# A Critical Review of the Evaluation of Framework for Agent Oriented Methodologies

**BY**

[1] **Omankwu, Obinnaya Chinecherem**
**Department of Computer Science,**
**Michael Okpara University of Agriculture,**
**Umudike Umuahia, Abia State, Nigeria.**
***saintbeloved@yahoo.com***

[2] **Anigbogu, S.O.**
**Department of Computer Science,**
**Nnamdi Azikiwe University,**
**Awka Anambra State, Nigeria.**
**dranigbogu@yahoo.com**

[3] **Nwagu, Kenneth Chikezie**
**Department of Computer Science,**
**Nnamdi Azikiwe University,**
**Awka Anambra State, Nigeria,**
**Nwaguchikeziekenneth@hotmail.com**

[4] **Anigbogu, Kenechukwu,**
**Department of Computer Science,**
**Anambra State Polytechnic**
**Mgbakwu Awka, Nigeria.**

## Abstract

Multiple agent-oriented methodologies were introduced in recent years. However no systematic evaluation of these was offered. In this work we presented an evaluation framework for agent-oriented methodologies: The review of this evaluation framework focused on four major facets of a methodology, namely: Concepts and Properties, Notations and Modeling techniques, Development Process, and Pragmatics. In analyzing the results, the author recognized that the mentioned facets of methodology need further improvements within the existing agent-oriented methodologies.

## 1.0 Introduction

In the last decade, many methodologies for developing agent-based systems have been developed. A methodology is the set of guidelines for covering the whole lifecycle of system development both technically and managerially. A methodology, according to Graham et al., (2017), should provide the following: a full lifecycle process; a comprehensive set of concepts and models; a full set of techniques (rules, guidelines, heuristics); a fully delineated set of deliverables; a modeling language; a set of metrics; quality assurance; coding (and other) standards; reuse advice; and guidelines for project management. The relationships around these components are shown in Figure 1.1. In figure 1.1, the UML notations are used to depict the relationships around the components. As depicted in the said figure, a methodology consists of a set of techniques, a modeling language and a lifecycle process. The set of techniques consists of metrics, quality assurance (QA) activities, a set of standards and tools. The modeling language comprises notations and a meta model. The lifecycle process consists of project management, a number of roles (e.g., an analyst or a designer), a number of procedures (e.g., how to move between development stages), and a number of deliverables (e.g., a design document, source code). In addition, Figure 1.1 also shows that the tools can be based on the Meta model of the modeling technique and they represent the modeling technique's notations. The deliverables use the modeling technique.

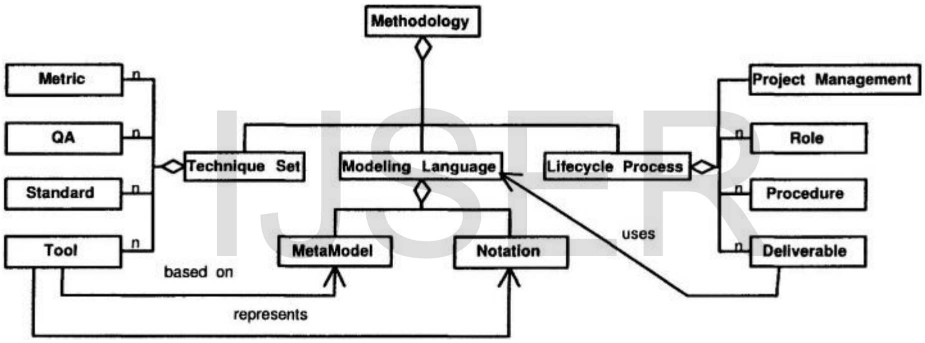

Figure 1.1. The components of a methodology and the relationships among them

## 2.0 Literature Review.

At present, more than twenty four agent-oriented methodologies exist. The multiplicity and variety of methodologies result in the following problems (Sturm and Shehory, 2003).

*(i)* Industrial problem: selecting a methodology for developing an agent-based system/application becomes a non-trivial task, in particular for industrial developers which hold specific requirements and constraints (Cernuzzi and Rossi, 2002);

*(ii)* Standards problem: multiple different methodologies are counter-productive for arriving at a standard. With no standard available, potential industrial adopters of agent technology refrain from using it (Sturm and Shehory, 2003).

*(iii)* Research problems: excessive efforts are spent on developing agent-oriented methodologies, in times producing overlapping results.

Additionally, as a result of allocating resources to multiple methodologies, no methodology is allocated sufficient research resources to enable addressing all facets and providing full-fledged agent-based methodology (Cernuzzi and Rossi, 2002).

A few evaluation of agent-oriented methodologies have been suggested. In (Yu and Cysneiros, 2002), the authors set a list of questions that a methodology should address. However, neither evaluation nor a comparison has been performed using that set. Another study (Cernuzzi and Rossi, 2002) suggested a framework for evaluating agent-oriented methodologies. That framework uses a set of evaluation criteria to examine methodologies' expressiveness. However, it does not examine other properties encompassed within the methodology definition. In (Kumar, 2002), the author performed an evaluation of five agent-oriented methodologies, but, referred only to some supported concepts such as organization design and cooperation. He did not refer to the broad set of attributes that constitute a complete methodology. In (Shehory and Sturm, 2001), the authors performed an evaluation of the modeling part within a methodology, while other parts which are concept and property, and pragmatics not evaluated.

In Dam and Winikoff, (2003), three methodologies were compared: MaSE Prometheus and ROADMAP. The comparison was performed by gathering feedback regarding the properties of the methodologies from students that used them, and from the methodologies' developers. The gathered feedback included several inconsistent answers. This resulted in difficulty in analyzing methodology properties.

Many studies that dealt with evaluating agent-oriented methodologies compared two or three methodologies, mainly with respect to the expressiveness of the methodologies and their supported concepts, and not with respect to other software engineering criteria.

The evaluation framework used in this study was based on a feature analysis technique. In other word, the features of each of the examined methodologies were evaluated. The evaluation was performed based on information regarding the examined methodologies available in publications. The framework's four facets were: concepts and properties, Notations and Modeling Techniques, Development Process, and Pragmatics. These facets, and the metric used in conjunction with them, are introduced below.

## 3.0 Evaluation of The different Agent Oriented Framework

The evaluation framework used in this work is based on a feature analysis technique. That is, the features of each of the examined methodologies are evaluated. The evaluation is performed based on information regarding the examined methodologies available. The framework's four facets are: Concepts and Properties, Notations and Modeling Techniques, Development Process, and Pragmatics

## Concepts and Properties

A concept is an abstraction or a notion inferred or derived from specific instances within a problem domain. A property is a special capability or a characteristic (Cernuzzi and Rossi, 2002). This section deals with the question of whether a methodology addresses the basic notions (concepts and properties) of agents and Multi Agent System. The following are the concepts according to which an agent-oriented methodology should be evaluated:

1. Autonomy: is the ability of an agent to operate without supervision;

2. Reactiveness: is the ability of an agent to respond in a timely manner to changes in the environment;

3 Proactiveness: is the ability of an agent to pursue new goals; and

4. Sociality: is the ability of an agent to interact with other agents by sending and receiving messages, routing these messages, and understanding them.

Meanwhile, the following are the building blocks that encompass the basic components of Multi Agent System (MAS). These building blocks are based on the work of Sturm and Shehory, (2003).

1. Agent: is a computer program that can accept tasks, can figure out which actions to execute in order to perform these tasks and can actually execute these actions without supervision. It is capable of performing a set of tasks and providing a set of services.

2 Belief: is a fact that is believed to be true about the world.

3 Desire: is a fact of which the current value is false and the agent (that owns the desire) would prefer that it be true. Desires within an agent may be contradictory. A widely used specialization of a desire is a goal. The set of goals within an agent should be consistent.

4. Intention: is a fact that represents the way of realizing a desire. Some-times referred to as a plan.

5. Message: is a means of exchanging facts or objects between entities.

6. Norm: is a guideline that characterizes a society. An agent that wishes to be a member of the society is required to follow all of the norms within. A norm can be referred to as a rule.

7. Organization: is a group of agents working together to achieve a common purpose. An organization consists of roles that characterize the agents, which are members of the organization.

8. Protocol: is an ordered set of messages that define the admissible patterns of a particular type of interaction between entities.

9. Role: is an abstract representation of an agent's function, service, or identification within a group.

10. Society: is a collection of agents and organizations that collaborate to promote their individual goals.

11. Task: is a piece of work that can be assigned to an agent or performed by it. It may be a function to be performed and may have time constraints.

## Notations and Modeling Techniques

Notations are technical system of symbols used to represent elements within a system. A modeling technique is a set of models that depict system at different levels of abstraction and different system's facets including structural and behavioral facets as stated by Shehory and Sturm, (2001). This section deals with the properties to which methodology's notations and modeling techniques should adhere. The list of these properties is adopted from Shehory and Sturm, (2001).

1 Accessibility: is an attribute that refers to the ease, or the simplicity, of understanding and using a method. It enhances both experts and novices capabilities of using a new concept.

2. Analyzability: is a capability to check the internal consistency or implications of models, or to identify aspects that seem to be unclear, such as the interrelations among seemingly unrelated operations. This capability is usually supported by automatic tools.

3. Complexity management (abstraction): is an ability to deal with various levels of abstraction (i.e., various levels of detail). Sometimes, high-level requirements are needed, while in other situations, more detail is required. For example, examining the top level design of a Multi Agent System (MAS), one would like to understand which agents are within the system, but not necessarily what their attributes and characterizations are. However, when

concentrating on a specific task of an agent, the details are much more important than the system architecture.

4. Executability (and testability): is a capability of performing a simulation or generating a prototype of at least some aspects of a specification. These would demonstrate possible behaviors of the system being modeled, and help developers determine whether the intended requirements have been expressed.

5. Expressiveness (and applicability to multiple domains): is a capability of presenting system concepts that refers to:

- The structure of the system;
- The knowledge encapsulated within the system;
- The system's ontology;
- The data flow within the system;
- The control flow within the system;
- The concurrent activities within the system (and the agents);
- The resource constraints within the system (e.g., time, CPU and memory);

- The system's physical architecture;
- The agents' mobility;
- The interaction of the system with external systems; and
- The user interface specification.

6. Modularity (incrementality): is the ability to specify a system in an iterative incremental manner. That is, when new requirements are added it should not affect the existing specifications, but may use them.

7 Preciseness: is an attribute of disambiguity. It allows users to avoid misinterpretation of the existing models.

A development process is a series of actions that, when performed, result in a working computerized system. This section deals with the process development facet of a methodology. This facet is evaluated by examining the following:

1 Development context: specifies whether a methodology can be used in creating new software, reengineering or reverse engineering existing software, prototyping, or designing for or with reuse components.

2 Lifecycle coverage: specifies what elements of software development are dealt with within the methodology. Each methodology may have elements that are useful in several stages of the development lifecycle. Here, the lifecycle stages are defined as follows: requirements' gathering, analysis, design, implementation, and testing.

Again having the development stages defined is not sufficient to render a methodology usable. A methodology should further elaborate the activities within the development lifecycle. Providing a detailed description of the activities included in the development lifecycle would enhance the appropriate use of a methodology and increase its acceptability as a well-formed engineering approach and to verify that a methodology provides detailed activity descriptions, we need to examine the details of the development process. This verification can be performed by answering the following questions regarding an evaluated methodology:

1 What are the activities within each stage of a methodology? For example, an activity can be the identification of a role, a task, etc. The methodology may consist of heuristics or guidelines helping the developer to achieve his/her system development goals.

2. Does the process provide for verification? This question checks whether a methodology has rules for verifying adherence of its deliverables to the requirements.

3 Does the process provide for validation? This question checks whether a methodology has rules for validating that the deliverables of one stage are consistent with its preceding stage.

4 Are quality assurance guidelines supplied?

5 Are there guidelines for project management?

**Pragmatics**
Pragmatics refers to dealing with practical aspects of using a methodology. This section deals with pragmatics of adopting the methodology for a project or within an organization. In particular, the framework suggests examining the following:

1.Resources: These are the (publicly available) publications describing in detail the methodology (e.g., textbooks and papers), users' groups, training and consulting services offered by third parties and automated tools (CASE tools) available in support of the methodology (e.g., graphical editors, code generators, and checkers).

2 Required expertise: This is the required background of those learning in agent oriented methodology. A distinguishing characteristic of many

methodologies is the level of mathematical sophistication required to fully exploit the methodology. A criterion within the required expertise may check the required knowledge in some discipline (Ardis, et.al. 2012)..

3. Language (paradigm and architecture) suitability: This is the level to which the methodology is coupled with a particular implementation language (e.g., object oriented programming language) or a specific architecture (e.g., BDI).

4. Domain applicability: This indicates the level of suitability of a methodology to a variety of domains (e.g., information systems, real-time systems).

5. Scalability: This is the ability of the methodology to be adjusted to handle various application sizes. For example, can it provide a lightweight version for simple problems.

And to enable ranking of the properties examined in the evaluation process, the framework proposes a scale of 1 to 7 with the following interpretations:

1. An Indication that the methodology does not address the property.

2. An indication that the methodology refers to the property but no details are provided.

3. An indication that the methodology addresses the property to a limited ex-tent. That is, many issues that are related to the specific property are not addressed.

4. An indication that the methodology addresses the property, yet some major issues are lacking.

5. An indication that the methodology addresses the property, however, it lacks one or two major issues related to the specific property.

6. An indication that the methodology addresses the property with minor deficiencies.

7 An Indication that the methodology fully addresses the property.

Thus far, we have described the evaluation framework, its evaluation criteria, and it’s metric.

## 4.0 Summary and Conclusion

We have reviewed the evaluation of Feature Based Analysis as a framework that examines the various facets of an Agent Oriented methodology. The results of that reviewed showed that Feature based Analysis framework covered the four major Agent Oriented Methodology facets. We summarize the evaluation thus (Ardis, et.al. 2012).

1 Concepts and properties: the autonomy and proactiveness criteria are properly addressed by the framework. The reactiveness is lacking in the sense that the connection between events and responses to them is not well specified. The sociality deals with organization rules, but not with multiple organizations or societies structure, nor with role hierarchy. The building blocks coverage is good; but none of the frameworks covers all of them. This coverage varies among the frameworks as a result of their different goals(Ardis, et.al. 2012).

2. Notations and modeling techniques: this facet was addressed to a limited extent. The accessibility of the framework was good even though it requires further enhancements. Current limitations resulted from the multiplicity of models and the use of logic within the specification stages(Ardis, et.al. 2012).

3. Pragmatics: Frameworks were not coupled to a specific programming language or an agent architecture, and therefore can be used for multiple domains. Scalability (i.e., the ability to be adjusted according to a specific project needs) was not supported by the methodologies (Ardis, et.al. 2012).

In conclusion, the examined framework provided an appropriate infrastructure. The framework considered four major aspects of methodologies namely: Concepts, Notations, Process, and Pragmatics. Each of these areas defined proper evaluation criteria regarding the methodology aspects in general and the agent-orientation concepts in particular. This framework can be utilized for identifying the strengths and weaknesses of agent-oriented methodologies, that for selecting methodologies for application development. It can also be used for promoting existing methodologies which may advance the acceptability of agent technology by introducing a mature, well-structured engineering approach.

Again the outcome of this review showed a need for further research and improvements. The agent-oriented methodologies evaluation frameworks, in order to promote and help in arriving at industry-grade methodologies. The evaluation performed in this work provided researchers and practitioners with a detailed framework among other agent-oriented methodology frameworks. The framework used in this study may be utilized by others to evaluate and compare other methodologies as needed.

## References

Allan, W., & Kurt, C. (2012). *A framework for evaluating software technology*. IEEE Computer Society Press.

Allan, M., & Wallnau, T. (2012). *The Rational Unified Process: An Introduction.* Addison- Wesley Pub Co.

Arazy, O., & Woo, C. (2012.). Analysis and design of agent-oriented information systems. *The Knowledge Engineering Review,* 17(2).

Ardis, A., John A., & James V. (2012). A framework for evaluating specification methods for reactive systems: Experience report. *In International Conference on Software Engineering,* pages 159-168.

Avison, D., & Fitzgerald, G. (2013). *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill: New York.

Avison, D., & Fitzgerald, G. (2011). *Information Systems Development: Agent Oriented Methodologies, Techniques and Tools.* McGraw-Hill: New York.

Barbara, A., Drogoul, A., & Benhamou, P. (2016). Agent-oriented design of a soccer robot team. In Proceedings of the Second International Conference on *Multi-Agent Systems*. Menlo Park, CA: American Association for Artificial Intelligence.

Barbara K., (2012). DESMET: a method for evaluating software engineering methods and tools. *Technical Report TR96-09*, University of Keele, U.K.

Bauer, B., & Odell, J. (2005). UML 2.0 and agents: how to build agent-based systems with the new UML standard, *Journal of Engineering Applications of Artificial Intelligence* Vol. 18, Issue 2, 2005.

Belina, F., Hofgrefe, D. & Sarma, A.,(2011). *SDL with Applications from Protocol Specification*", Prentice Hall Int., Hertfordshire, UK, 1991.

Behling, K., (2010). *Project Management – Theory and Practice*. McGraw-Hill professional.

Berard E. (2012). A comparison of object-oriented methodologies. *Technical report*, Object Agency Inc.

Bernon, C., & Glize, P. (2012). *The ADELFE methodology for an intranet system design*. Washington, DC: Cato Institute.

Berard E. (2011).*A comparison of object-oriented methodologies*, Technical report,Object agency Inc.

Bobkowska, A. (2005). *Framework for methodologies of visual modeling language evaluation*, Proceedings of the symposia on Metainformatics, ACM Press 2005.

Braubach, L., Pokahr, A., Moldt, D. and Lamersdorf W.(2005).*Goal representation for BDI agent systems*", In R. Bordini, M. Dastani, J. Dix . and A. El Fallah Seghrouchni, editors, Programming Multi-Agent Systems, second Int. Workshop (ProMAS'04), vol. 3346 of LNAI, Pages 44–65. Springer Verlag,.

Brazier, F., Jonker, C. & Treur, J. (2010).*Principles of compositional multi-agent system development*", In Proceedings of Conference on Information Technology and Knowledge Systems, Pages 347–360. Austrian Computer Society.

Bresciani, P.,Giorgin,N., Hiunchiglia, R., Mylopoulos,K., & Perini, A. (2014).*Tropos: An agent- oriented software development methodology*. Autonomous Agents and Multi-Agent Systems.

Bresciani, P., & Perini, A. (2010).Tropos: *An agent-oriented software development methodology*. Autonomous Agents and Multi-Agent Systems.

Buhr, R.,(2008). *Use Case Maps as Architectural Entities for Complex Systems*, IEEE Transactions on Software Engineering. Vol. 24, No. 12, Pages 1131-1155, 2008.

Buhr, R., & Casselman, R.(2011).*Use Case Maps for Object-Oriented Systems*. Prentice- Hall, USA.

Burmeister, B.(2010). *Models and Methodology for Agent-Oriented Analysis and Design*; In Working Notes of the KI'96 Workshop on

Agent-Oriented Programming and Distributed Systems, Saarbrilcken, Germany.

Booch, G. (2014). *Object-oriented analysis and design*. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Booch, G. Rumbaugh, J & Jacobson.I. (1998). *The Unified Modeling Language User Guide*. Addison Wesley.

Burrafato, P. & Cossentino, M. (2012). *Designing a multi-agent solution for a bookstore with the PASSI methodology*. In P. Giorgini, Y. Lespérance, G. Wagner & E. Yu (Eds.), Proceedings of the Agent-Oriented Information Systems.

Burrafato, P., & Cossentino, M. (2012). Designing a multi-agent solution for a bookstore with the PASSI methodology. In P. Giorgini, Y. Lespérance, G. Wagner & E. Yu (Eds.), Proceedings of the *Agent-Oriented Information Systems.*

Burmeister, F., & Cossentino,M,.(2011). Designing a multi-agent solution for a bookstore with the PASSI methodology. In Fourth International Bi-Conference Workshop on *Agent- Oriented Information Systems* (AOIS-2011), Toronto (Ontario, Canada).

Bush,G., Stephen, C., & Martin P(2011). The Styx agent methodology. *The Information Science Discussion Paper Series 2012*, Department of Information Science, University of Otago, New Zealand.

Brain, J., & Odell, J.(2010.). *Object-oriented methods: Pragmatics and considerations*. Upper Saddle River, NJ: Prentice-Hall .

Caire, G., & Leal,F. (2012): Recommendations on supporting tools. *Technical Information Final version, European Institute for Research and Strategic Studies in Telecommunications (EURESCOM), July 2012.*

Caire, G., & Massonet, P. (2011). Agent-oriented analysis using MESSAGE/UML. In M. Wooldridge, G. Wei, & P. Ciancarini (Eds.), *Agent-oriented software engineering II.*

Castro, J., Kolp, M., & Mylopoulos, J.(2010). A requirements-driven development methodology. In In Proceedings of the 13th International Conference on *Advanced Information Systems Engineering (CAiSE'01),* Interlaken, Switzerland.

Castro, J., Kolp, M., & Mylopoulos, J. (2012). Towards requirements-driven information systems engineering: *The Tropos project in Information Systems*.

Cavedon, L., & Sonenberg, L. (1998). On social commitment, roles and preferred goals. In Proceedings of the Third International Conference on *Multi-Agent Systems (ICMAS),* Paris. IEEE Computer Society.

Cavedon, L., & Sonenberg, L. (2012). On social commitment, roles and preferred goals. In Proceedings of the Third International Conference on *Multi-Agent Systems (ICMAS),* Paris. IEEE Computer Society.

Cernuzzi, L., & Rossi, G.(2012). *On the evaluation of agent oriented modeling methods*. In Proceedings of Agent Oriented Methodology Workshop, Seattle.

Cossentino, M., & Potts, C. (2012). A case tool supported methodology for the design of multi- agent systems. In The 2002 International Conference on *Software Engineering Research and Practice (SERP'02),* Las Vegas (NV), USA.

Cernuzzi, L., & Rossi, G. (2002). On the evaluation of agent oriented methodologies. In Proceedings of OOPSLA 2002 Workshop on *Agent-Oriented Methodologies*. Sydney, AUS: Centre for Object Technology Applications and Research.

Cernuzzi, L., & Rossi, G. (2002). On the evaluation of agent oriented methodologies. In Proceedings of OOPSLA 2002 Workshop on *Agent-Oriented Methodologies*. Sydney, AUS: Centre for Object Technology Applications and Research.

Collinot, C., & Treur, J. (2010). Deliberate normative agents: Principles and architectures. In N. Jennings & Y. Lespérance (Eds.), *Intelligent agents VI Berlin*: Springer-Verlag.

Chan, K., & Karunasekera, S. (2004). Agent-oriented software analysis. In Proceedings of 2004 Australian *Software Engineering Conference* (pp. 20-27). Los Alamitos, CA: IEEE Computer Society Press.

Chris, S., & Barbara, A. (2011). Evaluating software engineering methods and tools-part 4: the influence of human factors. ACM SIGSOFT Software Engineering Notes.

David, L., & Michael W.(2011) . Debugging multi-agent systems using design artifacts: The case of interaction protocols. In Proceedings of the First International Joint Conference on *Autonomous Agents and Multi Agent Systems*.

David,L.(2012).*Agent-Oriented Information Systems*. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Debenham, J., & Henderson-Sellers, B.(2010). Full lifecycle methodologies for agent-oriented systems the extended OPEN process framework. In Proceedings of *Agent-Oriented In-formation Systems*, Toronto.

DeLoach,S.(2012). Multiagent systems engineering: A methodology and language for designing agent systems. In *Agent-Oriented Information Systems '99 (AOIS'99),* Seattle WA.

DeLoach, S. (2012). Analysis and design using MaSE and agentTool. In Proceedings of the 12th Midwest *Artificial Intelligence and Cognitive Science Conference* (MAICS 2001), 2012.

Dumke, R.(2011). Metrics-based evaluation of object-oriented software development methods. In Proceedings of the 2nd Euromicro Conference on *Software Maintenance and Reengineering (CSMR'98),* pages 193{196, Florence, Italy.

E. Yu (2011). *Modelling Strategic Relationships for Process Reengineering*, University of Toronto, Department of Computer Science, 2011.

Eckert, G. (2010). *Improving object-oriented analysis*. Information and Software Technology.

Elammari, M., & Lalonde, W.(2010). “*An Agent-Oriented Methodology: High-Level and Intermediate Models*”, *HLIM*, Proceedings of AOIS Heidelberg.

Frank, U. (2012). A comparison of two outstanding methodologies for object-oriented design. *Technical Report*.

Frank, U. (2012). Evaluating modeling languages: relevant issues, epistemological challenges and a preliminary research framework. *Technical Report* 15, 2012.

Genesereth, C., & Nelson, T. (2011). The importance of dealing with uncertainty in the evaluation of Software engineering methods and tools. In Proceedings of the 14th international Conference on *Software engineering and knowledge engineering,* ACM Press.

Glaser, C., & Francisco, L. (2010). Agent oriented analysis using MESSAGE/UML. In Michael W., Paolo, C., & Gerhard, W., editors, Second International Workshop on *Agent-Oriented Software Engineering* (AOSE-2012.

Hans –Van, V.(2012). A CASE tool supported methodology for the design of multi-agent systems. In H.R. Ababnia & Y. Mun (Eds.), Proceedings of the 2012 International Conference on Software Engineering Research and Practice (SERP’12), Las Vegas.

Hans-Van, V. (2000). *Software Engineering: Principles and Practice.* John Wiley & Sons, second edition.

IEEE Std 610.12. *IEEE Standard Glossary of Software Engineering Terminology*, p.77, 1990.

Iglesias,S.(2012). Evaluating modelling languages: relevant issues, epistemological challenges and a preliminary research framework. *Technical Report 15*, University of Koblenz-Landau.

James, O.(2012). Objects and agents compared. *Journal of Object Technology*. Toronto

Jayaratna,N.(2012). *Understanding and evaluating methodologies, NISAD: A systematic framework*", Maidenhead, UK: McGraw-Hill.

Jennings, N., Sycara, K., & Wooldridge, M.(2012). A Roadmap of Agent Research and Development; *In Autonomous Agents and Multi-Agent Systems Journal*, Publishers, Boston.

Jennings, N., & Wooldridge, M.(2012). *Agent-Oriented Software Engineering,* in proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99), vol. 1647,
Springer-Verlag: Heidelberg, Germany.

Juan, T., Pierce, A., & Sterling, L.(2011). *Roadmap: Extending the gaia methodology for complex open systems*", In Proceedings of the 1st ACM Joint Conference on Autonomous Agents and Multi-Agent Systems (Bologna, Italy), ACM, New York.

Juan, T., Sterling, L. and Winikoff, M.(2002).*Assembling Agent-Oriented Software Engineering Methodologies from Features*", in the Proceedings of the Third International Workshop on Agent-Oriented Software Engineering, at AAMAS'02, Bologna, Italy, 2002.

Jefrey, M. (2011). An introduction to software agents. In Jefrey M. Bradshaw, editor, *Software Agents*, pages 3{46. AAAI Press / The MIT Press, 2011.

John, F. (2010). *Multi-agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley.

Kendall E.,(1996). *Agent Roles and Role Models: New Abstractions for Multi-agent System Analysis and Design*", Proceedings of the International Workshop on Intelligent Agents in Information and Process Management, Bremen, Germany, 1998.

Kendall, E., Malkoun, M., and Jiang, C. (2010).*A Methodology for Developing Agent Based Systems*", In Distributed Artificial Intelligence Architecture and Modeling, LNAI 1087. Springer-Verlag, Pages 85-99, Germany.

Khanh, H., & Michael W.(2010) . *Comparing agent-oriented methodologies*. In To appear at the International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2010), Melbourne, Australia.

Kruchten, P.(2000).*The Rational Unified Process: An Introduction*. Addison-Wesley Pub Co.

Krupansky, J. (2010).*Foundations of Software Agent Technology*", Agtivity: Advancing the Science of Software Agent Technology.

Kinny, T., Georgeff, B., & Rao, I.(1996). *Desire: Modeling Multi-Agent Systems in a Compositional Formal Framework*, Int.Journal of Cooperative Information Systems, Vol. 6. Special Issue on FormalMethods in Cooperative Information Systems: Multi-agent Systems.

Law,D., & Naem,A.(2013). *Methods for Comparing Methods: Techniques in Software Development*. NCC Publications

Lewis, R.(2014).Project Management. McGraw-Hill professional

Lin, P., & Michael, W., (2011). Prometheus: A methodology for developing intelligent agents. In Third International Workshop on *Agent-Oriented Software Engineering*, July 2011.

Lin, P., & Michael, W., (2011). Prometheus: A pragmatic methodology for engineering intelligent agents. In Proceedings of the OOPSLA 2002 Workshop on *Agent-Oriented Methodologies*, pages 97{108, Seattle, November 2011.

Lin, P., & Michael, W., (2011). Prometheus: Engineering intelligent agents. Tutorial notes, available from the authors, October 2011.

Lin, P., & Winkoff,A., (2012). *Prometheus: A brief summary*. Technical note.

Lin, P., & Michael W.(2012). *Prometheus: A methodology for developing intelligent agents.* McGraw-Hill: New York.

Lind, C., Erik, A., Steve J., Frank H., & Pascale, T. (2011). *Empirical studies of object- oriented artifacts, methods, and processes*: State of the art and future directions. Empirical Software Engineering.

Maes, J.(2011). *Understanding and Evaluating Methodologies: NIMSAD a Systematic Framework.* McGraw-Hill, New York, 2nd edition.

Mark, W., & DeLoach.S. (2012). An overview of the multiagent systems engineering methodology. In The First International Workshop on *Agent-Oriented Software Engineering,* Limerick, Ireland.

Michael, P. (2012). Evaluation of object-oriented modeling languages: A comparison between OML and UML. In Martin Schader and Axel Korthaus, editors, *The Unified Modeling Language Technical Aspects and Applications*,. Physica-Verlag, Heidelberg.

Mike, F. (2012). The Tropos software development methodology: Processes, Models and Diagrams. In Third International Workshop on *Agent-Oriented Software Engineering*.

Moulin, A.(1994). Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*.

Nicholas, A., & Wooldridge, M.(2012). *An Introduction to Multi-Agent Systems*. John Wiley & Sons, 2002.

Nwana, H. (2012). *Software agents: An overview*. Knowledge Engineering Review. Toronto

Odell, S. (2012). Requirements of an object-oriented design method. *Software Engineering Journal*, pages 102-113, March 2012.

O'Malley, S.(2011). Determining when to use an agent-oriented software engineering methodology. In Proceedings of the Second International Workshop On *Agent- Oriented Software Engineering (AOSE-2011).*

Omicini, J., Parunak H., & Bauer B. (2012). *Representing Agent Interaction Protocols in UML*. The First International Workshop on Agent-Oriented Software Engineering.

Padgham, .J., & Winikoff, A. (2012). *A cognitive foundation for comparing object-oriented analysis methods*. In J. F Nunamaker and IEEE Computer Society Press: R. H Sprague, editors, 26th Hawaii International Conference on System Sciences.

Padgham, J., & Michael, I.(2012). *Agent Oriented Methodology;* Addison Wesley.

Parson, C., Jazayeri, M., Mandrioli, D.(2011): *Fundamentals of Software Engineering*. PrenticeHall, Englewood Cliffs, N. J. ,pp.12,14.

Pollack, B., Fausto, M., & Anna P. (2010). *Troops: An agent-oriented software development methodology*. Technical Report DIT-02-0015, University of Trento, Department of Information and Communication Technology.

Rao, A.(2010). *A methodology and modeling technique for systems of BDI agents,* In Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi- Agent World, LNAI Vol. 1038, Springer-Verlag, Berlin.

Roel, W. (2011). *A survey of structured and object-oriented software specification methods and techniques.* ACM Computing Surveys.

Rumbaugh, E.(2011). A comparison of object-oriented methodologies. *Technical report,* Object Agency Inc.

Russel,M., & Norvig.(2011). Determining when to use an agent-oriented software engineering methodology. McGraw-Hill, New York.

Sabas,M., Badri,O., & Delisle,M.(2002). *The Gaia methodology for agent-oriented analysis and design.* Autonomous Agents and Multi-Agent Systems.

Scott, A.(2011). Specifying agent behavior as concurrent tasks: Designing the behavior of social agents. In Proceedings of the Fifth Annual Conference on *Autonomous Agents,* Montreal Canada.

Scott, A. (2012). *Applying agent oriented software engineering to cooperative robotics*. University of Toronto, Department of Computer Science.

Scott, A. (2012). Multi agent Systems Engineering. *International Journal of Software Engineering and Knowledge Engineering.*

Sharble, R., & Cohen,S.(2012). The object-oriented brewery: *A comparison of two object oriented development methods*. SIGSOFT Software Engineering Notes.

Sturm, A., & Shehory,O. (2003). Towards industrially applicable modeling technique for agent- based systems (poster). In Proceedings of International Conference on *Autonomous Agents and Multi-Agent Systems,* Bologna.

Tambe, M., & Jennings, R.(2012). *Applications of intelligent agents. Agent Technology:* Foundations, Applications, and Markets

Trans,O., & Law,M.(2010). *Understanding and Evaluating Methodologies:* McGraw- Hill, New York.

Verharen, G., & Dignum, A. (2012). Agent-Object-Relationship Modeling. In Proc. of Second International Symposium - from *Agent Theory to Agent Implementation together with EMCRS 2012*, April 2012.

Wagner, G., (2011). Agent-Oriented Analysis and Design of Organizational Information Systems. In Proc. of Fourth IEEE International Baltic Workshop on *Databases and Information Systems,* Vilnius (Lithuania), May 2011.

Wood,R.(2012). *Developing Intelligent Agent Systems*. John Wiley & Sons, Ltd.

Wood, R., Pethia, L.. Gold,A., & Firth,D. (2012). *A guide to the assessment of software development methods. Technical Report* 88-TR-8, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, PA

Wooldridge, M., & Jennings, R. (2010). *A guide to the assessment of software development methods.* Technical Report 88-TR-8, Software

Engineering Institute, Carnegie-Mellon University, Pittsburgh, PA, 2010.

Wooldridge, M., & Jennings, R. (2014). *Intelligent agents: Theory and practice*. The Knowledge Engineering Review.

Wooldridge, M., Jennings, R., & Kinny, D. (2011). *A methodology for agent-oriented analysis and design.* In Proceedings of the third international conference on Autonomous Agents.

Wooldridge, M., Jennings, R., & Kinny, D (2012). *The Gaia methodology for agent-oriented analysis and design.* Autonomous Agents and Multi-Agent Systems.

.