Simulation-C2 Interoperability Through Data Mediation: the Virtual Command and Control Interface

Don MacQuarrie
Captain Chris Taff
Ben Asselstine
Land Software Engineering Centre
305 Rideau Street
Ottawa, ON K1A 0K2
907-555-1234, 907-555-5678
don@ideasincontext.com,
taff.cp@forces.gc.ca,asselstine.bd@forces.gc.ca

Raj Hans
Scott Reid
SimFront Simulation Systems
594 Duff Crescent
Ottawa, ON K1J 7C5
613-294-4905, 613-796-2100, 613-889-1449
r.hans@simfront.com, s.reid@simfront.com

Keywords:
Simulation, C2, Interoperability, C2IEDM

ABSTRACT: As C2 systems become more sophisticated, constructive simulation is necessary to provide credible battle scenarios at a useful level of fidelity and resolution for training and experimentation. Interoperability, in the form of automated and semi-automated information exchange between the simulation and C2 domains, is essential.

This paper describes an approach to simulation-C2 interoperability employed by the Canadian Department of National Defence (DND). For the past several years, DND has developed the Virtual Command and Control Interface (VCCI), influenced largely by the evolution of C2 systems built on the Multilateral Interoperability Programme (MIP) Information Exchange Data Model (IEDM). VCCI is now fielded and in use within the Canadian and UK Armies. In the VCCI framework, simulation-specific components handle translation between the simulation(s) (such as the Joint Conflict and Tactics Simulation (JCATS) and others) and a IEDM-compliant database. This VCCI database, or VDB, becomes the hub for data exchange; it not only contains information from the simulation(s) and C2 system in use, but also meta-information about the larger training/experimentation context. Data exchange can then performed with the IEDM-compliant database used by the C2 system (in Canada's case, the Operational Database). Although this exchange can be largely automated, our experience has shown the value of maintaining a mediation agent, allowing human-in-the-loop data mediation in order to achieve training or experimentation objectives. By separating the data translation function (from the simulation to the VDB) from the information management function (between the VDB and the ODB), the VCCI framework allows each function to be controlled by the appropriate parties. Also, by maintaining contextual information within the VDB, after-action review (AAR) can be enriched.

The current state of the VCCI architecture is presented and described, recent experiences with VCCI use in preparing units for operational deployments are discussed, and areas for future exploitation are identified.

1. Introduction

In simulation-aided military training exercises, training value is increased greatly if training audiences can be insulated from the simulation. The audience can then operate with their normal command and control (C2) tools in a “train as you fight” mentality. Similarly, as C2 systems become more sophisticated, constructive simulation is necessary to provide credible battle scenarios at a useful level of fidelity and resolution for training and experimentation. Interoperability, in the form of automated and semi-automated information exchange between the simulation and C2 domains, is essential.

To train commanders and staffs, the Canadian Army frequently uses two constructive simulations to model the synthetic battlespace: the Joint Conflict and Tactical Simulation (JCATS)[1] and the ABACUS Command and Staff Trainer (CAST/ABACUS)[2]. In the C2 domain, a variety of automated tools of the Land Command Support System (LCSS) are used. Early attempts at data interchange between each simulation and each C2 application quickly grew into a “many to many” relationship which was unsustainable.
This paper describes an approach to simulation-C2 interoperability employed by the Canadian Department of National Defence (DND). For the past several years, DND has developed the Virtual Command and Control Interface (VCCI), influenced largely by the evolution of C2 systems built on the Multilateral Interoperability Programme (MIP) Information Exchange Data Model (IEDM)[3]. VCCI employs a modular architecture in order to reduce the impact of changing simulations or C2 applications. The VCCI system is now fielded and in use for training and experimentation within the Canadian and UK Armies.

This paper is organized as follows. Section 2 contains short descriptions of important concepts, and lists some of the requirements which drove key design decisions. Section 3 describes how these design decisions were incorporated into the VCCI architecture, and discusses the tradeoffs that were made. Section 4 details a typical implementation of VCCI, and Section 5 discusses important lessons learned during employment of this implementation. Section 6 contains a summary and a description of future work.

2. Concepts and Requirements

2.1 Concepts

Interoperability. A discussion of M&S-C2 interoperability must begin with a common understanding of the term. Obviously, interoperability is not an all-or-nothing state. It can range from simple interconnection at the physical level (e.g. two disparate systems sharing a TCP/IP network) to complete interoperation (e.g. two systems which not only share data in common and useful form, but also are aware of and react to each other’s operation).

For this paper, we use two descriptors of interoperability: the Levels of Conceptual Interoperability Model (LCIM) and the doctrine-representation-protocol view of Battle Management Language (BML).

In the LCIM, described in [4] and refined in [5], interoperability can range from Level 0 (No Interoperability) through Level 6 (Conceptual Interoperability). For practical reasons, we restrict our goals to Levels 1 – 4, quoted from [5] as:

- **Level 1**: On the level of Technical Interoperability, a communication protocol exists for exchanging data between participating systems. On this level, a communication infrastructure is established allowing the exchange of bits and bytes; the underlying networks and communication protocols are unambiguously defined.

- **Level 2**: The Syntactic Interoperability level introduces a common structure to exchange information, i.e., a common data format is applied. On this level, a common protocol to structure the data is used; the format of the information exchange is unambiguously defined.

- **Level 3**: If a common information exchange reference model is used, the level of Semantic Interoperability is reached. On this level, the meaning of the data is shared; the content of the information exchange requests are unambiguously defined.

- **Level 4**: Pragmatic Interoperability is reached when the interoperating systems are aware of the methods and procedures that each other are employing. In other words, the use of the data – or the context of its application – is understood by the participating systems; the context in which the information is exchanged is unambiguously defined.

The design goal for VCCI was to reach Level 3 through a common information exchange reference model, allowing for automated information sharing. Also, we attempted to approach Level 4, in a restricted fashion, by allowing human-in-the-loop intervention at run-time. This is arguably not true pragmatic interoperability, but was considered acceptable in our specific application.

Initiatives such as the Coalition Battle Management Language (C-BML) [6] attempt to define unambiguous grammars for describing data interchange. BML has the notion of three views: doctrine (every term in the language must be unambiguous and rooted in doctrine), representation (the language should provide one and only one way to represent battlespace events, missions, and tasks), and protocols (the language should specify the technical means of data exchange).

Again, our goal with VCCI was an interface that would respect the representation and protocols view, and offer a means of accessing the doctrine view.

Automated data exchange. The notion of automated data exchange within the simulation and C2 domains is
not new. In the simulation domain, standards such as the Distributed Interactive Simulation (DIS) [7,8] protocol and the High-Level Architecture (HLA) [9] are used to specify the nature and form of data that simulations can receive and publish. However, when VCCI design began, there was no common FOM which would interface with our C2 system.

At the same time, the MIP community had been showing excellent progress in C2 interoperability through the IEDM. Canada was a leader in this development, and our C2 systems were designed to be IEDM-compliant.

We therefore examined the IEDM as a model to allow automated data exchange between simulation and C2, with the understanding that mapping work would be required to the simulation interface (whether DIS, HLA, or other). This appeared to be a common approach in the M&S community.

2.2 Requirements

Although the specific requirements for the VCCI system are lengthy, they can be summarized for the purposes of this paper as the following. These are in rough order of priority, and thus correspond to the chronological development of VCCI:

- The system must interface to both JCATS and CAST/ABACUS, and provide the ability for future interfaces to new constructive sims as they are adopted for training or experimentation.

- The system must stimulate the Operational Database (ODB), the main data store of the Canadian C2 system\(^1\). The system must provide the following info to C2 users:
  - positional data
  - basic status data
  - holdings
  - advanced reports and returns (contact reports, etc)

- The system must allow the following C2 to Sim data interchange:
  - planning data (control features)
  - initial unit positions
  - obstacles
  - orders/missions/tasks

- The system should allow for robust after-action review (AAR).

- The system should allow for purpose-built federations of the existing simulations and other supporting simulation systems. This requirement was included because, at the time, HLA federations were not available or did not meet user requirements.

3. Design of the VCCI Architecture

We now turn to the VCCI architecture itself. Figure 1 shows a high-level view of the architecture, consisting of a constructive simulation, the VCCI system, and a typical C2 system.

![Figure 1: The VCCI Architecture](image)

The core of the VCCI system consists of a set of collectors, the VCCI database (VDB), and the VCCI Mediation Agent (VMA).

3.1 Collectors

A collector is specific to the simulation in use. It reads two different kinds of information from the simulation scenario: start-state information (which tends to be specific to the simulation in use), and dynamic battlespace events (which tend to be available in more common formats and protocols). The functionality of the collector depends on the capabilities of the simulation: it may monitor DIS/HLA traffic (e.g. for use with JCATS) or it may require use of a proprietary interface (e.g. with CAST/ABACUS). Start-state information may be read from simulation configuration files or databases.

\(^1\) The original intent was to use the MIP gateway to the Canadian C2 system, and have the simulation domain appear as “just another C2 system”. However, the MIP interface did not provide access to the full data set that users required.
“Collector” is a historical term and is somewhat of a misnomer as it implies only one-way information flow from simulation to C2. In early versions of VCCI, this was the priority and was the first functionality to be implemented. In future versions, the collector is intended to provide two-way exchange, transferring battlefield events to the C2 domain, while applying planning and task information from C2 to the simulation.

3.2 The VCCI Database

The heart of the VCCI architecture is the VCCI database, or VDB. This is an IEDM-compliant database used to store all appropriate information about the simulation context. Note that earlier versions of VCCI did not employ this intermediate database. Instead, they behaved according to the Message Exchange Mechanism (MEM) of the LC2IEDM [10]. At the time, VCCI performed basic data transformation at the syntactic level, reading simulation events and translating them into ADatP3-formatted [11] messages. These messages were received by the C2 system as if they had originated from any MIP-compliant C2 node. The adoption of the VDB in the current version of VCCI was an important design decision, and bears further explanation here.

Although the message-based versions of VCCI were initially sufficient to provide stimulation of MIP-compliant C2 tools, we found several advantages to using an intermediate database, over and above reducing the many-to-many interface problem described in Section 1. These include improved data mediation through use of an IEDM-compliant intermediate database, storage of the larger simulation context, and the ability to perform effects modeling.

Improved data mediation. The VDB uses the IEDM data model in order to facilitate information transfer. As seen in [12] and [13], the C2IEDM is a useful medium for interoperability provided certain conditions exist: that it is used for interoperability of entities and concepts, and that the two domains are of similar resolution. Both of these conditions can be met in our particular implementation and with the given use-cases for VCCI.

Thus, during implementation of the VCCI collectors and the VDB, use of the IEDM ensures that all required battlefield concepts are thought out and captured in the data model at design-time. Every simulation concept of interest is manually mapped to the IEDM, and generic database transactions are developed. This is expensive in terms of development effort, but it ensures that simulation data is correctly captured and stored; not just in terms of its syntax, but in terms of its place in the overall conceptual model. The mapping effort, once completed, gives users confidence that data in the VDB is correct (in terms of the simulation conceptual model) and able to be exchanged (in terms of the C2 conceptual model).

Storage of the larger simulation context. The VDB receives all “ground truth” information from the simulation(s) in use. This represents an all-knowing view of the battlefield. Since the users in the C2 domain are limited to only that information which can be obtained through sensors, reports, or joint/coalition forces, the VDB will, over time, contain much more information than will be reported to the C2 domain. During the design of VCCI, the VDB was intended to be the “centre of all knowledge”. Therefore, not only does it contain all appropriate information from the simulation, the intent was to add further metadata about the overall context in which the simulation is being employed: information which is available to the exercise staff but which exists outside of the simulation. This relationship is shown in Figure 2.

![Diagram showing the relationship between VDB and other databases](image)

Figure 2. Use of VDB to store larger context

Having this single source of all data allows for greatly enriched AAR: imagine a training audience that can quickly recall not only the events of the simulation, but also other non-simulation exercise injects. For example, it may assist AAR if the training audience can be shown other events that were occurring at the time (media injects, for example) even if they were not modeled in the simulation itself.
Effects modeling. Often, exercise designers wish to implement an effect for use by the training/experimentation audience, and are less concerned with the simulation steps which are required to achieve that effect. Certain effects are more difficult to model than others, based on the simulation(s) in use. For example, a constructive simulation such as JCATS can easily model force-on-force kinetic battlespace events but cannot easily show impacts of certain civil-military cooperation efforts. An important deduction made during VCCI design was that, in order to incorporate these effects, it was not necessary to model their component steps (poorly) in an existing simulation. Rather, the effect can be directly modeled in the VDB using the same mapping effort that had been done for simulation events. At run-time, an “effects server” can inject these effects through the collector into the VDB.

3.3 The VCCI Mediation Agent

The design and use of the VDB, therefore, could permit interoperability up to the semantic level (Level 3 of the LCIM), assuming that the VDB and ODB were connected with direct data replication. To add support for higher levels of interoperability, a Mediation Agent was added to the VCCI design, both to control the data transfer but also to allow for data mediation.

VMA has two modes: similar to the collector, it can transfer the static information elements (e.g. organization and equipment types, ORBAT info) and also the dynamic information elements (e.g. positions, holdings and status updates). Whereas the collectors populate the VDB with simulation data in real-time, the VMA populates the ODB with specific information (e.g. which subset of organizations), at a specific interval (e.g. every 10 minutes). It is precisely these settings that model the reporting from synthetic units (simulation entities) to real units (a commander being trained). The VMA also provides positional information for a unit based upon the position of that unit’s commander. This “Commander’s Vehicle” feature enables the translation of a simulation concept (i.e. the Leopard C2 operated by an armoured troop leader is at a certain position) to a C2 concept (i.e. 1 TP, A Sqn is near that location).

Simple situational awareness is, of course, a very small part of M&S-C2 interoperability. The designers of the VMA realized that true conceptual interoperability was impractical in implementation. At the time (and, arguably, today) there was no grammar nor reference model which could be applied as a straightforward series of mapping algorithms. Instead, the designers decided to present the information gaps to military subject-matter experts at run-time. They would use their knowledge and expertise to adjust various VMA settings and, in so doing, enable the correct C2 view of the simulation, and vice-versa.

Thus, the VMA can be said to perform data alignment and data transformation in an automated fashion. Unit and equipment type and instance information can be obtained, translated, and stored correctly into the ODB of the C2 system. In C-BML parlance, this could be referred to as the representation view. The VMA also performs a certain level of data mediation: imparting context to the data through the settings determined by the exercise staff. This varies from scenario to scenario, and is not easily automated. By providing this “human in the loop” interaction, the VMA allows the application of a portion of what C-BML refers to as the doctrine and protocol views.

3.4 After-Action Review

Note that Figure 1 also shows links to an After-Action Review (AAR) tool. Though not a necessary component of VCCI, integral AAR is a useful by-product of the VDB design. Although most modern constructive simulations include AAR capability, they tend to be proprietary and inconsistent across simulations. By using the VDB to store all simulation information in a common format, a single tool can be used to perform AAR regardless of the particular simulation employed. This reduces the skillset required to conduct AAR, and presents the information in a single, consistent form.

4. Implementing the VCCI Architecture

A typical implementation of VCCI is shown in Figure 3. Note the two operating modes shown here: Step 1, or “static” mode, and Step 2, or “dynamic” mode.
Figure 3. Example VCCI Implementation

The JCATS-specific collector reads start-state information from the JCATS configuration files (.setup, .fplan, and .fchar). It parses this information and converts it to LC2IEDM format for entry into the VDB. Table 1 gives an example of the mapping of simulation data to the IEDM. Note that this is a rather simple example. Not all data are mapped one-to-one, and the collector contains various inference rules to correctly handle these cases.

<table>
<thead>
<tr>
<th>JCATS Field</th>
<th>Value</th>
<th>Table</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit-OTHGold-Service</td>
<td>ARM</td>
<td>unit-type</td>
<td>unit-type-service-code</td>
<td>A</td>
</tr>
<tr>
<td>Unit-Agg-Mobility Class</td>
<td>Ground Forces</td>
<td>unit-type</td>
<td>unit-type-mobility-code</td>
<td>WTC</td>
</tr>
<tr>
<td>Unit-OTHGold-OrgType</td>
<td>MECHINF</td>
<td>unit-type</td>
<td>unit-type-category-code</td>
<td>B</td>
</tr>
<tr>
<td>Unit-OTHGold-Echelon</td>
<td>BN</td>
<td>unit-type</td>
<td>unit-type-size-code</td>
<td>BN</td>
</tr>
<tr>
<td>DIS Country Code</td>
<td>39</td>
<td>object-type</td>
<td>object-type-nationality-code</td>
<td>CA</td>
</tr>
</tbody>
</table>

Table 1. Example data mapping

Once the VDB has the initial information, and prior to running the simulation, the VMA is then run in static mode in order to generate a start-state ODB for use by the training audience. Using VCCI to populate initial holding and ORBAT info ensures that C2 system does not contain information which is inaccessible to the simulation.

With the start-state information promulgated, the simulation can then be executed. The collector shifts into “dynamic” mode, reading DIS Protocol Data Units (PDU) and updating the VDB. The collector has configurable time/distance thresholds to limit the number of database updates; a unit can be set to report its status every certain number of seconds or after moving a certain distance.

Table 2 shows a sample mapping of a Fire event (DIS FIRE PDU) to the VDB. Note that this mapping is not one-to-one; several fields in the VDB are populated from database lookups, rather than directly with PDU data.

<table>
<thead>
<tr>
<th>DIS PDU Fields</th>
<th>LC2IEDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDU Header Record</td>
<td>location/point/absolute-point</td>
</tr>
<tr>
<td>Firing Entity Id Record</td>
<td>organisation-point (unit) or materiel-point (materiel)(relationship of shooter/position/time)</td>
</tr>
<tr>
<td>Target Entity Id Record</td>
<td>reporting-data/reporting-data-absolute-timing (time)</td>
</tr>
<tr>
<td>Munition Id Record</td>
<td>reference (source data of event)</td>
</tr>
<tr>
<td>Event Identifier Record</td>
<td>action/action-event (the attack event)</td>
</tr>
<tr>
<td>Fire Mission Index Field</td>
<td>action-event-status (relationship of action to time)</td>
</tr>
<tr>
<td>Location In World Record</td>
<td>reporting-data/reporting-data-absolute-timing (use same objects as above)</td>
</tr>
<tr>
<td>Burst Descriptor Record</td>
<td>object-item/feature/control-feature (to model the aim point)</td>
</tr>
<tr>
<td>Velocity Record</td>
<td>object-item-type (to associate control feature instance to type)</td>
</tr>
<tr>
<td>Range Field</td>
<td>object-item-status/control-feature-status location/point/absolute-point</td>
</tr>
<tr>
<td></td>
<td>(target location)</td>
</tr>
<tr>
<td></td>
<td>action-objective/action-objective-item (relationship of target to the action)</td>
</tr>
</tbody>
</table>

Table 2. Example event mapping

At this point, the VMA is configured for dynamic transfer. Exercise controllers adjust settings such as the
appropriate commander’s vehicle for each reporting subunit, and decide upon reporting intervals. These intervals are chosen, partly to replicate the real-world performance of the C2 system, and also to inject exercise artifacts such as “fog of war”.

5. Lessons Learned

The current version of VCCI was fielded in March 2007. It is currently in use at five training sites across Canada, to train commanders and staffs at levels from Company to Brigade. Typical training events involve 4000 simulation entities, stimulating as many as 200 C2 clients. A typical exercise construct is shown in Figure 4.

![Figure 4. A typical exercise construct](image)

In these events, the simulation is normally controlled by exercise players outside of the training audience: the so-called HICON and LOCON, for example. These players also have access to the C2 system, in order to receive orders and other supplementary data. The training audience, as much as possible, does not interact directly with the simulation.

In the past year, VCCI use has been refined and many lessons have been learned. Some of the more pertinent are listed here.

5.1 Time management

Simulations have a variety of means for time management. The simplest employ offsets from an arbitrary start time, and assume that all distributed clients are using the same timeframe within an acceptable latency. Simulations which are connected as an HLA federate normally conform to the HLA concept of timeslots. This allows simulations to be paused, restarted, and so on, without affecting the overall flow of the exercise. On the other hand, our C2 tools are intended to function always in a “real-world” timeframe. There is no simple way to pause or reset time for a C2 system.

This became an issue during multi-day training events where the simulation is paused at the end of each day, and resumed the following morning. Simulation data was either rejected by the C2 system, or it was accepted but treated as aged: for example, friendly-force icons were displayed in a greyed-out state as if no recent position update had been received.

To resolve this, functionality was added to VCCI to allow for arbitrary offsets to be applied automatically to the timestamps received from simulation. Each morning, exercise staff could apply the correct offset to ensure that simulation timestamps would appear correctly to the C2 system. This approach is suboptimal, however—it violates the notion that all data from the simulation is “ground truth”. We are continuing to research ways to solve this problem without making arbitrary changes to simulation data.

5.2 Initial database synchronization.

As mentioned in Section 4, VCCI is used to generate start-state C2 databases, thus ensuring data synchronization between simulation and C2. Simulation centre staff perform this task on a regular basis for training and experimentation audiences. However, this can be problematic in practice: C2 users must often enter significant planning data into their C2 system prior to arriving at a training or experimentation event.

To accommodate this, VCCI is used to generate the start-state ODB which is the delivered to the users, often months prior to the exercise. This leads to configuration control issues and reduces the flexibility of the simulation centres to make short-notice changes.

5.3 Common reference types.

The use of an intermediate database for data mediation can only be useful if the database contains the combined data model of the simulation and C2 domains. Initializing the database requires a single authoritative source of reference information that spans both domains. As noted in [14] and [15], this common initialization is problematic.

In order to address this issue, the simulation designers use a common database of reference type information, which is maintained on their behalf by the C2
community. Any new data, whether required by C2 or by simulation, is entered into the database and thus available to both. During VCCI operation, the type information is first initialized in the simulation (as specific instances of each type), and then used to populate the C2 system. The disadvantage to this approach is that new type information cannot be added by the simulation at run-time (i.e. the scenario must be completely understood by the simulation designer prior to commencement). Also, configuration management is required for the type information, spanning the C2 and M&S community.

5.4 Performance

The use of VCCI in command and staff training has also led to unrealistic expectations regarding the performance of our C2 system.

VCCI is deployed in a simulation environment using commercial off-the-shelf (COTS) computer hardware and typical business-grade network infrastructure. For cost-savings, the C2 applications are also running on this “blue-wire” network, rather than in their operational environment of ruggedized computers and field radios. For many users, this is the first exposure to C2 of large numbers of units. Training audiences at higher levels may believe (incorrectly) that the C2 system is more responsive than can realistically be expected. To counter this, VCCI includes the ability to add arbitrary delay to information reporting in order to increase the realism of the C2 system.

6. Summary and Future Work

6.1 Summary

This paper has presented an approach to C2-M&S data mediation. By use of an intermediate database built upon the IEDM, the VCCI system can allow interoperability at the semantic level. The VCCI Mediation Agent allows human-in-the-loop intervention in data transfer, which relieves the demanding requirement for complex data mediation algorithms which are difficult to prove correct.

The intermediate data model ensures that simulation entities and concepts are (1) modeled correctly at the semantic level and (2) able to be interchanged with C2 in a semi-automated fashion. This VCCI database has provided other useful functionality, such as a common AAR tool, storage of the larger exercise context, and effects modeling.

The VCCI system has been demonstrated with simulation entity counts as high as 65,000; it is in daily use for training and experimentation, with average entity counts of 2000-4000, providing stimulation to hundreds of C2 clients.

By maintaining an IEDM-compliant data model, it is relatively straightforward to extend VCCI to interoperate with other MIP-compliant C2 systems.

6.2 Future Work

Some areas of future work include:

JC3IEDM support. As Canada transitions to the JC3IEDM as its data model in the C2 domain, VCCI will also adopt this model, continuing the assurance that simulation information is correctly stored. This will require changes to the VDB and to the collectors. The mapping work described in Section 4 will need to be reviewed and extended to incorporate changes in the model. Although somewhat of a duplication of effort, this work is justified by the information assurance and fine-grained control over the data interchange that is provided.

Common data initialization. As mentioned in Section 5.2, although VCCI addresses the problem of common data initialization by generating start-state C2 databases, this is not an ideal solution. We are investigating means to allow interoperability in the case of pre-existing ODBs containing unknown start-state information. Unfortunately, this is not simply a technical solution. This will likely involve the Canadian M&S and C2 communities agreeing on processes to ensure common authoritative data sources.

Two-way VCCI. The current fielded implementation of VCCI, while useful, is largely “one-way” from simulation to C2. We are now working on the next priority set of requirements: implementing data interchange from C2 to simulation. The first prototype allows transfer of control feature and overlay information, and is currently in the beta release stage.

Effects Servers. As mentioned in Section 3, use of the VDB allows for effects modeling, or direct inject of certain outcomes which are difficult to model in other constructive simulations. We plan to develop several task-tailored “effects servers”—the first, a simplified air mission server, in currently in development and has been successfully demonstrated to simulation staff.
7. References


Author Biographies

DON MACQUARRIE is a software engineer with the Land Software Engineering Centre of the Canadian military and the current project lead for the VCCI project. His interests include M&S/C2 interoperability, modeling of cultural factors and non-kinetic effects, and uses of serious games. Mr. MacQuarrie is a retired Signals Major, with experience at the Tactical and Operational level. He holds a B.Eng. in Computer Engineering from the Royal Military College and a M.A.Sc. in Software Engineering from the University of Waterloo.

CAPTAIN CHRIS TAFF is the Project Manager/Lifecycle Application Manager for C2 simulation for the Canadian Army. He is responsible for provision and maintenance of constructive simulation capability, and for development of tools including the VCCI system. He has seven years of experience in modeling and simulation. Captain Taff holds a B.A. double major in Honours History and Western Civilization (Liberal Arts College program) from Concordia University.

BEN ASSELSTINE is a software engineer with the Land Software Engineering Centre of the Canadian military and a former project leader for the VCCI project. He has nine years experience in the M&S field, and holds a B.Sc. in Anthropology and Computer Studies from Trent University.

RAJ HANS is an engineer with SimFront Simulation Systems on the Land Software Engineering Centre team of the Canadian military. He is currently developing and maintaining components of the VCCI architecture. Mr. Hans has worked on simulation-related projects in both the private and defence sector. He holds a B.Eng in Electrical Engineering with a Minor in Computer Science from McGill University.
SCOTT REID is CEO of SimFront Simulation Systems, a provider of some of the components of the VCCI architecture to the Canadian and UK Armies. His interests include AAR tools, and M&S/C2 interoperability. Mr. Reid is a retired Signals Captain and an original architect of the first two generations of VCCI. He holds a B.Eng. in Computer Engineering from the Royal Military College and a M.A.Sc. in Software Engineering from the University of Waterloo.