UNIVERSITÀ DEGLI STUDI DI BERGAMO

Dipartimento di Ingegneria

Corso di L Dottorato di ricerca in MECCATRONICA, INFORMAZIONE, TECNOLOGIE INNOVATIVE E METODI MATEMATICI

Classe n. XXVII

A Distributed Intelligent Maintenance System based on Artificial Immune Approach and Multi-Agent Systems

Relatore:

Chiar.mo Prof./Chiar.ma Prof.ssa Nome Cognome

Correlatore:

Chiar.mo Prof./Chiar.ma Prof.ssa Nome Cognome

Luca Fasanotti

Matricola n. 51482

ANNO ACCADEMICO 2013 / 2014

Index

1. Context Analysis

1.1.Introduction

Nowadays we are witnessing a growing interest about the automatized supervision an control of every typology of plants.

These are primary due to a couple of factors, the increasing need of ever higher performance and reduction in the cost of monitoring hardware.

Regarding the first factor, the increasing demand of performance, both in terms of throughput and efficiency of the plants depends by the evolution of the globalized market: nowadays the European and American companies are subject to competition from companies located in emerging countries.

These new competitor are characterized to a cost of labour significantly lower and a quality of the products that is reaching the quality of European products.

for the European company the only feasible strategy in order to keep the market's leadership is to push the evolution of production plants in order to increase the quality of the products, reduce inefficiencies in all the aspects of the production process, minimize the downtime and reduce the wasting of raw material.

In order to reach these goals all the aspect of the production process need to be monitored in order to keep them under control and help enable an early detection of failures and lack of performance.

1.2.Scope of the work

The main goal of this research is to develop an innovative methodology for advanced maintenance in case of large geographical widespread plant such as waste water treatment plant or oil / gas pipeline.

This kind of plants are characterized by a set of limitation factor that makes the adoption of standard maintenance techniques very difficult and also limits the performance of these kind of approach. So an innovative methodology need to be developed in order to overcome these limitations.

The set of limitation factor are the following

- Very huge size of the plant: dimension over one hundred kilometers end to end are not so uncommon in this kind of plants
- Hash environment: parts of these plants are often located in very harsh environment like underground, deserts, mountain, forest, tundra etc. etc. these environment are very aggressive with the plants.
- Unpopulated zone: related to the previous element these plant are often located in zone with no presence of stable human settlements, these affect primary the capability of fast react to a failure of the plants.
- Difficulty of continuous monitoring: due to the dimension of the plant and the type of land covered by the system is very difficult to cover all the plant with an exhaustive monitoring network. This is caused primary by the Unfeasible cost of a cabled monitoring system and the lack of stability of wireless communications systems.

In order to overcome these constraints an innovative architecture for an Intelligent Maintenance System based on Artificial Immune System and Multi Agent System has been developed.

2. Artificial Immune System

Since an Artificial Immune System finds its motivation and roots from its biological counterpart, the first part of this introductory section is devoted to a better understanding of the way biological immunity systems work.

An Immune System is a complex adaptive system of cells and molecules, distributed throughout our body, which provides us with a basic defence against pathogenic organisms [46]. It consists of a great number of different cells and organs that work all together. The most important components are reported in Table 1.

Immune system component	Main function
Thymus	Production of T cells
Spleen	Production of different components of IS
Bone marrow	Key component of the lymphatic system, producing the lymphocytes
Lymphocyte (B-cells)	White blood cells that cooperate to detect and assist in the destruction of pathogens
Helper T-cells	Control of the immune response of other cells by coordinating their evolution and mitosis
Killer T-cells	Kill infected cells
Plasma cells	Lymphocytes that secrete antibodies when activated
Memory cells	Long life lymphocytes that keep tracks of previous infection and provide a fast response in case of reinfection
Antibodies	Proteins capable of detecting non-self elements by binding to them, disabling or helping other cells to destroy the foreign elements.

Table 1 Main components of an Immune System and their main functionality

It is easy to understand that the immune response is not provided by a single biological element. It emerges from the interaction of more organs, subsystems and components that reciprocally influence and regulate each other.

In analogy, in literature an Artificial Immune System (AIS) is described as an "adaptive system inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving."[9]. This biomimetic approach implies that an AIS needs to follow some organizing principles which are the basis of an immune system, like [37]:

- *distributed nature*: the same as the lymphocytes, they are capable to autonomously detect the presence of a local infection; in an AIS a detector

works without the needs of central coordination; this greatly enhances the robustness of the system by removing single points of failure;

- *multi-layered*: in the biological immune system, no mechanism guarantees by itself a complete safety; multiple layers of different mechanisms are combined to provide an overall protection; this means that also an AIS needs to rely on different systems or strategies that work all together in order to obtain the maximum performance, avoiding the convergence of the system on a sub optimal solution;
- *diversity*: similarly to a biological immune system, which varies from person to person, an AIS embedded on several identical machines needs to be different, in order to increase the robustness against undetected faults or unwanted behaviors. This is usually performed by using a random selection of initial values and through random mutations (needed to detect new failure modes);
- *disposability*: no single component of the human immune system is essential, any cell can be replaced; every cell death is balanced by the production of new cells; in addition, in an AIS the number of threads or agents running in the system needs to be regulated in order to avoid saturation of the computational resources of the entire system;
- *autonomy*: the immune system does not require external management or maintenance; it autonomously classifies and eliminates pathogens, and it performs a self-repair by replacing damaged cells; also an AIS needs to be autonomous and should not affect the availability and the performance of the machine (as a transparent system);
- *adaptability*: the immune system learns to detect new pathogens and retains the ability to recognize previously seen pathogens through an immune memory; analogously, an AIS needs to implement some self-learning mechanisms based on soft computing techniques, such as adaptive control charts or artificial neural networks, in order to estimate the system´s normal behavior and to detect anomalies never previously recorded;
- *dynamically changing coverage*: the immune system makes a space/time tradeoff in its detector set; it cannot maintain a set of detectors (lymphocytes) large enough to cover the space of all pathogens; at any time it maintains a

random sample of its detector repertoire which circulates throughout the body; this repertoire constantly changes through cell death and reproduction (in an AIS with random changes and selection mechanism);

- *behavioral identity*: in order to detect diseases never met before, an immune system does not work on the basis of a full knowledge of any antigens; it works by identifying and classifying the behaviors as safe or unsafe (using peptides as indicator of behaviors); hence, an AIS has to work by analyzing the overall system behavior and not only with a set of previously detected unsafe behaviors;
- *imperfect detection*: by accepting imperfect detection, the immune system increases the flexibility with which it can allocate resources; for example, less specific lymphocytes can detect a wider variety of pathogens, but they will be less efficient at detecting any specific pathogen; thus, an AIS must not be designed to respond to very specific behaviors; it needs to be more slack in order to improve self-learning and allow the detection of new failure modes.

In order to implement these organizing principles, there are many approaches which can be adopted. The ones most used in literature are Clonal Selection, Negative Selection, Immune Network, and Danger theory. The foundations of these methodologies are summarized in Table 2 and explained in the next section.

Methodology	Overview
Clonal Selection	Multiple cloning of failure detectors with slight modification on the basis of the effectiveness of the detector (the more a detector works, the more clones are created with less modification)
Negative Selection	Random generation of detectors with casual parameters; each detector is tested with the system in good condition and if a false positive is detected, it is discarded.
Immune Network	Failure detection is made using a network of detectors linked together based on the detectors´ affinity; the detectors are cloned and mutated and, according to their level of affinity, the detector can be integrated on

	the network; the mutation process is activated on the basis of the neighboring detector.
Danger Theory	Failure detection is carried out not according to the pattern recognition of a specific fault signature, but considering the potential danger of the element under investigation through specific danger signs.

Table 2 Overview of different AIS methodologies

Clonal Selection [8] finds conceptually its roots on the assumption that only those cells that are able to recognize the antigen are allowed to proliferate and participate in the immune response. The main features of the Clonal Selection Theory are: (i) new cells are copies of their parents (cloning); (ii) each cloned cell is subjected to a mutation mechanism (somatic hyper mutation) that binds, in the opposite way, the cloning rate and the mutation rate to the performance of the detector: the more a detector is performant, the more clones are generated with minor mutations; (iii) elimination of newly differentiated lymphocytes carrying self-reactive receptors; (iv) proliferation and differentiation on contact of mature cells with antigens.

This kind of system is based on the clonal selection and affinity maturation principles. It is similar to mutation-based evolutionary algorithms and has several interesting features, as population size dynamically adjustable, exploitation and exploration of the search space, location of multiple optima, capability of maintaining local optima solutions, and a defined stopping criterion.

Negative Selection Algorithms [6] mimics the immune system's ability to detect unknown antigens while not reacting to self-cells. During the generation of T-cells, the receptors are built through a pseudo-random genetic rearrangement process. After the creation, a censoring process is performed in the thymus (called negative selection) in order to destroy T-cells that react against self-proteins; this implies that only T-cells that do not bind to self-proteins are allowed to leave the thymus. These matured T-cells circulate throughout the body to perform immunological functions and protect the body against foreign antigens. In AIS, this class of algorithms is typically used for classification and pattern recognition problems, where the problem space is modeled in the complement of available knowledge. For example, in the case of an anomaly

detection domain, the algorithm use a set of exemplar pattern detectors trained on normal (non-anomalous) patterns that model in order to detect unseen or anomalous patterns.

Immune Network [41, 50] is founded on the hypothesis that the immune system maintains an idiotypic network of interconnected B cells for antigen recognition. These cells both stimulate and suppress each other in certain ways that lead to the stabilization of the whole network. Two B cells are connected if the affinities they share exceed a certain threshold, and the strength of the connection is directly proportional to the affinity they share. In artificial immune network (AIN) models, a B-cell population is made of two sub-populations: the initial population and the cloned population. The initial set is generated from a subset of raw training data to create the B-cell network. The remainders are used as antigen training items. Antigens are then selected randomly from the training set and presented to the areas of the B-cell network. If the binding is successful, then the B-cell is cloned and mutated. The mutation yields a diverse set of antibodies that can be used in the classification procedure. Once a new B cell is created, an attempt is made to integrate it into the network at the closest B Cells. If the new B cell cannot be integrated, it is removed from the population. If no bind is successful, then a B-cell is generated using the antigen as a template and is then incorporated into the network.

Danger Theory [47, 55] is quite similar to the negative selection approach, since both are related to discrimination between safe and unsafe behaviors (safe and unsafe cells in NS). However, the central idea in the Danger Theory is that, in order to discriminate the state, the immune system does not respond to foreignness by detecting non-self pattern; it rather estimates the potential danger. This theory has been borne out of the observation that there is no need to attack everything that is classified as foreign, but only the dangerous ones. In this theory, danger is measured by damage to cells indicated by distress signals that are sent out when cells die of an unnatural death (cell stress or lytic cell death, as opposed to programmed cell death, or apoptosis). By using a Danger Theory approach in an AIS, the challenge is clearly to define a suitable danger signal, a choice that might prove as critical as the choice of fitness function for an evolutionary algorithm. In addition, the physical distance in the biological system should be

translated into a suitable proxy measure for similarity or causality in artificial immune systems.

Like shown before an AIS is a set of methodology that can be used to implement an algorithm able to solve a specific problem; the different approaches are focused on the behaviour of specific parts of immune system but all the approaches are focused on the resolution of generic problems.

Like the natural immune system, an AIS shows a marked flexibility and a good capability to solve optimization and decision making problems getting a coverage time equal or better respect other approaches.

Due to the high similarity between the detection of a pathogenic agent in a natural immune system and the detection of a failure in a plants, the adoption of AIS to solve maintenance needs appears natural and a lot of research are performed about this scope and was the starting point of this literature review.

Regarding performance, is very hard to make a general evaluation because the performance are strictly dependent about the specific implementations and can vary a lot in different applications but in general, respect to other kind of distributed methodologies like Multi Agents systems AIS shows to have a coverage time comparable, a greater adaptability and cooperation level. This better flexibility has the cost of a resource consumption significantly greater respect other kind of algorithms.

3. Multi Agent System

3.1.Introduction

In recent years there has been a remarkable evolution of programming paradigms, especially in the fields of artificial intelligence and experts system, one of the most famous architecture is the Agent oriented programming.

3.2.Multi Agent System

A multi-agent system (M.A.S.) is a computerized system composed of multiple interacting intelligent agents within an environment. Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent or a monolithic system to solve. The main reason for this is that the capacity of an intelligent agent is limited by its knowledge, its computing resources, and its perspective. This bounded rationality (Simon 1957) is one of the underlying reasons for creating problem-solving organizations.

3.3.Key Characteristics

in order to better understand this paradigm is useful identify the key characteristics of agent and multiagent system. There characteristics can be grouped in five different set, Intrinsic, extrinsic, system, framework and environment.

3.3.1. Intrinsic Characteristics

The intrinsic characteristics regroup all the aspect strictly related to the inner behaviour and setup. These characteristics are specific to each agent

Lifespan
Level of Cognition
Construction
Mobility
Adaptability
Modeling

Table 3 Intrinsic Characteristics of MAS

Lifespan

Lifespan indicates the expected life of each agents, depending by the applications an agent can have a life from a single shot, where an agent are created only to perform a single task and after that are immediately destroyed (transient Agent) to a very long life (long term agents) where the agents remain in running on the system in order to perform continuous operation like a monitoring agents.

Level of Cognition

Level of cognition defines the inner mode of operation of the agent, there are two different way : reactive or deliberative.

A reactive agent is an agent that operates only on the basis of external inputs, the actions taken can be selected using a set of condition-action rules or using a finite state machine or a fuzzy approach.

A declarative agents instead, possesses an explicitly represented, symbolic model of the world, and in which decisions (for example about what actions to perform) are made via symbolic reasoning.

This very different approaches lead to very different behaviour of the agents and need to be carefully chosen in order to obtain the desired behaviours.

usually a reactive agent is more computationally efficient, consumes less resources and is able to react faster to an event. to the other side a deliberative agent can make correlations between the inputs and the state of the system, achieving better performance in some cases.

Construction

Construction indicates the programming paradigm used to implement the agent, there are two different approaches usually used in agent's programming: procedural or declarative.

in particular the procedural approach means that the programmer must specify exactly how a computation should proceed, step by step on the other side, in a declarative approach, the programmer must specify what the agent should do without giving too much detail of how the function needs to be implemented at low level.

Mobility

mobility expresses the capability of the agent to migrate in another systems in order to perform the own action in a different environment.

an agent with these capability is defined as itinerant agent, an agent without this is defined Stationary. A static agent exist as a single process or thread instead, an itinerant one is capable to pick up and move their code and data to a new host where can continue the execution.

The value of the mobility depends by the rule of the agents, for some application an itinerant agents it is invaluable, instead other kind of agents don't necessitate of this characteristics.

Adaptability

Adaptability expresses the capability of the agent to change its behaviour in order to react to a change in the environment, this feature enable the evolution of the system and is of paramount importance in case of expert system or artificial intelligence system. anyway this characteristics is not suitable for all kind of agents, and exists three different level of adaptability: fixed, teachable and autodidactic

a fixed agent has the it's own behaviour hardcoded inside it's functions without parameterization or other techniques to modify it.

a teachable agent, instead, has some functions able to change the operation way of this components; this is the typical way to implement a decision system based on artificial neural network in which there is a specific training function that, using a training dataset, is able to tune the parameter of the network.

the last type of agent, the autodidactic , is the agent with more adaptability. this kind of components are able to use the data, usually used to perform the required task, also for training itself in automatic way.

Modeling

Modeling is the last feature that characterize the inner behaviour of an agent: it express the target of the agent: an agent can model the environment the environment, in order to react to external changes, can modelling itself, in case of autonomous agent or can modelling other agents in case of communication agent.

3.3.2. Extrinsic Characteristics

Extrinsic Characteristics refers to feature related to the behaviour of the agents respect to the environment and the interaction with other agents, these characteristics are the following

Locality
Social Autonomy
Sociability
Friendliness
Interactions

Table 4 Extrinsic Characteristics of MAS

Locality

Locality express the position of the target of an agent: an agent can be local or remote

A local agent operate on the same system in witch it is executed, instead a remote agent operate to an external system through the use of a communication network.

Social Autonomy

The social autonomy express the degree of decisional autonomy that the agent has respect the other ones.

An agent can be independent, when its behaviour is not under the control of another agent. otherwise the agent is called controlled.

Sociability

Sociability express the awareness of an agent respects the other ones that are currently running on the system and the degree of social interaction between them, this feature can assume several different values:

- Autistic: if the agent has no knowledge about other components and executes their own task without any care about the behaviour of other agents
- Aware: if the agent has awareness about the behaviour of the other agents but don't take any action to cooperate with them
- Responsible: if the agent has awareness about other agents and operate in order to coordinate and control the function of other agents
- Team Player: if the agents has full knowledge about the behaviour of other agents and cooperate with the other agents in order to perform the overall task

Friendliness

Friendliness express the different modality of interaction between different agents. There are three different modality of interaction: cooperative, competitive or antagonistic.

In a cooperative behaviour the different agents have the same objectives and collaborate one each other to perform them, is the typical configuration that is usually used for parallel computational tools.

Another behaviour is the competitive where the different agents fight in order to obtain a limited resource. This is the typical way how the auction based system works in order to solve optimization problems.

The last behaviour is the antagonistic where an agent is able to limit the functionality or destroy another agents in order to solve the task a using a mechanism similar to the Darwinian evolution.

Interactions

Interaction express all the low level characteristic of the communication between agents it can further split in three different feature

- Logistics: that express if the interaction occurs directly between the agents or instead by using a third-part in order to facilitate the message exchange
- Quality : that express if the iteration can be with a single agent or with the entire system under control
- Semantic Level: that express if the communication are procedural or declarative

3.3.3. System Characteristics

The System characteristic group collects all the attributed related to the multiagent system that are independent of the characteristic of the agents that constitute them. These characteristics are the following

Uniqueness
Granularity
Control Structure
Interface Autonomy
Execution Autonomy

Table 5 System Characteristics of MAS

Uniqueness

Uniqueness is a parameter that express the type of agents that constitute a multiagent system; if the system is composed only by agent of the same kind the system are defined homogeneous, otherwise if the system is composed by agent of different type the system are called heterogeneous.

Due to the nature of this kind of system, an homogeneous multi agent system is very rare and limited only to simple application, for a more complex system the adoption of an heterogeneous architecture is mandatory in order to provide the necessary flexibility to the system.

Granularity

Granularity express the detail of the control action implemented in the system; a system can have a fine granularity if the agents that compose the MAS . are able to control at low level all the aspect of the behaviour of the system. Instead , a coarse grained system are composed by agents that controls only the most important aspect of the system, leaving uncontrolled the less important things.

Is easy to understand that the adoption of a fine granularity allows to obtain the better performance but the adoption of these methodology requires the use of a very high number of agents with a strong impact on the resource needed to the system.

Control Structure

Control source defines the control mechanism that is used to manage and synchronize the task of different agents there are two different strategy that can be adopted:

- Hierarchy: in this configuration is defined a priori a hierarchy between the different agents with some agents that have more privileges respect other one and have the capabilities to control and force the execution of other agents
- Democracy: in a democracy approach, instead, the control structure is less rigid and an agent are able to control other agents, or not, depending on the condition of the system:

Is easy to understand that an hierarchy approach is more easy to implement and allow a better estimation of the behaviour of the entire system, conversely the use of a democracy control structure allows the creation of a more flexible system but it involves

some limitation like a less predictability of the conduct of the system, with the needs of an extended adoption of simulation techniques in order to validate the system.
Another limitation of a democracy approach concern to the reactivity of the system, usually a democratic system is slower to respond to an external input due to the time required to vote and define the control chain.

Interface Autonomy

Interface autonomy express the autonomy of the internal design of an agent respect the overall architecture; this feature, in particular, defines which mechanism, at low level are used to exchange information between the different agents.
There are several techniques that can be used for that, the most rigid is the adoption of shared environment variables that severely restricts the inner architecture of all the agents that must use the same variable with the same name and the same type in all the agents.
A more flexible approach, instead , is the use of Application Program Interfaces (APIs) that each agents exposes in order to abstract and hide the internal variable using only a standardized interface to communicate.
The strategy with the most interface autonomy is the use of an ontology in order to define the meaning and the message format freeing the agents to manage these messages in autonomous way.

Execution Autonomy

Execution autonomy correspond to the freedom that an agent has while execution in an environment.
For example a personal assistants can be constrained to behave, in a helpful manner, they must always speak the truth, believe and help the user. In other kind of system like agent that represent different interest, instead, some execution autonomy is generally required even the interface autonomy is still required.
Execution autonomy is crucial for the openness of the system and usually some grade of autonomy must be provided in order to take advantage of the use of multi agent system. Restriction in this autonomy may still be imposed by specific system for specific role.

3.3.4. Framework characteristics

This set includes a set of features related to the agent execution environment, the framework that enable the launch, control and the message exchange between different agents of the system

Design Autonomy
Communication Infrastructure
Directory Service
Message Protocol
Mediation Services
Security Service
Remittance Services
Operations Support

Table 6 Framework Characteristics of MAS

Design Autonomy

Design autonomy deals with how the agents are constructed. it is the autonomy of the designers. Design autonomy is crucial if the system need to be truly open in that other programmer should be able to contribute with their agents to the system while having to satisfy as few requirement as possible. Thus, design autonomy correspond to heterogeneity of the agents. It is orthogonal to control autonomy, because agents of a fixed design might choose their own actions, while agents of different design might be controlled externally.

Different Agent framework come with different requirements that impinge upon design autonomy; for example some framework require that all agents will be built using a specific language. Some approaches, usually used for agent communication require instead that the agent must necessarily represent beliefs; they must be able to perform logical inference and plan with a rational approach.

Communication Infrastructure

Communication infrastructure bring the same concept of interface autonomy applied in the framework context. It express the different approaches that the framework implement in order to enable the communication between agents.

This feature specify if the framework allows the use of a blackboard approach, based on the use of shared variables, or if use a message based protocol that implies the use of an ontology approach.

It also specify if the system use a connection or connectionless communication approach with synchronous or asynchronous communication protocol.

Another aspect that is defined in this characteristics is the target of the messages. A network communication can have three different schema and a framework can implement one or more of this

- Unicast: that implies the use of point to point communication between two different agents
- Multicast: that implies that the framework allows a group communication between an agent and a set of recipient
- Broadcast: that implies that all the messages are received by all the agents currently running on the system.

Directory Service

Directory service defines the discovery protocol implemented in the framework; this function is an essential part of the framework, it provides to an agent the capability to discover which other agent are currently running on the system and what services provide.

This system are mainly composed by two different subsystem called white and yellow page services

The white page service is the part of the system that keep track of the number and the type of the agent that are currently running on the system. This system provide also, for each agent an unique identification ID that act like destination address in order to enable to agents to communicate.

On the other side the yellow page service allows agents to publish one or more services they provide so that other agents can find and successively exploit them. Agents may register their services with this service or query the yellow page to find out which services are offered by other agents.

Message Protocol

Message protocol defines the communication protocols, supported by the framework, that are used to provide communication features to the agents.

There are several different protocols that can be used for this scope which arise from different sectors like standard internet communication protocols (HTTP, HTML, XML) or industrial communication protocol like OLE, OPC or CORBA.

Mediation Services

Mediation services express all the feature, implemented in the framework in order to solve conflict between the agents. In fact sometimes agents have conflicting goals or are simply self-interested.

The objective of the protocols is to maximize the payoffs (utilities) of the agents. In cases where the agents have similar goals or common problems, as in distributed problem solving (DPS), the objective of thisservice is to maintain globally coherent performance of the agents Without violating autonomy, ie, without explicit global control. For the latter cases, important aspects include how to

- determine shared goal
- determine common tasks
- avoid unnecessary conflicts
- pool knowledge and evidence

there are several mechanisms that can be used to perform this feature like the use of dedicated messages, specified through an ontology, or the use of transactional system to revert the operation of an agent and resolve a possible conflict in this way.

Security Service

Security Service express all the feature, implemented in the framework, regarding security.

There are two main aspects to security involving mobile agents. The first, and most commonly considered, is protection of the system against intentionally or accidentally malicious agents. The second is protection of a mobile agent against malicious servers or agents.

The former aspect has been dealt with extensively in the context of operating systems, which establish and maintain protection levels for process execution. Security in the latter aspect cannot be guaranteed, be- cause in order for the mobile agent's code to be executed, the agent has to expose both its code and data to the server. A detection, but

not prevention mechanism, is to have the agent return itself with its data, to verify that it has not been altered.

Authentication, integrity, confidentiality, and nonrepudiation are other important aspects of security. Authentication validates the identity of the person or agent with whom other agents are interacting.

Integrity ensures that What an agents see has not been tampered with, confidentiality ensures that what the designer intend to be private remains so; and nonrepudiation means that the agents are liable and cannot change your behaviour in an unexpected way.

Remittance Services

Remittance Service express the functionalities, implemented in the framework to model a "virtual currency model" in the system

This kind of system allows the implementation of auction based interaction model between the agents and other similar model to implement, in an efficient way, optimization process.

Operations Support

This characteristics included all the auxiliary features of the framework that are not strictly related to the functionalities of the agents but necessary to obtain a reliable system in particular some of these aspect are

- Archiving: that include all the features related to backup and save the data used by the system
- Redundancy: that include all the features that are necessary to preserve the functionality of the system even in case of failure
- Restoration: that include all the feature that are necessary to restore the system operation after a shutdown both intentional or caused by a catastrophic failure
- Accounting: that include all the feature related to the authentication and control of the operator which must be able to intervene on the system for updates or maintenance

3.3.5. Environment Characteristics

This set include a set of features able to characterize the environment where the multi agent system works

The main characteristics are the following

Knowable
Predictable
Controllable
Historical
Teleological
Real time

Table 7 Environment Characteristics of MAS

Knowable

Knowable express how the rules that governing the environment under the control of the multiagent system are known.

A knowable environment allows the implementation of white or grey box approach into the inner behaviour of the agents in order to implement an optimal control strategy.

At the other side if the environment's behaviour is not known only a black box approach can be used.

Predictable

Predictable express how the future behaviour of the environment can be forecast on the basis of the actual state.

This kind of feature make influence on the time horizon of the control that the mas can make on the system.

Controllable

Controllable express the ease with an agent can act some change in the behaviour of the environment a very controllable environment require usually a soft actuation; on the other side if the environment is very hard to control a strong actuation need usually to be used.

Historical

Historical express the capability to forecast the future behaviour of the environment by analysing the previous series of historical data.

This feature sets the basis for the development of a predictive system; if an environment is not historical is very difficult implement a predictive system so usually a reactive system are implemented.

Teleological

teleological express the main goal of multi agent system;

a MAS is teleological if has a well defined and global purpose or goal which it pursues through the cooperation of each agents.

Real time

Real time is a characteristics related to the dynamic of the environment. a MAS is real time if the time of change of the environment is comparable to the time required to the agents to deliberate; on the other side if the agents are much faster respect the environment, and if possible to assume that the environment is stationary during the deliberation, the system is defined as not real time.

3.4.FIPA Standard

The Foundation for Intelligent Physical Agents (FIPA) Standard is a body for developing and setting computer software standards for heterogeneous and interacting agents and agent-based systems.

FIPA was founded as a Swiss not-for-profit organization in 1996 with the ambitious goal of defining a full set of standards for both implementing systems within which agents could execute (agent platforms) and specifying how agents themselves should communicate and interoperate in a standard way.

In this section a little overview of the standard is provided in order to better understand the capability of JADE, a framework that implement FIPA.

3.5.JADE Framework

Jade, acronyms for Java Agent DEvelopment Framework is one of the most famous framework that can be used to implement a FIPA compliant multi agent system in java.

JADE system supports coordination between several agents FIPA and provides a standard implementation of the communication language FIPA-ACL, which facilitates the communication between agents and allows the services detection of the system.

JADE was originally developed by Telecom Italia in 1998 and nowadays is distributed as free software.

3.6. Multi Agent System Application Overview

In this section an overview of some examples of application of a MAS based methodology for maintenance purpose is provided. However, it must be kept in mind that this is a cutting edge approach. Thus, most of these studies are still under a prototypical phase or related to a very specific application.

3.5.1. IMS-TEMIIS

IMS-TEMIIS has been developed as a demonstration prototype of an integrated platform for remote and advance maintenance strategy (Iung 2003). This platform simulates a typical chemical plant with several pumps and valves controlled by a heterogeneous set of control systems on several different fieldbus.

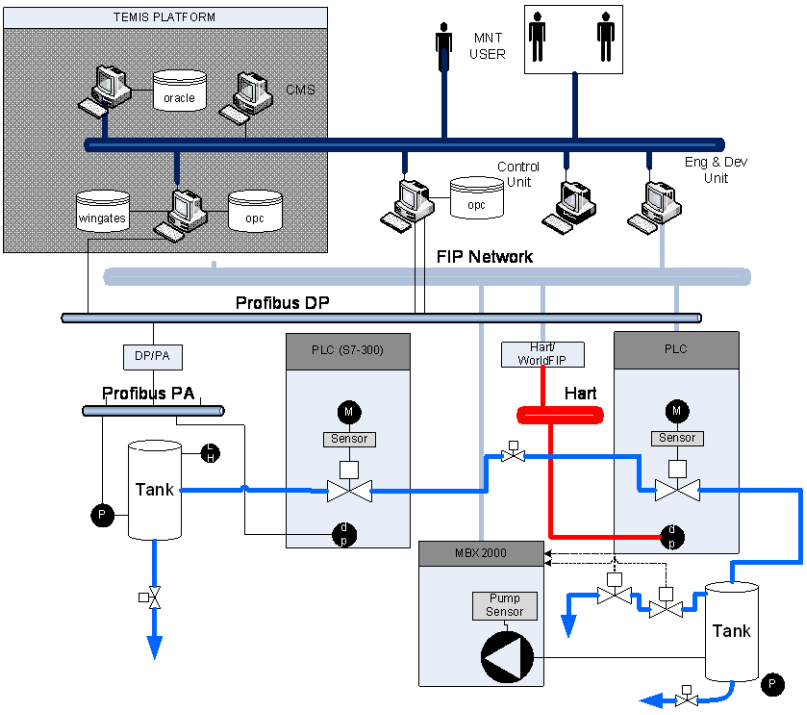

Figure 1 IMS-TEMIIS platform schema

Inside this platform a MAS based diagnostic system has been implemented. This system is composed by a limited set agents, as shown in Figure 2, which are loosely intercoupled.

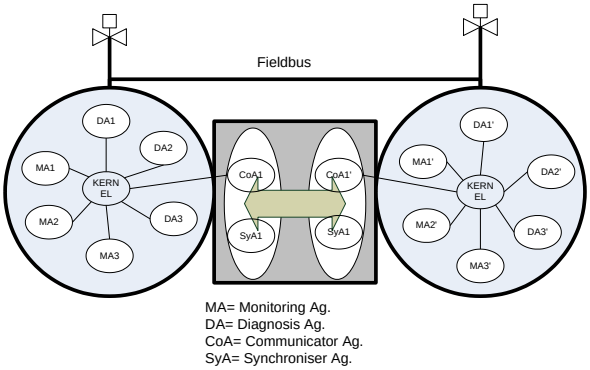

Figure 2 – The role of the single agents inside the IMS- TEMIIS platform

In this application, all the main functionalities, like diagnostic and monitoring, are implemented by specific agents that perform autonomously all the necessary operations. When an agent detects a failure, a more complex and cooperative decision process is activated, through the use of communicator and synchroniser agents, in order to share the knowledge between diagnostic agents and isolate the degraded component of the system.

This auxiliary agent also aims to ensure a reliable and affordable communication between different machines. The exchange of messages, required for this process, are managed by a custom middleware, named kernel, that abstracts the maintenance system from the hardware and allows the integration of different devices and communication protocol.

3.5.2. Intelligent Maintenance System for microsatellite

This application is a good example of a multi-agent architecture that is intended to replace human operators responsible for system maintenance (Sierra et al. 2004). The particularity of this application is the extreme complexity of the system under control combined with the impossibility to perform a maintenance intervention without incurring in very expensive space missions. The structure of the automation system is based upon collaborative agents designed to detect failures in any of the microsatellites components. The multi-agent system consists of a set of different agents devoted to failure detection, prevention and correction. Regarding correction, specific agents for each constitutive part of the microsatellite have been developed in order to take over the

necessary actions to solve any given problem in its operation. The detection agent decides which correction agent should take the control of the system, based upon the inference obtained from its knowledge base made up of rules for testing and diagnosis. Actions or corrections may imply the use of redundant systems, which can reconfigure themselves to avoid defective circuits. The prevention agent uses predictive models that have been developed for each significant failure mode. Statistical models are also used by this agent to determine the shape of the distribution of times to failure. The prevention agent selects the corresponding correction agent to which control is going to be transferred. This agent carries out the necessary actions to prevent the system failure. The overall intelligent system implements a blackboard architecture for communication and collaboration among agents.

3.5.3. Intelligent Maintenance System for wind farms

Another relevant example about the use of multi-agent systems for maintenance purpose is related to the development of a maintenance platform for wind farm and related power grid (Trappey et al. 2011).

In this application, it is particularly relevant the way the maintenance planner has been implemented. This is one of the implementations of a market-based approach where the entire protocol for the management of the auction has been realised with a set of standard messages to share the needs and the capabilities of each agent.

Three different steps for maintenance decision making have been implemented:

- strategic: the preparation and recovery stage (long term problem) considers the organizational and financial impacts of the decisions (evaluated according to the repair cost, the cost of preventive maintenance and the cost due to possible downtime);
- tactical: medium-term problems related to prediction and prevention of a failure;
- operational: short term problem related to failure detection and response, responsible for the management of corrective maintenance and related management and negotiation.

For implementing this complex system, a large set of different agents is required: a group of agents for data extraction, the monitoring agents (MA) and asset agents (AA)

that represent a single device of the system. Another group of agents is related to the failure detection, including the diagnostic Agents (DA) and the prognostic Agents (PA). The last group of agents is related to the functions of maintenance scheduler, composed by a couple of agents, Maintenance Decision Support Agents (MDSA) and the System Provider Maintenance Agent that manages all the tasks needed for the interventions planning. They interact with the Human Resource Agent (HRA) and Spare Part Agent (SPA) responsible for the management of personnel and spare parts.

All these agents give rise to a very complex interaction system, based on a set of different request–response protocols, in order to implement the overall system.

3.5.4. Maintenance Schedule for a bus fleet

This implementation relates to the maintenance management of a bus fleet (Zhou et al. 2004). In this application only maintenance scheduling operations have been implemented. Fault isolation and identification is not considered and consequently the system does not carry out any prognostic functionality. The architecture of this implementation is shown in the following figure

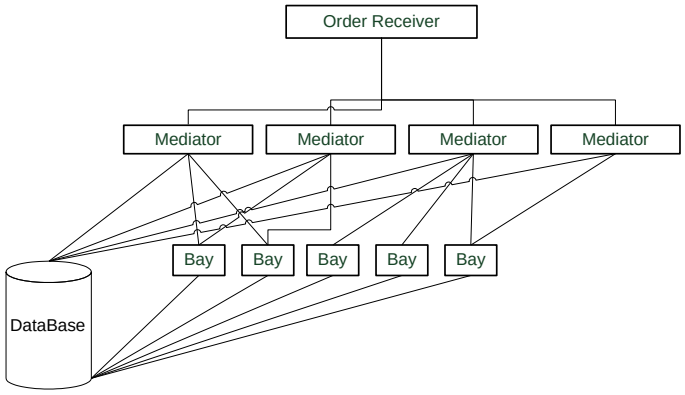

Figure 3 The MAS architecture in the bus fleet application

This approach is centralized because there is only a single agent responsible for the management of the request of maintenance operation. This agent, linked with a bus status tracking system and a bus service planning system (a couple of external systems responsible of the health assessment of the buses and of the planning activities of the vehicles) receives all the maintenance orders, rank them according to their priority and retains a global knowledge about the status of the overall system. It is linked to several mediator agents, each of them responsible of a specific maintenance activity. Each

mediator agent keeps a list of bay agents which are capable of the maintenance type that the mediator agent manages and selects for performing the task. Each bay agent is part of the system that manages and performs the physical maintenance operation. In order to keep in consideration the status of the several bay and maintenance orders, a database agent is responsible to store the actual state of the overall system in order to help mediator agents to choose the best bay to perform the task. The use of database agents allows individual agents to be aware of the overall status of the system in order to find the optimum solution. This choice was performed in order to minimize the impact of maintenance operations to the functionality of the transport system, and secondly in order to maximize the uptime of the maintenance bay.

4. Literature Review

In this chapter, a literature review on the use of AIS for industrial maintenance applications has been performed using web-based search engines in IEEE Explorer, Springer Link, Elsevier Journal Finder and Google Scholar. The reference sections of the most cited papers were also used to improve the search. The keywords used were "artificial immune systems diagnostic", "artificial immune systems prognostic", "artificial immune systems industrial", "artificial immune systems maintenance", "artificial immune systems self-learning", and "artificial immune systems self-healing". The literature review included 323 papers of which only 110 were eligible for the inclusion in the analysis consistently with the relevance to the topic of maintenance related industrial applications of AIS. This preliminary analysis was carried out with the autonomous judgment of two of the researchers involved in the study.

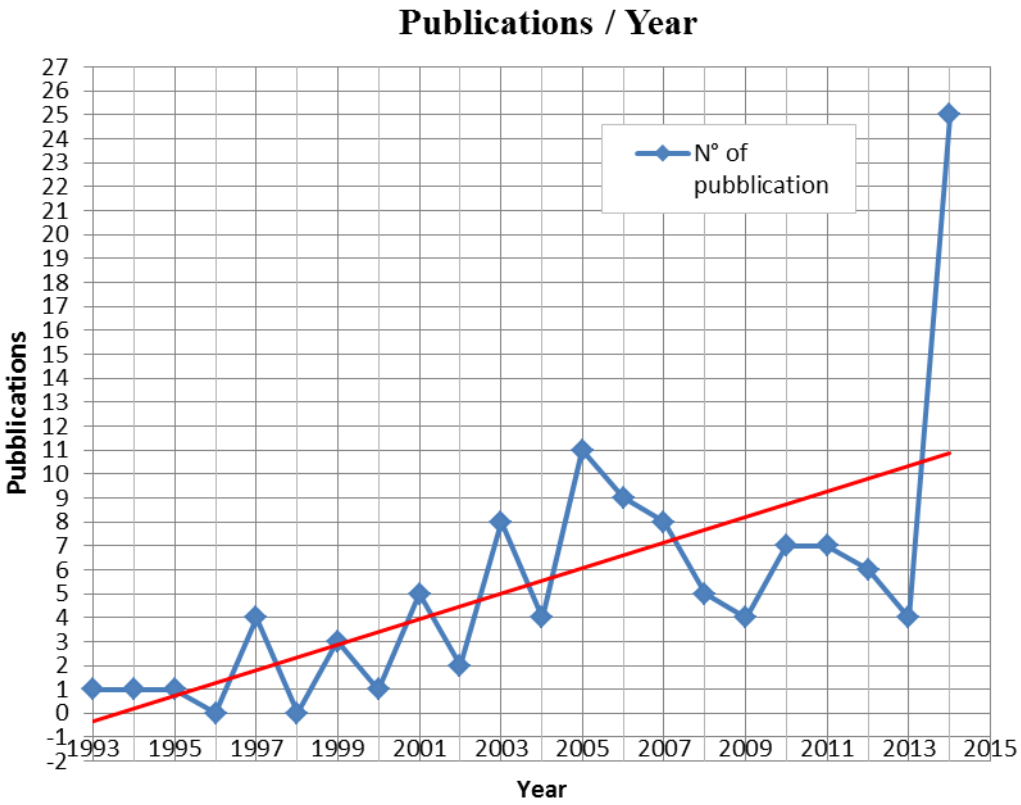

Figure 4 Temporal distribution of the literature review

The first work that proposed AIS as a tool for process diagnosis, "An Immune Network Model and its Application to Process Diagnosis" was presented by Ishida in 1993 [6], defining the timeline starting point for the paper search. Since then, the number of publications related to maintenance application of AIS has been growing, despite its irregular trend line. In particular, in 2014 the number of papers shows a very interesting growth which demonstrates the increasing interest in these topics.

The following table reports the methodology used for the review: each paper has been analysed according to seven different characteristics; for each of them a set of subcategories was defined and all the papers were evaluated with a score between 0 and 5 in each subcategory. This analysis was conducted in an independent way by the two researchers and the final score is the mean of the evaluation of each researcher. The next section presents the results, grouped for each classification criteria.

Application Domain	Application Targets	Algorithms	Implementation	Coverage Area	Self-Capabilities	Maturity degree
Fault detection	Electronics	Clonal Selection	OO programing	Part	Self-learning	Conceptual
Diagnosis	Mechanical	Negative Selection	Multi-agent systems (generic)	Equipment	Self-healing	Simulation
Prognosis	Electric Motors	Danger theory	Multi-agent systems framework	Plant		Prototype
Optimization	Software	Artificial Immune Network	VHDL (VHSIC Hardware Description Language)			Field application
Other	Smart Grid	Dendritic cell				
	Other	New proposal				

Table 8 Classification Criteria and Categories

4.1.Application Domain

The first analysis regards the application domain in order to identify the fields where the use of AIS are nowadays under study and where researchers expect a good result. Figure 5 shows how the application domain is distributed.

The most common use in industrial application is fault detection and diagnosis, a natural application to the pattern recognition and anomaly detection features of the AIS. Diagnosis is closely related to fault detection and some works use these two terms in the same sense. In particular, regarding fault detection, AIS are used mainly for weak signal analysis in order to perform the detection of a failure from the data provided by a set of sensors. On the other side, for fault diagnosis, AIS are commonly used for clusterization and to implement classifier algorithms.

Several studies apply the AIS to both tasks, like fault classification for rotating machines presented by Tang [14] and by Strackeljan [15]. Other works use AIS to fault detection, in combination with other techniques to perform the diagnosis, as the proposal of Amaral [16] that uses a negative selection algorithm to detect the fault and a quad tree space partition to classify it. There are also works focused on fault detection, like the hardware immune system proposed by Bradley to perform error detection for reliability measurement [17] or the new AIS proposed by Laurentys [18], based on Natural Killer (NK) immune cells. Prognostic methodologies using AIS were proposed by Hu and Qin [19] to forecast the health of electronic equipment in a vessel. Thumati and Halligam have used AIS, associated with a nonlinear observer, to predict faults in nonlinear discrete-time systems [20][21].

Another application domain for AIS is the optimization problem: a remarkable application in this field is showed by Bhuvaneswari and Ramachandran [22] [23], using AIS to implement a distributed control for a microgrid generation based on energy auction. An unusual application found in the survey is in mine detection: Sathyanath and Sahin [24] introduce an Artificial Immune System based on an Intelligent Multi Agent Model (AISIMAM) in order to optimize the control strategies for mobile robots dedicated to demining. In the ICT field, AIS have been used for fault detection and diagnosis [25][26][27] of an electronic system and for optimization [28][29]. Another common application is to implement an intrusion detection system [30][31].

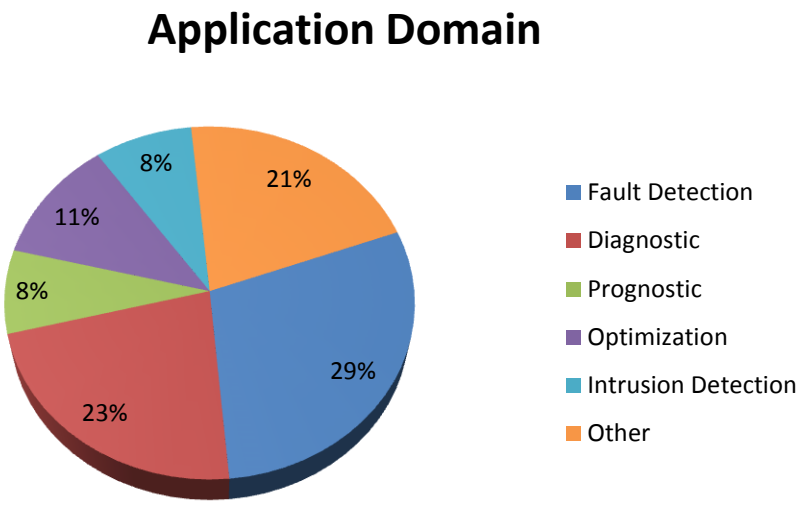

Figure 5 Application Domains distribution

4.2. Application Targets

The second performed analysis is about the system target. Various applications have been clustered in the following six categories: *Electronic Equipment*, *Mechanical Component*, *Electric equipment* (including motors), *Software*, *Smart grid* and *Other* (which collects targets not easily classifiable).

Software systems are the most frequently application target described in literature, mainly due, as shown before, to the early works in this field; in fact, almost all of the publications between 1993 and 2002 are related to theoretical approaches or applications in software system. Over the years, the methodology and the technology required to apply this methodology to physical systems became mature and the focus of many technical papers moved from software to mechanical and electrical systems. Figure 6 presents the distribution found in this survey.

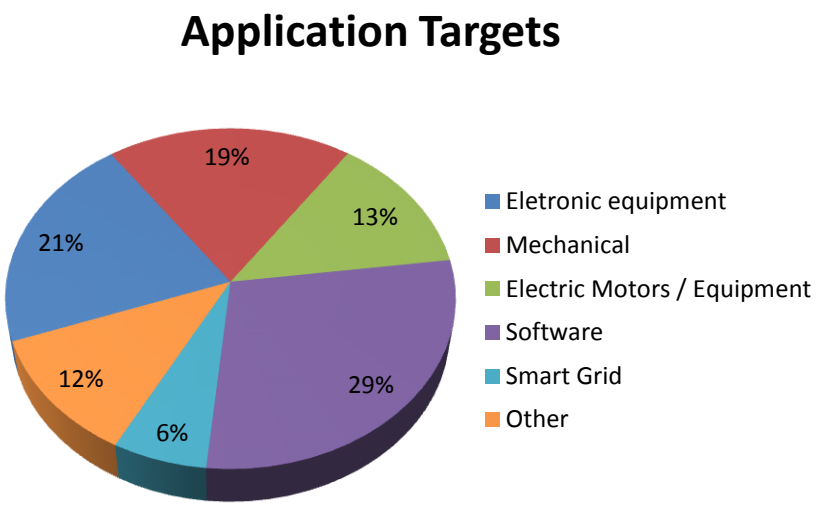

Figure 6 Application Target

Regarding the application target, remarkable applications founded in literature are the following. A diagnosis of induction motors is the main case study in the work of Costa Branco *et al* [32]. Laurentys *et al* and Amaral *et al* detect faults in analogic electronic circuits with an AIS that works on the impulsive response of the circuits [16] and wavelet signatures [33] for robotics application. Caham *et al* use AIS to detect error in twin robots controllers [34].

Mechanical applications of AIS are commonly studied for fault diagnosis. Halligan, Thumati *et al* [20][21] have applied AIS in axial piston pump, Strackeljan [15] in rotating machines. Another use is for a gas lift system, as described by Araujo *et a*l[35].

The most recent application targets are electrical energy systems and smart grids, a typical distributed application that fits well with the natural distributed nature of the AIS. An application that greatly shows the capability of an AIS system is the work of Xu *et al* [36] that proposes an AIS to identify causes of power distribution outage in electrical distribution system, that has a performance 163% better than a neural network method applied before. In the same field, Bhuvaneswari *et a*l [22][23] have implemented AIS as an auction system to manage and control a microgrid generation system.

4.3. Algorithms

The third analysis regards the different algorithms adopted in an AIS implementation. Since there are several different aspects of an immune system that can be imitated in order to implement an AIS, the analysis shows a good distribution of the work on the main methodologies; the results are resumed in Figure 7. The different approaches are related to the biologic behaviour that has inspired it. The two most applied algorithms, derived from consolidated immune response theories, are clonal selection (24%) and negative selection (23%) algorithms. A strong similarity with well-known genetic algorithms also justifies these choices.

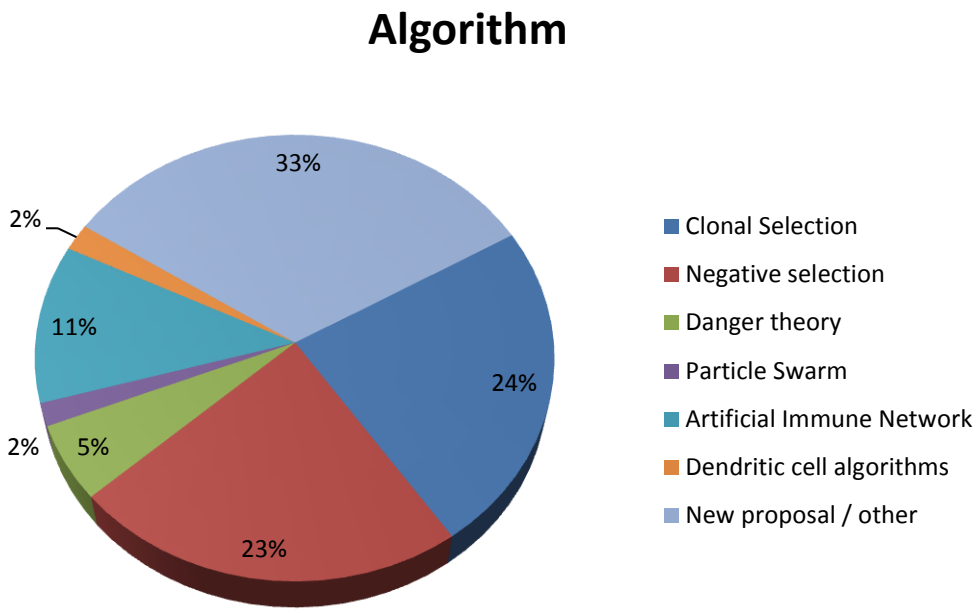

Figure 7 Algorithms used in AIS

Despite this, it is interesting to note that the largest category is that one concerning the proposal of new methodologies. Wang et al. have improved the performance of a Negative Selection algorithm by applying a semantic approach as an affinity function instead of, as usually used, Euclidian Distance.[37]. Adaptive approach is also used by Aydin et al. to improve the coverage and to minimize the set of detectors generated by clonal selection. This application aims to implement fault detection in induction electric motors by analyzing the energy consumption through the use of Hilbert transform [38] as next step of the previous work.

Another methodology is based on the use of a wavelet support vector machine to diagnose faults in a gearbox and has its parameters optimized with the support of an immune genetic algorithm, based on clonal selection [39]. A similar approach is used by Aydin et al. in [40], where an AIS system is used to train and optimize a support vector machine able to discriminate failures in inductive motors. Another typical methodology for fault detection is the combination between AIS and nonlinear estimators such as Fast Fuzzy Neural Network. Taniguchi et al. [41] used this approach to detect sensor faults in a UPS (uninterruptible power supply) control system. In the work of Jaradat et al. [42] a combination between sensor fusion techniques and negative selection is used for feature extraction and diagnostic. A different approach comes from the studies of Thumati et al. [21] [20] where, in order to predict a failure in an axial piston pump a nonlinear fault observer, trained by an AIS, is used.

4.4. Implementation

The analysis has shown that the majority of the works are on early stage and focus only on the design phase whilst implementation aspects are not currently addressed. They focus on the design of the model, on the operating rules and on the system architecture in general.

Only few articles describe details of implementation of AIS (Figure 8). The most used way is the development of a specific software by using classical object oriented programming. The analysis also shows a wide adoption of Multi Agent Systems (MAS) models to implement complex AIS system like in the works of Sathyanath and Sahin in [24], Ramachandran et al. in [23] and [22], Rodin et al. in [43], Ishida[44], Hua et al. [45] .

In order to reach a better standardization, some works implement AIS using standard MAS programming framework like JADE, a FIPA compliant Java based framework as shown by Sathyanath and Sahin [22] and Hua et al. [45]. Bradley and Tyrell proposed in [17] an architecture for the implementation of AIS on a hardware platform and, in [46], one application for this architecture is described. In both papers, the AIS is implemented in VHDL in order to be executed by an FPGA (Field Programmable Gate Array).

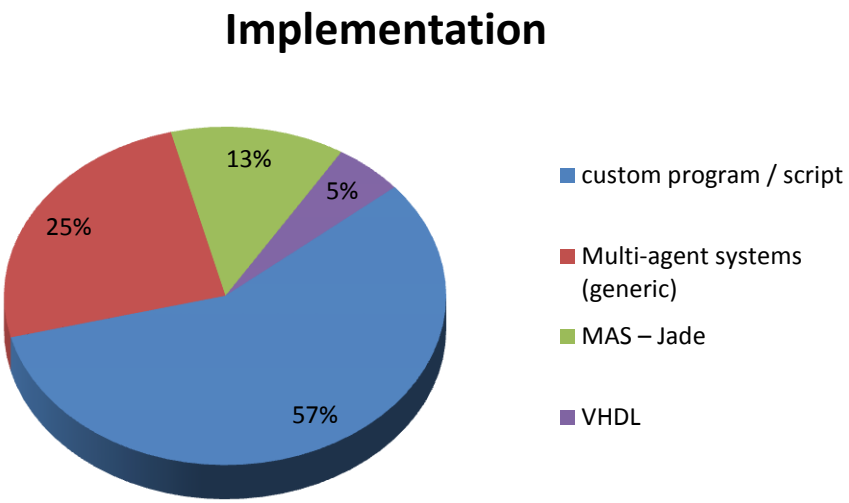

Figure 8 Technologies used in implementation

4.5.Coverage Area

In order to better understand the complexity of different implementations, an analysis of the coverage has been performed distinguishing between theoretical works, single parts of a machine, single machine of a plant or the entire plant. Theoretical works without any specific object of application dominate the coverage application area, as shown in Figure 9.

Looking at the papers that focus on specific applications, there is a fair distribution between works at the level of part, equipment and plant. Scalability feature is highlighted in this context, since AIS can be applied with low constraints regarding the size of the system or its complexity strategies that could cover all areas [47], [48],[40], single equipment or parts [21],[20],[49],[19],[50][41].

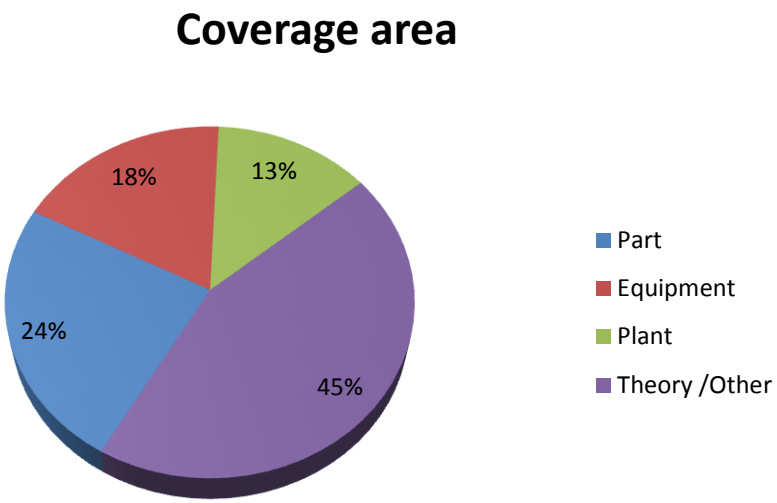

Figure 9 Coverage area

4.6. Self-Capabilities

Technologies inspired by biological systems try to reproduce important features of these systems. Considering the area of maintenance and diagnostics, features such as self-repair or self-healing are very desirable because they aim to improve the system reliability. Self-learning is also an important feature, since many of the algorithms employed are based on artificial intelligence techniques and must be trained to perform their goals. Few papers explore these advanced functionalities that could be implemented with AIS. Self-learning approaches is almost twice more used than self-healing, due to the natural difficulty to replace or repair a damaged part in automatic way.

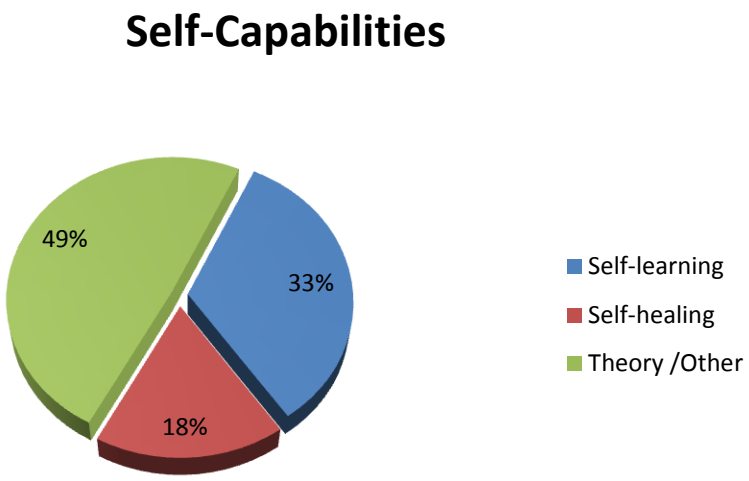

Figure 10 - Self-capabilities

Self-Healing are currently a feasible technique, but applicable only in a limited set of real applications like software or electronic systems with the availability of spare parts ready to use (e.g. like Hard Disk or Flash Memory) or systems based on reconfigurable hardware (like FPGA) A remarkable application is the work of Canham and Tyreell that proposed a hardware-based multilayered AIS that acts with an embryonic array in order to deploy a hardware fault tolerant system. Embryonic array is a homogeneous array of logic units, called cells. Each cell of the array can be configured in order to replace different components of the system, acting like staminal cells in human body. The embrionic array is designed in order to be implemented in ASICs (Application Specific Integrated Circuits), reducing the cost. For the experimental test, a commercial FPGA has been used [51].

AI algorithms must have an initial learning or training phase to build the detectors set or to optimize detectors parameters. Some papers extend this phase and keep learning

during the operational stage. This feature is used to define the self-learning capabilities. Such approaches are found in the works of Hongbing *et al.* [52] and Hua *et al.* [45].

4.7.Maturity Degree

In order to evaluate how much the solution is market ready, the works presented into the different papers have been classified into four categories: *conceptual*, when the proposal is just outlined; *simulation*, when a computational tool is developed/used to evaluate the proposal; *prototype*, when it involves the development of a test bench. The last category is *field application*, when AIS are used in a real application.

Figure 11 summarizes this classification. Conceptual proposals are the majority, followed by simulations. A few works reach the prototype level and no work presents so far a real field application. This is mainly due to the fact applications of AIS in the industrial world are quite recent, and this survey reflects it. Many works are only theoretical; the majority of the remaining papers go more deeply into the application with a simulative approach in order to evaluate the theory and draw the limits and efficiency of the proposal strategy. Prototypes could be seen in the papers [20],[28],[40], [14],[53] and [39].

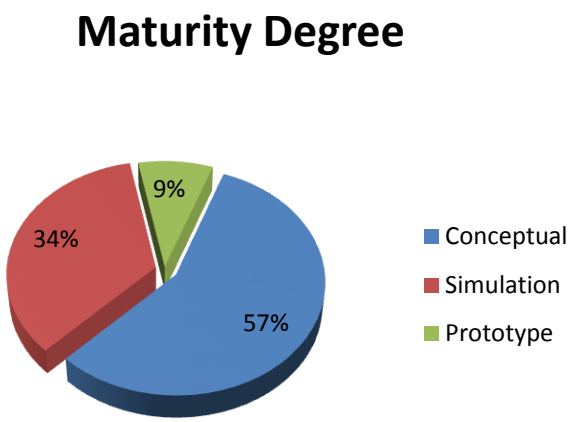

Figure 11 Level of maturity

4.8.Consideration about the literature review

Applications of AIS in the maintenance field explores the robust pattern detection feature to perform mainly fault detection and diagnosis, The analysis shows that AIS are very common in literature in order to develop new advanced maintenance systems. This is mainly due to the performance and robustness that these methodologies reach in pattern recognition and other operations commonly used for prognostic and diagnostic.

Several papers compare AIS to other solutions. They show that AIS based fault detectors have the same or a better performance than other traditional methods [18][47] [54][14] to fault identification, but have a large number of false positive [18]. This means that the definition of the right parameterization of identification algorithms is crucial in order to reduce false detections and to improve fault classifications [18][15]. Another major factor to be taken into account in order to provide a high performance in fault detection is the correct normalization of real data [48]. These problems could be avoided with the use of DOE (Design of Experiment) techniques to find the relevant parameters.

An interesting feature of AIS that well emerges from the analysis is the abnormality detection capability that allows these approaches to perform fault detection without the need of a training phase [54]. They assume that a normal system behaviour is given by its operation at the early stages of its functioning. Theoretical papers show that the performance of this approach, usually developed with Negative Selection algorithms [13], is highly dependent on the set of trained data assumed as "normal" and presents high false positive rates. Theoretical works try to overcome these limitations through the use of Danger Theory methodology[55].

Another field where AIS is commonly adopted is for the solution of optimization problems: a typical example, in maintenance field, is the scheduling of maintenance processes. The analysis shows that AIS is a good methodology to solve optimization problems, especially in case of very complex problems, where the distributed nature of these methodologies helps to reduce the computational time. Furthermore, in these applications AIS shows a good behaviour where different local optima are reached in few iterations, and after the system continuously tries to improve its own response moving towards a particular goal [13].

Concerning the application targets, the survey shows that AIS covers a broad range that fits with the maintenance needs. Mechanical devices [14][15] [20][21], electric motors [32] and electronic components [16][19] can be monitored using vibration, voltage, current and energy consumption as inputs for the determination of performance degradation .

Regarding the methodologies, AIS algorithms are mainly based on four immunologic theories (negative selection, clonal selection, Artificial Immune Network and Danger Theory). The research in all of them reaches different levels: the more advanced methodologies - that have reached the prototype evolutionary stage - are Clonal

Selection, Negative Selection and their variants. These methodologies are the simplest to use, demonstrating at the same time accurate performance in maintenance problems.

AIS is naturally a distributed system [56][4], but only few and recent applications take explicitly advantage of this feature[57]. Publications in the period between 2010 and 2014 show the use of AIS combined with Multi Agent Systems to manage operations in Smart Grids, as the proposals by Bhuvaneswari in [22] and Ramachandran in [23]. This is due to the great interest that this topic attracts and the great applicability of methodologies like MAS and AIS in this research field.

The learning or training phase is one of the relevant and critical steps to all AIS algorithms, in order to build detectors set or to initialize parameters. However, only in recent works some features like self-learning are explored, even if mainly applied in not critical applications like parameter optimization [52].

Another important outcome that emerges from the literature review is that, despite the natural distributed characteristic of an immune system, a centralized approach is adopted in the majority of the works. This is mainly due to the simplicity of the components or plants where AIS are currently implemented. In fact, distributed approaches are more advantageous when the size of the system under control scales up. For a simple system, a centralized approach results simpler to implement and not computationally too much expensive, as pointed out by [45]. By analyzing the type of applications, most of them are related to custom software developed only for one specific application without any abstraction to a more generic model.

Moreover, despite the keywords of the preliminary research are closely related to prognostic, diagnostic and AIS, in more than 49% of the papers, where the implementations of these are exposed, this is performed using Multi Agents. This reflects the idea that MAS is a good methodology to implement AIS. However, almost none of these applications use standard agent-based frameworks in their implementation (only three papers mention the Jade framework [22]). General purpose frameworks,

which have a wide utilization in other applications such as optimization algorithms and negotiation protocols (i.e. auction-based mechanisms), are not yet considered sufficiently powerful and flexible to be applied in a more complex field as advanced maintenance.

5. Artificial Immune Intelligent Maintenance System

5.1.Introduction

Artificial Immune Intelligent Maintenance System (AI2MS) is the name of the architecture, developed during this work which aims to overcome the limits of current predictive and preventive maintenance system when are faced to a very geographically widespread systems like oil transfer system via pipeline or waste water treatment systems.

These systems are composed by a huge number of devices, often placed in inaccessible areas with a large distance between them and often without possibility of data connection. For these scenarios it is mandatory that the maintenance system is distributed and, because the different devices inside the plants are activated very few times (typically once a day), the time needed to acquire enough data related to the behavior of a single device directly on field is too huge to implement a feasible traditional centralized system.

In order to overcome these limitations AI2MS has been conceived by integrating a MAS-based architecture with the main features of an artificial immune system (AIS).

Like shown in the previous chapters the use of artificial immune systems allows the implementation of a very reliable diagnostic and prognostic system even in case of unreliable network between the different stations of the system. This is primarily due to flexibility in the behaviour of AIS methodologies.

The choice of Multi Agent System to implement the architecture, on the other hand, allows to push further the flexibility of the systems by enabling an interesting mix of autonomy and distributed function that allows to overcome the strong limitations of the limited training data set (due to the few device actuations).

AI2MS is a complex architecture, constituted by several agents of various type that collaborate one each other in order to provide the maintenance functionality.

in this chapter an overview of the developed architecture is proposed.

In the first part a description of the different agents developed are proposed in order to better understand the role of the different agents in the overall system, after that an overview of the entire platform are proposed in order to show the cooperation between the different agent with a possible use case in an oil transfer pipeline system.

after that will be shown a possible implementation of the diagnostic part of the system into a real test bench, and in a more detail the ontology developed to perform this task, the implementation in Jade and the results of the test on the benchmark platform.

5.2. AI2MS Agents Description

The architecture is composed by ten different types of agents grouped in 4 different clusters on the basis of the role of the agents in the system; the cluster are:

- data provider agents;
- diagnostic agents;
- prognostic agents;
- service agents;

The first set includes the agents involved in the provision of data from the field, in the second are grouped the agents involved in the failure detection and identification tasks, the third set, instead , contains the agents responsible of prognostic features involved in the estimation of the remaining useful life of the machines, that are used to plan the maintenance interventions.

the last group include instead, all the agents that not are involved in task directly related to maintenance but collects all the agents responsible to the management and optimization of the overall platform; this last set provides the required specificity to the platform to operate into the different scenarios taken into account in this work.

5.2.1. Data provider agents

Data provider agents contains the agents responsible to manage the different sensors installed on each machines installed into the plant. in this section there are two different kind of agents: sensor agent (SA) and sensor diagnostic agent (SDA).

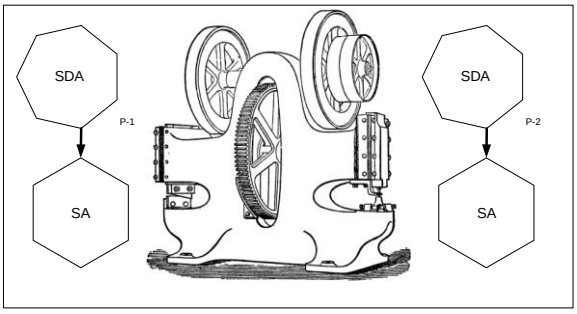

Figure 12 Data Provider Agents

Sensor Agent

Description

Sensor Agent is the agent that is responsible to the data acquisition from the field.

this agent provide the data required to other agents to provide the required functionality. the implementation of this agent can vary a lot depending by the different sensors that are managed by the single agents.

In general, in each machine are running several agents of this type and each agent provide only a specific type of data. this mean that for each sensor mounted in a machine at least one agent is continuous running in the platform.

In case of sensors that can provide a multiple set of data, such as for example a tri-axial accelerometer each axis is managed by a specific agent.

this feature is further strengthened in case of the use of smart sensors; in fact a smart sensor is often characterized by the adoption of a fieldbus for communication and besides the data provided by the sensor are also communicated a set of auxiliary information for fieldbus managing and to diagnose possible failures in the communications. in this case for each smart sensor a set of different agents will be in concurrent run a set for providing the sensor data and a set able to provide the diagnostic data.

Another task provided by this agent is the conditioning and preliminary manipulation of the raw data. this function is necessary because the raw data provided by the sensors are often too noisy to be directly used in prognostic and diagnostic algorithms and a preliminary phase of filtration, normalization, denoising is often necessary.

Moreover different kind of algorithms often require a different manipulation of the same data; in this case different agents are provided each of which implement a different filtration schema in order to serve the best data to the algorithms. however, even if the

different methodologies implemented into the system require only pre processed data, an agent for each sensor with the role of provider of the raw data is always running on the system. the data provided to this agent in fact are required to the Sensor Diagnostic agent.

The mode of operation of this agent is based on the producer-consumer paradigm, this means that each sensor agent continuously manages the sensor and executes the preprocessing of the input but the information are stored in the internal state of the agents and are not automatically provided to other agents.

When an agent require a data from a sensor a request response protocol must be used. this request response protocol is standardized between all the agents through the use of the specific ontology developed for AI2MS, discusses later in this chapter.

The mode of operation of this agent is similar to the behaviour of the sensor agent. in fact due to the similarity of the role of this agents that both are provider of information to other agents also in this case a producer consumer paradigm was adopted with the support of the same ontology used for the sensor agent for the standardization of the communication protocol in order to enable requests from all the other agents of the platform.

Sensor Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the sensor agent presents the following behaviour

Sensor Agent	
Intrinsic Characteristics	
Lifespan	Long term
Level of Cognition	Reactive
Construction	Procedural
Mobility	Stationary
Adaptability	Fixed
Modeling	External environment
Extrinsic Characteristics	
Locality	Local

Social Autonomy	Independent
Sociability	Autistic
Friendliness	Cooperative
Interactions	Logistic

Table 9 Key characteristics of Sensor Agent

Concerning the intrinsic set ,a sensor agent is characterized by a long term lifespan, due to the necessity to provide the sensor data as soon as possible, a reactive cognitive level, because the agent respond only to the other agent's requests. regarding the construction a SA is procedural because the operations needed to get the data from the sensor and for the preprocessing are coded directly into the algorithm in fixed way, this involves a very limited adaptability; about modeling, is easy to understand that SA model the external environment (a physical sensor).

Regarding the extrinsic set of characteristics, sensor agents shows an independent behaviour with an autistic social autonomy, this is due to the high grade of independence of this kind of agents that operate only as data provider in autonomous way without any control by other agent.

Concerning the friendliness SA shows a cooperative behaviour, because helps the other agents, but only with the data provided with logistic interaction due to producer-consumer paradigm adopted to model this agent.

Sensor Diagnostic Agent

Description

Sensor diagnostic Agent is an auxiliary agent with the aim of support a sensor agent in order to improve the performance and the reliability of the system into the typical scenarios of application of AI2MS.

the reason which has led to the development of this agent is located primarily into the problems related to the geographic dispersion of the machines that constitute this kind of plants; in fact in this plants some machines, like for example, an actuator for a interception valve in an oil pipeline can be located in a very harsh environment, that can damage the sensor in short time. furthermore this machine can also be located in a very difficult to reach zone like, classical examples are, in oil pipelines, equipment positioned on the ocean floor, in remote regions of Siberia or Russia in the far north etc.

etc. and the same considerations can be adopted also, with minor severity, in case of waste water treatment plants and waterworks, when the critical parts are often located underground, like ad example a well, or in a mountain region as in the case of a water source.

these limitations lead strong limitations to the possibility of fast intervention of maintenance personnel in order to solve a failure occurred to a sensors. in these cases SDA provides a reliable support to the sensor agent in order to ensure the correct functioning of machine and maintenance system even in case of degraded sensor.

this system, obviously, cannot restore, by itself, the correct functioning state of a broken system but can operate in the early stages of degradation; in fact, in case of complete failure of the sensors, the only feasible procedure to keep the system immune to this kind of failure is to implement a redundancy of the critical sensor with the installation of a secondary set of spare sensors.

the main role of Sensor Diagnostic Agent is to provide an health assessment of the sensors in order to evaluate the degree of reliability of the data provided by the different sensor agents.

these evaluation can be used for two main purposes: the first is to identify a damaged sensors and schedule a maintenance intervention, the second is to allow a proper use of the data provided by a broken sensor in expectation of a repair intervention. in this scenario in fact, the possibility of using this data, taken with the proper confidence value and safety margins can help to avoid a breakdown of the entire machine.

the health assessment evaluation can be mainly performed in three different ways, depending by the type of the sensor,

the first and simplest approach is an estimation of degradation driven by the time; in particular, some kind of sensors showing an accuracy that constantly decrease during time, due to wear or dirty, until a calibration phase is performed.

the second strategy is based about the evaluation of the results of automatic calibration processes that some sensors have implemented inside; in this cases SDA can force the execution of these processes and acquire the necessary information by performing requests of raw data or diagnostic data, if possible, directly to the sensor agents.

the third strategy are more reliable but expensive can be adopted only in case of redundant sensors. in this case by performing a comparison between the raw data

provided by the different sensors SDA can evaluate, with high accuracy, the health of each sensors and estimate in a precise way the correction factors or calibration curves to be applied to the data in order to compensate the wrong measurements.

The mode of operation

Sensor Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the sensor diagnostic agent presents the following behaviour

Sensor Diagnostic Agent	
Intrinsic Characteristics	
Lifespan	Long term
Level of Cognition	Declarative
Construction	Procedural
Mobility	Stationary
Adaptability	Fixed
Modeling	Sensor Agent
Extrinsic Characteristics	
Locality	Local
Social Autonomy	Independent
Sociability	Responsible
Friendliness	Cooperative
Interactions	Logistic

Table 10 Key characteristics of Sensor Diagnostic Agents

Like the related sensor agent, SDA shows a long term lifespan a stationary behaviour. regarding the adaptability this kind of agents is not flexible because it's behaviour is completely depending by the type of sensors under control and the configuration of the machine. it also easy to understand that the SDA need to model its related sensor agents. concerning the extrinsic characteristics instead, like SA SDA shows an independent social autonomy and a locality limited only to the machine where the proper SA is running and an independent behaviour.

the differences between the sensor agent must be found in the characteristics more related to the cooperation between agents; in particular SDA shows a responsible

behaviour, due to the intrinsic role of controller of SA, is cooperative with other agents that can require it's services with logistic interaction due to the product consumer paradigm used for the implementation.

5.2.2. Diagnostic Agents

Diagnostic agent regroups all the agents responsible about the fault detection and fault isolation that are necessary in order to provide a reliable diagnostic of the entire plant. due to the particularity of this kind of plants and due to the performance of the different methodologies used to implement an artificial immune system in the different tasks required to perform an reliable diagnostic, three different type of agents have been developed: fault detection agents (FDA), new fault detection agents (NFDA) and Cooperative Detection Agents (CDA). in particular, FDA and FDA act on a single machine with the scope of detect different kind of failures by adopting complementary methodologies, CDA instead is a migrate agent that operate on several machine in order to detect failures not related on a single machine.

Failure Detection Agent

Failure detection agent is the agent responsible of the detection of only a single specific type of failure that can occurs on a machine.

The reason on the basis of this choice of implementation is based on the assumption that every machine has some failures that have a probability of occurrence significantly higher respect than other ones.

This is mainly due to some critical components inside a machine or known problems in some equipment and these criticality can be easily detected using standard maintenance techniques like fmeca analysis or Weibull analysis based on historical reliability data.

The reason about the adoption of a dedicated agent class to detect this kind of failure, different from the agents responsible to detect generic failures, must be founded in the AIS methodology adopted: for this role, in order to increase the detection performances, avoiding meanwhile a waste of computing resources, a clonal selection approaches was adopted. clonal selection, in fact, shows the best trade off between detection capabilities and allocated resources respect other methodologies. this technique also, allows a fine tuning of these factors in independent way in each failure mode.

the implementation of this agent is based on a typical general purpose clonal selection approach in which for any kind of failures a set, constituted by an high number of agents, is in concurrent running on the system. the agents are not equal one each other, but each one has its own personalization on the parameter of the detector.

this personalization allows the system to keep in run an high number of detectors slightly similar one each other in which every detector can detect only a particular pattern with a very high specificity and very strict detection field.

this allow the system to maintain an high selectivity into the failure detection, decreasing in this way the false positive rate meanwhile maintaining a high flexibility due to the high number of detectors in concurrent run.

the regulation of the clonal selection methodology implemented is based on the following steps.

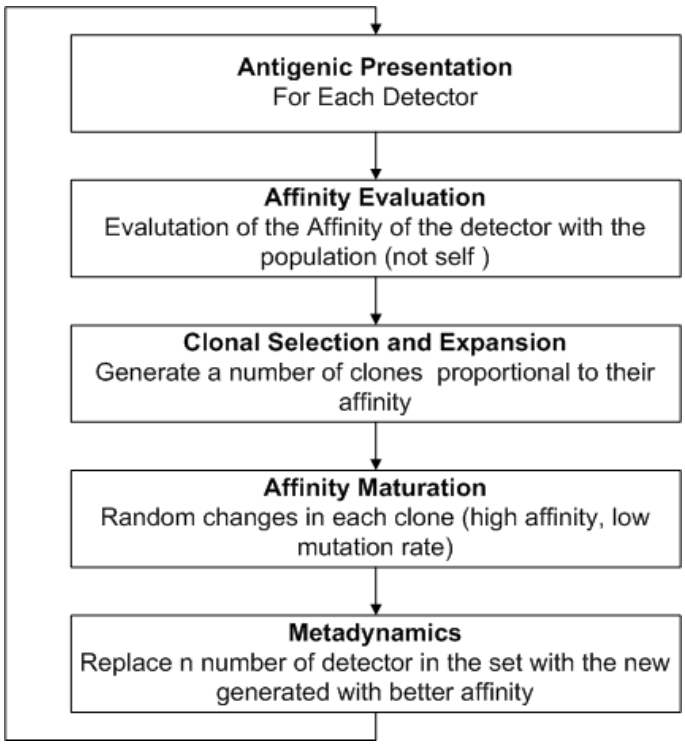

Figure 13 schema of clonal selection methodology

the first phase, the antigen presentation, is the execution of the detection procedures on a specific testing dataset, equal for each agent, that contain a set of signatures of the failure and a set of signatures of the machine.

In the second phases the affinity evaluation, the performance of the detector is evaluated through the analysis of the ratio of false positive and false negative into the testing

dataset. this value express the quality of the detector and has a strong impact on the next phases of the algorithm

the third phases, called clonal selection and expansion is the main phase of the algorithm. in this phases each detector is cloned N times. the number N of clones is directly related to the affinity of the detector. this means that more a detector is able to detect, in a precise way, the failure more clones of this detector are generated. on the other hand a poor detector is cloned a number N significant lower.

this procedure allow the system to reach, step by step through the different iterations, an increasing detection rate, and consequently a better quality of the fault isolation that it tends to reach, step by step the optimum.

the second and more important reason is avoid the local optima. in fact, in this kind of iterative algorithms characterized by successive approximations that bring to the optimization of the solution, is possible to obtain the problem of the convergence to a local optima.

In this case, if for some reason, like as example a bad initialization or due to the random nature of the changes all the agents could converge to a solution that is not the best.

An simplified example of this problem can be found in the following picture

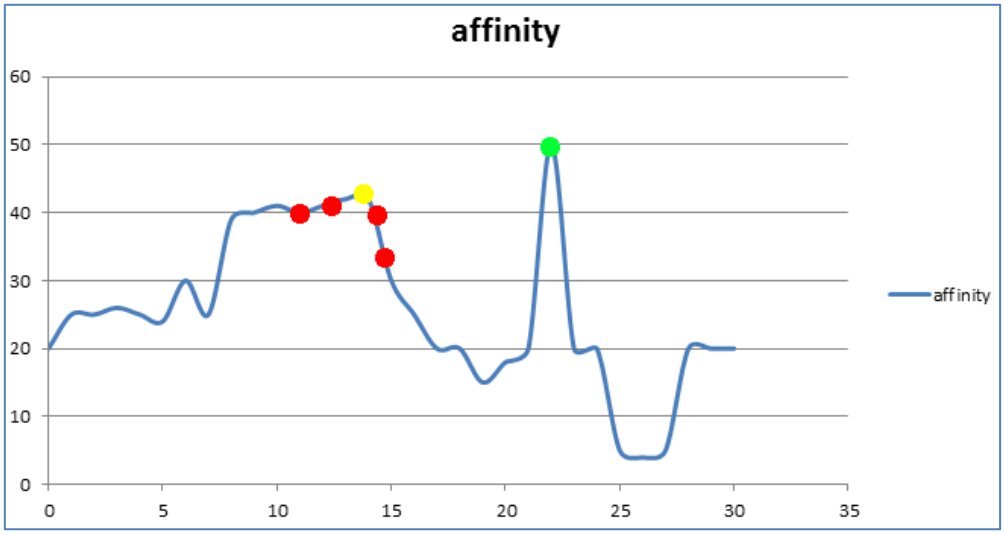

Figure 14 An example of local optimum problem

in the graph is shown in blue the value of an hypothetical affinity function that depends only by a single parameter, and with the red dots are representing the affinity value of the 4 agents that are currently running on the system. in this case all the agents, through

the iterations of the algorithm, will tend to the local optimum represented by the yellow dot and none of them will reach the global optimum that is represented by the green dot. the strong modification of the parameters of the poor detectors allows to solve this problem by spreading the agents on the entire feature space.

the last phase, called metadynamics, allows the evolution and the regulation of the resources used by the system, in this phase the new detectors created in the third and fourth step are compared with the old detectors in order to make a ranking between them based upon the estimated affinity, after this, if the new detectors shown a better affinity, a defined number of this detectors became part of the system replacing an equal number of agent with lowest performance.

In the implementation of AI2MS this procedure is applied, in autonomous way for each typical failure of each machine. in this way for each failure the better failure identification technique can be adopted.

regarding the number of agent in concurrent execution on the system with this architecture also this can vary from a failure mode to another one, and is mainly depending by the noisiness of the failure's signature. if a failure signature is very clear, in fact, the system require only few agent to achieve a good identification rate. on the other side, if the signature is very noisy more agents need to be used in order to obtain the same performance.

Failure Detection Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the failure detection agent presents the following behaviour

Fault Detection Agent	
Intrinsic Characteristics	
Lifespan	Short to long term
Level of Cognition	Declarative
Construction	Procedural
Mobility	Stationary
Adaptability	Teachable to autodidactic
Modeling	The machine
Extrinsic Characteristics	

Locality	Local
Social Autonomy	Independent
Sociability	Aware
Friendliness	Competitive
Interactions	Logistic

Table 11 FDA Key Characteristics

In this case , unlike the previous agents the lifespan is variable and depend only by the performance of the detector (short term) that affects the ranking and influences the probability of substitution into the pool; in a more general view, instead, is easy to understand that several FDA agents are always running on the system (long term).

concerning about the scope FDA agent are fixed agent able to model only a single machine with a great degree of flexibility. in particular FDA agents need to be teachable when are instanced in order to provide the needed initialization but through the iteration its behaviour is switched to autodidactic due to the natural evolution of the system.

Regarding the interaction with other agents, FDA shows an high autonomy, an aware sociability and a competitive behaviour that is expresses into the metadynamics phase where all the agents compete to remain in the execution pool.

New Fault Detection Agents

The clonal methodology implemented into the fault detection agents provides the better performance between the different AIS methodology in case the application requires the detection of specific signatures. however this methodology shows it limitations when the failure's signature is unknown. in this case in fact, the high selectivity of clonal selection detector imposes the use of a too high number of agents to cover all the feature space and make the system able to detect any kind of failure that can occur in a machine, even never seen before.

between the different AIS methodologies the one that shows the best efficiency in this task is the negative selection approach.

due to this considerations NFDA is implemented in AI2MS in order to support the FDA agent with the scope of provide the fault detection capability even in case of new failures.

NFDA in fact uses the negative selection methodology to implement a black box fault detection system that mimic the behaviour of T cells of mammalian immune system.
the schema of the algorithm on the basis of this approach is shown in the following picture

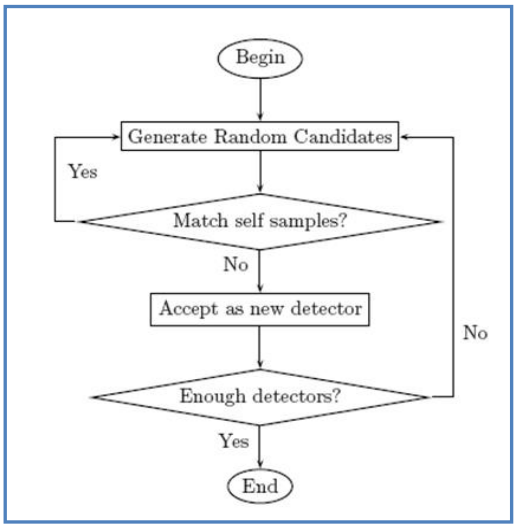

Figure 15 schema of the negative selection methodology

In particular the system works by generating a high number of detectors each of them are parameterized in random way.
each of this detectors is executed on a set of signatures of the machine in good state (negative selection) and if any detector is triggered even just one time ,in the entire dataset, it is discarded.
this process will be iterated a high number N of times generating a huge set of detectors that can cover all the feature space.
the number N agents currently running on the system is a parameter that mainly depends by the platforms, the strategy implemented in AI2MS is to generate an number N of agent able to saturate at maximum the 90 % of the unused computation resources of the machine
this choice was made due to the high relevance of failure detection capability for the scope of AI2MS that is secondary only to the fault identification provided by FDA agents; so while in FDA the number of agents are related only to the performance without any care about the performance of other agents in the NFDA management

instead, it was considered and the number of agents running in the platform varies during the time on the basis of the computational effort of the machine.

New Failure Detection Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the new failure detection agent presents the following behaviour

New Fault Detection Agent	
Intrinsic Characteristics	
Lifespan	Long term
Level of Cognition	Declarative
Construction	Procedural
Mobility	Stationary
Adaptability	Autodidactic
Modeling	The machine
Extrinsic Characteristics	
Locality	Local
Social Autonomy	Independent
Sociability	Aware
Friendliness	Competitive
Interactions	Logistic

Table 12 NFDA Key Chareacteristics

Regarding the intrinsic characteristics NFDA shows a behaviour similar to FDA, in fact also this agents are implemented in order to be executed only in a single machine (local and stationary) and have the role to model the entire machine. the main difference is that this agents have always a long life lifespan and its adaptability is only autodidactic due to the black box approach used in negative selection algorithms.

Concerning about the extrinsic characteristics, instead, the NFDA agents show the same characteristics of FDA, in fact also NFDA shows an independent behaviour with aware of the other agents running on the machine, a low level (logistic) of interaction wile the behaviour respect other NFDA and FDA agents are competitive because all of them try to detect the failures in independent way.

Cooperative Detection Agents (CDA)

The hybrid approach between clonal and negative selection methodology used in the previous agents allows a full coverage of the machine's feature space and consequently an optimal capability of fault detection and isolation. however in the kind of plants where AI2MS is developed there is an high possibility that a failure can occur outside a machine in a part of the plant not under control.

an example of these failure, one of the most famous and also one with the highest severity) is a leakage in an oil pipeline.

this failures cannot be detected using the agent description before because the internal state of both the machine, the one before the leakage and the one after the leakage, are not influenced by the failure.

in order to solve this problematic a specific agent has been implemented, the Cooperative detection agent (CDA)

CDA is a mobile agent that is able to migrate from each machine to another in order to provide these fault detection capabilities.

due to unreliability of the network that characterized the typology of plants under study, for this agent is chosen the adoption of standard prognostic algorithms in a multi agent environment avoiding the use of AIS methodology

in fact the high number of agents required to implement a reliable fault identification using any methodology derived from artificial immune systems would involve a massive usage of network communications, communications that can be very hard or very costly specially in oil pipeline applications.

the strategy adopted involve instead, the adoption of a single agents for each part of the plants not under control between two controllable machines. this configuration allows to minimize the time required for the transmission in order to increase the probability of successful transmission, and consequently the correct operation of fault detection tasks even in case of network subjected by an high failure rate.

with a standard approach, instead the entire fault identification can be achieved by using only one agent that continuously migrates between the two machines performing the required task from one side to the other one using only one shot communication protocol.

furthermore, in case of machines not linked in network the multi agents implementation adopted allows a minimal set of capabilities; this can be achieved by allowing the migration of the agents not only by using network connection but also by using some device carried by maintenance personnel.

however in this cases CDA agents cannot provide a full coverage of the plants with a fast response but this solution provide, in any cases , a better coverage respect any other solution.

Cooperative Failure Detection Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the cooperative failure detection agent presents the following behaviour

Cooperative Fault Detection Agent	
Intrinsic Characteristics	
Lifespan	Long term
Level of Cognition	Reactive
Construction	Procedural
Mobility	Itinerant
Adaptability	teachable
Modeling	The environment between two machine
Extrinsic Characteristics	
Locality	Local
Social Autonomy	Independent
Sociability	Aware
Friendliness	Autistic
Interactions	Logistic

Table 13 CDA Key Characteristics

regarding the intrinsic characteristics, the main difference respect other detection agents is the itinerant mobility, and a reduced flexibility respect FDA and NFDA.

this is primarily due to the limited set of failures that can occur outside the machines and due to the implementation consideration described before that require only a training phases, eventually repeated during time of the algorithm used.

regarding the extrinsic characteristics ,instead, CDA shows a more independent behaviour characterized by Independent autonomy, and aware sociability with an autistic friendliness.

this is mainly due to the implementation strategy chosen for AI2MS that enables only one CDA for each section; for this there is no need to cooperation between the different CDA running in the platform with no need of cooperation between them.

5.2.3. Prognostic Agents

prognostic Agent group a set of agent responsible of prognostic capabilities of AI2MS. the motivation on the basis of these agents is set on the basis of the evolution of maintenance needs in recent years.

More in detail, nowadays the company competitiveness requires to be able to use production facilities with a high level of reliability, availability and safety. In this context, most companies still need to undertake a breakthrough in carrying out their maintenance activities by shifting from a traditional ”Fail and Fix (FAF)” approach to a ”Predict and Prevent (PAP)” maintenance methodology: the former is a breakdown maintenance approach (or run-to-failure maintenance), which takes place only when a breakdown occurs, while the latter is a time-based preventive maintenance (also called planned maintenance), which sets a periodic interval to perform maintenance intervention regardless the health status of a physical asset Recently, other maintenance approaches such as Condition-Based Maintenance (CBM) have emerged to support companies on this challenging area. CBM is defined as maintenance carried out according to need as indicated by condition monitoring, which aims at promptly detecting, diagnosing, signaling and highlighting any deviation from the nominal operations of machines. The advantage of this strategy lies in the possibility of preventively maintaining the system only when necessary, thus, in principle, saving resources and system availability. The natural evolution of CBM is prognostic where the trigger for maintenance is not only a threshold value but an estimation of the Remaining- Useful-Life (RUL) of the machine.

This set contains two different agent, the Device Health Assessment Agent (DHAA) and the Plant Health Assessment Agent (PHAA) with different scope.

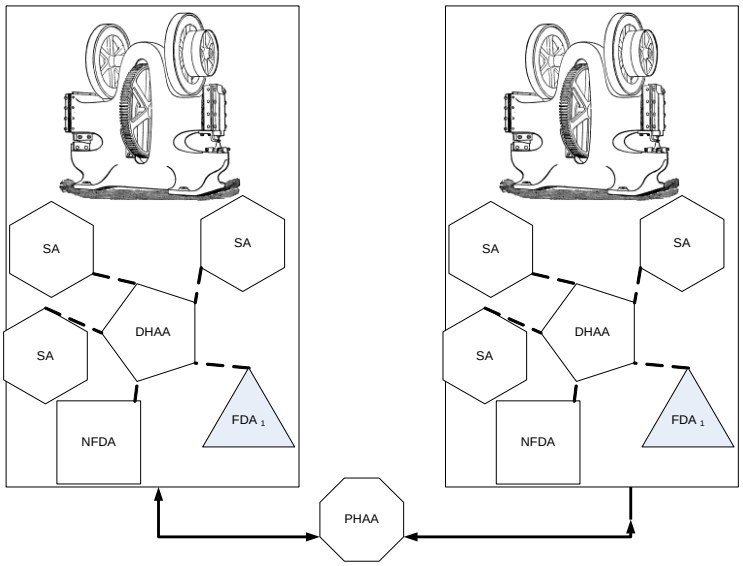

Figure 16 Prognostic Agents

Device Health Assessment Agent (DHAA)

Device health assessment agent is the agent responsible to the management of a single machine with the aim to estimate the Remaining useful life of the machine where this agent is executed.

this agent perform the specific prognostic algorithm developed by maintenance experts specifically for this machine using the data provided by the other agents currently run in the device.

the choice to leave the implementation to a maintenance expert derives from the consideration, that nowadays there are no algorithms able to provide an reliable prognostic on a generic machines but every algorithm need to be fine tuned in order to adapt to a specific machine.

furthermore not only the parameters of the system need to be tuned, depending on the machine also the methodology used need to be wisely chosen.

these depends by the fact that, even if a black box approach is adopted, the set of sensors used, the preprocessing of raw data, the filter methodology and the feature extraction techniques implemented can provide a wide range of performance depending by the typical failure mode of a machine and by the way where the machine component degrade during time.

this fact is the main motivation because, in the implementation of Sensor Agent, AI2MS provide a very high grade of flexibility by allowing the implementation of any pre processing methodologies required by the application.

In order to improve the computational efficiency, the DHAA can also acquire the results of FDA, CDA and NFDA agents in order to allow the system to reuse the data provided by avoiding the necessity of recalculate some data wasting computational resources that can be used in a better way by increasing the number of NFDA agents running in the platform.

Regarding the implementation AI2MS was configured in order to maintain a single DHAA agent running in each machine every time, this agents continuously update the estimated RUL of the machine on the basis of the new data provided by the other agents. like other sensor and diagnostic agents a request response protocol is implemented using the specific ontology in order to provide the computed RUL to the maintenance personnel and PHAA agent.

Device Health Assessment Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the device health assessment agent presents the following behaviour

Device Health Assessment Agent	
Intrinsic Characteristics	
Lifespan	Long term
Level of Cognition	Declarative
Construction	Procedural
Mobility	Stationary
Adaptability	teachable
Modeling	A single machine
Extrinsic Characteristics	
Locality	Local
Social Autonomy	Independent
Sociability	Aware
Friendliness	Autistic
Interactions	Quality

Table 14 DHAA Key Characteristics

Regarding the intrinsic characteristics DHAA is a long term agent that model a single machine, due to the general characteristics of prognostic algorithms DHAA shows a

declarative level of cognition with procedural construction. due to the scope limited on the single machine where the agent is executed its mobility is set to stationary where the adaptability are set to teachable. this is mainly due to the high specificity of prognostic algorithms that involves a fixed layout of the procedure where the parameters are often configured starting from a set of training data acquired from a good state machine. This data set can often been shared with NFDA agents reducing the number of data required for the application.

this is an interesting feature because one of the main problem in the implementation of prognostic algorithms in the typology of plant in the scenario under consideration is the lack of a large amount of data due to the few actuations of the devices.

concerning about extrinsic characterized, instead, DHAA is classified like an independent agent that work alone (autistic friendliness) in a local way because it scope is the machine where it runs, the level of interaction instead is usually higher respect other agents because for acquire the data needed to perform prognostics, it needs to communicate with all the other agents that are in execution.

Plant Health Assessment Agents

Plant health assessment agent is the agent responsible to estimate the remaining useful life of the entire plant.

Only one instance of this agent is in execution in the entire plant with the aim to collect all the RUL estimated by DHAA from each machine and, on the basis of these values, provide the health of the entire plant to maintenance personnel.

the motivation on the basis of the choice of implement or less this agent must be sought in the topology of these systems.

in fact both the aqueducts that waste water treatment plants and even more the oil pipelines present an high number of redundant systems, auxiliary systems and emergency systems developed with the aim to increase the availability of the plants and keep the system up even in case of serious damage in a section of the system.

due to these considerations is very hard to estimate the RUL of the entire plants using the health of each machines without a complete knowledge about the plant topology and the backup capabilities provided by each part of the system.

In order to solve this problem PHAA was developed like a itinerant agents that have built it inside all the knowledge related by the plant topology and by migrating from a

machine to another collects all the prognostic information of each machines and by analysing bottlenecks, single points of failure and redundancy architecture are able to detect the weak parts of the plants in order to provide an accurate estimation of the remaining useful life of the entire plant.

Plant Health Assessment Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the plant health assessment agent presents the following behaviour

Plant Health Assessment Agent	
Intrinsic Characteristics	
Lifespan	Long term
Level of Cognition	Declarative
Construction	Procedural
Mobility	Itinerant
Adaptability	teachable
Modeling	The entire plant
Extrinsic Characteristics	
Locality	Remote
Social Autonomy	Team Player
Sociability	Responsible
Friendliness	Cooperative
Interactions	Quality

Table 15 PHAA Key Characteristics

regarding the intrinsic characteristics PHAA shows a behaviour similar to other itinerant agents in AI2MS with a long term lifespan, a declarative level of cognition and, an itinerant mobility necessary to perform the required task.

The level of adaptability of PHAA is teachable, this is primarily due to the high stability of production topology that change only in case of a big revamp of the entire plant. this involves that the adoption of self adaptive mechanisms is not mandatory and only a preliminary configuration is require in order to enable the functionality of the agent.

concerning the extrinsic characteristics, instead PHAA shows a very complicated behaviour: in fact it is characterized by a remote locality, due to the fact that the

acquisition of a RUL of a machine can affect the health of a portion of the plant where this machine is not directly involved. also it social autonomy is set to the highest level, team player, because it collaborates, in a cooperative way, with other DHAA agents to perform the prognostics while it iteration level is quality due to the natural ability to interact with the entire system under control.

5.2.4. Service agents

Service agents are a set of agents not strictly related to the diagnostic purpose. It is a set of agents responsible for the evolution and adaptation of the overall maintenance system. The role of the service agents is to assist other agents and overcome some limitations intrinsic of the typology of plants analyzed in the scenario.
this cluster regroups three different agents: Update Training Agent (UTA), Failure Mode Update Agent (FMUA) and Evolution Agent (EA).

Update Training Agent

Update training agents (UTA) is developed starting by the assumption that one of the major problems in this application is the lack of large amount of data in order to create a dataset of the machine in good state enough wide to allows an effective training of the agent related to prognostic and diagnostic purposes using a single machine. In fact, even if each machine is actuated very few times (like as example a flow control valve in a oil pipeline), usually in a plants there are several machine of the same type and often of the same model.
The idea on the basis of Update training agent is the development of an itinerant agent that is generated when a new machine is installed and migrates on all the machine of the same type, using the same methodology of CDA like network connection or through devices carried by maintenance personnel.
the role of this agent is to share the training data between the different machines by acquiring the data from a machine and providing, to the same machine, the training data provided by other machine not currently hold by the machine where the agent is execution.

When a machine receive this new data an internal updating mechanism are activate in order to force the system to retest all the NFDA agents and updating the parameter of DHAA with the aim to improve the capabilities of this components.
furthermore, in order to optimize the resources, this agents have a limited life, when all the machines are updated this agency was destroying freeing resources that can be used to more important agents.

Update Training Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the update training agent presents the following behaviour

Update Training Agent	
Intrinsic Characteristics	
Lifespan	Short term
Level of Cognition	Reactive
Construction	Procedural
Mobility	Itinerant
Adaptability	Fixed
Modeling	The machine training set
Extrinsic Characteristics	
Locality	Local
Social Autonomy	Independent
Sociability	Responsible
Friendliness	Cooperative
Interactions	Logistic

Table 16 UTA Key Characteristics

regarding the intrinsic characteristics UTA shows a very simple behaviour. it is characterized by a short term lifespan with a reactive level of cognition, because it is triggered by the start of a new machine. with a fixed e procedural implementation. regarding the mobility it is itinerant due to its role in the platform.
concerning the extrinsic characteristic UTA works alone only in the machine where it is currently executed (Local locality and independent social autonomy) while it

cooperation a show logistic interaction due to the ability to communicate with other agents for sharing the data.

Failure mode update Agent

Failure mode update agent is an agent with a role very similar to UTA, the main task, taken in charge by this agent is the spreading, between the different machines of the same type , of the fault identification capabilities provided by FDA agents with the main scope to level the performance level of each machine at the highest value.

The requirement of a specific agent to do that derive from the characteristics of FDA agents. these component in fact are stationary and local agents without possibility to communicate between different machines. furthermore this agent are also independent and this may result, also due to the geographical dispersion of the machines, the possibility that an machine can execute a set of fault identification algorithms different from another machine of the same type.

this is mainly due a couple of possible reasons. the first is that the two machines can be set and parameterized by two different maintenance teams. on the other side, the second reason is more related of the evolution of the entire system and involves new FDA agents developed on a machine during the standard operative cycles through the use of a dedicated agent described in the following section.

regarding the implementation, FMUA use the same methodology used by UTA, where, when a new FDA is generated, both in case of new machine or in an already existing machine, a FMUA is generated with a short lifespan, this itinerant Agent migrate on all the machines in the plant of the same type and perform the updating procedure (if necessary). after its task is completed the FMUA agent is destroyed in order to free the allocated resource to other task.

Failure Mode Update Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the failure mode update agent presents the following behaviour

Failure Mode Update Agent	
Intrinsic Characteristics	
Lifespan	Short term
Level of Cognition	Reactive

Construction	Procedural
Mobility	Itinerant
Adaptability	Fixed
Modeling	The machine training set
Extrinsic Characteristics	
Locality	Local
Social Autonomy	Independent
Sociability	Responsible
Friendliness	Cooperative
Interactions	Logistic

Table 17 FMUA Key characteristics

FMUA shows a behaviour pretty similar to UTA, this is mainly due to the same role of these agents in the platform, in fact the only difference between these components are the kind of data that they share.

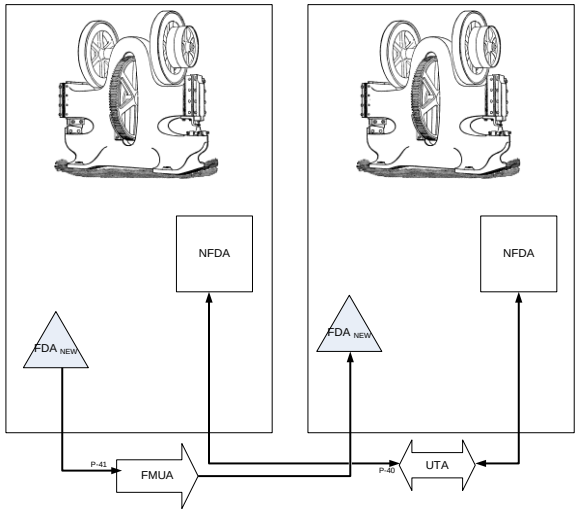

Figure 17 Failure mode update agent and update training agent overview

Evolution Agent

Evolution agent is the agent responsible of the evolution and upgrading of maintenance capabilities of the entire platform.

the evolution agent is developed starting by the assumption that, when a plant is installed or revamped, is impossible to have a full knowledge about all the devices installed in the plant, especially regarding the maintenance task. in fact the most critical maintenance information, and also the most difficult to obtain is the identification of

typical failure mode and the identification of the feature that can be analyzed in order to identify and prevent this failures.

these information can only by acquired during the years by analyzing the failure reports and the sensors data related to many of this events.

Obtain this huge amount of information before the start of the plant is often difficult and in order to overcome this limitation the evolution agent is developed.

the main task of this component is to provide a tools that can collect all the detection report provided by CDA and NFDA in the entire plant and support the maintenance expert to detect some trends or recurrent failure.

if a trend is discovered the maintenance expert can evaluate its relevance and in positive case, can provide to the implementation of a specific FDA. on the machine under analysis. Once the detector agent has been developed this is automatically spread to other machine, in automatically way, through the use of failure mode update agent.

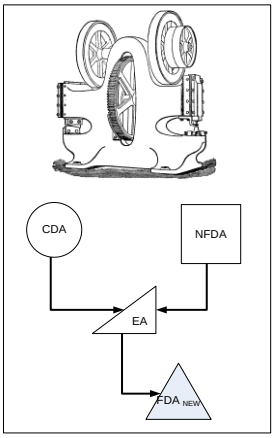

Figure 18 Evolution Agent overview

Evolution Agent Key Characteristics

Regarding the key characteristics of the multi agent implementation, the evolution agent presents the following behaviour

Evolution Agent	
Intrinsic Characteristics	
Lifespan	One shot
Level of Cognition	Reactive
Construction	Procedural
Mobility	Itinerant

Adaptability	Fixed
Modeling	The entire plant
Extrinsic Characteristics	
Locality	Local
Social Autonomy	Independent
Sociability	Aware
Friendliness	Cooperative
Interactions	Logistic

Table 18 EA Key Characteristics

Regarding the intrinsic characteristics, the evolution agents shows a one shot lifespan and a itinerant mobility , due the operation mode implemented, in fact evolution agent require the cooperation of maintenance personnel, for this reason keep it running continuously is a waste of resources and the better implementation is a manual activation. as level of cognition and construction EA is a reactive agent that works only on the basis of FDA currently running on the system in a procedural way.

concerning the extrinsic characteristics, instead, the evolution agent shows an independent autonomy in a local scope due to the fact that the agent work on a single machine to implement the new FDA, its sociability is set to aware due to the inner knowledge of the CDA and NFDA agents that work on the plant whit interact only to low level with a logistic approach; its friendliness is set to cooperative because it don't competes respect the other agent's task

5.3.AI2MS Platform Overview

AI2MS is composed by a set of agents of the types described in the previous sections of this chapter, in particular in a typical implementation of this platform there are:

at global level

- 1 PHA that continuously evaluates the health assessment of the entire plant
- 1 evolution agent for each maintenance team that currently investigating on the failure mode typical of this device
- 1 UTA for each new machine installed on the plant until the completion of its task
- 1 FMUA for each failure mode discovered by maintenance personnel until the completion of its task

at machine level

- 1 SDA for each sensor installed on the machine
- 1 SA for each presentation mode of each machine sensor
- N*M SDA where N is the number of known failure mode of the machine and M is the mean number of clones for each failure, due to clonal selection approach
- K NFDA agent where K depends on the free resources available on the machine
- J CDA where J expresses the number of critical not sensorized section between two machine

in addition to these agents, two auxiliary agents need to be implemented in order to allows the interaction between the main agents.

These auxiliary agents need to implement the yellow and white page services common on several multi agent system. in AI2MS this is been achieved with the use of directory facilitator (DF) and agent management system services provided by JADE middleware.

5.2.1. AI2MS Key characteristics

the typical multi agent implementation of AI2MS shows this key characteristics.

Artificial Intelligent Immune Maintenance System	
System Characteristics	
Uniqueness	heterogeneous
Granularity	Coarse
Control Structure	Hierarchy
Interface Autonomy	Ontology based
Execution Autonomy	Elevated
Framework Characteristics	
Design Autonomy	Medium Grade
Communication Infrastructure	Asynchronous connection oriented message protocol
Directory Service	Jade Directory Facilitator
Message Protocol	JADE HTTP based Message Transfer Protocol
Mediation Services	Not required
Security Service	TCP based connection

Remittance Services	Not required
Operations Support	JADE support
Environment Characteristics	
Knowable	Partially
Predictable	Partially
Controllable	Not controllable
Historical	Yes
Teleological	Yes.
Real time	No

Table 19 AI2MS Key Characteristics

Due to the nine different type of agents implemented in AI2MS is easy to understand that the platform is heterogeneous, furthermore, due to the coverage only of maintenance needs the granularity of the control is only coarse due to the inability of the system to control each functionality of each machine in the plant, instead, because AI2MS is designed in order to operate like an auxiliary and autonomous system respect the plant control system it execution autonomy is elevated.

regarding the message exchange between the different agents, that influences many characteristic of the platform the system, through the use of JADE message transport system was implemented an asynchronous communication channel based on connection oriented TCP packet.

The use of TCP ensures the correct delivery of the messages between the different agents, the management of the network and the congestion avoidance capability, necessary to the intrinsic unreliability of the network in these scenarios.

While TCP protocol ensures the safety of communication system, no methodology are implement in order to provide a secure communication media between the different agents. the reason on the basis of this choice is the limited computational power that is usually available in the machine systems. in fact most of the messages are exchanged in a local environment and furthermore ,adding a further encryption layer to the communication can decrease the fault detection capabilities by reducing the number of NFDA agents available.

In case this feature is mandatory, however, this can be easily implemented at higher level through the use of VPN or other technology that allow a secure communication in a transparent way.

regarding the environment it shows to be partially knowable and predictable. this is due to the consideration that a set of failure can be typical of the machine and can be predicted on the basis of the historical failure data but on the other hand there is always a level of uncertainty derived from the possibility of the occurrence of a new failure never seen before.

regarding teleological aspect instead, the environment is teleological because the application field on the platform is limited on the plant under control and it is specifically designed to perform a required task.

concerning real time characteristic the system is not considered real time due to the mean time required to a machine to degrade that is several time major respect the computation time of prognostic and diagnostic algorithms implemented in the agents.

5.4.AI2MS Implementation

For the development and the following implementation of AI2MS, GAIA approach has been used for the design of the platform.

5.4.1. Introduction to Gaia

Gaia is a conceptual framework that describes a series of steps and procedure in order to allow to an analyst to go systematically from a statement of requirements to a design that is sufficiently detailed that it can be implemented directly. Note that the requirements capture phase as being independent of the paradigm used for analysis and design.

An overview of the complete GAIA methodology is proposed in the following picture

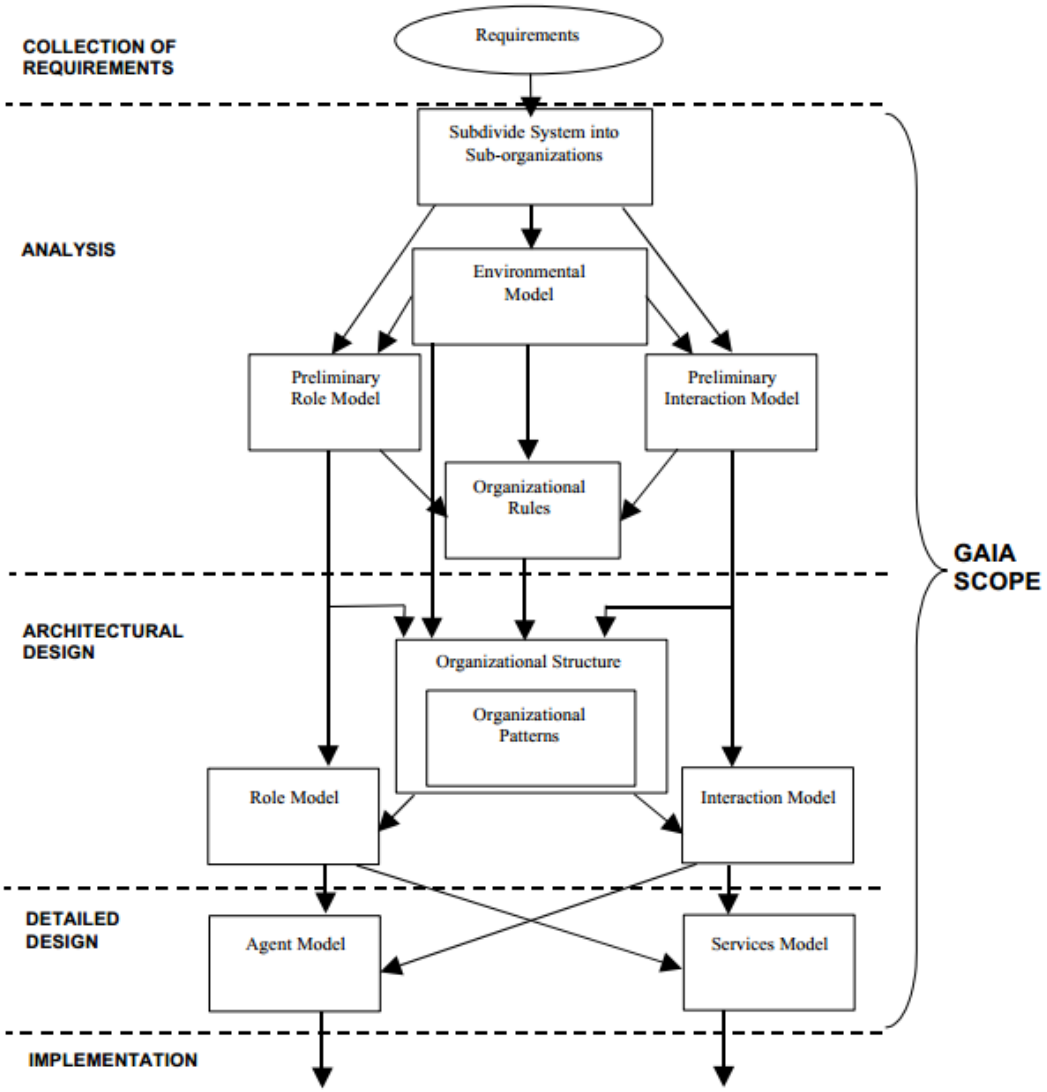

Figure 19 Overview of GAIA Methodology

in particular due to the low level of tasks that AI2MS need to cover it was made the choice to avoid the use of GAIA for the analysis phase. the phase that leads to the definition of the set of agents available was made in a traditional way.

the GAIA methodology has been used, instead, to model the role of the different agents, and the interactions between them.

this allows an easy definition of the messages required able to be directly translated in a FIPA compliant ACL messages.

5.4.2. The role model

for each role a set of characteristics needs to be analysed. this set is described in the following table.

Role Schema		Name of Role
Description		Short description of the role
Protocol and Activities		Protocol and activities in which the role plays a part
Permission		"rights" associated with the role
Responsibilities	Liveness	Liveness responsibilities
	Safety	Safety responsibilities

Table 20 Gaia Role Schema

In particular, A role will have associated with it certain permissions, relating to the type and the amount of resources that can be exploited when carrying out the role. In general, permissions can relate to any kind of resource. In a human organization, for example, a role might be given a monetary budget, a certain amount of person effort, and so on. However, in GAIA model, resources are relating only to the information or knowledge the agent has. That is, in order to carry out a role, an agent will typically be able to access certain information. Some roles might generate information (read); others may need to access a piece of information but not modify it (write), while yet others may need to modify the information(change).

The functionality of a role is defined by its responsibilities. These responsibilities can be divided into two categories: Liveness and safety responsibilities.

Liveness responsibilities expresses the rule that manage the lifespan of the agent. Liveness responsibilities tend to follow certain patterns. For example, the guaranteed response type of achievement goal has the form "a request is always followed by a response". The infinite repetition achievement goal has the form "x will happen infinitely often".

liveness properties are specified via a liveness expression, which defines the "lifecycle" of the role. These expressions define the potential execution trajectories through the various activities and interactions (i.e., over the protocols) associated with the role joined through the use of a set of specific operator defined in the following table

Operator	Usage	Interpretation
.	x.y	X followed by y
|	x|y	X or Y occurs

*	X*	X occur 0 or more time
+	X+	X occur 1 ore more time
ω	X^{ω}	X occur infinitely often
[ ]	[X]	X is optional
||	X||Y	X and Y interleaved

Table 21 GAIA Liveness Operator

The atomic components of a liveness expression are either activities or protocols. An activity is somewhat like a method in object-oriented terms, or a procedure in a PASCAL like language. It corresponds to a unit of action that the agent may perform, which does not involve interaction with any other agent. Protocols, on the other hand, are activities that do require interaction with other agents. Usually an activity is underlined to provide an easy visual identification.

the liveness predicates allows to specify the behaviour of the agent, however In many cases, it is insufficient because sometimes will be required to maintain certain invariants while executing. For example, we might require that a particular agent taking part in an electronic commerce application never spends more money than it has been allocated. These invariants are called safety conditions, because they usually relate to the absence of some undesirable condition arising. Safety requirements in Gaia are specified by means of a bullet list of predicates. These predicates are typically expressed over the variables listed in a user's permissions attribute.

5.4.3. The Interaction model and protocol definition

There are inevitably dependencies and relationships between the various roles in a multiagent organization. Indeed, such interplay is central to the way in which the system functions.

In Gaia, such links between roles are represented in the interaction model. This model consists of a set of protocol definitions, one for each type of inter-role interaction. Here a protocol can be viewed as an institutionalized pattern of interaction. That is, a pattern of interaction that has been formally defined and abstracted away from any particular sequence of execution steps. Viewing interactions in this way means that attention is focused on the essential nature and purpose of the interaction, rather than on the precise

ordering of particular message exchanges. This approach means that a single protocol definition will typically give rise to a number of message interchanges in the run time system.

The protocols are represented in a dedicated schema reported in the following schema

Protocol Name		
Initiator	**Partner**	**Input**
Description		**Output**

Table 22 Gaia Protocol Schema

A protocol definition consists of the following attributes:

- Protocol Name. brief textual description capturing the nature of the interaction (e.g., "information request", "schedule activity X" and "assign task Y");
- Initiator. the role(s) responsible for starting the interaction;
- Partner. the responder role(s) with which the initiator interacts;
- Inputs. information used by the role initiator while enacting the protocol; — Outputs. information supplied by the protocol responder during interaction;
- Description. textual description explaining the purpose of the protocol and the processing activities implied in its execution.

5.4.4. The service Model

The aim of the Gaia services model is to identify the services associated with each agent role, and to specify the main properties of these services. By a service, we mean a function of the agent. In OO terms, a service would correspond to a method; however, we do not mean that services are available for other agents in the same way that an object's methods are available for another object to invoke. Rather, a service is simply a single, coherent block of activity in which an agent will engage. It should be clear there every activity identified at the analysis stage will correspond to a service, though not every service will correspond to an activity. The services that an agent will perform are derived from the list of protocols, activities, responsibilities and the liveness properties of a role.

For each service that may be performed by an agent, it is necessary to document its properties. Specifically, we must identify the inputs, outputs, pre-conditions, and post-

conditions of each service. Inputs and outputs to services will be derived in an obvious way from the protocols model. Pre- and post-conditions represent constraints on services. These are derived from the safety properties of a role. Note that by definition, each role will be associated with at least one service.

5.4.5. AI2MS Schema

In order to Implement the platform the nine type of agents described in the previous chapter are mapped into the role schemas, in particular for each agent a particular role need to be developed. The results of this are reported in the following schemas

Role Schema		Sensor Agent
Description		Responsible to provide the data acquired by a sensor
Protocol and Activities		Register_DF, Query_DF ,Query_Sensor, Subscribe_SA, Inform_SARaw, Inform_SAFilter, Calibrate_SA, Calibrate_Sensor
Permission		Read Sensor Change rawData Change FilteredData Change CalibrationData
Responsibilities	Liveness	READ_SENSOR= (Query_Sensor)$^{\omega}$ SUBSCRIBE = Subscribe_SA.(Inform_SAFilter | Inform_SARaw)$^{\omega}$ INFORM = Inform_SAFilter | Inform_SARaw CALIBRATE = Calibrate_SA . Calibrate_Sensor
	Safety	True

Role Schema	Sensor Diagnostic Agent
Description	Responsible to provide a confidence value of sensor Data
Protocol and Activities	Register_DF, Query_DF Calibrate_SA, Inform_CV, Inform_SARaw, Inform_SAFilter,Estimate_CV ,

		Subscribe_SA
Permission		Change Timestamp Change CV
Responsibilities	Liveness	SUBSCRIBE= Subscribe_SA CALIBRATE = (Calibrate_SA)$^{\omega}$ CALCULATECV=(Inform_SAFilter | Inform_SARaw). Estimate_CV
	Safety	True

Role Schema		**Failure Detect Agent**
Description		Responsible of detection a known failure mode
Protocol and Activities		Register DF, Query DF, Inform_SARaw, Inform_SAFilter, Inform_CV, Inform_FDAFailureDetect, Perform_CS, Traning_FDA, Subscribe_SA, Inform_NewFDA, DetectFailure, getFDAList
Permission		Change FDAParameter Change FDAStatus Change CloneNumber
Responsibilities	Liveness	SUBSCRIBE= Subscribe_SA CLONING= (Perform_CS+). Traning_FDA UPDATING= Inform_NewFDA. CLONING CHECKFAILURE=((Inform_SAFilter | Inform_SARaw).[Inform_CV])*. DetectFailure)$^{\omega}$ INFORM_FAILURE= CHECKFAILURE.[Inform_FDAFailureDetect]
	Safety	CloneNumber<MAXCLONENUMBER

Role Schema	**New Failure Detect Agent**
Description	Responsible of detection a unknown failure mode
Protocol and Activities	Register DF, Query DF, Inform_SARaw, Inform_SAFilter,

		Inform_CV, Inform_NFDAFailureDetect, Perform_NS, Traning_NFDA, Subscribe_SA, Inform_NewTrainingSet, DetectFailure, getTrainingSet
Permission		Change NFDAParameter Change NFDAStatus Read FreeResources
Responsibilities	Liveness	SUBSCRIBE= Subscribe_SA CLONING= (Perform_NS+). Traning_NFDA UPDATING= Inform_NewTrainingSet. CLONING CHECKFAILURE=((Inform_SAFilter | Inform_SARaw).[Inform_CV])*. DetectFailure)$^{\omega}$ INFORM_FAILURE= CHECKFAILURE.[Inform_NFDAFailureDetect]
	Safety	FreeResources>10%

Role Schema		Cooperative Fault Detection Agent
Description		Responsible of detection a unknown failure mode that occur in a unmonitored part of a plant
Protocol and Activities		Register DF, Query DF, Inform_SARaw, Inform_SAFilter, Inform_CV, Inform_CDAFailureDetect, DetectFailure, Subscribe_SA
Permission		Change CDAStatus Change CDAParameter
Responsibilities	Liveness	SUBSCRIBE= Subscribe_SA CHECKFAILURE=((Inform_SAFilter | Inform_SARaw).[Inform_CV])*. DetectFailure)$^{\omega}$ INFORM_FAILURE= CHECKFAILURE.[Inform_CDAFailureDetect]
	Safety	True

Role Schema		Device Health Assessment Agent
Description		Responsible of health estimation of a single machine
Protocol and Activities		Register DF, Query DF, Inform_SARaw, Inform_SAFilter, Inform_CV, Inform_FDAFailureDetect, Inform_NFDAFailureDetect, Inform_CDAFailureDetect, Estimate_DHA, Inform_DHA, Subscribe_SA
Permission		Change DHAStatus
Responsibilities	Liveness	SUBSCRIBE= Subscribe_SA CHECKDHA=([(Inform_SAFilter | Inform_SARaw). [Inform_CV]]*.[Inform_FDAFailureDetect*].[Inform_NFDAFailureDetect*].[Inform_CDAFailureDetect*]. Estimate_DHA)$^{\omega}$ INFORM_DHA = CHECKDHA.[Inform_DHA]
	Safety	True

Role Schema		Plant Health Assessment Agent
Description		Responsible of health estimation of the entire plant
Protocol and Activities		Register DF, Query DF, Estimate_PHA, Inform_PHA, Inform_DHA
Permission		Change PHAStatus
Responsibilities	Liveness	CHECKPHA= ((Inform_DHA)+. Estimate_PHA)$^{\omega}$ INFORMPHA= CHECKPHA.[_ Inform_PHA]
	Safety	True

Role Schema	Failure Mode Update Agent
Description	Responsible of sharing FDA agent between the machine
Protocol and Activities	Register DF, Query DF, Inform_NewFDA, getFDAList checkFDAList
Permission	Read NewFDAParameter

Responsibilities	Liveness	CHECKFDA= getFDAList.checkFDAList SENDFDANEW= CHECKFDA.[Inform_NewFDA]
	Safety	True

Role Schema		**Update Training Agent**
Description		Responsible of sharing training data between the machine
Protocol and Activities		Register DF, Query DF, Inform_NewTrainingSet, checkNewTrainigset
Permission		Read NewTrainingSet
Responsibilities	Liveness	CHECKTrainigSet= getTrainingSet.checkNewTrainigset SENDNEWTRAININGSET= CHECKTrainigSet. [Inform_NewTrainingSet]
	Safety	True

Role Schema		**Evolution Agent**
Description		Responsible of detection creation of new FDA
Protocol and Activities		Inform_NFDAFailureDetect, AnalyseNFDA, Inform_NFDA_Analysis, request_NFDA_Analysis, Request_FDA_Creation, Create_FDA, Inform_NewFDA
Permission		Change NFDAReportList Change FDA Parameter
Responsibilities	Liveness	ANALYZENFDA= request_NFDA_Analysis. AnalyseNFDA SENDNFDAANALYSISRES= Inform_NFDA_Analysis CREATEFDA= Request_FDA_Creation. Create_FDA. Inform_NewFDA
	Safety	True

5.1.AI2MS Protocol

with the role definition, presented in the previous section, many different protocol have been introduced in order to define the different interactions that can occur between the different agents. the formal definition of these protocols are presented in the following tables.

Protocol Name Register_DF		
Initiator	**Partner**	**Input**
SA SDA FDA NFDA CDA DHA PHA UTA FMUA EA	White-page service provider	
Description Register the agent to the platform management system		**Output** Agent ID

Protocol Name Query_DF		
Initiator	**Partner**	**Input**
SA SDA FDA NFDA CDA	Yellow-pages service provider	Service required Type of Agent

DHA PHA UTA FMUA EA		
Description Interrogate the yellow page service in order to allow the communication		**Output** List of Agent ID

Protocol Name Subscribe_SA		
Initiator	**Partner**	**Input**
SDA FDA NFDA CDA DHAA	SA	
Description Subscribe a Sensor Agent in order to obtain a continuous stream of sensor's data avoiding continuous request.		**Output**

Protocol Name Inform_SARaw		
Initiator	**Partner**	**Input**
SDA FDA NFDA CDA	SA	

DHAA		
Description		**Output**
Request a single set of raw sensor data from a SA		Raw Sensor Data

Protocol Name		
Inform_SAFilter		
Initiator	**Partner**	**Input**
SDA FDA NFDA CDA DHAA	SA	
Description		**Output**
Request a single set of pre processed sensor data from a SA		Filtered Sensor Data

Protocol Name		
Calibrate_SA		
Initiator	**Partner**	**Input**
SDA	SA	
Description		**Output**
Order to a SA to perform a calibration of the sensor managed by the SA		Calibration Parameter

Protocol Name		
Inform_CV		
Initiator	**Partner**	**Input**
SDA	FDA NFDA CDA	

	DHAA	
Description Request response protocol to obtain from a SDA the confidence value of the sensor data provided by a SA		**Output** Sensor CV

Protocol Name Inform_FDAFailureDetect		
Initiator FDA	**Partner** DHAA	**Input**
Description Inform a DHAA of a failure isolated by a FDA		**Output** Type of Failure Detected

Protocol Name getFDAList		
Initiator FMUA	**Partner** FDA	**Input**
Description Request response protocol to obtain the list of the FDA currently running of the machine		**Output** List of FDA

Protocol Name Inform_NewFDA		
Initiator FMUA EA	**Partner** FDA	**Input** FDA Parameter
Description Manage the creation of a new set of FDA on the machine		**Output**

Protocol Name		
Inform_NFDAFailureDetect		
Initiator	**Partner**	**Input**
NFDA	DHAA	
Description		**Output**
Inform a DHAA of a failure detected by a NFDA		Failure Information

Protocol Name		
Inform_NewTrainingSet		
Initiator	**Partner**	**Input**
UTA	NFDA	
Description		**Output**
Send to a machine a new set of data for NFDA training		New Training Set

Protocol Name		
getTrainingSet		
Initiator	**Partner**	**Input**
UTA	NFDA	
Description		**Output**
Request response protocol to obtain the training set available on the machine		Current Training Set

Protocol Name		
Inform_CDAFailureDetect		
Initiator	**Partner**	**Input**
CDA	DHAA	
Description		**Output**
Inform a DHAA of a failure detected by a CDA		Failure Information

Protocol Name	

Inform_DHA		
Initiator	**Partner**	**Input**
PHAA	DHAA	
Description		**Output**
Request response protocol used to acquire the health assessment information from each devices in the plants		Device Health Assessment

Protocol Name Inform_PHA		
Initiator	**Partner**	**Input**
PHAA	Maintenance personnel	
Description		**Output**
Inform the maintenance personnel to the health value of the plant		Plant Health Assessment

Protocol Name Inform_NFDA_Analysis		
Initiator	**Partner**	**Input**
EA	Maintenance personnel	
Description		**Output**
Inform the maintenance personnel to the result of the analysis conducted on NFDA event		NFDA ANalysis

Protocol Name request_NFDA_Analysis		
Initiator	**Partner**	**Input**
Maintenance personnel	EA	
Description		**Output**
Request to a EA to perform an analysis on NFDA event		

Protocol Name Request_FDA_Creation		
Initiator Maintenance personnel	**Partner** EA	**Input** FDA Parameter
Description Order to EA to create a new FDA on the machine		**Output**

5.2.AI2MS FIPA Protocols

In order to implement the platform using JADE, the protocols defined in the previous section must be translated in a set of FIPA compliant messages according to the rules and message types defined in this standard. the result of this process is reported in the following table.

Protocol Name	ACL Message	ACL Type	Sender	Receiver	Content
Subscribe_SA	RequestSubscription	Subscribe	SDA FDA NFDA CDA DHAA	SA	
Inform_SARaw	RequestSARaw	Request	SDA FDA NFDA CDA DHAA	SA	
	ResponseSARaw	Inform	SA	SDA FDA NFDA CDA	Raw Data

				DHAA	
Inform_SAFilter	RequestSAFilt	Request	SDA FDA NFDA CDA DHAA	SA	
	ResponseSAFilt	Inform	SA	SDA FDA NFDA CDA DHAA	Raw Data
Calibrate_SA	RequestCalibration	Request	SDA	SA	
	ResponseCalibration	Inform	SA	SDA	Calibration Data
Inform_CV	RequestCV	Request	FDA NFDA CDA DHAA	SDA	
	ResponseCV	Inform	SDA	FDA NFDA CDA DHAA	Sensor CV
Inform_FDAFailureDetect	InformFDADetect	Inform	FDA	DHAA	FailureID
GetFDAList	RequestFDAList	Request	FMUA	FDA	
	ResponseFDAList	Inform	FDA	FMUS	List of FDA
Inform_NewFDA	InformNewFDA	Inform	FMUA EA	FDA	FDA Param.

Inform_NFDAFailureDetect	InformNFDADetect	Inform	NFDA	DHAA	
Inform_NewTrainingSet	InformNewTraining	Inform	UTA	NFDA	New training Set
GetTrainingSet	GetTrainingSet	Request	UTA	NFDA	Traning set
Inform_CDAFailureDetect	InformCDADetect	Inform	CDA	DHAA	
Inform_DHA	RequestDHA	Request	PHAA	DHAA	
	ResponseDHA	Inform	DHAA	PHAA	Device health
Inform_PHA	RequestPHA	Request	Extern	PHAA	
	ResponsePHA	Inform	PHAA	Extern	Plant health
Inform_NFDA_Analysis	ResponseNFDAAnal	Inform	EA	Extern	
Request_NFDA_Analysis	RequestNFDAAnal	Request	Extern	EA	Analysis result
Request_FDA_Creation	RequestFDANew	Request	Extern	EA	FDA Param

In particular all the protocols except Subscribe are mapped using only the FIPA ACL Request and response message type due to the one shot behaviours that characterize this events. in particular the protocols that implement a request-response type of message exchanges are modelled using a couple of message, the protocols that require only a one-directional communication, instead are mapped using only one ACL message.

the subscribe protocol, on the other side, use a subscribe message type due to the intrinsic characteristic of this protocol that is used to require a cyclic transmission of sensor data.

an overview of all these protocol is presented in the following schema

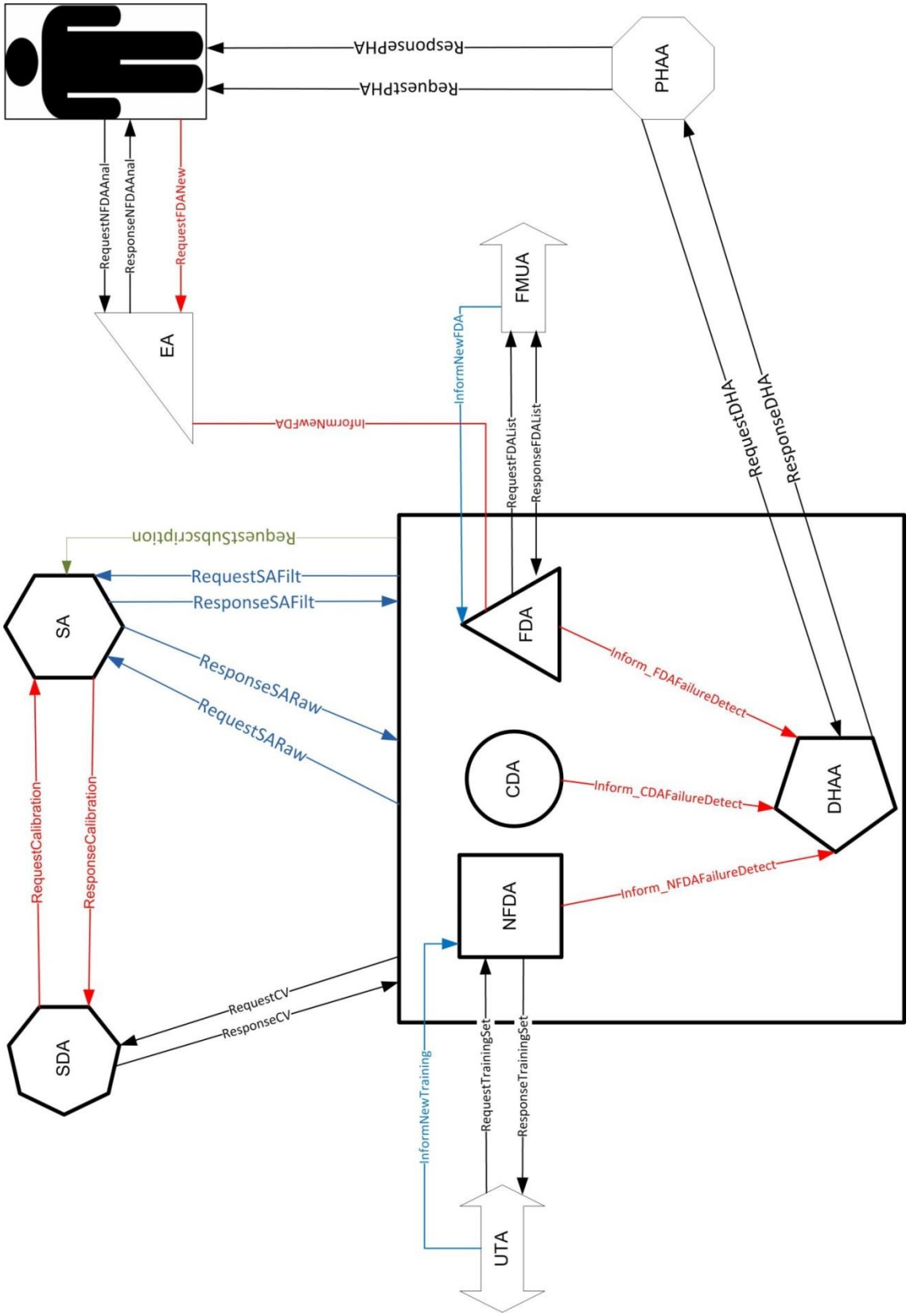

Figure 20 AI2MS messages Schema

6. AI2MS Testing

In order to validate its effectiveness, the AI2MS system has been applied on a specific laboratory test bench that is able to simulate a limited set of typical failure modes that can occur in an oil transfer system. This system is based on an electric valve actuator that is able to reproduce two different kinds of failures: a gear damage (by replacing the good gears with degraded one) and a problem on the actuator (by an increase of the resistive torque).

The test bench is equipped with vibrating sensors, positioned in the bearing of the motor shaft, that are managed by a Sensor Agent to provide information to a set of Diagnostic Agents. This group is composed by 50 agents, created using a clonal selection technique. Starting from the detector of the three failure modes. the size of the group is chosen in an empirical way in order to reach a trade-off between computational power and precision of the results.

Figure 21 AI2MS Test bench

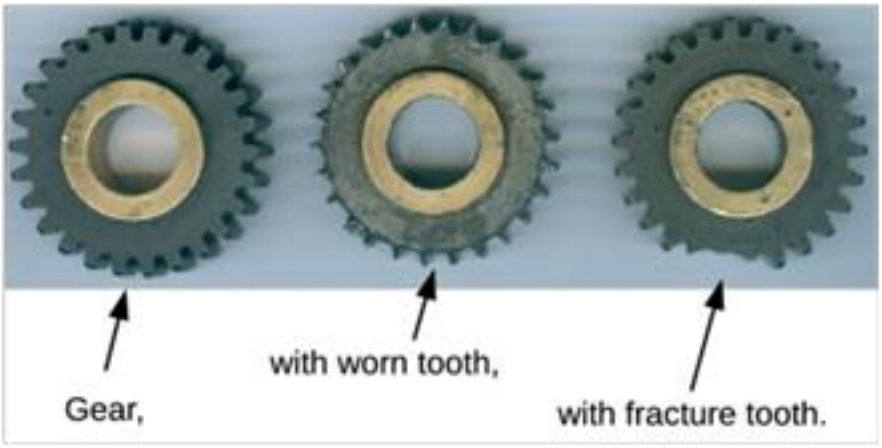

Figure 22 Gear Set used to simulate a failure

Wavelet Packet Energy (WPE) is the data processing used to generate the PerformanceSignature. WPE was pointed by (Qiu et al. 2006) as an effective method to extraction of week fault signatures from a bearing signal. Mother-wavelet Daubechies 6 achieved satisfactory results in previous works of (Gonçalves et al. 2011) and (Piccoli et al. 2012). A decomposition level of 4 was chosen, generating a PS of 16 elements. Next figure shows a signal and its Performing signature representing a normal behaviour

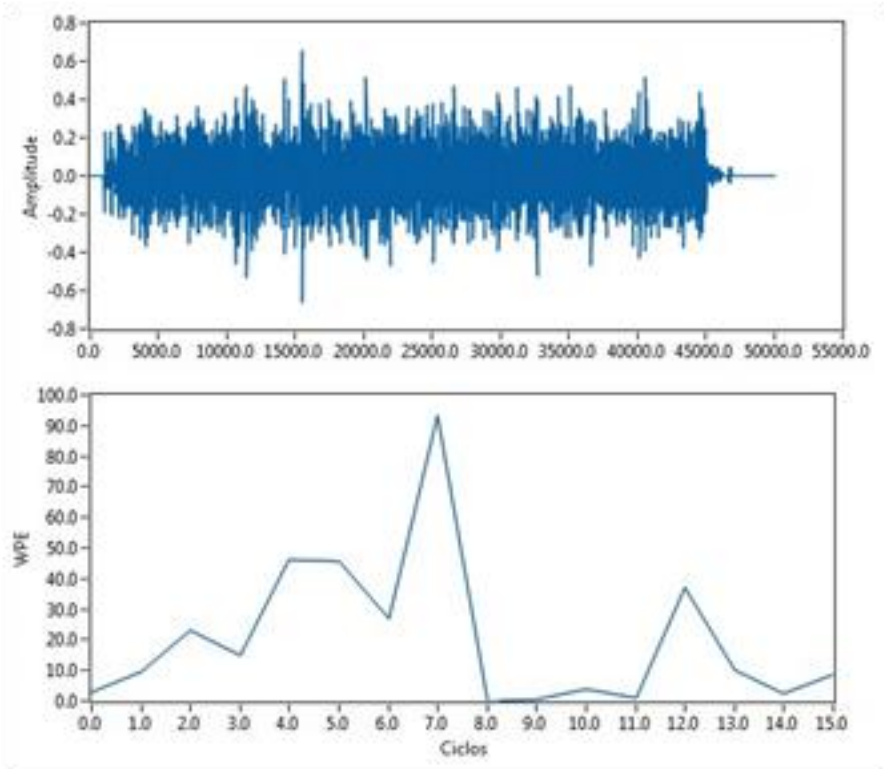

Figure 23 WPE of test bench in good state

next figure, instead presents a signal representing a degraded gear.

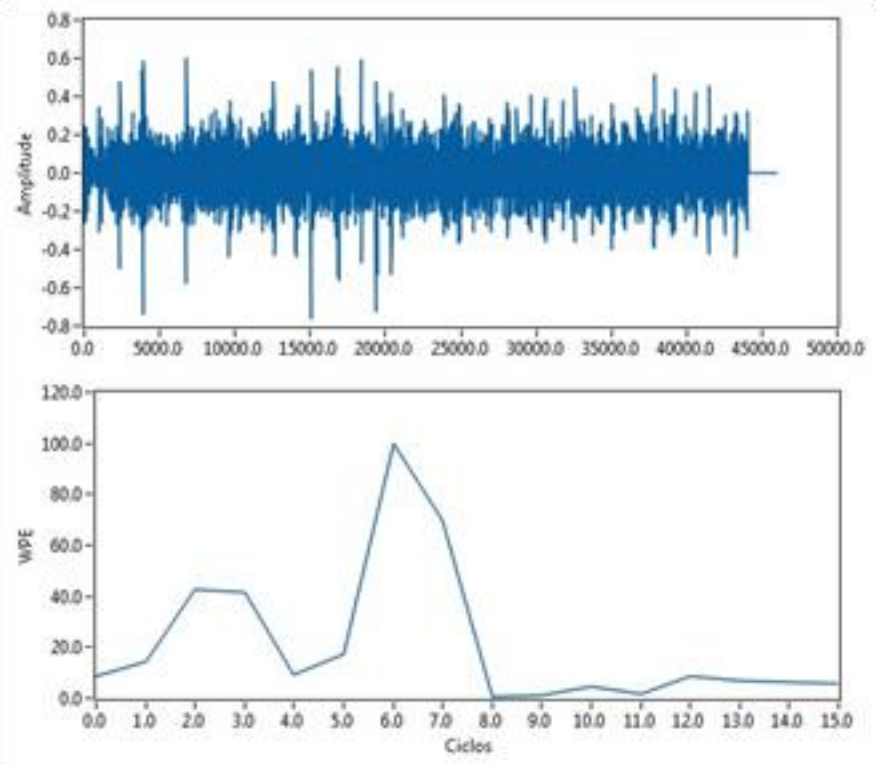

Figure 24 WPE of test bench in failure

The data set available is composed by 150 cycles of operation, witch 50 operations representing normal behaviour, 50 representing Fault 1 and 50 Fault 2.

6.1.RESULTS

Tests were performed to verify the ability of fault identification of the Diagnostic Agents implemented.

Application based on the case study was developed to simulate the behaviour. The system was configured to generate 50 Diagnostic Agents

A set of 20 cycles of each operational condition was randomly chosen to feed the New Fault Detection Agents and the two sets of known failures modes. The remaining signals where used to test the system. The following table presents the results.

Signal Type	*Number of signals*	*Faults detected*	
		Fault 1	*Fault 2*

Normal	30	1	1
Gear wear	30	29	0
Gear damaged	30	2	28

Figure 25 Testing Result

The first tests performed on the proposed diagnosing algorithm has produced 4,4% of false positives and 1,1% of false negatives. The performance of the system is, at the current state, lower with respect a typical solution (Laurentys et al., 2010). However, it is expected that, with the implementation of the service agents, the performance will increase to a value comparable to other solutions; this is because the data set used for the test is very small, comparable to a month of operation on a single device, and with the increse in the data set provided by Update Training Agents the performance will likely improve.

Further steps in the testing activities will be:

- Implementing a feedback mechanism to Evolution Agents, giving learning capabilities to the system.
- Modelling and implementation of the Collaborative Diagnostic Agent, to validate the collaborative approach proposed.
- Analysis of the intensity and quality of the message exchange, to evaluate the requirements to the network support and the possibility of integration with the plant control network.

7. Conclusion

In the present work, we have proposed a multi-agent system to support the development of a maintenance platform. Only basic operations and communications were addressed in this work.

The use of multi agent systems to implement a computerized maintenance management system is definitely a good solution whenever the system requires a good adaptability or the system is so complicated that is very hard to define a set of rules for the management of the system with a standard approach.

The overview of different maintenance needs, solved with a multi-agent implementation shown in this chapter, clearly shows the adaptability of this methodology to solve all the tasks required to a modern intelligent maintenance system also in case of very critical plants with rigid constraints.

The practical implication coming from the adoption of these technologies are quite evident: the opportunity to implement condition-based maintenance approaches in complex plants with a large number of assets delocalized allows the prediction of unexpected failures which, in turn, impacts both on the economical performance and on safety.

However, it is important to keep in mind the limits of this approach, mainly regarding the difficulty to predict the behavior of a maintenance system governed by a MAS. A wise design of the governance rule of the different agents, the communication between them and an exhaustive simulation approach for the validation of the system is still necessary in order to verify the proper functioning of the system.

Communication between agents plays a key role in the system, since the deployment of agents until the full operation. A carefully design of the interaction protocols and messages is a key element for a MAS implementation well succeeded.

To assure agent interoperability, two approaches were adopted in this work: use of a framework that implements FIPA protocols and creation of an ontology to build common concepts among different domains. Support of a modelling and design methodology was important to keep taking the specifications through the many phases of the project and the implementation. In the next phases of the AI2MS, a Model Driven

Engineering methodology to Agent development will be used to take advantage of the semi-automatic code generation of these tools.

Resource allocation tests shows that CPU occupation has a linear relation with the number of agents. The memory allocation seems related not only to agent creation, but likely related also to the intensity of message exchange.

8. References

[1] S. Elmeligy, M. Ghaffari, and J. Lee, 2010, "Transformation from Prognostics to Engineering Immune Systems," *Advanced Maintenance Engineering*, Vol. 1 no. 1 pp. 152-157.

[2] M. Zeschky, B. Widenmayer, and O. Gassmann, 2011,"Frugal Innovation in Emerging Markets," *Research Technology Management*, vol. 54, no. 4, pp. 38–45.

[3] J. Lee, M. Ghaffari, and S. Elmeligy, 2011,"Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems," *Annual Reviews in Control*, vol. 35, no. 1, pp. 111–122.

[4] U. Aickelin and D. Dasgupta, 2005, *Artificial immune systems*. Springer US, pp. 375–399.

[5] D. Dasgupta and N. Attoh-Okine, 1997, "Immunity-based systems: a survey," *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 1, pp. 369–374.

[6] Y. Ishida, 1993. "An Immune Network Model and its Application to Process Diagnosis," *Systems and Computers in Japan*, vol. 24, pp. 38–45.

[7] L. de Castro, 2001, "An Introduction to the Artificial Immune Systems," *5th International Conference on Artificial Neural Networks and Genetic Algorithms* pp. 22–25.

[8] D. Dasgupta, S. Yu, and F. Nino, 2011, "Recent Advances in Artificial Immune Systems: Models and Applications," *Applied Soft Computing.*, vol. 11, no. 2, pp. 1574–1587.

[9] J. Timmis, a. Hone, T. Stibor, and E. Clark, 2008,"Theoretical advances in artificial immune systems," *Theoretical Computer Science.*, vol. 403, no. 1, pp. 11–32.

[10] J. Timmis, 2006,"Artificial immune systems—today and tomorrow," *Natural Computing.*, vol. 6, no. 1, pp. 1–18.

[11] F. Liu, Q. Wang, and X. Gao, 2006, "Survey of artificial immune system,"*1st International Symposium on Systems and Control in Aerospace and Astronautics* , pp. 985–989.

[12] L. De Castro and F. Von Zuben, 2000,"Artificial immune systems: Part II–A survey of applications," *Technical Report RT DCA 01/99, Universidade Catolica de Santos*, pp. 0–64.

[13] E. Hart and J. Timmis, 2008,"Application areas of AIS: The past, the present and the future," *Applied Soft Computing*, vol. 8, no. 1, pp. 191–201.

[14] P. Tang, H. Kong, Z. Gan, T. Wuhan, and T. W. S. Chow, 2011, “Clonal Selection Programming for Rotational Machine Fault classification and Diagnosis", *Prognostics and System Health Management Conference*,pp.1-6.

[15] J. Strackeljan and K. Leiviskä, 2008, “Artificial immune system approach for the fault detection in rotating machinery”, *International Conference on Condition Monitoring & Machinery Failure Prevention Technologies*.

[16] J. L. M. Amaral, J. F. M. Amaral, and R. Tanscheit, 2006,“An Immune Fault Detection System for Analog Circuits with Automatic Detector Generation,” in *2006 IEEE Congress on Evolutionary Computation*, pp. 2966–2972.

[17] D. W. Bradley and a. M. Tyrrell, 2001 “The architecture for a hardware immune system,” *Proceedings. The Third NASA/DoD Workshop on Evolvable Hardware*, pp. 193–200,.

[18] C. a. Laurentys, R. M. Palhares, and W. M. Caminhas, 2011, “A novel Artificial Immune System for fault behavior detection,” *Expert System with Applications.*, vol. 38, no. 6, pp. 6957–6966.

[19] B. Hu and S. Qin, 2012,“Prognostic Methodology for Health Management of Electrical Equipments of Propulsion System in a Type of Vessel Based on Artificial Immune Algorithm,”, *IEEE Conference on Prognostics and System Health Management (PHM)*, pp. 1-8,.

[20] B. T. Thumati, G. R. Halligan, and S. Jagannathan, 2012.“A Novel Fault Diagnostics and Prediction Scheme Using a Nonlinear Observer With Artificial Immune System as an Online Approximator,” *IEEE Transaction on Control System Technology*, vol. 21, no. 3, pp. 1–10.

[21] G. R. Halligan, B. T. Thumati, and S. Jagannathan, 2011,“Artificial immune system-based diagnostics and prognostics scheme and its experimental verification,” *IEEE International Conference on Control Applications (CCA)* , pp. 958–963.

[22] R. Bhuvaneswari, S. K. Srivastava, C. S. Edrington, D. A. Cartes, and S. Subramanian, 2010,“Intelligent agent based auction by economic generation scheduling for microgrid operation,” in *Innovative Smart Grid Technologies (ISGT)*, pp. 1–6.

[23] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. a. Cartes, 2011,“An Intelligent Auction Scheme for Smart Grid Market Using a Hybrid Immune Algorithm,” *IEEE Transactions on Industrial Electronics.*, vol. 58, no. 10, pp. 4603–4612.

[24] S. Sathyanath and F. Sahin, 2002 “An Artificial Immune System Based Intelligent Multi Agent Model and its Application to a Mine Detection Problem,”, *Proceedings of the ICARIS 2002 1st International Conference on Artificial Immune Systems*.

[25] A. Dubey, G. Karsai, and N. Mahadevan, 2012, “Fault-Adaptivity in Hard Real-Time Component-Based Software Systems",*Software engineering for self-adaptive systems II*, pp. 294-323.

[26] S. Nordstrom and A. Dubey, 2006, “Ghost: guided healing and optimization search technique for healing large-scale embedded systems,”, *Engineering of Autonomic and Autonomous Systems, 2006. EASe 2006. Proceedings of the Third IEEE International Workshop on. IEEE*, pp.54-60.

[27] S. Schadwinkel and W. Dilger, 2006, “A dynamic approach to artificial immune systems utilizing neural networks,” *Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM,*, p. 131-132,

[28] M. D. Santambrogio, 2010, “From reconfigurable architectures to self-adaptive autonomic systems” *International Journal of Embedded Systems*, vol. 4 no.3, pp. 172–181.

[29] A. a. Freitas and J. Timmis, 2007, “Revisiting the Foundations of Artificial Immune Systems for Data Mining,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 4, pp. 521–540.

[30] S. T. Powers and J. He, 2008, “A hybrid artificial immune system and Self Organising Map for network intrusion detection,” *Information Sciences.*, vol. 178, no. 15, pp. 3024–3042.

[31] J. Kim, P. J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, 2007, “Immune system approaches to intrusion detection – a review,” *Natural computing.*, vol. 6, no. 4, pp. 413–466.

[32] P. J. Costa Branco, J. a. Dente, and R. V. Mendes, 2003,“Using immunology principles for fault detection,” *Industrial Electronics, IEEE Transactions on.*, vol. 50, no. 2, pp. 362–373.

[33] J. Amaral and J. Amaral, 2004, “An immune inspired fault diagnosis system for analog circuits using wavelet signatures” *... . 2004 NASA/DoD ...*, pp. 2–5, 2004.

[34] R. Canham, a. H. Jackson, and a. Tyrrell, 2003, “Robot error detection using an artificial immune system,” *Proceedings. The Third NASA/DoD Workshop on Evolvable Hardware*, pp. 199–207.

[35] M. Araujo, J. Aguilar, and H. Aponte, 2003, “Fault detection system in gas lift well based on artificial immune system,” *In the proceedings of the International Joint Conference on AI*, vol. 3, pp. 1673–1677.

[36] L. Xu, M.-Y. Chow, J. Timmis, and L. S. Taylor, 2007, “Power Distribution Outage Cause Identification With Imbalanced Data Using Artificial Immune Recognition System (AIRS) Algorithm*”, Power Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 198–204.

[37] C. Wang, S. Xia, and Y. Zhou, 2013,“Semantic Artificial Immune Model for Fault Diagnosis,” *Journal of Computers.*, vol. 8, no. 8, pp. 2059–2068.

[38] I. Aydin, M. Karakose, and E. Akin, 2010, “An adaptive artificial immune system for fault classification,” *Journal of Intelligent Manufacturing*., vol. 23, no. 5, pp. 1489–1499.

[39] F. Chen, B. Tang, and R. Chen, 2013, “A novel fault diagnosis model for gearbox based on wavelet support vector machine with immune genetic algorithm,” *Measurement*, vol. 46, no. 1, pp. 220–232, Jan. 2013.

[40] I. Aydin, M. Karakose, and E. Akin, 2007, “Artificial immune based support vector machine algorithm for fault diagnosis of induction motors,”, *Electrical Machines and Power Electronics, 2007. ACEMP'07. International Aegean Conference on. IEEE*, pp. 3–7.

[41] S. Taniguchi and Y. Dote, 2001, “Sensor fault detection for uninterruptible power supply (UPS) control system using fast fuzzy-neural network and immune network,” *Systems, Man, and Cybernetics, 2001 IEEE International Conference on. IEEE*, vol. 1, pp. 99–104.

[42] M. A. K. Jaradat and R. Langari, 2009, “A hybrid intelligent system for fault detection and sensor fusion,” *Applied Soft Computing*, vol. 9, no. 1, pp. 415–422.

[43] V. Rodin, a. Benzinou, a. Guillaud, P. Ballet, F. Harrouet, J. Tisseau, and J. Le Bihan, 2004, “An immune oriented multi-agent system for biological image processing,” , *Pattern Recognition*,vol. 37, no. 4, pp. 631–645.

[44] Y. Ishida, 1997, “Active Diagnosis Self-Organization: An Approach The Immune Network Metaphor”, *International Joint Conference on Artificial Intelligence*, Vol. 1084, pp.1089.

[45] X.-L. Hua, I. Gondal, and F. Yaqub, 2013,“Mobile agent based artificial immune system for machine condition monitoring,”, *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*, pp. 108–113.

[46] D. Bradley and A. Tyrrell,2002, “A hardwthuare immune system for benchmark state machine error detection” *Computational Intelligence, Proceedings of the World on Congress on*, pp. 813–818.

[47] C. a. Laurentys, R. M. Palhares, and W. M. Caminhas, 2010, “Design of an artificial immune system based on Danger Model for fault detection,”, *Expert Systems with Applications* , vol. 37, no. 7, pp. 5145–5152.

[48] C. a. Laurentys, G. Ronacher, R. M. Palhares, and W. M. Caminhas, 2010, “Design of an Artificial Immune System for fault detection: A Negative Selection Approach,”, *Expert Systems with Applications* ,, vol. 37, no. 7, pp. 5507–5513.

[49] S. Yuan and F. Chu, 2007,“Fault diagnosis based on support vector machines with parameter optimisation by artificial immunisation algorithm,”, *Mechanical Systems and Signal Processing*, vol. 21, no. 3, pp. 1318–1330.

[50] N. Sasaki and Y. Dote, 2002, “Diagnosis and control for multi-agent systems using immune networks,” *International Joint Conference on Neural Networks*, Vol.2 .

[51] R. O. Canham and A. M. Tyrrell, 2000, "A hardware artificial immune system and embryonic array for fault tolerant systems.", *Genetic Programming and Evolvable Machines*, Vol. 4, no.4, pp. 359-382

[52] W. Hongbing, L. Peihuang, and T. Dunbing, 2013, “Adaptive Dynamic Clone Selection Neural Network Algorithm for Motor Fault Diagnosis”, *Int. Journal Smart Sensing and Intelligent System*, vol. 6, no. 2, pp. 482–504, 2013.

[53] B. Jakimovski and E. Maehle, 2008, “Artificial immune system based robot anomaly detection engine for fault tolerant robots”, *Autonomic and trusted computing*, pp. 177–190.

[54] M. Gong, L. Jiao, W. Ma, and J. Ma, 2009, “Intelligent multi-user detection using an artificial immune system,”, *Science in China Series F: Information Sciences*, vol. 52, no. 12, pp. 2342–2353.

[55] U. Aickelin and S. Cayzer, 2008, “The Danger Theory and Its Application to Artificial Immune Systems,”, *arXiv preprint arXiv*, pp. 141–148.

[56] L. de Castro and F. Von Zuben, 1999, “Artificial immune systems: Part I–basic theory and applications,”, *Universidade Estadual de Campinas, Dezembro de, Tech. Rep*, 1999.

[57] D. P. Buse, J. Q. Feng, and Q. H. Wu, 2003, “Mobile agents for data analysis in industrial automation systems”, *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on,. IAT 2003.*, pp. 60–66.

[58] E. Scanff, K.L. Feldman , S. Ghelam, P. Sandborn, M. Glade, Foucher B. , 2007, "Life cycle cost impact of using prognostic health management (PHM) for helicopter avionics",, *Microelectronics Reliability* vol.47 pp.1857-1864.