

On the Evaluation of Agent Oriented Methodologies

Luca Cernuzzi
DEI - Universidad Católica “Nuestra Señora
de la Asunción”,
Campus Universitario, C.C. 1683
Asunción, Paraguay
lcernuzz@uca.edu.py

Gustavo Rossi
LIFIA - Universidad Nacional de La Plata
La Plata, Argentina
gustavo@sol.info.unlp.edu.ar

Abstract

An increasing number of methodologies and modelling methods are being proposed in the area of agent-oriented software engineering. However, one of the open problems in order for agent-oriented software engineering to become a “mainstream” is a lack of consensus between the different analyses and design methodologies that have been proposed. Thus, this study proposes a framework to carry out an analysis or evaluation of the agent-oriented analysis and design methodologies.

The proposal takes into consideration qualitative evaluation criteria employing quantitative methods. In order to clarify the proposal, this framework is also applied to a case study, and some interesting aspects are analysed from both a qualitative and a quantitative perspective.

Key words: Agent Oriented Methodologies, Methodologies Evaluation

1. Introduction and motivation

Agent technology has received a great deal of attention in the last decade and is now one of the most active areas of research and development activity in the computing field. However, in spite of the different agent theories, languages, architectures and successful agent-based applications developed, agent-oriented software engineering is still at an early stage of evolution.

The role of software engineering is to provide methodologies (set of methods, models and techniques) that make it easier to handle the complexity of the software development process increasing the quality of the resulting systems [11]. Thus, the role of agent-oriented methodologies is to assist an agent-based application in all of its life cycle phases. An initial comparative analysis of some agent-oriented

methodologies considering this specific perspective may be found in Cernuzzi and Giret [6].

Nowadays, a vast range of agent-oriented methodologies is available for agent-based system designers. The researchers have followed the approach of extending existing methodologies to include the relevant aspects of the agents [15]. These extensions have been carried out mainly in two areas: object oriented (OO) methodologies and knowledge engineering (KE) methodologies.

The following are a representative of the agent-oriented methodologies that take their inspiration from object-orientation: *Agent Oriented Analysis and Design* [5], *Agent Modelling Technique for Systems of BDI agents* [19] (and some extensions proposed by different authors), *MASB* [21], *Agent Oriented Methodology for Enterprise Modelling* [18] and *Gaia* [28]. Moreover, some specific notation for modelling agent-oriented systems have been proposed (i.e. Agent UML [4]). As representative of the agent-oriented methodologies that extend knowledge engineering it is possible to mention the following: *CoMoMAS* [13] and *MAS-CommonKADS* [14]. Moreover, some authors proposed agent oriented methodologies based on a formal specification framework. A good example may be found in using Z schemes for agent specification [20]. A survey of those efforts is presented in [6], [15], [27]. Finally, some interesting emerging methodologies are *TROPOS* [12], [22] and *SODA* [23].

However, as stated in [27], one of the open problems in order for agent-oriented software engineering to become a “mainstream” is a lack of consensus between the different methodologies that have been proposed. Moreover, in most cases, there is not even an agreement on the kinds of concepts the methodology should support. Given this state of affairs, it may be very interesting for agent-based system designers to carry out an analysis or evaluation of the existing methodologies that would be most appropriate to use in each case. An important contribution in this area is the qualitative evaluation proposed by Shehory and Sturm [26].

However, as argued in [17], quantitative data that showed, on a standard set of software metrics, the superiority of the agent-based approach over other approaches simply does not exist. Furthermore, there is no more specific data to show the superiority of an agent-oriented methodology over others.

For all these reasons, the main objective of the present study is to propose a framework for evaluating methodologies, so that agent-based systems designers and the authors of agent-oriented methodologies may carry out the evaluation and accumulation of experience useful both for their own work and for that of other future works as well.

The rest of the paper has been organised in the following manner: chapter 2 presents the framework for evaluating methodologies; chapter 3 presents the application of the framework to a case study considering the comparative analysis of two methodologies; and finally, chapter 4 offers some conclusions and presents some future works.

2. Evaluating methodologies

As previously mentioned, our objective is to find out appropriate forms for evaluating methodologies that support the agent-oriented systems and applications design process. Some results of the process are reflected in the product quality. We must take these aspects into consideration in the evaluation process; that is, some of the criteria will have to refer to characteristics of the product in order to be able to evaluate the methodology.

Important contributions related to the quality of the process may be found in patterns and models like the Capability Maturity Model [24], SPICE [2] and ISO 9000-3 [16]. Their purpose is more general, however, and does not adequately cover the need to identify and measure specific criteria in order to achieve a finer perception of the quality of the methodology under study. Furthermore, CMM and SPICE assess an organisation's use of a methodology, that is how it works in practice in several organisations, more than a comparative evaluation of different methodologies as the present study proposes.

Others interesting frameworks and metrics for evaluating methods and methodologies may be found in object-oriented (see for example [25]) as well as reactive systems [1] disciplines. Some aspects of these works have partially inspired the present study and may help future refinement of our proposal.

Here we present an evaluation framework based on works carried out by different authors [3], [6], [10], [7]. The proposal takes into consideration qualitative evaluation criteria employing quantitative methods.

2.1. A framework for the evaluation of AOSE methodologies

The proposed framework may be represented as in the following figure.

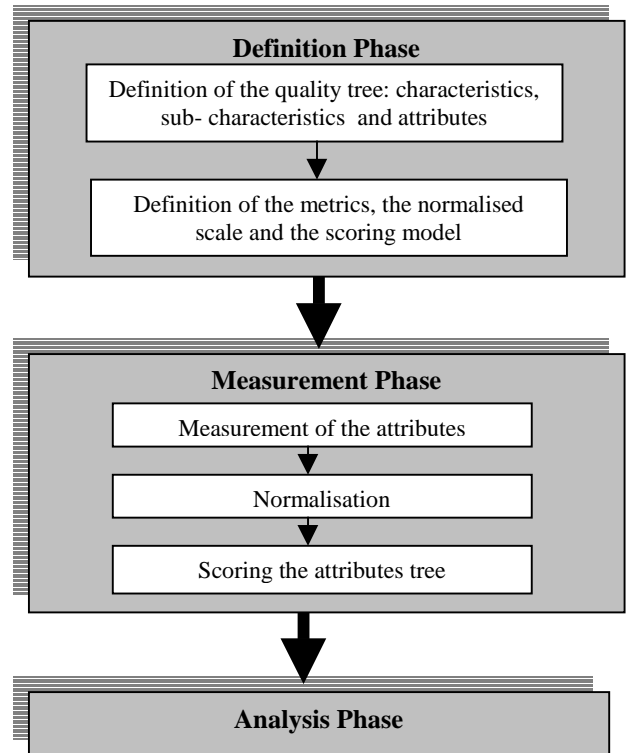


Figure 1. Framework for the evaluation of AOSE Methodologies

Step 1. Specification of an attributes tree. An attributes tree is defined (see section 2.2) in order to identify the more general characteristics and then to specialise them into finer sub-characteristics and attributes to obtain a set of quantifiable ones. This model is the base for measurement in later phases. It is important to observe that the attributes tree may change according to the evaluation goals. This characteristic of the framework offers evaluators a great flexibility to select the most adequate attributes according to specific interest or point of view.

Step 2. Definition of the metrics, the normalised scale and the scoring model [10].

Metric: the observations may correspond to empirical relationships among attributes, qualitative evaluations or quantitative evaluations, depending on the criterion (indicator) and the type of measurement needed. It may be useful to say that only leaves of the attributes tree are evaluated directly. The other values are obtained by indirect observations according to the scoring model. Hereinafter a guideline (set of rules) for assigning

numeric values of each directly valuable measurable attribute is presented.

For each attribute A_i , a variable X_i is associated taking a real value, i.e., the measured value. Normally, the possible result of the evaluation may be continuous (ranging from 0 to 1), discrete, absolute, or average according to the attribute. In the discrete case the rule assigns to the method value 1 if it meets with the attribute; value 0.5 if it partially meets with the attribute; and value 0 (zero) if it does not meet with the attribute. In the absolute case, the amount of items of the attribute is observed. In the average case a formula like the one below is used.

$$F = \frac{\sum_{i=1}^N f(i)}{N} \quad (1)$$

N corresponds to the total amount of items of the observable attribute, $f(i)$ may be the continuous, discrete, absolute or average result obtained for attribute i , and F is the average of attributes presented by the methodology.

Normalised scale: the objective is to define a normalised scale type and a set of rules for mapping the results obtained in the measurement process as they relate to the normalised numeric scale.

A Ratio Scale Type is adopted for the following reasons:

- It preserves the ordering, the size of intervals between entities and the ratios between entities.
- All arithmetic can be meaningfully applied to the classes in the range of the mapping.

To each attribute numeric values may be assigned, in the range of 0 to 10, that may be mapped according to the following rules (the mapping rules should preserve the representation condition [10]):

- If the possible values are continuous (range from 0 to 1) or discrete (0, 0.5 or 1) then they are multiplied by 10.
- For those measurements that are carried out starting from absolute values (count) and mathematical formulae, the simple rule of three formula is applied:

$$M = \frac{V_n \times 10}{May} \quad (2)$$

M represents the mapping result, V_n represents the previously evaluated value, and May represents the maximum evaluated value between methodologies.

- For those measurement that present inverse results (greater value implies worst behaviour), the inverse simple rule of three formula is applied:

$$A = \frac{Men \times 10}{V_n} \quad (3)$$

A represents the mapping result, V_n represents the previously evaluated value, and Men represents the minimum evaluated value between methodologies.

It is important to clarify that the rules previously specified keep the representation condition, since the results obtained in the empirical relationships are preserved when carrying out the mapping [10].

Scoring model: in the attributes tree (see section 2.2) each attribute that is not a leaf of the tree may be measured by means of indirect measurement. Actually, we propose a simple linear model applying the average of immediately direct sub-attributes. This simple model will be improved in future works.

Step 3. Measurement of attributes: it consists in applying the corresponding metric to criteria or indicators of each directly observable attribute or sub-characteristic.

Step 4. Mapping of the relationships to the normalised numeric scale: it allows designers and evaluators to apply a stepwise aggregation mechanism in order to obtain an indicator of more general (indirect) attributes for each competitive methodology or for a single methodology.

Step 5. Scoring the attributes tree (indirect measurement): starting from the indirect measurement it is possible to score a more general variable (characteristic, sub-characteristic or attribute), obtained from the measurement of defined attributes. This linear model is very simple and will probably be replaced by a more powerful aggregation mechanism, like the Logic Scoring of Preference [8]. Moreover, it is possible to associate a priority or specific weights to attributes to better cover significant aspects for the evaluation. However, we leave this consideration for future works.

This step assures designers and evaluators a more general vision of how the methodology supports different perspectives and facilitates the analysis of advantages and drawbacks of the methodologies under evaluation.

Step 6. Analysis of the results: the numeric scoring model facilitates the evaluators in the analysis process in order to highlight positive aspects and point out drawbacks of the methodologies under evaluation. Moreover, it facilitates a comparative evaluation between methodologies.

It is possible to observe that the proposed framework is sufficiently general to assure flexibility to the evaluators and, at the same, time it is sufficiently precise in the steps and the conceptual tools that support it to assure an interesting guide for the evaluation process.

2.2. Attributes Tree

The definition of an attributes tree is one of the most difficult tasks in the framework because the tree represents the basis for all the evaluation process.

In the specialised literature it is impossible to find a consensus about a set of characteristics that every agent-oriented methodology has to cover. However, some suggestions are presented by diverse authors [9], [6], [17], [4], [27], [22], [26]. A lot of those suggestions, enriched by our experience in the construction of agent-based systems, are compiled in the above attributes tree proposal (Table 1). We have decided to group together the attributes considering three different perspectives: those concerning the own characteristics of agents, those referred to the interaction process, and those more directly inherent to the design and development process. For clarity, we then explain each attribute. It seems

evident that a good methodology may offer to agent-based systems designers a set of models, techniques and mechanisms that possibly cover all the attributes in the most exhaustive way.

It may be argued that the attributes tree does not cover all the possible interesting characteristics of the design or implementation process. In effect, some interesting process requirements like security, flexibility, and predictability, have not been included because they are too general and are normally inherent to the run time and implementation phase. So, they depend more specifically upon the adopted development platform than the design methodology. Moreover, many important attributes related with general principle of software engineering have not been included. However, as we observed in step 1 of the framework, a different attributes tree may be used for different evaluation.

Table 1. Attributes tree model

1 Internal attributes	2 Interaction attributes	3 Other process requirements
<i>1.1 Autonomy</i>	<i>2.1 Social ability</i>	<i>3.1 Modularity</i>
<i>1.2 Reactivity</i>	2.1.1 Organisational relationships among agents	3.1.1 Decomposition
<i>1.3 Pro-activeness</i>	2.1.2 Interaction with others agents	3.1.2 Models' dependence
<i>1.4 Mental notions</i>	2.1.2.1 Types of agents interaction	<i>3.2 Abstraction</i>
1.4.1 Beliefs	2.1.2.2 Commitments	3.2.1 Abstraction inside each phase
1.4.2 Goals (Desires)	2.1.3 Conversations with other agents	3.2.2 Existence of design primitives and high level abstraction mechanisms
1.4.3 Intentions	2.1.4 Interfaces with other entities	<i>3.3 System view</i>
	<i>2.2 Interaction with the environment</i>	<i>3.4 Communication support</i>
	<i>2.3 Multiple Control</i>	3.4.1 Clear and precise models
	<i>2.4 Multiple Interests</i>	3.4.2 Systematic transition
	<i>2.5 Subsystems interaction</i>	

Internal attributes

- *Autonomy* [17]; [27]: agents encapsulate some state, and make decisions about what to do based on this state and its own objectives. So, they have control both over their internal state and over their own behaviour.
- *Reactivity* [17]; [27]: agents are able to respond in a timely fashion to changes that occur in their environment.
- *Pro-activeness* [17]; [27]: agents are able to act in anticipation of future goals by taking the initiative.
- *Mental notions* [19]
 - ✓ *Beliefs*: agents have to keep information about the environment, the internal state that may hold and the actions it may perform.
 - ✓ *Goals (desires)*: agents may adopt a set of goals (or desires) that may depend on the actual internal state.

- ✓ *Intentions*: agents may have plans they may possibly employ to achieve their goals or respond to events they perceive.

Interaction attributes

- *Social ability* [27]
 - ✓ *Organisational relationships among agents* [17]: when agents interact there is typically some underpinning organisational context that defines the nature of relationships between agents and influences their behaviour. This context may change during the agent's life. Thus it is important to support simple modifiability to the model.
 - ✓ *Interaction with others agents* [17]: may be necessary either to achieve their individual goals or to manage the organisational dependencies.
 - *Types of agent's interaction*: may vary from information interchanges, to perform a particular action, to co-operation and negotiation or competition, etc.
 - *Commitments* [9]: agents have obligations (conditions to comply) and authorisations about their relationships with other agents.
 - ✓ *Interfaces with other entities* [17]: agents may operate in a more general system composed by other types of entities so it is a need to specify well-defined interfaces.
 - ✓ *Conversations with other agents* [9]: different types of agents' interaction (e.g. negotiation, co-operation, etc.) implies a conversation process and therefore requires some kind of agent-communication language. It is important to capture the conversational messages and to facilitate the identification of conversational protocols used in communication.
- *Interaction with the environment* [17]: agents are situated in a particular dynamic environment; they receive inputs related to the state of their environment and they may modify the environment through effectors.
- *Multiple Control* [17]: interaction between multiple agents implies the administration of multiple loci of control.
- *Multiple Interests* [17]: since agents make decisions at run time, the goal that a specific agent wants to achieve may co-operate, be independent, or enter in conflict with the goals of other agents in the environment. The administration of multiple interests is imperative.
- *Subsystems interaction* [17]: agents may be grouped together into subsystems that may interact between themselves. The interactions within subsystems may be covered by the *Social ability* attributes.

Other process requirements

- *Modularity* [22]: increases efficiency of task execution, reduces communication overhead and usually enables high flexibility. It implies constraints on inter-module communication.
 - ✓ *Decomposition* [17]: the most basic technique for tackling complexity is to divide the large problem into smaller and more manageable parts each of which can then be dealt with in relative isolation.
 - ✓ *Models' dependence*: it is the average of all the relationships between the different models of the methodology. A high dependence on some specific models of a methodology may imply that if they are not well designed it may affect all the design; hence, lower dependence is better.
- *Abstraction* [17]
 - ✓ *Abstraction inside each phase* [7]: the methodologies present different stages, each stage uses defined models that take into consideration aspects that affect exclusively this stage.
 - ✓ *Existence of design primitives and high level abstraction mechanisms* [7]
- *System view* [9]: in order to understand the whole system, a macroscopic system-oriented model is required.
- *Communication support* [7]
 - ✓ *Clear and precise models* [7]
 - ✓ *Systematic transitions* [9], [7]: a good methodology should provide guidelines for simple and elegant transitions between the models.

3. Applying the framework: a case study

In this section a case study is presented with the aim of clarifying the application of the framework. In order to carry out an independent analysis for a specific methodology as well as a comparative analysis, at least two methodologies have to be considered. For this purpose we selected the Agent Modelling Technique for Systems of BDI Agents [19] and the MAS-CommonKADS methodology [14] for two reasons. First, they represent the two mayor tendencies (extensions of OO and KE). Second, we have already used these methodologies in different projects. It is important to say that we consider the original proposal of Kinny, Georgef and Rao methodology and not its extensions that probably reach better evaluation results.

3.1. Agent Modelling Technique for systems of BDI agents

This method [19] defines two main levels (external and internal) for modelling BDI (Belief, Desire and Intention) agents.

The *external viewpoint* consists of the decomposition of the system into agents and the definition of their interactions. This is carried out through two models: the *agent model*, for describing the hierarchical relationship between agents and the relationships between concrete agents; and the *interaction model*, for describing the responsibilities, services and interaction between agents and external systems.

The *internal viewpoint* carries out the modelling of each BDI agent class through three models: the *belief model*, which describes the beliefs about the environment; the *goal model*, which describes the goals and events an agent can adopt or respond to; and the *plan model*, which describes the plans an agent can use to achieve its goal.

3.2. The MAS-CommonKADS methodology

This methodology [14] extends the models defined in *CommonKADS*, adding techniques from object-oriented methodologies and from protocol engineering to describe the agent protocols.

The methodology starts with a *conceptualisation phase* which is an informal phase used to collect the user's requirements and to obtain a first description of the system from the user's point of view.

The methodology defines the following models:

- *Agent Model*: describes the main characteristics of the agents, including reasoning capabilities, skills, services, goals, etc.
- *Task Model*: describes the tasks (goals) carried out by agents, and task decomposition.
- *Expertise Model*: describes the knowledge needed by the agents to carry out the tasks. The knowledge structures distinguishes domain, task, inference and problem solving knowledge.
- *Co-ordination Model*: describes the conversations between agents, that is, their interactions, protocols and required capabilities. The development of the model defines two milestones. The former is intended to identify the conversations and the interactions. The latter is intended to improve these conversations with more flexible protocols such as negotiation and identification of groups and coalitions.
- *Organisation Model*: describes the organisation in which the MAS is going to be introduced and the organisation of the agent society.

- *Communication Model*: details the human-software agent interactions, and the human factors for developing these interfaces.
- *Design Model*: collects the previous models and is subdivided into three sub-models, *application design*, *architecture design* and *platform design*.

3.3. Evaluation of the Selected Methodologies

It is quite intuitive that steps 1 and 2 of the framework are independent of the case study (that is the methodologies under evaluation) while the others are strongly dependent. So, we present in this section just the application of the evaluation of qualitative and quantitative attributes, the mapping of the measurement to the normalised numeric scale and the indirect measurement of other attributes, sub-characteristics and characteristics inferred from those directly evaluated.

For space reasons we use the name AAI instead of Agent Modelling Technique for Systems of BDI Agents.

1. Internal attributes

1.1. Autonomy

AAI: the three models of the internal view cover this aspect. [Evaluation 1](#)

MAS-CommonKADS: through the experience and agent models the methodology covers this aspect. [Evaluation 1](#)

1.2. Reactivity

BDI: the goal model with the plan model satisfactorily cover this aspect. [Evaluation 1](#)

MAS-CommonKADS: is specified in the experience model. [Evaluation 1](#)

1.3. Pro-activeness

AAI: the plan model partially covers this aspect. In effect, it is not possible to specify how to dynamically assume different objectives. [Evaluation 0.5](#)

MAS-CommonKADS: it is possible to model the goals but not the fuzzy and subjective ones, as well as the evolutionary behaviour. [Evaluation 0.5](#)

1.4. Mental notions

1.4.1. Beliefs

AAI: there is a belief model, however it does not allow to model modifications in the beliefs related to the environment evolution. Moreover, it is impossible to model uncertainty since the model is based on first order theory. [Evaluation 0.5](#)

MAS-CommonKADS: it is covered by the expertise model, however it is impossible to model the fuzzy and subjective beliefs. [Evaluation 0.5](#)

1.4.2. Goals (Desires)

AAI: just three types of goals may be modelled: achieve, verify and test. It does not cover

subjective goals and evolutionary behaviour modelling. [Evaluation 0.5](#)

MAS-CommonKADS: through the task model this aspect is covered. [Evaluation 1](#)

1.4.3. Actions (Intentions)

AAIL: the plan model is complete and intuitive for this purpose. [Evaluation 1](#)

MAS-CommonKADS: the task model describes the actions and the methods of problem solving the agent may adopt for each goal. [Evaluation 1](#)

2. Interaction attributes

2.1. Social ability

2.1.1. Organisational relationships among agents

AAIL: the agent model describes the agent class hierarchy. [Evaluation 1](#)

MAS-CommonKADS: the organisational model covers this aspect. [Evaluation 1](#)

2.1.2. Interaction with others agents

2.1.2.1. Types of agents interaction

AAIL: the interaction model specifies the messages and their order. [Evaluation 1](#)

MAS-CommonKADS: the co-ordination model covers satisfactorily the agent interaction. [Evaluation 1](#)

2.1.2.2. Commitments

AAIL: it identifies responsibilities and services of each agent class. [Evaluation 1](#)

MAS-CommonKADS: the co-ordination model specifies the conversations among agents, and starting from there, the operations and services. [Evaluation 1](#)

2.1.3. Conversations with other agents

AAIL: the interaction model specifies the messages and their order. [Evaluation 1](#)

MAS-CommonKADS: the co-ordination model specifies the conversations among agents. [Evaluation 1](#)

2.1.4. Interfaces with other entities

AAIL: it does not cover this aspect. [Evaluation 0](#)

MAS-CommonKADS: the organisation model presents the agents relationships with other objects of the system. [Evaluation 1](#)

2.2. Interaction with the environment

AAIL: agents may know the environment through their sensors and may react according to the stimuli they receive. However, it is not possible to model changes in the beliefs. [Evaluation 0.5](#)

MAS-CommonKADS: the expertise model partially covers this aspect. [Evaluation 0.5](#)

2.3. Multiple Control

AAIL: not covered because it does not model a global state of the agent-based system. [Evaluation 0](#)

MAS-CommonKADS: the co-ordination model covers static aspects, not so the dynamic ones. [Evaluation 0.5](#)

2.4. Multiple Interests

AAIL: just focuses on agent goals considering each agent independently. [Evaluation 0](#)

MAS-CommonKADS: in the expertise model autonomous and co-operative problem solving methods may be distinguished. The latter partially meets the attributes. [Evaluation 0.5](#)

2.5. Subsystems interaction

AAIL: agents' class hierarchy relationships are modelled. However, it does not cover interaction with other sub-systems that are not agents. [Evaluation 0.5](#)

MAS-CommonKADS: the design and organisation models satisfactorily cover this aspect. [Evaluation 1](#)

3. Other process requirements

3.1. Modularity

3.1.1. Decomposition

AAIL: different models facilitate an intuitive use of this aspect. [Evaluation 1](#)

MAS-CommonKADS: different models cover this aspect. [Evaluation 1](#)

3.1.2. Models' dependence

AAIL: Considering the 5 models proposed and the 6 relationships, the average dependence (corresponding to the evaluation results) is [1.2](#)

MAS-CommonKADS: Considering the 7 models proposed and the 11 dependence relationships, the average dependence (corresponding to the evaluation results) is [1.57](#)

3.2. Abstraction

3.2.1. Abstraction inside each phase

AAIL: it contemplates abstraction levels in different phases. However, the resulting architecture is too abstract to be directly implemented and normally needs to be refined with complementary methods. [Evaluation 0.5](#).

MAS-CommonKADS: the first phases contemplate a higher abstraction level which is easily refined in design phase. [Evaluation 1](#)

3.2.2. Existence of design primitives and high level abstraction mechanisms

AAIL: it covers these aspects. [Evaluation 1](#)

MAS-CommonKADS: it covers these aspects. [Evaluation 1](#)

3.3. System view: macroscopic system-oriented model

AAIL: it does not cover this aspect. [Evaluation 0](#)

MAS-CommonKADS: the organisation model offers a global view of the system through the application design. Evaluation 1

3.4. Communication support

3.4.1. Clear and precise models

AAII: it satisfactorily covers this aspect. Evaluation 1

MAS-CommonKADS: it satisfactorily covers this aspect. Evaluation 1

3.4.2. Systematic transitions

AAII: the methods and techniques proposed in the first steps of design processes are related, while there is a lack of adequate orientation and mechanisms to translate from design to implementation. Evaluation 0.5

MAS-CommonKADS: the methodology offers simply transition mechanisms from higher abstraction levels up to the design and the implementation. Evaluation 1

Table 2. Evaluation results

Attributes Tree	Evaluation		Evaluation type	Final Values	
	AAII	MAS-CommonKADS		AAII	MAS-CommonKADS
1 Internal attributes			Average	7.92	8.33
<i>1.1 Autonomy</i>	1	1	<i>Discrete</i>	10	10
<i>1.2 Reactivity</i>	1	1	<i>Discrete</i>	10	10
<i>1.3 Pro-activeness</i>	0.5	0.5	<i>Discrete</i>	5	5
<i>1.4 Mental notions</i>	0.67	0.833	<i>Average</i>	6.7	8.33
1.4.1 Beliefs	0.5	0.5	Discrete	5	5
1.4.2 Goals (Desires)	0.5	1	Discrete	5	10
1.4.3 Intentions	1	1	Discrete	10	10
2 Interaction attributes			Average	3.5	7
<i>2.1 Social ability</i>	0.75	1	<i>Average</i>	7.5	10
2.1.1 Organisational relationships	1	1	Discrete	10	10
2.1.2 Interaction with agents	1	1	Average	10	10
2.1.2.1 Types interaction	1	1	Discrete	10	10
2.1.2.2 Commitments	1	1	Discrete	10	10
2.1.3 Conversations with agents	1	1	Discrete	10	10
2.1.4 Interfaces with other entities	0	1	Discrete	0	10
<i>2.2 Interaction with the environment</i>	0.5	0.5	<i>Discrete</i>	5	5
<i>2.3 Multiple Control</i>	0	0.5	<i>Discrete</i>	0	5
<i>2.4 Multiple Interests</i>	0	0.5	<i>Discrete</i>	0	5
<i>2.5 Subsystems interaction</i>	0.5	1	<i>Discrete</i>	5	10
3 Other process requirements			Average	6.25	9.7
<i>3.1 Modularity</i>			<i>Average</i>	10	8.9
3.1.1 Decomposition	1	1	Discrete	10	10
3.1.2 Models' dependence	1.2	1.57	Formula	10	7.64
<i>3.2 Abstraction</i>	0.75	1	<i>Average</i>	7.5	10
3.2.1 Abstraction inside each phase	0.5	1	Discrete	5	10
3.2.2 Design primitives and abstraction mechanisms	1	1	Discrete	10	10
<i>3.3 System view</i>	0	1	<i>Discrete</i>	0	10
<i>3.4 Communication support</i>	0.75	1	<i>Average</i>	7.5	10
3.4.1 Clear - precise models	1	1	Discrete	10	10
3.4.2 Systematic transition	0.5	1	Discrete	5	10

Starting from the evaluation results presented in Table 2 it is possible to carry out an independent analysis of each methodology as well as a comparative analysis of both. For example, it is quite evident that AAIL poorly covers the Interaction perspective and MAS-CommonKADS in all the perspectives presents better results than AAIL. A deeper and finer analysis of the methodologies is left for future studies.

4. Conclusions and future works

This work proposes a framework for the evaluation of agent-oriented analysis and design methodologies. The proposal is based on works carried out by different authors [3], [6], [10], [7] and takes into consideration qualitative evaluation criteria employing quantitative methods.

This framework that may be used by agent-based systems designer as well as for authors of agent-oriented methodologies contemplates 6 steps (see Figure 1):

1. Specification of an attributes tree model.
2. Definition of the metrics, the normalised scale and the scoring model [10].
3. Measurement of attributes.
4. Mapping of the relationships to the normalised numeric scale.
5. Scoring the attributes tree.
6. Analysis of the results.

For exemplification purposes, an application of the framework to the case of the AAIL [19] and the MAS-CommonKADS [14] has been introduced.

A first important advantage of the framework is that the present proposal was defined not just by considering heuristics but it was defined by employing formal aspects of measurements and metric presented by Fenton and Pfleeger [10]. It is therefore possible to carry out quantitative evaluations and not just qualitative ones. Another important contribution is the attributes tree model. This work presents a synthesis of different proposals introducing finer criteria. In effect, some of the criteria proposed in previous works have been considered indirect attributes and have been refined by means of more specific attributes that may be directly measured. However, it is important to observe that the attributes tree may change according the evaluation goals. For example, looking for a good method to design a single agent, almost all the interaction perspective attributes may be excluded from the evaluation. So, the framework offers evaluators a great flexibility to select the more adequate attributes according to specific interest or point of view.

Moreover, one of the main contributions to the modelling methods evaluation of our proposal is to explicitly convert qualitative evaluated attributes to a normalised numeric value. This conversion facilitates evaluators to score the quality tree and to carry out a

comparative analysis among different modelling methods.

As for future works, different possible lines may be seen:

- a better refinement of the proposed attributes tree;
- several ways to evaluate attributes, considering the type of agent-oriented system-to-be;
- the opportunity to associate a priority or pondered weight to attributes to better cover significant aspects for the evaluation of the quality of a modelling method
- and finally, a more powerful aggregation mechanism.

5. References

- [1] Ardis, M., Chaves, J., Jagadeesan, L.J., Mataga, P., Puchol, C., Staskauskas, M., and Von Olnhausen, J., "A Framework for Evaluating Specification Methods for Reactive Systems", *IEEE Transaction on Software Engineering*, 22 (6), June 1996, pp. 378-389
- [2] Bamford, R. C., and Deibler, W., "Comparing, contrasting ISO 9001 and the SEI Capability Maturity Model", *Software System Consulting. IEEE Computer*, October 1993.
- [3] Basili, V.R., and Rombach, H.D., "The TAME project: Towards improvement – oriented software environments", *IEEE Transactions on Software Engineering*, SE vol. 14 n° 6, 1988, pp. 758-773.
- [4] Bauer, B., Müller, J., and Odell, J., "Agent UML: A formalism for specifying multiagent software systems". In Ciancarini, P., Wooldridge, M. (eds.) *Agent-Oriented Software Engineering - Proceedings of the First International Whorkshop (AOSE-2000)*. Springer-Verlag: Berlin, Germany, 2000.
- [5] Burmeister, B., "Models and methodology for agent-oriented analysis and design", In: Fischer, K. (eds.): *Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems*, DFKI Document D-96-06, 1996.
- [6] Cernuzzi, L., and Giret A., "Methodological Aspects in the Design of a Multi-Agent System", In: *Proceedings of Workshop on Agent Oriented Information Systems. VII National Conference on Artificial Intelligence (AAAI-2000)*. Austin, USA, 2000, pp. 21-28.
- [7] Cernuzzi, L., and González, M., "A Framework for Evaluating WIS Design Methodologies", In Filipe, J., Sharp, B., Miranda, P. (Eds.): *Enterprise Information Systems III*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 198-207.
- [8] Dujmovic, J.J., (1996), *A Method for Evaluation and Selection of Complex Hardware and Software Systems*, The 22nd International Conference for the Resource Management

and Performance Evaluation of Enterprise Computing Systems. CMG 96 Proceedings, Vol. 1, 368-378.

[9] Elammari, M., and Lalonde, W., "An Agent-Oriented Methodology: High-Level and Intermediate Models", In: Proceedings of Workshop on Agent Oriented Information Systems (AOIS-99). Heidelberg, Germany, June 1999.

[10] Fenton N., and Pfleeger S., *Software Metrics: A rigorous and Practical Approach*, PWS Publishing Company, 1997.

[11] Ghezzi, C., Jazayeri, M., and Mandrioli, D., *Fundamentals of Software Engineering*, Prentice Hall, Englewood Cliffs, N. J., 1991.

[12] Giunchiglia, F., Mylopoulos, J., and Perini A., "The Tropos Software Development Methodology: Processes, Models and Diagrams", Proceedings of The Third International Workshop on Agent-Oriented Software Engineering (AOSE-2002), Bologna, Italy, July 2002.

[13] Glaser, N., "Contribution to Knowledge Modelling in a Multi-Agent Framework (the CoMoMAS Approach)", Ph.D. dissertation, L'Universtité Henri Poincaré, Nancy I, France, 1996.

[14] Iglesias, C., Garijo, M., González, J., and Velasco, J., "Analysis and design of multiagent systems using MAS-CommonKADS", In: AAAI'97 Workshop on Agent Theories, Architectures and Languages - ATAL, Providence, RI, 1997.

[15] Iglesias, C., Garijo, M., and González, J., "A Survey of Agent-Oriented Methodologies", In: Müller, J., Singh, M., Rao, A. (eds.): *Intelligent Agents V* (LNAI 1555). Springer-Verlag: Berlin, Germany, 1999.

[16] ISO 9003-3: Quality Management and quality assurance standards-part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software, 1991.

[17] Jennings, N., "On agent-based Software Engineering", *Artificial Intelligence* 117, 2000, pp. 277-296.

[18] Kendall, E. A., Malkoun, M. T., and Jiang, C., "A methodology for developing agent based systems for enterprise integration", In Luckose, D., Zhang, C. (eds.): Proceedings of the First Australian Workshop on DAI. LNAI, Vol. 1087. Springer-Verlag, Berlin, Germany, 1996.

[19] Kinny, D., Georgeff, M., and Rao, A., "A methodology and modelling technique for systems of BDI agents", In: van der Velde, W., Perram, J. (eds.): *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'96*. LNAI, Vol. 1038. Springer-Verlag, Berlin, Germany, 1996.

[20] Luck, M., and d' Inverno, M., "A formal framework for agency and autonomy", In: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95). San Francisco, CA, 1995, pp. 245-260.

[21] Moulin, B., and Brassard, M., "A scenario-based design method and an environment for the development of multiagent systems", In Luckose, D., Zhang, C. (eds.): Proceedings of the First Australian Workshop on DAI. LNAI, Vol. 1087. Springer-Verlag: Berlin, Germany, 1996, pp. 216-231.

[22] Mylopoulos, J., Kolp, M., and Giorgini, P., "Agent-Oriented Software Development", In: Proceedings of the Hellenic Conference on Artificial Intelligence (SETN-02), April, 2002.

[23] Omicini, A., "SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems", Agent-Oriented Software Engineering, 1st International Workshop (AOSE 2000), Limerick, Ireland, June 2000 (Revised Papers available in LNCS 1957, Springer-Verlag, 2001).

[24] Paulk, M., Garcia, S., and Chrissis M., "The continuing improvement of the CMM: Version 2", In: Fifth European Conference on Software Quality, September 1996.

[25] Rossi, M., and Bi, S., "Complexity Metrics for Systems Development Methods and Techniques", *Information Systems* 21(2), 1996, pp. 209-227.

[26] Shehory, O., and Sturm, A., "Evaluation of Modeling Techniques for Agent-Based Systems", In: Proceedings of Autonomous Agents, Montreal, 2001, pp. 624-631.

[27] Wooldridge, M., and Ciancarini, P., "Agent-Oriented Software Engineering: The State of the Art", In Ciancarini, P., Wooldridge, M. (eds.) *Agent-Oriented Software Engineering - Proceedings of the First International Workshop (AOSE-2000)* LNAI Vol. 1957. Springer-Verlag: Berlin, Germany, 2000.

[28] Wooldridge, M., Jennings, N., and Kinny, D., "A methodology for agent-oriented analysis and design". In Proceedings of the Third International Conference on Autonomous Agents (Agents 99), Seattle, WA, May 1999, pp. 69-76.