CHAPTER 3

GOALS AND EXPERIMENTATION METHODOLOGY

3.1 INTRODUCTION

Multi Agent Systems are found to be the prominent base for large-scale distributed computing environments. The research community is also accepted that the Multi Agent Systems offer enhanced solutions for its predecessors and indeed perform as solution base for its successors. Due to this unique significant feature, MAS entities are becoming the inherent components of its successors and it may be continued forever. Due to these prospective claims of MAS theory, various methodologies and models have been proposed as presented and discussed in the literature survey as in the Chapter 2. On the other side, there are only limited guidelines are proposed to validate the performance of the MASs. In particular, very few quantitative guidelines are proposed to validate the MASs both in terms of system’s (MAS) and applications’ perspectives. From these points of view, the aim of this research is defined to provide versatile set of metrics to assess the MASs in two different perspectives and so as to facilitate the developers in developing the MASs with the promotional features.

Irrespective of the potential facets of MASs, it is also widely accepted that the agent technologies are interesting to the users only if those technologies address the issues of interest to the users. This applicability theory is pertinent for other paradigms also. From this perspective, the experiments of the proposed research are designed such that to accommodate
diversified applications in the MAS environment and thereby to validate the proposed metrics framework on versatile applications. This chapter describes the goals of the research presented in this thesis along with the organization of the experiments carried out for the proposed research.

3.2 GOALS OF THE PROPOSED RESEARCH

In general, MAS can be employed where it is difficult by an individual agent to fulfill the requirements of the intended system (Gutierrez et al 2009). On the other hand, it can be stated that a MAS is a system composed of multiple agents oriented towards a common set of goals and they are suitable for the environments where it is necessary to have distribution of expertise, control and information with respect to the anticipated problem domain (Dorothy et al 1997). So, MAS offers comfortable solution methodologies particularly for distributed problem solving environments with rapid and consistent styles. Here the science of inter and intra-agent communication and coordination would relate communication strategies and indeed they are the pertinent aspects of distributed solution environments and problem-solving efficiency (Dorothy et al 1997).

Based on the types of communication and coordination strategies, there are four types of MAS architectures are devised (Kruchten 1995, Paul et al 2003, Paul et al 2005, Paul et al 2006); centralized synchronous architectures, centralized asynchronous architectures, distributed synchronous architectures and distributed asynchronous architectures. As the studies proved that the performances of MAS architectural styles are mostly of theoretical nature, they often need to be complemented with empirical studies and realizations using instantiations of these styles. In the distributed problem solving environments, the information about appropriate strategy selection may be used either statically or dynamically. In the former one, the strategy selection
issues are available in the expedient fashion which can be readily utilized, whereas in the latter case, everything is based on the evolving problem solving situation. Each scheme has its own theme specific influences and impacts both in the positive and negative dimensions. But, studies proved that the performance aspects of MAS with respect to the strategy selection issues are mostly of a practical nature and they often realized.

On the other hand, from the architectural point of view, all the environments will be treated as same (abstract interaction point of view only) and the architectural levels of discussions involves about different levels of communications rather than the purpose of the communications. Hence the levels of communicational issues in the MASs decide the performance of the overall system and they can be classified into various categories; negotiation, interaction, adaptation, learning, cooperation, collaboration, coordination, exception handling, application specific performance issues, etc…

From the above perspectives, in the research reported in this thesis, it is aimed at offering a generic evaluation framework, which is to assess different levels of communication and different types of coordination aspects in MAS models. The proposed evaluation framework contains different sets of software metrics of variety of measurement models related to communication and coordination aspects of MAS. In this respect, the goals of this research are derived such that to extend the support to the developers to evaluate the MAS at the fine grained levels, rather than at the course grained levels as follows:

IV. Improve the level of granularization of measuring the performance attributes in MASs.

V. Quantify the performance issues of MASs at the Extended Fine Grained Levels.
VI. Improve the performance evaluation schemes of MASs at different hierarchies.

Here the goals are much focused and more quantifiable. The first goal reflects the level of decomposition of the MAS models w.r.t. the performance evaluation and the second goal is measurement-oriented, in which it is supposed to measure the internal components of the outcomes of the first goal. The results of these two goals form a base for the third goal, which depicts the significance of the proposed approach.

In seeking to argue for these goals, it is clear that this work differs in flavor from the majority of the research works. It presents novel measurement schemes along with the new theorems for MAS performance evaluation, which results in quantitative outputs with experimental results. It is very important to note that this work quantifies the performance of the MAS at different hierarchies, which have not been attended by the related works. The intention of these analyses is to provide the quantitative justification theory of precisely why the further one more quantitative evaluation scheme is essential for Multi Agent Systems. The next section describes the empirical study undertaken to pursue these goals and the overall operation performed in this proposed research work.

3.3. EXPERIMENTATION METHODOLOGY

3.3.1 Need for Measurement in MAS

Though it is often argued that MASs offer variety of advantages in dynamic environments where flexible software solutions are needed (Kirn 2006), but exactly this flexibility is the cause for significant drawbacks of MAS in most of the cases, which is due to the unavailability of proper validation methodologies of MASs (Ingo et al 2009). Further it is also stated
that, if the dynamics in the MAS environments are properly validated in all probable environment states, then it is possible to utilize and establish the property of flexibility at the required levels appropriately. This claim is only based on the availability of level specific validation methodologies of MASs. Including flexibility, this is also applicable for all other ability factors of MASs, but on the other side, the availability of level specific validation methodologies and guidelines is so limited (Ingo et al 2009). This potential drawback attracts the researchers to strive to find better validation models for MASs.

It is also discussed that, irrespective of the domain, the measurement process is manifest in the Multi Agent Systems and almost all the research works are centered on this either to contribute to it or to make use of it. The scope for extending the measurement strategies in MASs is highly evident and this can be proved by analyzing the life cycle of an agent in a MAS set up in a very brief manner; the individual performance of an agent in the MAS set up is purely depends on the load offered to that particular agent, the assignment of work load to the individual agent is depends upon the state analysis and negotiation, whereas the negotiation performance depends on the effectiveness of communications and interactions between the agents in the MASs set up.

Hence it can be realized that the scope for fine grained measurement strategies in MASs is so obvious and it can be extended to various levels such as independent and dependent load execution strategies, negotiation, interaction and communication schemes, and etc. Since it is realized that a framework of fine grained measurement schemes for MASs is necessary at this scenario, a systematic experimental methodology has been devised in order to develop and validate the same.
3.3.2 Experimentation Framework

The layered view of the experimentation framework is illustrated in the Figure 3.1. This framework consists of five different layers; attributes layer, mapping layer, metrics layer, execution layer and validation layer. The responsibilities of each layer are defined and described in the following sections.

3.3.2.1 Attributes Layer

Prior to derive the appropriate metrics of validation framework, as a prerequisite, the corresponding anticipated quality attributes of MAS are to be defined and refined at the appropriate levels. Some of these attributes may be specific for each set of applications, e.g., application specific performance related attributes and majority of them are domain independent. This layer defines and classifies the performance attributes of the Multi Agent System in different perspectives. The performance related attributes are classified into two primary categories; communication oriented attributes and coordination oriented attributes.

**Communication Oriented Attributes:** Designers of MASs are responsible for describing the models of communication in order to guide the agent interactions within and outside of the Multi Agent Environments. So, communication models form the base for interaction protocols in the MASs and due to this, almost all of the related researchers of MASs intended to offer much importance towards defining and validating the communication models (Bauer et al 2001, Jiao et al 2005, Gutierrez et al 2009).
<table>
<thead>
<tr>
<th>MAS Performance Attributes</th>
<th>Communication Oriented Attributes</th>
<th>Coordination Oriented Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC&lt;sub&gt;1&lt;/sub&gt;</td>
<td></td>
<td>AT&lt;sub&gt;1&lt;/sub&gt;</td>
</tr>
<tr>
<td>AC&lt;sub&gt;2&lt;/sub&gt;</td>
<td></td>
<td>AT&lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
<tr>
<td>AC&lt;sub&gt;ac&lt;/sub&gt;</td>
<td></td>
<td>AT&lt;sub&gt;at&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mapping Schemes</th>
<th>One To One Mapping</th>
<th>One To Many Mapping</th>
<th>Many To One Mapping</th>
<th>Many To Many Mapping</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Metrics of MAS</th>
<th>Communication Oriented Metrics</th>
<th>Coordination Oriented Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC&lt;sub&gt;1&lt;/sub&gt;</td>
<td></td>
<td>MT&lt;sub&gt;1&lt;/sub&gt;</td>
</tr>
<tr>
<td>MC&lt;sub&gt;2&lt;/sub&gt;</td>
<td></td>
<td>MT&lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
<tr>
<td>MC&lt;sub&gt;ar&lt;/sub&gt;</td>
<td></td>
<td>MT&lt;sub&gt;ar&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Execution and Analysis</th>
<th>Test Bed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing Applications</td>
<td></td>
</tr>
<tr>
<td>Result Analyses</td>
<td></td>
</tr>
</tbody>
</table>

| Validation and Decision Support | Validation over { MC } | Validation over { MT } |

Figure 3.1 Layered view of the Experimentation Methodology
Following this, the work presented in this thesis proposes a set of metrics that account the levels and types of communication activities in several scopes in the MAS environments. Here the corresponding attributes are defined in a set as $\{AC_1, AC_2, \ldots\ldots AC_{ac}\}$ where, ‘AC’ refers to the communication orientated attributes and ‘ac’ is the total number of attributes in the related category.

**Coordination Oriented Attributes:** It is widely accepted that the Multi Agent Systems represents most promising approaches for supporting distributed applications, because of their ability to use multi-agent coordination protocols to dynamically self-organize themselves (Jennings et al 1998, Wooldridge et al 1999). So, in case of MAS environments, the role of coordination is apparent in respect of management oriented responsibilities for both small and large scopes (Nima et al 2005, Rajiv et al 2007). Proofs are there about the individual agents within the groups often may access to local information only, but not able to access the global information and of not able to realize the behavior of neighborhoods (Parrish et al 1999).

Though, the corresponding task completion and final goal achievements are the responsibilities of the individual agents, the individual agents must have the responsibility of offering helping hands to its neighbors in completing the overall goal especially in a MAS environment. But this is not sufficient to perform the set of goal oriented collaborative activities, particularly in MAS environments. In respect to this, a dedicated set of metrics are suggested in order to quantify the levels and variants of coordination activities in different scopes in the MAS environments. Here the corresponding attributes are defined in a set as, $\{AT_1, AT_2, \ldots\ldots AT_{at}\}$ where, ‘AT’ refers to the coordination orientated attributes and ‘at’ is the total number of attributes in the related category.
### 3.3.2.2 Mapping Layer

This layer is responsible for addressing the relationships and deriving the correlations among the performance attributes of MAS, both domain independent and dependent. The output function of this layer is a set of measurement schemes for MAS performance validation. Here the process of mapping follows four possible types of associations; *One-To-One Association, One-To-Many Association, Many-To-One Association and Many-To-Many Association.*

In case of One-To-One Association, single performance attribute may be mapped with the only one measurement scheme and the output can contains only one metric. In the second case, the input function contains only one performance attribute, whereas the output function can contains more than one metrics and the corresponding measurement schemes. In case of Many-To-One Association, more than one performance attributes may be mapped with the only one measurement scheme and the output function can contains the corresponding single metric. In the last case, both the input and output functions contains more than one performance attributes and metrics respectively along with the corresponding measurement schemes.

### 3.3.2.3 Metrics Layer

This layer composite the core activities of this research, whose functions include, defining and deriving the metrics, proposed through this research. This layer produces different sets of metrics, which can be classified in accordance to two different perspectives. As like in the classification schemes in case of performance attributes, here also the corresponding metrics are classified into
two categories; *communication oriented metrics* and *coordination oriented metrics*.

These metrics are all newly defined and derived particularly for MAS evaluation schemes in order to fulfill the intended purpose and they can be described as follows:

\[
NM = \left\{ \left( MC_1, MC_2, MC_3, \ldots, MC_{cr} \right), \left( MT_1, MT_2, MT_3, \ldots, MT_{tr} \right) \right\} \tag{3.1}
\]

where,

- \( NM \) is the overall set of New Metrics of all categories.
- \( MC \) and \( MT \) represent the set of New Metrics w.r.t. the communication oriented attributes and coordination oriented attributes.
- \( cr \) and \( tr \) represent the number of available metrics in the communication and coordination categories respectively.
- At any condition, \( < cr, tr > \neq 0 \). i.e., all performance attributes must be quantified and appropriate measurement scheme(s) should be defined and derived.

These two classes of metrics are to be treated accordingly in the Execution and Validation layers.

### 3.3.2.4 Execution Layer

This layer is responsible for applicability based validation and justification of the proposed metrics. Here, applicability based validation means
the process of ensuring the worthiness of the proposed metrics w.r.t. a Multi Agent Application System (MAAS), which meets the criteria specified for MAS.

Test bed in MAS environments will be normally useful for experimental analysis of relationships between distributed problem structure, distributed problem-solving strategy choices, and problem-solving efficiency (Paul et al 2006, Dorothy et al 1997, Gutierrez et al 2009). In particular, in this work, the experimental analyses are focused towards all the performance attributes as described in the attributes layer. In general, for any kind of validation, particularly for MASs, it is impossible to find single MAAS available that is optimal with respect to all the attributes intended to be validated (Paul et al 2006). Thus, by considering this guideline, a MAAS has to be constructed such that to accommodate all kinds of performance attributes those are discussed in the attributes layer. In order to achieve this, it is proposed to construct a MAAS with the generic definitions and properties of agents’ w.r.t. the standard guidelines of a generic MAS.

In the last few years, many diverse Agent Oriented Software Engineering (AOSE) standards and methodologies are being proposed as discussed in the Chapter 2. Unfortunately, no silver bullet methodology fits for all the needs of any problem domain (Wooldridge et al 1995, Iglesias et al 1999, Behrouz 2003) w.r.t. the MAS computing environments. One way to advance the state of research in agent-oriented methodologies is to define a suitable example problem that can serve as a focal point for discussion and exchange of research ideas and results.
In these points of view, the test bed in this work is defined as a framework with the combination of the properties stated in the attributes layer and this combination is achieved through careful design of the objective framework. It is not possible to construct a framework easily, essentially one which has all the properties by combining the specific features of the methodologies. From the above perspectives, an objective framework is constructed here to solve a specific problem, software testing, which does not comprise any methodology for fabricating the agents but, all the generic features are incorporated in favor of usefulness, which is common for the attempts of new research initiations (Carolyn 1999, David et al 1999, Barbara et al 2002, Dag et al 2002).

Another important design issue which is to be considered with the construction of agents is the agent-architecture. The term ‘agent-architecture’ refers to a particular methodology for building agents and specifies how the agent can be decomposed into a set of component modules (Wooldridge et al 1995). There are two types of agent architectures; deliberative architecture and reactive architecture. The deliberative agent or agent architecture can be defined as the agent that contains an explicitly represented, symbolic model of the world, and in which decisions (for example about what actions to perform) can made via logical reasoning, based on pattern matching and symbolic manipulation.

Reactive agents have their roots in the criticism of deliberative agents as in (Brooks 1991). A reactive architecture is based on considering the behavior of an agent as the result of competing entities trying to get control over the actions of the agent. Reactive agents respond to the present state of the environment in which they are situated and this characteristic is the strength of this approach because the agents do not need to revise their world model as it changes.
However, purely reactive systems suffer from two main limitations: first, reactive agents can conclude the decisions based on local information, not based on the overall systems’ information and so they are not responsible for global behavior; second, the relationship between individual behaviors, environment, and overall behavior is not understandable, which leads to a laborious process of experimentation, trial and error to engineer an agent or MAS.

Hence, for most of the problems, neither a purely deliberative nor a purely reactive architecture is appropriate, but hybrid architectures can combine the merits of both. Typically, these architectures are realized as a number of software layers, each deal with different abstraction levels. Most of the architectures find that three layers are sufficient (Ferguson 1992, Wooldridge et al 1995): a reactive layer, which makes decisions based on raw sensor input; the deliberative layer abstracts away from the raw sensor input and deals with a knowledge-level view of the agent’s environment; the communication layer of the architecture tends to deal with the social aspects of the environment. Coordination with other agents is typically represented in this layer.

So, in this research, the agents are constructed using hybrid architecture with three layers and the distinct functionalities of each layer are defined as follows:

- Reactive layer is responsible for defining the set of agents to be involved in the particular testing session. The definition should be based on the testers’ specification as one of the components of the input from the external world.
• Deliberative layer is responsible for making decisions on number of test cases required based on the input test piece and the history information.

• Communication layer is responsible for coordination and collaboration activities within the multi-agent systems.

In this work, the definitions and descriptions of the agents are based on those properties as discussed above and the works described by Wooldridge et al (1995), Wooldridge (1997), and Jennings et al (2001) as presented in the following paragraphs:

Situatedness: Situatedness or reactiveness of an agent means that the agent receives input from the environment in which it is active and can also effect changes within that environment. Examples of environments in our proposed test bed include resource negotiation in the distributor agents and test case definition based on the defect detector value in the testing agents.

Autonomous: Autonomous system is one that can interact with its environment without the direct intervention of other agents. In the proposed test bed, the testing agents are autonomous since, while the simulations are executed, testing agents act within intervention by a human operator and without instructions from any other agents.

Flexibility: Flexible agents are both responsive and proactive depending on its current situation. A responsive agent receives stimuli from its environment and responds to them in an appropriate and timely fashion. A proactive agent does not simply respond to situations in its environment but is also able to be opportunistic, goal directed, and have appropriate alternatives for various
situations. The distributor agent, for example, in our test bed, would be able to go back to the user with the remaining products with their specifications. They will be transmitted to another distributor for next level of resource negotiation.

Societal: If an agent is social then it can interact, as appropriate, with other software or human agents. An agent is only a part of a complex problem solving process; the interactions of the social agent are oriented towards the goals of the larger multi-agent system. This social dimension of the agent system must address many difficult situations. Examples of interactions include cooperation in working towards a common goal and coordination in organizing problem-solving activity. In the proposed test bed, schemes of distribution of tasks to multiple testing agents and collection of environmental test reports for individual testing agents include this kind of interactions.

Once a designer or a researcher has made the decision to use a MAS, a number of methodologies exist for building multi-agent systems (Brauer et al 2001, Bresciani et al 2001, Caire et al 2001, Rana 2001, Zhu 2001). But unfortunately, in the existing MAS, only few are described with sufficient details to use them for real world problems (Behrouz 2003, Khanh et al 2003). The widely accepted solution is to construct and use objective specific frameworks that are best suited for the given problem space.

This problem-specific framework should allow decomposing the problem domain in terms of autonomous agents that can engage flexible, high-level interactions and decentralization in control. This decentralization, in turn, reduces the system’s control complexity and results in a lower degree of coupling between components. These should initiate the involvement of MAS to represent
the decentralized nature of the problems, the multiple loci of control, the multiple perspectives or the competing interests.

The MAS considers how a particular problem can be solved by a number of agents, which cooperate by dividing and sharing the knowledge about the problem. Based on these observations, here an objective specific multi agent application framework for software testing is developed and used for the experimental purposes to validate and justify both domain independent and domain dependent performance attributes of Multi Agent Systems.

3.3.2.5 Validation Layer

It is the responsibility of the proposers to ensure the worthiness of the proposed metrics in terms of their technicality and applicability, which is nothing but the process of metrics validation (Andrew et al 2010, Lionel et al 1995, Norman 1994, Barbara et al 1995). The applicability and technicality based validations cum justifications over the metrics proposed in this research have to be done in this layer. In general, the technicality based validation includes checking the type of the standard scales and their corresponding transformation functions, ensuring the properties of metrics and confirmation through the provision of objective evidence for specific use (Andrew et al 2010, Barbara et al 1995). The research presented in this thesis follows the above guidelines and performed applicability based validations.

3.4 SUMMARY

The principal concern in this research is to develop a software metrics framework for validating the performance of MASs. A Five layer
experimentation framework is developed for the research narrated in this thesis. Attributes layer defines and classifies the performance attributes of the MAS in different perspectives. The performance attributes are devised into two categories; communication oriented attributes and coordination oriented attributes.

Mapping layer is responsible for addressing the relationships and deriving the correlations among the performance attributes of MAS. Four possible types of association schemes are followed; One-To-One Association, One-To-Many Association, Many-To-One Association and Many-To-Many Association. The output function of this layer is a set of measurement schemes for MAS performance validation.

Metrics layer produces different sets of metrics, which can be classified in accordance to the classification schemes as in the attributes layer. The execution layer is responsible for applicability based validation and justification of the proposed metrics. For applicability based validation, a multi agent based distributed software testing application system, which meets the criteria specified for MAS, has been proposed to construct as a test bed. The validation layer is responsible to ensure the worthiness of the proposed metrics in terms of their applicability in the anticipated environments.