Adapting a Commercial Simulation Framework to the Needs of Information Fusion Research

Pontus Hörling, Vahid Mojtahed, Per Svensson
Swedish Defence Research Agency
Stockholm, Sweden
{hoerling, vahid, pers}@foi.se

Brad Spearing
Ternion, Inc.
Huntsville, AL, U.S.A.
spearing@ternion.com

Abstract - *The paper develops and argues usage requirements emanating from Information Fusion research on simulation frameworks. Based on these requirements, a next-generation simulation framework is currently being designed by one framework vendor.*

Keywords: Simulation frameworks, information fusion, geographical information management, requirements specification, evolutionary development, command and control

1 Introduction

FOI, the Swedish Defence Research Agency, has developed and experimented with computerized simulation models since the late fifties. A decision was recently made to design and build a simulation laboratory for Information Fusion (IF) research at FOI. The laboratory should allow IF algorithms and ideas to be tested in a software plug-in type of open simulation development environment, or simulation framework.

The Information Fusion research at FOI [1, 2, 3, 4, 5] is aimed at creating, evaluating, and demonstrating new information fusion techniques intended for applications in tactical intelligence analysis of military operations in various environments and scenarios. The focus of the IF simulation laboratory is on software and algorithm design, and in particular, methodology evaluation and demonstration, rather than, say, on powerful visualization capabilities or high execution speeds. Two early examples are algorithms for force recognition using transportation [2] and organizational [4] doctrines, respectively, see Figures 2 and 3.

Since few future requirements on such a laboratory can be foreseen today, the IF simulation laboratory needs to be built around a development environment concept as flexible as possible, based on an open architecture, with a simulation kernel scheduling and orchestrating the different objects and functions in an evolving scenario. Development tools need to allow for extensive re-use, including gradual extension and rewrite, of both software components and simulation scenarios.

2 Requirements on a simulation development environment for IF research

The purpose of the IF simulation laboratory is to support the definition, creation, and execution of scenarios where information fusion concepts, methods, and algorithms are tested and evaluated. Here, information fusion denotes data fusion processes which exploit a dynamic target situation picture produced by multisensor fusion, by combining its information with any available and relevant *a priori information*, in order to refine and interpret the battlespace situation picture. This intelligence interpretation process aims at delivering a comprehensive picture of the opponent´s options and, based on an evaluation of these options, suggest his likely intentions.

Focusing henceforth on ground-based scenarios, the a priori information will typically consist of geographical data, other important information about the tactical environment such as the location of civilian populations and protected buildings, intelligence about the opponent´s tactics, equipment and organization, known facts about the opponent´s logistics situation, as well as other kinds of tactical knowledge [1]. Detailed geographical a priori information will be needed in particular to support calculation of sensor-to-target detection, classification, and tracking parameters, spatial reasoning about target behaviour based on tactical doctrine, and real-time terrain-dependent management of own collection resources.

Information will be transmitted from actors (e.g., sensors) to other actors (e.g., Command and Control, *C2*, sites). At the C2 sites the information will be fused and interpreted. Finally, the interpretation will be used to develop and issue control messages intended to improve sensor utilization in relation to perceived surveillance objectives.

Information fusion also comprises techniques for proactive or reactive planning and management of own information collection resources such as sensors and sensor platforms, in order to make best use of these resources in relation to identified intelligence requirements. Resource management and high-level sensor management is one focus of the IF research at FOI [5].

A simulation for testing information fusion algorithms is thus likely to include participating actors such as mobile platforms or C2 sites, models of their behaviour including control of their motion and their sensors, inter-actor communication, and sometimes, reactive behaviour of actors belonging to the opposing force. Models need to be adapted to the scale of command and control IF scenarios, typically involving hundreds or even thousands of actors in a mainly ground-based environment encompassing, say, between 10 and 1000 km^2, cf. Figures 2 and 3. To keep complexity, development time, and cost within acceptable limits, the development of simulations for IF research need to be based on extensive reuse of both software and scenarios and should adopt an evolutionary development approach [6].

2.1 Current simulation environment

As a testbed for this research, we have previously used simulation software originally designed during the first half of the 1990's, in cooperation between FOI and several Swedish defence industries. This software, called *FbSim™* [7], was later further developed by one of the industry partners and has subsequently been used mainly for various defence materiel evaluation purposes.

FbSim supported well the first stages of our evolutionary demonstrator research and development project. From an IF research point of view, its main strengths were its integrated terrain data manager, offering a both attractive and powerful user interface for scenario presentation, and the fact that the software is delivered with a set of ready-made scenarios which matched our objectives well during early stages of our research. The terrain data manager provided Swedish standard terrain map data at a suitable resolution. This module also provided terrain attribute classification, integrated terrain elevation grid data, and powerful algorithms for evaluating shortest/best paths on and off road networks, and for performing line-of-sight calculations between two arbitrary points in the terrain.

However, we eventually found FbSim hard to adapt to our growing requirements. To create new scenarios with, or even without, the introduction of new kinds of actors proved difficult and time-consuming, the system's architecture and scenario notation turned out not to be flexible enough for our needs, and at the time, adequate documentation and training products were unavailable. Finally, we needed a system which could support the creation of terrain databases from many different areas in the world, as well as manage simulation of scenarios involving a large number of objects and units. Neither of these features were offered by FbSim.

2.2 Simulation Frameworks

A *simulation framework* is a term denoting a class of commercial products which provide a generic development environment, or toolset, for modelling and simulation. A research organization like FOI needs to develop simulations for many different purposes under an evolving set of requirements. Thus, the need for a development methodology based on reusable components and techniques has long been evident. A simulation framework may offer a significant productivity increase in designing and building simulations. This is due to faster development of scenarios and models, extensive reuse of scenarios, models, and program modules, and the establishment of a basis for cost-sharing and cooperation with other organizations working with similar tasks.

Obviously, these advantages come at a price:

- the license costs for the framework can become quite high, and may cover development costs of software irrelevant to a user's own organization
- the increased productivity can frequently only be exploited over the longer run, and in the form of higher-quality research products, rather than as short-term cost savings, since it requires exploitation of a substantial software and knowledge re-use base
- there is a need for in-house expertise in each of the several user roles a simulation framework supports, and sustainable framework-based system development activities will require the availability of a group of specialists internal or external to the research or development organization. The cost of sustaining this competence typically needs to be shared among several competence groups and projects.

Some years ago, two research groups at FOI more or less independently acquired a simulation framework, called *Flames™* [1] [8], now being used by a growing number of projects within the agency. The alternatives to making such generic investments are either to abstain

[1] Flames is a trademark of Ternion Corporation

completely from using complex scenario-based simulations, to develop them in-house using less specialized tools (such as mathematical program libraries, or in simple cases generic development and analysis environments such as MATLAB), or to buy them piecemeal from simulation software developers. All these approaches have been used repeatedly by FOI and certainly will continue to do so in the future. The simulation framework approach, however, has by now proved its ability to offer increased productivity in a number of cases.

2.2.1 What does a simulation framework provide?

To expand on the terminology introduced above, a simulation framework is an open and extensible simulation system which can be used to simulate the behavior of a broad range of event-based systems, using a combination of software supplied with the framework and software written by the user in a standard programming language.

It is a well-documented commercial product which is subject to continuous development, and whose maintenance cost is largely carried by the framework provider, thus shared among software licence holders.

The most important characteristic of a simulation framework is its underlying architecture, allowing the applications supplied with the framework to be customized and new framework-based applications to be developed quickly and inexpensively.

A useful simulation framework must be a fully functional, end-to-end simulation system complete with off-the-shelf applications that support definition, execution, post-processing and visualization of scenarios.

This architecture typically consists of:

(1) A kernel, which provides

- an infrastructure that includes common services and facilities for models, e.g. object management, time management, memory management, execution control, interface for client applications, data base management, network communication, etc. These services provide advanced features to simulations and reduce the size of models,

- an architecture enabling the development of application-specific models which are cleanly separated from the services of the kernel,

- a standard framework for model development and interoperability.

(2) A set of standard applications which support scenario definition, execution, post-processing, and visualization, as well as an environment and associated tools that support and simplify the development of user-defined applications.

(3) In order to satisfy these requirements, a simulation framework should:

- provide a model library containing environment, equipment and cognitive models, making the framework ready to use on delivery,

- contain multi-level documentation, which guides both the end-user and the developer through the framework,

- possess a stable trademark that vouches for future availability of maintenance and development,

- be a multi-purpose system, i.e., a single simulation framework should be capable of supporting many different kinds of simulation applications.

This list could easily be extended by additional requirements like level of fidelity, resolution, reusability, etc.

As stated in Section 2 above, a geographical information management subsystem needs to be included in the simulation environment of the IF laboratory. The need for such *geoinformatics (GI) functionality*, although strongly felt in our application area, is however poorly satisfied by simulation frameworks currently known to us.

2.2.2 Requirements for GI functionality

FOI's requirements for GI functionality when using a simulation framework in IF research are summarized below. The complete list of requirements were included in a 10-page Request for Offer, sent by FOI in April, 2001, to the three simulation framework vendors known to us at the time.

Some of the GI functionality is needed only in a preprocessing phase, some in the simulation framework (and the final simulator) itself. We distinguish between *geographical database preparation*, which is presumably best carried out by specialists for each geographical area of interest in a preprocessing step, and *geographical database usage*, which is going on within the simulator whenever a scenario is being set up, executed, or assessed.

To create and populate different kinds of terrain databases, a tool for preprocessing of commercially available GI data will be required. These preprocessing functions should be separated from the simulation framework and are probably best performed by an external specialist organization.

The GI subsystem should have general facilities for coordinate transformation between commonly used geodetic datums and map projections.

GI functions should be reachable via a graphic interface using the same principles by which actors are modified during scenario generation.

In order to be able to develop new functions and algorithms and classes using or operating on the terrain model there is a need for low-level access to the terrain database.

Providing a suitably detailed map background to the output from a simulation will often radically enhance information presentation. In order to reduce the amount of tedious programming tasks, FOI needs a high-level but flexible tool for map presentation integrated in the simulation framework.

In addition to 2D presentation, it would be useful to be able to visualize a dynamic 3D view of the simulation, including of course its geographical environment.

2.2.3 Vendors' response

None of the three vendors who received the RFO provided a response which could form a basis for product procurement by FOI. However, one of them, *Ternion Corporation*, responded with a proposal to investigate in a two-phased effort if and how the capabilities of their previously mentioned Flames framework could be augmented to satisfy the GI-related requirements of the IF simulation laboratory. This proposal was accepted by FOI.

3 Third party product evaluation

The first phase of the effort was begun by defining a list of specific criteria suitable for the evaluation of commercial *geographic information system* (GIS) packages. A description of these criteria is presented in Table1, below.

3.1 GIS Product Evaluation

Numerous commercial GIS packages were evaluated for their ability to satisfy the criteria listed above. Over one hundred GIS products were considered in the evaluation. The GIS products to be evaluated were identified from Internet searches and from referrals from other GIS vendors. Each GIS vendor was contacted and provided with a questionnaire. Nineteen companies responded. In most cases, the vendors that responded were contacted again to clarify their answers or to get additional information.

3.1.1 GIS Product Evaluation Results

The market search and product evaluations did not identify a single GIS product that could meet the evaluation criteria. It was also clear that deficiencies in the API could not be corrected by custom software development due to the lack of underlying support for attribute, road, or 3D data. In addition, the need to access a disk resident database for each query is unacceptable from a performance standpoint.

Based on this market search and evaluation, the study came to the conclusion that existing GIS products are not designed to support simulation applications. They serve a different market with a very different set of requirements.

3.2 VisSim Product Evaluation

Ternion then expanded their effort to include the evaluation of commercial *visual simulation (VISSIM)* packages. Twenty-five Visual Simulation (VISSIM) product vendors were considered in this evaluation. A vendor list was compiled by searching for support of the OpenFlight format among the products evaluated in the *Survey of Terrain Visualization Software* [9]. Responses to a brief questionnaire were solicited. The area of principal concern was the ability to query analytical information from a non-graphical application (such as the standard application for scenario execution in Flames, called *Fire*). Twenty companies responded to the questionnaire. Most of the companies that responded were eliminated from consideration after evaluation of their response. Products from six companies were selected for further evaluation.

As was found to be the case with GIS products, no VISSIM product could meet the evaluation criteria. The study concluded that VISSIM products are not designed to support analytical simulations. However, the underlying databases used by VISSIM products might be useful for analytical simulation purposes.

4 Enhanced terrain management design analysis

Based on the phase 1 findings, Ternion continued by defining and evaluating an approach that would satisfy FOI's requirements. This approach was focused on developing new software for Flames using data similar to that used by VISSIM products. Ternion eventually proposed to meet FOI's requirements by developing four optional enhancements to Flames, as detailed below.

4.1 Terrain Database Generation

Several terrain database generation tools were further evaluated. Based on this evaluation, Ternion selected the *Terra Vista*™,[2] family of products from Terrex as the best tool for building terrain databases for use in Flames. This selection was based primarily on Terra Vista's ability to write correlated terrain databases in a variety of formats. Using Terra Vista, a terrain database can be developed by importing terrain and feature data from several different sources. Once developed, the database can be written in several formats, all of which are correlated with each other. Furthermore, Terra Vista can write a database as a set of *ARC Shape files* [10], Figure 1.

Some of the Shape files contain polyhedral TIN information describing the "terrain skin", including the surface of roads and waterways. Other Shape files contain vector data that describe such things as the centerline of roadways and the location of forested areas. Shape file data is usable for mathematical calculations and can include attribute information. Terra Vista can specify feature and attribute information using codes from the *Feature and Attribute Coding Catalogue* (FACC) [11]. These codes will be used for feature and attribute information within Flames. The polyhedral TIN data representing the terrain skin and the vector data describing features are correlated by Terra Vista. This means, for example, that the vectors representing the centerline of a road will lie on the surface of the polygons that represent the road.

In addition, Terra Vista can store a feature ID with each polygon that represents a surface of a feature. This provides a cross-reference between the polygon data and the feature data.

4.2 Enhanced Terrain Option

A new terrain model, called ETO (*Enhanced Terrain Option*), will import the ARC Shape files containing polyhedral TIN data as generated by Terra Vista.

ETO will support the existing Flames FTerrain functions (such as FTerrainIsMasked, FTerrainCheckIntersection, FTerrainGetElev, and FTerrainComputeECRToBody). In addition, FTerrain functions will be added to provide attribute information associated with a given surface polygon (such as soil type, road, water, etc.) using FACC coding, and to provide the ID of the feature associated with a given polygon (if any).

One of the attributes that ETO will provide for a given polygon will be the ID of the associated feature, if one exists. For example, if a ground vehicle is positioned on a road, the ID of the road feature at the vehicle's position can be accessed via a function call. ETO will include a ground vehicle example model in source code, supporting vehicle movement on a road, given its road ID.

4.3 Enhanced Feature Option

A new set of feature models, called EFO (*Enhanced Feature Option*), will support importing ARC Shape files containing vector data as generated by Terra Vista. Imported data will be used to create instances of new Flames feature classes that will be developed as a part of EFO. These features, when loaded into Flames, can be queried by models that need to consider features in their calculations. The feature classes that will be developed are road, body of water, bridge, forested area, building, and individual tree/bush. For objects of these classes, the position, type, and other relevant data will be accessible.

4.4 Enhanced Display Options

Two new subsystems will add support for display-ing 2D and 3D image data in *Forge* and *Flash*, the standard scenario development and visualization applications, respectively, of Flames. The 2D option uses 2D image files in the format generated by Terra Vista. The 3D option will use OpenFlight data and will be rendered using the visual simulation toolkit *VTree*™,[3] .

5 Conclusions

FOI has evaluated Ternion's proposals and drawn the conclusion that, although they do not satisfy all of FOI's stated requirements for geographical data management functionality, critically important such functionality would be provided by the proposed enhancements. Furthermore, these enhancements create a platform for geoinformatics functionality development by framework users. Perhaps more important strategically, however, is the fact that the proposed new geodata management architecture would form a solid foundation for future vendor-provided generic functionality in the Flames simulation framework.

By proposing to base their geodata development strategy on state-of-the-art third party tools and de-facto industry standards, Ternion's proposal may contribute to the closing of the methodology gap that has existed between geodata management and simulation framework software manufacturers.

[2] Terra Vista is a trademark of Terrain Experts, Inc. (TERREX)

[3] Vtree is a trademark of CG^2, Inc.

Criterion	Description
1. API for Analysis	The chosen solution must provide an API that supports all the analytical operations to be performed using the spatial data.
2. C-callable API	The product's API must be directly callable from a program written in the C programming language. (e.g. an API implemented in C or C++ is almost always acceptable. An API implemented in Java would not be acceptable.)
3. Attributes of Objects	The chosen solution must store more than geometry of the objects in the database. It must also store attribute data (such as type of soil or material) about each object to support the required analysis.
4. 3D Representation of Objects	The chosen solution must describe all the objects in the database in three dimensions. These descriptions must be consistent and complete (no holes/gaps in the data and no redundant/ overlapping data).
5. Multi-platform Support	The chosen solution must work on all platforms supported by Flames (PC, SGI, and Sun) and have the potential to be ported to other platforms in the future.
6. Performance	The chosen solution must support large spatial databases and large scenarios with potentially thousands of queries per second to the database.
7. 3D Visualization	The chosen solution must support real-time 3D visualization. Not every Flames-based application will require this visualization, but some will. If the solution does not support visualization, no application will ever be able to use it. This is the case with the current Flames DTED option. Real-time visualization is not possible using a DTED database directly. The biggest issue in supporting real-time 3D visualization is spatial database correlation. The only way to accurately render a scenario in 3D is to ensure that the spatial database used internally in Fire, the execution environment of Flames, is correlated with the spatial database used by the 3D visualization tool.
8. Compatibility with Other Databases	The chosen solution should support a database format for which converters are readily available. There are many different types of databases and database standards in existence, and it is unrealistic to attempt to support all types. A database format with converters can convert other types of database formats to the chosen format.

Table1. Evaluation criteria for geoinformatics products.

References

[1] Per Svensson, *Information Fusion in the Future Swedish RMA Defence*, Swedish Journal of Military Technology, No. 3, pp. 22-28, 2000.

[2] Johan Björnfot and P. Svensson, *Modeling the Column Recognition Problem in Tactical Information Fusion*, Proc. Third Int. Conf. on Information Fusion (Fusion 2000), Paris, July 10-13, 2000.

[3] Mats Bengtsson and J. Schubert, *Dempster-Shafer clustering using Potts spin mean field theory*, Soft Computing Vol 5, No. 3, pp. 215-228, 2001.

[4] John Cantwell, J. Schubert, and J. Walter, *Conflict-based Force Aggregation*, In Proceedings of the Sixth International Command and Control Research and Technology Symposium, Annapolis, USA, 19-21 June 2001.

[5] Ning Xiong and P. Svensson, *Multi-Sensor Management for Information Fusion - Issues and Approaches*, Information Fusion, in press.

[6] Jesper Fredriksson, P. Svensson, and T. Risch, *Mediator-based Evolutionary Design and Development of Image Meta-Analysis Environments*, J. Intell. Inf. Syst., Vol. 17(2/3), December 2001, pp. 301-322.

[7] *FbSim User Manual*, Sjöland och Thyselius Datakonsulter AB, Stockholm, Sweden, Sept. 2001 (in Swedish).

[8] //www.ternion.com/product.htm

[9] //www.tec.army.mil/TD/twvd/survey/survey.txt, U.S. Army Topographic Engineering Center, 27 June, 2001.

[10] *ESRI Shapefile Technical Description*, ESRI White Paper, July 1998.

[11] //www.digest.org/html/gp40.htm, DIGEST Part 4, Edition 2.0, June 1997.

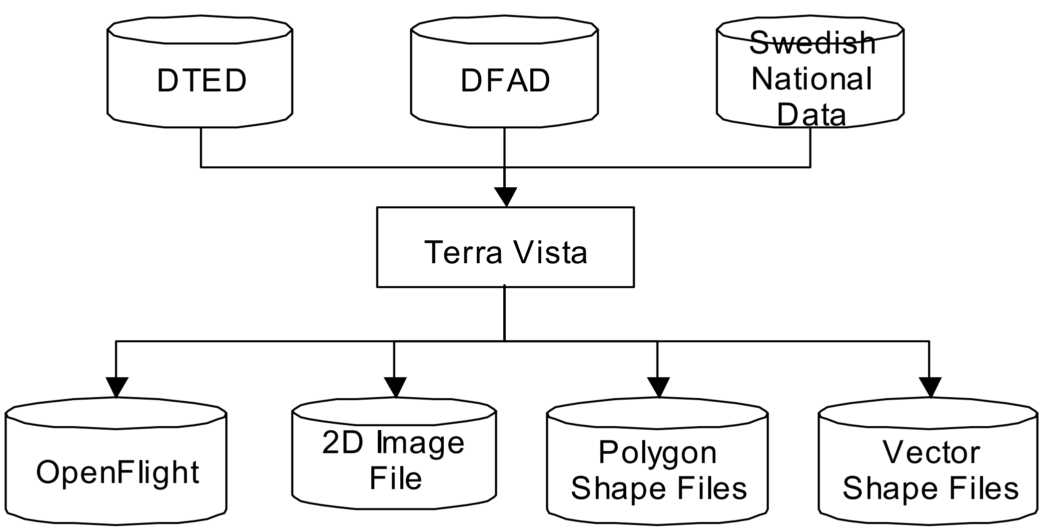

Figure 1. Terrain database generation data flow.

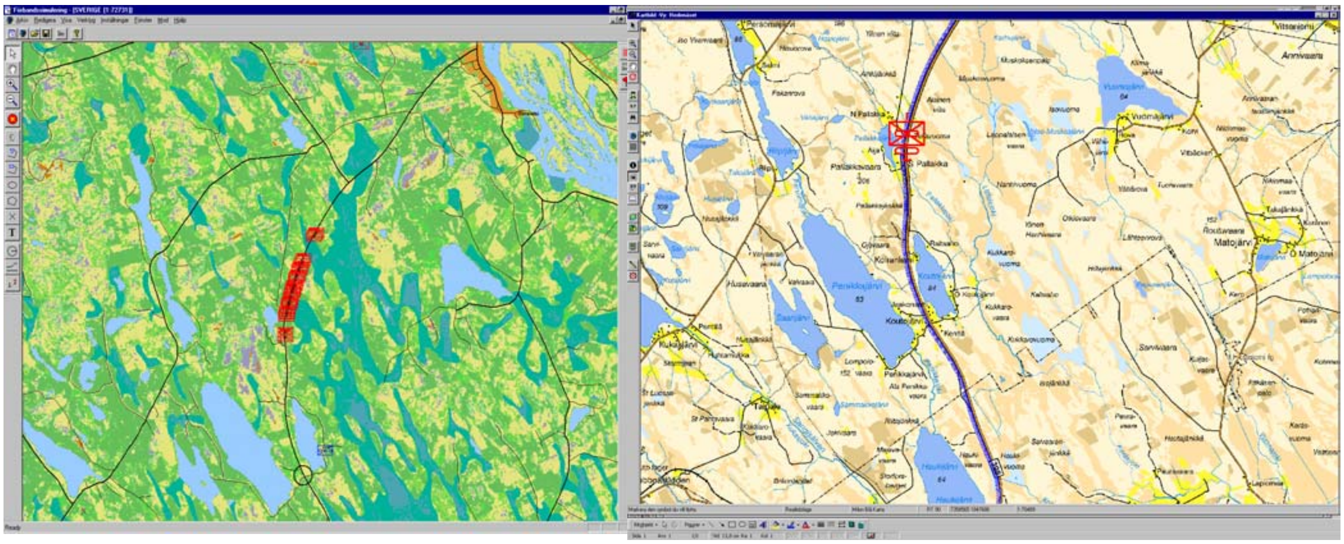

Figure 2. Simulating a moving vehicle column (left) and displaying result of HMM-based company-level force aggregation (right), cf. [2].

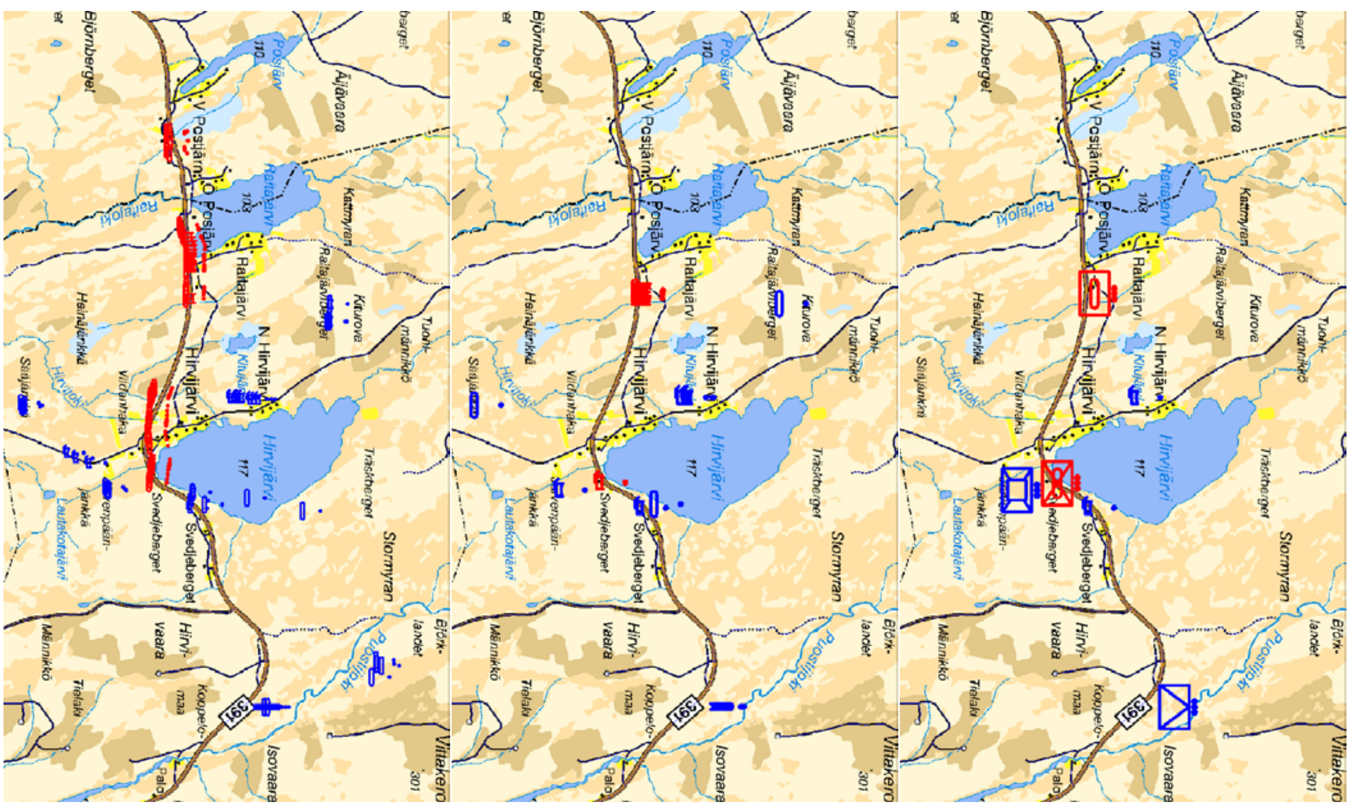

Figure 3. Simulating intel reports from two opposing forces (left), displaying result of vehicle recognition based on Dempster-Shafer clustering (center), and result of platoon-level force aggregation (right) , cf. [4].