Towards Automated Controlling of Human Projectworking Based on Multiagent Systems

Manfred Mauerkirchner and Gerhard Hoefer

Partner of University of Derby in Austria, A-4060 Leonding, Austria
m.mauerkirchner@eduhi.at
g.hoefer@htl-leonding.ac.at

Abstract. Calculating optimized project plans, consisting of an arbitrary number of activities of different types and within a dynamic available pool of human resources characterized by specific profiles can be a most challenging task. We focus mainly on the project control process based on such project plans, which also provide the possibility to integrate different types of external disturbances that normally influence a project workflow. Observed deviations of precalculated time intervals of activities immediately lead to the activation of the so-called self learning components, which automatically tune the personal parameters of each involved workgroup member. At least there are two processes to observe and to synchronize permanently: Real project workflow (the so-called Real System) and its realtime emulation (the so-called Simulation System). Due to all these dynamic requirements and to gain scalability of the system, the implementation selected is based on the usage of different intelligent software agents.

1 Introduction

Due to unpredictable changes of previously planned project resources the control of running projects can be a very demanding task. To automate external generated feedback and corresponding system adaptation we propose a flexible and scalable model which can be implemented best by means of multiagent methodology. At least two processes have to be observed and synchronized simultaneously: real project workflow (Real System) and its real-time emulation (Simulation System), both using a common, permanent adapted and optimized project plan. Such a mechanism could be a suitable means to integrate specific events like allocation, modifying or deleting of project resources in real-time without corrupting the comprehensive system.

As we focus mainly on individuals as project resources we are forced to add several characteristic properties which are typical concerning human resources: Specific skills and capacities and costs to perform certain activities, heuristic relationships between arbitrary persons to describe their abilities to permit groupworking, etc. Finally, the usage of appropriate self learning components should enable the storage of actual knowledge about the involved persons of the resource pool in a transparent way, in order to achieve at least our overall goal: Automated controlling of project working.

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 281–290, 2005.

1.1 Planning Subsystem

A simple way to start with is a directed graph of so-called *project subtasks* (without specific types) to define predecessor-succesor relations between subtasks with estimated execution durations and an arbitrary set of non specific human resources. Because of the very high degree of complexity the use of exact solution methods (i.e. based on the Branch-and-Bound principle) are of minor practical relevance today. Over the years various techniques have been developed that attempt to solve this classic NP problem, known as *Resource Constrained Project Scheduling Problem* (RCPSP). The implementation of heuristic models has been preferred because they generate solutions iteratively by checking so-called *priority rules* and, in addition, weighted by adaptive random sampling methods [6]. Our proposed model is characterized by an arbitrary directed graph of so-called *project activities* (of any type), a pool of human resources with different skills/capacities/costs and a matrix schema to describe heuristic relationships between all members of the pool [7].

Our planning subsystem consists of two parts, each using another methodology: *Local Optimizer* searches optimal project groups within a dynamic environment for activity-specific requests and using appropriate genetic algorithms [4], *Global Optimizer* simulates any project workflow and produces project plans, which are optimized by a numeric non-gradient method by Rosenbrock [11]. As a result all project plans contain a complete workflow of project activities and schedules of all involved persons [8]. The most important requirements of such a planning system are:

- calculation of complete project plans very efficiently and whenever needed
- re-planning of partly executed plans (from any moment on)
- re-planning of partly executed plans with regard to an addional restriction: existing group memberships remain unchanged
- delivery of schedules/plans between two arbitrary dates

1.2 Controlling Subsystem

Based on a common actual project plan, both systems (Real System, Simulation System) are active and running synchronously. Real project workflow becomes emulated and visualized by the so-called *Controlling Subsystem*, used by a responsible project manager. Using a so-called *Controlling Interface* a great part of different external disturbances (category-1 disturbances) that influence real project working will be captured explicitly, i.e. allocation and/or deleting of human resources, deviations from pre-calculated time intervals of already finished project activities, etc.[9] The category-2 type of disturbances concerns personal skills of project members and their relationships within project groups whose integration is completely automated by the methodology described in chapter 2. (Optionally) accompanied by strategic decisions (i.e. modification of logical project flow and/or project structure, etc.) and based on all modified previously stated project parameters, the project manager starts the (re)planning

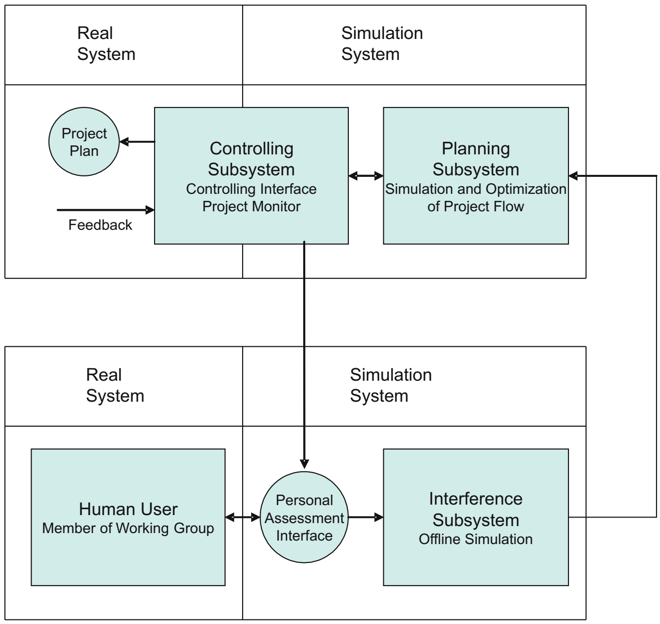

Fig. 1. System View

subsystem asynchronously while both main systems keep running without interruption (see Fig.1).

1.3 Synchronization

Synchronization in our context means that a newly generated project plan (part of a former active plan) becomes effective as actual base for the next future. Activation of new plans may happen automatically on the occasion of a project milestone (begin or end of project activity) or enforced explicitly by a decision of the project manager. Both cases ensure a fully synchronized project system based on an optimized and modified project plan with regard to the actual project conditions. It should be clear that such a system provides many possibilities to inform interested stakeholders about potential risks which eventually could hinder achieving planned results.

2 Self Learning Components

The components described in this section enhance the performance of the overall system by adding the capability of self learning based on personal feedback of all participating group members. The ability of self learning is achieved by adding two components described in this section and helps to increase the problem specific adaptation of the overall system. This kind of adaptation deals with a set of system inherent parameters that represent the characteristics of all human resources participating in the various activities necessary to achieve global goals. This characteristics are classified in two distinct groups : The first group is made up of so-called *skill parameters* whereas the second one contains so-called *interaction parameters*. Both classes are crucial for the synchronization of the real and the simulated project flow. Only realistic assumptions for their values keep both flows in track permanently. Determining the entirety of these values is a demanding task and usually only rough estimations are available in advance. Therefore, tuning all parameter values in a proper way is an inevitable requirement for long time synchronicity of the real and the simulated project flow. This tuning is realized as an offline adaptation process which is executed in two successive steps : The *personal assessment interface* collects suggestions for improving of both skill and group interaction parameter values from all members of the concerned groups. The *interference subsystem* tries to gradually improve the tuning of system inherent parameters by iteratively incorporating the collected suggestions into the actual parameter values (see also Fig.1). Self learning is regarded as a steady process of adaptation that maintains the synchronicity of both project flows by tuning all system relevant parameters.

2.1 Personal Assessment Interface

The tuning process is triggered by an apparent disruption of the synchronicity of the real and the simulated project flow. However, a relevant difference in calculated and real existing execution time of any project activity must not be neglected. As an immediate response the personal assessment interface requests all members of the affected working groups to notify which parameter values may not be chosen properly. Therefore, the feedback of each group member consists of suggestions for the revision of the potentially incorrect parameter values.The resulting suggestions are formalized comments on parameter values of problem relevant user skills and of critical member interactions in each working group. This formalized inquiry is based on standardized questionnaires with regard to all the relevant parameters mentioned above. Questionnaires should have the following structure : One scale bar per each relevant parameter displays the actual parameter value. The displayed values may be adjustable by all individual group members according to their latest group related experiences. Therefore, the scale should represent all possible values of judging skills and interaction that are proposed by modern social psychology [5]. Questionnaires may be realized as electronic documents that are transportable over the intranet. Furthermore, the personal assessment interface has to transfer the potentially conflicting statements of this inquiry to the central interference subsystem and store them in its database.

2.2 Interference Subsystem

The interference subsystem generates context specific updates of the parameter values used by the central planning subsystem in order to re-synchronize the real project flow and its emulation. The collection of stored statements for improval forms the basis for the generation of these updates which is carried out in the following distinct steps.

Step 1 : Create a population of sets that encode one statement for each characteristic parameter respectively.

Step 2 : Determine the fitness of each set by repeating the latest simulation of the planning subsystem offline.

Step 3 : Choose sets that are able to improve the synchronization of the latest activities in the real system and the simulated one.

Step 4 : Generate new populations of bests by performing genetic operations.

Repeat steps 2 to 4 until some improved steady state level is reached.

The updated parameter values are checked by the controlling subsystem for plausibility and then transferred to the planning subsystem for future simulations of the project flow.

Each of these steps needs to be elaborated more in depth :

Step 1 guaranties that the retrieved information about user and group specific parameters is encoded in a proper way for further processing especially for the employment of heuristic optimization strategies. Therefore, it is of great importance that potentially conflicting suggestions of the various group members are distributed over the entire population. This means that in each set there is exactly one group specific gene encoding all group relevant parameters. Group relevant parameters form a set of parameter values containing the skill and interaction values of all group members (see Fig.2).

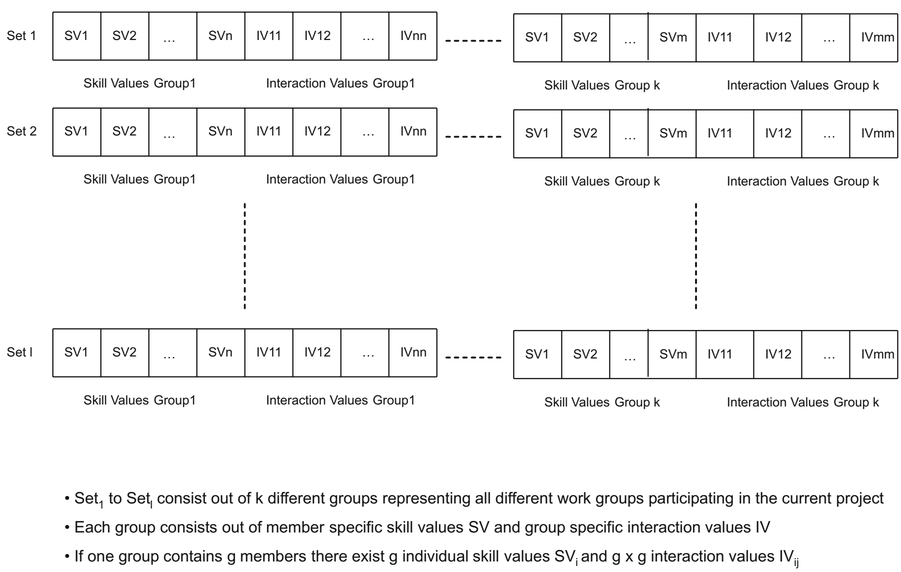

Fig. 2. Population Structure

Step 2 assesses the power of all individual sets of this population to simulate the actual situation in each group of the real project flow. Therefore, an offline re-run of the latest simulation is carried out for all these sets. The potential of each simulation run to re-establish the synchronicity of the real and the simulated project flow might be a convincing measure of fitness. This measure may be obtained by sampling the deviations in real and simulated execution times of all actual group relevant activities. The onset of the sampling interval and its length are of high relevance. The onset of re-running the simulation with the population of manipulated parameter values should be the point in time when activities that cause that deviations in execution times have been started. The length of the sampling interval may be the time interval until all real and simulated activities are finished. The measure is sampled as the sum of all deviation times of the latest activities of the real project flow and their offline simulation. In conclusion, the potentially fittest set of the population yields the shortest sample time.

Step 3 selects the sets that constitute the bases of the population for the next adaptation cycle. Selection may be executed by employing one of the various well established strategies like roulette wheel or tournament selection depending on the individual fitness measures determined in step 2.

Step 4 deals with the organization of the next population. One convenient strategy might be to perform complete or partial crossover of genes that encode for the same group activity and to induce mutations at a low rate for all parameter values. As a result this restriction in crossover operations causes the fact that the parameter specific information retrieved by the personal assessment interface are not spread out on other genes or mixed up in identical genes. Crossover operations are employed to find combinations of manipulated parameter values that may be regarded as being a close representation of all internal affairs of each group. Additionally, mutations are also introduced to check whether some parameter values are sensitive to some further tuning. Mutations should only slightly alter the encoding of one parameter value [10].

2.3 Integration

The components that enable self learning capabilities are extensions of the overall system and realized as the above described personal assessment interface and the interference subsystem. Both extensions are integrated in the overall system in a straight forward way since the interference subsystem is connected to all human users via the personal assessment interface. On the other hand, the interference subsystem acesses the central planning subsystem for its offline test runs which also does not need any change in the overall system architecture.

3 Multiagent Architecture

Considering Fig.1 depicting all features of a project controlling mechanism from a system viewpoint we have to reflect next which underlying software architecture could meet our requirements best: We propose the implementation of

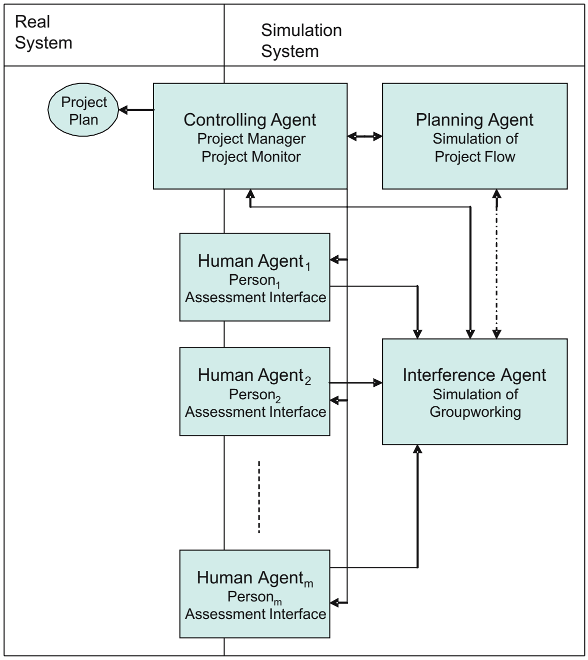

Fig. 3. Multiagent Structure

a so-called *multiagent system*, more exactly the usage of different (intelligent) *software agents*. Fig.3 illustrates the transformation of our previously designed model in a diagram showing all involved issues using agent technology. The next paragraphs show the reasons why the selected technology is appropriate and promising concerning our problem in more detail.

To achieve a basic understanding of the tasks and functioning of intelligent agents we start with characteristic properties that differentiate an intelligent agent from traditional software programs [1]. It should be mentioned that not every agent must provide all stated properties, it rather depends completely on its complexity. The characteristics of intelligent agents can be grouped into two main categories: *internal* and *external properties*. Internal properties define the internal being of an agent determining the actions within the agent. They include the ability to learn, reactivity, autonomy and goal orientations. The external properties comprehend all those characteristics that affect the interaction of several agents or human agent communication. Not all characteristics can be assigned to just one group, rather parts can belong to both groups. The property *character* of an agent is a well known example for the issue which significant parts determine the internal behavior and which parts also play an important role in the external communication [12].

The principle property of agents is *reactivity* which means that agents are able to realize their environment and to react in an acceptable time. All types of agents active in the system described above are basically realized as reactive agents since all activies are initiated primarily by the project manager and individual agents react in a hierarchical order.

Proactivity is regarded as a property grouped a level above reactivity. It does not only react to the changes of its environment but takes itself the initiative under specific circumstances. This ability normally requires that the agent has well defined goals or even a complex goal system in addition. This property is named *goal-orientation*. The control subsystem may be realized as a proactive agent provided that strict rules for initiating a synchronisation are laid down.

Because the complexity of an agent can range widely also areas like rule-based-systems, knowledge-based-systems or neural networks from classical artificial intelligence can be suitable. Reasoning capability and the ability to recognize the environment gives the agent the chance to make specific decisions as a result of changing environment. This *reasoning/learning* is a property of the interference subsystem which is realized using adaptive evolutionary strategies.

The ability of an agent to follow its goal to a certain user defined degree and without user intervention is named *autonomy*. It is one of the key characteristics of intelligent agents. To act autonomously an agent must have both control over its actions and internal states. Autonomy is a property of all agents active in this system since all activities of individual agents are carried out without any monitoring. Only the project manager is allowed to monitor and verify the final outcome of all simulations and learning steps.

The property *mobility* can be described by the ability to move from one computer to another in electronic communications networks. This results in high demands on the network environment, security, data privacy and data management. Every involved computer must be able to pack a mobile agent and send it to another computer as well as to receive new agents. This property is essential for the presented system since the personal assessment interface is realized as a set of mobile agents. All individual human user communication interfaces are made up by the corresponding personal agent.

The previously mentioned property *character* collects all features to interact with human beings over a virtual person, i.e. using an avatar.

The types of agents described above have to interact in a concerted way. The intended way of interaction is determined by three main elements of interaction [2]. These elements are the aim of the agent, its individual abilities, and the resources it demands. The aims of interacting agents may be compatible or imcompatible, whereas their abilities and demanded resources may be sufficient or insufficient to achieve certain tasks. Compatible aims mean that reaching the aim of one agent does not prevent any other agent from reaching its individual aim. An agent with sufficient abilities is able to solve the complete problem on its own. Sufficient resources make sure that interacting agents do not have to compete for the use of certain resources. In the system described above the individual aims of all agents are compatible since the complete system is designed in a modular and hierarchical way. So agents interact in a well defined way that lacks the potential of conflicting aims. The abilities of each participating agent is sufficient to deal with its individuak task but insufficient to take charge of any other task. The resources demanded by each agent may be available without problematic restrictions posed by the overall system since each agent is realized as a distributed process running in its own context. The overall type of this interaction scheme is classified as simple cooperation and typical for communication systems like the one introduced above.

4 Summary and Preview

The overall intention on our proposed way towards automating the project control process is to integrate (nearly) all kinds of external disturbances that influence the precalculated project flow. We distinguish between

- disturbances of category-1, which enforce an immediate decision of the project manager, to start the (re)planning subsystem and furthermore, to synchronize the real project work and its emulation process using the same new calculated project plan.
- disturbances of category-2, which are not neglectable deviations of durations of project activities, normally caused by not corresponding values of personal parameters of the involved human resources (skill parameters and group interaction parameters). In such cases the so-called self learning components will be executed automatically trying to tune the set of personal parameters by requesting suggestions of all group members and optimizing them by using a specific evolutionary methodology. Optimal solutions are recommendations for the project manager, followed by the same procedure like described with category-1 disturbances.

It should be clear that such a permanent modified and adapted system can be used as a resource of information for interested stakeholders about potential risks of the running project at any time. Another obvious benefit is that a permanent updated database of parameters of all involved human resources can be reused within any new projects - a possible way to store specific knowledge.

The transformation of our proposed system in a system of different software agents is a matter of current work. In addition, the implementation of the so-called interference subsystem with the aspect of satisfying all performance requirements, has to be mastered in the next future. Finally it should be mentioned that the base system (planning subsystem, controlling subsystem, integration of category-1 disturbances) has been implemented as a running prototype since 1998.

References

1. W.Brenner, R.Zarnekow, H.Wittig, (1998): Intelligent Software Agents, Foundations and Applications, in *Springer-Verlag Berlin Heidelberg*
2. J.Ferber, (1999): Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence, in *Addison-Wesley Publishing Company*

3. http://www.fipa.org, visited 15-12-2004
4. D.E.Goldberg, (1989): Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley Publishing Company*
5. K.Holm, (1991): Die Befragung I, in *UNI-TB, Stuttgart*
6. R.Kolisch, A.Drexl, A.Sprecher, (1997): Neuere Entwicklungen in der Projektplanung, in *Zeitschrift für betriebswirtschaftliche Forschung, Jg.49*
7. M.Mauerkirchner, (1997): Event Based Modelling and Control of Software Processes, in *Engineering of Computer-Based Systems - Proceedings ECBS'97 Monterey, California*
8. M.Mauerkirchner, (2001): A General Planning Method for Allocation of Human Resource Groups, in *Lecture Notes in Computer Science - Proceedings EUROCAST'01 Las Palmas de Gran Canaria, Spain*
9. M.Mauerkirchner, (1999): Decision Based Adaptive Model for Managing Software Development Projects, in *Lecture Notes in Computer Science - Proceedings EUROCAST'99 Vienna, Austria*
10. H.Pohlheim, (2000): Evolutionäre Algorithmen, in *Springer-Verlag Berlin Heidelberg*
11. H.H.Rosenbrock, (1960): An Automatic Method for Finding the Greatest or Least Value of a Function, in *Computer Journal Vol.4*
12. M.Wooldridge, N.R.Jennings, (1995): Intelligent Agents: Theory and Practice, in *The Knowledge Engineering Review, 10(2)*