

A COMPARISON BETWEEN EXPERT SYSTEMS AND AUTONOMIC COMPUTING PLUS MOBILE AGENT APPROACHES FOR FAULT MANAGEMENT.

UNA COMPARACIÓN ENTRE LOS ENFOQUES BASADOS EN SISTEMAS EXPERTOS Y COMPUTACIÓN AUTÓNOMA MÁS AGENTES MÓVILES PARA LA GESTIÓN DE FALLAS.

SERGIO ARMANDO GUTIÉRREZ

Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia. Email: saguti@unal.edu.co

JOHN WILLIAN BRANCH

Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia. Email: jwbranch@unal.edu.co

Received for review September 16th, 2010, accepted December 1th, 2010, final version February 26th, 2011

ABSTRACT: In this article, a revision of literature on application of expert systems to fault management is presented. Then, a new approach, based on the autonomic computing paradigm and mobile agents technology is introduced. The main features of this new approach are compared to equivalent ones of expert systems. Benefits of the schema based on mobile agents are presented.

KEYWORDS: autonomic computing, mobile agents, expert systems, fault management

RESUMEN: En este trabajo se presenta una revisión de trabajos relacionados a la aplicación de sistemas expertos a la gestión de fallas. Se introduce también un enfoque novedoso, basado en el paradigma de la computación autónoma y la tecnología de agentes móviles, y sus principales características son comparadas con respecto a las equivalentes de los sistemas expertos. Los beneficios del esquema basado en agentes móviles son presentados.

PALABRAS CLAVE: computación autónoma, agentes móviles, sistemas expertos, gestión de fallas.

1. INTRODUCTION

The capacity of detecting failures in a given system is a very important challenge which arises in many engineering disciplines. When dealing with expensive equipment's, large and complex systems, or an infrastructure on which critical services or processes depend, requirements of availability and reliability increase dramatically. This is a reason why fault management becomes a very important research topic in areas such as chemical, aerospace, nuclear, and electrical engineering, among many others [1]. An emergent field in which expensive equipment and critical systems arise is telecommunications (TELCO).. Today's telecommunications networks are very complex systems involving hundreds and even thousands of components, every one having multiple variables and features to be evaluated in order to assess how well the services running on them

are performing. Due to need of quality service, and because of the growing dependency that organizations and people have on these services, networks (and the infrastructure and platforms where services run), need to have a high degree of availability as well as the capacity to react to failures and problems which might arise during their operation. Every day, *critical mission* is becoming a more commonly used term for designating the infrastructure which is used to run TELCO services. Critical mission means that systems should be available 100% of the time, and must not be interrupted under any circumstance, as an interruption might mean that a critical process can not be performed [2]. This requirement of high availability implies that any event associated to failures in any component of the system be managed in such a way that the degradation or unavailability of a system be avoided or minimized as much as possible. This is the reason why fault management has become a big issue

in the field of operation and engineering. As in many cases, the ability of human operators and supervisors is not quick and efficient enough to respond in the best way to the event of a failure. This work presents a comparative discussion and analysis of the features of one of most relevant artificial intelligence techniques which is reported in the literature for fault management, in particular at the detection and diagnosis phases as is expert systems, compared to an approach defined using the paradigm of autonomic computing. The rest of this paper is organized as follows: Section 2 presents some relevant works on expert systems applied to fault management. Section 3 outlines the proposed framework for fault management based on autonomic computing. Section 4 presents a comparison between artificial intelligence techniques and the proposed framework, and Section 5 presents the conclusions.

2. RELATED WORKS

Expert systems belong to a branch of artificial intelligence named knowledge engineering. An expert system can be understood as a software mechanism used to store and represent the knowledge acquire from one or several human experts. For representation and storing, symbolic tools as Neural Networks, graphs, rough sets or rules, among other. The core of an expert system is the inference engine, which allows it to obtain solutions for problems within the knowledge domain corresponding to the knowledge base of the expert system.

A basic expert system is composed of an inference engine, a knowledge base, a knowledge-acquiring subsystem, a human machine interface, and an interpreter. Figure 1 illustrates the relationship among the components and the human user [3].

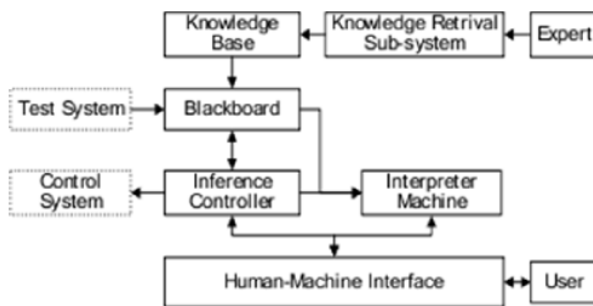


Figure 1. Expert system structure [3]

Literature reports the application of expert systems to many fields in engineering, although, it is worth mentioning that a few articles report the application of this technology to the specific field of TELCO networks and platforms. However, for the purpose of discussing the main features of expert systems applied to fault management, and comparing them to the proposed approach based on autonomic computing plus mobile agents, the application of expert systems to different engineering fields is considered. Also, is valid if TELCO platforms are understood as complex systems, formed by many interrelated components, whose behavior is dynamic and time changing;

Xian-Ming *et al* [4] describe the application of an expert system based on neural networks to fault diagnosis in an artillery radar. This system has two knowledge bases. One is for static knowledge, mainly corresponding to fault descriptions. The other base is for dynamic knowledge, formed by the reality data taken from an expert user, and by data coming from the reasoning process. The reasoning engine of this system works in two ways. One is automatic reasoning, applied to the knowledge base by using the fault information. The other is dialogue reasoning, which is triggered when a user interrogates the system; For the implementation of the expert system, no specialized software is used; General purpose database and programming language are used to implement the system.

Tongle *et al* [5] present the application of an expert system to diagnosis in a field of Mechanical Engineering, the Rolling-Bearing faults. An expert system is used as mechanism to detect possible reasons for a failure, and after reasoning, inform it to a Human Team. For this implementation, authors present the use of CLIPS, which is a specialized language to build expert systems.

Xianyong *et al* [6] present the development of an expert system applied to diagnosis in Storage Systems. This work proposes a different coding scheme for representing knowledge, and uses Neural Networks for the creation of the inference model.

Yougang [7] presents a model based on Rough Set Theory and expert systems for fault diagnosis. In this work, Rought Set Theory is used to complement expert systems. By using sets notation, the components of the system are described. The most important of

the components is the decision table. As many of the variables related to system behavior might be continuous, a mechanism for discretize the values of variables is provided. The combination of Rough Set Theory and expert systems shows to be useful in filtering redundant information and eliminate false alarms.

Jian *et al* [8] describe a solution combining expert systems and Fault Tree. In this approach, the authors try to address common problems associated to expert systems as knowledge acquisition, performance for large and complex systems and knowledge base management. This solution is implemented by using general purpose software tools.

Krivoulya *et al* [9] propose the application of a Fuzzy expert system in diagnose of computer system failures. Firstly, authors describe a series of features or requirements which are not fulfilled by static expert systems, and then, authors explain how the concept of fuzzy for an expert system is very natural, as the expert system is meant to store human knowledge, which is usually expressed in qualitative terms rather than in quantitative terms. Authors show how this kind of expert system is suitable for solving inaccurate and subjective problems, and how fuzzy logic is a good mechanism to store human knowledge.

WenBin *et al* [3] analyze several aspects of knowledge base structure, which is an important topic in expert system design. Authors define a data model, specify mechanisms for conflict resolution and some tips for increasing or decreasing preconditions in rules. The implementation is based on general purpose software tools.

Ding [10] presents the application of expert systems to Remote Diagnosis by defining mechanisms to interconnect several of them through Internet. Interconnected expert systems have the capacity to acquire knowledge manually (being fed from a human expert), automatically (from experience) or semi automatically (combination of manual and automatic).

Xue *et al* [11] propose the use of FPGA (Field Programmable Gate Array) circuits for building the reasoning engine of expert systems. The results show this approach fits naturally for implementing reasoning trees and offer a very good performance for small

systems, but present problems for larger systems, and does not fit for reverse reasoning.

Ma and Wang [12] propose a general purpose expert system, based on the concept of expression parsing. This mechanism is used to represent the human knowledge, initially by means of logic expression and at reasoning phase, by using mature expression parsing.

Finally, Zaw and Soe [13] present the design and implementation of a rule based expert system applied to Fault Management. In this case, the expert system is employed for solution of very simple faults in the environment of a personal computer. The user interface of the expert system is very simple to use. Authors do not report the use of a specific shell or tool to build the knowledge base and inference, but it is created by using general purpose tools (Relational Database and Java programming language)

3. PROPOSED FRAMEWORK

In this section, a description of the proposed framework for Fault Management by means of Autonomic Computing will be introduced.

3.1 Introducing Autonomic Computing

Autonomic Computing is a concept inherited from bioinspired computing in its beginnings. The first work where Autonomy Oriented Computing is introduced is the work by Liu *et al* [14]. This work defines the fundamentals of Autonomy in computing and the four main characteristics which should be exhibited by a computing system according to this paradigm:

- **Autonomy:** The entities in the system are rational individuals, capable to act in an independent way; in other words, the system does not have a central component and manager.
- **Emergent:** When system entities cooperate and work together, they exhibit behaviors not available or possible to obtain by individual entities separately.
- **Adaptive:** System components are capable of modify their behavior according to changes present in the environment where system operates.

• **Self-organized:** The system components are capable of organize themselves to achieve the previous commented behaviors. This paper also presents the several types of Autonomy Oriented Computing according to how autonomy is achieved by the system.

In Jin and Liu [15], Autonomy Oriented Computing is formally defined, by employing set notation to express the concepts associated: Environment, computing entity, state, behavior and goal are some of the concepts which are expressed in a formal language. From a more applied point of view, and again from a bioinspired perspective, Horn [16] presents the approach to Autonomic Computing from perspective of a industry leader as IBM.

Following Horn, there are eight keys elements present in an Autonomic Computing System:

• **Self-Awareness:** This is, the system should know itself. It will need detailed knowledge of its components, current status, ultimate capacity, and all connections with other systems to govern itself. It will need to know the extent of its “own” resources, those it can borrow or lend, and those that can be shared or should be isolated.

• **Self-Configuration:** System configuration or “setup” must occur automatically. Also, the system must modify itself, in such a way

that its configuration, must be the adequate to cope with environment conditions.

• **Self-Optimization:** This is, the system will always try to find other ways to improve its functioning. It will monitor its constituent parts and fine-tune workflow to achieve predetermined system goals.

• **Self-Healing:** System must be able to recover from events that might cause malfunctioning. The system also

must be able to detect problem or potential problems, and according to this, define alternate ways to perform, applying reconfiguration to keep the system working.

• **Self-Protection:** Starting from the fact of the potentially aggressive and hostile environment where system resides, it must be able to detect, protect and identify potential attacks or vulnerabilities, so that it can protect itself and keep working in a secure and consistent state.

• **Self-adaptability:** This is, the system must know its environment, the context surrounding it when operating, and the other entities cohabitating with it. It must be able to adapt to this environment, and its changing conditions, by reconfiguring itself or optimizing itself.

• **Openness:** The components in system must be open to communicate each other, and must be able to work with shared technologies. Proprietary solutions are not compatible with Autonomic Computing philosophy.

• **Self-containment:** Components within an Autonomic Computing System must be able to perform the task or tasks they have assigned, not requiring external interventions

for the performing itself, and hiding the complexity to the end user.

Complementing the work by Horn, Lin *et al* [17] review Autonomic Computing from the perspective of Software Engineering. Authors present a proposal of metrics which could be used to evaluate the quality of frameworks based on Autonomic Computing. In other side, the works of Sterritt [18], Magedanz *et al* [19], Ionescu *et al* [20] and Tizghadam *et al* [21] evidence how Autonomic Computing has become a very important.

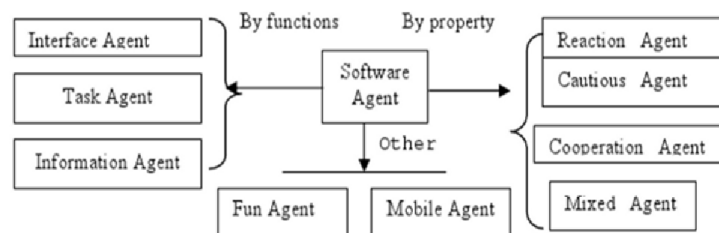


Figure 2: Software Agent Classification [23]

Research topic in the field of Information Technologies, both for academic communities and for industries, in solving many problems that are becoming difficult to face with traditional and conventional approaches.

3.2 Introducing Mobile Agents

Agents are enjoying a lot of popularity as a novel abstraction for structuring distributed applications. It is a technology from the field of Artificial Intelligence. Although, there is not a precise and widely adopted definition for Agents, the tendency is to define it through the features they should expose [22].

Following to Yubao and Renyuan [23], an agent is an entity possessing the following characteristics:

- **Self-government:** Agents should have the ability to govern themselves, without external interference from the outside world while they are performing their tasks.
- **Smart:** Agents should implement certain functions and should be able to choose the required

information to complete their tasks. They also should be able to get knowledge from the performance of their tasks.

- **Lasting:** The agents should survive, according to their participation in the tasks they perform.

Co-relation: This is the social behavior defined by theory. In the real world, the cooperation is presented as messages exchange among the agents in the system.

Figure 2 presents a classification of software agents according to their functions, their properties, or other relevant features which can be assigned to them.

Specifically, a feature relevant in the approach to be proposed is mobility. That is, the capacity of the agent to move itself autonomously across the environment where it runs, with the goal to execute its designated task at other locations where it is required.

3.3 Autonomic Computing and Mobile Agents applied to Fault Management

For the goal of offering high availability and reliability on critical systems, increasing the reliability of Fault Management (FM) component or components is a very important goal to achieve. The elements detecting Faults, with the goal of detecting hem, isolating them

or even correcting them should become very smart and reliable, and should adapt to the complexity of the systems where they are meant to be used.

The approach to FM which is proposed, consists in implementing Mobile Agents (MA), which exhibit the eight fundamental features of AC defined by Horn [16]. These agents will be designated to monitor specific resources, and will use a Knowledge Database which will store criteria defining faults for the associated resource. According to these criteria, the agents will be able to anticipate to faults on the component, by analyzing and detecting patterns on the designated assess metrics for the resource and by means of mobility feature, agents would be able to move across the system to monitor the designated resource at other locations.

The architecture is composed by the following components:

- **Knowledge Database (KDB):** Is a database containing the criteria which are going to be used by the agents to determine faults and how to react before them.
- **Data Collector (DC):** It is a component receiving data coming from the agents, and capable of sending them information regarding changes in the criteria for fault detection.
- **Mobile Agent (MA):** It is an agent which will perform the actions on the resource and will traverse the network monitoring similar resources in other devices.

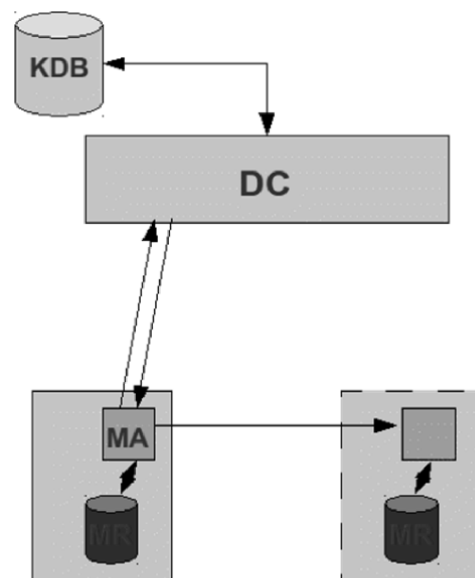


Figure 3. Proposed Approach Architecture.

- **Managed Resource (MR):** It is the resource in the hosts which will be supervised by the agents, for possible faults and problems in its performance. This resource can be a hardware element, a software element, or the behavior of a particular metric or behavior in service.

Database as mentioned, will have two purposes. One will be to contain the data reported by the agents, for example, whenever it performs a correction because of a failure, or according to periodic monitoring whether defined. The other purpose will be to store rules or instructions defining the criteria to act when a faulty behavior is detected. For example, if a failure is a full file system, the criteria could be what files to delete, or what kind of files to delete. Also, according to the functionality which performs the system component, the collected data would be useful to predict that a failure is yet to come, and so, indicate a proactive action triggered on the MA.

Data collector would be the entity communicating with the MA. It will have two purposes. The first, collect the data from the MA, by receiving on them, or by triggering a data query to them. The other will be to perform smart analysis on data collected, and if possible, to modify the criteria of fault detection and notify these changes on criteria to MA.

For example, if MR is a file system, the data collector could analyze the growing rate of a particular file and predict that it might cause a failure by a full file system and notify this to MA.

Also, the data collector would be useful to notify exceptions to MA, so that they could ignore the application of a criteria. The DC could be an agent itself.

Mobile Agent would be the entity performing the detection and correction of failures on the hosts. It would be implemented in such a way to apply criteria to solve faults in a particular resource, and it could choose to travel or to clone itself to send a copy of itself to another host requiring supervision in the same resource.

Managed Resource would be the system component under supervision and control by Mobile Agents. It might be a hardware or software element or even a service or task.

4. COMPARISON

This section will discuss a comparison between the most relevant features that can be generalized for the reviewed works on Expert Systems applied to Fault Management, and the equivalent ones in the proposed approach for Fault Management.

Despite the wide application of expert systems [12], and the fact that expert systems are a technology which has been used for a long time [8], there are some features which are inherent to its nature [11],

[7] limiting its applicability. The following subsections detail some of these features, and introduce how they could be addressed in the proposed approach.

4.1 Knowledge Acquisition

Most of the expert systems, and specifically the study cased reviewed and referenced in Section 2, have interfaces which capture the knowledge directly from a human expert, or a group of human experts. Through direct input of rules in the mechanism used for the specific implementation (See below), this knowledge is taken into the Knowledge Base.

Because of the inexact and qualitative nature of the human language, certain measures have to be taken so that the system is able to cope with this ambiguous data, for example, the usage of fuzzy logic.

It is unusual in an expert systems to have the capacity of feedback. That is to say, that the expert system is able to learn from experience. The Knowledge Base has to be modified, extended and corrected through human intervention. A relevant exception to this feature is the work of Ding [10] which reports an expert system capable to acquire knowledge from experience.

In the approach using autonomic computing and mobile agents, although the starting point is a predefined knowledge database, containing the criteria specifying faults, the Autonomic Features of the agents to be used for monitoring allow them to feedback the knowledge database with the goal of generalize a fault defined in specific way. For example, if a defined fault is a Full Filesystem on a Server, the monitoring agents might modify the knowledge database to include a particular

file for observation. In a certain way, it might be defined that the main mechanism of feedback for the approach is the experience acquired by agents during their execution.

4.2 Knowledge Representation

As expert systems are usually implemented using purpose specific tools, this tools impose restrictions on the syntaxis and the symbols used to represent the knowledge into the knowledge database.

The usage of rule languages, rough sets and even mathematical symbols is common as the input for these systems. This way of knowledge representation imposes that operator interacting with the knowledge database of an expert system requires Artificial Intelligence knowledge, or at least, deep knowledge of the tool or tools used for this representation, not only in the syntactic aspect, but also in the implications of adding or modifying rules in the proper operation of the expert system.

In the proposed approach, as previously mentioned, the knowledge database contains the criteria which define faults to be supervised and corrected by the Agents. These criteria would be expressed in descriptive way, classifying the faults according to resource to be monitored, and some thresholds which should be verified. As for the storage a relational database is proposed, no particular rule syntax would be required as for ES implementation tools. For this case, a deep domain knowledge is required, but no specific Artificial Intelligence artifacts are required for manipulating the knowledge database.

4.3 Human Interaction

The most common mechanism of interaction between human an operator and expert system are questions. Human Operator inputs some kind of initial symptoms, usually through a keyboard, and from that initial input, the expert system asks questions to the user, which depending on the answers provided by the user, outputs a diagnosis. These questions are used to guide and trigger the inference and reasoning processes within the inference engine of the expert system.

Another limited interaction which is performed by the user is the task of modifying the knowledge base of the expert system.

This is performed through special interfaces provided by the tool implementing the expert system, or through interfaces accessing the database, to store rules.

In the proposed approach, human interaction is required to populate the knowledge database which stores the criteria describing failures. User also might interact with data collector to query reports on the information stored by the agents, or to modify execution parameters of the agents.

It is worth to mention that despite the eight main features of Autonomic Computing, a very important component in every system developed according to this paradigm is an interface where human operators can take control on the system operation. This is to avoid situations where a system might run out of control, presenting unexpected or undesirable behaviors.

4.4 Reasoning Process

The reasoning process, that is to say, evaluate the rules contained into the knowledge base of the expert system according to input received from the user is usually triggered for each answer. Reasoning is not automatic, it can be described as synchronous, as it is started, modified and triggered according to the user input. Also, as much as the knowledge database grows and extends, reasoning process tends to be slow. An exception to this case is the implementation of reasoning by using hardware [11].

In the proposed approach, as agents are implemented to be autonomic and domain specific, the reasoning process is simplified to assess and evaluate the current conditions of the component or resource under monitoring. Although monitoring agents communicate with the data collector and might communicate among themselves, the execution of their tasks is not conditioned to previous execution of other agents' tasks.

4.5 Fault Correction

Expert systems are basically limited to diagnose. After processing the input of the user, an expert system outputs one or several possible diagnosis options, and it is up to the user to apply the suggested corrections on the failed component. In certain way, the intervention on the components which expose faults is out of the scope of expert systems.

The proposed approach not only performs diagnosis and detection of failures, but offers the option to correct and prevent the detected failures. One of the primary goals of the Monitoring Agents is correcting the detected failures, and even preventing them according to the information collected during execution.

4.6 Scalability, Applicability, Performance

By nature, expert systems have a limited applicability. Features such as requiring deep Artificial Intelligence skills for interventions on a knowledge database, the algorithms ruling and guiding the reasoning process, and the mechanism to traverse the knowledge database, reduce the performance of expert systems for real world problems. Most of the reviewed literature state that expert systems exhibit degradation when applied to large and complex problems.

By its distributed nature, the presented approach, based on mobile agents, might offer a higher performance making it more suitable for real world applications, on large and complex systems. Al-Kasassbeh *et al* [24] and Adhichandra *et al* [25] report result on application of Agents for Fault Management, showing better performance than in other approaches.

5. CONCLUSIONS AND FUTURE WORK.

An approach based on Mobile Agents and Autonomic Computing has been introduced and its main features were compared to equivalent ones on Expert Systems. From this comparison, approach based on Mobile Agents present several advantages, which could be taken to improve Fault Management tasks for large and complex systems.

As future work, it is proposed to implement and assess these advantages, and even trying to merge both approaches to obtain a robust solution for Fault Management, fitting the needs of modern networks, platforms and services.

REFERENCES

- [1] I. HWANG, S. KIM, Y. KIM, AND C. E. SEAH, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Transactions on Control Systems Technology*, vol. 18, pp. 636–653, 2010.
- [2] T. FUJISAKI, M. HAMADA, AND K. KAGEYAMA, "A scalable fault-tolerant network management system built using distributed object technology," in *Proc. First International Enterprise Distributed Object Computing Workshop.*, 1997.
- [3] C. WENBIN, L. XIAOLING, H. CHANGJIU, AND L. YIJUN, "Knowledge base design for fault diagnosis expert system based on production rule," in *Asia-Pacific Conference on Information Processing*, 2009.
- [4] S. XIAN-MING, Y. HE-YONG, AND Z. PENG, "Fault diagnosis expert system of artillery radar based on neural network," in *International Conference On Computer Design And Applications (ICCD 2010)*, 2010.
- [5] X. TONGLE, P. XINCAI, AND W. XIUTING, "The study of expert system in rolling bearing faults diagnosis," in *International Conference on Educational and Network Technology*, 2010.
- [6] X. J. L. ZHANCHEN, X. ZENGHUI, AND L. YINGCHUN, "An expert system for comprehensive diagnosis in store management system," in *2nd International Conference on Advanced Computer Control (ICACC)*, 2010.
- [7] Z. YOU GANG, "Fault diagnosis model based on rough set theory and expert system," in *International Colloquium on Computing, Communication, Control, and Management*, 2009.
- [8] Z. JIAN, Y. JINSONG, AND D. JING, "A universal fault diagnosis expert system based on fault tree," in *The Ninth International Conference on Electronic Measurement & Instruments*, 2009.
- [9] G. KRIVOULYA, Z. DUDAR, D. KUCHERENKO, AND M. SAMI, "Fuzzy expert system for diagnosis of computer failures," in *10th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics CADSM.*, 2009.
- [10] Z. DING, "Research on internet-based open remote fault diagnosis expert system," in *International Conference on Information and Multimedia Technology*, 2009.
- [11] J. XUE, L. SUN, C. Q. M. LIU, AND H. QIAN, "Implement fault diagnosis high speed reasoning expert system with FPGA," in *Proc of IEEE International Conference on Mechatronics and Automation*, 2009.
- [12] L. MA AND H. WANG, "General-purpose fault diagnosis expert system based on the method of expression

parsing,” in IEEE Circuits and Systems International Conference on Testing and Diagnosis, 2009.

[13] S. M. M. SOE AND M. P. P. ZAW, “Design and implementation of rulebased expert system for fault management,” World Academy of Science, Engineering and Technology, vol. 48, pp. 34–39, 2008.

[14] J. LIU, K. TSUI, AND J. WU, “Introducing autonomy oriented computation,” in Proceedings of First International Workshop on Autonomy Oriented Computation, 2001.

[15] X. L. JIN AND J. M. LIU, Agents and Computational Autonomy: Potential, Risks, and Solutions., ch. From Individual Based Modeling to Autonomy Oriented Computation, pp. 151–169. University of Bath, 2004.

[16] P. HORN, “Autonomic computing: Ibm’s perspective on the state of information technology,” tech. rep., IBM Corp, 2001.

[17] P. LIN, A. MACARTHUR, AND J. LEANEY, “Defining autonomic computing: A software engineering perspective,” in Proc. of the 2005 Australian Software Engineering Conference (ASWEC 05), 2005.

[18] S. R., “Soft computing and fault management,” in Proceedings of Fourth International Symposium Soft Computing and Intelligent Systems for Industry, Paisley, Scotland, United Kingdom, 2001.

[19] T. MAGEDANZ, “Report on fet consultation meeting on communication paradigms for 2020,” tech. Rep., Fraunhofer FOKUS, 2004.

[20] D. IONESCU, B. SOLOMON, M. LITOIU, AND M. MIHAESCU, “A robust autonomic computing architecture for server virtualization,” in International Conference on Intelligent Engineering Systems, 2008. INES 2008., 2008.

[21] A. TIZGHADAM AND I. A. LEON-GARCIA, “Autonomic traffic engineering for network robustness,” IEEE journal on selected areas in communications, vol. 28, pp. 39–50, 2010.

[22] S. FRANKLIN AND A. GRAESSER, “It’s an agent or just a program?: A taxonomy for autonomous agents,” in Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages., 1996.

[23] M. YUBAO AND D. RENYUAN, “Mobile agent technology and its application in distributed data mining,” in First International Workshop on Database Technology and Applications, 2009.

[24] M. AL-KASASSBEH AND M. ADDA, “Analysis of mobile agents in network fault management,” Journal of Network and Computer Applications, vol. 31, pp. 699–711, 2008.

[25] I. ADHICANDRA, C. PATTINSON, AND E. SHAGHOUEI, “Using mobile agents to improve performance of network management operations.”.