

Technical Report CS-2014-22

Multi-Agent Based Simulation in Software Project Management: Scope Management Representation and Visualization

Davy Baía¹, Carlos J. P. Lucena¹, Paulo Alencar², Rafael Rocha¹, and
Donald Cowan²

¹ Pontifical Catholic University of Rio de Janeiro, Rj - Brasil
dbaia;lucena;rrocha@inf.puc-rio.br

² University of Waterloo, On - Canada
palencar;dcowan@uwaterloo.ca

Abstract. Software project management (SPM) has increasingly become an essential task in many organizations, especially with the increase in size, complexity of current software systems. However, software projects often change during development time and there is a lack of models and supporting tools to support the SPM process, especially when it comes to simulations that can assess scenarios involving software project dimensions such as scope, time, cost and quality. In this paper we present our work on a multi-agent based simulation approach to decision making in software project management, with a focus on the scope management processes. In the context of these processes, we will present an instance of our approach in which we provide two essential outcomes to support the human interface aspects, namely (i) the representation of the scope processes ; and (ii) visualization techniques using a Work Breakdown Structure (WBS) to show the dynamics of activity flow sequences. To evaluate our approach, we have implemented the representations in JaCaMo, an agent-based framework, and, in terms of coverage, we have shown how it can be mapped to some processes of the SCRUM project management method and to the PMBOK Guide (Project Management Body of Knowledge) scope management process. Based on an exploratory study and our experience, we believe these results help to advance the research area involving the intersection of agents and project management. Especially in terms of modeling and simulation complex and dynamics under conditions pertaining agents, organizations and their environment.

Keywords: Multi-agent, Simulation, Project Management, Scope Planning, Representation, Visualization

1 Introduction

Software project management (SPM) has increasingly become an essential task in many organizations, especially with the increase in size, complexity of current software systems. In SPM, software development projects involve numerous

elements, including resources, project managers, activities, stakeholders, sponsors and the project environment, and changes often happen during development time. These elements related to each other in many ways in order to realise the interactions required to fulfill the project activities and implement the project successfully, and need to be represented and visualized to support software project scope management.

This article focuses on our ongoing work on a Multi-Agent Based Simulation (MABS) approach that incorporates the necessary features needed to simulate software project development and, as consequence, support software project management. MABS modeling provides the ability to capture all relevant attributes in the software development processes, products, and personnel relates to planning, tracking, and controlling involved in software project management [1, 2]. Regarding scope management, simulations needs to be based on the representation of the relations between tasks and their dependencies, critical paths and activity flow sequences. The simulation models can also be used in simulations involving finite resources, delays and estimated costs. These simulations are useful to generate options to decision making, such as options related to cost and estimated schedule and, therefore, can lead to significant improvements. Project managers working in teams can use these simulations to carefully plan and coordinate their efforts so that their projects can be successful [1, 2]. However, there is a need to represent the relations among project elements and provide visualization support in a simulation framework. The representation can be based on rules, norms, missions, beliefs, desires, and intentions (i.e. using the Belief-Desire-Intention, BDI, model). The BDI model can be supported by tools such as JaCaMo [3], which uses models that represent (i) agents and their interactions, (ii) organizations, and (iii) environment.

In this paper we present our work on a multi-agent based simulation approach to decision making in software project management, with a focus on the scope management processes. In the context of these processes, we will present an instance of our approach in which we provide two essential outcomes to support the human interface aspects, namely (i) the representation of the scope processes; and (ii) visualization techniques using a Work Breakdown Structure (WBS) to show the dynamics of activity flow sequences. To evaluate our approach, we have implemented the representations in JaCaMo, an agent-based framework, and, in terms of coverage, we have shown how it can be mapped to some processes of the SCRUM project management method and to the PMBOK scope management process. Based on an exploratory study and our experience, we believe these results help to advance the research area involving the intersection of agents and project management. Especially in terms of modeling and simulation complex and dynamics under conditions pertaining agents, organizations and their environment.

2 An Overview

The presented work is an extension of a Multi-Agent Based Simulation approach [12], which fits closely with the practices described in the PMBOK Guide and focuses on time, cost, scope and human resource issues addressed in the guide. The approach leads to a unique simulation model that can be applied to software development processes (e.g., for the waterfall model). Figure 1 shows four components necessary to depict a software project. As it can be seen in the figure, in general, our approach can be supported by different simulation environments. In previous work, MABS has relied on the CORMAS environment, which provides support to social aspects and visualization.

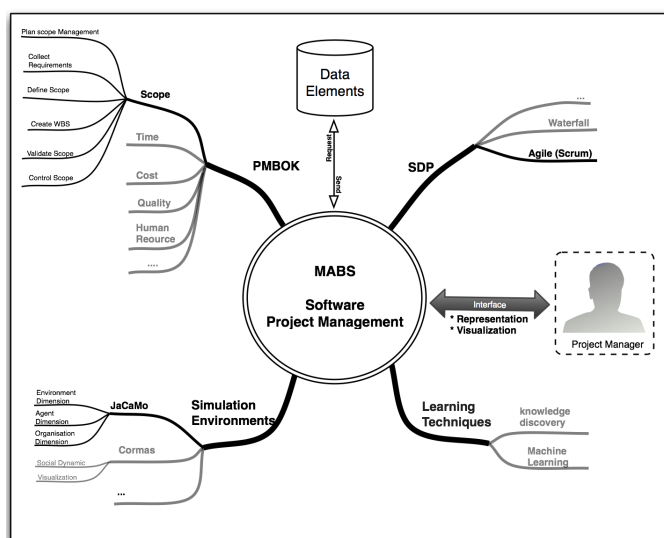


Fig. 1. Overview of the approach.

Figure 1 shows project-related topics such as PMBOK, Software Development Processes (SDP), Simulation Environments, Learning techniques and Interface. In this paper we focus on the topics depicted by darker lines. In terms of PMBOK, we focus on scope management processes. Regarding Simulation Environments, we rely on the JaCaMo framework, which supports the features needed in our representation. Regarding Software Development Processes, we focus on an Agile Methodology (i. e., SCRUM). Finally, as it can be seen at the central right-hand in the figure, we focus in this paper on interface aspects, namely representation and visualization. As a whole, the proposed extensions, illustrated by the new branches in the figure, contribute to enhance the set of techniques and tools that can assist project managers in their decision-making process.

3 Related Work

Work related to this paper involves three main lines of research: (i) agents in business process management; (ii) agents in project management; and agent-based tool and applications. First, different workflow systems have been provided to automate the processes of a business (e. g., [13, 6]).

Second, software agents have been used in many ways to support project management activities. Work has been provided to support the management process [14] and the scheduling of tasks within the supply chain in the project management environment [20]. Similarly, agents have been used industrially in supply chain management [24, 15]. In addition, methods and tools have been developed to support workflow planning and management, especially in design domains (e. g., [11]). Some researchers have also focused on resource management in virtual organizations [21] and plan tracking [24]. Planning for software project management has relied in some cases in adapted simulation-based planning algorithms [7]. Regarding control and monitoring, software agents have been used at various sites in distributed software engineering processes [4]. Regarding quality assurance, a multi-agent environment framework was provided to support process and quality assurance, in which an agent plays the role of a project manager and helps to assign and support activities such as testing and reviews [10]. Other approaches rely on software agents to support risk management in robot systems [9] and to prevent technical risks in small and medium enterprises [18].

Finally, concerning the JaCaMo agent-based framework specifically [3], we are using in our approach, numerous applications have been provided, including work on smart co-working spaces [8], agent-based machine-to-machine management infrastructure [16], and ontology-based knowledge management for knowledge-intensive workflows in business processes [22].

In general, in contrast with related work, our approach focuses on agent-based simulation and takes advantage of the different features supported by JaCaMo, which uses models that represent different aspects related to agents and their interactions, organizations, and environment. Thus, assisting in the development process that is complex and dynamics and often change. These aspects involve complex relationships and interactions that need to be explicitly represented. How can we represent and provide simulations and their visualization based on these relationships. The underlying assumption is that research geared towards improving our understanding about these relationships, which involve elements such as resources, project managers, activities and the environment, and their representation and visualization can help to support the associated complex and inter-related software project management processes.

4 Project Scope Management and Multi-Agent Based Simulation

According to PMBOK[17] "Project Scope Management includes the processes required to ensure that the project includes all the work required, and only

the work required, to complete the project successfully.” Software project scope management involves six processes, namely Plan Scope Management, Collect Requirements, Define Scope, Create Work Breakdown Structure (WBS), Validate Scope, and Control Scope. Each of these processes has its related inputs, tools and techniques, and outputs.

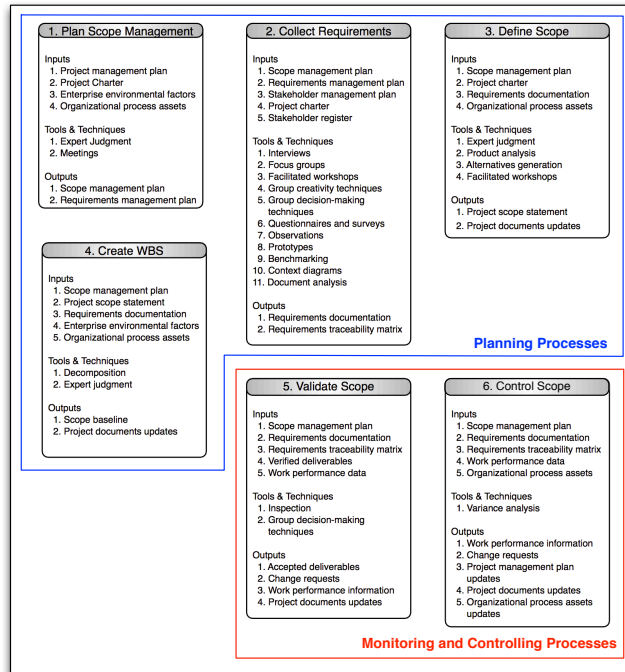


Fig. 2. Groups and related Processes.

For each process mentioned before, we can formulate the inputs using the JaCaMo platform and thus create a new tool to support the project manager decision-making.

4.1 Multi-Agent Based Simulation with JaCaMo

JaCaMo is a platform to support multi-agent oriented programming. JaCaMo was built upon three existing platforms: Jason for programming autonomous agents, Moise for programming agent organizations, and Cartago for programming shared environments. As a result, JaCaMo provides a unifying perspective in programming agents, organizations, and environments.

According to Boissier et al[3], in the agent dimension in JaCaMo, the agent is an entity composed of a set of beliefs, representing the agent’s current state

and knowledge about the environment in which it is placed, a set of goals, which correspond to tasks, that the agent has to achieve, and a set of plans which are courses of action, either internal or external, triggered by events, which the agents can dynamically compose, instantiate and execute to achieve their goals. Events can be related to changes either to the agent's belief base or to its goals. For example, the available resource agents have specific beliefs in a certain state and perform certain activities to achieve their goals based on a strategy (i.e., a plan). In the next section we will show how to represent beliefs, goals and plans.

In JaCaMo [3], the environment dimension is composed of one or more workspaces, which are used to define the topology of the environment. Each workspace is a logical place containing a dynamic set of artifacts, defining the environment structure and behaviour, and representing the resources and tools that agents can create, discover, perceive, and use at runtime. Each artifact provides a set of operations and observable properties defining an artifact's usage interface, used by agents to observe and operate the artifacts. The execution of operations can generate updates to the observable properties and specific observable events. For example, in a software development project, the artifacts are tasks that will be executed by resources, and each task set is related to a workpackage.

Regarding JaCaMo's [3] organizational dimension, an organisation is described from a structural point of view, in terms of group and role entities (e.g., to specify how much and which resources are necessary for project). From a functional point of view, in terms of a social scheme, mission and goal entities are introduced. The main project goal can be divided into multiple subgoals. However, there are missions that can be related to activities, and a set of missions can be a subgoal. From a normative point of view, in terms of norms that bind roles to missions, it is possible to constrain an agent's behaviour with respect to sets of goals that have to be achieved when it chooses to enter a group and play a certain role in it (e.g., an activity has a norm of the type obligation that is related to the mission that must be fulfilled).

Finally, these three dimensions incorporate important features to assist the software project simulation and its complex relationships and, since JaCaMo relies on these dimensions, we will use this framework to support and represent project management relationships from an agent-oriented perspective and define an approach based on the supporting techniques and tools to assist project managers in the scope processes.

4.2 Plan Scope Management

Plan Scope Management is a process in the planning phase. This process documents how the project scope will be defined, validated and controlled. According to the good practices described in PMBOK[17], the key benefit of this process is that it provides direction on how the scope will be managed throughout the project. Figure 2 shows the inputs related this process.

We can rely on the features of the three dimensions mentioned previously and its features to formalize these inputs and its complex relationships. In this way,

we can use and simulate and use the formalized inputs. For example, we can rely on the organisational dimension to formalize the project participant roles. This formalization has features related to the inputs (e. g., enterprise environmental factors). Code List 1.1 shows this formalization.

Listing 1.1. Role definition to project.

```

1 <role-definitions>
2 <role id="project_owner" />
3 <role id="sponsor" />
4 <role id="building_resource" />
5 <role id="software_requirement_analyst" > <extends role="
   building_resource" /> </role>
6 <role id="programmer" > <extends role="building_resource"
   /> </role>
7 </role-definitions>
8
9 <group-specification id="project_group">
10 <roles>
11 <role id="project_owner" min="1" max="1" />
12 <role id="software_requirement_analyst" min="1" max="2" /
   >
13 <role id="programmer" min="1" max="4" />
14 <role id="sponsor" min="1" max="4" />
15 </roles>
16 </group-specification>

```

The *role-definitions* tag defines the roles and this tag can extend other elements. For example, *software_requirement_analyst* role extends *building_resource*. Thus, the *software_requirement_analyst* has all the features related to *building_resource* in addition to his or her own features. We also need to define the participants related to each role and their number. The *group-specification* tag assigns roles to groups and specifies the number of participants.

The project manager can use these representations and relationships in the simulation. In this way, these types of relationships as techniques can support support the scope management process and help to create the requirements management plan.

4.3 Collect Requirement

Collect requirements is a process belonging to the planning phase. This process determines, documents, and manages the stakeholders' needs and requirements to meet the project objectives. According to the good PMBOK's [17] practices, the key benefit of this process is that it provides the basis for defining and managing the project scope, including the product scope. Figure 2 shows the inputs related to this process.

Stakeholders have an important role in project management. Their active involvement can contribute to a project's success because the project is created for

them and with them. From the stakeholders' perspective, we need to discover and decompose their needs into specific requirements. However, we also need formalize these requirement and create a structure of the goals to be achieved to comply with the requirement. For this purpose, we can use the organisational dimension and its features to formalize the inputs of the project-related needs and their complex relationships. For example, we will use the project charter's inputs to formalize how the goals (one of the expected results) are the decomposed into tasks. Thus, we can create tasks to be executed to accomplish these goals. Code List 1.2 shows this formalization.

Listing 1.2. Definition to project tasks.

```

1 <scheme id="build_project_sch">
2   <goal id="project_built">
3     <plan operator="sequence">
4       <goal id="sw_Requirement_Specification" ttf="3 days"
5         />
6       <goal id="software_Prototyping_done" ttf="3 days" />
7     </plan>
8   </goal>
9 </scheme>

```

The *build_project_sch* tag is defined by an schema as follows. It incorporates the goal *project_built* that has a plan that complies with the project needs. This plan can have sequence or parallel goals. In addition, we can define the time to target the goal. We also need define the tasks related to these goals. For this purpose, we use the environment dimension. In Section 4.4, we will show this definition. Figure 2 shows the inputs related this process.

Stakeholders can perform simulations using this formalization. Thus, we can use this representation, as we are doing in this paper, to support our approach or incorporate it into existing tools or techniques (e.g. group decision-making techniques, prototypes). In this way we are supporting project managers in the creation of the requirements documentation and the requirements traceability matrix.

4.4 Define Scope

After completing the requirements collection process we have to define the scope process. The objective is to develop a detailed description of the project and the product. According to PMBOK [17], the key benefit of this process is that it describes the product, service or result boundaries by defining which of the collected requirements will be included or excluded from the project scope. Figure 2 shows the inputs related this process.

Define scope is a process in the planning phase. This is an important process because a detailed preparation of the project scope statement is critical to project success and leads to the major deliverables, assumptions, and constraints that are documented during project initiation. For this purpose, we can use the

environment dimension and its features to formalize the scope definition (inputs) and its complex relationships. For example, we can use the requirements documentation (an input) to support the creation of these tasks. Thus, we can simulate the tasks and check whether the related goals are fulfilled. Code List 1.3 shows this formalization.

Listing 1.3. Activities for objectives.

```

1 +!sw_Requirement_Specification <-
   specify_software_requirement .
2 +!software_Prototyping_done <- softwarePrototype .

```

As we have mentioned previously, project goals are related to specific tasks. Therefore, every goal has a tasks list. For example, in Code List 1.3, to comply with the *sw_Requirement_Specification* goal the task *specify_software_requirement* needs to be executed. Thus, we can use these specifications to support the creation of the project scope statement and to update the product documents (e.g., stakeholder register, requirements documentation).

4.5 Create WBS

Create WBS is the last process in the planning phase. This process subdivides the project deliverables and project work into smaller, more manageable components. According to PMBOK [17], the key benefit of this process is that it provides a structured vision of what has to be delivered. The WBS is a hierarchically organized scope of work to be carried out by the project team to fulfill the project objectives and create the required deliverables.

It is used in the formalization of all other processes mentioned previously with function as inputs to this process. Figure 2 shows the inputs related this process. In addition, we add more information related to the enterprise environmental factors (e.g., the specific software development process). In section 5, we will describe how the WBS creation depends on the specific software development process adopted in the enterprise.

Listing 1.4. Role appropriate to task.

```

1 task_roles("Software_Requirement_Specification", [
   software_requirement_analyst]) .
2 task_roles("Software_Prototyping", [programmer]) .

```

Code List 1.4 shows the additional information necessary for the creation of the WBS. Each activity is associated with a role, and this is a specific rule to be followed. We use the organization dimension and its features to formalize these rules. Thus, the project relationships can be defined as rules and the system has to follow them. For example, it must be ensured that only a *software_requirement_analyst* can perform the *Software_Requirement_Specification* task.

The Create WBS process uses all dimensions to provide enough features to incorporate the necessary relationships among software project elements. In this

way, we have a formalization that can be used in our approach, which supports the creation of the scope baseline and the project document updates.

4.6 Validate Scope

Validate scope is a process in the project monitoring and controlling phase. This process formalizes the acceptance of the completed project deliverables. According to PMBOK [17], the key benefit of this process is that it brings objectivity to the acceptance process. Figure 2 shows the inputs related this process.

Customers or sponsors need to review and verify the project deliverables in order to ensure that they are completed satisfactorily. This process uses outputs of the other processes mentioned previously, to perform the validation and to accept the final project or product. We can formalize the inputs of this process, e. g., verified deliverables or work performance data, using the organization environment. Figure 2 shows the inputs related this process. Further, we can formalize the customer or sponsor expectations with respect to each deliverable using missions. In addition, norms can be used to specify constraints involving roles and the project mission. Code List 1.5 shows this formalization.

Listing 1.5. Mission for role .

```

1 <mission id="management_of_project_building" min="1" max="
  1">
2   <goal id="project_built" />
3 </mission>
4 <mission id="specify_SoftwareRequirement" min="1" max="
  1">
5   <goal id="software_Requirement_Specification_done" />
6 </mission>
7 <mission id="software_Prototype" min="1" max="1">
8   <goal id="software_Prototyping_done" />
9 </mission>

```

The *management_of_project_building* tag defines the project mission. It incorporates the goal *project_built* that has to be fulfilled by a role. This role can be defined using norms. In addition, we can define the roles that can execute the missions. We also need to define the norms related to these mission. To define these norms we use the environment dimension again.

Using these features, the validate scope process has the necessary formalization that can be used in our approach, and these representations can support project managers to deal with work performance information, change requests and acceptance of the deliverables.

4.7 Control Scope

Control Scope is a process in the monitoring and controlling phase. This process monitors the status of the project and product scope and changes it according

to a baseline. According to PMBOK [17], the key benefit of this process is that it allows the scope baseline to be satisfied throughout the project.

The project manager uses this process to assess the evolution of the project. In addition, it is used to manage changes. To support this process we use the formalization of all other process mentioned previously with act as its inputs. Figure 2 shows the inputs related this process. After this step, the simulator is able to instantiate the formalizations previously defined, and simulations involving scope control can be executed.

Variance analysis is a technique used in this process. Its objective is to determine the cause and degree of differences between the baseline and the actual performance [1]. The baseline can be generated by the simulator. In this way, the project manager has the necessary support to conduct relevant comparisons, and the simulation and its features can support the execution of the outputs of this process, e. g., change requests and plan updates.

5 Relation between PMBOK and SDP

Software Projects that adopt PMBOK often have a related Software Development Process (SDP). Two of these software development processes are the Agile and the Waterfall methods, which have been developed to support project teams in the software construction process. In previous work [12], we have used the Waterfall Model in conjunction with PMBOK as an example. In this paper, we use an Agile model (i. e., SCRUM) to support the PMBOK guidelines. Users of the Agile method are increasing day by day and this method has attracted increasing attention because its processes are simple to adopt. In fact, agile is friendly and has fewer rules (e.g., little documentation) than other methods. On the other hand, Agile does not create the documentation required for projects that require a higher level of detail. However, this method is adopted in projects that require a great deal of interaction among project participants and little documentation. It is used because it is an excellent method to support these features. Some researchers have described approaches for improving the documentation related to the Agile method (e.g., [19]).

The Waterfall method was the first method introduced to support software development processes. This method has been important because several methods are based on its approach. Further, it provides a well-defined structure. Large-scale projects require several artifacts, and this method incorporates artifacts that can be adapted to different needs, which is an advantage. Nevertheless, in order to provide large number of artifacts, a great number of resources are required.

According to Highsmith [5], the definition of "Agile" is the ability to both create and respond to changes in order to profit in a turbulent business environment. There are three main focal points within Agile. The first one is that organizations are "chaordic", having characteristics of both chaos and order. The second point is that trust is the basis for all internal and external relationships, with a focus on collaboration and not on contract negotiations. The final focal

point is that the focus lies first on people and interactions, and second on processes [23]. Within the Agile development world there are several development approaches. We are using the SCRUM approach in this paper. We have chosen SCRUM because it has been used in some projects conducted in the Software Engineering Lab (LES) at the Pontical Catholic University of Rio de Janeiro.

According to SCRUM, the development process happens through iterations called sprints. Daily SCRUM meetings (30 minutes or less) are held with the entire team to identify roadblocks and provide status feedback. Requirements are held constant through the sprint to provide some stability in a quickly changing environment. Our approach can be used to build the sprints of the SCRUM methodology. Thus, project complexity and dynamics can be supported by simulations. Each sprint uses the simulation results to support its definition. For example, project managers can run several simulations and then choose the option which complies with the project goal. In addition, simulation results and their representation can be seen a document type since they were built according to project requirements.

Several enterprises have used these two approaches (the Agile and the Waterfall methods) in combination and have benefitted from such a hybrid solution. Merging brings agility and helps to create the documentation required in some projects. However, in this case it is necessary that an experienced manager leads the project effectively and that appropriated management tools as used. As mentioned in Section 4.5, the scope management process depends on the model adopted by the enterprise, in this case SCRUM. Thus, each iteration can be defined by a task set and each of the tasks may have a subtask set. However, the persons involved in a project can define their interactions and daily meetings using the simulator, which provides in this way support to the software management processes.

5.1 Relation among PMBOK, Scrum, and JaCaMo

Table 1 shows the relation among PMBOK, Scrum, JaCaMo, and JaCaMoPM. JaCaMoPM is the approach and tool that we are describing in this paper, which is built on top of JaCaMo, an agent-based framework that considers the three dimensions that need to be represented in the scope management process (i. e., the agent and their interactions, the organization and the environment).

Assuming that the focus is on the scope management process, this table shows that all the processes (P1-P6) defined in PMBOK can be mapped to SCRUM activities. For example, the SCRUM activity "Perform domain analysis for building domain model" can be associated with the PMBOK processes P1 (Plan Scope Management) and P2 (Collect Requirements). Similarly, we have shown that all the PMBOK processes (P1-P6) can also be mapped to JaCaMo or JaCaMoPM. The PMBOK processes P1 to P6 can be mapped to the JaCaMo representation features and the PMBOK process P6 can be mapped to the JaCaMoPM visualization features (i. e., the WBS visualization and the WBS status presentation).

Table 1. Relation among PMBOK, Scrum, and JaCaMo

PMBOK	SCRUM	JaCaMo	JaCaMoPM
P.1 Plan Scope Management	Perform domain analysis for building domain model. (P.1 , P.2)	Role-definitions, Group specification (P.1, P.3, P.4, P.6)	WBS visualisation (P.4, P.5, P.6)
P.2 Collect Requirements	Development of a comprehensive product backlog list. (P.2)	Scheme, Goal, Plan (P.1, P.2, P.5)	WBS status presentation (P.6)
P.3 Define Scope	Development of a comprehensive product sprint backlog. (P.3)	Tasks for goal (P.3, P.4)	
P.4 Create Work Breakdown Structure (WBS)	Definition of the functionality that will be included in each release. (P.4)	Role appropriate to task (P.3, P.4)	
P.5 Validate Scope	Selection of the release most appropriate for immediate development. (P.5)	Mission rules (P.5)	
P.6 Control Scope	Review of progress for assigned backlog items. (P.6)		

6 WBS Visualization and Dynamic Status

Figure 3 shows the simulations being executed. The simulator creates a WBS with all the tasks required to be performed. In this way, the simulation can flow from task to task as each task is executed. In this figure, each blue box represents a scheduled task, which is named by a task name and have no associated resource name. After a task is performed, its associated box changes its color (from blue to green) and is also labelled by the resource name that completed the task. In this way, not only the proposed approach can support the representation of the necessary features needed in the scope management process, but can also provide a convenient WBS visualization. A special feature of the WBS visualization is that it allows project managers, through changes in color, to follow the dynamics of the activity flow sequences, which help to support the validation and control of the project scope.

7 Conclusions and Future Work

In this paper we have presented our work on a multi-agent based simulation approach to decision making in software project management, with a focus on the scope management processes. In the context of these processes, we will present an instance of our approach. which we call JaCaMoPM, in which, according to the Figure 1's darker branches, we focus on the PMBOK scope management processes, SCRUM as a software development process (SDP), JaCaMo as an

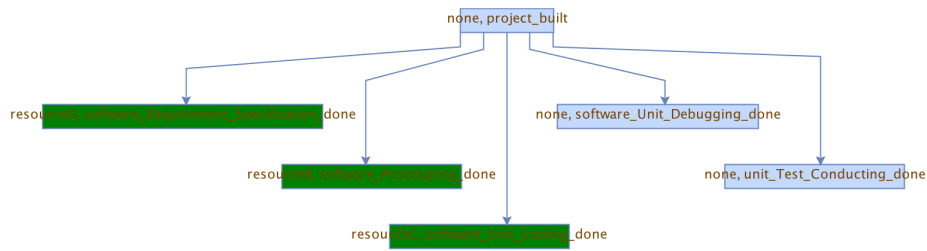


Fig. 3. WBS created by simulation.

agent simulation environment, and two human interface aspects (i. e., representation and visualization). Representation refers to the representation of the scope activities and the Work Breakdown Structure (WBS), and visualization refers to techniques that show how a visualization of the dynamics of activity flow sequences.

To evaluate our approach we have implemented the representations in JaCaMo, an agent-based framework, having extended it to provide the necessary visualization support, and, in terms of coverage, we have shown how it can be mapped to the SCRUM project management method and to the PMBOK Guide (Project Management Body of Knowledge) scope management process. The combination of PMBOK guidelines, the SCRUM development model, and a representation (using JaCaMo) and visualization platform provides our simulation-oriented approach, JaCaMoPM, with valuable features to support project managers throughout the software development process. Based on an exploratory study and our experience, we believe these results help to advance the research area involving the intersection of agents and project management. Especially in terms of modeling and simulation complex and dynamics under conditions pertaining agents, organizations and their environment.

References

1. Agarwal, R.: A Flexible Model for Multi-Ageent Based Simulation of Software Development Process. Ph.D. thesis, Auburn University (2007)
2. Agarwal, R., Umphress, D.: A flexible model for simulation of software development process. Proceedings of the 48th Annual Southeast Regional Conference (2010)
3. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with jacamo. *Science of Computer Programming* 78(6), 747–761 (Jun 2013)
4. Gaeta, M., Ritrovato, P.: Generalized environment for process management in cooperative software engineering. Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment pp. 1049–1053 (2002)
5. Highsmith, J.: *Agile Software Development Ecosystems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2002)

6. Jennings, N., Wooldridge, M.: Applications of intelligent agents. *Agent Technology: Foundations, Applications and Markets* pp. 3–28 (1998)
7. Joslin, N., Poole, W.: Agent-based simulations for software project planning. *Proceedings of the Winter Simulation Conference* (2005)
8. Kameas, A.: Towards the next-generation of ambient intelligent environments. *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)*, 2010 19th IEEE International Workshop on pp. 1–6 (2010)
9. Korb, W., Engel, D., Boesecke, R., Eggers, G.: Risk analysis for a reliable and safe surgical robot system. *International Congress Series* pp. 766–770 (2003)
10. Leung, H., Poon, C.: Multi-agent environment for software quality assurance, autoq. *EUROMICRO Conference, 1999. Proceedings. 25th* (1999)
11. Maurer, R.: Project coordination in design processes. *IEEE International 5th Workshop for Enabling Technologies (WET) : Infrastructure for Collaborative Enterprises* pp. 191–198 (1996)
12. de Medeiros Baia, D., Lucena, C.J.P., Cowan, D., Bommel, P., Valadares, C., Oliveira, T.C.: A multiagent-based simulation model to support management decision making in software development. *Tech. Rep. CS-2014-04*, University of Waterloo, David R. Cheriton School of Computer Science (2014)
13. Muller, J., Bauer, B., Friese, T.: Programming software agents as designing executable business processes: A model driven perspective. *Programming Multi-Agent Systems (LNAI) 28*, 49–71 (2004)
14. Nienaber, R., Barnard, A.: A generic agent technology framework to support the various software project management processes. *Proceedings of the Informing Science and Information Technology Conference (INSITE)* pp. 108–118 (2007)
15. Parunak, H., Baker, A., Clark, S.: The aaria agent architecture: An example of requirements-driven agent-based systems design. *Proceedings of the First International Conference on Autonomous Agents* pp. 482–483 (1997)
16. Persson, C., Picard, G., Ramparany, F.: A multi-agent organization for the governance of machine-to-machine systems. In: *Proceedings of the IEEE International Conference on Intelligent Agent Technology* (2011)
17. *PMBOK: A Guide To The Project Management Body Of Knowledge - PMBOK Guides - Fifth Edition*. Project Management Institute (2013)
18. Rigaud, E., Guarnieri, F.: Towards an agent oriented virtual organization dedicated to risk prevention in small and medium sized companies. *Proceedings of the 13th International Workshop on Database and Expert Systems (DEXA)* (2002)
19. Ruparelia, N.B.: Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes* 35, 8 (2010)
20. Sauer, J., Applerath, H.: Scheduling the supply chain in teams of agents. *Proceedings of the 36th Hawaii International Conference on System Sciences* (2003)
21. Szymczak, M., Frackowiak, G., Ganzha, M.: Resource management in an agent-based virtual organization. *Proceedings of the IEEE MaSeB Workshop* pp. 458–462 (2007)
22. Toledo, C., Bordini, R., Chiotti, O., Galli, M.: Developing a knowledge management multi-agent system using the jacamo platform. In: *Proceedings of ProMAS (AAMAS)* (2011)
23. Udo, N., Koppensteiner, S.: Will agile development change the way we manage software projects? agile from a pmbok® guide perspective. *Proc. PMI Global Congr., Project Manage. Inst* pp. 22–23 (2003)
24. Wu, C., Simmons, D.: Plan tracking knowledge base. *Proceedings of the 24th Annual International Computer Software and Applications Conference (COMPSAC)* pp. 299–304 (2000)