On The Evaluation Of Agent Oriented Modelling Methods

Luca Cernuzzi[1] and Gustavo Rossi[2]

[1]Universidad Católica "Nuestra Señora de la Asunción", DEI
Campus Universitario, C.C. 1683, Telephone (+595-21) 334650
Asunción, Paraguay
lcernuzz@uca.edu.py
[2]Universidad Nacional de La Plata, LIFIA
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 analysis and design methods that have been proposed. Thus, this study proposes a framework to carry out an analysis or evaluation of the agent-oriented analysis and design modelling methods.

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 Modelling Methods, Modelling Methods Evaluation

1 Introduction and Motivation

Agent technology has received a great deal of attention in the last decade and now is 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 probably 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 [9]. Thus, the role of agent-oriented methodologies is to assist an agent-based application in all of its life cycle phases. A comparative analysis of some agent-oriented methodologies considering this specific perspective may be found in Cernuzzi and Giret [5].

Nowadays, a vast range of agent-oriented methodologies is available for agent-based systems designers. The researchers have followed the approach of extending existing methodologies to include the relevant aspects of the agents [12]. These extensions have been carried out mainly in two areas: object oriented (OO) methodologies and knowledge engineering (KE) methodologies.

As representative of the agent-oriented methodologies that take their inspiration from object-orientation it is possible to mention the following: *Agent Oriented Analysis and Design* [4], *Agent Modelling Technique for Systems of BDI agents* [16], *MASB* [18], *Agent Oriented Methodology for Enterprise Modelling* [15], Gaia [23] and Agent UML [3]. As representative of the agent-oriented methodologies that extend knowledge engineering it is possible to mention the following: *CoMoMAS* [10] and *MAS-CommonKADS* [11]. Moreover, some authors proposed agent oriented methodologies based on formal specification framework. A good example may be found in using Z schemes for agent specification [17]. A survey of those efforts are presented in [5], [12], [22].

However, as stated in [22], one of the open problems in order for agent-oriented software engineering to become a "mainstream" is a lack of consensus between the different analysis and design modelling methods 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 systems designers to carry out an analysis or evaluation of the existing modelling methods that would be most appropriate to use in each case. An important contribution in this area is the work of Shehory and Sturm [21]. However, as argued in [14], quantitative data that showed, on a standard set of software metrics, the superiority of the agent-based approach over other techniques simply does not exist. Moreover, there is no more specific data to show the superiority of an agent-oriented modelling method over others.

For all these reasons, the main objective of the present study is to propose a framework for evaluating modelling methods, so that agent-based systems designers and the authors of agent-oriented modelling methods 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 modelling methods ; and finally, chapter 4 offers some conclusions and presents some future works.

2 Evaluating Modelling Methods

As previously mentioned, our objective is to find out appropriate forms for evaluating modelling methods that support the agent-based systems and application design process. Some results of the process are reflected in the product quality. We must take into consideration these aspects 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 modelling method. Important contributions related to the quality of the process may be found in patterns and models like the Capability Maturity Model [20], SPICE [1] and ISO 9000-3 [13]. 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 modelling method under study.

Before continuing, it is necessary to clarify the meaning of the term evaluation in this context. Evaluation is considered to be the comparison between a dimension and a criterion or its indicators. In our case, the main objective is the usefulness of a specific modelling method, normally associated to a methodology, for the design of agent-based systems and applications, taking into consideration different dimensions and several criteria or indicators that would help evaluators to verify the objective.

Here we present an evaluation framework based on works carried out by different authors [2] [5], [8], [6]. The proposal takes into consideration qualitative evaluation criteria employing quantitative methods.

2.1 A Framework for the Evaluation of AOSE Modelling Methods

Step 1. Application of the paradigm Goal-Question-Metric GQM [2]. The main objective of this stage is to determine exactly what is needed to be measured and which criteria to take into account to reach the prospective objectives.

Objective: The main objective of the evaluation is to: "Highlight the potentiality and weakness of modelling methods for the design and development of Multi-Agent Systems considering different perspectives (i.e. internal or interaction attributes, as well as other process requirements)."

Questions: Considering the main objective some questions that should be answered are: Which criteria are needed to carry out the evaluation of the modelling methods ? How is it possible to classify those criteria under different perspectives?

Step 2. Specification of an attributes tree [8]: Beginning with the results of the GQM, an attributes tree is created (see section 2.2). The objective of creating an attributes tree is to identify the more general criteria and then to specialise them into finer criteria to obtain a set of quantifiable ones. So, it is possible to apply numeric measures (measurements) of all criteria to reach the tree's root. This model is the base for measurement in later phases. It is important to observe that the attributes tree may change according the evaluation goals stated in step 1. This characteristic of the framework offers evaluators a great flexibility to select the more adequate attributes according to specific interest or point of view.

Step 3. Definition of the empiric relationships, and evaluation of qualitative and quantitative attributes:
An attribute tree is defined and direct or indirect evaluations measurements are carried out. The observations may correspond to empiric relationships among attributes, qualitative evaluations or quantitative evaluations, depending on the criterion and the type of measurement needed. It may be useful to remember that only leaves of the attributes tree are evaluated directly. The other values are obtained by indirect observations. 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 continuos (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)=0 if the method does not meet with attribute *i*
f(i)=0,5 if the method partially meets with the attribute *i*
f(i)=1 if the method meets with attribute *i*
F is the average of attributes presented by the method

Step 4. Definition of a normalised scale type and rules to carry out the mapping In this step, the objective is to define a normalised scale type and a group of rules for mapping the results obtained in Step 3 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 [8]):

- If the possible values range from 0 to 1 then they are multiplied by 10.
- If the result is discrete, the value 0 is assigned to that which does not meet the attribute and 10 to that which does.
- For those measurements that are carried out starting from absolute values (count) and mathematical formulas, the simple rule of three formula is applied:

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

M represents the mapping result
$\mathbf{V_n}$ represents the previously evaluated value
May represents the maximum evaluated value between methods

- 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
$\mathbf{V_n}$ represents the previously evaluated value
Men represents the minimum evaluated value between methods

It is important to clarify that the rules previously specified keep the representation condition, since the results obtained in the empiric relationships are preserved when carrying out the mapping [8].

Step 5. Mapping of the relationships to the normalised numeric scale: This allows designers and evaluators to apply stepwise aggregation mechanism in order to obtain an indicator of more general (indirect) attributes for each competitive modelling method or for a single method.

Step 6. Indirect measurement of other attributes and analysis of the results: Starting from the indirect measurement it is possible to infer a latent variable attribute or a more general variable attribute, not directly observable, obtained from the measurement of defined attributes. 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 applying the average of immediately direct sub-attributes. Other aggregation mechanisms, like the Logic Scoring of Preference, may be used. Moreover, it is possible to associate a priority or pondered weight 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 modelling method supports different perspectives and facilitate the analysis of advantages and drawbacks of the methods under evaluation.

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 activities in the framework because the tree represents the basis for all the evaluation process.

Table 1. Attributes tree model

1 Internal attributes	**2 Interaction attributes**	**3 Other process requirements**
1.1 Autonomy	*2.1 Social ability*	*3.1 Modularity*
1.2 Reactivity	2.1.1 Organisational relationships among agents	3.1.1 Decomposition
1.3 Pro-activeness	2.1.2 Interaction with others agents	3.1.2 Models' dependence
1.4 Mental notions	2.1.2.1 Types of agents interaction	*3.2 Abstraction*
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 Actions (Intentions)	2.1.4 Interfaces with other entities	*3.3 System view*
	2.2 Interaction with the environment	*3.4 Communication support*
	2.3 Multiple Control	3.4.1 Clear and precise models
	2.4 Multiple Interests	3.4.2 Systematic transition
	2.5Subsystems interaction	

In the specialised literature it is impossible to find a consensus about a set of characteristics that every agent-based has to cover. However, some suggestions are presented by diverse authors [7], [5], [14], [3], [22], [19], [21]. 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 reasons, we then explain each attribute. It seems evident that a good modelling method may offer to agent-based systems designers a set of models, techniques and mechanisms that possibly cover in the most exhaustive way all the attributes.

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, adaptability, flexibility, and predictability, have not been included because they are too general and normally are inherent to the run time and implementation phase. So, they depend more specifically upon the adopted development platform than the design modelling method . Moreover, many important attributes related with general principle of software engineering have not been included because our actual purpose is oriented to modelling methods. However, as we observed in step 2 of the framework, a different attributes tree may be used for different evaluation.

Internal attributes

- *Autonomy* [14]; [22]: 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* [14]; [22]: agents are able to respond in a timely fashion to changes that occur in their environment.
- *Pro-activeness* [14]; [22]: agents are able to act in anticipation of future goals by taking the initiative.
- *Mental notions* [16]
  - ✓ *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* [22]
  - ✓ *Organisational relationships among agents [14]: 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 agents life thus it is important to support simple modifiability to the model*
  - ✓ *Interaction with others agents [14]: may be necessary either to achieve their individual goals or to manage the organisational dependencies.*
    - *Types of agents interaction*: may vary from information interchanges, to perform a particular action, to co-operation and negotiation or competition, etc.

  - *Commitments* [7]: agents have obligations (conditions to comply) and authorisations about their relationships with others agents.
  - ✓ *Interfaces with other entities [14]: 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 [7]: 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* [14]: 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* [14]: interaction between multiple agents implies the administration of multiple loci of control.
- *Multiple Interests* [14]: 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* [14]: 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* [19]: increases efficiency of task execution, reduces communication overhead and usually enables high flexibility. It implies constraints on inter-module communication.
  - ✓ *Decomposition [14]: 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 modelling method . A high dependence on some specific models of a modelling method may imply that if they are not well designed it may affect all the design; hence, lower dependence is better.*
- *Abstraction* [14]
  - ✓ *Abstraction inside each phase [6]: 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 [6]*
- *System view* [7]: in order to understand the whole system, a macroscopic system-oriented model is required.
- *Communication support* [6]
  - ✓ *Clear and precise models [6]*
  - ✓ *Systematic transitions [7], [6]: a good modelling method 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. For this purpose the Agent Modelling Technique for Systems of BDI (Belief, Desire and Intention) Agents [16] and the MAS-CommonKADS modelling method [11] have been used and compared.
It is quite intuitive that the steps 1, 2, and 4 of the framework are independent of the case study (that is the modelling methods 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 mensuration to the normalised numeric scale and the indirect mensuration of other attributes inferred from those directly evaluated.

For space reasons we use the name BDI instead of Agent Modelling Technique for Systems of BDI Agents and we present just few example of direct attributes evaluation.

- *Pro-activeness*

BDI: 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

- *Beliefs*

BDI: 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

- *Goals (Desires)*

BDI: 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 satisfactorily covered. Evaluation 1

- *Interfaces with other entities*

BDI: 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

- *Multiple Control*

BDI: not covered because it does not model a global state of the agent-based system. Evaluation 0
MAS-CommonKADS: static aspects are covered by the co-ordination model; not so the dynamic ones. Evaluation 0.5

- *Multiple Interests*

BDI: just focuses on agent goals considering each agent independent of the others. 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

- *Subsystems interaction*

BDI: 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

- *Models' dependence*

BDI: Considering the 6 models proposed, the average dependence (corresponding to the evaluation results) is 1.333

MAS-CommonKADS: Considering the 7 models proposed, the average dependence (corresponding to the evaluation results) is 1.714.

- *System view: macroscopic system-oriented model*

BDI: 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

Table 2. Evaluation results

Attributes Tree	**Evaluation** (steps 3)		**Evaluation type**	**Final Values**	
	BDI	*MAS-CommonKADS*		BDI	*MAS-CommonKADS*
1 Internal attributes			**Average**	**7.92**	**8.32**
1.1 Autonomy	*1*	*1*	*Discrete*	*10*	*10*
1.2 Reactivity	*1*	*1*	*Discrete*	*10*	*10*
1.3 Pro-activeness	*0.5*	*0.5*	*Discrete*	*5*	*5*
1.4 Mental notions	*0.67*	*0.83*	*Average*	*6.7*	*8.3*
1.4.1 Beliefs	0.5	0.5	Discrete	5	5
1.4.2 Actions (Intentions)	0.5	1	Discrete	5	10
1.4.3 Goals (Desires)	1	1	Discrete	10	10
2 Interaction attributes			**Average**	**1.7**	**6**
2.1 Social ability	*0.75*	*1*	*Average*	*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
2.2 Interaction with the environment	*0.5*	*0.5*	*Discrete*	*5*	*5*
2.3 Multiple Control	*0*	*0.5*	*Discrete*	*0*	*5*
2.4 Multiple Interests	*0*	*0.5*	*Discrete*	*0*	*5*

2.5Subsystems interaction	*0.5*	*0.5*	*Discrete*	*5*	*5*
3 Other process requirements			**Average**	**6.25**	**9.72**
3.1 Modularity			*Average*	*10*	*8.9*
3.1.1 Decomposition	1	1	Discrete	10	10
3.1.2 Models' dependence	1.33	1.714	Absolute	10	7.8
3.2 Abstraction	*0.75*	*1*	*Average*	*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
3.3 System view	*0*	*1*	*Discrete*	*0*	*10*
3.4 Communication support	*0.75*	*1*	*Average*	*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

Table 2 shows the mapping of the results to the normalised ratio scale as defined in step 4. It is possible to observe that the attributes obtained by indirect measurement, mentioned in step 6, were obtained by averaging the attributes related to each indirect measurement shown in Table 1.

Starting from the results presented in Table 2 it is possible to carry out an independent analysis of each modelling method as well as a comparative analysis of both. For example, it is quite evident that MAS-CommonKADS in all the perspectives presents better results than BDI. Moreover, in the interaction perspective the difference is very pronounced. For space reasons a deeper and finer analysis of the modelling methods is left for future studies.

4 Conclusions and Future Works

This work proposes a framework for the evaluation of agent-oriented analysis and design modelling methods. The proposal is based on works carried out by different authors [2], [5], [8], [6] 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 modelling methods, contemplates 6 steps:

1. Application of the paradigm Goal-Question-Metric [2] to determine the objective of the evaluation.
2. Specification of an attributes tree model [8] that constitutes the base for the measurement process in later phases.
3. Definition of the empiric relationships and evaluation of qualitative and quantitative attributes.
4. Definition of a normalised scale type and rules to carry out the mapping from each mensuration to this scale.
5. Mapping of the relationships to the normalised numeric scale.
6. Indirect mensuration of other attributes inferred from those directly evaluated and analysis of advantages and drawbacks of the evaluated methods.

For exemplification purposes, an application of the framework to the case of the Agent Modelling Technique for Systems of BDI Agents [16] and the MAS-CommonKADS modelling method [11] has been introduced.

A first important advantage of the framework is that the present proposal was defined not just considering heuristics but it was defined employing formal aspects of mensurations and metric presented by Fenton and Pfleeger [8]. 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. 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 facilitate agent-oriented systems designers to obtain evaluation of indirect attributes and to carry out a comparative analysis among different modelling methods.

As for future works, at least three possible lines may be seen:

– a better refinement and formalisation of the proposed evaluation framework;
– the application of the framework to several cases that cover different types of modelling methods with emphasis on more recent methodologies and comparative studies;
– and finally, 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.

References

1. Bamford, R. C., Deibler ,W.: Comparing, contrasting ISO 9001 and the SEI Capability Maturity Model. Software System Consulting. IEEE Computer, October (1993)
2. Basili, V.R., Rombach, H.D.: The TAME project: Towards improvement – oriented software environments. IEEE Transactions on Software Engineering, SE vol. 14 nº 6 (1988) 758-773
3. Bauer, B., Müller, J., Odell, J.: Agent UML: A formalism for specifying multiagent software systems. In Ciancarini, P., Wooldridge, M. (eds.) Agent-Oriented Software Engineering - Proccedings of the First International Whorkshop (AOSE-2000). Springer-Varlag: Berlin, Germany (2000)
4. 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)
5. Cernuzzi, L., 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) 21-28
6. Cernuzzi, L., González, M.: A Framework for Evaluating WIS Design Methodologies. (to be published) In Filipe, J., Sharp, B., Miranda, P. (Eds.): Enterprise Information Systems III. Kluwer Academic Publishers, Dordrecht, The Netherlands (2002)

7. Elammari, M., 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)
8. Fenton N., Pfleeger S.: Software Metrics: A rigorous and Practical Approach. PWS Publishing Company (1997)
9. Ghezzi, C., Jazayeri, M., Mandrioli, D.: Fundamentals of Software Engineering. Prentice Hall, Englewood Cliffs, N. J. (1991)
10. 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)
11. Iglesias, C., Garijo, M., González, J., Velasco, J.: Analysis and design of multiagent systems using MAS-CommonKADS. In: AAAI'97 Workshop on Agent Theories, Architectures and Languages - ATAL, Providence, RI. (An extended version of this paper has been published in INTELLIGENT AGENTS IV: Agent Theories, Architectures, and Languages, Springer Verlag, 1998) (1997)
12. Iglesias, C., Garijo, M., 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)
13. 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)
14. Jennings, N.: On agent-based Software Engineering. Artificial Intelligence 117 (2000) 277-296
15. Kendall, E. A., Malkoun, M. T., 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, Germay (1996)
16. Kinny, D., Georgeff, M., 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)
17. Luck, M., 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) 245-260
18. Moulin, B., 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) 216–231
19. Mylopoulus, J., Kolp, M., Giorgini, P.: Agent-Oriented Software Development. In: Proceedings of the 2nd Hellenic Conference on Artificial Intelligence (SETN-02), April (2002)
20. Paulk, M., Garcia, S., Chrissis M.: The continuing improvement of the CMM: Version 2. In: Fifth European Conference on Software Quality, September (1996)
21. O. Shehory and A. Sturm, Evaluation of Modeling Techniques for Agent-Based Systems, In: Proceedings of Autonomous Agents, Montreal, (2001) 624-631
22. Wooldridge, M., 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-Varlag: Berlin, Germany (2000)
23. Woolddridge, M., Jennings, N., 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) 69-76