The IFD03 Information Fusion Demonstrator

Simon Ahlberg, Pontus Hörling, Karsten Jöred, Christian Mårtenson, Göran Neider, Johan Schubert, Hedvig Sidenbladh, Pontus Svenson, Per Svensson,* Katarina Undén, and Johan Walter
Swedish Defence Research Agency, SE-172 90 Stockholm, Sweden
firstname.familyname@foi.se

Abstract - *The paper discusses a recently developed demonstrator system where new ideas in tactical information fusion may be tested and demonstrated. The main services of the demonstrator are discussed, and essential experience from the use and development of the system is shared.*

Keywords: Scenario simulation, demonstrator concept, force aggregation, group tracking, sensor allocation, sensor modelling, terrain modelling.

1 Introduction

In the *tactical information fusion* process, data from various sources are used: sensor data, as well as data on probable behavioral patterns of the opponent, cultural and geographic characteristics of the area of operations, etc. Such data have previously been fused to an operational picture through time-consuming manual analyses and discussions. As the availability of sensor data explodes as a result of technological advances, this manual fusion process becomes a serious bottleneck.

Automating the information fusion process is still largely a research issue. In particular, it is not yet clear how the basic methodology of computerized information fusion should be structured. Clearly, this structuring is not a purely technical task, but an issue which must eventually closely involve users of information fusion systems.

In the Swedish defence research and development (R&D) environment, at least, there are few opportunities to achieve the required user involvement until some credible information fusion demonstration platforms have been introduced to prospective users. Such platforms need to be based on scenario simulation, the only known methodology likely to offer the required versatility, dynamics, traceability, and repeatability of situations to be analyzed and techniques to be applied. Thus, simulation-based systems allowing prospective users first to learn about, later to try out and put strain on proposed information fusion methods and their user interfaces will be a prerequisite for the evolution and gradual user acceptance of these emerging methodologies. No less important are the twin requirements of being able to apply a sequence of fusion methods to various analysis problems, and then objectively evaluate their combined effectiveness and performance.

The paper presents early experiences and conclusions from the development of such a system that integrates research results in the areas of force aggregation, ground tracking, and sensor resource management within a state-of-the-art scenario simulation environment.

2 Conceptual overview

Largely according to the description given in [1], the FOI project *Information fusion in the command and control system of the future network-based defence* recently completed the development of a reusable [2] information fusion demonstrator system, the Infofusion demonstrator 03 (IFD03). In IFD03, level 2 information fusion is treated as the interpretation of a flow of observations, realized as a scenario-based simulation of a physical process in space and time. This simulation describes the stochastic interaction between an observation system, a complex target system, in this case a hierarchically organized enemy unit, and a complex environment.

Information is transmitted from simulated sensors to a simulated Command and Control, C2, site. At the C2 site information is fused and interpreted, using automatic information fusion processes. Some of these interpretations are then used by the C2 site as basis for issuing control messages intended to improve sensor utilization in relation to a predefined surveillance objective. A key component of the demonstrator is the visualizer, which provides a movie-like, interactively controllable multi-screen playback display of a set of parallel views of the prerecorded simulation.

The IFD03 system was used to perform a demonstration in mid-December 2003, based on a simple battalion-level ground force attack scenario. An outline of this scenario was given in [1]. Although several details were later changed, this outline provides most of the background information needed in the following.

The development methodology that was partly reused, partly developed by our project in order to facilitate the construction of the demonstrator proved to be highly cost-effective although far from problem-free (see Section 5).

*Corresponding author.

The demonstrator implementation is based on three large development environments, the *problem solving environment* (for an in-depth study of this concept, see [3]) MATLAB™ [4], the simulation framework FLAMES™ [5], and the terrain modelling system TerraVista Pro Builder™ [6]. In the project, FLAMES and MATLAB were tightly integrated, and FLAMES' new handling of advanced terrain models, generated by TerraVista, was specified and at least partly financed. Finally, the FLAMES software for visualization of simulation results using the new terrain modelling feature was restructured and both functionally and computationally substantially improved.

2.1 Demonstrator objectives

The demonstrator is not a design and certainly not a prototype of a deployable system. To create a prototype, a significant additional R&D effort would be required. Our primary purpose has been, instead, to investigate how IF methods can be combined into a system and work together in the context of that system. We also wanted to create and exercise an effective mechanism whereby information fusion concepts can be communicated to our customers and other interested parties. Finally, we wanted to create a basis for discussions with customers and prospective users about how research in the IF area should be prioritized.

2.2 Use cases

The major use cases [7] we had in mind when creating the system were:
(1) performing a demonstration addressing a possibly "infofusion-naive" audience. This is communication, not research, but could be developed into a methodology to present, visualize, and later analyze in detail properties of new components and subsystems,
(2) performing studies and experiments with sensor models, terrain and other environment models, fusion methods, doctrine models, scenario assumptions, etc., in various combinations, to test different hypotheses about opportunities and limitations related to Network-Based Defence (NBD) and information fusion (IF),
(3) developing and testing methodology and models for IF, i.e., specification, development, and test of new concrete methods and fusion concepts. The size and complexity of a demonstrator can be a severe drawback here, at least early in the research and development process, which leads to the question: how could detail studies in a separate test environment best be combined with system tests involving the demonstrator platform? The demonstrator provides at least a partial answer to that question.

Potential advantages from using such a simulation-based R&D process include:

- shorter turn-around time and lower cost for the modelling activity; this can be exploited to create a better dialog with prospective users/customers,
- higher quality through such a dialog and improved opportunities to pre-test a proposed system in synthetic but increasingly realistic and perhaps ultimately dangerous situations,
- improved basis for the estimation of total system construction costs.

2.3 Scenario display

During the demonstration, three adjacent overhead projection screens showed:
(1) reports and ground truth data displayed on a synthetic map background,
(2) results from the different information fusion methods displayed on map backgrounds, and
(3) dynamic plots of various statistics and other information about the current state of the fusion processes.

At the beginning of the scenario only a few reports had arrived. These were indicated on the first screen (Fig. 1) and then appeared as clustered objects on the second screen. This was the first chain of fusion events shown during the demonstration. At the same time the process could be followed on the third screen where plots of the number of received reports and the estimated number of objects were displayed.

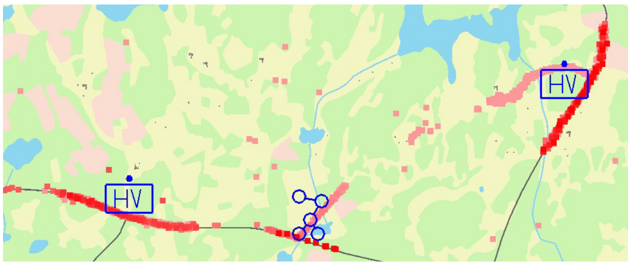

Fig. 1. Sensor reports view. Home guard patrols and a ground sensor network are symbolically shown.

As the scenario progressed, more surveillance resources were allocated and therefore many more reports were delivered. On the second screen, views showing clustered vehicles and clustered platoons were displayed. Here, vehicles and platoons had been automatically classified into more or less specific categories, when possible into specific types. The categories or types were displayed using standardized army symbols (Fig. 2). A comparison between the two levels showed a good correspondence.

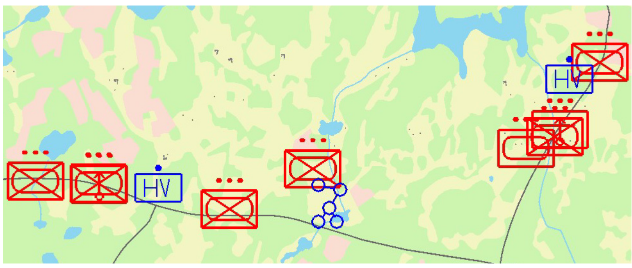

Fig. 2. Platoon view. There are three different types of platoon in the picture.

Controlled by the operator, the demonstrator display could switch momentarily between different aggregation levels such as reports, vehicles, platoons and companies. This showed how the different information fusion methods worked at different levels. In a more general sense, this flexibility of the demonstrator indicated how a future system also could and should be flexible.

To indicate to the audience how the fusion methods were performing, various results could be compared with ground truth as it was displayed during the demonstration. Having access to all scenario information, the ground truth view showed the location of all vehicles and all sensors in the displayed area over time.

During the scenario, the correspondence between different aggregation levels and between ground truth and the aggregation levels varied. The scenario display was paused by the operator at several occasions, to show how correspondences varied in different situations and between different levels of the scenario. By zooming in on any desired display area, detailed situations could be visualized and discussed (Fig. 3).

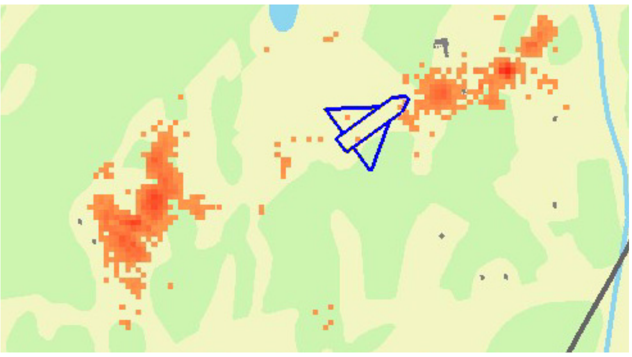

Fig. 3. Particle histogram view. Shades of red are used to represent variation in probability density.

Some of the information fusion methods used in the scenario required a specific context to best show their capability. One example is the particle filter tracking which is more effective in terrain than on roads and was therefore also displayed during a terrain passage. Results of this method were displayed as histograms overlaid on the terrain map, using red squares with increasing intensity for increasing probability density of objects (Fig. 3).

3 Fusion node methods

The analysis module has three main tasks and uses four different methods. The tasks are force aggregation, group tracking and sensor allocation. They are performed using Dempster-Shafer clustering and Dempster-Shafer template matching for force aggregation, probability hypothesis density (PHD) particle filtering for group tracking, and random set simulation for sensor allocation.

3.1 Force aggregation

In force aggregation, intelligence reports with given position, time, and type information are used. Initially, all pairs of intelligence reports are evaluated. Then the task is to find a set of facts which speak against the proposition that two reports are referring to the same object: different types of vehicle? distance too long or too short? different directions? wrong relative positions? etc. This yields a potential conflict between each pair of intelligence reports. From this, a conflict matrix is constructed and supplied to the clustering algorithm. The Dempster-Shafer clustering algorithm [8−10] is used to partition the set of reports into subsets, each subset corresponding to one object. Then, objects are classified by fusing all available intelligence using Dempster's rule. This method continues, first from sensor reports to vehicles, then from vehicles to platoons, then from platoons to companies, etc., upwards level by level in the organization hierarchy.

When clustering reports to vehicles, the subproblem of determining the optimal number of clusters, n_0, was solved heuristically using the following empirical observation: Let $Mc_{opt}(n)$ be the approximate optimum value of the metaconflict function [9] found by the clustering algorithm, given n clusters. Then $\log(Mc_{opt}(n))$ tends to be approximately linear both above and below $n = n_0$, while changing slope near n_0.

At the vehicle-to-platoon level (Fig. 2), vehicles are clustered and groups of vehicles are classified using Dempster-Shafer matching against templates [11]. At all levels in clustering and template matching we use the full descriptive power of Dempster-Shafer theory, carrying several alternative hypotheses represented by a belief function that is the result of fusing all intelligence pertaining to the cluster. Each alternative hypothesis is matched and evaluated against all templates (Fig. 4) and a weighted average is calculated for each potential template. The best template is selected if its matching value is above a given threshold.

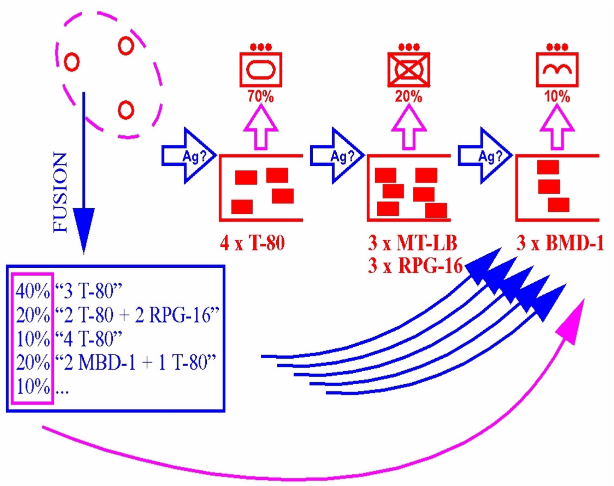

Fig. 4. Scheme for aggregation of vehicles to platoons. Each hypothesis is evaluated against all templates, to give an overall fitness for each template.

The average measured computation time per completed reports-to-vehicle aggregation in the demonstration scenario was initially about 30 min. on a 2.6 GHz single-processor PC. A few hundred reports were clustered each time, usually yielding an optimum number of 10−30 clusters (vehicles). Computation time for clustering, the time-dominant operation in this aggregation process, is quadratic in the number of clustered items. In this problem ten clustering trials were made for each of 21 different cluster sizes in each force aggregation.

A systematic, profiler-guided code optimization effort involving, i.a., hand-translation of the most time-consuming loops into C, has by now improved total clustering execution speed by about a factor of ten. In addition, the use of parallel processing could reduce elapsed time for this problem by a further factor of 200, to about 1 second.

Result quality in terms of classification error statistics remains to be measured and analyzed. Subjectively, however, classification accuracy up to platoon level was sufficient to yield credible results when compared to ground truth data.

3.2 Group tracking

Particle filtering [12] has attracted a great deal of interest during the last ten years. The method is suitable in situations where the motion or observation model is non-linear. Since the ground target tracking problem displays a number of non-linear properties, we use particle filtering in our system instead of, e.g., Kalman filtering.

We use a PHD particle filter [13] for simultaneous tracking of an unknown number of targets in terrain. The input to the filter are intelligence reports with given positions and velocities. Furthermore, information about the type of terrain can be obtained for every position. In each time step, both the number of vehicles and their positions and velocities are estimated in an iterative Bayesian manner, based on the reports, the terrain and on estimates of the observation and motion noise. The PHD particle filter is an implementation of the PHD filter introduced by Mahler and Zajic [14].

The method may also take aggregated results from the force aggregation as observations in order to track multiple units. Thus, one PHD particle filter is used on each hierarchical level.

The computation time for particle filters is in general high in comparison to Kalman-type trackers. However, this does not pose a great problem in a scenario such as ours. In the demonstrator, computation time for each time step of the tracking was about five seconds on a 2.6 GHz single-processor PC, while the real time length of each tracking step was ten seconds.

Although results seem promising from the few experiments yet carried out, performing systematic studies of computational performance and result quality will be an important future task.

3.3 Sensor allocation

Starting from an assumption about which information is to be collected, when and where it should be collected, and when it is needed, available sensor resources are to be allocated [15]. For IFD03, we chose to implement a simple version of a sensor allocation method based on random sets [16], which is currently being developed at FOI. This method is based on simulating the opponent's future movements and choosing sensor control parameters to maximize a utility function.

The method used can be informally described as follows. Given a situation picture $\mathbf{x_0}$, we generate all possible future situation pictures $\mathbf{x}$ that are consistent with the positions in $\mathbf{x_0}$ at time 0. For each of these, we calculate the utility of each sensor control scheme $\mathbf{s}$. This utility is calculated by simulating observations of $\mathbf{x}$ using $\mathbf{s}$. The $\mathbf{s}$ whose average over all $\mathbf{x}$ is best is then chosen.

Several utility functions were implemented and tested. All currently tested utilities are based on simulating a tracking algorithm and comparing its output with the "ground truth" in $\mathbf{x}$. Several simplifying assumptions were made in the implemented algorithm. All movement in $\mathbf{x}$ was restricted to a road network that was automatically extracted from the terrain database. Also, the set of possible sensor control schemes was kept relatively small.

Work is ongoing to extend the simple sensor allocation method presented here, but no significant result quality or computational performance data have yet been produced.

4 Modelling issues

4.1 Modelling sensor observations

In [1], we discussed general properties which environment and sensor models should possess to enable sensor models to provide relevant information. The "environment", which in our case includes the terrain and the physical actors (vehicles) out there, should be modelled in neither more nor less detail than needed for the given level of scenario abstraction. The fusion methods we want to test with our simulator expect input such as observation times, target positions and velocities with their uncertainty estimates, as well as target types with uncertainty within a given target classification hierarchy.

It is not enough to build only a sensor model at a certain level of detail. The subject is threefold; a sensor detects energy (light, vibrations, radio signals, etc.), that has been emitted from somewhere, as well as been propagated and attenuated, reflected etc. on its way to the sensor. For detection to take place in a sensor, some kind of threshold level must be exceeded. The detected signal-to-noise ratio (SNR) from a target must be high enough for a signature-extracting mechanism to find the features it is trained to discern. The higher SNR, the more detailed estimates can be produced. The main issue is how the SNR should be realistically modelled for a chosen sensor in a typical terrain where a typical target is located.

4.1.1 Image sensors

For image sensors working in the UV-Visible-IR wavelength region the terrain cover, e.g., tree canopies, and the ground topographic structure are important. In our scenario we have a diversified terrain with many forested areas of different sizes, open croplands, roads, lakes and even littoral regions. For real observations with a UAV carrying a video working in the visible region, this type of terrain would give sparse bursts of reports when targets happen to pass by sufficiently transparent areas of the tree canopy, when they pass by open areas, and when targets are travelling on roads. Other routes through the terrain provide better camouflage. It is desirable to reproduce this behavior in order to study how the non-smooth flow of reports will affect the performance of fusion methods.

Image sensor detection probability is based on the "Johnson criterion" [17]. It gives a relation between the amount of resolved pairs of light/dark bars in a bar pattern of the same size as the target minor extension projected towards the observer, and the probability of detection/classification/identification. Actually, this is an empirical relation based on how well a trained human can identify different objects. The amount of resolved bars is related to the contrast between light and dark regions in the image, also interpretable as SNR. For a sensor in the visible region, this is the level of reflected light contrast within and at the bounds of a potential target. This is dependent on target surface reflectance variations, and the strength and directivity of the ambient light. For an IR sensor in the thermal region this contrast instead follows the target and background temperature variations. Outside the target itself, a contrast-rich (clutter-rich) background makes detection more difficult.

The attenuation of light is modelled for an image sensor observing from a UAV. The attenuation factor is dependent on terrain cover type (forest/open land) and, in the forest case, on the angle between the line-of-sight (LOS) and the vertical direction. It is easier to see through the canopy in a close to vertical direction, since the LOS path through foliage is shorter, and small, fully open regions between trees will appear. Large attenuation reduces SNR.

4.1.2 Ground sensor networks

The ground sensor network model was implemented in the simplest possible manner. An integrated tracker was assumed to motivate the removal of terrain effects as well as the influence of the individual positions of the network nodes. This led to a statistically homogeneous detection quality inside the range of the network. The purpose of such a sensor is to contribute intelligence carrying high quality position and speed measurements, but poor classifications.

4.1.3 Human observers

The model of human observers (home guard patrols equipped with advanced measuring binoculars) is less detailed than the image sensor discussed above. This is mainly due to difficulties of modelling the complex fusion performed by the human brain. Thus, the model was built on a phenomenological basis. A basic relationship of detection quality proportional to distance was assumed and model parameters were then adjusted so as to produce reasonably realistic output.

Common to sensors and observers is the fact that their ability to discern the type of a detected target increases with SNR. Recognition type hierarchies relevant for the different sensors' energy classes were constructed. I.e., an image sensor can discern (with decreasing discernibility) <T80>/<tank>/<tracked vehicle with 6 track rollers>/<vehicle>, and a seismic sensor can discern <heavy tracked vehicle>/<vehicle>. Each entry in these hierarchies can be expanded into all vehicle types belonging to it, limited by the size of the database of candidate vehicle types. This allows comparisons based on evidential reasoning to be made between type information from different sensor types.

4.1.4 COMINT interceptors

Simulated radio messages are exchanged within platoons in the scenario. This exchange follows two patterns: "commander gives an order, subordinates answer one after another", and two-party dialogues. Orders are more frequent when platoons pass certain geographical regions in the scenario where it is likely that they must communicate because of commanders' change of route etc. These radio messages can be intercepted by own COMINT interceptors deployed in the terrain. The interceptors give quite coarse information about bearings to emitters. Bearing crossings are computed to get an indication of the position of an emitter. Information about position and communication pattern is transmitted to the fusion node, which tries to find out who is communicating with whom, in order to resolve the platoon structure. This resolution can be obtained by looking at the time pattern of the intercepted messages. A conflict matrix can be built from these, and evidential reasoning methods for handling both attracting and conflicting evidence [9, 10] can be applied to resolve possible communication links. Most of this functionality now exists in IFD03. However, final fused results remain to be obtained and evaluated.

4.2 Environment model

4.2.1 Terrain modelling

The terrain model used by the FLAMES system in IFD03 was created using TerraVista Pro Builder [6]. This is primarily a tool for creating terrain databases for visual simulation, but it also has options for creating vector files or raster images for, e.g., GIS applications. All output is correlated, i e, taken together, it constitutes a geometrically and topologically consistent terrain model. TerraVista's terrain generation process had to be extended using its built-in scripting engine, in order to make the output contain the desired feature attribute names. The terrain features were grouped into seven classes defined by FLAMES: bridge, building, canopy, land region, lake, river, and road. This is a rather coarse classification, although feature classes themselves can have some feature-specific attributes (Figs. 1, 5). The available attribution is sometimes insufficient in that one might want to extend the feature classes to describe, e.g., forest density and composition, average building height for urban areas, etc. The modelling itself is straightforward and uses

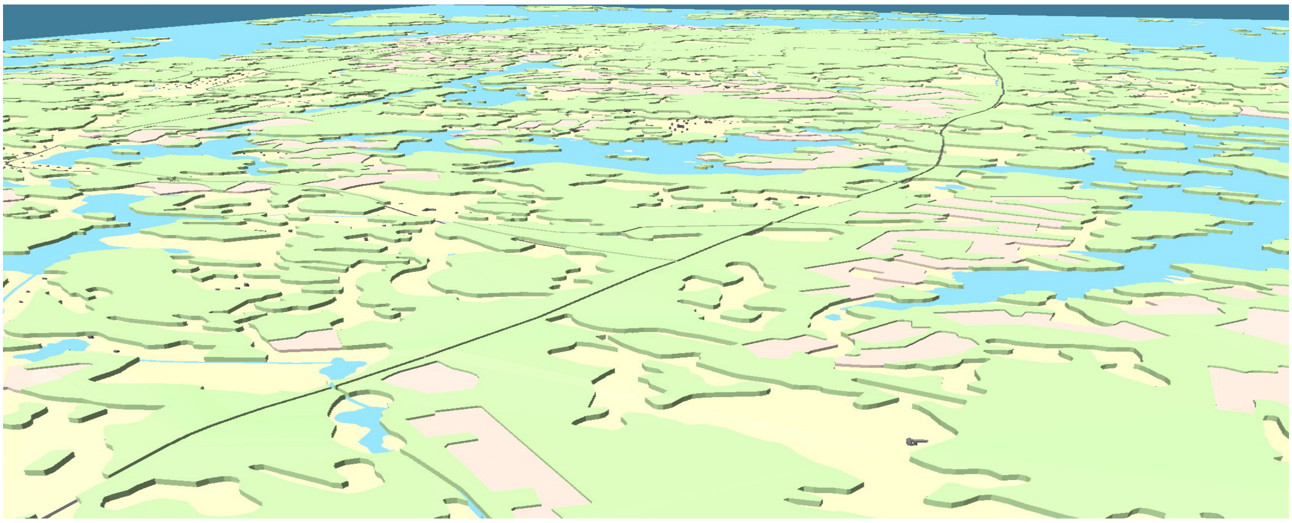

Fig. 5. Visualizer perspective view eastward, covering a part of the scenario area of interest. Note the different heights of forest canopies (green) and built-up areas (pink).

standard procedures. However, due to some yet undetermined cause, some terrain areas remain unattributed after terrain and feature generation, leading to undesired conditions during the simulation. Neither is it possible to load arbitrary terrain tiles into FLAMES, also for reasons which are yet unknown. Finally, a large number of irrelevant attributes are unavoidably propagated to the terrain polygons during terrain generation, making the terrain model more space-consuming than necessary. The latter problem, however, should be straight-forward to correct in future system versions.

4.2.2 Terrain data

The 50x50 m^2 ground elevation database from which our triangulated 3D model was built is probably detailed enough for our needs. However, for realistic modelling of the strongly varying tree canopy transparency, we would need, say, 25x25 m^2 raster information on tree population density, as well as typical tree height and mixing ratio of coniferous and deciduous trees. What was available was a 1:50000 standard topographic map and vector coverage divided into approximately 20 feature classes, which were then condensed into the seven classes supported by FLAMES. The CORINE Land Cover project [18] aims at obtaining higher specificity databases of land cover information for the EU countries. Work on this map product for the scenario area of interest, although ongoing, was not completed in time to be used in our demonstration.

4.3 Doctrine, organization, and equipment models

These models describe the behavior and motion of enemy ground forces according to their doctrine, i.e., the set of tactical rules that is expected to guide the behavior of the opponent's army. This includes telecommunication and transportation along a road network of mechanized forces in hostile territory.

The enemy battalion model consists of approximately 60 vehicles: battle tanks, armed personnel carriers, antiaircraft missile launch vehicles, grenade launcher vehicles, and two types of truck. To create models for these target objects, a table of normalized detection, classification, and identification probabilities, assuming fixed target distance, were needed for each type of sensor. In these tables, objects are assumed to be viewed against a clutter-free image background, noise-free seismic or acoustic environment, etc. Properties of the environment, in particular ground clutter properties at the location of the object and relevant features along the line of sight (atmosphere, vegetation) will then reduce these probabilities.

As source for military doctrines, unclassified Swedish Armed Forces publications were used. The application of these doctrines to our scenario was developed in dialog with military experts.

Five battalion options were included:

- a mechanized infantry battalion plus optional extension consisting of a tank or a howitzer company
- tank battalions with 3 or 5 companies

The descriptions include unit hierarchy down to vehicles of optional types. From these resource descriptions the application *march under low threat* was developed, which includes the sequence of and distance between vehicles and units from vehicles via platoons to companies.

In order to simplify the scenario description, it was decided to use a strict military hierarchy, hence a few vehicles for repair and for transport of ammunition and other supplies were ignored.

The information used in modelling the radio communication needed to stimulate COMINT interceptors described the commanding hierarchy and simple communication rules.

5 Evolutionary development and its environment

In general terms, evolutionary system design and development [19] may be described as a methodology where large development projects are partitioned into an organized collection of separately agreed subprojects or phases. Each phase is developed according to a predefined design contract, which can and must be operationally verified by a set of "users" representing the "customer" organization. Originally, the main rationale of evolutionary design and development is to facilitate close customer and end-user involvement in the development process. But a similar iterative design and development process is also well suited to the needs of a research group which develops comprehensive software while striving to retain much individual responsibility for design and work planning [20]. This was also our experience in the IFD03 project.

5.1 Combining C and MATLAB

The decision to use MATLAB instead of C or some other language (CommonLisp was a seriously advocated alternative) for developing the `FusionNode` was taken because we wanted to spend as little time as possible developing and debugging the implementation of our algorithms, and focus our work instead on algorithm design. The fact that most of our group has significantly more experience in the use of MATLAB than of C influenced this choice. The decision was made easier by the availability of the MATLAB Compiler software, which generates C or C++ code from MATLAB code, enabling us to easily integrate `FusionNode` code into the FLAMES framework. In summary, the decision proved successful, contributing significantly to the productivity of the project.

Using MATLAB had both positive and negative consequences. On the pro side, new ideas may be quickly implemented using MATLAB's rich variety of built-in functions. MATLAB algorithms could often be conveniently debugged by loading input data, previously generated and then `saved` during execution of the compiled system, into an interactive MATLAB session. Also, test code could very easily be added, such as plotting the input or output of a function.

On the con side, MATLAB does not provide fully automatic garbage collection. Instead, the MATLAB system handles allocation and deallocation of memory for objects by use of a heap mechanism. Memory space is automatically reallocated when an object grows. Initially, this caused severe memory fragmentation problems. To diagnose and fix such problems, MS Windows™ diagnostic functions had to be used to obtain information on memory availability. Ultimately, the cause of these problems may be found in a MATLAB programming style not adjusted to the development requirements of large systems. MATLAB allows preallocation of matrices that will contain a large number of data. This is the style to be preferred when developing large systems in MATLAB [4]. It is, however, not necessarily easy to apply this style consistently and our efforts to remove all memory fragmentation problems from the system have not yet been successful.

MATLAB, designed as an interactive environment, will not catch many errors when using the MATLAB Compiler. Even simple things like misspelling a function or variable will cause run-time errors. However, MATLAB Compiler does issue compilation warnings for many errors like these. Thus, MATLAB Compiler messages should be closely watched.

The large size of our terrain database meant that we ran close to the Windows upper limit of 2 GB per process memory size. Using a larger terrain database size would thus not be possible using current technology. Conceivable solutions of this problem include switching to a computer system with 64 bit address space, or changing the terrain database part of FLAMES to use a disk-stored database. In a short term perspective, both approaches seem unrealistic.

5.2 Code versioning and documentation

The CVS (Concurrent Versions System) configuration manager [21] played an essential part in our system development process. While the use of CVS requires considerable discipline from developers (e.g., not committing untested code, writing proper change logs), we would probably not have been able to interface the different parts of the system without using it, or some similar system. We are currently investigating alternatives to CVS for source-code management. We would, for instance, like to be able use MS Visual Studio™ for controlling also the MATLAB Compiler. Visual Sourcesafe™ might then be a viable alternative to CVS.

Since the project became quite hectic as the date for the demonstration drew closer, comprehensive system documentation had to postponed. Several problems would likely have been avoided if a previous research project had properly documented its detailed procedures, so we were brusquely reminded that such documentation is indispensable also in small-to-medium-scale computational research work. Thus, the system documentation of IFD03 was a high-priority work item early this year.

6 Conclusions and future work

The primary purpose of IFD03 is to demonstrate information fusion methods. The next step should be to evolve fusion research by including more realistic settings and therefore involving the end-user, to find out how information fusion can be used in systems of the future. Here it is important to consider the needs of the end-user at each command level in the context where the system is being used, while giving attention to all the different applications of the system. Further work is also needed to deal with issues of how and when information fusion is to be used.

Using IFD03 as a test bed, we further intend to address effectiveness issues in the future, such as:

- what improvement in effectiveness (measured, perhaps, as increased information quality [22], or information gain [15]) of various aspects of situation modelling can be expected from information fusion methods?
- what improvement in effectiveness can be expected from a network-based defence technology, with and without information fusion methods?
- how does delays and "inertia" of various kinds, arising from, e.g., information transmission or information processing, influence expected improvements in effectiveness?

In the near term, the aggregation, tracking, and sensor resource allocation methods need to be completed to a level where systematic Monte Carlo experiments can be performed. In these experiments, computational performance and result quality of all the fusion methods planned for IFD03 will be studied, using the already established scenario environment.

References

[1] Per Svensson and Pontus Hörling. Building an information fusion demonstrator. In *Proc. Sixth Int. Conf. Information Fusion*, pages 1316−1323, Cairns, Queensland, Australia, 8−11 July 2003. Int. Soc. Information Fusion, Sunnyvale, CA, 2003.

[2] Bernard Coulange. *Software Reuse*. Springer Verlag, London, 1998.

[3] David W. Walker, Omer F. Rana, Maozhen Li, Matthew S. Shields, and Yan Huang. The software architecture of a distributed problem-solving environment. *Concurrency: Practice and Experience*, 12(15):1455−1480, 2000.

[4] MATLAB (2004, Apr.) [Online]. Available: http://www.mathworks.com

[5] FLAMES (2004, Apr.) [Online]. Available: http://www.ternion.com/index.htm#products/need.htm

[6] TerraVista Pro Builder (2004, Apr.) [Online]. Available: http://www.terrex.com

[7] Ivar Jacobson. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, Reading, MA, 1994.

[8] Mats Bengtsson and Johan Schubert. Dempster-Shafer clustering using Potts spin mean field theory. *Soft Computing* 5(3):215−228, 2001.

[9] Johan Schubert. Clustering belief functions based on attracting and conflicting metalevel evidence. In Bernadette Bouchon-Meunier, Laurent Foulloy, and Ronald R. Yager, editors, *Intelligent Systems for Information Processing: From Representation to Applications*, pages 349−360, Elsevier Science, Amsterdam, 2003.

[10] Johan Schubert. Clustering belief functions based on attracting and conflicting metalevel evidence using Potts spin mean field theory. *Information Fusion*, to appear.

[11] Johan Schubert. Evidential Force Aggregation. In *Proc. Sixth Int. Conf. Information Fusion*, pages 1223−1229, Cairns, Queensland, Australia, 8−11 July 2003. Int. Soc. Information Fusion, Sunnyvale, CA, 2003.

[12] Neil J. Gordon, David J. Salmond, and Adrian F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107−113, 1993.

[13] Hedvig Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proc. Sixth Int. Conf. Information Fusion,* Pages 800−806, Cairns, Queensland, Australia, 8−11 July 2003. Int. Soc. Information Fusion, Sunnyvale, CA, 2003.

[14] Ronald P. Mahler and Tim Zajic. Multitarget filtering using a multitarget first-order moment statistic. In Ivan Kadar, editor, *Proc. SPIE Vol. 4380 Signal Processing, Sensor Fusion and Target Recognition X,* pages 184−195, Orlando, FL, USA, 16−18 April 2001. SPIE, Bellingham, WA, 2001.

[15] Ning Xiong and Per Svensson. Sensor management for information fusion – issues and approaches. *Information Fusion*, 3(2):163−186, 2002.

[16] Irwin R. Goodman, Ronald P. Mahler, and Hung T. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic Publ., Dordrecht, 1997.

[17] J. Johnson and W. Lawson. Performance modelling methods and problems. In *Proc. IRIS Imaging Systems Group*, 1974.

[18] CORINE Land Cover project (2004, Apr.) [Online]. Available: http://www.gsf.de/UNEP/corine.html

[19] James O. Coplien. Reevaluating the Architectural Metaphor: Toward Piecemeal Growth. *IEEE Software,* 16(5):40−44, 1999.

[20] Jesper Fredriksson, Per Svensson, and Tore Risch. Mediator-based evolutionary design and development of image meta-analysis environments. *J. Intell. Inf. Syst.*, 17(2/3):301−322, 2001.

[21] CVS configuration manager (2004, Apr.) [Online]. Available: http://www.cvshome.org

[22] Stefan Arnborg, Joel Brynielsson, Henrik Artman, and Klas Wallenius. Information awareness in command and control: precision, quality, utility. In *Proc. Third Int. Conf. Information Fusion*, pages ThB1/25−32, Paris, France 10−13 July 2000. Int. Soc. Information Fusion, Sunnyvale, CA, 2000.