Creating Project-Specific Agent-Oriented Methodologies Using a Feature-Based Evaluation Framework and Situational Method Engineering

Zohreh O. Akbari

***Abstract*—Current state of Agent-Oriented Software Engineering (AOSE) paradigm reports relative lack of industrial acceptance compared to others. This paper proposes a solution to this shortcoming, following the existing approaches that aim to use Situational Method Engineering (SME) in a collaborative manner among AOSE methodology designers, and suggesting the use of a methodology evaluation framework along with the process as well. This framework is a mean to collect the best method fragments and evaluate consecutively the methodology during the development process for possible methodology improvements. The proposed solution as a result, helps software development organizations to reach the fifth level of Capability Maturity Model (CMM).**

***Index Terms*—Agent-Oriented Software Engineering (AOSE), Capability Maturity Model (CMM), Evaluation Framework, Methodology, Project-Specific, Situational Method Engineering (SME).**

I. Introduction

Due to the complexity of software development process, wide range of software engineering paradigms has been devised (e.g. structured, object-oriented (OO), procedural, and declarative) [1]. During the past two decades, with the high rate increase in complexity of projects associated with software engineering, agent concepts that originated from Artificial Intelligence (AI) have been considered to devise a new paradigm for handling complex systems [1]-[6].

This paper aims to outline the current state of AOSE paradigm (section II) and propose a solution to its relative lack of industrial acceptance, which is then readjusted to present a plan for software development organizations to reach the fifth level of CMM (section III). Key building blocks of this approach include an evaluation framework for AO methodologies and a project-specific methodology building framework, described in section IV and V, followed by a case study presenting the plan, in section VI.

II. A Survey of AOSE Paradigm

In order to outline the current state of AOSE paradigm, this section starts with defining AOSE methodologies then briefly goes over its history, followed by the list of existing AOSE methodologies and their strengths and weaknesses.

A. The Definition of AOSE Methodology

Regarding [7]-[13] the definition considered for a software engineering methodology in this paper is as follows: An economical process of developing software, equipped with distinct concepts and modelling tools [14].

An agent-based system is a system in which the key abstraction used is an agent [6] (for agent definition see [1], [6], [15]). Thus by AOSE we mean a software engineering paradigm in which the key abstraction used is an agent. Considering this description and the above mentioned definition, an AOSE methodology can be defined as an economical process of developing software, equipped with distinct concepts and modelling tools, in which the key abstraction used in its concepts is that of an agent.

B. The History of AOSE Paradigm

AOSE Paradigm, which was first proposed by Yoav Shoham in 1990, is based on a societal view of computation [4], [5]. The main source of this paradigm is AI [6], [16] or precisely, Distributed AI (DAI) [17], [18]. Nevertheless, in AOSE, agents are about software engineering more than they are of AI [6]. AOSE paradigm has promoted a lot during the past two decades, and although it was first limited to academic researches, it has interested the industry within the last years as well [16], [18]. It should be pointed out that the progress of this paradigm has faced a great transformation, which some researches refer to as the entrance to the new generation of AOSE methodologies [18], [19]. The main idea of this transition is based on SME [20] and the unification strategy of existing issues [21] in order to build a framework for designing project-specific methodologies. This approach is the researchers' solution to eliminate the relative industry rejection of this paradigm, or eliminate its weaknesses [18], [21], [22]. Such issues can be found in [18], [23], and [24], which will be described later.

C. Existing AOSE Methodologies

The number of existing AOSE methodologies is very high despite of their newness. Table I, lists examples of these methodologies in order of the year of presentation. It should be pointed out that items in rows 37, 43, and 57 are more than just simple methodologies, and are project-specific methodology building frameworks (described in V).

Manuscript received December 23, 2009. This paper has been prepared as an overview of the author's Master thesis.

Zohreh O. Akbari is with the Payame Noor University, Tehran, Iran (e-mail: z.o.akbari@ gmail.com).

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I: LIST OF EXISTING AOSE METHODOLOGIES

#	Methodology	Year	Reference
1	ARCHON	1991	[25]
2	MADE	1992	[26]
3	DRM	1993	[27]
4	TOGA	1993	[28]
5	CIAD	1994	[29], [30]
6	Agent Factory	1995	[31], [32], [33], [34]
7	AOMfEM	1995	[35]
8	Cassiopeia	1995	[36], [37]
9	AAII (KGR)	1996	[38], [39]
10	AOAD	1996	[40]
11	AWIC	1996	[41]
12	CoMoMas	1996	[42]
13	MASB	1996	[43]
14	MAS-CommonKADS	1996	[44]
15	AALAADIN	1997	[45], [46]
16	AMBSA	1997	[47]
17	AOIM	1997	[48]
18	CaseLP	1997	[49]
19	DESIRE	1997	[50]
20	Adept	1998	[51]
21	AMBIA	1998	[52]
22	AOAaD	1999	[53]
23	HIM	1999	[54]
24	MaSE	1999	[55], [56]
25	MASSIVE	1999	[57], [58]
26	ZEUS	1999	[59]
27	ASEfIA	2000	[60]
28	Gaia	2000	[61], [62]
29	MESSAGE/UML	2000	[63], [64]
30	SODA	2000	[65]
31	Agent-SE	2001	[66]
32	AOSM	2001	[67]
33	Styx	2001	[68]
34	Tropos	2001	[69], [70], [71], [72], [73]
35	ADELFE	2002	[74]
36	ALCCIG	2002	[75]
37	CAOMF	2002	[24], [76], [77]
38	IEBPM	2002	[78]
39	INGENIAS	2002	[79], [80]
40	MESMA	2002	[81]
41	Nemo	2002	[82]
42	ODAC	2002	[83]
43	Agent OPEN	2002	[16], [18], [84]
44	PASSI	2002	[85], [86]
45	Prometheus	2002	[87], [88], [89]
46	ROADMAP	2002	[90]
47	SABPO	2002	[91]
48	SADDE	2002	[92]
49	MAGE	2003	[93], [94]
50	OPM/MAS	2003	[95]
51	RAP/AOR	2003	[96], [97]
52	RoMAS	2003	[98]
53	SONIA	2003	[99]
54	AMBTA	2004	[100]
55	AODM	2004	[101]
56	CAMLE	2004	[102]
57	FIPA	2004	[23], [103]
58	MAOSEM	2004	[104]
59	RAOM	2004	[105]
60	MAHIS	2005	[106]
61	MAMfHMS	2005	[107]
62	OMASM	2005	[108]
63	OWL-P	2005	[109]
64	ADMuJADE	2006	[110]
65	MOBMAS	2006	[111], [112]
66	WAiWS	2006	[113]
67	ADEM	2007	[114], [115]
68	ASPECS	2007	[116]
69	ForMAAD	2007	[117]
70	ANEMONA	2008	[118]
71	MASD	2008	[119]
72	MASIM	2008	[120]
73	PerMet	2008	[121]
74	AOMEIS	2009	[122]
75	ODAM	2009	[123]

D. Strengths and Weaknesses of AOSE Methodologies

In this section the necessity of agent-orientation usage is discussed in terms of AOSE methodologies' strengths and the relative industrial rejection in terms of its weaknesses.

1) AOSE Methodologies Strengths

- *Inclusion of other paradigms' capabilities and presentation of more abilities:* AOSE paradigm includes all the capabilities of other paradigms (e.g. OO, knowledge engineering (KE) and service-oriented (SO)) and even presents more abilities. As stated by Shoham [5], the agents can be considered as active objects with mental states [124], meaning that despite the common characteristics between objects and agents, they are not just simple objects but they present more capabilities [124]. In addition, most of the problems subject to KE methodologies are also present in designing Multi-Agent Systems (MAS), such as knowledge acquisition, modeling, and reuse. Furthermore, these methodologies conceive a knowledge-based system as a centralized one; thus, they do not address the distributed or social aspects of the agents, or their reflective and goal-oriented attitudes [124]. Moreover regarding SO methodologies it should be pointed out that service is only one of the several concepts presented by an agent, and that agents are not just service performers, but also predictors [3].
- *Suitability with new software development requirements:* As mentioned before, due to the complexity of software development process, wide range of software engineering paradigms has been devised; but recently, with the high rate of increase in complexity of software engineering projects, agent concepts which originated from AI, have been considered to devise a new paradigm for handling complex systems [1]-[6]. Some special applications of this paradigm are presented in [125].

2) AOSE Methodologies Weaknesses

- *The lack of attraction for methodology user to use the AOSE paradigm:*
  - Lack of AO programming languages [18]
  - Lack of explicit statement of AOSE advantages [18]
  - Relative difficulty of learning [12]
  - High cost of agent-orientation acquisition [18]
- *The lack of attraction for methodology user to use existing AOSE methodologies:*
  - Relative immaturity [19]
  - Marketing of multiple AOSE methodologies [18]
  - Lack of confrontation with wrong expectation of one-size-fits-all methodology [21], [126]
  - Lack of confrontation with user willingness to setup his own project-specific methodology [21]

III. PROPOSAL SOLUTION TO AOSE PARADIGM PROMOTION

The progress of AOSE paradigm is dependent to the elimination of its weaknesses (II.D.2). Clearly, when the software development organization becomes justified for using AOSE by its strengths (II.D.1), it will accept its cost and learning effort much easier, since it knows that in the long-term, this paradigm will not just pay back this cost but also benefits more than others.

With the emergence of industry willingness for agent-orientation, the next problem to be eliminated would be the lack of attraction for AOSE methodologies. It is obvious that identifying the strengths and weaknesses of each methodology can be the first step to its progress and wide industrial acceptance as well [14], [19], [21]. In addition, the marketing of multiple methodologies which is an obstacle to the ease of selection, lack of the presence of a one-size-fits-all methodology, and the need of project-specific methodologies show the necessity for exploitation of a project-specific methodology building framework. Thus

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

it is suggested that software development organizations use an evaluation framework for AOSE methodologies - such as the one suggested in IV - in order to choose the best for their project, and in case of finding no fitting match to exploit the evaluation results for building effective project-specific methodologies. This might be done by improving existing methodologies by replacing their weak parts with strong ones from other methodologies, using one of the frameworks for creating AO project-specific methodologies – described in V. Thus a consolidated approach as also expressed in [18] could give a better signal to the industry. With this regard, it is suggested that instead of competing, AOSE methodology designers collaborate with each other by evaluating their own methodologies using an appropriate evaluation framework to collect the method fragments with their rankings in order to use these information for SME. This is quite feasible since most of the AOSE methodologies are academic and not commercial products.

This approach would (i) help to improve existing methodologies by identifying their weaknesses, (ii) make the availability of multiple methodologies an advantage (having wide range of method fragment options), (iii) do away with the wrong expectation on one-size-fits-all methodology, (iv) answer user willingness to setup his own project-specific methodology. Thus this approach will attract methodology users to use AOSE methodologies, and results to industrial acceptance of the paradigm. More so, the usage of the frameworks for creating AO project-specific methodologies will not only make it possible to use programming languages from other paradigms which are suitable for AOSE, but the industry willingness for this paradigm will encourage language designers as well.

This solution to AOSE weaknesses may also be readjusted to propose a plan for development organizations to reach the fifth level of CMM. Fig. 1 explains this plan. In CMM organizational maturity framework [127], [128], 5 maturity levels are distinguished [20]: Initial, Repeatable, Defined, Managed, and Optimizing. Since the proposed plan exploits the SME in order to build project-specific methodologies, it is clear that it satisfies the third level of CMM. In addition, since the evaluation framework assesses the methodologies for management plans and thus the management plans' method fragments are constructed to methodology, both process and products are regularly evaluated by the project management team to satiate the forth level of CMM. The feedback that is given by the organization while employing the methodology using the evaluation framework causes the methodology correction to take place continuously and concurrent with its exploitation, and satisfies the fifth level of CMM.

What has taken placed by now is the growth of repository by adding all the AOSE methodology's components without considering any evaluation (e.g. [22], [129], [130]). But the approach presented here is the usage of an evaluation framework and a project-specific methodology building framework simultaneously together. So, each methodology would first be evaluated, and the method fragments with their grades enter the repository. This makes the selection of method fragments with desired grades possible at the methodology building stage which better implements SME.

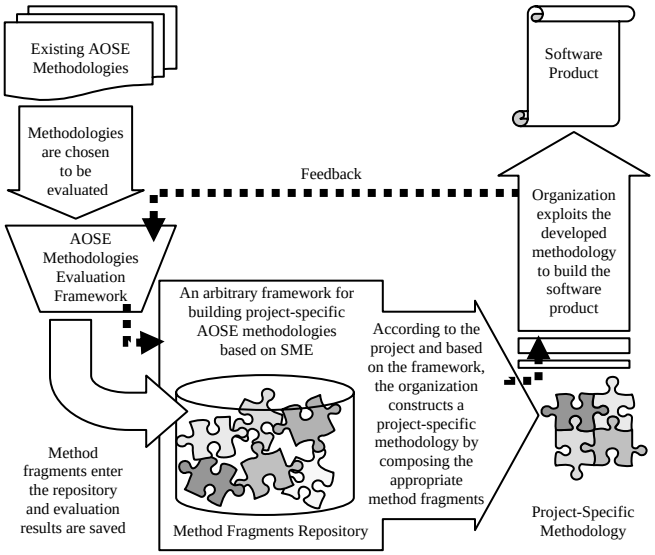

Fig. 1 Proposal Plan for Agent-Oriented Software Development Organizations to Reach the Fifth Level of CMM

IV. Existing Approaches for Evaluating AOSE Methodologies

Researches considering the evaluation of AOSE methodologies are limited to [12]-[14], [19], [112], [126], [131]-[141], and some other studies that compare two or three methodologies, only with respect to the expressiveness and the concepts supported by the methodology [12]. Most of the mentioned evaluation frameworks suffer from one or both of the following shortcomings: (i) Lack of coverage for all of the methodology aspects, (ii) Lack of definition of a precise evaluation metric.

Regarding the methodology definition mentioned in II.A, methodologies can be considered in six major aspects: concepts, notation, process, pragmatics, support for software engineering, and marketability. In addition, evaluation metric should be able to present different levels of methodology support for each criterion. The framework presented in [14] and completed in [134] evaluates methodologies from all aspects mentioned and defines a metric with 7 levels of support; thus it perfectly overcomes the mentioned weaknesses of most evaluation frameworks.

V. Existing Approaches for Creating Agent-Oriented Project-Specific Methodologies

Existing approaches for creating AO project-specific methodologies are based on SME also known as Methodology Engineering [142] – which is directed towards a controlled, formal, and computer-assisted construction of situational (project-specific) methods out of method fragments (a description of an Information System (IS) engineering method, or any coherent part thereof) [20]. Despite their strengths, these approaches also have some weaknesses: (i) Lack of methodology evaluation and result saving while storing a methodology in method fragments repository, (ii) Lack of consideration of method fragment capability while creating a project-specific methodology.

To eliminate these shortcomings, the best approach is to

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

evaluate and store all the method fragments with their corresponding evaluation results, and postpone the selection of method fragments with desired grades to methodology building stage. This approach again shows the necessity of joining these frameworks with an evaluation framework. The project-specific methodology building frameworks are briefly introduced in section V.A to V.C.

A. Agent OPEN Method

OPEN, which stands for OO Process, Environment and Notation, was first outlined in [143] and published in [144] as a full life cycle methodology [9]. OPEN Process Framework (OPF) consists of (i) a process metamodel, (ii) a repository, and (iii) a set of construction guidelines. The major elements in OPF metamodel are [130]: Work Units (Activities, Tasks, and Techniques), Work Products, Producers, and two auxiliary ones (Stages and Languages).

To extend this approach to support AOSE, [145] analyzes the differences between AO and OO approaches in order to be able to itemize and outline the necessary additions to the OPF's repository in the standard format provided in [146]. A list of method fragments added to OPF from existing AOSE methodologies can be found in [147].

B. Feature-Based Method

In [24] is proposed a modular approach enabling developers to build customized project-specific methodologies from AOSE features. An AOSE feature is defined in [76] to encapsulate software engineering techniques, models, supporting Computer-Aided Software Engineering (CASE) tools and development knowledge such as design patterns. It is considered a stand-alone unit to perform part of a development phase, such as analysis or prototyping, while achieving a quality attribute such as privacy. Comparing to Agent OPEN method, an AOSE feature can be defined in terms of these notions as a Work Unit performed by one or more Producers in support of a specific software engineering Stage resulting in one or more Work Products represented in the respective Languages [77]. Differing from Agent OPEN approach, this method does not regard it necessary to rely on the formal metamodel of method fragments and has demonstrated in [24], [76], [148] and [149] that informal approach to methodology composition works equally well and is more likely to be adopted in industry.

This method identifies and standardizes the common elements of the methodologies which could form a generic agent model on which specialized features might be based. The remaining parts of the methodologies would represent added-value that they bring to the common elements, and should be componentized into modular features. The small granularity of features allows them to be combined into the common models in a flexible manner. By conforming to the generic agent model in the common elements, it is expected that the optional features semantics remain consistent [24].

C. FIPA Methodology Technical Committee Method

This work refers to the FIPA Methodology Technical Committee activity and it consists in a quite open approach that allows the composition of elements coming from a repository of fragments of existing design processes that could be expressed in terms of a standard notation. Specifically dealing with the methods integration problem in this contribution, two different approaches have been considered to obtain methods integration: (i) guided by a (MAS) meta-model; (ii) guided by a development process.

In the first approach, the designer has to preliminary identify the elements that compose the meta-model of the MAS; then choose the method fragments that are able to produce those elements. The second approach focuses on the instantiation of some software development process that completely cover the development of MAS, by selecting suitable method fragments of AO methodologies proposed in the literature or ad-hoc defined [21], [23], [103].

VI. Case Study

In this section the proposed plan in section III is used in a case study. The Agent OPEN method is used as the project-specific methodology building framework, and Gaia and INGENIAS, which are already evaluated in [134], are considered as the input methodologies. Evaluation results prove that these methodologies are both strong, but since there is no one-size-fits-all methodology, they can not fit all projects as well. Suppose that the goal is to design a methodology for an organization that repeatedly works in the same domain, and accepts large, complex, business-critical projects; thus, according to [9] the organization needs a methodology with strong management plans and consideration for development context. INGENIAS achieved grades equal and higher than 4 in the evaluation (performed by Prof. Juan Pavon, one of the designers of INGENIAS) with the metric having highest grade of 6 for the best implementation of features [134]. The only features wherein INGENIAS achieved grade 4 are:

- *Concepts Criteria*: Service and Norm (Rule)
- *Process Criteria*: Project Management Plan, Configuration Management Plan, Verification and Validation Plan, Quality Assurance Plan, and Development Context

As it is seen they cover the features needed by the organization of the case study. According to the section III, it is suggested to add appropriate method fragments from Agent OPEN to INGENIAS. Below is the list of activities needed to be added to INGENIAS from Agent OPEN [9]:

- *Management Plans method fragments*: Project Management, Configuration Management, Risk Management, Metrics Engineering, Quality Engineering, Evaluation, Test, Training
- *Product Line Supporting method fragments*: Reuse Engineering, Programme Engineering, Resource Planning
- *Proposal Plan's Feedback method fragments*: Process Engineering

For Service and Rule features, since Gaia has achieved grade 6 in these features [134], it is suggested to use the related tasks from Gaia [22] in INGENIAS:

- *Specifying services of each agent*
- *Defining organizational rules*

Table II to IV present the Task/Stage possibility matrices of additional tasks regarding [9]. Due to the limited space

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

only 3 matrices are presented here[1].

It should be pointed out that assigning a possibility level to a task in a certain stage depends on: (i) the nature of the task, (ii) possibility (availability of necessary resources), and (iii) conformity with the life cycle standard. Life cycle standard of INGENIAS is USDP [150], which has 4 phases, and thus has got one phase more than INGENIAS (Transition). Since some of the tasks added to INGENIAS in this case study, need to be executed in this stage, this phase is added to the new methodology.

TABLE II: ADDITIONAL ANALYSIS TASKS/STAGES MATRIX

Additional Analysis Tasks \ Stages	Inception	Elaboration	Construction	Transition
Defining organizational rules	F	R	M	F

TABLE III: ADDITIONAL DESIGN TASKS/STAGES MATRIX

Additional Design Tasks \ Stages	Inception	Elaboration	Construction	Transition
Specifying services of each agent	F	R	M	F

TABLE IV: PROJECT MANAGEMENT TASKS/STAGES MATRIX

Project Management Tasks \ Stages	Inception	Elaboration	Construction	Transition
Undertake feasibility study	M	F	F	F
Undertake project planning	F	M	F	F
Execute plan	F	M	M	M
Establish change management strategy	F	M	M	M
Establish policy on component acquisition	F	M	M	M
Establish policy on COTS	F	M	M	M
Establish policy on outsourcing	F	M	M	M
Manage contract(s) / Negotiate contract(s)	M	M	F	F
Manage contract(s) / Select subcontractors and partners	D	M	F	F
Manage contract(s) / Oversee subcontractors and partners	F	M	M	M
Manage contract(s) / Sign off deliverables	M	M	M	M
Manage costs / Create a project budget	F	M	F	F
Manage costs / Apply cost containment	F	M	M	M
Manage costs / Report on project costs	M	M	M	M
Manage the customer relationship	M	M	M	M
Manage human resources / Create an appropriate organizational structure	R	M	F	F
Manage human resources / Interviewing prospective staff members	R	M	F	F
Manage human resources / Hiring prospective staff members	R	M	F	F
Manage human resources / Assigning staff members to Roles and Teams	R	M	F	F
Manage human resources / Evaluating staff member performance	R	M	M	M
Manage human resources/ Specifying individual goals	R	M	F	F
Identify project roles and responsibilities	R	M	F	F
Choose project team	R	M	F	F
Allocate tasks	R	M	F	F
Manage resources	R	M	M	M
Ensure schedule management / Estimate time	R	M	F	F
Ensure schedule management / Develop schedule	R	M	F	F
Ensure schedule management / Communicate schedule	R	M	F	F
Ensure schedule management / Control schedule	R	M	M	M
Obtain business approval	M	M	M	M

VII. CONCLUSION

The study of AOSE paradigm strengths shows the necessity of its usage; yet its current state of reports relative lack of industrial acceptance compared to others. This paper proposes a solution to this problem which aims to eliminate the weaknesses of this paradigm by the usage of an evaluation framework and a project-specific methodology building framework, simultaneously in a software development organization. The usage of SME, considerations for project management plans, and continuous improvements in the methodology through a wise combination of these frameworks may also lead the organization to reach the fifth level of CMM.

In this regard, following future works are suggested:

- *Activities towards implementation and exploitation of the proposal plan:* enriching the method fragment repository, storing the methodologies' evaluation results.
- *Activities towards completion proposal plan details:* enforcing the identification of the method fragments related to each criterion while storing a methodology, defining a change management plan for continuous changes that occur in proposal plan structure and data (towards improving the evaluation framework, and/or the methodology in use).
- *Activities towards adding more capabilities to the proposal plan:* preparing possibilities to design Domain-Specific Languages (DSL), preparing possibilities to determine the proper paradigm for the project and change dominant paradigm of the proposal plan.

[1] The rest may be found in the author's forthcoming Master thesis.

REFERENCES

[1] Jennings N. R. and Wooldridge M., Agent-oriented software engineering, In Handbook of Agent Technology (ed. Bradshaw J.), AAAI/MIT Press, 2000.

[2] Genesereth M. R. and Ketchpel S. P., Software agents, Communications of the ACM, Vol. 37, NO. 7, 1994, pp. 48-53.

[3] Jennings N. and Wooldridge M., Software agents, IEEE Review, 1996, pp. 17-20.

[4] Shoham Y., Agent-oriented programming, Technical Report STAN-CS-1335-90, Computer Science Department, Stanford University, Stanford, CA 94305, 1990.

[5] Shoham Y., Agent-oriented programming, Artificial Intelligence, 60(1), 1993, pp. 51-92.

[6] Wooldridge M., Agent-based software engineering, IEE Proc. on Software Engineering, 144(1), 1997, pp. 26–37.

[7] Brinkkemper, S., Method engineering: engineering of information systems development methods and tools, Information & Software Technology, 38, 1996, pp. 275-280.

[8] CMS, Selecting a development approach, Centers for Medicare and Medicaid Services (CMS), Original Issuance: February 2005, Revalidated: March 2008.

[9] Firesmith D.G. and Henderson-Sellers B., 2002, The OPEN process framework: An introduction, Addison-Wesley, UK, ISBN 0-201-67510-2, 2002.

[10] Lyytinen, K. A taxonomic perspective of information systems development: theoretical constructs and recommendations, In Critical Issues in Information Systems Research, Jr., R. J. B. and Hirschheim, R. A.(eds.) John Wiley & Sons Ltd., 1987, pp. 3-41.

[11] Standards Coordinating Committee of the Computer Society of the IEEE, IEEE standard glossary of software engineering terminology, IEEE Standards Board, IEEE Std 610.12-1990.

[12] Sturm A. and Shehory O., A framework for evaluating agent-oriented methodologies, In Giorgini P. and Winikoff M. (Eds.), Proceedings of the Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems, Melbourne, Australia, 2003, pp. 60-67.

[13] Sudeikat J., Braubach L., Pokahr A., and Lamersdorf W., Evaluation of agent-oriented software methodologies: examination of the gap between modeling and platform, Proc. of the Workshop on Agent-Oriented Software Engineering (AOSE), New York, USA, 2004.

[14] Akbari Z.O. and Faraahi A., Eveluation framework for agent-oriented methodologies, World Congress on Science, Engineering and Technology (WCSET 2008), Paris, France, Vol. 35, ISSN 2070-3740, 2008, pp. 419-424.

[15] Wooldridge M. and Jennings N. R., Intelligent agents: theory and practice, The Knowledge Engineering Review, 10(2), 1995, pp. 115–152.

[16] Debenham J.K. and Henderson-Sellers B., Full lifecycle methodologies for agent-oriented systems – the extended OPEN process framework, Proceedings of Agent-Oriented Information

Systems, Giorini P., Lespreance Y., Wagner G. and Yu E. (Eds.), 2002, pp. 87-101.

[17] Bond A. H. and Gasser L. (editors), A Survey of Distributed Artificial Intelligence, Readings in Distributed Artificial Intelligence, Morgan Kaufmann Publishers: San Mateo, CA, 1988.

[18] Henderson-Sellers B. and Gorton I., Agent-based software development methodologies, White Paper on OOPSLA 2002 Workshop on Agent-Oriented Methodologies, COTAR, Sydney, 2003.

[19] Dam K. H. and Winikoff M., Comparing agent-oriented methodologies, Proceedings of the 5th Int. Bi-Conference Workshop on Agent-Oriented Information Systems, Melbourne, Australia, 2003.

[20] Harmsen A.F., Situational Method Engineering, Doctoral dissertation University of Twente, With ref., index and summary in Dutch, ISBN: 90-75498-10-1, 1997.

[21] AOSE Technical Forum Group, AL3-TF1 Report, The first AgentLink III Technical Forum (AL3-TF1), Rome, Italy, 2004.

[22] Henderson-Sellers B., Debenham J., and Tran Q.N.N., Adding agent-oriented concepts derived from GAIA to Agent OPEN, In Advanced Information Systems Engineering: 16th International Conference, CAiSE 2004, Riga, Latvia, Berlin: Springer-Verlag, 2004, pp. 98-111.

[23] Cossentino M. and Seidita V., Composition of a new process to meet agile needs using method engineering, Software Engineering for Large Multi-Agent Systems, Vol. III, LNCS Series, Elsivier Ed, 2004.

[24] Juan T., Sterling L., and Winikoff M., Assembling agent oriented software engineering methodologies from features, Giunchiglia F., Odel J., and Weiss G. (Eds.), Agent-Oriented Software Engineering III, Third International Workshop, AOSE 2002, Bologna, Italy, LNCS 2585, Berlin, Germany: Springer-Verlag, 2002, pp. 198–209.

[25] Cockburn D. and Jennings N.R., ARCHON: A distributed artificial intelligence system for industrial applications, In O'Hare G.M.P. and Jennings N.R. (editors), Foundations of Distributed Artificial Intelligence, John Wiley & Sons, 1996, pp. 319–344.

[26] O'Hare G.M.P and Wooldridge M.J., A software engineering perspective on multi-agent system design: Experience in the development of MADE, In Avouris N.M. and Gasser L. (Eds.), Distributed Artificial Intelligence: Theory and Praxis, Kluwer Academic Publishers: Boston, MA, 1992, pp. 109–127.

[27] Singh M.P., Huhns M.N., and Stephens L.M., Declarative representations of multiagent systems, IEEE Transactions on Knowledge and Data Engineering, 5(5), 1993, pp. 721–739.

[28] Gadomski A.M., TOGA: A Methodological and Conceptual Pattern for modeling of Abstract Intelligent Agent, Proceedings of the First International Round-Table on Abstract Intelligent Agent, Gadomski A.M. (editor), 1993, pp. 25-27.

[29] Verharen E. and Weigard H., Agent-oriented information systems design, In Ras Z. and Zemankova M. editors, Poster Proceedings of the International Symposium on Methodologies for Intelligent Systems (ISMIS'94), Amsterdam, 1994, pp. 378- 392.

[30] Verharen E.M., A language-action perspective on the design of cooperative information agents, PhD thesis, Katholieke Universieit Brabant, the Netherlands, 1997.

[31] Collier R., The realisation of Agent Factory: An environment for the rapid prototyping of intelligent agents, M. Phil Thesis, UMIST, UK, 1996.

[32] Collier, R.W., O'Hare G.M.P., Agent Factory: a revised agent prototyping environment, in 10th Irish Conference on Artificial Intelligence and Cognitive Science (AICS), 1999.

[33] Collier R.W., Agent Factory: a framework for the engineering of agent-oriented applications, PHD Thesis, National University of Ireland, 2002.

[34] O'Hare G.M.P., Collier R.W., Conlon J., and Abbas S., Agent Factory: an environment for constructing and visualising agent communities, 9th Irish Conference on Artificial Intelligence and Cognitive Science (AICS), 1998.

[35] Kendall E.A., Malkoun M.T., and Jiang C., A methodology for developing agent based systems, In Lukose D. and Zhang C. (Eds.), Distributed artificial intelligence architecture and modelling: Proceedings of the first Australian workshop on DAI, LNCS 1087, Springer-Verlag: Heidelberg, Germany, 1996, pp. 85-99.

[36] Collinot A. and Drogoul A., Using the Cassiopeia method to design a soccer robot team, Applied Artificial Intelligence (AAI) Journal, 12 (2-3), 1998, pp. 127-147.

[37] Collinot A., Drogoul A., and Benhamou P., Agent-oriented design of a soccer robot team, In Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS'96), Menlo Park, CA: American Association for Artificial Intelligence, 1996, pp. 41-57.

[38] Kinny D. and Georgeff M., 1996, Modelling and design of multi-agent systems, Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ATAL-96), LNAI 1193, Berlin: Springer-Verlag.

[39] Kinny D., Georgeff M. and Rao A., A methodology and modelling technique for systems of BDI agents, Proc. of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, Springer, 1996, pp. 56-71.

[40] Burmeister B., Models and methodology for agent-oriented analysis and design, In Fischer K. editor, Working Notes of the KI'96 Workshop on Agent Oriented Programming and Distributed Artificial Intelligence, Dresden, 1996.

[41] Muller H.J., Towards agent systems engineering, International Journal on Data and Knowledge Engineering, Special Issue on Distributed Expertise, 23, 1996, pp. 217–245.

[42] Glaser N., The CoMoMAS methodology and environment for multiagent system development, In Zhang C. and Lukose D. (Eds.), Multi-agent systems methodologies and applications, 2nd Australian Workshop on DAI, LNAI 1286, Springer-Verlag, 1996, pp. 1-16.

[43] Moulin B. and Brassard M., A scenario-based design method and an environment for the development of multiagent systems, Lukose D. and Zhang C. (Eds.), Proc. of the First Australian workshop on DAI Intelligence, LNAI 1087, Springer-Verlag, 1996, pp. 216-231.

[44] Iglesias, C. A., Garijo, M., Gonzalez, J. C., and Velasco, J. R., Analysis and design of multiagent systems using MAS-CommonKADS. Singh M.P., Rao A., and Wooldridge M.J. (Eds.), Intelligent Agents IV (LNAI 1365), Springer-Verlag: Berlin Germany, 1998, pp. 313-326.

[45] Ferber J. and Gutknecht O., Aalaadin: a meta-model for the analysis and design of organizations in multi-agent systems, rapport de recherche, Lirmm, univ. de Montpellier, 1997.

[46] Ferber J. and Gutknecht O., A meta-model for the analysis and design of organizations in multi-agent systems, Proc. of the 3rd International Conference on Multi Agent Systems (ICMAS98), Paris, France, 1998.

[47] Neal Reilly W.S., A methodology for building believable social agents, Proceedings of the First International Conference on Autonomous Agents (Agents '97), Marina del Rey, CA, USA, ACM Press, New York, ISBN 0-89791-877-0, 1997, pp. 114-121.

[48] Kindler C., DeLuke R., Rhea J., and Kunz J.C., Development and demonstration of an agent-oriented integration methodology, Kaman Sciences Corporation, Rome , NY, Contract Number F30602-94-C-0216, 1997.

[49] Martelli M, Mascardi V. and Zini F., CaseLP: a complex application specification environment based on logic programming, Proc. of ICLP'97 Post Conference Workshop on Logic Programming and Multi-Agents, 1997, pp. 35-50.

[50] Brazier F.M.T., Dunin-Keplicz B.M., Jennings N.R., and Treur J., DESIRE: modelling multi-agent systems in a compositional formal framework, Int Journal of Cooperative Information Systems, 6(1), 1997, pp. 67-94.

[51] Jennings N.R., Norman T.J., and Faratin P., ADEPT: An agent-based approach to business process management, SIGMOD Record 27(4), pp. 32-39, 1998.

[52] Gao X. and Sterling L., A methodology for building information agents, In Yang Y., Li M. and Ellis A. (Eds.), Web technologies and applications, ch. 5, International Academic Publishers, 1998, pp. 43-52.

[53] Wooldridge M., Jennings N. R. 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, 1999, pp. 69–76.

[54] Elammari M. and Lalonde W., An agent-oriented methodology: high-level and intermediate models, Proceedings of AOIS 1999 (Agent-Oriented Information Systems), In the Third International Conference on Autonomous Agents, Seattle, USA, 1999.

[55] DeLoach S.A., Multiagent systems engineering: a methodology and language for designing agent systems, Proceedings of the First International Bi-conference Workshop on Agent-Oriented Information Systems (AOIS '99), In the Third International Conference on Autonomous Agents, Seattle, USA, 1999.

[56] DeLoach S.A., Multi-agent systems engineering: an overview and case study, In Henderson-Sellers B. and Giorgini P. (Eds.), Agent-oriented methodologies, ch. 11, Hershey, PA: Idea Group, 2005.

[57] Lind J., Iterative software engineering for multiagent systems, The MASSIVE Method, LNAI 1994, Berlin: Springer-Verlag, 2001.

[58] Lind J., MASSIVE: Software Engineering for Multiagent Systems,

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

PhD Thesis, University of Saarland, Saarbrucken, 1999.
[59] Nwana H., Ndumu D., Lee L., and Collis J., ZEUS: A Tool-Kit for Building Distributed Multi-Agent Systems, In Applied Artifical Intelligence Journal, Vol 13 (1), 1999, pp. 129-186.
[60] Zamboneli F., Jennings N.R., Omicini A. and Wooldridge M., Agent-oriented software engineering for internet applications, Omicini A., Zambonelli F., Klusch M. and Tolksdorf R. (Eds.), Coordination of Internet Agents: Models, Technologies and Applications, ch. 13, Springer, 2000.
[61] Wooldridge M., Jennings N. R. and Kinny D., The Gaia methodology for agent-oriented analysis and design, Journal of Autonomous Agents and Multi Agent Systems, Vol. 3, No. 3, 2000, pp. 285-312.
[62] Zambonelli F., Jennings N.R. and Wooldridge M., Multi-Agent Systems as Computational Organizations: The Gaia Methodology, In Henderson-Sellers B. and Giorgini P. (Eds.), Agent-oriented methodologies, ch. 6, Hershey, PA: Idea Group, 2005.
[63] Caire G., Leal F., Chainho P., Evans R., Garijo F., Gomez J., Pavon J., Kearney P., Stark J. and Massonet P., Agent oriented analysis using MESSAGE/UML, Wooldridge M., Weiss G., and Ciancarini P. (Eds.), Agent-oriented software engineering II, LNCS 2222, Berlin: Springer-Verlag, 2000, pp. 119-135.
[64] Evans R., Kearney P., Stark J., Caire G., Garijo F. , Gomez Sanz J., Pavon J., Leal F., Chainho P., and Massonet P., MESSAGE: Methodology for engineering systems of software agents, EURESCOM Technical Information, 2001.
[65] Omicini A., SODA: societies and infrastructures in the analysis and design of agent-based systems, Agent-Oriented Software Engineering, LNCS 1957, Berlin: Springer-Verlag, 2000, pp. 185-193.
[66] Far B.H., Agent-SE: A methodology for agent oriented software engineering, Jin Q., Li J., Zhang N., Cheng J., Yu C. and Noguchi S. (Eds.), Enabling Society with Information Technology, Springer, pp. 357-366, 2001.
[67] Shi Z., Jiao W., and Sheng Q., Agent-oriented software methodology, CEEMAS 2001, Cracow, Poland, 2001
[68] Bush G., Cranefield S., and Purvis M., The Styx agent methodology, The Information Science Discussion Paper Series, Number 2001/02, ISSN 1172-6024, Department of Information Science, University of Otago, Dunedin, New Zealand, 2001.
[69] Bresciani P., Giorgini P., Giunchiglia F., Mylopolous J., and Perini A., Tropos: An agent-oriented software development methodology, Autonomous Agents and Multi-Agent Systems, 8(3), 2004, pp. 203-236.
[70] Bresciani P., Perini A., Giorgini P., Giunchiglia F. and Mylopoulos J., A Knowledge Level Software Engineering Methodology for Agent-Oriented Programming, Proc. of the Fifth International Conference on Autonomous Agents, Montreal, Canada, 2001, pp. 648-655.
[71] Castro J., Kolp M. and Mylopoulos J., A Requirements-Driven Development Methodology, Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01), 2001, pp. 108-123.
[72] Castro J., Kolp M., and Mylopoulos J., Towards requirements-driven information systems engineering: the Tropos project, Information Systems, 27(6), 2002, pp. 365-389.
[73] Mylopoulos J., Kolp M., and Castro J., UML for agent-oriented software development: the Tropos proposal, Proc. of the 4th International Conference on the Unified Modeling Language, UML'01, Toronto, Canada, Springer, 2001, pp. 422-442.
[74] Bernon C., Gleizes M.P., Picard G., and Glize P., 2002, The ADELFE methodology for an intranet system design, In Giorgini P., Lespérance Y., Wagner G., and Yu E. (Eds.), AOIS-2002, pp. 1-15.
[75] Zhang T., Kendall E.A. and Jiang H., An agent-oriented software engineering methodology with application of information gathering systems for LCC, Proceedings of the Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002 at CAiSE*02), 2002, pp 1-15.
[76] Juan T., Sterling L., Martelli M., and Mascardi V., Customizing AOSE methodologies by reusing AOSE features, In Rosenschein J. S., Sandholm T., Wooldridge M., and Yokoo M. (Eds.), Proc. of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03), Melbourne, Australia, 2003, pp.113–120.
[77] Taveter K., Sterling L., 2008, Features as loosely defined method fragments, AOSE TFG08.
[78] Taveter K. and Wagner G., A multi-perspective methodology for modelling inter-enterprise business processes, In Arisawa H. and Kambayashi Y. (Eds.): ER 2001 Workshops, LNCS 2465, Springer-Verlag Berlin Heidelberg, 2002, pp. 403-416.
[79] Pavon J. and Gomez-Sanz J., Agent oriented software engineering with INGENIAS, Proc. of the Third International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003), Marik V., Muller J., Pechoucek M. (Eds.), Multi-Agent Systems and Applications II, LNAI 2691, Spring-Verlag, 2003, pp. 394-403.
[80] Pavon J., Gomez-Sanz J.J., and Fuentes R., The INGENIAS methodology and tools, In Henderson-Sellers B. and Giorgini P. (Eds.), Agent-oriented methodologies, ch. 9, Hershey, PA: Idea Group, 2005.
[81] Cuesta P., Gomez A., Gonzalez J.C., and Rodriguez F.J., The MESMA approach for AOSE, 4th Iberoamerican Workshop on Multi-Agent Systems (Iberagents'2002), a workshop of IBERAMIA'2002, The VIII Iberoamerican Conference on Artificial Intelligence, 2002.
[82] Huget M., Nemo: an agent-oriented software engineering methodology, In Proceedings of OOPSLA Workshop on Agent-Oriented Methodologies, Debenham J., Henderson-Sellers B., Jennings N.R. and Odell J., Seattle, USA, 2002.
[83] Gervais M.P., ODAC: an agent-oriented methodology based on ODP, Journal of Autonomous Agents and Multi-Agent Systems, Vol. 7, 2002, pp. 199-228.
[84] Henderson-Sellers B., Tran Q., Debenham J., and Gonzalez-Perez C., Agent-oriented information systems development using OPEN and the agent factory, Information Systems Development Advances in Theory, Practice and Education: 13th International Conference on Information Systems Development, ISD 2004, Vilnius, Lithuania, New York: Kluwer Acadmic / Plenum Publishers, 2005, pp. 149-160.
[85] Cossentino M. and Potts C., A CASE tool supported methodology for the design of multi-agent systems. In Ababnia H.R. and Mun Y. (Eds.), Proceedings of the International Conference on Software Engineering Research and Practice (SERP'02), Las Vegas, 2002, pp. 315-321.
[86] Cossentino M., From requirements to code with the PASSI methodology, In Henderson-Sellers B. and Giorgini P. (Eds.), Agent-oriented methodologies, ch. 4, Hershey, PA: Idea Group, 2005.
[87] Cervenka R., Modeling Notation Source: Prometheus, Version: 03-04-02, Foundation for Intelligent Physical Agents, 2003.
[88] Padgham L. and Winikoff M., Prometheus: A methodology for developing intelligent agents, In Giunchiglia F., Odell J., and Weiß G. (Eds.), Proc. of the Third International Workshop on Agent-Oriented Software Engineering (AAMAS'02), LNCS 2585, 2002, pp. 174-185.
[89] Padgham L. and Winikoff M., Prometheus: A pragmatic methodology for engineering intelligent agents, In Debenham J., Henderson-Sellers B., Jennings N., and Odell J.J. (Eds.), Agent-oriented Software Engineering III, Proceedings of the Workshop on Agent-oriented Methodologies at OOPSLA 2002, Seattle, 2002, pp. 97-108.
[90] Juan T., Pearce A. and Sterling L., ROADMAP: Extending the Gaia methodology for Complex Open Systems, Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy, 2002.
[91] Dikenelli O., Erdur R.C., SABPO: A standards based and pattern oriented multi-agent development methodology, ESAW 2002, 2002, pp. 213-226.
[92] Sierra C., Sabater J., Agusti J., Garcia P., Evolutionary Programming in SADDE, AAMAS'02, ACM, Bologna, Italy, 2002, pp. 1270-1271.
[93] Shi Z., Zhang H., Dong M., and Zhao Z., MAGE: Multi-Agent Environment, Proc. of the International Conference on Computer Networks and Mobile Computing (ICCNMC'03), 2003, pp. 181-188.
[94] Shi Z., Zhang H., Cheng Y., Jiang Y., Sheng Q., Zhao Z., MAGE: An agent-oriented programming environment, IEEE ICCI, 2004, 250-257.
[95] Sturm A., Dori D., and Shehory O., Single-model method for specifying multi-agent systems, Proceedings of Second International Joint Conference on Autonomous Agents and Multi Agent Systems, 2003, pp. 121-128.
[96] Taveter K. and Wagner G., Towards radical agent-oriented software engineering processes based on AOR modelling, In Henderson-Sellers B. and Giorgini P. (Eds.), Agent-oriented methodologies, ch. 10, Hershey, PA: Idea Group, 2005.
[97] Wagner G., The agent-object-relationship meta-model: Towards a unified view of state and behavior, Information Systems, 28(5), 2003, pp. 475-504.
[98] Yan Q., Shan L., Mao X. and Qi Z., RoMAS: a role-based modeling

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

method for multi-agent systems, Proceedings of International Conference on Active Media Technology, 2003, pp. 156-161.
[99] Alonso F., Frutos S., Martinez L., and Montes C., SONIA: a methodology for natural agent development, In Gleizes M.P., Omicini A., and Zambonelli F. (Eds.): ESAW 2004, LNCS 3451, Springer-Verlag Berlin Heidelberg, 2005, pp. 245-260.
[100] Sardinha J., Milidiu R., Lucena C. and Paranhos P., A methodology for building trading agents in electronic markets, Technical Report, Computer Science Department, PUC-Rio, Brazil, PUC-RioInf.MCC36/04, 2004.
[101] Tian J., Foley R., and Tianfield H., A new agent-oriented development methodology, Proceedings of the Intelligent Agent Technology, IEEE/WIC/ACM International Conference, 2004, pp. 373 – 376.
[102] Shan L. and Zhu H., Software engineering for multi-agent systems III: Research issues and practical applications, In Choren R., Garcia A., Lucena C., and Romanovsky A. (Eds.), Proceedings of the Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems, Berlin: Springer-Verlag, 2004, pp. 144-161.
[103] Garro A., Fortino G. and Russo W., Using method engineering for the construction of agent-oriented methodologies, In Proc. of WOA 04, Torino, Italy, 2004, pp. 51-54.
[104] Wang L. and Guo Q., Mobile Agent Oriented Software Engineering (MAOSE), In Karmouch A., Korba L., and Madeira E. (Eds.): MATA 2004, LNCS 3284, Springer-Verlag Berlin Heidelberg, 2004, pp. 168-177.
[105] Giret A. and Botti V., Towards a recursive agent oriented methodology for large-scale MAS, In Giorgini P., Muller J.P., Odell J. (Eds.), AOSE 2003, LNCS 2935, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 25–35,.
[106] Li C. and Liu L., MAHIS: an agent-oriented methodology for constructing dynamic platform-based HIS, Australian Conference on Artificial Intelligence, 2005, pp. 705-714.
[107] Giret A., Botti V., and Valero S., MAS methodology for HMS, In Marik V., Brennan R.W. and Pechoucek M. (Eds.): HoloMAS 2005, LNCS 3593, Springer-Verlag Berlin Heidelberg, 2005, pp. 39-49.
[108] Villaplana E.A., A proposal for an organizational MAS methodology, AAMAS'05, 2005, pp. 1370.
[109] Desai N., Mallya A.U., Chopra A.K. and Singh M.P., OWL-P: a methodology for business process development, Kolp M., Bresciani P., Henderson-Sellers B., Winikoff M.(Eds.), Agent-Oriented Information Systems III, 7th International Bi-Conference Workshop (AOIS 2005), 2005, pp. 79-94.
[110] Nikraz M., Caire G. and Bahri P.A., A methodology for the development of multi-agent systems using the JADE platform, Computer Systems Science and Engineering, Vol. 21, No. 2, 2006, pp. 99–116.
[111] Tran Q.N.N., Beydoun G., and Low G., Design of a peer-to-peer information sharing MAS using MOBMAS (ontology-centric agent-oriented methodology, In Advances in Information Systems Development, Springer, 2007, pp. 63-76.
[112] Tran Q.N. and Low G., MOBMAS: a methodology for ontology-based multi-agent systems development, Information and Software Technology 50, 2008, pp. 697–722.
[113] Lu H. and Chhabra M., A methodology for agent oriented web service engineering, In Shi Z. and Sadananda R. (Eds.): PRIMA 2006, LNCS 4088, Springer-Verlag Berlin Heidelberg, 2006, pp. 650-655.
[114] Cervenka R. and Trencansky I., The agent modeling language – AML, A comprehensive approach to modeling multi-agent systems, Whitestein Series in Software Agent Technologies and Autonomic Computing, ISBN: 978-3-7643-8395-4, 2007.
[115] Whitestein Technologies, LS/TS Product Brochure, Available at http://www.whitestein.com/library/WhitesteinTechnologies_LS-TS_ProductBrochure.pdf, 2008.
[116] Cossentino M., Gaud N., Hilaire V., Galland S. and Koukam A., ASPECS: an agent-oriented software process for engineering complex systems, In Proc. of the Fifth Agent Oriented Software Engineering Technical Forum (AOSE-TF5), Hammameth, Tunisia, 2007.
[117] Hadj-Kacem A., Regayeg A. and Jmaiel M., ForMAAD: a formal method for agent-oriented application design, Web Intelligence and Agent Systems, 5(4), IOS Press, Amesterdam, Netherland, ISSN: 1570-1263, 2007, pp. 435-454.
[118] Giret A., ANEMONA: a multi-agent methodology for holonic manufacturing systems, Springer Series in Advanced Manufacturing, First edition, ISBN-13: 978-1848003095, 2008.
[119] Abdelaziz T., Elammari M., and Branki C., MASD: towards a comprehensive multi-agent system development methodology, Meersman R., Tari Z., and Herrero P. (Eds.), OTM 2008 Workshops, LNCS 5333, 2008, pp. 108 – 117.
[120] Clancey W.J., Sierhuis M, Seah C., Buckley C., Reynolds F., Hall T., and Scott M., Multi-agent simulation to implementation: a practical engineering methodology for designing space flight operations, In Artikis A. et al. (Eds.): ESAW 2007, LNCS 4995, Springer-Verlag Berlin Heidelberg, 2008, pp. 108-123.
[121] Grislin-Le Strugeon E., Anli A., and Adam E., A methodology to bring MAS to information systems, In Kolp M. et al. (Eds.): AOIS 2006, LNCS 4898, Springer-Verlag Berlin Heidelberg, 2008, pp. 90-104.
[122] Athanasiadis I.N. and Mitkas P.A., A methodology for developing Environmental Information Systems with Software Agents, Whitestein Series in Software Agent Technologies and Autonomic Computing, Birkhauser Verlag Basel/Switzerland, 2009, pp. 119-137.
[123] Mao X., Zhao J., and Wang J., Engineering adaptive multi-agent systems with ODAM methodology, In Ghose A., Governatori G., and Sadananda R. (Eds.): PRIMA 2007, LNCS 5044, Springer-Verlag Berlin Heidelberg, 2009, pp. 380-385.
[124] Iglesias C.A., Garijo M. and Gonzalez J.C., A survey of agent-oriented methodologies, Intelligents Agents IV: Agent Theories, Architectures, and Languages, LNAI 1555, Springer-Verlag, 1999, pp. 317-330.
[125] Wooldridge M. and Ciancarini P., Agent-oriented software engineering: the state of the art, In Ciancarini P. and Wooldridge M. (Eds.), Agent-Oriented Software Engineering, Springer-Verlag LNAI 1957, 2001.
[126] Henderson-Sellers B. and Giorgini P., Agent-oriented methodologies, Idea Group Publishing, ISBN 1-59140-587-4, 2005.
[127] Humphery W.S., Managing the software process, Addision-Wesley, Reading, MA, 1990.
[128] Paulk M.C., Curtis B., Chrissis M.B., and Weber C.V., Capability maturity model for software, Ver. 1.1, Technical Report CMU/SE1-93-TR-24, Software Engineering Institute, Pittsburgh, PA, 1993.
[129] Henderson-Sellers B., Debenham J., Tran Q. N., Cossentino M. and Low G., Identification of reusable method fragments from the PASSI agent-oriented methodology, In Agent Oriented Information Systems III, LNCS 3529, Springer-Verlag GmbH, 2006, pp. 95-110.
[130] Henderson-Sellers B., Giorgini P. and Bresciani P., Enhancing Agent OPEN with concepts used in the Tropos methodology, In Proceedings of the Fourth International Workshop Engineering Societies in the Agents World, Imperial College London, UK, 2003.
[131] Abdelaziz T., Elammari M., and Unland R., A framework for the evaluation of agent-oriented methodologies, Fourth International Conference on Innovations in Information Technology (IIT '07), 2007, pp. 491-495.
[132] Al-Hashel E., Balachandran B.M., and Sharma D., A comparison of three agent-oriented software development methodologies: ROADMAP, Prometheus, and MaSE, Apolloni B. et al. (Eds.), KES 2007/WIRN 2007, Part III, LNAI 4694, 2007, pp. 909-916.
[133] Alinaghi T., Ghoroghi C., Saborui A., and Basseda R., A framework for dependency evaluation of the agent oriented methodologies work flows, Third IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE 2009), 2009, pp. 289-290.
[134] Akbari Z.O. and Faraahi A., A feature-based framework for agent-oriented methodologies evaluation, In Proceedings of CCSR 2009, Tehran, Iran, 2009, pp. 125-133.
[135] Cernuzzi L. and Rossi G., On the evaluation of agent oriented methodologies, In Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies, November 2002.
[136] Kumar M., Contrast and comparison of five major agent oriented software engineering methodologies, Available at http:// students. jmc.ksu.edu/grad/madhukar/www/professional/aosepaper.pdf, 2002.
[137] Lin C., Kavi K.M., Sheldon F.T., Daley K.M. and Abercrombie R.K., A methodology to evaluate agent oriented software engineering techniques, Software Agents and Semantic Web Technologies Minitrack, IEEE Proc. HICSS-40, Big Island HI, 2007.
[138] Sabas, A, Badri, M and Delisle, S. A multidimensional framework for the evaluation of multiagent system methodologies, 6th World Multi Conference on Systemics, Cybernetics and Informatics (SCI-2002), 2002, pp. 211-216.
[139] Saremi A., Esmaeili M., and Rahnama M., Evaluation complexity problem in agent based software development methodology, Second International Conference on Industrial and Information Systems (ICIIS 2007), 2007, pp. 577-584.
[140] O. Shehory, A. Sturm, Evaluation of modeling techniques for agent-

based systems, Agents 2001, 2001, pp. 624-631.
[141] Yu E. and Cysneiros L.M., Agent-oriented methodologies – towards a challenge exemplar, Fourth International Workshop on Agent-Oriented Information Systems (AOIS'02), May 2002.
[142] Kumar K. and Welke R.J., Method engineering, a proposal for situation-specific methodology construction, Systems analysis and design: a research agenda, Cotterman and Senn (Eds.), Wiley, 1992, pp. 257-268.
[143] Henderson-Sellers B. and Graham I.M., OPEN: Toward Method Convergence, IEEE Computer 29, 4, 1996, 86–89.
[144] Graham I., Henderson-Sellers B. and Younessi H., The OPEN process Specification, Addison-Wesley, 1997.
[145] Debenham J. and Henderson-Sellers B., Designing agent-based process systems - Extending the OPEN process framework, In Plekhanova V. (Ed.), Intelligent agent software engineering, ch. VIII, Hershey, PA: Idea Group Publishing, 2003, pp. 160-190.
[146] Henderson-Sellers B., Simons A.J.H. and Younessi H., The OPEN toolbox of techniques, Addison-Wesley, UK, 1998, 426 pp.
[147] Henderson-Sellers B., Evaluating the feasibility of method engineering for the creating of agent-oriented methodologies, Pechoucek M., Petta P., and Varga L.Z. (Eds.), CEEMAS 2005, 2005, pp. 142–152.
[148] Juan T., and Sterling L., The ROADMAP meta-model for intelligent adaptive multiagent systems in open environments, In Giorgini P., Muller J., and Odell J. (Eds.), Agent-Oriented Software Engineering IV, 4th International Workshop, AOSE 2003, Melbourne, Australia, LNCS 2935, Berlin, Germany: Springer-Verlag, 2003, pp. 53–68.
[149] Sterling L. and Taveter K., The art of agentoriented modeling, Cambridge, MA, London, England: The MIT Press, 2009.
[150] Ambler S.W., Nalbone J., and Vizdoz M., Enterprise Unified Process: extending the Rational Unified Process, Prentice Hall PTR, ISBN: 0-13-191451-0, 2005.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)