Agent-based framework for simulation and support of Dynamic Engineering Design Processes in PSI

Vladimir Gorodetsky[2], Vadim Ermolayev[3], Wolf-Ekkehard Matzke[1], Eyck Jentzsch[1], Oleg Karsayev[2], Natalya Keberle[3], Vladimir Samoylov[2]

[1]Cadence Design Systems, GmbH, Mozart str., 2, 85622, Feldkirchen, Germany
{wolf,jentzsch}@cadence.com
[2]SPIIRAS, 39, 14-th Liniya, St. Petersburg, 199178, Russia
{gor, ok, samovl}@mail.iias.spb.su_
[3]Zaporozhye National Univ., 66, Zhukovskogo st., 69063, Zaporozhye, Ukraine
{eva, kenga}@zsu.zp.ua_

Abstract. The paper reports on the first results of the Productivity Simulation Initiative (PSI) project of Cadence Design Systems GmbH. The project addresses the problem of fine-grained modeling and simulation of dynamic engineering design processes in order to attempt to assess and to enhance their productivity. The application domain of PSI is Semiconductor and Electronic Systems design. PSI uses multi-agent approach and models design processes as collaborative orchestrated activities of designers' teams. Rational collaboration and team formation is arranged through enhanced Contract Net negotiations. The paper outlines the modeling approach, reports on the methodology, and the rapid prototyping tool used for PSI Simulation Prototype implementation.

1 Introduction

"Design – a signature of human intelligence – was always a great challenge for artificial intelligence (AI) research" (cf. [21]). Observations of how humans act in design inspired several fundamental ideas in AI, e.g., automated problem solving and reasoning [SIM69]. In return, AI community has attacked the problems of design domain by attempting to engineer systems and infrastructures that are capable of supporting humans in accomplishing intelligent tasks.

Engineering design processes are far from being fully automated yet in a satisfactory way, though some attempts have been undertaken. These attempts have used agents to create intelligent software systems to support design processes performed by designer teams and comprising contributions from various disciplines ([1], [3], [4], [5], [18], [19]). These attempts revealed the fact that automating a design process is the task, which due to its complexity is similar to that of AI challenges like natural language processing, human-like decision making, etc. In both cases available theories, frameworks, methodologies, and technologies are still too immature to approach a solution (the state of the art is discussed in [8]). However, some of the vital problems in design process analysis, optimization and management may be solved at least partially automatically.

One of these problems is the modeling, the assessment and the prediction of the productivity of the teams performing design in order to be capable to optimize and manage a time-cost trade-off “on the fly” preserving the high quality of the expected final design product.

The task of building a software system able to reliably assess, predict and optimize the productivity in a Dynamic Engineering Design Process (DEDP) is at least threefold. The first aspect is that the system needs the adequate representation model of the world – i.e., the environment comprising the processes and the collaborative teams of autonomous actors who play these processes. The second aspect is that the system needs the adequate model of a DEDP. And the third aspect is that the system needs well defined and reliable productivity metrics and their assessment and prediction mechanisms. It is also important to notice that the mentioned representation models, metrics and mechanisms should be well grained and balanced to constitute adequate, feasible, and reliable framework.

The goal of the 1-st phase of the PSI project is to develop and to validate an agent-based simulation framework designated for future use in assessment and prediction of the design process productivity thus forming the prerequisites for the design process optimization and on-line management. Particular subtasks of this phase are: to develop the formal framework for modeling the world and the processes, to implement the demonstrator of this framework in the form of multi-agent simulator prototype (further on referred to as DEDP-MAS), and to use it for experimenting with several application scenarios for assessing the feasibility and the further development of the approach.

The paper is structured accordingly. Section 2 presents our problem statement and the rationale for the focus of our activities. Section 3 sketches the approach, modeling and implementation methodology used. Section 4 describes the two of our application scenarios and the experimental settings. Section 5 reports on the simulation experiments of different types. The reminder of the paper discusses the related work provides conclusions and our plans for future work.

2 Problems Addressed and Benefits Gained

PSI project bases itself on the understanding that engineering design processes in the vast majority of cases and industrial branches are weakly defined and heavily influenced by unpredictable dynamic changes. Therefore, these processes should be performed in quite a flexible manner to ensure meeting the objectives and to demonstrate at least near-optimal productivity and quality of results. Gaining such flexibility seriously depends on the capability to assess the feasibility of the initial plan, to react to the changes in the process in timely and flexible manner through re-planning, to evaluate the plan in terms of the predicted productivity and the result quality. Hence, one of the tasks of PSI was to implement a software tool capable to play DEDP simulation games for both:

- Evaluating the initial plan for an engineering design project using predictive simulation, and

– Partially automating the process of dynamic planning of collaborative activities of a designer team through the simulation of their negotiations

Another goal of the reported PSI phase was to create the initial DEDP Simulation Testbed by recording the logs of the DEDPs of the application scenarios (Section 4). This testbed contains the logs of the DEDPs application scenarios and the mental models of the agents playing the roles of design team members in experiments (Section 5).

The initial set of experiments performed on the created prototype clearly showed the approach feasibility to model engineering design processes. E.g., experimenting with the PSI application scenarios showed that DEDP-MAS simulator may be efficiently used for the planning and the adjustment of the project design plans according to the unexpected changes in designers' capabilities. DEDP-MAS framework prototype is now used in PSI project as the basis for further development of a more accurate model of design processes. We finally aim to obtain a software tool which will help in enhancing the productivity of DEDPs.

3 Approach, Methodology, and Agent Platform

DEDP participants are conceptually structured and form a kind of organization comprising individual human **Actors**[1] and groups of **Actors** at different hierarchical levels. **Activities** of an organization and its members are regulated by **Policies**. **Actors** form dynamic **Teams** on **Project** and/or **Task** basis. The organization and its members own certain knowledge. Substantial part of this knowledge are the **Capabilities** of the **Actors** to perform certain activities. Sub-sets of these **Capabilities** together with respective **Authority** specifications form human **Roles** in a design process. The environment of the above organization is formed by the structured specification of DEDPs under performance which actually imposes partial ordering and other relations on activities being composed in the design process.

DEDP-MAS prototype has been designed and implemented in Multi Agent System Development Kit (MASDK) [14]. According to the Gaia methodology [22] which forms the methodological basis of the MASDK platform the system’s organization is described at analysis stage in terms of *Roles*, *Protocols,* and *Agent classes* allocated single or more Roles they have to perform. Fig.1 illustrates the organization of the current version of DEDP-MAS prototype as it looks in the window of the *Meta-model editor* provided by MASDK. It comprises three problem-oriented (*Task_manager*, *Executor* and *Tool_Provider*) and one auxiliary (*Time_Simulator*) roles. A human actor can play *Task_manager* and *Executor* roles, so they are assigned to one agent class – *Designer*. The main activities executed by the *Task_manager* role are 1) revealing design artifacts, 2) specifying activities to be executed and the order of their execution as Pert chart, 3) assigning activities and 4) monitoring their execution. The main activities carried out by the *Executor* role are 1) scheduling assigned activities and 2) simulating their execution. Software tools are

[1] The mentioned entities of DEDP Ontology are **bold**.

considered here as resources used for some activities execution and the task of their usage scheduling is executed by the *Tool_Provider* role. The tasks executed by the *Time_simulator* role which is considered as auxiliary one are 1) providing human actor with the interface for inputs description, and 2) synchronizing agents' operation.

Interaction model includes eight protocols. Initiator of each protocol is indicated by triangle. The *Initialization* protocol specifies interaction of the *Simulator* and *Designer* agent classes at the DEDP start up. In particular, when this protocol has been initialized the human actor assisted by the *Simulator* agent selects the task to execute, inputs the initial data, determines the settings of simulation parameters, etc. Then, according to the *Initialization protocol*, the above data is communicated to the agents of the *Designer* class. The simulation itself is then being performed in day-by-day mode under the control of the *Simulator* agent class. Three protocols, *Interruption, Negotiation* and *Simulation* may be performed in each work day. They are initialized by the *Simulator* agent one after another.

The *Interruption* protocol is started up if the human actor assisted by the *Designer* agent class intends to view and analyze the workload and schedule of each designer via respective user interfaces and to modify the workload of designers for the remaining part of DEDP simulated. If the workload of a designer is modified the re-scheduling of the remaining activities is automatically executed.

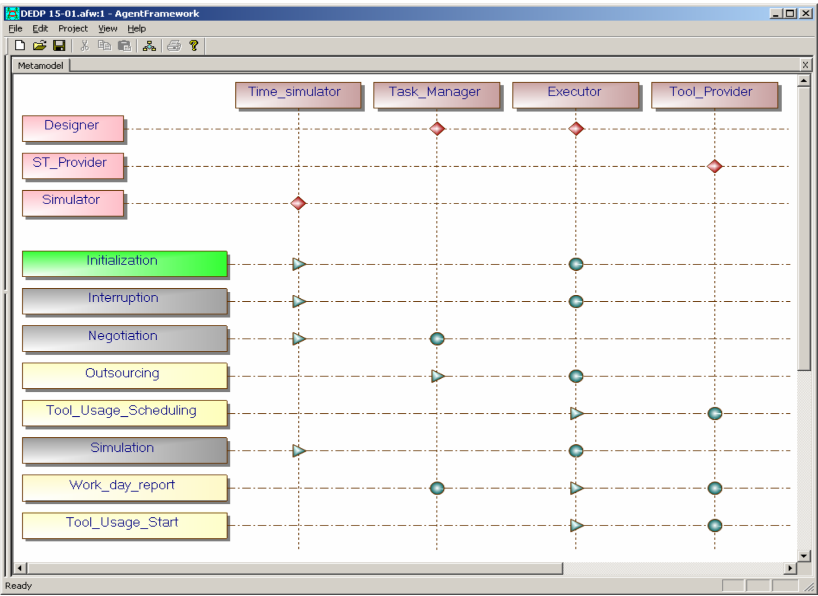
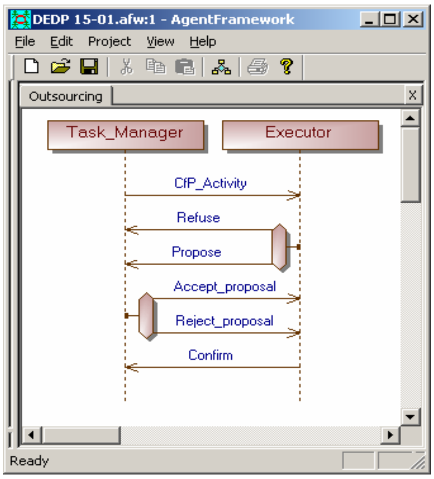


Fig. 1. Meta model of DEDP MAS prototype

During the *Negotiation* protocol execution, the *Task manager* role initiates the *Outsourcing* (nested) protocol (Fig. 2) based on the Contract Net Protocol (CNP) [13] to perform the assignment of the activities to the designers., If an agent performing the *Executor* role during the CNP–based negotiation intends to use a software tool it initiates negotiation with a *ST_provider* agent using the *Tool_usage_scheduling* (nested) protocol.



After completion of the assignment and scheduling procedures (when all the abovementioned protocols are finished) the *Simulator* class agent initiates the *Simulation* protocol while simulating operation of the agents of the *Designer* class performing the *Executor* roles according to the schedule for the current day. If certain agents of the *Designer* class use software tools in simulation progress they initiate the *Tool_usage_start* (nested) protocol. After the work day activity simulation is completed the *Simulation* agent class using *Work_day_report* (nested) protocol reports the simulation results to the agents performing *Task_manager* and *Tool_provider* roles. Conceptual description of the above protocols is made using the *Protocol editor* of the MASDK platform in the standard style like depicted in Fig. 2 for *Outsourcing*.

At the design stage [22] a formal specification of (i) agent classes and (ii) their services (in terms of state machines) is developed. Specification of each agent class is reduced to identifying its services associated with respective protocols in which agents of the class take part, and activities executed according their assigned roles. E.g. in the current version of DEDP-MAS the specification of *Designer* agent class comprises eighteen services, such as *Assignment management, Outsourcing, Activity scheduling, Proposal computing, Activity simulation,* etc. Detailed description of graphic editors used in MASDK for specifying agent classes and services (in terms of state machines) can be found in [14].

4 Application Scenarios and Experimental Settings

PSI simulator is used in two application modes: descriptive and predictive. In descriptive mode the simulation is used to assess the performance of the DEDPs which have been accomplished in the past.

The predictive mode supports project managers in planning of starting and re-planning of running design projects in case of emergent problems e.g. late changes to the design objective, sudden unavailability of the team members, the changes in the workload of the designers according to the influence of the other projects, etc.

Based on these usage modes PSI testbed comprises the following two parts. The first part (the initial testbed) contains the detailed records of 1 – 2 ongoing design projects to extract the knowledge of the acting humans and log every event during the execution of the project as accurately as possible. DEDP model and the adequacy of the implemented interaction mechanisms are evaluated based on these "logs of DEDP execution". As the result the corrective factors are extracted to improve the quality of the simulation. The performance of the process can be accessed based on this improved simulation of a design process.

The second part of the testbed is under creation and covers the prediction capability. A set of 3 to 5 design artifacts will be used to create detailed project plans by experienced project managers (2-3 per artifact). As the part of this planning process all decisions and their reasons will be recorded to extract the know-how of the project manager.

Initial set of experiments has been performed on the two simplified scenarios: the process of the design of a digital multimedia encoder [15] and the process of the design of an analog controlled amplifier [23]. These processes have been described according to the data collected by lead designers of Cadence Design Systems GmbH in their previous design projects. Execution logs have been created for the respective DEDPs through filling in the DEDP questionnaires [16]. These logs formed the initial testbed for DEDP-MAS prototype.

The scenarios were simplified to keep the complexity at a low to medium level. For example, the digital scenario is characterized by:

– 5 designers
– Design artifact comprising 4 functional blocks
– The process from RTL design up to tape-out in GDSII format resulted in 36 activities.

5 Experiments for Framework Prototype Checking

Experiments with the DEDP-MAS prototype are performed in frame of the real world project of low complexity and amount of work under some simplifications. It was assumed that the execution log does not exist at the beginning of the process, but is gradually developed in line with the DEDP flow. The objective of the experiments was to develop the so called prediction-correction methodology. The methodology should predict the development of the process up to the next checkpoint through the simulation based on the initial task and existing agents' mental models. General view of the experiments scenario is outlined in Fig. 3.

DEDP-MAS prototype configuration (the set of software agents with the accordingly prepared mental models) corresponds to the project design team. It is assumed in the experiments that only one agent of the *Designer* class performing the *Task manager* role in DEDP-MAS prototype assists the human *Project leader* and the rest of the agents of this class simulate the activities of the other design team members.

In the experiment along with the development of the DEDP the human expert repeatedly carries out the following activities:

– Gradually develops the execution log via specific event-based log editor, and
– Analyses the predictions of the further development of the process via interaction with the *Task manager* assistant agent and ordering the simulation in the predictive mode

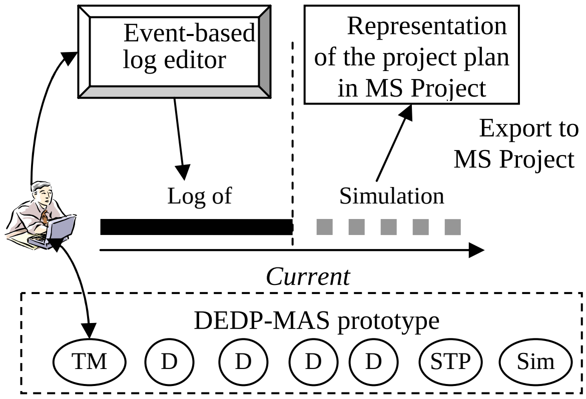


Fig. 2. General scenario of experiments supported by DEDP-MAS

Log editor allows to record all kinds of events that may occur in simulation: activity assignments, activity accomplishments etc. For example activity assignment record includes: 1) assignment time, 2) list of designers possessing required capabilities and believed to be potential executors of the activity under assignment, 3) the log of negotiation with these designers, 4) the information about the winner of the accomplished CNP. The data associated with the events and which may be useful for computing different design process metrics is also recorded to the log.

Simulation aiming at prediction of the future DEDP development is initiated after the playback of the log corresponding to the accomplished part of the process at the given point in time. During simulation the human *Project leader* may trim the available capacities of the designers and revise the assignments of the activities arranged by the assistant *Task_manager* agent. Accordingly the *Project leader* can evaluate the predicted path of DEDP in *What-if* mode. Simulation results are exported in Microsoft Project format. This facility allows the *Project leader* to compare the Gantt charts of DEDP paths executed in real life to that predicted by simulation. In DEDP-MAS prototype these checkpoints are allowed ones per simulated project day Export to the Microsoft Project and the above comparison are done on daily basis. Therefore the *Project leader* can monitor and dynamically influence DEDP development in the mentioned checkpoints.

Thus, the experiments with the developed and implemented DEDP-MAS prototype support the solution of the following practically important problems:

1) Estimation of the adequacy of the world model representation in the DEDP-MAS prototype.
2) Estimation of adequacy of DEDP model used in the DEDP-MAS prototype. The adequacy is assessed through the comparison of actual processes with the ones simulated by the DEDP-MAS prototype in the predictive mode.

6 Discussion and Related Work

The constellation of projects pioneered R&D in agent-based engineering design support and automation began to appear about a decade ago e.g. ([3], [4], [1]). These findings initially motivated PSI project. Some projects of the "second wave" ([18], [5]) helped to specify the focus of PSI in automating the near-optimal arrangement of DEDPs in terms of their productivity. Besides that PSI aims to provide the industrial strength solution in the mentioned niche.

DEDP modeling framework and simulator prototype implementation is based on research in: dynamic planning process modeling, methodologies and frameworks for the design and implementation of multi-agent systems.

DEDP modeling framework in the part of organizational and actor-related knowledge representation bases itself on the frameworks ([12], [20], [6], [7]). PSI contribution in this part is the incorporation of roles, actors with its specific subclasses, teams of actors, negotiation context in one coherent ontology and its binding to the engineering design domain by, e.g., introducing the sub-ontologies of Design Artifacts and Software Tools [9]. The main emphasis of PSI DEDP ontology is the model of a dynamic team of designers which are formed through contracting negotiations and perform dynamically orchestrated processes. In the part of process modeling PSI borrows the ideas and the approach from ([2], [6], [11]). In DEDP ontology engineering design processes are modeled as tasks composed of subtasks and atomic activities. Similarly to [17] subtasks and activities may have weak and strong dependencies. However the knowledge on these dependencies is in PSI local and differs from actor to actor as specified in their partial local plans. Similarly to [11] activities have pre-conditions, post-conditions and post-effects. However PSI ontology constrains the semantics of pre-/post- conditions and effects by making them sub-classes of an event concept. Material inputs and outputs semantically and structurally belong to PSI Design Artifacts ontology. Some inspirations for the development of agent reasoning mechanisms in PSI were provided by RAPPID set based reasoning framework [18] and RACING negotiation framework [6]. PSI extends these initial percepts to the family of negotiation mechanisms: task allocation, design re-use, choice of a software tool to perform a design activity [10].

7 Conclusions and Future Work

As reported in the paper the first outcomes of PSI are (i) the formal framework for modeling DEDPs, (ii) implemented simulation prototype of DEDP simulator software tool, (iii) initial PSI testbed comprising DEDP execution logs built for two mentioned application scenarios, (iv) results of the initial experiments which prove the feasibility of concept implementation.

The contribution of the modeling framework is the provision of the DEDP model in the form of the set of DEDP-full ontologies. These ontologies were simplified to DEDP-light version and used in the design of the meta-model of DEDP-MAS, in the implementation of PSI DEDP execution log questionnaire and editor as well as in the implementation of local knowledge models of DEDP-MAS agents. Another input of the modeling framework is the set of coordination and negotiation mechanisms. It provides CNP-based negotiation mechanisms for task or activity allocation, design artifact re-use and the choice of the proper software tool. Planned future work in this direction will develop the extensions for welfare-based DEDP productivity assessment, process and result quality assessment and refine negotiation strategies and dynamic re-planning mechanisms.

The experiments with the DEDP-MAS prototype support solution of several practically important DEDP tasks e.g. the estimation of the adequacy of the world model and DEDP model itself represented in the DEDP-MAS prototype. It also provides a computational framework for development and evaluation of the reliable metrics concerning the design process productivity and to discover *sensitive parameters* of the design process influencing the mentioned metrics that is important for future design processes optimization. An important property of the DEDP-MAS prototype from the industrial viewpoint is the integration with Microsoft Project providing dynamic visualization of DEDP progress and the results of the *Project leader* intervention thru it.

The development of the initial testbed allowed to adjust the requirements to the prototype as wel as to prepare initial evaluation experiments. The initial set of experiments performed on the created prototype clearly showed the feasibility of this approach to model engineering design processes. For example experiments with PSI application scenarios showed that DEDP-MAS simulator may be efficiently used for the planning and the adjustment of the design project plans according to the unexpected changes in design team members' capabilities. Planned experimental work will be to build the extension of the testbed by recording the execution log of currently running design project at Cadence Design Systems, GmbH and to develop the methodology for the evaluation of the initial design project plan through the usage of predictive simulation mode.

References

1. Balasubramanian, S. and Norrie, D. H.: A multi-agent intelligent design system integrating manufacturing and shop-floor control. In: Proc. First Int. Conf. on Multi-Agent Syst., San Francisco, pp. 3-9, 1995
2. Buhler, P. and Vidal, J.M.: Enacting BPEL4WS specified workflows with multiagent systems. In Proc. of the Workshop on Web Services and Agent-Based Engineering, *2004*
3. Cutkosky, M.R. et al: PACT: An Experiment in Integrating Concurrent Engineering Systems. *IEEE Computer* 26(1), p. 28-38, 1993
4. Darr, T. P., Birmingham, W. P.: An Attribute-Space Representation and Algorithm for Concurrent Engineering. CSE-TR-221-94, University of Michigan, Department of Electrical Engineering and Computer Science, Ann Arbor, Michigan 48109-2122, 1994.
5. Danesh, M. R. and Jin, Y.: An Agent-Based Decision Network for Concurrent Engineering Design. CERA 9(1), 37-47, 2001.
6. Ermolayev, V., et al: Towards a framework for agent-enabled semantic web service composition. *Int. J. of Web Services Research*, 1(3), 63-87, 2004.

7. Ermolayev, V. Keberle, N. and Tolok, V.: OIL Ontologies for Collaborative Task Performance in Coalitions of Self-Interested Actors. In: H. Arisawa, Y. et al (Eds.): Conceptual Modeling for New Information Systems Technologies ER 2001 Workshops, Yokohama Japan, November 27-30, 2001. LNCS vol. 2465, 390-402, 2001.
8. Ermolayev, V.: The State of the Art in Agent-Based Modeling and Simulation of Design Processes. TR-PSI-2-2004. Cadence Design Systems, GmbH, 25 p., 2004.
9. Ermolayev, V. and Keberle, N.: DEDP-MAS Ontologies Specification v.1.0. TR-PSI-05-2004, VCAD EMEA Cadence Design Systems GmbH. Oct. 2004.
10. Ermolayev, V. et al: Agent-Based Dynamic Engineering Design Process Modeling Framework. Technical Report. Cadence Design Systems, GmbH, 29 p., 2004.
11. Fensel, D. and Bussler, C.: The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications* 1(2): 113-137, 2002.
12. Fox, M.C. and Gruninger, M.: Enterprise Modelling. AI Magazine 19(3), 109–121, 1998.
13. Foundation for Intelligent Physical Agents. FIPA Contract Net Interaction Protocol Specification. Ref. No XC00029E. 2001.
14. Gorodetski, V. Karsaev, O., Samoilov, V., Konushy, V., Mankov, E., and Malyshev, A.: Multi Agent System Development Kit: MAS software tool implementing GAIA Methodology. In: Z. Shi and Q. He (eds.) Int. Conf. on Intelligent Information Processing (IIP2004), Beijing, Springer, pp.69-78. 2004.
15. Jentzsch, E. and Matzke, W.-E.. Case Study of a Digital Design Process. VCAD EMEA Cadence Design Systems GmbH. May 17, 2004.
16. Keberle, N. and Weber, S.: Questionnaire to create formal record of an Analog Design Process & A Walk-through Example. Cadence Design Systems GmbH, VCAD CIC2, Aug 20, 2004.
17. Nagendra Prasad, M. V., and Lesser, V. R.: Learning situation-specific coordination in cooperative multi-agent systems. *Autonomous Agents and Multi-Agent Systems*. 2(2), 173-207 1999.
18. Parunak, H.V.D. et al: The RAPPID Project: Symbiosis between Industrial Requirements and MAS Research. *Autonomous Agents and Multi-Agent Systems* 2: 111-140, 1999.
19. Shen W. and Barthes J.-P.: An Experimental Multi-Agent Environment for Engineering Design, *Int. J. of Cooperative Information Systems*, 5(2-3), 131-151, 1996
20. Uschold, M. et al: The Enterprise Ontology. Knowledge Engineering Review, 13(1), 1998.
21. Vancza, J.: Artificial Intelligence Support in Design: A Survey. Keynote paper at the *1999 International CIRP Design Seminar*, Kluwer, 1999.
22. 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*, 3(3), 285-312, 2000.
23. Weber, S.: Case Study of an Analog Design Process. VCAD CIC2 Cadence Design Systems GmbH., 2004.