14th ICCRTS
‘C2 and Agility’

Agility through Adaptive Autonomy

Topic 7: C2 Approaches and Organization

Martijn Neef and Bob van der Vecht
TNO Defence, Security and Safety

Point of contact: Martijn Neef
TNO Defence, Security and Safety
Oude Waalsdorperweg 63
P.O. Box 96864
2509 JG, The Hague, The Netherlands
telephone: +31-70-3740864
e-mail: martijn.neef@tno.nl

Agility through Adaptive Autonomy

Martijn Neef[1] (martijn.neef@tno.nl) and Bob van der Vecht[1,2] (bob.vandervecht@tno.nl)
[1] Command & Control and Information Management Department,
TNO Defence, Security and Safety, The Hague, The Netherlands.
[2] Department of Information and Computing Sciences, Utrecht University, Utrecht.

Abstract. In any multi-actor environment, there is an inevitable trade-off between achieving global coordination of activities and respecting the autonomy of the actors involved. Agile and resilient behaviour demands dynamic coordination capabilities, but task and resource allocation quickly becomes a demanding challenge in joint NEC environments because of individual constraints and demands. In this paper, we present work on adaptive autonomy in multi-agent organizations. We have been researching the relationship between autonomy and coordination, and developed an agent reasoning model that enables collaborative task coordination, but also guarantees individual autonomy – the capability to self-manage behaviour. We define autonomy as the amount of influence other agents have on one's decision making process. We have given the agent options to adapt its openness to external influences, so it can change its own level of autonomy. This allows agents to select the level of autonomy that best fits the circumstances, given a certain tasking, individual policies and organizational structure. We have incorporated this concept in a practical model and added heuristics for environmental events, information relevance and organizational rules. Our approach addresses fundamental collaborative challenges in NEC environments, and may bring about new perspectives on autonomy in collaborative environments.

1. Introduction

Achieving high levels of agility and resilience in networked military organizations will require new ways of thinking about command and control. While traditional military organizations are gradually being readied for network-centric missions, it becomes obvious that we need rethink coordination strategies. In future arenas, there will be many more parties involved, and the chain of command will be much less transparent. We will need to rely more on distributed processes and accept that traditional centralized command and control strategies will not lead to agile capabilities.

Our work on adaptive autonomy in agent systems might provide some interesting perspectives on agility and resilience in NEC environments. We have been researching the topic of autonomy in multi-agent systems, and in particular the relationship between *autonomy* and *coordination*. In any multi-actor environment, there is an inevitable trade-off between achieving global coordination of activities and respecting the autonomy of the actors involved. If decision making processes and operational tasks are distributed over many parties, respect of autonomy becomes an increasingly important issue. Obviously, this is a crucial issue in NEC organizations, especially in joint and combined missions where there might be conflicts of interest.

In this paper, we present work on adaptive autonomy in multi-agent organizations. We have been exploring the relationship between autonomy and coordination in agent organizations. We are working on collaborative decision-making models that respect the agent's own autonomy, but at the same time take organizational roles and operational conditions into account. We believe that our approach addresses fundamental issues in network-centric operations, and may bring about new perspectives on autonomy and agility in collaborative, networked environments. In this paper, we explain our approach, demonstrate how it can be used to model organizational behaviours and discuss its relevance in practice.

2. Autonomy and agility in agent systems

Artificial agent research is usually regarded as an academic practice, with limited practical applicability for complex environments. This is unfortunate, because research into agent organizations revolves around many of the same issues that complicate the development of NEC organizations, such as collaborative decision making, decentralized coordination strategies and dynamic organizational structures. Actor autonomy is another important trait that both agent organization and NEC organizations share. We will discuss agent autonomy, and explain why autonomy is important for agile coordination in NEC organizations.

The role of autonomy in agent systems
Autonomy is an important aspect of artificial agents. It is usually regarded as one of the defining features of an agent (Jennings, 2000, Castelfranchi, 1995). Agents have, by definition, control over both their internal state (e.g. beliefs, desires, intentions) and behaviour. Agents act goal-oriented, and exert their independence and problem solving capacities to reach their objectives. In multi-agent systems, there is communication and coordination of activities between agents.

Coordination implies that agents enter into a work agreement, and agree to a certain interdependence and interaction. This implies that agents will influence each other, and possibly make demands that may affect each other's degree of autonomy. For instance, a supervising agent may demand that one of his subordinate agent perform a certain task. This demand challenges the autonomy of the subordinate agent. If the relationship between the supervisor and the subordinate agent are clear and agreed upon by both parties, then the subordinate agent will not object to the request, since it will also advance its own goals (*successfully follow orders and help to fulfill the organizational goals)*. Moreover, you could say that agreements on relationships actually define the level of autonomy that either party may exhibit. In figure 1, such agreements would set the 'Permitted Actions' boundary.

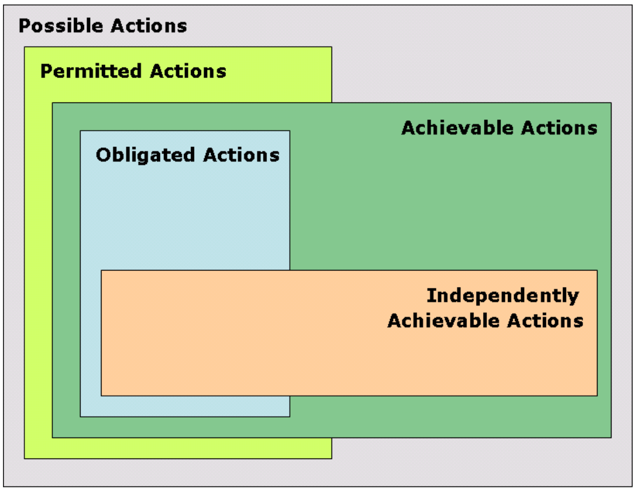

Figure 1: Basic dimensions of autonomy (Bradshaw, 2003)

On the other hand, if the demand directly opposes the agent's beliefs or own objectives, then the agent will find itself challenged in autonomy. Such a situation might be a cause to question the demand itself, or the interaction agreement. Should the agent follow orders, and perform an act that in detrimental to its own goals, or exert autonomy, and let his own motivations prevail over demands from others? Such situations are reminiscent of the difficult autonomy challenges that Asimov's Laws of Robotics brought about (Asimov, 1990), and many ethical questions in warfare. The relationship between ethics, policies and autonomy is an important but precarious one, and not easily captured in a model. We will not diverge into a discussion on the role of human ethics in decision making. However, for coordination purposes in multi-actor environments, we do need a way to model and deal with autonomy in multi-agent coordination. We believe that is essential to give agent autonomy a central role in coordination activities. Most conventional coordination tactics do not explicitly give the affected actors a say in the process, and consequently negate the essence of autonomy: giving actors the freedom to object or accept proposal, based on their own interest.

Autonomy and coordination mechanisms
Since, by definition, agents need to be in control of their own internal state and behaviour, the question rises how agent decision making is actually impacted by external influences. How

can an agent maintain control over his own behavior, but at the same time cooperate with other agents to achieve coordination? The traditional way to achieve coordination is by developing a top-down coordination mechanism. The designer of a multi-agent system specifies the tasks and interaction mechanisms that the agents will follow. The rules of the coordination mechanism are embedded in the decision-making process of the agent. This allows the agents to jointly find a correct division of labour. One can argue that such agents are not truly autonomous, since they can only behave in line with commands that are set forth at design-time. The agent has no means to enforce its autonomy, and cannot pro-actively deviate from plans. Of course, in many systems, this is a desirable feature: we want the system to do what it was designed for. In some cases, however, it might be favorable to grant agents a degree of sovereignty. One could think of situations where standard procedures fail, and agents are left to their own devices to create alternative plans. For instance, if the chain of command collapses because of communication breakdown, deployed agents need to be able to take matters into their own hands. In other words, they need to adapt their autonomy from being commanded to self-ruling.

Adaptive autonomy and agility

In our perspective, autonomy is about the level of independence of decision making. The degree of autonomy of decision making can be defined as the degree of intervention by other agents on the decision making process of one agent (Barber, 2001). An agent that is heavily influenced by other agents in its decision making is displaying obedient behaviour. An agent that does not allow any external influence in its decision making is being ultimately independent. By altering the amount of external influence on its decision making, an agent can adapt its own level autonomy. In this fashion, agents can actively select the level of autonomy that best fits the circumstances. In effect, an agent that changes it level of autonomy in response to changing conditions shows *adaptive autonomy*.

Agility refers to the ability of an entity to quickly and gracefully respond to a changing environment. Resilience refers to the ability of an entity to withstanding disturbance, and to readily recover. Both are very desirable features for organizations, and, understandably, very much in the spotlight of C2 and NEC research. An agile and resilient organization must be able to cope with changes in circumstances (e.g. unexpected events, new objectives) and changes in organizational structure (e.g. failing elements, modified chain of command), and respond promptly with a new course of action or a new organizational layout (e.g. new coordination scheme). A traditional top-down approach to coordination will not yield agile behaviour in a modern network-centric organization, because of the transient nature of NEC organizations. There are simply too many actors and systems involved to create a course of action that fits all individual capabilities and norms. We believe that agility in a dynamic network-centric organization must find its roots in individual adaptive autonomy.

3. A Model for Adaptive Autonomy

We have developed a reasoning model for artificial agents that implements the notion of adaptive autonomy as described above. We give the agent means to manage incoming external events (observations, messages) through an *influence control* process. This process controls which events affect the internal state of the agent, and thus influence decision making and behaviour. With the process of influence control, we refer to a conscious process, where the agent is aware of the external events and uses its knowledge to determine the effect on the mental state. This makes perception an active process, similar to the psychological Perceptual Control Theory (Powers 1960, Farrell 1999), instead of a passive process.

The agent itself is in command of the influence control process, and can configure the process to fit his own objectives. This approach guarantees the autonomy of the agent (Van der Vecht, 2007), because it makes the agent ultimately responsible for its own behaviour. Figure 2 shows the position of influence control process in relation to decision making process of an agent. The *internal state* holds the agents' current beliefs, goals and plans. The *decision making* process processes the elements of the internal state to derive decision and courses of action.

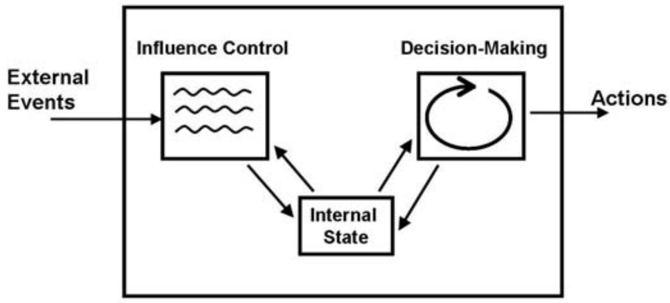

Figure 2: Schematic reasoning model of an adaptive autonomous agent

Event Processing

The influence control process uses event-processing rules to decide on adoption or rejection of certain influences. We propose to use rules of the form:

```
<EVENT> <- <CONSTRAINTS> | <EFFECT>
```

The event is the trigger of the rule; it should match the incoming event. There are three types of events: *observations, inform messages,* and *request messages.* The first two contain information, whereas the latter contains a task or a goal.

The constraints describe situational constraints for the rule. They should match the beliefs and goals of the agent to make the rule applicable.

The effect of the rule specifies an internal action of the agent that holds the effect on the mental state. The possible effects depend on how the mental state of the agent is constructed. Receiving new information generally leads to belief updates. Concerning tasks, an agent can add a goal, as a consequence of accepting a request. Another option is to specify rules that result in ignoring the external event. For example, when information is irrelevant, or when the agent is busy. We propose to use three internal actions and the corresponding effect on the mental state:

- Update Beliefs
- Add Goal
- Ignore Event

As we stated, this set does not contain all possible results of event-processing rules. For example, another internal action can be to drop a goal, as response to a prohibition. Furthermore, a belief update is not necessarily a straight-forward process, as adopting a new belief can lead to inconsistent beliefs. However, in the examples presented in this paper, the presented three are sufficient to demonstrate control over external influences

Basic Attitudes

The event-processing rules specify the attitude of how agents perceive the world. The attitudes are directly related to levels of autonomy of the decision-making process. For all events the agent can choose to adopt or reject them.

Event	Effect	Basic Attitude
Observation	Update Beliefs Ignore Event	Self-reliant Non-self-reliant
Inform message	Update Beliefs Ignore Event	Trusting Non-trusting
Request message	Add Goal Ignore Event	Cooperative Non-cooperative

Table 1. Basic Attitudes

When combining the three event-types and the two processing options, we can construct eight general attitudes. We present three general attitudes and show how they are constructed from the event-processing rules:

- *Non-social*: An agent with a non-social attitude does not interact with other agents. It adopts own observations, but messages from others are ignored. Therefore, the goal/task determination is free from external influences. The agent creates and selects its own goals and plans. Influence via environmental modification is still possible; it is possible to influence the agent's behavior by manipulating the environment. The following event-processing rules for messages and observations are active:

```
observation(X) <- TRUE | UpdateBeliefs(X)
message(Sender, Performative, X) <- TRUE | IgnoreEvent()
```

- *Self-reliant/trusting*: A self-reliant, trusting agent will process messages from others and believe the content. Its beliefs are influenced by others. The agent adopts own observations as well. The agent determines its goals and tasks by itself. We have implemented the \emph{self-reliant, trusting} attitude by the following event-processing rules:

```
observation(X) <- TRUE | UpdateBeliefs(X)
message(Sender, inform, X) <- TRUE | UpdateBeliefs(X)
message(Sender, request, X) <- TRUE | IgnoreEvent()
```

- *Self-reliant/cooperative*: If agent A is cooperative with respect to agent B, it will do what agent B asks for, without considering other options. Its tasks and goals are determined by agent B. Agent B can send a request message to agent A. A processes the message by adding the request to its goal base. The agent adopts own observations in order to create an own world perspective. The following rules specify the self-reliant, cooperative attitude:

```
observation(X) <- TRUE | UpdateBeliefs(X)
message(S, inform, X) <- TRUE | IgnoreEvent()
message(S, request, X) <- TRUE | AddGoal(X)
```

These are three examples of the eight possible attitudes. Of course, it is possible to construct more complex profiles by specifying the event-processing rules differently, for example by specifying the situational constraints.

These constraints use local knowledge from the agent's internal state to permit or bar external events from influencing the agent's beliefs. Therefore, sensible knowledge should be used. These reasoning rules are effectively *heuristics*.

Meta- knowledge for influence control

What knowledge should be taken into account by the reasoning rules? We have explored several heuristics that seem appropriate to control external influences. One of the heuristics is relevance of information. If an agent can determine the relevance of information with respect to a certain goal, it can focus itself on a specific type of information, or prevent itself from information overload by filtering incoming information on relevance. Information relevance is important for influence control. Another related heuristic is the *state of mind* of the actor. An actor will react differently when it is busy than when relaxed, or when it feels endangered. We cluster such heuristics as *self knowledge*. Examples of event-processing rules using this type of knowledge are:

```
observation(X) <- relevant(X) | UpdateBeliefs(X)
message(Sender, Performative, X) <- busy() | IgnoreEvent()
```

Self knowledge creates heuristics for an agent to determine how it is influenced. The agent should also be able to control by whom it is influenced. The reasoning rules for event control can use knowledge about the existing organization or about the agent's social context. Can the sender of a message be trusted? Does a request originate from a superior or from an unfamiliar source? Agents can achieve coordination by allowing influence on the internal state based on social and organizational knowledge, for example:

```
message(S, inform, X) <- trusted(S) | UpdateBeliefs(X)
message(S, request, X) <- superior(S) | AddGoal(X)
```

One can think of several other reasons to allow or disallow influence on the internal state in a certain context. A specific coordination type puts requirements on the environment, such as availability of communication or information resources. Critical changes in the environment can be used to determine the proper level of autonomy, and thus provides another important heuristic for influence control: *environmental knowledge*.

Table 2 summarizes the above types of knowledge that can be used in reasoning rules for influence control. This is obviously not an exhaustive list, but these three main types of knowledge seem to capture relevant factors.

Type of knowledge	Examples
Self knowledge	Is this information relevant for my objectives? Does my state of mind permit new requests?
Organizational/Social knowledge	Relation to information source Can the source be trusted?
Environmental knowledge	Availability of communication Availability of information sources

Table 2. Examples of meta- knowledge for influence control

When we use the meta-knowledge in the event-processing rules, we create agents with adaptive autonomy. It depends on the situation whether an agent allows influences on decision-making or not and thus makes the agent locally responsible for its own behavior. Furthermore, it is a powerful feature to design dynamic coordination mechanisms, as we will see in the next section.

4. Using Adaptive Autonomy for Coordination

When designing a coordination mechanism, we specify the desired behavior of the participants and the way they exchange information and delegate tasks. Dekker (Dekker, 2005) presents a taxonomy of coordination mechanisms in networked organizations. He distinguishes two dimensions: homogeneity/heterogeneity of the participants and value-symmetry/non-value-symmetry of the roles they perform. From these dimensions he creates eight different coordination models. The three basic types are: centralized control, request-based coordination and emergent coordination. The other five are combinations of the basic types.

In agent research many work has been done on specifying organizational models, for example OperA (Dignum, 2004). The organizational model describes the roles and relations between actors, and specifies behavioral rules in terms of norms. It is constructed based on functional requirements of the organization. The behavioral guidelines for the roles are described in contracts. Taking up a role in an organization means that an agent is expected to act following the contracts.

OperA describes a coordination mechanism based on organizational at an abstract level. The organizational model is separated from the reasoning process of the agents that will fulfill the roles. Earlier work has shown that contracts specified in OperA can be translated to event-processing rules for the agent (Van der Vecht, 2008). However, the specified behavioral rules in the organizational model are static, which has drawbacks for agility.

Agile Organizations

All static coordination mechanisms have their advantages and drawbacks. In a dynamic situation it is not possible to choose one coordination type that will always lead to the best performance. The main reason is that unexpected situations can occur that were not known at design time and that may not fare well with the selected coordination mechanism. There are two ways to achieve agility in an organization:

- Top-down: a new organizational model is defined, and the agents change their contracts with the organization. As a consequence they adopt different reasoning rules for influence control, which will change the coordination.
- Bottom-up: the agents change the (priority of) reasoning rules for influence control by themselves if they notice that the organizational model fails. They adjust their autonomy to repair the organizational failure.

The top-down dynamics can be achieved by carrying out structural changes, whereas bottom-up dynamics originate in autonomous choices of the agents. In both situations, organizational adaptation implies changes in the reasoning process of individual agents. The model of adaptive autonomy presented here, enables to achieve the required adaptivity in the reasoning process.

5. Example application scenario

We use a simple maritime NEC scenario to demonstrate the approach. A combined task force is in charge of ensuring a safe transit for a high value unit. The coalition exists of four vessels from three nations. Each vessel has specific capabilities. Each nation has specific operational policies.

There are different ways to organize tasks, as shown in the C2 taxonomy of Dekker (Dekker, 2005). Additionally, there are other circumstances that require dynamic reorganization, such as a necessary switch between decentralized and centralized coordination, upsizing or downscaling of the organization, and a change in the rules of engagement. Figure 3 illustrates some of these situations.

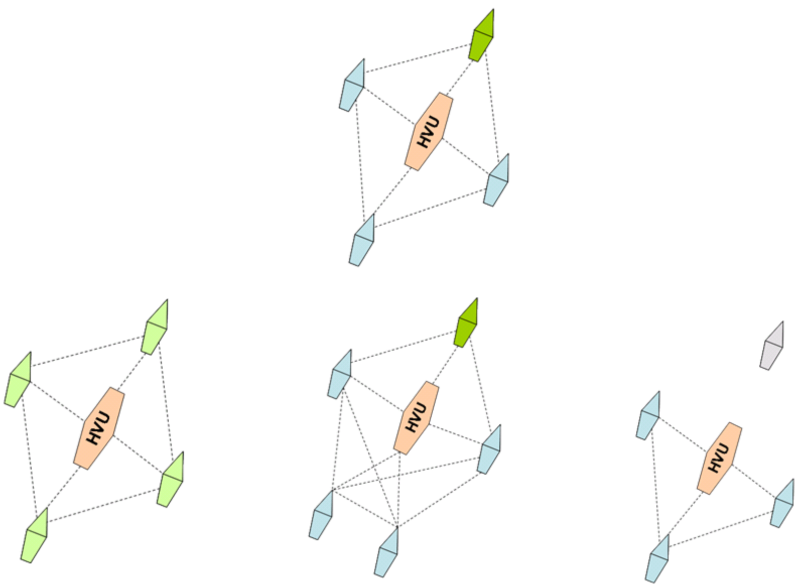

Figure 3: Example situations in which reorganization is necessary. The top figure shows the default mode, with a single command ship, three additional vessels and a high value unit (HVU). The bottom row shows (a) decentralized command, (b) addition of a vessel and (c) loss of command.

Bottom-up Dynamics

First we show bottom-up dynamics in the organization. We achieve this by changing the autonomy level of the decision-making process of the agents. We start with a hierarchical organization with one commander and three followers. We have implemented the followers with the following rules for event processing:

- *IF* request from commander *THEN* follow request
- *IF* I am in danger *THEN* ignore requests and follow own goals
- *IF* no communication *THEN* follow own goals

The rules ensure that the agents follow the commands, but if communication fails, they will pursue their goal using local observations. Also, in case of danger the agent will take care of its own safety.

Under these rules, fleet members might be disobedient to the requests of the commander at certain times. When they feel, based on their local belief, that they are not in danger, and in clear communication with the commander, they will follow the orders. In this manner, the organization effectively switches between different coordination mechanisms. Although the dynamics of the coordination mechanism has been designed and predefined, the responsibility of the choice for coordination type is in hands of the actors themselves.

Top-down Dynamics
Top-down dynamics can be achieved by carrying out structural changes. In our coordination model it means that the contracts between the agents and the organization need to be changed. The modular approach in the agent's reasoning model provides a mechanism to adopt organizational rules into the decision-making process. This can be done dynamically by changing contracts at runtime (Van der Vecht, 2008). For instance, the commander may need to put a new set of rules of engagement into effect. He can implement these rules by issuing new social contracts to all his fleet members. These contracts put the new rules of engagement into practice.

Let us describe some simple scenarios that show how the adaptive autonomy model facilitates agile behaviour. We start with the hierarchical organization of the fleet. The vessels have adopted the corresponding organizational rules as event-processing rules. The rules specify a coordination mechanism in which the agents are obliged to inform the commander about a status change, and that they have to answer an extinguish request from the commander with an accept or reject message. The commander is the highest in the hierarchy. We assume that he can take the initiative to change the coordination process.

If, for some reason, the current coordination mechanism causes too much communication over the network, the commander can take the initiative to change the interaction protocol. The agents can decide on a new protocol that leaves the reply of the vessels out. The vessels translate the new interaction contract to event-processing rules and ends up with the same set of rules, but without the rule saying to explicitly answer the request. The new interaction protocol has of course an effect on the information circulation within the organization, and therewith possibly on the performance.

A more rigid example of reorganization can lead to a new role division. Imagine that the commander agent gets overloaded by too many tasks. It can assign the new commander role to a vessel agent to take over the coordination in a certain area. Furthermore, he informs other vessels that they get a new superior. The vessel agent becoming coordinator adopts the behavior rules belonging to the new role. The vessels who get a new superior update their beliefs according to the new situation.

These scenarios describe top-down dynamics in the organization. Coordination in the fleet is guided by contracts, and agile behaviour results from adaptation of the contracts. The modular approach in the agent's reasoning model makes this possible, and still leaves room for the individual agent to implement its preferences.

6. Practical application in NEC environments

How does the above model translate to operational environments? The most evident use is in supporting coordination activities in multi-party environments. Agile and resilient behaviour demands dynamic coordination capabilities. Task and resource allocation quickly becomes a demanding challenge in joint and combined NEC environments because of individual constraints and demands. Artificial agents could support this process by acting as *proxies*: mediating representatives for all parties involved.

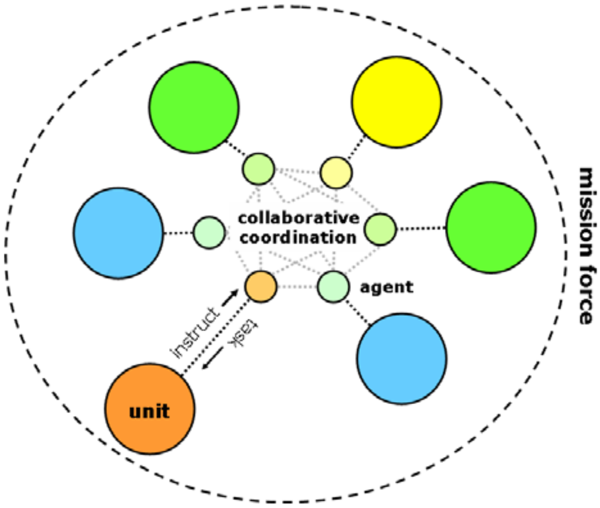

Figure 4: Using mediating agents for collaborative coordination

Such proxy agents could support mission planning and resource sharing, and make it easier to respect individual constraints and policies. Each force member can instruct its proxy agent by tuning the various attitudes to influences. For example, a unit may not be allowed to engage in offensive measures because of local rules of engagement, but may be allow its resources to be used in defensive actions. It could instruct its representative agent accordingly by configuring its openness to external influences. The agent would decide to filter out requests for offensive capabilities, but join the coordination process when dealing with defensive goals. It would use its delegated autonomy to actively accept of refuse requests. In a hectic conflict, there may not be enough time to deal with such individual constraints or resolve potential conflicts through ordinary communication. Artificial agents may help to cope with the dynamics of multi-party collaborations and improve agility. When the organization changes structurally because of leadership changes or the arrival of extra units, the embedded organizational knowledge in the agents

Such autonomy-related *configurations* could be further facilitated by using meta-reasoning models. In a meta-reasoning model, the agent does not just reason about particular external events, but also over the relationship between various goals and information. This approach gives us a way to use prioritization of decisions, and in effect construct 'attitudes'. For example, given a certain operational event, the agent's reasoning model may conclude that, based on internal knowledge and the agent's attitude, it prioritizes to force protection over self-defense. This means that the agent has received information that it is relevant for the success of two goals (self-defense and force protection), but that it actively chooses to let only the force-protection rule succeed. It blocks the information for the self-defense rule that would lead to retreat. Because of the modularity of the autonomy model, it is relatively easy to develop and implement such meta-reasoning models and their corresponding attitudes. In practice, such attitudes could serve as a way to implement doctrines.

7. Conclusions

In this short paper, we briefly introduced our work on adaptive autonomy in agent systems. We have developed a decision making model for artificial agents in which coordination can be defined in terms of organizational norms and rules, but that also guarantees autonomy. We have given the agent capabilities to self-adapt its openness to external influences, so it can change its own level of autonomy. We distinguish several types of external influences that may impact decision making, such as environmental events, information relevance and organizational rules, and give practical means to define attitudes and preferences.

We believe that our approach addresses some fundamental challenges in the progress towards higher NEC maturity levels. Agility and self-synchronization can only be achieved when participants in a NEC organization have adopted practical methods to manage their autonomy. We recognize the essence of having local autonomy, but we also recognize the necessity of coordinated activities. Our model shows that it is possible to relate autonomy and global coordination, and define simple mechanisms that enable adaptive behaviour. The underlying concept is relevant for understanding and facilitating task coordination in NEC environments. For instance, networked parties might interact through the use of mediating agents, that represent a party, and guards its autonomy. There will be many issues in implementing such a model, but it may inspire NEC developments, and bring about new perspectives on autonomy in collaborative environments.

8. References

Asimov, I. (1990), *Robot Visions*. New York: Visual Publications.

Barber, K.S., Martin, C.E. (2001), *Dynamic adaptive autonomy in multi-agent systems: Representation and justification*. International Journal of Pattern Recognition and Artificial Intelligence 15(3), pp. 405-433.

Bradshaw, J. M., Sierhuis, M., Acquisti, A., Feltovich, P., Hoffman, R., Jeffers, R., Prescott, D., Suri, N., Uszok, A., and Van Hoof, R. (2003), *Adjustable autonomy and human - agent teamwork in practice: An interim report on space applications*. In H. Hexmoor, R. Falcone, & C. Castelfranchi (eds.), Agent Autonomy, Dordrecht, The Netherlands: Kluwer, pp. 243-280.

Castelfranchi, C. (1995), *Guarantees for autonomy in cognitive agent architecture*. In: Intelligent Agents, Lecture Notes in Computer Science, Volume 890, Springer Verlag, pp. 56-70.

Dekker, A. (2005), *A taxonomy of network centric warfare architectures*. In Systems Engineering, Test and Evaluation Conference. SETE 2005, Brisbane, Australia.

Dignum, V. (2004), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. Utrecht University, PhD Thesis.

Farrell, P. S. E., Hollands, J. G., Taylor, M. M., & Gamble, H. D. (1999). *Perceptual control and layered protocols in interface design: fundamental concepts*. International Journal of Human-Computer Studies, 50, 489-520.

Jennings, N.R. (2000), *On agent-based software engineering*. Artificial Intelligence 117(2), pp. 277-296.

Powers, W. T., Clark, R. K., and McFarland, R. L. (1960). *A general feedback theory of human behavior: Part 1. Perceptual and Motor Skills*, 11, pp. 71 - 88.

Vecht, B. van der, Dignum, F.P.M., Meyer, J-J.Ch., Neef, R.M. (2007), *A Dynamic Coordination Mechanism Using Adjustable Autonomy*. In Proceedings of the MALLOW 2007 Workshop on Coordination, Organisation, Institutions and Norms (COIN@MALLOW 2007), 3-4 September 2007, Durham, UK, pp. 169-180.

Vecht, B. van der, Dignum, F.P.M., Meyer, J-J.Ch., Dignum, V. (2008), *Organizations and Autonomous Agents: Bottom-up Dynamics of the Coordination Mechanism*. In Proceedings of the AAMAS 2008 Workshop on Coordination, Organisation, Institutions and Norms (COIN@AAMAS 2008), 12-16 May 2008, Estoril, Portugal.