

THE EXCLUSION ZONE MODEL – A DEVELOPMENT METHODOLOGY

Paul Rendell

Air Warfare Centre (OAE)
Joint Headquarters Building (RAF)
Northwood
HA6 3EP
UNITED KINGDOM

Alan Cowdale

Air Warfare Centre
Thomson Building
RAF Waddington
Lincoln LN5 9NB
UNITED KINGDOM

ABSTRACT

This paper describes the application of prototyping and rapid development techniques to the support of real Operations when meeting a specific military requirement.

The Exclusion Zone Model (EZM) has been selected as a Case Study as it illustrates how the techniques can be applied, while highlighting the problems encountered by developers operating in a highly reactive environment.

1 INTRODUCTION

The Operational Analysis Element (OAE) of the Air Warfare Centre provides scientific support to front line units of the Royal Air Force, and consequently is required to develop, maintain and use simulation models and tools. Depending on the nature of the task, the models and tools can vary in size and complexity from detailed wargames requiring several days, if not weeks, to prepare and execute, down to simple spreadsheet models which can be prepared and utilised in minutes.

Many of the problems faced by military planners require responses and actions within very tight time-scales. Consequently, if simulations are to contribute to the decision making process, the models will usually need to have been developed, tested and configured to meet the requirements of the task well in advance. In practice, there will be occasions when the analyst does not have the necessary simulation tools immediately available. This paper describes one such occasion within the OAE, and chronicles the subsequent development of a model designed to aid the analysis of No-Fly Zones during coalition air operations, illustrating how simulation methodologies can be employed to develop models which meet customer requirements over a range of development time-scales.

2 MODEL 1 – SPREADSHEET MODEL AS A CALCULATOR

Simple spreadsheet models are usefully employed by the OAE as a precursor to larger more complex models by, for example, performing basic geometric calculations to validate input parameters.

For relatively straightforward problems, a spreadsheet may be used to provide the complete solution when considering issues such as:

- a) Is the distance travelled by an aircraft within its range ? If not would air-to-air refuelling assets be available ?
- b) Are alternative targets or divert airfields within range ?
- c) Will the overall duration of the sortie ensure that the mission is achieved in daylight or under cover of darkness ?
- d) Will the sortie be co-ordinated in time and space with other sorties associated with the mission ?

3 MODEL 2 – A CUSTOMISED SPREADSHEET MODEL

In 1997, the OAE were asked by air planning staff to devise a method to assess the effectiveness of No-Fly Zones. The general equations of 'Model 1' were used to provide a graphical plot of intercept positions for a scenario with known radar and defensive fighter locations and ingressing hostile aircraft. This first model was presented to the customer within 24 hours of the original tasking, Figures 1 and 2.

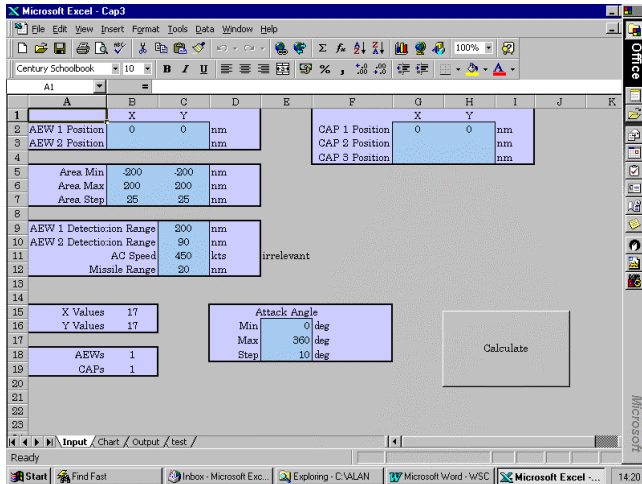


Figure 1: Data Entry Form of Model 2

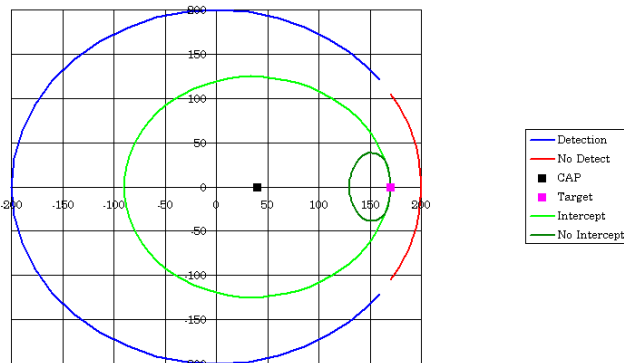


Figure 2: Graphical Output of Model 2

4 MODEL 3 – DEVELOPMENT OF A MORE GENERIC MODEL

The algorithm of Model 2 was extended to iterate through a large number of intercept positions. Consequently, it was possible to produce a contour plot showing the percentage of ingress geometries that could be intercepted by a given configuration of radars and fighters, see Figure 3. The OAE were required to generate some typical results for a briefing the following day.

In the limited time available, a Visual Basic model was developed with the capability of creating contour graphs for two radar sites and three fighter locations. The model was used to illustrate air defence coverage over an area of UK interest. This was well received by the sponsor. Consequently, the OAE were tasked to deliver a robust version of the model prior to a number of planning exercises scheduled during the next few months.

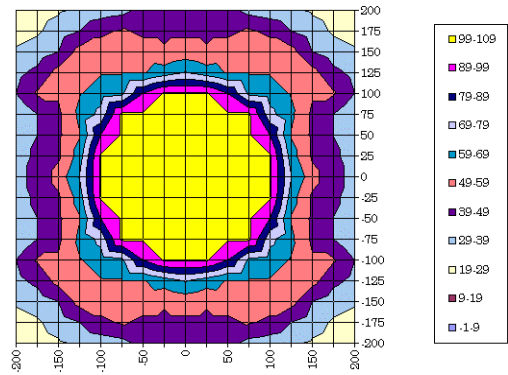


Figure 3: Graphical Output of Model 3 Illustrating the Percentage of Ingress Geometries That May be Intercepted

5 MODEL 4 – DEVELOPMENT OF DELIVERABLE DEMONSTRATION PRODUCT

In practice, considerable work was required in order to convert Model 3 into a tool of practical use. The output of this model required manual amendment to include symbology, titles and legends for ease of understanding. Such features were automatically included in Model 4, enabling the user to visualise input parameters and provide a suitably annotated output.

Additionally, by including the locations of airbases and a more detailed aircraft representation, the feasibility and sustainability of the proposed defensive formation could be assessed.

This was the first model iteration to be formally designed and tested and for which comprehensive documentation was produced.

Following delivery, the model was used as a planning aid during a number of exercises that took place in the early part of 1998.

6 MODEL 5 – THE REQUIREMENT

Although there were a number of enhancements that both the OAE and our sponsor would have liked to add to the model, the requirement to have a fast running model that could be deployed on a standard lap-top computer had prevented any further development in 1998.

Towards the end of 1998 the OAE took delivery of the next generation of lap-top computers with sufficient memory and clock-speed to make enhancing the model a practical proposition. The main enhancements that have been proposed for the model are:

- a) The provision of representative geographical mapping.
- b) The enhancement of the model from a purely air model to include a maritime component

and assess the area which a given disposition of naval vessels could patrol.

The current model is being integrated with a Geographical Information System (GIS), which can be used to overlay features such as missile engagement zones and exclusion areas on to a geographical backdrop.

7 MODEL 5 – THE EXCLUSION ZONE MODEL

Policing a designated Exclusion Zone can be a problematic issue; analysts and planners must consider a wide variety of factors if the available assets are to be used effectively and intelligently. Issues such as achieving maximum coverage from Airborne Warning and Control Systems (AWACS) or determining optimal Combat Air Patrol (CAP) positions must be given due consideration.

Equally, minimising the risk of interception when infiltrating enemy airspace can also be a complex problem. The Exclusion Zone Model is intended to act as an intelligent aid to this decision-making process. The methodology used during development and the model architecture permit the model to evolve naturally with both the requirements of the analyst and of the planner. This has been achieved through the use of Component Object Model (COM) technology.

8 THE COMPONENT OBJECT MODEL (COM)

What exactly is the Component Object Model and what benefits does it provide that cannot easily be achieved using more traditional approaches to software development?

Microsoft define COM as:

‘...a platform-independent, distributed, object-oriented, system for creating binary software components that can interact.’

COM is a *binary standard* – a standard that applies to a program that has been translated into binary machine code. It does not dictate the structure, language or implementation details of the program. As a consequence, a component may interact with other components that have been developed using different languages, rely on a different operating system or reside on a different hardware platform. Developers who write COM Servers and Clients have no involvement in the underlying communication mechanism. This mechanism, known as *marshalling* is handled by the operating system.

The object-oriented nature of COM facilitates code reuse and enables different teams of developers to independently evolve their components without

jeopardising the stability of client applications. According to Microsoft:

‘...the key to building reusable components is to be able to treat the object as a black box. This means that the piece of code attempting to reuse another object knows nothing, and needs to know nothing, about the internal structure or implementation of the component being used. In other words, the code attempting to reuse a component depends upon the behaviour of the object and not the exact implementation.’

In summary, COM technology:

- a) Provides an object-based service to be used or reused.
- b) Allows components to be built independently and evolve independently.
- c) Allows components to be written in many different computer languages, use any operating system and operate across thread, process and network boundaries.

9 USING THIRD PARTY COMPONENTS

Figure 4 illustrates the way in which COM technology been exploited during the development of the Exclusion Zone Model.

The Microprocessor Applications Group (MAG) of Cranfield University have independently developed *Flora* and the *Flora Comms Server*. Together these act as a mapping Graphical User Interface (GUI) for a suite of Land-based Operational Analysis (OA) tools known collectively as *Thistle*. The tools are currently used by the Headquarters of the ACE Rapid Reaction Corps (HQ ARRC), NATO Consultation, Command and Control Agency (NC3A) and the Defence Evaluation and Research Agency (DERA).

Like the other *Thistle* components, *Flora* has been written in Borland C++ but because *Flora* functionality is exposed through a well-defined COM interface, the choice of language is not a concern. Furthermore, *Flora* is in a constant state of evolution but this does not adversely effect development undertaken by the Air Warfare Centre because client applications are only concerned with the *Flora* interface; this may be expanded to include additional methods but existing methods are immutable.

10 THE MODEL COMPONENTS

Each of the components developed by the Air Warfare Centre has been implemented using the Active Template Library (ATL). This is a library of template-based C++

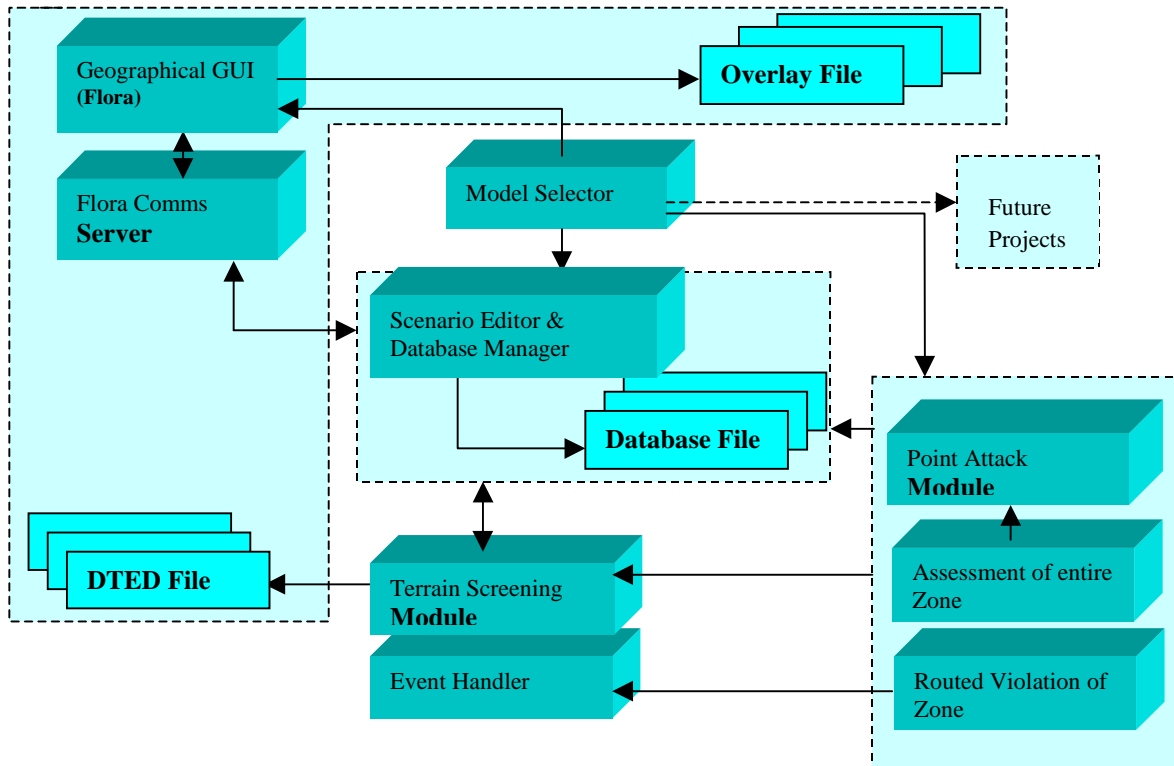


Figure 4: Schematic Illustrating the High-Level Architecture of the Model. Each Solid Box Indicates a Module That Uses COM.

classes that may be used to rapidly create small and fast COM objects. Development was undertaken using Visual C++ versions 5.0 and 6.0 as part of the Visual Studio Suite of Microsoft products.

This section provides a brief description of each component, including its purpose and how it interfaces with other components.

11 GEOGRAPHICAL INFORMATION SYSTEM GUI – FLORA

Flora acts as the main user interface, permitting the display of simple overlays on a geographical backdrop, Figure 5. The content and nature of an overlay is dependent upon the client application that is responsible for its management. Symbol information is passed to *Flora* through conventional COM calls in conjunction with a text-based communication protocol called Common Data Transfer Format (CDTF). This protocol was originally developed by HQ ARRC to provide a method for defining symbols within *Flora* based on the NATO APP-6 standard, Military Symbols for Land Based Systems.

It is a long-term goal of the OAE to develop a suite of air-based OA tools akin to the Thistle suite of models. The first member of this tool set will be the Exclusion

Zone Model although it is apparent from the discussion below that many of the COM objects used by the model may be re-used or evolved to accommodate future projects.

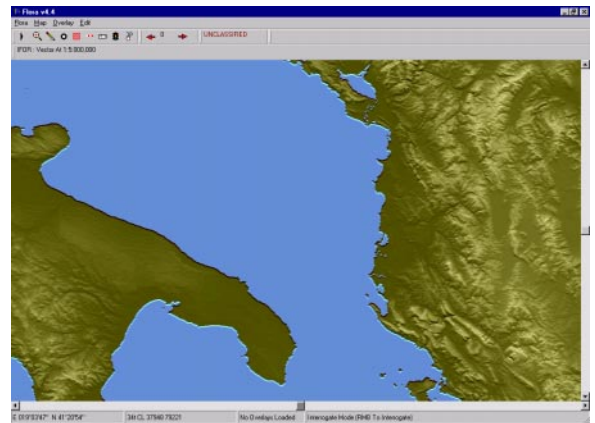


Figure 5: An Example Screenshot from the *Flora* Application

One of the principal advantages of *Flora* is that it is already well established amongst front-line OA organisations.

12 SCENARIO EDITOR AND DATABASE MANAGER

The Scenario Editor enables the user to position and define a variety of entity types in preparation for running other model components. Use of the editor is both simple and intuitive.

Firstly, the user opens an existing database or chooses to dynamically create a new database. The appropriate entity type is selected from a toolbar and then positioned by clicking on the desired map location. Clicking the right hand mouse button whilst the mouse pointer is over the entity symbol performs an interrogate operation. This in turn displays a property sheet containing a number of property pages. One property sheet in particular is pertinent to the type of symbol under interrogation and is under the control of the Scenario Editor client application, despite appearing from the user’s perspective to be a seamless part of Flora.

By editing the data fields of the property page, the entity may be named, re-positioned or other attributes may be defined or modified.

All data is stored in the named database and symbol information is stored in an appropriately named overlay file using CDTF.

The Editor uses the Microsoft Data Access Object (DAO) wrapper classes to create native Microsoft Jet engine databases (*.mdb files) and to interrogate or amend stored information.

The database file is opened for shared access so that other components may query it for specific entity information, Figure 6.

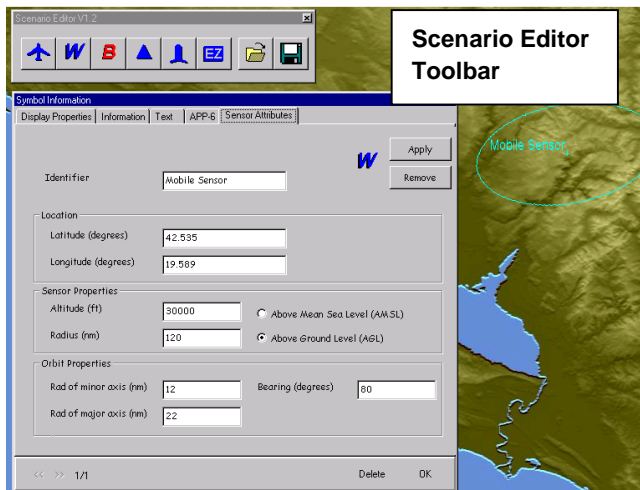


Figure 6: Format of the Display Following the Interrogation of a Symbol That Has Been Created Using the Scenario Editor

The Scenario Editor component exposes two custom interfaces : ICommsServer and IScenEd. The first of these

contains the same methods as those exposed by the Flora Communications Server and all method calls are delegated appropriately. This is an example of COM Containment and is a form of *black box inheritance* in which the Scenario Editor (the Outer Object) inherits the functionality of the Communications Server (the Inner Object). This structure is illustrated in Figure 7. The IScenEd interface includes methods for displaying the toolbar and determining the name of the active database. Note that the IUnknown interfaces are exposed by all COM objects and form part of the COM standard.

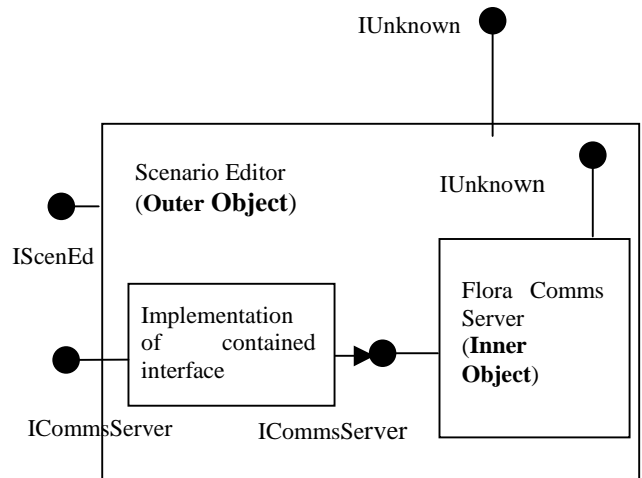


Figure 7: An Example of COM Containment

13 TERRAIN SCREENING

The Terrain Screening module has a dual role. Firstly, it is used to determine whether terrain obscures the Line of Sight (LOS) between two generic entities (at known location and elevation). This functionality is exposed to client applications through the ILosAssess COM interface.

Secondly, the same algorithms are employed to generate a LOS rosette over a designated region, Figure 8. The rosette indicates those areas where LOS is not obscured for a target at known elevation above ground level or sea level. The calculations are currently based on terrain only and use Level One Digital Terrain Elevation Data (DTED) as supplied by the Defence Mapping Agency. However, a more sophisticated radar detection algorithm may be incorporated in the future.

All sensing entities are created and modified using the Scenario Editor. A spherical Earth model is assumed.

14 POINT ATTACK

This is the first of three modules that are used to assess the effectiveness with which an Exclusion Zone may be

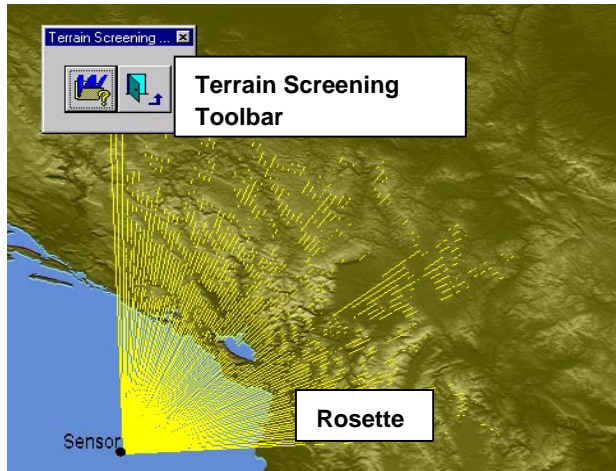


Figure 8: A LOS Rosette for an Airborne Sensor at 29,000ft with a Range of 200nm and a Target Elevation of 500ft Above Ground Level

policed. The User creates or loads a scenario using the Scenario Editor and then nominates a point target, a hostile entity, one or more intercepting entities and one or more sensors. The module evaluates the probability that the hostile entity is detected and subsequently intercepted before it is able to service the point target.

The hostile entity is assumed to approach from a user-defined range of possible bearings and advances along a straight-line path at constant speed for each of these bearings.

The positions of detection and interception are represented within a Flora overlay by two separate lines. If the interception line completely encloses the designated target when all angles of approach by the hostile entity are considered, then the target is not vulnerable to attack for the given scenario.

If the user selects multiple sensors then the chosen sensor is the one that is able to detect most rapidly. Similar reasoning is used to discriminate between interceptors when a number of these entities are selected.

The display is colour coded to indicate whether interception occurs before or after the offensive entity is able to engage its intended target.

15 ASSESSMENT OF ENTIRE EXCLUSION ZONE (Under Development)

The Scenario Editor enables the user to define an exclusion zone by clicking on the locations of the north-west and south-east vertices. Note that lines of constant latitude and longitude are used to represent the zone boundaries.

When assessing the level of protection afforded to an exclusion zone, the Point Attack algorithm is applied repeatedly to evenly spaced points. The point spacing is determined by a user-defined resolution. The algorithm returns the probability of successfully defending each point

from a nominated ingressing hostile entity. The resulting two-dimensional array of probability values can then be processed to generate a colour coded contour plot of the data that may be displayed as a Flora overlay.

It is possible to define a translucency attribute for symbols within Flora, thereby ensuring that the contour plot does not completely obscure the underlying terrain. The user is able to vary the translucency of a symbol from opaque to transparent, if desired.

All assumptions embodied by the Point Attack module are inherent within this component; namely, hostile entities are assumed to travel in straight lines at constant speed and approach from a user-defined range of possible bearings.

16 ROUTED VIOLATION OF EXCLUSION ZONE (Future Development)

This component introduces a time dependency to the analysis of an exclusion zone and also relaxes some of the assumptions made by the previous two components.

The user defines one or more hostile missions by clicking on a series of points. Each route is composed of a number of straight-line sections and will necessarily violate one or more exclusion zones. It may also include one or more designated point targets, although this is not a requirement of the module.

The simulation begins at time $t=0$ (unless specified as otherwise) and proceeds until all hostile entities are successfully intercepted or have exited the exclusion zone or zones to which they have been assigned. The tasking of interceptors is governed by pre-defined Rules of Engagement (RoE). One or two examples of these are listed below:

- a) Interceptors have no prior knowledge of hostile mission routes and therefore assume that hostile entities will maintain a constant bearing – this then corresponds to the *anticipated route*.
- b) Interceptors will only take action if interception may be achieved before the anticipated route breaches a Missile Engagement Zone (MEZ).
- c) Interceptors that are able to reach the hostile entity most rapidly are tasked when a sensor first acquires the entity.

If the rule can sensibly be discounted or if a number of alternatives exist then the user may select which rules are taken into account.

A Flora overlay is used to display the paths that are followed by interceptors, points at which the hostile entity is acquired or lost by a sensor and points of interception.

The final COM object, the Event Handler, handles all time dependence.

17 EVENT HANDLER (Future Development)

This COM client exposes a simple interface that facilitates the scheduling, removal and processing of different events. It is not specific to the types of event that occur within the Exclusion Zone Model and may therefore be incorporated into other COM-based time-dependent simulations.

18 USING THE DYNAMIC SYSTEM DEVELOPMENT METHOD (DSDM) TO MEET AN EVOLVING REQUIREMENT

The Exclusion Zone Model has been developed in an environment where:

- a) The requirement specification evolved over time.
- b) Each stage of development was time-constrained.
- c) Deliverable products were regularly required by the sponsor for review.

Clearly this is not a conducive environment for producing high quality software, however the development team are still required to provide an auditable software product which has been demonstrated to be 'fit for purpose'.

Consequently the development team have employed the Dynamic System Development Method (DSDM) to facilitate the controlled evolution of the model. DSDM focuses on the frequent delivery of products and the active involvement of the user in order for the system to converge on an acceptable solution within the allowed time. Put tersely, the DSDM process involves:

- a) Defining a baseline for high level requirements.
- b) Listing and prioritising the desired functionality.
- c) Allocating the functionality to a number of timeboxes.

Several reviews are held between developers and customers during each timebox. The prioritised functionality may be refined as a consequence of a review but the timebox deadline is not allowed to shift. The deadline for the final timebox coincides with project completion. The Exclusion Zone Model lends itself well to development within the DSDM framework.

19 SUMMARY

The Exclusion Zone Model has adopted a number of forms during its evolution that have varied considerably in both nature and complexity.

The current version of the model, developed using Microsoft Visual C++, makes use of Component Object Model (COM) technology to achieve language-independent, reusable functionality within an easily extensible architecture.

Development has focussed on the active involvement of the task sponsor and the frequent delivery of products in order for the system to converge on an acceptable solution within the allowed time. This is the essence of the Dynamic System Development Method (DSDM) and this approach has enabled the OAE to successfully provide a series of applications that address the key issues in a timely fashion.

The OAE philosophy for supporting front line operations, is that 80% of the solution in 20% of the time will often best meet the customer requirement.

REFERENCES

Microsoft Developer Network Library for Microsoft Visual Studio Version 6.0.

AUTHOR BIOGRAPHIES

PAUL RENDELL received an MA (Hons) in Physics from Wadham College, Oxford University. He is employed as a Software Engineer by the Centre for Operational Research and Defence Analysis (CORDA), a subsidiary of British Aerospace (BAe) and has specialised in the development of object-oriented and component based applications.

Whilst acting as a contractor to the Air Warfare Centre, these applications have been used to assist Operational Analysts in their role of providing support to the Royal Air Force (RAF).

ALAN COWDALE is the Senior Scientific Advisor at the Air Warfare Centre, RAF Waddington, UK. He holds a B.Sc. in Mathematics from the University of Southampton and an MSc in Operational Research from the University of Birmingham. He has undertaken a wide range of simulation projects for both the Royal Navy and Royal Air Force in connection with assessing weapon system performance and effectiveness.