11th ICCRTS
Coalition Command and Control in the Networked Era

Agent Coordination Mechanisms for Multi-National Network Enabled Capabilities

(C2 Architecture, C2 Experimentation, C2 Modeling and Simulation)

P.P.A. Storms and T.J. Grant

P.P.A. Storms
Y'All B.V.
Grotestraat 182
5141 HD Waalwijk
The Netherlands
phone: +31 (0)416340019
email: patrick@yall.nl

T.J. Grant[1]
NLDA
P.O. Box 90002
4800 PA Breda
The Netherlands
phone: +31 (0)765273261
email: tj.grant@nlda.nl

Point of contact: T.J. Grant

[1]Also with Department of Computer Science, University of Pretoria, South Africa

Abstract

Modern advanced information technology enables military organisations to share information, such that decision making occurs at all levels within the chain of command. In a network centric approach to warfare, assets like sensors, shooters and C2 systems are interconnected in an infostructure, or information grid. By sharing information and combining capabilities, these assets work together to achieve enhanced capabilities. In the Netherlands, this is termed Network Enabled Capabilities (NEC). Under NEC, assets can dynamically form temporary "teams" to fulfill a specific task. To realize such agile configurations of assets, the problem of managing tasks between assets has to be tackled.

In this work we argue that specialized software components, called intelligent agents, are suitable for coordinating tasks between assets. NEC systems can be viewed as a type of multiagent systems (MAS) in which agents represent assets and connect them to the information grid. Now, coordination mechanisms, as explored in the field of MAS design, are also applicable in a NEC setting.

The aim of the research reported in this paper is to identify which coordination strategies are suited to NEC. We take the following approach. First, we give an overview and classification of agent based coordination mechanisms and their properties. We suggest a taxonomy of agent coordination strategies. Next, we identify the key requirements for coordination mechanisms in a NEC setting. Based on these requirements we argue that explicit, centralized coordination strategies with low communication overhead in a cooperative environment are best suited as primary coordination strategy. Finally, we compare our research with other initiatives, employing agent technology.

We recognize that there is no single best way to coordinate, and that for local clusters of agents other types of coordination strategies might be preferable. Therefore we suggest a hybrid NEC coordination strategy, composed of a global primary coordination strategy, with subordinate clusters of agents that use local coordination strategies. This can be a mechanism for handling multi-national coalitions. In the proposed architecture, agents are organized in a nested structure of clusters, or holons.

1 Introduction

1.1 Network Enabled Capabilities

Post-Cold War conflicts predominantly involve a wide range of global and regional actors (major powers, international agencies, neighbouring states, terrorist groups and criminal networks) and modern warfare has to cope with asymmetric threats and networks of decentralised, loosely coordinated, fighting groups. Moreover missions are no longer a single-nation effort, but are executed in Joint force, in conjunction with Allied and coalition partners. This imposes new requirements to militairy operations such as increasing the tempo of operation, dealing with changing goals during mission execution, being interoperable with own and coalition forces, efficient deployment and coordination of assets, and sharing information with all elements to achieve a high level of shared situational awareness [1].

Modern rapidly advancing information technology enables military organisations to share information, such that decisionmaking occurs at all levels within the chain of command [2]. Technological changes give organisations the opportunity to take full advantage of all available information and to bring all available assets to bear in a rapid and flexible manner [3]. Thanks to the web structure, multiple redundant paths for information sharing are possible. This is exactly what drives the current transformation of hierarchical, platform centric organisations to agile, network centric organisations. In the next subsection we will take a closer look at visions on military networked organisations.

Network Centric Warfare (NCW) is a military concept in which information superiority enables operations to generate increased combat power. By networking sensors, decision makers and shooters, NCW aims at increasing shared awareness, speed of combat, tempo of operations, lethality, survivability and the level of self synchronisation. In essence, NCW translates information superiority into combat power by effectively linking knowledgeable entities in the battle space [4]. NCW is a formal US networking concept and doctrine that seeks to develop into a fully-fledged warfighting capability. The tenets of NCW are:

1. A robustly networked force improves information sharing.
2. Information sharing enhances the quality of information and shared situational awareness.
3. Shared situational awareness [1] enables collaboration and self-synchronization, and enhances sustainability and speed of command.
4. These, in turn, dramatically increase mission effectiveness.

By sharing information, NCW aims at achieving a heightened state of shared situational awareness and knowledge among all elements. Consequently an increased effectiveness is reached with the same number of assets, enabling a webbed network of multiple military organisations to counter modern threats.

The Dutch and UK Ministries of Defence use the term Network Enabled Capability (NEC) to denote evolving capability by bringing together sensors to gather information, a command and control (C2) network to fuse, communicate and exploit the information, and strike assets to act rapidly to deliver the required effect. All available assets pool their information by networking, in order to achieve enhanced capabilities [5]. NEC shares the tenets of NCW, but is more limited in scope. It is not a doctrine, but rather a conceptual and technical framework for gradually implementing NCW theory to actual enhanced capabilities. Realization of NEC is a process of change and evolution, starting with automating and digitising current operational (decision) processes and integrating previously dispersed systems. Throughout this paper we will use the term NEC.

1.2 Assets connected in information grids

NEC theory states that available assets can only be fully exploited if there is a high degree of decomposition and logical decoupling. For instance, if a sensor (e.g. tracking radar) is logically decoupled from a weapon system (e.g. close-in weapon system) both the weapon and the sensor can be controlled independently and can be shared with other assets and deployed more efficiently. Yet the sensor remains physically mounted on top of the weapon system. Now, one can think of a tracking radar supplying range information to complement angle-only tracks from an infra-red sensor. Also the

weapon system might acquire a target using data from other sensors than its own, e.g. in case of a faulty sensor, to surprise the enemy, or if more accurate target data is already available.

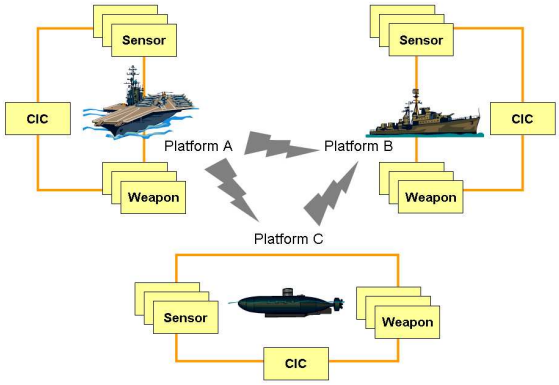


Figure 1: Cooperation at platform level

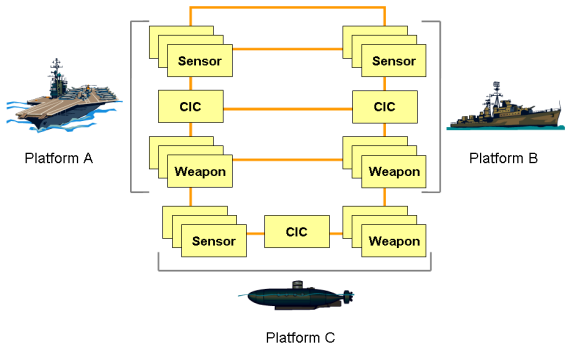


Figure 2: Sharing information and control of resources in a NEC grid

Figures 1 and 2 illustrate the difference between a platform centric and a network centric approach, respectively. In a platform centric approach each platform (in this case an aircraft carrier, a frigate and a submarine) is an indivisible unit in a mission. Although the platforms can exchange tactical information using datalinks, their capabilities are limited to the assets that are collocated on the platform itself. The assets are typically sensors, C2 nodes (such as the Command Information Center) and weapons. Each platform has its own decision cycle (e.g. surveillance, target detection, threat assesment, sensor-weapon assignment, target aquisition and interception) and mainly has to use its own resources for execution.

In a NEC setting, the available assets are logically decoupled from the platforms and organised as nodes in an *information grid*. The grid enables information sharing and control of assets between all nodes. Consequently assets can be used independently of the platform, yielding new capabilities such as *integrated fire control* (IFC) [6]. IFC combines sensors, C2 nodes and weapons of different platforms to enable collaborative engagements. An example of an IFC capability is *Engage On Remote*, as illustrated in Figure 3: one platform tracks a threat and uploads fire control data to a firing unit, which is responsible for interceptor guidance. Using the same assets, the effective range of weapons is extended. Currently, US Navy developed IFC capabilities for anti air warfare within the Cooperative Engagement Capability (CEC) programme.

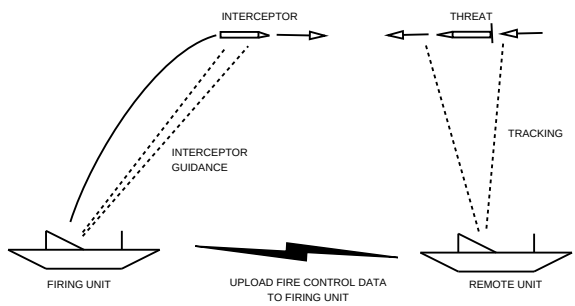


Figure 3: Integrated Fire Control Engage On Remote Capability

Another example of decoupling is the Unmanned Airborne Vehicle (UAV), a remotely controlled reconnaissance and combat airplane. Here the pilot is physically decoupled from the aircraft. Since the pilot is no longer required to be actually present at the battlefield and the UAV is semi-autonomous, the pilot becomes an asset that can be shared. From a remote control center he is able to operate multiple UAVs simultaneously. The UAV pilot can be assigned to multiple missions, in different organisational structures.

1.3 Agents, Multi Agent Systems and Agent Coordination

To realise NEC, existing assets need to be connected to the infostructure, and their information, services or capabilities need to be accessible for other assets. To achieve this a number of problems have to be tackled. Amongst others, these problems concern interoperability, security and coordination of tasks between assets. In this work we envision that specialized software components, called *intelligent agents*, are suitable to fulfill this task.

Several different perspectives, property sets and classifications for intelligent agents are described in the literature [7, 8, 9, 10]. There is a consensus that agents are automated entities (machines, software processes) that have some degree of intelligence and autonomy [11] in pursuing a set of goals. Agents are able to perceive their environment and respond to changes in a timely fashion by acting upon their environment. Action can be both reactive and proactive. Finally, agents should be able to interact with other agents, humans and non-agent systems to offer their services, take action on behalf of them (*agency*), and cooperate to achieve a set of goals. These goals can be common or conflicting.

The term *multiagent system* (MAS) denotes a network of agents. Multiagent systems are typically (large scale) distributed systems, comprised of multiple individuals or services and engaged in more than one task. Furthermore multiagent systems have: goal-directed behaviour (where goals can change), the ability to affect and be affected by the environment, legal standing and the presence of knowledge, culture, memories, history and capabilities distinct from any single agent [12].

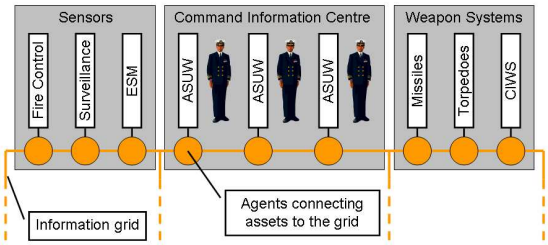


Figure 4: Agents connecting assets to the grid

Aspect	NEC	MAS
assets	sensors, shooters, control systems, military units	agents, humans, non-agent systems
network	information grid sensor grid engagement grid	agent platform
services	information sharing, situation awareness, cooperative engagement	problem solving
goal	execute mission, achieve effect	perform tasks, maximise utility

Table 1: Relation between NEC and MAS

NEC systems can be viewed as multiagent systems in which agents represent assets and connect them to the information grid. Figure 4 shows, in a simplified example, how naval assets are represented by agents. The agents connect the assets to the grid, and assets can only be accessed through their corresponding agent. Thereby, all assets are shielded from each other's specific technical and behavioural characterists. Furthermore, the agents can guard availability of the assets and access policies. Finally, the agent can take care of dynamically forming temporary "teams" of assets to fulfill a specific task.

Table 1.3 further illustrates the relation between NEC and MAS. The primary assets in a MAS are agents which can represent other assets such as humans or non-agent systems. An *agent platform* takes care of transporting messages between agents, and provides additional networking services such as agent lifecycle control, addressing and lookup facilities. In a NEC context the assets are sensors, shooters, control systems and military units, connected in a grid. A MAS provides problem solving services, in order to perform a set of tasks or to optimise some utility function. The services and goals strongly depend on the application domain. In the NEC application domain, the services include information sharing, situation awareness and cooperative engagements. The goals of the network are: executing a mission, or achieving a desired effect.

The increased gain in agility offered by NEC comes at a price. Dynamically organising assets yields substantial coordination efforts, which are not required in case of static, pre-defined organisa-

tional structures. To realise NEC systems we need to introduce coordination mechanisms that respect the specific requirements of the military domain in general, and of network centric operations in particular. In the field of MAS design, coordination has been studied extensively [13, 14, 15, 16, 17]. Theory and concepts stemming from this research can be applied to tackle NEC related coordination problems.

1.4 Problem statement and approach

As we pointed out in section 1, the gain in agility caused by a network centric approach to warfare comes at the cost of increased effort for establishing coordination. We also argued that coordination mechanisms as explored in the field of MAS design seem promising for a NEC setting, because of the similar characteristics of NEC and multiagent systems.

The aim of this work is to identify coordination strategies suited to NEC. We take the following approach. First, we will give an overview and classification of agent based coordination mechanisms and their properties in section 2. Next, in section 3 we zoom in on specific coordination issues in NEC, and identify the chief requirements to coordination mechanisms in a NEC environment. We evaluate which types of agent based coordination methods, as identified in section 2, meet the NEC requirements. Based on this result we propose a NEC coordination architecture. Section 4 discusses the results, related work and future work.

2 Coordination in Multiagent Systems

2.1 What is coordination?

Agent coordination is concerned with the control of the activities and the regulation of object flows from an operational perspective within a MAS. Here, *object* denotes an artifact that can be produced, consumed or tranformed by agents. For instance, an object can be data, information, knowledge or a physical entity. Objects can have various properties, i.e. they can be discrete (e.g. messages) or continuous (e.g. energy), divisable or indivisable, sharable (e.g. can be accessed by multiple agents simultaneously) or exclusive, and static or dynamic (i.e. the properties of an object change in time). With *activity* we mean the act of consuming, transforming and distributing objects. Each such sub-activity is called a *job*, i.e. *consumer job*, *transformation job* and *distribution job* respectively. Activities are part of *primitive tasks*, which are at the lowest level of a task decomposition hierarchy.

A common sense definition for coordination is: *Coordination is the act of working together harmoniously* [13]. If we apply this definition to MAS, the act of "working together" implies that agents perform activities and that interdependencies exist between these activities, such as consumer / producer relations. The predicate "harmoniously" implies that either no conflicts exist, or conflicts are resolved. For establishing a conflict-free system some kind of management process is required [13, 17].

As pointed out by Corkill and Lander [15] coordination is an essential activity in multiagent systems, since it permits agents to perform complex composite tasks and achieve (common) goals by means of interaction. Note that, considering the definitions of object, activity, job and task, the act of task decomposition is outside the scope of coordination. We assume that in a MAS tasks are already decomposed to a level of granularity that fits the capabilities of individual agents, and the interdependencies between the sub-tasks (e.g. the order of execution) are explicitly known. So, if agents communicate on coordination this communication will only involve objects, activities and jobs.

Agents can be organised in an authority structure. For instance, in a hierarchical organisation agents with managerial responsibilities delegate tasks to subordinate agents. Within an organisation different control regimes (coordination) can exist. In general, an organisation sets general norms to agent behaviour, coordination patterns and authority. These norms are respected by the members of the organisation for a potentially long term. Coordination takes place within the "rules of the game" of an organisation, and is concerned with the relatively short term management of specific activities. We must therefore always consider coordination strategies in the context of an organisation [18]. For an overview of organisational paradigms

in multiagent systems, see Horling et al. [19].

2.2 Communication on coordination

In a MAS, part of the interaction between agents will involve communication about coordination. For instance, agent A delegates an activity to agent B and requires a notification when B has finished. If A does not receive a notification within x seconds, A will assume that B has failed to complete the activity. Apart from a description of the activity itself, A has to inform B about the procedure. This implies that agents need to be interoperable at the level of coordination.

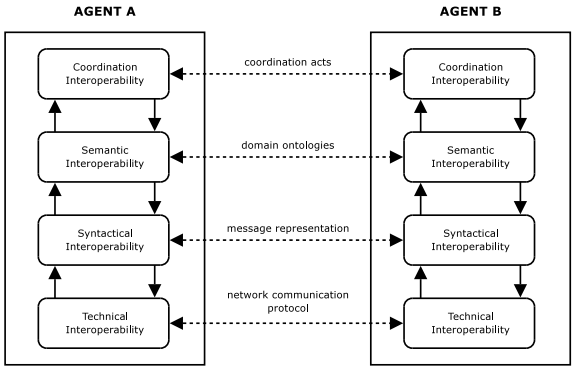


Figure 5: Levels of interoperability

In a model for agent interoperability [17], that is similar to the ISO-OSI Network Model [20], coordination is positioned at the top layer. See Figure 5. The lower-level layers are concerned with semantic (i.e. the meaning of messages), syntactical (i.e. the structure of messages) and technical (i.e. the physical distribution of messages) interoperability.

Communication on coordination can concern different dependency relations between activities. Ossowski [21] identified the following dependency relations between activities:

1. Producer-consumer dependencies. An agent may produce one or more objects that are to be used by other agents. Sometimes transportations (i.e. transactions within the flow of objects) have to be performed to move objects between agents. There are different ways to manage this dependency. A producing agent may either be reactive (i.e. wait for consumer demand) or pro-active (i.e. produce objects in advance). In both cases the producing agent can either inform consumer agents about the presence of an object (event notification), or directly distribute the objects.

2. Shared resource dependencies. Multiple agents may need mutually exclusive access to a single object. Resource allocation actions can manage this dependency.

3. Simultaneity dependencies. Multiple agents may need to perform activities that cannot be performed at the same time. Synchronisation actions restrict the periods in which activities can occur.

4. Task dependencies. A group of activities may accomplish tasks that jointly attain come overall goal. Selection, delegation, intervention and aggregation actions must respectively ensure that activities are selected in the right order, responsibility for performing activities is delegated to available and competent agents, responsibility is taken back in case of failure, and results are aggregated to complete the task.

These dependency relations can be seen as coordination constructs of a MAS. Each relation defines *what* agents talk about when communicating on coordination. *How* coordination is realized, is defined by a *coordination strategy*. These are discussed in the following section.

2.3 Taxonomy of agent based coordination strategies

Agent coordination strategies are designed to enable a group of agents to work together in some way to complete a single task or well defined set of tasks (also called a problem-solving episode). In this section we give an overview of various types of coordination strategies. Based on the classification criteria described in literature [14, 21, 18], we distinguish the following dimensions to classify coordination strategies:

1. Implicit versus explicit coordination. See 2.3.1.

2. Dynamic versus static coordination. See 2.3.2.

3. Coordination strategies for cooperative versus competitive agent systems. See 2.3.3.

4. Centralised versus decentralised coordination. See 2.3.4.

In addition to these dimensions, we will consider the coordination strategy's design metaphor as a discriminating property. This is discussed in section 2.4.

2.3.1 Implicit and explicit coordination

For implicit (communication-less) coordination strategies there is no explicit inter-agent communication related to coordination. Either agreements on coordination are shared by all agents in a MAS, or agents operate under local sensing and control. In the latter case system-level behaviour arises from interaction between individual agents through their environment. An example of such a mechanism is stigmergy [22, 23, 24, 25].

With explicit (communication-based) coordination strategies, agents explicitly communicate information related to coordination. Agents make use of a coordination strategy, which can be seen as a decision-making and communication pattern among a set of agents that perform activities to coordinate task execution [26]. Approaches to explicit coordination are many and various. Among others we distinguish market-based, negotiation-based and organisation-based approaches [27, 28, 29].

As observed by Jones et al. [30] the distinction between implicit and explicit coordination is not crisp. It is rather a continuous spectrum since (1) there will always be some coordination-related communication and (2) there will always be some kind of shared model on coordination.

2.3.2 Dynamic versus static coordination

Dynamic coordination strategies allow agents to alter their coordination strategy at runtime. This can be done by either fine tuning the configuration of a specific coordination strategy, or replacing the current coordination strategy by another.

With static coordination the coordination strategy for a MAS is determined and configured *a priori*, e.g. at design time.

2.3.3 Coordination in cooperative and competitive agent systems

A cooperative MAS is an association of agents that join together to carry out an activity of mutual benefit. This sets specific requirements on the coordination strategy, since agents need to share their resources and to orchestrate their activities such that the group can perform certain tasks better than a single individual agent. The coordination strategy has to support the agents to maximise some global utility. Individual preferences or goals are of secondary priority.

In a competitive MAS agents are self-interested and will primarily pursue their individual goals. The overall performance of the group is of secondary interest. The goal of a coordination strategy in a competitive environment is to "persuade" agents to cooperate in performing a task, by satisfying individual goals or preferences. In a competitive MAS agents may have conflicting goals and might not be willing to share all information.

Typical techniques used in cooperative agent systems are *blackboards*, *Contract Net* (CNET) and *distributed constraint satisfaction* [31, 32, 33]. Blackboard systems allow agents to contribute to a common goal by sharing information and partial solutions with other agents. CNET is a well-known mechanism for task allocation, based on bidding and contracting [34]. Techniques that stem from the field of distributed constraint satisfaction are suitable for determining a task allocation, without relying on a central contractor (as is the case with CNET).

Note that these techniques are only suitable in a cooperative environment. The use of a blackboard architectures implies that agents are willing to share information. Using CNET and distributed constraint satisfaction for task allocation requires agents to be *honest* (i.e. they do not provide false information on availability and suitability) and *benevolent* (i.e. an agent does not benefit from accepting a task).

In a competitive MAS agents will typically require some kind of "reward" to perform a certain task, will value the resources assigned to them, or are willing to "pay" other agents for providing a service. Typical coordination strategies for competitive agent systems are:

- Market mechanisms. If in a MAS the main

relation to be managed between agents is *consumer-producer* or *producer-consumer* (see 2.2) then coordination can modeled as an artificial economy in which agents trade objects in exchange for some utility. The value of objects is determined by economic rules of supply and demand. An example of a well-known market mechanism in agent systems is the *auction*, such as the Dutch, English, first-price or Vickrey auction [27, 35].

- Negotiation mechanisms. If a MAS mainly has to deal with resolving conflicting goals then negotiation mechanisms are suitable coordination strategies. Whereas CNET is applicable in a cooperative setting, negotiation mechanisms aim at maximising an *egalitarian social welfare function* or *Nash product* seem more applicable in a competitive setting. The general idea is that agents negotiate on objects (tasks, resources) such that the utility of the least-satisfied agent is maximised. For background reading see [28].

The differences between these two strategies are subtle. Typically, market mechanisms are modelled as auctions where third-party agents (auctioneers) mediate between producers and consumers. In case of negotiation, agents make (a sequence of) bi-lateral trades, without intervention of a third-party agent. Since both strategies involve the non-benevolent exchange of objects to establish an allocation of objects to agents, negotation can be seen as a market mechanism. Vice versa, an auction can be seen as a form of mediated negotiation. For the sake of clarity, we will use the term *market-based coordination* to denote both strategies.

Note that for both cooperative and competitive agent systems, overall goals are eventually reached. Even though in a competitive MAS agents may have conflicting goals, conflict can be resolved by negotiation and trade. In this work we do not consider *malicious* agents, i.e. agents aimed at deliberately corrupting the functionality of a MAS. In the context of NEC, we assume that all agents (both cooperative and competitive) in a military coalition's grid are somehow willing to contribute to successful execution of a mission. Still, the grid itself should be protected against possible intruding malicious agents of the opposing force. Examples of malicious agents are viruses, spyware and computer worms.

2.3.4 Centralised and de-centralised coordination

In a MAS where coordination is centralised, a separate set of (computational) assets can be distinguished that is solely occupied with handling coordination. All the other agents in the MAS have no capabilities to coordinate, other than informing the central coordination mechanism about their state and obeying its instructions. Note that centralised coordination does not necessarily mean that coordination is performed by a single central system. It can also be a distributed system that is functionally separated from the MAS that is under control.

In a MAS with decentralised coordination each agent has the capability to coordinate, as well as their functional (problem solving) capabilities.

The distinction between centralised and decentralised coordination is not crisp. For instance, if an agent in a MAS with decentralised coordination delegates a set of activities to a number of subordinate agents, then this agent will temporarily act as a centralised coordinator for a subset of the MAS.

2.4 Agent coordination design metaphor

A metaphor is a comparison which imaginatively identifies one concept with another dissimilar concept, and transfers or ascribes to the first some of the qualities of the second. Metaphors are especially powerful when used to help understand a concept that is unfamiliar or unapproachable. In software engineering, metaphors are an instrument to conceptualise the structures and designs of information systems, and to communicate their meaning. Applied to this work, we can distinguish agent coordination by their underlying design metaphor. The number of design metaphors that can be applied to agent coordination is virtually unlimited. In this section we highlight a few commonly used metaphors.

2.4.1 Organisational metaphor

A commonly applied metaphor for realizing coordination in a MAS is the organisational metaphor, where multiagent systems are modelled as human organisations. An organisational structure consists of stakeholders (i.e. individuals, groups, physical or

social systems) that coordinate and interact with each other to achieve common goals. Hence organisations can be seen as a type of coordination [14].

Organization theorists have proposed patterns such as the structure-in-5 [36], the matrix, the chain of value and the like to define organizational structures and behaviours. Within a human organisation individuals or units (groups of individuals) act according to a set of (social) rules and hold a *position* or *role*, that is associated with certain responsibilities and skills. For instance, in an organisation one may distinguish units like planning, sales, marketing, procurement, product design, production, customer support, etc. Within these units one may find (operational) roles that are specific for that unit (like planners, designers, assemblers) or more generic roles that can also be found in other units (like management, administration). Agreements are made on how these units and positions operate, how they are controlled and to whom they report.

Organisational concepts can be mapped onto the design of multiagent systems [37, 38], where agents are arranged in organisational structures according to function, skill, location, time, client or location. Popular patterns used in the organisational metaphor for agent coordination are derived from Mintzberg's [36] organisational structures [17, 29, 39, 40]. Malone et al. [26] point out that the goal of coordination is to manage interdependencies between positions and activities performed to achieve goals. A classification of dependencies is given by [21] (see also section 2.2). Mintzberg has mechanisms that can be applied to coordinate these dependencies. These mechanisms are:

Direct Supervision This coordination mechanism achieves coordination by having one individual take responsibility, which is taking all decisions for the work of others, issuing instructions to them and monitoring their actions. This mechanism can be seen as a pattern for one central reasoning service (i.e. the *Manager*) with several information providing processes (i.e. *Operators*). This form of coordination is suited when there is a clear distinction between decision-making and operation.

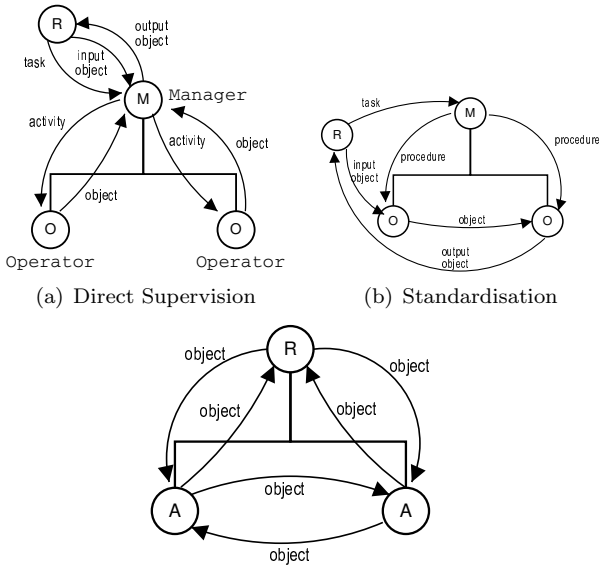
Standardization of Work This mechanism achieves coordination by specifying the content of the technical activity. The content of the activity is specified in every step, from getting the input objects (*consume job*), what to do with it (*transform job*) and to whom to distribute it (*distribute job*). In MAS design this means a (hard coded) procedural program dictates the behavior of an agent, without any room for negotiation with other agents. Conflicts will be reported to the supervisor (e.g. Manager).

Standardization of Output Objects This mechanism achieves coordination by only specifying the result (i.e. the objects produced) of the activities. In MAS design this means standardizing the agent's interfaces (e.g. specifiying output objects) and exchange mechanisms, like languages and ontologies.

Standardization of Skills Here, coordination is achieved by only specifying what competences are needed for the activity. In MAS design the knowledge required for specific activities has to be specified and agents need to be equipped with knowledge about the competences of other agents. By means of protocols, they can exchange knowledge and discover competences of others.

Mutual Adjustment Coordination is achieved by a process of informal communication between positions. This means that positions are capable of solving coordination issues by themselves. In MAS design this means that agents have social abilities in the sense that they are capable of interacting and reasoning about each other's interfaces, knowledge and competences and activities to achieve, with little standardization or protocols.

Figure 6 gives a schematic representation of the coordination mechanisms (borrowed from [17]). The circles represent roles of agents. M stands for Manager, O for Operator and R for requesting agent. Authority structures are represented by straight lines that connect positions. The curved lines represent message flows. The figure representing Standardisation is valid for Standardisation of Work, Output of Objects and Skills. In the case of Mutual Adjustment there are no Manager or Operator roles. Here A denotes an agent capable of solving coordination issues.



(a) Direct Supervision (b) Standardisation

(c) Mutual Adjustment

Figure 6: Three coordination mechanisms.

At the level of coordination the organisational metaphor allows engineers to design multiagent systems as if they were human organisations. Thereby, engineers can make design decisions based on ideas and concepts from the field of organisation theory. Note that the metaphor allows for numerous types of coordination strategies. Based on the Mintzberg's structures, coordination by organisation typically yields explicit coordination strategies since agents directly communicate on coordination.

2.4.2 Market metaphor

Multiagent systems comprised of competitive agents have to deal with non-benevolent, selfish agents that aim to satisfy private goals. The coordination strategy exploits the competitive behaviour to optimise some global utility function, e.g. by treating the MAS as a virtual market place of negotiating agents. In this market *buying agents* may request or place bids for a common set of objects such as services, information or access to resources. *Seller agents*, or third party *auctioneer agents*, are responsible for processing bids and determining the winner [19]. So, allocation of objects to agents is either facilitated by a central auctioneer agent or a by means of a sequence of (bilateral) local negotiations between buyers and sellers.

The arrangement of agents in buyers and sellers, competing for objects, creates a system that excels in coordinating producer-consumer type relationships. It closely mimics real-world market economics. Market-based coordination mechanisms borrow concepts from economics and trade, such as the notion of *auctions*. Various auction-based mechanisms exist, and can vary along a number of dimensions:

- In *private* or *sealed-bid* auctions participants do not see competing bids, as opposed to *public auctions*.
- An auction can either be *single shot*, or iterate over several rounds either as an *ascending* (English) or a *descending* (Dutch) auction.
- In a *reverse auction*, sellers bid rather than buyers.
- If either the buyer or the seller maintains a fixed price, the auction is *one-sided*. If both parties compromise, the auction is *two-sided*.
- In a *combinatorial auction* participants bid on collections of objects, rather than single objects. Here, the value of a combination of objects is not necessarily the sum of individual object values.
- Typically, the highest bidder wins the auction. In a Vickrey auction [35] the highest bidder wins but pays the second highest bid price. This promotes truthful bidding.

An alternative approach for establishing an allocation of objects in a competitive environment is by means of local negotiations, without the intervention of an auctioneer. Typically, this involves a series of bi-lateral exchanges of objects. Each participating agent assigns a value (utility) to the current set of objects it owns, and interacts with a peer to see if an exchange (deal) can be made such that both parties benefit. An agent might offer side-payments to its peer, to compensate for possible loss of utility. Ultimately, the goal is to reach an allocation of objects such that some global utility function is optimised.

Current research focuses on establishing convergence and complexity properties for different kinds of negotiation, for different types of deals, utility functions and data representations [28]. One important result is that any sequence of mutually beneficial deals without side payments will converge to a Pareto optimal[2] allocation [41]. Although no guarantee can be given on the number of deals needed to establish this optimal solution, the overall utility will increase after every deal. In time-critical applications, as found in the military domain, a system can start with an initial sub-optimal allocation (that might be computed by a set of fast heuristics) and improve this result in the remaining computing time.

2.4.3 Biological metaphor

In nature, communities of living entities are found that coordinate their activities in a robust and adaptive way. For instance, many social insects deposit and sense chemicals (*pheromones*) in a shared physical environment, that participates actively in the system's dynamics. The presence and characteristics of pheromone influences the behaviour of individual insects. Although the individual behaviour follows simple rules, the community as a whole will display some kind of complex, intelligent behaviour. This phenomenon is known as *swarm intelligence*. The act of communicating via pheromone markers in the environment is called *stigmergy*.

Computer scientists and software engineers are inspired to model coordination strategies for multiagent systems using swarm mechanisms found in nature. For computer systems Beni et al. [42] define swarm intelligence as a property of systems of non-intelligent robots exhibiting collectively intelligent behaviour. Alternatively the term *emergent intelligence* or *emergent behaviour* is used, to indicate that intelligent behaviour arises from the collective rather than the individual. In multiagent systems simple (mobile) agents correspond to the living entities in a swarm. The agents interact via artificial equivalents of pheromones, i.e. digital markers in the environment. Swarm-based systems have the following qualities [43, 44]:

1. Simplicity. The logic for individual agents is fairly simple. The agents are easier to program and prove correct at the level of individual behaviour. Also, they can run on extremely small platforms, such as "smart dust" micro chips [45, 46].

2. Robustness. Swarm-based systems favor large numbers of entities that are continuously organizing themselves. Therefore, the systems performance is robust against the loss of a few individuals. The simplicity and low expense of each individual agent means that such losses can be tolerated economically.

3. Scalability. A swarm-based system is very scalable from the coordination point of view. Since agents have some form of shared behavioural model about how to react to each other's behaviour, markers (pheromone) in the environment, or external conditions, the strategy requires no direct or extensive communication between the agents. This yields efficient inter-agent communication.

In multiagent systems swarming has been applied to various applications. The agents may either be physical entities in the real world (e.g. vehicles), objects that travel through a computer network (where the nodes of the network can be seen as "nests"), or software processes that represent a state in a computation (e.g. a set of image processing jobs). Some examples of applications are:

Region detection Bourjot et al. [47] used the collective web building strategy of social spiders[3] to develop image processing algorithms for detecting contours and regions.

Network optimisation A model of the foraging behaviour of ants can be used to solve routing problems [48] in networks (e.g. road traffic management). Alternatively, this model can be applied for computer network management applications [49].

UAV control Parunak et al. [50] suggest a stigmergy-based system for controlling a set of UAVs. Each UAV must effectively cover a large search space and revisit locations regularly, maximizing detection probability based

[2] An allocation is Pareto optimal if and only if it is not possible to improve the individual utility of an agent without making any of the others worse off.

[3] *Anelosimus eximius*

on known characteristics of the target (e.g., visibility angle), while not exhibiting any obvious systematic search patterns that would permit mobile targets to execute simple avoidance strategies. Alternatively, Hawthorne et al. [51] suggest a behaviour based swarm algorithm for unmanned vehicle control.

In a broader context, swarming has come into vogue in the military to describe a battlefield tactic that involves decentralized, pulsed attacks [52, 53].

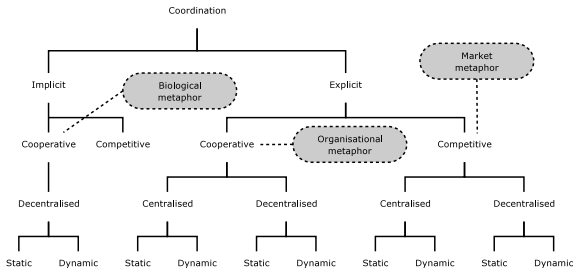


Figure 7: Coordination taxonomy

2.5 Taxonomy overview

Figure 7 shows a diagram of the coordination taxonomy. Although the distinction between the values of some of the dimensions are not crisp, we choose to represent the first four dimensions of our taxonomy in a tree structure. In this tree a "branch" represents the predominant value of a dimension.

Note that in the taxonomy diagram not all combinations of values are exhaustively enumerated. We argue that implicit coordination strategies are primarily applicable in cooperative environments, since competitive environments suggest explicit interaction (e.g. negotiation) between agents about allocations of objects. An exception is reactive behaviour, not driven by explicit rules, to an opponent's actions. Examples of this are the Cold War arms race and dynamics in predator-prey populations.

Furthermore, we argue that there cannot be a centralised coordination strategy without explicit coordination. In a centralised setting, decisions on coordination must somehow be communicated to the agents under control. Although this could be realised with communication-less mechanisms such as markers in the environment (stigmergy), such mechanisms are only applicable for one-way unaddressed messages, rather than the two-way communication that full coordination requires. Hybrid forms are possible, in which there is implicit coordination one way, and explicit coordination the other way.

The fifth dimension, the coordination design metaphor, cannot be placed in such a tree structure, since it is impossible to enumerate all possible metaphors a designer might imagine. Therefore, we position the three metaphors discussed in section 2.4 as "shortcuts" in the tree structure. Coordination strategies that are based on a biological metaphor are usually implicit (e.g. using stigmergy), decentralised and suitable for a cooperative environment. The organisational metaphor typically results in explicit coordination, since agents delegate work, give instructions on how to perform an activity, request for services, report the status of an activity, and so on. All agents obey the authority structures of the organisation, and will benevolently execute the tasks assigned to them. Therefore these coordination strategies are applicable in cooperative multiagent systems. Finally, market-based coordination is a form of explicit coordination because agents negotiate or bid on the allocation of objects. One can also imagine that the coordination could use implicit mechanisms such as stigmergy, e.g. when an agent bids by leaving a marker in the environment. In section 2.4.2 we argued that market-based coordination is applicable in competitive multiagent systems.

3 Coordination in NEC

In this section we identify which coordination strategies are suitable for realising network enabled capabilities. We take the following approach. First we identify the coordination related requirements of NEC, in section 3.1. Then, in section 3.2 we discuss what types of coordination strategies meet these requirements, expressed in terms of the taxonomy defined in section 2.3.

3.1 Requirements

Before we identify requirements for coordination in NEC, we assume that an *infostructure* is available. This infostructure connects all available assets and allows them to share information. As discussed in section 1.3, we envision an infostructure in the form of a grid of interconnected agents, where agents fulfill the role of mediator between the grid and the military assets (sensor, shooters, C2 nodes). Since assets can only be accessed through their corresponding agents, all assets are shielded from each other's specific technical and behavioural characterists. In this way, agents handle interoperability issues between assets. Also the agents coordinate and schedule activities, and guard the availability of the assets. The infostructure is a prerequisite for agile deployment of assets.

As a starting point for identifying coordination related requirements for NEC, we take a look at the NEC Core Themes as defined in the UK Outline Concept for NEC [5]. The Core Themes can be seen as general requirements to NEC. The following three themes are directly related to coordination, since they concern dynamic (re)configuration of assets, cooperation between assets, task assignment, team formation and managing interdependencies between assets and activities.

1. **Flexible Working.** Enabling assets to rapidly reconfigure to meet changing mission needs, allowing them to work together with minimum disruption and confusion.

2. **Agile Mission Groups.** Enabling the dynamic creation and configuration of Mission Groups that share awareness and that coordinate and employ a wide range of systems or a specific mission.

3. **Synchronized Effects.** Achieving overwhelming effects within and between Mission Groups by coordinating the most appropriate assets available in the battle space through dynamic distributed planning and execution.

The fact that three NEC Core Themes involve coordination, proves the importance of coordination within NEC. Together, they yield the most important requirement to coordination in NEC: *flexibility* to reorganise and reconfigure assets in order to meet changing mission needs.

The second requirement can be derived from the fact that network enabled capabilities typically involve many different types of assets. Each of these assets has distinct characteristics, concerning for instance quality of service (e.g. track accuracy of a sensor), computational constraints (e.g. response time), physical constraints (e.g. maximum velocity of a vehicle) and communication constraints (e.g. transmitter bandwidth). This heterogeneity will increase in joint and combined operations. Moreover, NEC does not require replacement of existing assets by new, network enabled assets. Rather, existing assets are incorporated in the network. This means that even within a single-nation force there will be a mixture of legacy and state-of-the art assets. Hence, a coordination strategy cannot assume uniform behaviour of all assets. Being able to deal with *heterogeneity* is the second requirement to coordination in NEC.

In the military domain we must address the issue of *security*. In a networked environment, malicious agents might deliberately corrupt the functionality of a MAS. One can think of Red Force (enemy) agents intruding a Blue Force (friendly) grid, claiming vital resources (comparable to a *denial of service attack*) or corrupting agent interactions such that normal operations are disrupted. A coordination strategy should be resilient to malicious agents. This security requirement should be considered at the level of coordination. We assume that basic security precautions (such as authentication, authorisation and encryption) are taken irrespective of the coordination strategy.

The presence of an infostructure does not imply a high quality *communication* network. The physical communication infrastructure, underlying a NEC grid, is a hybrid one. It can be a mixture of secure phone lines, satellite communications, digital radio, and so on. Between nodes there will be strong differences in network availability, bandwidth and latency. Since sufficient bandwidth or low latency is not guaranteed, a coordination strategy should yield minimum communication overhead at the level of coordination interoperability (see figure 5).

The next requirement is concerned with *robustness*. In general this means that if parts of the networked system fail, the system can proceed with degraded functionality. Specifically, if for some reason assets are not available (e.g. due to damage,

1. Flexibility: ability to readily reorganise and reconfigure assets.
2. Heterogeneity: ability to coordinate heterogeneous assets.
3. Security: Resilience to malicious agents.
4. Communication: resilient to various quality of service levels.
5. Robustness: graceful system degradation or fallback in case of malfunctions.
6. Tempo: no negative effect on tempo of operations.
7. Scalability: ability to handle increased system complexity.

Table 2: Requirements to coordination in NEC

malfunction or failing communications), the coordination strategy should come up with an alternative configuration of assets to meet mission needs. Moreover, if the coordination strategy fails to come up with a suitable configuration at all, the networked system should, in the worst case, degrade to a platform centric force.

One of the merits of NEC is increased tempo of operations because of direct interaction between assets. Coordinating activities will require some computational effort. A requirement to the coordination strategy is that it shall not negatively effect the gain in *tempo*.

The final requirement considered in this work involves *scalability*. Since NEC allows assets to join or part the network, the coordination strategy should be able to handle an increased number of assets without impacting performance.

Table 2 summarizes the requirements to coordination in NEC. In the next section we will evaluate which coordination strategies are suitable for NEC, bearing in mind the requirements identified in this section.

3.2 Evaluation

Considering the coordination taxonomy, as depicted by figure 7, we must first determine whether implicit or explicit coordination mechanisms are applicable, given the requirements for coordination in NEC (table 2). Implicit coordination implies that either all agents have a shared understanding on how to manage activities, or agents interact through their environment. In both cases, there is very little communication between agents on the level of coordination. This satisfies the communication requirement. However, there are some major drawbacks to applying implicit coordination in a NEC context. First, implict coordination implies a level of homogeinity in the behaviour of agents (e.g. "ant" agents in an ant-based system). A NEC system should be open to agents of other (coalition) participants. These agents are likely not to have the same characteristics, since they have been developed independently. The problem of heterogeneity can be resolved by defining shared models of coordination. Then, each agent has to have knowledge on the behaviour of its peers. However, if the coordination strategy changes, or agents with significantly different behaviour enter the grid then these models would have to be updated. This could result in a substantial maintenance effort. Second, systems in which agents interact through their environment (e.g. by means of stigmergy) are subject to errors or manipulation. Other agents, possibly with malicious intents, can deliberately or accidentally change or remove the marks in the environment, thereby corrupting system behaviour. In short, implicit coordination is less suitable for the open systems we envision, because the requirements to heterogeinity and security cannot be met.

We will now look at explicit coordination strategies. However, we should only consider strategies with a low communications overhead, such that the communication requirement is met and tempo of operations is not negatively influenced. According to our coordination taxonomy, there are two options: coordination strategies for cooperative and competitive agents. In case of competitive agents we have to deal with non-benevolent, selfish agents that aim at satisfying private goals. We already established in section 2.4.2 that market-based coordination strategies are suitable for competitive multi-agent systems. However, there are some drawbacks to a market-based approach [19]. The first concerns the potential complexity of the system, requiring reasoning about the bidding process (*counterspeculation*) and determining the auction's outcome. The latter can be particularly difficult in combinatorial auctions (i.e. agents negotiate about multiple objects), which is known to be a NP-complete problem. If the number of agents that participate in a market increases, allocation of objects might

be become computationally infeasible. The second drawback concerns security. Auctions and negotiations are vulnerable to cheating and manipulation. For example, intruding malicious agents can easily corrupt negotiations or auctions by starting false negotiations or by artificially increasing the price at auctions. Finally, auctions and negotiations usually involve a lot of communication, since bids and preferences have to be communicated between buyers, sellers and auctioneers. In short, there is a risk to designing NEC as a competitive MAS since, from a coordination perspective, it is hard to meet the scalability, security and communication requirements. Furthermore, computational complexity threatens tempo of operations.

Consequently, we prefer to design NEC as a MAS of cooperative agents, possibly using an organisational metaphor. To respect the scalability requirement, a centralised approach seems promising. Agents that perform coordination tasks are separated from agents that represent assets, so both groups can be scaled independently. Note that a centralised coordination approach does not mean that a single computational node is responsible for coordination. Rather, we envision a grid consisting of two types of agents: agent that coordinate and agents that are coordinated. This separation of concern will decrease the complexity of the agents that represent the assets, since they will have virtually no capabilities to coordinate. Moreover, the robustness constraint is respected because there is no single point of failure.

In short, for NEC we primarily prefer *explicit, centralised coordination strategies with low communication overhead in a cooperative agent environment.* In order to meet the communication and tempo requirements, the agents should be arranged in a flat organisation with short communication paths. Note that we have derived general properties for NEC coordination. In practice, certain clusters of agents in a NEC grid might require different coordination strategies that better fit the specific characteristics. In the following section we propose a coordination architecture for NEC, that allows for sufficient flexibility.

3.3 Proposed NEC coordination architecture

In this section we propose a coordination architecture for NEC, that is based on the considerations discussed in section 3.2. This coordination architecture puts the coordination strategy in an organisational contexct. We envision a hybrid NEC coordination strategy, composed of a global explicit, centralised coordination strategy with low communication overhead in a grid of cooperate agents, with subordinate clusters of agents that use local coordination strategies. To realise this, there should exist an abstraction between the global coordination strategy and the local clusters of agents. The global strategy is leading and treats the subordinate local structures as single functional units. At the local level, a limited set of agents and assets take responsibility for local coordination, possibly using a strategy that differs from the global one.

A structure that fits this description is the *holarchy*, or *holonic organisation.* A holarchy is a self-similar organisational structure comprised of multi-leveled, grouped organisations. Each grouping, or *holon*, has a character derived but distinct from the entities that are members of the group. At the same time, this same holon contributes to the properties of one or more holons above it. The nesting of holons within a holarchy is a hierarchy in which subordinate holons relinquish some of their autonomy to the superordinate holon. Figure 8 shows a graphical representation of a holarchy. Black dots correspond to agents, the circles represent holons and the arrows represent control flows.

The chief characteristic of a holarchy is that requesting agents require little or no knowledge on how a request is handled at the local level. Subordinate holons have a high degree of autonomy in how they fulfill a request. In short, a holarchy can be seen as a hierarchy that allows some amount of cross-tree interactions and local autonomy. This characteristic closely relates to the NEC concept of *power to the edge* [54], where entities at the bottom of a hierarchy synchronise their activities. Also, there is a resemblance between holarchies and federations, especially when dealing with flat holarchies that use agents as intermediates between the holons. For a concise description of holarchies, hierarchies, federations and other organisational paradigms, see the work of Horling et

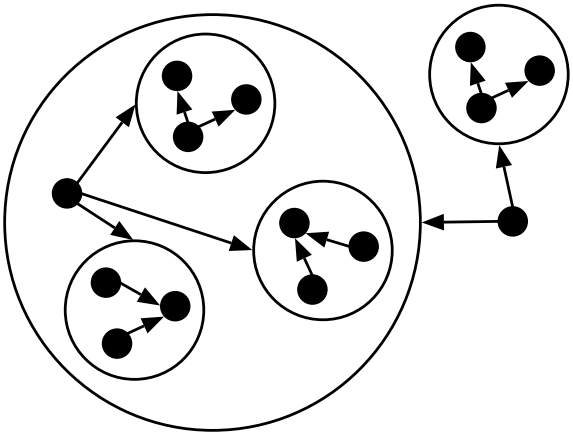


Figure 8: A holarchical organisation

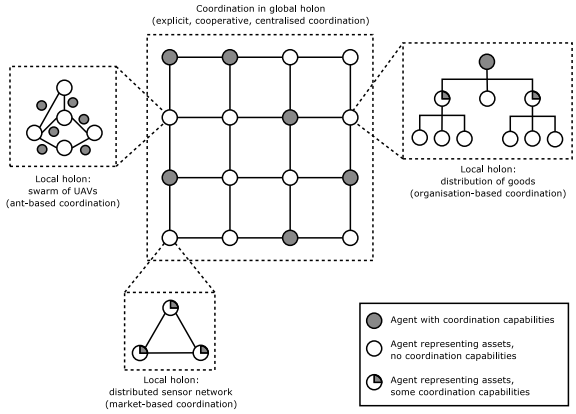


Figure 9: NEC holonic coordination archtecture

al. [19].

If we put NEC coordination in the context of a holarchic organisational structure, an organisation arises in which the top-level holon is formed by a global grid structure of agents and assets. This grid is comprised of two types of agents: agents that coordinate and agents that are subject to coordination. Agents of the first type are solely occupied with coordination tasks. Although distributed in a grid, these agents form the centralised coordination core of the system. Agents of the latter type have little or no coordination capabilities at the global level. These agents either represents assets (sensors, shooters, C2 nodes) or are subordinate holons that are treated as single functional units at the global level.

The subordinate holons themselves are multiagent systems of coherent assets. With "coherent" we mean that the assets can be managed with the same coordination strategy, and that from the global perspective the assets can be considered as a functional unit. For example, consider a set of UAVs that cooperate in a reconnaissance task. At the global level the coordination strategy can treat this set of UAVs as one logical sensor that provides information on enemy positions, buildings or terrain features, without having to bother about individual UAVs. The coordination of activities between individual UAVs is left to the subordinate holon. This holon can use a coordination strategy that best fits the specific characteristics of the tasks, the agents and the assets.

Figure 9 illustrates a holonic coordination architecture for NEC. Agents are represented by circles. The colouring denotes the coordination capabilities of an agent. At the global level agents are organised in a grid. Some nodes in the grid are holons in which agents are organised and coordinated differently. Note that communication paths are short, since agents in the global holon take direct orders from the coordinating agents. Between holons communication paths can be kept short by limiting the number of nestings within holons.

Coordination at the top-level holon can be realised with COMPASS, or similar technology. Coordinating agents in the grid must have some knowledge of the available agents and assets, their capabilities, properties and interrelations. This can be realised with a distributed knowledge base, from which coordination plans are continuously inferred. If a new agent or holon is added to the grid, a new entry is added to this knowledge base. Besides planning and delegating tasks to agents, coordinating agents should monitor the execution of tasks and intervene if necessary.

4 Discussion

In this work we recognised the need for tackling coordination related issues in NEC. As more and more military assets (sensors, shooters, C2 systems) are deployed in networked structures, such that their combined capabilities can be fully exploited, the problem of coordination gets more

prominent. We observed that NEC systems can be seen as a special kind of multiagent systems. Hence, theory and concepts concerning coordination explored in the field of MAS design can also be applied to NEC systems.

We explored theory on MAS coordination and identified five dimensions by which coordination strategies can be classified, yielding a coordination taxonomy. Next we identified the requirements that have to be met by a NEC coordination strategy. These requirements concern flexibility, heterogeneity, security, communication, robustness, tempo and scalability. Based on these requirements we argued that explicit, centralised coordination strategies, with limited communications overhead, in a cooperative agent environment are best suited for managing activities in a NEC grid. Other types of coordination might still be useful for specific clusters of agents and assets, and they can be incorporated in a hybrid holarichal coordination architecture.

Although we identified the most promising type of strategy for global NEC coordination, many design decisions for actual implementations still have to be made. The results of this work should be seen as a starting point in the design and implementation process of coordination mechanisms for NEC. Furthermore, the analysis has been purely theoretical. The proposed coordination architecture should be further refined and its qualities should be evaluated by means of simulations and prototypes.

Promising technologies for this future research are COMPASS and Cougaar. COMPASS, developed by defence industry (Thales Naval Netherlands[4]), is a prototype of a multi-platform middleware system that is aimed at the dynamic configuration of capabilities in network centric systems. This prototype features mechanisms for explicit, centralised coordination with limited communication overhead, as required for coordination at the global level in our proposed architecture.

Cougaar[5] is an agent platform and framework developed by DARPA[6], for the development large-scale distributed agent-based applications. Aimed at defence applications, Cougaar has some powerful features for reliable inter-agent message transport, task planning and service discovery. Within Cougaar, agents can be organised in logical groups or *communities*, and can be addressed based on community membership. Using Cougaar's community features, agents can easily be organised in holonic organisations. Both Cougaar and COMPASS possess specific military-standard system qualities such as robustness, security and fail-safety.

Defence is not the only domain in which the network centric paradigm is explored. The merits of a network centric approach to operations is also recognised in the crisis response domain. Similar to the trend in the military domain, (Dutch) crisis response organizations tend to shift from hierarchical, regionally organized structures to dynamic assemblies of people and resources. Research projects that, among other things, focus on coordination in agile networked crisis response organisations are ICIS[7] and COMBINED[8], both supported by the Dutch Ministry of Economic Affairs and hosted by the Decis Lab[9]. Because of the similarities in topics, further research activities should be combined to establish cross-fertilisation between crisis response research and defence research.

References

[1] M.R. Endsley. Theoretical Underpinning of Situation Awareness. In M.R. Endsley and D.J. Garland, editors, *Situation Awareness Analysis and Measurement*. LEA, Mahwah, NJ, USA, 2000.

[2] N. Blatt. Trust and Influence in the Information Age: Operational Requirements for Network Centric Warfare. In *Proceedings of the 10th International Command and Control Research and Technology Symposium*, McLean, Virginia, USA, June 2005.

[3] D.S. Alberts, J.J. Garstka, R.E. Hayes, and D.A. Signori. *Understanding Information Age*

[4] `http://www.thales-nederland.nl`

[5] Cognitive Agent Architecture, `http://www.cougaar.org`

[6] Defense Advanced Research Projects Agency, `http://www.darpa.gov`

[7] Interactive Collaborative Information Systems, `http://www.icis.decis.nl`

[8] Chaotic Open world Multi-agent Based Intelligently NEtworked Decision support Systems, `http://combined.decis.nl`

[9] Delft Cooperation on Intelligent Systems, `http://www.decis.nl`

Warfare. DoD C4ISR Cooperative Research Program, August 2001.

[4] D.S. Alberts, J.J. Garstka, and F.P. Stein. *Network Centric Warfare: developing and leveraging information superiority*. DoD C4ISR Cooperative Research Program, 2nd edition, 1999.

[5] Ministry of Defence. Outline Concept for the UK Network Enabled Capability (nec), May 2003.

[6] B.W. Young. Future Integrated Fire Control. In *Proceedings of the 10th International Command and Control Research and Technology Symposium*, McLean, Virginia, USA, June 2005.

[7] S. Franklin and A. Graesser. Is it an agent, or just a program? A taxonomy of Autonomous Agents. In *Proceedings of the Third International on Agent Theories, Architectures and Languages*, 1996.

[8] H. Nwana. Software Agents: An Overview. *Knowledge Engineering Review*, 11(3):205–244, 1996.

[9] J.M. Bradshaw, editor. *Software Agents*. AAAI Press, 1997.

[10] N. Jennings. On Agent-Based Software Engineering. *Artificial Intelligence*, 177(2):277–296, 2000.

[11] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley and Sons Ltd., Chichester, UK, 2002.

[12] A. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann publishers Inc., San Mateo, CA, USA, 1988.

[13] Thomas W. Malone and Kevin Crowston. What is coordination theory and how can it help design cooperative work systems? In *CSCW '90: Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, pages 357–370. ACM Press, 1990.

[14] N.R. Jennings. Coordination techniques for distributed artificial intelligence. In *Foundations of distributed artificial intelligence*, pages 187–210, New York, NY, USA, 1996. John Wiley & Sons, Inc.

[15] D. Corkill and S. Lander. Agent Organizations. *Object Magazine*, 8(4), April 1998.

[16] A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, editors. *Coordination of Internet Agents: Models, Technologies, and Applications*. Springer, 2001.

[17] C.J. van Aart. *Organizational Principles for Multi-Agent Architectures*. Whitestein Series in Software Agent Technologies. Birkhäuser, Basel, Switserland, 2005.

[18] S. Willmott. *Coordination Structures for the Intelligent Control of Dynamic Distributed Systems*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2002.

[19] V. Lesser B. Horling. A Survey of Multiagent Organizational Paradigms. *The Knowledge Engineering Review*, 19(4):281–316, 2005.

[20] ISO Open Systems Interconnection Reference Model. http://www.iso.org.

[21] Sascha Ossowski. *Co-Ordination in Artificial Agent Societies: Social Structures and Its Implications for Autonomous Problem-Solving Agents*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.

[22] R. Beckers, O. Holland, and J. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In *Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems*, pages 181–189, Cambridge, MA, USA, 1994.

[23] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, New York, NY, USA, 1999.

[24] O. Holland and C. Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5(2):173–202, 1999.

[25] Z. Mason. Programming with stigmergy: using swarms in construction. In *Artificial Life VIII: Procedings of the Eighth International Conference on Artificial Life*, pages 371–374, Sydney, Australia, 2002. MIT Press.

[26] T.W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.

[27] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated Negotiation: Prospects, Methods and Challenges. *International Journal of Group Decision and Negotiation*, 10(2), 2001.

[28] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemare, N. Maudet, J. Padget, S. Phelps, J. A. Rodruez-Aguilar, and P. Sousa. Issues in Multiagent Resource Allocation. *Informatica*, 2005. Accepted for publication.

[29] P. Giorgini, M.Kolp, and J. Mylopoulos. Multi-Agent Architectures as Organizational Structures. *Journal of Autonomous Agents and Multi-Agent Systems*, 2005. in press.

[30] C. Jones, D. Shell, M.J. Matarić, and B. Gerkey. Principled Approaches to the Design of Multi-Robot Systems. In *Proceedings of the Workshop on Networked Robotics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-04)*, Sendai, Japan, 2004.

[31] M. Yokoo, E.H. Durfee, T. Ishida, and K. Kuwabara. The Distributed Constraint Satisfaction Problem: Formalization and Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):673–685, 1998.

[32] H. Jung, M. Tambe, and S. Kulkarni. Argumentation as distributed constraint satisfaction: applications and results. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous Agents*, pages 324–331. ACM Press, 2001.

[33] T. Khedro and M.R. Genesereth. Modeling Multiagent Cooperation as Distributed Constraint Satisfaction Problem Solving. In *ECAI*, pages 249–253, 1994.

[34] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, pages 1104–1113, 1980.

[35] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, pages 8–37, 1961.

[36] H. Mintzberg. *Structures in fives: designing effective organizations*. Prentice Hall, Englewood Cliffs, NJ, USA, 1993.

[37] M.S. Fox. An organizational view of distributed systems. In *Distributed Artificial Intelligence*, pages 140–150, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

[38] A. Fuxman, P. Giorgini, M. Kolp, and J. Mylopoulos. Information systems as social structures. In *Proceedings of the 2nd International Conference on Formal Ontologies for Information Systems, FOIS01*, Ogunquit, USA, October 2001.

[39] M. Kolp. Organizational Structures for Multi-Agent Architectures: Application to Lego Mindstorms Mobile Robots. IAG Working Paper 28/02, January 2002. http://www.cs.toronto.edu/ mkolp/pubs.html.

[40] T. T. Do, S. Faulkner, and M. Kolp. The Structure-in-5 as an Agent Architectural Pattern. Working Paper IAG 38/02, May 2002. http://www.cs.toronto.edu/ mkolp/pubs.html.

[41] U. Endriss, N. Maudet, F. Sadri, and F. Toni. On optimal outcomes of negotiations over resources. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 177–184. ACM Press, 2003.

[42] G. Beni and U. Wang. Swarm intelligence in cellular robotic systems. NATO Advanced Workshop on Robots and Biological Systems, Il Ciocco, Tuscany, Italy, 1989.

[43] H.D.V. Parunak and S.A. Brueckner. Engineering Swarming Systems. In F. Bergenti, M.P. Gleizes, and F. Zambonelli, editors, *Methodologies and Software Engineering for Agent Systems*, Amsterdam, The Netherlands, 2004. Kluwer.

[44] J.A. Sauter, R. Matthews, H.D. Parunak, and S. Brueckner. Performance of digital pheromones for swarming vehicle control. In *Proceedings of the 4rd International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2005)*, pages 903–910. ACM Press, July 2005.

[45] K.S.J. Pister, J.M. Kahn, and B.E. Boser. Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes. Highlight Article in 1999 Electronics Research Laboratory Research Summary.

[46] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for Smart Dust. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278. ACM Press, 1999.

[47] C. Bourjot, V. Chevrier, and V. Thomas. A new swarm mechanism based on social spiders colonies : from web weaving to region detection. *Web Intelligence and Agent Systems*, 1:47–64, 2003.

[48] G. Di Caro and M. Dorigo. AntNet: A Mobile Agents Approach to Adaptive Routing. Technical Report 97-12, IRIDIA, Universit Libre de Bruxelles, 1997.

[49] N. Foukia and S. Hassas. Managing Computer Networks Security through Self-Organization: A Complex System Perspectives. In G. Di Marzo Serugendo, A. Karageorgos, O.F.Rana, and F. Zambonelli, editors, *The First International Workshop on Engineering Self-Organising Applications (ESOA)*. Springer Verlag, July 2004.

[50] H.V.D. Parunak and S. Brueckner. Swarming Coordination of Multiple UAV's for Collaborative Sensing. In *Proceedings of the Second AIAA 'Unmanned Unlimited' Systems, Technologies, and Operations Conference*, San Diego, CA, USA, 2003.

[51] C. Hawthorne and D. Scheidt. Moving emergent behavior algorithms from simulation to hardware: Command and control of autonomous uxvs. In *Proceedings of the 10th International Command and Control Research and Technology Symposium*, McLean, Virginia, USA, June 2005.

[52] John Arquilla and David Ronfeldt. Swarming and the Future of Conflict. RAND National Security Research Division (NSRD), `http://www.rand.org`, Documented Briefings, no. DB-311-OSD, 2000.

[53] S.J.A. Edwards. Swarming on the Battlefield: Past, Present, and Future. RAND National Security Research Division (NSRD), `http://www.rand.org`, Documented Briefings, no. MR-1100-OSD, 2000.

[54] D.S. Alberts and R.E. Hayes. *Power to the Edge*. Information Age Transformation Series. DoD C4ISR Cooperative Research Program, 2003.