



UNIVERSITY OF  
CALGARY

TUTORIAL on

**PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS**

**Prepared for:**  
**Agent Based Software Engineering SENG 609.22**  
**Dr. Behrouz H. Far**

**Prepared by:**  
**Mircea Cobzaru**

**December 20, 2002**

## TABLE OF CONTENTS

<b>ABSTRACT</b> .....	2
<b>1. INTRODUCTION</b> .....	3
<b>2. PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS</b> .....	4
<b>2.1 Software Quality Metrics</b> .....	4
2.1.1 Product measurement and evaluation.....	5
2.1.2 Process measurement and evaluation.....	5
2.1.3 Resource measurement and evaluation.....	6
<b>2.2 Measures for Intelligence</b> .....	7
2.2.1. Abilities of Intelligent Systems as Measures of Intelligence....	7
2.2.2. The Vector of Intelligence.....	9
<b>2.3 Other Considerations on Performance Metrics for         Intelligent Systems</b> .....	10
<b>3. UTILITY FUNCTIONS FOR INTELLIGENT SYSTEMS</b> .....	10
<b>4. CONCLUSIONS</b> .....	11
<b>5. REFERENCES</b> .....	12

## **ABSTRACT**

The process of evaluating performance for intelligent systems is a difficult process due to the complexity of the technologies used in constructing such architectures and environments. This area of science and engineering is quite new, without a complete quantified knowledge, which makes it difficult to use existing evaluation criteria and metrics to establish the advantages and disadvantages of different systems.

This tutorial presents some issues in the software measurement process of intelligent systems and introduces a number of evaluation criteria for these types of systems as a starting point for deriving a set of useful metrics to measure their performance.

## 1 INTRODUCTION

Currently, intelligent systems are a new technique without a complete quantified knowledge about the technology. This tutorial gives an idea of possible aspects in software measurement of intelligent systems and suggests some evaluation criteria for these types of systems. To assess the current state of the science and technology and evaluate the advantages and disadvantages of different systems, we need *metrics*, without which we cannot reward success and penalize failure. The *metrics* are quantitative measures for the evaluation of a specific utility or software quality attributes. [3, 6]. Moreover, science and engineering are not possible without quantitative measurements.

The use of metrics for the software developed with object-oriented and procedural methodologies is a difficult process but when applied to software agents that are supposed to exhibit anthropomorphic characteristics, it becomes even more difficult and complex. That is because the intelligent systems field is not a single area of study, but rather many disciplines, from control theory and neural nets to artificial intelligence and cognitive sciences.

We find that the use of metrics for intelligent systems involves some questions that need to be answered:

- ✓ “What are the proper measures for intelligent systems?”
- ✓ “What is that distinguish intelligent systems from so-called traditional software?”
- ✓ “What can be measured?”
- ✓ “What is the best way to approach a new software paradigm in terms of performance metrics?”
- ✓ “Can we just modify the traditional metrics of the object-oriented system development to match the agent software development process, or do we need to extend them by adding specific performance measures and criteria to get a more comprehensive picture of intelligent systems capabilities?”
- ✓ “How can we compare systems of different design”? (i.e. compare systems based on genetic algorithms with those based on expert systems)
- ✓ “How should we compare systems based on reactive behaviors with those based on planning”?
- ✓ “Should artificial intelligence be compared against human capabilities”?
- ✓ “Is there a set of tests, tools and models for evaluation that can be designed and used”? “Can we agree upon and formalize them”?

This tutorial does not tackle all these questions but provides a brief introduction to performance assessment of intelligent systems and its sub-components taking into account specific measures of performance for the intelligence of multi agent systems.

We proceed from the standpoint of classifying the performance metrics into two classes –*Software quality metrics for intelligent systems* and –*Measures for intelligence* Measuring performance for intelligent system is difficult. It is known fact that agent based developments are one of the most complex construction tasks in terms of the complexity of the constituent components.

The first attempt to measure the performance of an intelligent system was the well-known Turing Test proposed by Alan Turing in 1950 [14], in an attempt to give a mathematical definition of "algorithm" or "mechanical procedure". To pass this test, an Intelligent System must be able to carry on a conversation with a human in a way that is identical to a conversation between two humans. The test is very difficult, and it is not an appropriate metric for the performance of many Intelligent Systems. Besides, it does not evaluate intelligence but rather the ability of a system to *pretend being intelligent*. Nevertheless, the concept of a Turing machine has played an important role in the philosophy of mind, behaviorism, conditions for intelligence granting and other aspects related to the operational definitions of intelligence.

In the following section we will define some evaluation criteria set in order to measure and evaluate the product, process and resources characteristics considering the models of intelligent systems in respect to their particular aspects.

## 2. PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS

In order to evaluate an intelligent system with regard to its performance we have to take into account software quality criteria and characteristic measures related to the intelligence of a given system.

Unfortunately, software quality criteria can neither be easily measured, nor clearly defined and thus the assessment of a certain software product is relative because acceptance criteria depends on the context of use, the purpose for which quality characteristics are being described and even on the user. Standards provide little guidance as to what exactly should be measured and how the results are used to better assess the quality of software so there is no general rule on how measures should be combined to produce an overall assessment of quality [8].

With these ideas in mind, we will not try to evaluate the criteria by assigning them qualitative or quantitative scales, factors and weights but we will present a general assessment framework that divides the performance measures into two categories:

(i) Software quality metrics (derived from several standards such as ISO 9000, ISO 9126, ISO 9294, IEEE, 829 and others) and (ii) Measures for intelligence.

### 2.1 Software Quality Metrics

Generally, we can apply most of the object-oriented software metrics for *product (code) measurement and evaluation*, *process measurement and evaluation* and *resources measurement and evaluation*. We will use the above three classes from software measurement as a guideline for grouping several criteria for agent-based software systems.

### **2.1.1 Product measurement and evaluation**

In this subsection we will take look at some of the software metrics with applicability in intelligent software agent system and give the appropriate references.

#### ***Size estimation***

This measure takes into account the functional size and the physical size of a software agent and also the agent system size, which includes the potential number of (active) agents and their contents; the size is also related to the environment.

Evans [1] proposes the *total number of links* within a network and within special region boundaries.

#### ***Performance level:***

This set of metrics considers the task related performance of an agent system and the ability to realize special tasks

Evans [1] defines some formula to estimate the *response time*, the *time of delay* for the actions of notify, delete, create and commit of software agents.

### **2.1.2 Process measurement and evaluation:**

#### ***Behavior simulation:***

This defines the degree in which the agents' behaviors within the system are implemented related to social behaviors.

The MECCA system [4] implements a component to simulate the agents' scenarios and interactions.

#### ***Modeling of agent-based systems:***

This defines the model on which the construction of agent based system is based in respect to types of agents used and the kinds of interactions. Falchuk [2] describes twenty icons for the different types of software agents and six types for the kinds of interactions for a better usability of the agent-based system modeling.

#### ***ACL evaluation:***

It refers to the ability of the agents in intelligent systems to communicate through common agent communication languages to better achieve collaboration and cooperation. Singh [13] defines some criteria to evaluate agent communication languages for an appropriate use.

#### ***Reasonableness of agent deriving:***

The rationale behind the construction of an agent given an agent template and the possibility of deriving learning capabilities from peer agents is considered.

Joshi [7] defines a measure for the reasonableness to automatically generate exemplars to learn the mapping from a problem to an agent.

### **2.1.3 Resources measurement and evaluation:**

#### ***Middleware evaluation:***

This metric is based on the communication infrastructure used to support flexibility and efficient messaging. Poslad [11] describes an evaluation of the middleware aspects in agent-based systems.

#### ***Vendor evaluation:***

This metric considers the reliability and reputation of the vendors for agent-based components and systems.

Guilfoyle [5] gives an overview about the level of vendors of the agent technology.

#### ***Paradigm evaluation:***

The appropriateness of the chosen software basis and used software components for the implementation of the software agents and the agent-based system is evaluated.

Wong [15] motivates for mobile agent implementation with Java.

## **2.2 Measures for Intelligence**

Intelligent software agents need specific metrics that can measure intelligence if this technology is to rise to the expectations and make an impact on the field of artificial intelligence. One characteristic distinguishing software agents from software developed with object-oriented and procedural methodologies is the intelligence that agents should display.

### **2.2.1 Abilities of Intelligent Systems as Measures of Intelligence**

Measuring intelligence effectively [12] depends on the ability to describe it by:

- ✓ Pro-activeness and goal-orientation
- ✓ Reactiveness (reactive agents)
- ✓ Autonomy (rational agents, and others)
- ✓ Mobility (mobile agents)
- ✓ Learning and reasoning ability (deliberative agents, and others)
- ✓ Social ability: communication and cooperation (multi-agent systems)
- ✓ Making decisions and constructing plans to achieve goals

Newell [10] expands the features that a system must have to qualify as intelligent to more abilities:

- ✓ Recognize and make sense of a scene
- ✓ Understand a sentence
- ✓ Construct a correct response from the perceived situation
- ✓ Form a sentence that is both comprehensible and carrying a meaning of the selected response
- ✓ Represent a situation internally
- ✓ Be able to do tasks that require discovering relevant knowledge

Let us take a look at a few characteristics of an intelligent system based on software agents:

### ***Learning***

Learning is always associated with intelligence and it serves as a critical characteristic of it that determines both the success and failure. An agent is considered intelligent if it can learn from its environment and modify its behaviors and goals to respond to changes in the environment. The ability to learn for an agent is coupled with the ability to perform resource and knowledge discovery. On the other hand, reactive agents that need a quick response time may not have much learning and reasoning because the “operating cost” will not be justified.

### ***Mobility***

For mobile agents the degree of mobility can be a measure of their intelligence because it requires resource discovery. Mobility as a measure of intelligence requires the agents’ ability to autonomously perform processes on remote systems and their ability to restart their execution autonomously in case of interruption without resetting.

### ***Social Ability***

The degree of social interaction and the agents’ ability to exhibit social behavior are an important criterion for multi-agent systems. The type of social interaction between agents affects knowledge acquisition and interpretation. The social model affects the goals and may affect the survival of the system if we consider that intelligent systems have the characteristic of being open.

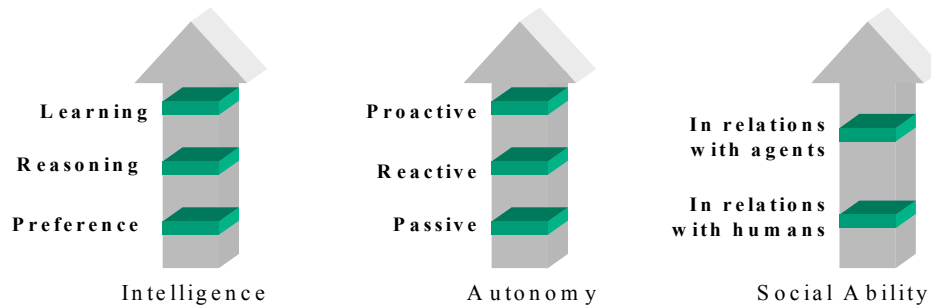
### ***Communication***

Agent-communication languages should enable heterogeneous agents communicate and this brings up the issue of interoperability in respect to a common language or a standard. Another interoperability issue is the lack of a shared content language and ontology. Agent-communication languages such as KQML and FIPA ACL meet the requirements of interoperability but complete standards and specifications are not in place yet.

### ***Ontology***

Ontologies represent an important criterion for the metrics of intelligent software agents, in particular for agents exhibiting the social abilities of communication and cooperation. An ontology expresses the set of terms, entities, objects, and classes and the relationships between them with formal definitions. The use of ontologies contributes to knowledge sharing and reuse across systems. The degree of completeness and consistency of ontologies can provide a quantifiable measure. A shared terminology and syntax is also a step toward achieving the sought interoperability

A few of the agents’ characteristic with regard to their intelligent abilities is summarized in figure 1 where the arrows point in the direction of increased level of intelligence.



**Fig 1 A model of agents' intelligent abilities**

### 2.2.2 The Vector of Intelligence

A different and more complex approach to evaluation of intelligence involves a so-called Vector of Intelligence. This measure has been proposed by Meystel et. al [9] because of the lack of precise measures to evaluate intelligence: the Vector of Intelligence (VI), represents the level of success of the system functioning when this success is attributed to the intelligence of the system. The need to construct a VI emerges in many areas. A possible Vector of Intelligence would be constructed from 25 parameters that set some coordinates for measuring intelligence. Among these we mention just a few:

- ✓ Memory temporal depth
- ✓ Number of objects that can be stored (number of information units that can be handled)
- ✓ Number of levels of granularity in the system of representation
- ✓ Horizon of planning at each level of resolution
- ✓ Horizon of extrapolation at a level of resolution
- ✓ Response time
- ✓ Ability of problem solving intelligence
- ✓ Constraints upon parameters
- ✓ Cost-function of solving the problem

and many other parameters related to problem solving abilities, knowledge and system behavior.

The Vector of Intelligence (VI) is expected to integrate all these parameters of intelligence in a comprehensive and quantitatively applicable form. The set  $\{VI_{ij}\}$  could allow us to require a particular target vector of intelligence  $\{VI_t\}$  find the mapping  $\{VI_t\} \rightarrow \{VI_{ij}\}$  that will provide an intelligent machine with a minimum cost (C) mapping  $[\{VPT\} \rightarrow \{VI_{ij}\}] \rightarrow \min C$ .

Of course this problem is related to measurement only and not to design.

## 2.3 Other Considerations on Performance Metrics for Intelligent Systems

The measurement of the criteria presented is directly related to the experience expressed in values of the measured attributes for the evaluation. The process of measurement has to take into consideration the followings:

- ✓ The *scale type* and the *unit*
- ✓ The *determination of the favorable values* for the evaluation of the measurement component (by discussion in the development team, analyzing and use the examples in the literature, use of the thresholds of the metrics tools, etc.)
- ✓ The *tuning of the thresholds* (as approximation during the software development from other project components, application of a metrics tool for a chosen software product, etc)
- ✓ The *calibration of the scale* (transformation of the numerical scale part to the empirical part)

Generally, the evaluation metrics are of *qualitative* type (nominal and ordinal scales) or of the *quantitative* type (interval or ratio scales).

## 3 UTILITY FUNCTIONS FOR INTELLIGENT SYSTEMS

Utility functions are performance measures that normalize metrics with different values to a common scale to allow for their analysis and integration.

Developers and users of intelligent systems look for measurements of performance of new systems to quantify progress and to identify application benefits. Evaluation of Intelligent System performance is limited by the deficiency of accepted performance metrics. Each of the manifestations of intelligence presented in the previous section can be in principle measured by combined metrics (utility functions) such as **accuracy**, **speed**, **efficiency**, and **cost/benefit ratio**, fact that provides a basis to compare Intelligent Systems over a range of similar application areas and technologies.

**Accuracy** can be defined as the quality of the resulting design, as measured against multiple attributes such as function and life cycle cost.

**Speed** is directly measurable in comparison to the maximum speed feasible with current CPU rates and other limitations in design time reduction imposed by practice.

**Efficiency** will be measured in the improvements seen in a set of business drivers

**Cost/benefit ratio** function is dependent of the relationship between costs in the design process and the manufacturing costs that are determined by design decisions.

Thus, to simplify the problem at a starting point, we can evaluate the performance of Intelligent Systems against these four metrics. For example, *neural nets* would rate highly for speed, efficiency and cost/benefit ratio, due to the ability to acquire knowledge from data rather than the process of knowledge acquisition and

rule generation as required for expert systems. *Expert systems* would be expected to be accurate within their range of expertise, by combining the expertise of multiple human experts. *Expert systems*' cost/benefit is an issue due to costly development processes. Execution speed of *expert systems* would be slower than that of the neural nets due to the rule parsing time-consuming techniques. However, the explanation capabilities provided by *expert systems* are a benefit. *Fuzzy logic* (where suitable to apply) reduces development time compared to expert systems because qualitative relationships are more easily expressed (linguistically). Mixed systems such as fuzzy / neural systems tend to minimize the costs and increase the functionalities of the individual technologies.

#### 4 CONCLUSIONS

Given the complexity of the issues and the diversity of viewpoints regarding intelligent systems' performance measurement, it is obvious that the matter in discussion raises many unsolved problems. It is still possible to refine these ideas to the point of coming up with some useful metrics to measure the performance of intelligent systems.

In the case of humans, performance depends not only on the quality of the mind, but also on the capabilities of the body. Similarly, the performance of intelligent machines depends not only on software, but also on hardware. The mind and body work and progress together. Performance can depend on hardware, computational power, sensors, etc. as well as on algorithms, data structures, and software engineering.

The Intelligent System performance metrics have to be tested to establish their validity over the range of domains that enable Intelligent Systems and their applications.

Indeed, we would like to measure what is really necessary in a practical situation, but in modern software development methodologies such as multi-agent systems this aspect is not quite clear and unanimously agreed upon.

With the increasing number of intelligent systems, users and developers attempt to find the best techniques and qualitative/quantitative measures in order to evaluate the systems' performance and how well they match their requirements.

In conclusion, intelligent systems can be judged as being made of intelligent standalone components, as an intelligent organization or as an environment in which components (agents) execute. The environment should include the knowledge repositories and ontologies, which are some of the key aspects to the agents' degree of intelligence. That is the reason why the measure of intelligence for an agent-based cannot rely on the intelligence of a system controller only and needs to include the intelligence of the environment

## REFERENCES

- [1] Evans, R. et al.: *A Multi-Agent System Architecture for Scalable Management of High Performance Networks: Applying Arms Length Autonomy*. In [Hayzelden 99, pp. 86-111]
- [2] Falchuk, B.; Karmouch, A.: *Visual Modeling for Agent-Based Applications*. IEEE Computer, December 1998, pp. 31-38
- [3] Fenton N. and Pfleeger S.L. *Software Metrics: A Rigorous and Practical Approach* PWS Publishing Company, Boston, MA, 1997
- [4] Gerber, C. et al.: *Resource Adaptation for a Scalable Agent Society in the MoTiV-PTA Domain*. In [Hayzelden 99, pp. 183-206]
- [5] Guilfoyle, C.: *Vendors of Intelligent Agent Technologies: A Market Overview*. In [Jennings 98, pp. 91-104]
- [6] Hudli, R., Hoskins, C., Hudli, A., *Software Metrics for Object-oriented Designs*, IEEE, 1994.
- [7] Joshi, A.; et al.: *Multi-agent system support of networked scientific computing*. Internet Computing, May/June 1998, pp. 69-83
- [8] Kececi N. and Abran A. *Analyzing, Measuring and Assessing Software Quality within a Logical Based Graphical Framework*. Qualita 2001 The Fourth International Congress on Quality and Reliability, Annecy, France, March 22-23, 2001
- [9] Meystel A., Albus, J., Messina, E., Evans, J. Fogel, D., Hargrove W. *Measuring Performance of Systems with Autonomy: Metrics for Intelligence of Constructed Systems* Available at [http://www.isd.mel.nist.gov/conferences/performance\\_metrics/metrics-final.pdf](http://www.isd.mel.nist.gov/conferences/performance_metrics/metrics-final.pdf)
- [10] Newell, A. *The Knowledge Level*. In Artificial Intelligence, 18(1), 87-127, 1982
- [11] Poslad, S. et al.: *Agent-Oriented Middleware for Integrating Customer Network Services*. In [Hayzelden 99, pp. 221-246]
- [12] Pouchard L. *Metrics for Intelligence: the Perspective from Software Agents - Collaborative Technologies* Research Center Computer Science and Mathematics Division Oak Ridge National Laboratory Available at [http://www.cs.utk.edu/~pouchard/pub/Nist\\_workshop.pdf](http://www.cs.utk.edu/~pouchard/pub/Nist_workshop.pdf)
- [13] Singh, M. P.: *Agent Communication Languages: Rethinking the Principles*. IEEE Computer, Dec.1998, pp. 40-47
- [14] Turing A. "Computing machinery and intelligence. Mind", 59(236) 433-460, 1950
- [15] Wong, D.: et al.: *Java-Based Mobile Agents*. CACM, 42(1999) 3, pp. 92-105