14th ICCRTS
"C2 and Agility"

"Flexible Command and Control Architecture to Achieve Agility"

Topic 9: C2 Architectures and Technologies

I. Pérez, L. Hernández, M. Esteve, C. Palau

Professor Manuel Esteve Domingo

Technical University of Valencia, Spain
Camino de Vera S/N 46022 Valencia, Spain
+34 96 3877305
mesteve@dcom.upv.es

Abstract

SIMACOP is a command and control system for small units in the friendly force tracking systems domain which has self-inspection and autopoietic capabilities to achieve robustness, agility and flexibility in the development of missions as well as to enhance shared situational awareness as a result of global adaptation. The system is composed of several command and control posts located at every hierarchical echelon from battalion down to squad, each one consisting of an application layer acting as user interface, a core module where GIS, database (C2IEDM data model) and self-state monitoring and reconfiguration capabilities are located and a hardware abstraction layer which isolates previous components from underlying networks. Previously remarked capabilities are achieved by means of having an inter and intra node software architecture that monitors its state and the state of the environment as well as user inputs and reacts modifying each individual node and reconfiguring (in a distributed and self-organizing memory-less approach) the global network and information flow architectures. Both intra and inter node levels enforce maintenance of the overall system structure (defined as a goal-oriented mission file which can be dynamically modified) in an autopoietic way thus providing robustness and agility.

Keywords: autopoiesis, agility, flexibility, robustness, resilience, command and control, Friendly Force Tracking

1. Introduction

The main goal in the command and control loop is to enforce the achievement of effectivity in the accomplishment of a mission in an effects-based approach of operations. Such an effectivity must be enforced or supported in several ways, as stated in Dr. Alberts works[Alb02][Alb03][Alb06] but fundamentally by following the Network Centric Warfare (NCW)[Alb99][Alb00] approach and principles. In the tenets of NCW presented in [Alb02] it is outlined:

- A robustly networked force enhances information sharing among its constituents.
- Information sharing and collaboration enhance the quality of information as well as shared situational awareness.
- Shared situational awareness enables collaboration and self-synchronization as well as agility. This leads to an improvement in sustainability and speed of command.
- These, in turn, dramatically increase mission effectiveness.

On the other hand, is difficult to find an agreement on the precise definition of the term as there are several definitions of the concept. As stated by Alberts [Alb03] and Atkinson [Atk05], the concept of agility refers to its capacity to adapt to a dynamic environment. As it is stated in Power to the Edge [Alb03] agility does not just come by itself, it has to be boosted by means of the right organization structure, operational concepts, associated systems and trained personnel and, only the synergy of those factors can lead to its achievement.

In the same work it is outlined that agility has six main attributes:

- Robustness: the ability to maintain effectiveness across a range of tasks, situations, and conditions.
- Resilience: the ability to recover from or adjust to misfortune, damage, or a destabilizing perturbation in the environment.
- Responsiveness: the ability to react to a change in the environment in a timely manner.
- Flexibility: the ability to employ multiple ways to succeed and the capacity to move seamlessly between them.
- Innovation: the ability to do new things and the ability to do old things in new ways.
- Adaptation: the ability to change work processes and the ability to change the organization.

It can be seen in the literature the fundamental role that plays agility in the achievement of mission effectiveness. As an example it can be seen in the works of Dod et al. [Dod06] and Dekker [Dek06a] as well as in experimental works such as McEver research [Mce06].

Then, a crucial feature of every command a control system that follows the NCW principles is the encouragement of agility at all levels.

As we have seen agility is associated with terms as adaptation to the environment, innovation and flexibility but, on the other hand, never letting the overall system loose its fundamental features and structure, that is where concepts as robustness and resilience play their role. Then, to fulfil with the agility principle, a system must be flexible enough to adapt itself as well as robust enough to avoid environment disturbances with a component of self-modification and self-configuration to cope with both requirements.

Then, as a usual approach in latter times in engineering, one may find it some sort of *zeitgeist* or the spirit and trend of the time, we can pick concepts and ideas from what is envisioned nature does, developing the so-called biologically inspired systems [For04]. In particular, some theories about what living beings are and how they survive can be useful to characterize what we need to achieve such agility whilst keeping system consistency. Those theories are basically all the related to self-organizing or configuring emergent structures that try to characterize what the living is [And72] [Crut94] and how to apply it to engineered systems [Bon99][Bro91]. In particular we have focused on the concept of autopoiesis.

Autopoiesis is a neologism introduce in 1973 by Chilean biologists H. Maturana and F. Varela [Var73] [Mat80] which literally means "self-creation" from the Greek words auto(*αυτό*):self and poiesis(*ποίησις*): creation or production.

> "An autopoietic machine is a machine organized (defined as a unity) as a network of processes of production (transformation and destruction) of components which: (i) through their interactions and transformations continuously regenerate and realize the network of processes (relations) that produced them; and (ii) constitute it (the machine) as a concrete unity in space in which they (the components) exist by specifying the topological domain of its realization as such a network." ([Mat80], page 78)

The term was originally conceived as an attempt to characterize the nature of living systems. A typical example of an autopoietic system is the eukaryotic cell which is made of various biochemical components such as nucleic acids and proteins, and is organized into bounded structures such as the cell nucleus, cell membrane, several organelles and cytoskeleton. These structures, based on an external flow of molecules and energy, produce the components which, in turn, continue to maintain the organized bounded structure that gives rise to these components.

A related and complementary concept would allopoiesis. An allopoietic system is one that produces something different from itself. An example of such systems can be a factory which uses raw materials (components) to generate a good (an organized structure) which is something *other* than itself (the factory).

In a more general approach the term autopoiesis can be assimilated to the dynamics of a non-equilibrium system of the general theory of systems [Wie48]. In such a model it can be seen as a set of organized states (some authors call them dissipative structures [Pri77]) that remain stable for long periods of time despite a continuous flow of matter and energy through them. Other features of those systems would be that every process within the system directly helps maintaining the rest and systems are structurally coupled with their medium following its dynamics and creating new ones within it due their course of actions.

Autopoiesis is quite related to the concept of complex adaptive system (CAS) but there are some slightly differences among them. An autopietic system is a complex adaptive system that has very strong self-monitoring mechanisms that allow itself to keep its main state and features whatever changes happen on the environment and on itself due to the adaptation-to-the-changes process. Then the key point is the self-state keeping capabilities.

2. Related work

There has been a lot of work about autopoiesis in the theoretical domain. Best examples are the works of Maturana and Varela such as [Mat98] [Var91] [Var99], as well as previously cited work, where the authors present their ideas and extend their concepts.

In the social sciences domain, the theories of Niklas Luhmann [Luh89] [Luh95] have been widely spread and have had a considerable acceptance. The author remarks that autopoiesis is not an exclusive feature of physical systems and extends the work to cognitive and social domains. On of its main contributions is its effort to explain self-organization and self-production in human societies confronted to risky and contingency situations.

Based on the seminal work on Luhmann, in the theory of organisations area of research there has been some work to model organisations as autopoietic entities. One can find studies as [Mur97] for strategic management, [Teu88] for a systemic approach of law structures, [Buc06] for an approach more related to politics, among others.

In a more technological area, focusing in the computation and communications fields several works can be found. For instance, in the artificial intelligence and cognitive areas, there are researches that try to define paradigms where cognitive processes do not start from a representationist point of view but from the autopoiesis concept, as can be seen in the work done by Bitbol et al. [Bit04].

There are very interesting studies from Belgian researcher Luc Steels where sets of robots self-organise themselves to achieve a common goal (for instance to cluster objects randomly distributed) and where a common communication 'language' emerges and is used by those agents to achieve their mission. Such a structure can be seen as an autopoietic system [Ste96a] [Ste96b].

In the area of agent systems there are several works that explore the concept application. One can point out the approach of Di Paolo et al. [Dip97][Dip05], the STARCAT architecture [Lew06] designed to develop autopoietic software frameworks, or multi-agent architectures where agents have autopoietic capabilities as it can be seen in the work of McMullin y Gross[Mcm01].

The key concept in the following surveyed works is to treat software, and thus develop, use and maintain it, as a living autopoietic organism. That can be seen in the work of Gabriel and Goldman[Gab07] from Sun Microsystems, who point out that software systems are constructed nowadays as:

“A finished product [that] is constructed at the factory and shipped to its final destination where it is expected to act like any other machine—reliable but oblivious to its surroundings and its own welfare.”

They claim that software systems should monitor their state and their environment and adapt to it, catch error and recover from them and configure and customize themselves depending on conditions. Authors propose a shift from paradigms and locate their framework inspired in biological concepts, mainly autopoiesis.

It can be seen that this is the approach taken by some real software and system implementations. It is the case of some Google infrastructures, as it is pointed out in [WEB1]. As authors show, there are several Google systems, such as their file system (know as Google File System), the BigTable large-scale database and the Chubby lock system (for distributed resource access locking) which exhibit those autopoietic features. Systems are highly distributed and one of the main goals of their design is to provide a high degree of fault-tolerance. Cited systems monitor themselves, replicate data among failed components and self-manage on top of a pool of potentially faulty servers. As it is remarked in [WEB1], one can see those systems as 'implementing a metabolic environment for user programs that is composed from cells (servers) that routinely die and are replaced, while the entire system has a dynamic, self-sustaining continuity.'

There is an initiative started by IBM in 2001 [IBM1], called autonomic computing, that aims to develop computer systems capable of self management. They are considered to make decisions on their own, based on high-level policies, they are constantly supervising their status and the environment and adapting to conditions. As a real product, IBM has developed WSDM (Web Service Distributed Management) standard for Service Oriented Architectures. Starting from that initiative, several frameworks have been proposed, projects like CASCADAS [CAS] and SOCRATES [SOC] as well as for autonomic networking (concept considered to networks that monitor and adapt) like projects Autonomic Network Architecture [ANA] or Bisnet [BIS].

In the command and control research community only in very few works the concept is outlined. We can find it in [Lee07] where there is an approach to a methodology to engineer complex systems with strong human interaction and where the autopoiesis concept is highlighted as a consequence of the constructivist philosophical approach confronted to the classical, positivist approach, and more related to the NCW sensemaking concept. On another research carried by Allen et al. [All06], knowledge and sense making as a consequence are seen as autopoietic structures since knowledge impacts the interpretation that defines the knowledge itself. This approach follows a previous work done by Von Krough [Von98] where the self-constructivism view is developed.

There are also other works, like [Ste07] where there is a characterization of a conflict and its dynamics like an autopoietic system. In works like ‘complexity theory and the network centric warfare’ by Moffat [Mof04] terms such as far from equilibrium dynamic systems are used to explain and characterize concepts as a conflict or the self synchronization of a force but there is no trace of the autopoiesis concept, which in our opinion is a 'one step beyond' from others used in that work as 'complex emergent behaviour'.

Anthony Dekker is another author that has explored the concepts of emergent and self-organizing systems related to the discipline of command and control and networking (see for instance [Dek06b][Dek06c]) but never has pointed out the need for an autopoietic system as an extension of a reflexive, self-healing system.

3. Design approach

Once outlined the main concern, enforce agility, and some design principles we want to start from, autopoietic systems framework, we can show our approach to obtain those desired goals. The aim of this work has been to adapt existing autopoietic frameworks and strategies to the technical part of command and control systems as we consider that they can enhance considerably agility. Also we have found a gap in the literature in such area. This approach is taken in wider command and control system called SIMACOP, which stands for small units command and control system in Spanish, which will be explained in following sections.

What we want is to benefit from the possible outcomes that arise from approaching the system design as a complex adaptive system, particularly an autopoietic one, so our design and the final system will exhibit the features related to those, that is: self adaptation to environment changing conditions without loosing its main state, features and patterns.

In the following figure we can see the classical command and control loop based in control and systems theory

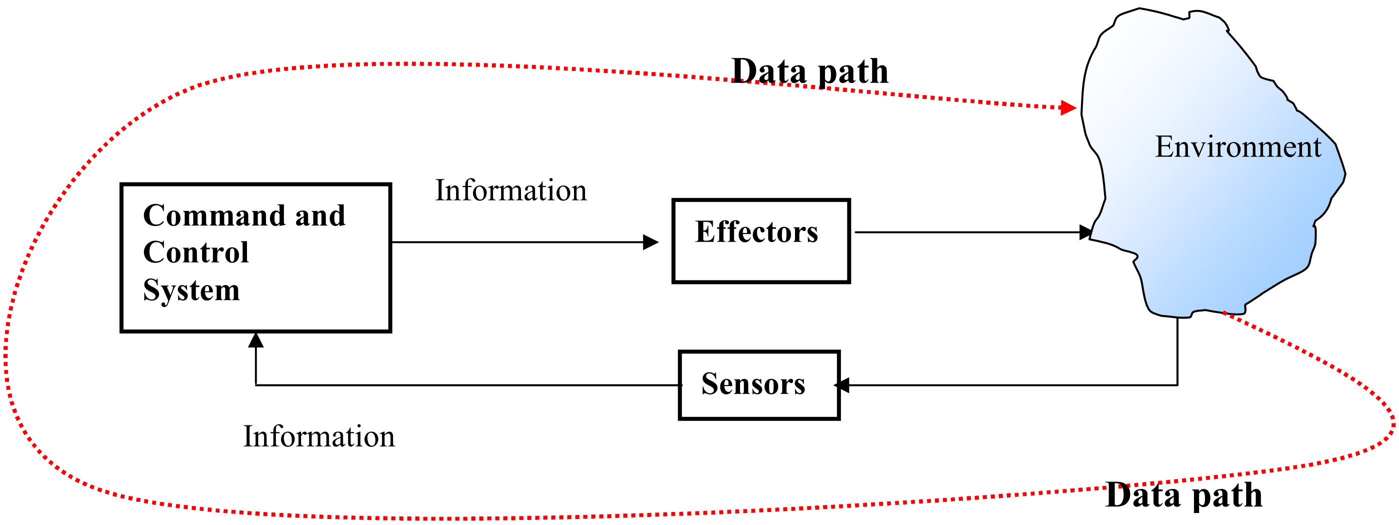

Figure 1: Basic command and control data paths

Starting from the previous classical loop seen in the previous figure, we introduce the concept of an autopoietic NCW system as one that is made of various information components and is organized into bounded structures such as networks devices, computers, vehicles, human beings and so on. Those structures, based on an external flow of information produce new information components, continue to maintain the organized bounded structure that, in turn, gives raise to the components (information structures, decisions, commands, effects, etc). In this case the goal that drives and directs system behaviour is two-folded: to maintain structure and to adapt to changes but in both cases limited (and complementarily directed) by the overall goal: mission effectiveness. Thanks to the approach stated we consider that our developed system enforces and supports agility to a higher degree.

The classical command and control loop, or modern interpretations as those from Dr. Alberts[Alb06] can be extended as the command and control system is modified during the course of an action at all levels (organizational, procedural, technical, ...) in order to adapt to environment changes rapidly gaining agility.

In this model there are going to be two kinds of data paths: autopoietic and allopoietic data paths. Those data paths are going to be information flows that traverse the command and control loop from the inputs taken from the environment to the outputs of the system. Both data paths are two-folded inter and intra node as information is going to be consumed/produced inside each of the systems node and at the overall nodes scope. The first data paths are going to be those used by processes and applications, inside of the system, in charge of maintaining the overall state and adapt to new environmental conditions in a two fold approach, intra-node and inter-node, and the second data paths are going to be the 'real' data for the command and control system and applications. In our approach, processes in charge of maintaining the state (thus using the autopoietic data path) are going to monitor the environment and the system state and, depending on the needs, modify parameters and even running code from the applications that use the allopoietic data path. The applications that use the allopoietic data path ignore the details of underlying resources they use as well as the modification that can happen to their running code, they just consume the first and execute themselves residing on top of the autopoietic data path but isolated, or not needing to know, about it. It is worth to notice that, depending on the nature of the scenario of operations and the operation itself, allopoietic information may become autopoietic or at least influence the self-regulatory behaviour. That can happen when sensed information about the scenario can trigger not only the operators' interest but also the homeostatic mechanisms.

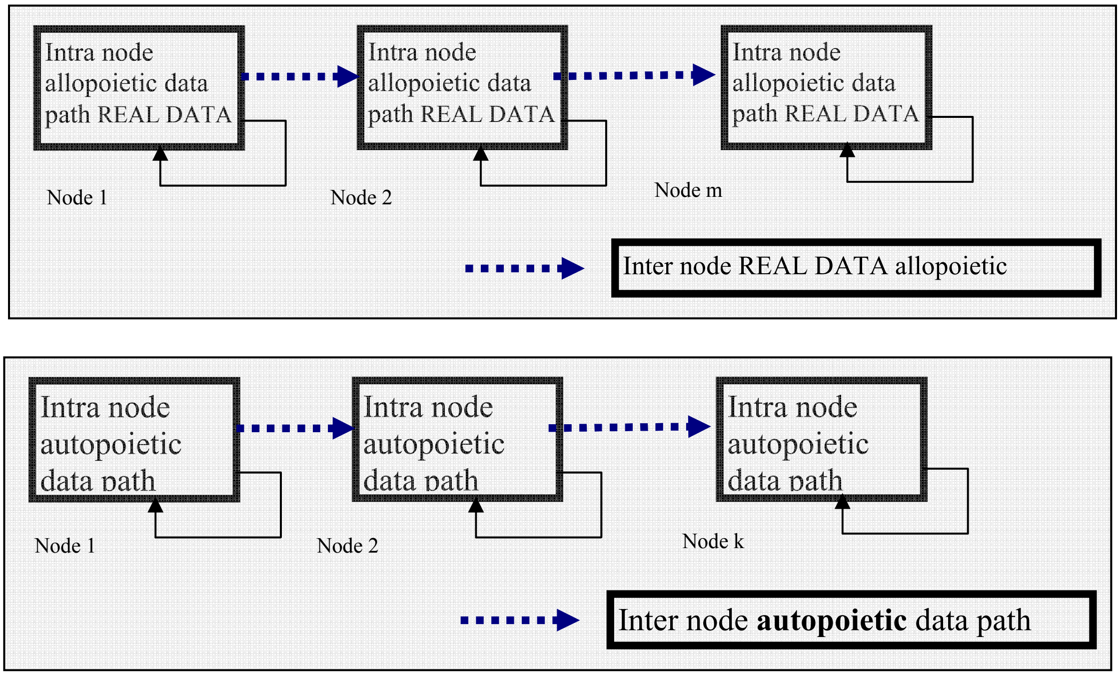

Figure 2: Autopoietic and allopoietic data paths

In the previous figure both data paths can be seen. All of the start from information generated/gathered at the environment, traverse the command and control system and then back to the environment. In the case of intra node data path (both for the allopoietic and autopoietic) we are considering information extracted from the environment, processed and sent to the environment (as an effect) only inside the node's scope, not affecting other nodes. In the case of inter node data path (for both allopoietic and autopoietic) that is information that traverses more that one node s the interaction of them is required for a given process or action.

Obviously there must be trade-off among flexibility and system predictability and performance. Systems that follow this approach must pay attention on the fact that the searched flexibility cannot penalize system behaviour determinism to an extent that makes it useless or far away from its intended operational ranges.

In the following sections a fielded command and control system for small units called SIMACOP is presented. The system has been developed with the described philosophy in order to achieve the desired agility. On the other hand the system has more capabilities than just autopoiesis and has been developed with NCW concepts in mind to enforce self-synchronization, 'power to the edge', information superiority, 'network-to-the-sensor', etc.

4. Proposed architecture and developed system

4.1 Overall description of the system

SIMACOP is the Spanish acronym for small units command and control system is a fielded and operational command and control system developed at the technical University of Valencia, Spain. It is a very tactical system, located in the gap existing between Friendly Force Tracking Systems (FFT) and Battlefield Management Systems (BMS). In fact, it is a fully featured FFT with some of the capabilities of a BMS. The system is currently in use by the Spanish Army and has been tested as a FFT interoperability node at the NATO Coalition Warrior Interoperability Demonstration (CWID)'07 and CWID'08 exercises by means of the NFFI (NATO Friendly Force Information) 5527 STANAG usage.

The system has the following architecture:

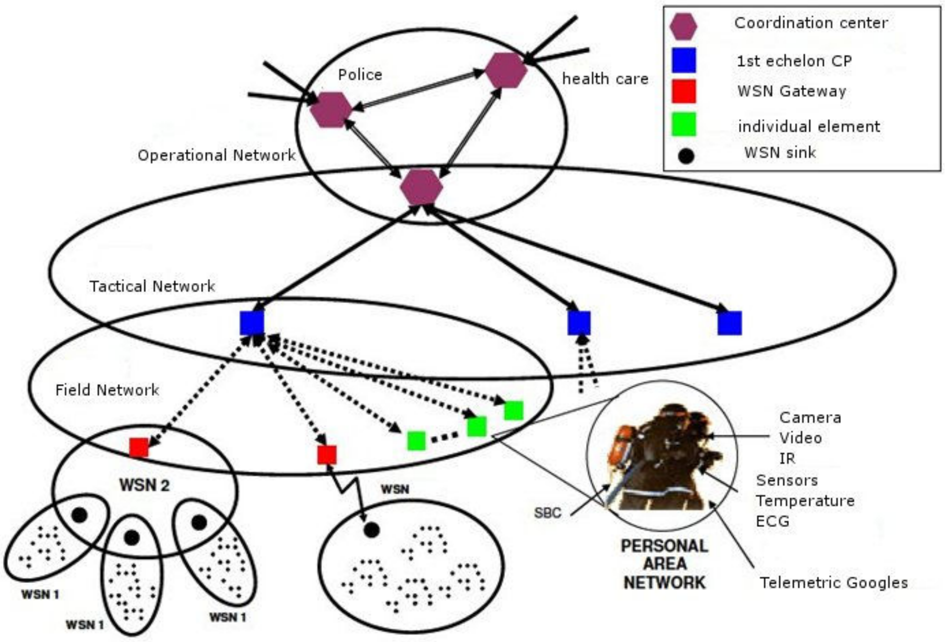

Figure 3: Overall architecture

It can be seen that there are some nodes grouped in several sub-networks and located in vehicles which constitute the basic elements of the architecture. Those nodes have the FFT (Friendly Force Tracking) command and control application and interchange data in order to share the same common operational Picture (COP). There can be also dismounted nodes for individual soldiers that currently have two versions: Single Board Computers (SBC) and PDA.

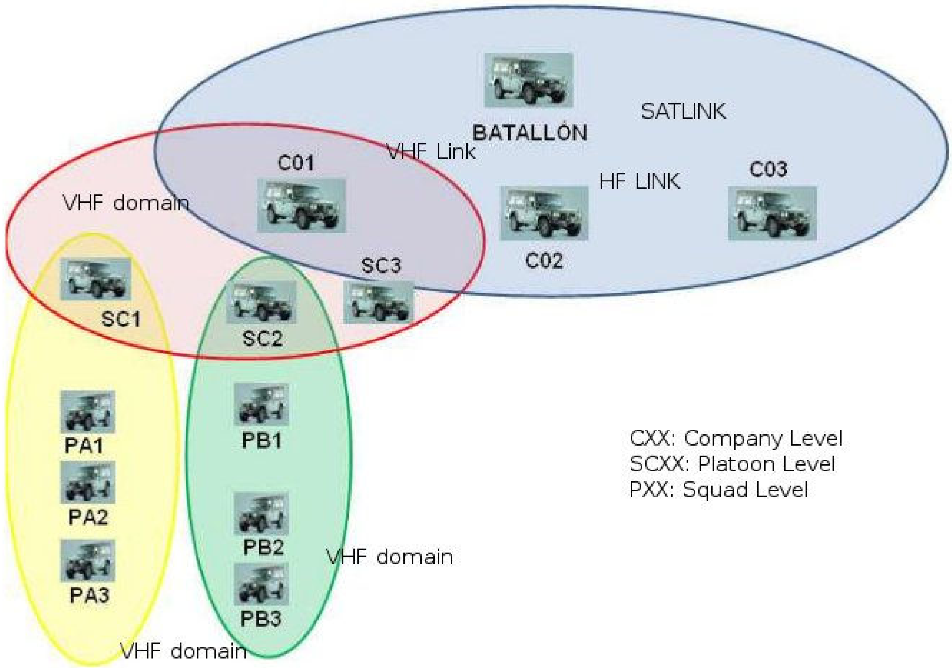

Figure 4: typical deployment

Basic features of the system are:

- Friendly forces location and tracking capabilities
  - Command and control capabilities from battalion level to the dismounted soldier
    - Alarms
    - Threats introduction and broadcasting
    - Short messages in tactical chat format
    - Objects marking and broadcasting
    - ORBAT (ORder of BATtle) Generation
    - Units filtering
    - Interoperability via NFFI (IP1, IP2 and SIP3 SOA (Service Oriented Architecture) based profiles) with NATO Land Track Interoperability Service (LTIS) and others FFT.
    - Interoperability with SIMACET (Spanish Army Great Unit Command and Control System) via COE (Common Operational Environment).
    - C2IEDM (Command and Control Information Exchange Data Model) and MIP (Multilateral Interoperability Programme) Block II compliant
- Multimedia capabilities: basically the inclusion of high-quality real-time video flows (where the link bandwidth allows as, for instance with some satellite links and UHF mesh networks).
- Integration of any kind of sensors:
  - Video
  - Infra red (IR)
  - Biosensors
  - Telemetric binoculars
- Compatible with any kind of GIS and maps formats:
  - SIGMIL (Spanish Army Cartography)
  - Any kind vector formats
  - Web Map Service (WMS) and Web Feature Service (WFS)
  - Satellite images

Being a very tactical system, its user interface is extremely simple and intuitive, can be run on tactile displays and every action is far from another by three clicks, as much.

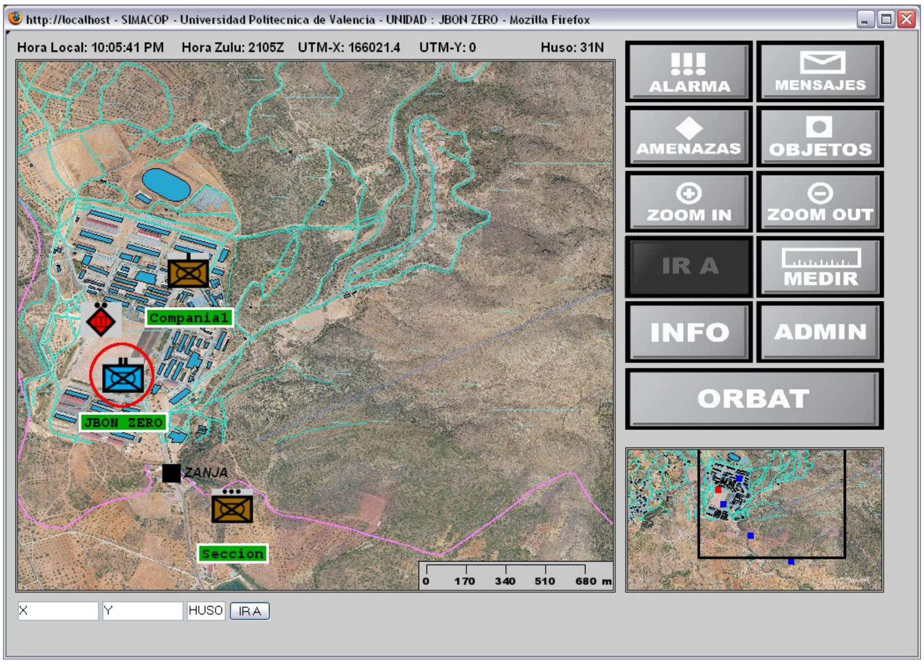

Figure 5: Current application

The system can be used with the most important tactical radio means nowadays in operation, both military and civil. We can point out in the VHF radios segment the PR4G V3 F@snet radio and PR4G V2 radios. In the HF segment it can use the Harris 5800 Radio and the Thomson HF Radio. Satellite means Inmarsat, Iridium and Thuraya as well as satellite means TLX-50 and TLX-5. Also it has been tested with Tetra and Tetrapol means as well as Alcatel/Lucent WiMAX technology and ITT Spearnet Personal Radio in the UHF mesh domain.

4.2 Autopoietic aspects of the system

Flexibility, robustness and resilience are acquired by the system due to its autopoietic features. As it was pointed out in previous sections, this goal is achieved in an interrelated two-layered approach, intra-node and inter-node. Inter node reconfiguration is achieved by means of network topology and features changes, and intra node reconfiguration is achieved by means of changes in the code of the application that runs on each node to reflect the changes in network architecture as well as other system attributes.
We will consider that that our system is an autopoietic machine in the sense that it is continuously regenerating and transforming itself (due to the effort on adapting to environment conditions) whilst keeping some basic patterns or skeleton. There are going to be two main domains or entities where these concepts take place: at the components of a node (or intra-node autopoiesis) and at the network level (or inter-node autopoiesis), so we are going to consider that the software at a particular node is an entity that regenerates itself and transforms due to adaptation to the environment whilst keeping its basic structure (intra-node) and, on the other hand, we are going to consider the overall network as an entity that regenerates itself and transforms due to the adaptation to the environment whilst keeping its basic structure. This is going to be explained in the following points, with detailed examples.

4.2.1 Intra-node self-configuration

In the intra-node approach, there are mechanisms on every node that inspect its state, environment conditions and user inputs and, depending on local policies modify local configuration and even their own or other SIMACOP applications running code. On the other hand, on every node there are stand-alone watchdog processes that monitor the state of other crucial processes (such as video server, GPS server, communications infrastructure, etc.) and, if a strange behaviour is detected, the component is returned to a consistent state. Thus, these monitoring processes try to adapt intra node parameters and applications to changing conditions as well as keeping fundamental processes alive and consistent. The node elements of

SIMACOP react to external inputs keeping their main structure. The autopietic data path consists of the information regarded from outside and processed by previously stated processed and afterwards transformed in effects that modify parameters or the running code. On the other hand, the allopoietic data path is the real data that the command and control application processes and shows to the human operators, for instance GPS positions gathered and displayed at the GIS of the command and control application.

In this approach two solutions have been developed, one for the dismounted system and another for the vehicular system. In the first one the system runs over an embedded hardware (SBC or PDA as stated previously) which has to be kept in operation with no human intervention and recover from failures (power restarts, network disassociation, etc) autonomously.

In such a case there are two levels of operation: operative level and code level. In the operative level there are several processes that inspect and monitor the state of the crucial processes (communications service, video server, GPS server, and bio signals server) and evolution of the data flows from and to other nodes.

That can be seen in the following figure:

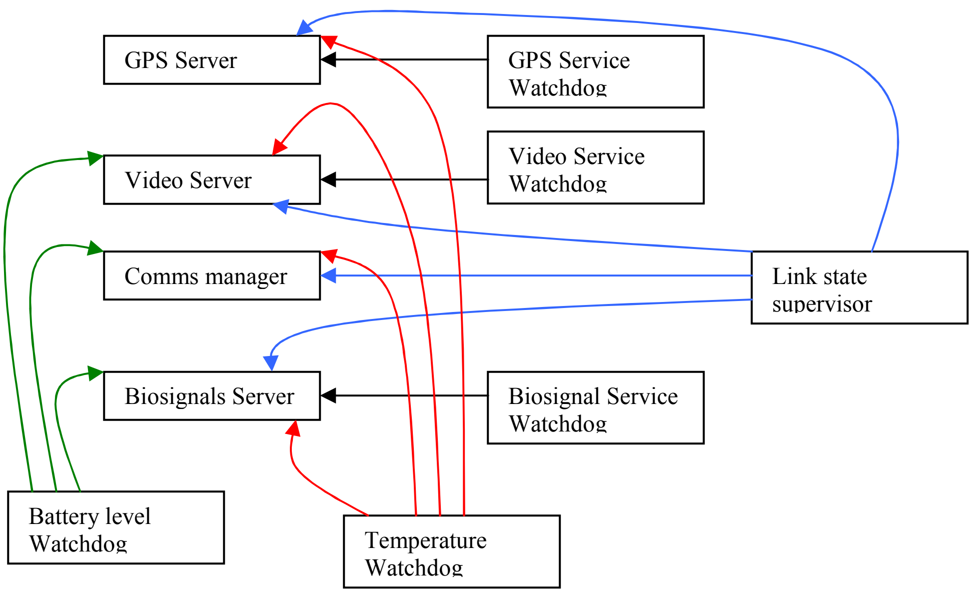

Figure 6: Operative level process management

At the code level there are processes that detect modifications on the environment and modify executing code in order to adapt. This is the case, for instance, when network link state is monitored as experiencing some modifications or problems. Other example is when some environmental conditions are detected to be changing (such as link state or video parameters like illumination or movement) and the video server reduces its kilobit per second rate. Another example can be seen when the SBC approaches to a previously deployed sensor network, it can communicate via ZIGBEE (802.15.4) with its nodes and reconfigure their structure and routing following and energy-consumption minimization function. In such cases the system rewrites some parts of the code in order to adjust. It has to be remarked that in the dismounted solution (both SBC and PDA) there is no compiler so the code that is rewritten is interpreted code.

In the vehicular version there are also processes that inspect the correct operation of sensors and restart server in order to setup them in a consistent state in case of malfunction. On the other hand, the on-the-fly code rewriting process is more relevant in this solution as these nodes are the ones that implement core functions of the inter-node and intra-node self-configuration processes. Then, the system reacts to monitored inputs (environment or user) in order to gain efficiency and flexibility and the process leads to a more autopoietic system that if just configuration parameters modifications where done.

For instance the code rewrites itself in order to switch from one GPS configuration to another (depending on user inputs or an incorrect GPS received data), to switch from one communication mean downlink or uplink depending on three different factors: user inputs, monitored link state or received from other node overall architecture reconfiguration. The last one will be explained in the following section as it fails in the inter-node autopoiesis case.

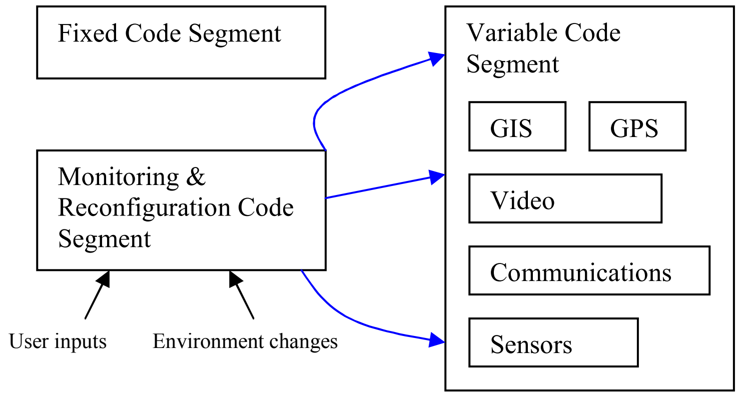

Figure 7: intra-node code rewriting process

4.2.2 Inter-node self-configuration

The main idea behind inter-node autopoiesis is to keep the overall network structure consistent as well as adapting to outside inputs. In our opinion, the overall network structure means that nodes connectivity must be kept providing main system features such as message exchange, Common Operational Picture sharing and so on, during and after a network rearrangement due to an adaptation to the environment changes. Those can be the insertion of new nodes, removal of nodes, changes on links among nodes and so on. In this case the autopoietic data path is all the information that, starting from out-of-the-system gathered data provokes information exchange among nodes that produces effects such as topology reconfiguration. The allopoietic data path is all the other information that flows inter-node such as a tactical chat message exchange among two command and control posts. In the inter-node autopoietic approach there are several mechanisms to modify overall nodes configuration and network setup. The core of the SIMACOP system is a small chunk of information (typically 40 Kbytes) called Mission File (FDM due to the acronym in Spanish) where all the specifications of the operation hierarchy and network setup, as well as messaging and cartography of a given operation are described. Then, in order to reconfigure network state, this core information has to be consistent among nodes so exchange procedures have to be defined.

Following some detailed examples of network reconfiguration are detailed. In the first example, node network joining is considered. When a new unit joins the mission, the node needs to know minimal information either the current network architecture by means of the current mission file or at least the contact information of its superior unit.

First step in the process is contacting the superior unit to register into the mission (ORBAT_JOIN_MSG: ORBAT join message) as shown in figure 7, the superior unit receives that request and validates the subordinate trough a PKI (Public Key Infrastructure), after successful validation, superior unit updates its databases with the new subordinate unit and sends the modified mission file to the subordinate (ORBAT_UPDATE_MSG: ORBAT update message) as well as to its superior unit (in case it exists). The superior unit will also update its database and forward the ORBAT_UPDATE_MSG to the rest of its subordinate units, these subordinates units will apply the changes to its databases and forward the ORBAT_UPDATE_MSG to its subordinate units as well, this way the changes will spread across the network (figure 8).

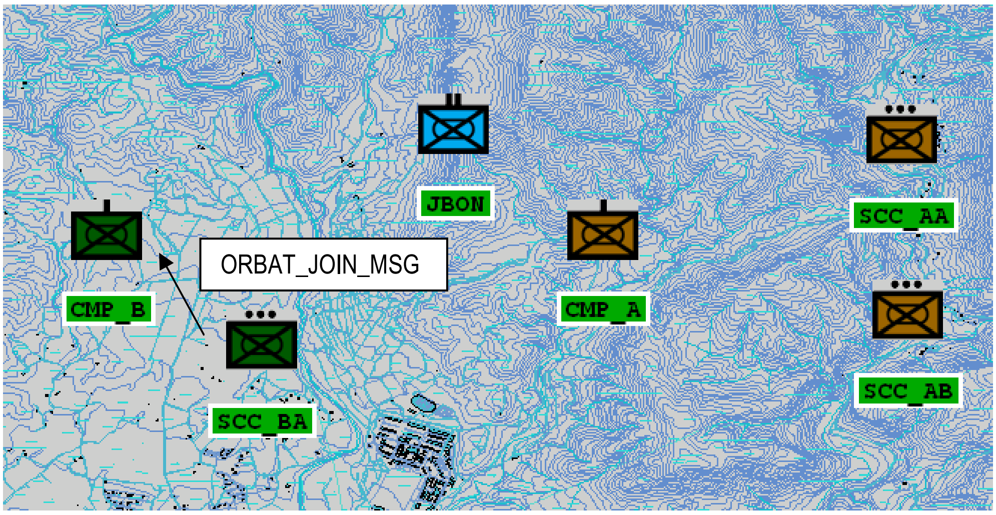

Figure 8: A new unit joins the operation

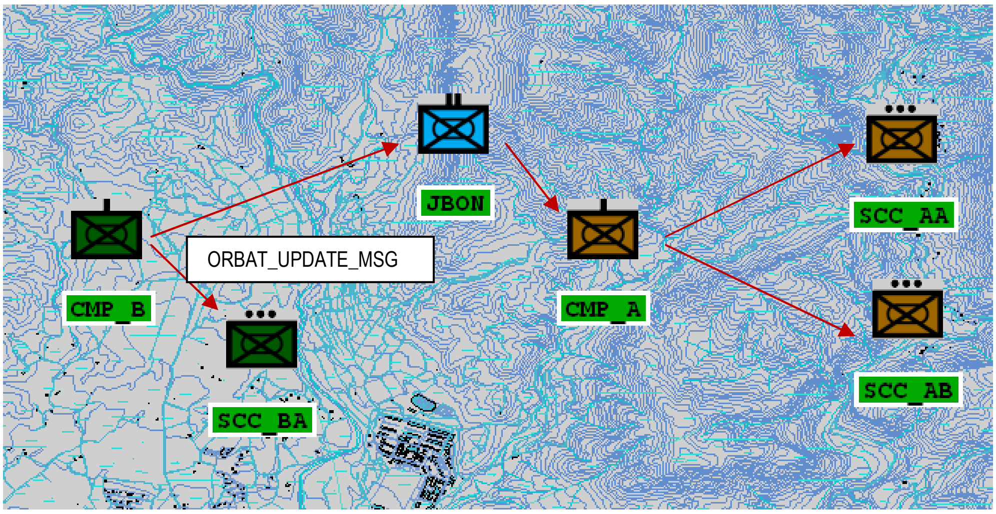

Figure 9: Superior unit updates the ORBAT

When all subordinates units in a level have updated the information they will send an ORBAT_APPLYOK_MSG (ORBAT apply changes OK message) to the superior unit (figure 9). On reception of this message by the global coordination unit (GCU) from all its subordinates this will mean that every node in the network has updated its local databases with the new unit information so the GCU floods the network with an ORBAT_UPDT_MF_MSG (ORBAT update mission file message). So far changes were made at DB level but not to the application itself, so when a node receives this message intra-node reconfiguration comes into play, application will reread the DB and generates the code to make this new unit visible on the GIS, in case the node has no subordinates it will send a ORBAT_APPLYOK_MF_MSG (ORBAT apply mission file changes OK) to the superior unit and in case it does have it will forward the ORBAT_UPDT_MF_MSG to its subordinates and wait until it receives the ORBAT_APPLYOK_MF_MSG from all of them before reporting successful update to the superior unit. When the global coordination unit (GCU) receives this message from all its subordinates this will mean that every node in the network has updated and rewritten its application, from now on the unit is fully operative in the network. (Figure 10)
After this point, the overall network has reconfigured to adapt to external inputs as well as information flow between nodes has not been disturbed by this modifications.

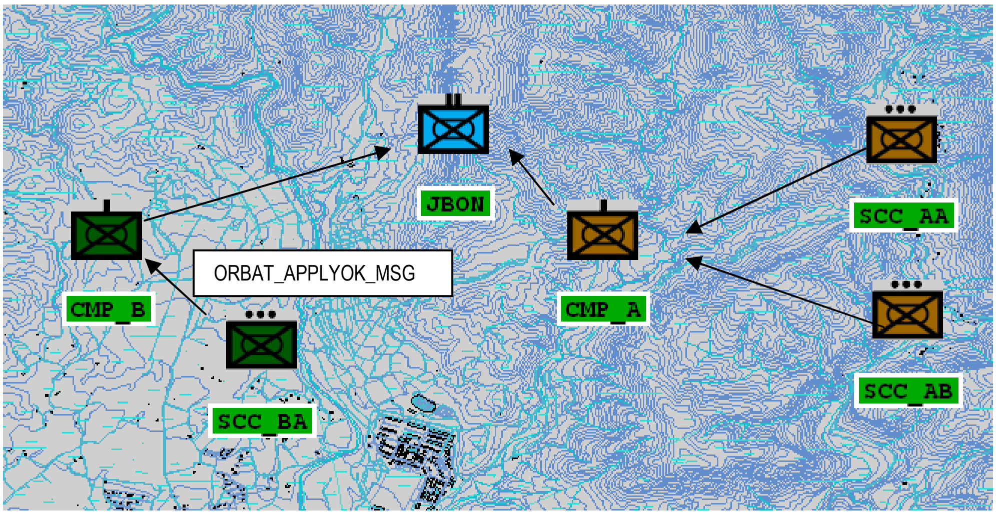

Figure 10: Acknowledge of ORBAT changes

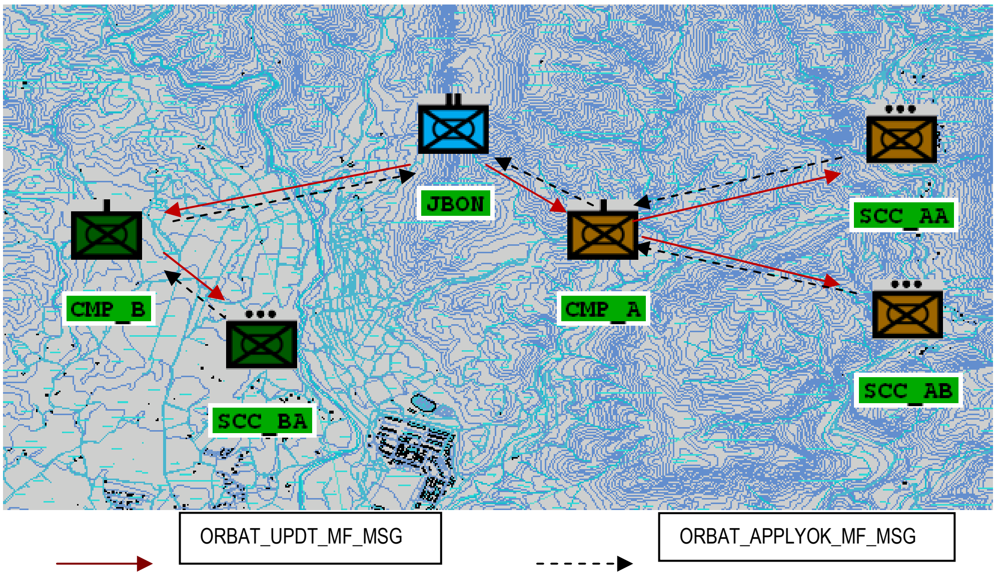

Figure 11: Mission file redistribution and intra-node self-configuration

In the case of link state modification, inter node reconfiguration will occur but we have to differentiate between two cases: a) a node wants to access the network with a different transmission media previously declared on the mission file, b) a node wants to access the network with a different transmission media not previously declared on the mission file.

Regarding first case, SIMACOP application monitors the state of the transmission media; in case of link failure it will use the following backup media to contact the unit. Each transmission media gets assigned a weight in function of its data rate, transmission bandwidth and duplex mode, the larger the weight the better the channel is for data communication. In addition to this each node maintains a transmission matrix with all active media transmission with its neighbours and the possible link backups in case of failure organized by weight.

Whenever there is a link failure the node will try to contact the neighbour trough the next available transmission media by sending a CHG_TX_MED_REQ (Change Transmission Media Request) request and will wait for TX_CHANGE_RES (Transmission change response) response. The receiving node will update its database, restart the replication services towards this unit and acknowledges the message with a TX_CHANGE_RES response, from now on data communication and replication data will be sent through this channel. If the node fails to respond to the CHG_TX_MED_REQ, the unit will try the rest of available media transmission in a round robin way until it gets a response in any of the available media.

Figure 11 illustrates the process.

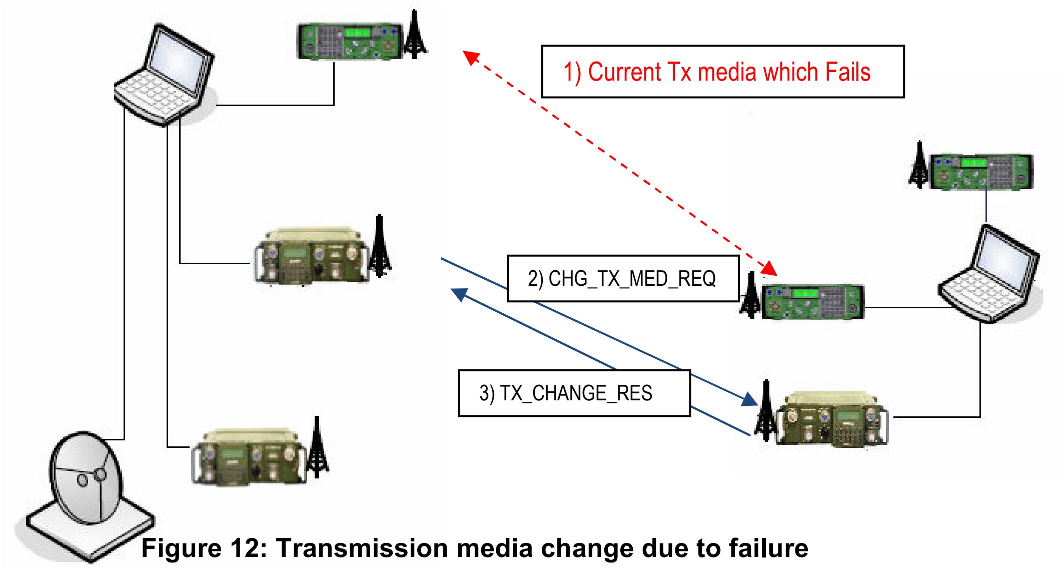

Figure 12: Transmission media change due to failure

Inter node reconfiguration will also occur when a unit wants to access the network with a different transmission media not previously declared on the mission file, the operator will configure all necessary details of this new transmission media like IP address, connection name, etc. This will update the local database.

In order to continue the process, we have to differentiate between two cases, if the unit is changing its transmission media either towards its subordinates or its superior node. In the latter, the node sends a JOIN_NET_REQ request through this new configured media, upon reception of this request the superior unit updates its database, restart the replication services towards this unit and acknowledges the request with an ACCEPT_JOIN_RES response, from now on data communication and replication data will be send through this channel.

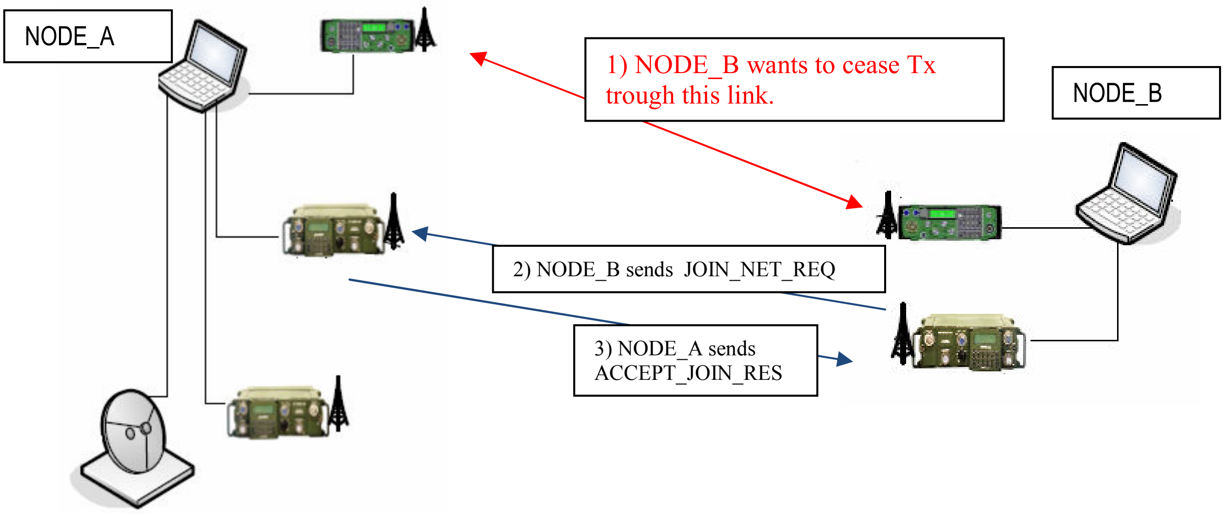

Figure 13: Change of transmission media on subordinate site

In the former, provided that at least one of the subordinates also have a data radio on this new communication channel, the superior unit will send an UPDT_NET_MSG message to its subordinates, the receiving node will update its database, restart the replication services towards this unit and acknowledges the message with an APPLYOK_NET_RES response, from now on data communication and replication data will be send through this channel. This change in the network configuration will only apply to the affected nodes; it will not be spread over the entire network, thus reducing bandwidth. Figure 13 illustrates the process.

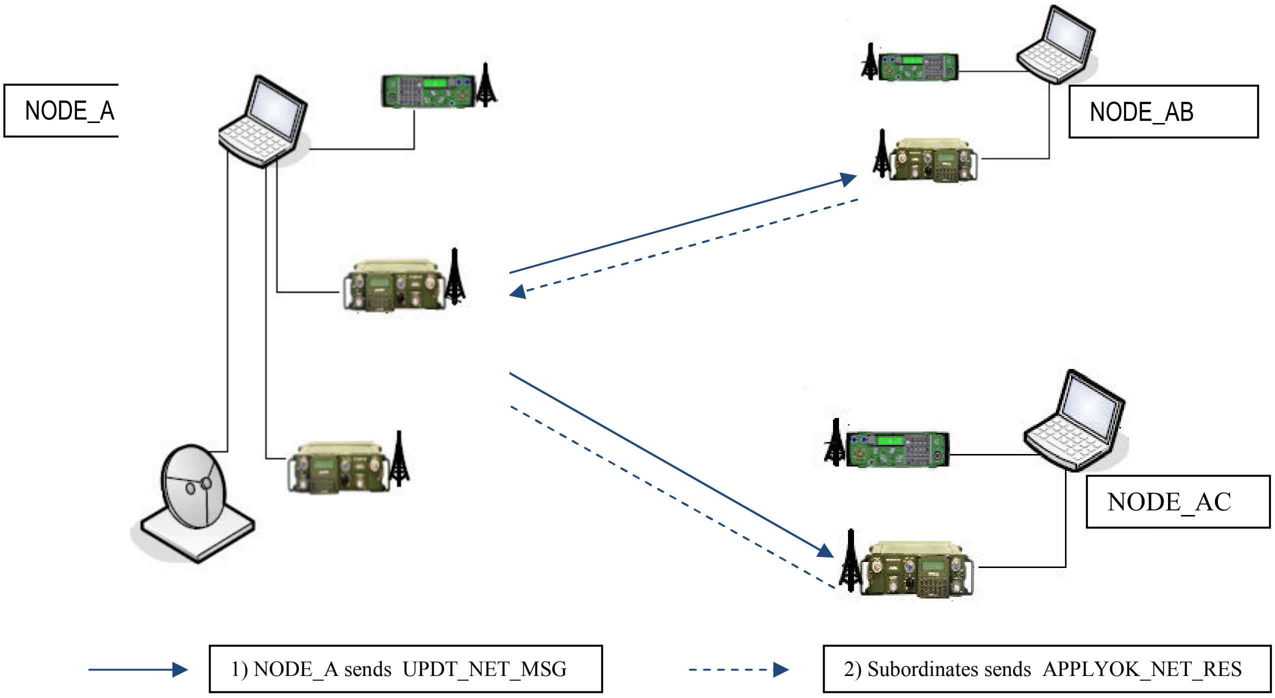

Figure 14: Change of transmission media on coordination site

This last example reflects the idea behind the autopoietic approach, system self-configuration and adaptation at the network level whilst keeping network consistency.

On the other hand, in some of the system deployments both dismounted and vehicular solutions use UHF mesh networking solutions as communications means. In such a case, the system benefits from their reconfiguration and self-organization features and enforces such behaviour. For instance, mesh radios have ISO/OSI layer 2 MAC mechanisms to have dynamic clusters of nodes and act as relays for buddies when needed. SIMACOP system enforces such behaviour by means of the usage of dynamic routing algorithms as well as neighbour routing tables.

The mechanisms described can be considered as complex adaptive systems but, in our opinion go one step beyond as, at any time, they enforce the basic structure keeping while regenerating themselves in order to adapt to the changes.

5. Results and future work

The validity of proposed autopoietic architecture has been proved as the system has participated in many Spanish Army manoeuvres and currently is in a fielded state and presented features have helped the system to be more agile in the desired dimensions (flexible, adaptable, robust and resilient). Also the system has been used successfully in NATO exercises for the past two years.

- Current system does not exhibit full autopoietic capabilities so we intend to extend it by means of:
- Full integration with self-configuring sensor networks
- Cartography mission changes on-the-fly
- Higher degree of system auto-inspection and reflexivity.
- Evaluation of the feasibility of agent architectures to enhance and maximize the software modules independence and interaction.

References

[Alb99] David S. Alberts, John J. Garstka, and Frederick P. Stein, “Network Centric Warfare”, US Department of Defense Command and Control Research Program publication, 1999

[Alb00] Alberts, D.S., Garstka, J.J., Stein, F.P., “Network Centric Warfare: Developing and Leveraging Information Superiority”, CCRP Publ., 2nd Edition (Revised). Aug 1999, Second Print Feb 2000

[Alb02] Alberts, D.S., (2002), Information Age Transformation: Getting to a 21st Century Military, Washington, DC, CCRP Publications. pp. 7-8. First published 1996.

[Alb03] David S. Alberts, R.E Hayes, “Power to the edge”, US Department of Defense Command and Control Research Program publication, 2003

[Alb06] David S. Alberts, R.E Hayes, “Understanding Command and Control”, US Department of Defense Command and Control Research Program publication, 2006

[ANA] http://www.ana-project.org/

[And72] Anderson, P.W., "More is Different: Broken Symmetry and the Nature of the Hierarchical Structure of Science", Science 177 (4047): 393-396, 1972

[BIS]http://dssg.cs.umb.edu/wiki/index.php/BiSNET/e

[Bit04] M. Bitbol, P. Luisi “Autopoiesis with or without cognition: defining life at its edge”, *Journal of The Royal Society Interface*, Vol. 1, No. 1. (22 November 2004), pp. 99-107.

[Bon99] Eric Bonabeau, Marco Dorigo and Guy Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems”, New York: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity, 1999

[Bro91] Brooks, R.A., “Intelligence without representation. Artificial Intelligence”, Journal 47: 139—159, 1991

[Buc06] E. Buchinger, “The sociological concept of autopoiesis: Biological and philosophical basics and governance relevance” Kybernetes Journal, vol.35, issue 3-4, pages 360-374, Emerald Group Publishing Limited, 2006

[CAS] http://www.cascadas-project.org/dissemination.php

[Cru94] Crutchfield, James P., “The Calculi of Emergence: Computation, Dynamics, and Induction”, Special issue on the Proceedings of the Oji International Seminar: Complex Systems — from Complex Dynamics to Artificial Reality", Physica D, 1994

[Dek06a] Dekker, A.H. “Measuring The Agility Of Networked Military Forces” In Journal of Battlefield Technology, Volume 9, Number 1, March 2006, pages 19-24

[Dek06b] Dekker, A.H. “Centralisation *vs* Self-Synchronisation: An Agent-Based Investigation”, in 11th International Command and Control Research and Technology Symposium, Cambridge, September 2006

[Dek06c] Dekker, A.H. “Agility in Networked Military Systems: A Simulation Experiment”, in 11th International Command and Control Research and Technology Symposium, Cambridge, September 2006

[Dip97] E. Di Paolo, “Social coordination and spatial organization: Steps towards the evolution of communication”, Proceedings of the Fourth European Conference on Artificial Life (ECAL'97), 1997

[DiP05] E A Di Paolo “Autopoiesis, adaptivity, teleology, agency “Phenomenology and the Cognitive Sciences, 2005

[Dod06] Dodd, Richardson, Alston, Beautement, “Investigation into the C2 arrangements for Edge Organisations”, QinetiQ, 2006

[For04] Nancy Forbes, “Imitation of Life: How Biology is Inspiring Computing”, MIT Press, Cambridge, MA 2004

[IBM1] http://www-01.ibm.com/software/tivoli/autonomic/

[Lew06] Lewis, J., Demange, L. “An Implementation of the Copycat Architecture Using the Starcat Adaptive Computational Framework” in *Proceedings of the 2006 Las Vegas International Conference on Computer Science*, pp. 502-508 (ICAI’06), 2006

[Luh89]N. Luhmann,, “*Ecological communication”*, Cambridge: Polity Press, 1989

[Luh95]N. Luhmann, “*Social Systems”*, Stanford University Press, 1995

[Mat80] MATURANA, H., and F. VARELA, “Autopoiesis and Cognition: The Realization of the Living”, Boston Studies in the Philosophy of Science [Cohen, Robert S., and Marx W. Wartofsky (eds.)], Vol. 42, Dordecht (Holland): D. Reidel Publishing Co., 1980.

[Mat98] H. Maturana and F. Varela, “The Tree of Knowledge: The Biological Roots of Human Understanding”, Boston 1998

[Mce06] J. McEver, D. Martin “Operationalizing C2 Agility” Information Age Metrics Working Group (IAMWG), December 2006

[Mcm01] Barry Mcmullin, Dominique Groß, “Towards the Implementation of Evolving Autopoietic Artificial Agents”, In Proceedings of the 6th European Conference on Advances in Artificial Life, Prague, Czech Republic, 2001

[Mof04] J. Moffat “Complexity theory and network centric warfare” Information age transformation series” USA department of Defense Command and Control Research Program Publications, 2004

[Mur97] K. Murthy, “Autopoiesis and strategic management in emerging economies”, Portland International Conference on Management and Technology, (PICMET '97), July 1997

[Pri77] Prigogine, Ilya; Nicolis, G. “Self-Organization in Non-Equilibrium Systems”, Wiley Publishing 1977

[Ste96a] Luc Steels, Fondazione Carlo, Erba Milano, “The origins of intelligence”, in proceedings of the Carlo Eba foundation meeting on artificial intelligence, Italy 1996

[Ste96b] L. Steels, “Self-organizing vocabularies” In Proceedings of Fifth Conference on Artificial Life (1996)

[Ste07] S. Stetter (editor), “Territorial Conflicts in World Society”, Modern Systems Theory, International Relations and Conflict Studies, Routledge 2007

[SOC] http://www.fp7-socrates.org/

[Teu88] G. Teubner (editor), “Autopoietic law: a new approach to law and society”, European University Institute 1988

[Var73] Varela, Francisco J.; & Maturana, Humberto R., “De Máquinas y Seres Vivos: Una teoría sobre la organización biológica.” Santiago de Chile: Editorial Universitaria., 1973

[Var91] F. Varela, E. Thompson and E. Rosch: The Embodied Mind: Cognitive Science and Human Experience, Cambridge: The MIT Press, 1991

[Var99] F. Varela: Invitation aux sciences cognitives, Paris: Seuil, 1999

[Von98] Von Krogh, G. “Care in knowledge creation”, California Management Review 40, p. 133, 1998.

[WEB1] http://blog.robjsoftware.org/2007/09/conscientious-software-part-1.html

[Wie48] Weiner, N., “Cybernetics – Communications and Control in Man and Machine”, MIT Press, 1948