Technical Report CS-2014-04

A MultiAgent-Based Simulation Model to Support Management Decision Making in Software Development

Davy Baía[a], Carlos J. P. Lucena[a], Donald Cowan[b], Pierre Bommel[c], Carolina Valadares[a], Toacy Oliveira[d]

[a] *Pontifical Catholic University of Rio de Janeiro*
[b] *University of Waterloo*
[c] *CIRAD - GREEN Unit*
[d] *Federal University of Rio de Janeiro*

Abstract

Context: Techniques to support the development of software projects and describe the quality of the results have received substantial attention from the research community because software projects often do not achieve their intended results in terms of factors such as time to completion and quality of the final product.
Objective: In this paper, we propose a MultiAgent-Based Simulation Model to support project managers in the decision-making process that is required create software systems. This Model is based on the PMBOK areas such as cost, time, scope and human resource.
Method: We present an illustrative example that was drawn from our experience with real-world software development projects conducted within our research group. Such example demonstrates the benefits and drawbacks of using MultiAgent Systems simulation as a support for the project manager decision-making process.
Results: We have created a simulation model and associated scenarios that allow a project manager to analyze strategic options throughout the project. Each option is based on the PMBOK practices allowing project managers to investigate the best scenario that meets the project needs such as cost, time and quality.
Conclusions: Some factors are crucial to the success or failure of a project. Decisions made by the project manager are one such set of factors that determine if the project goals are achievable. Through using MABS to simulate

Preprint submitted to INFORMATION AND SOFTWARE TECHNOLOGY March 4, 2014

scenarios we have become convinced that this approach can contribute to the understanding of the phenomena that occur during a software development project and provide real assistance to the project manager.

Keywords: Software development process; Software Project; Project Manager; MultiAgent systems; Simulation; Decision-Making

1. Introduction

Software development in large organizations involves teams working collaboratively to solve problems that are often quite complex. Members of these teams typically follow development processes defined by the organization in which they work. The Project Managers role is to lead the team and the Project Managers activities include: allocating resources, setting priorities, monitoring and measuring progress, and keeping the team focused on the project goal [[1]. To perform these activities effectively, the project manager needs to have knowledge about the progress of the project and related resources. One key activity of the project manager is to assign tasks to each team member. To ensure the best task allocation, the manager requires prior knowledge about the team members and their capabilities (available resources). For instance, in a software development project the developer resource has specific characteristics, such as skills, knowledge and performance levels.

The project manager needs to know these developer characteristics to allocate the tasks in the best possible manner. Assigning a task to a resource that does not fit the required characteristics can cause serious problems such as late deliveries and poor quality products. Moreover, the project managers activities usually involve enormous amounts of information, particularly in the case of large, complex software projects. In such cases, manual recording and data analysis are not really feasible, because of the risk of human error and overall cost and time constraints. Recent surveys indicate that Information Technology (IT) projects continue to have a high failure rate [2, 3] and there is also increasing evidence that the skills of the project manager may be critical for efficient and effective performance of the project team in improving the teams results [4]. Thus, the project manager needs an appropriate toolkit for management support.

Currently, most of the available management tools are based on process simulation [5]. However, there are proposals in the literature to use simulation based on MultiAgent Systems (MAS) to emulate real-life observations [6, 7, 8]. We believe that simulation associated with a model that incorporates features of the software development process such as resources (developers, testers, architects, etc) can reflect reality. The advantages of using MultiAgent-Based Simulation (MABS) are: its ability to support realistic aspects of the problem domain incorporating agent-based features, agent autonomy (e. g., rule-based decision-making, agent delegation) and self-organization (e. g., dynamic resource adaptation) of the environment.

In this paper, we propose a MultiAgent-Based simulation model that incorporates features, which are needed to simulate software project development and as a consequence can support software project management.

Finally, based on a realistic illustrative example, we explain the advantages and disadvantages of using MAS to simulate a software project. In addition, based on the results of the examples design and implementation we highlight benefits and limitations of the use of MABS to support the project managers decision-making activities during a software development project.

2. Related Work

Literature about the use of simulation to assist software development projects first appeared in 1999 [5, 9]. Kellners article [5] describes software process simulation modeling in terms of why, what, and how. The article points out the benefits of simulation to support various aspects of software development management such as process improvements and software project management training. Rus [9] describes the use of a process simulator to support software project planning and management. The model focuses on software reliability but is limited to the factors of cost and schedule.

MABS have also been studied and applied in several other areas. They have been used to improve previous types of simulation such as micro-simulation, object-oriented or individual-based simulation. One of the positive characteristics of MABS is the ability to model individual features [10]. Thus, this approach is able to incorporate aspects of the real world such as the adaptability of social systems. This flexibility contributes to the choice of these simulation features in complex systems.

Recently, articles by Wickenburg, Agarwal and Cherif [6, 7, 8] propose approaches to support the software development process and software project management using MABS. Wickenburg [6] provides a set of general guidelines about when to use MABS and also shows three concrete examples where MABSs seem particularly promising.

Agarwal [7] shows multiple ways in which MABS can be applied to the software development process. She showed that it is possible to simulate the performance of software development teams using performance data of individual team members in small-scale software projects. This approach allows managers to reconfigure the model dynamically without having to change the simulation code. For example, by allowing the addition or removal of developers from the development team, it becomes possible to understand the impact of such changes on a project.

Cherif [8] presents an actual application of MABS to software process simulation modeling. He uses the same basic models to execute both MABS and System Dynamics (SD) simulators. The approach allows relevant comparisons between SD and MABS.

In this paper, we propose a different MABS from the ones already mentioned. The MABS we describe accounts for the adaptability of social systems. The proposal is a MABS for project management that fits more closely with the practices described in the PMBOK Guide (Project Management Body of Knowledge) [1]. From PMBOK it is possible to derive good practices related to a given software development process. We focus on time, cost, scope and human resource issues of PMBOK, which leads to a unique simulation model to be applied to Software Development Processes.

3. Project Management and Software Development Processes

The importance of good project management has been recognized by various companies from all sectors of the economy as an essential factor in the success of their initiatives. A project is a temporary endeavor in which people are engaged to create a product, service or other result [1]. The outcome from the project is meant to solve a concrete problem, by transforming ideas into actions. Some projects are defined through their goals or objectives and within their limitations of resources and time. The projects objectives should be clear and feasible. Every project has a beginning, middle and end. Some projects have a starting date, while others have a delivery date. The delivery date specifies the delivery deadline of the product(s) and/or service(s) established with the projects stakeholders. Another key factor is the set of human resources that should be optimally deployed to ensure the product or service has appropriate quality and is delivered on-time and within budget as specified by the stakeholder(s).

Management attempts to minimize the possibility of failure of a project through the use of a plan covering all project deliverables. The plan takes into account monitoring and control of project activities, thus hopefully minimizing errors that may lead to exceeding estimated costs and delivery dates of the project. Project management is the application of knowledge, skills, tools and techniques to project activities in order to meet stakeholder requirements [1]. For this, the PMBOK Guide [1] proposes 10 knowledge areas for project management, namely: integration, scope, time, cost, quality, human resources, communications, risk, procurement and stakeholder. To produce the expected results, these areas need to be managed from the definition of the project scope to the final delivery of the project results.

As mentioned before, the focus on project management covered by this paper includes the following areas: scope, time, cost and human resources. In the following, we describe the relationship between PMBOK areas and the elements of the software development process.

3.1. The Project Management Processes

The project management process of PMBOK comprises five process groups: initializing, planning, executing, monitoring and controlling, and closing. The areas that we focus on have processes in three groups. In Figure 1, we illustrate the groups and related processes. Some processes in Figure 1 are part of our approach.

According to PMBOK [1], *"Project scope management includes the processes required to ensure that the project comprises all the work required, and only it, to complete the project successfully. The main concern consists of defining and controlling what is or what is not a task included in the project."* By deciding on what constitutes the set of project tasks the project manager can determine all the necessary activities to formulate a good operational plan. The proposed model supports some of this process. In the planning group of the scope area, it will support activities such as plan scope management, requirements collection, scope definition, and creation of the Work Breakdown Structure (WBS). The simulator sets up a WBS by collecting all activities and setting corresponding dates. WBS is one of the main documents in this area of project management as it refers to the work that needs to be accomplished to deliver a product, service, or result with the specified features and functions.

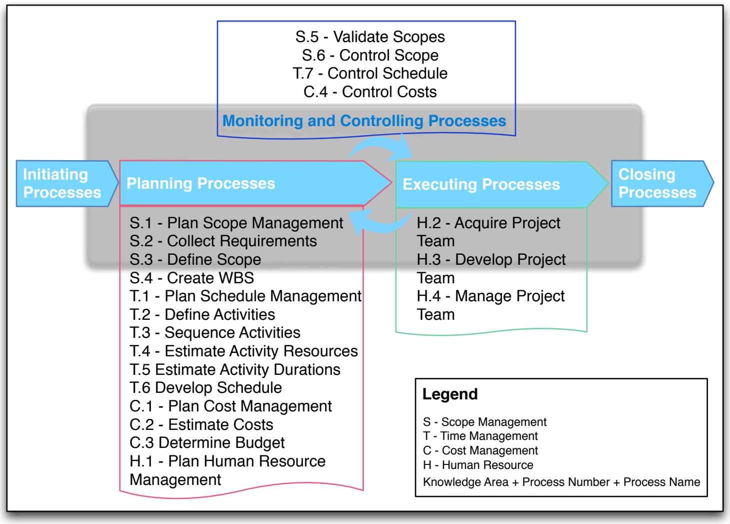


Figure 1: Groups and related Processes.

In the **time area**, the aim is to ensure that the project finishes on time. In the planning group, it has processes such as plan schedule-management, define activities, sequence activities, estimate each activitys required resources, estimate each activitys duration and develop a schedule. Project time management is an important skill for any successful project manager. It is quite common for the real cost to exceed the budgeted cost (scheduled in the project planning process) and one of the possible reasons for this situation to occur is poor time management. Such bad management can cause financial damage and even derail the project. We will handle and support some processes in this area. For example, in the process to estimate activity resources, our model creates one scenario to prioritize the time. Thus, we check the availability of resources and choose the best resource to comply with the time estimate.

In **cost management**, the aim is to develop and control the project budget. Its processes in the planning group are plan cost management, estimate cost and determine budget. According to PMBOK [1], project cost management mainly manages the cost of resources needed to complete scheduled activities. However, it should also consider the effect of project decisions on the cost of operation, maintenance and support of the project product, service or result. We handle and support the processes such as estimating costs and determining budgets. For example, we are able to simulate various teams with distinct costs, thus, bringing one more type of support to the project managers decision process. Thus, the project manager can choose an option that meets costs.

Human resources management of the project includes the processes that organize and manage the project team [1]. The project team is composed of people with skills and responsibilities. Team members should be involved in much of the project-planning process and also in decision-making. The type and number of members of the project team can often change as the project evolves. Human resources management in the planning group has only one process, Plan for Human Resource Management. We will deal with support for some processes in this area. We need all the features of the entire set of resources. From these specifications, we can simulate various teams with the aim of complying with time and resource availability to support the project manager further.

These three areas and their respective processes consume a large amount of the project managers time. In addition, incorrect decision-making can generate unexpected results. The simulation of this process can assist the project manager by checking possible alternatives before making a decision. In the present work we focus on the association between the project management areas and the software development process used in the project.

3.2. The Software Development Process

For the Software Development Process we have used the software process model used at the Software Engineering Lab (LES[1]) at the Pontifical Catholic University of Rio de Janeiro. LES has worked extensively on coordinating and carrying out software systems development for web and desktop in different domains such as petroleum and e-commerce. To perform its various projects, personnel in LES use an adaptation of the Waterfall Model which was first documented by Benington [11] and has been modified over time [12]. We take advantage of the familiarity of the software development community with this model so as to focus the readers attention on the use of MABS to support software management decision-making rather than processes to support software system development. We use the LES process because it meets all the requirements necessary for our work on simulation.

[1]http://www.les.inf.puc-rio.br/

Nevertheless our approach could be used with any other well-documented development process.

Figure 2 shows the elements generated by the software development process used at LES in support of our illustrative example. Each process phase is responsible for some activities. For example, the design phase has results such as **activities list** that run throughout this phase. We can adopt these activities and their features such as duration, person in charge, and knowledge to verify the results of the simulation scenarios. In the next section, we describe the simulation model and how we apply these elements.

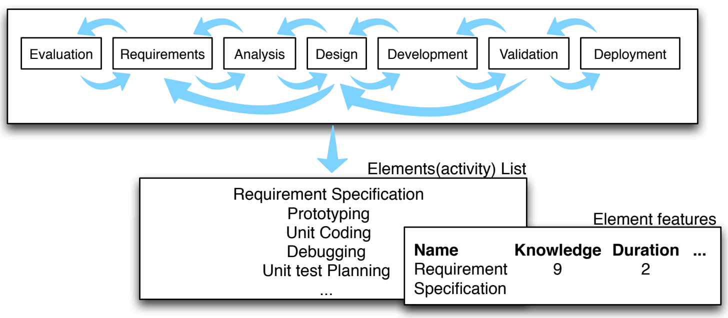


Figure 2: Software development process used at LES.

4. MultiAgent Based Simulation for a Software Development Project

According to Luck [13], there are positive indications that MASs provide strong models to represent real-world environments that are both complex and dynamic. Some of these situations apply in the areas of ecological environment, social science and economics [14, 15, 16]. Others focus on the software development process [6, 7, 8]. The use of MABS to simulate real-world domains can guide solutions to complex physical or social problems, which would probably be unachievable otherwise. For example, we could model the impact on the project of a specific development team or when members of a team change. In these complex situations MABS provide faster and more effective methods for resource allocation than centralized approaches [13]. Thus, we chose MABS to simulate management of software projects.

Software Process Simulation Modeling (SPSM) is based on composing human resources and tasks. Human resources are responsible for the allocation of tasks and their execution. The tasks must be executed by people that have the appropriate skill set. Resources such as developers, test analysts and project managers have unique individual characteristics such as experience and the tasks have distinct features such as level of difficulty. On the other hand, some models based on homogeneity such as a dynamic system [8] consider the features of the developers as represented by the group. This is a model limitation because the model is not able to explain certain observed facts in real life [8]. Since software development is an intense human activity, there is an interest in incorporating social considerations into SPSM.

In this paper we propose a MABS to incorporate the individual characteristics of a feature such as resources and activities in the software development process simulation as it makes the simulation closer to the real world. In the next section we describe the design of the model.

4.1. Model Description

The purpose of this research is to analyze the effectiveness of software design through a MABS by assessing the phenomena that happen during the progress of the project. In section 3 we present the software project composed of activities and human resources. We describe these two components using the flow diagram of Figure 3.

The MABS have agents and entities. They are as follows:

- Resources (Agent): Each human resource is modeled by a simple reactive agent, employing rules of action such as the agents mechanism for running a task. Every agent incorporates features about its nature such as performance and experience. The project manager needs to know the available resources and their features. The project manager also needs to set these resource features through the use of some tool, such as OpenProject[2].

- Project Manager (Agent): In the initial phase, the Project Manager Agent assigns resources to activities and sets dates for activities. The manager is also responsible for allocating the correct activity to the most appropriate resource. If a resource was allocated to an activity that is unavailable then the project manager must look for another available resource.

[2]https://www.openproject.org

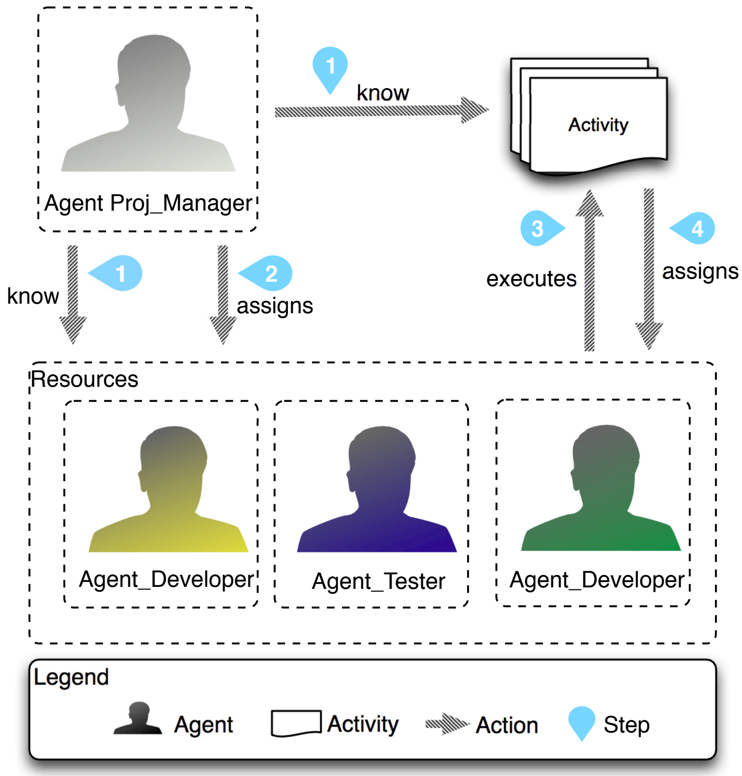


Figure 3: MABS for Software Project.

- Activities (entity): The model needs information about each of the activities. This may be accomplished through the use of a tool such as OpenProject. This tool must have all the information about each activity necessary to be inserted in the model.

Figure 3 illustrates the sequence of steps of the simulation, and these steps are described next.

1. In the first step, the Project Manager needs to know both the activities and resources. Each activity has its own agent-based characteristics and each resource has individual features. In Section 4.2 we describe each one of them.
2. In the next step, the Project Manager assigns the activities based on the characteristics of each activity and its resource profile (i. e., agent

delegation). This step reduces the chance of allocating an activity to a resource that is less suitable in a given simulation. This step also sets the final expected date of the project, the start and final expected date of each activity, the start and final expected date of the work packages. It thus creates a Work Breakdown Structure (WBS).

3. In the third step, the Agents (Resources) execute the activities.
4. The fourth step only initializes when one activity finishes, thus freeing its resources for other activities. Moreover, it sends an alert to all dependent activities informing them that the preceding activity has completed. Thus, all dependent activities are allowed to be executed. In addition, if in the initial moment of an activitys execution it is determined that a resource that had been assigned to that activity is not available, the project manager agent needs to determine what resources are available and which resource has the best profile to execute the activity. This choice is based on a set of rules. Agents can be dynamically allocated to a specific activity/task depending on factors such as availability. The project manager agent assigns this activity. This ensures the autonomy of agents and self-adaptation. The third and fourth steps are repeated until all activities are completed.

4.2. Structural Model of the Simulation

The simulation class diagram has four main classes: Project, Proj_Manager, Resources and Activity. Figure 4 illustrates the set of basic classes for the simulation and their relationships.

To initialize, we need to define some attributes and have a set of activities and their respective characteristics. The attributes are:

- Project - name: project name startDate: start date of the project, finalExpectedDate: expected date for completion;

- AbstractElement An abstract class that can be used to create sub-activities and sub-workpackages.

- Activity - name: activity name, knowledgeLevel: knowledge level expected of the resource that will perform the activity, standardDuration: estimated duration, superActivity: activity that must be complete for this activity to initialize;

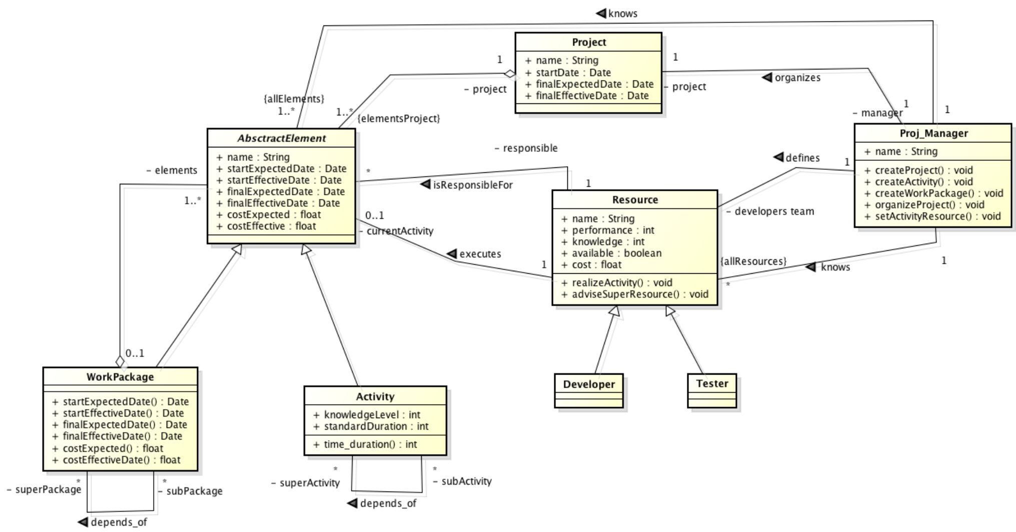


Figure 4: Class Diagram for the model of Software Project Management.

- Resource - name: resource name, performance: resource performance, knowledge: resource knowledge, available: availability of the resource, all are available initially;

- Proj_Manager - name: project manager name.

After the first step, shown in the Figure 3, the second step sets some attributes, such as:

- Activity - startExpectedDate: expected date to initialize the activity, finalExpectedDate: expected date to finish the activity, startEffectiveDate: Actual date of initialization, responsible: assigned resource, costExpected: expected cost of the activity, costEffective: effective cost of the activity.

- WorkPackage - startExpectedDate: expected date to initialize the workpackage, finalExpectedDate: expected date to finalize the workpackage, costExpected: expected cost of the workpackage, costEffective: effective cost of the workpackage.

During the simulation, changes can be made to some of these settings. For example, if an activity needs a resource in order for the activity to start and the resource is unavailable then the project manager will assign another resource to the activity. This is one of the advantages of using the MAS: the possibility of self-adaptation of the system and the autonomy of its agents.

4.3. Activities Diagram for the Project Manager Agent

To illustrate the second, third and fourth steps of Figure 3 better, we create an UML activity diagram of the project manager agent. For each action of the project manager, one activity is chosen. That is, the project manager agent is able to make choices based on its embedded rules.

Initially, the project manager agent receives an activity and then chooses a strategy (cost, quality or time). Each strategy has a rule to be followed. After this, the availability of an associated resource is checked. If the resource is not available the project manager chooses another activity. On the other hand, if the resource is available, the activity will be executed by the project manager agent.

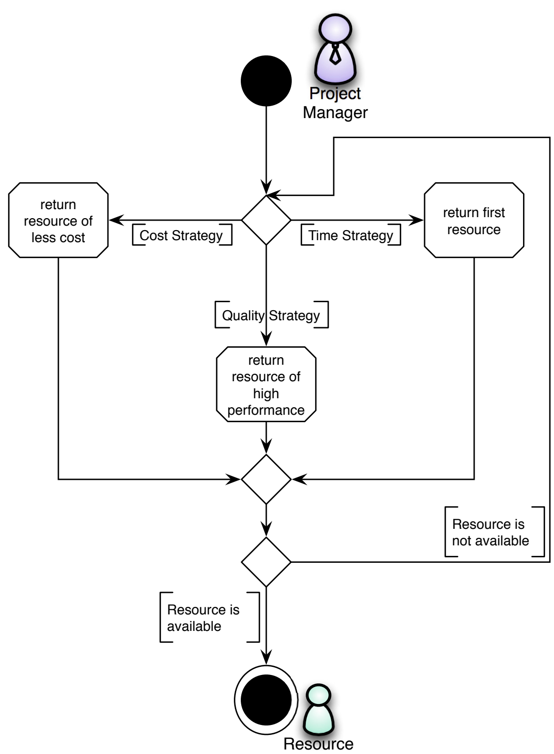


Figure 5: Activities Diagram

5. Realistic Illustrative Example

Our proposed simulation model has been implemented using a simulation environment called CORMAS [17]that is free and open-source software dedicated to natural resource management. Cormas is built on VisualWorks, a cross-platform implementation of the Smalltalk language.

Such a simulation environment has proven to be useful in the understanding of complex interactions between environment and social dynamics. We chose to use the CORMAS simulation environment because our model simulates a set of agents interacting and modifying their environment. In our case the collection of elements represents the elements of a software development process. Moreover, the principles on which CORMAS is based are expressed by different choices. When building a model into CORMAS, one has to consider three aspects of the MVC (Model-View-Controller) design pattern [18]. The first aspect concerns the agents and their interactions with their environment or through direct communication between agents. The second part concerns the overall control of the dynamics, while the third part seeks to define one or several observations of the system via the view concept.

There are several simulation environments that exist such as MASON [19], Madkit [20], TurtleKit [21], Presage [22], Netlogo [23] or JaCaMo [24]. However, we decided to implement our model on the CORMAS platform, because of the simplicity of the framework and its orientation towards social dynamics. Social dynamics are an important paradigm for our problem domain. The other reason was the debugging tools offered by Smalltalk, an interpreted and pure object-oriented language that has proven to be a valuable asset for quickly prototyping our model. When working in a Smalltalk system, the typical edit-compile-run cycle is avoided. Using breakpoints, it is easy to examine the state of the running program, but also to be able to interact dynamically with the agents on the execution stack (i.e. call methods on them, and examine their output), and also to change code and resume its execution. Changing, adding or removing methods dynamically at runtime makes it possible to see a complex application emerging as you develop the components. By altering the value of a parameter on the fly or by voluntarily placing an agent in an extreme situation, it is easier to verify the consistency of the agents behavior.

For better understanding of the illustrative example, we divide it into sections such as the general guidelines, the initial data to start the simulation, and the result of the simulation followed by a discussion.

5.1. General Guidelines

First, we designed our MultiAgent-Based Simulation Model using the PMBOK guidelines, as described in Section 4. Then, we used the CORMAS framework to implement the simulation and to run it. To assess the efficiency of our MABS, the simulations were instantiated from one development project of LES by relating its elements to PMBOK good practices.

5.2. The initial data to start the simulation

Data is needed to initialize a simulation. In our illustrative example, the data consists of the activities, work packages, human resources, and project manager of a real project. For some of these elements, standard information is required. For example, each activity needs a name, a knowledge level and a standard duration. But it also needs to know its mother activity and the work package to which it belongs. Only the name is required for work packages and for the project manager. For human resources, it is necessary to know the resource name, performance, knowledge and cost. Finally, the project needs a name and a starting date. To support this phase, we used the tool for the creation of activities, called OpenProject[3]. The activities were created through the software development process of LES. This data is available in CORMAS [4] and Figure 6 shows sample data. After having inserted the necessary data for the simulation, we can run it and check its results. The simulation run follows the steps of the simulation already mentioned in Section 4.1. The next subsection presents and discusses the results.

The model needs information about the resources and the activities and this information is added by the project manager through a tool (OpenProject), as mentioned before. For our illustrative example the main resource features are knowledge and performance. The project manager ranks the resource knowledge between 1 and 10. Each number is a knowledge area in which the resource is able to run activities. With this, the activities also have the knowledge feature that is needed for the resource knowledge to run this activity. There is also the ranking for resource performance. The values are between 1 and 3, in which 1 means low performance and 3 high performance. These features are ranked according to a project managers experience. Figure 7 describes these features.

[3]https://www.openproject.org

[4]`http://cormas.cirad.fr/fr/applica/MABS_Project_Management.htm`

Activities				
Name_Activities	level_knowledge	standard_duration	superActivity	workpackage
Software Requiment Specification	9	2		Software Design
Software Prototyping	7	4		Software Design
Software Unit Coding	3	4	Software Prototyping	Software Design
Software Unit Debugging	3	3	Software Unit Coding	Software Build
Unit Test Conduct	9	5	Software Unit Debugging	Software Build

Resource			
Name_Resource	performance	knowledge	cost
Davy	1	9	50
Marx	2	4	20
Rafael	2	9	60

Figure 6: Example of Data

Name_Knowledge_Area	Knowledge_level
Test	1
Coding (Java)	3
Software architecture	7
Software requirement. de Software	9

Performance	Performance_level
Low	1
Standard	2
High	3

Figure 7: Resource knowledge and performance evaluation

5.3. Scenarios

For this illustrative example we have created three scenarios. With them, the project manager is able to analyze more strategic options. Thus, the MABS model assists the project manager in decision-making. Scenarios have been created for time, cost and quality. Each scenario uses its own strategy.

The cost scenario uses a strategy focused on lower cost. In the second step, illustrated in the Figure 3, the project manager checks which resources are able to run the activity and chooses the resource with the smallest cost. By proceeding this way the project manager creates a team with activities with the smallest cost. Algorithm 1 describes the pseudocode of the assignment made by the project manager.

TThe time scenario uses a strategy with a focus on time. In the second step illustrated in the Figure 3, the project manager checks the resources

Algorithm 1 Assign resource to activity

```
function AssignResource(Senarios, Activities, Resources)
    for all scenario S : (Cost, Time, Quality) do
        for all activity A Ai : An do
            reourceR.knoledge meet activityA.knoledge;
            assign activity A to the resourceActivity(S, A, resource_enable)
function resourceActivity(S, A, resource_enable)
    if S=Cost then return MIN resource.cost
    if S=Quality then return MAX resource.performance
    if S=Time then return FIRST Resource
```

available to run the activity and chooses the first resource. Algorithm 1 describes pseudocode that specifies this assignment. This allows the project manager to reduce the duration of the project.

Algorithm 2 Resource is not available

```
if Resource is not available then return
    function AssignResource(Senarios, Activities, Resources)
```

Finally the quality scenario has a strategy that focuses on performance. Look at the second step in Figure 3, where the project manager checks resources available to run the activity and chooses the resource with the higher performance. Thus, the project manager creates a team with high performance to accomplish the activities. Algorithm 1 contains pseudocode that includes this assignment.

But if this resource is busy, the project manager chooses another available resource. Algorithms 2shows pseudocode that specifies this approach.

5.4. Result of the simulation and discussion

The simulation is run step by step. In the first step, the project manager agent identifies all activities and resources. After this, he/she creates the Work Breakdown Structures (WBS) and sets the expected initial and final dates, ordered by sequence of activities. In the second step, the simulator finds and starts the resources running. This choice of resources depends on the strategy and scenario that he/she intends to create. The settings for strategy and scenario are some of the simulators support features for the project-manager decision-process where the settings take place automatically.

But this setting may not be the best choice because the simulator accounts for some details such as whether the resource is busy. But there is a solution, which we will describe later. On the other hand, to run these settings through the simulator resource features are needed. This is one limitation of the simulation.

The third and fourth steps cycle until the activities are finished. The resource agent receives one activity and executes it. There are strategies such as time, cost and quality. In the time scenario, if a resource is busy, the resource agent of the mother activity sends one message to the project manager agent that checks if it has other resources available. If they are found, the project manager agent checks if this resource has appropriate features to run this activity. This activity is another support to decision making because the simulator takes care of self-organization. In the scenario with cost or quality, the simulator already has one strategy from its initial configuration. By setting the initial configuration to cost strategy, the simulator finds a resource of lower cost. For the quality strategy the simulator finds a resource of high performance.

Finally, after all activities are finished, the simulator generates a report with the summary of the simulation as illustrated in Figures 8(a), 8(b) and 9 that also describes the options to support the project manager. Figure 8(a) is a chart that shows the ratio between the total cost and the scenarios. Figure 8(b) shows the ratio between the total days and the scenarios. And Figure 9 indicates the ratio between suggested team and scenarios.

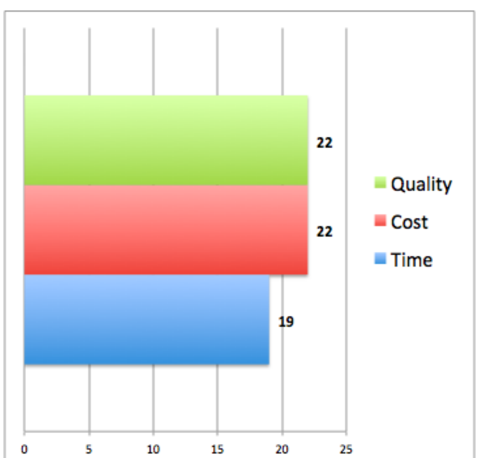
With this simulation, the project manager is assisted in the decision making process while setting up the project. For example, if priority is cost, project manager has suggestions for a team and a simulation with all the steps of the project. With this data, the project manager has solutions to assist in his/her activities. Furthermore, the simulation only needs a model to create several scenarios. The arguments presented so far provide positive evidence that MABS may be used to assist the project manager.

6. Conclusion and Future Work

Some factors are crucial to the success or failure of a project. Decisions made by the project manager are one such set of factors that determine if the project goals are achievable. Through these experiments we have become convinced that MABS can contribute to the understanding of the phenomena that occur during a software development project and provide real assistance to the project manager. With this in mind, we transformed the issues of project management in the PMBOK Guide into software process elements. We also created a simulation model for software projects with the necessary attributes for the simulation.

This paper offers an approach to managing software development projects that incorporates individual features of the environments that make up the development scenarios. This approach requires an effort to collect data and

(a) Project Cost x Scenarios.



(b) Project Days x Scenarios.

Figure 8: Report with the summary of the simulation.

	Team				
Time	Davy	Rafael	Marx	Marx	Davy
Cost	Davy	Davy	Marx	Marx	Davy
Quality	Rafael	Rafael	Marx	Marx	Rafael
	Software Requiment Specification	Software Prototyping	Software Unit Coding	Software Unit Debugging	Unit Test Conduct

Figure 9: Project Team x Scenarios

parameterize the agents. The advantages of using MABS are: its ability to support realistic aspects of the problem domain, agent autonomy and self-organization of the environment. For this, we have to follow a structured and well-defined model namely the simulation model.

In this context, we use MABS to model a software project simulation. The conclusion of the study serves as an initial evaluation of the usefulness of MABS to simulate software projects. In order to assess our model, we designed and implemented a realistic example to illustrate how our MABS helps to support a project manager in decision-making during a software development process.

For future work, we will implement other scenarios such as mergers of the scenarios mentioned in this paper. For this, we will use formalism found in JaCaMo [24] to improve the rules mentioned before. We will also study specific phenomena that occur during the management of the project. In summary, we will continue the research on the possible causes that lead to the success or failure of software projects and the use of MABS to assist in software project management.

[1] PMBOK, A Guide To The Project Management Body Of Knowledge - PMBOK Guides - Fifth Edition, Project Management Institute, 2013.

[2] D. Rubinstein, Standish group report: There's less development chaos today, Software Development Times.

[3] A. Budzier, Why your it project may be riskier than you think.

[4] H. Taylor, J. P. Woelfer, Leadership behaviors in information technology project management: An exploratory study, in: Proceedings of the 2011 44th Hawaii International Conference on System Sciences, 2011.

[5] M. I. Kellner, R. J. Madachy, D. M. Raffo, Software process simulation modeling: Why? what? how?, Journal of Systems and Software 46 (1999) 91–105.

[6] T. Wickenberg, P. Davidsson, On multi agent based simulation of software development processes, Multi-Agent-Based Simulation Ii 2581 (2003) 171–180.

[7] R. Agarwal, D. Umphress, A flexible model for simulation of software development process, Proceedings of the 48th Annual Southeast Regional Conference.

[8] R. Cherif, P. Davidsson, Software development process simulation: multi agent-based simulation versus system dynamics (2010) 73–85.

[9] I. Rus, J. Collofello, P. Lakey, Software process simulation for reliability management, Journal of Systems and Software 46 (1999) 173–182.

[10] A. Drogoul, D. Vanbergue, T. Meurisse, Multi-agent based simulation: Where are the agents?, Multi-agent-based Simulation II (2003) 1–15.

[11] H. D. Benington, Production of large computer programs, Annals of the History of Computing 5 (4) (1983) 350–361.

[12] N. B. Ruparelia, Software development lifecycle models, ACM SIGSOFT Software Engineering Notes 35 (2010) 8.

[13] M. Luck, P. McBurney, Computing as interaction: agent and agreement technologies.

[14] D. F. Adamatti, J. S. Sichman, C. Rabak, Jogoman: A prototype using multi-agent-based simulation and role-playing games in water management, Joint Conference On Multi-Agent Modelling For Environmental Man- Agement (CABM-HEMA-SMAGET).

[15] H. G. Morales, P. Arbeletche, P. Bommel, J. C. Burges, M. Champredonde, J. Corral, J. F. Tourrand, Modelliser le changement dans la gestion des terres de parcours en uruguay, Cahiers Agricultures 19 (2010) 112–117.

[16] C. Le Page, N. Becu, P. Bommel, F. Bousquet, Participatory agent-based simulation for renewable resource management: The role of the cormas simulation platform to nurture a community of practice, Journal of Artificial Societies and Social Simulation.

[17] F. Bousquet, I. Bakam, H. Proton, C. Le Page, Cormas: Common-pool resources and multi-agent systems (1998) 826–837.

[18] G. E. Krasner, S. T. Pope, A cookbook for using the model-view controller user interface paradigm in smalltalk-80, J. Object Oriented Program. 1 (3) (1988) 26–49.

[19] S. Luke, Mason: A multiagent simulation environment, SIMULATION 81 (2005) 517–527.

[20] O. Gutknecht, J. Ferber, The madkit agent platform architecture, Multi-Agent-Based Simulation Ii 1887 (2001) 48–55.

[21] G. Beurier, F. Michel, The turtlekit simulation platform: A multi-agent tool for designing and studying complex systems.

[22] B. Neville, J. Pitt, Presage: A programming environment for the simulation of agent societies (2009) 88–103.

[23] S. Tisue, U. Wilensky, Netlogo: A simple environment for modeling complexity (2004) 16–21.

[24] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, A. Santi, Multi-agent oriented programming with jacamo, Science of Computer Programming 78 (6) (2013) 747–761. `doi:10.1016/j.scico.2011.10.004`.