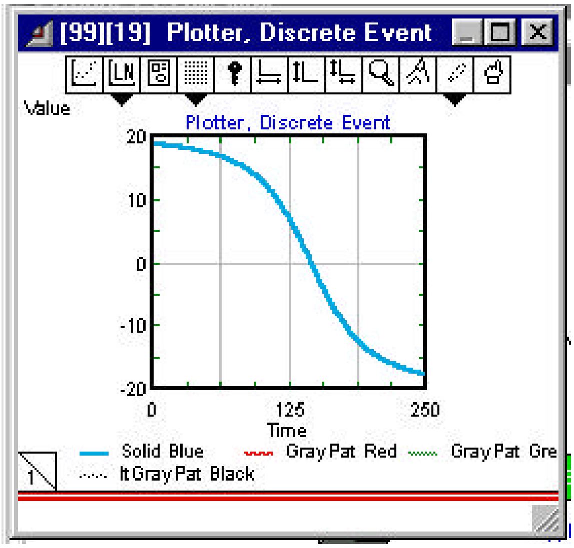

Figure 12: Doppler Velocity of Passing Submarine

As the particular metrics to be gathered will vary depending upon the nature of the tactics being investigated, it is impossible to cover all possibilities. However, EXTEND ™ has a number of data gathering methods which facilitate this collection and can be included by the analyst.

7 CONCLUSIONS

The system described in this paper provides a method by which the analyst can modify and investigate tactics in a controlled and timely manner.

Tactical building blocks from pre-defined libraries are used to construct tactics that control the actions of Real World Units within a companion simulation.

The simulation and the model of the tactics have been designed to be as flexible and as generic as possible, allowing the analyst to focus on the development of the tactics and not be hampered by having to modify the simulation.

8 WAY AHEAD

The tactical modelling system continues to be developed, to refine the concept and to build up the library of Performer and Decision Blocks. It is currently being assessed as a tool to aid the development of ASW detection and tracking tactics. Additional and differing tactical situations will be used to exercise the model further. It is anticipated that the range and complexity of the Real World Units will increase as the breadth of their use expands.

ACKNOWLEDGEMENTS

The author acknowledges Dr Colin Mason, Mr Terry Newman and Ms Sara Smithers of CORDA, for their help and support during the development of the concept and in the preparation of the paper.

REFERENCES

Imagine That, Inc. 1997 *Extend Software Manual*. San Jose CA.

AUTHOR BIOGRAPHY

PHILLIP MARTIN is a senior analyst and consultant within the Centre for Operational Research and Defence Analysis (CORDA), a subsidiary of British Aerospace. He is a graduate of Aeronautical Engineering from Bristol University, and has been an Operational Analyst for over twenty years. During this time he has been involved with projects for all the UK Armed Services. He is currently responsible for the front line analysis undertaken by CORDA staff for the UK MoD and was himself a front line analyst working for the Maritime Warfare Centre (Fleet OA Staff) for six years.