Automations in Computer Network Management Utilizing Computational Intelligence

Edison Xavier, Fernando Koch, Carlos Westphall

Network and Management Laboratory – Federal University of Santa Catarina (UFSC)
Postal Box 476 – 88.040-970 – Florianópolis – SC – Brazil

{xavier, koch, westphal}@lrg.ufsc.br

Abstract. *Computer network management, due to the complexity and scale of networks nowadays, has necessarily come to be a pro-active and automated activity. In this type of management there are several points of decision-making or reasoning where computational intelligence can be utilized. This paper discusses one of the most important points of decision-making in management: how to implement data collecting and the sending of actions to the environment. From the discussion it is presented proposals to aid decision-making through computational intelligence with the aim of automating this task.*

1. Introduction

Computer network management is traditionally composed of stages executed in a logical and continuous sequence, having as a result the management itself. [KOCH 2001] suggests the stages as presented in Figure 1.

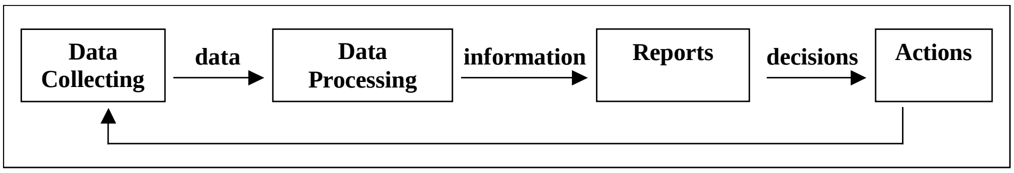

Figure 1. Traditional stages of computer network management

In some of these stages there are points in decision-making that utilize some subsidies for their resolution, such as information obtained from data gathered in the environment and historical behavior .Therefore, management decisions are based on data collected from and actions sent to the environment. These incursions deserve due attention, as they are users of the environment by themselves:

- They must consume the smallest quantity of resources, inasmuch as that is precisely the aim of management;
- Performance in the execution is a key factor for the good management process.

These points lead directly to a questioning: how to implement data collecting or actions that characterize them as “well behaved” users towards the environment and still obtain good results for the management?

Among the options available, there is the paradigm of mobile agents that have some interesting characteristics of implementation ([OHARE 1996]), and that they can be utilized conveniently according to the need. As seen in [XAVIER 2002], the paradigm allows for various solutions for implementation for the same data collecting or a management action. It is up to the manager to choose the most appropriate.

Nevertheless these decisions have become unfeasible for the human manager due to the complexity and scale of environments. For this reason, it is desirable to offer the manager some tool for automation of decision, based on simulation of his reasoning. To this end, we seek paradigms in computational intelligence that simulate human reasoning and are capable of implementing such decisions.

This study composes the [KDEMA] project, whose aim is to study applications of computational intelligence in each one of the phases of computer network management. In this paper we focus on automation of decision-making on how to implement data gathering or the sending of actions to the environment under management control. It is divided as follows: in section 2 we present the state-of the-art in the utilization of computational intelligence in network management; in section 3 we present theoretical justifications for the utilization of mobile agents; in section 4 we present our proposal for automation of decision-making on the most suitable configuration, explaining each step considered; finally, in section 5 we present some conclusions and indicate some future directions.

2. State-of the-art

The implementation of network management through other paradigms and their comparisons have been the target of various studies. There are studies comparing the behavior of mobile agents with SNMP, such as that of [RUBINSTEIN 2001], which makes the comparison based on implementations and approximate models. [ARANTES 2002] proposes mathematical models for calculating the time spent in management by mobile agents and SNMP, based on case studies.

One of the great advantages of mobile agent paradigms in network management lies in the aspect of decentralization of the manager figure, in which the agents are conveniently distributed by the environment and execute tasks that would normally be the responsibility of the manager. [THOTTAN 1998] cites this decentralization as an attempt to solve problems of scalability and make pro-activity possible.

Questions of the implementation of mobile agents are interesting and very important, since different configurations may have the same final result, but with quite different metrics – such as management time or produced octets. [RUBINSTEIN 2001] mentions alternatives in these configurations that seek better results. [XAVIER 2002] shows that different configurations really have quite distinct metrics, which are of interest to the management. [BOHORIS 2000] utilizes mobile agents only as transport mechanisms and shows unfavorable results to their utilization, since they do not exploit other characteristics of impact, such as intelligence.

Network management also requires implementations that involve intelligence. Conditional data gathering or decentralized pro-active detection are great examples of such a utilization. Computational intelligence can be used in all situations that involve decision-making. Research that utilizes computational intelligence in mobile agents can be found in [KARNOUSKOS 2002], where a neuro-fuzzy combination is utilized for proposing an intelligent and dynamic system of messages, adaptable to the conditions of the user. In [DAS 2002] we have computational intelligence acting in the distribution of tasks to mobile agents for the resolution of complex research problems on distributed bases. The metrics analyzed are "time" and "size" of the research. It shows three research configurations: one agent; multiple agents or conventional research, without agents. We perceive that the work must implement the configurations in order to reach decisions, that is, the manager cannot always anticipate the most suitable decision during the project time.

3. Mobile agents

Mobile agents may be utilized in the implementations of various stages of network management. [KOCH 2001] proposes the adoption of mobile agents in all stages of network management. Other studies like that of [ARANTES 2002] make comparisons between mobile agents, only as data collectors, and SNMP.

Actually, the mobile agents can be more than simple transport mechanisms for collecting or directly altering pre-defined data, as is the case of SNMP. There are characteristics found in [OHARE 1996] and explored in [XAVIER 2002] that present mobile agents as good candidates for more elaborate implementations, which seek decentralization of management through adaptability, mobility and intelligence. Mobile agents can carry any computation, theoretically meeting any need, superimposing the static characteristics of the client-server models of network management.

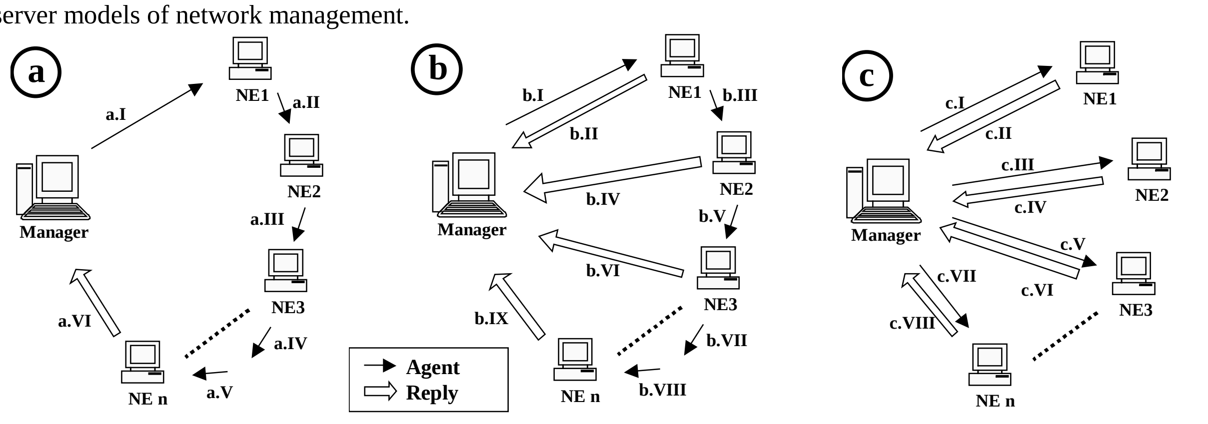

Figure 2. Different configurations of mobile agents applied to network management

We have as an example, seen in Figure 2, three configurations of mobile agents with the same objective: all must collect data in elements under management control (NEs) or perform some action to be executed in a given place. We have then, according to Figure 2, the three examples explained below:

a) An agent is created in an NE (a.I), it collects data or executes actions and moves on to the next NE (a.II) without sending any reply to the manager; it continues collecting or acting and moving through all the NEs (a.III, a.VI, a.V) until, at the end of the last collection or action, it sends the replies to the manager all at once (a.VI);
b) An agent is created in an NE (b.I), collects data or executes actions and sends replies to the manager (b.II), immediately moves on to the next NE (b.III) executing new collecting or actions and new replies to the manager (b.VI); this process continues until all the NEs are visited (b.V, b.VI, b.VII, b.VIII, b.IX);
c) One agent is created in each NE (c.I, c.III, c.V e c.VII), does collecting or actions and sends replies to the manager (c.II, c.IV, c.VI, c.VIII).

As one can see, the variation of constructive characteristics of mobile agents offers a variety of solutions that substitute classical models, besides introducing new possibilities, such as the sending of actions, conditional collecting or unrelated operations.

4. Computational intelligence in network management

A pro-active network has various points that are subject to reasoning and decision-making, each one with its characteristics within the process. There are situations of classification, of choices based on rules or even conclusions based on 'data mining'. For this reason, there exists the possibility of applying many paradigms of computational intelligence in network management, such as fuzzy logic, neural networks, genetic algoritms, tree-based decisions or especialist systems. They may be combined or utilized individually, in one point or all points of decision-making.

The [KDEMA] project is a component activity of the [HOPE] and [TAGERE] projects, developed by the Network and Management Laboratory of the Federal University of Santa Catarina, which sets out to study the application of computational intelligence in all points of network management. In this article we present studies on the points in which network management must utilize the environment for passing on information, such as data collecting or sending of actions. The utilization of mobile agents in this point is justified by the findings shown in Section 3.

4.1 Utilization of environment for network management

The main purpose of network management is to guarantee resources for the clients of a given environment, but the management is also a client, through its different kinds of utilization, such as collecting or sending actions, when they are submitted to that environment. The choice of the best implementation for these kinds of utilization may be of great value to management, since it can collaborate with the environment in matters of consumption of resources or with the management itself in matters of efficiency.

When network management needs to collect data or send actions to the environment, it defines the specific characteristics of such a need at the start:

- In data collecting, it defines the data to be collected and the targets of the collecting;
- In actions, it defines what must be done and the targets of the actions.

The management must then define "how" to implement and execute that need. Our study proposes the definition of an automation system for this decision, composed of four stages, as seen in Figure 3:

- Generator of configurations, responsible for analyzing the needs of this utilization of the environment and showing the configurations of possible mobile agents (according to Section 3);
- Accountant, which generates values for relevant metrics under each of the configurations considered; these values are obtained by simulations;
- Classifier, which utilizes algorithms to establish classifications among the metrics found; and
- Rules for decision-making, which utilize simulation of reasoning to define the best configuration, among those considered.

This system of support to decision-making, under a higher abstraction, has inputs and desired outputs:

- Inputs: characteristics of the environment to be utilized or managed and characteristics according to the management need; and
- Outputs: characteristics of the implementation defined as the most suitable.

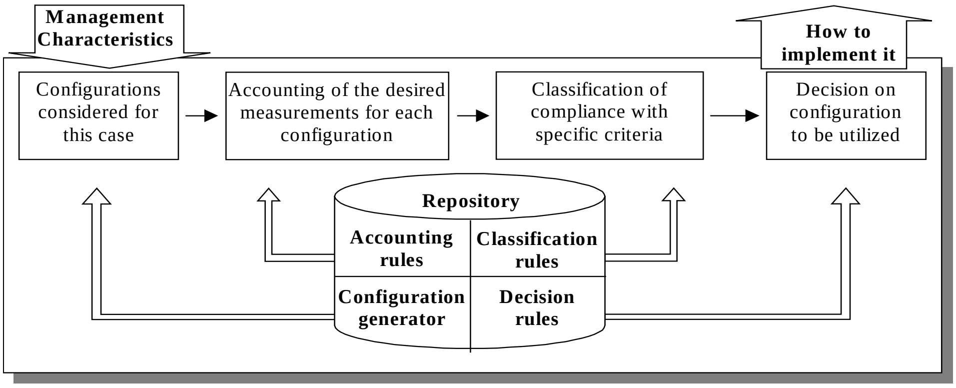

Figure 3. Stages of automation of how to implement a management

In matters of modularity and clarity, the stages are treated independently. In matters of interoperability, exchanges of information between the stages are codified in XML [XML 2003] according to the ontology agreed on. That codification makes the exchange of information possible among the different management groups, through the collaboration of their different kinds of reasoning, results or experiences.

4.2 Configuration generator

The configuration generator is the first stage of the decision process. It receives information from the management targets, as well as some characteristics of the environment such as existing connections, with their latencies and bandwidths. From this, it defines all the possible configurations of mobile agents for the execution of this management.

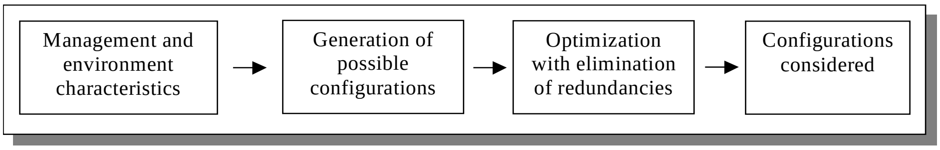

Figure 4. Stages of configuration generator

The steps of this stage, showed in Figure 4, are:

- Receiving the characteristics of the management and the environment, such as targets, actions, links, latencies, band widths;
- Generating all the configurations of possible mobile agents through specific algorithms; and
- Optimizing the result, detecting and eliminating possible redundancies.

```
<?xml version="1.0" encoding="UTF-8"?>
<environmentSpecs>
	<network name="headquarter" net="192.168.0.0/24" type="ETH-100Mbits"/>
	<network name="wan" net="192.168.100.0/30" type="PPP-64Kbits"/>
	<network name="sales" net="192.168.1.0/24" type="ETH-100Mbits"/>
	<host name="network manager" snmp="yes" agent="yes">
		<ip>192.168.0.2</ip>
	</host>
	<host name="router cisco-headquarter" snmp="yes" agent="no">
		<ip>192.168.0.1</ip>
		<ip>192.168.100.1</ip>
	</host>
	<host name="router cisco-salesposition" snmp="yes" agent="no">
		<ip>192.168.1.1</ip>
		<ip>192.168.100.2</ip>
	</host>
	<host name="position_1" snmp="yes" agent="yes">
		<ip>192.168.1.1</ip>
	</host>
	<host name="position_2" snmp="no" agent="yes">
		<ip>192.168.1.2</ip>
	</host>
```

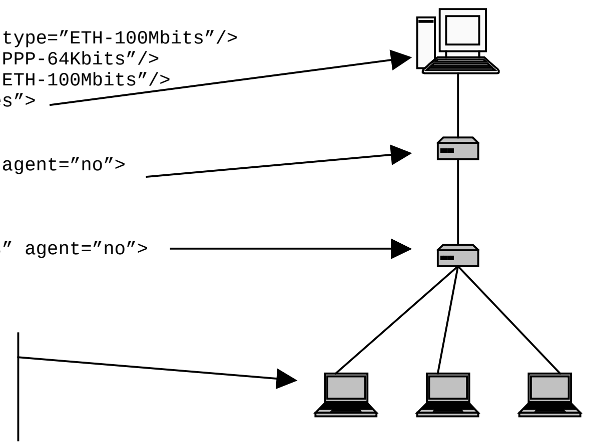

Figure 5. Example of environment description using XML and known ontology

From the management characteristics, the algorithm perceives the targets and their capacities, defining the possible interactions for the agents. The algorithm also defines the quantities of agents possible for a given management. Furthermore, the algorithm uses the characteristics of the environment (Figure 5) to define transport possibilities for these agents, considering physical characteristics such as links, networks and routes. The algorithm combines the possibilities, generating all the possible configurations for the management implementation, as seen in Figure 6.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<agentConfigurationSpecs>
      <configuration name=”a”>
            <agent name=”007”>
                  <location ip=”192.168.1.1”>
                        <getData target=”192.168.1.1”>
                              <type>snmp</type>
                              <id>0.1.3.6.0.0.1</id>
                        </getData>
                  </location>
                  <location ip=”192.168.1.2”>
                        <getData target=”192.168.1.2”>
                              <type>java-class</type>
                              <id>get_disk_space.class</id>
                        </getData>
                  </location>
```

Figure. 6. Example of configuration proposed by the generator using XML

4.3 Measurement Accounting

The configurations obtained in the previous stage must be evaluated from the angle of important requisites for network management. Such an evaluation is made based on metrics that are convenient for the manager, according to his needs. The following are examples of metrics:

- Quantity of octets transited in the manager, in each NE and in each connection; or
- Time spent by management, total and partial.

To obtain the metrics desired for each configuration, this stage provides simulation algorithms adjustable by characteristics of environment such as latency or bandwidth or even characteristics of NEs such as how to be able to determine the processing power or space on the disk. We have some studies, like [RUBINSTEIN 2001], that utilize simulations for calculations of time spent by mobile agents, or [XAVIER 2003], which offer mathematical formulas to calculate the production of octets that pass through various configurations.

As inputs of this stage are:

- Characteristics of links, such as latencies and bandwidths;
- Characteristics of NEs, such as processing power;
- Configurations considered and their characteristics, obtained in previous stage.

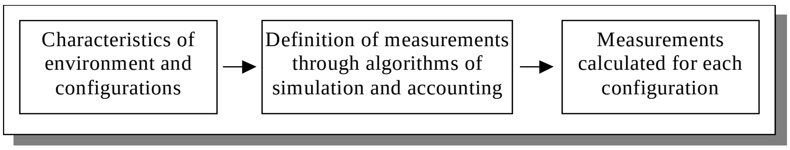

Figure 7. Stages for measurement accounting

Finally, the output of this stage is a set of measurement values for each of the configurations considered (Figure 8). These metrics are a necessary subsidy for the decision process to conclude. A human manager, at this point, can utilize these metrics to opt for the suitable configuration, through his reasoning. Nevertheless, this work further proposes two subsequent stages with the aim of automating the whole decision process.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<metricResults>
      <metric name=”transited octets” unit=”bytes”>
            <scope target=”192.168.0.2”>
                  <data configuration=”a”>29000</data>
                  <data configuration=”b”>29006</data>
                  <data configuration=”c”>34020</data>
            </scope>
            <scope target=”192.168.1.1”>
                  <data configuration=”a”>38029</data>
                  <data configuration=”b”>41334</data>
                  <data configuration=”c”>34020</data>
            </scope>
```

Figure 8. Example of XML document with values obtained by simulation

4.4 Classification process

This stage is responsible for classifying the metrics obtained. The values received for each metric are basically numbers that can be classified by their absolute values with consequent identification of two extremes. Probably one of them is the most and the other the least interesting to the management, depending on the context.

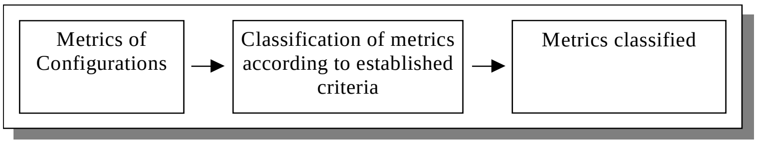

Figure. 9. Stages of configuration classification process

Nevertheless, in human reasoning, comparisons and decisions are not totally absolute. An absolute classification is always an great subsidy, but not considering very close values as possible "technical ties". Despite such bivalent logic ("best" or "worst"), we can introduce concepts of fuzzy logic [BARRETO 1999] in which values of the metrics can be classified for example as "very high", "high", "medium", "low" or "very low", through concepts of fuzzy logic. In this way, close values have the same fuzzy value and the "technical ties" are detected, approximating the decision process of human reasoning.

4.4.1 Classification rules for *fuzzyfication* of metric values

As an alternative to absolute classification, we propose the fuzzyfication of metrics considered within the universe {very high, high, medium, low and very low}, according to the fuzzy pertinence function in Figure 10.

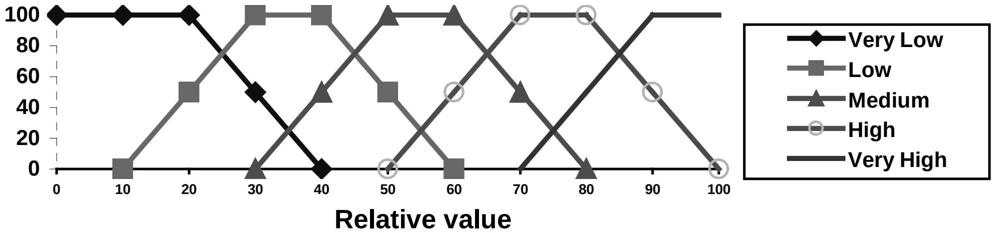

Figure 10. Fuzzy pertinence function

Considering each metric, we attribute to the results, new values of 0 to 100% based on the largest result of the set, obtaining relative values. We then place each of the values on an axis of the ordinates of the fuzzy function, obtaining percent values for each of the fuzzy variables that might be under this point.

```
<?xml version="1.0" encoding="UTF-8"?>
<classificationResults>
      <metric name="transited octets" unit="bytes">
            <scope target="192.168.0.2">
                  <data configuration="a">
                        <absoluteValue>29000</absoluteValue>
                        <relativeValue>85%</relativeValue>
                        <fuzzyVariable name="muito alto" value="67%"/>
                        <fuzzyVariable name="alto" value="67%"/>
                  </data>
```

Figure. 11. Example of XML with results of classification using fuzzy variables

The values of metrics can now be seen as {very high, high, medium, low and very low} accompanied by their percentages (Figure 11), so that the classification can now offer to the next step – that of decision itself – better constructions for operating within decision rules.

4.5 Decision process

This last stage shows the configuration considered to be the most suitable for the intended management. It has as input the metrics classified from all the configurations considered. In fact, the human manager is now able to decide for the most suitable configuration, utilizing for this reasoning similar to that of conditional and relational operations (equal, bigger, smaller), operating within classified metrics. This reasoning becomes the decision process a candidate for paradigms of computational intelligence that deal with decisions based on rules.

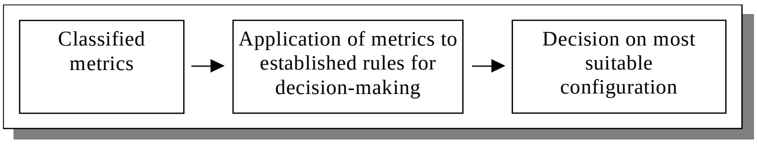

Figure. 12. Stages in decision process

4.5.1 Decision for inference rules

Reasoning in decision-making can be implemented in a sequence of inference rules of the type: “if expression then reaction”. We can draw together a series of rules that translate human reasoning into that decision-making. We have as an example:

a) if (production of octets in manager equal to very low) “or” (management time between NEs equal low) then (suitable configuration);
b) if (production of octets in manager less than low) “and” (management time between NEs lower than medium) then (suitable configuration).

```
<?xml version=”1.0” encoding=”UTF-8”?>
<decisionRules>
	<rule>
		<expression>
			<metric>produced octets</metric>
			<scope>management</scope>
			<relationalOp>greater than</relationalOp>
			<fuzzyValue>high</fuzzyValue>
		</expression>
		<logicalOp>or</logicalOp>
		<expression>
			<metric>transited octets</metric>
			<scope>192.168.0.1</scope>
			<relationalOp>less than</relationalOp>
			<fuzzyValue>very low</fuzzyValue>
		</expression>
		<result>accept</result>
```

Figure 13. Inference rules represented in document XML

The rules are evaluated for each of the configurations considered. The evaluation is made observing the percentage value of the respective fuzzy variable, received from the classification process. Each configuration has its winning rule, which is the highest percentage. In the process of decision-making, the configuration that fulfills the rule with the best percentage was the winner. At the end of the decision process, the result is passed on to the manager, composed of:

- Configuration chosen, with all their constructive details;
- Other configurations, with respective evaluations.

5. Conclusions and future studies

From what we have seen, computer network management has various points of decision, making them candidates for automation through paradigms of computational intelligence. The automation proposed here seeks to decrease manager’s workload, through simulation of his/her own reasoning on some of these points. More than this, it seeks to substitute decision points for the human manager in extremely complex environments and in those of large scale.

All the correlate works investigated apply intelligence in management manipulating data so as to decide on “what” must be done, but do not discuss “how” it must be done. In this study we show a proposal for the utilization of artificial intelligence based on the problem of “how” to implement management in its moments of incursion into the environment. This proposal works through sequential stages of data treatment available to management. These stages modulate the problem and allow for future exchanges of information or knowledge among completely different points of management. Its validation will take place through a prototype, now being developed in the research group.

The human manager can apply the results of this study in several different ways, differentiated by the level of responsibility attributed to the process:

- Total responsibility for the process: In this case the manager automates the entire decision process and the implementation of management, optimizing his/her time to deal with other management activities not yet automated;
- Partial responsibility for the process: in this case the manager verifies the decision taken by this process, confirming or correcting situations that s/he judges necessary;

- Minimum responsibility for process: in this case the manager only consults the information and reports produced by the decision process, as a subsidy for making his/her own decision.

In utilizing this resource in a real environment, it is recommended that the manager begin on the third level of responsibility, verifying his/her own behavior and adjusting to the decision rules and other variables. The transition to the second level and from there to the first will take place gradually, as automation comes to offer the desired degree of reliability or accuracy.

Our present stage of work leads us to the following roads:

- Study alternatives for all the algorithms proposed, notably that of decision;
- Seek environments with execution suitable to the project, since the management would be prejudicated in cases of slow execution. There are studies that utilize clusters or grids to this end, such as [ASSUNÇÃO 2003], who explores the utilization of the intelligent agents in grids;
- Establish an ontology or semantics for the exchange of management information, in particular, that generated by the stages proposed;
- Establish service proposals shared with the management, in which a management nucleus can be the client or collaborator with others. These services may be offered via Web Services with all the aggregated technologies (XML, UDDI, SOAP), guaranteeing interoperability. It is a proposal that seeks for network management what ebXML [EBXML] seeks for electronic commerce.

6. Bibliographical References

[ARANTES 2002] ARANTES, Juliana; WESTPHALL, Carlos; CUSTÓDIO, Ricardo. Modelo Analítico para Avaliar Plataformas Cliente/Servidor e Agentes Móveis Aplicado à Gerência de Redes. Proceedings of 20th Brasilian Symposium of Computer Networks – SBRC 2002. (Vol. I). Búzios (RJ), p. 424-439.

[ASSUNCAO 2003] ASSUNÇÃO, Marcos; WESTPHALL, Carlos; KOCH, Fernando. Grids de Agentes Aplicados à Gerência de Redes de Computadores e Telecomunicações. 21th Brasilian Symposium of Computer Networks – SBRC 2003 – Natal – Brazil – May/2003.

[BARRETO 1999] BARRETO, Jorge M. Inteligência Artificial no Limiar do Século XXI, 2nd Edition. Florianópolis, 1999.

[BOHORIS 2000] BOHORIS, C.; PAVLOU, G.; CRUICKSHANK, H. Using Mobile Agents for Network Performance Management – In: NOMS 2000, Vol 07, pp 637-652. Sep/2000.

[DAS 2002] DAS, S.; SHUSTER, K.; WU, C.,. ACQUIRE: agent-based complex query and information retrieval engine. In: First International Joint Conferente on Autonomous Agents and Multiagents Systems.

[EBXML] Electronic Bussiness using eXtensible Markup Language. In: http://www.ebxml.org.

[HOPE] Hot-Topics em Gerência de Redes. Network and Management Laboratory – UFSC. Florianópolis – SC – Brazil. In: http://www.lrg.ufsc.br/page/hope.pdf.

[KARNOUSKOS 2002] KARNOUSKOS, S.; VASILAKOS, A, Neuro-fuzzy applications: Active electronic mail. In: Proceedings of the 17th Symposium of the 2002 ACM Symposium on Applied Computing.

[KDEMA] The KDEMA Project. In: http://www.lrg.ufsc.br/~kdema. Access in Jan/2003.

[KOCH 2001] KOCH, Fernando; WESTPHALL, Carlos. Decentralized Network Management using Distributed Artificial Intelligence. Journal of Network and Systems Management. Plenum Publishing Corporation. Meddletown, USA (2001, Vol. 9, No. 4). p. 291- 313, Dec/2001.

[OHARE 1996] OHARE, G.; JENNINGS, N. Foundations of Distribuited Artificial Intelligence, A Wiley-Interscience Publication, Nova York, 1996.

[RUBINSTEIN 2001] RUBINSTEIN, Marcelo G. Avaliação do Desempenho de Agentes Móveis no Gerenciamento de Redes. Thesis submited to obtain the Doctor Degree in Electrical Engineering - COPPE/UFRJ. Rio de Janeiro. 2001.

[TAGERE] Tópicos Avançados em Gerência de Redes de Computadores e Telecomunicações. Network and Management Laboratory – UFSC. Florianópolis – SC – Brasil. In: http://www.lrg.ufsc.br/page/tagere.pdf.

[THOTTAN 1998] THOTTAN, Marina; JI, Chuanyi. Proactive Anomaly Detection Using Distributed Intelligent Agents, IEEE Network, Vol 12, pp. 21-27, Sep/1998.

[XAVIER 2002] XAVIER, Edison; KOCH, Fernando; WESPHALL, Carlos. Avaliação de Variações da Configuração de Agentes Móveis na Gerência de Redes. I2TS – Oct/2002. Florianópolis – SC – Brazil.

[XAVIER 2003] XAVIER, Edison. Aplicação de Inteligência Computacional na Gerência de Redes Através da Automatização do Uso de Agentes Móveis. Dissertation submited to obtain the Master Degree in Computer Science. UFSC. Florianópolis. 2003.

[XML 2003] The Extensible Markup Language. In: http://www.w3.org/xml.