Wargaming Tools with Chemical Biological and Radiological Capability

A. Skvortsov, R. Gailis, M. Ling

Human Protection and Performance Division,
Defence Science and Technology Organisation

ABSTRACT

The transformation of general military strategy and planning guidance to specific Defence Capability Strategy is where major future war fighting concepts are explored. Insertion of CBRN scenarios or challenges into the future operating concept exploration studies would provide some tests of the robustness of concepts to realistic threats. The provision of specific CBRN hazard information is a particular element required in the general NCW architecture, which would generally provide situational awareness and a common operating picture at various levels of the Force.

This paper provides an update on a current project that is integrating the LOD wargaming tool jSWAT (Joint Seminar Wargaming Adjudication Tool) with the HPPD modelling tool HPAC (Hazard Prediction and Assessment Capability), in order to provide realistic stimuli into Seminar Wargames based on CBRN threat scenarios.

A scenario to be modelled comprises a hazardous plume of some CBR agent with which the appropriate response agencies must deal with. A discussion of the integration issues and the final technical architecture of jSWAT/HPAC modelling tool is discussed. Following an initial demonstration, the intention will be to explore further iterations through more detailed simulation, using other LOD tools.

1. Introduction

The threat of CBRN attack is a frequent feature of the modern battlefield. Faced with such a threat a Commander needs to be able to rapidly access current CBRN hazard areas and determine the optimum plan for force protection and maintaining operational momentum. That is why incorporation of CBRN events in any operational model of the modern battlefield becomes a critical success factor.

To make a positive start on the development in this area, HPPD proposed a concrete project with real, measurable outcomes that will provide some practical experience in integration of CBRN events in warfare operation models. We also wished to draw on other expertise within DSTO in the Operations Research (OR) field. Given the fact that CBRN is predominantly seen as a Land issue, a logical first choice is to draw on the large OR capabilities that exist within Land Operations Division (LOD). LOD have a suite of OR tools and models to draw upon. The intention is to integrate one of the CBRN dispersion models with an agent based distillation model and a complex urban environment, to model entities such as military and civilian players.

A scenario we planned to develop comprised a hazardous plume of some CBR agent with which the intelligent agents must interact. A Proof of Concept (POC) and feasibility of the proposed approach would result in a demonstration of the integrated HPPD/LOD models through an adjudicated seminar wargame. As a first step, the LOD wargaming tool jSWAT (Joint Seminar Wargaming Adjudication Tool) and HPPD modelling tool HPAC (Hazard Prediction and Assessment Capability) were selected as the best candidates for trial integrations.
1.1 Why jSWAT?

The Joint Seminar Wargaming Tool is an LOD developed software package (programmed in Java) that aims to facilitate the seminar wargaming process that army currently uses to explore new concepts. Seminar wargaming is applicable to the way in which the CBRN exercises are conducted, and by using jSWAT it enables us to control the scenario and capture the required data in an automated fashion.

jSWAT’s design philosophy is based around the transparent map overlays that are used as part of the Military Appreciation Process (MAP). Different levels of tactical or strategic information are laid over the underlying terrain image to provide different insights into the planning process. jSWAT utilises this idiom in its layout, with different layers of information being able to be switched on and off at will (see Figure 1). HPAC’s CBRN information is displayed as another overlay. It provides information related to which CBRN sensors are activated and allows the operators to plot the spread of the plume.

Figure 1: jSWAT in operation

1.2 Why HPAC?

The Hazard Prediction and Assessment Capability (HPAC) models the release of CBRN materials to the atmosphere and the associated dispersion using detailed meteorological information [1]. It provides an estimate of effects on the physical environment and to a lesser extent the resulting impact of that environment on exposed population. Developed in the US by the Defence Threat Reduction Agency (DTRA) [2,3,9], HPAC was initially developed to meet the needs of the war fighter, but in recent years has expanded its capabilities to ensure it is relevant in the civilian counter-terrorism context. As its primary role is in emergency planning and response, HPAC is designed to run in real-time whilst still providing reliable predictions. Figure 2 below shows an example HPAC output for a fictional radiological dispersion device in Melbourne. The coloured contours show the different areas of radiological contamination that might be expected from such a device, overlaid on some freely available satellite imagery (obtained from Google Earth).

In general, the HPAC software predicts the effects of various hazardous material releases into the atmosphere and its collateral effects on civilian and military populations. This counterproliferation/counterforce tool assists warfighters in destroying targets containing weapons of mass destruction and responding to hazardous agent releases. It employs integrated source terms, high resolution weather forecasts and particulate transport algorithms to rapidly model hazard areas and human collateral effects.

At the heart of HPAC is the Second-order Closure Integrated Puff (SCIPUFF), a Lagrangian puff dispersion model developed in FORTRAN [2]. Detailed descriptions of material releases, meteorology, terrain, and other inputs are fed to SCIPUFF, which calculates the transport and dispersion and tracks material concentrations, depositions, and doses.

Figure 2: HPAC output for a fictional radiological dispersion device

In addition, several models provide a more abstract or operational description of events, referred to as incidents. These models include conventional attacks on biological and chemical facilities, accidents at nuclear facilities, nuclear weapon...
incidents, nuclear weapon detonations, chem-bio weapons, missile intercepts, and smoke and obscurants.

The HPAC Project Editor is a rich and fully featured user interface to create the description of some event that releases CBRN material into the environment (an “incident”). Currently, OpenMap [8] is used for the bulk of map data display. HPAC cleverly characterises each incident as “Where”, “What” and “When” and allows an incident to transform to a “release” (being a detailed physical description of the NBC material that is released into the environment).

Architecture wise, HPAC 4.0 is designed as a networked application. The foundation of the system is a set of servers designed to be accessed from a variety of clients, not only the client application built for HPAC. HPAC client applications may be written in any language and targeted for any platform capable of communicating with the server via TCP/IP and the Common Object Request Broker Architecture (CORBA) [5,9].

Unfortunately HPAC has an internal critical limitation in terms of jSWAT integration: it is not possible for two clients to simultaneously access a running server.

### 1.3 What we mean by integration.

While the term “integration” sounds straightforward, it needs to be clearly specified in the current context of wargaming tools integration. On functional level we needed to evaluate two options

- Duplex information exchange and display,
- Mutual models influence (i.e. output of one model may impact result of the other).

For the POC, the first option was adopted, i.e. HPAC data is displayed in jSWAT, but it has no direct influence on model entities. Thus as a result of models integration, we provided in jSWAT:

- An environment to assist the adjudication of CBRN effects by displaying contoured data associated with an incident release. No automation is provided of casualties or manoeuvre effects as a result of exposure to a CBRN release or the need to adopt protective (MOPP) measures.

- Sensors and detectors allow the non-adjudicating players to effectively interact with the physical model of the plume. However, an adjudicator may intervene to shape the player's awareness of the source term in any manner desired.

The integration methods may vary depending upon their level of coupling between components and the amount of internal modifications required to enable them. We also evaluated two extreme options

- Strong coupling (each model uses shared components of each other and makes direct functional calls)
- Loose coupling (each model has no detailed knowledge of the other, and only receives / sends broadcasted messages).

The second option was adopted since it is less dependent on model source code and more flexible in terms of deployment and version control.

### 2. Bridging the Architectures

At runtime, the HPAC application can be thought of for development purposes as comprising three key elements:

- The SCIPUFF server,
- The Project Editor, and
- The Interface for Consequent Effects (ICE).

For deployment, each of these components may execute on a separate machine and are networked through the agency of a CORBA middleware layer.

When considering jSWAT integration with HPAC, we add two elements [3,6]:

- The jSWAT Server, and
- One or many jSWAT Clients.

The jSWAT Server and Clients are networked using Jini technology to provide a mechanism for the Clients to discover instances of the Server

---

1 Jini™ (pronounced like genie) is a network architecture for the construction of distributed systems.
running on the network. A JavaSpace\(^2\) is used as the communication technology once clients have been found. The JavaSpace in this case functions as a middleware layer, providing a simple network paradigm for sharing objects between Server and Client analogous to the CORBA middleware layer [4,5].

The jSWAT Server acts as the mediator of the JavaSpace and controls the “truth” of the active wargame. That is, the Server populates the space with objects representing the current game state and is the sole author of object written to the space.

In order to integrate the two applications with their differing middleware layers, it is sufficient to create a linkage between the jSWAT Server and the ICE. This linkage acts as a bridge between the two middleware layers and suffices to translate between the object structure of HPAC on the one hand and the representation of the world in the jSWAT space on the other.

Because each client in the HPAC architecture does not share the server but rather is allocated its own short-lived server process, there is no capacity in the HPAC architecture for two clients to share a project at runtime. This key finding mentioned above drives the POC integration towards the following user process, using the saved Project file as the mechanism to transfer incidents from the HPAC Project Editor to the jSWAT Client:

1. A skilled user creates incidents with the HPAC Project Editor including source term, terrain and weather conditions.
2. The user saves the incident definition as an HPAC Project in a well known location.
3. The jSWAT server is started and pointed to the HPAC transfer Project.
4. The Project is loaded to the SCIPUFF server.
5. The jSWAT JavaSpace is populated with objects corresponding to incidents for display on the jSWAT Client.
6. Periodically or at the culmination of the game, the Project file is saved to disk.

The process flow described above can be visualised (with some alternate potential pathways) per Figure 4. This figure highlights the critical issue of the separate “per client” process space for the HPAC Server and illustrates the necessity of a file based transfer between process spaces in the situation where the Project Editor is used to characterise the release.

---

2 A JavaSpace is a Jini service that provides a distributed persistence and object exchange mechanism.
A further refinement on this process would allow the HPAC user to save the Transfer Project during a running jSWAT game allowing automatic (or more likely manual) reloading and remapping of the transferred releases into the jSWAT object space. This idea, while technically feasible, falls outside the brief of the POC demonstration.

The SCIPUFF server is able to take a range of named samplers and compute (at least) concentration values as a time series at these static points. Little information has been available on the format of inputs and outputs necessitating a degree of effort to reverse engineer this process. Beyond the POC, direct contact with the developers of this element of the SCIPUFF codebase may substantially enhance our usage of the sampler interface.

### 3. Preliminary Results and Future Work

The POC for jSWAT/HPAC integration was successfully demonstrated at a Land Warfare Development Centre (LWDC) Workshop in August 2006.

Here we demonstrate some results for illustration purposes only. Figure 5 shows an example of a plume calculated in HPAC.

**Figure 4: JSWAT/HPAC integrated process flow.**

In Figure 6 the same plume is exported and displayed in jSWAT.

The LWDC workshop has demonstrated the POC capability to an ADF audience, as well as participants from the Land Operations Division (LOD). The workshop generated much discussion with positive feedback on the usefulness of the tool. A number of additional functionalities have been identified at the workshop and these are now being implemented under a new development contract.

The jSWAT/HPAC tool was also demonstrated to a National Security and State Emergency Services audience at a joint LOD and HPPD workshop in December 2006. The outcome of this workshop is being reported by Grieger, Ling and Nunes-Vaz at SimTecT 2007.
4. References