

# An Agent Based Information System for Project Portfolio Management

José Alberto Arauzo, Adolfo Lopez Paredes, and Javier Pajares

INSISOC, University of Valladolid  
{arauzo,adolfo,pajares}@insisoc.org

**Abstract.** With the increase in globalization, integration of world markets, and rise in the concept of transnational corporations, the importance of project management has increased many folds. An efficient project management helps new enterprises to achieve competitiveness, speed of response, and closeness to customer demands by improving their flexibility and agility, while maintaining their productivity and quality. But for project management to be successful in the new competitive and dynamic environment management, systems should incorporate new characteristics. In these circumstances, the current challenge is to develop information and control systems for project management that exhibit intelligence, robustness and adaptation to the environment changes and disturbances. The introduction of multi-agent systems paradigms addresses these requirements, bringing the advantages of distribution, autonomy, scalability and re-usability. This paper proposes a distributed information and control system for project portfolio management that integrates strategic issues, planning and control into a community of software agents.

**Keywords:** information systems, multi-agent systems, project portfolio, scheduling.

## 1 Introduction

Projects affect success rate of the organization and thus are vital for them. A project results in bringing something new. It can be a new product or service, increasing sales or capturing of new markets, reducing cost, or an innovative feature in the existing product. Project management helps organizations working in various places integrating their operations and achieving goals.

Project portfolio managers are involved in developing methodologies for valuing projects, selecting the best ones, balancing the portfolio in terms of risk, cost, etc, and coordinating the joint execution of individual projects, so that synergies can be achieved. The output of this process is a set of projects to be done, ranked in terms of strategic and financial importance to the firm.

Once the firm knows the “right projects to do”, it is necessary to plan how “to do things right”. Therefore, project schedules need to be developed, taking into account the availability of resources during each period.

Moreover, uncertainty overruns and dynamic changes make portfolio management more complex. Priority changes over time because of the addition of new interesting

projects, changes in corporate strategy or simply, because of feedback information about individual project overruns affecting their expected returns.

In this paper, we propose an auction approach for online dynamic scheduling in multi-project environments, where projects can be rejected as a result of the scheduling tatonnement process. We use a multi-agent system where both the resources and the projects are artificial agents which play in real time an auction to maximize the portfolio efficiency.

This agent-based approach has two distinctive aspects with respect to other works: the integration of strategic decisions (accepting or rejecting new projects) and operative aspects (resource allocation), and the ability to manage resource flexibility.

## 2 Multi-agent Systems for Project Planning and Control

Projects are characterized by complexity (they include many components and dependencies), uncertainty (about the availability of resources, task durations), dynamic behavior (changes in the scope of the project, adding or removing unexpected tasks, re-scheduling processes) and are inherently distributed (each task may be completed by different resources or in different geographical locations). In the case of a multi-project environment, each one of these features is severely intensified.

In the last decade, many ideas of Artificial Intelligence (AI) have been proposed to solve operation management problems [7]. Multi-agent systems are a distributed IA paradigm precisely designed to deal with problems of these characteristics. As Jennings and Wooldridge point out, multi-agent systems are suitable for problems having the following properties: *complexity*, *openness* (components of the system are not known in advance, can change over time, and are highly heterogeneous, dynamic in project management terms), with *dynamical and unknown environments changing over time* (uncertainty) and *ubiquity* (the activity is distributed over the complete structure) [5][11].

Depending on the physical or organizational structures, many problems can be abstracted and managed as distributed systems using the multi-agent paradigm as metaphor. This decentralized approach in project management has been used since the last decade [13], but it is in the last years when market based approaches [2] are receiving a growing interest. Recently, Lee, Kumara and Chatterjee [8][9] have proposed a multi-agent based dynamic resource scheduling for distributed multiple projects using market mechanisms. Following the same research line, Confessore et al [3] propose an iterative combinatorial auction mechanism as coordination mechanism to resolve the same problem.

Other examples of agent based approaches in project management field can be found in the work of Kim and colleagues [6], Wu and Kotak [12] or Cabac [1]. As underlined in the introduction, our work makes use of a market metaphor and an auction mechanism to help project managers to take portfolio decisions about resources and about portfolio composition.

## 3 Multi-project Environment Definition

We define a multi-project scheduling environment in which the information system must operate.

At any instant  $t$  there are  $I$  projects in the system, each one denoted by  $i$ . Each project is characterized by a value  $V_i$ , that can be interpreted as the revenue obtained for the project, a weight  $w_i$  representing the strategic importance given to the specific project, a desirable delivery date  $D_i$ , a limit delivery date  $D_i^*$  that cannot be exceeded, an arrival date of the project to the system,  $B_i$ , and a limit answer date  $R_i$  that represents the latest date to decide to reject the project.

The system is considered dynamic: while some projects are being developed other projects can be included or rejected in real time.

Each project  $i$  consists of  $J_i$  activities, each one denoted by  $ij$  where  $i \in \{1, 2, \dots, I\}$  and  $j \in \{1, 2, \dots, J_i\}$ . Let  $d_{ij}$  the standard duration associated to activity  $ij$  and  $M$  the set of available resources. Each resource  $m \in \{1, 2, 3 \dots M\}$  can just be assigned simultaneously to one activity (figure 1).

A set  $H$  of  $K$  competences  $H = \{h_1, h_2, \dots, h_K\}$  are necessary to complete the projects. Each resource is endowed with a given cost rate per unit of time,  $c_m$ , and a subset  $H_m$  of  $H$  of competences that can be performed. Each resource has a certain grade or ability to perform a competence. Therefore, the work capacity of resources can be symbolized by means of a vector of abilities per resource  $e_m = (e_{m1}, e_{m2}, \dots, e_{mK})$ , where  $e_{mf} \geq 0$  shows the ability degree of resource  $m$  to perform the competence  $h_f$ . If  $e_{mf} = 0$  then the resource  $m$  has not got the competence  $h_f$ , if  $0 < e_{mf} < 1$  the resource is able to perform inefficiently the competence  $h_f$ , if  $e_{mf} = 1$  it has standard efficiency to perform the competence, and if  $e_{mf} > 1$  it will do it efficiently.

Every activity  $j$  of project  $i$  is associated with a competence  $h(i,j)$ . Any activity  $ij$  with a given  $h(i,j)$  can be performed by a resource  $m$  just if  $e_{m,h(i,j)} \geq 1$ . The duration of the activity  $ij$  depends on the resource assigned to perform it. We denote this duration as  $d_{ijm}$  (duration of activity  $j$  of project  $i$  in resource  $m$ ). It is calculated according to  $d_{ijm} = d_{ij} / e_{m,h(i,j)}$ , where  $d_{ij}$  is the standard duration of activity  $j$  of project  $I$  (activity duration when is performed by a resource that has standard efficiency).

We include explicitly the option of reassigning resources in real time when a new project arrives to the system or when a disruption happens. The activities can be assigned to any resource that has the specific competences to perform it.

The overall efficiency ( $E$ ) of the system will be evaluated by the average benefit obtained in a certain time interval  $T$  according to [1]:

$$E = \frac{B_T}{T} = \frac{\sum_i (V_i - TC_i)}{T} \tag{1}$$

where  $i$  are each one of the projects finished in  $T$  and  $TC_i$  is the total cost to complete the  $i$  project. This cost has two components, the resource direct cost and the delay cost (expression 2).

$$Cost_i = \sum_{j=1}^{J_i} c_{m(i,j)} \cdot d_{ijm} + w_i \cdot \phi \tag{2}$$

$$where \quad \phi = \begin{cases} 0 & si \ F_i \leq D_i \\ w_i \cdot (F_i - D_i)^2 & si \ D_i < F_i \leq D_i^* \\ \infty & si \ F_i > D_i^* \end{cases}$$

The first addend corresponds to the resource direct cost to finish each activity  $j$  of project  $i$ .  $m(i,j)$  denotes the resource selected to comply with activity  $ij$ , and  $c_{m(i,j)}$  the cost rate per unit of time of selected resource. The second addend is the delay cost associated to the project, where  $F_i$  is the real conclusion date of the project. This cost is zero if the project is concluded before the desirable delivery date  $D_i$ . If the project conclude between  $D_i$  and the limit delivery date  $D_i^*$ , the cost will be proportional to the square of the delay over  $D_i$  ( $F_i - D_i$ ). If the project is concluded after  $D_i^*$  the cost will be consider infinite. The problem considers the decision to reject projects if the benefit of perform the project don't compensate the increment of total cost.

## 4 Multi-project Environment Definition

The system is mainly composed of two groups of agents: project and resource managers. A third type of agent is required in the system: The Monitoring-Auctioneer-Creator Agent (MAC). There is only one agent instance of this type which receives bids and asks from projects and resources and computes the exchange price to allocate resources to projects.

### 4.1 Project Manager Agents

Each project is represented by a Project Manager Agent (PMA) which is created and deleted in real time (new projects are evaluated to be accepted in the existing project portfolio, while the projects which finish are removed from the portfolio). Each one is characterized by its tasks, precedence relationships, due date, value, local programs and their execution state. The individual objective of each project agent is to look for 'contracts' with resources that can perform the required activities and hence completing successfully the project. In order to achieve their goal, project agents make a plan that takes into account only their own activities (local schedule).

### 4.2 Resource Manager Agents

The system includes as many Resource Manager Agents as resources are considered. They are defined by their competences or skills and efficiencies. Each resource agent is aiming at performing tasks from project agents. The use of a resource is controlled by means of a *Resource Manager Agent*. Its goal is to increase it's the level of occupation of the resources and the incomes. Project manager agents and resource manager agents make contracts to perform pending tasks of projects. The contracts will be based in the prices that emerge from an auction which is handled by the MAC Agent (Monitoring-Auctioneer-Creating Agent).

### 4.3 Monitoring-Auctioneer-Creating Agent (MAC)

The MAC agent plays three roles of centralized nature: (i) it creates manager Project Manager Agents when new projects are added to the system, (ii) it monitors the resource activities, and (iii) it plays as an auctioneer in a market procedure. This agent compiles real time data of resource states (i.e. availability, allocated activities, and current operations in resources) and shows a summary of this information. Moreover,

as an auctioneer, it bids available time intervals of resources to projects, initiating an auction procedure that allocates activities to time intervals of resources.

#### 4.4 Auction Interactions

In our distributed multi-project system, the decision-making system is decentralized; it is distributed among every agent. Thus, each project creates its own schedule (local schedule) based on its own goal and information. In principle, this procedure entails some weaknesses, e.g. it can bring incompatible local schedules (several projects use simultaneously the same resource) and the local schedules can be globally inefficient, profitable projects may be rejected; most important projects may be delayed, etc. Accordingly, if we aim to design a functional decentralized scheduling process we need to resolve these difficulties. To do that, we propose a market mechanism (it was initially proposed by Davis R and Smith RG [4]) that ensures that local schedules are nearly compatible and globally efficient according to the expression (1). This market based multi-project scheduling approach is founded on Lagrangian Relaxation [10], [14], a decomposition technique for mathematical programming problems.

In order to apply the market metaphor, the periods of time when resources are available are subdivided in a set small time intervals or time slots. Each time slot of every resource is modelled as a ‘good’ that can be sold in an auction where the auctioneer acts as a seller. Thus, a local schedule is a bundle of time slots that has been allocated to a project.

The auctioneer (MAC Agent) proposes a price for each time slot from the current time to the end of the scheduling horizon. The scheduling horizon changes dynamically by coinciding with the latest time slot that some project has asked at any moment.

Project agents play the role of ‘bidders’ in the auction mechanism. They bid for the required set of time slots needed to complete their pending tasks at the current time. Project agents try to find a set of time slots ( $Z_i$ ) through the resource pool while incurring the minimum possible local cost ( $LC_i$ ). This cost has two components (expression 3), the sum of the price of the selected time slots and the delay cost.

$$LC_i = \sum_{m \in Z_i} p_{mt} + w_i \cdot (D_i - F_i)^2 \tag{3}$$

where  $p_{mt}$  is the price of the time slot (t) of the resource (m).

If a project agent do not find a set of time slots that allows to schedule all the pending tasks before  $D_i^*$  with a cost smaller than the project value ( $V_i$ ), it will not ask for any set of time slots. This indicates that the project is unprofitable at the correspondent round of bidding and must be rejected.

Depending on the demand of bids, each resource agent modifies the price of its time slots to maximize its revenue, reducing at the same resource conflicts. In order to get this goal a subgradient optimization algorithm is used to adjust prices at each round of bidding. By means of this algorithm resource agents increase the price of the time slots where there is conflict (i.e. more than a project have asked for this time slot) and reduce the price of those time slots that have not been demanded. The process of price adjustment and bid calculation continues indefinitely decreasing the number of resource conflicts at each round.

### 4.5 Contract Interactions

The auction mechanism described above allows project agents to build compatible and globally efficient local schedules for their pending activities. Agents also interact through a complementary process to make firm agreements based on their local schedules. These agreements determine fixed programs for earliest scheduled tasks. When an agreement is closed, project agents remove the task involved in the agreement from the pending task queue.

## 5 Working Example

To show the system performance we present a little example. We consider three different resources (R1, R2 and R3), endowed with the competences C1, C2 and C3 respectively. In table 1, we show a portfolio of five projects, and the tasks needed to complete the project.

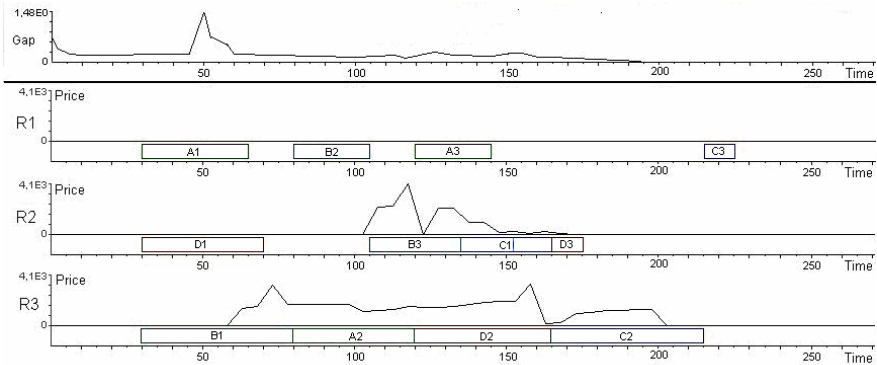
**Table 1.** Example

Projects	Task 1	Task 2	Task 3	Arrival date	Starting Date	DD1	DD2	Value
A	C1 35	C3 40	C1 25	0	0	120	180	30000
B	C3 50	C1 25	C2 30	0	0	120	180	10000
C	C2 30	C3 50	C1 10	50	90	150	270	15000
D	C2 40	C3 45	C2 10	0	0	180	240	12000
E	C3 45	C2 20	C3 50	50	90	150	270	30000

Each task is defined by means of the pertaining competence and expected standard time to be completed. The arrival date is the date when the project is included in the system. Projects can start in the starting date; otherwise, they should have been rejected before this date. Due Date 1 (DD1) is the most desirable duration whereas Due Date 2 (DD2) is the maximum allowed. All the projects have a weight of 1.

Figures 1 shows the system state at a given time (current time). In the upper area of the figures the relative duality gap evolution is presented. The prices of time slots are the solution of the dual problem and the duality gap is a measure of the difference between the primal and dual objective function, and hence it quantifies the quality of the solution [10]. The relative duality gap is calculated as the duality gap divided by the dual solution. A small relative duality gap means that the prices are representative of the system state, thus, a good solution is achieved. In the lower part of the picture, charts of resources are presented.

In figure 1, we show the evolution of the tasks performed by each resource and the prices of the time slots after finishing the simulation. This picture is from the monitoring provide by the MAC Agent. We can see how the duality gap increases when the projects C and E arrive to the system. At this moment previous prices did not reflect the new system state (there are new projects in the system). After some time, the prices change to adapt themselves to the new system conditions, and the gap decreases again. The system not only gives us the dynamic schedule and the rejected projects, but the value of each resource as well.



**Fig. 1.** Tasks performed by resources.  $T_{ij}$  denotes Task  $j$  of project  $P_i$

## 6 Conclusions

Project portfolio management is a management process designed to help an organization to acquire and view information about all of its projects, then sort and prioritize each project according to certain criteria. Currently, in the global market, these activities have high influence in the enterprise competitiveness and.

We propose a multi-agent system and an auction mechanism for online dynamic project selection, scheduling and control in multi-project environments. Projects have tasks to be completed, so they compete for the resources endowed with the capabilities required to do some pieces of work. The prices of resources emerge endogenously by means of an auction process.

We show some of the possibilities of our multi-agent approach to deal with some of the decisions that a manager needs to take within multi-project environments. The system allocates in real time resources to projects, and decides about project acceptance or rejection taking into account project value, profitability and (feedback) operational information. We also show how it is possible to discover which resources are the most valuable, so they should be preserved or added to the firm.

## References

1. Cabac, L.: Multi-agent system: A guiding metaphor for the organization of software development projects. In: Petta, P., Müller, J.P., Klusch, M., Georgeff, M. (eds.) *MATES 2007*. LNCS (LNAI), vol. 4687, pp. 1–12. Springer, Heidelberg (2007)
2. Clearwater, S.: *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific Publishing Company, Singapore (1996)
3. Confessore, G., Giordani, S., Rismondo, S.: A market-based multi-agent system model for decentralized multi-project scheduling. *Annals of Operations Research* 150, 115–135 (2007)
4. Davis, R., Smith, R.G.: Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence* 20, 63–109 (1983)

5. Jennings, N.R., Wooldridge, M.J.: Applying agent technology. *Applied Artificial Intelligence* 9, 357–369 (1995)
6. Kim, K., Paulson, J., Levitt, R.E., Fischer, M.A., Petrie, J.: Distributed coordination of project schedule changes using agent-based compensatory negotiation methodology. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, vol. 17, pp. 115–131 (2003)
7. Kobbacy, K.A.H., Vadera, S., Rasmy, M.H.: AI and OR in management of operations: history and trends. *Journal of the Operational Research Society* 58, 10–28 (2006)
8. Kumara, S.R.T., Lee, Y.H., Chatterjee, K.: Distributed multiproject resource control: A market-based approach. *CIRP Annals - Manufacturing Technology* 51, 367–370 (2002)
9. Lee, Y.H., Kumara, S.R.T., Chatterjee, K.: Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism. *Journal of Intelligent Manufacturing* 14, 471–484 (2003)
10. Luh, P.B., Hoitomt, D.J.: Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique. *IEEE Transactions on Automatic Control* 38(7), 1066–1079 (1991)
11. Wooldridge, M.J.: *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd., New York (2002)
12. Wu, S., Kotak, D.: Agent-based collaborative project management system for distributed manufacturing. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1223–1228 (2003)
13. Yan, Y., Kuphal, T., Bode, J.: Application of Multi-Agent Systems in Project Management. In: *Working Notes of the Agent-Based Manufacturing Workshop 1998*, pp. 160–170 (1998)
14. Zhao, X., Luh, P.B., Wang, J.: Surrogate Gradient Algorithm for Lagrangian Relaxation. *Journal of Optimization Theory and Applications* 100(3), 699–712 (1999)