Paper I-179

12th International Command and Control Research and Technology Symposium

"Adapting C2 to the 21st Century"

June 19-21, 2007, Newport, RI

Track: T8: C2 Technologies and Systems

Title: **Managing Areas of Interest in Command and Control Information Systems**

Author: Dr. Thomas Nitsche

Organization: FGAN/FKIE

Address: Neuennahrer Straße 20
53343 Wachtberg-Werthhoven
Germany

Phone: +49 . 228 . 9435 - 505

Fax: +49 . 228 . 9435 - 685

E-Mail: nitsche@fgan.de

Managing Areas of Interest

in Command and Control Information Systems

Thomas Nitsche

FGAN/FKIE
Neuenahrer Straße 20, 53343 Wachtberg-Werthhoven, Germany
nitsche@fgan.de

Abstract

A command and control information system (C2IS) has to provide mechanisms to filter the information available in the C2IS to the mission-specific parts that are relevant to the corresponding military commander or other C2IS users. Areas of interest *(AOI) are geographic areas like the surrounding of the user's position and his area of responsibility where the user wants to get informed about other military objects, e.g., own and foreign forces, that are within that specific area.*
The simple form of AOIs can be extended by 1) taking the impact of other objects on us into account by using their effective range, and 2) incorporating future positions as described by plans and orders. For the first extension we can use object properties like their speed, their direction of movement or the range of the object or their weapons to determine a distance from within they can be a potential threat (in case of red forces) or supporter (in case of blue forces). For the second extension we can use machine-readable information about plans and orders to predict our own (planned) positions in the future as well as those of other forces. Examples include a patrol or helicopter crew that can thus include blue forces and potential enemy threads along their (planned) route into their operational picture.
To efficiently *observe such areas, i.e. to compute the AOI within a C2IS, we introduce the concept of* region services*. These services contain all objects of a fixed geographic region. Regions can be defined in a regular manner, hierarchically based on quad-trees or by using areas of responsibilities as a basis.*
A user-specific C2IS instance can now directly and efficiently establish subscription-relations to the relevant *objects around its AOI in order to obtain information about the position, status and behaviour of these objects. If objects including the current user itself now dynamically change their position we merely have to update the information relations to those few objects that enter or leave a region within the AOI, instead of having to consider all objects within the global information Grid.*
Region services thus do not only improve the efficiency for generating a static common operational picture (COP) but can also handle any dynamic changes of object positions.

1. Introduction

The paradigm of network-centric warfare (NCW) [1], [16], [26] is a powerful concept to employ the opportunities of modern IT systems. Its aim is to improve military effectiveness by achieving information superiority [1]. This improves mission effectiveness and helps dominating the battlespace [12], because it allows deploying military power more rapidly and effectively.

For example, in the execution of air-strike missions in Afghanistan during Operation Enduring Freedom (OEF), the time gained from using NCW capabilities was used for contingency planning. The enhanced situational awareness thus allowed the staff officers "to do more tactical and strategic thinking" ([16], page 6).

The global information Grid (GIG) [12] forms the technical basis for realizing the power of the network-centric warfare concept. It is based on the global connection of all military systems, ranging from sensors (e.g., reconnaissance systems) over the command and control systems to actors (effect systems) (cf. Figure 1).

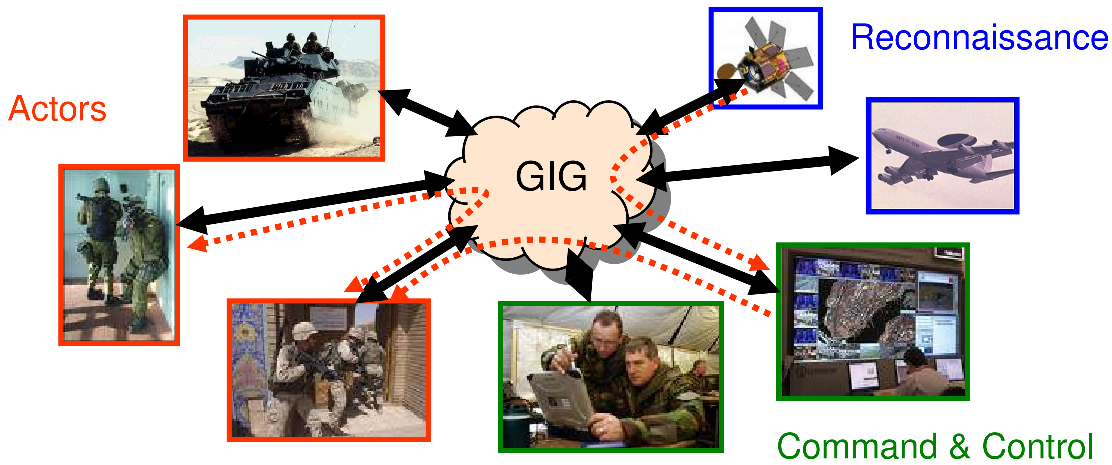

Figure 1 – Combining sensors (reconnaissance systems), actors (effect systems) and command and control systems to form a global information Grid (GIG)

In principle, every system or person may be able to communicate with everybody else in the network. In the information grid, all available information can thus be shared among the different users, provided that the required security criteria (like, e.g., [9] or [10]) are satisfied. As a result, potentially all the available data about own (i.e., blue) and enemy (i.e., red) forces as well as other information becomes available in the GIG.

Sharing all available information is, however, not sufficient for successfully accomplishing a mission. From the information point of view, using *all* available data leads to an explosion of the information space, which cannot be handled appropriately by humans and computers. To handle the problem of information overload [8] we have to restrict the data in an operational picture to only that information that is relevant for the corresponding user and his current mission [7], [13], [21] (Figure 2).

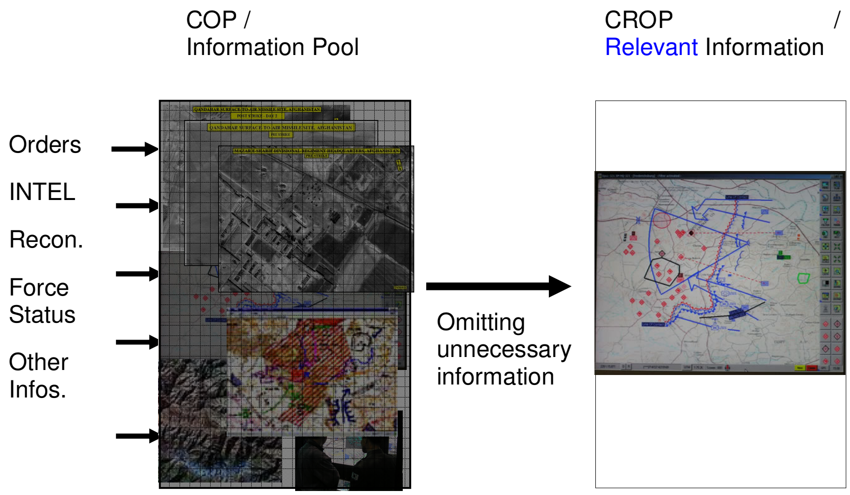

Figure 2 – Restricting the available information to only mission-relevant for the corresponding user

A command and control information system (C2IS) therefore has to provide corresponding filtering mechanisms. *Areas of interest* (AOI) are one such concept.

The remainder of this paper is organized as follows. In Section 2 we describe how geographically based areas of interest can be defined, while in Section 3 we extend the simple form of AOI by 1) taking the impact of other objects on ourselves into account by using their effective range, and 2) incorporating future positions as described by plans and orders. The computation of areas of interest in a command and control information systems (C2IS) is discussed in Section 4. An efficient algorithm based on the concept of region services is described in Section 5. Finally, Section 5.4 concludes.

2. Areas of Interest

An *area of interest* (AOI) is the area of concern to the commander. While in its general form this can include information categories like, e.g., data about current terrorist activities or information about the political or economic situation, we concentrate on geographically based areas here.

The area of interest to a commander or another user of a C2IS will generally include the area of responsibility (AOR) of the corresponding military unit, i.e. the geographical area where the unit is supposed to conduct its operations. Moreover, the AOI will normally also include some surrounding area of the AOR to monitor the behaviour of neighbouring forces as well as enemy forces who could influence the successful completion of the users mission.

In a C2IS the AOI of a user is thus defined as an area like the surrounding of his position or the surrounding of his area of responsibility which he wants to observe. That means that he wants to get informed about other objects that are within that specific area. This is feasible since most of the data presented in a common operational picture (COP) by a C2IS have some spatial reference. This includes positions of blue and red forces, spatial data like information about streets or bridges, and others.

The shape of an AOI can be a simple circular or rectangular surrounding of the current position of the corresponding user. In this case all objects within a specific proximity of that base object are contained with the AOI and are thus to be visualized as the users operational picture. Figure 3 shows a simple example how objects outside the AOI are filtered out.

Figure 3 – Filtering all objects within a certain area of interest (AOI)

By combining such simple shapes we can achieve more complex shapes of the AOI, similar to the data field model (cf., e.g., [19]). Additionally we could use spatial data of streets, rivers and others to describe borders of AOR and AOI, respectively.

To simplify the presentation we abstract from the exact shape of the AOI for the remainder of this paper. We only require that we are able to check if a certain position is inside a given shape. This assumption can obviously be satisfied very easily within a C2IS. Moreover, we also abstract from the exact kind of objects stored in the C2IS, be they blue-force units, red-force units, or others. We only need some attribute values about their position and status (see Section 3.1 below), but do not distinguish otherwise between own or enemy forces here.

In Figure 4 we show an abstracted, simplified view of the AOI to filter relevant objects in an operational picture. Such an abstract view is sufficient to demonstrate the algorithms to compute the AOI, while in a real C2IS we obviously would visualize the objects according to their corresponding military symbols (cf., e.g., [11]) over appropriate maps.

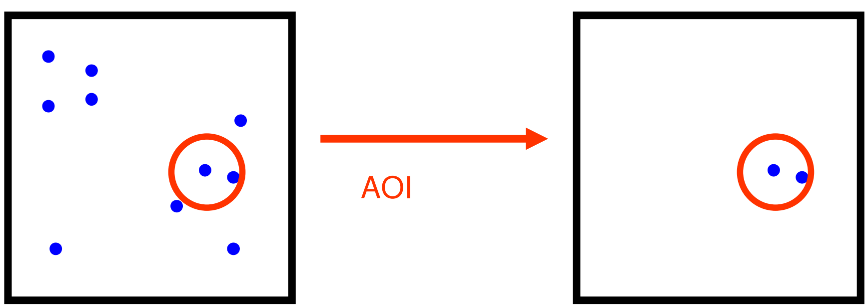

Figure 4 – Filtering of objects within a certain AOI (Abstracted, simplified visualisation of objects as blue dots and AOI as a red circle around an object.)

3. Extended Definition of AOIs

The simple form of AOIs defined above in Section 2 can be extended based on two basic principles: First the potential impact of objects on us on their effect range and second their potential impact in the future based on planned activities.

3.1. Incorporating the Effect Range

For the first extension we can use object properties like their speed, their direction of movement or the range of the object or their weapons to determine a distance from within they can be a potential threat (in case of red forces) or supporter (in case of blue forces).

Mobilized artillery units, for instance, have a much wider effect range than a non-motorized sniper in an urban environment. These object-specific effect ranges can than be used as a basis to define our area of interest. The AOI thus includes all those objects that are either directly located within (the shape of) the AOI, or whose effect range intersects with the AOI.

In Figure 5, for instance, we filtered all objects whose effect range (which is simplified visualized as a green circle) does not intersect with the AOI. Thus only those two objects located directly within the AOI itself (cf. also Figure 4) and the object in the lower right part of the figure remain relevant in this example.

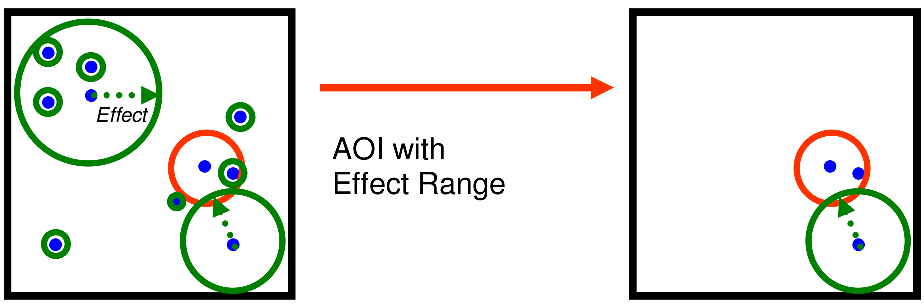

Figure 5 – AOI (simplified visualized as red circle) with effect range (simplified visualized as green circles) of objects (blue dots) taken into account

3.2. Incorporating the Future

For the second extension we can use machine-readable information about plans and orders to predict our own (planned) positions in the future as well as those of other forces. This information about the future can come from plans and orders in the MIP data model [20], from a more formalized battle management language (BML) [22] or from other machine-readable formats.

A patrol or helicopter crew, e.g., can thus include own forces and potential enemy threads along their (planned) patrol route or flight path into their operational picture (cf., e.g., Figure 2 in [17]). Here the user may be interested in the current and/or predicted objects along his route.

Figure 6 shows the construction of such an extended AOI with future positions: The combination of (the shape of) a simple AOI around the planned positions of a user forms the shape of the extended AOI.

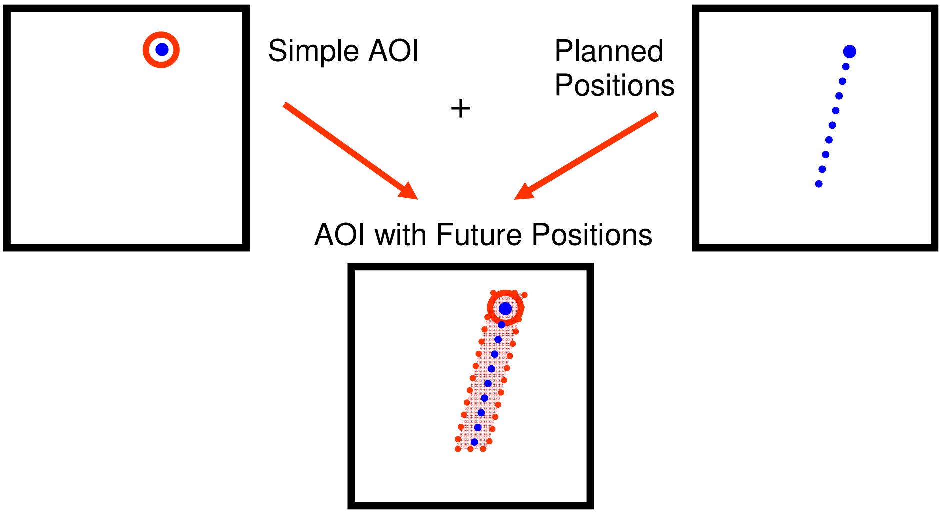

Figure 6 – An area of interest (see the red area) described by a simple AOI and planned positions (see the blue dots). Objects within the AOI except the current and future positions of the user itself are not shown here.

4. Computing Areas of Interest

4.1. Service-Oriented C2IS Architecture

Such user-parameterised areas of interest as described before thus define a set of objects that the user is interested in, i.e. that are relevant for his operational picture. Before we derive how a C2IS can actually compute the AOI we first take a look into the architecture of C2IS.

A command and control information system provides, among others, the following basic functionality: It generates the operational picture for each user and visualizes it. Provided that the information is shared in a global information grid, the C2IS can thus generate a common operational picture (COP) which helps to gain a shared situational awareness.

In a service-oriented architecture a C2IS thus contains at least of COP- and visualization services.[1] Each of them usually has user-specific instances [15]:

- The user-specific visualization service merely displays the content of its corresponding COP-service [24].
- The user-specific COP-service stores the data about the user itself and its current status like its position. This information can be delivered to other COP-services using a publish-subscribe-approach. Subscription-relations to the COP-services of those objects relevant for the current user deliver information about these objects. This implies that the user-specific COP-service can subscribe to the relevant objects to become informed about their position, status and future changes.

[1] The Plato system, an experimental service-oriented network-based C2IS that is currently developed at FGAN-FKIE is built on such a design [25], [15], [18].

In case of multinational and other operations where different C2IS are used, we merely have to import the data like blue-force information from the other systems by using e.g., the MIP [20] data replication mechanisms, and then create COP-services for the imported, remote objects.

Note that such a service-oriented architecture allows the different services to be distributed onto different computers in the GIG which allows the C2IS to operate in a decentralized manner. The shared data of the global information space is thus not necessarily contained in a central database but can also be distributed within the GIG such that each (COP-)service only holds a local portion of that information.

4.2. Simple Approaches and its Limitations

However, before a COP-service can establish subscription-relations to the relevant objects, we first have to determine these relevant objects, i.e. we have to find out which objects are actually within our AOI.

In case that we were satisfied with an AOI containing only objects in our direct neighbourhood we might get the location information of these objects just for free. A special routing protocol for the radio communication in mobile ad-hoc networks (MANETs) additionally transmits GPS data [2]. Unfortunately this is a proprietary protocol, so it only works if all radio communication devices use this protocol which we cannot assume to be valid, especially in multinational operations. Moreover, this approach is restricted to the tactical level and the direct surrounding of ourselves. Objects that are located far more remote but are still relevant for our AOI, e.g. because of their large effect range, cannot be handled here. The same restriction holds for objects representing higher-level command posts that are connected directly via fibre cable instead of radio communication.

Thus in general we have to compute the AOI explicitly.

Let N be the total number of objects available within the GIG. If no further information is available, the AOI computation for a *single* user requires $O(N)$ time, since we have to check the position of all N objects.

Unfortunately, we are not the only user in the system. According to the NCW approach in principle all blue force units can do the same, i.e. they may have an associated COP-service that computes its own AOI. This implies that all objects may define their own local areas of interest for which they have to check the positions of all other objects in turn. Implemented naively, this would lead to an algorithm of quadratic time-complexity $O(N^2)$, while synchronized all-to-all algorithms can do this in $O(N \log N)$ time [4].

This is, however, still not satisfactory. The reason is that the above complexity only holds for a single AOI computation with static object positions. In practice, however, we have a situation where the objects are moving, i.e., they change their positions. This implies that some objects leave our AOI, while others may enter it. Subscription-relations to objects within out AOI only provide us with position updates of those objects that we are already monitoring. So we can determine if an object leaves the AOI, but we will not know if another objects moves towards us. Even worse, we may (and in general will) move ourselves towards other objects that are not in our AOI yet and hence are not monitored by us. As a result, dynamic changes of object positions will soon make the AOI outdated.

As a consequence we had to re-compute the AOI in regular intervals in order update it and hence the CROP. The dynamic behaviour of the objects like blue-force units thus forces us to execute the above described algorithm over and over again. Since it is a global algorithm this not only takes the computation time but – in case of distributed services – also the time for the all-to-all communication scheme wasting a lot of bandwidth, especially if done on the tactical level.

But even if we ignored the above described computational and communication efficiency problems of the algorithm and simply re-computed the AOI in very short time intervals we still would have the problem that we do not know if and when the AOI becomes outdated due to un-monitored objects entering our AOI without notice.

5. Region Services

The reason why the above approach is so inefficient is that we repeatedly check the positions of all available objects within the GIG, while there are in practice only a few objects relevant for our area of interest, their number being in general much less than *N*, i.e., the total number of objects. In the naïve implementation we therefore filter out objects according to their position from the large object set rather than just checking if there exist any objects with a specific position and combining these small sub-sets directly.

5.1. Concept

Approaches in multi-cast communication schemes [5], [6], [23] lead us to the idea of a *region service*. Such a service defines a certain geographical region of the world and contains a list of all objects that are located within this region.

Based on a C2IS software architecture consisting of COP- and visualization services, we hence can extend this by a set of region services, each of them being responsible for a certain region. In its simplest form we can divide the earth (or at least our full operational area) into regions of the same size.

Figure 7-(a) shows the division of an area into regular regions, each of them being of the same size. In such a regular division the borders of each region can be computed very easily and the test if a certain position falls within a specific region can be done very efficiently.

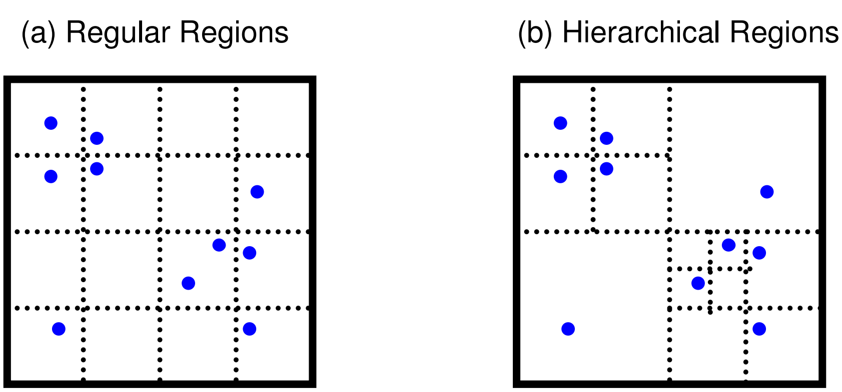

Figure 7 – Division of an area into (a) regular regions with the same size each, and (b) hierarchically defined regions based on a quad-tree division with different sizes but containing the same number of objects

Each of the spatial objects can thus be added to one of the region services in constant time. To detect the objects within the area of interest (AOI) for a user, we now only have to request the objects from those region services that overlap with the user's AOI. This takes *O(N/R)* time on average for each region service, with *R* being their number. If the number *R* of region services is sufficiently large we can thus achieve that each region service is – on average – responsible for only a restricted number of objects.

Unfortunately, the objects in a military domain are in general not regularly distributed around the world but concentrated on the battlefields. We can hence expect many units being located in, e.g., Kabul or Iraq, but probably almost none in, e.g., the Antarctic. This implies that the number of objects located within each regular region may vary significantly. An adaptive approach to the definition of regions takes the actual distribution of objects into account. Organizing region areas hierarchically as a quad-tree (in the case of two-dimensional coordinates, see Figure 7-(b)) or as an oct-tree (in the case of three-dimensional coordinates) allows region services to be defined in such a way that they all contain (almost) the same number *K* of objects. This leads to a *O(K)* constant time algorithm for retrieving the objects within an AOI, provided the size of the AOI is small compared to the whole battlefield, i.e. the AOI covers only a fixed number of known regions. However, the check which region a certain position belongs to requires *O(log (N/K))* time in this case, so dynamic object movements are more expensive here than in the case of regular regions.

Another distribution of the regions can be defined by using the areas of responsibility for the different units, potentially extended by some additional regions to handle areas not covered by any AOR. Such AOR based regions have the advantage of being adapted to the actual distribution of units on the ground, with a natural hierarchy based on the chain of command. However, while they are close to actual military regions, their computation is generally rather difficult, since their borders are not regular but may use any complex curve following e.g. the form of rivers or streets. Since the region services are not directly visibly to the user of the C2IS but only used internally to compute the AOI, the other two forms of regions (regular and hierarchical) are probably preferable in terms of computational efficiency.

The concept of region services not only improves the efficiency for generating a static COP but can also handle dynamic changes of object positions. In general all objects – except for those installed at a fixed place – can and do move. This means that the objects within and near the AOI of a user may change their positions and hence may leave or enter the AOI dynamically. However, not only the surrounding objects but also the user itself may move. Therefore a static computation of the AOI (possibly extended by a small surrounding of the AOI) does not work. Instead we have to compute and update the AOI dynamically. If an object is located in the area of one of the region service that overlaps with our AOI, we can subscribe to that object to be informed about changes of its position and other attribute values. If that object finally leaves the area of the region service and is thus out of our AOI, we can cancel the subscription of that object. If, on the other hand, a currently un-monitored object changes its position in such a way that it is entering the area of one of the region services that overlap with our AOI, we will be informed by the region service about this object and can immediately subscribe to it. We therefore only have to check for a limited amount of objects if they are within our AOI.

Note that a region service should in average contain not only one but multiple objects, because this shall give the best trade-off between 1) the time required to manage the objects within the region service and their potential dynamic movements from one region to another region on one hand, and 2) the time to determine the region service(s), i.e. to check where a certain object belongs to, on the other hand.

5.2. AOI Computation using Region Services

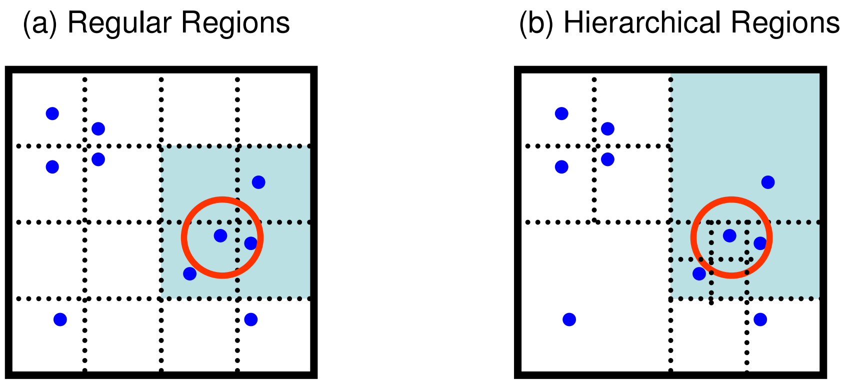

Figure 8 – Computing the area of interest using region services: (a) in case of regular regions, (b) for hierarchical regions

The actual algorithm for computing the areas of interest with the support of region services works as follows (see also Figure 8):

1. First all objects have to register themselves at their corresponding region service, i.e. they publish themselves.

   In case of regular regions this takes constant time for each object and can be done in parallel for each region service, while in case of hierarchical regions it takes O(log R) time for each object to find its proper region, and since adding new objects may change the region hierarchy this has to be done sequentially.

   Once all the objects are registered at their corresponding region service, their COP-services take care of their movements: If an object leaves the area of one region and enters the area of another region, the service automatically de-registers the object at the old region service and registers itself at the new one. These dynamic updates may happen in parallel to the AOI computation. Provided each object is registered at least with one region service at all times, this does not matter. We thus only have to ensure that an object is registered at the new region service before it de-registers itself in the old region.

   The re-registration of moving objects could also be done by the region service rather than by the object itself. This required the region services to establish a subscription relation to the objects located within their region in order to get informed if they left the region. In this case the region service had to register the object at the new region service before de-registering it locally. However, here we had to transfer all position changes of objects to the region service in order to check if they have left or not, leading to a lot of unnecessary communication. It is therefore better to operate as described above and do the check on the side of the objects themselves, as this smart push approach [13] is more efficient.

2. To compute the AOI of one user, we have to determine the relevant region services. This includes all those regions that intersect with the shape of the AOI.

3. Then we read all objects from the relevant region services determined in phase 2 before. We thus get a list of object identifiers, or links to their corresponding COP-services. In order to get informed if that object set changes (due to moving objects) we have to create a subscription-relation to the region-service.

4. Read further information, especially the location, of these objects. This can be done by creating subscription-relations to the corresponding COP-services, which also informs us about later changes of their values and positions.

5. Finally we check which of the objects actually belong to the AOI and filter out the other ones.

   Note that if one of the objects received in phase 3 changes its position we have to check again if it has move into or out of the AOI. However, since from phase 3 we get only objects in the direct neighbourhood of the AOI, there are generally much less objects to be filtered out than had been in the simple algorithm of Section 4.

The different phases of the AOI computation algorithm are shown in Figure 9.

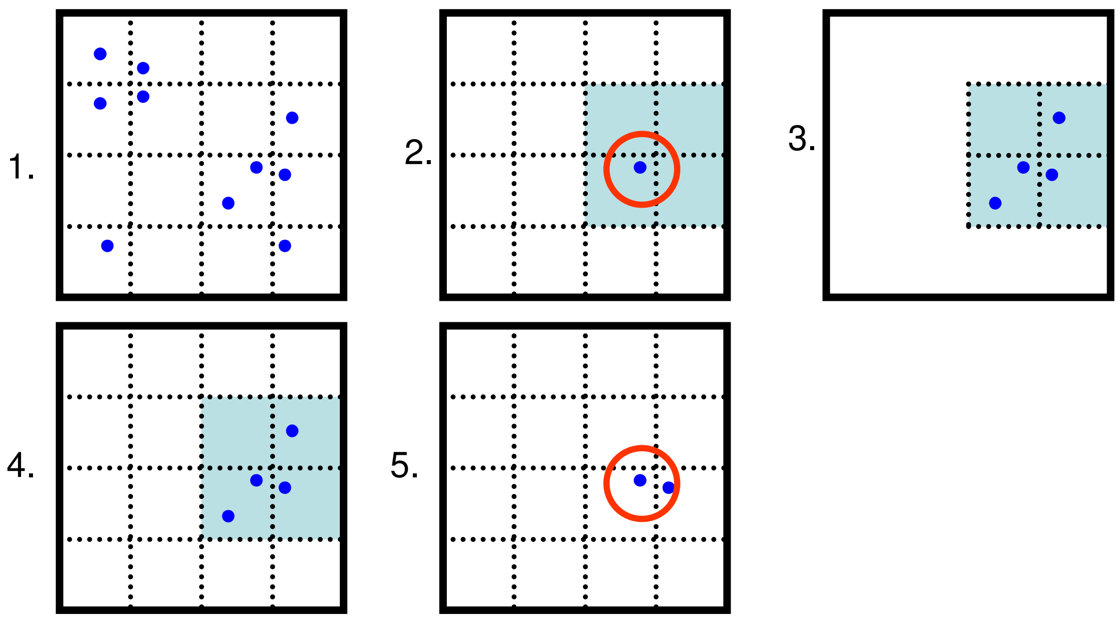

Figure 9 – Different phases of the algorithm to compute the AOI based on region services with regular regions: (1) Register objects at corresponding region services, (2) Determine relevant regions for AOI, (3) Get objects from relevant region services, (4) Read object positions, (5) Determine objects within the AOI

5.3. AOI Computation using Region Services with Effect Range

The computation of areas of interest with effect ranges (see Figure 10) is merely a small extension (in phases 2, 5 and 7, respectively) of the above base AOI computation algorithm.

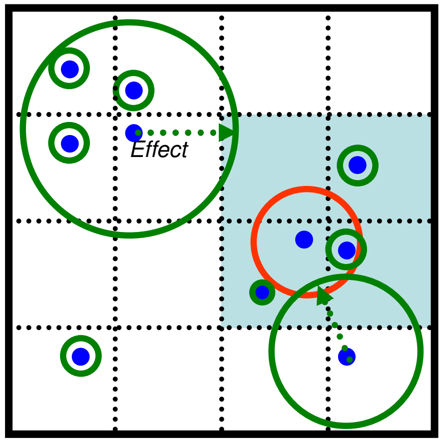

Figure 10 – Computing the area of interest with effect ranges using region services for regular regions

Its different phases (see Figure 11) are as follows:

1. First register all objects at their corresponding region services as done before.

2. Then all objects additionally have to register at all those regions that they have an effect on. Note that in this case one object may be registered at multiple region services.

3. Determine the relevant region services for the AOI as before, i.e., check which regions intersect with the shape of the AOI.

4. Read all objects from those relevant region services by establishing proper subscription relations. In addition to the objects located within certain regions here we also get objects that might have an effect within that region.

5. Since effect objects may be registered at multiple region services, we have to merge the object sets from different regions.

   For example, in Figure 11-(4) the object in the lower right part of the figure is – due to its effect range – registered at two of the relevant regions (shown in light blue).

6. Subscribe to the corresponding objects and thus read their positions.

7. Finally filter out objects if neither their position nor their effect range intersects with the shape of the AOI. The other objects are than part of the AOI and to be displayed.

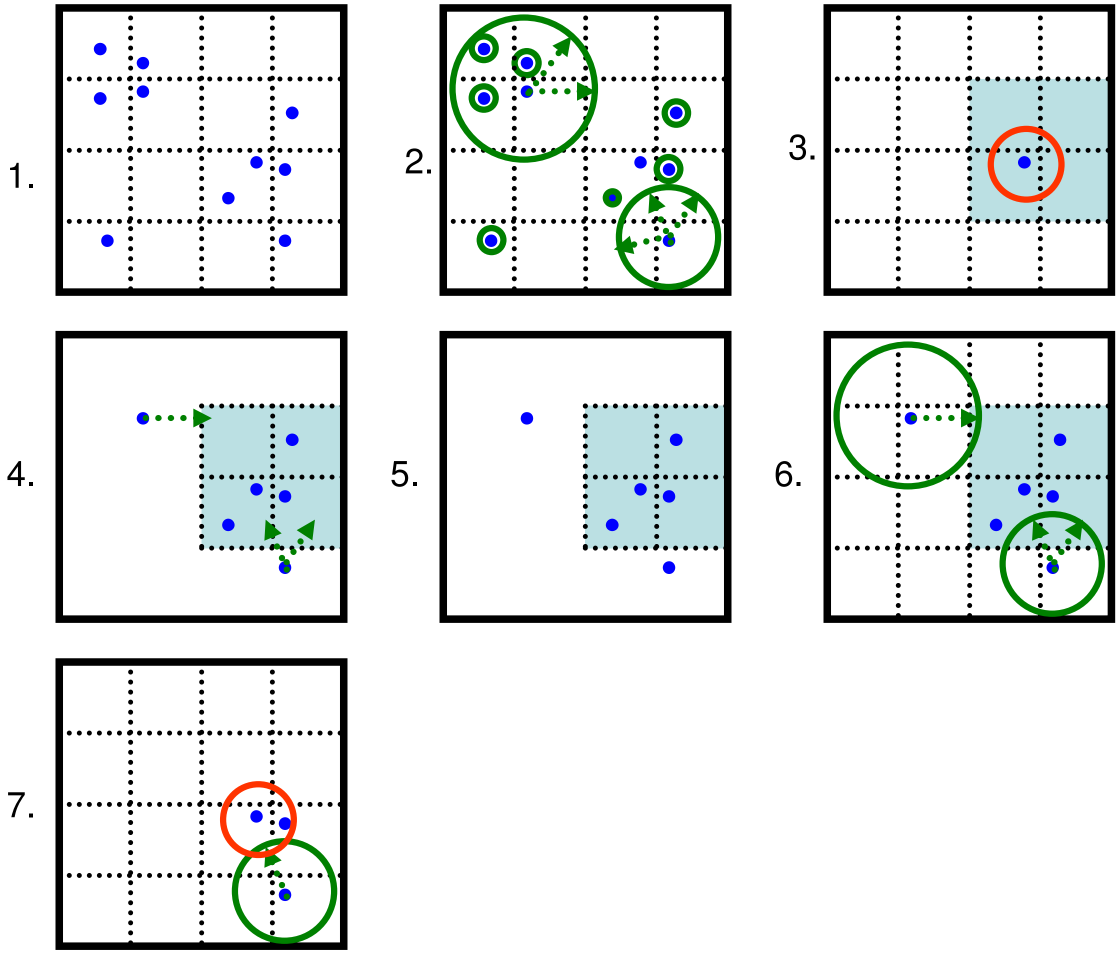

Figure 11 – Different phases of the algorithm to compute AOI with effect ranges based on region services with regular regions: (1) Register objects at corresponding region services, (2) Object registration at region services with effect range, (3) Determine relevant regions for AOI, (4) Get object sets from relevant region services, (5) Merge object sets, (6) Read object positions and effect range, (7) Determine objects within the AOI

The AOI computation based on hierarchical regions works analogously to the computation on regular regions shown in Figure 11.

5.4. AOI Computation using Region Services with Effect Range and Future Positions

In the previous section we extended the base algorithm for computing areas of interest using region services from Section 5.2 by the effect range. A similar extension could be done if we want to incorporate future positions, i.e. planned (or predicted) positions of own forces and predicted positions of enemy forces.

Similarly to the effect range we have to compute the shape of the future positions of each object. This line or area determines the regions where the object may be located, i.e. at which region services the object has to be registered.

The shape of the AOI is a combination of the simple AOI shape and the future positions (cf. Figure 6). The same holds for the potential effect range, although here we could take the (decreasing) fuel status of the unit into account, i.e. the effect range of an object might decrease in the future according to its predicted consumption.

The extended shape of the AOI as well as those of the future positions of objects combined with the effect range can now be used to compute the AOI, as shown in Figure 12.

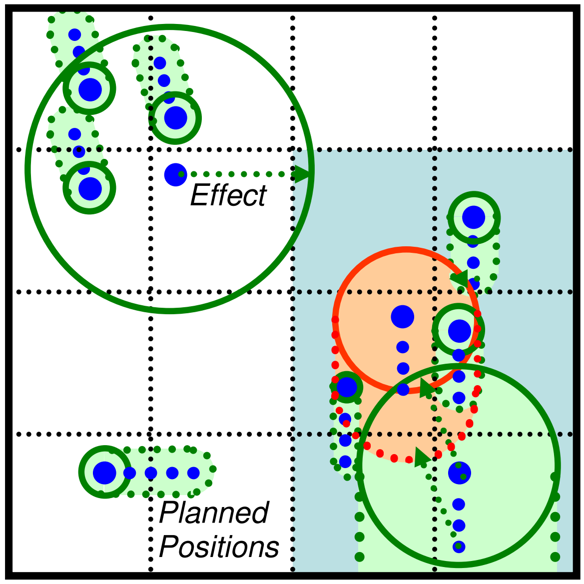

Figure 12 – Computing the AOI with effect ranges and future positions using region services for regular regions

The different phases of the algorithm are similarly to that of Section 5.3, with the modification in the shape of the effect range and that of the AOI: Here we potentially have to take future positions into account:

1. First we determine for each object, if there are potential future positions available, e.g., defined by some plans and orders for the corresponding unit, or a prediction on its potential behaviour. This calculation can in principle be done in parallel. As a result we in general get an area where the object may be located. This area can be a simple connected list of lines or even a single point if no future positions are available.

2. Based on the basic shape of the AOI and the planned positions of ourselves we compute the shape of the AOI as show in Figure 6.

3. The same has to be done for the effect ranges: They also have to incorporate potential future positions.

4. Now we proceed as done in the simpler versions of the algorithms before, i.e., we continue with phase 1 of the algorithm from Section 5.3 (AOI computation with effect ranges).

   First we register all objects at their corresponding region services. This step could be omitted, since in general the current object position is part of the area of the effect range. However, handling the object positions explicitly without their effect range or potential future positions allows more freedom for the user of the C2IS. If the local calculation of the full AOI (cf. phase 10 below) is too complex and takes too much time, the user can decide to ignore the future positions of objects and use the AOI only with the current object values.

5. Next all objects additionally have to register at all those regions that they have an effect on. In case of future positions this includes the effect range the object may have in the future at its planned or predicted position. In general we will probably register each object at potentially multiple region services, if the effect range or the future position area covers larger areas.

6. Determining the relevant region services for the AOI now requires checking which regions intersect with the shape of the AOI now or potentially in the future.

7. Get those objects that are located within the relevant regions or that might have an effect on that region now or potentially in the future. As before this information can be gained by establishing proper subscription relations to the corresponding region services.

8. Due to the effect ranges and the potential future positions, objects may be registered at multiple region services. We thus have to merge the object sets from the different regions.

   For example, in Figure 13-(7) the object in the lower right part of the figure is – due to its effect range – registered at four of the six regions relevant for the AOI (shown in light blue).

9. By subscribe to the corresponding objects and we can read their positions, status values determining their effect range as well as potential plans and orders that influence its future positions.

10. Finally we have to determine which objects are within the AOI. Here we have to take into account both the effect range as well as potential future positions of ourselves and of the other objects.

    This local calculation can become quite complex if there are larger areas for potential future object positions. It becomes even more complicated if we tried to eliminate those objects that may never have a potential effect on us at any moment now or in the future.

    In the example of Figure 13-(10) the upper right object is such an example. Its effect range intersects with the AOI sometimes in the future. However, if we assume that this object is moving with the same speed and direction as we do, the potential effect only happens at a time when we have left already that area (cf. also Figure 12). Thus this object could actually be eliminated from the AOI, since it has no effect on us. The same holds for the object directly on the lower left of our current position: If it moved synchronously with us it would never appear in our AOI. However, if that object stopped it would appear within our AOI.

Figure 13 visualizes the different phases of the final algorithm.

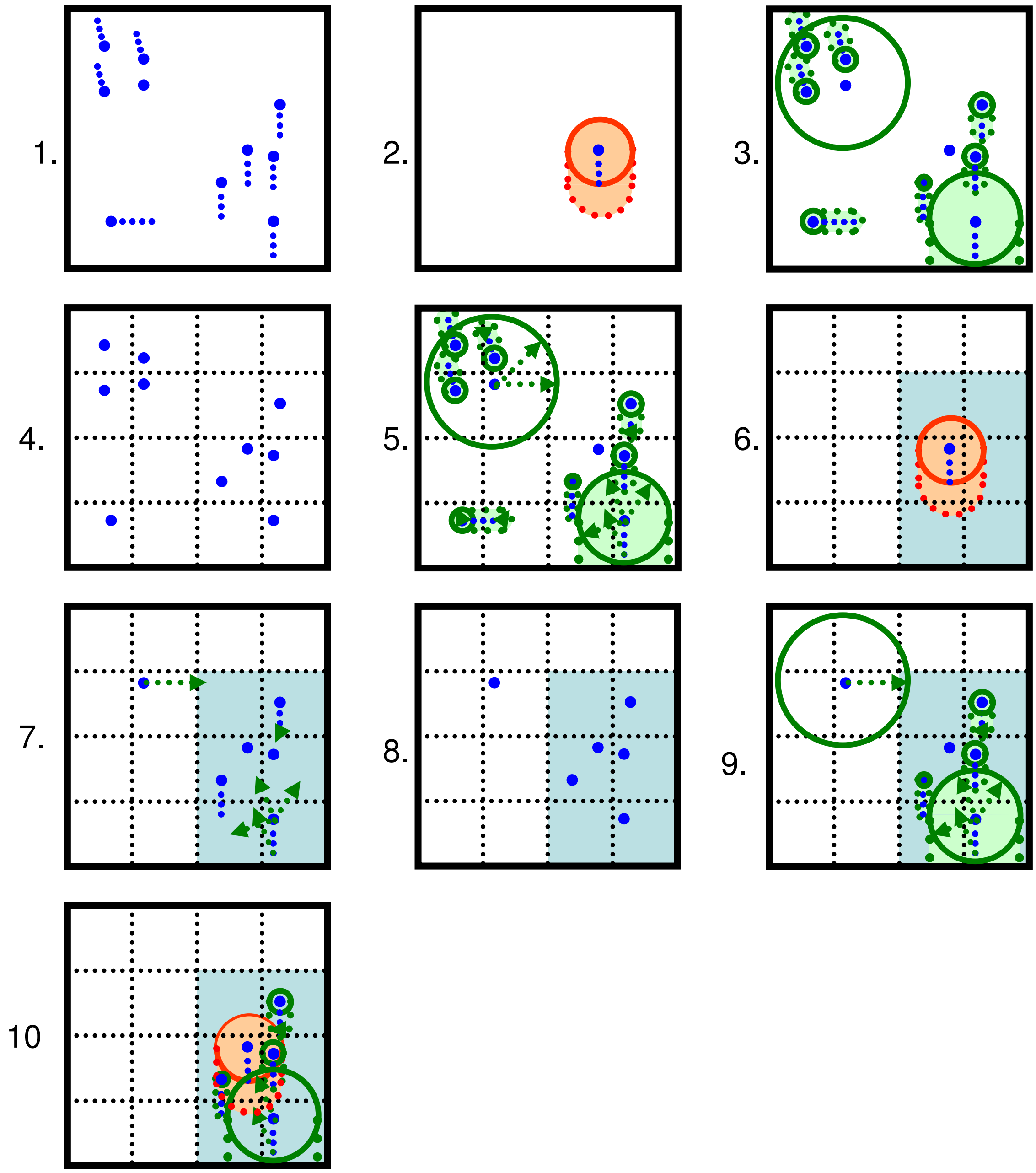

Figure 13 – Different phases of the algorithm to compute AOI with effect ranges and future positions based on region services with regular regions: (1) Determine the future positions of objects (if available), (2) Compute the shape of the AOI incorporating future positions, (3) Compute the effect range of objects incorporating potential future positions, (4) Register objects at corresponding region services, (5) Object registration at region services with effect range and future positions, (6) Determine relevant regions for AOI, (7) Get object sets from relevant region services, (8) Merge object sets, (9) Read object positions and other attributes (effect range status and planned positions), (10) Determine objects within the AOI, incorporating potential future positions

6. Conclusion

A command and control information system (C2IS) has to provide mechanisms to filter the information available in the C2IS to the mission-specific parts that are relevant to the corresponding military commander or other C2IS users. *Areas of interest* (AOI) are geographic areas like the surrounding of the user's position and his area of responsibility where the user wants to get informed about other military objects, e.g., own and foreign forces, that are within that specific area.

The simple form of AOIs can be extended by 1) taking the impact of other objects on us into account by using their effective range, and 2) incorporating future positions as described by plans and orders. For the first extension we can use object properties like their speed, their direction of movement and the range of the object or the range of their weapons to determine a distance from within they can be a potential threat (in case of red forces) or supporter (in case of blue forces). For the second extension we can use machine-readable information about plans and orders to predict our own (planned) positions in the future as well as those of other forces. Examples include a patrol or helicopter crew that can thus include blue forces and potential enemy threads along their (planned) route into their operational picture.

To *efficiently* observe such areas, i.e. to compute the AOI within a C2IS, we introduce the concept of *region services*. These services contain all objects of a fixed geographic region. Regions can be defined in a regular manner, hierarchically based on quad- or oct-trees or by using areas of responsibilities as a basis.

A user-specific C2IS instance can now directly and efficiently establish subscription-relations to the *relevant* objects around its AOI in order to obtain information about the position, status and behaviour of these objects. If objects including the current user itself now dynamically change their position we merely have to update the information relations to those few objects that enter or leave a region within the AOI, instead of having to consider all objects within the global information Grid.

Region services thus do not only improve the efficiency for generating a static common operational picture (COP) but can also handle any dynamic changes of object positions.

We are currently about to implement the AOI information filtering and their efficient computation using region services within the Plato system, an experimental service-oriented network-based C2IS [25], [15], [18], [24].

As future work we want to evaluate the different kinds of region services (regular and hierarchical) in more detail using simulations of realistic object movements and the corresponding changes of AOIs.

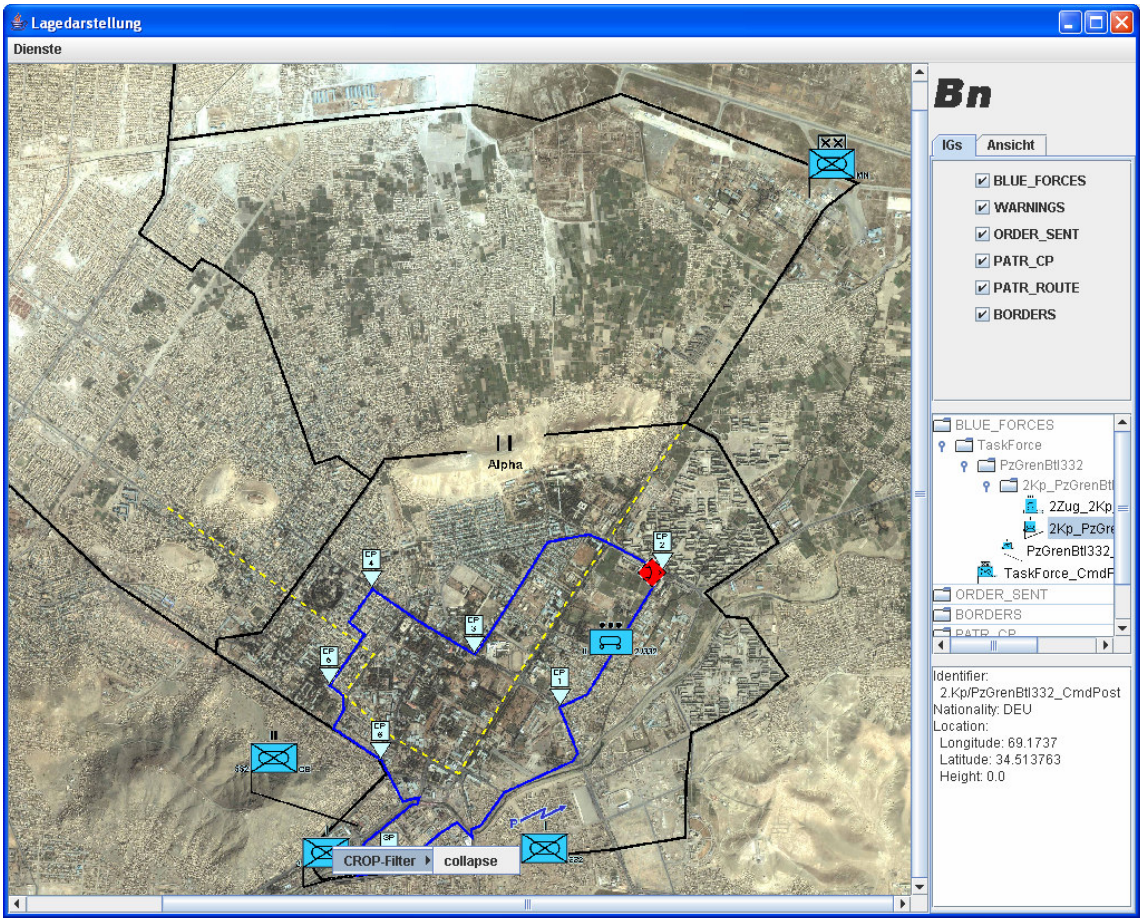

Figure 14 – Experimental service-oriented network-based C2IS

Acknowledgements

The author would like to thank Ret. Col. Gerd Dürr, Marc Spielmann, Marko Jahnke and Christian Riechmann for providing valuable inspirations and comments.

References

[1] David S. Alberts and Richard E. Hayes. Power to the Edge: Command and Control in the Information Age. CCRP Publication Series, Washington, DC, 2003.

[2] Thomas Bachran, Harald H.-J. Bongartz and Alexander Tiderko. A Framework for Multicast and Quality based Forwarding in MANETs. In *CCN'05: Proceedings of the 3rd IASTED International Conference on Communications and Computer Networks*, Marina del Raym California, USA, ACTA Press, 2005.

[3] Alexander B. Bordetsky and Frederick Hayes-Roth. Hyper-Nodes for Emerging Command and Control Networks: The 8th Layer. In *11th International Command and Control Research and Technology Symposium (ICCRTS) – Coalition Command and Control in the Networked Era*, Cambridge, UK, Sept. 2006.

[4] Joshua Edward Barnes and Piet Hut. A hierarchical *O(N log N)* force calculation algorithm. *Nature*, Vol. 324, No. 4, pp. 446-449, Dec. 1986.

[5] Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf. Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems*, Vol. 19, No. 3, pp. 332-383, Aug. 2001.

[6] Antonio Carzaniga, Mattew J. Rutherford, and Alexander L. Wolf. A Routing Scheme for Content-Based Networking. In *Proceedings of IEEE INFOCOM 2004*, Hong Kong, China, March 2004. http://www.ieee-infocom.org/2004/Papers/19_1.PDF.

[7] Edward Dawidowicz, Vairzora Jackson, Thomas E. Bryant and Martin Adams. The Right Information … and Intelligent Nodes. In *8th International Command and Control Research and Technology Symposium (ICCRTS)*, Fort McNair – National Defense University, Washington, DC, 17-19 June 2003.

[8] Peter J. Denning. Infoglut. *Communications of the ACM*, Vol. 49, No. 7, pp. 15-19, July 2006.

[9] U.S. Department of Defence. DoD Directive 5230.11, "Disclosure of Classified Military Information to Foreign Governments and International Organisations". http://www.dtic.mil/whs/directives/corres/pdf/d523011_061692/d523011p.pdf, June 16, 1992.

[10] U.S. Department of Defence. DoD Directive 8500.1, "Information Assurance", http://www.dtic.mil/whs/directives/corres/pdf/d85001_102402/d85001p.pdf, October 24, 2002.

[11] U.S. Department of Defence. DoD Interface Standard, "Common Warfighting Symbology", January 30, 1999.

[12] U.S. Government Accountability Office. Defense Acquisition – The Global Information Grid and Challenges Facing Its Implementation. GAO-04-858, Report to Subcommittee on Terrorism, Unconventional Threads, and Capabilities, Committee on Armed Services, House of Representatives. July 2004.

[13] Frederick Hayes-Roth. Two Theories of Process Design for Information Superiority: Smart Pull vs. Smart Push. In *Command and Control Research and Technology Symposium (CCRTS) – The State of The Art and the State of the Practice*, San Diego, CA, June 2006

[14] Norman Jansen and Marc Spielmann. Personal Communication. 2006.

[15] Norman Jansen, Thomas Nitsche, Marc Spielmann and Irmtrud Trautwein. An Architectural Framework for Network-Based Service-Oriented Command and Control Information Systems (Ein Architekturrahmenwerk für netzwerkbasierte serviceorientierte Führungsinformationssysteme; in German). Technical Report, FKIE, Wachtberg, Germany, in preparation, 2007.

[16] John Kruse, Mark Adkins and Kimberly A. Holloman. Network Centric Warfare in the U.S. Navy's Fifth Fleet – Web-Supported Operational Level Command and Control in Operation Enduring Freedom. In *Proceedings of the 38rd Hawaii International Conference on Systems Sciences (HICSS)*, IEEE Computer Society, January 2005.

[17] Samuel R. Kuper, Robert G. Eggleston and Ron Scott. Using Work-Centered Support System Technology to Enhance Command and Control. In *8th International Command and Control Research and Technology Symposium (ICCRTS)*, Fort McNair – National Defense University, Washington, DC, 17-19 June 2003.

[18] Sylvia Käthner and Marc Spielmann. A Base Component for Network-Based Service-Oriented C4ISR Systems. In *Coalition C4ISR Architectures and Information Exchange Capabilities (RSO-IST-042-RSY-014)*, The Hague, The Netherlands, 27-29 September 2004.

[19] Björn Lisper and Per Hammarlund. The Data Field Model. MRTC report ISSN 1404-3041 ISRN MDH-MRTC-42/2001-1-SE, Mälardalen, Real-Time Research Centre, Mälardan University, June 2001.

[20] Multilateral Interoperability Programme. The C2 Information Exchange Data Model (C2IEDM Main). http://www.mip-site.org.

[21] Thomas Nitsche. Information Access in Tactical Command and Control Information Systems. In *11th International Command and Control Research and Technology Symposium (ICCRTS) – Coalition Command and Control in the Networked Era*, Cambridge, UK, Sept. 2006.

[22] Ulrich Schade and Michael R. Hieb. Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests and Reports. In *11th International Command and Control Research and Technology Symposium (ICCRTS) – Coalition Command and Control in the Networked Era*, Cambridge, UK, Sept. 2006.

[23] Francesc Sebé and Josep Domingo-Ferrer. Scalability and security in biased many-to-one communication. *Computer Networks*, Vol. 51, Issue 1, pp. 1-13, Jan. 2007.

[24] Marc Spielmann. Visualization of Spatial Data in Command and Control Information Systems. In *Military Communication and Information Systems Conference (MCC 2006)*, Gdynia, Poland, Sept. 2006.

[25] Marc Spielmann. Concept of a Distributed Command and Control Information System to Support Network Centric Warfare (Konzeption eines verteilten Führungsinformationssystems zur Unterstützung der Vernetzten Operationsführung; in German), FKIE-Report, Wachtberg, Germany, to appear 2007.

[26] Clay Wilson. Network Centric Warfare: Background and Oversight Issues for Congress. *CRS Report for Congress RL32411*, June 2004.