Using an Ontology for Entity Situational Awareness in a Simple Scenario

Joyce A. Nagle
U.S. Army Engineer Research and Development Center
Cold Regions Research and Engineering Laboratory
Hanover, NH 03755
Joyce.A.Nagle@us.army.mil

Paul W. Richmond
U.S. Army Engineer Research and Development Center
Geotechnical and Structures Laboratory
Vicksburg, MS 39180
Paul.W.Richmond@usace.army.mil

Curtis L. Blais
MOVES Institute
Naval Postgraduate School
Monterey, CA 93943
clblais@nps.edu

Niki C. Goerger
U.S. Army Engineer Research and Development Center
Geotechnical and Structures Laboratory
Vicksburg, MS 39180
Niki.C.Goerger@usace.army.mil

LTC Robert H. Kewley
U.S. Military Academy
Department of Systems Engineering
West Point, NY 10996
Robert.Kewley@usma.edu

Robin K. Burk
U.S. Military Academy
Department of Electrical Engineering & Computer Science
West Point, NY 10996
Robin.Burk@usma.edu

The use of ontologies to represent data and knowledge combined with service-based software provides new opportunities for the integration of command and control systems, simulations, and dynamic data. The U.S. Department of Defense's Global Information Grid (GIG) is envisioned to integrate complex communications networks, data from disparate sources, and services-oriented applications. The GIG will provide analysis, decision support, and information visualization to both human and automated users. This paper describes concepts and architecture for an experiment which demonstrates the use of knowledge-based technologies to support tactical maneuver. Open source and in-house software were linked together to conduct a simple simulation of a realistic mission (move from A to B along a "safe" route). Events which may affect the route are injected, reported, and stored in the knowledge base to simulate battlespace dynamics. Through the use of an automated reasoner, events which affect the route are identified and passed to the decision maker to stimulate possible replanning. Follow-on efforts will convert the maneuver related software into web services, moving closer to the GIG concept. Discussed is the value of a formal ontology within the domain of military maneuver, architectural approach, and lessons learned.

Keywords: ground vehicle mobility, ontology, assured mobility, global information grid, data modeling, service oriented architecture, vehicle route planning, simulation

1. Introduction

The Global Information Grid (GIG)[1] is emerging as the next-generation architecture to enable network-centric operations (NCO) and support shared situational awareness (SA). The target vision for the GIG as stated in the GIG Architectural Vision, June 2007, is "for an agile, responsive, and unified GIG that enables the Department [of Defense] to fully leverage the power of information and collaboration across the Enterprise to the forward edge of the battlespace" [1]. To achieve the vision, it will be necessary for users to find and exchange the data and information they need in a timely, flexible, and effective manner.

This paper describes an initial demonstration prototype for information services of the sort envisioned for the GIG. Concepts and the architecture for a simple experiment which investigates and demonstrates the use of knowledge-based technologies to support tactical maneuver were developed. The work builds upon our previous efforts to create a data model and ontology for the Mobility Common Operational Picture (M-COP). We describe a military operations demonstration which makes use of potential web services utilizing portions of the M-COP combined with an ontology and movement and maneuver algorithms. The functionality provided includes route planning and monitoring combined with intelligently replanned routings based on external events and informed by reconnaissance data.

Specifically, we have created a demonstration of services that contribute to creation of a Common Operational Picture to aid commanders in planning ground maneuvers and responding to unplanned events in the battlespace. In so doing, we bring together several initiatives: the M-COP, current practices regarding the in-theater use of intelligence information, semantic technologies (specifically, formal ontologies) and research initiatives including Common Maneuver Networks based on the U.S. Army Engineer Research and Development Center's (ERDC) Battlespace Terrain Reasoning and Analysis (BTRA) framework. The resulting

[1] "The globally interconnected, end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers and support personnel." (DoD Directive 8100.1, GIG Overarching Policy, September 19, 2002, Enclosure 2.)

The July 2006 edition of CrossTalk http://www.stsc.hill.af.mil/crosstalk/2006/07/index.html provides several articles concerning Net-Centric Information and the GIG.

demonstration prototype will serve as the basis for a series of network science experiments aimed at providing insight into design, usability, and complexity issues associated with providing mobility-related information services to ground commanders.

1.1 The Mobility Common Operational Picture (M-COP)

To support integrated joint operations by ground, air, and sea forces, the Joint Chiefs of Staff have called for a Common Operational Picture (COP) to be generated and provided to all commands [2]. The COP is presented as a single identical display of relevant information shared by the commands and allows drill-down to supporting data [3]. Generation and dissemination of regularly-updated COP displays is the responsibility of software services provided by the GIG.

In addition to the overall COP, the Joint Chiefs have called for generation of a M-COP to support operational commanders [3]. The M-COP presents information needed to determine where to maneuver in the battlespace. It must support and in some cases provide the integration and analysis of extensive information about the terrain, friendly and hostile forces, equipment, weather, and other factors to facilitate planning and execution of movement by troops, manned equipment, and autonomously mobile robotic systems across the battlespace. Creating the M-COP requires creating a unified knowledge space regarding ground vehicle mobility and maneuver in the current and future force.

1.2 Current In-theater Use of Intelligence Data Relating to Ground Mobility

In current ground operations, analysts collect information from various reporting sources in-theater to construct and maintain Route Status Advisory documents. These documents provide maps of various locations overlaid with graphics to show segments of movement routes used for conducting various missions. Color coding of the route segments indicates status, such as “route closed,” “high threat,” “medium threat,” or “low threat.” Moreover, annotations on the maps provide additional detail regarding recent incidents that can affect movement through the area, such as high levels of Improvised Explosive Device (IED) activity or information on damage sustained to bridges. The collective set of advisory information represents judgments made by intelligence staff based on their interpretation of reported information. The staff has to consider the quality of the data, the credibility of the sources of the reports, and the operational environment in preparing their assessments.

As data volumes increase in more highly automated information architectures such as the GIG, sophisticated information processing will be required to assist staff members in evaluating and correlating the information reports. Traditional software development has approached this requirement through specialized code and database management systems that have proven expensive to develop, even more expensive to maintain, and prohibitive to agility. This specialized code is an analog of the logical processes exercised by the human analysts. As conditions change in the maneuver space, the human analyst fluidly adapts logic. However, specialized software code requires expensive changes.

What if the knowledge and logical processes exercised by the human analysts could be captured in data structures in such a way that the software engineering problem became one of data update rather than code change? Semantic technologies, specifically formal ontologies, enable this shift of focus for software architecture and design.

1.3 Capturing Meaning, Not Just Data: Formal Ontologies

Today's web-based standards provide the means for capturing knowledge in formalized ontologies against which logic-based software tools can reason about the implications of new information. Ontologies play a major role in the emerging Semantic Web and are widely used in knowledge management systems, biomedical informatics, medical terminologies, and artificial intelligence applications. They are also of increasing importance in the GIG, where they may be used to support the discovery, composition, execution, and monitoring of GIG services.

There have been many attempts to define an ontology. Currently the most commonly accepted definition in the informatics community is "*an explicit specification of a conceptualization*" [4]. A formal ontology of this sort includes a vocabulary together with specification of the meanings of the terms in the vocabulary. This specification includes identification of the fundamental categories in the domain and identification of the ways in which members of the categories are related to each other, along with constraints on the ways in which the relationships can be used.

Constructing an ontology is important in the development of knowledge-based systems since ontologies capture shared understanding about the key concepts associated with some domain and about the properties of objects within the domain. The advantages of ontologies have been widely described, and include enabling sharing of knowledge, reuse of knowledge, and better engineering of knowledge-based systems with respect to acquisition, verification, and maintenance.

In the context of software and information systems, ontologies are usually expressed in a logic-based language, so that detailed, accurate, consistent, and meaningful distinctions can be made among the classes, properties, and relations of the ontology. Ontology tools have also been developed which perform automated reasoning by applying the concepts and rules in the ontology to a knowledge base which reflects specific facts known about a given domain situation.

1.4 Network Science and the GIG

In order to realize the Department of Defense's vision for Network-Centric Operations and the supporting GIG architecture, networked information, hardware, and social systems of unprecedented complexity must be created. In 2006, the National Research Council (NRC) of the National Academies issued a report [5] which identified challenges inherent in this complexity and which recommended that the Army take a number of steps to foster the emerging new discipline of Network Science. The boundaries and content of the discipline are still evolving, but there is general agreement that it addresses the creation, growth, and behavior of highly complex physical, data, cognitive, and social networks including those which play a central role in Network-Centric Operations.

In response to the NRC's report, in April 2006 the U.S. Military Academy (USMA) at West Point, NY, stood up a new center of excellence for Network Science. The USMA Network Science Center encourages cross-disciplinary network science research, both foundational and applied, on the part of faculty and cadets and in cooperation with other academic and Army organizations. Initial projects include identification of key predictors for jihadi/insurgent social network changes and design of a dismounted strike warning and tracking system, including evaluation of the communications, cognitive, and operational impacts of its deployment for IED attacks.

In keeping with the Network Science initiative, USMA's Operations Research Center joined with the Network Science Center to initiate a series of experiments regarding the network implications of ontology use in GIG software services based on the M-COP.

2. Background

To achieve information dominance, information must flow seamlessly between Battle Command (BC) systems and other information sources and users, including models and simulations (M&S). Current embedded training and BC systems do not share tactical maneuver data. The battlespace COP is therefore inconsistent between these systems potentially leading to severe consequences from incorrect decisions about maneuver potential during training, planning, rehearsal, and execution of operations. Previous work by this team developed the concept of a M-COP for ground vehicle movement and maneuver, and defined data components, an ontology, and potential services [6] - [11]. The data model identified what information is important in planning ground vehicle movements and provided a standardized data element dictionary suitable for data mediation among legacy application software. The ontology provides a platform and application independent formalization of the concepts and relationships represented in that information. The services identified software applications that could be distributed across the net-centric architecture for discovery, access, and use by any software on the network.

Although only a subset of the overall COP, the M-COP is a challenging mix of information provided by decision aids, environmental databases, platform performance data, doctrinal behaviors, and simulation processing that will be distributed across the GIG. These sources of data and processes use a variety of data models that need to be reconciled through metadata and data mediation to enable the information to be effectively merged to create the M-COP.

The M-COP is expected to be provided to commanders in the form of one or more web services available via the GIG. A web service is an abstract resource or interface that provides a set of tasks comprising coherent functionality from the point of view of providers' entities and requesters' entities. A number of well-established technical specifications such as the Web Services Description Language (WSDL) exist for the definition and publication of these tasks. To be used, a service must be realized by a concrete provider agent. Under the web service architecture model, service providers advertise their availability to requesting software. New requestors may be added and old ones die off without requiring modifications to the provider agent. This loosely coupled architecture is an instance of the Services Oriented Architecture (SOA) approach which has significant industry sponsorship.

A web service which provides integrated information in a standard format need not be – and often is not – the originating collector of the source data. A large part of the value of the SOA approach is that it specifically decouples these actions, allowing the addition or evolution of new data sources without disruption to users of the data. Achieving the envisioned capabilities of the mature GIG will require parallel development of sensors, systems, and sources for data collection, on the one hand, and the definition and instantiation of service tasks that integrate, evaluate and present information based on that data, on the other hand. The scenario against which we developed the prototype service described here (Section 3 below) assumes the availability of real-time or near real-time data regarding conditions and events in the battlespace. We assume, and believe it to be likely, that timely collection and integration of intelligence, surveillance, and reconnaissance (ISR) data will increasingly be automated as net-centric equipment is fielded and the GIG matures to support fully networked operations.

2.1 Movement and Maneuver

In force-on-force simulations or in actual ground vehicle movement planning, the principal goal of long-range route finding (as opposed to short-range or reactive planning, which is concerned with collision avoidance) is to be able to move to a given point avoiding obstacles (buildings, rivers, etc.), following a road, trail, or off-road path. In more advanced route finding, the route can be optimized based on a criterion (on-road only, off-road, shortest, fastest, most concealed, safest, etc.). One way to do this is to develop a network graph which represents the maneuver environment and to apply a cost to each arc or segment of the network. In the simplest case, the length of an arc is the cost, but costs can also be associated with the terrain and estimated vehicle speeds. The network data is stored in a forward star structure – the most efficient format for representing networks [12]. Mathematically, this is known as a shortest path problem, and two well-known solution methods are Dijkstra's [13] and the A* [14] algorithms. Richmond *et al.* [15] describe the implementation of route planning based on a maneuver network in the OneSAF Test Bed.

In the simplest case, a maneuver network is the local road network made up of road segments between road intersections or locations of interest. Current commercial web-based systems and navigation systems use this methodology for determining routes based on shortest or fastest criteria. Software that supports development of ground maneuver networks for military operations is more complex since it must represent both the local road network and the potential for maneuver off-road (cross-country). One such software system under development is the Commercial Joint Mapping Toolkit[2] (C/JMTK). C/JMTK is the Department of Defense's standard geospatial and visualization toolkit based on Environmental Systems Research Institute, Redlands (ESRI), CA products. The U.S. Army ERDC BTRA research project is also producing tools which support ground vehicle mobility analysis and tactical maneuver planning. For actual unit route planning, it uses ESRI's proprietary network analyzer package, which utilizes the A* algorithm. A specific product of interest is a maneuver network which can be exported as a shapefile. Formal documentation of the BTRA maneuver network attributes is limited at the time this paper is being written; however, it currently contains approximately 56 attributes [16] which can be used to develop a cost for finding a route. Many of these attributes are based on the vehicle-terrain interaction of the 12 vehicle classes represented in the standard mobility (STNDMOB) application programming interface (API) [17].

Although initial planning produces an optimal maneuver network, during military operations it cannot be assumed that the network remains static. Events can and probably will occur which may significantly impact the network (e.g., bridges can be destroyed, tactical bridges emplaced, and hostile forces detected). As a result, the knowledge or data (in our demonstration the ontology knowledge base) which represents the M-COP will also be dynamic. To support commanders with real-time situational awareness and enhanced sensemaking during operations, M-COP software should monitor battlefield events, identify changes of interest, and initiate appropriate actions (e.g., notifications to replan) automatically.

3. Scenario

In order to illustrate these dynamic planning capabilities, the research team developed the following notional scenario. A patrol of four Stryker M1127 Reconnaissance Vehicles is on a

[2] http://www.cjmtk.com/

mission to meet with the local mayor of the city at 0800 at the City Hall. Prior to departure, the patrol is given a route that has been cleared within six hours prior to departure from the unit's forward operating base. This route takes into consideration road and bridge conditions, other convoys on the road, IED and ambush threats, and other conditions that could affect the movement. Intelligence has told the patrol that there are two IED hot spots and two bridges along the selected route. The patrol begins its travels cautiously along the route. As the patrol progresses along the route, a bridge ahead of its position is destroyed. Passage along the selected route becomes impossible. When the unit receives the information, it will need to replan a route to its objective. The unit is part of the network-enabled battle command system and has registered its route with a route monitoring service. The route monitoring service is able to gather data and information from various sources regarding conditions and activities in the operational environment. Moreover, the service uses the ontological structure of the knowledge base to reason whether changes in the operational environment may impact the patrol. If there is a need to alert the patrol, the service does so. The route monitoring service picks up information regarding: (a) there is a destroyed bridge; and (b) the damaged bridge location. Since the bridge is along the intended route, and the unit has not yet crossed the bridge, the route monitoring service makes a notification through the battle command system that the route is infeasible and the patrol employs the route finding service to identify a new route in near real time. The route finding service contains updated data (timely data) concerning conditions in the operational environment (such as any other damaged bridges, IED incidents, convoys on roads, traffic reports, non-combatant traffic flow, congestion on routes, etc.), and plans the new route accordingly. The route finding service then provides the information via command systems to the patrol. The patrol proceeds along the second route. While en route, an ambush is reported along this second route, requiring a second alert and a replanning cycle to compute a third route to reach the objective. Vehicles proceed along this new route to finally reach City Hall. Figure 1 provides an illustration of this scenario using the Joint Readiness Training Center (JRTC) at Fort Polk, LA, as the geographical setting.

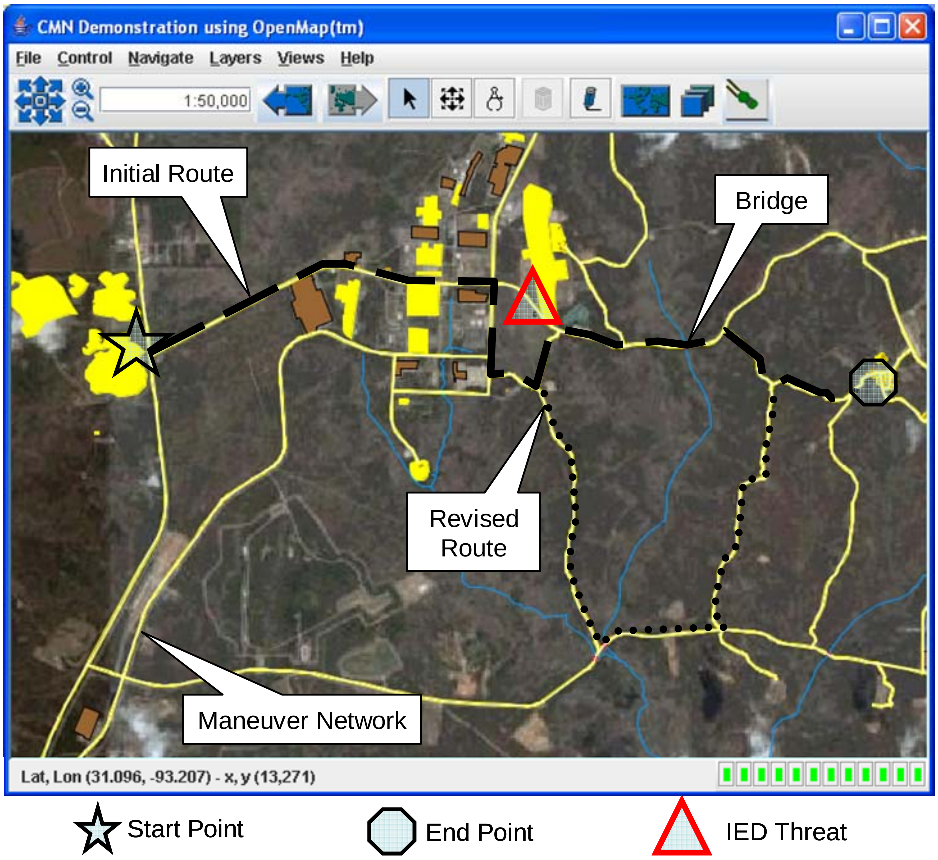


Figure 1. Scenario demonstration GUI, Joint Readiness Training Center (JRTC), Fort Polk, LA.

4. Demonstration

In order to demonstrate the above scenario using simulation, a conceptual demonstration architecture was developed. Shown in Figure 2 is the initial concept, with the services interacting with the OneSAF simulation and a Battle Command System (BCS) through network connections. As an initial experiment, we replaced the BCS and network with a simple Graphical User Interface (GUI) and an OpenMap^{TM} viewer, connected to a discrete event simulation (DES) API (Simkit) acting in place of OneSAF. This simple map-based DES allowed animation of the vehicle movement in the previously described scenario. Lines in Figure 2 indicating web service requests were handled by Java-based software running multiple processes using the API's described below. Figure 3 shows a process diagram indicating user inputs and resulting code processes.

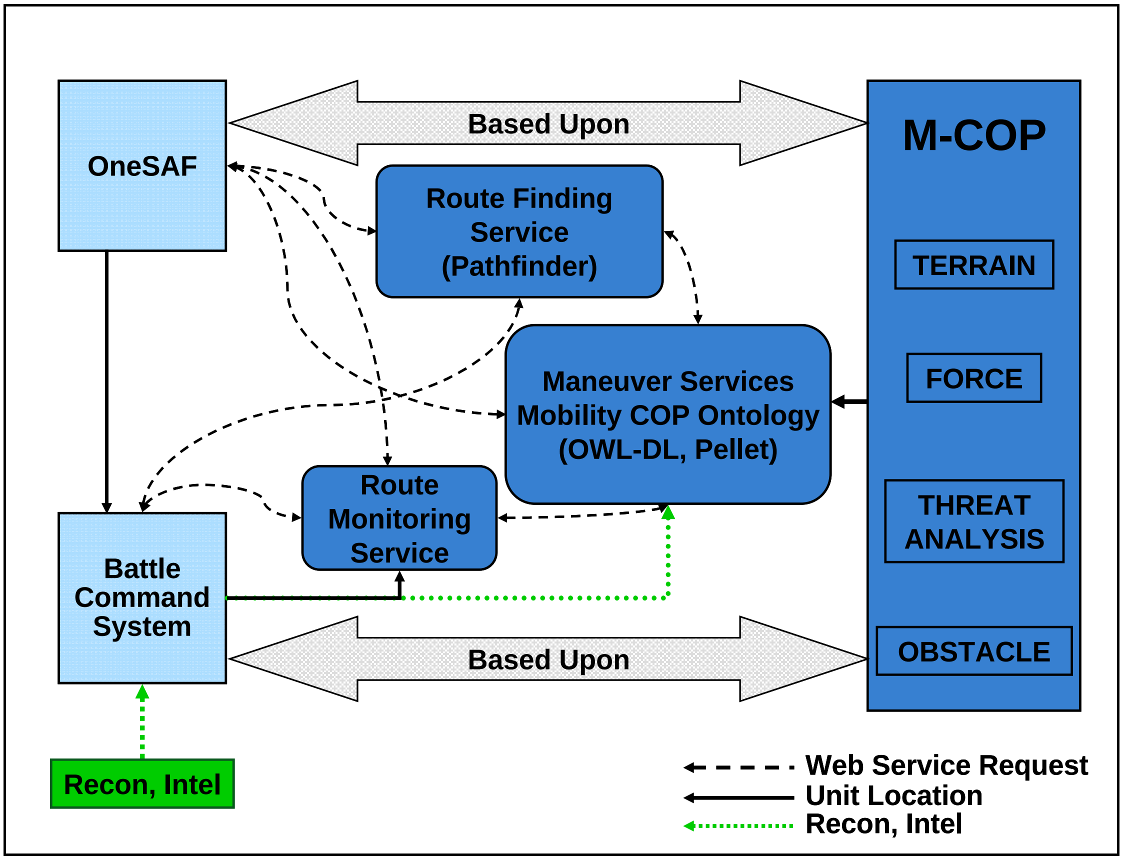


Figure 2. Conceptual demonstration architecture.

4.1 Visualization

A visualization tool was developed for the demonstration based on the open source software OpenMap™. OpenMap is a JavaBeans™-based geospatial toolkit (BBN Technologies, Cambridge, MA) and is available from the OpenMap website[3]. Using the standard OpenMap API, code was developed and integrated to display the maneuver network, call the route planning service, display the resulting route, and plot locations of the vehicles. The visualization tool displayed the results of the services and dynamic effects with simple vehicle icons that follow the provided routes at constant speeds.

[3] http://openmap.bbn.com/

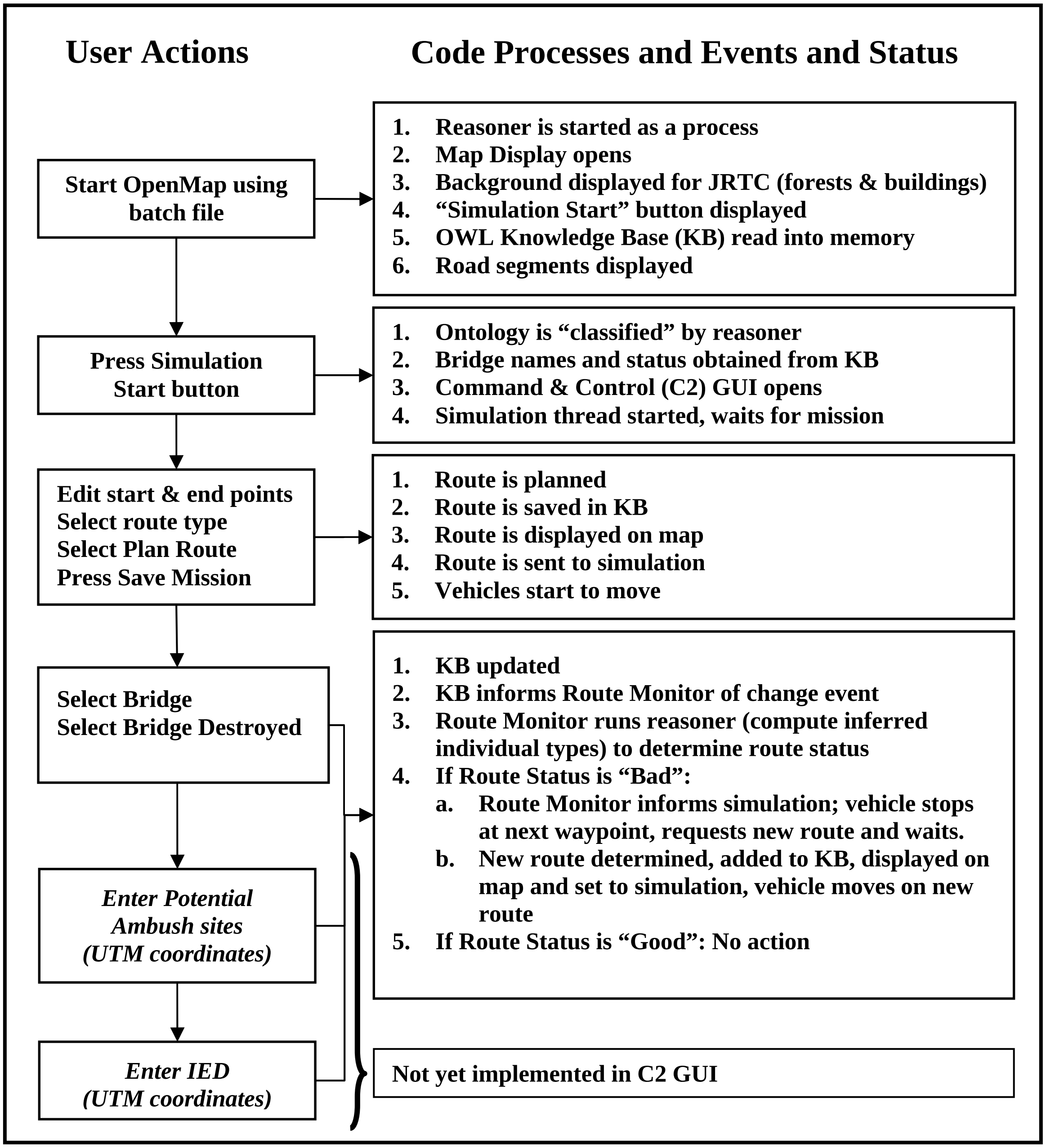


Figure 3. Scenario and code processing flow diagram.

4.2 Simulation Engine

Simkit[4] is an open source, DES engine. Buss and Sánchez [18] discuss how to simulate vehicle movement, including a simple methodology for animation. We followed their methodology and assumed steady vehicle speeds (no acceleration) as our intent was not to simulate vehicle performance, but only create a dynamic representation of its location along a route. We

[4] https://diana.cs.nps.navy.mil/Simkit/

considered using a simple velocity equation to determine vehicle location, but by using Simkit, we allow future experiments which could involve preprogrammed events initiated by the simulation.

4.3 Battle Command System GUI

A full set of M-COP software services on the GIG will require integration of route planning, monitoring and updates with data sources (e.g., potentially including intelligence estimates of likely IED danger on a route segment) and with battle command systems. Section 6 discusses our follow-on plans in this regard, along with the opportunities for use of this system in training and in the evaluation of responses to new enemy tactics. For the initial experiment, we used a simple GUI shown in Figure 4. Intelligence "updates" are based on the scenario, and are manually entered into the GUI, initially allowing only bridges to be destroyed. As these updates are received, they are injected into the knowledge base.

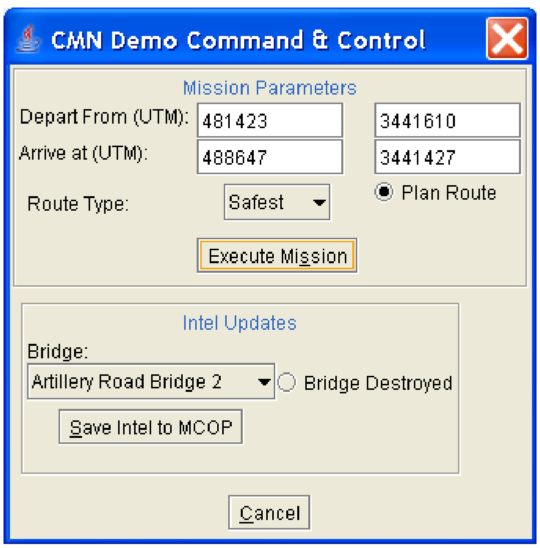


Figure 4. Demonstration C2 GUI.

4.4 Services

Table 1 contains our initial web services descriptions. There are four services defined to support our scenario: Route Finding, Route Monitoring, and two Maneuver Services -- Update and Query. Route Finding consists of determining a series of points based upon the maneuver network, the desired route type (fastest, shortest), starting and destination points, along with any points in between that should be visited (way points). A shortest path algorithm is used to produce a list of points (end points of each segment in the route), which are then used by the vehicle during movement. The Footprint-to-Pathfinder project, formally entitled "Integration of Urban Characterization, Munitions Effects and Threat Assessment for Ground Vehicle Planning in Urban Environments," was a three-year effort sponsored by the Army G-3 Battle Command Simulation and Experimentation Directorate[5] as part of an initiative to improve representation of

[5] Deputy, Army Modeling & Simulation Office, HQDA, G-3/5/7 Battle Command, Simulation, and Experimentation Directorate

urban operations in models and simulations. The target simulations for this project were the Combined Arms Analysis Tool for the 21st Century (Combat XXI) and One Semi-Automated Forces (OneSAF). One of the project objectives, to represent vehicle route planning in urban environments, resulted in Java code for performing vehicle route planning based on network graphs, which were used in this effort (see [19] for a description of the Footprint-to-Pathfinder path-finding algorithm and overall review of path-finding methodologies). The code allowed us to develop cost functions based on distance to objective and threat encountered along the route.

Route monitoring implies that an application or service is aware of a previously determined route and that the knowledge base is monitored for changes; e.g., a bridge that lies on the route is subsequently destroyed, so it will now have an increased cost to traverse. Based on the current position of vehicles along the route, the route monitoring service should notify the vehicles that the route is no longer viable.

Table 1. Application/Service descriptions.

Service	**Description of Service**	**Input/Output Descriptions**
Route Finding	A least cost route for ground vehicles and units	Input • Route Start, End, *and Waypoints*[1], *Unit or vehicle type, and formation* • *Route type (On-road, Cross-country, both)* • Planning factor (*time*, distance, threat, …) • *Command Speed* • Route name Output • Waypoint list, estimated average speed between waypoints, *estimated total time, and distance* • *Turn by turn directions*
Route Monitoring	Notification that a route is no longer viable, based on a planned route, current unit position and route influencing events	Input • Route Name (implies unit assigned) • *Events that can effect viability of route* Output (only if and when route becomes invalid) • Notification *and reason for route change status*
Maneuver – Update	Maintains a maneuver network graph for route planning.	Input • Updates to M-COP data
Maneuver – Query	Responses to specific queries related to M-COP.	• Request for asserted & inferred data (queries) - Maneuver network - *Status of specific route edges* - *List of damaged bridges* - *List of threats to routes (candidate targets)* - *List of inferred obstacles* - ….. Output (responses to specific queries) • Maneuver network (edges, nodes and cost)

		• *Status of specific road edges* • *Specific lists (as above)*
[1] italicized terms are future or potential capabilities		

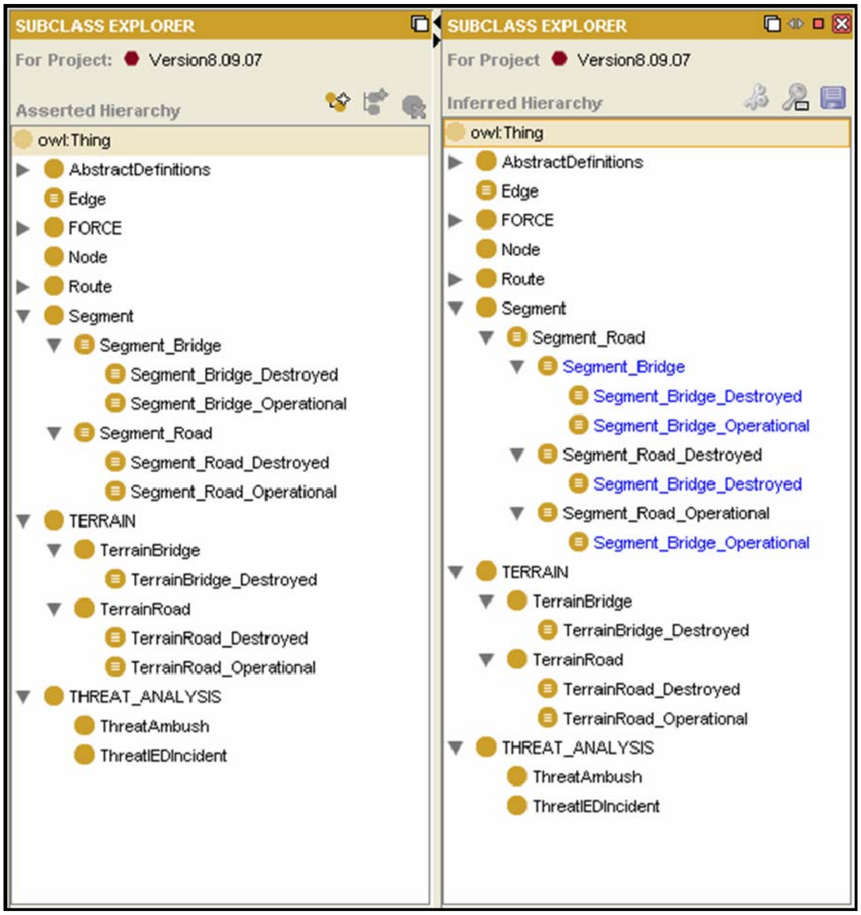
The maneuver services are based on an ontology developed to support our scenario. The Web Ontology Language (OWL) is a language for defining and instantiating web ontologies and has become the standard for representing ontologies on the Semantic Web[6]. An OWL ontology consists of individuals, properties, and classes. Individuals represent objects in the domain of discourse. OWL properties represent relationships between two individuals. OWL classes are sets that contain individuals. They are described using formal mathematical descriptions that state precisely the requirements for membership of the class. Classes may be organized into a superclass-subclass hierarchy, which is also known as a taxonomy. Subclasses specialize ('are subsumed by') their superclasses. In this paper, components (class, property, and individual names) of the M-COP ontology are written in a Franklin Gothic Book font like this.

One of the key features of ontologies that are described using OWL-DL (Description Logics) is that they can be processed by a reasoner. OWL-DL is based on Description Logics (DL), a decidable fragment of First Order Logic and therefore amenable to automated reasoning. A reasoner that understands the semantics of OWL can automatically compute the classification hierarchy and check for inconsistencies in an ontology that conforms to OWL-DL.

The demonstration M-COP ontology was implemented, refined, and evaluated using the Protégé-OWL[7] open source ontology development tool. In Protégé-OWL, the 'manually constructed' class hierarchy is called the 'asserted hierarchy'. Figure 5(a) shows the initial asserted hierarchy M-COP ontology. In our scenario, the vehicles (class FORCE) are traveling along a road. Each road consists of a series of class Segment(s). In this demonstration, each class Segment is a named class[8] Segment_Road and if a bridge is present, it is also a named class Segment_Bridge. The properties of the named classes contain information on the number of lanes, speed limit, damage codes for the condition of the class Segment, and threat codes for the presence of IEDs or ambushes.

A DL reasoner, Pellet, was used to compute the ontology class hierarchy [20]. Pellet is open source, was originally developed at the University of Maryland's Mindswap Lab, and is now commercially supported by Clark & Parsia LLC (Washington, DC)[9]. The Pellet reasoner was used to check three important characteristics of the ontology: consistency, ensuring that it does not contain any contradictory facts; concept satisfiability, determining whether it is possible for a class to have any instances; and classification, computing the subclass relations between every named class to create the complete class hierarchy.

[6] http://www.w3.org/TR/owl-features/
[7] http://protege.stanford.edu
[8] OWL classes fall into two main categories – named classes and anonymous (unnamed) classes. Anonymous (unnamed) classes are formed from logical descriptions. They contain the individuals that satisfy the logical description. A named class is an explicitly defined anonymous class. Only individuals that satisfy the restrictions may belong to this named class.
[9] http://pellet.owldl.com/



(a) (b)

Figure 5. (a) Asserted class hierarchy and (b) inferred class hierarchy shown in the Protégé GUI.

In an ontology class hierarchy a class can be a subclass of several classes. The class hierarchy that is automatically computed by the reasoner is called the inferred hierarchy. If a class has been reclassified, then the class name will appear in a blue color in the inferred hierarchy in the Protégé GUI (Figure 5(b)). One of the major benefits of building an ontology using OWL-DL is that a reasoner can automatically compute the class hierarchy and find any inferred hierarchies. Inferred hierarchies represent what the ontology will look like, not how it was built. When very large ontologies are built, the use of a reasoner to compute relationships becomes imperative. Building the initial asserted hierarchy and then using the reasoner to find any multiple inheritance keeps the process clean. Figure 6 and 7 show the asserted and inferred hierarchies, respectively, for a Segment. It is clear to see that the asserted hierarchy is simpler than the more complex inferred hierarchy.

In addition to the classification hierarchy expressed in OWL, rules can be constructed to express conclusions to be drawn based on what is currently known about the properties or attributes of the classes. The Semantic Web Rule Language (SWRL) is an emerging standard for expressing rules [21]. For example, in our scenario we used a rule that states that if a Route (?x) contains Segment (?y) that has property hasDamageCode with value of Destroyed, then the Route is to be considered destroyed:

Route (?x) ^ Segment (?y) ^ hasSegment (?x, ?y) ^ hasDamageCode (?y, Destroyed) →

hasDamageCode (?x, Destroyed)

The reasoner will reassess defined rules as conditions change in the operational environment (for our scenario, when new data was inserted into the ontology). The Protégé GUI was used to design, populate, and test the knowledge base and SWRL rules prior to implementation in the Protégé-OWL API for execution in the demonstration code. Once the demonstration ontology was fully developed, we used several data sources to populate the initial knowledge base. Geospatial vector ESRI shapefile formatted data was used to create the JRTC maneuver network.

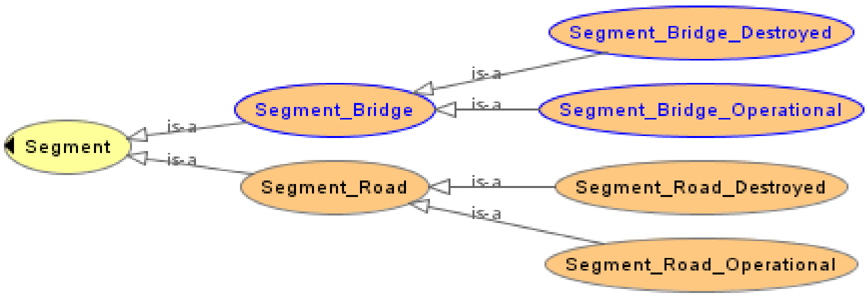


Figure 6. The asserted hierarchy for the class Segment.

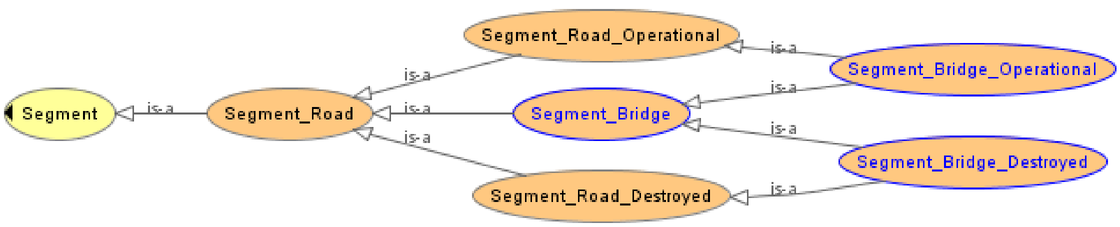


Figure 7. The inferred hierarchy for the class Segment.

5. Results and Discussion

We have developed a relatively simple scenario with a supporting knowledge base and have used this to provide situational awareness to a simulated unit. By connecting the knowledge base to software that conducts route planning and monitoring, the unit can be notified of events potentially impacting its operations. This assessment and notification was accomplished using capabilities developed in OWL, SWRL, and open source API's. This research and development demonstrates a proof of concept and sets the stage for extending capabilities to support situational and shared awareness. The lessons learned and insights gained from this work will prove valuable in making extensions to implement more complex scenarios and associated reasoning.

We found that the ontology was hard to design in meaningful ways (in other words, in ways that really add value over writing procedural code and "exhaustive" enumeration of possible cases) while trying to make use of embedded knowledge with the use of a reasoner. Ontology design is an iterative process – model, apply, model. We started with a model of the ontology that included the M-COP categories terrain, weather, obstacles, forces, maneuver analysis, threat, and utilities. Selecting a focused and scoped use case to instantiate caused us to revisit and restructure components of the ontology. Fleshing out the ontology required continuing iterations of modeling the ontology, instantiation of use cases, and refining the ontology.

The Protégé application provided us with both a GUI in which to develop the ontology and access to the API with which to access the ontology programmatically. We found that the inferred classes could not be developed based on floating point numbers but by creating nodes indexed with a unique ID and attributes of floating point coordinates. This also led to some "by hand" adjustments to our source data to insure that the bridges exactly aligned with roads. Creating the classes which were combined to form an abstract network for route planning was problematic and may have resulted in redundant information being stored in the knowledge base (roads, segments, and edges contain overlapping attributes).

It is true that any particular inference that can be achieved in software using reasoning techniques against a knowledge base can also be achieved with standard procedural code. At issue, therefore, are the relative advantages and disadvantages obtained through the use of the formal ontology in the construction of prototype M-COP services.

It is perhaps not surprising that we experienced some difficulty in crafting the formal ontology we produced for this prototype. Although the ontology was small as measured by the number of concepts and attributes it describes, it was ambitious in that it deliberately attempted to formalize the conceptual relationships that contribute to human judgment-making regarding (for example) trafficability of a given road segment by a given type of vehicle as a result of specific events. Ontologies that describe relatively tangible, objective entities are straightforward to define. Such entities and their attributes are also straightforward to represent in relational databases. For such applications, the tradeoff between data structure and code complexity using standard software architectures versus those using the ontology and reasoner can be quantified using common complexity metrics.

The M-COP route-related ontology was more complex, and more difficult to define, because it embodied knowledge and human judgments that were less constrained by physical relationships. There were, for instance, several alternative ways we considered representing the relationship between the road segments that reside on an overpass with the road segments over which it crosses. Unlike relational databases, where the rules of normalization guide proper data table and index definition, with ontologies the deciding criteria have to do with the relatively open-ended range of inferences which might be supported by the ontology design choice.

The advantages of using an ontology to provide M-COP services for the GIG are realized, not with regard to the initial iteration of software development, but rather in the flexibility and power with which software can approximate human judgments in complex situations as the ontology and the knowledge base grow. Ontologies address what Alberts and Hayes [22] call the cognitive dimension of the network-centric battlefield. They embody, not data exchange standards (which are syntactic in nature), but rather the semantics or meaning of events and data. Thus, the partial M-COP ontology defined for this prototype enables the key task of sensemaking and of understanding the implications of an event (in this case, the mobility implications of a destroyed bridge).

As described in [11] we found that neither a top-down concept decomposition nor the techniques of formal concept analysis were adequate tools to guide the development of the ontology for this purpose. Instead, we made progress by working from the bottom up: defining classes, attributes, and class/sub-class relationships associated with physical equipment, terrain features, etc. Once those were in place we then turned to the use of description logic to capture the more complex cognitive relationships associated with making judgments about trafficability. The IEEE's Suggested Merged Upper Ontology [24] attempts to provide very high-level concepts under which increasingly detailed domain-specific ontologies can be defined by industry standards groups. While this approach may well be useful for such tasks as searching the Semantic Web for news articles or for online access to application databases, our experience suggests that a bottom-up approach is more feasible for providing battlefield commanders with technical aids that reduce the cognitive load of tracking local conditions while maintaining appropriate situational awareness in combat and near-combat operations.

The demonstration project described in this paper begins, but by no means completes, the ontology needed to fully take advantage of machine reasoning regarding mobility. An IED which damages an overpass is just one example of an event which humans would judge as creating an obstacle to movement through a planned route. The earlier papers regarding the M-COP data model suggest other such events, e.g., encountering a damaged vehicle that partially blocks a road lane. In such a scenario, an ontology that describes both roads and classes of equipment would provide a flexible and powerful means for inferring whether or not the damaged vehicle will in fact obstruct movement of the particular unit currently assigned to that route. Accounting for all such alternatives would require significant amounts of procedural code. Moreover, adding new code in response to changing enemy tactics would be cumbersome and error-prone. The use of an ontology and knowledge base, on the other hand, focuses changes on Soldier understanding of those tactics – how to discern and respond to them. Changes would need to be made to the ontology and knowledge base, but would not destabilize existing software functionality.

It is not surprising that capturing professional judgments at the cognitive, sensemaking level is difficult. An investment in doing so, however, promises to equip our Soldiers with software services that provide targeted, actionable information and advice (in this case, regarding route changes in response to events that may be outside their direct awareness). A key challenge in network-centric operations is to provide quality information while avoiding information overload. Reasoning against an ontology does not offer short-term advantages in time or effort for initial software development. It does, however, promise such advantages over the life cycle of applications. More importantly, it offers ways to provide judgment-rich, targeted information services to Soldiers in high-stress environments.

6. Future

The prototype M-COP services constructed for this demonstration require expansion in several areas. Our next step is to integrate with current Army simulation and Battle Command systems being used in experimentation. We will do this using the Geospatial Battle Management Language (GeoBML) demonstration architecture [23] as a starting point. This expanded system will be the platform for a series of network science experiments to be performed by the USMA Operations Research Center, a partner of USMA's new Network Science Center. Successful implementation of the GIG in support of new capabilities envisioned by the Army's Future Combat Systems and Future Force will require a deeper and more quantitative understanding than we currently have regarding computational and network traffic loads, tactical implications, and user interface impacts. We plan to exercise the expanded prototype in a series of experiments designed to provide insight into these issues. A key step in the design of these experiments will be to explicitly define and defend measures of effectiveness for a software service that provides this sort of low-level sensemaking to Soldiers. We anticipate that these measures will address both technical performance of the system and also the impact on Soldiers of its use. It is our hope and expectation that this prototype and the experimental results we gather will contribute to success as the GIG and network-centric operations move from concept to implementation.

7. References

[1] Department of Defense (DoD). *Vision for a Net-Centric, Service-Oriented DoD Enterprise.* June 2007. Available from http://www.defenselink.mil/cio-nii/docs/GIGArchVision.pdf

[2] Joint Forces Command. *Capstone Requirements Document: Global Information Grid (GIG)*. JROCM 134-01. 30 August 2001.

[3] Joint Chiefs of Staff. *Doctrine for Joint Operations, JP 3-0*. 10 September 2001. Available from http://www.dtic.mil/doctrine/jel/new_pubs/jp3_0.pdf

[4] Gruber, T. A translation approach to portable ontology specifications. *Knowledge Acquisition* 1993; 5(2): 199-220.

[5] National Research Council. *Network Science*, The National Academies Press, 2006

[6] Richmond PW, Blais CL, Nagle JA, Goerger NC, Gates BQ, Burk RK, Willis JB, Keeter R. "Standards for the Mobility Common Operational Picture (M-COP): Elements of Ground Vehicle Maneuver," U.S. Army Engineer Research and Development Center, Geotechnical and Structures Laboratory; 2007. Report No.: ERDC/GSL TR-07-4.

[7] Blais CL, Goerger NC, Richmond PW, Gates BQ, Pace M. "Data Mapping and Ontology Design for Common Maneuver Networks," Paper 05S-SIW-140, Spring Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, San Diego, CA, 2005.

[8] Blais CL, Goerger NC, Richmond PR, Gates BQ, Willis JB. "Global Information Grid Services and Generation of the Mobility Common Operational Picture," Paper 05F-SIW-107, Fall Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, Orlando, FL, 2005.

[9] Goerger NC, Blais BL, Gates BQ, Nagle JA, Keeter, RR. "Toward Establishing the Mobility Common Operational Picture: Needs Analysis and Ontology Development in Support of Interoperability," Paper 06S-SIW-044, Spring Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, Huntsville, AL, 2006.

[10] Nagle JA, Blais BL, Burk RK, Gates BQ, Goerger NC, Richmond PW. "The Mobility Common Operational Picture Data Model: A Foundation for Conceptual Interoperability in the Domain of Ground Vehicle Mobility and Maneuver" Paper 06F-SIW-005, Fall Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, Orlando, FL, 2006.

[11] Burk RK, Goerger NC, Gates BQ, Blais BL, Nagle JA, Goerger S. "Knowledge Representation for Military Mobility Decision-Making by Humans and Intelligent Software". In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*. Springer. New York, 2007.

[12] Gallo G, Pallottino S. "Shortest paths algorithms." *Annals of Operations Research* 1988;13: 3-79.

[13] Dijkstra EW. "A note on two problems in connexion with graphs." *Numerische Mathematik* 1959; 1: 269-271.

[14] Sedgewick R, Vitter, JS. "Shortest paths in Euclidean Graphs." *Algorithmica* 1986; 1(1): 31-48.

[15] Richmond PW, Scoggins RK, Gates BQ, Yamauchi H. "Common Ground Vehicle Route Planning for Army Simulations" Paper 07S-SIW-062, Spring 2007 Simulation Interoperability Workshop, Norfolk, VA, 2007.

[16] Blais CL, Goerger NC, Richmond P, Gates B, Pace, M. "Data Mapping and Ontology Design for Common Maneuver Networks" Paper 05S-SIW-140, Spring 2005 Simulation Interoperability Workshop, San Diego, CA, 2005.
[17] Baylot EA Jr, Gates BQ, Green JG, Richmond PW, Goerger, NC, Mason GL, Cummins CL, Bunch LS. "Standard for Ground Vehicle Mobility." Vicksburg (MS): U.S. Army Engineer Research and Development Center, Geotechnical and Structures Laboratory; 2005. Report No.: ERDC/GSL TR-05-6.
[18] Buss AH , Sanchez P. "Simple Movement and Detection in Discrete Event Simulation." In *Proceedings of the 2005 Winter Simulation Conference, pp.992-1000. Orlando, FL, 4-7 December 2005* M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds. Available from http://www.informs-cs.org/wsc05papers/118.pdf
[19] Beeker E. "Potential error in the reuse of Nilsson's A Algorithm for path-finding in military simulations." *Journal of Defense Modeling and Simulations* 2004; 1(2): 91-97.
[20] Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y. "Pellet: A practical OWL-DL reasoner." *Journal of Web Semantics* 2007; 5(2): 51-52.
[21] Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", 19 November 2003. Available from http://www.daml.org/2003/11/swrl/
[22] Alberts DS, Hayes RE. *Understanding Command and Control*. Department of Defense Command and Control Research Program. 2006. Available from http://www.dodccrp.org/files/Alberts_UC2.pdf.
[23] Kleiner M, Carey S, Roberts J, Altenau M, Yu H, Gates BQ, Nelson E. "Geospatial Battle Management Language: Bridging GIS, C2 and Simulations." Paper #2107 ESRI Users' Conference. 2007. Available from http://gis.esri.com/library/userconf/proc07/papers/abstracts/a2107.html
[24] IEEE Standard Upper Ontology Working Group. "Suggested Merged Upper Ontology." Available from http://suo.ieee.org/SUO/SUMO/index.html

Acknowledgements

This research was sponsored by the U. S. Army Engineer Research and Development Center, Geotechnical and Structures Laboratory (ERDC-GSL). Permission was granted by the Director, Geotechnical and Structures Laboratory to publish this information. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Author Biographies

Joyce A. Nagle is a research engineer with the U.S. Army Engineer Research and Development Center, Cold Regions Research and Engineering Laboratory (ERDC-CRREL), in Hanover, NH. She is currently working in the area of the use of ontologies and knowledge-based technologies to support tactical maneuver. She is a graduate of the US Army Management Staff College and the Office of Personnel Management's Women's Executive Leadership Program. She received her B.S. in Botany and M.S.E. degrees in Water Resources and Environmental Engineering from the University of Michigan and her Ph.D. in Engineering Sciences from Dartmouth College.

Paul W. Richmond is a mechanical engineer at the U.S. Army Engineer Research and Development Center, Geotechnical and Structures Laboratory (ERDC-GSL), in Vicksburg, MS, where he develops ground vehicle mobility models for use in simulations, simulators and performance analysis models, specifically related to terrain interaction and off-road performance.

Curtis L. Blais is a research associate at the Naval Postgraduate School (NPS) with the Modeling, Virtual Environments, and Simulation (MOVES) Institute. His principal research areas include agent-based simulation of non-traditional warfare and application of Semantic Web technologies to improve interoperability of Command and Control Systems and Modeling and Simulation Systems. Mr. Blais has a B.S. and M.S. in Mathematics from the University of Notre Dame. He is currently a Ph.D. candidate at the MOVES Institute at NPS.

Niki C. Goerger is a research engineer with the U.S. Army Engineer Research and Development Center, Geotechnical and Structures Laboratory (ERDC-GSL), in Vicksburg, MS. Her expertise is in the areas of military modeling and simulation and in operational modeling and analysis. Dr. Goerger served as a research associate and the Defense Model and Simulation Office Visiting Professor (M&S) at United States Military Academy (USMA), West Point, NY from 2004-2008. She is currently the ERDC Liaison to the Office of the Assistant Secretary of Army (Acquisitions, Logistics, and Technology) in Arlington, VA.

Robert H. Kewley is currently the Director of the Operations Research Center at the United States Military Academy (USMA) Department of Systems Engineering. He was commissioned in 1988 from the USMA as an armor officer. His armor assignments include Task Force 1-32 Armor in the 1st Cavalry Division and Task Force 1-70 Armor in the 194th Separate Armored Brigade. He served as a tank company executive officer during Operations Desert Shield and Desert Storm. He has commanded both a tank company and a cavalry scout training troop. His analysis experience includes a teaching assignment at West Point and a tour at the Center for Army Analysis. LTC Kewley's research interests focus on command and control systems. He has a M.S. in Industrial Engineering and a Ph.D. in Decision Science and Engineering Systems, both from Rensselaer Polytechnic Institute.

Robin K. Burk is an assistant professor in the Electrical Engineering and Computer Science Department at the United States Military Academy (USMA), West Point, NY. Prior to joining the faculty in 2001, she spent over 27 years in software engineering, engineering management, program management and as a corporate executive in high-technology firms serving both defense and commercial markets. She is the author and technical editor of a number of well-selling books for software and networking practitioners, including TCP/IP Blueprints and several editions of UNIX Unleashed. Robin is a Ph.D. candidate at the State University of New York at Albany in Information Science with a research focus on knowledge representation and artificial intelligence techniques for autonomous mobile robotic equipment.