LOCATION MANAGEMENT IN DISTRIBUTED, SERVICE-ORIENTED COMMAND AND CONTROL SYSTEMS

Thomas Nitsche
Research Institute for Communication, Information Processing and Ergonomics (FGAN-FKIE)

Keywords: Location management, non-homogenous distribution, neighbourhood, areas of interest, network centric operations, command and control.

Abstract: In this paper we propose an efficient location management scheme for large amounts of mobile users and other objects in distributed, service-oriented systems. To *efficiently* observe geographic areas of interest (AOI) in command and control information systems (C2IS), 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. To handle in-homogenous distributions of objects we propose a combination of regular and hierarchical regions. 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 but can also handle any dynamic changes of object positions.

1 INTRODUCTION

Location management, i.e. the determination and tracking of object positions, plays an important role in all systems where such location information is used. In ubiquitous systems it delivers valuable information about the geographic context of a user and his surrounding.

In this paper we handle the question how such location-awareness can be realized in large distributed systems. Objects in such systems are mobile users, sensors as well as spatial objects representing specific geographic locations like bridges or street corners. These (mobile) objects can be represented as service instances or agents. To adapt themselves to the current situation they must be able to observe the relevant context, i.e. the other objects in their direct neighbourhood, but also potentially far more remote objects that might have an effect on them.

In network-centric operations (NCO) (Alberts & Hayes, 2003; Kruse, Adkins & Holloman, 2005; Wilson, 2004) as well as in crises management for disaster scenarios (Arnold, et al., 2004; Denning, 2006a; Jungert, Hallberg & Hundstad, 2006) command and control information systems (C2IS) (Alberts & Hayes, 2006) are used for managing the operations. In both cases the user has to make proper decisions and react quickly, i.e. in real-time, according to the current situation.

Since the number of objects and users is too large to be directly observable – it may reach millions in network-centric operations – the C2IS has to provide mechanisms to filter the available information to the mission-specific parts that are relevant to the corresponding users.

Areas of interest (AOI) are geographic areas like the surrounding of a user's position and his area of responsibility where the user wants to get informed about other objects and users that either are within that specific area or that may have an effect on that area. The objects themselves are in general distributed in-homogenously over the overall operational area (cf., e.g., (Mitschke & Peter, 2001) for distribution of users within a disaster scenario) and the objects are not static but may change their positions with different speed: While pedestrians are moving slowly, vehicles or even helicopters or air-planes may change their position at very high speed. Since each of the objects may correspond to a user of the C2IS, we therefore have to manage the location of potentially millions of mobile users in parallel. Since the applications require a reaction in

Nitsche T. (2007).
LOCATION MANAGEMENT IN DISTRIBUTED, SERVICE-ORIENTED COMMAND AND CONTROL SYSTEMS.
In *Proceedings of the Second International Conference on Software and Data Technologies - PL/DPS/KE/WsMUSE*, pages 144-151
DOI: 10.5220/0001347001440151

real-time, the corresponding location management and AOI computation has to be done in an efficient manner as well.

The remainder of this paper is organized as follows. In Section 2 we describe the global information grid and how geographically based areas of interest can be defined in command and control information systems. The location management in C2IS is discussed in Section 3. An efficient algorithm based on the concept of region services working for non-homogenous distributions is described in Section 4. Finally, Section 5 concludes.

2 AREAS OF INTERST IN COMMAND AND CONTROL SYSTEMS

2.1 The Global Information Grid

The global information grid (GIG) (Blais, Goerger, Richmond, Gates & Willis, 2005; U.S. Government Accountability Office, 2004) forms the technical basis for realizing the power of the network-centric operations concept. As is the computational or data grid formed as a connection of compute or data centres with the users (Foster & Kesselmann, 1998), so is the information grid: It is based on the global connection of all users and systems, ranging from sensors over the command and control systems to actors. In principle, every system or user may be able to communicate with everybody else in the network and can have access to remote services and data. In the information grid, all available information can thus be shared among the different users, provided that the required security criteria are satisfied.

Sharing all available information is, however, not sufficient. 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 (Denning, 2006b) 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 (Hayes-Roth, 2006; Nitsche, 2006).

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

2.2 Areas of Interest

An *area of interest* (AOI) is the area of concern to the user. Here we concentrate on geographically based areas. The area of interest to a user of a C2IS will generally include the (geographical) area of operation of the corresponding user. Moreover, the AOI will normally also include some surrounding area to monitor the behaviour of neighbouring users (be they friendly or hostile), who could influence the successful completion of the user's 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 operation 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) (Mittu & Segaria, 2000) by a C2IS have some spatial reference. This includes (physical) positions of mobile users (Hightower, Borriello, 2001) representing own and foreign forces, spatial data like information about streets or bridges, and others.

The shape of an AOI can be seen as a combination of simple circular or rectangular surroundings of the current user's position and other spatial data (cf., e.g., (Lisper & Holerin, 2002)). 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. Moreover, we also abstract from the exact kind of objects stored in the C2IS. We only need some attribute values about their position and status, but do not distinguish otherwise between them here.

2.3 Incorporating Effect Ranges

The simple form of AOIs defined above can be extended based on the potential impact of objects on us on their effect range, but also on their potential impact in the future based on planned activities (Schade & Hieb, 2006).

For incorporating the effect range of objects 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 or supporter (cf., eg., (McEneaney & Singh, 2007; Scott, Wan, Rico, Furusho & Cummings, 2007)).

In Figure 1, 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 area of interest itself and the object in the lower right part of the figure remain relevant in this example.

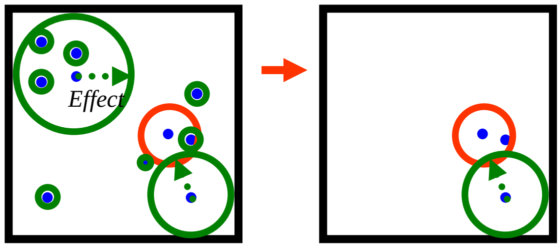

Figure 1: Area of interest (simplified visualized as red circle) with effect range (simplified visualized as green circles) of objects (blue dots) taken into account.

3 LOCATION MANAGEMENT IN C2IS

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. For its computation we hence have to determine which objects are contained with the AOI because they are within a specific proximity of that user or might have an effect on him.

3.1 Service-Oriented C2IS

In this paper we assume a service-oriented architecture of the C2IS (Käthner & Spielmann, 2004). The overall functionality is provided in the form of services with user-specific instances, each of them can be distributed onto different computers in the grid 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 grid among the different service instances, where each of them only hold a local portion of the overall information.

Relevant for the location management is a so-called COP-service. The user-specific instance of that 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. A user-specific COP-service can subscribe to those objects relevant for this user in order to become informed about their position, status and future changes.

3.2 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 (Bachran, Bongartz & Tiderko, 2005). 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 or disaster scenarios incorporating different organisations with different equipment. 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 that are connected directly via fibre cable instead of radio communication.

Note that an IP-based location management scheme (cf., e.g., (Shin, Park, Jung & Kim, 2006)) also does not work here: Due to the different organisations involved that may use their own sub-networks as well as command and control hierarchies the geographic neighbourhood is not necessarily correlated with that in a network, even if we used tactical radio on the lower tactical levels. Similar holds for RFID-based tracking systems like (Satoh, 2006).

Thus in general we have to compute the AOI explicitly.

Let N be the total number of objects available within the grid. 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 NCO approach in principle all users (except for hostile 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 (Barnes & Hut, 1986).

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 mobile, i.e., they change their positions at any time. 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 to update it. The dynamic behaviour of the objects 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.

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.

4 REGION SERVICES

The reason why the above approach is so inefficient is that we repeatedly check the positions of all available objects within the grid, 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 full object set rather than just checking if there exist any objects with a specific position (in our surrounding) and combining these small sub-sets directly.

4.1 Concept

Approaches in multi-cast communication schemes (Carzaniga, Rosenblum & Wolf, 2001; Carzaniga, Rutherford & Wolf, 2004; Sebé & Domingo-Ferrer, 2007) 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-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 (cf. Figure 2-(a)).

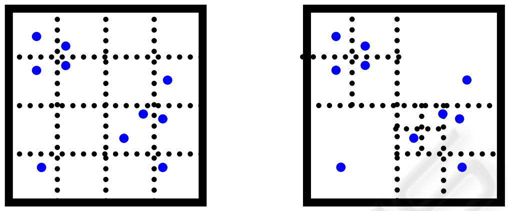

Figure 2: Division of an area into (a) regular regions with the same size each, and (b) hierarchically defined regions based on a quad-tree division with different sizes but containing approximately the same number of objects.

In cellular radio networks such regions appear naturally due to the limited range of radio communication. A mobile user thus has connections to only a few cells that determine the user's location area. Location management here means paging and location update (Chew, Yeo & Kuan, 2007).

However, in network-centric operations the users are connected via different communication channels which include radio networks, satellite connections and fibre lines. In the global information grid, where all objects are connected to, is the geographic proximity in general not directly visible. Even if two users are in direct neighbourhood of each other, they may use different kinds of radio communication devices due their affiliation to different organisations or nations and hence do communicate not directly with each other via radio but via their corresponding home organisations.

Moreover, the location of a user is actually not a single position but potentially a larger area of its effect range. We can thus do not directly use the concepts used in (cellular) radio networks for general command and control information systems. We hence use directly the spatial position information of an object (like its GPS position).

Figure 2-(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. (This computation of the affiliated region to a certain position corresponds to the *paging* process in cellular radio networks.)

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 regions 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. This implies that the number of objects located within each regular region may vary significantly: While some regions may be (almost) empty, other regions may contain a large amount of objects. This not only affects the efficiency of the algorithm described before but may even prevent the required real-time behaviour of the command and control information system.

A similar problem arises in massively multiplayer online real-time games. Here a zoning approach is used to distribute the server load onto parallel servers, each of them being responsible for a different partition, i.e. zone, of the game world. However, while this zoning concept scales with the number of players and the size of the users, it does not scale with the density of players within a certain area. If the object density within one region is too large, the server does not respond in real-time anymore. (Müller & Gorlatch, 2006) proposes the replication of heavily populated zones onto different game servers on the grid. Here each server contains the whole game state, but only processes the so-called *active entities* as part of the global state and afterwards broadcasts its local computation result.

While in games such a bad responsiveness is at most annoying, in real-life like in network-centric operations the inability to locate the different objects in time may actually result in real casualties.

The region services in our approach are basically storing data (the local objects in that region) and are not computing-services. (The actual computation is done in the COP-services. Note that the mapping of service-instances to actual servers is outside of the scope of this paper.) The replication thus does not make sense here. However, the idea of parallelizing the load onto different (sub-)servers can be adopted if we divide a region (with its object set) into sub-regions (with its corresponding sub-sets of associated objects).

An adaptive approach to the definition of regions therefore has to take the actual distribution of objects into account. Organizing region areas hierarchically as a quad- or oct-tree (in case of two- or three-dimensional coordinates, see Figure 2-(b)) 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 area, i.e. the AOI covers only a fixed number of 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 for regular regions.

The concept of region services not only improves the efficiency for generating a static operational picture but can also handle dynamic changes of object positions. In general all user objects are mobile. Thus the objects within and near the AOI of a user 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 does not work, but we have to compute and update the AOI dynamically: If an object is located in the area of one of the regions that overlaps with our AOI, we can subscribe to that object for position updates. If that object finally leaves the area of that region 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 regions 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.

4.2 Location Management using Region Services

The actual location management algorithm for computing the areas of interest with the support of region services is described here. See also Figures 3 and 4 for an illustration.

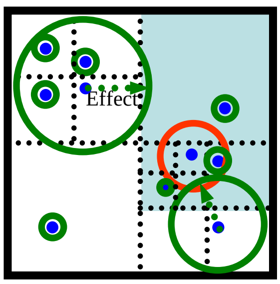

Figure 3: Computing the AOI with effect ranges using region services. The resulting area is shown in Figure 1.

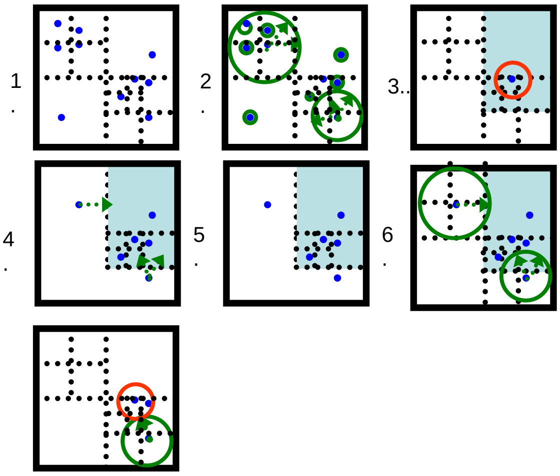

Figure 4: Phases of the location management process.

Phases 1-2 (see Figure 4) set the pre-conditions by initializing the region services. Phases 3-6 fetch the location information of the objects in the user's surrounding of the current user. Finally in phase 7 the AOI is computed locally. Phases 2 and 5 can be omitted for the location management without effect ranges.

1 *Register objects at corresponding region services of current position:* First all objects have to register themselves at their corresponding region service, i.e. they publish themselves.

   In case of regular regions it 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 we ensure that an object is registered at the new region service before it de-registers itself in the old region.

2 *Register objects at region services of current effect range:* Then all objects additionally have to register at the corresponding region service of 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 relevant region services for the AOI*: To compute the AOI of a user, we have to determine the relevant region services for this user. This includes all those regions that intersect with the shape of the AOI.

4 *Get object sets from the relevant regions for the AOI*: Then we read all objects from the relevant region services determined in phase 3 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.

   In addition to the objects located within certain regions here we also get objects that might have an effect within that region.

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

   For example, in Figure 3 the object in the lower right part of the figure is – due to its large effect range – registered at two of the relevant regions (shown in light blue, c.f. Figure 4-4.).

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

7 *Determine objects within the AOI:* Finally we check which of the objects actually belong to the AOI and which does not. Those objects where their position or their effect range intersects with the shape of the AOI are to be displayed, while the others are filtered out.

   Note that if one of the objects received in phase 4 changes its position we have to check again if it has moved into or out of the AOI. However, since from phase 4 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 3.

4.3 Dynamic Location Updates

The presented algorithm works well for a static setting where the objects do not move. Here we discuss that is also appropriate for the dynamic case.

If the user is moving, the shape of his AOI may intersect with other regions than before. In this case we have to update the subscription relations to the corresponding region services. This will provide us with the objects within these regions such that we can update the AOI.

If, on the other hand, another object (that may be monitored by the user's AOI) is moving in such a way that it leaves the area of its current region, it has to register itself at the new region-service (and de-register at the old one). This region service will automatically inform the user about the change within the object set such that the AOI can be adapted locally.

Moreover, the object registration at the new region service will increase the number of objects handled by it, if not another objects leaves the region at the same time. In the hierarchical region model based on quad- or oct-trees this would mean that we had to split the region into sub-regions, if a certain maximal number of objects had been reached. (Note that for efficiency reasons we do not split immediately but allow more than one object within each region.) If a region is split, we create new region-services for the sub-regions, while the old region-service takes the role of the parent service.

However, one problem still remains with the hierarchical regions. In order to check which region service a specific location belongs to, we have to retrieve the corresponding region service by traversing the region-tree, starting from the root region which denotes the whole area. This leads to *O(log R)* messages which is not appropriate for small bandwidth radio networks, and may put high loads onto the root region server.

For this reason we combine regular and hierarchical regions: The regular regions as in Figure 2-(a) serve as base regions, since here the object location mapping can be done in a single step. Only if there are too many objects within that region due to an in-homogenous distribution we further sub-divide that region in a hierarchical manner. The region service for the corresponding hierarchical base region serves as the root for the sub-tree of regions, which finally leads to a forest of quad- or oct-trees.

The performance of the different location management schemes is evaluated in (Nitsche, 2007).

5 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 either within that specific area or that may have an effect on that area.

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 or hierarchically based on quad- or oct-trees.

To handle in-homogenous distributions of objects we propose a combination of regular and hierarchical regions: The regular base regions are used to directly derive the basic region service responsible for the corresponding area, thus avoiding the communication overhead necessary if we had to read the data-dependent region-distribution from a root region service (and its sub-services). The concept of hierarchical regions is used in case of larger densities of objects within a region. In the latter case we split the region into sub-regions which are managed by separate services.

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 but can also manage the dynamic changes of object locations.

The proposed location management scheme based on region services can not only be used in command and control information systems but in all distributed, service-oriented systems with large amounts of mobile users.

REFERENCES

Alberts, D.S., Hayes, R.E (2003). *Power to the Edge: Command and Control in the Information Age*. CCRP Publication Series, Washington, DC, USA.

Alberts, D.S., Hayes, R.E (2006). *Understanding Command and Control*. CCRP Publication Series.

Anthony, R.W. (1999). Relating Large and Small in C2 and Operations. In *Command and Control Research and Technology Symposium (CCRTS)*, Newport, RI.

Arnold, J., Levine, B., Mammatha, R., Lee, F., Shenoy, P., et al. (2004). Information-Sharing in Out-of-Hospital

Disaster Response: The Future Role of Information Technology. *Prehospital and Desaster Medicine,* 19(2): 201-207.

Bachran, T., Bongartz, H. H.-J. & Tiderko, A. (2005). A Framework for Multicast and Quality based Forwarding in MANETs. In *Proc. 3rd IASTED International Conference on Communications and Computer Networks (CCN'05),* ACTA Press.

Blais, L., Goerger, N.C., Richmond, P., Gates, B. & Willis, J.B. (2005). Global Information Grid Services and Generation of the Mobility Common Operational Picture. In *2005 Fall Simulation Interoperability Workshop (SIW)*, Orlando, Florida.

Barnes, J.E., Hut, P. (1986). A hierarchical *O(N log N)* force calculation algorithm. *Nature,* 324(4), 446-449.

Carzaniga, A., Rosenblum, D.S., & Wolf, A.L. (2001). Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems*, 19(3): 332-383.

Carzaniga, A., Rutherford, M.J. & Wolf, A.L. (2004). A Routing Scheme for Content-Based Networking. In *Proc. IEEE INFOCOM 2004*, Hong Kong, China.

Chew, Y.H., Yeo, B.S., Kuan, D.C.M. (2007). Sensitivity study of location management area partitioning in cellular communication systems. *Computer Networks*, 51(6): 1555-1573.

Denning, P.J. (2006a). Hastily formed networks. *Communications of the ACM,* 49(4), 15-20.

Denning, P.J. (2006b). Infoglut. *CACM*, 49(7): 15-19.

Foster, I., Kesselmann, C., editors (1998). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

Hayes-Roth, F. (2006). Two Theories of Process Design for Information Superiority: Smart Pull vs. Smart Push. In *Command and Control Research and Technology Symposium (CCRTS),* San Diego, CA.

Hightower, J., Borriello, G. (2001). Location Systems for Ubiquitous Computing. *IEEE Computer*, 57-66.

Jungert, E., Hallberg, N., Hundstad, A. (2006). A Service-Based Command and Control Systems Architecture for Crises Management. *International Journal of Emergency Management,* 3 (2-3): 131-148.

Kruse, J., Adkins, M. & Holloman, K.A. (2005). Network Centric Warfare in the U.S. Navy's Fifth Fleet. In *Proc. 38rd Hawaii International Conference on Systems Sciences (HICSS),* IEEE Computer Society.

Käthner, S., Spielmann, M. (2004). A Base Component for Network-Based Service-Oriented C4ISR Systems. In *Coalition C4ISR Architectures and Information Exchange Capabilities*, The Hague, The Netherlands.

Lisper, B., Holerin, J. (2002). Development and Verification of Parallel Algorithms in the Data Field Model, In *Constructive Methods for Parallel Programming,* 127-141, Nova Science Publishers.

McEneaney, W.M., Singh, R. (2007). A Computationally-Feasible Algorithm for Estimation of Opponent Strength in Urban Combat. In *12th International Command and Control Research and Technology Symposium (ICCRTS)*, Newport, RI, USA.

Mitschke, T., Peter, H. (2001). *Handbuch für Schnell-Einsatz-Gruppen*, 3rd ed. S+K Verlag, (in German).

Mittu, R., Segaria, F. (2000). Common Operational Picture (COP) and Common Tactical Picture (CTP) via a Consistent Networked Information Stream (CNIS). In *Command and Control Information Systems,* Monterey, CA.

Müller, J., Gorlatch, S. (2006). Scaling Online Games on the Grid. In *Proc. 4th International Game design and Technology Workshop and Conference (GDTW 2006).*

Nitsche, T. (2006). Information Access in Tactical Command and Control Information Systems. In *11th International Command and Control Research and Technology Symposium (ICCRTS)*, Cambridge, UK.

Nitsche, T. (2007). Evaluating Region-Based Location Management Schemes for Service-Oriented Command and Control Information Systems. Subm. to *Mil. Communication and Information Systems Conference (MCC 2007).*

Satoh, I. (2006). Location-based services in ubiquitous computing environments. *International Journal on Digital Libraries*, 6(3), 280-291, Springer.

Schade, U., Hieb, M.R. (2006). 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),* Cambridge, UK.

Scott, S.D., Wan, J., Rico, A., Furusho, C., Cummings, M.L. (2007). Aiding Team Support in Command and Control Operations with Large-Screen Displays. In *Proc. of HSIS 2007: ASNE Human Systems Integration Symposium*, Annapolis, MD, USA.

Sebé, F., Domingo-Ferrer, J. (2007). Scalability and security in biased many-to-one communication. *Computer Networks*, 51(1): 1-13.

Shin, S.Y., Park, S.-H., Jung, B.H., Kim, C. (2006). Dynamic Location Management Scheme Using Agent in a Ubiquitous IP-Based Network. In *Management of Convergence Networks and Services, 9th Asia-Pacific Network Operations and Management Symposium (APNOMS 2006),* LNCS 4238, 491-500, Springer.

U.S. Government Accountability Office (2004). *Defense Acquisition - The Global Information Grid and Challenges Facing Its Implementation.* Report GAO-04-858.

Wilson, C. (2004). Network Centric Warfare: Background and Oversight Issues for Congress. CRS Report for Congress RL32411.