



HAL
open science

M-Maintenance Approach based on Mobile Agent Technology

Khadija Abid, Ikbal Arab-Mansour, Chantal Bonner-Cherifi, Leila Hayet Mouss, Okba Kazar

► **To cite this version:**

Khadija Abid, Ikbal Arab-Mansour, Chantal Bonner-Cherifi, Leila Hayet Mouss, Okba Kazar. M-Maintenance Approach based on Mobile Agent Technology. IJOQM, International Journal of Operations and Quantitative Management, 2017, 23 (1), pp.1 - 21. hal-01534490

HAL Id: hal-01534490

<https://hal.science/hal-01534490>

Submitted on 13 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

M-Maintenance approach based on mobile agent technology

Khadija Abid

University of Batna2 LAP Laboratory
(abid.khadija39@gmail.com)

Ikkal Arab-Mansour

University of Lyon LIRIS Laboratory
(ikkal.mansour@insa-lyon.fr)

Chantal Bonner-Cherifi

University of Lyon 2 DISP Laboratory
(chantal.bonnercherifi@univ-lyon2.fr)

Leila Hayet Mouss

University of Batna2 LAP laboratory
(hayet_mouss@yahoo.fr)

Okba Kazar

University of Biskra LINFI Laboratory
(kazarokba@yahoo.fr)

This paper proposes an approach of mobile maintenance (m-maintenance) that aims to reduce the maintenance cost and to overcome the unavailability of experts. The Condition-Based Maintenance strategy is chosen and a three-layered framework based on mobile agent technology and web services is proposed. To ensure the reliability, Petri Nets are used in order to formally model and verify the proposed approach. Agents are modeled and their behavior with Reconfigurable Object Nets formalism. A case study on a production line is conducted to evaluate the proposed approach using LabVIEW environment. Results provide evidence of the applicability of the model.

Keywords: Mobile maintenance, Web services, Agent technology, LabVIEW, Petri nets, CBM

1. Introduction

In developing countries such as Algeria, the failures of industrial equipment pose serious problems for the companies and significantly hinder their proper functioning. In complex industrial systems, over 10,000 of devices and various software systems make maintenance tasks difficult to manage. One of the important problems in manufacturing and production plan that can make or break a business is the maintenance cost. Depending on the specific industry, it can represent from 15% to 40% of the costs of produced goods (Mobley, 1990). According to the study of T. Wireman (Wireman, 1990), the estimated cost of maintenance for a selected group of companies increased from \$200 billion in 1979 to \$600 billion in 1989. Several researches have been devoted to failure-based plant maintenance such as in (Nguyen, 2012) where the authors propose an approach leveraging the ubiquitous aspect of Enterprise Resource Planning Systems ERP. As maintenance is a key activity of companies that contributes significantly to their performance, several issues must be solved in this area such as: the lack of operator's comprehension, resources and availability of the operator.

The development of Information and Communication Technologies (ICT) allowed the emergence of a new maintenance generation, known as e-maintenance. This generation provides considerable solutions to the traditional maintenance issues. In another side, the chosen maintenance strategy has an important role in the proposed solutions. The e-maintenance gradually replaced the traditional maintenance. Several research works focused on e-maintenance (Campos et al., 2007; Campos et al., 2009; Mangina et al., 2000; Lee, 1997; Lee, 2001; Kezunovic et al., 2002; Buse et al., 2003; Wang et al., 2005; Münch et al., 2014; Fumagalli et al., 2015).

Mobile maintenance, known as m-maintenance, is a part of e-maintenance. It is defined as the "Maintenance support which includes the resources, services and management necessary to enable proactive decision process execution. This support includes e-technologies (i.e. ICT, Web-based, tether-free, and wireless) but also, e-maintenance activities (operations or processes) such as e-monitoring, e-diagnosis, etc" (Muller et al., 2008). Drawing on the work of I. Rasovska (Rasovska, 2006), this paper presents general m-maintenance architecture (Figure 1).

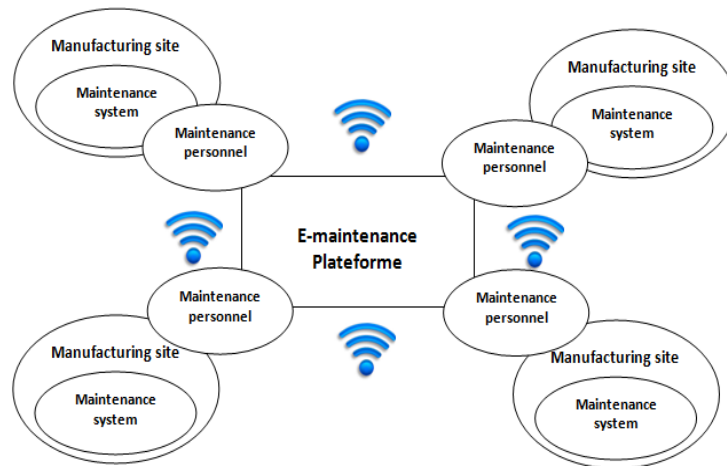


Figure 1 General *m*-maintenance architecture

As e-maintenance, m-maintenance applied one of maintenance strategies. In order to choose one of them, these strategies are studied and CBM strategy (Condition-Based Maintenance strategy) is picked. The details of this study are presented in this paper.

Furthermore, the aim in this paper is to propose m-maintenance approach in order to: to reduce maintenance costs, solve the problem of lack of experts, and achieve near-zero-downtime performance. For this reason combination of mobile CPS (Cyber-Physical Systems) and mobile agents' technology with the CBM strategy is proposed. Also web services are integrated to ensure the interoperability between different software applications distributed across a network. Besides, web services allow a possible reusability of several components and offer a high degree of security. They represent an interesting approach where huge amount of data are needed to be managed. In the other side, mobile agent showed great success in various application domains thanks to their advantages such as: autonomy, mobility, cloning ...etc. They are free and can travel among the machines in the network. Thus, it can transport its state and code to another execution environment in the network. The mobile agent technology allows to overcome the network latency and to reduce the network traffic in the system.

In order to have a secure data system, three layers architecture is adopted. This later can ensure: distributing applications between different services, communication between the layers, and layers can cooperate while being located on heterogeneous and separate machines. Moreover, with this architecture it is easier to reuse, add, or maintain some functionalities of the system without major changes to other layers.

This approach is firstly verified formally by using Reconfigurable Object Nets (RON) (Abid et al., 2013). RONs are high level Petri Nets, where tokens can be token-net and can change their structure. They allow catching mobile aspects of mobile agents and deal with the formal modeling of agents, communication, cloning and displacement.

The proposed approach is implemented and a tested on a case study, the juices plant in Algeria by using LabVIEW (Laboratory Virtual Instrument Engineering Workbench). This platform allows to elaborate higher quality projects in short time with fewer people involved. LabVIEW has the ability to scale in order to meet the needs of a given application. Furthermore, a faithful Android application, which is programmed in Java, and is meant for mobile maintenance and exchange of real-life data. The resulting application is intended to serve as a central interface between a technician and the factory.

The remainder of this paper is organized as follows. Section 2 introduces the related works. A discussion is conducted on the problem description in which the core of the studied problematic is presented. Section 3 presents the proposed approach. It details the static and dynamic modeling of the agents and it elaborates the formal verification using Petri net. To illustrate the proposed approach, a case study in Section 4 is presented. This case study deals with the establishment of mobile maintenance for a production line using LabVIEW environment. Finally, this paper finishes with a conclusion and draws some perspectives for future works.

2. Related Works

Wang et al. (2007) use mobile devices and Internet in a remote fault diagnostic system. The proposed system is based on three parts: the data acquisition and processing, the diagnosis data publishing, and the client machine. Data acquisition consists of a software that analyses the data with the remote monitoring devices. The diagnosis is made with Smart Asset Maintenance Systems (SAMS). SAMS analyze the digital signals and determine the state of the equipment. In case the machine was detected as being dysfunctional, a message is sent from the

server and a phone call is made to the maintenance engineer. Data will be ready to be accessed by the client machines, as Personal Digital Assistant (PDA) for example.

Lee (Lee, 1997; Lee, 2001) is one of the first works where agent technology was introduced in remote monitoring. The author argues that intelligent decisions can be exploited through web-enabled agents that achieve smart e-service solutions. He adopts a watchdog agent that could serve to inform about the deviation from predetermined values and could alert the maintenance staff about an impending failure. The watchdog agent is an embedded agent which obtains the data via sensors using wireless communication. It does not cooperate with other agents to achieve a common goal.

The following works introduce the notion of layer and illustrate the use of agent technology in the different layers.

The idea of using agents in e-maintenance can be also found in (Campos et al., 2007; Campos et al., 2009; Mangina et al., 2000). Authors develop a multi-layer architecture dedicated to e-maintenance. In the work of Campos et al. (Campos et al., 2007; Campos et al., 2009), the multi-layer e-maintenance process does not use agents in all layers. Only one intelligent agent is developed. Its main function is to monitor some values inserted into a database. A notification is sent to the maintenance engineer in real time in case of a high priority level, through the user interface. This interface is the first layer where the user interacts with the system. The second layer consists of the application, web services and the web server. The third layer is the database servers, where the data are entered and stored periodically. Mangina et al. (2000) propose the COndition-Monitoring Multi-Agent System (COMMAS). COMMAS architecture is based on agent technology for condition-monitoring. The architecture consists of three layers. Each layer contains an agent. The first agent senses the signal from the machine. The role of the second agent is to extract information from the data received from the first agent. The authenticity of the sensor signal is also checked and in case of a failure, an attempt is made to find the origin of such failure. The last agent is an inference engine. It has an overview of the system and evaluates the state of the plant.

In another work, more recently, L. Fumagalli et al. (2015) proposed in their paper e-maintenance architecture without agent technology. They discussed some design choices of a flexible e-maintenance platform, built based on the combined use of web services and statistical analysis. CBM was introduced in order to demonstrate the flexibility and SOA (Service-oriented Architecture) was adopted. Although the SOA present several advantages as reusability and maintainability but, implementation of SOA requires a large upfront investment by means of technology, development, and human resource.

E-maintenance and m-maintenance are realized through large networks. These maintenance processes require transmission of a huge amount of data via a network. Transmission is unsuitable for a wireless sensor network where bandwidth is limited and connection is unstable. Thus, mobile agents have the potential to greatly increase the performance of these processes under these low conditions. These advantages motivate several works to adopt mobile agents in maintenance applications. The base line technologies in the following works consist of web technologies and mobile agents.

Kezunovic et al. (2002) argue that the mobile agent can help to store and to retrieve all the information needed to perform maintenance or to repair work. Based on mobile agent and web technologies, they propose a software system for maintenance which is utilized to exchange information among agents. The idea is to make an itinerary according to the knowledge about the location of information sources, then the mobile agents travel to each source and collect the information.

Buse et al. (2003) present a system that uses mobile agents and a multi-agent system in power system environment and power substations. There are two objectives: firstly providing data analysis facilities in a power system environment, secondly furnishing data retrieval, remote analysis and reporting functions for stored data in power substation databases. Mobile agents are used as messengers to achieve communication between different agents of the system.

Xue Wang et al. (2005) propose to use a wireless sensor network for intelligent maintenance. A mobile agent collects information over the sensor network then analyzes different data. The system had to maintain a numerical control machining center. Each mobile agent associated to a sensor node acquires real-time signals and makes an elementary analysis. If a failure is detected, a failure-value is stored in the local memory of the sensor node. If a failure-value exceeds a specific threshold, a message is sent to another agent. The last agent receives the message and decides which agent will be dispatched. This approach benefits from the advantages offered by mobile agents in different phases of the maintenance process which allow obtaining better results.

More recently, T Muench et al. (2014) present an infrastructure approach for virtual enterprises developed by the consortium of the European research project ComVantage. They discuss the impact of this infrastructure with respect to the mobile maintenance domain. Their approach focuses on the core aspects of interoperability between heterogeneous environment of different machine types and interfaces. Then, a semantic abstraction layer is proposed regarding the inter-organizational access to enterprise data.

In the same direction, this work proposes a three-layer based approach where agents are mobile. A three-layer environment is easier to control and guarantee best information security. This is due to the centralized application server (the middle tier) which is more easily managed. The most difference that brings our work with respect to the other presented works, resides in the proposition of mobile agents and their exploitation in the three layers of the system. The three layers architecture provides us several advantages as described above. The introduction of web services and intelligent agents in the different steps of the maintenance process is also one of the most advantages supplied by the proposed approach.

3. Proposed Approach

In order to develop maintenance approach, maintenance strategy has to be chosen, for this reason the most common maintenance strategies are studied (Figure 2).

Corrective maintenance is carried out following detection of an anomaly and aimed at restoring normal operating conditions. To fix the damage, a preparation time is required which generates a waste of time and money. In this case of maintenance catastrophic faults may occur.

The second maintenance strategy is scheduled maintenance. Maintenance is carried out periodically at predetermined intervals, aimed to reduce the failure risk of the equipment by replacing the system components according to a schedule. Scheduled maintenance does not wait for any event. This strategy generates very important fees especially when the equipments are in good conditions.

Condition-based maintenance is based on the equipment performance monitoring. The actual equipment condition is continuously assessed by on-line detection. Maintenance is carried out when certain indicators give a signal that the equipment may fail. It is based on condition monitoring. It involves data acquisition, data processing, data analyzing, data interpreting, and information extracting from this data (Campos et al., 2009).

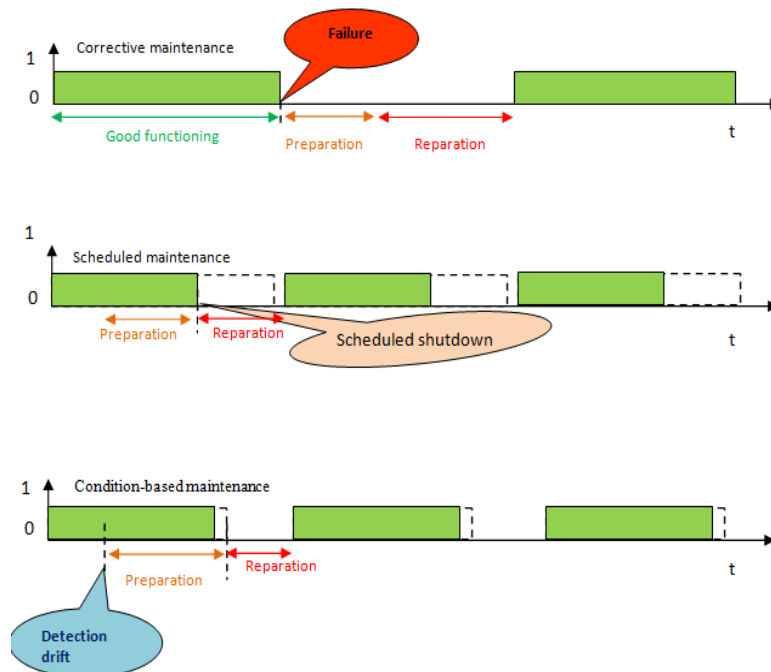


Figure 2 Common maintenance strategies

Chosen maintenance strategy should avoid unexpected passage to failure mode which may cause catastrophic faults. Besides, it has to significantly reduce maintenance costs by minimizing the number of unnecessary operations scheduled. Scheduled maintenance require total or partial shutdown of the system. Moreover, the chosen maintenance strategy should reduce the frequency of failures.

According to this analyze, CBM strategy is the best maintenance strategy that allow having best results. It overcomes serious issues and reduces maintenance costs. Applying CBM strategy allows the availability of equipments as long as possible which increase productivity.

According to (Campos et al., 2009; Mangina et al., 2000), CBM is the most commonly used strategy and multi agent system is well appropriate to support it. However, few researchers apply agent technology to manufacturing system and maintenance (Chandra et al., 2008; Weiming et al., 2006; Wernz and Deshmukh, 2007; Iung, 2003; Fletcher et al., 2003).

1. Agent Based Approach

In order to realize the proposed platform, the mobile maintenance must be made up of several elements presented in the decision system (upper part of Figure 3). The operational level (lower part of the figure) presents the machines of the manufacturing system. The two parts are connected by an internet connection that allows information exchange. The approach ensures real time monitoring and data must be acquired on-line. Data come from different signals as vibration, current, sound, etc. For this reason, a wireless connection is needed to link the different parts of the system. A database allows the storage of data and information. For mobile and real time monitoring, a technological support is necessary. CPS need to be used as: Smartphone.

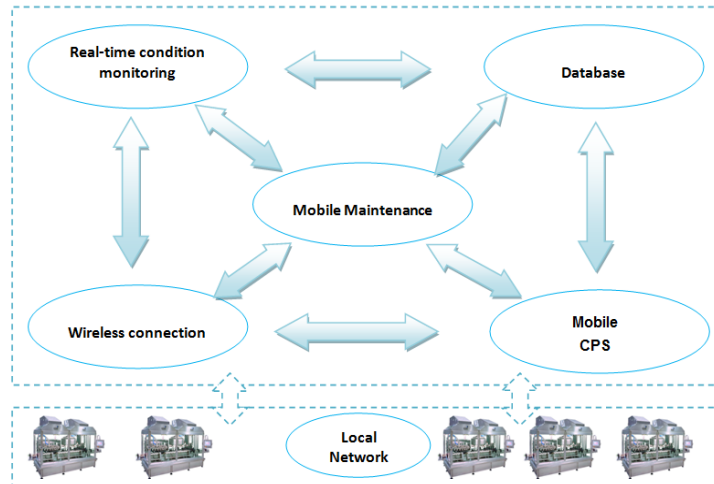


Figure 3 A framework of the proposed mobile maintenance system

The proposed architecture of mobile maintenance is shown in Figure 4. This three-layer architecture is based on mobile agent technology, web services and mobile CPS. The presentation layer is composed of a Graphical User Interface (GUI), which is managed by the IA. This layer can be a Smartphone. The Smartphone is more friendly and efficient than a laptop or a normal computer. The use of Smartphones represents one of different way of mobile computing. According to M. Nosrati et al. (2012), mobile computing means: being able to use a computing device while changing location properties. The Smartphone is a mobile phone built on a mobile Operating System, with more advanced computing capability. The Operating System in here is Android 4.4 (Kit Kat). Android is a Linux-based operating system developed by Google (Espiau, 2012). It is used in a large category of Smartphones and its major advantage resides in APIs (Application Programming Interfaces). These APIs are very comprehensive and very easy to access. Android is extremely portable, it adapts to a lot of structures, Smartphones, tablets, in the presence or absence of keyboards or trackballs and on different processors. Android applications are written in the Java programming language. They run within the Dalvik Virtual Machine, an open source technology.

The second layer is the application layer. It contains mobile agents, MAdA, MAA, MDA, and MCA. Web services are found in this layer. They serve as middleware since they are also the application software that is designed to support interoperability among the applications distributed across a network. The sending of messages from and to the client machines is facilitated by this technology.

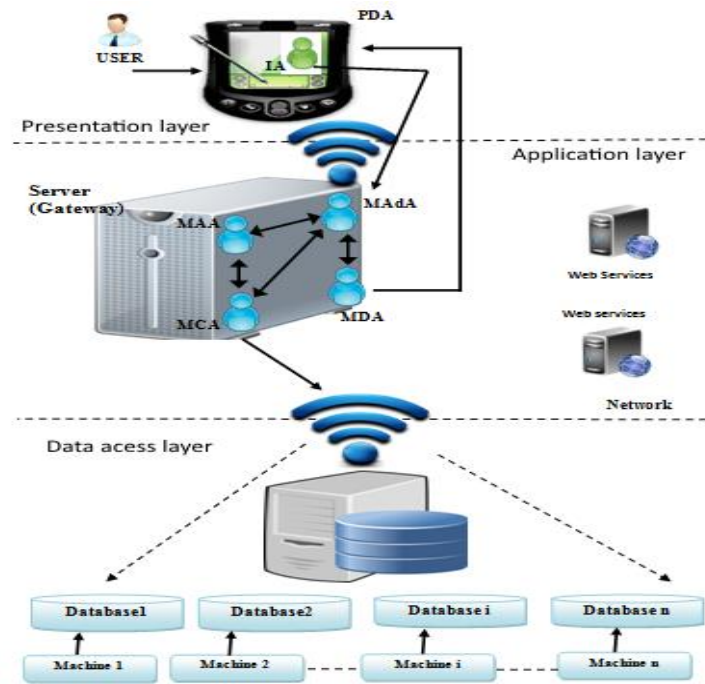


Figure 4 Layered architecture of the mobile maintenance framework

The last layer is the data access layer. All the machines are located in this layer; each machine has its own database. Each database stores the data entering into the system. The databases are characterized by various factors such as their ability to provide long term reliable data storage, multi-user access, concurrency control, query, recovery, and security capabilities. These factors are important in maintenance due to the need of gathering and storing data, or for example for the purpose of health monitoring machine.

The agents dispatched in this architecture try to accomplish CBM steps as shown in Figure 5. A set of mobile and stationary agents are implemented. One stationary agent is implemented, the Interface Agent (IA). The other agents are mobile: Mobile Collector Agent (MCA), Mobile Analyzer Agent (MAA), Mobile Administrator Agent (MAdA) and Mobile Diagnosis Agent (MDA). These agents must interact between each other to meet the overall objective as follows:

- First step. Data acquisition is realized by a MCA. This agent will move from one machine to another in order to collect information on the status of the machine, acquire real-time signals (vibration, current or temperature ...), and coordinate with the MAA and the MAdA. This step ensures real time monitoring.
- Second step. Data processing is realized by MAA. Through its analysis and interpretation, this agent makes collected data more understandable. The first phase of data processing is data cleaning. It ensures, or at least increases the chance, that cleaned data, i.e. error free data, are used for further analysis and modelling as described in (Jardine, 2006). Many factors can cause data errors by many elements including the human factor. The analysis in this phase is carried out through a comparison between the information in databases of the system and the information collected by the MCA on the current state of the system. This agent receives messages from MAdA and MCA. To realize data processing step, data and information are transmitted across a wireless network and stored in the database.
- Third step. Maintenance Decision Making is realized by MDA. This agent can apply two methods: current condition evaluation-based method (CCEB) and/or future condition prediction-based method (Shahrul and Rosmaini, 2012). FCPB and CCEB are two CBM decision making methods. The first one predicts the future trend of the equipment condition. The second one evaluates the current equipment condition.

The presented approach, propose to apply the CCEB method. Indeed, there is always an error rate in predicting, and without prediction results will be more realistic. In CCEB method, once the current related condition data are collected, the modelling process is carried out to estimate the actual equipment condition. This actual equipment condition will be compared to a predefined failure threshold. If the equipment condition level reaches or exceeds the threshold, the equipment will be called for maintenance. Otherwise, the equipment is assumed to be in good condition and can still be used. In most cases, this method is performed via periodic monitoring or inspection monitoring activities in order to collect current condition data. To display the result of this step, a technological support, as Smartphone, is required. To that end, IA is used. The IA is activated once the expert is connected from its Smartphone or its PDA. This agent checks the authentication of the expert,

activates MAdA, receives messages from MDA, and displays this information on the screen of the technological support.

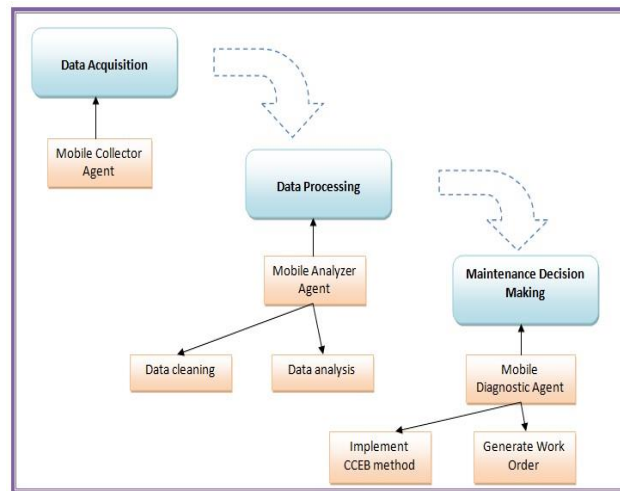


Figure 5 CBM steps in the proposed mobile maintenance approach

The MAdA specifies the location of the degraded machines. It sends the information to MAA and MDA. It is also responsible of (i) the creation of the other agents, MCA, MDA and MAA, (ii) the backup of information on the industrial system in the server database, and (iii) the communication with other agents in the system.

2. UML Representation

Agents are modelled using AUML (Agent UML) (Bauer, 1999) and MA-UML (Mobile Agent UML) (Loukil et al., 2006). AUML and MA-UML are the most complete agent-based extension of UML (Edgardo et al., 2003). These extensions provide concepts for the modelling of strong mobility, remote execution and cloning. The static modelling of the agents is represented by a MA-UML class diagram presented in Figure 6. The diagram has two super classes, “Mobile Agent” and “Stationary Agent”, and five subclasses; IA, MCA, MAA, MAdA and MDA. The “Mobile Agent” super class has five attributes. Machine information represents machine details as its name, its role ...etc. Knowledge represents a pool of logical sentences to infer conclusions about the production line. Itinerary is a guidebook for the agent. Host attribute stores the address of the machine. ID represents the order number of the machine. All attributes are private in order to make them only accessible for the class itself. Mobile agents need five operations to accomplish their tasks. Clone() operation allows to create new agents. MoveTo() operation permits displacement of agents between machines. Return() operation allows agents to return into the initial host. Send() operation sends information between agents. Receive() operation allows any agent to receive messages and information from another agent. The first two are protected in order to allow access by a class that inherits from the “Mobile Agent” class. The operations Return(), Send() and Receive() are public. MCA, MAA, MAdA and MDA inherit from the “Mobile Agent” super class.

“Stationary Agent” super class has two attributes: machine information and knowledge. These two attributes are the same as in the “Mobile Agent” super class. They are also private to make them only accessible for the class itself. The “Stationary Agent” super class has four operations. The protected Activate() operation allows this agent to activate MAdA at the first connection of the user. The three public operations, Send(), Receive() and Inform() have the same role as in the “Mobile Agent” super class. The only agent that inherits from this class is IA.

MA-UML specification categorizes associations as semantic relationships. The relationships in Figure 6 represent associations, i.e. semantic relationships that can occur between agents.

MAdA is activated by IA as shown in relation “Activate”. Clone relations allow MAdA to produce and to generate several MCA, MAA and MDA as needed in the system. The exchange of information between the agents of the system occurs with “Inform” relations as shown in the figure below.

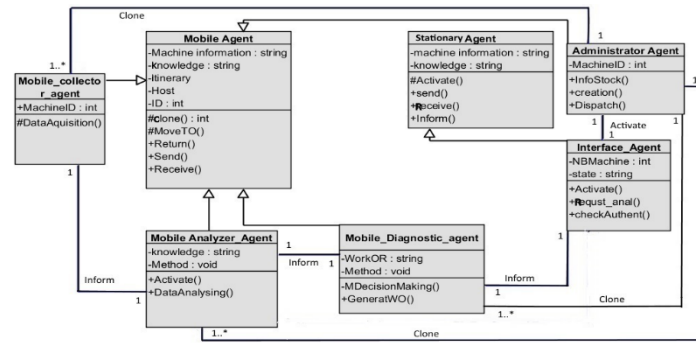


Figure 6 Class diagram of the agents operating in the mobile maintenance framework

The dynamic aspect of the agents is represented by a UML sequence diagram. In Figure 7, the diagram shows agents as objects, and their interactions arranged chronologically. It depicts the sequence of messages exchanged between the agents that are needed to carry out the functionality of the scenario. The vertical dimension of the diagram shows the sequence of messages/calls in the chronological order in which they occur. The horizontal dimension shows the object instances which the messages are sent to. After user authentication, IA is activated. MAdA is activated by IA in its turn. MAdA clones the MAA and MCA. MCA is dispatched to the host machine in order to wait for certain types of information to become available. The interaction between the agents is performed by the FIPA-ACL protocol (Agent Communication Language). This protocol is a proposed standard language for agent communications. It relies on speech act theory and defines a set of performatives, also called Communicative Acts, and their meaning (Diagnum and Greaves, 2000).

After the information collection, MCA sends the information to MAdA using ACL protocol. The latter activates the MAA which makes its analysis and returns the information to MAdA. In case of failure, MDA is created in order to make a decision. Otherwise, the user is informed that the system is working properly.

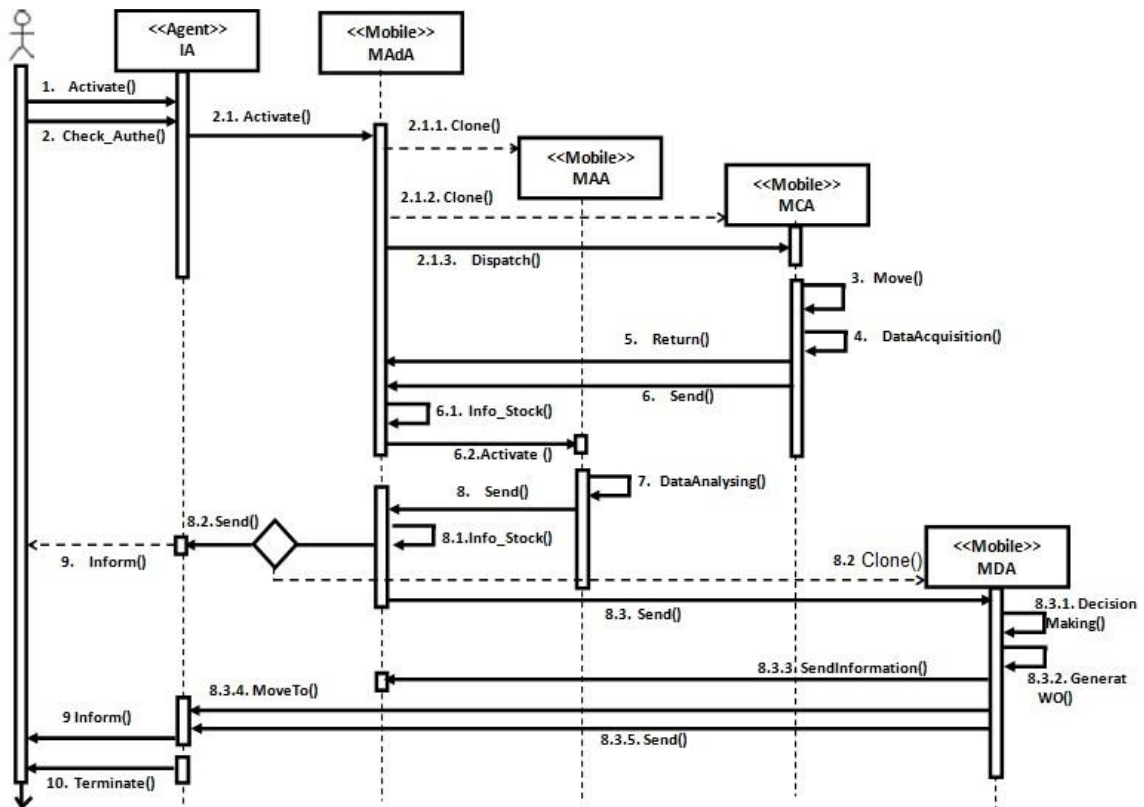


Figure 7 A sequence diagram of the interactions between Agents in the mobile maintenance framework

3. Agents Modelization

Each agent in the system is modelled as a token, an ordinary Petri net, in the Reconfigurable Object Net. At the token level, token-nets describe the behavior of the agents, and token-rules describe the structural transformations on token-nets. The system level describes the whole system where all the agents coexist, move,

and communicate. In the system level, places can be “net-places” or “rule-places”. Net-places contain token-nets. Rule-places contain token-rules. Figure 8, Figure 9 and Figure 10 illustrate the formal model under RON-tool.

Figure 8 illustrates the token-rule used in the transformations necessary to establish communication connections between agents. The figure consists mainly of three editor panels, one for the Left-Hand Side (lhs), one for the Right-Hand Side (rhs) and one for a Negative Application Condition (nac 1). The rule in this figure shows the gluing of the two object nets (token-nets): MCA and MAdA illustrate in lhs. This gluing will allow these two agents to communicate. This communication will be achieved by adding two transitions: send and receive and a place: Info shown in rhs.



Figure 8 Token rule of agent communication

The formal model of communication between MAdA, MCA and MAA is represented on Figure 9. Agents are represented by a square. Each agent has several places, denoted by P, and transitions, denoted by T. Transition T7 corresponds to send in MCA and transition T9 corresponds to receive in MAdA, whereas the place P12 represents intermediary place, used when exchanging information occurring between the agents. A so-called negative application condition (NAC) forbids the application of the rule if send and receive transitions already exists.

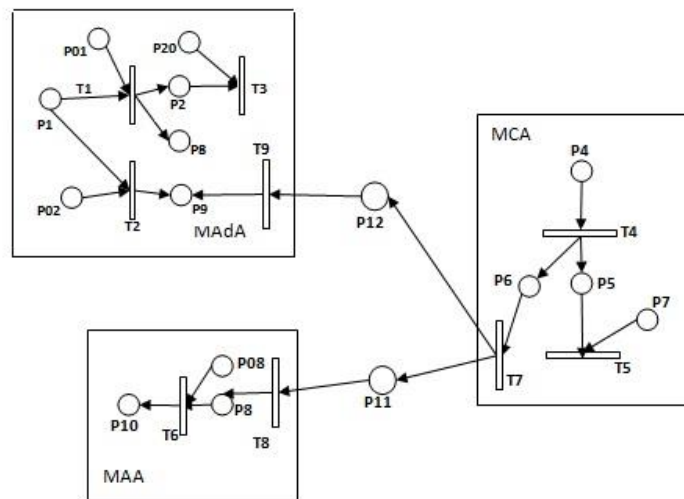


Figure 9 Formal model of communication between agents with RON formalism

Figure 10 shows the distributed system, in RON editor, where several agents may interact with each other. The blue spheres with “O” represent different object nets of the system. The green ones with “R” represent the different rules to be applied. The entire gray box with different icons shows the transitions.

For agent’s interactions, a MAdA object net can be merged with a MCA object net by firing the “Orientation” APPLYRULE system transition, as described in Figure 8. Note that the Orientation system transition controls which pair of agent will interacts.

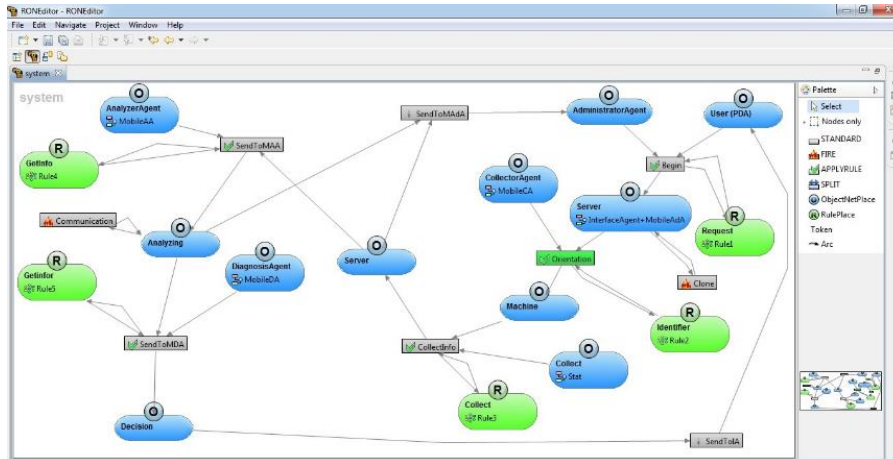


Figure 10 The whole agents under RON-tool in communication phase

More details on the formal verification can be found in the work of Abid et al. (2013).

4. Evaluation

This section explores the studied production line, its formal validation to performance study and its implementation under LabVIEW.

1. The Production Line

The juices production chain acquires important production machines, which are a great investment in the plant. Production takes place in a well-established production program. The studied chain consists of eight machines where each machine has its own inputs and outputs, modeled in Figure 11.

The unpacker machine puts the cans one after another on the conveyor in the direction of the next machine. To work, this machine needs only the cans. The second machine is the filling machine. It fills the entering cans with juice previously mixed, puts the closure and sends the cans onto the second conveyor in order to transmit it to the next machine. To accomplish this task, this machine needs the caps and the juice "product". The next step is the pasteurization. The role of the pasteurizer machine is to reduce the number of viable pathogens, using a high temperature. This machine needs the closed cans, water and a heater. In the scanner machine, two cameras rotate in order to ensure that the cans are not damaged. The date stamp machine puts the date and the hour on the cans. The two shrink wrapper machines require glue, cardboard and film to accomplish their tasks. These machines apply heat on cardboard and film to pack the final product. The first one wraps six cans and the second one wraps twenty four cans.

This production line system is well suited for our case study. Some machines exhibit interesting features. Indeed, they have several inputs corresponding to parameters in this study, and some of them are easy to control. This is the case of the pasteurizer machine which has three types of inputs: temperature, water and product. These parameters play an important role in the pasteurization process. They have a direct impact on the quality of the final product.

The implementation of the case study uses data from the pasteurizer machine. All the cans need to be pasteurized at different steps. The machine needs to heat the cans at the beginning and the end of pasteurizing to 47 °C for a few minutes. In the middle phase, the temperature is increased between 85 - 95 °C to kill the micro-organisms.

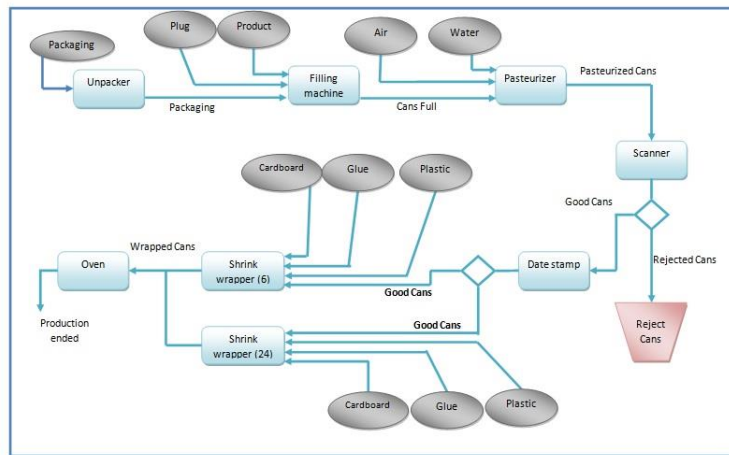


Figure 11 The studied juice production line

2. Implementation with LabVIEW

National Instruments (2014) defines LabVIEW as "... a highly productive development environment for creating custom applications that interact with real-world data or signals in fields such as science and engineering".

LabVIEW allows to program and to present the automata of the selected machine. The process can be followed through visual inspection of the different steps. The visualization is essential in analyzing complex data sets. All the data can be stored from LabVIEW into the databases. The MCA has access to these data and the scenario will be completed as described in section 3.

The connection between LabVIEW and the Siemens automata can be achieved through several protocols. The first one is OPC (OLE for Process Control) industry standard. It provides a real plug- and-play software technology for process control and factory automation. The second one is TCP/IP. TCP/IP functions are implemented into LabVIEW to control the way of transmitting information reliably across the network.

LabVIEW programs are called Virtual Instruments (VIs). Each VI has three components, a block diagram that contains the graphical source code, a front panel which is built using controls and indicators controls representing the inputs and indicators representing the outputs, a connector panel that represents the VI in the block diagrams of other VIs. The graphical G Programming Language is central to LabVIEW. It allows to quickly tie together data acquisition, analysis and logical operations. This language has several advantages (National Instrument, 2014) as intuitive and flowchart-like dataflow programming model, shorter learning curve than traditional text-based programming, naturally represents data-driven applications with timing and parallelism.

Figure 12 presents the VI of the automata corresponding to the pasteurizer machine. The start button must be used to warm the water in the tanks if they are full. The VI contains four tanks of pasteurizer operations. Indicators let the user set the temperature of each tank. The thermometers will increase step by step to indicate the temperature of the water in each tank. In this figure, the process has not started. The user must click on the valve sequentially in order to fill the tanks. Figure 13 shows how the tanks are filled. Each tank must have between 2000 and 2200 liters represented in the programmed software by 20 and 22 units. As soon as the water level reaches 20 units, the valve is closed (shown in red), and the water level in the next tank begins to increase, and so on until all four tanks are filled. If the tanks are empty, the heater does not work and the temperature cannot increase, as shown on the left side of Figure 14. Otherwise, the heater works, and the water is warm as represented on the right side of Figure 14. If the heater does not work well, the state lamp will light up red and the data are stored in the database. The programmed software is done with real parameters corresponding to the pasteurized machine.

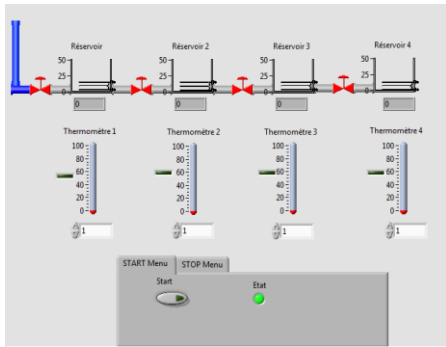


Figure 12 The VI of Pasteurizer machine

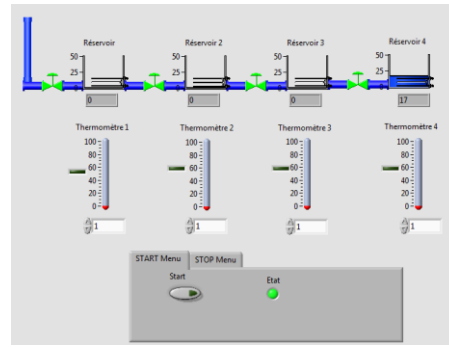


Figure 13 The filling of the tanks

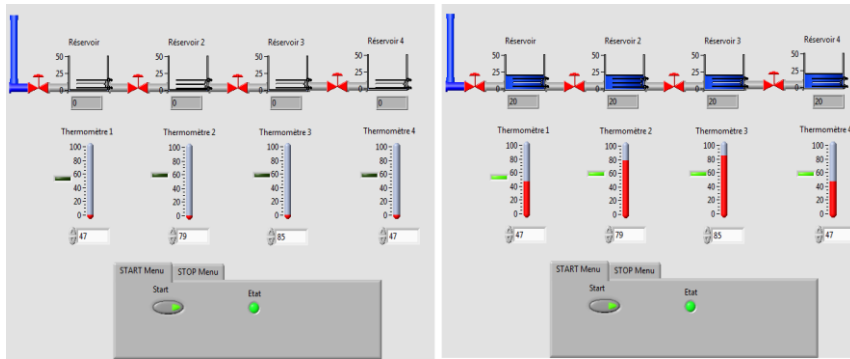


Figure 14 The temperature is defined but the heater doesn't work and temperature doesn't change

Figure 15 shows a part of the block diagram. Later on, all the objects placed on the front panel will appear on it as terminals. The block diagram contains structures and functions which perform operations on controls and supply data to indicators. It also contains the graphical source code of the LabVIEW program. The figure shows how the indicators of the temperature are linked to the commands. When the value of the command is changed, the indicator will show the new defined one. Conditional statement is shown in the figure. For example, in the case of true (vrai), i.e. when the start button is activated, the command of the temperature as “Température désirée 3” in the left side of the figure will be linked to the indicator “T. Désirée 3”. Otherwise, in the false case (faux), the commands and the indicators are not linked and the value will not be changed.

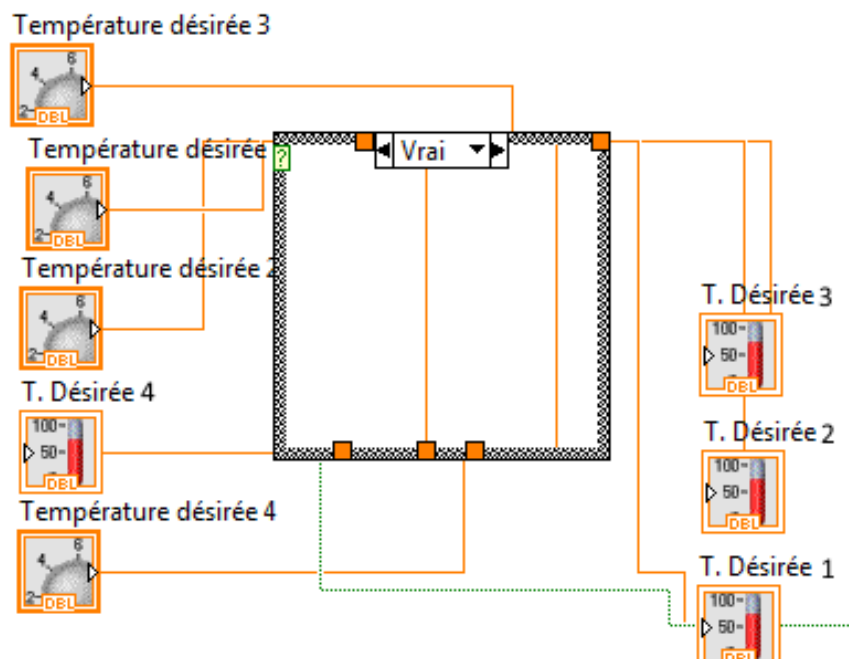


Figure 15 Temperature block diagram

3. Results and Discussion

In the studied production line, when all the machines are in good condition the production line produces in one hour:

- 15 pallets
- 1200 cardboards
- 28800 cans

In one day:

- 120 pallets
- 9600 cardboards
- 230400 cans

In case of dysfunction in a period of 10–15min, the production line produces, in one hour:

- 12 pallets
- 960 cardboards
- 23040 cans

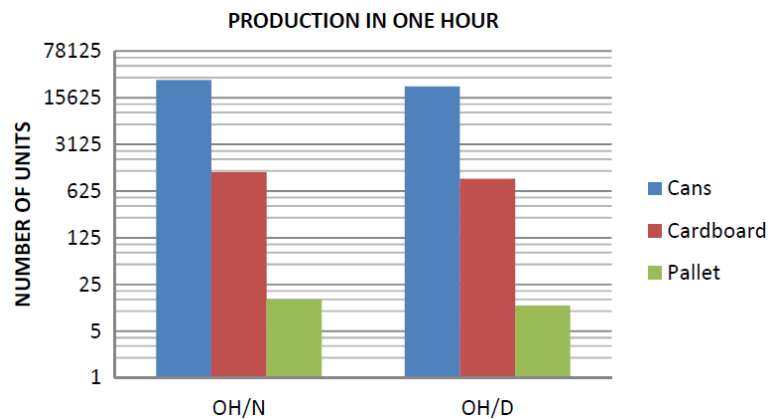


Figure 16 Production in the normal case and the dysfunction case

Table 1 Label of the chart

Acronym	Interpretation
OH/N	One Hour in Normal case
OH/D	One Hour in Dysfunction case (10–15min)

According to the production manager of the juices plant in Algeria; this method of maintenance increases production with 18% per day.

It is clear, from Figure 17, that this method of maintenance increases the production.

Production of cans (blue pyramid) has increased from 230400 cans/day to 271872 cans/day, which has increased the number of cardboards (red pyramid); from 9600 cardboard/day to 11328 cardboard/day and pallets (green pyramid) from 120 pallets/day to 140 pallets/day. In one day 41472 cans are produced.

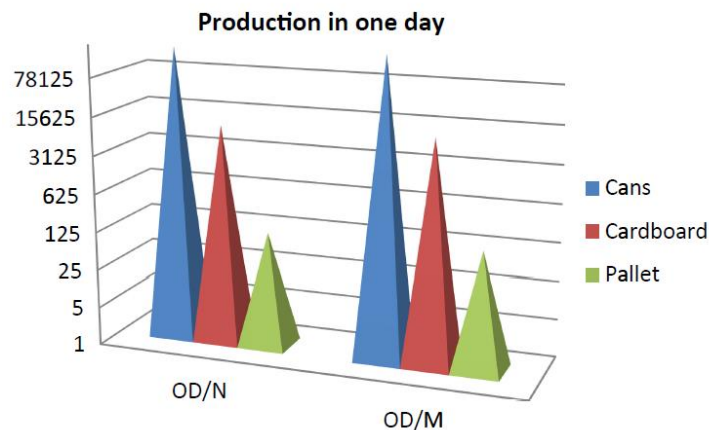


Figure 17 Result after method of maintenance

Table 2 Label of the chart

Acronym	Interpretation
OH/N	One Hour in Normal case
OH/D	One Hour in Dysfunction case (10–15min)

5. Conclusion

This work, propose an approach based on mobile agents technology and mobile CPS for mobile maintenance (m-maintenance). Also, general m-maintenance architecture is proposed in this paper.

M-maintenance is a new generation of maintenance implemented using advanced technologies. It aims to find solutions of several problems in industrial sector as quality and cost issues. In this way, maintenance strategies are studied in order to select the best one: CBM. CBM is preventive maintenance strategy that tries to avoid and eliminate unnecessary tasks. This type of maintenance considerably reduces costs by minimizing the number of unnecessary operations programs.

The proposed approach tries to benefit from the advantages of intelligent agents where, a stationary agent is implemented on the mobile CPS to execute their tasks. CBM steps are implemented by mobile agents assigned on three layers architecture. The adoption of this architecture meant to obtain a secure data access where the end user does not have direct access to the databases. The web services paradigm is also adopted to ensure the interoperability between different software applications distributed across a network.

MA-UML standard is chosen to model the structure of agents and sequence diagram for the interactions between them. To formally verify the mobility of the agents, RON high level Petri net is used. Furthermore, a juice production line system is taken as a case study. The implementation results of the studied system using LabVIEW environment are provided.

The combination of the different technologies, mobile CPS, CBM, mobile agent, web services, multi-layer architecture is the corner stone of our proposal.

This research work can open several valuable perspectives:

1. Integrate cloud computing where shared resources, data and information of factories are provided on-demand.
2. Use Internet of Things to have advanced connectivity of devices, systems, services and machines.
3. Use wireless mobile ad hoc networks. They are self-configuring and dynamic networks in which nodes are free to move.
4. Introduce various aspects for continuous improvement of the production chain, and more widely of the supply chain like sustainability (Savino et al., 2013; Savino et al., 2014; Savino et al., 2015), quality management (Savino et al., 2009), as well as batch scheduling that can be impacted by maintenance activities (Savino et al., 2010).

References

Abid K, Kahloul L, Mouss L.H, Kazar O, (2013), “Formal Specification of a Mobile Agent Based Maintenance for Manufacturing Systems”, Proceeding of 7th International Workshop on Verification and Evaluation of Computer and Communication Systems, Italy.

Ajmone Marsan M, Balbo G, Conte G, Donatelli S, Franceschinis G, (1994), “Modelling with Generalized Stochastic Petri Nets”. ISBN: 0471-93059-8.

Bauer B, (1999), “Extending UML for the Specification of Interaction Protocols”, Proceeding of 6th Call for Proposal of FIPA.

Belloni E, Marcos C, (2003), “Modeling of Mobile-Agent Applications with UML”, Proceedings of the Fourth Argentine Symposium on Software Engineering, JAIIO (Jornadas Argentinas de Informtica e Investigacin Operativa), pp 3-13.

Biermann E, Ermel C, Hermann F, Modica T, (2007), “A Visual Editor for Reconfigurable Object Nets based on the ECLIPSE Graphical Editor Framework”, Proceeding of 14th Workshop on Algorithms and Tools for Petri Nets, GI Special Interest Group on Petri Nets and Related System Models, pp. 2-7.

Bucci G, Carnevali L, Ridi L, Vicario E, (2010), “Oris: a tool for modeling, verification and evaluation of real time systems”, International Journal on Software Tools for Technology Transfer, vol. 12, n° 5, pp. 391-403.

Buse D.P, Feng J.Q, Wu Q.H, (2003), “Mobile agents for data analysis in industrial automation systems”. Proceedings IEEE/WIC International Conference on Intelligent Agent Technology, pp. 60-66, Canada.

- Campos J, Jantunen E, Prakash O, (2009), "A web and mobile device architecture for mobile e-maintenance", *The International Journal of Advanced Industrial Technology* vol. 45, n° 1, pp. 71-80.
- Campos J, Jantunen E, Prakash O, (2007), "Development of a maintenance system based on Web and mobile technologies", *Journal of International Technology and Information Management*, vol. 16, n° 4, pp. 1-18.
- Campos J, (2009), "Development in the application of ICT in condition monitoring and maintenance", *Computers in Industry Journal*, vol. 60, n° 1, pp. 1-20.
- Chandra S, Kumar C, Kumar S, Tiwari M, (2008), "Development of an intelligent agent based AGV controller for a flexible manufacturing system", *International Journal of Advanced Manufacturing Technology*, vol. 36, n° 7, pp. 780-797.
- Du TC-T, Wolfe PM, (1997), "Overview of emerging database architectures", *Computers Industrial Engineering*, vol. 32, n° 4, pp. 811-821.
- Diagnum F, Greaves M, (2000), "Issues in agent communication", ISBN: 3-540-411.
- Dingle N.J, Knottenbelt W.J, Suto T, (2009), "PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets", *Acm Sigmetrics Performance Evaluation Review*, vol. 36, n° 4, pp. 34-39.
- Espiau F, (2012), "Créez des applications pour Android", ISBN: 979-10-90085-37-4.
- Fumagalli L, Macchi M, (2015), "Integrating maintenance within the production process through a flexible E-maintenance platform", *Proceeding of 15th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2015)*, vol. 48, n° 3, pp. 1457-1462.
- Fletcher M, Brennan R.W, Norrie D.H, (2003), "Modeling and reconfiguring intelligent holonic manufacturing systems with Internet-based mobile agents", *Journal of Intelligent Manufacturing*, vol. 14, n° 1, pp. 7-23.
- Hoffmann K, Ehrig H, Mossakowski T, (2005), "High-Level Nets with Nets and Rules as Tokens", *Proceeding of 26th international conference on Applications and Theory of Petri Nets*, pp. 268-288.
- Iung B, (2003), "From remote maintenance to MAS-based e-maintenance of an industrial process", *Journal of Intelligent Manufacturing*, vol. 14, n° 1, pp. 59-82.
- Jardine A.K.S, Daming L, Dragan B, (2006), "A review on machinery diagnostics and prognostics implementing condition-based maintenance", *Mechanical Systems and Signal Processing Journal*, vol. 20, n° 3, pp. 1483-1510.
- Jing R.L, Li P.K, Shu B.T, (2005), "An object-oriented intelligent disassembly sequence planner for maintenance", *Computers in Industry*, vol. 56, n° 7, pp. 699-718.
- Kezunovic M, Xu X, Wong D, (2002), "Improving Circuit Breaker Maintenance Management Tasks by Applying Mobile Agent Software Technology", *Transmission and Distribution Conference and Exhibition: Asia Pacific, IEEE/PES*, vol. 2, pp. 782-787.
- Kim D.N.T, (2012), "A Practical Approach to Failure-based Maintenance Management with ERP Software", *International Journal of Operations and Quantitative Management*, vol. 18, n° 4, pp. 269-280.
- Kuo C-H, Huang H-P, Muder J, Li-Der J, (2005), "Separation model design of manufacturing systems using the distributed agent-oriented Petri net", *International Journal Computer Integrated Manufacturing*, vol. 18, n° 2-3, pp. 146-157.
- Lange D.B, Mitsuru O, (1999), "Seven good reasons for mobile agents", *Communications of the ACM*, vol. 42, n° 3, pp. 88-89.
- Lee J, (1997), "Strategy and challenges on remote diagnostics and maintenance for manufacturing equipment", *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 368-370.
- Lee J, (2001), "A Framework for Web-Enabled E-Maintenance Systems", *Proceedings Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, pp. 450-459.

Loukil A, Hachicha H, Ghedira K, (2006), "A proposed Approach to Model and to Implement Mobile Agents", *IJCSNS International Journal of Computer Science and Network Security*, vol. 6, n° 3, pp. 125-129.

Mangina E, McArthur S.D.J, McDonald J.R, (2000), "Autonomous agents for distributed problem solving in condition monitoring", *Lecture Notes in Computer Science*, vol. 1821, pp. 683-692.

Mobley R.K, (1990), "An Introduction to Predictive Maintenance", Van Nostrand Reinhold, New York, ISBN: 9780750675314.

Muller A, Marquez A.C, Iunga B, (2008), "On the concept of e-maintenance: Review and current research", *Reliability Engineering and System Safety*, vol. 93, n° 8, pp. 1165-1187.

Münch T, Hladik J, Salmen A, (2014), "Collaboration and Interoperability within a Virtual Enterprise Applied in a Mobile Maintenance Scenario", DOI: 10.4018/978-1-4666-5142-5.ch007.

National Instrument, (2014), <http://www.ni.com/newsletter/51141/en/>, February 2014.

Newcomer E, (2002), "Understanding Web Services: XML, WSDL, SOAP, and UDDI", Addison Wesley Professional", ISBN: 0201750813.

Nosrati M, Ronak K, Hasanvand H.A, (2012), "Mobile Computing: Principles, Devices and Operating Systems", *World Applied Programming*, vol. 2, n° 7, pp. 399-408.

Piateski G, Frawley W, (1991), "Knowledge Discovery in Databases", ISBN: 0262660709.

Rasovska I, (2006), "Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas : Application au diagnostic dans une plateforme d'e-maintenance", PhD thesis, University of Franche Comté.

Sahai A, Morin C, (1998), "Mobile agents for enabling mobile user aware application", *Proceedings of the 2nd International Conference on Autonomous agents*, pp. 205-211.

Savino M, Sekhari A, (2009), "A quality management system based on fuzzy quality pointers in ISO 9000", *International Journal of Product Development*, vol. 8, n° 4, pp. 419-430.

Savino M, Mazza A, (2014), "Toward Environmental and Quality Sustainability: An Integrated Approach for Continuous Improvement", *IEEE Transactions on Engineering Management*, vol. 61, n° 1, pp. 171-181.

Savino M, Manzini R, Mazza A, (2015), "Environmental and economic assessment of fresh fruit supply chain through value chain analysis, A case study in chestnuts industry", *Production Planning and Control*, vol. 26, n° 1, pp. 1-18.

Savino M, Ouzrout Y, Mazza A, De Chiaro S, (2013), "The sustainability of product compliance within lifecycle design", *International Journal of Product Lifecycle Management*, vol. 6, n° 3, pp. 288-309.

Savino M, Meoli E, Ming L, Wong M, (2010), "Dynamic batch scheduling in a continuous cycle-constrained production system", *International Journal of Services Operations and Informatics*, vol. 5, n° 4, pp. 313-329.

Shahrul K, Rosmaini A, (2012), "An overview of time-based and condition-based maintenance in industrial application", *Computers & Industrial Engineering Journal*, vol. 63, n° 1, pp. 35-149.

Vatn J, Hokstad P, Bodsberg L, (1996), "An overall model for maintenance optimization", *Reliability Engineering & System Safety*, vol. 51, n° 3, pp. 241-257.

Wang W, Tse P.W, Lee J, (2007), "Remote machine maintenance system through Internet and mobile communication", *International Journal of Advanced Manufacturing Technology*, vol. 31, n° 78, pp. 783-789.

Wang X, Aiguo J, Sheng W, (2005), "Mobile Agent Based Wireless Sensor Network for Intelligent Maintenance", *Advances in Intelligent Computing*, vol. 3645, pp. 316-325.

Weiming S, Qi H, Hyun JY, Douglas N, (2006), "Applications of agent-based systems in intelligent manufacturing: an updated review", *Advanced Engineering Informatics*, vol. 20, n° 4, pp. 415-431.

Wernz C, Deshmukh A, (2007), “Decision strategies and design of agent interactions in hierarchical manufacturing systems”, *Journal of Manufacturing Systems*, vol. 26, n° 2, pp. 135-143.

Wireman T, (1990), “World Class Maintenance Management”, Industrial Press, New York, ISBN: 0831130253.

About Authors

Khadija Abid is a PhD student at Batna 2 University, Batna, Algeria. She received the Master degree in 2010 at Biskra University, Biskra, Algeria, working on artificial intelligence field. She participates as a member of program committee in international conferences. She is interested and working in artificial intelligence field and multi-agents systems with their applications and also advanced information systems, manufacturing systems, formal modeling.

Ikbal Arab-Mansour received a PhD in Industrial Informatics from the University of Nancy1, France in 1998. From 2002 to September 2013, she was Associate Professor at the University of Aleppo-Faculty of Informatics-Syria. Since October 2013, she is contract academic teacher/researcher at National Institute of Applied Sciences (INSA) Lyon-Computer Science Department (LIRIS Laboratory), Industrial Engineering Department (DISP Laboratory) and Preparatory Level Department (LIRIS Laboratory - SOC Team). Ikbal Arab-Mansour has mainly worked on Business-IT alignment, Information Systems, SOC/SOA, Modelling and Simulation.

Chantal Bonner-Cherifi received her PhD in computer science from Corsica University, France, in 2011. She is working as an associate professor at the DISP laboratory, University of Lyon 2, France. Her current research interests focus on Information Systems Agility with SOA architectures, Smart Cities and Network Science. She serves as program committee member in several international conferences. She has published about 30 international refereed conference and journal papers.

Leila Hayet Mouss received the B.Sc. degree in Electrical Engineering, in 1979, from the National Polytechnic School of Algiers, Algeria; the M.Sc. degree in Electrical and Computer Engineering, in 1982, from the ENSERB, France; and finally the Ph.D. degree also in Electrical and Computer Engineering, in 1985, Bordeaux University. She joined the University of Batna, where she is an Associate Professor of Electrical and Computer Engineering. She is the head of Automatic and Computer Integrated Manufacturing Laboratory. Her current research interests include industrial Diagnosis of production system using the artificial intelligence techniques in the LAP (Laboratoire d’Automatique et Productique).

Okba Kazar was born at Biskra (Algeria) in 1962. He obtain his magister degree in 1997 from the Constantine University (Algeria) working on artificial intelligence field. He obtained his PhD degree from the same university in 2005. He is member of editorial board of some Journals. He is author of some publication in international journals and session chair in international conferences, he also publish a book “Manual d’Intelligence artificielle”. He is interested to the multi-agents systems and their applications, ERP, advanced information systems, Web services, semantic Web, bigdata, internet of things and cloud computing. Actually Okba Kazar is full professor at computer science department of Biskra University and director of intelligent computer science laboratory (LINFI).