Multi-agent Software Tool for Management of Design Process in Microelectronics

Vladimir Gorodetski
Oleg Karsaev
Victor Konushy
St.-Petersburg Institute for Informatics and Automation, St.-Petersburg, Russia
{gor, ok, kvg}@iias.spb.su

Wolf-Ekkehard Matzke
Eyck Jentzsch
Cadence Design Systems, GmbH, Feldkirchen, Germany
{wolf,jentzsch}@cadence.com

Vadim Ermolayev
Zaporozhye National Universitet, Zaporozhye, Ukraine
eva@zsu.zp.ua

Abstract

The paper presents a software tool to support the management of design process in microelectronics. It is developed as a multi-agent prototype intended for automated project planning, scheduling and simulation as well as project tracking and control while managing cooperative work of a design team.

1. Introduction

The task of Design Process Management (DPM) is the coordination of a design project team in resource constrained settings in order to gain near optimal performance. Although the field of computer aided support of such coordination has been intensively researched in the past, at the moment, the research community both in academia and industry exhibits growing interest in further exploration of this very important and highly vibrant domain based on the new technologies and for new applications. One of such new application domains in project management is the design in microelectronics. So far design process coordination and optimization in this application domain has not been fully supported by efficient software tools.

DPM in microelectronics possesses a number of specific features that demand the development of specialized tools. The former research in this field is reported, e.g., in [1], [5], [10]. More detailed coverage of the state of the art is presented in [3]. Domain knowledge representation using interrelated ontologies is given in [4]. Some basic peculiarities of a design process in microelectronics are described in [5].

This paper presents a software tool to support the management of Dynamic Engineering Design Processes (DEDP). The theoretical basis of this tool is presented in our earlier publication [5]. The current version has more sophisticated functionalities and user interfaces which provide the support for electronic device design project management practically along its full life cycle. The software tool is developed as a multi-agent prototype intended for automated project planning, scheduling, simulation and project tracking and control while managing cooperative work of a design team.

The rest of the paper is organized as follows. Section 2 presents the general problem statement outlining the basic design tasks supported by the software tool and its multi-agent architecture. Sections 3 describes its basic functionalities. Section 4 outlines the scenarios of a microelectronics design process. Section 5 concludes and outlines the plans for the future research and development.

2. Problem Statement. Software Tool Multi-agent Architecture

A design process in microelectronics generally is composed of the following four basic subtasks. *The first subtask* is, given an electronic device specification, to design the Work Breakdown Structure (WBS) decomposing the electronic device into the subblocks and presenting the design project as a structured set of atomic activities. Practice proved that, based on the experience and similar devices designed in the past, a leading engineer can propose several alternatives of device decomposition and initial attributes of WBS. At the starting point it is not clear which of these alternatives is better in the terms of the design performance. Indeed, each WBS alternative represents the project via the specific types and combinations of atomic activities and requires different types, qualities, and quantities of the resources (staff and skills of designers, specialized software, reusable solutions, etc.). The development of a WBS is a creative task and it would be desirable:

- To provide a Project Manager (PM) with a possibility of, at least, semi-automatic generation of WBS alternatives;
- To enable further rapid evaluation of WBS alternatives from multiple viewpoints: required resources on the time scale, project cost and duration, WBS robustness.

This task is knowledge intensive. To solve it, the supporting software tool needs to maximally use the knowledge of domain experts represented in the terms of domain ontology, develop the generation scheme, specify the task at hand and perform the inference of a WBS. To the best knowledge of the paper authors, this task has not been considered so far. It determines one of the functionality of the software tool in question.

The second important subtask of electronic device design process management is resource constrained scheduling (RCPS) of activities composing WBS. Although the RCPS problem has been researched quite intensively (see, e.g., [7], [8], [9], etc.), electronic device design process management problem is peculiar in some respects. In our framework implemented in the presented software tool the RCPS task is solved via negotiation of software agents modeling a PM and potential contractors (designers) using two-step algorithm. The first one is the selection of a particular activity for allocation and scheduling. The second step is assigning of the selected activity to the particular designer (agent) using argumentation based protocol.

In contrast, with the well known standard RCPS problem [2], electronic device design process is suffers from a number of uncertainties: activity span is a random value distributed differently for different activities and designers, a scheduled resource may randomly become unavailable at the scheduled time, some activities may require to be repeated (possibly, several times) in case if, at a further step, it happens that the intermediate result is not satisfactory. If so, the re-execution of the failed activity may affect several other activities on the same path in the WBS. Other uncertainties also exist. That is why each WBS and the subsequent schedule have to be necessarily validated via simulation under uncertainties. Simulation is also needed for control (the validation of the modified WBS and the re-built schedule). Thus, the simulation subtask is *the third* one composing the entire electronic device design process management routine.

The forth process management subtask in design of microelectronic is project tracking and control aimed to monitor the status of design process and to correct it if noticeable variations of the design process appear.

These four subtasks constitute the generic routine of design process management in microelectronics.

The multi-agent architecture of the developed software tool prototype, further referred to as DEDP-MAS, is outlined in Fig.1. It implements the above described design process management routine. Additionally, a team formation task, which is basically solved by a PM, is a functionality supported by the user interface. It is considered below as a subtask of WBS generation.

This multi-agent system was designed, implemented and deployed using Multi –Agent System Development Kit, MASDK [8], that is a multi-agent platform for analysis design, implementation and deployment of multi-agent applications. MASDK is the tool kit implementing the Gaia methodology [11]. MASDK provides rich graphical user friendly interfaces for a designer.

The architecture includes the agents of two classes (Fig.1): agent class PMA (PM Assistant) and agent class DA (Designer Assistant). The number of PMA agent instances corresponds to the number of projects in progress, whereas the number of DA agent instances is equal to the total number of designers involved in these projects. Each PMA agent instance is "alive" during the respective project execution. MASDK Lite is one more component used for particular project specification, i.e. defining the agent instances of each class, specification of their initial mental models according to the project-related data, and the deployment of the resulting multi-agent system to the given computer network. The specification of agent classes done in MASDK terms are considered as input data of MASDK Lite component.

The Initial mental models of PMA agents include: (i) the knowledge representing "skills" of the designers in regard to perform the activities of various types; (ii) the types of the design artifacts (e.g., functional blocks) to which the activities are to be applied; (iii) the types of the intermediate and final results to be computed referring to the input and output data types to each activity type.

DA agents get their knowledge about the skills and the other information attributed the designers, e.g., the estimations of the activity difficulties, etc. The name spaces formalizing this knowledge are provided by PSI Family of Ontologies [4].

While managing the project, a PM interacts with his /her PMA agent-assistant through user interface. The main window of this interface is shown in Fig.2. The results of project planning, scheduling and performance are visualized by the means of standard MS Project API.

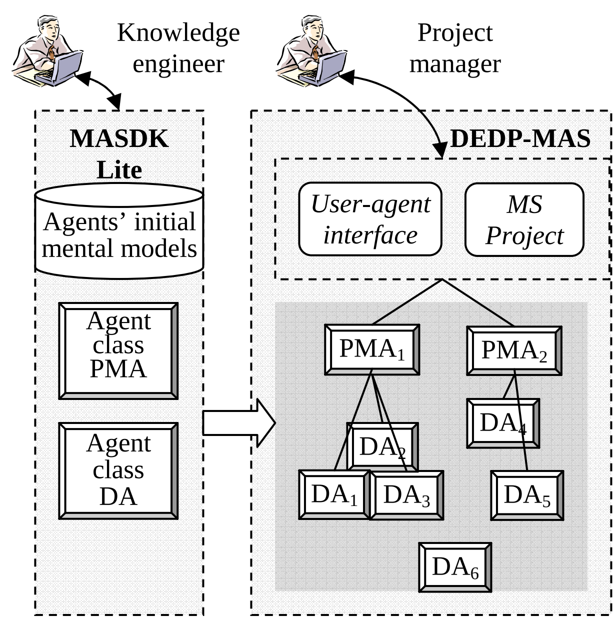

Fig.1. Advanced Multi-agent Software Prototype structure

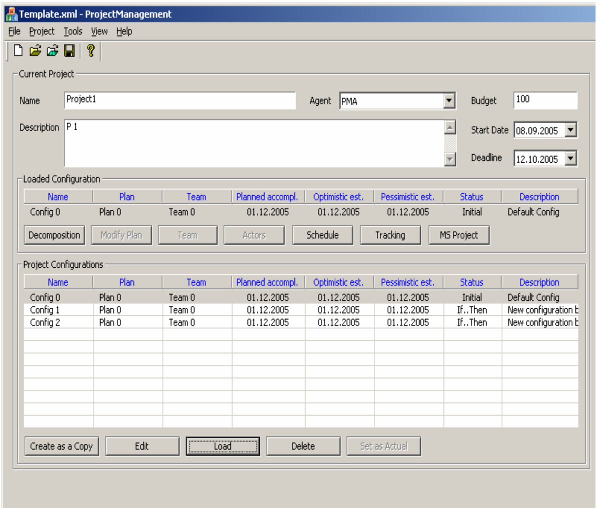

Fig.2. Main window of PM GUI

3. DEDP-MAS functionalities

PMA agent behavior is initiated by a PM using the interface presented in Fig.2. It provides the capability to specify project input data and other information. In the reminder of this section, an outline of how the tool implements the subtasks of a design process management routine (Section 2) and their interactions is done.

3.1. WBS generation

WBS generation is the functionality of a PMA agent. The initial data include the types of design artifacts composing the device (e.g., a chip) under design, the types of initial and final results to be computed for each design artifact. Additional data include the estimates of design artifact difficulties, the upper bound of activity duration, etc. If some duration exceeds the given threshold it is decomposed into sub activities.

PMA agent services constructed of domain experts' knowledge are capable to automatically generate one or several alternatives of the WBS for the same functional decomposition of a device. At this phase of the process management routine, the team formation task is also solved. In this task, the PMA agent asks the DA agents about the abilities of their designers. The PMA agent also should take care that the design team contains at least one potential executor for each type of WBS activities. In general case, several such executors for each type of activity may be the members of a design team.

3.2. Scheduling

Scheduling the WBS activities is classified as MMRCPSP/Gen problem [2]. Schedule optimization is done according to the minimization of overall project performance time. Scheduling is managed by PMA agent. It also conducts the negotiations with DA agents. Scheduling procedure is heuristic step-by-step procedure using sequential schedule generation scheme [7]. At any step, two subtasks are solved. The first one is to select activity to be scheduled. For this selection, a set of activities belonging to the so-called decision set [7] are considered as candidates and selection is done using heuristics rules. The next subtask of scheduling step is the choice of a designer to be allocated the selected activity to execute. The last step is solved through negotiations between the PMA and DA agents in frame of the argumentation-based protocol implementing contracting procedure. In some cases, several activities can be selected for scheduling and resource allocation if some additional relations are imposed on the activities by WBS. In this case, a special coordination of the search is used.

DA agents participating in bargaining for an activity are to take into account their capabilities to perform this activity, its difficulty and complexity, the availability diagrams indicating the time intervals when the designers are free of other commitments, etc.

3.3. Schedule simulation

Simulation is an event-driven procedure generating events indicating the start and the accomplishment of activities executed in uncertain environments according to the developed schedule. Activity durations are randomly generated using β–distribution which is specific for each designer and each type of activity. The probabilities are computed using activity attributes given in WBS.

Event-driven procedure follows all ordering constraints represented in WBS, availability of designers (it may fail due to delay accumulated during performance of previous activities), parallelity relations (if any), etc. Simulation component computes various statistics:
- Either estimating the quality of the developed schedule and resource allocation of the given WBS (probability distribution of the project accomplishment time),
- Or estimating the new schedule developed at project tracking and control phase.

3.4. Project Tracking and control

The objective of project tracking is notification of the PM about the events associated with the project performance to enable him to control the design process to meet the requirements and constraints. The events which happen are recorded with the help of *WBS modification* and Project tracking functions. The execution of these functions, in turn, invokes *re-scheduling*.

3.4.1. WBS modification. The WBS modification is reduced to the repeated inserting of some sequence of activities in it. Such modifications are typical in design projects in microelectronic domain.

3.4.2. Project tracking. This function is intended for registration the factual start and accomplishment times of activities as well as the data describing the rate of activities' progress. These records may later be used for the prediction of the deviation of the scheduled accomplishment time for the activities that are in progress.

3.4.3. Re-scheduling. This function is conceptually the same as the Scheduling function tuned for specific usage. Only non-started activities are scheduled. They can be scheduled in two modes. In the first mode, the schedule is computed from scratch while admitting that for activities the other designers may be allocated and new schedule is computed. In the second mode, the existing schedule is only repaired. It means that the schedule is updated without reallocation of the designers.

4. Typical scenario of functionality usage

This section outlines two basic scenarios used to evaluate the methodology and the developed software prototype. The first one is used in the initial planning while the second one used in design process tracking and control.

4.1. Initial planning

The goal of this initial stage is the choice of an optimal WBS and the team of designers to perform the design. At this stage, several alternatives of WBS are generated and, for each of them, one or more variants of design team are considered. Next, for each pair <WBS, design team> called as *project data configuration*, a schedule is computed, simulated and evaluated. This allows to compare the terms of project execution in different cases and to choose the best pair <WBS, design team>.

4.2. Process tracking and coordination

At a process execution phase, the periodical usage of project tracking and control functions allows to update (correct) the schedule in the cases of unexpected events or delays. The schedule simulation function allows for predicting possible violation of the project deadline. If the schedule is corrected new project data configurations is generated to avoid undesirable violations.

5. Conclusion and Future Work

The plans for the future work assume dropping the simplifications that were agreed upon for the current version of the software prototype. In particular, the following modifications of DEDP-MAS are considered.

- In the current version of DEDP-MAS the agents of DA class so far simulate designers. In the upcoming version they will play role of a designer assistant and are provided by interface to human designers.
- A new agent class playing role of PMA agents' coordinator will be introduced in order support simultaneous coordinated execution of several projects and to resolve conflicts between the project plans, etc.
- A new agent class playing role of a knowledge engineer assistant will be introduced in order to provide the possibility of modification of agents' mental models in their life cycle.

References

1. Balasubramanian, S., Norrie, D. H.: A multi-agent intelligent design system integrating manufacturing and shop-floor control. In: Proc. First International Conference on Multi-Agent Systems., San Francisco (1995), 3-9.
2. Brucker, P., Drexl, A., Mohring, R., Neumann, K. and Pesch, E. Resource-constrained project scheduling: Notations, classification, models and methods, European Journal of Operational research 112, (1999), 3-41.
3. Ermolayev, V.: The State of the Art in Agent-Based Modeling and Simulation of Design Processes. TR-PSI-2-2004. Cadence Design Systems, GmbH (2004).
4. Ermolayev, V., Jentzsch, E., Keberle, N., Samoylov, V., Sohnius, R.: The Family of PSI Ontologies. Reference Specification. V.1.4 Technical Report. Cadence Design Systems, GmbH, Feb., 2006, 47 p.
5. Gorodetsky, V., Ermolaev, V., Matzke, W.-E., Jentzsch, E., Karsaev, O., Keberle, N., Samoilov, V. Agent-based Framework for Simulation and Support of Dynamic Engineering Design Processes in PSI. In *Lecture Notes in Artificial Intelligence*, vol. 3690, Springer, (2005), 511-520.
6. Gorodetsky, V., Karsaev, O., Samoylov, V., Konushy, V., Mankov, E., Malyshev. A. Multi Agent System Development Kit. Chapter in book R.Unland, M.Klusch, M.Calisti (Eds.) Software Agent-Based Applications, Platforms and Development Kits. Whitestein Pub.,(2005).
7. Kolisch, R. and Padman, R. An Integrated Survey on Project Scheduling. Techn. Report 463, University of Kiel, (1997).
8. Kolish, R. Serial and Parallel Resource–Constrained Project Scheduling Methods Revisited: Theory and Computation. European Journ. of Operat. Research, 90, 1996, 320–333.
9. Kolisch, R. and Drexl, A. Adaptive Search for solving Hard Project Scheduling Problems. Naval Research Logistics, vol.43, 1996, 23–40.
10. Shen, W. and Barthes J.-P.: An Experimental Multi-Agent Environment for Engineering Design, Int. J. of Cooperative Information Systems, 5(2-3) (1996) 131-151
11. 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) (2000) 285-312.