Model-based provisioning and management of adaptive distributed communication in mobile cooperative systems

Sakkaravarthi Ramanathan, Ismael Bouassida Rodriguez, Khalil Drira, Christophe Chassot, Sibilla Michelle, Thierry Desprats

Model-based provisioning and management of adaptive distributed communication in mobile cooperative systems

Sakkaravarthi Ramanathan [1], Ismael Bouassida Rodriguez [1], Khalil Drira [1], Christophe Chassot [1], Sibilla Michelle [2], Thierry Desprats [2]

1 : Laboratoire d'analyse et d'architecture des systèmes (LAAS) CNRS : UPR8001 – Université Paul Sabatier - Toulouse III – Institut National Polytechnique de Toulouse - INPT – Institut National des Sciences Appliquées de Toulouse

2 : Institut de recherche en informatique de Toulouse (IRIT) CNRS : UMR5505 – Université des Sciences Sociales - Toulouse I – Université Toulouse le Mirail - Toulouse II – Université Paul Sabatier - Toulouse III

Abstract

Adaptation of communication is required to maintain the reliable connection in collaborative activities. Within the framework of wireless environment, how can host entities be handled in the event of a sudden unexpected change in communication and reliable sources? This challenging issue is addressed in the context of Emergency rescue system carried out by mobile devices and robots during calamities or disaster. For this kind of scenario, this book proposes an adaptive middleware to support reconfigurable, reliable group communications. Here, the system structure has been viewed as participants named control center, coordinators and investigators. Control center possess high processing power and uninterrupted energy and it is responsible for global task. Coordinators and investigators are entities e.g. autonomous robots and firemen that own smart devices to act locally in the mission. Adaptation at control center is handled by semantic modeling whereas at local entities, it is managed by a software module called communication agent (CA). Modeling follows the well-known SWRL instructions which establish the degree of importance of each communication link or component. Providing generic and scalable solutions for automated self-configuration is driven by rule-based reconfiguration policies. To perform dynamically in changing environment, a trigger mechanism should force this model to take an adaptive action in order to accomplish a certain task, for example, the group chosen in the beginning of a mission need not be the same one during the whole mission. Local entity adaptive mechanisms are handled by CA that manages internal service APIs to configure, set up, and monitors communication services and manages the internal resources to satisfy telecom service requirements.

Keywords: Autonomic computing, Wireless Communication, Middleware, Software Agent, Semantic Model, Event-Based Communication, Self-adaptation, Self-Reconfiguration, Ontology

Chapter 1
Introduction, state of the art and background

The book aims at studying and developing means to design, specify and implement a set of mobile autonomous entities with well-established properties particularly in terms of self-heal ability and to achieve a set of missions and self-adaptation in a dynamic environment. This research is supported by the French project ROSACE (RObots et Systémes Auto-adaptatifs Communiquants Embarqués).

This book focuses on the associated software (models, algorithms and systems) that require for ROSACE project. It also addresses in a systematic way about the relationships among the software layers and the specific constraints imposed to the middleware layer corresponding to the real-time systems as well the management of network resources and communication services.

ROSACE brings together a strong research consortium composed of research teams from three laboratories (CERT-ONERA, IRIT and LAAS-CNRS,) for making real progress in this area: an active and central object - namely a fleet of cooperative robots - is critical for keeping the difficult and ambitious scientific and technical work well grounded in relevant realities and well focused on actual needs.

Document Overview

This document describes the operational scenario aims to illustrate de environment and the collaborative behaviour of the entities evolving in this environment, for achieving ROSACE's mission objectives.

The document is organized as follows. After this introduction, chapter 2 gives an overview of ROSACE scenario and presumed solution. ROSACE assumptions, concepts, notation, and views are also described in this section. The next sections are devoted to define operational scenarios, use cases and scenes for each project perspective. Section 3 describes the communication network and its adaptive mechanisms. Section 4 presents the middleware agent followed the ontology modeling.

State of the art

In the literature, several studies focus on middleware approach with automated management techniques for adaptation. This means introducing changes to a program or maintaining functionalities and, whenever possible, improving performance. In [1], an adaptive framework supporting multiple classes of multimedia services with different QoS requirements in wireless cellular networks is proposed. The work in [2] envisions middleware architecture for service adaptation based on network awareness to manage resources in an adaptive context.

Further research [3] provides frameworks for designing transport protocols whose internal structure can be modified according to the application requirements and network constraints.

We also found that, in the case of architecture adaptation, middleware plays an important role for system transformations those changes according to the environment and requirement. To cope with evolving constraints, behavioral and architectural adaptability is required at several levels. This entails coordination management without which performance would drop much below target. In [4], a schema is described for dynamically managing distributed computing resources by continuously computing and assessing QoS. Here, resource utilization metrics are determined a posteriori and adaptive distributed system reference architecture is equally put forward.

After developing an information system, there might be many reasons for adaptation, e.g corrective, evolutional or perspective [5]. If we consider the corrective adaptation, application does not behave as expected. The solution is to identify the application module that causes the problem and replace it by a new module providing the same functionality as before. In case of developmental adaptation, all features are not taken into account in the design phase. But with changing user needs, the application needs to support the user's request and this could be achieved by adding new modules or modifying the existing functionality while keeping the same application architecture. Finally in perspective adaptation, the idea is to improve performance. For example, if a module receives numerous requests that result in performance degradation, it can be duplicated to share the workload.

QoS is important for self-healing service-based distributed interactive applications. It requires the ability to deal with permanently changing constraints at the communication and execution levels. The work in [6] presented a self-healing middleware framework furnishing properties for QoS management in these kinds of distributed applications.

High-availability applications with time-dependent resource requirements demand certain resource level assurances to operate correctly. QoS resource management techniques are being successfully developed that allow network systems to provide such assurances. The work in [7], developed a middleware agent that mediates application resource usage so as to ensure that applications get the resources they need in order to provide adequate performance.

The authors in [8] describe architecture metamodel for adapting components that implement coordination for reflective middleware distributed across peer devices. This work also investigates the reconfiguration types in various environmental conditions. The work in [9] is presenting the dynamic control behaviour middleware that incorporates reinforcement machine learning support of autonomic control of QoS management and thus reduces the overall system knowledge required by the system developers.

Self-adaptable systems dynamically adapt to satisfy new functionality requirements, to optimize performance and to adapt to variable runtime conditions. Context-aware middleware acquires and utilizes information about the context of a communication device and the

network status to provide an action. The work in [10] modelled the context middleware across three layers of abstraction (runtime, context changes and applications). The aim is to achieve self-adaptation of the composition without jeopardizing the integrity and usability of the overall system. The authors in [11] address the model of self-adaptive distributed components that enables applications by specifying adaptations and communications separately at runtime without loss of information.

Motivated by the above discussion, we present a novel ontology-based support for reconfigurable adaptive group communication architecture at control center. This approach improves the distributed decision making that readily acts in a time-constraint situation. At local entities, we have designed a communication agent with autonomic computing properties to take a decision by itself depending on the situation or trigger the control center if there is no solution.

Background

The scenario has the following main phases:

1. **Fire detection and evaluation.** Fire evidence may be acquired by different means: forest sensors, satellite images, human observations and others. Observations and alarms are sent to a control center where they are correlated and evaluated to decide the appropriate response.
2. **Public organization response.** In most of the cases an organization is set up according to the gravity and the estimated impact of the fire. Initial Decisions are taken by the head of the organization which acts according to well established plans and operating protocols. Common activities during this phase are:
   1.1 Set up the intervention procedure
   1.2 Resource selection, mobilization, and allocation
   1.3 Resource deployment plan
   1.4 Monitoring and control of resources
3. **On site resource deployment**. This phase is the consequence of the execution of the plans procedures, and protocols prepared in the previous phase. These plans might be adapted to specific circumstances of topography, fire evolution, and availability of the resources on site. Specific objectives and tasks should be defined according for humans and resources according to the mission priorities and the specific context. Typical activities during this phase are:
   - Deployment of intervention teams in different areas
   - Deployment of common mission resources on site, i.e. telecommunications, energy, logistic, transportation, etc.
   - Establish initial priorities and goals according to mission plan
   - Goal and task distribution among team members
4. **On site intervention.** In this phase all the resources are coordinated to achieve the mission whose main objective is to minimise the impact of fire ensuring life safety, protecting the environment and property. Activities during this phase are highly dependent on the specific environment where the fire occurs. When human life is in danger intervention teams should focus on saving lives. This involves;
   - Location of people in possible danger
   - Advice and guidance for moving to a safe location
   - Location of injured
   - Providing first aid and transportation to safe places
   - Meet basic human needs
   - Others

   Operational activities include but are not limited to:
   - Location of the fire areas to be extinguished
   - Resource coordination for fire isolation and extinction in specific areas
   - Logistic management for supporting intervention activities
   - Protection of resources (radio antennas, roads…)

- Self-protection

Activities necessary to maintain operationality and efficiency of the resources on site are.

- Monitoring and location of team members
- Monitoring of fire status and estimation of fire progress
- Communication among team members, and the mission control center
- Others

5. **End of mission.** Teams and resources involved in the mission should return to initial locations. Evaluation reports should be elaborated to summarize the activities performed, the challenges, strengths and weakness of plans, procedures, and protocols, efficiency on resource utilization, and assessment of mission results. These reports will be taken into account for improving achievement in future missions.

Assumptions

The following intervention teams will be considered in the scenario:

- Firefighter team. This team is made up of firefighters with the appropriate equipment to achieve on site intervention goals – extinguishers, fire blankets, vehicles, communication equipment, first aid kits, etc.
- Autonomous Ground Vehicle (AGV) team . This team is composed by the LAAS robots
- Autonomous Aerial Vehicles (AAV) team. This team is composed by the AAVs from ONERA

Perspectives

The scenario will be partitioned into different use cases to show the functionality to be achieved, the goals, activities, and cooperative behavior among the different entities. We illustrate the collaborative behaviour of a group of actors situated in an environment, which can play different roles for achieving the mission objectives. This book also aims to specify significant situations to determine the expected behaviour of the system to achieve the mission objectives satisfying the mission's constraints.

Here, graphical notations are necessary to represent both the relationships among the entities and their behaviour participating in mission goals. UML notation will be used for two main reasons: 1) It provides standardized graphical formalisms which are widely accepted in industry and academia; 2) Notations are supported by tools which are available for free download as open source, and are also commercially available. In order to facilitate UML modeling, each use case should describe a goal-oriented sequence of interactions between actor's roles and the mission resources necessary to achieve the functionality that satisfies a specific goal. A use case is initiated by an actor's role with a particular goal, and completes successfully when that goal is satisfied. Different scenes could be defined as instances of a use case. These scenes represent a single path through the use case. Thus, one may construct a scene for the main flow through the use case, and other scenes for each possible variation of

flow through the use case (e.g., triggered by options, error conditions, security breaches, etc.). Scenes may be depicted using sequence diagrams and communication diagrams.

Throughout this book, we will cover the following topics:

- Detailed explanation of communication network and its telecom perspective in ROSACE scenario
- Detailed adaptation management techniques for monitoring the communication system in order to satisfy current requirements and supported activity in order to handle the evolution of these requirements
- Presenting adaptive communication acts on different layers (transport and middleware)
- Cooperating with different layers by receiving notifications and by sending alarms when adaptation is not possible
- Description of deployment configurations using dynamically reconfigurable protocols and software architectures
- Definition of architecture of the multi-level adaptability management system (distribution of adaptation on different levels)
- Technologies and tools to integrate adaptable properties in the usability of system architectures

Discussion

Chapter 2

ROSACE Project and overall challenges

Infallible and efficient communication is absolutely important for public safety and disaster recovery operations. Many natural disasters have proved that there exist significant inadequacies in the monitoring system to communicate the problem to the rescue team. One such major problem that stopped rescue teams and emergency services during the calamities was the lack of system to take decisions by its own in case of hierarchical dependency. Another problem is the unavailability of terrestrial communications infrastructure such as traditional landline but it is crucial for the system to connect all the rescue teams in order to send/receive messages to save the people. Here, sensors or alarm systems are installed to monitor the catastrophic areas such as forest, flood regions etc. When there is an accident, the sensors send data to the Control Center (CC) to take action (sometimes, the control center is informed by the people who are in trouble or by volunteer). Once the information data like video, voice, pictures, messages, etc. are received, CC takes an appropriate action by using decision model depending upon the nature of the calamity.

After analyzing the situation, a team has to be set up to in that incident area to help the people who are in real danger or to protect the natural resources. The team shall be a truck with firemen, robots or helicopters (we use the term"participants" to represent the team members). All these participants have ROSACE devices embedded with Telecom Agent (TA), also called Communication Agent (CA) and WiFi cards.

The team has three major actors: mission supervisor, coordinators, and field investigators. The supervisor's function is to monitor, manage, decide and authorize actions to coordinators and investigators. Coordinator's task is to report to the supervisor and to manage the investigators during the mission and assign tasks. The investigator's role is to explore the operational field, observe, analyze, and report about the situation. There exist two major steps, i.e., "Exploration step" and "Action step". To support this, we have coordination and cooperation flows. Coordination flows take place between investigators and their coordinator and between the coordinators and the supervisor. Cooperation flows occur between the investigators within the same group or between the investigators of different groups.

To make it simpler for the reader, Figure 1 explains briefly a situation with the flows and network connection. The two trucks (coordinators) have the WiFi access points to communicate with the investigators like robots, firemen whereas the supervisor communicates with coordinators using satellite and another by using the WiFi access point. But if there is a network interruption on WiFi access point between the supervisor and a coordinator, a solution could be to contact the other coordinator and its investigator to reach the lost coordinator. Another solution is to create a new network, thanks to the walker's mobile devices; communication can be established as shown in the diagram. In the following chapters, we use the term "group" to indicate the coordinators and investigators using the same SSID or using the same WiFi modes like Infrastructure, Adhoc, etc.

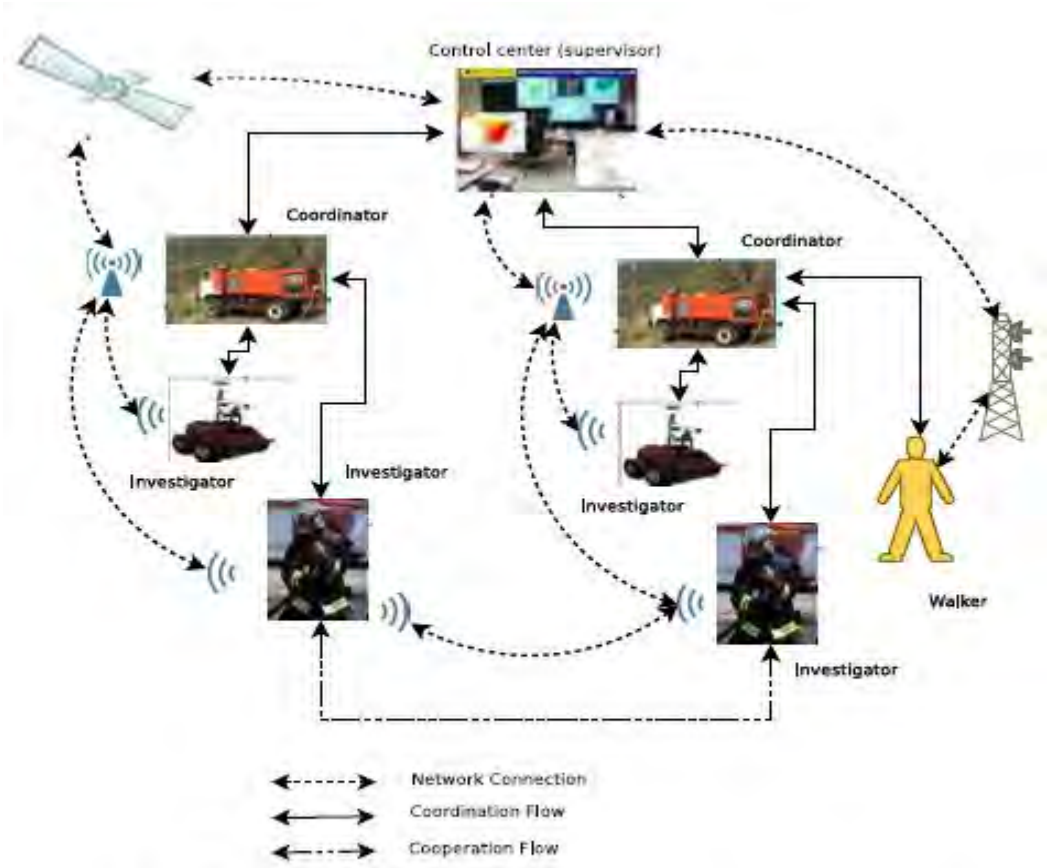


Figure 1: ROSACE scenario Description

Communication Network

There are three network regions in this scenario:

Investigator's network: By interconnecting the investigator „devices, network is established. At first, all the investigator's devices connect to its coordinators using WiFi infrastructure mode and in case of non-detection; the devices tries to connect to its own peers using adhoc mode to form the network. Here, the devices should be equipped with GPS position.

Coordinators network: At incident area, infrastructure networks are normally unavailable; coordinators establish a temporary communication network for that specific duration. Controlling the investigators, sharing mission critical data and coordination of recovery efforts are performed here. For example, routers can be deployed at selected location at the incident site to create an instant wireless hot zone. The coverage of this network can be

extended by mobile terminals/PDA carried by the investigators participating in the routing and forwarding of packets.

Supervisor network: It is a permanent network to collect data, commanding/supporting the decision making and cooperate with coordinators and investigators. It provides wide area connectivity thanks to satellite technology, GSM, WiMax, etc.

Communication is essential as it allows information exchange among the participants. For that, we need to consider a comprehensive set of requirements that need to be met by the current communication systems. These functional requirements are as follows:

Service support: Audio and video are the main services required for this kind of mission. Interactive data services like instant messaging, database queries, internet connection are also considered. Downloading the map, GPS position also plays a vital role.

Communication modes: Depending upon the mission requirements, devices should be capable to activate unicast, multicast and broadcast ideally. If the network technology allows using anycast and geocast, the devices must have the ability to utilize them. Furthermore, the devices need to be able to operate in peer-to-peer mode if there is a problem in standard modes like infrastructure or adhoc.

Network management: Performance, configuration and failure management are covered here. Supervisor assigns roles to the investigators and controls their priority. Also, global network monitoring and decisions for action reinforcement are handled here.

In our scenario, we are focusing the management of telecommunication resources available in the mission in order to provide the best communication possible to all participants working in the operating scenario. It could be achieved by allowing the participants to connect others to establish the network and also to communicate. Figure 2 shows the flexible connection and the scope of communication. Here, if there is a radio connection exists between the different entities, then the source will be able to communicate with the destination.

Imagine, at a particular incident, the team has a supervisor, two coordinators and each coordinator has 3 investigators. The arrow represents the communication between the participants. The idea is to allow all the participants to connect each and everyone. For example, a coordinator connects to the supervisor, its peer and to its investigators. But if a coordinator wants to connect to the investigator belonging to other group, there should not be any problem. Similarly, the investigators contact their peers in the same group and also with the peers of other group. By achieving this, a versatile communication is achieved.

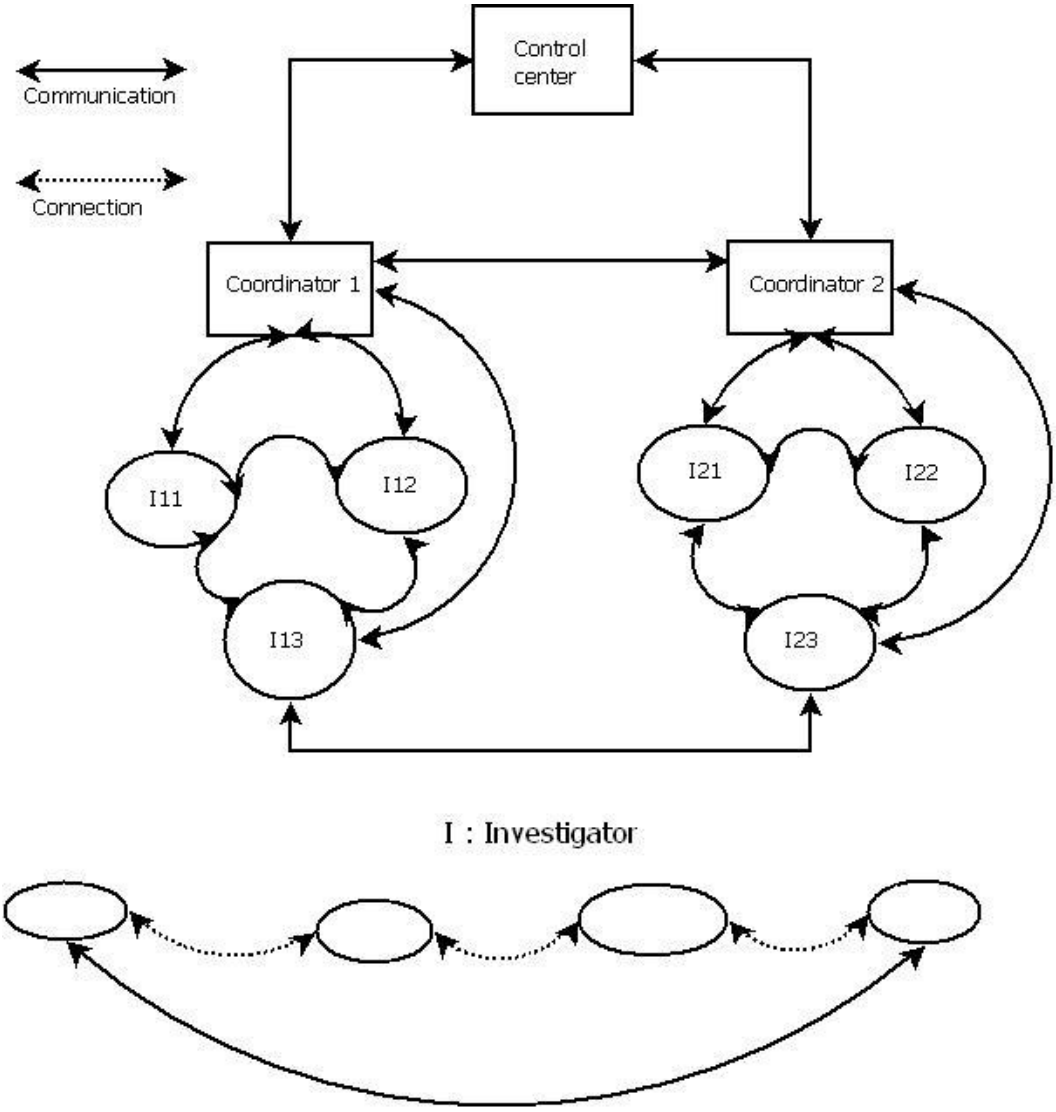


Figure 2: A versatile communication

Hence, the specific goals are:

- Establishing a local network to provide permanent connectivity among in site intervention teams and the supervisor.
- Managing telecommunication resources to guaranty a permanent connectivity among mission participants.
- Providing best-possible quality of service (QoS) according to communication goals and available resources.
- Preserving the quality of communications (performance and consistency with activity requirements).

Infrastructure assumptions

WiFi access points are installed on firefighters trucks, ambulances, Aerial ground vehicles (AGVs) and also on robots. The operational assumptions including networking infrastructure, technologies, models, and communication services are depicted in the following and specific situations are also described in order to illustrate the expected behaviour of the mission. Use cases help identifying the high level entities such as agents or roles which could be in charge of achieving the overall behaviour. They also used to define the higher level tasks and information needed by those entities to perform identified behaviour.

Note:

A telecommunications network is a collection of terminals, links and nodes which connect together to enable communication between nodes/terminals. The links connect the nodes together and are themselves built upon an underlying transmission network which physically pushes the message across the link. There are wide ranges of technologies to establish a communication network. Each one has its own merits and demerits. For example, Wireless local area network (WLAN) uses both a high-frequency and low frequency radio technology and uses spread spectrum technology to enable communication between multiple nodes in a limited area. In the following section, we highlight the possible wireless network topologies that are suitable for our scenario. The technical explanations of these terminologies are briefly included in appendix. This document assumes the reader will have sufficient knowledge about mission visions, goals and the structure of participant's communication medium.

Wireless network modes

The standard form of wireless network has two operating modes:

- Infrastructure mode (IFM), in which wireless devices are connected to an access point. This is generally the default mode for 802.11b cards. Even two 802.11 devices that are side-by-side in infrastructure mode must send data to each other through the access point. The supervisor is connected to the coordinators in IFM mode similarly the investigators are connected to their respective coordinators in IFM.
- Ad hoc mode, in which devices are connected to one another without any access point. We identify the advantages and disadvantages of this different connection modes thus paving a way to find the most suitable one for the mission.

Note:

Figure 3 represents the actual communication scenario in ROSACE context. It gives an ideal mode of network connection in which there is a supervisor, two coordinators, 3 firemen and 3 robots. The supervisor and coordinators have WiFi routers that are interconnected and coordinators control firemen and robots respectively. The firemen and robots are connected to their respective coordinators through WIFI adhoc. If fireman 2

wants to communicate to robot1, the path will pass through coordinator, supervisor and coordinator of that robot.

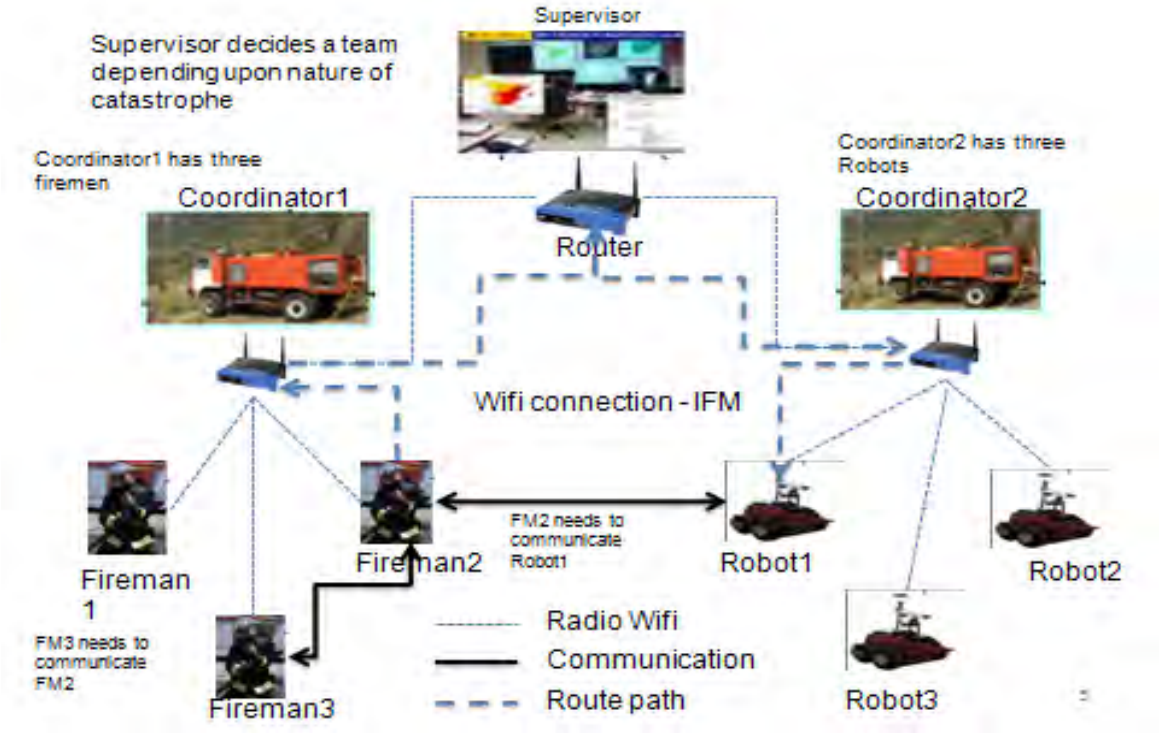


Figure 3: ROSACE Communication

Topology Classification

Possible modes and different technologies allow us to widely analyse the possible network topology that suits our mission. The communication is achieved through the technologies like traditional access point, MANET, MESH, WIMAX or multihop. Each technology can be activated through IFM or adhoc mode. Here, investigator is provided with one or two WiFi cards and a GSM card. By further exploring all the possibilities of using SSID, channels and card, we have got a broad classification (Figure 4). For example, in traditional access point, there are three modes possible: Infrastructure, adhoc and mixed. Configurations like SSID, channels and card are used in each mode. SSID is an identification key to connect to a subnetwork and at any given time, a card uses only one SSID. But if a device has more energy, we can activate another card for another SSID. So, here two cards use two different SSIDs to connect to two different subnetworks. But each card owns two channels, for example, channel 6 uses audio flow whereas channel 11 uses video flow. To be precise, consider that a device has two cards using two different channels with a single SSID. It means, at same time, it can communicate to two other devices for audio and video. More explanations are detailed in the appendix.

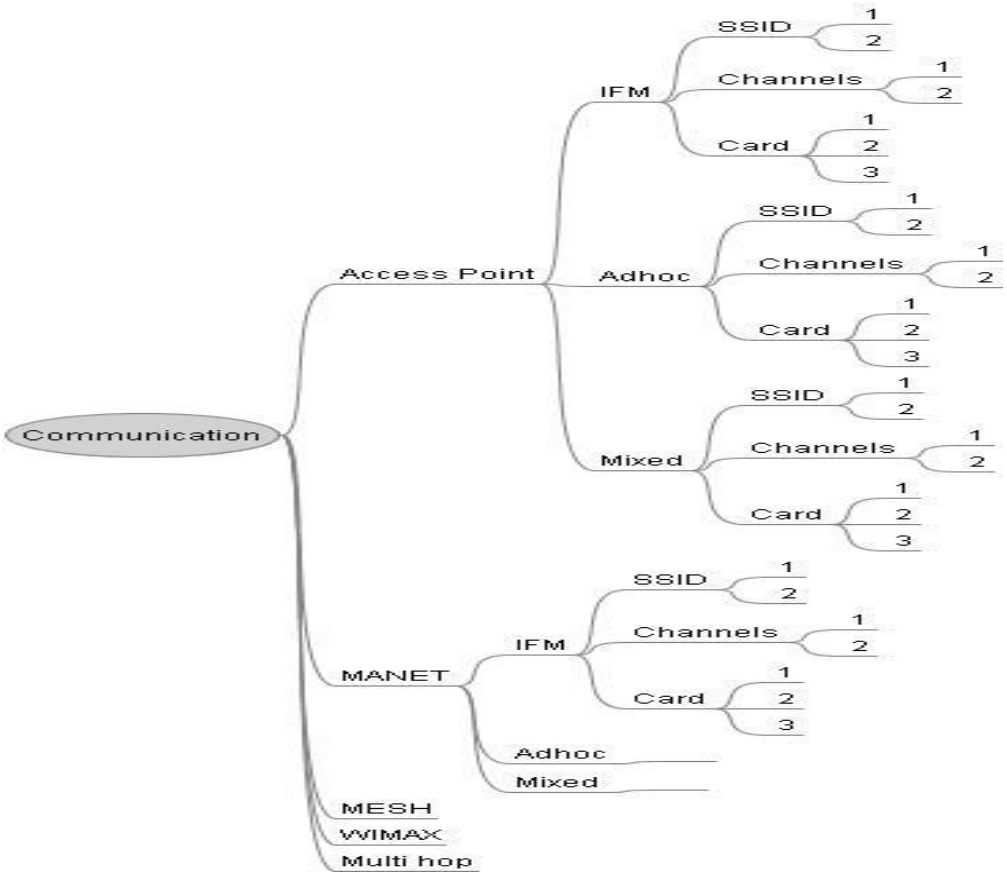


Figure 4: Broad classification of connection configurations

Criterias

This broad classification also helps us to focus on the criterias. For example, imagine a device has 2 WiFi cards. At any time, the device uses a card and if it wants to communicate with other device, it can switch to the channel corresponding to that device and it is possible by configuring the hardware. In case of failure in the card, the device activates the second card (redundancy). But if a device has 3 cards, we can configure 2 cards to connect 2 different SSIDs and third card in case of failure. By using 3 different cards, we can show the following criterias are fulfilled:

- Robustness/resilience/reliability
- Connectivity
- QoS
- Connection with non-Rosace entities
- Configuration overhead

Table 1 shows relation between the configuration and criterias.

Configuration	1 Interface card		2 cards		3 cards	
Criterias	Channel 1	Adaptation action: Switching to another channel	Channel 1	Adaptation action: Switching to another channel	Channel 1	Adaptation action: Switching to another channel
Robustness/ Resilience					Yes(*)	
Connectivity	Yes (*)	Yes (*)	Yes (**)	Yes (**)	Yes(**)	Yes(**)
QoS		Yes(*)	Yes (*)	Yes(**)	Yes(**)	Yes(**)
Connection with Non-Rosace devices						Yes(*)

Table 1: Criterias vs Configuration

With one card, a device can increase the connectivity by an adaptation action like switching the channels. In some specific case, it also supports QoS. For example, imagine a situation where the supervisor requests an investigator who is in communication already to move to another place as this action has some critical importance like capturing a video, acting as a bridge between the supervisor and another device for sensitive data communication. By doing so, the QoS of the whole system might be improved which is an added value. With 2 cards and also with an adaptation action, connectivity and QoS might be served. Also, by including the third care, we can assure that robustness and connection with non-rosace devices are possible. Here * represents the added value to existing potenial capability and ** indicates that this value is higher.

Challenges

In wireless network, there are many complex problems need to be addressed. Devices are energy constrained and devices itself non-responding, intermittent or no connectivities, etc. Low throughput, excessive delay and jitter are common problems. Physical environment, such as obstacles, weather, and noise influence the link quality. Bandwidth deficiency could be caused by the nearby devices that content for the shared wireless medium. Protocol misconfiguration like non-compliant device may not join the subnet even if they have link-layer connectivity.

Here the challenges are described below.

- Link qualities fluctuation: Multi-path fading and interference from the environment, can result in widely varying fluctuations in its quality. This results in routing path challenges and cause breaks in established connections.

- Device mobility: As the route to destination changes, reachable becoming unreachable after movement. Moreover, mobility cause transient breaks in a connection between nodes, thereby preventing the delivery of collected information.
- Protocol fault. It introduces undesired problems sometimes. (For ex, some routing protocols will create a routing loop/black hole at certain times, overly agile routing protocols cause route flapping that degrade network throughput).
- Traffic congestion. When the load approaches/exceeds the link capacity, a wireless experience congestion or even collapse. Also, the devices near gateways experience more congestion due to aggregated packet relay.
- Resource constrained devices: As they are resource constrained, these devices are characterized by low processing power, limited disk space, and low energy. The allocation of limited resources for monitoring can result in poor system performance.

In figure 5, imagine the investigator P13 loses its network connection from coordinator1 and it moves close to P23. If P13 detects the radio signal of P23, then the goal is to establish a connection automatically between P13 and P23 without having any trouble. By default, the wireless technology has the capability to achieve this connection (it depends on the implemented technology). This new connection is decided and forwarded to its control entities and this is activated by an intelligent component. This high level intelligent component encapsulates knowledge about telecom infrastructure, networking and transmission technologies. This component offers its capabilities to the rest of entities as services which are provided through standard interfaces.

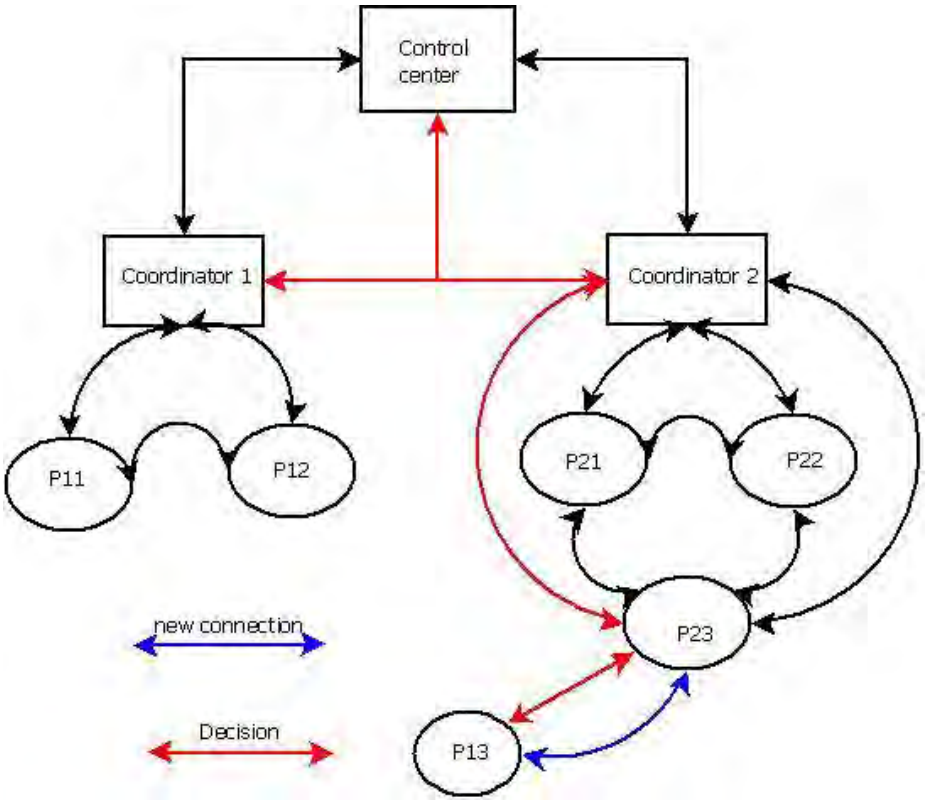


Figure 5: Connection and decision

This component is represented as a "Communication Agent" (CA) and has the following capabilities:

- Mission information awareness, CA could have access to mission information such as participants, roles, priorities, location and other information which may be used for telecom service provisioning
- Contextual information awareness. This includes: a) The internal context of the underlying node containing where the CA is located, and b) the environmental context, where are situated the rest of nodes and CA's participating in service provisioning.
- Seamless Management of telecom resources in hosting nodes to achieve adaptive communication needs. This includes a) managing internal service APIs and related middleware to configure, set up, and monitoring communication services, b) to asses' service quality according to mission needs, c) dynamic re-configuration and management of internal resources to satisfy telecom service requirements.
- Cooperation with the rest of CA to achieve Organizational goals. This includes a) achieving its own goals sharing common resources with other CAs, b) managing internal resources to satisfy collaboration requests, c) interacting with other CAs to exchange control information and data to guaranty end to end service provisioning and quality control.

CAs will be embedded in mobile robots and other telecommunication devices such as PDA's. They are expected to manage efficiently the overall telecom resources in the hosting node where they are deployed. As service providers for other internal units they should report errors and exceptions to their internal device components, and when possible suggest corrective actions to the control or decisional units.

Telecom Perspective Assumptions

The Telecom system might be viewed as a team of cooperating CA´s. The team as a whole will cooperate with the rest of ROSACE teams in order to achieve the objectives of the mission. Some of the specific responsibilities:

- Provide adaptive communication services to intervention teams. This will require
  - To set up set up a local network at the operational scenario using the available telecom infrastructure
  - Integration and interoperation of the local network with public and private networks in order to provide connectivity with the supervisor and the rest of teams participating in the mission
- Active monitoring of networking infrastructure
- Pro-active service management and quality control of the telecom services

Functional needs

Although centralized control is simpler to design and implement, it introduces a single point of failure, which can impede service reliability and scalability. Single point of failure can be avoided by not centralizing any single actor controlling the entire decisions. By doing so, we can prevent overload and incapacity of a node to respond the critical demands.

By distribution, current scenario is retained thus influencing the results of future situation (thanks to prior knowledge database). A distributed decision features high availability requirements and should always be available to actors. For example, if coordinators or investigators fail to reach the supervisor for an important decision, this will entail the worst consequences. By decentralizing, the different situations can potentially be handled much more easily than centralized ones. But distributing intelligence is not easy particularly in the case of adaptation, which needs special attention. To this end, the middleware has to actively monitor, control and allow an actor to make decision by his own. We have highlighted some of the challenges here:

Dynamicism: As operation conditions change, new participants can come in and out. Moreover, different services can be required in different places at different times.

Data Communication: As data may need to be exchanged among participants, bandwidth must be carefully used to avoid congestions and flooding. Problems like undesired delay or unpredictable jitter could be handled by data flow priority, use of another actor as forwarder, etc. Priorities and choice of alternative paths should not be static, but can dynamically be changed according to the requirements and changes in the network such as device' failures. Also, the data must be received soon enough so that successive dependent data can be used as well. Otherwise, not only is the data unnecessary, but sending and processing the data has consumed limited resources [19].

Adaptation Needs

After establishing the local network to provide reliable voice, video and messages between the participants, the CA should aware of the mission evolution. As each participant has a role to play, the location or a sudden change in the mission might cause this participant to change his previous role. The agent has to monitor, detect the changing environment and adapt to the evolution. While doing so, the data flow and priorities between the participants could be changed and thus the agent should handle and manage the connection between the participants. If there is a failure in doing so, it should be smart enough to repair by activating suitable functions. If the decision model can't find a solution to rectify the problem, it has to process the request to supervisor. In our case, there are two or three access points are

connected to establish the local network. Each access point controls its own group members and it manages the communication between them.

Middleware correction mechanism needs

The middleware should satisfy the following qualities:

- Gathered data from information providers need to store the user preferences in database.
- Deploying computational resources including free memory, processing power and possibly also some bandwidth, battery, etc.
- Dependencies like some components rely directly on the presence of another component to operate correctly.
- Determining when to modify the parameters of a particular transport protocol or switch from one protocol to another.

Management of participant profile related to network management is another task. For example, default interface, priority of each interface, interface selection policies need to be analyzed. According to the policies, the agent should open a channel for the traffic flow (a policy denotes the criteria for the selection of the best current network interface). But, certain data forwarding traffic of this particular interface may overload a node in the path from the data gathering points to the final node. A solution could be to send messages to other nodes into sleep mode and waking them only when it is required to forward data to neighboring nodes.

Data Flow

There are two kinds of data exchanged among nodes in the network: control data and application data. Control data is small and may not experience latency or unexpected delays to achieve its destination. So, control data is segregated from application data by receiving higher priority to be forwarded. On the other hand, there are several kinds of application data, e.g. simple values (integers and floats), video stream and character string. In spite of this sort of data have a priority lower than control data; they must fulfill the QoS requirements of the application.

Adaptation Triggers

After the deployment of certain mission by the control center, unexpected events might disrupt the mission. The middleware then has to execute a series of actions, in order to ensure the mission, despite the occurrence of these events. We term such events as adaptation trigger events. Some events are highlighted in Figure 23 like nodes movement, resources variation, etc. But triggers could change the entire mission or only affect the communication level (classified by different line textures in the figure).

Communication level triggers are represented by the connection/disconnection of nodes, variation in the rate of sending data, channel failure (non-delivery of data to destination), etc. Mission level trigger is the participant action (adding a new actor, removing an actor, change the role of an actor, etc).

Adaptation may be ruled by architecture or behavior-based transformation laws. In general, the adaptation is behavioral (or algorithmic) when the behavior of the adaptive service can be modified, without modifying its structure. Standard protocols such as TCP and specific protocols such as ETP provide behavior-based adaptation mechanisms. Behavioral adaptation is easy to implement but limits the adaptability properties. Indeed, the addition of new behaviors may be required. In this case, the component has to be recompiled and the adaptation can no longer be performed during run-time.

The adaptation is architectural when the structure of adaptive components can be modified. Some frameworks are provided for designing Transport level protocols whose internal structure can be modified according to the application requirements and network constraints. The replacement of a processing module by another(s) can be easily implemented, followed by a plug and play approach where the new component has the same interfaces as the replaced one.

Context adaptation

Context adaptation is applied to the services of the application (services adaptation), to the exchanged data with the nodes (content adaptation) and to the resources (resource adaptation). All these adaptations can be static or dynamic. Static adaptation is obtained by providing different pre-built versions of a resource for different context situations. Dynamic adaptation is done at runtime by filtering the service input and output according to the context.

Chapter 3 General Solutions

In such a dynamic environment, the CA installed in participants should be autonomous (adaptive), i.e., being capable of self-managing, to overcome the growing complexity of mission task and to reduce the barrier of problem diagnosis. In other words, an autonomic system makes decisions on its own, using high-level policies; it will constantly check and optimize its status and automatically adapt itself to changing conditions. In our mission, the adaptation consists of three phases (Figure 6), i.e., detection, decision and the action. These three phases are executed locally in investigator, globally in supervisor and distributed with peer entities. Also, it is a continuous process and dependent to each other (as soon as the symptoms are detected, the decisions are made and actions are taken while detection starts its process once again).

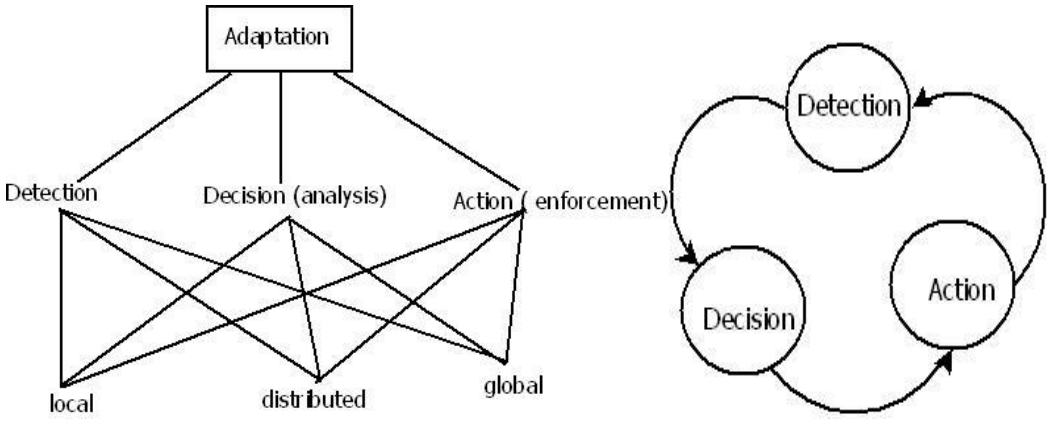


Figure 6: Adaptation phases

3.1 Autonomic problem management

It is hard to diagnose problems manually in wireless communication system. A good solution is to automate a problem management task by continuously monitoring network condition, analyze the problem when it is detected, and taking adaptation actions for self-recovery. We have to detect, identify, isolate and determine the root cause of problems to recover the system. Notifications of problem detection, maintaining and examining logs, tracing and identifying problems by examining environmental changes are some of the main tasks.

3.1.1 Problem detection

There are roughly three categories of problem detection measurements: local, neighborhood, and global measurements.

- Local measurements. Resource usage monitoring(CPU & memory) is handled local to estimate its own health status. Measuring the channel conditions such as the external

noise level, signal strength of its neighbors, and its perceived medium utilization can also be achieved. Statistics on link quality, relayed and dropped packets are collected. Other configuration parameters, such as transmission power level, routing table, neighbor list are used for problem diagnosis.

- Distributed measurements. A device may cooperatively work with its neighbors to provide some measurements, such as accurate link quality. For example, in case of wireless mesh network, a routing protocol may use link quality as the basis of path-selection metrics (ex, Expected Transmission Count, Weighted Cumulative Expected Transmission Time). In shared wireless medium, a device overhear its neighbor's transmission to collect relevant statistics for problem diagnosis. For example, a device may use the difference of a neighbor's received and relayed packets as an indication of misbehaving forwarding function .
- Global measurements. It is necessary to collect measurements such as network topology and routing state, to detect routing anomalies (loops and flapping). Channel conditions over the network as a whole should be considered when setting radio channel to achieve both well connected topology and high network throughput.

The figure 7 explains the difference between local, distributed and global decision. Local adaptation techniques are handled at investigator itself and in some case, it contacts its peer to make the decisions and this is totally different from the interactions held between the supervisor and the investigators. Also, the global decisions are updated to all the participants whereas the local decisions need not to be informed to its superiors except if it is critical. For the simplicity, we exclude the coordinator.

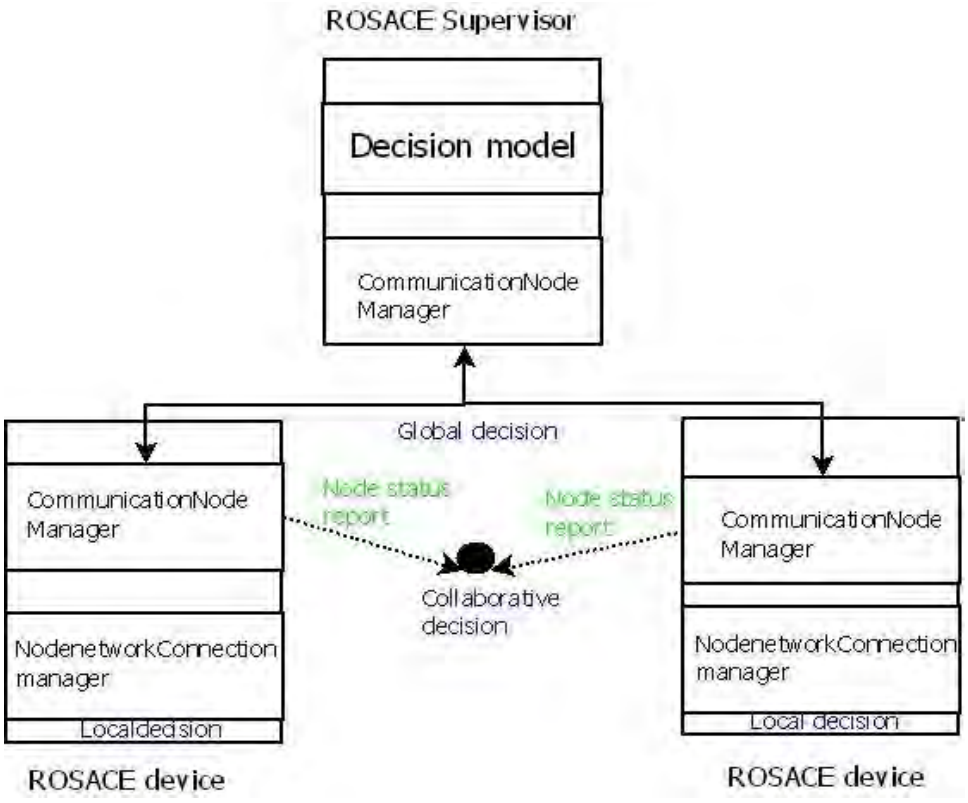


Figure 7: Global, collobarated and local decisions

As single node's observations are usually incomplete, it is necessary to correlate measurements from multiple sources for problem diagnosis. The supervisor collects all network measurement data and performs the analysis. The advantage being here is having a complete network observations and the disadvantage is that to obtain further measurement cause route disruption and network partition. For example, the supervisor takes long time to access the local measurement during connection problems. Solution is to spread the measurement information when a problem occurs.

But in case of distributed approach, the supervisor does not require to control all the measurement data thus saving bandwidth and avoiding the single point of failure. For example, a local problem is easily detected and diagnosed by cooperating among adjacent nodes. On the other hand, diagnosis takes valuable resources if performed by the devices themselves, which may degrade routing performance.

3.1.2 Diagnosis Algorithms

With sufficient measurement, it is probable to locate the problem and determine whether it is a link or device failure, protocol error, or traffic congestion. It is, however, challenging to pinpoint the root cause, based on the knowledge base to make an autonomic recovery decisions to correct the problem.

Problem diagnosis determines the root causes from observed symptoms, such as abnormal events derived from the network measurements. Machine learning is another solution to derive these diagnostic rules.

In model-based diagnosis approach, the key idea is to build a structural and behavioral model, where the structure is a acyclic graph representing influence relationship among components and the behaviors are expected performance output. The combination of structure and expected behaviors can by used to systematically track down and return the root cause. In the context of diagnosing communication networks, the structure must be constructed by network measurements because the topology is not known beforehand. Once the network model is obtained, it is possible to detect a problem if the observed performance does not match what is predicted by the model.

As model-based diagnosis is flexible to handle new problem types, the network model is constructed automatically and the reasoning logic remains more or less the same. The challenge here is to build accurate network models, including physical channels, network topology, and traffic workload, by network measurements (packet dropping, link congestion, external noise sources), so the prediction can be useful for diagnosis. Like all other diagnosis approaches, it is challenging if there are multiple problems occurred in the network.

The figure 8 explains the diagnostic mechanisms. As we have already explained, there are two modes in diagnosis; local, collaborative and global. Local device is controlled by the device itself without the intervention of any other participants whereas the collaborative is the collective decision between the peers in case of failures. If there is no appropriate solution, there must be a global diagnosis to detect the failure cause. Collaborative decision is very

useful to confirm the failure and it helps the system assure that the decisions taken don't disturb the other communication.

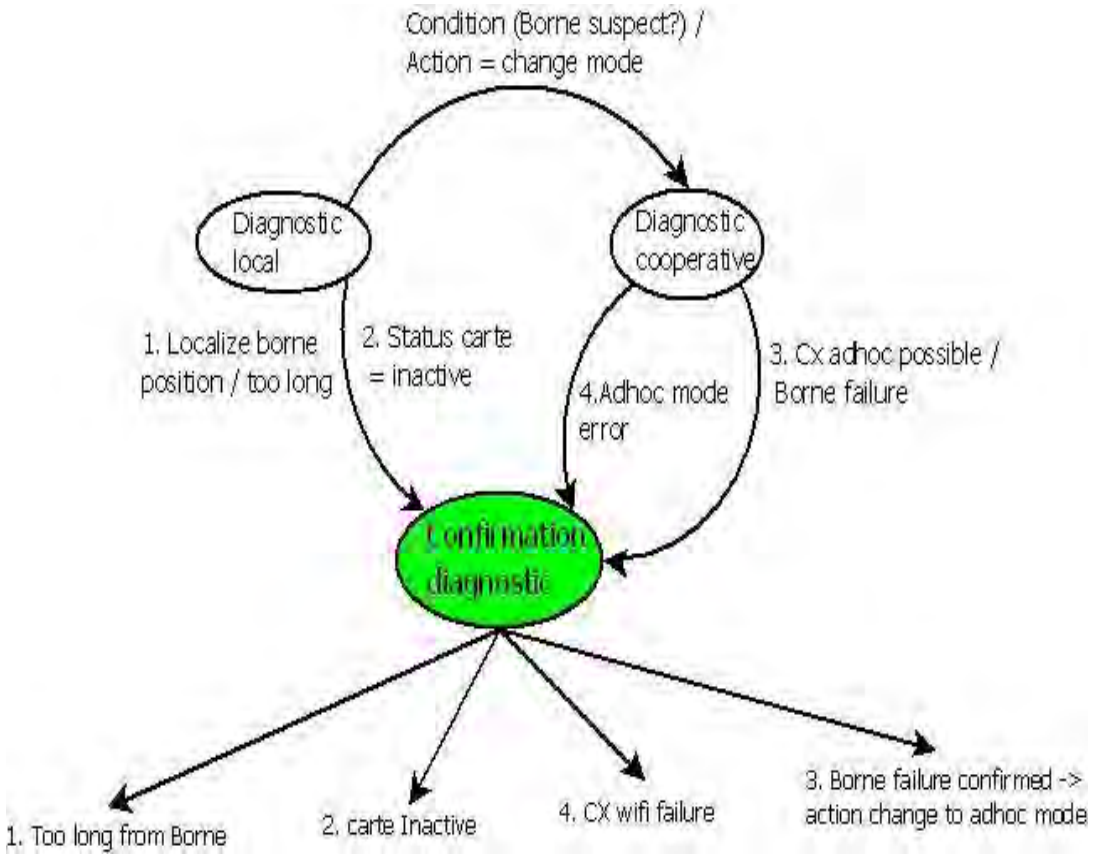


Figure 8: Diagnosis mechanisms

4. Algorithms

Here, we illustrate the algorithm in case if there is a problem in WiFi connection. To start our mission, the first step is to establish the communication network between the participants. As soon as the participants enter the intervention area, the robots and fireman's devices need the network without any problem. This algorithm explains if a fireman' device or a robot has a problem to connect the network. As there are many reasons, we have to consider the possible cases and an appropriate solutions to handle. Our algorithms use UML entity diagram for the instructions and the commands. Each algorithm is a list of well-defined instructions for completing a task. Starting from an initial state, the instructions describe a computation that proceeds through a well-defined series of successive states, eventually terminating in a final ending state.

4.1 IFM with 1 WiFi interface (capable of switching to ad-hoc mode) + GSM: local adaptation mechanism

If an investigator's device has a problem to connect the network, the symptoms are the non-detection of radio signal, inactive card or the device is far from the access point. Here, figure 9 explains the mechanism to solve the problem. Our adaptive TA analysis the causes and take an decision. For example, in case of device is far from the access point, the global decision is to localize the position (X,Y) of this device and the adaptation is requesting to move to a new position so that it can connect to the network (position's coordinates is sent to the robot through an interface).

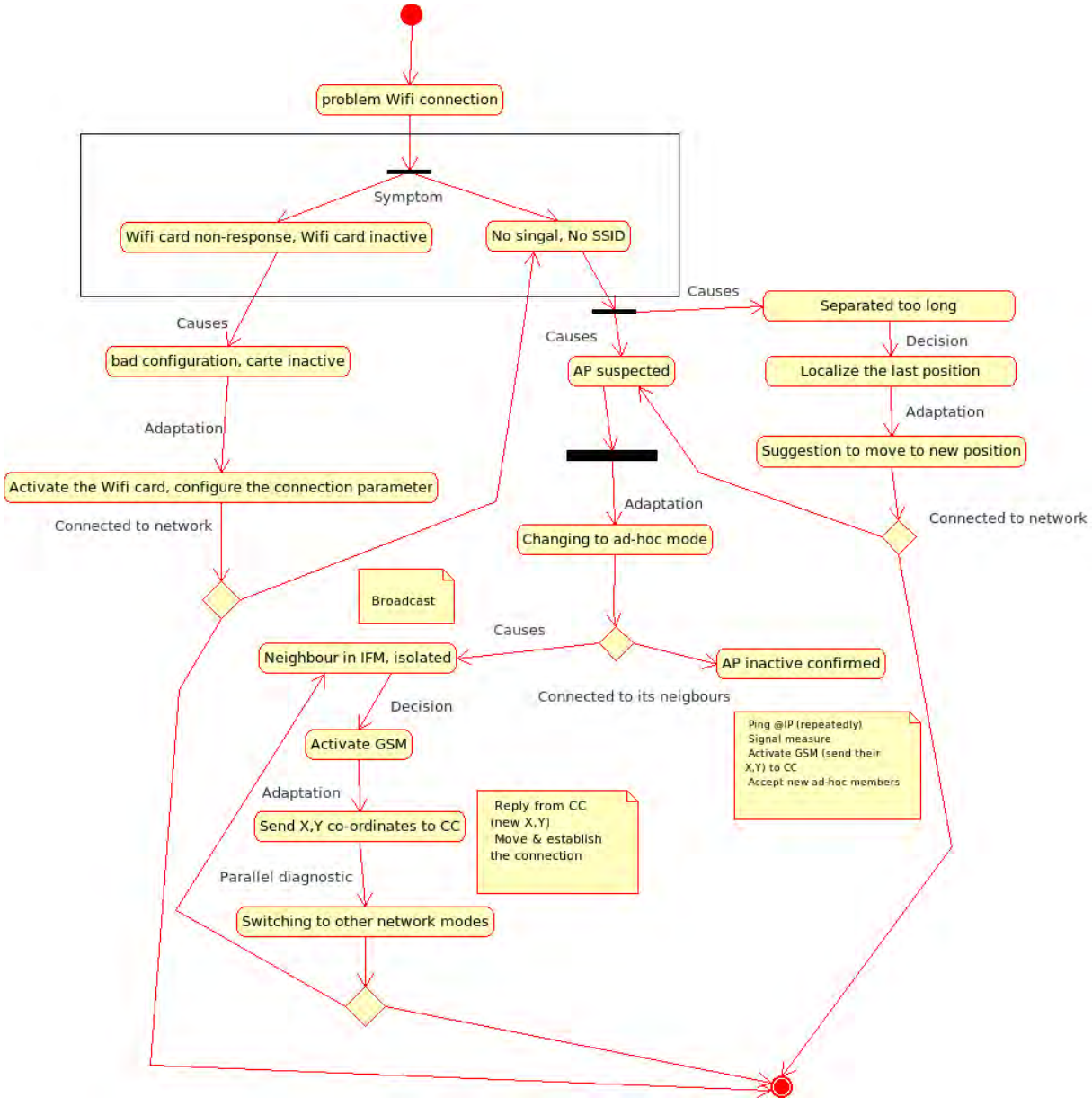


Figure 9: Local decision algorithm for IFM with 1 card

This is the last step for the adaptation because before doing that, the local autonomic system analysis all other possible chances to establish the network. One solution is to switch to ad hoc mode and if it connects to one of its peer (means other peer also switch to adhoc mode as it could not connect to acess point), means the devices are far from the access point or access pont is in failure state. Then the peers are connected and ready for the communication. Imagine a device is totally isolated from its peer, access point etc. In that case, the device switches to adhoc mode and if the device can not detect any radio signal, it has to activate the GSM card, an another adaptation mechanism to send its XY coordinates to the supervisor.

4.2 IFM with 1 WiFi interface (capable of switching to ad-hoc mode) + GSM:global adaptation mechanism

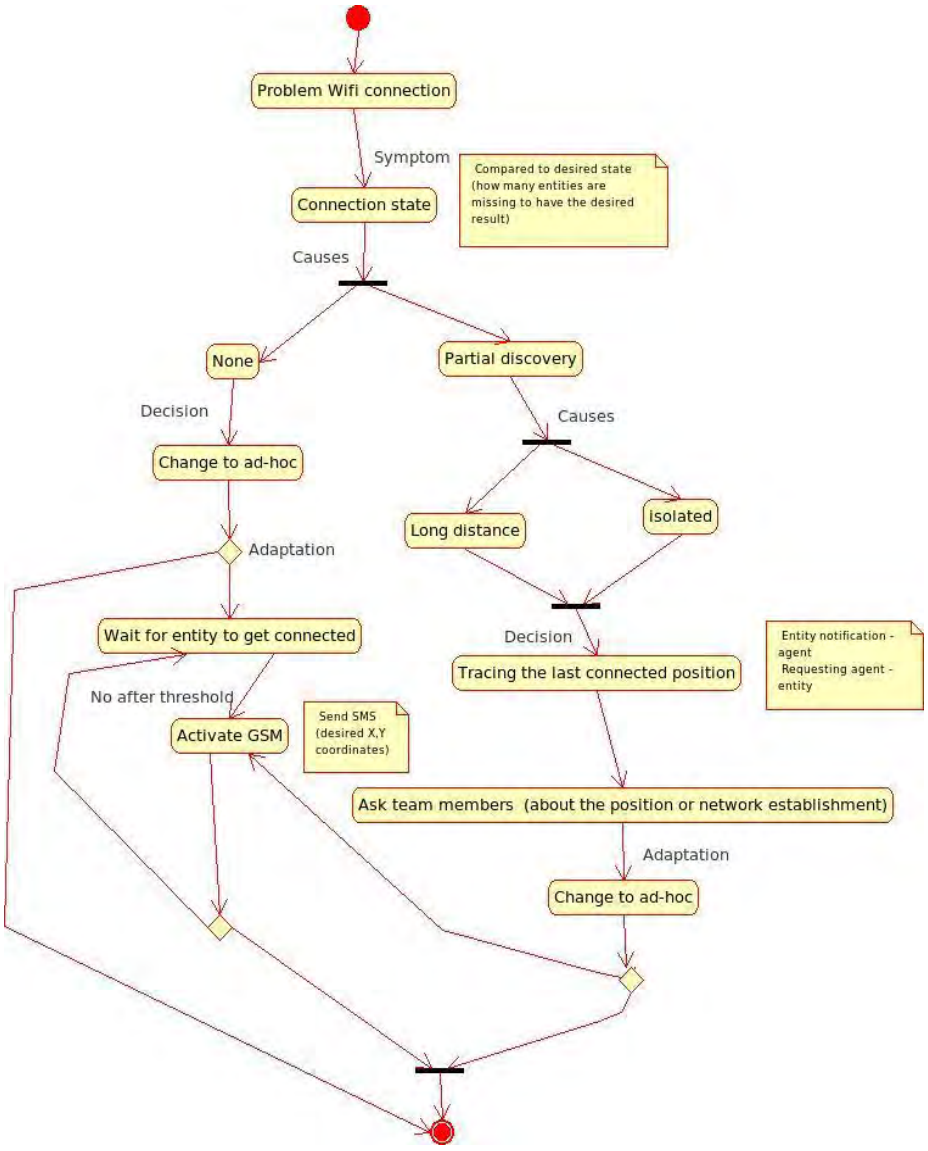


Figure 10: Global decision algorithm for IFM with 1 card

Similarly, the autonomic system at the COMM layer of the supervisor diagnosis, decide and take adaptative actions that are different from the investigator (Figure 10). When the participants deployed on the field, the supervisor's role is to assign the tasks to the investigators. To do that, he has to connect and communicate with all the desired participants. If there is a partial connection or no connection with any of the participants compared to the desired state of the mission, it has to diagnosis and take actions. Initially, it is connected in IFM mode. As soon as it doesn't find any of the participant, it has to change to adhoc mode to check whether any of the investigator wants to communicate with him. If there is a new connection to one of the investigators, then the communication starts. Otherwise, it has to activate the GSM card to let the investigators know its position. If the supervisor finds that if one of the investigator is far from the access point or it is totally isolated, then it has to trace the last connected position of this investigator and send a request to another investigator to move to this position. This is a mission task and it totally depends on the situation. The challenge here is to consider the priorities of the investigators whenever the supervisor makes a decision.

To conclude, it is simpler to use only one SSID for the whole mission except a new SSID to connect the foreign Mobile. If we use different SSIDs, the impacts and consequences are listed below:

- Switching challenge
- Repeated connection (resetting the network)
- Energy constraints (2 WiFi interfaces activating at the same time?)
- more steps to establish the connection
- With ad-hoc, no potential advantage of using 2 WiFi network card

5. Adaptation mechanisms

The adaptation mechanism requires the ability to deal with permanently changing constraints at the communication level. As there different situations that cause the failure, monitoring mechanism need to distinguish the causes. For example, monitoring to extract information about the device health, analysis to detect possible degradations, diagnosis & decision to identify the degradation source and executing the repair actions are analyzed here.

The Diagnosis/Prognosis & Decision

Proactively or reactively on receiving warning, the modules inspect the behavior on the basis of the disruption caused by the communication. This allows the identification/prediction of the past/eminent deficiency. The decision is based on reconfiguration actions for prognosis and repair actions for diagnosis [6]. For reactive situation, the monitoring services cooperate with the diagnosis services to detect service degradation and to react appropriately by repair plans. And for the predictive recovery, the monitoring services cooperate with the prognosis services to predict service degradation and to act appropriately by reconfiguration plans. Thus

it is a good act by preventing future nodes communicate to the nodes that currently involved in the degradation.

Hotspot transition

Another perception of communication adaptation is hotspot transition. Initially, a device could be a hotspot so that all other devices are connected. In this case, all data exchanged between the devices are passed through this hotspot. If there is a change in the mission, for example, the participants are moved to a new position; there might be a case where this hotspot could not able to connect all the devices. So, the adaptation could be to select a device to act as a hotspot so that all the neighbor's devices can be connected (figure 11).

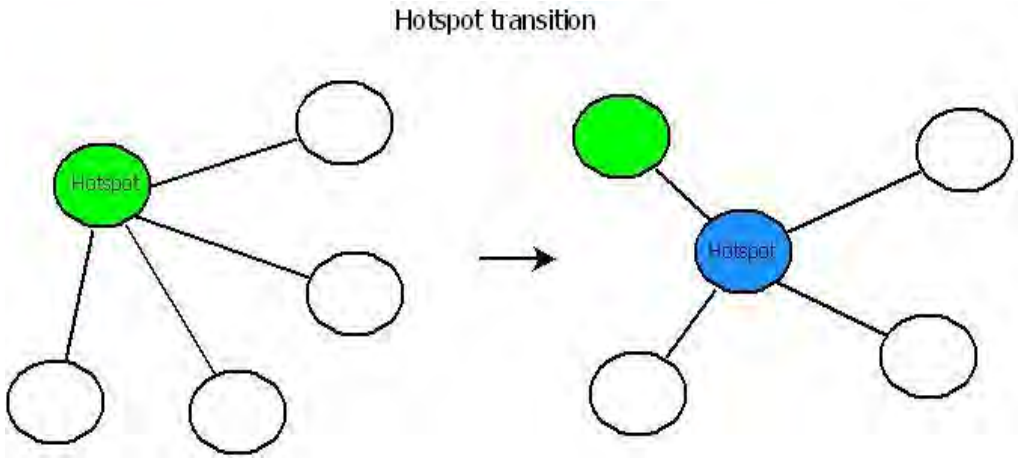


Figure 11: Hotspot transition

Delay management

By configuring the access points to send information about itself (identity, transmission channel etc) to the supervisor, changes are updated. As a device attaches itself to AP, the supervisor updates this device about its neighbors, through the AP of attachment. The device that is located in a region does active multi-channel scanning where it send probe requests for each channel scanned to locate the neighbor APs and notes their AP's received signal strength indication values along with the channel of transmission. Once the device moves, it already knows the channel where it located the next AP and avoids channel scans. The supervisor can also update as it is constantly updated of an AP's channel of transmission. This would reduce the probe delay significantly.

Note: When the device gets connected to any AP, it would initiate authentication with all the neighboring APs, horizontally, vertically and diagonally. Later when it moves to another AP there would be no authentication delay, as the device is already authenticated. Once connected to the new AP, the device again initiates authentication with all the neighboring APs, if it's

Neighbour Discovery

With the help of nearby devices, this technique takes advantage of the beaconing and probing mechanisms of IEEE 802.11 to ensure that connected devices do not pay unnecessary overheads for detecting disconnected devices. The technique also presents a simple technique for finding the approximate location of disconnected devices. It is useful for diagnosing wireless network performance problems [18].

• It allows disconnected devices to communicate via other nearby connected devices; this mechanism can be used to bootstrap wireless devices and resolve certain connectivity problems (figure 12).
• It is used for detecting and diagnosing a variety of faults: locating disconnected devices, diagnosing performance problems, etc.

Note: Devices have the capability of starting an infrastructure network (i.e., become an AP) or an ad hoc network on their own; this ability is supported by many wireless cards, e.g., Atheros, Native WiFi. Whenever faced with a choice of starting an ad hoc or an infrastructure network, infrastructure mode is better for many reasons.

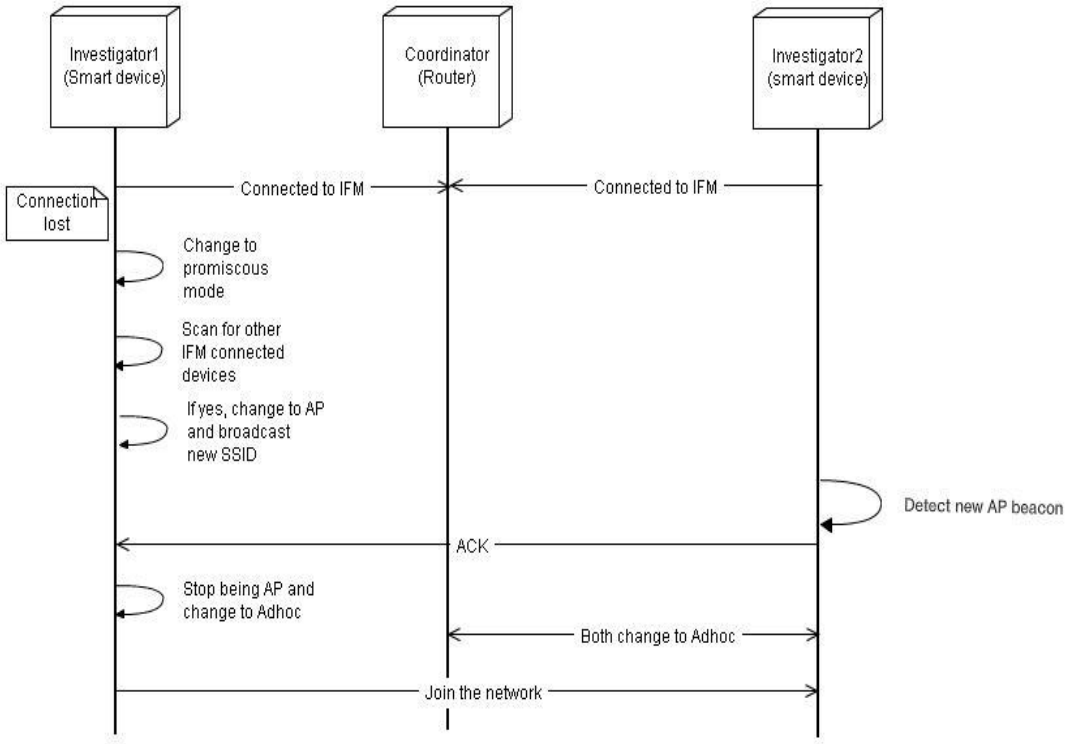


Figure 12: Neighbor discovery

Context Adaptation

Transport Layer

There are many advantages in using adaptive measures in transport layer. We can envisage our autonomic system in an efficient way by diagnosing the problems from the network, and decisions and action are handled by the higher layer, here the transport layer. If there is no solution, then it triggers the mission. Some of the adaptive measures at transport layer:

- Hide error losses from sender
- Switching the protocol
- Packet delay
- Discarding packets
- Determine packet loss

There are some adaptive transport protocols of adhoc technology. They are

- Ad Hoc Transport Protocol (ATP)
- Ad Hoc Transmission Control Protocol (ATCP)

Note: Transmissions in ATP are rate-based, and quick start is used for initial rate estimation. The congestion detection is a delay-based approach, and thus ambiguity between congestion losses and non-congestion losses is avoided. Moreover, in ATP, there is no retransmission timeout, and congestion control and reliability are decoupled. ATCP improves TCP performance by maintaining the high throughput since TCP's unnecessary congestion control is avoided. Also it saves network resources by reducing number of unnecessary re-transmissions.

Possible parameters

Transport layer:

1. Missing packet rate
2. Packet out of order rate
3. Packet retransmission rate
4. Application layer packet size
5. Maximum data rate handled by the receiver
6. Effective network data rate
7. Maximum network transmission unit
8. Transport layer delay overhead
9. Round-trip-time average
10. Round-trip-time variance
11. Fragmentation rate
12. End-to-end bulk delay
13. End-to-end Packet Jitter

14. End-to-end hop count
15. Application specified parameters if any

Routing

Proactive routing: Here, the nodes maintain a global state information and consistent routing information are also stored. When there is a change in network topology, this information is propagated to all the nodes and the corresponding state information is updated.

On-demand (Reactive) Routing: A path is computed only when the source needs to communicate with a destination. Here the source node initiates a *Route Discovery Process* in the network and after a route is discovered, the path is established and maintained until it is broken or is no longer desired.

Performance Metrics

- Hop Count – more could lead to poor throughput
- Link quality – all links do not have the same quality
- Stronger links can support higher effective bit rates and less errors/retransmissions
- Interference also can affect link quality
- Link quality is proportional to the SINR (Signal to interference and noise ratio)

Possible parameters

Network layer:

1. Routing cost (may be quantified using round-trip-time/bulk delay/ number of hops to destination host etc)
2. Delay jitter (variation of delay between arriving packets irrespective of order of transmission)

MAC/Data Link Control layer:

1. Bit Error Rate,
2. Frame loss rate (frames ACKed/framessent),
3. Effective throughput (over both short and long durations),
4. Variance in throughput
5. Variance in bit error rate
6. Link congestion indicator
7. Maximum frame size
8. Frame overhead (as a percentage value)
9. Retransmission attempt rate
10. Frame error rate
11. Congestion Indicator

Chapter 4 Proposed solutions

In ROSACE scenario, actors like Fireman, Robot need to communicate each other to reach a common goal. Each actor is equipped with handheld devices (PDAs) and communication technologies, and should carry on specific tasks. The whole team carries a process and collaborates through the interleaving of all the different processes. When a disaster happens, the control center is responsible to make an action and to provide emergency assistance for victims by setting up a team. The team tries to stabilize the situation and reduce the probability of secondary damage and speed recovery actions (aim to return the living conditions to normal).

The control center acts as back-end, providing advanced services requiring high computational power whereas ROSACE front-end devices provide services to team operators. The control center is constituted mainly by high power computer integrated with data, knowledge and content. Team operators equipped with mobile devices, connected in an ad hoc that carries on a process, in which the adaptive to connection/ task anomalies is fundamental.

After a disaster such an earthquake, a team equipped with mobile devices is sent to that area to help the victims. Before this process, the control center must gather information about the site map, a list of the important objects at the site, and some previous reports and materials obtained by organizations like Telecommunication industry, public and private organizations, etc.

NOTE:

The actors owning the ROSACE device can form a mobile ad hoc network in which the team leader's device coordinates the other team member's devices by providing appropriate information and assigning activities. Mobile Ad hoc NETworks are networks of mobile devices that communicate with one another via wireless links without relying on an underlying infrastructure. This distinguishes them from other types of wireless networks: for instance, cell networks or infrastructure-based wireless networks. To achieve communication in a manet, each device acts as an endpoint and as a router forwarding messages to devices within radio range. Manets are a sound alternative to infrastructure-based networks whenever an infrastructure is no longer available, or can't be used, as in emergency scenarios.

We use Unified Modeling Language (UML) to specify, visualize, modify, construct the ROSACE framework as it offers a standard way to visualize a system's architectural blueprints, including elements such as, actors, activities, etc.

Actors

Thanks to UML, formalizing the main actors (supervisor, Robots, firemen) and their roles in ROSACE scenario is shown in figure 1. The main functions of the supervisor are to control and communicate with the actors and to create the communication network which is the same for coordinator that is to create the local communication network and to control the

investigators. Coordinator also monitors and sends information to investigators. Whereas, the role of investigator is to communicate with other RSOACE team member in order to help the victim or its own team member.

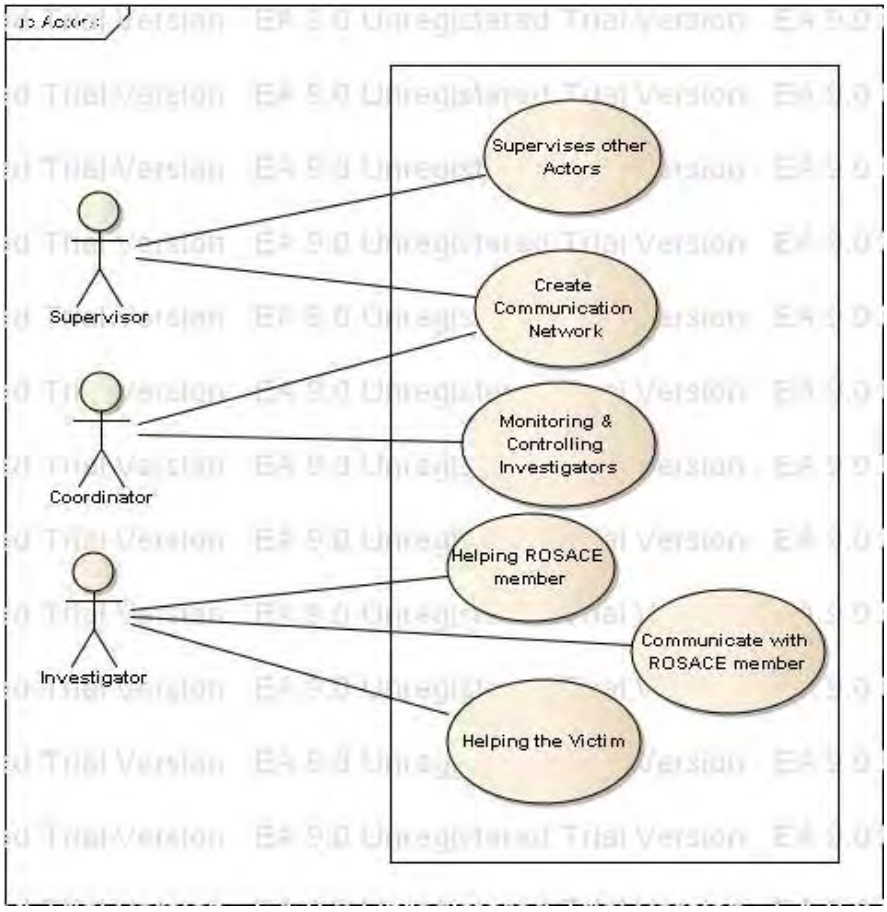


Figure 9.1: Use cases associated to Actors

Coordinator

The coordinator actions depend on the type of deployment but the main goal is to act like a mediator between the supervisor and investigator. The communication between supervisor and other coordinator is called coordination flow. In our scenario, there are mainly two types of coordinator, i.e. Robots coordinator and Firemen coordinator.

Note:

The following diagrams give an example of team's observable behaviour. Communication diagrams and Messages Sequence diagrams are used later. The goal could be achieved through more sophisticated dialogue among supervisor, coordinators and investigators, for example to allocate new telecom resources.

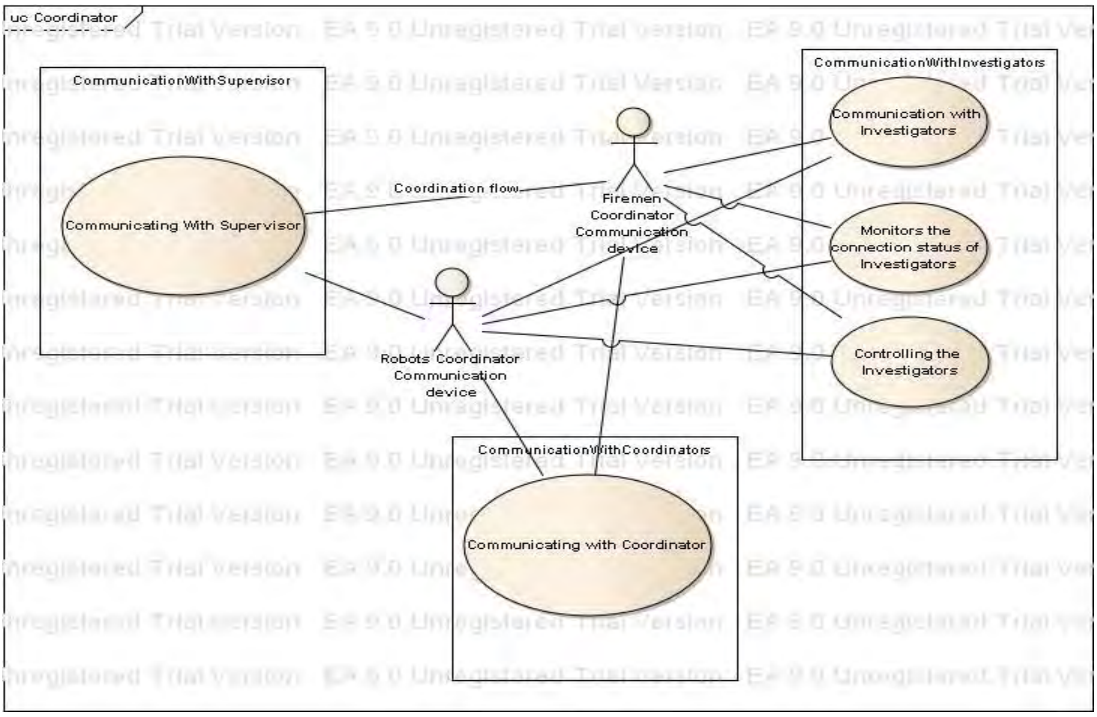


Figure 9.2: Use cases for Coordinator's functionalities

Investigator

The main communication functionality of investigator is to change the communication mode with coordinators and with other investigators.

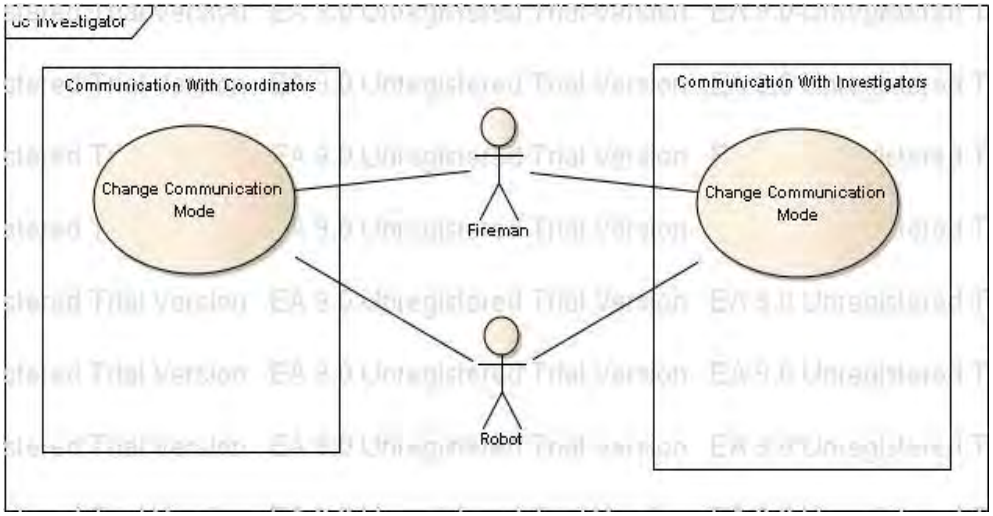


Figure 9.3: Use cases for Investigators

9.1 Communication Agent's (CA) Information Processing model

Figure 4 depicts a set of CA's components which could be considered for implementing the functionality defined in the following use cases. This CAs need to be validated or modified

during the analysis and could be the starting point for identifying the agents and the computing entities that should detailed in the design phase. The main components of CA are explained below.

Communication node manager

This manager collects and organizes the local network configuration and monitors the context information. This information is used for the purpose of both normal functions and adaptation. Local network information includes networkID, connected member list, number of network interfaces (e.g. Bluetooth, WLAN, GSM, etc). The other task is to create and maintain a channel (the logical link between physical application components that are located in different devices e.g. PDA or mobile). To transfer data, each channel uses a specific type of connection and if there is a change in policy or when there is a change in the network context, the channel can be dynamically changed.

End-to-end network information is monitored. It includes connection information (route availability, best route), network QoS parameters (availability, available bandwidth, delay, response time, jitter, loss). By explicit query, we can obtain these pieces of network information from underlying infrastructure.

Communication resource Manager

The manager is used for controlling the configuration parameters, traffic workload at local interface, error rate, signal strength, power consumption, and operation status (e.g. available, operable, connecting, connected, sleeping, idle, transmitting, receiving, unconnected, unreachable, disabled, etc.). It calculates packet statistics information, e.g. received, sent, etc. and the dropped protocol packets. Thanks to this manager, communication with protocol stack is achieved. For the energy management, the resource manager calculates the remaining time left out for the services after checking the energy consumption utilized by WiFi, Bluetooth, etc. According to the default values, the resource manager notifies the communication node manager that a threshold is reached.

Communication Service Manager

This manager computes the resource, monitors the application status, reports in case of failures, and negotiates the policies. The manager performs fault detection and recovery activities, to negotiate resource utilization and QoS requirements with the underlying infrastructure.

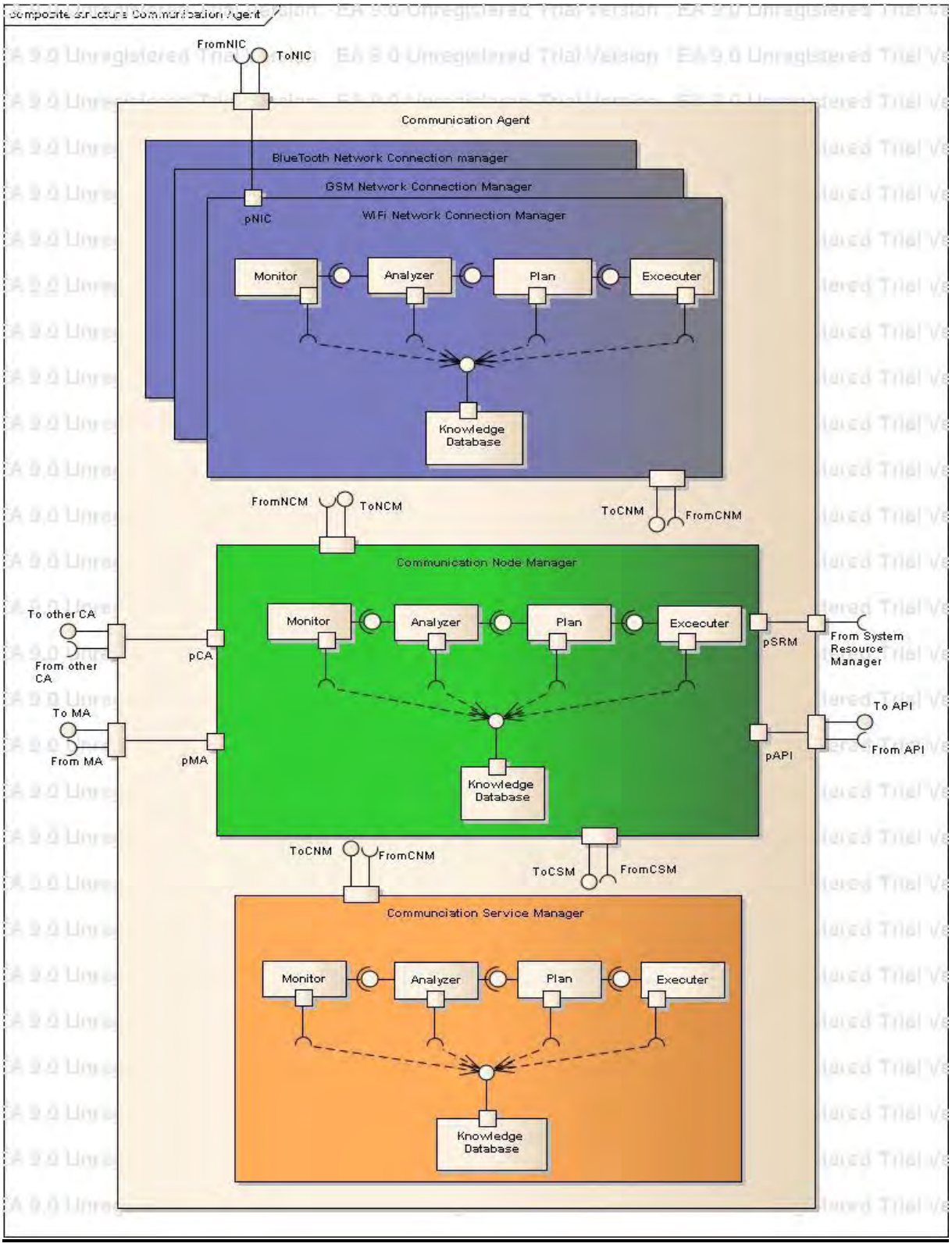


Figure 9.4: Communication Agent Composite Structure diagram

Decision Module/Plan

Plan is used to clarify the adaptation purpose and the boundaries of the decision to be made. It gathers information from the knowledge database and foresees the future if something gets affected by the decision. It identifies and suggests the knowledge or expertise to context management to make an action and recommends the currently existing resources to help with the decision process. It communicates to other functional modules and use appropriate knowledge for the integration of ideas and also the negotiation and prioritization of ideas. Then it helps the adaptation management to implement the decision. Finally, it summarizes this action and log into the database for future predictions. We use SWRL rules to define our adaptation policy. The application designer defines these rules according to context changes he wants to handle. As we explained earlier, each level executes its own rule when there is an appropriate need. Thanks to the decision model at every level, not all requests are passed to the supervisor. If there is no solution for an event at a particular level, then it triggers the higher level. At high level, SWRL will be used if there is a need to establish a cooperation flow between the investigators among the different groups. At low level, if the energy level of a participant's device reduces below the threshold, decision will be made by triggering the SWRL rule.

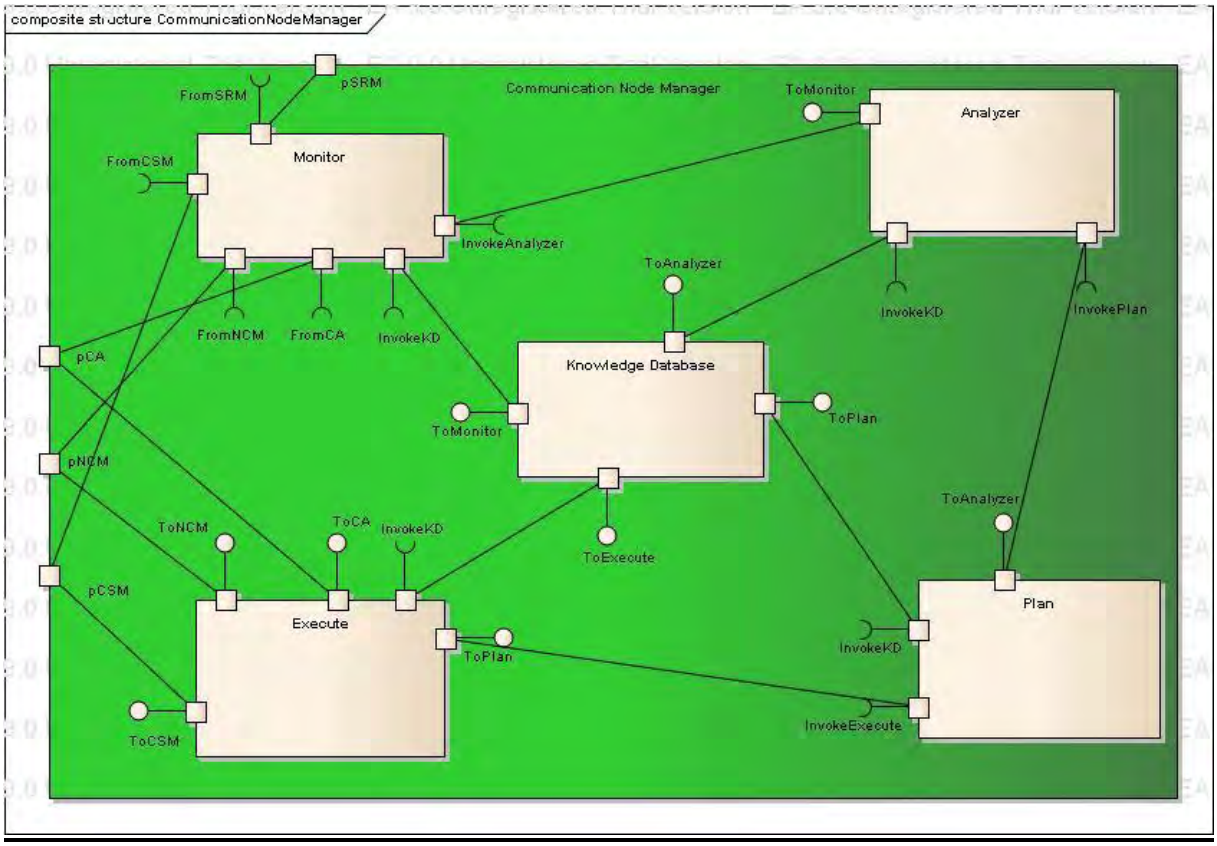


Figure 9.5: Communication Node Manager Composite structure diagram

Adaptive Executor

It is responsible for adapting the application by calling and properly linking the adaptation services. After the execution of the services adaptation process, the executor ensures the adaptation to the context situation. Also, it triggers the higher level for more complex issues. Context management is highly dependent on the environment, i.e. on the physical context properties and the dynamic environment, the mission. We encapsulate such dependencies in a generic context provider, which is also responsible for managing the aspects of context related to the node, such as priority and role. Since the captured context may not be meaningful to the mission in present situation, this executor contacts with other components to trigger the desired action. The trigger could be the low level (communication) context or high level (mission) representation. Once the decision is made, interactions between the components assure the desired action.

Adaptive Framework

Figure 9.5 provides the composite structure diagram of communication node manager. It's based on MAPEK architecture and it clearly indicates the entry point for this module and also the exit point. The ports allow the internal components to invoke each other for a particular task.

Note :

The plan (decision model) is not the same for communication node manager, network connection manager and communication service manager. Each module has its own algorithm And thus the adaptive mechanism is distributed. It is also important to mention that the functionalities of investigator's CA are bit different from functionalities of coordinator's CA. This section provides a brief overview of studies that have been discussed in the scientific community on network monitoring.

Resources

Managed resources are controllable components located in the environment to capture metrics or act on it. This may be a single resource or a set. These sets can be composed of other resources. Then there is the presentation hierarchy. All the resources form the environment in which the manager can act, he may be composed of resources supervision or not.

Checkpoints

They serve as an interface for resource management supervision. They are two types of checkpoints: sensors and actuators.

- Sensors (or checkpoints dedicated catches) are typically used to transmit events or properties to an autonomic manager.
- Actuators (or checkpoints dedicated actions) are used to make changes to the environment to change the configuration properties or to change a state.

The combination of sensors and actuators form the management interface that allows the use of an autonomic manager. Checkpoints have properties like a name, a state, a set of measurement (sensor), a configuration.

The components of monitoring

The monitor connection interface observes the information collected by sensors; this information can be found in Knowledge Base of autonomic manager. We can check the status of wireless interfaces, verify the initial configuration and receive notifications if an interface fails.

As mentioned, the system is initially configured in infrastructure mode (config 1). Therefore, surveillance and detection will be on the state of Wi-Fi to keep the communication. The verification of the initial configuration is taken into account automatically as it was applied directly on all ROSACE devices by CommunicationNodeManager.

Communication protocol

The IEEE 802.11 protocol is designed to manage and reduce contention in the channel of communication of the wireless network in a fair manner [21]. SNMP provides a means to analyze records of network devices and provides statistics on the 802.11 standard. SNMP provides data management in the form of variables in managed systems [22]. These variables can then be queried by management applications. SNMP does not define the variables that are accessible, but instead uses an extensible design, where the available information is defined by management information bases (MIBs) that are often owned by individual suppliers. MIBs describe the structure of the data management sub-system device in a namespace (namespace) hierarchy containing the object identifiers (OID). Each OID identifies a variable that can be read or set via SNMP.

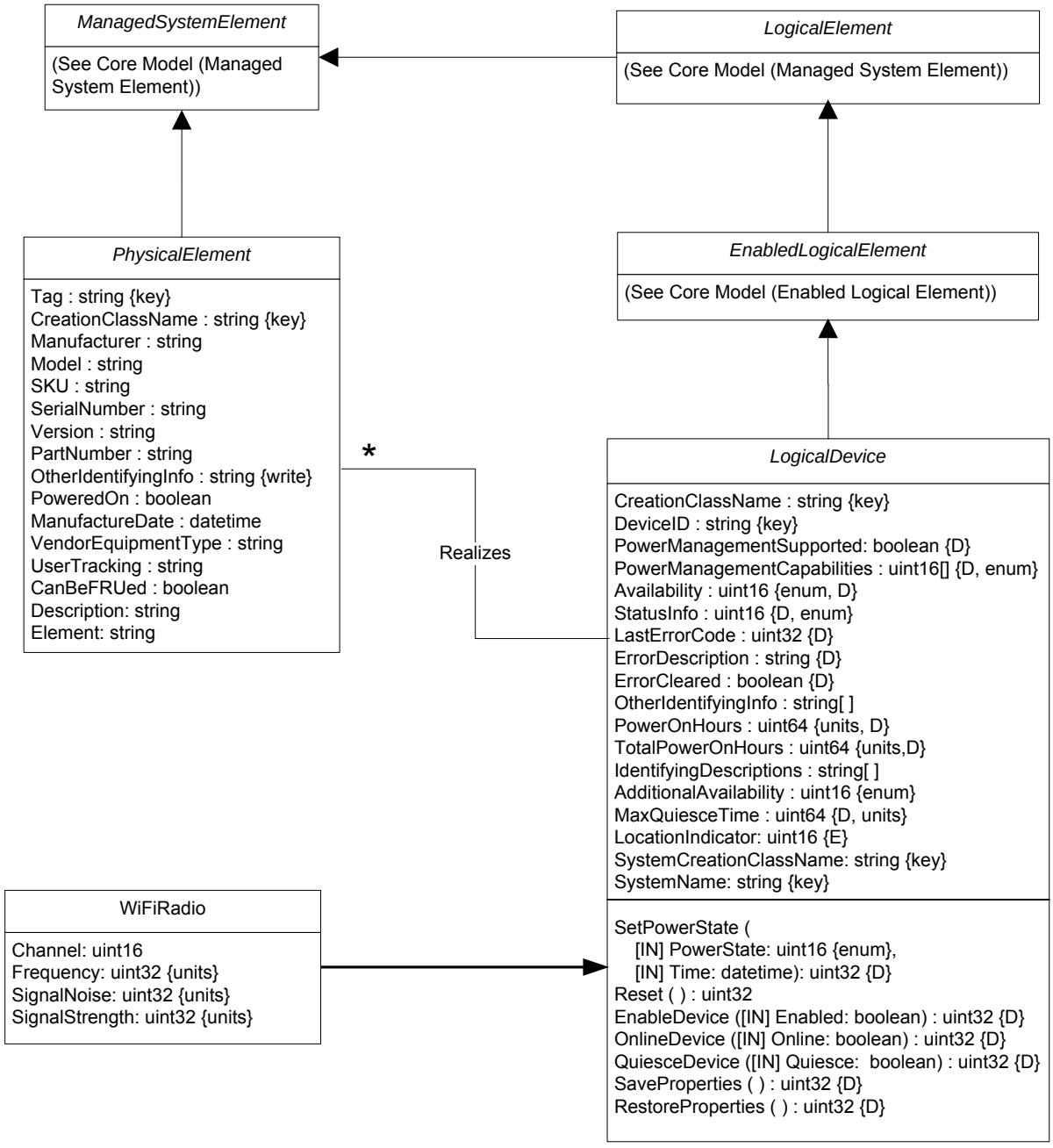
There is a standard MIB for IEEE 802.11 wireless networks, "IEEE802dot11-MIB." Some of the OIDs have defined AP configuration settings [23]. But Cisco has expanded the base of the 802.11 MIB with extensions that monitor events that are specific to users. This extension is called "Airespace-WIRELESS-MIB" [24].

Airespace MIB-WIRELESS-MIB collects statistics that are specific to our work through the SNMP table "bsnMobileStationStatsTable". This information includes the number of packets sent and received, the signal / noise ratio (SNR), an indication of the received signal strength (RSSI), etc.. And because of the hierarchical design of SNMP, we may be able to query this OID [24].

Information Model

The information model introduces the concept of managed object. A managed object is an abstract representation of a logical or physical resource that must be managed. This information is represented in the database management. Each managed object is named in a

unique way with a general structure of naming objects based on the concept of naming tree. Schematic model representing the CIM managed objects:



CIM reference model for monitoring

The common information model (CIM) offers as part of the consortium WBEM (Web-Based Enterprise Management) of DMTF standards of existing instrumentation and management through a common formalism for the specification of managed resources. Managed elements

are organized in the form of class diagrams close to the UML specification. We use CIM as a model of information management based on standard UML (Unified Modeling Language).

A meta-model, a naming system, a number of kernel models are grouped into three levels of abstractions, a specification language and a graphical notation, have been specified for the definition of models. It has been defined for CIM meta-object-oriented model. Thus, it takes the concept of class, property, method, association, reference, and schema qualifiers. More concepts from the management of networks and services have been integrated: the indications, the qualifiers Read / Write, naming.

This model is semantically rich and highly extensible. The concept of association is a very interesting one. In modeling phase, it allows you to specify semantic relations such as high dependency or the composition. In the operational phase, it can navigate and explore bodies in connection from a source instance.

The state machine

Informally, a state machine can be characterized by a finite set of states, a finite set of inputs, a finite set of outputs and a function for each state provides the next state based on the value of inputs and the internal state of the machine [Courvoisier and Valette 1986]. The graphical representation of a state machine is:

• A state is represented by a circle,
• State changes are represented by arcs labeled by the combination of inputs to the change of state matching.

Table 1 : Table of detection

Observed Points	Actions	Messages
No connexion (no SSID)	Changement of mode (configuration)	AP Suspected
No Signal (no RSSI)	Changement of position	Out of range
NIC inactive	Activation of second WIFI card	Carte Wifi inactive

Graphical representation of state changes

Encoding of states

* 0 - No signal NoSignal
* 1 - Weak Weak signal
* 2 - Poor Signal a little weak
* 3 - Fair signal through
* 4 - Good Signal Satisfactory
* 5 - Excellent Excellent Signal
* 6 - Signal Weak2 still low after T seconds

* 7 - Init Initial state, no information yet on the signal

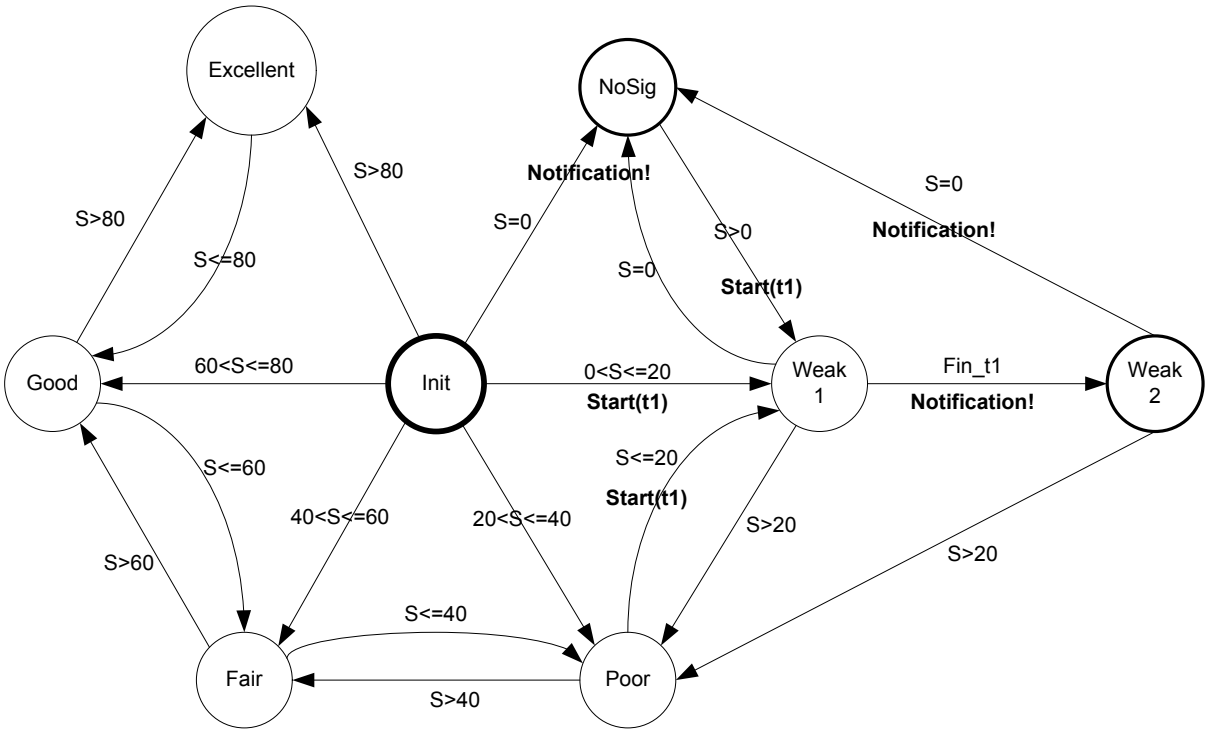


Figure 20: State Machine for the detection

Figure 20 shows a state machine to an initial entry and show all the state changes that may contain the signal.

7.4 Reconfiguration

RECONFIGURATION means the change of the configuration parameters of a system. In other words, changing the configuration of a system so that it adapts to new conditions.

Reconfiguration occurs after failure detection, it can correct the application in order to find an appropriate way of responding to the healthy functioning system [10]. The reconfiguration is then the step to implement the new configuration.

Types of Reconfiguration

Two approaches are possible to complete the reconfiguration; one is static and one dynamic [11]:

a) Static Reconfiguration

Adaptation and static configuration are those carried out before or at the beginning of the execution of the application. For example, adaptation can be static at compile time; the reconfiguration takes the form of a recompilation. But it can also be launching the application

through settings or configuration files. In all cases, this requires knowing the execution context at the time of static adaptation. Such adaptation will therefore have meaning only if the context does not change (or little) when running. In the static reconfiguration, the application is stopped to make the changes and then restarted.

b) Dynamic Reconfiguration

If the adaptation and reconfiguration are performed during the execution of the application, they will say they are dynamic. Such adaptation may occur several times during the execution of an application. This case is suitable when the context is subject to change at any time. The application requires in this case a mechanism for dynamic reconfiguration.

In dynamic reconfiguration, changes are being implemented with, not the total cessation of the application, but just stop that part of the application concerned by the change. Here the term flexibility comes into its own in the sense that a change does the complete cessation of the application but may be running.

Illustration of reconfiguration

we have an example of adaptation and reconfiguration of the unit of Fireman 2, after it lost its connection with config1 or in Infrastructure mode to switch to config2 in ad hoc mode. This process is done after the detection of the failure, then comes the decision to change the configuration and then apply the new configuration by the Action

The interface can have several states: on, off or unknown.

 The IEEE 802.11x (also called the NetworkInterface) provides information on the type of wireless network and identification of the service. The X is a variant that can take the values a, b, g, n ... in general, 802.11b is used by default.

 The acronym ESSID is a string of 32 characters that identifies the domain that owns the network interface. This is the first option to configure when the installation of a station to a new network.

 This option "channel" provides information from a wireless interface based on the frequency of the channel used for communication.

 To secure the network, it uses a key of hexadecimal digits or ASCII (also called the encryption key or encryption). Encryption can be a key 64, 128 or 152-bit rarer (is not often implemented in the current network cards). The higher the level, the higher the "discovery" of the key is difficult and therefore the better the security.

 It is the MAC address (aka BSSID) of the access point is associated with the interface and the quality parameters of the radio coverage.

Illustration of use cases

Case 1: Configuration in Infrastructure mode

In this case, the establishment of such a network requires at regular intervals to ask terminals (PA) in the area to be covered by the network. Terminals and devices must be configured with the same network name (SSID = Service Set IDentifier) and uses a channel in order to communicate and verify who is accessing the network.

On the following scenario, we have the first phase of the connection. All devices are tested before being deployed, which means they all have a valid configuration, by default, which is the body config1 to connect to the network "FiremenInfraWifi." The firemen are connected to the PA Coordinater1 through channel 1.

Case 2: Configuration in Ad Hoc

Mode "Ad-Hoc" is a mode that allows you to directly connect computers with a Wi-Fi card, without using a third-party hardware such as an access point. The establishment of such a network is limited to configure the devices in ad hoc mode (instead of Infrastructure mode), the selection of a channel (frequency), a network name (SSID) and common to all if necessary, an encryption key.

In this scenario, the devices are initially connected in infrastructure mode and Firemen 1 and 2 have lost connectivity given the criticality of the connection. For this, so they changed network or configuration (config1 to config2), to continue to maintain some connectivity in Ad Hoc mode.

Case 3: Configuration in Ad Hoc mode to test

This instance of the class Config3 Wifi_AdHoc_Setting used to test to find where the shortcomings in terms of connectivity are. In this case, the devices must pass in promiscuous mode.

This scenario show that has lost its connection FM3, and FM2 is still connected to the PA is trying to switch AdhocTest, or Config3, in order to identify the problem of disconnecting the FM3.

Operational scenarios

We describe some use cases to achieve mission goals aiming to set up the connectivity infrastructure and the connectivity services in the intervention area.

Use case 1 - Establishing Network Connection (Local Network Set-UP)

One of the first tasks to be achieved on site is to deploy a local telecom network which provides the global network connectivity for actors involved in the mission. The diagram 6 shows actors and the telecom resources needed to set the local network.

Situation

The ROSACE team is situated at the intervention area. Fireman and Robots are also in the area for fire supervision and to find an injured victim. The CA embedded in AP which is a coordinator here (truck or robot) takes the responsibility of creating the local network. This process should be done automatically.

Goal

The aim is to define the activities and actions of the CA for creating the network. Identification of possible collaboration with other team members, and utilization of specific resources is also considered.

Scope & Level

The use case will focus on illustrating the flow of activities, actions and information, needed for achieving the goal. Further refinement of activities should be considered in order to complete goal's resolution life cycle.

Environment

Specific assumptions for the Use case are the following:

- Hot spots are located at truck and propagates the network ID
- Fireman' ROSACE terminal has WiFi capablities
- CA's embedded in Fireman's Terminal, Robots and AP

Hypothesis

- *GSM*

The idea is to deploy the WIFI network as the primary one and in case of failure of WIFI, CA chooses the available network access as secondary one (GSM). For example, sending video data using GSM is not recommendable as it lacks performance. Thus GSM acts as a substitution network means when there is a connection loss at primary one, GSM will be used to reactivate the WIFI network.

- *Deploying Adhoc*

Switching between WIFI status (IFM or Adhoc) depends purely on the status of the mission. Let's consider a lost fireman's CA sends its GPS position to its coordinator through GSM. The coordinator chooses one of the connected robots and send request to move to a position to get connected with the lost fireman. After receiving the message, robot's CA changes its IFM to Adhoc. Thus, when the robot approaches the Fireman's CA, the connection will be automatic as the device is already in Adhoc mode. The time dependant of this activity will be detailed in the plan.

CA's internal state

Once connected to the local network, the CA's state will be either connected or not connected. This result is achieved by the coordinator's CommunincationNodeManager which role has the responsibility to set up of internal telecom resources. Figure 7 depicts activities involved in Node connectivity set up.

Modeling network connectivity status for a single node

Initially, ROSACE devices connect to its coordinators through IFM. But due to change in the environment, the connectivity status of a single node varies due to different reasons. In our assumption, the other two possible connections are either through adhoc or through GSM which is shown in the diagram 8.

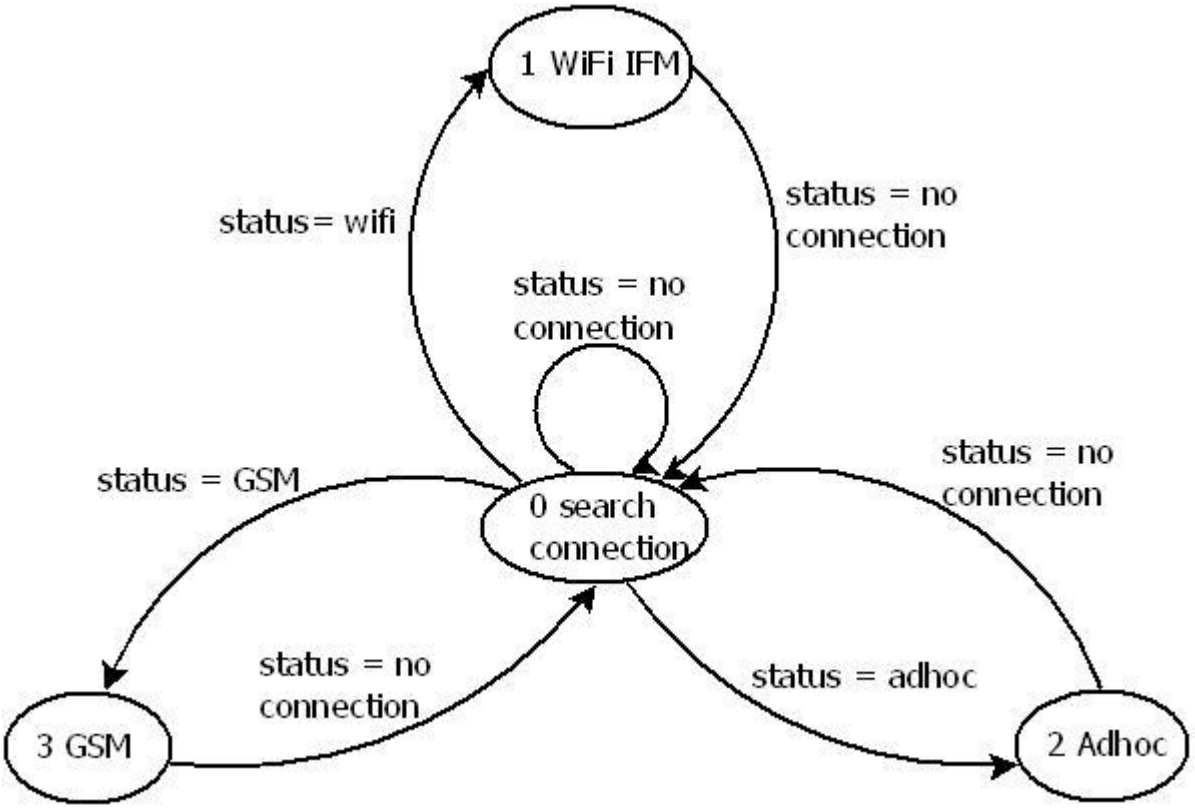


Figure : State diagram for network connection

The basic assumptions to achieve network connection will be followed:

• Each device should include hardware that lets it know its communication distance from the surrounding devices that are within radio range. Specific techniques and methods are easily available, i.e., TDOA (time difference of arrival), SNR (signal-to-noise ratio), and the Cricket compass.

• Each device is equipped with GPS hardware, even though the devices are low-profile.

• At start-up, all devices are connected (that is, each device has a path to any other device). Each device doesn't have to be within range of any other device, but it requires at least a loose connection, guaranteed by appropriate routing protocols.

• The coordinator predicts disconnections and manages its members for reassignment of the process tasks.

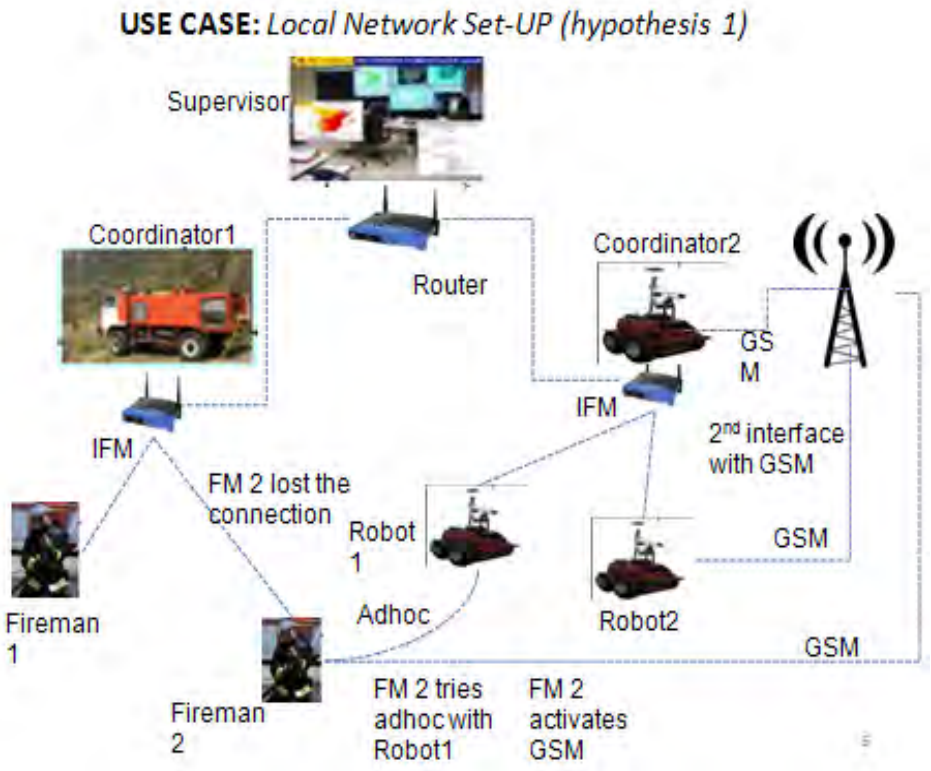


Figure : State diagram for real time communication

Note: Here we use GSM as a substitution network as it lacks performance for high bandwidth.
Role of Coordinator's CA

Coordinator's CA would direct a "bridge" device to follow the device/PDA that's going out of range, maintaining the connection and ensuring a path between the devices. In this way, the CA, on the basis of the disconnection prediction, schedules the execution of new, unforeseen activities. Such an adaptive change of the process is managed by the coordinator, which has knowledge about the status of all the devices and takes into account idle devices, operations that can be safely delayed, and so on. In order to support such a scenario, we have to investigate novel adaptive techniques for cooperative work and workflow management among ROSACE devices.

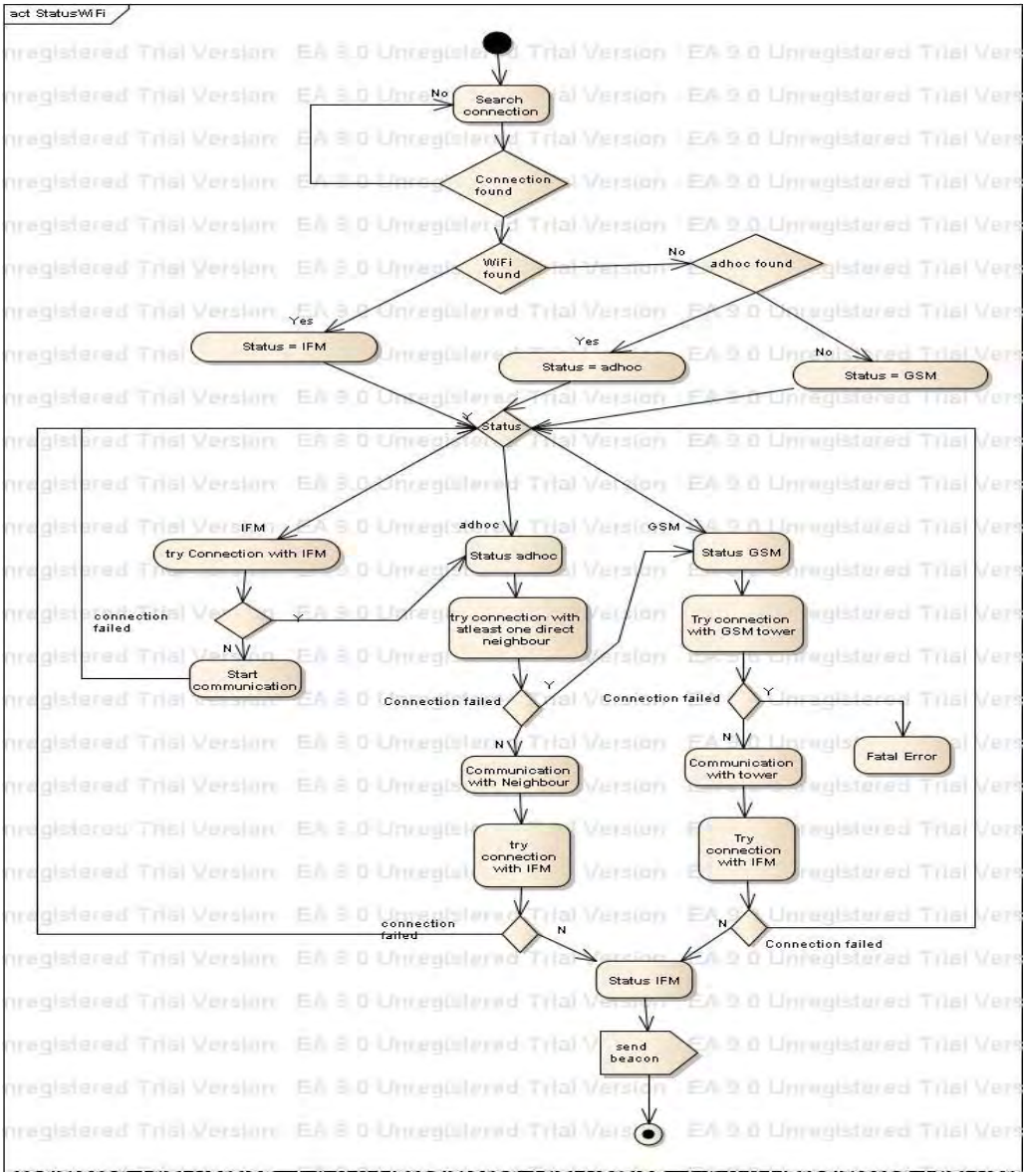


Figure : Activity diagram for WiFi connection

Figure 9 explains in detail the connectivity status of a single actor. The idea is to keep the devices at IFM whenever possible thus allowing coordinator to take the important decisions. We should keep in mind that the adaptive mechanisms used in our scenario is distributed means if there is a no solution locally, trigger will be sent to coordinator to take an action.

9.2.1 Initial connection setup (automatic)

After the team has reached the intervention area, the ROSACE devices have to connect automatically to access point when the actors launch their devices. If there is only one AP, the devices connect to it automatically once they receive the Beacon. In case of two APs, the device connects to the AP from which it received its first Beacon allowing the fast network setup. As the messages between CA's internal components are very important, this draft focuses on formalizing the messages using UML.

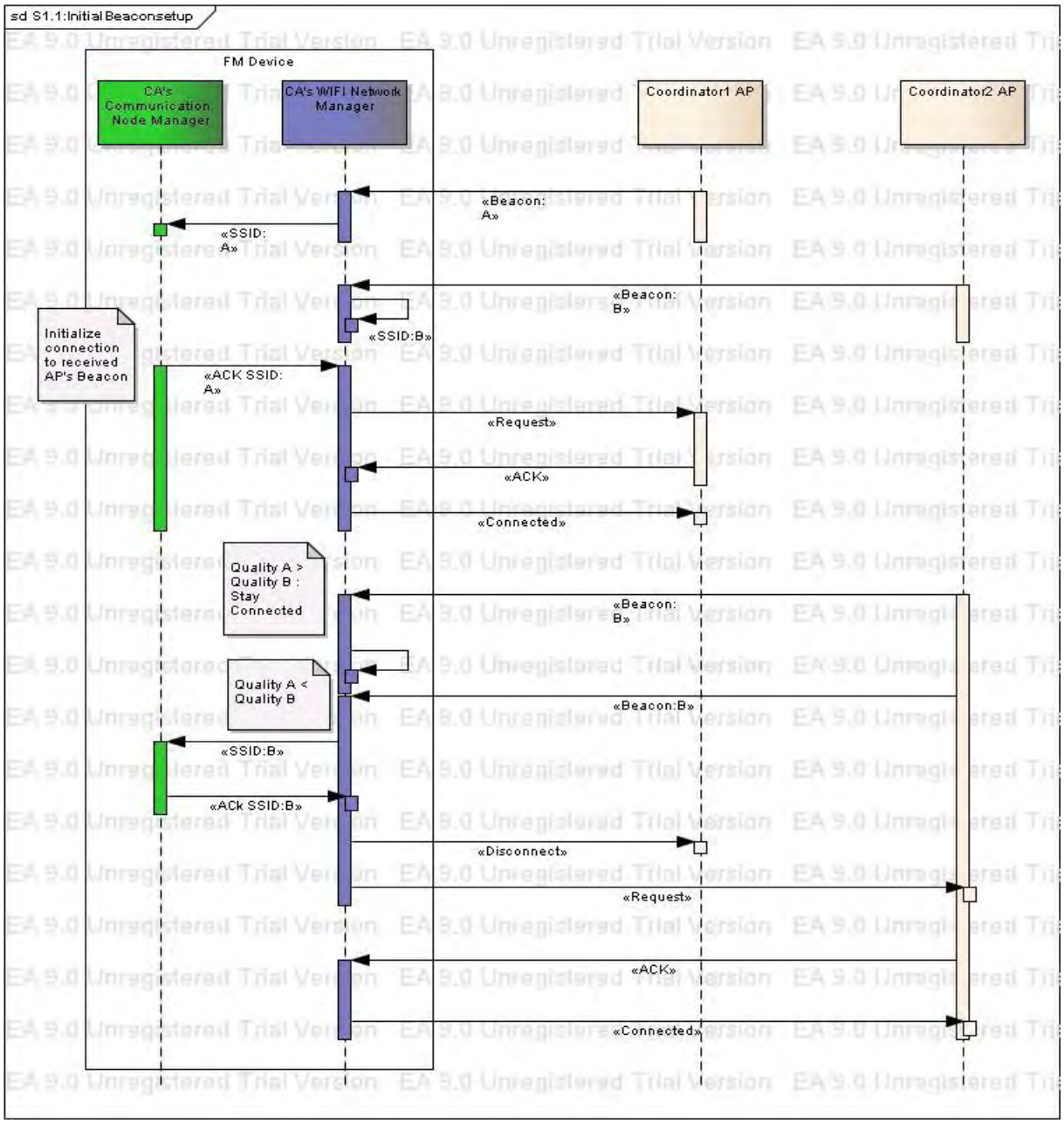


Figure : Sequence diagram for Initial connection

Quality A: Quality of signal between Fireman and Hotspot

In case of two APS, CA monitors the beacons and if there is no data transaction, CA analysis the quality of signals from the two received Beacons. If the quality of the connected AP is greater than the other one, it retains the connection otherwise; it will connect to other AP (fig 10).

Figure 11 depicts the activities involved in node connectivity setup. This activity diagram is the result achieved by CA's communication node manager when the device tries to connect the access point.

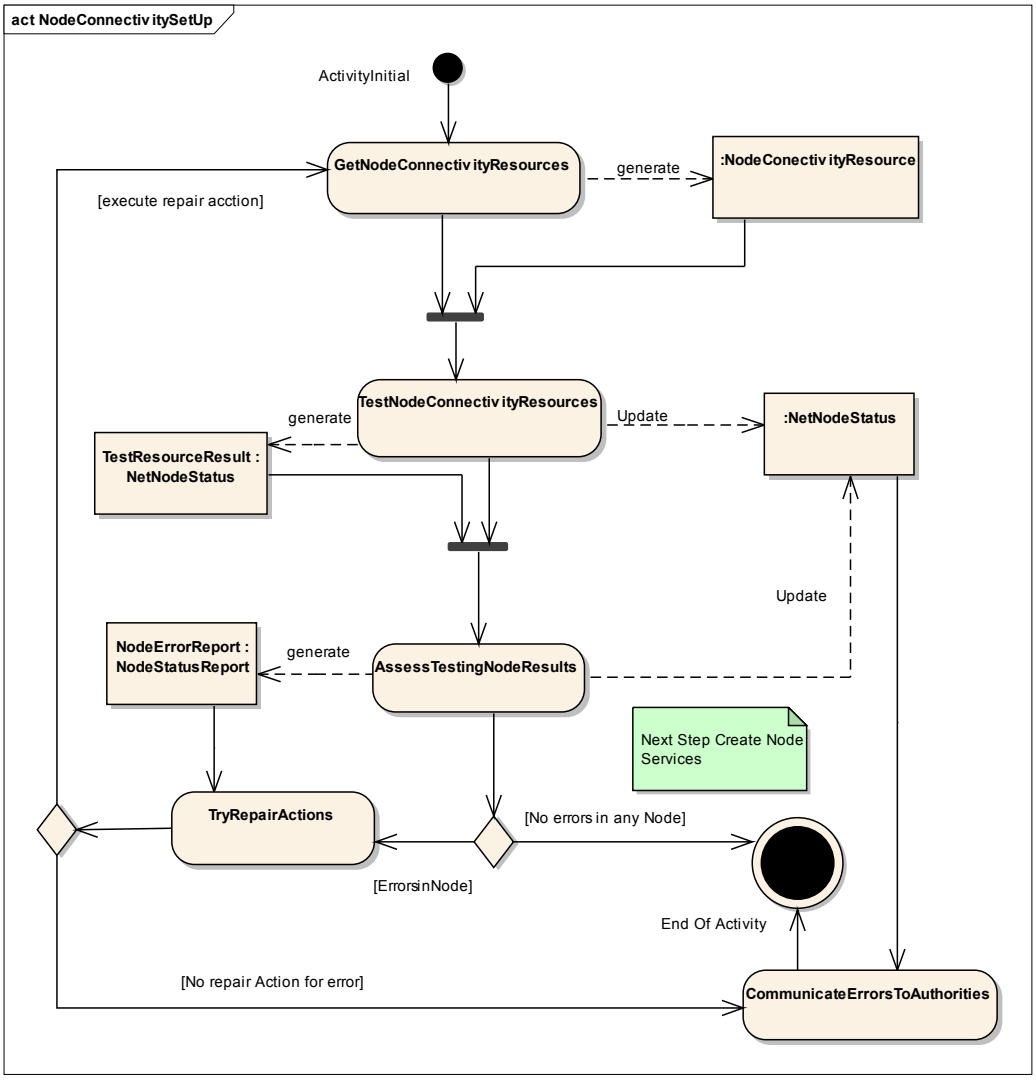


Figure : Activity diagram for Node Connectivity set up

9.2.2 Changing to new Interface

Once the monitoring module detects the failure in connection, CA analyzes the reason (no beacon received for a particular amount of time). The decision could be to wait for some more seconds (it is possible to detect a new beacon) and after this time-out, the decision module chooses the new interface. Activation of this new interface depends on the following policies. Once detected the signals of this new Interface, it will connected automatically. Here the policy is to sense alternative channels periodically (actively or passively) to choose the new interface. Figure 12 and 13 explains the steps taken to change to adhoc and GSM respectively.

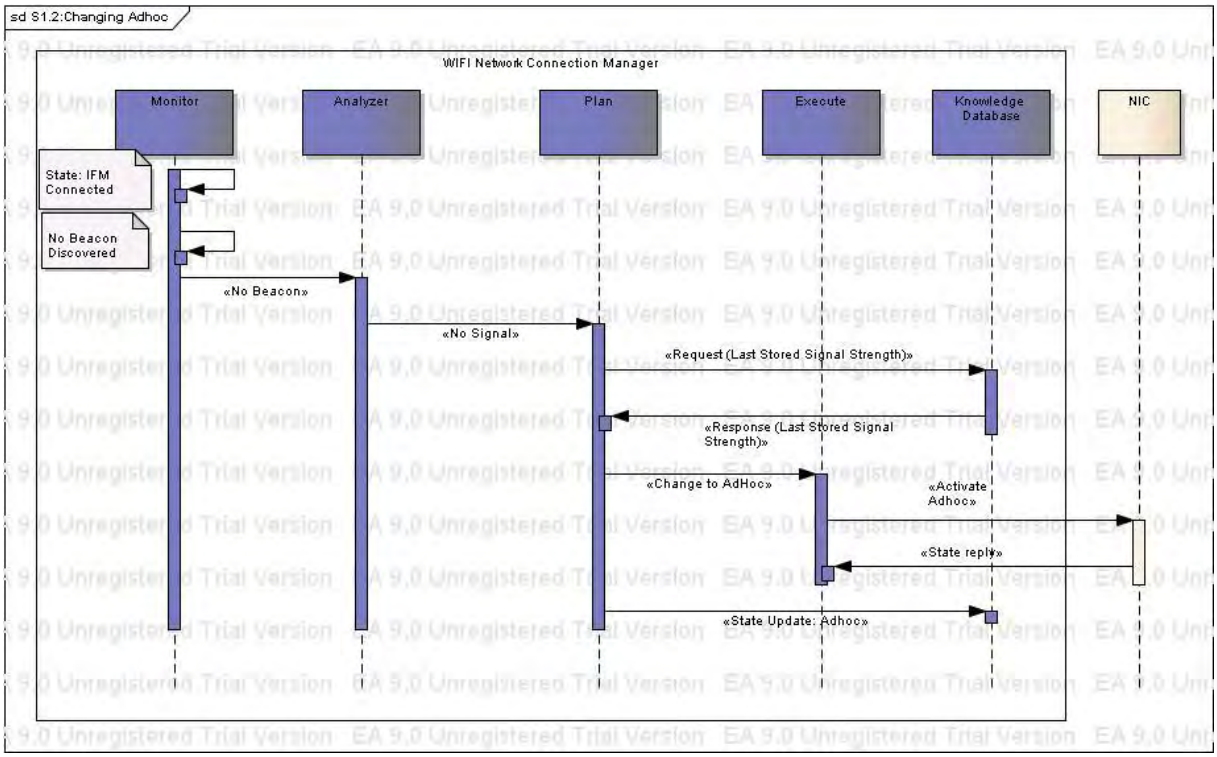


Figure : Sequence diagram for choosing Adhoc

Real challenges and solutions

It is important to describe the diagnostic methods used to analyze the state of the device: either connected or disconnected. The challenge is to find a mechanism that helps to identify when devices are not connected. We should identify the metrics needed to be monitored at different layers (PHY, MAC or IP). This measurement normally depends on the scenario.

Solution

- Timer: Periodical checking whether the device connected to network
- Routing table: Storing Mac address once connected

9.2.3 Not connected (after time out)

If the device is in non-connected state, communicationNodeManager retrieves information from the NetworkConnectionManager about the last connection status. This information contains signal strength of available access technology. According to the policies, CommunicationNodeManager send message to NetworkConnectionManager to activate the second interface card. The decision could be local if the mode changes from IFM to Adhoc mode. The messages between communication between NetworkConnectionManager and CommunicationNodeManager should be periodic. Smart plan tries to take action if the

module forsees a future connection loss. Trigger will be passed in between the modules when there is no connection after a threshold time out.

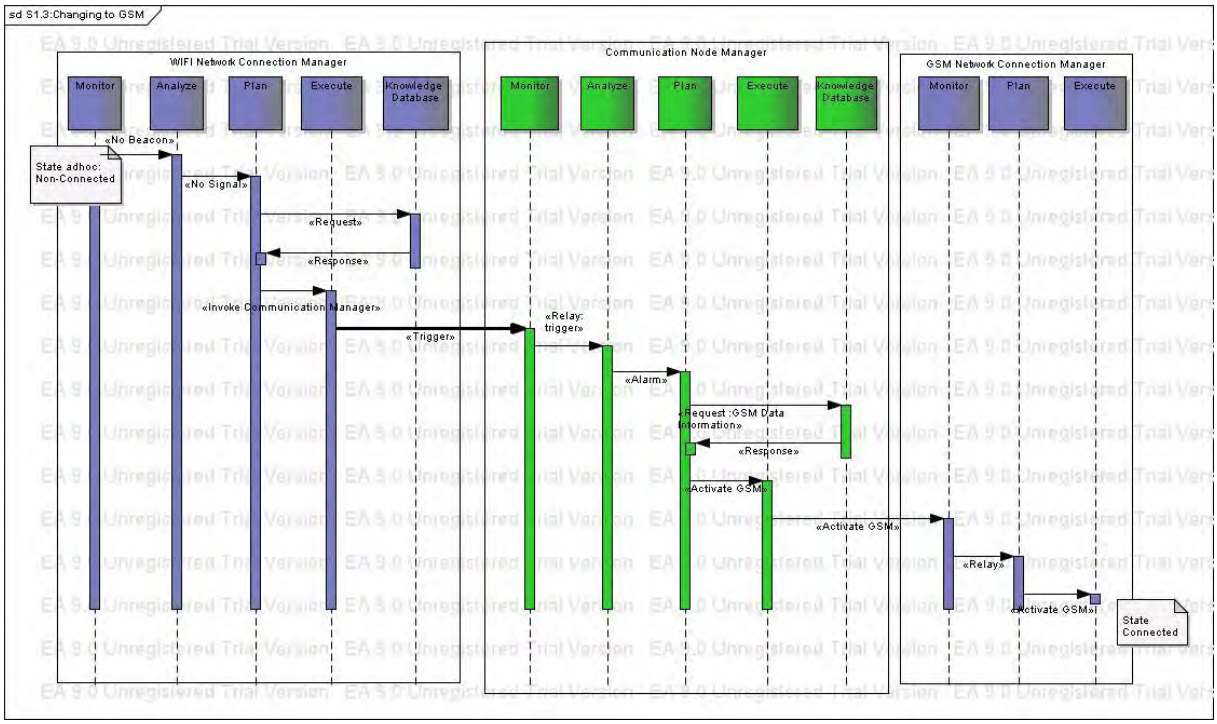


Figure : Choosing GSM

9.2.4 Deployment adhoc network

Once the fireman's device lost WiFi IFM connection, CA will connect to its coordinator with GSM. It will send its GPS position and changes to Adhoc mode. Coordinator decides an appropriate Robot or Fireman to move to a position so that the lost fireman device can be connected in the network. Here the policy is to change Adhoc mode instantly or after sometime. Once the network controller of GSM tower broadcast the GPS position of the lost fireman, the Coordinator chooses the appropriate actor (position closer to lost actor). Once chosen, it will send information to either connected Fireman's device or connected Robot to move to the desired position to get connected with lost fireman in Adhoc mode (figure 14). The policy here is to make the robot to trace the lost fireman if the device is in adhoc mode continuously.

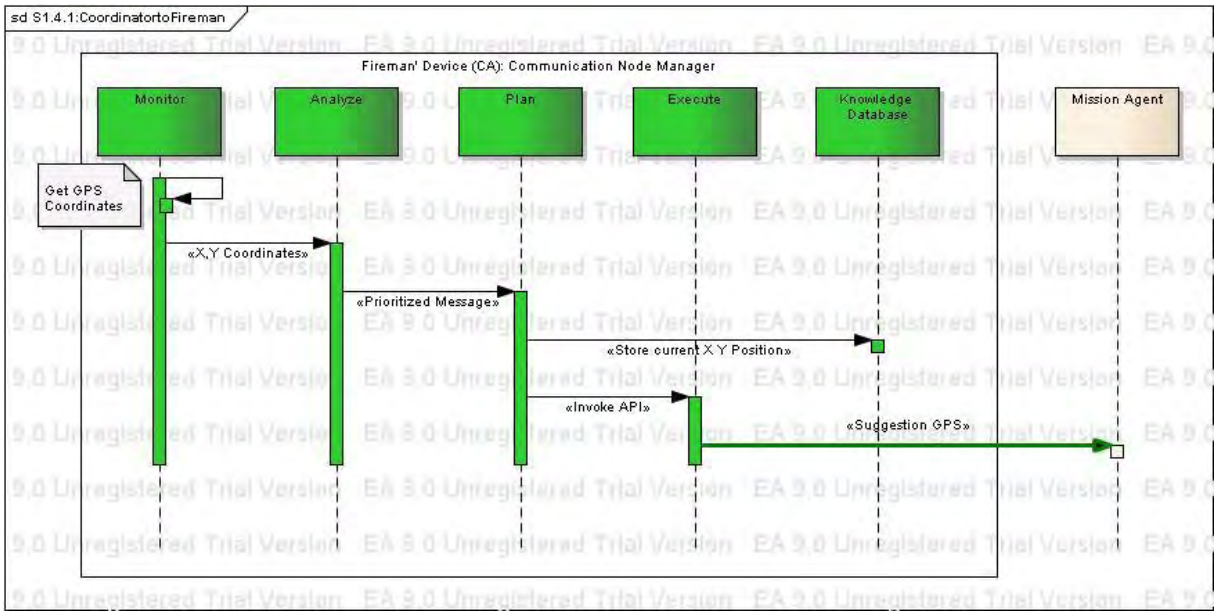


Figure 9.15 Cordinator to Fireman

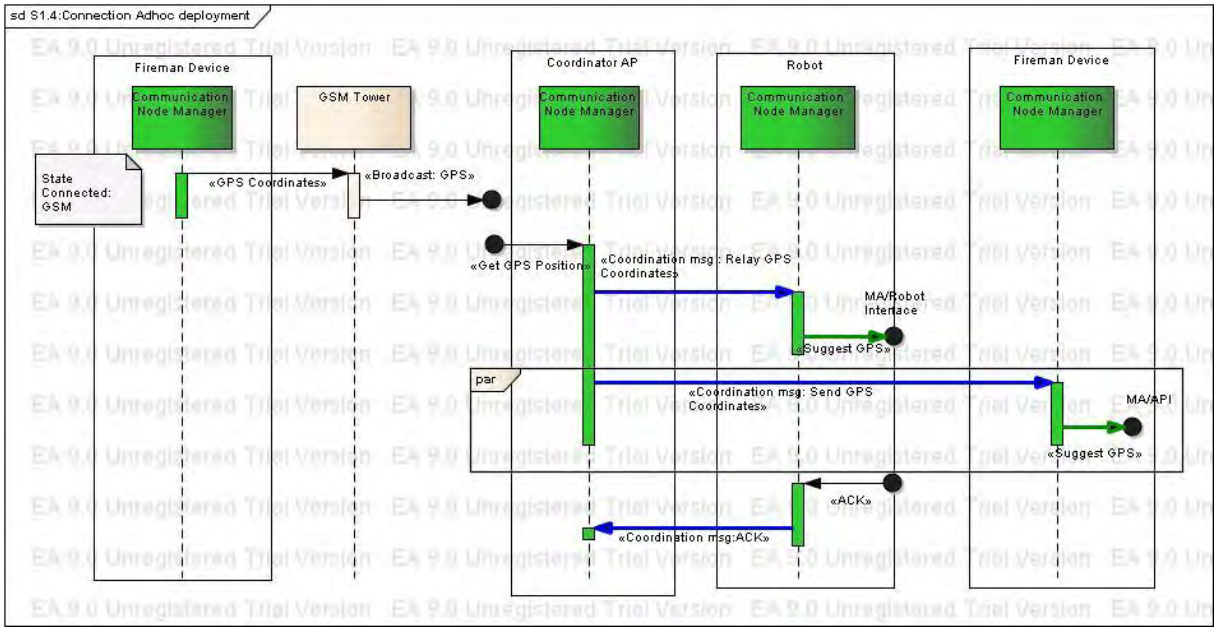


Figure 9.16 : Adhoc deployment sequence diagram

Note:

Notification to coordinator from the lost fireman device could be either by SMS or by email. But it is not sure that the data will reach the destination. Also, the plan should able to control the status of device, e.g, idle, active, sleep, etc.

9.2.5 Collaborative deployment

Collaboration is viewed as a committed effort on the part of two or more actors to devise a new understanding or solution for a decision task. In collaboration, participants agree to work on different tasks and share results. Figure 15 displays the sequence diagram of collaborative adhoc deployment.

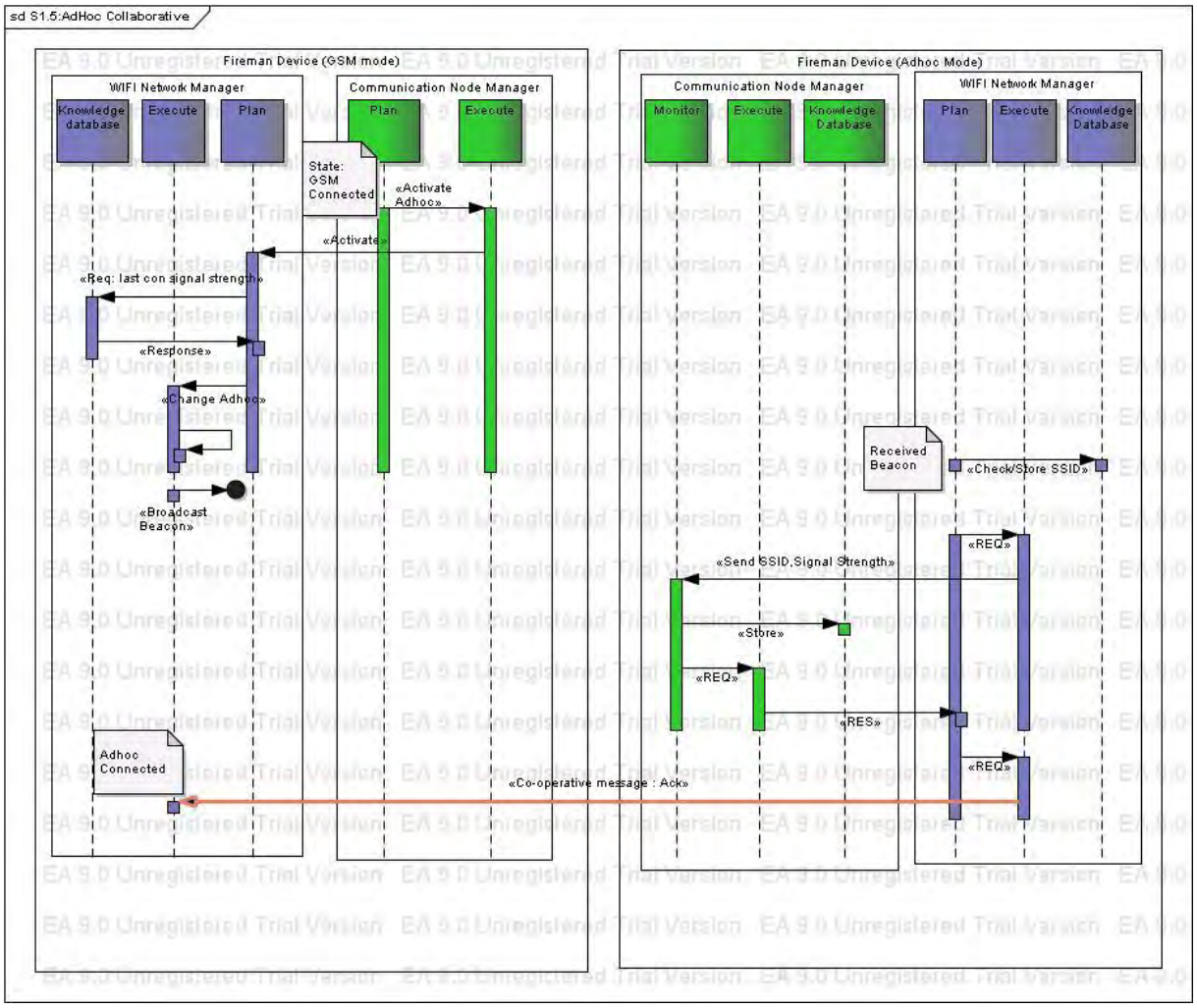


Figure 9.17: Collaborative adhoc deployment

Note:

One of the main challenges of ROSACE scenario is to monitor the status of the connection. In autonomic computing, the monitoring agent revokes information from NIC and thus the status will be updated to the database. But again, state-level monitoring is important, i.e

sometimes radio connection is not sufficient for data communication. This document shows explains the sequence diagram of of detection/dialog local of each client (connected to adhoc to his neighbor) and also suggestion to mission about relocating robot to re-establish the connectivity. Proactive telecom activities like contribution of communication layer to mission and vice-versa are explained in this draft.

9.3 Context management

Supervisor and Coordinator maintain a consistent state of the network and of each participant in the network. It manages the network topology (and its predicted next states) and the tasks each actor is in charge of, as well as services that offer. On the basis of that information, the coordinator applies algorithms for choosing a bridge and/or executes workflow task reassignment when needed. The coordinator's CA manages situations when a participant is going to disconnect, by applying algorithms for choosing a bridge, and by executing workflow schema restructuring and workflow task reassignment when needed. In case of no solution, it triggers the mission layer.

Note:

Context adaptation is needed especially when mobile devices are collaborating through heterogeneous environment. Context adaptation is considered as a set of rules on the following parameters: connectivity, user availability and user location. These rules anticipate the changes that may occur on the environment parameters in order to provide an adaptable service. A decision is used to help extracting generalized rules from a variety of contextual information.

Mobile environment constitute different situations that have to be considered in the collaborative session management. The adaptation to the context is considered as a set of rules that responds to the environment constraints, imposed by the context elements. Most work deal with the adaptation rules in order to allow adapted services. Nevertheless, the huge numbers of the context parameters make the construction of all rules difficult that cover all the possible situations. The decision tree is one of these learning-based methods which lead to construct a generic set of rules that approximate different contextual situations of the environment.

We define the context as a set of constraints, associated with the environment (figure 18). These constraints should be considered so as to provide a better service or a more suited one. They can be related to the mobile terminal (memory), or to the interaction relations between users (connection mode, availability). One variation of the values of these variables (constituting a particular contextual situation) can significantly have influences on the expected behavior of the system.

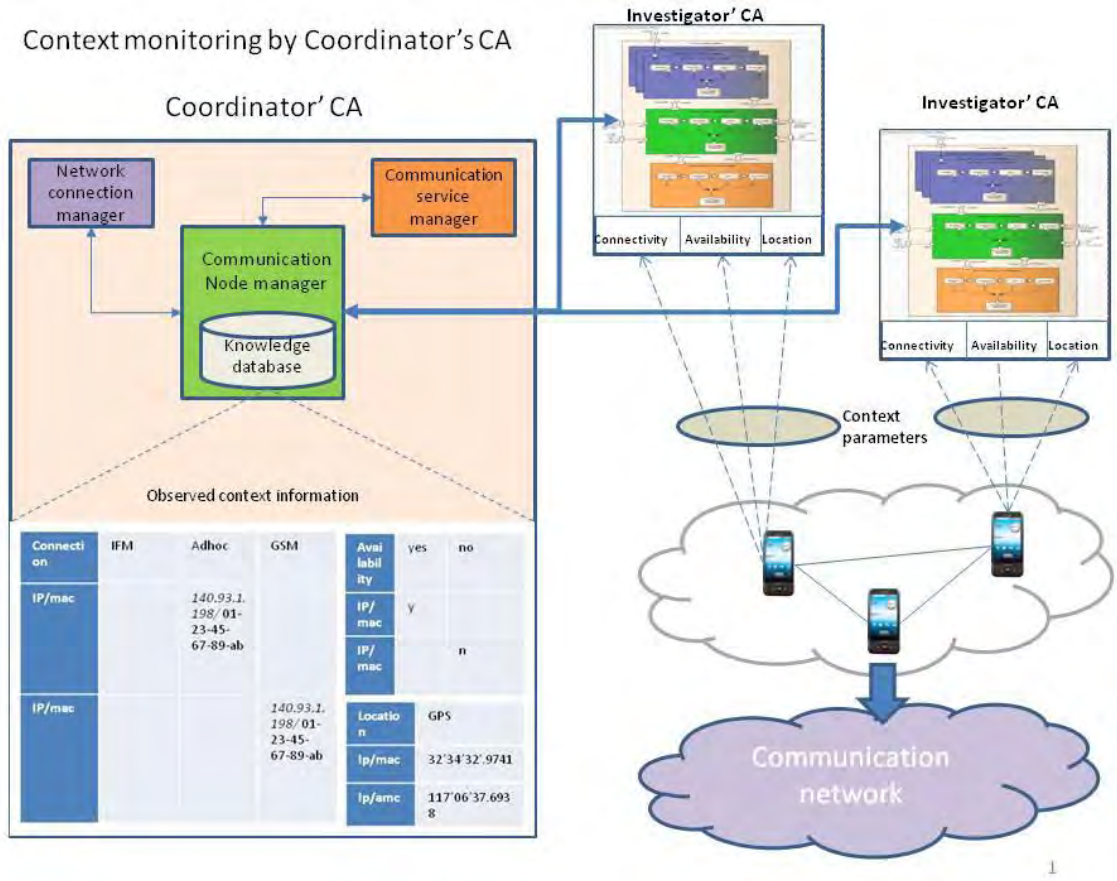


Figure 9.18: Context monitoring by Coordinator's CA

CA should support local connection management among devices that consists of monitoring and checking one-hop communications between a device and its neighbors. There are special services running on hand-held devices that implement techniques for estimating and calculating distances and relative positions (angle and direction of arrival) between a specific device and its direct neighbors. Each device has a wireless stack consisting of a wireless network interface and the hardware for calculating distances from neighbors. Communication Node Manager is responsible for sending and receiving messages to and from other devices, by abstracting over the specific routing protocols. Offered services are accessible to other devices and can be coordinated and composed cooperatively. Some of these services are applications that don't require human intervention. Others act as proxies for humans (for example, the service for instructing fireman to follow a participant is a simple GUI that alerts the user by displaying a pop-up window on his device or by emitting a signal).

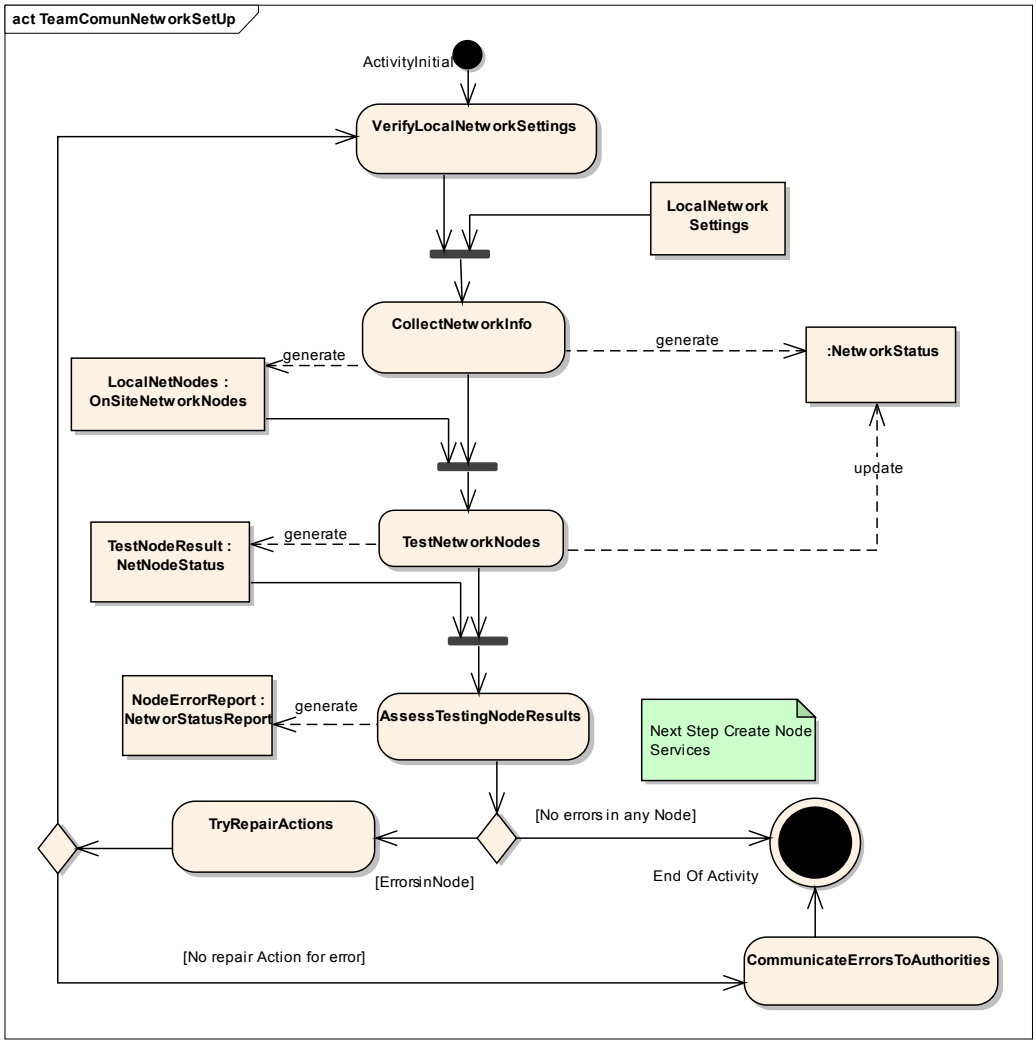


Figure 9.19: Activity diagram for team network set up.

9.4 Use case 2 - Helping to injured location through isolated node detection

Situation

The ROSACE team is situated at the intervention area. A person owning a PDA has fallen down into a deep ravine when trying to abandon the area jeopardized by fire. This person is now isolated and unconscious, then unable to use the PDA which still on. A robot or firemen wearing a PDA with ROSACE CAs, moves close to the wounded while working for fire isolation or any other activities related to the mission. They are not aware of the presence of the isolated injured since they are focused on its own goals. The CA in the robot or in the firemen's PDA, will detect the presence of injurer's PDA, and then conclude that this could belong to a person which needs help.

Goal in Context

The aim is to define the activities and actions of the CA for detecting new connections then identifying nodes and taking decisions according to mission context and goals. Identification of notifications to internal units where the CA is deployed, and the control center is also considered.

Hypothesis

- Missing person traced by telephone number (telecommunication provider, etc)

Supervisor obtains the GPS position of victim's position through the telecommunication provider. Then, this XY coordinates will be send to investigators with a prioritized notification, thus the victim is saved.

- Victim's PDA is turned on and sending probe

If there is no information from the telecommunication provider, then we have to assume that WIFI of missing person's mobile is turned on (AP mode or sending probe to find AP). If ROSACE team detects this probe, it is possible to trace the position. Thanks to triangulation or similar technologies, collaborative actions among the fireman and robot help to identify the victim. It can also be achieved by only one actor by accessing its own moving position with RSSI technology.

Note:

There are many technologies to identify the signal strength of receiving beacons. By storing the various XY coordinates of received beacons at different positions, CA's Communication Node Manager invokes this information from the knowledge database. Also, whenever it receives the new SSID, it can alarm the application to beep the device so that the fireman will be alerted that there is a new device around him.

Locally, after receiving the beep, the fireman looks for the device assuming there might be a victim around that area and same time, the signal's strength will be stored by the CA. After threshold time, CA sends the XY coordinates to device's application where it registered maximum signal strength to the device's application.

Collaborative: When there is a communication between two ROSACE' devices, this maximum signal strength's XY coordinated are exchanged to estimate the approximate position of Victim' device.

Hardware solution: By triangulation method, a single ROSACE actor can able to identify the victim's position. By connecting multiple antennas to the ROSACE device, the region where the maximum signal strength received are notified (XY position). The XY position then sends to the device's application thus paving a way for the fireman to move to that direction.
Victim cant reachable: There could be a situation where the ROSACE actor is close to the victim but he cannot trace him due to an obstacle (wall) in between them. In this case, CA

sends the XY coordinates to other actor.

Global: When the Super visor receives the XY coordinates from a single actor, it checks the database whether it already received this position from the other actor. If it so, it concludes that the victim could be around that position and send a coordinating message to the two actors to move to the position if actors are not in any prior mission. Monitoring can be done at two levels

- At Mac layer:
  - Passive sniffing (get RCPI – RSSI)
  - From two or more location, compute an azimuth (triangulation)
- At PHY layer:
  - Get antenna gain from signal
  - Apply for beamforming techniques

Technologies:

- Beam forming: Techniques using antenna arrays
- If fireman's CA gets the probe of victim's device, then there is a necessity to gather data from the robot for processing (to investigate whether robot has all information needed to track)

9.4.1 Receiving signals (beacon) from Victim's mobile

Once the Robot or Fireman's device receives signals from the victim's device, the CA will analyze whether the signals have enough information to identify its position. The first solution is to move around to estimate the position of the victim and it's purely depending on mission. If there is no sufficient information, it will pass this information to his neighbour to compute the position. Figure 19 and 20 explain the necessary steps to trace the victim. The various technologies will be explained briefly in the Appendix.

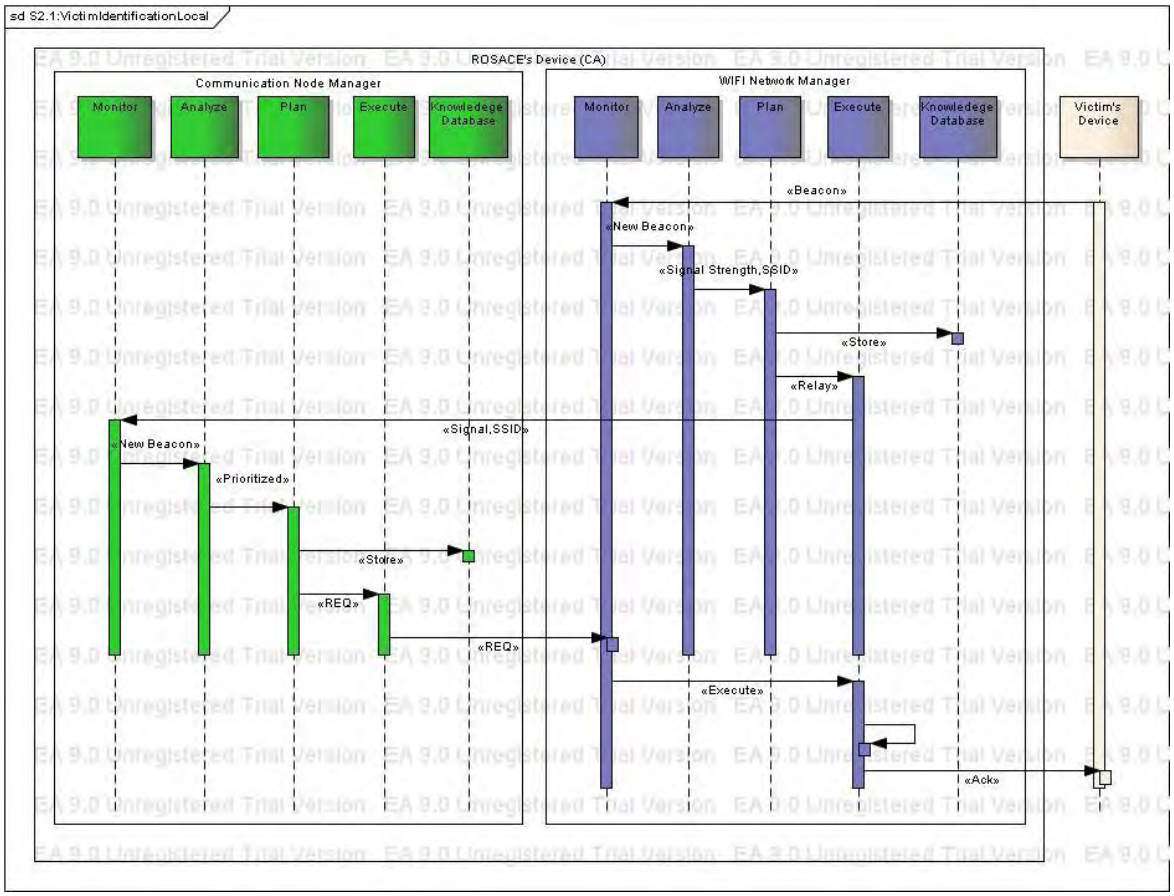


Figure 9.22: Sequence diagram for tracing beacon from the victim's device

CA's internal state

- Network Connection Manager : ready to analyse incoming information
- CommunicationNodeManager : controlling the connections between the neighbors

9.4.2 Modeling fireman's activity

The different use cases allow us to identify the various activities of fireman and robot. As the goal of ROSACE is to help the victim, there exist four states. The first state is to trace the position of victim or ROSACE members in case of danger. Once traced, depends on the situation, fireman moves to assist the injured person. This state sometimes involves no interference of coordinator but at times, the coordinator send request to fireman to locate the injured. The transition from discovery to giving help is triggered by actions mentioned in the figure 21. Also, we need to consider the victim that needs help which will be detailed briefly in the next section. Once the alarm is activated from the ROSACE team, the neighbor can recognize this situation and appropriate actions are taken. This state diagram helps the designer to understand the interactions between the coordinator and investigator in an abstract way. Each state has its own activity diagram and revealing each activity diagram results in sequence diagram. Figure 21 and 22 explains the state behavior diagram and activity diagram (discovery) of fireman respectively and remaining activity diagram will be explained later.

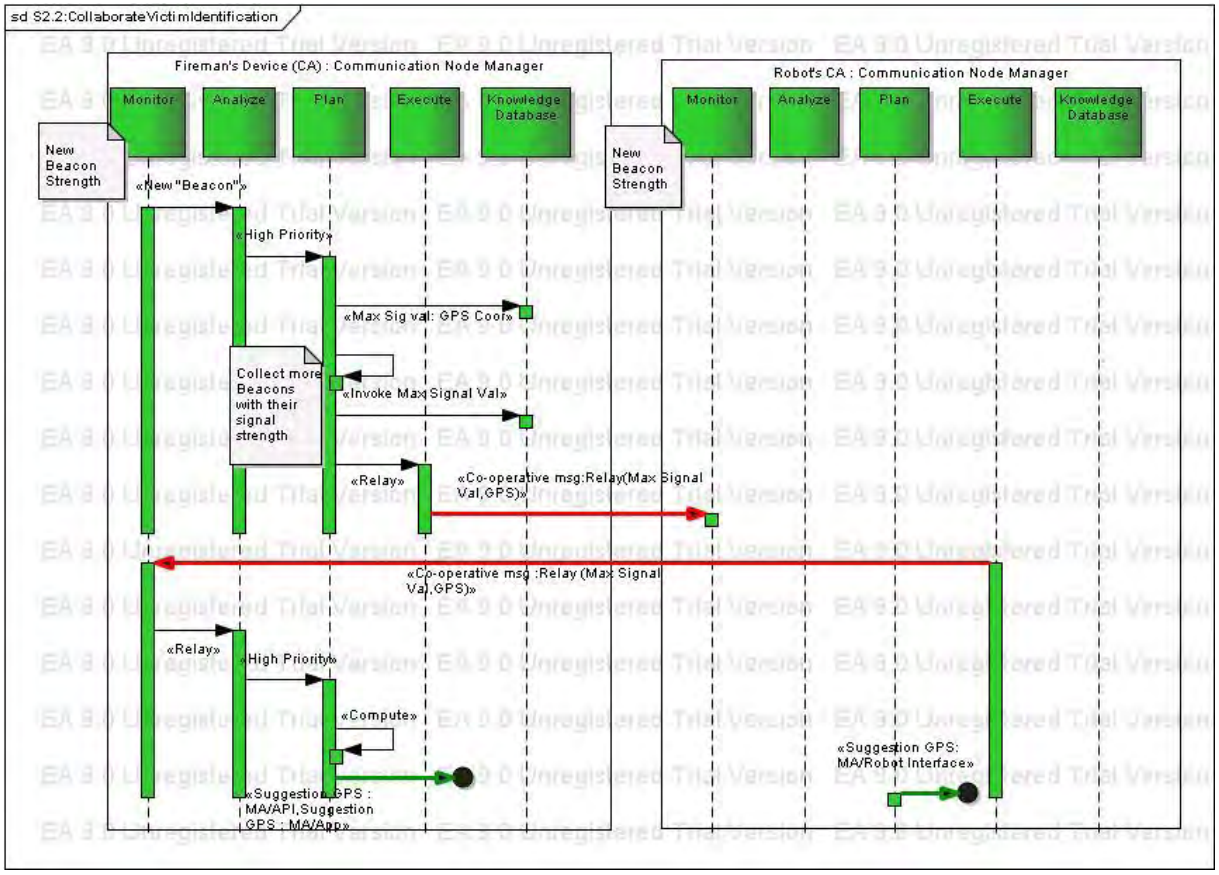


Figure 9.23: Sequence diagram for collaborative actions to find the victim's position

9.5 Use case 3 – Team member's self-protection through isolated node

Situation

The ROSACE team is situated at the intervention area. Firemen and robots are also in the area for fire supervision an injured location. A fireman wearing ROSACE PDA has fallen down into a deep ravine during fire fighting. This person is now isolated and unconscious, then unable to use the PDA which still on. A robot with ROSACE CAs moves close to the wounded working for fire isolation or any other activities related to the mission. They are not aware of the presence of the isolated injured since they are focuses is in other goals. The CA in the robot will detect the presence of injurer's PDA, and then conclude that its owner is in serious danger.

Goal in Context

The aim is to define the activities and actions of the CA for detecting new connections then identifying nodes and taking decisions according to mission context and goals. Identification of notifications to internal units where the CA is deployed, and with other CAs and the control center is also considered.

Hypothesis

- Video and audio virtualization

This use case provides an immense range of possibilities to discover the available technologies to find a solution. Imagine a situation, where two actors are communicating with each other through voice and video to achieve a common task for example, searching a victim. If one of the actor is in danger, through voice call, he can notify his neighbor or to its coordinator. But there could be a situation when the actor is unconscious and CA of his device could able to identify the situation. To achieve this, typical sensors embedded in fireman's clothes sense the well- being of this actor. These sensors could be accelerometer (detecting a fall), blood pressure, heart beat, steps count etc. Once the information received by the sensors, CA takes an appropriate decision. Local decision is to broadcast an alarm with current GPS position.

In collaborated mode, by treating the received messages from the neighbor, CA will take an action. Once the alarm is received from the victim's device, the neighbor who is connected to the victim send "HELLO" messages five times to make it sure that the victim is danger. If there is a reply from the victim's device, the neighbor notifies this situation to the coordinator and tries to locate the victim. The CA's plan module should possess algorithm to find a solution if there is no response from the victim's device.

If the victim is connected to the coordinator and there is no proper response from the victim's device, CA of the coordinator should analyze the different states of the situation like communication failure, actors' health, etc. which is explained in the previous use case. The unresponsiveness of investigator to coordinator's message leads to analyze the different state of actor from the coordinator's point of view.

9.5.1 Broadcasting Alert from ROSACE Victim's mobile

Once the Robot or Fireman's device receives signals from the sensors, the CA will analyze the conditions of the actor from predefined threshold values by consulting the knowledge database. Then according to the appropriate plan, it broadcast alert message and also set the device into safety mode (figure 26).

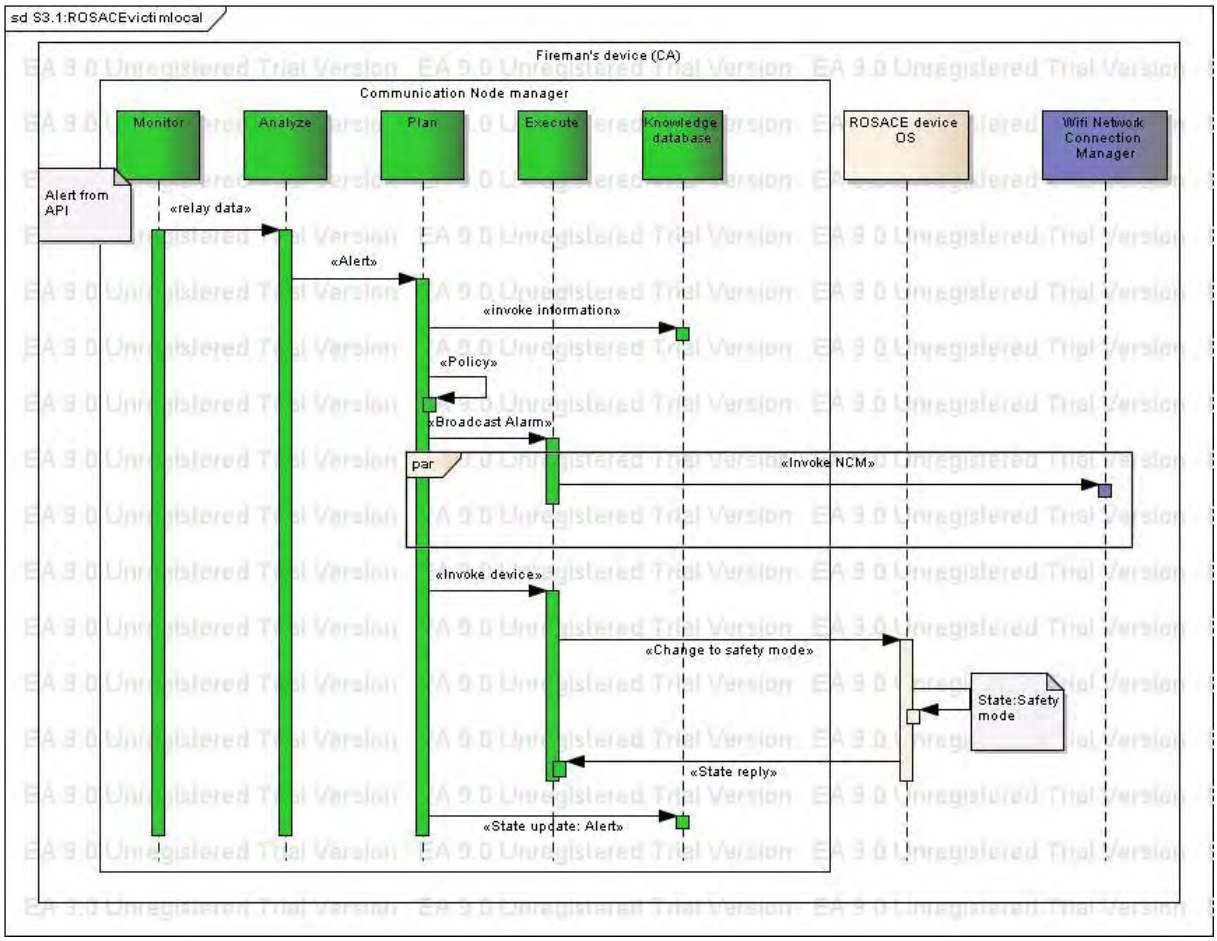


Figure 9.27 : Sequence diagram to broadcast alert message from ROSACE victim's mobile

9.5.2 Collaborative action to trace ROSACE Victim's mobile

Once the alert message broadcasted from one of team victim's device, the investigators or coordinators who receive this information should act according to the mission. One way is to check the SSID and position that is included in the broadcast message. Once authenticated, the neighbor assures that fireman is in real danger and asks the device for its GPS position and then notify to the coordinator (figure 27).

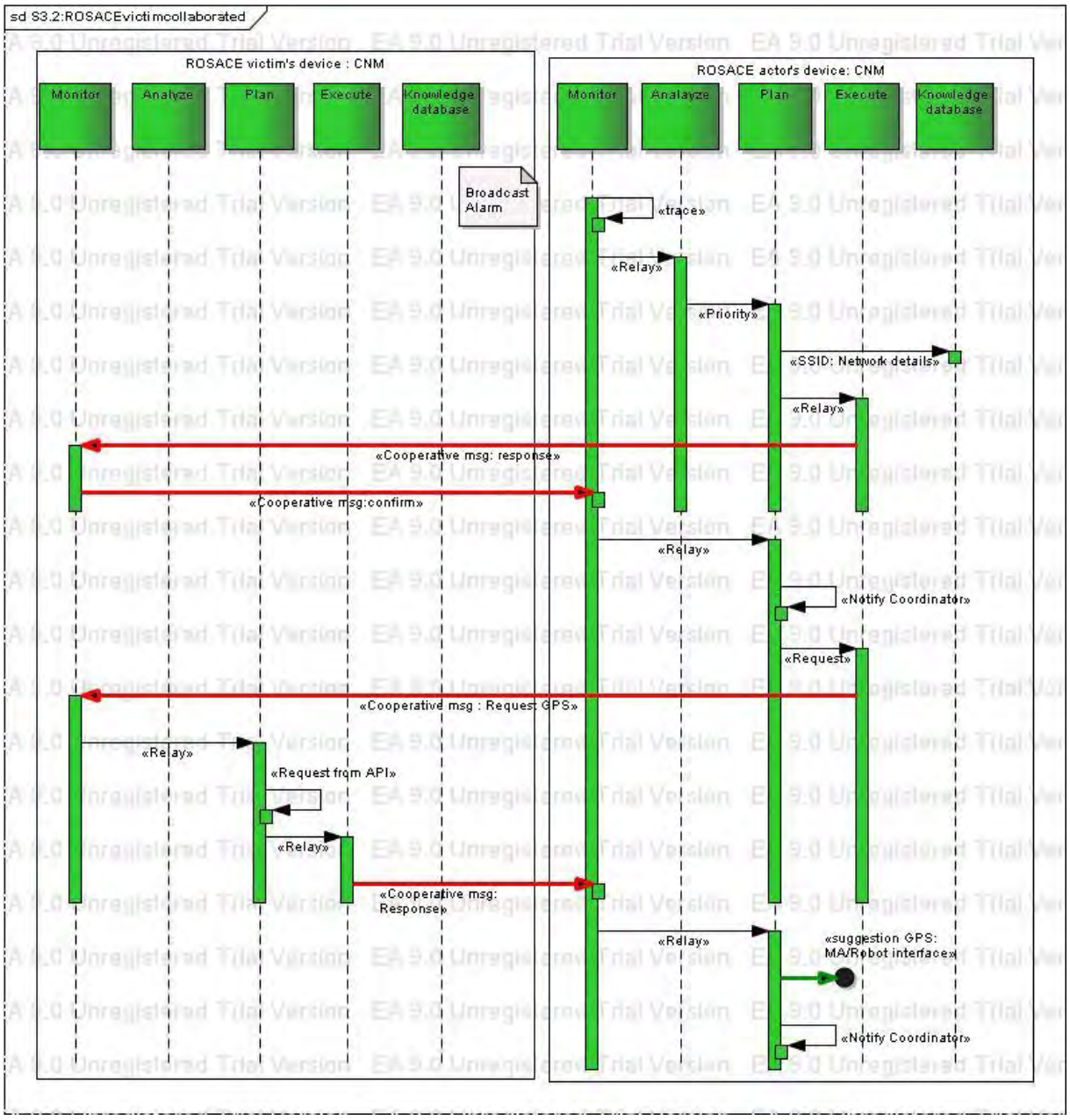


Figure 9.28: Sequence diagram for collaborative action to trace ROSACE victim's mobile

9.5.3 Modeling fireman's activity

We have included fireman's activity in this use case because; the behavior of fireman is also to help its team members. That is why there are different states in the diagram that clearly indicates the action of fireman. States like "giving help" ,"need help" and "waiting for coordinator" are explained in the following diagram. The messages between the states are sometimes refer the trigger, suggestion, coordination and cooperation messages. We distinguish these messages by a legend (using colors) in the sequence diagram.

Note:

When talking about the local plan, we have to consider the functional and non functional requirements. Services and functions a fireman can do are the functional requirements whereas QoS, feasibility are non functional requirements.

9.6 Use case - Improving team routing management by detecting lose of connection

Situation

The ROSACE team is situated at the intervention area. Firemen and robots are also in the area for fire supervision and also to locate the injured victim. A WiFi hotspot is located in a robot. A WiFi local network has been created around the mobile Hot spot. The CA network manager detects lose of connection and notifies the decision components of the robot indicating possible location/trajectory where connectivity will be OK.

Goal in Context

The aim is to define the activities and actions of the CA for detecting connection lose, then finding out possible solutions and communicating these solutions to decision units where the CA is deployed. Possible alternatives for finding solutions could be considered: a) Nodes could communicate among them to adapt their location to find the connectivity. b) The network manager will be in charge of finding out this solution, c) mixed alternatives where first the node try something but doesn't succeed then reports to network manger which in turn try something different and then reports to the CC if connectivity cannot be restored.

Hypothesis

- Signal history and its GPS position

ROSACE device possess some special application for its GPS coordinates. Thanks to this special application, CA invokes the XY coordinates and stores in its database. Whenever actor's device is connected to its peer or to its coordinator, communication node manager stores the XY position and its signal strength. Signal strength will be calculated by network connection manager. By storing the history of GPS position, CA can suggest the mission to take an action in case if it detects the deterioration in the signal strength. CA's plan should be strong enough to estimate locally the trajectory where the connection will be fine.

- Notify peer

CA can also notify its peer when there is connection deterioration before the loss or shift to adhoc mode to connect with its peer to calculate the good position.

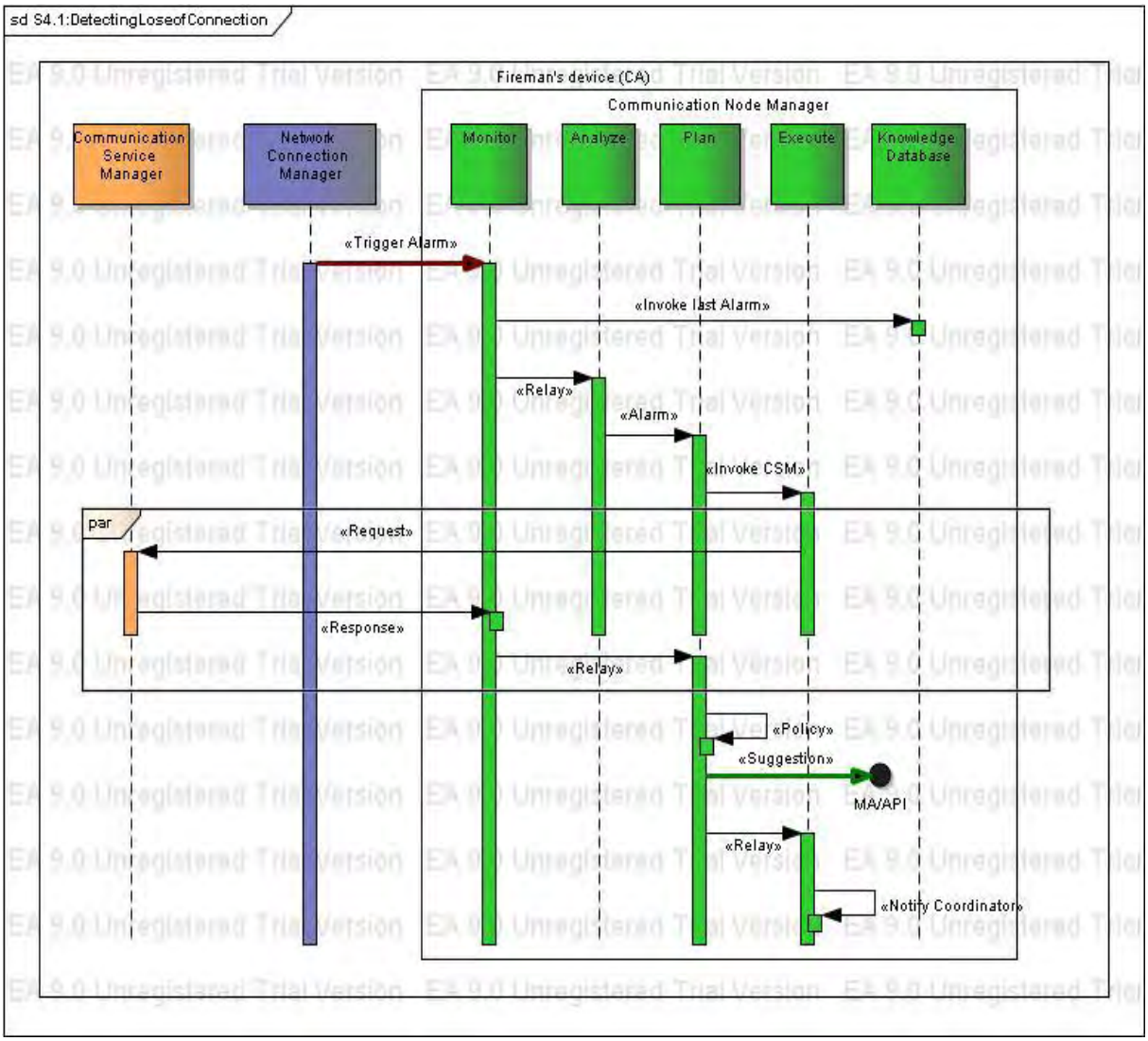
CA's internal state

Network connection manager detects weakness of connectivity

Communication node detects possible degradation of node communication services
Coordinator controlling the local network connections detects degradation of connectivity / connectivity loss

Note:

This draft details about the interaction between the different entities like fireman device, robots and coordinator. The messages between the internal components are explained by the sequence diagram. This will be briefly supported by communication protocol itself to achieve this task. Suitable examples are shown for the reader to follow.



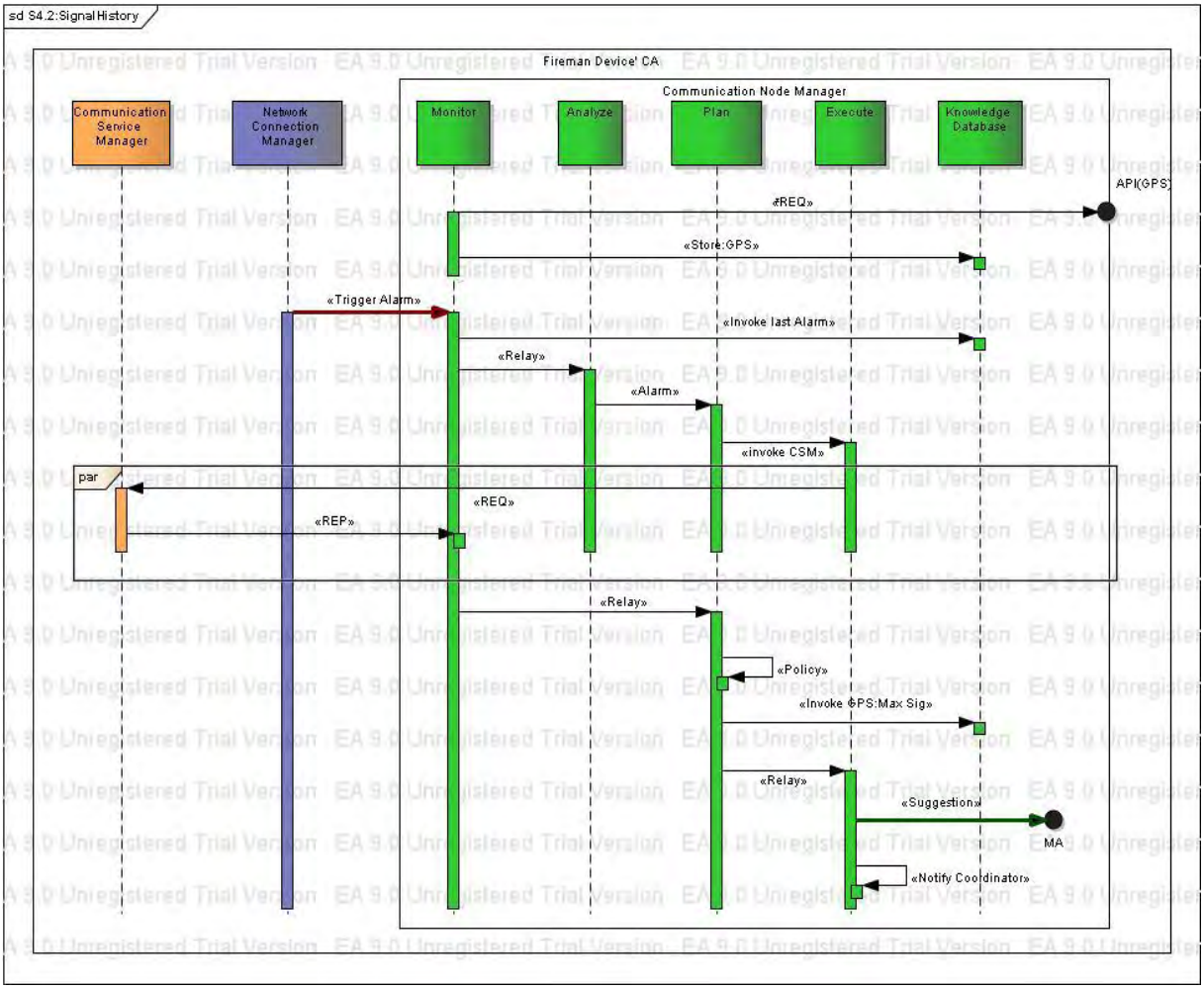
9.31 Sequence diagram for detecting connection loss

External view

The figure 32 shows ROSACE team's observable behaviour. The first case will be based on a hierarchical model where the Coordinator has the responsibility of finding out a solution. Other cases will be considered as variations of this case.

Detecting connection loss:

If there is deterioration in signal, there will be a trigger from network connection manager to communication node manager. Once it receives the trigger, node manager consults the database regarding the solution (thanks to the message stored last time when it was triggered) to find a solution. To confirm the loss, node manager consults with service manager if there is a loss in QoS. After the message communication, node manager suggests mission agent about the coordinates where it can have the maximum signal and then it notifies to coordinator.



9.32 Sequence diagram for detecting connection loss through Signal History

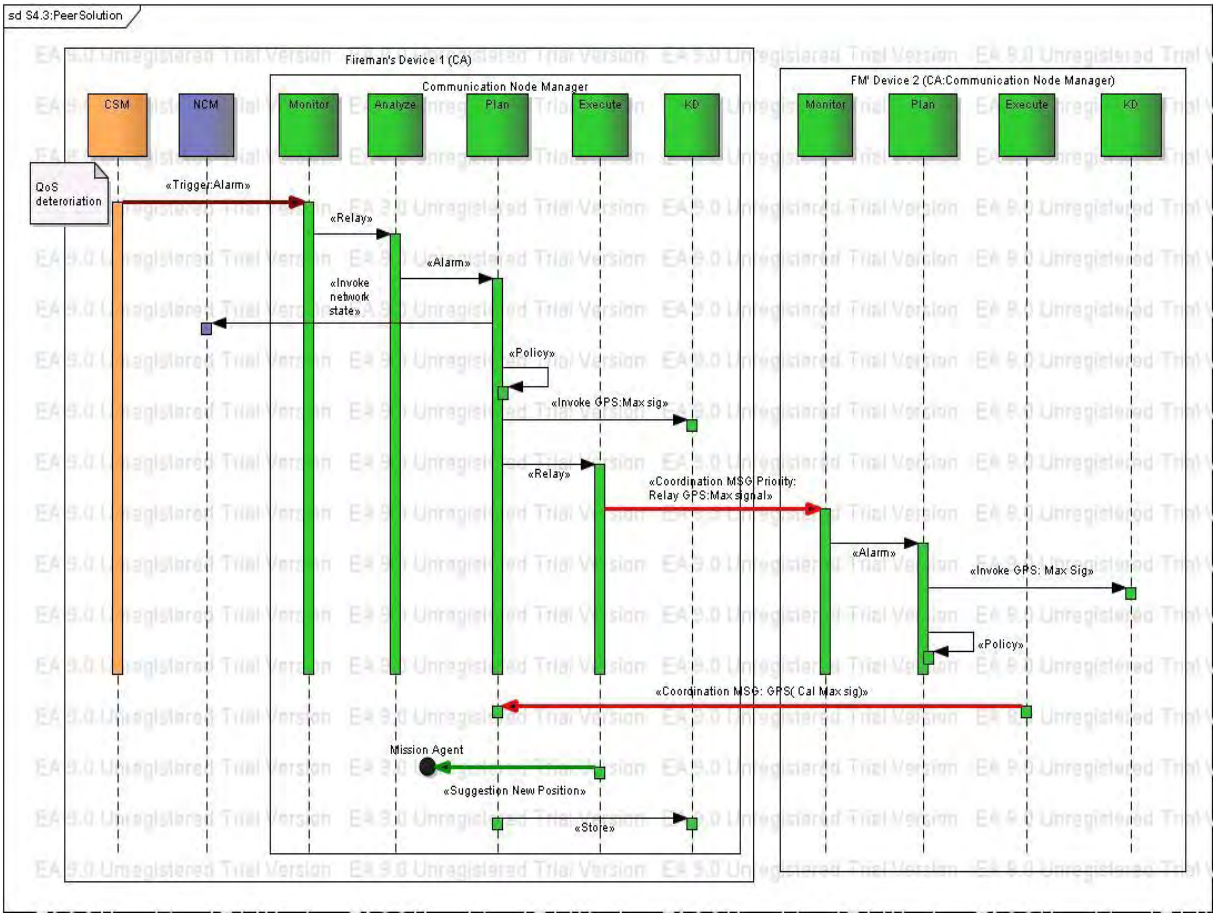
Note: Local solution to analyse the database:

Through application programming interface, the node is capable to identify its GPS position. The idea is to store this GPS position and also the signal strength at database.

When there is deterioration, it communicates both with connection and service manager about the loss to understand whether the service is a prioritized task or not. Once there is a loss in QoS, it suggests to Mission Agent about the GPS position where the device will have a chance to have the maximum signal.

Peer Solution

Once the GPS position identified by node manager, it could also send to its peer about this position. By that, a distributed solution is possible, because the neighbour compare this result in its database whether this received GPS position has the maximum signal. If not, it will send a new GPS position where the losing one will have the connection. In this coordinated message, trigger will carry the GPS position with its signal value.

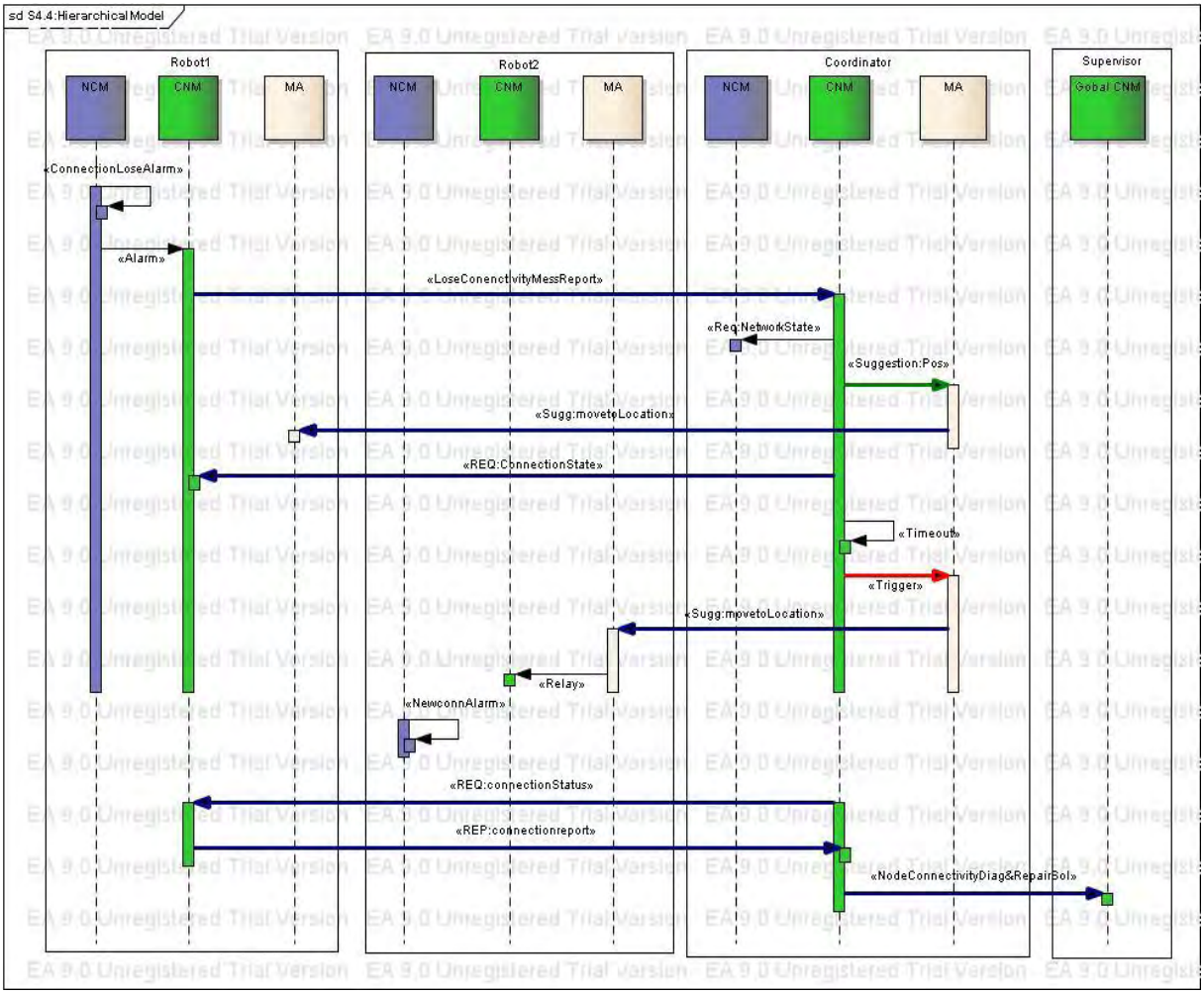


9.33 Sequence diagram for detecting connection loss through Signal History by Peer Solution

Hierarchical Solution

The main assumption in our scenario is that the nodes are connected to neighbours through infra structure mode. It means, the signal deterioration can also be traced by the network manager at coordinator CA. The node manager of investigator sends a message

about the loss report and coordinator takes an appropriate action. It will send a cooperated message about the new GPS position where the investigator will have the signal. If there is no confirmation about a certain time, coordinator will understand that the investigator might loss the connection and find a best neighbour node and send a message to follow the lost one. It will be done by analysing whether the helping robot is in prioritized task or not. If the neighbour robot is not in important action, it will send the robot a message to move a GPS position. The message contains the GPS position and a request to change the mode of communication.



9.34 Sequence diagram for detecting connection loss through Signal History by Hierarchical Model

Internal view

Figure 33 shows internal activities of a coordinator needed to achieve the functionality of this use case. More decentralized approach could be considered based on each node having the capability to calculate best location and asking close node to move to this location.

Goal in Context

To define a decentralized model of connectivity recovery where each node is capable of detecting connectivity lose with a close neighbour, it estimates if it is possible for him to do something for maintaining connectivity. If a solution is found it is redirected to its decision unit in order to be executed, otherwise notifications are sent to close nodes for contributing to find a solution.

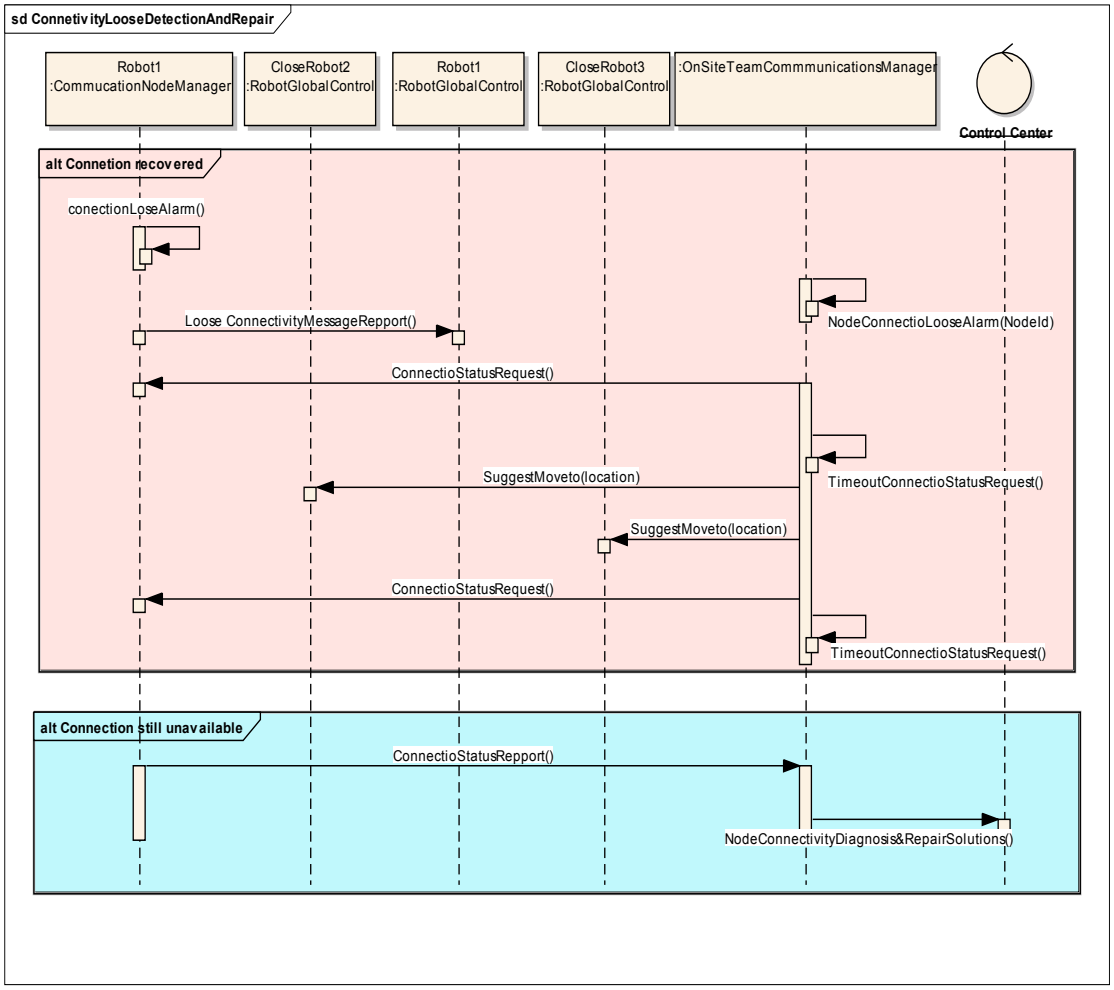


Figure 9.35: Sequence diagram for detecting loss and mission suggestion

Note:

Figure 9.35 shows the functional sequence diagram of coordinator's action once it receives an alarm from the investigator regarding the connection deterioration. Here, we show the different types of global messages and functional calls regarding the control center, the coordinator and also the robots 1 and 2, whereas robot 1 is having the problem with the connection and robot 2 is helping robot 1 to have the connection back. It is very important to note that, all the decisions are notified to the control center. The solutions taken at the coordinator sometimes might not the help the situation. In that case, the supervisor takes a global action to resolve the problem. It could be change of communication mode, movement of coordinator, or changing of roles. We are working on those issues on coming chapter.

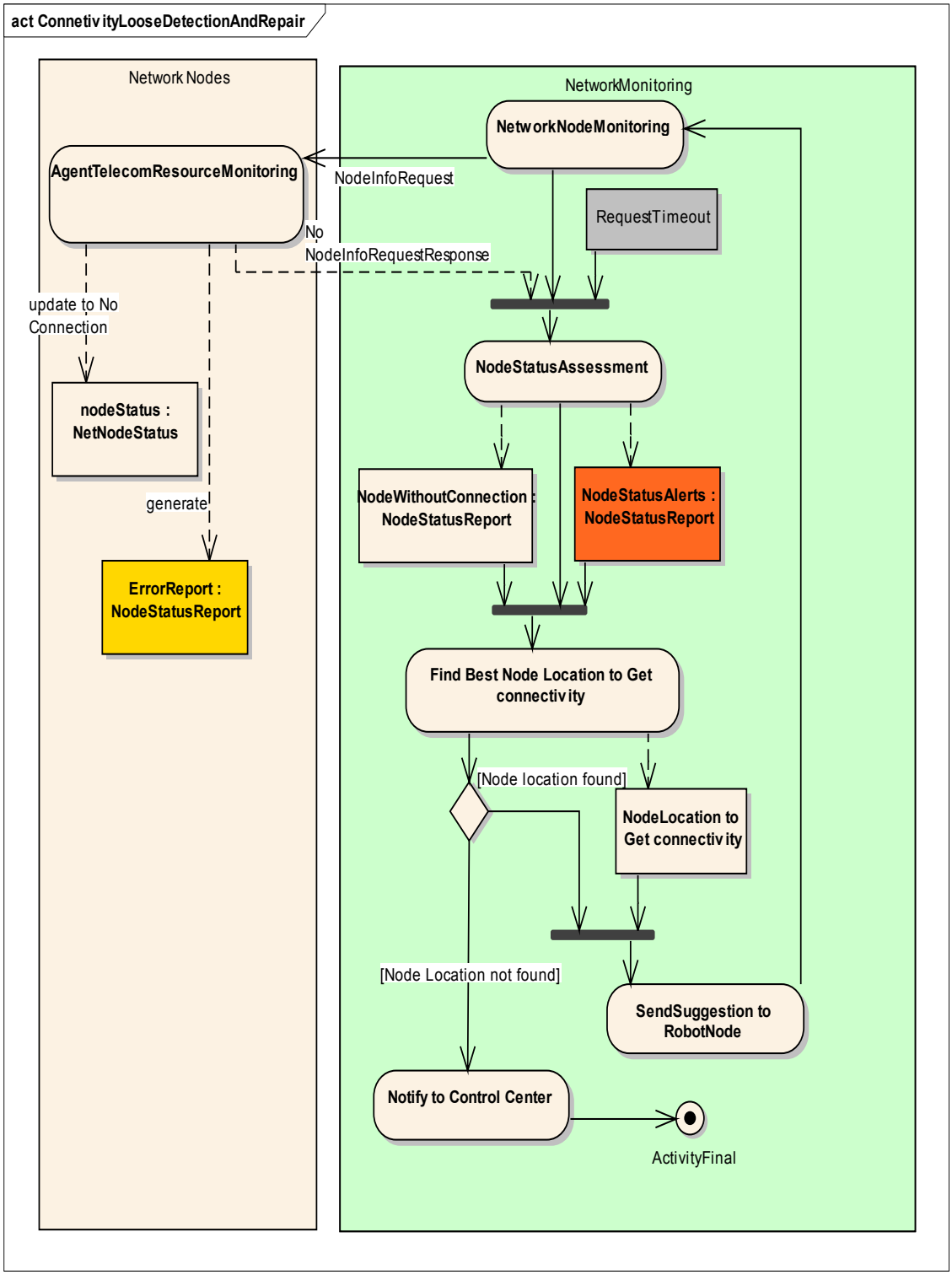


Figure 33: Activity diagram of coordinator for finding the best node

3.7 Use case - Adaptive Service Management in Critical Situations

Situation

The ROSACE team is situated at the intervention area. Firemen and robots are also in the area for fire supervision an injured location. A WiFi hotspot is located in a robot and a local network has been created around the mobile Hot spot. The robot intervention receives a notification from the control center for helping an injured at a location X. Following an internal decision one member of the team leaves the group to assist the injured. In its way to injured location the robot lose connectivity. The CA detects lose of connection and notifies the decision components of the robot indicating possible location/trajectory where connectivity will be OK. The robot ignores this notification since its priority is to assist the injured. The CA then try to know the task and the priority of the robot's goal in order to make the necessary for recovering the connection and providing the communication services needed for achieving its goal.

Hypothesis

- Increase the signal strength
- Task refining

Goal in Context

The aim is to define the activities and actions of the CA for detecting connection loss, then finding out connectivity recovery using knowledge about the Robot's goal. The QoS of the connection should be adapted to robot needs. When robot's goal has high priority the CA service manager should be aware of this and possibly create or ask for urgent creation of new connectivity resources to guarantee the telecommunication services with the QoS necessary to achieve this goal

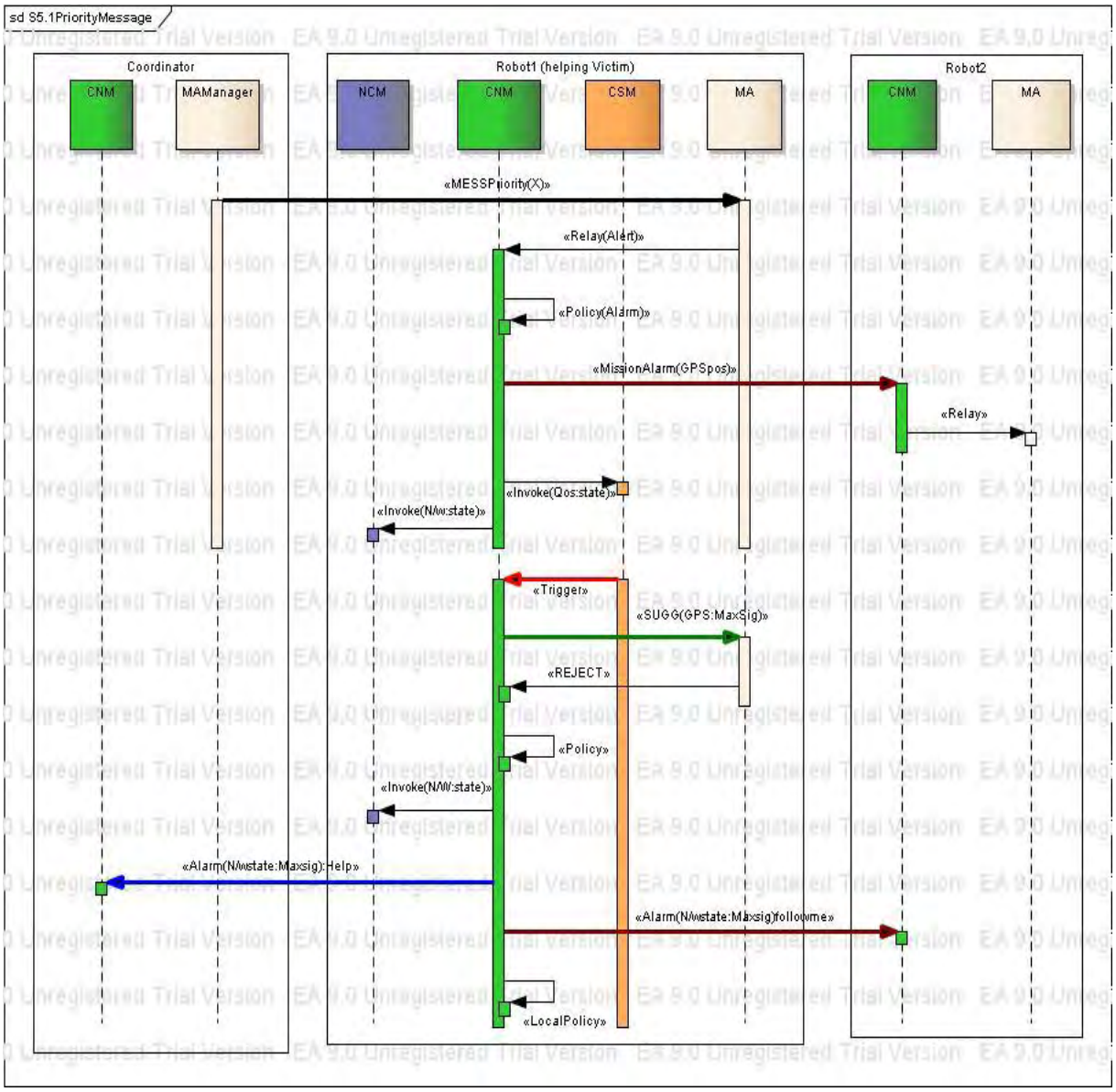


Figure 9.34: Sequence diagram showing the communication between MA to highlight the priority message

MA to highlight the priority message

If there is a task allocated to a node either by coordinator or supervisor, it will be notified through to the MA. And if the task is a prioritized one, the MA sends a message to node manager. Thanks to this message, node manager will alarm this message to its neighbour. At the same time, it invokes the QoS properties from service manager if there is any ongoing service between the node and its neighbour. Depending upon the situation, if the service manager sends a trigger to node manager, it suggests to MA about the GPS position. But the MA rejects this suggestion as it is assigned to a prioritized task. Once it get the rejection from the MA, it alarms the neighbour and also the coordinator.

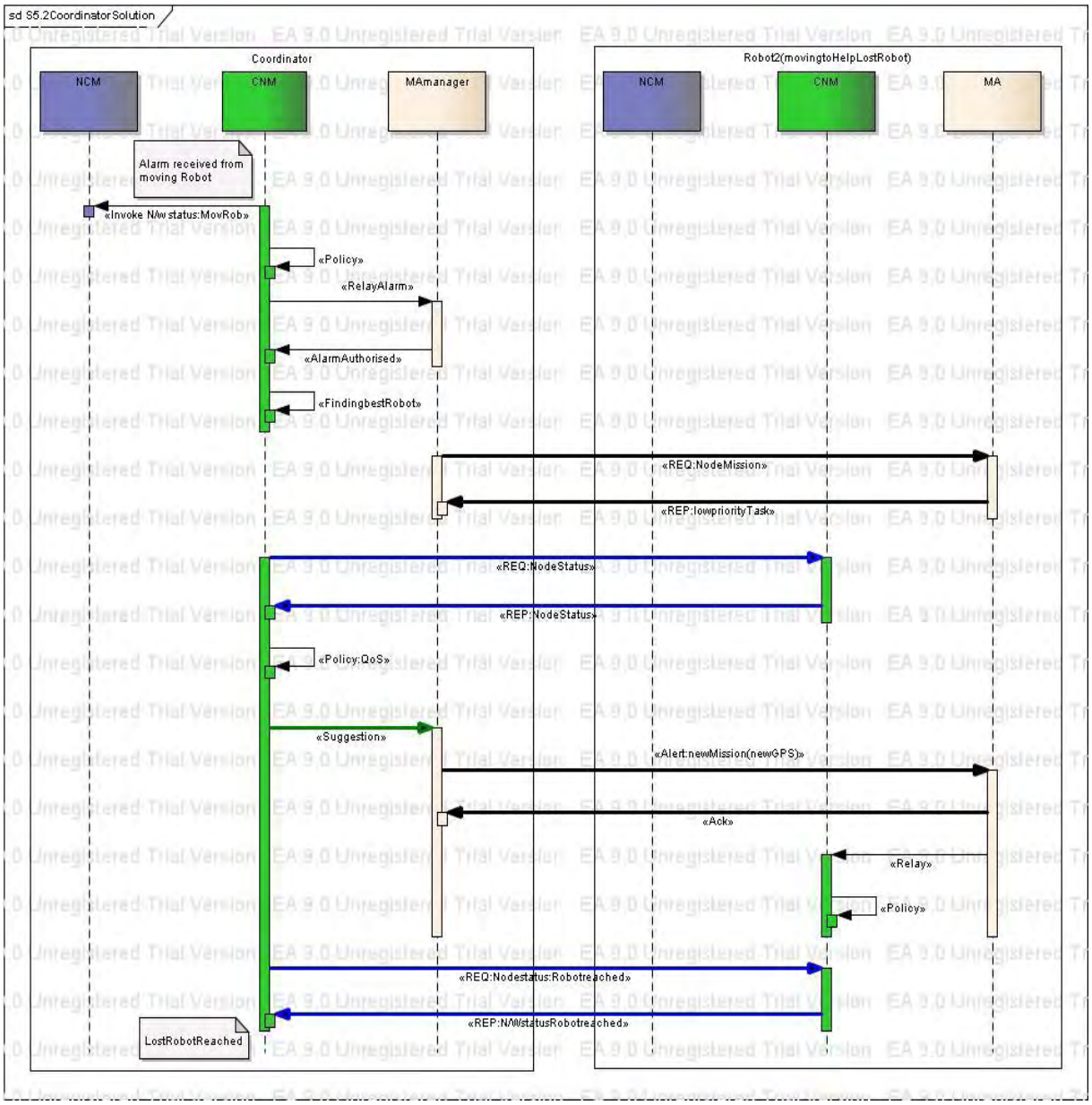


Figure 9.35: Sequence diagram showing the coordinator solution

CA's internal state

Communication node manager from robot detects weakness of connectivity, Communication service manager detects possible degradation of node communication services

Coordinator controlling the local network connections detects degradation of connectivity / connectivity loss

External view

Figure 34 shows team's observable behaviour. The first case will be based on a role oriented model where the coordinator the responsibility of finding out a solution. Other cases will be considered as variations of this case. CA detects lose of connectivity of one node. It tries to know the role and goals of the robot, in order to find out the communication services better adapted to achieve these goals.

Coordinator solution

The network manager which resides inside the CA of coordinator monitors the communication. If a fireman sees a victim, he will automatically move to help the victim and he doesn't mind to be in connection, because saving a victim is his prioritized task. In this case, the coordinator traces deterioration and sends an alarm to MA to move to suggested position. But this message will not affect the fireman and still the communication loses its connection. In this case, the coordination will assume that this fireman will be saving a life and find a neighbour robot or fireman to save the communication between the lost one.

Internal view

In fig 34, coordinator's CA identifies a connection loss in one node. As a result the control monitoring activity is notified, including a node status alert report which contains information about the node. This information is received in the context of the control monitoring activity which should assess the status alert report, and then decide how to proceed for recovering the connection and the services supported by this connection. A possible recovering plan is in the following figure 35.

Extensions

The diagram gives a specific plan for restoring connection. Other alternative plans are possible. These plan as well as internal activities should be defined in detail. More decentralized approach could be considered based on each node having the capability to calculate best location and asking close node to move to this location.

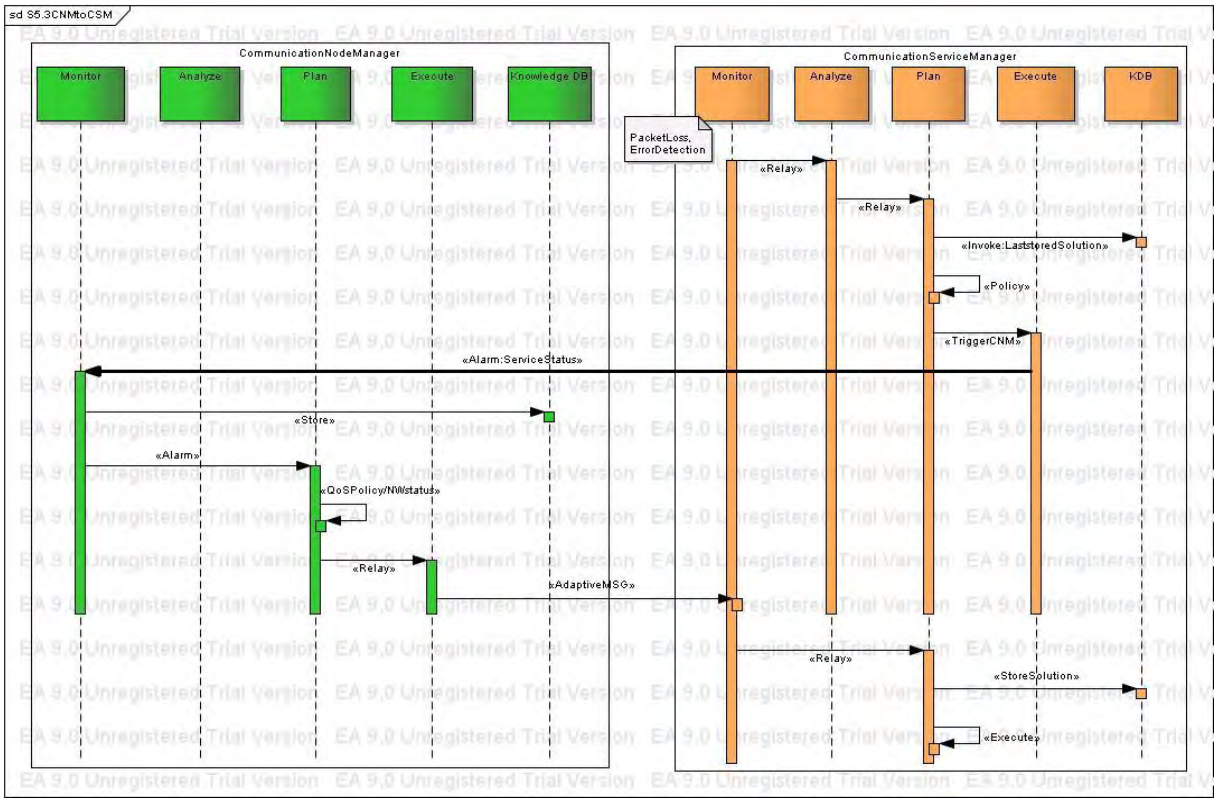


Figure 9.35: Sequence diagram between CSM and CNM

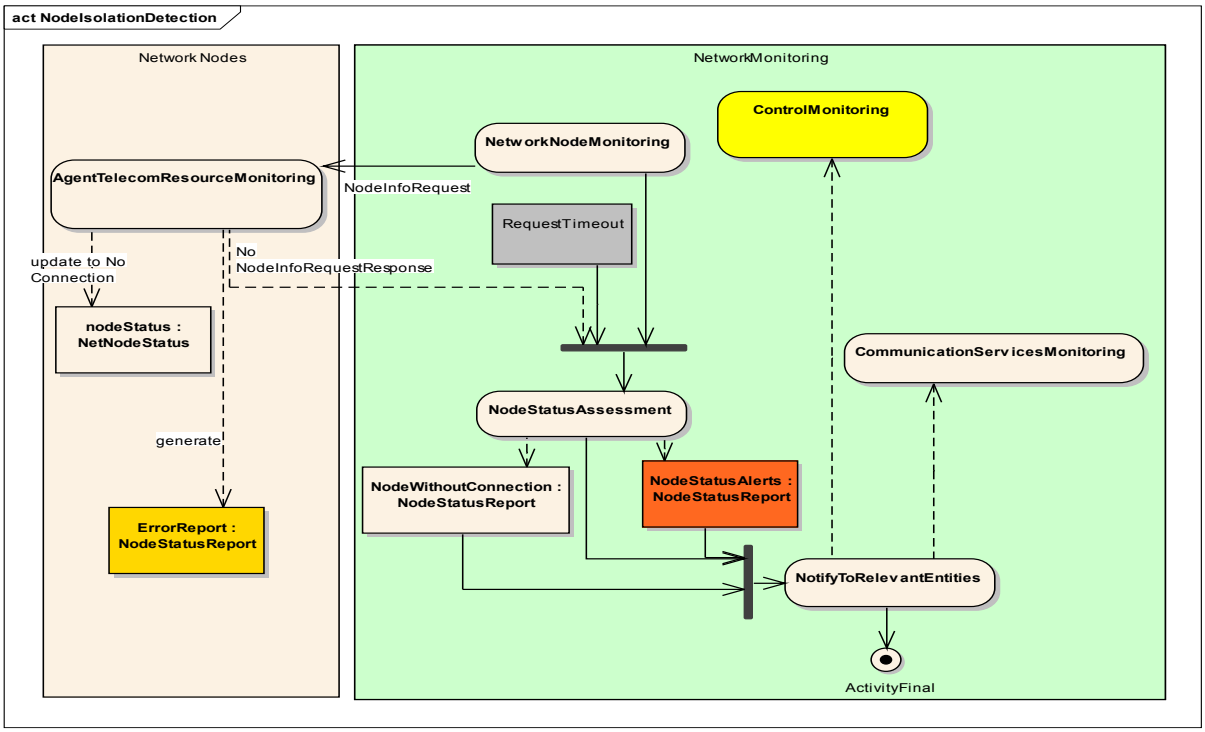


Figure 9.36: Activity diagram of coordinator for analysing the team's connection loss

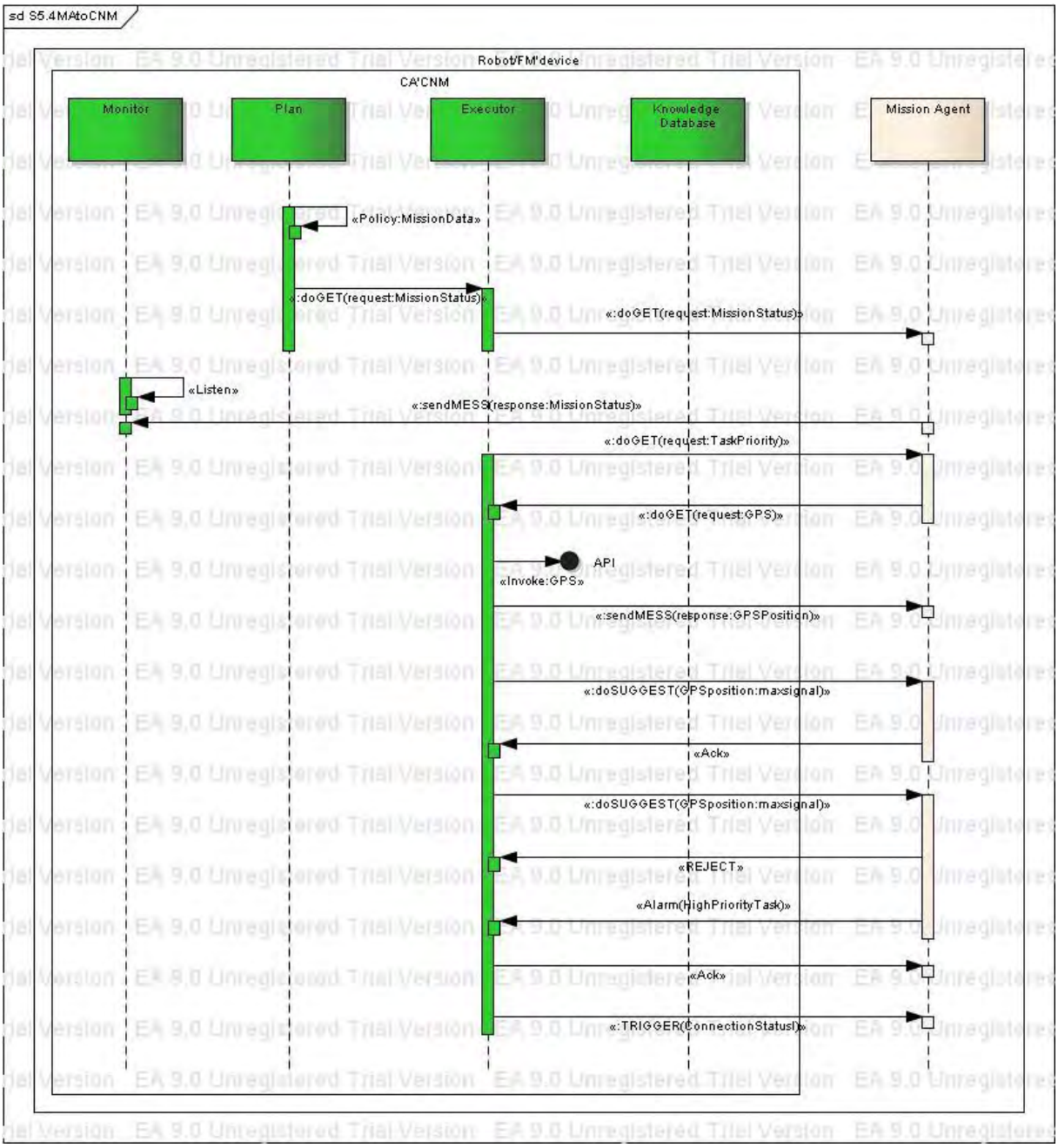


Figure 9.37: Sequence diagram between CNM and MA

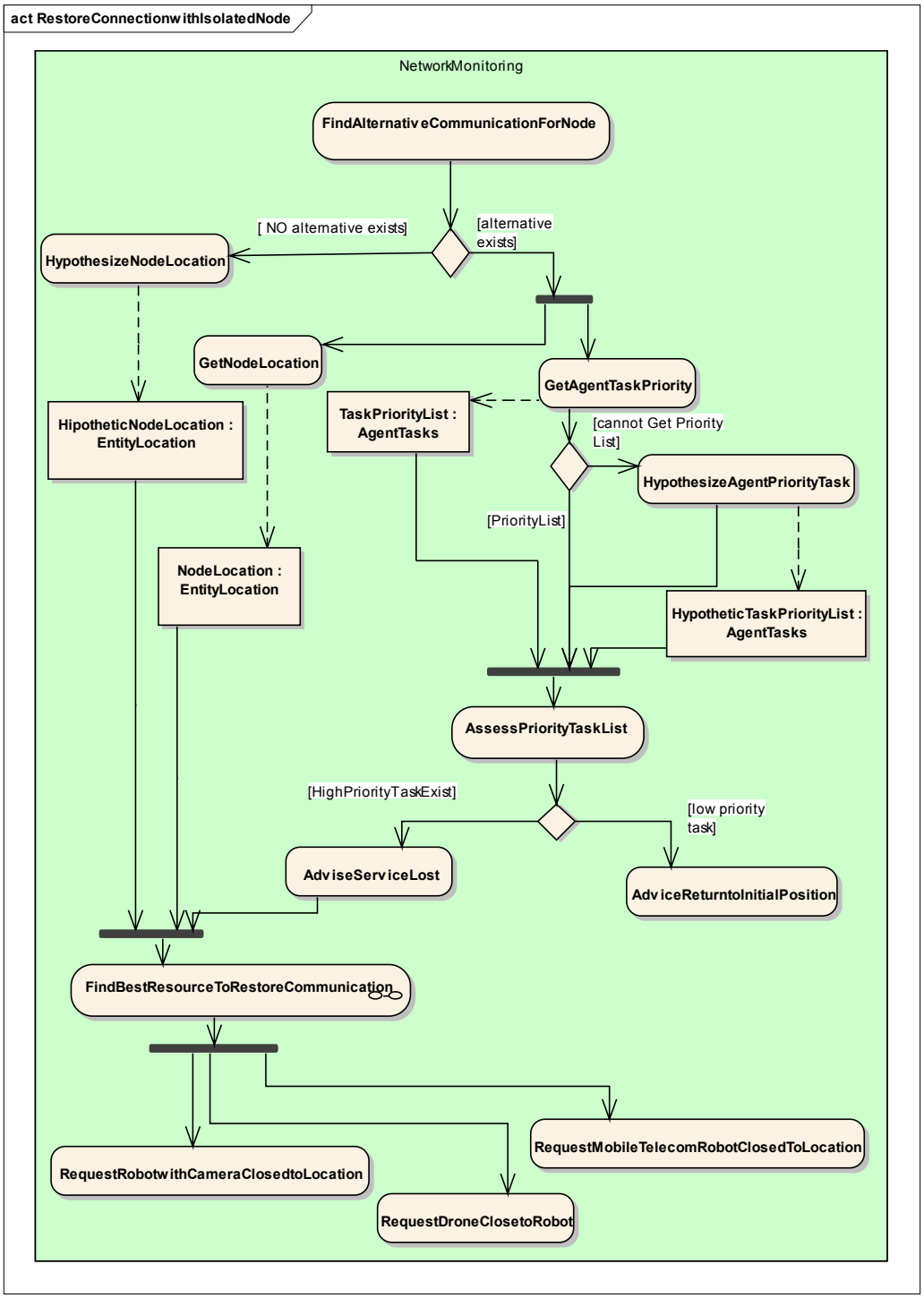


Figure 35: Activity diagram for possible recovery plan

10 Ontology based reconfiguration

Adaptation at control center is handled by semantic modeling whereas at local entities, it is managed by a software module called communication agent (CA). Modeling follows the well-known SWRL instructions which establish the degree of importance of each communication link or component. Providing generic and scalable solutions for automated self-configuration is driven by rule-based reconfiguration policies through ontology. To perform dynamically in changing environment, a trigger mechanism should force this model to take an adaptive action in order to accomplish a certain task, for example, the group chosen in the beginning of a mission need not be the same one during the whole mission. Local entity adaptive mechanisms are handled by CA that manages internal service APIs to configure, set up, and monitors communication services and manages the internal resources to satisfy telecom service requirements. Our work presents ontology based reconfiguration at control center and autonomic computing capabilities of CA followed by some illustrations like changes in the activity where node can arrive/leave and resource constraints, e.g., energy level, connection loss.

CA is responsible for local adaptation and trigger the control center if there is no solution. Once alarmed at control center, semantic model, thanks to SWRL rules reconfigure dynamically its topology to ensure minimum guarantees to maintain the communication. Thus, in order to deal with these very distinct capabilities, we use ontology based approach to solve the communication challenges at control center. And at local entity, the agent can: manage communication resources to ensure uninterrupted connectivity between mission nodes; provide the best-adapted solution according to communication goals and provide resources depending on availability.

Motivated by the above discussion, we present a novel ontology-based support for reconfigurable adaptive group communication architecture at control center. This approach improves the distributed decision making that readily acts in a time-constraint situation. At local entities, we have designed a communication agent with autonomic computing properties to take a decision by itself depending on the situation or trigger the control center if there is no solution.

The adaptive techniques at two different states are explained in this section. We will elaborate the techniques used at the control center followed by the CA at local entities. At control center, the main feature is the clear partitioning of adaptive functionalities into different levels, in which each level only takes care of the functions that are most suitable to be concerned by it. Each level encapsulates issues into a specific model, thus abstracting complexity to a higher level. The modules at the higher level are abstract system representations that tend to resemble human activities, while lower ones are much closer to real implementations of abstractions supporting these activities. Relevant levels are identified

and adaptation at the highest levels should be governed by changes in development of activity requirements. Adaptation at the lowest levels should be driven by execution context constraint changes.

In order to clearly separate different concerns in our approach, a multi-level architecture is proposed. Each level encapsulates issues into a specific model, thus abstracting complexity to a higher level. The modules at the higher level are abstract system representations, that tend to resemble human activities, while lower ones are much closer to real implementations of abstractions supporting these activities.

Relevant levels are identified and adaptation at the highest levels should be governed by changes in development of activity requirements. Adaptation at the lowest levels should be driven by execution context constraint changes. The levels retained are depicted in Figure 2 and detailed in the following paragraphs.

Here, we have chosen an ontology-based model because it constitutes a stan- dard knowledge representation system, allowing reasoning and inference. More-over, ontologies facilitate knowledge reuse and sharing through formal and real world semantics. Therefore, ontologies are high-level representations of business concepts and relations. These representations are close to developers' minds and therefore well suited to depict application level models. We have chosen to de- scribe these models in OWL [8], the Semantic Web standard for metadata and ontologies.

Activity Level

This level represents the applications that need collaboration inside the group of users and/or devices. It includes software elements implementing the activity's business, as well as user interfaces, security modules, etc. Among these elements, (at least) those relevant to collaboration are represented in the architectural model corresponding to this level of abstraction. Only collaboration-related elements of the activity level model will be taken into account in the refinement process. Nevertheless, other business elements (non collaboration-related) are also included and can be used in order to represent the whole activity.

This level provides, also, a session level abstraction. It describes the way members in a group are organized within sessions, where they can send and receive data flows. The main issue is that of determining a high-level collaboration schema that meets the needs of activity's collaboration. Hence, it supports collaborative sessions and can determine those elements needed to implement these sessions.

Activity Level Model

We have chosen an ontology-based model because it constitutes a standard knowledge representation system, allowing reasoning and inference. Moreover, ontologies facilitate

knowledge reuse and sharing through formal and real world semantics. Therefore, ontologies are high-level representations of business concepts and relations. These representations are close to developers' minds and therefore well suited to depict application level models. We have chosen to describe these models in OWL \cite{owl}, the Semantic Web standard for metadata and ontologies.

The activity level model is a graph, inspired by dynamic collaboration diagrams~\cite{villemur_dcd}, shows the detailed structure of one or more session. A session is a set of data flows. Each data flow goes from a sender component to a receiver component (components are deployed on nodes). Sender and receiver components may have text, audio or video as types. Flows are labelled with data types (audio, text and video) and the session to which they belong. This graph is expressed in the GraphML language (an XML dialect for representing graphs \cite{graphml}). This model details the structure of one or more sessions (a set of data flows). Each flow occurs between the sender and the receiver component (deployed on nodes).

1 G_Refine()
2 {
3 Let $\mathbb{A}_n$, $\mathbb{A}_{n-1}$ be the set of configurations
at level n and level $n-1$.
4 Let $A_{n,i} \in \mathbb{A}_n, i \in \mathbb{N}$, be a given configuration
5 Compute $\mathbb{A}_{n-1,i} = \{A_{n-1,j} \in \mathbb{A}_{n-1}$ such that:
$\exists p_1 \dots p_k \in P : A_{n,i} \xrightarrow{p_l \cdots p_k} A_{n-1,j}, j \in \mathbb{N}\}$
6 }

Figure 10.1 Activity level Refinement

Middleware Level

This level provides a middleware model that masks low-level details (like TCP sockets, UDP datagrams, IP addresses, multicast, etc.) in order to simplify the representation of communication channels. In actual fact, this level furnishes an abstract view of distributed systems, so that they become transparent for upper levels. For example, this model may be based on abstractions like Event-based Communications, Peer-to-Peer, Remote Procedure Calls or Remote Method Invocation.

Middleware Level Model

For the middleware level, we have retained the Event-Based Communication (EBC)\cite{ebc}. It represents a well established paradigm for interconnecting loosely coupled components and it provides one-to-many or many-to-many communication pattern. This model is a detailed graph containing a set of event producers (EP), event consumers (EC)

and channel managers (CM) connected with push and pull links. Multiple producers and consumers may be associated through the same CM. Since this model represents a graph, it can also be expressed in the GraphML language.

Activity Middleware Refinement

As the activity level and the middleware level models are represented by graphs, graph grammar theories represent an appropriate formalism to handle the refinement process. We provide a graph grammar-based implementation of the refinement. This implementation, called Grefine() (see \autoref{tab:ref_pro}), corresponds to the application of a set of graph grammar productions p1… pk that implement the refinement of an architectural configuration from level n to level n-1.

We use a graph grammar, that addresses the refinement of a given activity level architecture to all possible EBC level architectures. The productions of this graph grammar consider data collaboration components (e.g. \texttt{ReceiverComponent and SenderComponent) as non-terminal nodes and EBC entities (EPs, ECs and CMs) as terminal nodes. A session involving several senders and receivers is refined as a CM connected to several EPs and ECs.

$GG_{ACT \rightarrow EBC} = (AX, NT, T, P)$ with:
$T = \{CM(cm, s, m), EC(ec, m), EP(ep, m)\}$,
$NT = \{R(ar, m), S(as, m)\}$ and
$P = \{p_1, \ldots, p_4\}$

$p_1 = ($
$L = \{R(ar, m1), A(as, m2), S \xrightarrow{data,s} R\}$;
$K = \{\ \}$;
$R = \{EC(ec1, m1), EP(ep1, m2), CM(cm1, s, m1),$
$CM \xrightarrow{push} EC, EP \xrightarrow{push} CM\}$;
$C = \{$
$ic2 = (CM(cm1, s, m1), audio, s/push, S, out/out),$
$ic1 = (CM(cm1, s, m1), audio, s/push, R, in/in)\})$

$p_2 = ($
$L = \{S(as, m2), CM(cm1, x, m1), S \xrightarrow{data,s} CM\}$;
$K = \{CM(cm1, s, m1)\}$;
$R = \{EP(ep1, m2), CM(cm1, s, m1), EP \xrightarrow{push} CM\}$;
$C = \{\ \})$

$p_3 = ($
$L = \{R(ar, m2), CM(cm1, s, m1), CM \xrightarrow{data,s} R\}$;
$K = \{CM(cm1, s, m1)\}$;
$R = \{EC(ec1, m2), CM(cm1, s, m1), CM \xrightarrow{push} EC\}$;
$C = \{\ \})$

$p_4 = ($
$L = \{CM(cm1, s, m1), CM(cm2, s, m2)\}$;
$K = \{\ \}$;
$R = \{CM(cm1, s, m1)\}$;
$C = \{\ \})$

Figure 10.2 Activity- Middleware Refinements

In order to refine a given collaboration architecture into a set of EBC architectures, the graph grammar GGACT□EBC, detailed in the \autoref{tab_ref_gg}, is used. In this graph grammar, non-terminal nodes are collaboration entities while terminals nodes are EBC entities. For clarity's sake, only the case of audio sessions is considered. Therefore, the productions of this graph grammar refine ReceiverComponent and SenderComponent (R and S) into EPs, ECs and CMs. Similar grammar productions have been developed for text and video components.

The production p1 refines the pattern consisting of an S (denoted as as) connected to an R (denoted as ar) by the introduction of an event consumer, an event producer and a channel manager for a specific session (denoted here by x). Connection instructions ic1 and ic2 consider the push options. Other Rs (resp. Ss) linked to ar (resp. as) are connected to the created channel manager.

The production p2 refines the pattern consisting of an S connected to a channel manager. The production p3 refines the pattern consisting of an R connected to a channel manager for a specific session (denoted here by x). The production p4 guarantees that only one channel manager is kept for each session.

Reconfiguration Rules

To make this system adaptable, reconfiguration rules are needed to adapt the ontology instance to the current situation. As events play a major role in our application, the transformation of entities needs to be triggered. Here, events could occur at activity level, i.e, addition of new participants, changing an action, transfer of a participant from one group to another, new connection between investigators from different groups, etc. The events could also take the form of resource context changes, e.g, parameters like the energy of a device, bandwidth range, CPU processing capacity, RAM availability etc. We use SWRL\cite{swrl} rules to define our adaptation policy. The application designer defines these rules according to context changes he wants to handle. As we explained earlier, each level executes its own rule when there is an appropriate need. Thanks to the decision model at every level, not all requests are passed to the control center. If there is no solution for an event at a particular level, then it triggers the higher level. In SWRL, the head points to the adaptation transformations whereas the body indicates the context of ontology elements. This reconfiguration rules are really useful for adapting the scenario dynamically. In our case, adaptation events are classified into 2 types:

High Level Event Reconfiguration

Table 3 shows an example in the activity level evolution. Consider the supervisor is managing two coordinators and each coordinator has their own investigator group of two robots. As we explained previously, actors communicated through coordination flows. When there is a need to establish a cooperation flow between the investigators among the different groups, the

following rule is applied. The conditions for establishing the flow is explained in the scenario illustration.

$$Supervisor(?S) \wedge CoordinatorGroup(?CG) \wedge managesGroup(S, CG) \wedge Coordinator(?CO1) \wedge Coordinator(?CO2) \wedge hasMember(CG, CO1) \wedge hasMember(CG, CO2) \wedge InvestigatorGroup(?IG1) \wedge InvestigatorGroup(?IG2) \wedge managesGroup(CO1, IG1) \wedge managesGroup(CO2, IG2) \wedge Investigator(?I1) \wedge Investigator(?I2) \wedge hasMember(IG1, I1) \wedge hasMember(IG2, I2) \wedge CoordinationFlow(?CF1) \wedge hasReceiver(CF1, CO1) \wedge hasSender(CF1, S) \wedge CoordinationFlow(?CF4) \wedge hasReceiver(CF4, CO2) \wedge hasSender(CF4, S) \wedge CoordinationFlow(?CF3) \wedge hasReceiver(CF3, I1) \wedge hasSender(CF3, CO1) \wedge CoordinationFlow(?CF5) \wedge hasReceiver(CF5, I3) \wedge hasSender(CF5, CO2) \rightarrow SWRLb : createOWLThing\ (CooperationFlow(?CPF1)) \wedge hasSender(CPF1, I2) \wedge hasReceiver(CPF1, I1)$$

Table 4: High Level Event Reconfiguration example

Low Level Event Reconfiguration

Table 4 lists an example of changes in the communication level constraints. If the energy level of a participant's device surges below the threshold (e.g, below 20 %), a transformation is performed to move the channel manager from this device. In this case, the decision is made by the network level of the coordinator that manages the participant having an energy problem by triggering the SWRL rule. This clearly indicates that the intelligence is distributed and not all decisions are taken by the control center. More details will explained in the next section about the *dischargeParticipantDevice*.

$$Supervisor(?S) \wedge CoordinatorGroup(?CG) \wedge managesGroup(S, CG) \wedge Coordinator(?CO1) \wedge hasMember(CG, CO1) \wedge InvestigatorGroup(IG1) \wedge managesGroup(CO1, IG1) \wedge Investigator(?I1) \wedge Investigator(?I2) \wedge hasMember(IG1, I1) \wedge hasMember(IG1, I2) \wedge CoordinationFlow(?CF1) \wedge hasReceiver(CF1, CO1) \wedge hasSender(CF1, I1) \wedge CooperationFlow(?CPF1) \wedge hasReceiver(CPF1, I1) \wedge hasSender(CPF1, I2) \wedge Device(?device) \wedge hasID(?device, ?idDevice) \wedge hasHardwareProfile(?device, ?hardware) \wedge hasParameter(?hardware, ?param) \wedge sameAs(?param, powerlevel) \wedge hasValue(?param, ?value) \wedge swrlb : lessThan(?value, 20) \rightarrow adapt : dischargeParticipantDevice(?idDevice)$$

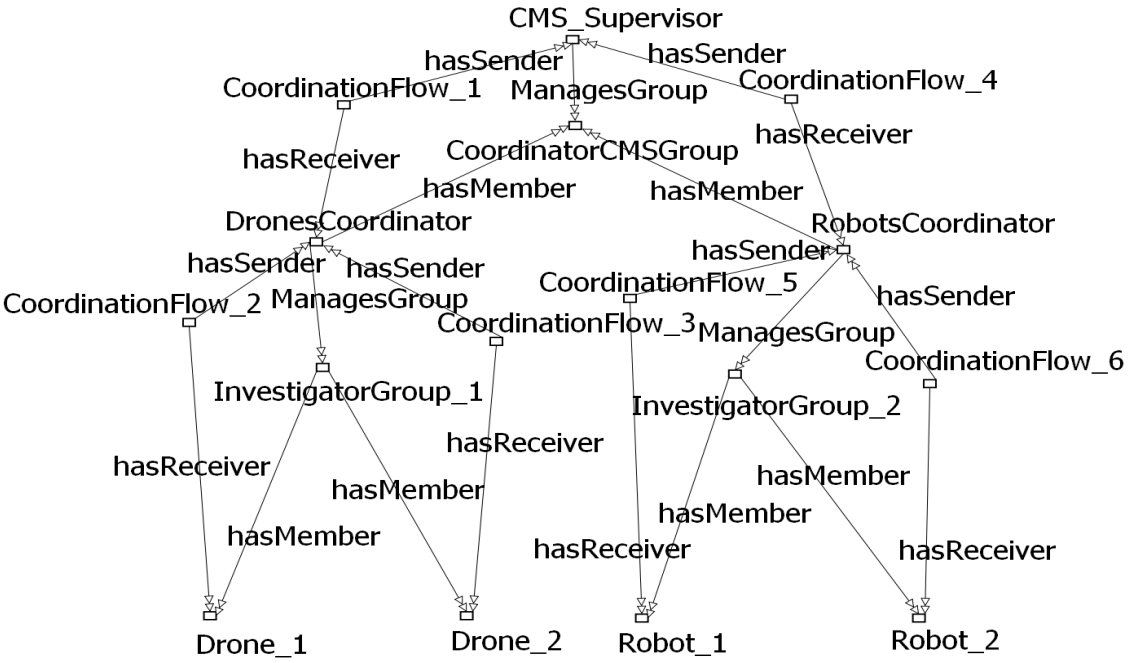
Table 5: Low Level Event Reconfiguration example

This section presents a top-down approach with respect to the architecture models used for the activity and middleware levels. It also illustrates the refinement processes that exist between levels in the adaptation process. The architectural configuration of the activity level is captured and represented in the CMS ontology instance {as shown in the upper part of

Figure~\ref{fig:initPhase}}. Here, concepts and relations from the activity-specific and from the generic collaboration ontologies are instantiated. To refine this activity model, rules are processed over these ontological instances. Firstly, activity-specific rules are processed and then, generic collaboration rules are applied. Then, rules are processed for each instance of \texttt{GCO:CommunicationFlow} found, thus creating the corresponding channel managers, event consumers and event producers.

The resulting set of ontological instances form a collaboration level graph. It is translated into GraphML language by means of XSLT transformation. In order to refine this collaboration level, a detailed graph grammar is used. This produces a valid configuration that contains terminal nodes only (i.e. nodes belonging to the EBC level). It is obtained by application of the sequence graph grammar Production. This refinement creates a detailed deployment descriptor used by a deployment service in order to utilize the indicated components on each device, thus implementing the required activity level session.

Here two situations are considered. The first one deals with an activity-based event and second with resource change event. Consider a situation where an investigator (Drone) intends to drop water in an area where another investigator (denoted as Robot2) is already busy. The simplified version of this situation at activity and communication levels is shown in Figure~\ref{fig:initPhase}.



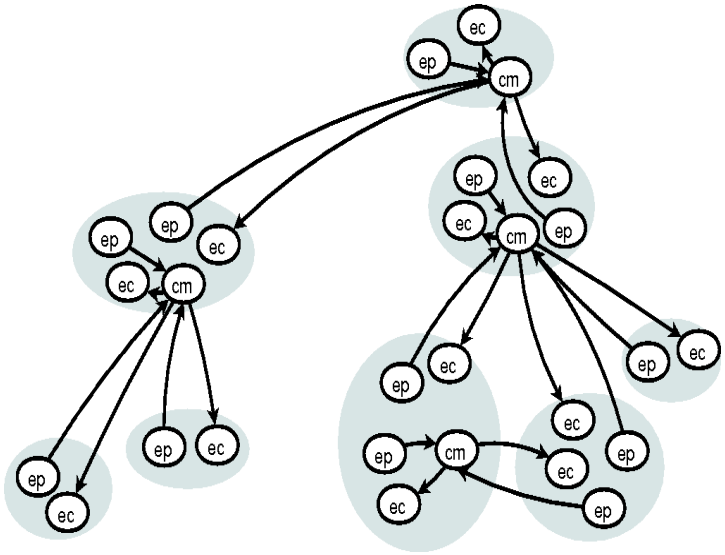


Figure 10. 3 EBC Deployment Descriptor After Initial Refinement

In this case, the plane2 has to be notified as soon as possible, not to drop water. Another investigator (denoted as Fireman1), aware of the situation, establishes a coordination flow to its coordinator and subsequently the coordinator reports to the supervisor. The latter already knows the position of the approaching plane2 and notifies the plane not to drop any water in that area via the plane coordinator.

As there is no connection between the plane2 and the Fireman2, the latter has to obtain the supervisor's decision. The other simpler solution could be to establish a connection between the plane2 and Fireman1 through use of a new cooperation flow obtained by running the SWRL rule (Table \ref{rule1}). Thus, Fireman1 can warn the pilot not to drop any water. After processing the rule, the activity level and the communication level have been changed as shown in Figure~\ref{fig:reconfResult}.

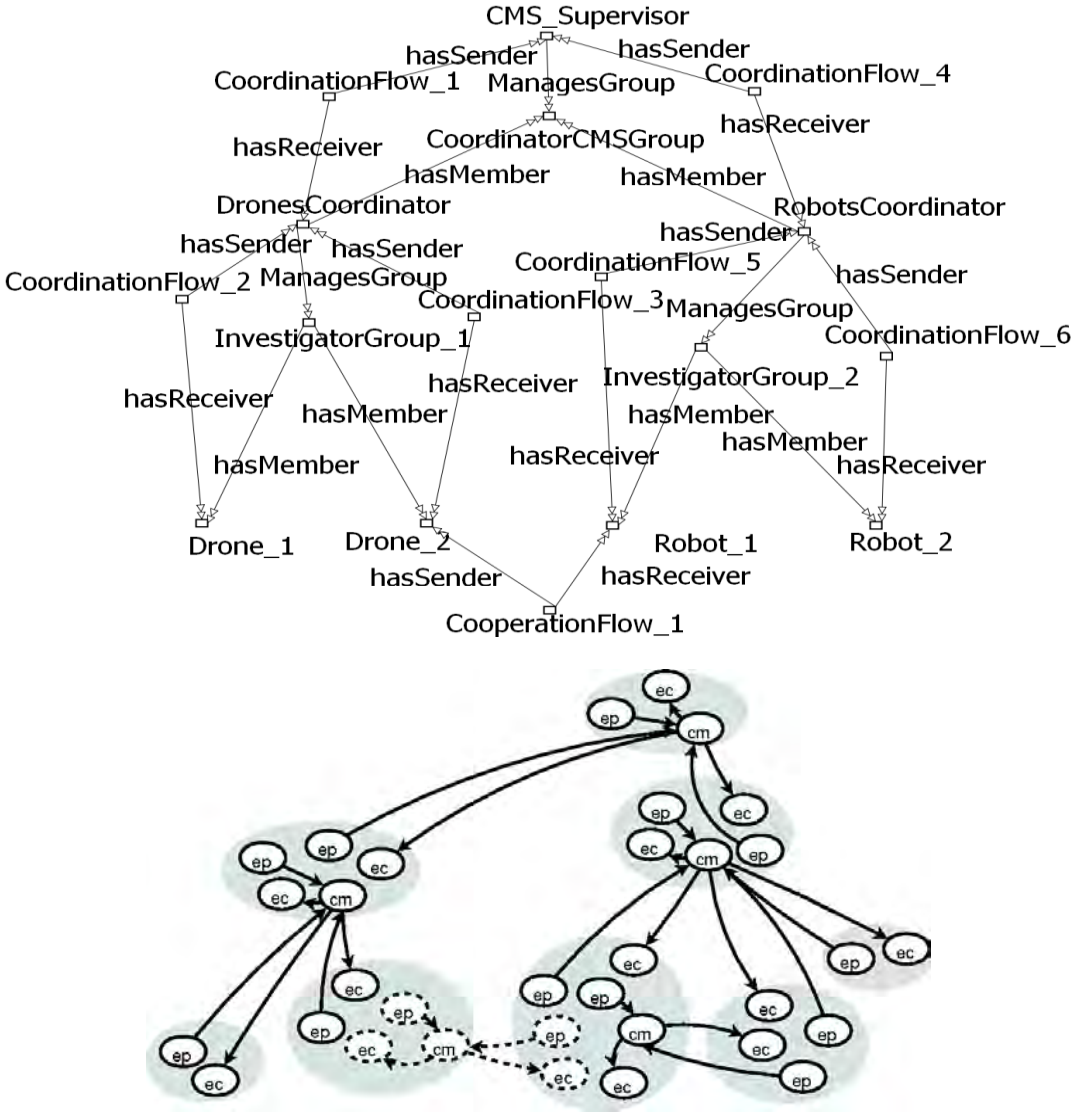


Figure 10. 4 EBC Deployment Descriptor After reconfiguration event

With the new cooperation flow between Robot1 and the Drone, a channel manager is needed in Robot1's device and the corresponding event producer and event consumer must be deployed in the Drone (represented by dotted line on Figure 4). Here, our approach does not allow the supervisor to take a decision as it takes numerous steps but acting locally thus solving time-constraint problem.

An example for context-awareness is illustrated in the following example. Consider an investigator (denoted as I2) sending messages to his coordinator via another investigator (denoted as I1). As each one is entrusted with his own task, may arise a situation where I1 could not handle this communication as he needs to move to another location or his power runs out. So he triggers the coordinator to make a decision because communication between

I2 and coordinator is very important. In case of I1 moving to somewhere else, a solution could be another investigator establishing a new connection with I2.

In case of I1's power deficiency, this operator can be discharged from EBC elements that can be running on one of his neighbors. This decision is explained in the rule shown in Table \ref{rule2}. In the first part of the rule, the current situation is identified and due to the power deficiency, the procedure dischargeParticipantDevice is triggered. In our case, the channel manager moves from I1 either to coordinator or to I2. By doing so, I2 can send this prioritized flow for a sufficient time through I1 to coordinator. Clearly, we act only at the communication level, i.e, on EBC. Nothing has been modified in the activity level and the new EBC descriptor is shown in Figure~\ref{fig:ebcReconf}.

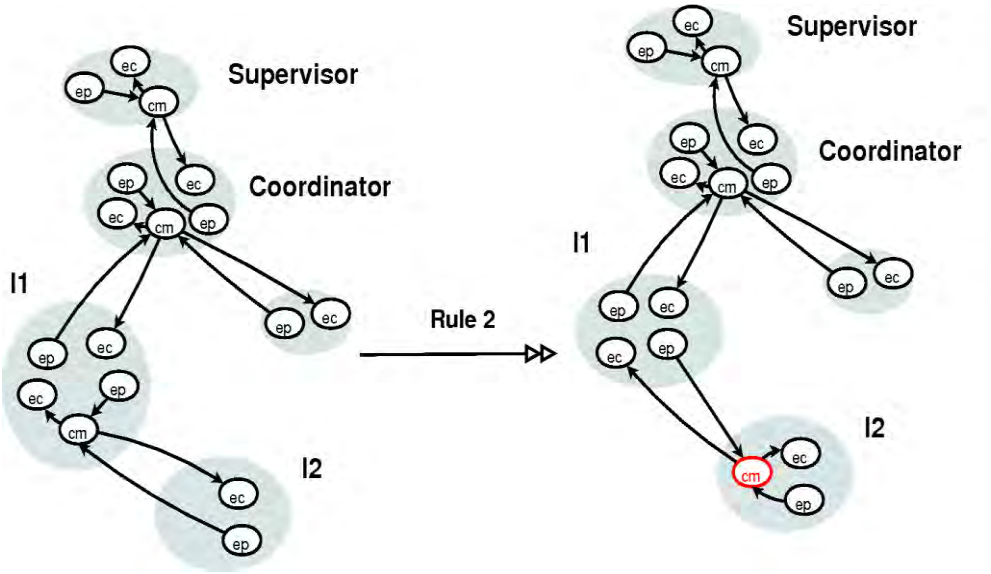


Figure 10. 5 EBC Redeployment After Power Diminution Event

In this section, we will consider three cases: adaptive mechanisms to trace the neighour, adaptation due to communication lost and adaptation due to energy level diminution.

Adaptation due to energy level diminution

The resource manage monitors the node's running resources. Before the end of the energy's lifetime, information is sent to the context manager to take actions. When a node's energy reaches the threshold value, with the 2 interfaces working at same time, a trigger enables an adaptation action. The network interface used by the lowest flow priority is switched off. The adaptation is not instantaneous and the time taken for this action is indicated by Δt (in Figure~\ref{fig:energy}). By doing this, the battery life time is extended. The curves in Figure~\ref{fig:energy} indicate that the available energy percentage at 10 % is 60 minutes if the two interfaces are used. But the available energy percentage at 10 % is 90 minutes if the adaptation action is taken (using only one interface).

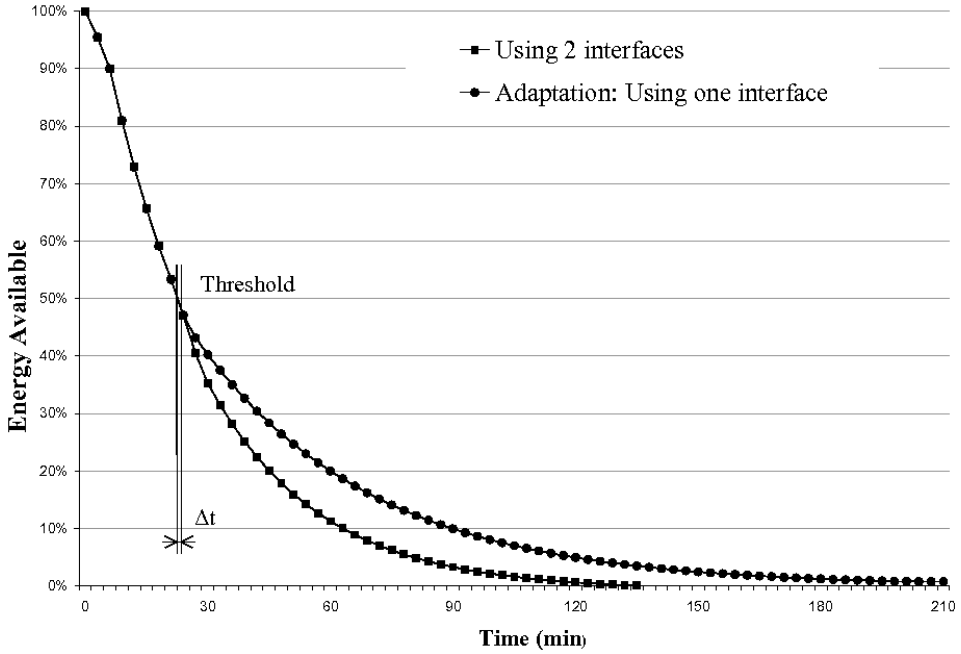
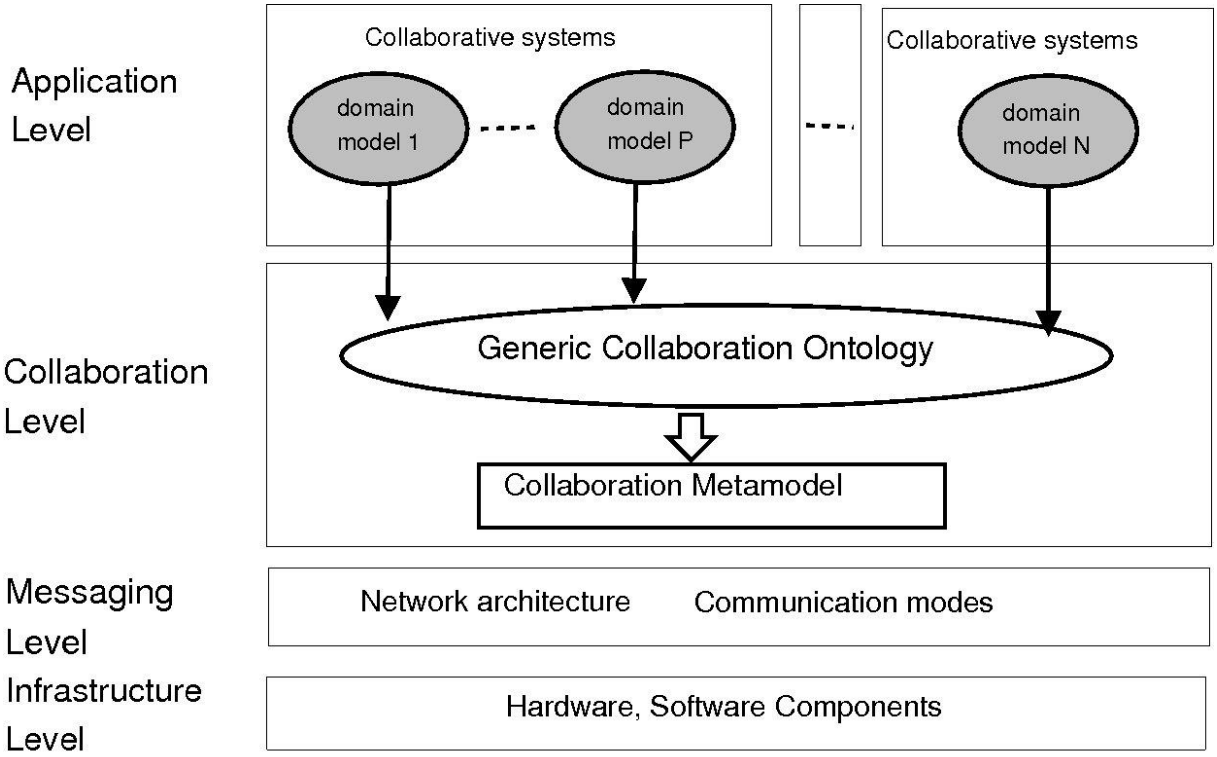


Figure 10.6 Adaptation due to energy level diminution

Implementation

We implemented our work using FACUS (Framework for Adaptive Collaborative Ubiquitous Systems), an architecture that supports semantic adaptation enabling the awareness of the presence/absence, roles and tasks of collaborators. This framework is based on a generic multi-level modeling approach that ensures multi-level adaptation. A generic collaboration model, based on Semantic Web technologies is proposed in order to support real-time collaboration between groups of participants working together in different tasks. The framework defines common interfaces for collaborative systems to enable the management of cooperative actions.



In this framework, a node represents a communicating entity which takes part in a collaborative activity. Nodes may represent human users (i.e. human-controlled software components) but also autonomous software components, agents, etc. Whether a node is an autonomous software component or it is a human-controlled component, it has to be executed on a physical machine. Such machines are represented by the concept Device (Node is linked to Device by the property hasHostingDevice). The execution context of the node will depend on the resources of the device that hosts it.

At the present time, a minimal set of device properties is considered, containing IP addresses (hasIpAddress), operating system (hasOS), available memory (hasAvailableMemory), CPU load (hasAvailableMem) and battery level (hasBatteryLevel).

The concept Flow represents a communication link between two entities. Therefore, Flow is linked to Node by two properties: hasSource and hasDestination. In this ontology, flows are considered as being unidirectional, and thus if a bidirectional communication between two nodes is required, it will be represented by two instances of Flow with two opposite directions.

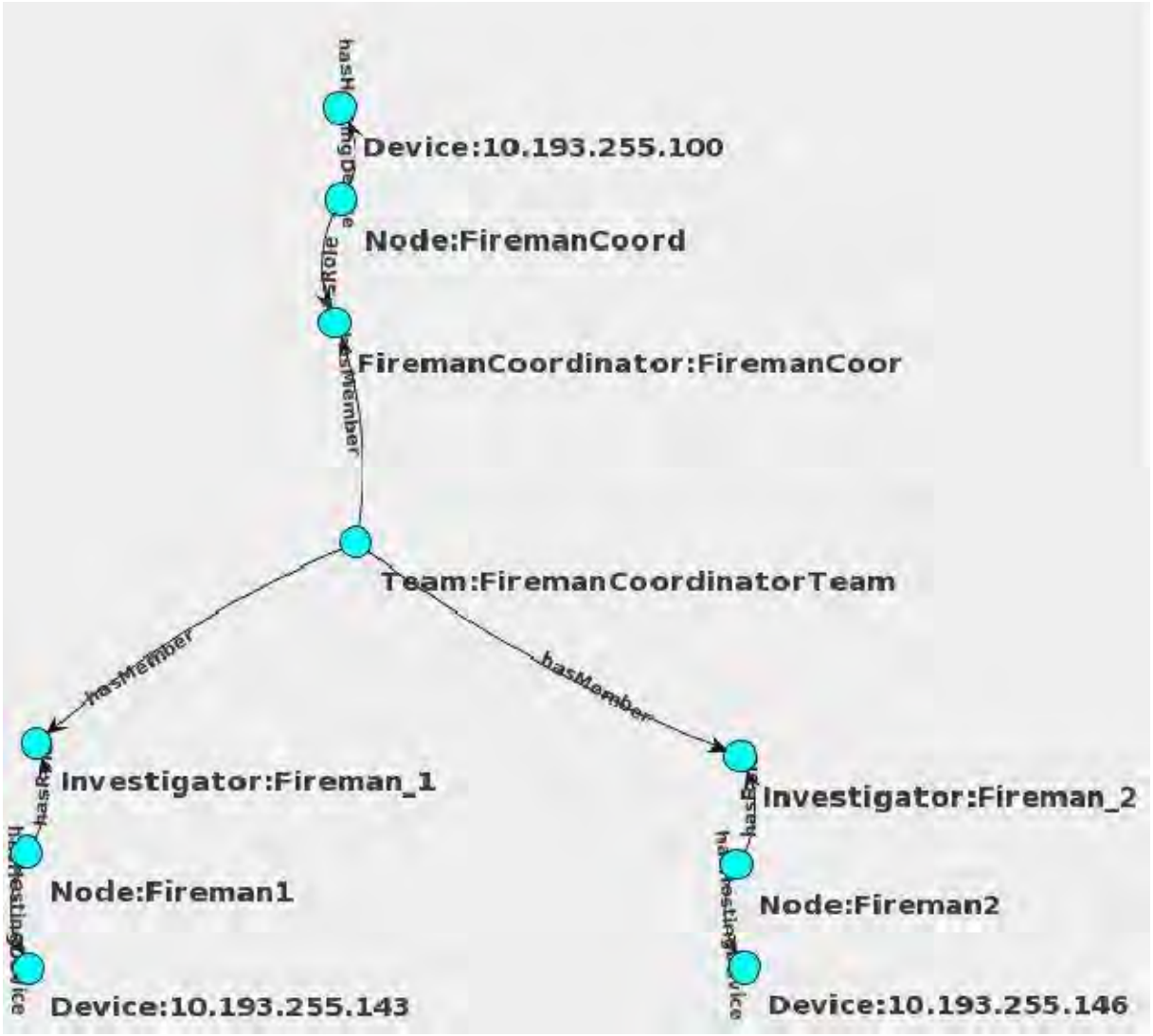
In order to handle data flows, nodes use external software components that are deployed on the same device as them. These external components are represented by the Tool concept.

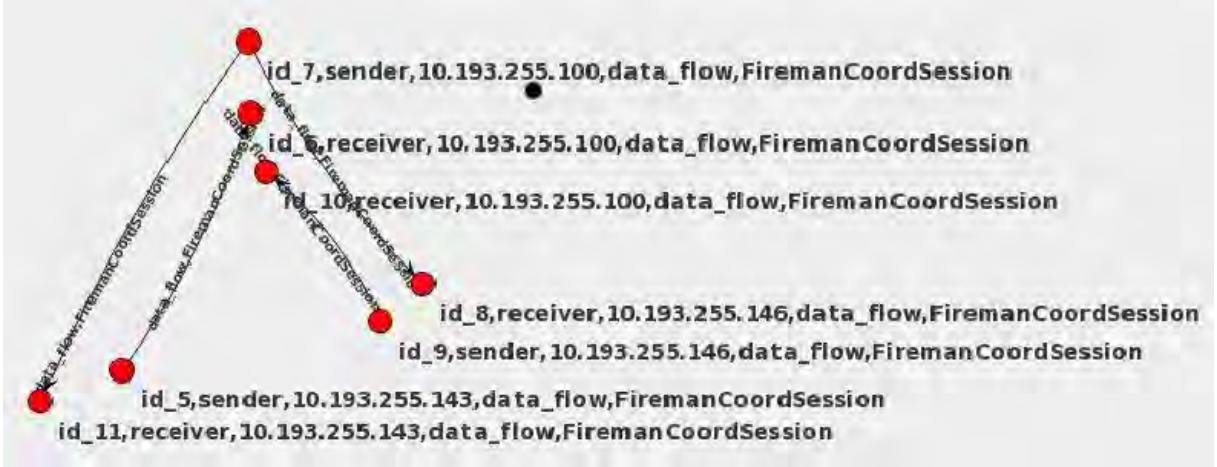
Tools are composed of several components, e.g., a sender component and a receiver component. Therefore the Tool concept is related to a concept called Component through the property hasComponent. Since components handle flows, a property called managesFlow links Component and Flow. Components have a data type (the same as the data type of the flow that they manage) and are deployed on a single device (isDeployedOn property which links Component and Device). SenderComponent and ReceiverComponent are linked to Flow by two sub-relations of managesFlow: sendsFlow and receivesFlow, respectively.

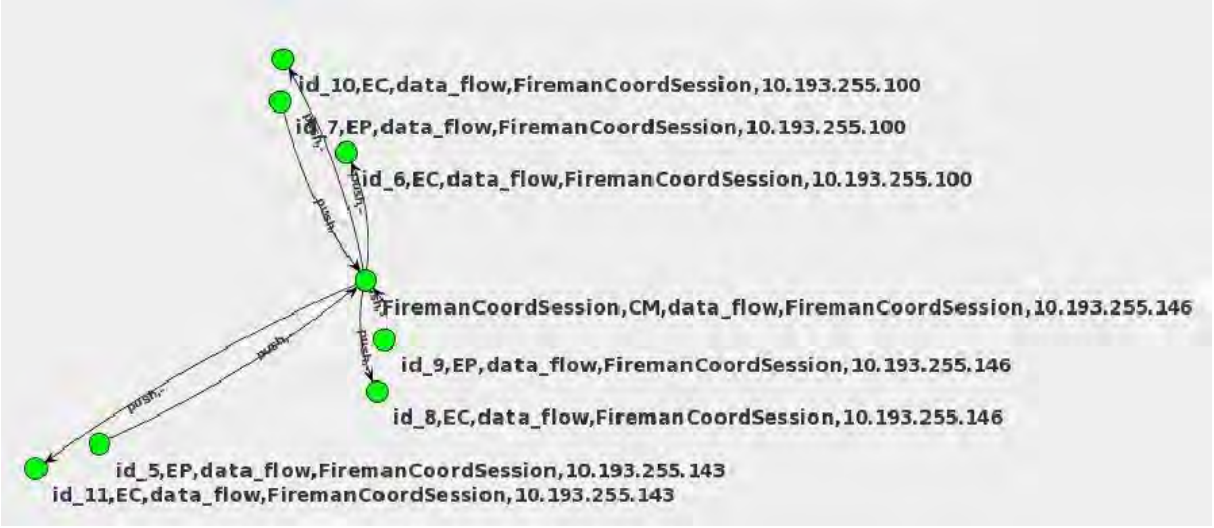
Finally, the Session concept represents a set of flows belonging to the same collaborative activity. The hasFlow property relates a session to a flow. The inverse property, belongsToSession, is functional, i.e., a flow belongs to a single session. Since flows are related to nodes, nodes are indirectly related to one or more sessions depending on the flows that connect them to other entities.

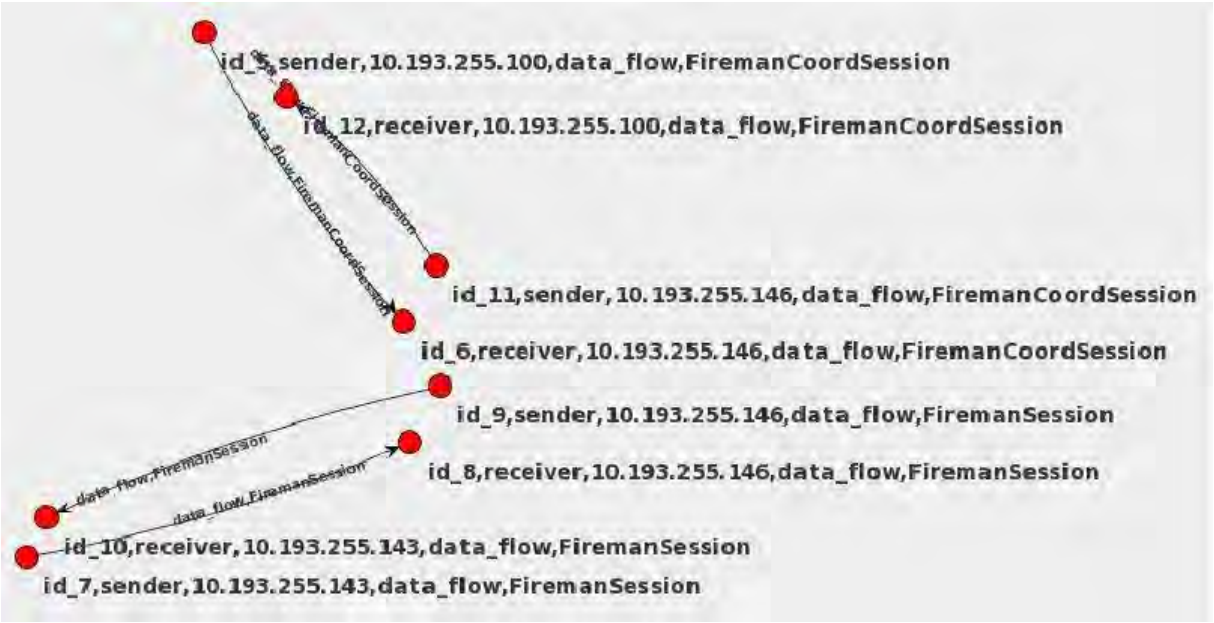
In FACUS, we have chosen one group (fireman)to show the adaptation. Initially, the fireman1 and fireman2 are connected to firemancoordinator at WiFi infrastructure mode. This initial stage of this situation at application, collaboration and massaging levels are shown in DIGRAM

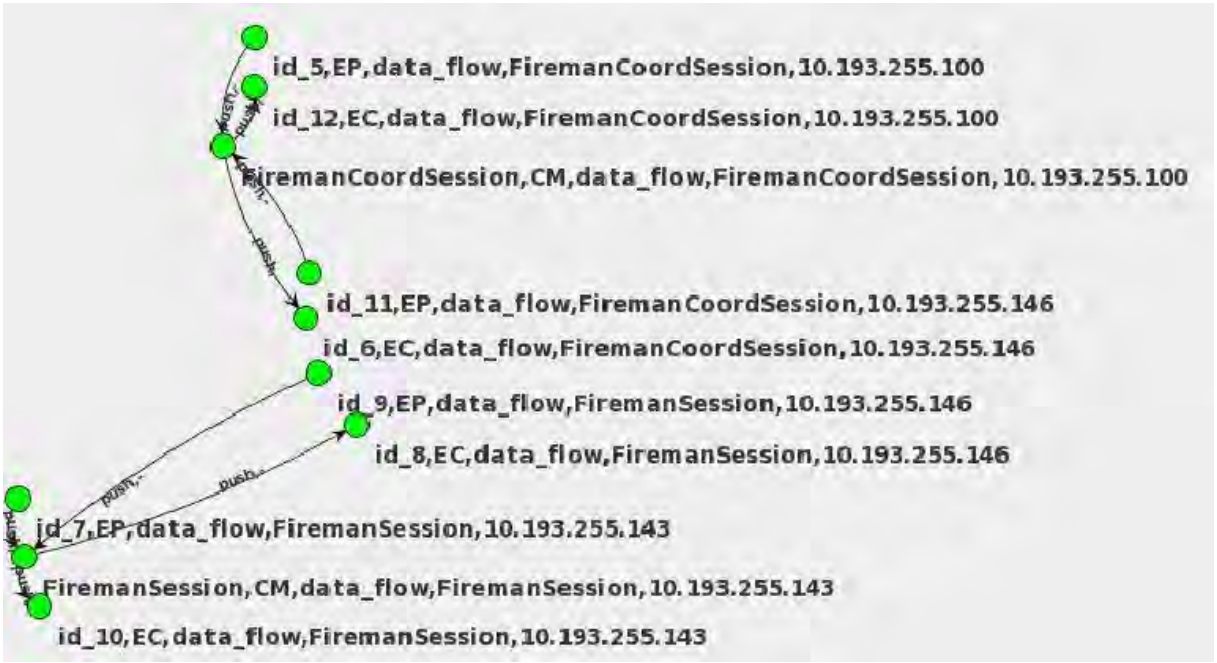
Consider fireman1 lost the connection with its coordinator while searching for a victim. Once the connection is lost, the coordinator aware this situation and thanks to our policies, the coordinator and the other investigator will shift to adhoc mode. Also, the local decision of the lost investigator changes to adhoc mode automatically such that communication is established. Figure and Figure show the adaptive collaboration and middleware graph in the implementation.











We use Morse to simulate the different scenarios. Morse \cite{morse,gilberto} is a versatile simulator for multi-robots applications. It enables realistic and dynamic environments with other interacting agents like humans, objects, etc. It is built on top of Blender, using its powerful features and extending its functionality through Python scripts. Simulations are executed on Blender's Game Engine mode, which provides a realistic graphical display of the simulated environments. A realistic simulation produces exactly the same data as the CA of the actual robot, and it is possible to demonstrate the adaptive mechanisms at command panels. It will allow the designer to simulate the real time scenario at various levels of abstraction as it allows him to specify the functions he wants to simulate and the ones he wants to evaluate. For instance, when evaluating an adaptive communication, it is not necessary to worry about lower level actions but an abstract simulation is sufficient. For example, if there is a communication loss between the participants, the object turns red. After adaptive mechanism, it turns to green so that the designer understands that the communication is retained (the adaptive algorithms are implemented in Java). The Figure 6 shows objects like robots, buildings, victims and the terminal (command mode) is the end result of adaptive algorithms.

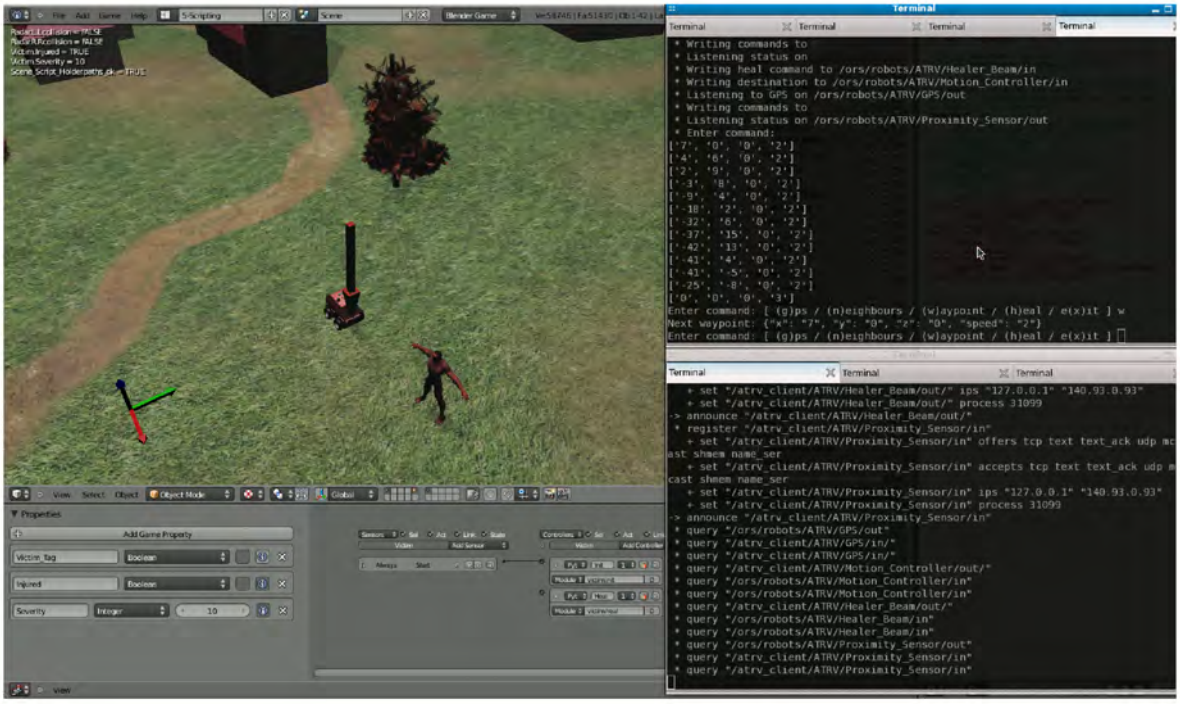


Figure 10.7 : A screen shot of MORSE simulator

LEGEND:

The different colors of the components used to identify the base functions. For example, green color represents the communication node manager whereas violet represents the network connection manager etc. Also, we have distinguished the messages by indicating the colored arrow (red, blue or green).

11 Architecture Deployment

Communication agent is a software module that will be installed in the ROSACE device and there are many ways of interpreting the type of deployment in real life.

Centralized

Here, supervisor controls firemen and robots without the intervention of coordinator. Messages flow between the different actors through supervisor. CAs communicates to each other thanks to WIFI IFM.

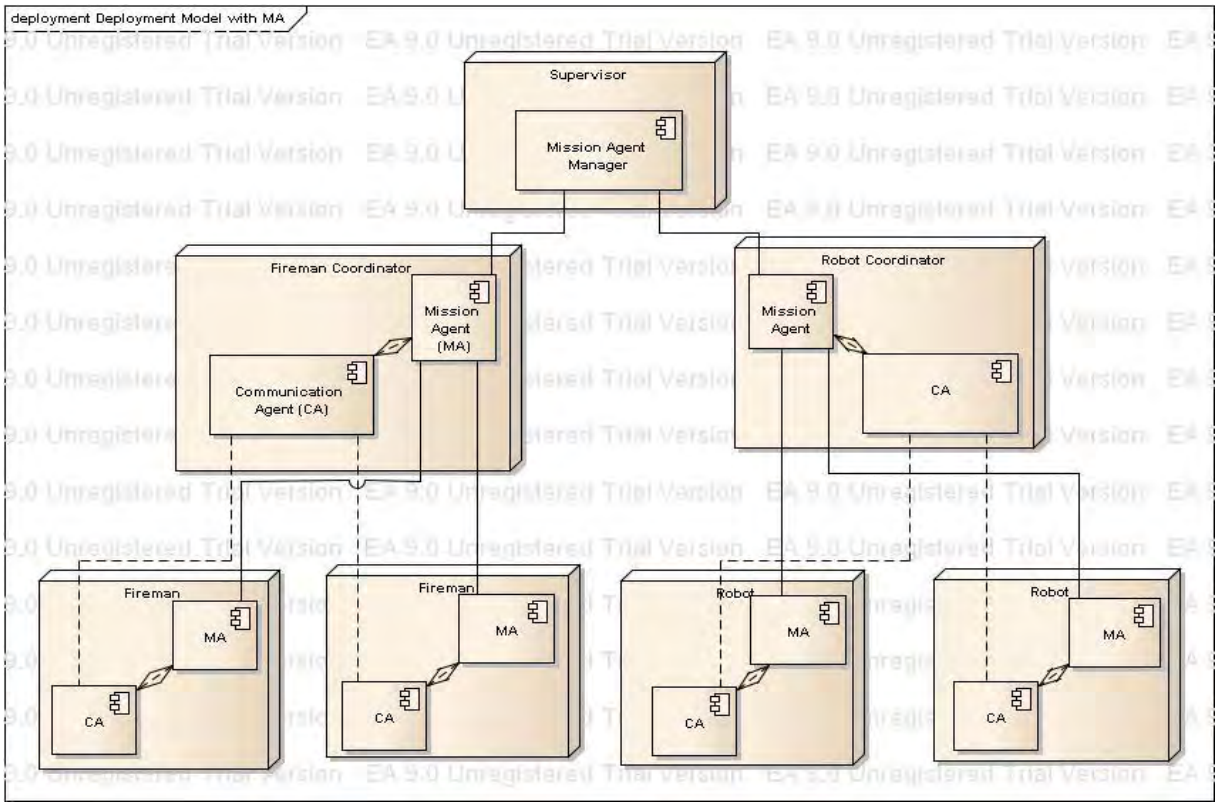


Figure 11.1 Implementing ROSACE devices

Also, we can include one coordinator to control both firemen and robots and coordinator is directly connected to supervisor. If supervisor wants to contact any of the investigators, it should pass through the coordinator. These diagrams only show the IFM mode but it's up to the operational team to choose the one suitable for their needs.

In centralized mode, there are many disadvantages like single point of failure. In case of failure in the supervisor or network loss, investigator's communication will be interrupted. Also, we have to take into account the decision components, whether it's centralized or distributed.

Distributed

Our scenario uses this deployment architecture for many reasons. One such being is to distribute the knowledge and redundancy of sources. It's also feasible and reasonable to have coordinators for each type of investigators.

Hybrid

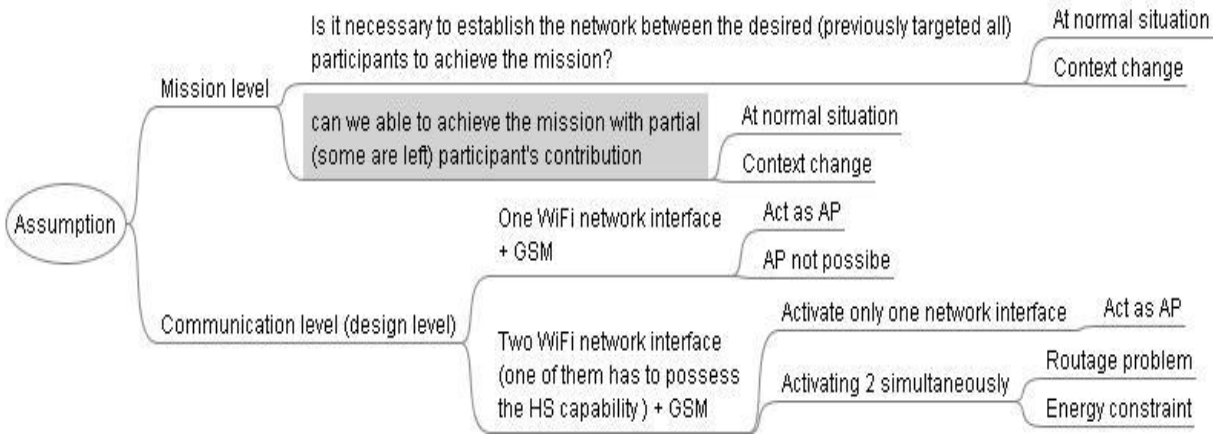
In case of hybrid, supervisor includes some special functionality that acts as a coordinator. Even though it has some potential advantages, implementing such type is expensive.

Conclusion and Open Issues

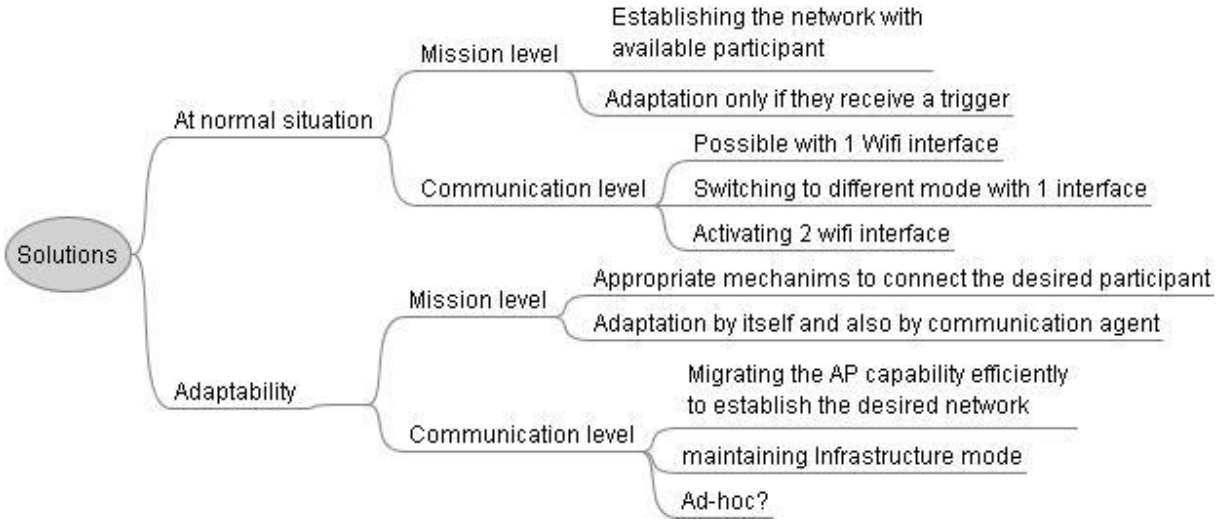
In this book, a multi-level modeling approach designed to support group communications has been detailed. For such a complex system, the whole scenario has been divided into different levels. Ontology has been used at the top two levels while we retained event-based communication at the third level to establish the flows between devices. The relations and transformations from top level to lower levels are presented. If a change arises in the environment, reconfiguration can be achieved by using SWRL rules. This is key for an appropriate management in case of changing resources in the environment. By using this approach, we could allow the architectural reconfigurations at run-time to handle the activity's evolving conditions. Unlike previous approaches, distributed decision model allows our architecture to make adaptive mechanism at lower levels and not influencing the higher ones.

Many open issues need to be discussed in our scenario. In case of environment change, triggers play an essential role to notify the decision components for initiating the adaptive policies. Even though the triggers are asynchronous and synchronous messages, a generic model for failure cases need to be analyzed. Also, assigning priorities to flows as well as monitoring resource deficiencies are our main challenges. Non-cooperative situations like low QoS and performance degradation are the topics worth for future work.

Some Mindtree files to illustrate our autonomic computing



Assumption



Solutions

EstablishingLocalnetwork

[1] Nasser, N., Hassanein, H.: Adaptive bandwidth framework for provisioning connection-level qos for next-generation wireless cellular networks. Canadian Journal of Electrical and Computer Engineering 29(1) (2004) 101–108

[2] Sun, J.Z., Tenhunen, J., Sauvola, J.: Cme: a middleware architecture for network-aware adaptive applications. In: Proc. 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. Volume 3., Beijing, China (2003) 839–843

[3] Exposito, E., Senac, P., Diaz, M.: FPTP: the XQoS aware and fully programmable transport protocol. In: Proc. The 11th IEEE International Conference on Networks (ICON'2003), Sydney, Australia (2003)

[4] Welch, L.R., Masters, M.W., Madden, L.A., Marlow, D.T., Irey, IV, P.M., Werme, P.V., Shirazi, B.: A distributed system reference architecture for adaptive qos and resource management. In: Proceedings of the 11 IPPS/SPDP'99 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing, London, UK, Springer-Verlag (1999) 1316–1326

[5] Ketfi, A., Belkhatir, N., Cunin, P.Y.: Adaptation dynamique, concepts et experimentations. In: Proceedings of ICSSEA. (2002) In French.

[6] Halima, R.B., Drira, K., Jmaiel, M.: A qos-oriented reconfigurable middleware for self-healing web services. In: ICWS. (2008) 104–111

[7] Brandt, S., Nutt, G., Berk, T., Mankovich, J.: A dynamic quality of service middleware agent for mediating application resource usage. In: RTSS '98: Proceedings of the IEEE Real-Time Systems Symposium, Washington, DC, USA, IEEE Computer Society (1998) 307

[8] Grace, P., Coulson, G., Blair, G.S., Porter, B.: A distributed architecture meta-model for self-managed middleware. In: ARM '06: Proceedings of the 5th workshop on Adaptive and reflective middleware (ARM '06), New York, NY, USA, ACM (2006) 3

[9] Vienne, P., Sourrouille, J.L.: A middleware for autonomic qos management based on learning. In: SEM '05: Proceedings of the 5th international workshop on Software engineering and middleware, New York, NY, USA, ACM (2005) 1–8

[10] Preuveneers, D., Berbers, Y.: Multi-dimensional dependency and conflict resolution for self-adaptable context-aware systems. In: ICAS '06: Proceedings of the International Conference on Autonomic and Autonomous Systems, Washington, DC, USA, IEEE Computer Society (2006) 36

[11] Phung-Khac, A., Beugnard, A., Gilliot, J.M., Segarra, M.T.: A Model of Self-Adaptive Distributed Components. In: Proceeding of the 4th ECOOP Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT'07), Berlin, Germany (2007)

[12] Bouassida-Rodriguez, I., Lacouture, J., Drira, K.: Semantic driven self-adaptation of communications applied to ercms. In: The 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), Perth (Australie) (2010 Avril)

[13] Smith, M.K., Welty, C., McGuinness, D.L.: OWL Web ontology Langage Guide. W3C Recommendation (2004) Url : http://www.w3.org/TR/owl-guide/.

[14] Sancho, G., Tazi, S., Villemur, T.: A Semantic-driven Auto-adaptive Architecture for Collaborative Ubiquitous Systems. In: 5th International Conference on Soft Computing as Transdisciplinary Science and Technology (CSTST'2008), Cergy Pontoise (France) (2008) 650–655

[15] Meier, R., Cahill, V.: Taxonomy of distributed event-based programming systems. In: ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems, Washington, DC, USA, IEEE Computer Society (2002) 585–588

[16] Ramachandran, K.N.;Belding-Royer, E.M.; AImeroth, K.C. DAMON: a distributed architecture for monitoring multi-hop mobile networks. In: Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on Issue Date, 4-7 Oct. 2004, On page(s): 601 – 609

[17] Chiang, C.-Y.J. ; Chadha, R. ; Levin, G. ; shihwei Li ; Yuu-Heng Cheng ; Poylisher, A. ;AMS: an adaptive middleware system for wireless adhoc networks, In. Military communications Conference, 2005. MILCOM 2005. IEEE issue date : 17-20 Oct, On page (s) : 2870-2876 Vol. 5, Atlantic City, NJ

[18] Soon-Hyeok Choi ; Perry, D.E. ; Nettles, S.M. ; A Software Architecture for Cross-Layer Wireless Network Adaptations: In. Software Architecture, 2008. WICSA 2008. Seventh Working IEEE/IFIP Conference , Issue Date : 18-21 Feb. 2008, 281 – 284, Vancouver, BC

[19] Atul Adya, Paramvir Bahl, Ranveer Chandra,Lili Qiu : Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks: In. MobiCom '04 Proceedings of the 10th annual international conference on Mobile computing and networking

[20] Joe Hoffert: Maintaining QoS for publish/subscribe middleware in dynamic environments: In. Proceeding DEBS '09 Proceedings of the Third ACM International Conference on Distributed Event-Based Systems

[21] Mocito, J. ; Rosa, L. ; Almeida, N. ; Miranda, H. ; Rodrigues, L. ; Lopes, A. ; Context adaptation of the communication stack :IN. Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference, Issue Date : 6-10 June 2005, On page(s): 652 - 655