A Component Based System for S-maintenance

Mohamed-Hedi Karray
AS2M
Femto-st Institute,
Besançon, France
Hedi.karray@femto-st.fr

Brigitte Chebel-Morello
AS2M
Femto-st Institute,
Besançon, France
Brigitte.morello@femtost.fr

Christophe Lang
LIFC
Franche-comté University,
Besançon, France
christophe.lang@univ-fcomte.fr

Noureddine Zerhouni
AS2M
Femto-st Institute,
Besançon, France
Noureddine.zerhouni@femtost.fr

***Abstract*-- Thanks to ICT, Web emergency and Internet, the achievement of maintenance services and monitoring can be performed automatically, remotely and through various distributed information systems. Hence the emergence of the concept of services offered through maintenance architectures, ranging from autonomic systems to integrated systems where knowledge management, cooperation and collaboration are vital to any operation. Into this context, new services like intelligent maintenance, self maintenance, etc are required. To this end, a new concept called s-maintenance is emerged. This concept defines a new generation of maintenance systems founded on a knowledge based system. While existing systems don't respond to the characteristics of this new generation of systems, we design in this paper an architecture of a maintenance component based system respecting the characteristics of s-maintenance. Each component in the system is defined to respond to one or many characteristics of this concept.**

Keywords- s-maintenance; e-maintenance; component based system; knowledge based system.

I. INTRODUCTION

The role of maintenance is strategic in the industrial environment. Hence, while having this interest in company; industry has been seeing the developments of different generations of maintenance systems. New technologies of information and communication technologies (ICT) have helped to establish and evolve these systems [1]. E-maintenance and s-maintenance are considered as the most important resulting concepts that have been developed and applied in the industry. A variety of definitions of e-maintenance are presented in the literature and summarized in [1]. Building on this diversity of work, we propose a definition to the concept of e-maintenance encompassing the entire maintenance process, as well the e-maintenance system. Hence, we define e-maintenance as the carrying out of maintenance where the all technical, administrative, and managerial actions or activities interact and collaborate electronically, using network or telecommunications technologies. Regarding the e-maintenance system, we define it as a collaboration system offering a set of software components and software services of maintenance support (integrated and/or distant) all that enable maintenance actors to find each other and the information they need and to be able to communicate and work together to achieve maintenance process via a variety of devices .

Vis-à-vis s-maintenance, it comes as an evolution in maintenance systems extending the concept of e-maintenance and transforms the maintenance system from an integration tool to the core of the maintenance process.

This concept is presented by Rasvoska et al [2] as a concept enlarging e-maintenance. Then in a previous work, we presented this work as a new generation of maintenance system focusing in semantic interoperability [3]. Then, a more generic definition is provided in [4] extending the s-maintenance concept to be more adaptable to the new industrial needs [21]. So, s-maintenance concept is defined as the carrying out of maintenance based on the domain expert knowledge, where systems in the network manage this knowledge (formalization, acquisition, discovery, elicitation, reasoning, maintenance use and reuse) and share the semantics to emerge new generation of maintenance services (as self-X services, Worker self-management, improvement of various types of maintenance, new collaboration methods, etc.) while including e-maintenance characteristics." When we speak about characteristics we mean what distinguish this concept according to its definition. In the other hand, s-maintenance system is a based knowledge distributed and collaborative system providing self-management services [5] to the overall maintenance process. It is a self learning system [6] having the possibility to evolve its intelligence degree, including computational intelligence and knowledge [7]. Based on semantics of the domain experts knowledge, it is a scalable, adaptable and contextual system (semantic interoperability, user oriented, intelligent HMI, processes orchestration, tracking users interactions, new process generation).

This concept emerges from the aggregation of the power of knowledge engineering and the integration of new aspects into the maintenance scope. The strong point of this concept is that the knowledge management system presents the core of the information system and allows manipulating the expertise knowledge of the domain (see figure1).

The s-maintenance system must ensure a high level of knowledge acquisition (based on self learning and tracking), elicitation, reasoning and re-use. This level enables more advanced predictive maintenance decision making basing on analytical reliability models, enables the achievement of intelligent maintenance and enables the automation of the maintenance process based decision making approach.

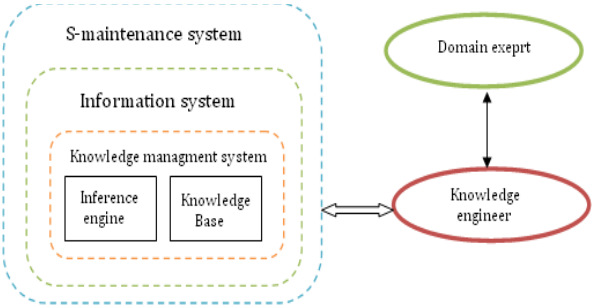


Figure1. S-maintenance system

Several maintenance systems have been developed and are in use today (ENIGMA, CASIP, ICASAME, Remote Data Sentinel, INTERMOR, INID, IPDSS, WSDF, MRPOS, PROTEUS, TELMA, etc.). These systems are a result either of the industrial world or of the academic one. Muller et al classify them as: proprietary platforms (i.e. ICAS), platforms developed within projects (i.e. PROTEUS) or platforms for research and education (i.e. TELMA). For more details about these systems, please refer to the survey presented in [1]. When analyzing these systems according to the s-maintenance definition, we note the lack between the characteristics of s-maintenance and what is provided by these systems. Most of these systems are built respecting some of e-maintenance definition. So, after a thorough analyze, we conclude the absence of a system responding to the s-maintenance characteristics and respecting the s-maintenance definition.

From the different methods of software engineering, Component Based System (CBS) seems the best candidate to model and set up an s-maintenance system respecting its definition given the philosophy of CBS which is integration centric with a focus on selecting components that match stakeholder requirements [8].

The computer-based system consists of all components necessary to capture, process, transfer, store, display, and manages information. Components include software, processors, networks, buses, firmware, application-specific integrated circuits, storage devices, and humans (who also process information) [9].

Hence, we propose in this paper a CBS for s-maintenance. There are different methods, approaches and languages to design and describe a CBS like summarized and proposed in [10], [11], [12] and [13], but while our system is specific to the concept of s-maintenance, we don't adopt any of these approaches. We just adopt a simple approach containing two phases, pre-analyze phase concerning the identification of the interesting components mach the s-maintenance requirements; and the second phase concerns the modeling of the system architecture by defining relations as well as interactions between these components.

The rest of this paper is organized as follow. Section 2 summarizes some related works on maintenance systems. Section 3 presents our proposed CBS architecture for s-maintenance. Section 4 discusses our proposal as well as validation and some perspectives. Finally, Section 5 concludes the paper and sets guidelines for future work.

II. OVERVIEW ABOUT EXITING SYSTEMS

After a thorough study on maintenance systems developed in the academic setting and in research projects based on the review presented in [1], we identified two systems developed respectively by Zhang and Cao that we consider as the closest systems to the definition of s-maintenance system. These systems don't perfectly ensure s-maintenance characteristics.

Zhang proposed a multi-agent system for e-maintenance based on knowledge management. This multi-agent system enables the exchange of knowledge/information between industrial automation systems [14]. All sources of information in the industrial automation systems are analyzed and modeled by agents. The relationships between these sources are highlighted in the context of maintenance, including proactive maintenance.

The system architecture consists mainly of a AMC (Agent Management Core) and four types of agents namely Knowledge Agent (KA), Configuration Agent (CA), Diagnostic Agent (DA), Field Agent (FA) and management agent (MA).

From an s-maintenance point of view, this system is consistent with some characteristics of this concept such as semantic interoperability and exploiting knowledge engineering. But the main issue of this concept (i.e. the dynamics of the services provided by the system) is not treated in this work. Moreover, the crucial point is that the AMC the most important component of the system is not knowledge oriented because it is not based on domain knowledge. Hence, this point let us wonder on its operation as component managing the creation and deletion of agents as well as the integration of new applications in the system.

In the other side, Cao et al develop the IIEMD (Integrated Intelligent Equipment Maintenance Decision) system for intelligent, distributed and cooperative Maintenance that provides a set of decision support tools for various maintenance activities [15]. IIEMD is a hybrid system supported by an ontology where it operates two types of logical reasoning that are the RBR (Rules Based Reasoning) [16] and CBR (Case Base Reasoning) [16] for the service of equipment maintenance decision-support in order to improve the performance of general maintenance process. IIEMD system is based on a Service Oriented Architecture (SOA) [17]. In this architecture, decision tools for maintenance (defined as Web services) can store, retrieve information from other tools and distribute information to other tools integrated into the system. This system integrates a knowledge base including an ontology and an inference engine.

We note in this work, the presence of some basic elements of s-maintenance characteristics. Firstly, this system is based on knowledge engineering. Thus, although it is not mentioned that the ontology was also used for the

benefit of semantic interoperability, this is obvious since web services used in the SOA handle the ontology concepts. Compared to the s-maintenance characteristics, this system uses the knowledge engineering services for decision making in maintenance (intervention, repair) and not include all activities in the maintenance process (as diagnosis, prognosis, optimization of resources, planning) although it can integrate these systems thank to the SOA. But, it is also clear the lack of other basic elements of s-maintenance as the self-learning and especially the generation of new services; it addresses only decision support services. In conclusion we can say that this platform is closest to the envisaged s- maintenance system but many aspects are neglected.

So after a brief analysis of existent systems, it seems clear that the overall characteristics of s-maintenance are not insured by these systems. This is understandable given the difference between the objectives of the initiated projects and the adopted concept. These systems are built to create e-maintenance system and try to respect s-maintenance characteristics. We remember that e-maintenance is about the integration of various systems of maintenance field and s-maintenance is about exploitation of existing knowledge in the system to emerge new services of maintenance while also including the integration aspect.

III. PROPOSED SYSTEM

We propose an architecture of a maintenance component based system respecting the characteristics of s-maintenance concept. The design approach is composed in two phases. In a first time, pre-analyze phase, we nominate the interesting components which must be present in the system. Then, in a second time, modeling phase, we design the system's architecture by defining relations as well as interactions between these components.

A. Pre-analyze

While a system is a collection of interrelated components that work together to achieve some objective, CBS success depends on the ability to select suitable components. An inappropriate component selection can lead to adverse affects such as short listing components that barely fulfill the needed functionality or they introduce extra costs in integration and maintenance phases [8]. Individual components usually provide fix capabilities that might not satisfy all system requirements and some of them may be unnecessary in a given system. This reduces the chance of a match between a component and stakeholder requirements.

That's why we begin by identifying the components that are generic and common for the proper functioning of the whole process of maintenance. Then, we identify the specific components for the s-maintenance.

To carry out maintenance of an automation system , it is typically recommended to use the following integrated applications [18]: a data acquisition system (SCADA, Supervisory Control And Data Acquisition), Diagnosis system, CMMS (Computerized Maintenance Management System), an ERP (Enterprise Resource Planning), a documentation system (eDoc), a Prognosis system , a scheduling system and finally a geo-location system. Each system relies on the model of the enterprise, the physical system or the equipment. All of these components can be integrated as extern applications in the system.

Then, we proceed now to identify specific components of s-maintenance. Concerning the Self management of maintenance process characteristic, a component for process management is needed.

Regarding to the Semantic based Maintenance, Self maintenance [22], Smart Maintenance, Intelligence and Expert knowledge characteristics; they require the presence of two major components which are a knowledge base to share semantics and to contain domain and expertise knowledge, as well as a reasoning engine inferring new knowledge and intelligence form the knowledge base.

Collaboration and cooperation need a coordination component; to share the knowledge it needs a knowledge base; a component ensuring security and manages users' roles as, well as a manger for human machine interface to adapt content to users' context and roles, then providing dynamic and contextual human machine interface.

Vis-à-vis semantic interoperability, a mediator component is needed to ensure this functionality based on the shared semantics.

For the Self Learning characteristic, a component ensuring traceability as well a component for reasoning is highly recommended to save and then analyze interactions in the system to learn and understand these interactions and save a new knowledge in the system.

And finally, About service generation which is a crucial point in s-maintenance, the presence of the reasoning engine is unavoidable, also a component for generation of new services must be present as well a directory to put on the new generated services.

In summary, we identify this variety of components which must be included in an s-maintenance system responding to its requirements (see figure 2). These components are: *coordinator, mediator, HIM-human interface manager, AC-access controller, KBC--Knowledge Base Controller, KB - Knowledge Base, Reasoner, Slib - Service Library, SG- Service Generator, PM - Process Manager, TM - Traceability Manager*.

The definition of these components and their functionalities will be provided in the next section.

B. System architecture

The core of this architecture is composed by four software layers allowing the integration of s-maintenance characteristics and applications in the system. Figure 3 gives an overview of the different layers of the core of the platform associated with the applications integrated into the platform as well as the smart technologies used by these applications. We can observe that components and mediator layers are based on the ontology layer.

We give now a brief description of these components and their functionalities.

a- Coordinator: it coordinates between all the system's components and the integrated applications. In the system core, all components are connected via the coordinator. In addition, this component is also a conduit of communication between users with the system and between applications with the core.

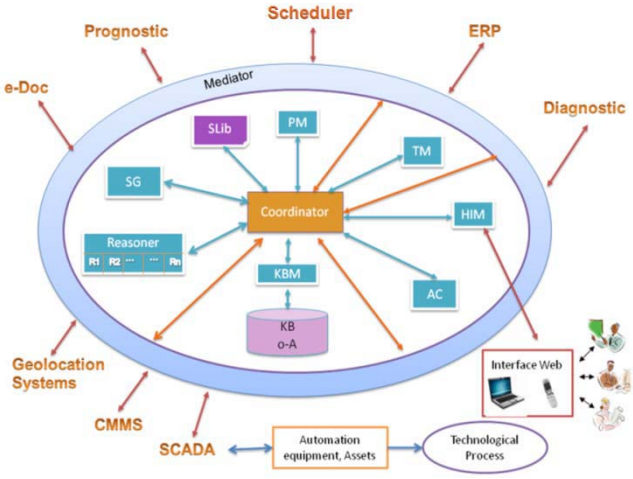


Figure 2. Composition of the proposed system

b- Mediator: it aims to ensure semantic interoperability between applications via the system. The architecture of s-maintenance system is based on knowledge and semantics represented by an ontology shared between its integrated systems. The local ontologies of these systems grow independently of each other. When ensuring semantic interoperability between these systems, the mediator takes into account the evolution of the local ontologies of these integrated systems [23].

c- HIM (Human Interface Manager): it manages all the postings on the Web interface in good shape and good content. Each actor has access to the maintenance system through HIM. A Maintenance actor (e.g. manager, expert, operator, etc) must be able to connect to the system via a Web browser on any type of client terminal. The client terminal may be a traditional PC, PDA, mobile phone or any other type of client, which it is possible to connect to the Internet.

d- AC (Access Control): Different actors can intervene on the system. It must therefore be able to manage user rights as well as applications. In this case, the use of an authentication database can limit unauthorized access The AC gives access permission to a component or application requested by a query issued by a user or application.

e- KBM (Knowledge Base Manager): it checks the consistency of outputs /inputs of knowledge with the ontology base (all knowledge that the other components inquire or register in the KB, are consistent with the ontology).

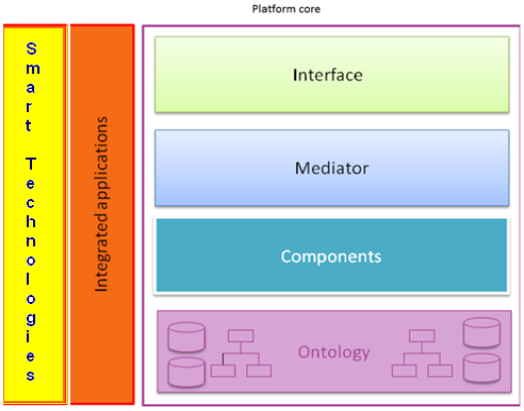


Figure. 3. Layers of the s-maintenance system

f- KB (Knowledge Base): In general, the ontology and all individual instances of concepts provide a knowledge base. In our case, knowledge base contains the ontology (concepts and rules [ontological meta-model]), ontology (instances of ontological meta-model), data (A: assertions [instances of ontologies]).

g- Reasoner (reasoning engine): The overall goal of a reasoning engine is to seek information and relationships of the knowledge base and provide answers, suggestions and predictions similar to the way a human expert. That is to say, new knowledge is generated from existing knowledge through reasoning engine. The reasoning engine contains general algorithms that can afford to manipulate the knowledge stored in the knowledge base.

h- SG (Generator Service): When a not exited service is required, Coordinator asks the SG to establish the new service (from requirements, provided information, reasoning, etc). After generation of this new service, it is registered in the SLIB. It may be a composition of basic services defined in the SLIB. Currently, we seek a mechanism or algorithm to implement this component.

i- Slib (Service Library): it contains all services that are not provided by existing applications (integrated in the platform) such as: Traceability of functional mode of the equipment or Financial Analysis by indicators. For now, we will offer each service description as an XML file.

j- PM - Process Manager: it orchestrates all the processes that pass through the system. That is to say that the PM component is responsible for managing the order process passing through the platform.

k- TM (Traceability Manager): it allows knowing the actions done through the system. While all internal communications between the platform and the

applications pass through the Coordinator, therefore the TM need just to be connected to the coordinator to trace the overall platform's interactions. The TM allows the tracking of all the interactions between applications, users with the platform and the interactions between components of the platform.

C. *System functioning*

The possible interaction between the user and the s-maintenance system are shown in figure 4. There are two types of interactions, authentication request and service requests (already integrated or generated by the system).

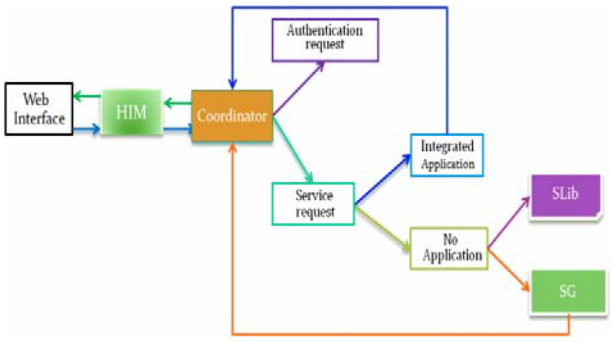


Figure 4. User and system Interactions

When user requests to access to the system through the web interface, his request passes from the HIM to the Coordinator. Coordinator sends the authentication request to the component AC to verify if that user can access the platform or not?

After a successful authentication to connect to the system, the user can querying services provided by the system according to his role rights.

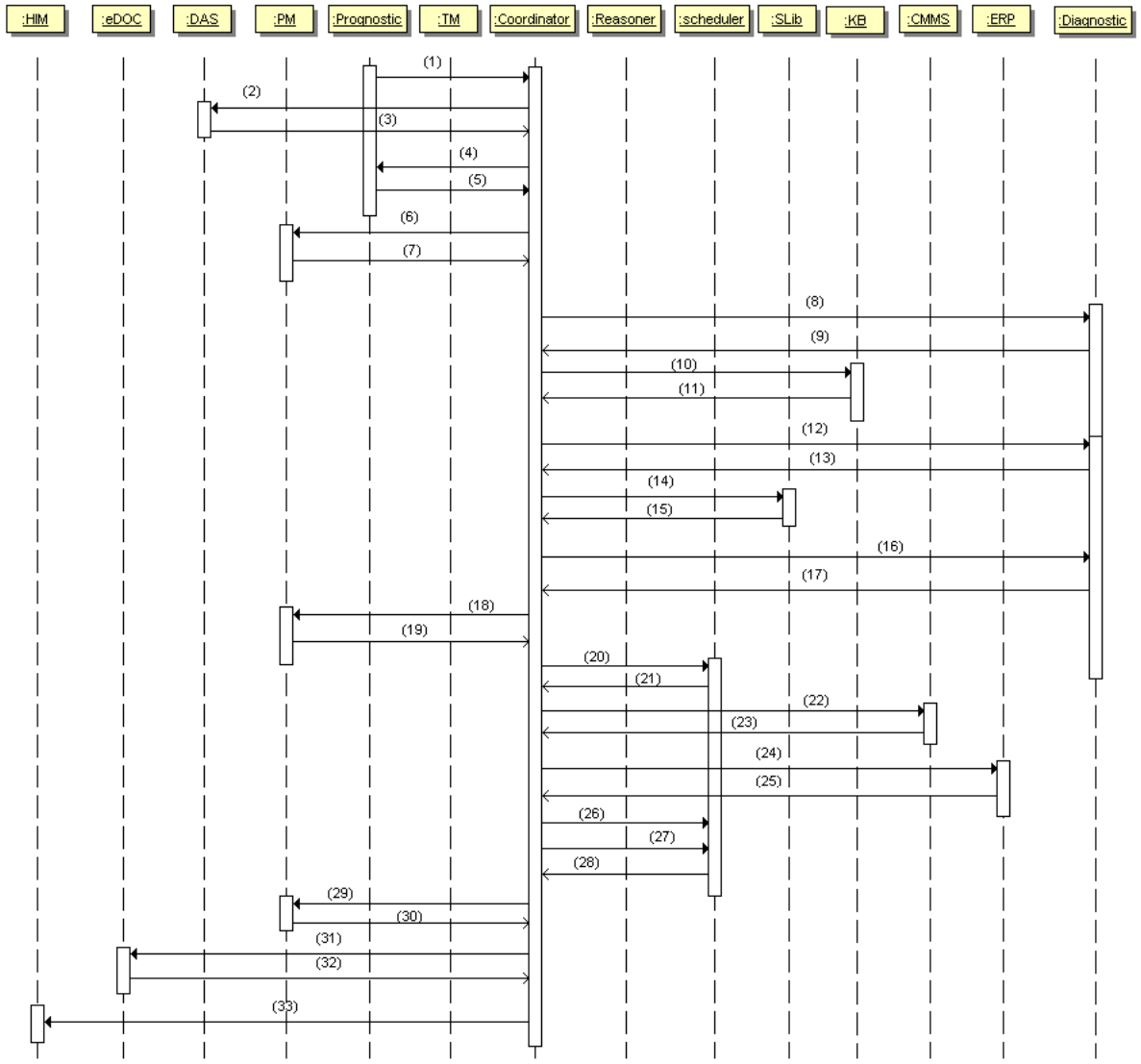
When the Coordinator receives a query that requests a service provided by the system, two cases are possible. The First one, these services are provided by the applications integrated into the system. In this case Coordinator locates the right application which provides the requested service. Then it communicates with this application. The application is launched; if needed, it has the possibility to ask the Coordinator for communicate with other components of the system core.

The second case is if the requested services are not provided by the integrated applications. In this case, Coordinator seeks in the SLib (service library) to verify if these services exist or not. If the SLIB library contains these services then Coordinator launching these services. Else the Coordinator asks the SG (service generator) to generate these services through the use of existing knowledge.

Moreover, to illustrate the system execution, we present an example of a self-management of maintenance service provided by the system. This service aims to ensure automatically the OSA-CBM process without the intervention of human operators only in the intervention task. In this example we illustrate interactions between the system components throughout the OSA-CBM process [19]. The example treats the RUL (Remaining Useful Life) [20] generated by the prognostic application about a component x into an equipment Y.

Figure 5 shows the sequence diagram of these interactions. We note that the OSA-CBM process is already instanced in the knowledge base and used by the PM in the illustration. We note also that the diagnostic application used in the illustration is based on the equipment model as well as the behavior model of the functioning of the equipment. The diagnosis system identifies causes of failures and actions that will be done.

The prognostic application sends a query to coordinator asking for data about component x of equipment Y from sensor. Coordinator asks the SCADA for data about component x of equipment Y. The SCADA send the asked data to the Coordinator which forwards it to the prognostic application. After making its analyses, this later sends an urgent RUL to the coordinator that the component X will fail in 10 hours. When receiving the RUL, the coordinator doesn't know what action it should do, so it sends a query to the PM asking for what action it will be done in the scope of the OSA-CBM process. The PM responds to the coordinator that it will launch an expertise. So, the coordinator sends a query request for a diagnosis expertise about the equipment Y from the diagnosis application. This later asks the coordinator for sensors data and all information about the equipment model. The coordinator gets this information from the KB and the SCADA and forwards them to the diagnosis application, which asks also for the behavior model of the equipment Y. while this model is not given from any application integrated in the platform, coordinator consults the SLib to ask if there is a service providing the behavior model of an equipment. So when finding this service in the SLib, coordinator asks for the execution of this service for the equipment Y. Then, coordinator forwards the generated behavior model to the diagnosis application. After analyses and reasoning, this latter edits the expertise report recommending the change of component x and component z which causes the failure of the component x. The coordinator asks the PM what to do after the expertise. The PM mentions that it will plan an intervention. So, the coordinator asks the scheduler application to plan an intervention to change components x and z of equipment Y before 10 hours. The scheduler asks about availability of spare parts and human resources. So, the coordinator gets this information from the CMMS and the ERP applications and forwards them to the scheduler which books spare parts and maintenance operators and edits the intervention. After the intervention planned, the coordinator asks the next action form the PM which recommends the edition of the work order. So the coordinator asks form the e-doc application to edit this latter. While edited, the coordinator sends the work order to the operators planned in the intervention via the HIM.



Messages Key: [? (ask for)] [S (Give)] [! (what to do?)] [A (answer)] [O (order)]

1: ? data: (component x; equipment Y, sensor)
2: ? data: (component x; equipment Y)
3: S [data (component x)]
4: S [data (component x, equipment Y)]
5: S [urgent RUL]
6: ! [RUL urgent]
7: A (expertise)
8: O (expertise (component x, equipment Y)
9: ? data: (component x; equipment Y, sensor) && ? Information: (equipment Y)
10: ? expertise model (equipment Y)
11: S[expertise model (equipment Y)]
12: S[data && expertise model (equipment)]
13: ? behavior model (equipment Y)
14: O service (behavior model: equipment Y)
15: S[behavior model (equipment Y)]
16: S[behavior model (equipment Y)]
17: A (expertise report [change component x, 10 hours] && [change component z])
18: ! (expertise done)
19: A (plan [intervention])
20: O (intervention (change component x & component z ,10 hours))
21: ? availability (spare parts && human resources)
22: S [book (spare parts)]
23: S[confirm (book(spare parts))]
24: ? availability (maintenance operators)
25: S[availability (operators)]
26: S[confirm (book(spare parts))]
27: S[list(operators)]
28: S[plan(intervention)]
29: ! [plan(intervention)]
30: A (work order)
31: ? edit (work order (intervention))
32: S[edit (work order)]
33: O (work order, operator)

Figure 5. Interactions between platform components into self

IV. IMPLEMENTATION AND DISCUSSIONS

After simulating a scenario in form of a sequence diagram, we are currently simulating the architecture with a multi-agent system MAS. Each component is simulated by an agent. After that, 4 groups of agents and their roles according to the definition and the behavior of the component are defined. Then two scenarios are specified to show three main characteristics of the s-maintenance. The first scenario is about self-learning through the TM component and self-management through the PM. The second scenario is about the generation of new knowledge via the SG. Some experiments should be planned to test some features of the system. In addition, the MAS will permit to test the charge of the messaging traffic for each particular component in the system. MADKIT platform is used to implement the MAS architecture.

Concerning the implementation of components, the KB is already developed by constructing a domain ontology of maintenance using the knowledge description language PowerLOOM [3]. The mediator component is also already developed in a previous work [23].

In the other hand, concerning consistency of the proposed system, we identify two critical components in the platform which are the Coordinator and the KB. Indeed, we suppose that a large volume of queries can burden the Coordinator component that generates a large response time for query execution or even a crash on this component *(to be tested and verified via the MAS)*. A failure on the coordinator means a total failure of the system. Different others problems may occur if any component fails. Such problem can cause a disturbance in the functioning of processes and workflow of the system. Even worse in the case of a crash at the knowledge base, the system is no longer usable. For this, we propose to replicate all components of the platform core, and especially highly recommend replicating the critical components. These replications can be put in the same server and/or in distant server or site. In addition, we can put a query dispatcher to manage the queries pile between the replicated coordinators and balance the processing load and the availability. As well, technologies and mechanisms of autonomic computing and self-healing systems can be added to each component to remedy this deficiency and enhance the robustness of the system. To conclude this section, we note that presented aspects about the consistency of the system will be investigated in future work.

V. CONCLUSION

After analyzing the existing maintenance systems we found that none of these systems meet all characteristics of s-maintenance concept that present the next generation of maintenance systems. For this purpose, we have proposed in this paper an architecture of a component based system responding to these requirements. Each component in this architecture answers to one or more of them. It should also be noted that the components of this architecture are defined according to their functionalities and not their technical specifications. These later are left for the engineers' choice when developing the system to implement them as software modules, web services or integrated applications. A scenario of execution is simulated via an UML sequence diagram. Also, other simulation scenarios via a multi-agent system are already ongoing.

Future works will be dedicated to integrate new services in the system. We are also working to develop all of the system components. Further, mechanism to the SG (service generator) component to generate complex new service will be done by composing elementary services provided in the SLib component. In addition, possibilities and benefices are studied to elaborate s-maintenance system by a cloud computing architecture to allow more scalability, potency and efficiency to the system.

ACKNOWLEDGMENT

This work was carried out and funded in the framework of SMAC project (Semantic-maintenance and life cycle), supported by European program Interreg IV between France and Switzerland.

REFERENCES

[1] Muller A, Marquez C & Iung B. (2008) On the concept of e-maintenance: Review and current research. Journal of Reliability Engineering and System Safety 93 1165–1187. Amsterdam: Elsevier Science Publishers.

[2] Rasovska, I., Chebel-Morello, B. and Zerhouni, N. (2005). Process of s-maintenance: decision support system for maintenance intervention. Proc. of the 10th IEEE conference on emerging technologies and factory automation; volume 2, Catania, Italy. pp. 679-686.

[3] Karray M H, Morello-Chebel B, Zerhouni N (2009) Towards a maintenance semantic architecture. Proceedings of the Fourth World Congress on Engineering Asset Management (WCEAM 2009) Athens.

[4] Morello-Chebel B, Karray M H, Zerhouni N, "New perspectives of Maintenance systems: "towards s-maintenance", workshop of Sustainable products and production, services and maintenance. Zurich (May 2010) .

[5] Ganek A. G and Corbi T. A. (2003) the dawning of the autonomic computing era. IBM Systems Journal, vol. 42(1):pp. 5–18.

[6] Leondes, Cornelius T. Knowledge – Based systems: Techniques and Applications, ISBN – 13:9780124438750, Academic Press, 2000.

[7] Poole D, Mackworth A, and Goebel R. (1998) Computational Intelligence: A Logical Approach, Oxford University Press.

[8] Mahmood, S., Lai, R.: "Analyzing Component Based System Specification" Proc. of AWRE 2006, Adelaide, Australia, 2006.

[9] http://encyclopedia2.thefreedictionary.com/Computer-based+systems

[10] Stein, J. L. and Louca, L. S., "A Component-Based Modeling Approach for Systems Design: Theory and Implementation," International Conference on Bond Graph Modeling and Simulation, Las Vegas, NV, 1995.

[11] D. Garlan, R. T. Monroe and D. Wile. Acme: An architecture description interchange language. In Proceedings of CASCON'97, pages 169-183, Ontario, Canada, November 1997.

[12] Kotonya, G., Hutchinson, J.: "Viewpoints for Specifying Component-Based Systems" Proc. of CBSE 2004, LNCS 3054, Springer 2004, Edinburgh, UK, May 24-25, 2004.

[13] Saulius Gudas, Edvinas Pakalnickas "Component-based modelling at the system design stage". Issn 1392 – 124x information technology and control, 2006, vol.35, no.3a

[14] W. Zhang, Knowledge Management for E-Maintenance of Industrial Automation Systems, Studies in Computational Intelligence (SCI) 42, 139–155 (2007)

[15] Xiangang Cao, Pingyu Jiang, « Development of SOA Based Equipments Maintenance Decision Support System », Intelligence Robotics and Applications, volume 5315, pages 576-582

[16] Rissland, E.L., & Skala , D.B. (1989). Combining case-based and rule-based reasoning: A heurestic approach. In, Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89: pp. 524-30, Detroit, Michigan.

[17] THOMAS ERL, "Service-oriented Architecture: Concepts, Technology, and Design", Upper Saddle River: Prentice Hall PTR. ISBN 0-13-185858-0, 2005.

[18] Bangemann T, Reboul D, Scymanski J, Thomesse J-P, Zerhouni N. (2006) PROTEUS—An integration platform for distributed maintenance systems. Comput Ind [special issue on e-maintenance]; 57(6):539–51.

[19] Thurston, M. and Lebold, M., Open Standards for Condition-Based Maintenance and Prognostic Systems, "MARCON", 2001, www.osacbm.org (2003-03-10).

[20] Tobon-Mejia, D. A.; Medjaher, K.; Zerhouni, N. & Tripot, G. A Mixture of Gaussians Hidden Markov Model for failure diagnostic and prognostic. IEEE Conference on Automation Science and Engineering, CASE'10, 2010

[21] Lee, J., Liao, L., Lapira E., Ni, J., and Li, L., "Informatics Platform for Designing and Deploying e-Manufacturing Systems," in Collaborative Design and Planning for Digital Manufacturing, 2009, pp. 1-35

[22] Labib, A.W., Next generation maintenance systems: towards the design of a self-maintenance machine. Proceedings of 4th IEEE International Conference on Industrial Informatics, INDIN'06., Singapore (2006)

[23] Karray M.H., Chebel-Morello B., Zerhouni N. A contextual semantic mediator for a distributed cooperative maintenance platform. Proceedings of 8th IEEE International Conference on Industrial Informatics, INDIN'10., Japon (2010)