

5-5-2010

An Agent-based Model of Team Coordination and Performance

Jose A. Rojas-Villafane

Florida International University, jrojas@suagm.edu

DOI: 10.25148/etd.FI10081217

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>

Recommended Citation

Rojas-Villafane, Jose A., "An Agent-based Model of Team Coordination and Performance" (2010). *FIU Electronic Theses and Dissertations*. 250.

<https://digitalcommons.fiu.edu/etd/250>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

AN AGENT-BASED MODEL OF TEAM
COORDINATION AND PERFORMANCE

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

INDUSTRIAL AND SYSTEMS ENGINEERING

by

José A. Rojas-Villafañe

2010

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by José A. Rojas-Villafañe, and entitled An Agent-Based Model of Team Coordination and Performance, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Martha A. Centeno

José A. Faria

Marc L. Resnick

Ronald E. Giachetti, Major Professor

Date of Defense: May 5, 2010

The dissertation of José A. Rojas-Villafañe is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Interim Dean Kevin O'Shea
University Graduate School

Florida International University, 2010

DEDICATION

I dedicate this dissertation to my dearly loved family Maria, Ishaq, and Isabel. Thanks for your patience, support, and love. You were my inspiration during this journey.

ACKNOWLEDGMENTS

I wish to thank Antonio Feliciano, may he rest in peace, for the opportunities to start a career teaching engineering and to pursue my Ph.D. at FIU. A special acknowledge to the Universidad del Turabo, my fellow faculty members, and administrative personnel for your support during my doctoral studies. I also must acknowledge the Air Force Research Laboratory for partially funding this research through the grant “Simulation modeling and statistical network tools for improving collaboration in military logistics”.

Very special thanks to my Mother, Margarita, and my in-laws, Francisco and Josefina. I would not been able to complete my studies and my dissertation without knowing that you were there to support my wife while I was absent.

Thanks to the members of my committee for sharing your knowledge through your advice and for generously finding time in your agenda for me. Thanks to Dr. Martha Centeno for introducing me to the exciting field of simulation, and thanks for your dedication and commitment to your students. I hope to follow your example during my career. Thanks to Dr. Ronald Giachetti for your guidance, wise advice, and for your unlimited patience. Thanks for always impel me to strive for perfection. I know this dissertation is just the start of the road, but I will carry those lessons for the rest of the way.

ABSTRACT OF THE DISSERTATION
AN AGENT-BASED MODEL OF TEAM COORDINATION
AND PERFORMANCE

by

José A. Rojas-Villafaña

Florida International University, 2010

Miami, Florida

Professor Ronald E. Giachetti, Major Professor

This research is based on the premises that teams can be designed to optimize its performance, and appropriate team coordination is a significant factor to team outcome performance. Contingency theory argues that the effectiveness of a team depends on the right fit of the team design factors to the particular job at hand. Therefore, organizations need computational tools capable of predict the performance of different configurations of teams.

This research created an agent-based model of teams called the Team Coordination Model (TCM). The TCM estimates the coordination load and performance of a team, based on its composition, coordination mechanisms, and job's structural characteristics. The TCM can be used to determine the team's design characteristics that most likely lead the team to achieve optimal performance.

The TCM is implemented as an agent-based discrete-event simulation application built using JAVA and Cybele Pro agent architecture. The model implements the effect of individual team design factors on team processes, but the resulting performance emerges

from the behavior of the agents. These team member agents use decision making, and explicit and implicit mechanisms to coordinate the job. The model validation included the comparison of the TCM's results with statistics from a real team and with the results predicted by the team performance literature.

An illustrative 2^{6-1} fractional factorial experimental design demonstrates the application of the simulation model to the design of a team. The results from the ANOVA analysis have been used to recommend the combination of levels of the experimental factors that optimize the completion time for a team that runs sailboats races.

This research main contribution to the team modeling literature is a model capable of simulating teams working on complex job environments. The TCM implements a stochastic job structure model capable of capturing some of the complexity not capture by current models. In a stochastic job structure, the tasks required to complete the job change during the team execution of the job. This research proposed three new types of dependencies between tasks required to model a job as a stochastic structure. These dependencies are conditional sequential, single-conditional sequential, and the merge dependencies.

TABLE OF CONTENTS

| CHAPTER | PAGE |
|---|------|
| 1 INTRODUCTION | 1 |
| 1.1 Research Background | 3 |
| 1.2 Research Problem | 9 |
| 1.3 Goal and Objectives | 19 |
| 1.4 Research Methodology | 20 |
| 1.5 Organization of Dissertation | 25 |
| 2 LITERATURE REVIEW | 26 |
| 2.1 Team Effectiveness and Performance | 26 |
| 2.2 Factors Affecting Team Performance | 31 |
| 2.3 Frameworks and Models to Study Coordination | 38 |
| 2.4 Team Simulation | 42 |
| 2.5 CybelePro Agent Infrastructure | 46 |
| 3 TEAM COORDINATION MODEL | 50 |
| 3.1 Conceptual Model | 50 |
| 3.2 Team System | 60 |
| 3.3 Team Member Model | 75 |
| 3.4 Team Coordination Model Implementation | 82 |
| 4 MODEL VALIDATION | 91 |
| 4.1 Test Scenarios | 91 |
| 4.2 Model Verification and Validation Strategy | 97 |
| 4.3 Data Analysis and Model Verification | 100 |
| 4.4 Model Validation | 112 |
| 4.5 Verification and Validation Conclusion | 120 |
| 5 EXPERIMENTATION | 123 |
| 5.1 Design of Experiments | 123 |
| 5.2 Experimental Results | 125 |
| 5.3 Team Optimization | 135 |
| 6 CONCLUSIONS | 140 |
| 6.1 Summary of Research | 140 |
| 6.2 Research Contributions | 142 |
| 6.3 Limitations and Future Work | 144 |
| REFERENCES | 149 |
| VITA | 156 |

LIST OF TABLES

| TABLES | PAGE |
|---|------|
| Table 1: SDL symbols used to model a project structure | 23 |
| Table 2: Summary of Factors Affecting Team Performance | 32 |
| Table 3: Coordination Mechanisms in the Model..... | 53 |
| Table 4: Job Structure Factors | 56 |
| Table 5: Team Composition & Team Structure Factors..... | 57 |
| Table 6: Centralization Levels Definition..... | 58 |
| Table 7: Formalization Levels Definition..... | 59 |
| Table 8: Type of Interdependencies included in the model | 63 |
| Table 9: Attributes of a Dependency | 64 |
| Table 10: Task Execution States..... | 65 |
| Table 11: Attributes that Define a Task..... | 67 |
| Table 12: Comparison between Asynchronous and Synchronous Media Types..... | 69 |
| Table 13: Summary of Model Validation & Verification..... | 98 |
| Table 14: Results of the Job Structure Verification..... | 102 |
| Table 15: Functions and Features Checked During Model Verification | 104 |
| Table 16: Problems detected during the verification of the model..... | 107 |
| Table 17: Problems detected during the verification of the model (Continuation) | 108 |
| Table 18: Results for the Internal Validity test..... | 109 |
| Table 19: Results for the Internal Validity test..... | 109 |
| Table 20: Results from Sensitivity Analysis to Variations in Task Durations | 111 |
| Table 21: Comparison with Actual System Results | 113 |

| | |
|--|-----|
| Table 22: Comparison with Actual System Results with adjusted time units | 115 |
| Table 23: Parameters Tested for Expected Effects | 116 |
| Table 24: Operationalization of Experimental Factors | 124 |
| Table 25: Results for Each Experimental Treatment | 126 |
| Table 26: Recommended Values for the Individual Experimental Factors | 136 |
| Table 27: Optimization of the Completion Time | 138 |

LIST OF FIGURES

| FIGURE | PAGE |
|---|------|
| Figure 1: Relationship between Team Performance and Effectiveness..... | 6 |
| Figure 2: General Formulation for the Team Design Problem..... | 11 |
| Figure 3: CODE Framework..... | 40 |
| Figure 4: An Integrated Framework of Team Coordination and Performance..... | 40 |
| Figure 5: Andres and Zmud's Research Model | 41 |
| Figure 6: Cybele Agent Framework for Agent Applications..... | 47 |
| Figure 7: CybelePro Agent Application Model..... | 48 |
| Figure 8: Conceptual Team Coordination Model..... | 51 |
| Figure 9: Representation of the System Model..... | 61 |
| Figure 10: State Diagram for Task Execution | 66 |
| Figure 11: Taxonomy of Communications Events..... | 69 |
| Figure 12: Implementation of Reciprocal Dependencies..... | 71 |
| Figure 13: Negotiation Process between Teammates..... | 72 |
| Figure 14: Team Coordination Model Architecture | 83 |
| Figure 15: Relation between Data Classes and Simulation Agents..... | 84 |
| Figure 16: Composition of the Job Environment Agent..... | 86 |
| Figure 17: Composition of the team member agent..... | 87 |
| Figure 18: Flow of Task Processing Events | 88 |
| Figure 19: Campus Hurricane Preparedness Job Network | 94 |
| Figure 20: Job Structure for the Race Committee Scenario | 97 |
| Figure 21: Simplified Network of Race Committee Job Structure..... | 103 |

| | |
|---|-----|
| Figure 22: Time Table for the Leader Agent of the Race Committee Scenario | 106 |
| Figure 24: Results of Coordination Load vs. Centralization/Formalization..... | 118 |
| Figure 23: Completion Time vs Centralization/Formalization..... | 118 |
| Figure 25: Pareto of Effects for Completion Time | 127 |
| Figure 26: Interaction Plots for Tap vs. Experience | 128 |
| Figure 27: Interaction Plot for Team Size vs. Centralization | 129 |
| Figure 28: Interaction Plot for Team Size vs. Experience | 130 |
| Figure 29: Cube Plot for the ABD Interaction..... | 131 |
| Figure 30: Pareto of Effects for the Coordination Load | 131 |
| Figure 31: Interaction Plot for Team Size vs. Centralization | 132 |
| Figure 32: Interaction between TAP, Centralization, and Formalization..... | 133 |
| Figure 33: Pareto of Effects for the Rework Time | 134 |
| Figure 34: Interaction Plot for Team size vs. Centralization | 134 |
| Figure 35: Conceptual Team Coordination Model | 141 |

CHAPTER 1:

INTRODUCTION

Teamwork is a key element in the functioning of most every organization. A primary reason for the use of teams by organizations is that the solution of complex problems often requires gathering multi-disciplinary expertise, which necessitates forming a team within which each member brings their expertise and perspective. For example, teamwork is essential for any business process improvement effort, and it is a backbone of the main process improvement methodologies used today such as “lean six sigma” (George 2002) and business process reengineering (Manganelli et al. 1996). In production organizations, the complexity of some products and production processes often required the utilization of teams of production workers and design engineers (Doolen et al. 2003). Effective teamwork is essential in healthcare organizations to provide better service and on occasions to save lives (Burke et al. 2004). In addition to traditional teams, virtual teams are gaining relevance in the business world since organizations are expanding their operations geographically and advances in telecommunications allows people to collaborate from distant locations (Wong et al. 2000).

Given the importance of teams to organizations, a common question is how can teams be designed so that they perform better? The research on teams suggests that organizations can design teams for high performance based on the selection of team composition, job structure, and coordination strategies (Espinosa et al. 2004; Stewart 2006). Team composition is decided before, and usually remains unchanged during, the

execution of the job. Job structure is usually dictated by the nature of the work and agreed upon before the job execution. Coordination is the factor that affects team performance during job execution and should have a key role in the study and design of teams.

This research postulates that the better teams coordinate, the better they will perform. Coordination is considered the essence of teamwork (Brannick et al. 1995; Marks 2000), and teamwork is what allows teams to adapt their strategies during the execution of a job to optimize their performance (Paris et al. 2000). Hence, team's performance measures should focus on coordination requirements.

In this research, an agent-based simulation model is created based on contingency theory so that teams can be modeled and designed for optimal performance. Agent-based simulation provides a close fidelity to how teams perform. Contingency theory is from the organizational sciences and states that the optimal organizational structure depends on contextual factors. In the case of teams, we restate the theory to say that the optimal team design depends on contextual factors. Using the simulation, an analyst could evaluate different team designs and determine a team design that leads to optimal performance under a given scenario.

The remainder of this chapter provides an introduction to this research. First, a brief background on team performance and coordination is provided, followed by a description of the problems addressed by this research, its goals and objectives, and its methodology.

1.1 Research Background

Teams are an alternative to perform a task that cannot be completed effectively by a single individual or by the aggregated independent efforts of a group of individuals (Marks 2000). Brannick and Prince (1997) define a *team* as a group of people “with different tasks who work together adaptively to achieve specified and shared goals.” What differentiates a team from simply a group of individuals is that in a team there is interdependence between team members’ tasks. Managing this interdependence requires teamwork or in other words coordination. Additionally, a common goal and a shared mental model are necessary conditions for the existence of a team (Beavers et al. 2001). A shared mental model is a conceptual construct that implies that each team member is aware of being in a team, believes that he can contribute to the achievement of the team goal, and believes that other team members also intend to pursue the same goal. A shared mental model is part of the self-identify a person possesses with respect to a team.

Cohen and Bailey (1997) identified four types of teams: work teams, parallel teams, project teams, and management teams. Work teams are responsible for producing goods or providing services, and they work on an ongoing basis. Parallel teams are constituted by members from different units of the organization to perform a function or task, usually a task such as problem solving or system improvement, which is not assigned to any of the regular organizational units. Project teams are formed on a temporary basis to produce a one-time output such as a new product, service, or project. Management teams are created to laterally integrate independent units of the same organization, by coordinating their respective functions according to a single strategic direction.

One of the main motivations for this research is a void in the team modeling literature of models that have the capabilities to represent teams from all four of the classifications described in the previous paragraph. Most of the research literature on team modeling focuses on work or project teams that have a specific number of tasks to perform and that have the dependencies between those tasks well-defined. However, many teams have to perform jobs that do not have those characteristics. For example, a team directing the emergency preparedness efforts for a city might have to decide between opening shelters in one neighborhood or evacuate it completely. These courses of actions, which depend on how the emergency unfolds, require very different tasks, resources, and different set of skills.

1.1.1 Team Performance Background

Starting largely in the 1980's, the growing relevance and importance of teams to organizations has encouraged researchers in many diverse fields to pay attention to team performance (Baker et al. 1997), and to the characteristics and processes that contribute to superior team performance (Marks 2000). For example, recent researches draw attention to the importance of team performance measures in fields such training (Burke et al. 2004) and risk assessment of complex engineering systems (Kim *et al.*, 2006).

Frequently, in the literature concerning teams, the terms team performance and team effectiveness are used interchangeably. Since this research only addresses team performance, it is important to establish the distinction between the two terms. Team effectiveness is defined as the degree to which a team achieves its goals (Daft 1995;

Kraiger et al. 1997); therefore, team effectiveness is “highly contextual” and should be evaluated under the context of other organizational factors (Pagell et al. 2002).

Team performance, on the other hand depends on the efforts and skills of team members. Team performance includes process and outcomes performance that can be measured at both the individual and team level (Cannon-Bowers et al. 1997; Paris et al. 2000). Process performance focuses on how the task was accomplished, and might include analysis of coordination strategies, team communication, information flow, leadership, error correction, among others (Paris et al. 2000). Team outcome performance refers to the quantity and quality of the team’s outcomes (Cohen et al. 1997), and it is commonly measured along the dimensions of time, cost, and quality (Kim et al. 2003; Piccoli et al. 2004; Mathieu et al. 2006).

Figure 1 illustrates the relationship between team performance and team effectiveness based on the model proposed by Doolen et al (2003). Team performance is divided in process or teamwork performance, hereby known as teamwork performance, and in outcome or task performance, hereby known as outcome performance. This separation between process and outcome performance is consistent with the *Input-Process-Output* model that has guided team research for years (Mathieu et al. 2006), and promotes the development of the proper set of skills required for each (Morgan et al. 1993). Teamwork and outcome performance complement one another, and jointly provide a better picture of team performance (Paris et al. 2000). Team effectiveness is a result of both, teamwork and outcome performance, in the context of organizational and environmental factors.

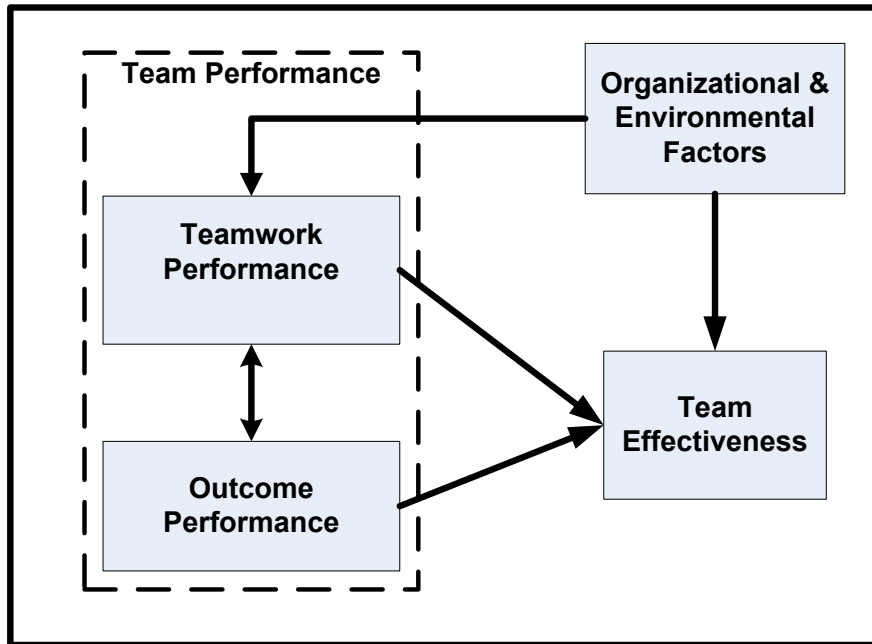


Figure 1: Relationship between Team Performance and Effectiveness

In summary, measures of effectiveness consider the impact of the team’s output on the organization, while team performance focus only on the quality of team’s output and the quality of the process to generate the output. To illustrate the difference between team performance and team effectiveness, consider a team of engineering students from a small university that enters a design contest. Suppose they submit a design that exceeds all the minimum requirements of the contest by at least 50%, and their professors put in high regard their creativity and work done given the limited resources available. The morale of the team was high throughout the process. In spite of this, the team ends up fourth in the competition. The performance of this team probably could be rated as outstanding in every category, but since they end up short on the main goal of winning the competition, their effectiveness should be rated lower. The explanation is that other

intervening factors, in this case, external environmental factors also influence team effectiveness.

This research focuses on teamwork performance because it is directly within the control of the team design, team leaders, and team members. Team effectiveness depends on factors outside the control of the team members or its managers and is consequently uncontrollable, although teams can be designed to be robust with respect to environmental conditions. However, it is noted that improving teamwork performance does improve team effectiveness.

1.1.2 Team Coordination Background

Understanding what contributes to team performance is a requirement in order to design, train, manage, and evaluate teams. A main postulate of this research is that team coordination is a key determinant of team performance. Coordination according to Malone et al. (2001) is “managing dependencies between activities”. A dependency means that the completion of one activity is subject to the completion of another. Team coordination focuses on coordinating tasks within the group, and dependencies may be in the form of information, materials, or reciprocal inputs (Stewart 2006). These dependencies might require team members to adjust to each other tasks either simultaneously, sequentially, or both (Brannick et al. 1997). For example, a team member may need some information or material product of another team member’s work in order to start or finish his own work.

According to Espinosa et al. (2004), coordination can be interpreted as both a process and an outcome. Coordination as a process involves the activities performed by a team to manage dependencies. Coordination as an outcome “can be defined as the extent to which dependencies have been effectively managed.” This research takes on the process view of coordination.

Team coordination occurs in the context of four dimensions: goals, resources, information, and tasks (Wang et al. 2001). Goal coordination involves the selection and decomposition of the team goal, and the development of strategies to achieve it. Resource coordination involves the allocation of scarce resources among the decision makers and activities. Information coordination involves determining and disseminating the information requirements of the different activities. Task coordination is the act of scheduling or planning interrelated activities required to complete the team task.

This research assumes that goal coordination occurs before a team starts working on a job. A team, according to its definition, is created with a purpose or goal in mind. Therefore when a team starts working on a new job, in most cases, the members already have agreed on the goal.

Teams use a blend of implicit and explicit coordination (Wang et al. 2001). Implicit coordination occurs when a team member takes actions or decisions that affect the team, or other team members’ activities, based on situational information. Implicit coordination depends on team members having precise mental models about the team’s goals, current situation and other members’ needs. Explicit coordination occurs through communication between team members. Communication provides team members with explicit information about other members’ needs and team’s current situation.

1.2 Research Problem

This research addresses the problem of how to model teams so that they can be designed for optimal performance. Within the scope of this problem, one particular void in the literature is that current modeling approaches usually assume well-defined deterministic job structures. In this research, we find that oftentimes job structures are stochastic in that the precise sequence of actions taken by the team cannot be predicted because it depends on various other factors that only unfold as the team works together on the job. Consequently, an important problem is how to model the stochastic job structure and incorporate it into an agent-based model. The following subsections elaborate further on the research problem.

1.2.1 Team Design Problem

Organizational design has been concerned with searching for coherence or a fit between tasks, strategies, and individuals. The goal of organizational theory is to provide a rational decision process to choose an organizational structure that improves the effectiveness of that organization (Galbraith 1977). There is evidence that the performance of an organization changes when its design is changed (Carley 1995). The principles behind organizational theory have evolved and have been adopted by other disciplines such as enterprise engineering. Enterprise engineering focuses on how to model, analyze, and design enterprise systems in which the organization view is one of many possible ways to view the enterprise (Giachetti 2004).

Teams are in essence organizations and can be designed as well. Stewart (2006) concludes that teams can be designed for high performance and design factors exist for the design of team composition, task structure, and organization context. Salas et al. (2005) provide an extensive list of factors required to model and design teams.

Based on the literature, this research formulates the team design problem in a very general form. The formulation presumes that the team has an objective to either minimize the completion time of the team task or to maximize the quality of the results. The decision makers have to decide the values of some factors under their influence (decision variables) while satisfying some factors they cannot change (constraints). Figure 2 shows a sample formulation for the problem, which is the one that applies to the cases and examples used by this research.

A common objective of team design is to optimize team performance given a job structure and specific job requirements. Team performance can mean different things in different scenarios. For example, a team might have the objective of performing the job assigned as soon as possible (minimizing completion time), and/or complete the job while minimizing the number of mistakes in the tasks (maximizing quality of results).

As previously defined, a job structure is composed of n tasks required to complete the job and q dependencies between them. The job requires a set of s skills to perform the tasks appropriately. Each of the tasks may or may not require one or more of the skills, and each task has a level of complexity associated with it.

Organization characteristics, team composition, and coordination mechanisms are likely to be in the control of the decision makers, therefore are the decision variables of the problem. Organization characteristics influence the way that teams operates, and the

team inherits these characteristics from its organization. Common organization characteristics are centralization and formalization. Team composition variables include team size, each team members' skills and experience. Coordination mechanisms can be implicit, such as task assignment plan, and/or explicit such as communication activities. The decision variables listed are either integer or binary variables.

| |
|---|
| <p>Objective:</p> <p>Optimize Team Performance (Min <i>Completion Time</i>, Max <i>Quality of Results</i>)</p> <p>Subject to:</p> <p><i>Job Structure</i> = (<i>Task_j</i>, <i>Dependency_k</i>) $j = 1, 2, \dots, n$ $k = 1, 2, \dots, q$</p> <p><i>Job Requirements</i>:</p> <p>$Skills_{ja} \in \{0, 1\} \quad \forall j = 1, 2, \dots, n \quad \text{and} \quad \forall a = 1, 2, \dots, s$</p> <p>$Task\ Complexity_j \in \{low, med, high\} \quad \forall j = 1, 2, \dots, n$</p> <p>Variables:</p> <p>Organization Characteristics:</p> <p>$Centralization = \{low, med, high\}; \quad Formalization = \{low, med, high\}$</p> <p><i>Team Composition</i>:</p> <p>$Team\ Size > 1,$</p> <p>$Members\ Skills_{ia} \in \{low, med, high\} \quad \forall i = 1, 2, \dots, Team\ Size \quad \text{and} \quad \forall a = 1, 2, \dots, s$</p> <p>$Experience_i \in \{low, med, high\} \quad \forall i = 1, 2, \dots, Team\ Size$</p> <p>Coordination Mechanisms(CM):</p> <p>$Implicit\ CM_l, \quad Explicit\ CM_m$ are binary</p> |
|---|

Figure 2: General Formulation for the Team Design Problem

The formulation can be changed such that what is a decision variable in one instance of the optimization model becomes a constraint in another instance of the optimization model. For example, a team design problem could be to determine how to

divide a job into tasks (job structure) given certain team composition and organizational characteristics.

Furthermore, in some instance of the problem the objectives could become constraints and the variables can become objectives. For example, a team design problem might have the objective of minimize the team size, while keeping the job completion time under some value (due date).

1.2.2 Limitations on Traditional Research Methods for Teams

The design factors have been defined and understood through traditional empirical research methods using human subjects such as case studies and controlled laboratory experiments. These empirical research methods that use human subjects have some limitations that prevent their application to design teams and to further advanced the research on team performance. For example, Wang et al. (2001) argue that contradictions found in the team coordination literature can be explained in part by the complexity of human behavior and the lack of encompassing theories about team performance. However, these complexities are too difficult to capture by either field studies or controlled experiments with human subjects, therefore, limiting the development of comprehensive theories relating team coordination and performance.

One limitation of both, field studies and controlled experiments, is that large sample sizes are difficult to obtain due to difficulties in designing, managing and conducting such experiments. Sample sizes limit the model complexity, range of factors, and number of scenarios that can be tested (Ancona et al. 1992; Wang et al. 2001; Piccoli

et al. 2004; Nuñez 2006). These limitations makes that good controlled experiments frequently confront the trade-off of face validity for tractability. If researches want the execution of their experiments to be manageable, often has to do it by limiting the scenarios and/or variables included in the experiment. This trade-off leads to serious limitations in the researcher's ability to generalize or extend the results to other teams types or working environments (Ancona et al. 1992; Mathieu et al. 2006), and make it difficult to use the results for prescribing actions (Pagell et al. 2002).

Finally, traditional research methods frequently rely on subjective methods, such as interviews and surveys, to collect data which increase the variation and limit the reliability of data collected. For example, Doolen et al. (2003) confront the limitation that the data collection method might have reduced the discrimination between factors due to the use of the same measurement instrument for all of the performance measures.

Coordination should be one of the main focuses of team design efforts. Coordination is often considered the essence of teamwork, and teamwork is what allows teams to achieved their goals (Brannick et al. 1995; Marks 2000). Nevertheless, results are somewhat contradictory about the impact of coordination on team performance. For example, Fussell et al. (1998) found that team performance increases according to the degree teams reported to being better coordinated. On the other hand, Piccoli et al. (2004) found no empirical support to the hypothesis that team output performance improves if coordination effectiveness improves.

Despite the many contradictory findings relating coordination and team performance, there is a general consensus that the better teams coordinate, the better they will perform. Measures of teamwork performance focus on the requirements to

coordinate a team, and teamwork is what allows teams to adapt and optimize their performance (Paris et al. 2000). Therefore, coordination and teamwork are aspects of a team that need to be carefully considered when assembling or designing a team for a task, or when management wants to improve performance for some existing teams in an organization.

To further the study of team coordination and team performance, experimental methods or tools should be developed to overcome the limitations of case studies, survey assessments, or laboratory studies to analyze a large number of factors, at large range of levels, for different types of teams. Research methods, traditionally employed by organizational scientists, have produced a large amount of knowledge and empirical evidence about the relationships between the different structural, behavioral and contextual factors affecting team performance as exemplified by (Cohen et al. 1997; Stewart 2006). This body of knowledge could be implemented using other research methods to test comprehensive theories related to team performance.

1.2.3 Computational Models and Contingency Theory

One alternative to develop and test theories on team coordination and performance is through computational models. As discussed and exemplified by Burton et al. (1995), computational models have been used for decades to test hypothesis, to explore organizational processes, and to study theoretical and practical issues. Computational models have the potential “to move theories of organization beyond empirical description to generative formalizations” (Carley 1995). However, up to

fifteen years ago, there was limited use of computers as tool in the design of social aspects of organizations (Jin et al. 1995). Wang et al. (2001) asserted that the study of teams' coordination issues will be benefited by combining various discipline perspectives through modeling and empirically studies.

Another imperative application of computational tools is to design organizations and teams. Carley (1995) pointed that organizational design is one the most relevant issues within an organization because changing its design allows organizations to adapt to its task environment and alter its performance. This author also pointed that computational models are particularly useful to evaluate organizational design alternatives. This assertion can be applied to teams considering that teams are a form of organization.

The need for a computational tool to test different team design alternatives is crucial when contingency theory is considered. Contingency theory proposes that the effectiveness of an organization depends on the “fit” between some organizational characteristics, known as contingency factors, and the external conditions affecting the organization (Howell et al. 2010). Different external conditions require different values of the organizational characteristics. The more traditional congruency models propose unconditional association among the variables in a model, while contingency models propose a conditional association of two or more variables with a dependent outcome (Umanath 2003).

Contingency theorists argued that a simple theory or general design guidelines cannot exist, because the right design choice depends on the particularly situation or task environment of the team (Carley 1995). Contingency theory implies that an analyst or

manager will have to identify the best team characteristics for the particular job at hand. Therefore, it can be easily argued that computational models will be more efficient than other research methods to evaluate different design alternatives for teams. For example, it will be very costly and time consuming to test team design configuration using real individuals every time a team needs to be assembled.

1.2.4 Simulation Models

Computer simulation is the most appropriate computational tool for team design, particularly under the basic assumption contained by Contingency Theory. Simulation models provide many advantages as a computational tool to conduct studies about teams and organizations. A simulation model allows the implementation of social sciences theory, and it allows a precise and testable representation of conceptual entities with their functions, structure, and behaviors (Kunz et al. 1998). Researchers using simulation models can measure and test with precision some variables while controlling other factors, and can test a wide range of organizational variables (Kim et al. 2003). These characteristics allow simulation-based studies to overcome the limitations of controlled laboratory studies with human subjects. Furthermore, for highly complex systems, such as teams, computer simulation is the only viable method for system analysis and evaluation (Cheng 1992).

Simulation models have been used previously to study organizations and teams. One of the most prominent examples is the work of the Virtual Design Team (VDT) at Stanford University that developed a computational discrete simulation system to model

organizations at the micro or individual member level (Kunz et al. 1998). The VDT system has been used mainly to simulate project teams such as product development or construction design teams (Christiansen et al. 1997; Kunz et al. 1998; Christiansen et al. 1999; Levitt et al. 1999). The VDT software also has been used to study the relation between various team characteristics and team performance (Wong et al. 2000; Kim et al. 2003). It would be appropriate to note that the VDT software was design for project management, and most of its applications have been to study project teams. It may have limitations to represent the behavior of other type of teams such as parallel teams and management teams were the task structure and interdependencies cannot be well defined. The VDT simulation application also has limitations on the representation of the shared mental model, since the only implicit coordination mechanisms provided is a network that represents the perception of each member about the skill level of other members.

Other examples of organizational modeling are found in the agent-based simulation literature (Loper et al. 2005; Yen et al. 2006). Loper and Presnell (2005) used an agent-based simulation model of the Georgia Emergency Management Agency to evaluate the performance of the agency at the individual and at the aggregate level. Yen et al. (2006) developed an agent-based architecture to model team processes, including components necessary to represent teamwork like share mental model, and goal management. These two research efforts provide good examples on how to model teams using agent-based simulation. Nevertheless, the number of factors involved in these studies is limited, neither of them provide for the study of coordination requirements of teams.

The use of organizational simulation to model team performance is becoming “increasingly important”, but the field is still in its infancy (Salas et al. 2005). Salas et al. (2005) points out that there is a need for better simulation tools that capture the complexities of team performance. These tools should be “fluid, flexible, and adaptable” and should capture realistic team performance.

This research identifies that simulation-based tools allow researchers and team analysts to model jobs with stochastic structures. A stochastic job structure represents the case when the execution of one or more tasks depends on the results from predecessor tasks. The result from the predecessor task can be model as a stochastic event that with a probability p generates a task, and with probability $1 - p$ generates other task or no new task.

Teamwork simulation models found in the literature use a rigid structure to model the job to be performed by the team (Christiansen et al. 1997; Kunz et al. 1998; Yen et al. 2006). A rigid structure means that teams always perform the same tasks to complete a specific job, and the dependencies between the tasks are always enforced. Nonetheless, some teams work in complex environments that often include elements of uncertainty and surprise (Altman-Klein et al. 2005) that changes the tasks to be performed by the team (Salas et al. 2005). Being able to model and simulate dynamic and stochastic job structures makes the simulation tool more flexible and adaptable to the environment of most teams than just being able to model rigid job structures.

1.3 Goal and Objectives

The goal of this research is to create an agent-based model of teams performing jobs with a stochastic structure, and to develop a methodology to apply the model to design teams. The resulting agent-based simulation model is called the *Team Coordination Model* (TCM). The TCM estimates the coordination load and performance of a team based on its composition, coordination mechanisms, and job's structural characteristics.

The methodology proposed by this research is called *Contingency Team Design Methodology* (CTDM). The CTDM consists on the following steps:

- Gather data about the team and the job.
- Develop the structure model for the job including tasks and dependencies, and input the scenario into the TCM simulation application.
- Validate input data and preliminary results from the simulation.
- Run a factorial design of experiment to determine what team composition characteristics, coordination mechanisms, and task structure will lead a team to achieve optimal performance and coordination efficiency. The experimental factors should be limited to the factors under the control of the analyst or team manager.

In order to achieve the goal, the research pursues the following objectives:

- Conceptualize the *Team Coordination Model*. The independent and dependent factors to be included in the model and the relationships among them are identified according to the research literature in team

performance and coordination. An operational definition is developed for each factor.

- Develop the Agent-based Simulation Model. The research implements the Team Coordination Model as an agent-based, discrete-event simulation application.
- Build test scenarios to verify, validate, and experiment with the model. An actual team scenario, based on a Sailboat Race Committee, is employed to test the Team Coordination Model for external validity and used it as a based scenario for the experimental study.
- Verify and Validate the Team Coordination Model. The simulation model is subjected to various verification and validation techniques to assess the suitability of the model for its intended purpose.
- Application of the Contingency Team Design Methodology. Once validated, the simulation model is used to determine the team design that optimizes performance for the scenarios in the case study. This study exemplified the application of the Team Coordination Model for practical purposes.

1.4 Research Methodology

Agent-based modeling and simulation (ABMS) captures the behavior of complex systems from a bottom-up approach (North et al. 2007). The ABMS process starts with the conceptualization of the behavioral rules of individual components that are

later used to create the agent-based model. The model is executed to emulate and analyze the behavior of the whole system. The ABMS's approach to modeling systems allows making connections between the behavior of individual components and the emergent system-level behavior. According to North and Macal (2007), organizations that develop agent-based models are able to determine which combination of individual level actions will yield better results. This is the challenge at hand when designing a team.

The ABMS methodology described by North and Macal (2007) is the based for the development of the Team Coordination Model and for the Contingency Team Design methodology proposed by this research. The remaining of this section provides more detailed about the steps and tools employed to conduct this research.

1.4.1 Conceptual Model and its Implementation

The conceptualization of the Team Coordination Model is based on contingency relationships between team design factors and performance. The factors included in the Team Coordination Model are the ones that the literature has identified as having the greatest influence on team coordination and outcome performance. Extensive literature exists about the relationships between team characteristics and performance (Cohen et al. 1997; Stewart 2006) and about factors affecting coordination in teams and groups (Fiore et al. 2003; Espinosa et al. 2004).

An agent-based, discrete-event simulation application implements the conceptual Team Coordination Model. An agent-based model represents a system as a collection of autonomous decision-making entities called agents, and assumes that social structures

and organizations are created by the interaction of individuals (Kiel 2005). The benefits of agent-based simulation are its ability to capture emergent behavior of a system from the interactions between its agents or components, provides a natural description of a system, and it is a very flexible in terms of augmenting the complexity or size of the model (Bonabeau 2002).

The TCM simulation application allows quantifying team coordination and performance based on tasks characteristics, team structure and composition, and coordination mechanisms. The simulation model quantifies the individual effect of each factor on the coordination performance and output performance. The contingency effects on team performance emerge from the behavior of the agent-based simulation.


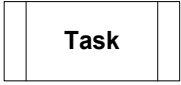

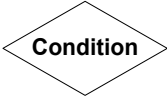


1.4.2 Simulation Tool Development

CybelePro is an agent infrastructure developed by Intelligent Automation Inc (IAI) that runs on top of a Java platform. CybelePro provides the classes and services to create and execute the agent-based, discrete-event simulation model. The graphical user interface of the application is built exclusively using JAVA classes.

The development of an agent-based model requires an incremental strategy (North et al. 2007), the Team Coordination Model was not an exception. Several phases were required to build the application, each one added functionalities and complexity to the model. The model was verified after every of these phases. Debugging was the main verification technique applied to each development phases.

Various hypothetical scenarios and one real team scenario are used to verify and validate the Team Coordination Model. The scenarios vary in complexity and duration of the activities. These scenarios are discussed in more details in Chapter 4. Specification and Description Language (SDL) was selected to model the job structures and to model the processes built in the simulation. SDL is an object-oriented formal language intended for the specification of complex event-driven applications involving many different activities that communicate using discrete signals (IEC 2007). Processes in an agent-based, discrete-event simulation model fits well the intended application of the SDL. Table 1 shows the SDL symbols used in this research.

Table 1: SDL symbols used to model a project structure

| Element | Symbol | Description |
|-----------|---|--|
| Start |  | Indicates the start of a project |
| Task |  | Represents a task to be perform by one agent |
| Document |  | Indicates an information resulting from a task that it is pass to another task |
| Condition |  | Indicates a stochastic event in the project representing a decision by an agent or a variable result of a task |
| Connector |  | Merges the result of two or more tasks when only one of them is needed to start performing the next task (logical OR). |
| End |  | Indicates the end of a project |

The intent of the validation process is to demonstrate that the model captures the dynamics of a team, and that the model is capable of differentiating, with enough accuracy, the influence that different design configurations have on the team performance and coordination. This research suffers one important limitation on the validation process, the ability to experiment with an actual system to compare results from various systems' scenarios with the corresponding results produced by the model. The TCM validation includes only a comparison of the model results with the data from just one scenario of the real system.

1.4.3 Application of the Team Coordination Model

A fractional factorial design of experiment is performed, using an actual team scenario, to complete the application of the Contingency Team Design methodology proposed by this research. The experiment has three objectives:

1. Identify significant interactions between design factors and compare them with the ones already identified from the literature.
2. Determine the level of the design factors that optimize team performance.

The first objective serves to demonstrate contingency relationships present in the simulation model. The second objective demonstrates the application of the CTDM to the optimization of a real team.

1.5 Organization of Dissertation

The documentation of this research effort is distributed in the following five remaining chapters. Chapter 2 provides a review of literature on team performance, team coordination frameworks, and simulation of teams. Chapter 3 describes the Team Coordination Model, and chapter 4 documents the verification and validation of the model. Chapter 5 provides an example of the application of this research, and chapter 6 summarizes and concludes this research effort.

CHAPTER 2:

LITERATURE REVIEW

Research literature related to teams is very extensive; it includes studies made by social scientists, business researchers, engineers, computer scientists, as well as multidisciplinary approaches. This chapter reviews the literature on teams related to this research. The first section discusses definitions and measuring approaches for team effectiveness and performance. The second section shows the factors that influence team performance. The third section discusses the theoretical frameworks to study team coordination, and the last section examines the literature on team simulation.

2.1 Team Effectiveness and Performance

A requisite to develop theories or techniques on how to better form or train teams is to define and measure what makes a team successful or effective. Team performance is definitely a key factor influencing team effectiveness. However, there is a difference between team success or effectiveness and team performance, since there are factors that affect effectiveness that are outside of the control of the team or its supervisors. The literature on teams is still fuzzy on this difference between team effectiveness and performance, and even sometimes these terms are used interchangeably. This section review definitions and measures of team effectiveness and team performance found on the literature, and the relationship between the two.

Daft (1995) defined organizational effectiveness as “*the degree to which an organization realizes its goals*”. This definition could be applied to every organizational

unit, including teams. This definition is also supported by Kraiger and Wenzel (1997) and by Hexmoor and Beavers (2002). Kraiger and Wenzel (1997) state a clear difference between team effectiveness and performance. These authors defined effectiveness to be an indicator of how well a team accomplishes its tasks or objectives, or the quality of the team outcome. In the other hand, they defined performance as an indicator of the quality of the process carried out to complete the task or product. However, Brannink and Prince (1997) asserted that a comprehensive assessment of team performance should measure elements of both process and outcomes.

2.1.1 Measuring Effectiveness

Measuring effectiveness in organizations is as complicated as the organization itself and the number of goals it pursues. Daft (1995) asserts that effectiveness is a multidimensional concept and can not be measure with a single indicator. This section explores different effectiveness measures present on team literature.

The multidimensionality of team effectiveness measurement is present in the work of different authors. For example, Cohen and Bailey (1997) categorized team effectiveness according to the team's impact on three major dimensions: quantity and quality of outputs, member's attitudes, and behavioral outcomes. These dimensions are supported by Komaki (1997), who develop a measure for team effectiveness in theater productions based on excellence of execution, fulfillment of staff's vision, and factors affecting group morale.

A new dimension of team effectiveness definition could be appreciated on Pagell and LePine (2002). These authors used the results from multiple case studies in manufacturing plants to identify factors that affect team effectiveness, in particular contextual factors, which are factors pertaining to team's external environment. They argue that team effectiveness is "highly contextual", and classify a team to be effective according to management perception of its contribution to firm's competitiveness. As the result of this study, the authors identified three main contextual factors affecting team effectiveness, level of trust on team, relevance of problems or challenges faced, and operational interdependence. Although, the authors admit that further research is needed to empirically confirm these results, this study brought to perspective that team effectiveness not only should measure the results of a team, but put those results in the context of the organization.

Another study that links contextual factors to team effectiveness was performed by Doolen et al. (2003). The contextual factors studied were organizational culture, organizational systems, and management processes. The authors administered a set of surveys to team members and leaders to measure the relation between these factors and team effectiveness and performance. The authors define a set of measures for team performance and a set of measures to evaluate effectiveness; performance measures were focus on team process and individual member satisfaction, effectiveness measures were based on accomplishment of goals. The measurement of team effectiveness was based on team managers' perception, while the measures of team performance were based on both managers and team members. The results showed that in general contextual factors were more correlated to team effectiveness than to team performance.

Mathieu et al. (2006) implicitly measure team effectiveness using customer satisfaction in a study where the subjects were teams of customer service engineers. The authors had another set of quantitative measures for performance related to the quality, time and cost of the service. Also, this study measure separately the team process performance based on transition toward achieving performance vision, coordination actions, and interpersonal attitudes.

These studies make a clear distinction between team effectiveness and team performance. The common ground on this distinction is that team effectiveness depends on the perception on its accomplishments by agents external to the team.

2.1.2 Measuring Team Performance

Measuring team performance is probably even more complicated than measuring effectiveness. Team performance should be measured according to several aspects such as purpose of measurement, attributes or behaviors to be measured, and measurement process-related aspects (Brannick et al. 1997). Team performance could be analyze at individual or team level, or could be a measure of team process or team outcome (Cannon-Bowers et al. 1997). For some applications, like teamwork training or individual evaluation, it would be appropriate to measure the performance of individuals within a team. While for others applications it will be more convenient to evaluate the performance of the team as a whole. The same way, outcome performance will tell you how well the team did its job regarding how they did it, but when teams' performance is to be track and improve through time; process performance should also be measured.

Baker and Salas (1997) postulate that the development of team performance measures should be part theoretical and part empirical, and “*must capture the dynamic nature of teamwork*”. Performance measures should address the behavioral, cognitive, and attitudinal aspects of a team to fully assess team performance since it could not just be measure by what team members do. Performance measures should assess team member shared mental models and experiences within the team. We can argue that the experience of a team member will influence the attitudes toward working with the same teams or other teams in the future.

Ancona and Caldwell (1992) emphasize the use of tasks process as a way to relate team composition factors to team performance, instead of processes aimed to team cohesion. Task processes focus on organizing the members of a team to complete their tasks. This should be considered particularly for ad-hoc or short duration teams. The authors used internal group processes measures and communications with external groups, rated by team members, as the measures of team functioning. The internal group measures were the team’s ability to define goals, develop plans, and prioritize activities. Managers rated team performance based on efficiency, quality of innovation on design, adherence to budget, compliance with timelines, ability to resolve conflicts, and overall performance.

Wong et al. (2000) measured performance when studying virtual teams through a simulation model by total task completion time, coordination time, and re-work time. Similarly, (Kim et al. 2003) measure performance in a simulation study about project teams in term of project duration, project cost, and project quality.

In summary, team effectiveness should measure the result of the work performed by the team or the degree that the team reached its goals, and the contribution or impact of these team results on its organization's overall effectiveness or in its environment. The effectiveness of a team is influenced by factors internal and external to the team. In other hand, team performance should measure how well the team worked to achieve its results or goals, and should focus only on factors under the influence of the team or its manager. We can segregate team performance in teamwork or process performance, and in task or outcome performance.

2.2 Factors Affecting Team Performance

This section discusses the factors or team characteristics that researchers have found to have an impact on or to be correlated with team performance. These factors or team design characteristics can be classified on four big categories: team composition, team structure, task design, and organizational context (Cohen et al. 1997; Stewart 2006). In addition to these three categories of team design characteristics, we review other factors pertaining to team members important to team performance, the psychosocial traits of the team (Cohen et al. 1997). A fifth category, team structure, is also included. The factors in this category sometimes are associated with task design or with team composition. Table 2 provides a summary of the factors discussed in this section.

Table 2: Summary of Factors Affecting Team Performance

| Category | Example of Factors | Sample of Research Studies |
|------------------------|--|--|
| Team Composition | team size, individual skills, and diversity | Ancona et al. (1992), Brannick et al. (1995), Cannon- Bowers et al. (1995), Stewart (2006) |
| Team Structure | physical dispersion, centralization, and formalization | Christiansen et al. (1999), Kim et al. (2003), Wong et al. (2000) |
| Task Design | task interdependence, uncertainty and complexity | Andres et al. (2002), Christiansen et al. (1999), Kim et al. (2003) |
| Organizational Context | Leadership and empowerment, autonomy, organizational culture, organizational systems | Doolen et al. (2003), Mathieu et al. (2006) |
| Psychosocial Traits | share mental models | Wong et al. (2001), Petre (2004) |

2.2.1 Team Composition

Group composition describes how the team is assembled in terms of individual members' characteristics. The important question about team composition is if these individual characteristics combine to improve team performance (Stewart 2006). Team composition factors include team size, individual skills, and diversity.

The impact of team size on performance is not well understood yet. Stewart (2006) found contradictorily results on the literature about the benefits of having a large team. His meta-analysis found a small, but significant effect of team size on the performance of management and project teams, but the effect was not significant for production teams. Cohen and Bailey (1997), on their study of team literature, note size may have an inverted U-shape relationship with performance, but this may not hold for every type of teams.

Both task work and teamwork individual skills are critical to team performance. Task work represent what the team does to achieve its goals, and it depends heavily on task competences as well as teamwork (Marks et al. 2001). Task expertise should be

complemented with teamwork skills development to have effective team performance (Burke et al. 2004). Brannick et al. (1995) showed that there is correlation between team skill composition and process performance measures through a study involving military air crews in simulated missions. Cannon-Bowers et al. (1995) developed an extensive list, from team literature, of skills and attitudes that influence performance. Examples of these skills and attitudes are: adaptability, leadership, communication, decision making, task integration, team cohesion, shared vision, mutual trust, etc.

Diversity refers to the variability or heterogeneity of some trait among team members. Some common types of diversity in teams include functional diversity and demographic diversity. Functional diversity refers to how diverse is the team in terms of area of expertise or function within the organization, while demographic diversity refers to how heterogeneous is the composition of the team in terms of demographic factors such as age/tenure, culture, etc (Ancona et al. 1992). Stewart (2006) found that heterogeneity is more desirable in teams doing creative work, and less desirable for management teams. Ancona and Caldwell (1992) studied the influence of functional and tenure diversity in performance of teams. The authors measured diversity in a team using the coefficient of variation for the tenure dispersion, and an entropy-based index for functional diversity. Each type of diversity showed different effects on group process performance, but seems to have the same inverse relationship with outcome performance.

2.2.2 Team Structure

Team structure refers to the nature and the strength of relationships among team members (Wong et al. 2000). Some factors commonly studied are physical dispersion, centralization, and formalization. The degree of physical dispersion of team members was studied by Wong and Burton (2000). The authors found that physical dispersion as a measure, for virtual team structure, has an effect on team performance.

Centralization was define by Kim and Burton (2003) as the level of position at which decisions are taken within the team. The authors studied the fit of centralization and uncertainty to predict team performance, and found that decentralize teams perform better in terms of quality than centralized ones. However, in terms of cost and duration of project, decentralized teams work better only under high uncertainty environments. Formalization refers to the hierarchical level in which the information is exchange and the format requirements of the communication (Andres et al. 2002). These authors found a significant and positive correlation between coordination strategy and team productivity. In this study, the coordination strategy was the combination of the level of formalization and the level of cooperativeness.

2.2.3 Task Design

Task design represents how the activities to be performed by the team are differentiated and integrated (Stewart 2006). The parameter of team design most studied in the literature is task interdependence (Pagell et al. 2002). Task interdependence refers to the extent to which a task requires exchange of products, information, and resources

with other tasks; and the extent to which the outcome of a task affect the outcome of another (Andres et al. 2002).

The interdependencies types are pooled, sequential, and reciprocal Christiansen et al. (1999). A pooled interdependence identifies that the two activities have no information dependency between them. A sequential interdependence indicates that one activity (the successor) depends on the output of another activity (the predecessor). A reciprocal interdependence indicates that both activities need information from the other one. Andress and Zmud (2002) found that task interdependence is positive correlated to team productivity; and Stewart and Barrick (2000) found U-shaped relationship between task interdependence and performance for teams performing conceptual tasks, and an inverted U relationship for teams performing behavioral tasks.

Other task characteristics that impact performance are uncertainty and complexity. Task uncertainty could be defined in terms of the extent to which the information required to perform a task is available to the individual (Kim et al. 2003). These authors found that uncertainty negatively affects team performance measures of cost, quality, and time. Task complexity is a measure of the level of cognitive demand the task imposes to the individuals (Campbell 1988). By optimizing the load of cognitive demand of the task, performance will also be optimized (Mitchell 2000).

2.2.4 Organizational Context

Leadership is the link between the team and the organization, and it has been present in the majority of team contextual research (Stewart 2006). According to this last

author's study, leadership correlates positively with team performance. This study found that both types of leadership, empowerment and transformational, also correlated positively with performance.

Similarly, Mathieu and Gilson (2006) studied the relationship between team empowerment to team process performance, quantitative or task performance, and customer satisfaction for teams of customer service engineers. The authors defined empowerment as the collective belief among team members that "*they have the authority to control their proximal work environment and are responsible for their team's functioning*". Results showed that empowerment is directly correlated to team processes and to quantitative task performance, but not to customer satisfaction (team effectiveness).

Autonomy of teams to take decisions seems to have a different impact on performance according to the type of teams and organizational environment. Teams performing physical work seems to benefit more from having autonomy than teams doing knowledge work (Stewart 2006). Autonomy seems to have a positive effect on team performance only when the organization environment favored innovation and produce high work pressure.

Doolen et al. (2003) classified the organizational context factors in three categories: management processes, organizational culture, and organizational systems. Management processes are defined as those used by organization's leadership to meet its goals, including establishing clear goals and resource allocation to teams. Organizational culture factors the extent to which the organizational culture supports teamwork, the integration of the team into the rest of the organization, and the cooperation between

teams. Organizational systems refer to the processes established to manage human resources. Some organizational systems that need to support teams are reward, feedback, training and education, and information systems. Doolen et al. (2003) found that most of the organizational context variables studied has a significant effect on team member satisfaction, around half has an impact on team effectiveness, but few has an significant effect on team performance.

2.2.5 Psychosocial Traits

The main psychosocial trait of a team is the share mental model. A share mental state is a requisite for the existence of a team (Hexmoor et al. 2002). A share mental model is a common representation among team members of the team and its objectives, roles and behavioral norms of its members, and individual and collective expectations. Shared mental models are employed by team members to anticipate the decisions and information needs of teammates (Wang et al. 2001). Kraiger and Wenzel (1997) proposed measuring share mental models by assessing how the teams process information and structure knowledge, the attitudes that enable the coordination of actions or information such as cohesion, and share expectations. Coordination requires a team to have a common mental model of the solution (Petre 2004). Other psychosocial factors present on the team literature are individual effort (Hoegl et al. 2001), cohesion (Hoegl et al. 2001), mutual support (Hoegl et al. 2001), team trust (Fiore et al. 2003), and level of cooperation (Doolen et al. 2003).

2.3 Frameworks and Models to Study Coordination

Models in the literature of Organizational Theory can be group across many different classifications. The models can be classified according to the theoretical approach it is based, like for example structural theory, information processing theory, and contingency theory (Carley 1995). Models also can be classified according to the purposes for which the model formulated. For example, Burton and Obel (1995) divided the computational organizational models based on four categories: descriptive, illustrative, normative, and man-machine simulations models.

This section discusses some relevant models or framework used to study coordination and performance in groups or teams. The models are classified in either Input-Process-Output (IPO) models or in Contingency models. IPO models view team performance as the result of how well teamwork and task work processes operate on some specific inputs. Therefore, performance can improve by improving the processes or by improving the inputs. Contingency theory argues that in order for a team perform well, the team context (inputs) and structure (processes) should fit together (Umanath 2003), and the way these factors should fit is situational specific (Carley 1995). Therefore contingency models tend to be normative rather than descriptive.

A classical example of a IPO model of coordination is found in Malone (1987). The author analyzed coordination in organizations as a function of their coordination structures. He defined four types of coordination structures: product hierarchy, functional hierarchy, decentralized market, and centralized market. The author developed mathematical functions to predict three types of cost associated to these structures: production costs, coordination costs, and vulnerability cost.

Wang et al.(2001) applied a Coordination and Decision (CODE) framework to study coordination and individual decision making in teams, developed a mathematical normative-descriptive model, and use the model as a simulation tool to study the impact of communication on team performance. The CODE framework is shown in Figure 3. According to this framework, each team member receives information from two channels, communication with other teammates, and situational channels. Based on this information and the mental model, the team member takes decision about actions and information to pass on to other team members and the environment. Team coordination is governed by these processes. The authors developed a mathematical model based on this framework and use it as a computational tool. This work provides a good reference to model the decision making involved in team coordination processes.

Espinosa et al. (2004) proposed a framework to study the effects of team cognition on team coordination and performance. The framework is shown in Figure 4. The framework follows an input-process-output model, where the inputs are task characteristics, team characteristics, and context factors. The team uses a mix of explicit and implicit coordination mechanisms to manage the dependencies emerging from the combination of factors. The result from this process is the level or state of coordination, which is an antecedent of team performance. This model was developed with the purpose of studying the process of coordination and the degree it influences team performance.

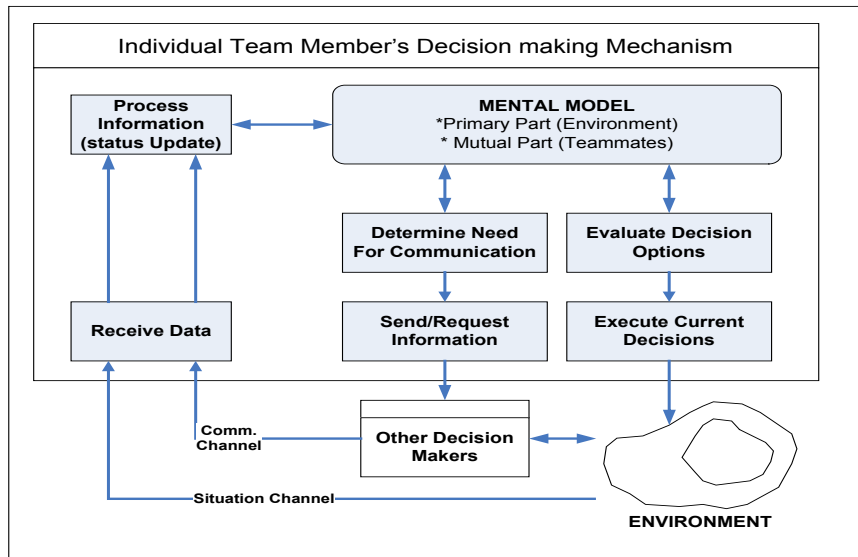


Figure 3: CODE Framework

(Wang et al., 2001)

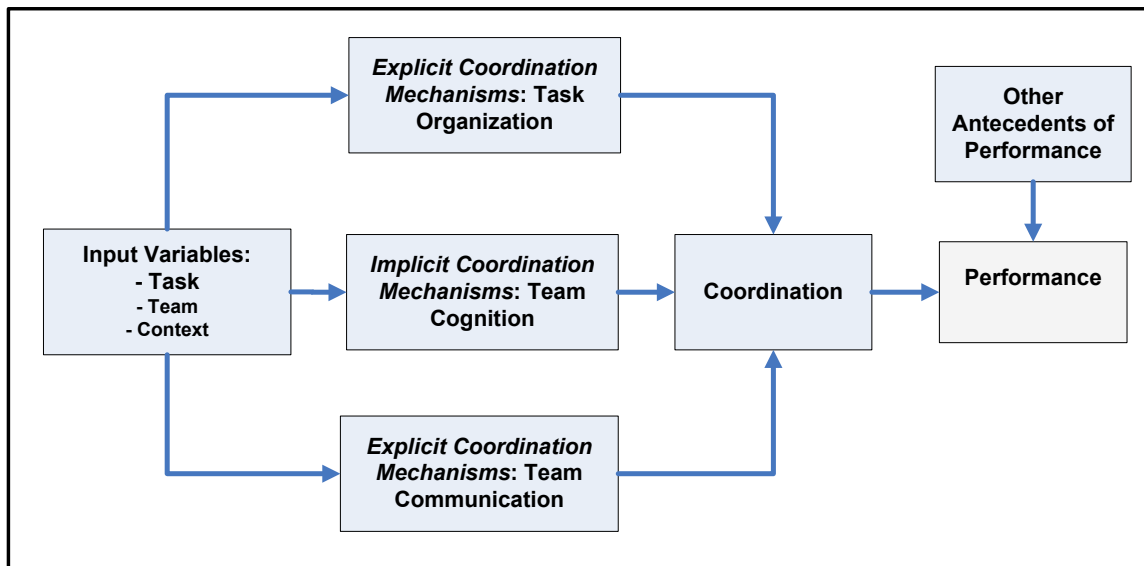


Figure 4: An Integrated Framework of Team Coordination and Performance

(Espinosa et al., 2004)

Two important asseverations derive from the work of Espinosa et al. (2004) are influential to this research. The first one is that the different combinations of inputs produce different types of dependencies and work arrangements that require different

coordination strategies. In this research, coordination strategy will be defined as the mix of coordination mechanisms used by a team. This asseveration suggests that a contingency model might be more appropriate to study coordination and performance in teams than input-output models. The second asseveration is that coordination explains part of the variance of team performance, but there are other antecedents or factors that influence performance as well. The model being develop will incorporate this asseveration, being task related skills the antecedent of performance included.

One Contingency model of coordination is found in Rathnam (1995), who studied the implications of team coordination gaps for the design of information systems. The authors proposed that coordination gaps are the results of lack of fit between coordination technology (IT) and process characteristics. They used four characteristics of the process in their frameworks: inter-connectedness, input uncertainty, distance between team member, and role conflict.

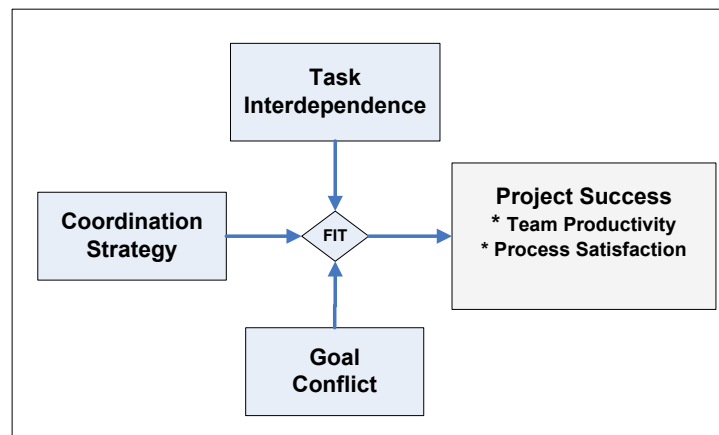


Figure 5: Andres and Zmud's Research Model

Andres and Zmud (2002) studied the coordination on software development projects with the purpose of found ideal team design configurations that optimize

performance. They used a contingency research model, shown in Figure 5, where task interdependence, coordination strategy, and goal conflict were the contingency factors. The authors argue that the fit between these contingency factors dictate the extent of information exchange and decisional autonomy required for project success. The authors implement the model through a factorial experimental design.

2.4 Team Simulation

There are three different approaches to simulate organizational behavior: mathematical, heuristics-based, and model-based (Jin et al. 1995). Both, the mathematical and the heuristic-based approaches used the aggregate organization as the unit of analysis, while the model-based approach allows for analysis at the micro or individual level. A Simulation model allows more precise implementation of social sciences theory, and allows a precise and testable representation of conceptual entities with their functions, structure, and behaviors (Kunz et al. 1998). Furthermore, mathematical or heuristics approach are not appropriate to test a contingency model of team coordination because the underlying assumption of Contingency Theory.

2.4.1 Agent-Based Simulation

Agent-based simulation (ABS) is emerging as the standard approach to simulate organizations and social groups. Agent-based modeling assumes that social structure and organizations are created by the interaction of individuals (Kiel 2005). Among the principles of ABS discussed by this author are that no rule in the model dictates the

global behavior, therefore each agent contains its own rules on how to react or behave in different situations. In the team modeling context, the team process and outcome will emerge from the behavior and decisions of each member, and the interactions between them. Each individual may choose to react different for different events or situations they encounter.

In ABS, each individual in the team is modeled as an agent. The main characteristic of agents is the capability to make independent decisions (Macal et al. 2005). Other characteristics of agents in a model are that are self contained, is autonomous, self-directed, and goal directed. An agent may contain attributes, behavioral rules, memory, resources, decision making sophistication, and rules to modify behavioral rules.

The use of ABS is justify when the mathematical relations of the model are just partially solvable or intractable (Kiel 2005), when agents are a natural representation for units in the system, when it is important that agents learn and adapt, when agents should have a dynamic relationship with other agents, and when the past is no predictor of the future (North et al. 2007). Macal and North (2005) provide a review of agent modeling and simulation development tools, such as MASON, SWARM, and REPAST.

2.4.2 Agent-Based Modeling of Teams

Agent-based models of teamwork can be classified in two main groups according to their purpose. One group focuses on developing artificial intelligent (AI) agents capable to work on teams to perform a job. The ultimate goal of these models is to

improve the effectiveness of team processes by having these AI agents to interact with or substitute humans in teams, and augmenting team decision making capabilities. Examples of these models are STEAM (Tambe 1998), CAST (Yen et al. 2006), and R-CAST (Fan et al. 2007). Agents in these models have pre-defined roles and work in a dynamic job environment.

The second group of agent-base models seeks to simulate processes in human teams with the purpose of deciding the best team and/or job configuration. The ultimate goal of these models is to predict with precision the performance of the team considering all relevant job, organizational, and individual factors. Examples of these models are Virtual Design Team (Kunz et al. 1998), Team-RUP (Yilmaz et al. 2007), and the one developed by Dong and Hu (Dong et al. 2008).

The VDT software is a “computational discrete event simulation system” which models organizations, at micro level, as information processing structures (Jin et al. 1995) (Kunz et al. 1998). The Virtual Design Team (VDT) was originally designed as a software tool that could be used to design organizations following the same design process used by engineers to design tangible objects or constructions (Levitt 2003). The first target of this tool was to simulate projects in the construction industry, although recent commercial versions allow for more general applications. The VDT model assumes a static job structure and pre-defined task assignments. The VDT research used Galbraith’s information processing view of organizations to develop the computational model of the software. The VDT applications range from modeling project organizations in the architecture and civil engineering industry (Christiansen et al. 1997; Christiansen et al. 1999), to the aerospace industry (Kunz et al. 1998; Levitt et al. 1999).

The VDT software also has been used to study teams. For example, Wong and Burton (2000) used the VDT software to simulate and analyze different virtual team models varying in virtual context, virtual team composition, and virtual team structure. Based on the study results, the authors developed a typology of situational considerations when designing virtual teams based on organizational or task requirements for coordination volume and tolerance for errors. Kim and Burton (2003) performed a simulation study using the VDT software to explore the relationship between task uncertainty, level of centralization, and project team performance. Based on this study, the authors make a set of generalizations about the best organizational structure for teams, under high uncertainty conditions, to improve each performance measure.

Loper and Presnell (2005) used agent-based simulation to evaluate the performance at individual and at aggregated level for the Georgia Emergency Management Agency (GEMA). Although, GEMA do not necessarily meet all the criteria for a team, the individuals required considerably amount of coordination to perform their tasks during an emergency. This work is also relevant for this research since the authors basically modeled the organization and its agents as information processing units. The authors implemented the model using an open source agent framework called OpenCybele.

The Team-RUP model was developed to study the effects of team behavior on the performance of software development teams (Yilmaz et al. 2007). Team-RUP was developed using RePast agent simulation toolkit. This simulation model considers a dynamic job environment and provides flexibility in terms of the organizational structure and size of the organization being model, furthermore, the model allows one agent to

represent an engineering team itself. This model is a good example of the potential of using simulation to study and design team and organizations, although its applicability domain is limited to software development organizations.

Another example on the use of simulation to study teams is presented by Dong and Hu (2008). This model was developed in RePast to study the effect on the team effectiveness of the interactions between members and between members and tasks. An interesting characteristic of this model is that it considers many type of relationships between members including friendship. However, the applicability of the model is limited to highly centralized teams that process one task at a time and only one team member works on the task.

2.5 CybelePro Agent Infrastructure

An agent infrastructure provides the runtime environment for the agents and the services needed to build the agents behaviors (IAI 2006). An infrastructure is composed of a set of classes and packages to develop the agent application, and an execution environment that runs it. The Team Coordination Model agent-based simulation application was developed using *CybelePro*, an agent infrastructure developed by Intelligent Automation Inc (IAI) that runs on top of a Java platform as shown in Figure 6. The *Cybele.kernel* package contains the classes need by a developer to build the agent application; those classes are access through the Activity-Oriented Programming Interface (AOPI).

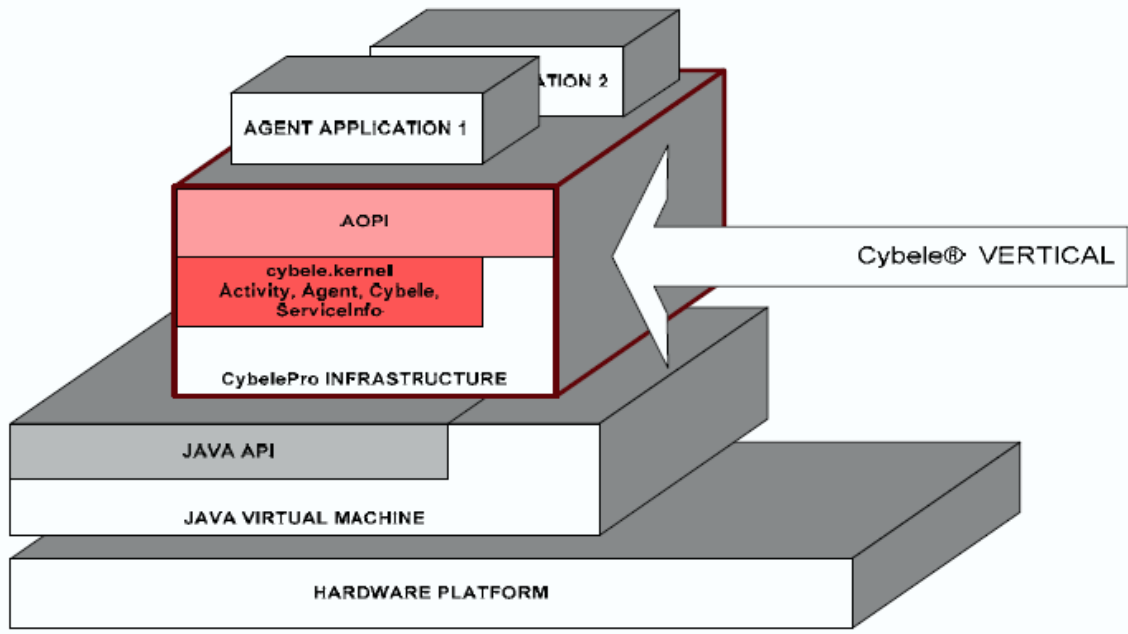


Figure 6: Cybele Agent Framework for Agent Applications
(IAI 2006)

CybelePro applications have three levels of encapsulation: container, agents, and activities as illustrated in Figure 7. The container enables Cybele’s run time environment and services for a particular application. Agents are objects with independent execution that interact with each other through a communication protocol. The simulation is performed by the system of agents in the Cybele container. The agents in the system can be classified in two categories: the ones modeling the real system and the support agents. CybelePro’s AOPI enforces an Activity Centric Programming (ACP) paradigm in which an autonomous agent is view as an encapsulated set of event-driven activities that share data, thread, and have a concurrency structure to manipulate their execution (IAI 2002). Activities are objects with independent data and execution that provide the different functions or roles performed by an agent. Each activity can be in one of six different states: runnable, active, hold, event-blocked, activity-blocked, or done. Agents interact

with each other by sending event objects between activities. Each agent has autonomy on how to interpret and act upon each event received.

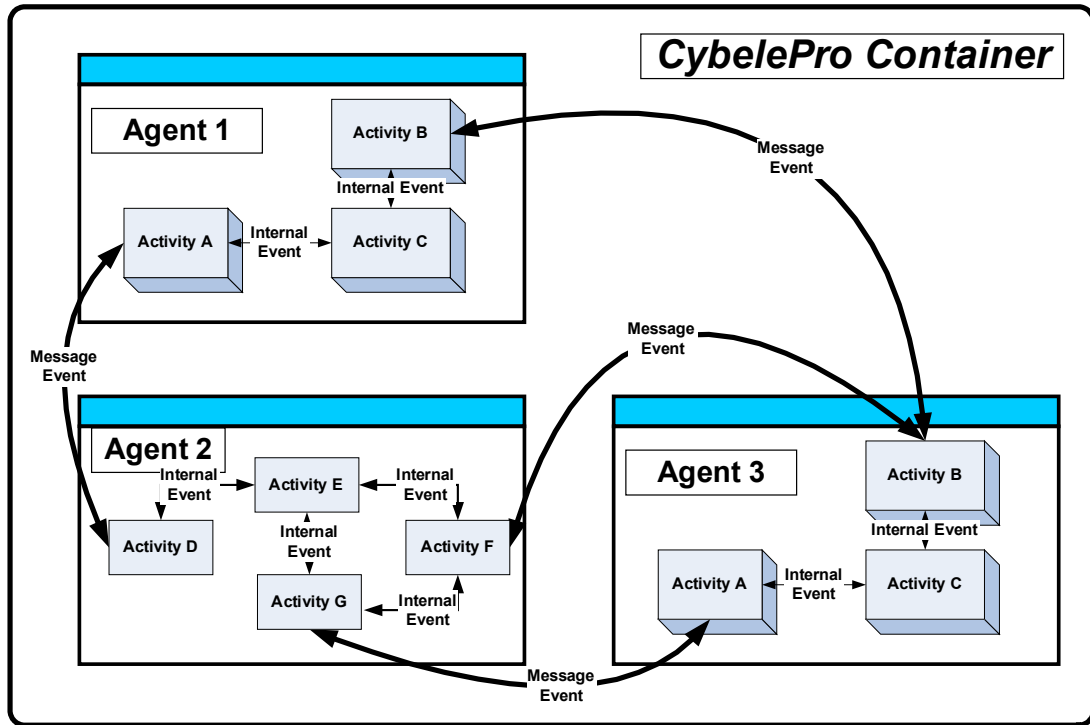


Figure 7: CybelePro Agent Application Model

CybelePro provides the option of using a discrete clock for agent-based simulation applications. The discrete clock ensures the repeatability of a simulation application, and also the time serialibility of events. Therefore, an event is not executed until all the events with a smaller time stamp are completed. Nonetheless, if two events have the same time stamp, there are executed without any particular order, but always in the same order when the simulation run is repeated.

Every application created with Cybele is event-driven in nature. CybelePro supports three types of events used to execute the flow of an agent-based simulation: messages, internal event, and timers. Cybele events trigger a method or procedure of an activity within the same agent or in another agent. Messages events are generated when an agent or activity sends or publishes a message to another agent(s) or activity with a tag attached. The message will trigger every event method setup with the particular tag. Messages events are particular used to enable the communication between agents. Internal events behave the same as the messages events with the exception that only trigger methods setup with a matching tag in activity objects within the same agent. The timer events are trigger when a timer, setoff within the same agent, expires.

Parts of the application, like the graphical user interface, were built using pure JAVA classes. Net Beans was the JAVA's integrated development environment (IDE) used to build the code of the application. This IDE simplified greatly coding and debugging the application.

CHAPTER 3:

TEAM COORDINATION MODEL

The *Team Coordination Model* (TCM) is an agent-based discrete simulation model that estimates the coordination and outcome performance of a team, based on its composition, coordination mechanisms, and job structure. This chapter describes the model and its development process. The development of the model starts by conceptualizing the relation between independent and dependent factors, and ends with the implementation of the model in an agent-based simulation application. The development process includes the adaptation of modeling languages, such as UML and SDL, to create graphical models of teams, their processes, and their jobs.

3.1 Conceptual Model

Figure 8 presents the conceptual Team Coordination Model. The model has two outcomes: task performance and coordination performance. These outcomes are affected by a series of factors grouped on the following classifications: task structure, team composition, team structure, and teamwork training and experience, and individual task and teamwork skills.

The proposed TCM model is based on contingency theory and contains two main propositions. The model proposes that outcome performance will be a fit between individual task skills and coordination effectiveness, measured as coordination load. According to the classification of contingency models found on (Umanath 2003), the TCM is a Contingency Moderation Model. The underlying assumption is that the

contingency factors are independent from each other, and the interaction between them produces the observed effect on the resulting measure. Nonetheless, the model also proposes that the coordination mechanisms, task structure, team structure, team composition and individual teamwork skills will have a contingency effect on performance through the coordination performance. This type of contingency effect is known as transitive effect. The model assumes that coordination is necessary to transmit the effect of the contingency factors to team performance. The rest of this section defines and operationalizes the performance measures and contingency factors included in the Team Coordination Model.

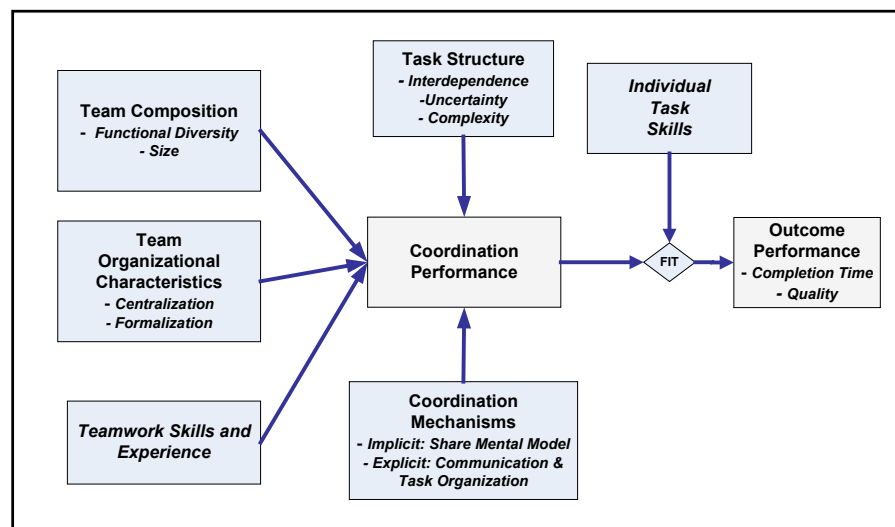


Figure 8: Conceptual Team Coordination Model

3.1.1 Performance Measures

The Team Coordination Model (TCM) considers two type of performance: outcome performance and coordination performance. Outcome performance considers the final result of individual and team efforts without considering the process of obtaining those results. Outcome performance is assessed through the following measures: the

quality of the team product or output and the total time to complete the job. These measures of team outcome performance are consistent with the literature on teams (Kim et al. 2003; Piccoli et al. 2004; Mathieu et al. 2006). For example, hurricane emergency management team performance during the preparedness could be evaluated by the total time it takes to have the population ready.

Coordination performance is used as a surrogate for teamwork performance. This is consistent with Paris et al.(2000), whom state that teamwork performance measures should “focus on coordination requirements between team members”. The measure of coordination performance in this research is coordination load. Coordination load is defined as the proportion of time spent by team members on coordination activities compare to the total time spent on the task (Nuñez 2006). The time team members spent on coordination activities is measure as the time spent on communication activities.

3.1.2 Coordination Mechanisms

The coordination load of a team is influenced by the balance between implicit and explicit coordination. Acquiring and passing information through communication channels is more costly in terms of time and effort than acquiring information through situational channels (Wang et al. 2001). Therefore, coordination load increases considerably by using explicit coordination mechanisms instead of implicit coordination. This section describes the implicit and explicit coordination mechanisms implemented in the TCM. Table 3 summarizes these mechanisms.

Espinosa et al. (2004) describe two types of explicit coordination mechanisms, which are task organization and communication. The authors group the implicit coordination mechanism under the label of team cognition. The authors found that the three mechanisms have an influence on team performance. However, the degree of influence varies with the type of task, the degree of interdependence between activities, and other team and context variables. This provides a strong support to the contingency approach to model team coordination and performance.

Table 3: Coordination Mechanisms in the Model

| Factor | Definition | Model Variables |
|---------------------|--|--|
| Team Communication | Time spent sharing information or making decisions | Media Type (memo, e-mail, etc) * Message Preparation Time * Message Transmission Time * Transmission Delay Time * Reception Time * Media Reliability Frequency (periodic or As required) |
| Task Organization | Use formal method that defines how the task should be performed | Used (Yes/No) |
| Shared Mental Model | Overlapping knowledge among teammates about teams objectives, structure, and process | Shared Mental Model factor (value between 0 and 1) |

The two explicit coordination mechanisms described in Espinosa et al. (2004), communication and task organization, are included in the TCM. Team communication refers to the time spent by individuals sharing information or making decisions with other team members, and it is modeled through media of message and frequency of communication.

Task organization refers to the use of a formal method that defines how the task should be performed. The use of task organization is evident when teams produce schedules, plans, manuals, operating procedures, administrative orders, etc. Task

organization is operationalized as a binary variable (used, not used). In the absence of a task organization method, the team will have to coordinate more through communication (Espinosa et al. 2004), therefore a positive value of this variable should decrease the frequency of communication requirements. Nonetheless, as discussed by Petre (2004), the availability of an external mental image of problem solution helps the formation of internal mental models in team members. Therefore, the availability of a task organization method should also improve the implicit coordination.

Implicit Coordination is modeled as a shared mental model. Shared mental models represent overlapping knowledge among team members about team's objectives, structure, process, roles, and behavioral and interaction patterns (Kraiger et al. 1997; Yen et al. 2006). Shared mental models are employed by team members to anticipate the decisions and information needs of teammates (Wang et al. 2001). As discussed in Yen et al. (2006), a shared mental model can be measured in terms of the level or degree of the overlapping among teammates' knowledge.

The TCM implements the shared mental model as the probability of an individual sending the information required by another team member before it is requested, and the probability that the information content and format fits the requirements of the recipient. The level of the shared mental model also will influence the probability of rework when integrating two activities due to incompatibility of outputs.

The level of shared mental model factor used by the simulation is entered by the users as two real variables: a base value and a modifying value. The modifying value modifies the base value according to other variables pertaining to each team member.

Therefore, the resulting share mental model factor (SMM), a value between 0 and 1, might vary among team members according to Equation 1.

$$SMM = baseSMM + \Delta SMM * MF(\text{Teamwork skill level}) + \Delta SMM * MF(\text{Team Experience level})$$

Where:

baseSMM = base value parameter

ΔSMM = modifying value parameter

$$MF(\text{level}) = \begin{cases} -1, & \text{if level} = \text{low} \\ 0, & \text{if level} = \text{Med} \\ 1, & \text{if level} = \text{high} \end{cases}$$

Equation 1: Shared Mental Model Factor Calculation

| Factor | Definition | Model Variables |
|----------------------|--|--------------------------------------|
| Task Complexity | Cognitive demand on team members | Complexity Level (low, medium, high) |
| Task Interdependence | Information requirements between the tasks | Dependency type |

3.1.3 Job Structure

Job structures dimensions or factors represented in the model are task complexity, and task interdependence.

Table 4 shows the definition and operationalization of the Job structure dimensions.

Task complexity can be defined in terms of the cognitive demand the task places on the team member performing it. Complexity can be measure objectively by the information load, information diversity, and rate of information change (Campbell 1988). These authors identify four objective characteristics of a task that can be used to determine the complexity level of a task: (1) the presence of multiple ways to arrive at the desired outcome, (2) task has multiple desired outcomes, (3) there are conflicting interdependence among desired outcomes, (4) the relation between processes and outcomes is uncertain. These characteristics are the based for the assessment of a task complexity level. The complexity level is parameterized into the TCM as one of three levels: low, medium, or high. A high level of complexity will occupy a larger portion of the cognitive capacity of individuals, therefore limiting the amount of information they can handle simultaneously. According to Christiansen et al. (1999), the solution complexity of a task will determine the probability that individuals make errors performing their activities.

Table 4: Job Structure Factors

| Factor | Definition | Model Variables |
|----------------------|--|--------------------------------------|
| Task Complexity | Cognitive demand on team members | Complexity Level (low, medium, high) |
| Task Interdependence | Information requirements between the tasks | Dependency type |

Task interdependence represents the information relationships between the tasks. An analysis of the interdependencies between tasks is fundamental to enhance coordination (Albino et al. 2002). Task interdependence is represented in the model as a

network, where the arc directionality represents the interdependence type and the arc capacities represent the information requirements. The types of dependencies included in the model are discussed in Section 3.2.

3.1.4 Team Composition

The team composition factors included in the model are functional diversity, demographic diversity, and team size. Table 5 summarizes the team composition and team structural factors.

Table 5: Team Composition & Team Structure Factors

| Factor | Definition | Variables |
|-----------------------|--|--|
| Size | Number of members in team | Total Number of members in the team |
| Functional Diversity | Heterogeneity of functional backgrounds and functional skills | Variability of the set of skills of each member. |
| Demographic Diversity | How heterogeneous is the composition of the team in terms of demographic factors such as age/tenure, culture, etc. | Experience level (low, medium, high) |
| Centralization | Hierarchical level that has authority to make decision | Centralization level (low, medium, high) |
| Formalization | Degree of formal/written communication and documentation in the organization | Formalization type (vertical, horizontal) |

Diversity on team composition significantly influences the effectiveness of its communication and on its performance (Ancona et al. 1992; Wong et al. 2000). Functional diversity is the degree of heterogeneity in terms of areas of expertise or functions within the organization. Functional diversity is achieved in the model through the set of functional skills and each member expertise level on each skill. Demographic diversity represents the variation in demographic variables of a team. The demographic variable included in the TCM is experience on the organization. The experience level

reflects the number of years the team member has been on the organization. The level is defined as low, medium, or high since a numerical value might represent a different level of experience in different situations. For example, the level of experience of an individual with 5 years of experience in an organization that has only 6 years of existence can be consider high; while the level of experience of an individual who has been working 5 years with a century old organization might be consider low.

Team Size is measured by the number of individuals that are part of a team. Stewart (2006) found a small, but significant effect of team size on performance for management and project teams, but not so for production teams. However, some of the studies he studied found that large teams tend to struggle on coordination. The optimal number of team members varies depending on the type of team and task to be performed.

3.1.5 Team Structure

The team structural characteristics included in the TCM are centralization and formalization. Centralization refers to the hierarchical level where decisions are made (Kim et al. 2003). Centralization can be parameterized as low, medium, and high. High centralization levels imply that decisions are made by a supervisor or manager, while in low centralization levels, decisions are made by team members. Low centralization corresponds to low coordination requirements. Table 6 contains more specific definitions for the centralization levels.

Table 6: Centralization Levels Definition

| Centralization Level | Level Definition |
|-----------------------------|--|
| High | Team members need to consult all or most of the decisions with the team leader or supervisor |
| Medium | Team members take some decisions themselves, but consult other |

| | |
|-----|---|
| | decisions with the team leader |
| Low | Team members take most of the decisions themselves, and only consult with the leader the most difficult decisions |

Formalization refers to the hierarchical level in which the information is exchange and the format requirements of the communication. Formality is categorized as horizontal and vertical (Andres et al. 2002), where horizontal formalization encourage the informal communication between peers and vertical formalization encourage passing information through the leaders or managers supervising the team. Also, horizontal formalization may open the use of more open, fast, and informal communication methods such as phone calls, e-mails, etc. Vertical formalization required the use of rigid or formal channels of communications like memos, meetings, etc. The TCM includes formalization as a three levels (low, medium, high) variable. Table 7 defines the formalization levels.

Table 7: Formalization Levels Definition

| Formalization Level | Level Definition |
|----------------------------|--|
| High | Vertical formalization is required most of the time |
| Medium | Both vertical and horizontal formalization are used. |
| Low | Team uses horizontal formalization most of the time. |

3.1.6 Individual Skills

One obvious factor affecting outcome performance is individual skills on the task domains. No matter how well a team coordinates its work, if team members do not possess the requisite skills and knowledge, then it is likely the team performance will be poor. In the TCM, individual functional or task skills are measure qualitative as having three levels of expertise (low, medium, and high). As in the VDT approach, high team skill will decrease the probability of an activity to have to be reworked. The level of

expertise of each team member for each skill is an input of the TCM. A low level of expertise means the team member does not possess formal training or previous experience on skill domain. A high level of expertise means the team member does possess all the training and/or experience on the skill domain. A medium level implies some training and/or some experience on the domain.

Teams that have better teamwork skills demonstrate greater teamwork effectiveness (Hirschfeld et al. 2006). Teamwork skill will have the same operationalization as functional skill; however it will not affect the task work, but the effectiveness of the coordination mechanisms.

3.2 Team System

The TCM is an agent-based simulation model that implements the conceptual model of team performance and coordination described in the previous section. The system emulated by the TCM is a team that is composed of two or more persons, and has one or more well-defined jobs to complete, as illustrated in Figure 9. By well-defined we mean the goal, outcomes, and tasks of the job are known. The team divides the tasks among themselves, works on the tasks, communicates among themselves, and coordinates their work until the team objectives are met. The model is stochastic; we allow for failures, conditional tasks, random task durations, and other random events that the team must respond to. The total time to complete the job is the primary measure of team performance. We also calculate coordination load as the team process measure.

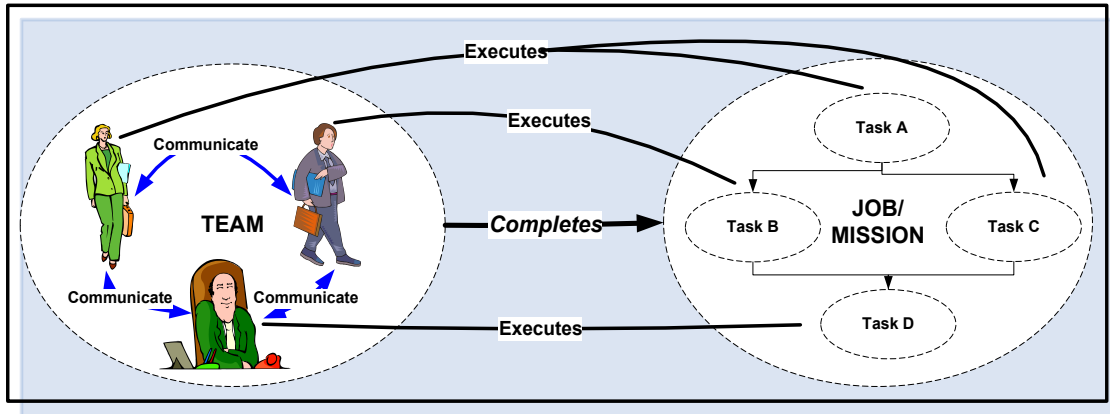


Figure 9: Representation of the System Model

3.2.1 Team Job Model

The only elements external to the team included in the model are jobs. This research defines a job as a unit of work that the team, as a group, needs to complete to accomplish its mission or purpose. The job has well-defined completion criteria such as the generation of a certain output. This means we know when the job is completed. A job is subdivided into two or more tasks. This precludes trivial jobs that have only one task. The tasks in a job will have dependencies between them.

This research models a dynamic job environment in which the team starts with a fix objective, but the job structure to achieve the objective varies according to team decisions, and environmental factors. The variation in the job environment of the team consider by this research are:

- Stochastic task duration
- Unknown number of task to be executed in a job
- A task might require just one of the predecessors to be completed instead than all of them.

This research models the job structure as an activity network where the nodes represent the tasks and the arcs between nodes represent dependencies. The job structure is defined by the dependencies between its tasks. If the dependencies are all deterministic, then the structure is static. In the other hand, the number of tasks and the order in which the tasks are executed might vary as the result of conditional or stochastic dependencies. Most team modeling research assumes static interdependencies as is typical in project PERT networks. The TCM also considers conditional interdependencies, which give rise to an uncertain and dynamic task structure.

There are three basic types of dependencies are pooled, control sequential, information sequential, and reciprocal (Giachetti 2006). Pool dependencies arise when two or more tasks require the same resource, in this case same team member, to be completed. This type of dependency was included in the model implicitly since a team member agent in the simulation might have to decide one task to perform among two or more available. Control sequential dependencies arise when a task can not be started until a predecessor task finishes. In the information sequential dependencies, a task requires the information output of a predecessor task before start its execution. In reciprocal dependencies, two tasks require some information outputs from each other before they can be completed.

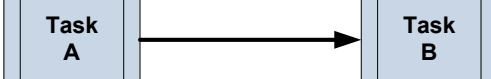

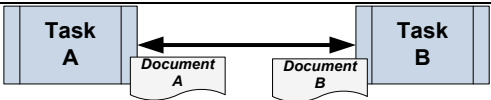

This research introduces additional types of dependencies that differ from the basic types in terms of the execution type. The execution type denotes if the dependency is always required or its execution is conditional to some event. There are three categories of execution for dependencies: deterministic, conditional, or single-conditional. The three basic types of dependencies are deterministic and are always

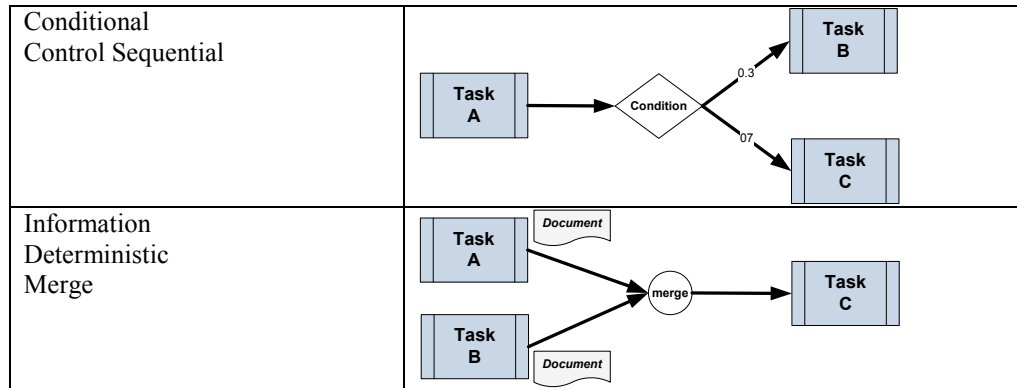
executed. For example, in a deterministic information sequential dependency, the successor task always received the information from the predecessor task.

A conditional dependency represents the case when a subsequent task is chosen from two or more mutually exclusive tasks depending on a condition. The condition can be a decision taken by the team or a random event. Each event has a distinct occurrence probability. Probabilities can be estimated from historical data or from expert experience.

A single-conditional task dependency is a special case of a conditional-sequential dependency in which the condition results in only two possible events; one that triggers the subsequent task and one that triggers no task. Table 8 shows graphically the dependencies types explicitly included in the TCM. The conditions are resolved after the predecessor task is completed. The probabilities of the successor tasks form a discrete user distribution used by the simulation to decide which of the successor task will be executed.

Table 8: Type of Interdependencies included in the model

| Dependency | Dependency Diagram |
|--|--|
| Control Sequential Deterministic |  |
| Information Sequential Deterministic |  |
| Reciprocal |  |
| Single-Conditional Control sequential |  |



Another type of dependencies introduced by this research is the merge dependency. A merge dependency is based on a logical OR; in this dependency a task can start when any of one or more predecessor tasks are completed. It is appropriate to clarify that the single-conditional and the conditional dependencies can be control or information dependencies and that only deterministic reciprocal dependencies will be included in the model. Table 9 list all the attributes needed to define all the types of dependencies.

A task is defined as a portion of the overall job that transforms inputs into outputs and can be assigned to a single team member. A task might have one or more inputs and one or more outputs. Each task is assigned a priority. The priority indicates the recommended order in which tasks should be completed. Task priorities are an output of a critical path algorithm performed by each team member.

Table 9: Attributes of a Dependency

| Attributes | Description / Values |
|--------------------|---|
| Predecessor Task | The ID of the predecessor task |
| Successor Task | The ID of the successor task |
| Dependency Type | Control sequential, Information Sequential, Reciprocal |
| Execution Type | Deterministic, Conditional, Single-Conditional |
| Merge Indicator | Merge / Not Merge dependency |
| Information Object | ID for the information object or document that defines an Information sequential dependency |

| | |
|------------------------|---|
| Condition | ID for the event that defines a conditional dependency |
| Dependency Probability | Probability that the dependency will be selected from a conditional event |
| Merge ID | ID to identify a Merge of dependencies |

During the process of completing a job, tasks pass through a series of states. Table 10 defines the seven possible states for a task. Figure 10 shows the allowable transitions between task states. A job is complete when all its tasks reach either the *Not Required* or *Done* absorbing states.

Table 10: Task Execution States

| Task State | Description |
|-------------------|---|
| Hold | Initial State for a task. State for tasks that have not completed the dependencies requisites to start execution. |
| Pending | State of tasks with all the requisites for execution that is waiting for the assigned team member to be available. |
| On Process | State of a task currently being executed by a team member |
| Pending Rework | State of a task that suffered a failure during its execution and its waiting for the assigned team member to become available again |
| On Rework | Task that is being re-worked |
| Done | Absorbing state for tasks that have completed the processing. |
| Not Required | Absorbing state for task with conditional dependencies that were not chosen during a conditional event. |

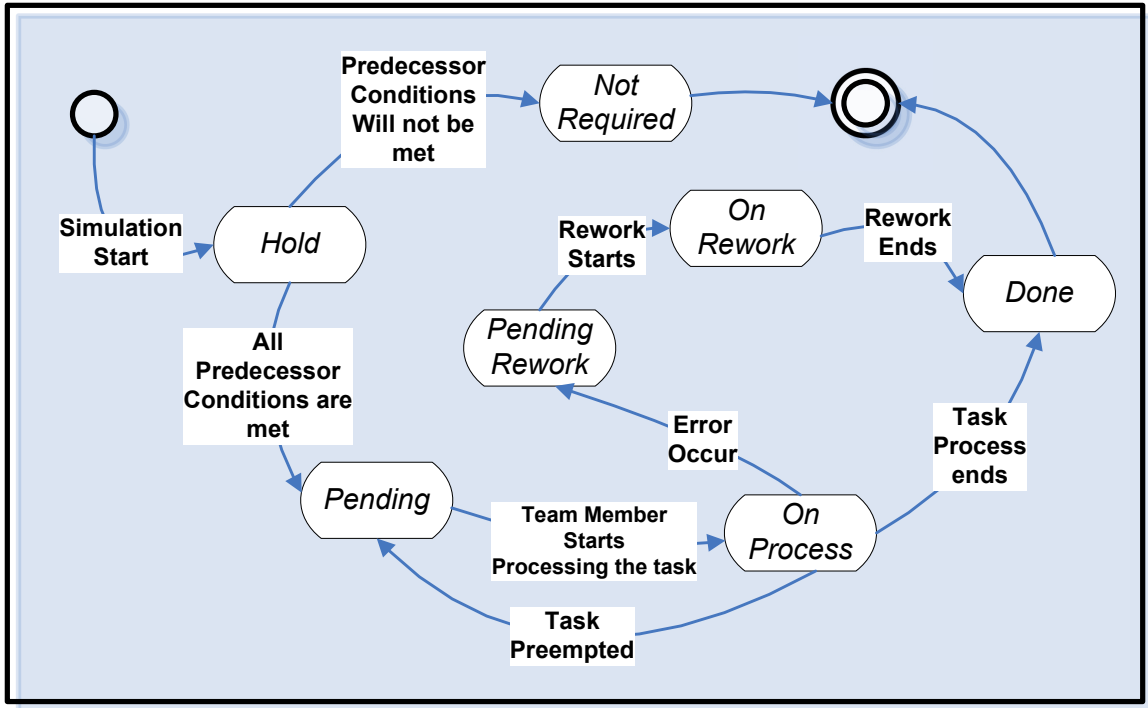


Figure 10: State Diagram for Task Execution

Tasks are characterized by their duration and complexity. Task complexity can be defined in terms of the cognitive demand the task places on the team member performing it, and was discussed in Section 3.1.3. Task duration is assumed stochastic following a triangular distribution. The triangular distribution is used since it is easy to implement and frequently the data available is insufficient to fit to other probability distributions. The triangular distribution only requires the analysts to estimate the minimum, the most likely, and the maximum duration for the task. Nonetheless, additional distributions could be added to future versions of the TCM.

The output of a task could be visible for other team members, besides the one performing it. If a task or its output is visible, the need for communication is reduced. Each task requires a specific set of skills that the team member should have to complete it

efficiently. This research assumes that the match between the team member skills and the skills required by a task impacts the duration of the task and the probability of a failure during task execution. Table 11 summarizes the information required to define a task.

Table 11: Attributes that Define a Task

| Attributes | Description / Values |
|-------------------|--|
| Task ID | Identifier of the task |
| Complexity | Cognitive demand on team members |
| Duration | Defined as a triangular distribution (min, most likely, max) |
| Priority | Indicate the recommended order in which a task should be performed related to the other tasks in the job. |
| Assigned Agent | Team member recommended or assigned to perform the Task |
| Skills | Set of skills required to perform the tasks adequately |
| Visibility | Indicates if team members can perceive the status of this task from the environment, visually or through another sense, without the need to receive communication from the agent working it. |

3.2.2 Team Communication

The only explicit coordination mechanism modeled in the simulation is the communication between team members. The model conceptualizes communication as an event that transfers a message from a sender team member to a receiver team member over a media. The communication media refers to the channel that carries the message from the sender to the receiver. Examples of communication media are meetings, phone calls, e-mails, faxes, and one-to-one conversations. Messages transmitted by these different methods vary in terms of the time it takes to prepare the message and time to transmit the information by the sender, delay in the reception of the information by the receiver(s), time to read and/or convert the information to a usable format, and the reliability of the media (probability that the information will be received on time). Teams

might employ more than one method of communication each with different frequency of use.

Figure 11 shows the taxonomy of communication events developed for this research. The purpose of this taxonomy is to simplify the classification and processing of messages in the simulation. The communication can be asynchronous or synchronous depending on the media used (See comparison in Table 12). Also, each communication event is classified by its intent, content, and message. The communication intent indicates the purpose of the message. The communication intent is either to transfer, request, or acknowledge information. The communication content refers to the type of message to be transferred or requested. The content could be information, a decision, or an action. The message specifies the information, decision, or action of the communication event. The communication media defines the process used to transfer the message; and the combination of intent, content and message defines the actions taken by the receiver to process it. For example, a team member could send an e-mail to his supervisor asking him to confirm his next task assignment. In this case, the media is asynchronous, the intent is to request, the content is a decision, and the message is the request itself.

In the model, teammates communicate for the following general purposes:

- To send information about a task
- To request information about a task
- To solve a conflict emerging when two members are assigned to the same task.

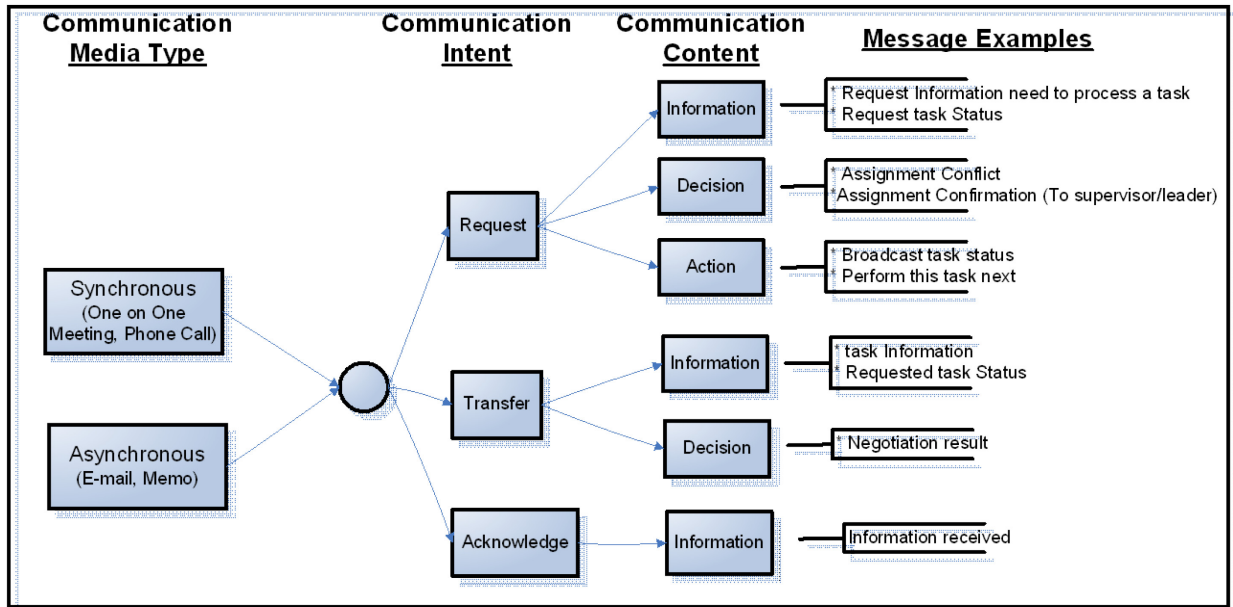


Figure 11: Taxonomy of Communications Events

Table 12: Comparison between Asynchronous and Synchronous Media Types

| | Asynchronous | Synchronous |
|----------------------|---|---|
| Modeling Assumptions | <ul style="list-style-type: none"> * Significant Time to create message, follow triangular time distribution * reading time is negligible * Message is not instantaneously available to the receiver after it has been send. Time to receive follows triangular distribution. * There is a probability $P_{AsyFail}$ that the message did not reach its destination. | <ul style="list-style-type: none"> * Sender spends a short amount of time to establish communication with the receiver * Communication time occurs simultaneously for the sender and receiver, lag time is negligible. * No significant time is needed to create or receive the message besides the communication time. * There is a probability $P_{SynFail}$ that the media will fail at some point during the communication |
| Predominant Use | Message Urgency: Medium to low Org. Formality: Medium to High | Message Urgency: Medium to high Org. Formality: Medium to low |
| Examples | e-mail, memo | One-to-one conversation, phone call, radio call |

The communications between teammates included in the model are:

- *Task Status* – These messages are triggered when a team member completes the execution of a task or when a team mate request the status of a task. The owner of the task will send a message to those team mates

that are in charge of executing the successor tasks to notify about the status of the predecessor task. If the dependency type between tasks is information sequential and the status of the predecessor task is “*Done*”, then the message contains the information document.

- *Request Information/status about Task* - These messages are triggered when a team member cannot perform any of his tasks because all are in the “*Hold*” state; the team member will select his/hers highest priority task and sends a request message to every team mate that is assigned to the predecessor tasks. The team member receiving this message will respond with one of the following messages:
 - *Task Status* – If the receiver of the request has the task assigned.
 - *Not My Task* – If the receiver of the request does not have the task assigned.
- *Communication due to Reciprocal Dependencies* - The model assumes that in reciprocal dependencies, the information from the reciprocal task is needed at the mid-point of a task execution. This assumption is made for simplification purposes. Then, at midpoint of the execution of a task with reciprocal dependencies, a team member will send a message with the reciprocal information to the team mate assigned to the reciprocal task. Then, if the reciprocal information has been received, the team member continues the execution of the task; otherwise he/she preempts the task until the reciprocal information is received. Figure 12 illustrates this

protocol. If a team member preempts a task, he/she also sends a *Request Task Status* message to the team mate assigned to the reciprocal task.

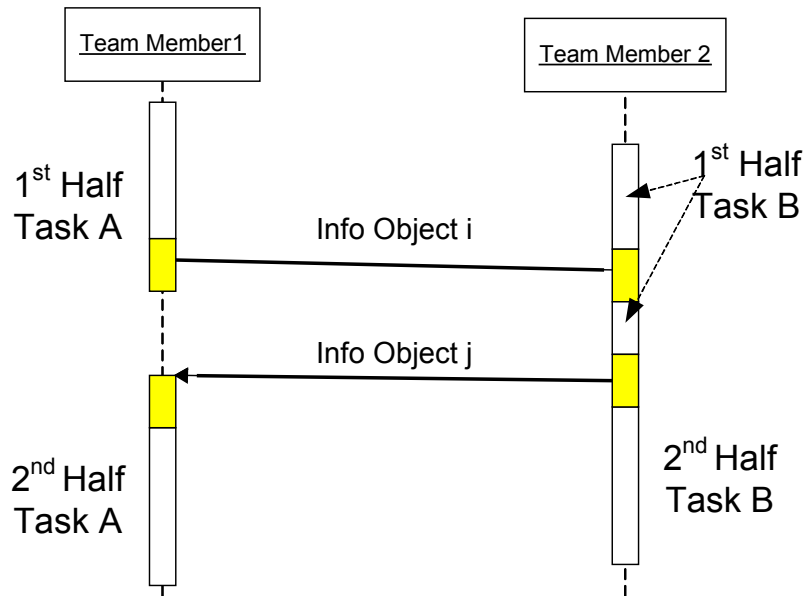


Figure 12: Implementation of Reciprocal Dependencies

- *Negotiation* - A conflict emerges between two team mates when both are (or think they are) assigned to the same task. Team members employ negotiation to solve this conflict as illustrated in Figure 13. The process starts when a team member receives a notification from a team mate that is starting processing a task. If the receiver of the task recognize the task as one of the task he/she is suppose to process, he/she will send a “*Negotiation required*” message to the sender of the notification. When the member processing the task receives this message, he/she will evaluate who should keep the task, and returns a negotiation response. This response could be one of the following messages:

- *Yield Task* - if the member who starts the negotiation will keep the task assignment.
- *Claim Task* – if the member processing the task will keep the task assignment.
- *Negotiation Tie* – if neither of the members wins the task over the other. The team members will resort to the leader to solve the conflict.

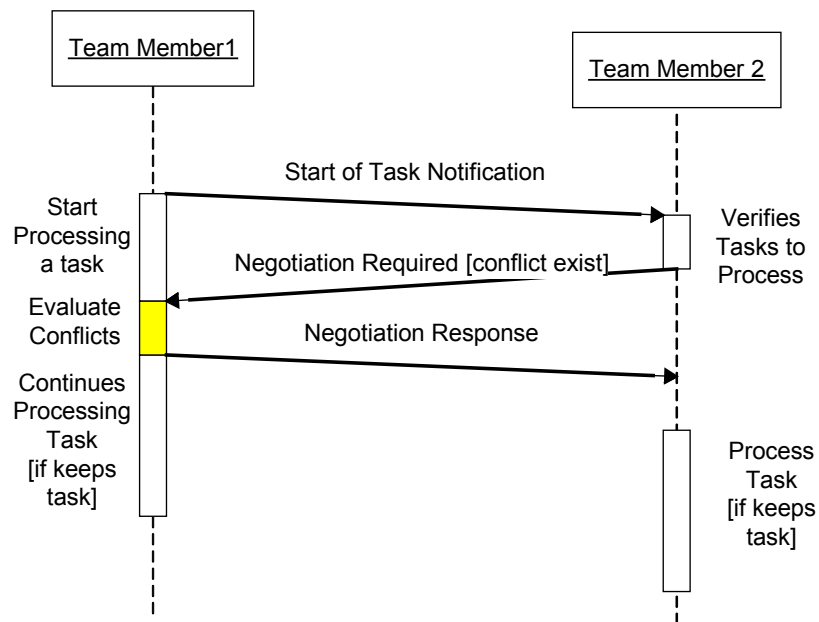


Figure 13: Negotiation Process between Teammates

This research assumes that the negotiation is solved using a simple set of rules. The team member that keeps the task assignment is the one that can finish the processing of the task first. Tiebreakers are solved using the seniority of the team members, the most experience get the task. If the tie continues after the first two rules are applied, then the conflict is solved by the leader.

Leadership is a key component for the success of a team. The model requires that one of the members be the team leader which performs the following tasks:

- Make changes in assignments
- Confirm a task assignment
- Order a team member to report the status of a task
- Resolve the conflict in assignments between teammates

The frequency the team leader performs these tasks depends on the centralization level of the team. Since the leader depends on the communication with the rest of the team to perform his/hers responsibilities, the centralization level increases the coordination load of the team. The communications between team members and the team leader included in the model are:

- *No Task To Process* – A team member (sender) notify the leader that he/she are idle because all of his/hers tasks are on “*Hold*” state or he/she have no tasks assigned. The leader responds with one of the following messages:
 - *Perform Task* – If the leader finds a task assigned to the sender with a “*Pending*” status (the sender might not have received a predecessor information); OR otherwise, the leader finds a task with “*Pending*” status which required skills are a good match with the skills of the sender.
 - *Task Status*- If the leader do not found a tasks with “*Pending*” status to assigned to the sender, then he/she will reply with the status of every predecessor task with a, “*Done*” or “*Not Required*” status, of the highest priority task assigned to the sender. Also, the leader will

send a *Request about Task* message to every team member with a predecessor (of the sender's task) task assigned that has a "*Hold*" or "*Pending*" status.

- *No Task to Perform* – The leader sends this message if he/she does not find a task assigned or to assign to the sender.
- *Confirm Assignment* - When a team member is ready to start the execution of a task, it might ask the leader to confirm if the task is the one he/she is suppose to do. The probability of a member sending a confirmation message is influenced by the team centralization level. When the leader receives a confirmation message, he/she will respond with one of the following messages:
 - *Go Ahead* – If the task is the one the team member is suppose to do next.
 - *Perform this Task Instead* - If the member is suppose to execute another task instead
 - *No Task to Perform* – if the leader founds no task for the member to execute.
- *Identify Owner* – This message is send to the leader when a team member receives a "*Not My Task*" message from a teammate. The leader, after receiving the "identify" message, founds out who is the owner of the task in question and returns a "*Task Owner*" message.
- *Assignment Conflict* - When the leader receives this message from a team member, he/she determines who of the two members involved in the conflict

is supposed to execute the task. Then, he/she sends a “*Perform Task*” message to the member responsible for the task, and a “*Drop Task*” message to the other members involved.

3.3 Team Member Model

This research models a team member in terms of the functions needed to work in a team. Team members’ functions include executing the tasks need to achieve the team mission, communicating with teammates, and processing the information exchange during communication. Also, team members should exert some decision making abilities to determine which tasks to execute, when to communicate with teammates and leader, and follow orders from the leader. One of the most important functions of a team member is the ability to coordinate with teammates the execution of the team’s job. Besides, explicit coordination abilities implemented in the model through communication, a team member should have implicit coordination abilities. The implicit coordination is included in the model in the form of a team member share mental model.

This section describes the conceptualization of the team member model. First, the share mental model is explained, then the decision making and task processing functions of the team member. The model of communication abilities were discussed in the previous section.

3.3.1 Shared Mental Model

The need for a mental model in the simulation is based on the premise that individuals form a mental construct about their roles or responsibilities in the team, and

the roles of their teammates. In other words, individuals form their mental plans on how to execute the team job or mission. This mental model might not be optimal and might not be the one the team finally executes. Another premise of the mental model is that individuals form their mental plan either based on some explicit plan developed before the job begins or following some simple rules to allocate the tasks based on their mental assumptions about their teammates (*Team Mental Model*).

This model assumes that if the team has an explicit job execution plan (called a *Task Assignment Plan*), individuals will form their mental models following this plan. This will be the case of the majority of teams since it is a common and good practice to have a plan or training on team member's responsibilities before starting the job.

On the other hand, if there is no Task Assignment Plan available, individuals will form their mental plans based on some simple rules that they can follow mentally. Assuming that individuals want the team to have a good performance, it can also be assumed that individuals will consider the following criteria for task allocation:

- *Assign the tasks to those individuals that are better qualified to perform them.*
- *Distribute the task assignment as evenly as possible.*

This last assumption implies that is better for team performance to assign some tasks to less qualified individuals that have time available to do it than assign it to highly skilled individuals that already have many previous assignments. Also, these rules might make an individual to differ with the explicit execution plan.

The level of organizational centralization will have an impact on the process of dividing the job's tasks among team members. In an organization with a high level of

centralization, it will be most likely for team members to consult or confirm with the team leader the tasks they should perform, and it will be less likely they object their assignments when they differ. On the other hand, on low centralization organizations, team members will rely more on their mental models for decisions on their task assignments.

Based on these ideas and assumptions, the Model implements the following processes related to team members mental models:

1. Team members develop their mental models.
 - If the team uses a Task Organization Mechanism, team members will develop their initial mental models based on this plan.
 - If there are not a Task Organization Mechanism, team members will develop their initial mental model following the criteria for task allocation explained previously.
 - The initial mental model will include the following information:
 - The member that will be responsible to perform each task, including the tasks that only will be required under certain conditions.
 - The order or priority of processing each task.
2. Team leader updates his/her mental model and changes assignments.
 - The team leader updates his/her mental models every time a team member notifies a task completion or that a task needs rework.

- The team leader checks if the total completion of the job will be affected considerably in result of the task completion event. The leader will explore changes in the current task assignments if the job will experience a delay. The changes will occur if a team member is busy to perform a task assigned to him/her that has a “Pending” status; in this case, the leader finds out a idle team member which skills matches positively with the skills required by the task in questions.

3.3.2 Decision Making

Team members require doing at least some simple decision making in order to complete the job. This sub-section describes the decision making functions included in the Team Coordination Model.

1. *Decide which task to perform* – The rule used by a team member to decide which task to perform is always do the task with the highest priority among the tasks in a “*Pending*” state. After decide which task to perform, a team member will do the following decisions:
 - *Send a message to team mates communicating the start of the task* – a team member might or might not communicate an “*On Process*” task status to team mates.
 - *Confirm the task assignment with leader*- Depending on the centralization level of the team, a team member might decide to

ask the leader to confirm that the task he/she is about to start processing is the best thing for him/her to do.

2. *Decide to accept a synchronous communication request* – When a team member receives a request from a team mate to engage in a synchronous communication, he/she might do one of three behaviors:

○ *Accept the request*- If the team member is idle OR performing a not urgent task OR preparing a low priority asynchronous message, then the member will establish the synchronous communication with the teammate requesting it.

○ *Reject the request*- If the team member is performing an urgent task OR creating a high priority asynchronous message, then will communicate to the teammate requesting the synchronous communication that he/she rejects the request. Then the requesting team member will save the message and might try to communicate later on.

○ *Ignore the request*- If the team member is busy on another synchronous communication OR, is performing a task AND working on an asynchronous communication, the member chooses to ignore the request and do nothing about it. The requesting team member will try again to send to establish the communication.

3. *Decide between perform a task or perform a coordination function like send a message to a team mate or receive asynchronous messages sent*

by team mates – The team member should constantly evaluate what to do between perform a task, send a message, or process a received message. The member will decide what activity to do on the basis of the respective priorities of tasks and messages.

4. *Solve conflicts on task assignment* – If the team is not perfectly coordinated two team mates might end up with the same task assignment. The team mates then will have to negotiate, or let the leader decide, who should end up performing the task in conflict. The negotiation rules were discussed on the previous section.

3.3.3 Task Processing

A team member model should include some functionality to perform tasks that are assigned to the member. The Team Coordination Model includes the following task processing functions as part of the team member model:

- *Perform a task* – The team member can select a task for processing once all the predecessor tasks are completed (task attains the “*Pending*” state). The task processing is model as an activity the member is engage during a period of time. The task duration is stochastic and is specified as a triangular random variable. The duration variable is further modified based on the skill fit between the team member executing the task and required task skills. The actual task processing duration is a random variate of the duration variable.

- *Stop the task execution momentarily when required and restart it afterwards*

While performing a task, a team member can engage in communication activities. The model assumes that a member has to stop processing a task while engaging in synchronous communications or when preparing a asynchronous communication. However, the model assumes that the member can receive asynchronous communication while processing a task without the need to stop. The member will resume the processing of a task right after the communication activity is done.

- *Preempt a task if necessary* – One rule implemented into the model is that the team member always will process the task available with the highest priority. The member can preempt a task he/she is performing if another task assigned to him/her reaches the “*Pending*” state and has a higher priority than the one being performed. The preempted task will be continued after the new task in process is done.

- *Rework the task if an error occurs* – Team members can make mistakes during the execution of a task, this task will require rework. The mistake can occur at any point during the processing of a task. The amount of rework is estimated stochastically as a percentage of the remaining time of the task. The rework will be performed by the same agent that commits the mistake; nonetheless, the leader can assign the task needing rework to

somebody else. The rework process is identical to the regular task performing process.

3.4 Team Coordination Model Implementation

The Team Coordination Model (TCM) was implemented as an agent-based simulation developed with Cybele Pro, an infrastructure for the development of agent-based systems that runs on top of a Java platform. The agents were designed as a combination of Cybele Activity Objects that provides the functionality of the agents, and Java Objects that store and manage the data of the entities included in the model.

Figure 14 illustrates the architecture of the agent-based simulation model. The job environment is modeled as an agent that releases the jobs to the team members. The team is represented by two or more team member agents that have coordination, decision making, and task processing functions that interact to obtain and process the information required, and perform the tasks. Team members also interact between each other through communication that simulates the use of different media types.

In addition to the team members agents and the job environment agent, there are two more agents created in the simulation: the Simulation Controller agent & the Statistics Reporter agent. The Simulation Controller controls the creation of the rest of the agents, and the start and end of each replication. The Statistic Reporter collects the statistics from the other agents and prepares the report of each replication. The user interface was created using Java classes.

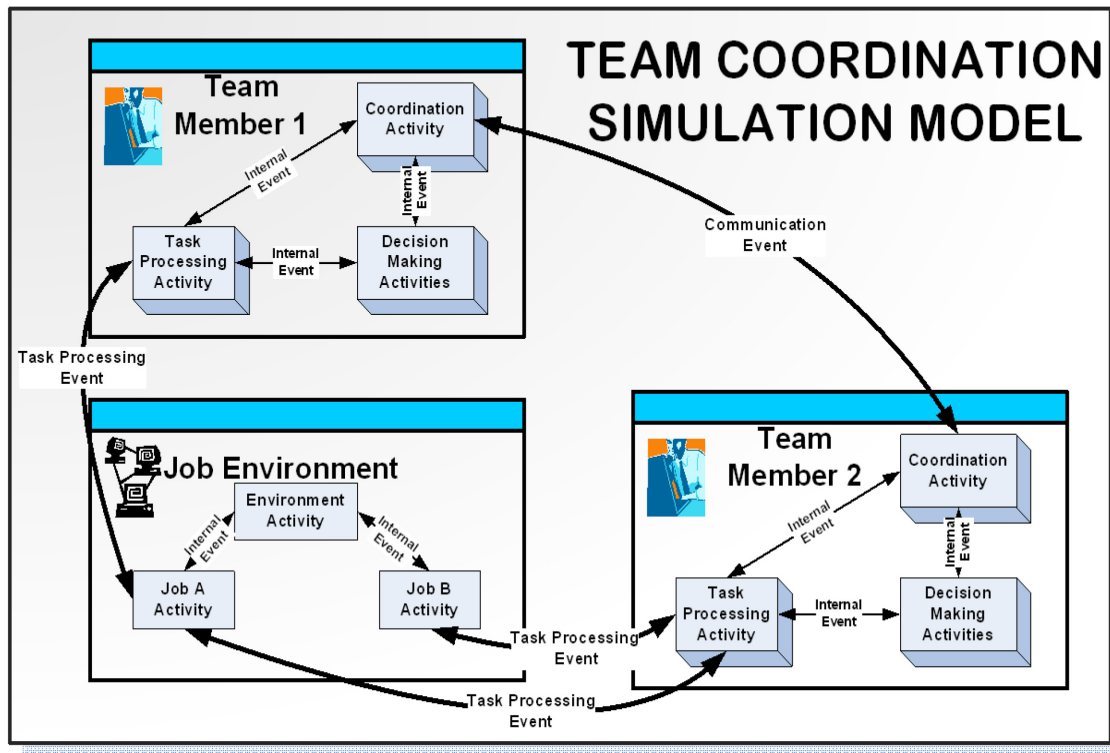


Figure 14: Team Coordination Model Architecture

3.4.1 Java Data Objects

The simulation uses a set of serializable Java objects to store and manage the data of the entities included in the model. These objects are required to transfer the data between the different agents and activities. Classes need to implement the *Serializable* interface in order to pass objects in messages in a Cybele Event, and to create arrays of objects of this class. Figure 15 shows the relationship between the data classes and the agents.

The data classes include in the model are:

- *Time Triangular* – this class is used to model the duration of task and coordination activities of the simulation, which are assumed to follow a

Triangular probability distribution. The class includes methods to generate random variates of the variable.

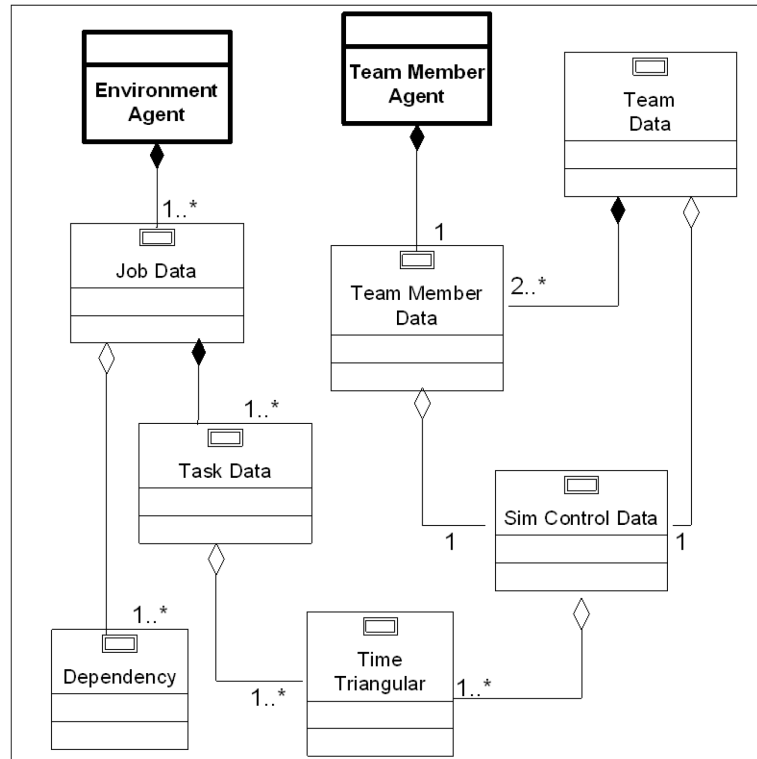


Figure 15: Relation between Data Classes and Simulation Agents

- *Dependency* – This class contain all the information needed to define and carry out dependencies between tasks.
- *Task Data* – This class contains all the input parameters that define a task, some variables needed during the simulation to describe the task status, and the variables needed to collect statistics about its execution. The class includes methods to modify and retrieve the information that it stores about the task.

- *Job Data* – This class contains the data that define and support the *Job* agent. It contains arrays of dependency data objects and task data objects.
- *Communication Media* – This class is used to store all the parameters that define a communication media.
- *Sim Control Data* – This class stores all the simulation parameters used during the simulation, other than the parameters used to describe the tasks, jobs, and team members. Among the data carry through this data objects are the communication media objects.
- *Team Member Data* - This class contains all the data that define a team member.
- *Team Data* - The Team Data object do not belongs to any agent in the simulation and its primary use is to deliver the team member data objects together.
- *Messages* – This class contains the data that defines a communication message to be transfer between team members.
- *Message List* – This class stores an array of Messages data objects, and includes methods to manage the array.
- *Task List* – This class stores an array of Task data objects, and includes methods to manage the array.

3.4.2 Environment and Team Member Agents Models

The *Job Environment* agent is composed of two types of Cybele activities objects and three types of serializable Java objects, shown in Figure 16. The *Environment* agent

activity object is the first created by the Simulation Controller agent, and the one that defines the agent. This object creates the *Job* activity objects based on the *Job Data* Java objects. The *Job* activities send the tasks to the team member agents and collect the statistics related to the tasks execution.

The job data objects are serializable Java objects that stores all the information that defines a job and its statistics. Each job data object contains a task data object for each of its tasks and a dependency data object for each dependency between two tasks.

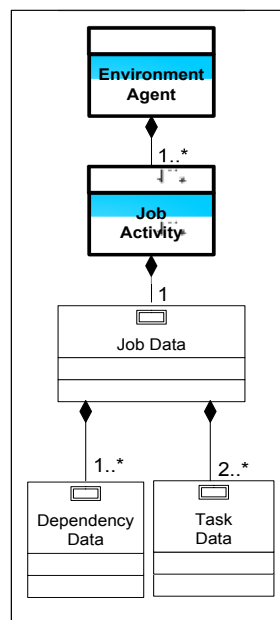


Figure 16: Composition of the Job Environment Agent

Each team member agent is composed of various Cybele Activity objects and three types of Java objects as illustrated in Figure 17. The *Team Member* activity object is the one that defines the agent and creates the other activity objects; however it does not execute any functionality of the team member. This object receives the *team member data* object that contains all the data that defines the team member and it is also used to collect the agent's statistics.

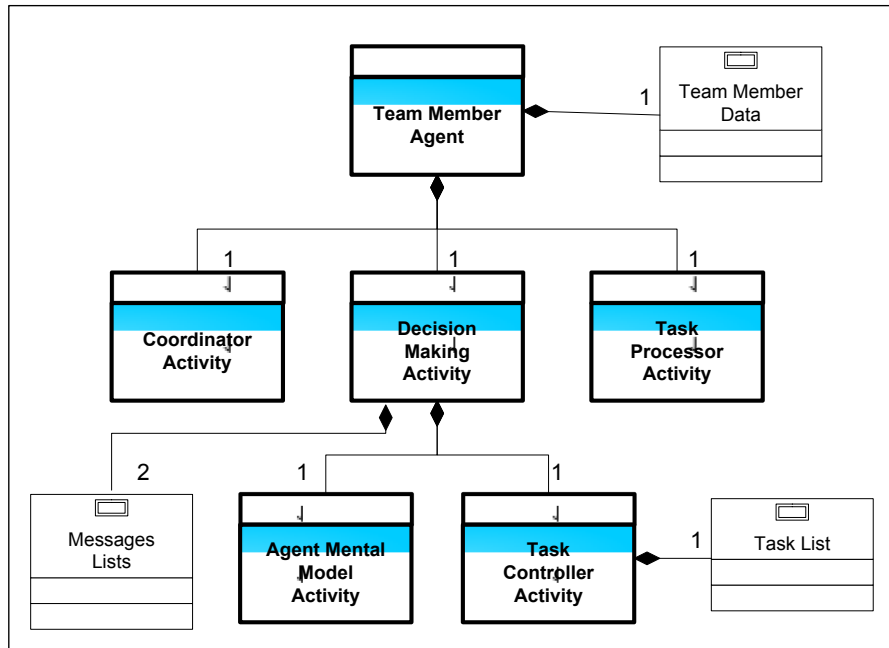


Figure 17: Composition of the team member agent

The *Task Processor* activity object controls the processes that simulate the execution of tasks by the team members. This object also performs the communication with the *Environment* agent. The *Coordinator* activity object of the agent controls the processes that simulate the asynchronous and synchronous communication with other *Team Member* agents. The *Decision Maker* and *Task Controller* activities object are in charge of most of the decision making and time allocation of the agent. An *Agent Mental Model* activity object is created and used by the *Decision Maker* object to implement the share mental model.

The *Decision Maker* object also creates and uses two *messages list* Java objects. One of these objects is used to store and manage the asynchronous messages received by the agent from other team member agents; the other messages list object is used to store

the messages the agent needs to send to other agents. The *Task Controller* object creates and uses a *Task List* object to store and manage the tasks assigned to the agent.

3.4.3 Execution of the Simulation

Figure 18 shows the flow of task processing events that drive the simulation of a team executing a job. The task processing starts when the environment agent releases a job to the team member agents. The team member receives the job and forms a mental model about the job. The job mental model contains the agent perception about how long the job should last and about which agent is suppose to perform each task. The mental model sends the tasks assigned to its agent to the Task Controller object.

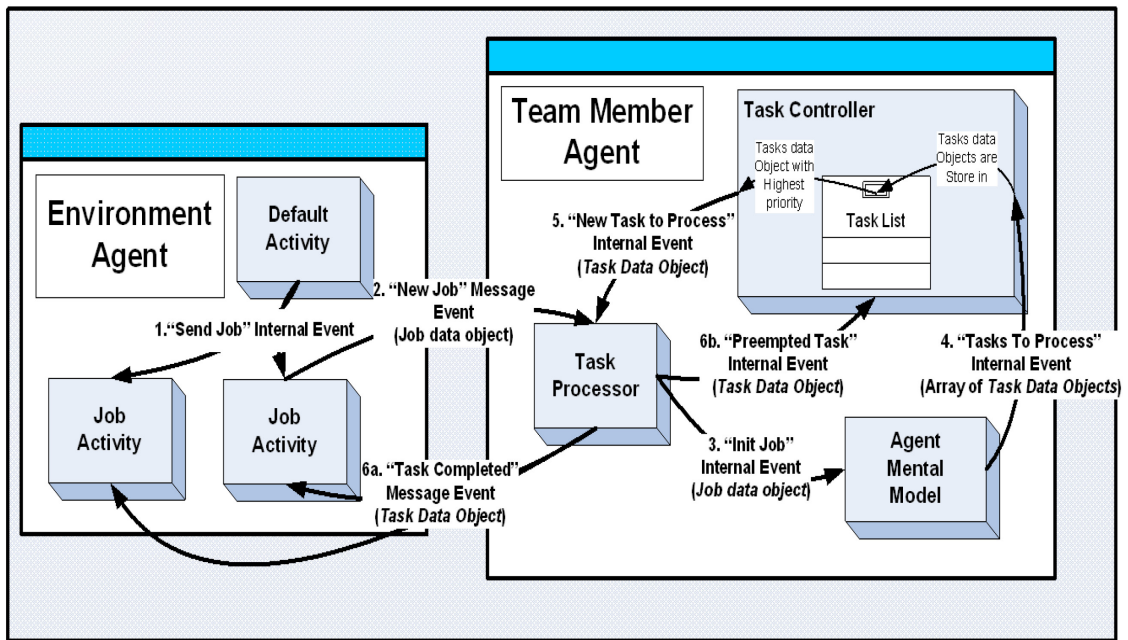


Figure 18: Flow of Task Processing Events

The tasks assigned to each agent are placed in their task list. The agent selects among the tasks with a “*Pending*” status, the one with the highest priority and sends it to the Task Processor object. The task processing duration is generated as a random variate of the duration triangular variable.

During the task processing the team member agent may receive communications from other team members. These messages interrupt the task execution. The team member agent pauses task execution, attends to the message, and then resumes task execution based on the priority of the message. Also, during the execution of the task, an agent could make a mistake that requires the task to be reworked. A task to be reworked is returned to the Task List in the Task Controller object.

When an agent receives a message regarding the status of a task that is a predecessor of a task in its list, the agent proceeds to update the task data object in the list. If a task attaining the “*Pending*” status has a higher priority than the one currently on process, the agent proceeds to execute the task with higher priority. The current task is preempted and returned to the Task List on the Task Controller object.

Once a task is completed, the agent sends a message notifying the event to teammates. The decision to send the notification to all teammates or to only those with successors is made stochastically. If the agent decides to send the notification to every team mate, it will increase the communication overhead. On the other hand, the agent might commit a mistake if decide to send the notification to only teammates with successor tasks since its mental model might be erroneous or outdated.

The Task Data object of a completed task is transferred back to the job activity object in the Environment agent. Once the team member agent successfully completes the task,

he will start the next task in his list and the above process is repeated until no more tasks are in the “*Pending*” state. If an agent do not have a task to process, it will remain idle but receiving messages and updating task information. If a task reaches a “*Not Required*” status, the team member agents also sends the Task Data object back to the Environment agent. The simulation ends when all the Job objects in the Environment Agent receive all of its task data objects back from the team member.

CHAPTER 4:

MODEL VALIDATION

A simulation model is built upon beliefs and assumptions about the behavior of an actual system (Garrido 2001). The model verification and validation process seeks to prove that these beliefs and assumptions about the actual system made during the modeling part are implemented correctly and are adequate for the purpose for which the model was built. Giachetti (2010) establishes that validity is not a yes/no qualification of the model, but a matter of the degree of confidence the users can have on the model as a representation of the system, and in the results, decisions, and analysis derived from its used. Hence, the validity of a model defines how well it can be used for its intended purpose.

The purpose of the Team Coordination Model (TCM) is to represent team behavior and estimate performance so that the teams can be designed for greater expected performance. To verify and validate whether the TCM sufficient serves this purpose, various techniques were employ to demonstrate data validity, programming verification, internal validity, conceptual validity, and operational validity. This chapter starts by describing the scenarios used to verify and validate the TCM model and explaining the overall validation strategy. Then it describes each method used in this process including a discussion of the respective results.

4.1 Test Scenarios

This section describes two complex scenarios developed for the purpose of verifying, and validating the Team Coordination Model. Both scenarios described in this

section were critical during the verification of the simulation model and the debugging of the computer application. Some toy scenarios were also used during the development of the simulation model but are not described in this document because they were built only to test some specific function of the program and discarded afterwards.

The IMT scenario is based on a hypothetical situation that provides the necessary complexity to test the model with a relatively large number of tasks and team members. The tasks are of relatively long duration and the structure is complex enough to test the model handling of the different types of interdependences.

The Race Committee scenario is based on a real team and consists of a relatively small number of team members and tasks. The duration of the tasks and communications is much smaller than those of the IMT scenario. Nonetheless, the Race Committee scenario's job structure is also complex enough to test possible conflicts due to task interdependences.

4.1.1 University Incident Management Team

The University Incident Management Team (IMT) describes a team of administrators that is activated to prepare the University for a Hurricane Emergency. The purpose of the IMT is to coordinate all the activities required to prepare the institution for the emergency as faster as possible while minimizing the resulting impact of the emergency incident. Also, the IMT is the source of the official communications from the institution to all its constituents regarding the preparedness, response, and

recovery of the university. The main responsibilities of the IMT during the preparedness stage for a hurricane are:

- Recommend courses of actions about hurricane preparedness to the university management (Chancellor, provost, etc.)
- Perform press conferences (if required), and generate official statements regarding the university preparations for the emergency.
- Make sure that all the precaution measures are taken to minimize the impact of the hurricane on the university community and operations.

The IMT is activated 72 hours before the estimated landfall of the hurricane, if the university area is on the potential range of impact. The team has until the hurricane warning is emitted (around 60 hours \pm 6 hours) to finish all the preparedness.

The IMT team is composed of 9 members, and is lead by the university's Emergency Management Coordinator. The job is composed of 31 tasks interconnected by 52 dependencies. The job structure is shown in Figure 19. The job structure is represented as a network of tasks joined by dependencies. SDL's symbols were used to build the network. The numbers within the parenthesis in each task node represent the minimum, most likely, and maximum duration for the task.

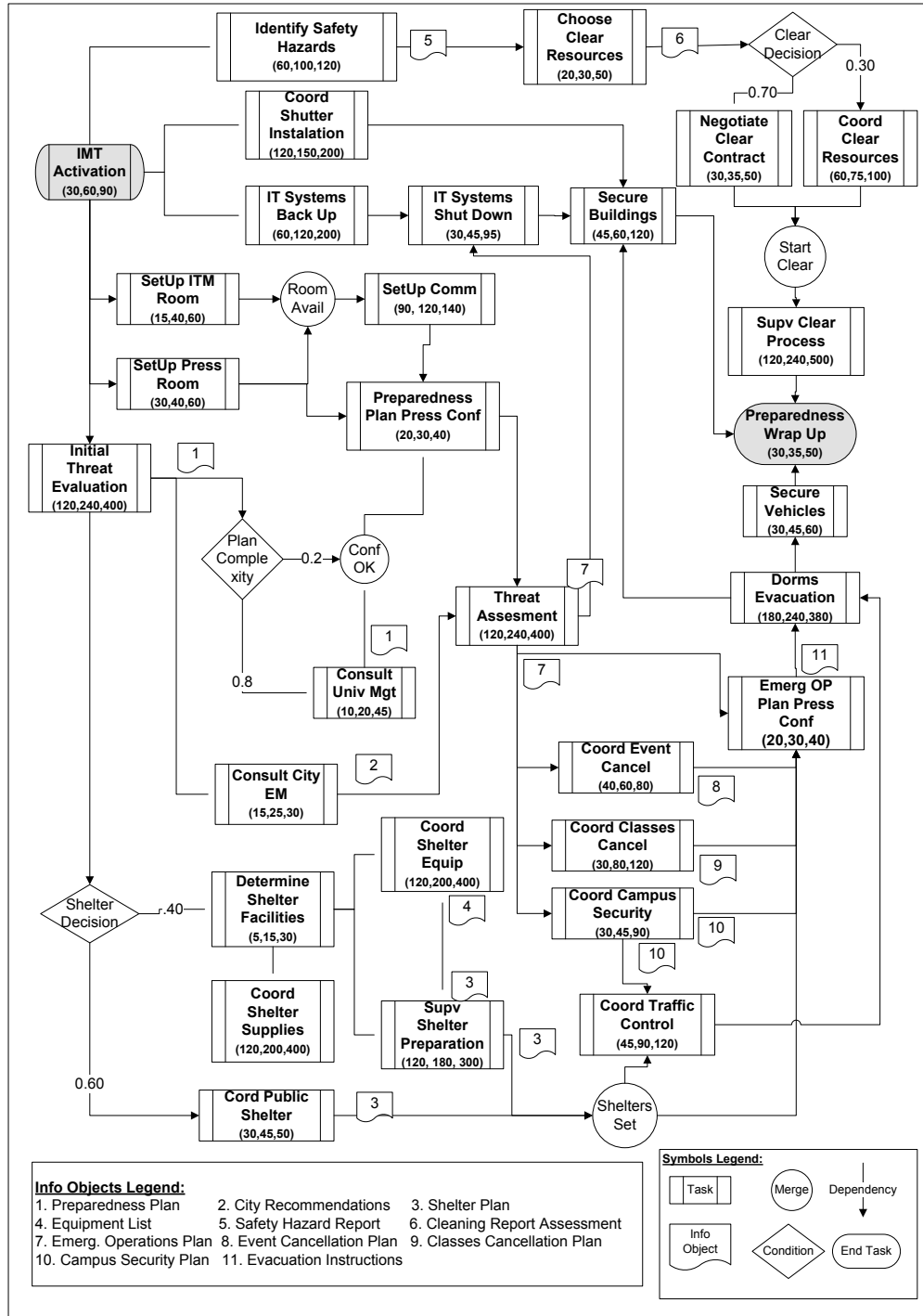


Figure 19: Campus Hurricane Preparedness Job Network

4.1.2 Race Committee Team

The Race Committee is a team of volunteers who run sailboat races. Their job is to setup the race course, run the race, and then score the results. In this simulation we focus on the first part of their job, to setup the race course. A sailboat race course is setup so that the racing sailboats start at the leeward (downwind) mark or starting line, and then they race into the wind to the windward mark. They turn at the windward mark and race back to the leeward mark. They do the loop once more, and then on their return to the leeward mark they finish the race. The task of the race committee is to setup the race course by determining the wind direction and strength so that the marks can be correctly set. The race committee then must start the race, which involves a timed sequence of flags and sound signals to notify all the racers. The race committee monitors the start to see that all racers comply with starting rules to ensure a good start.

The Race Committee operates on three boats: the signal boat, the pin boat, and the mark boat. The signal boat is where the Principle Race Officer (PRO) is situated. The PRO is the team leader, runs the race, and is the final authority for all decisions on race management. On the signal boat, there is a person to do the flags, a person to score, and a person to do the timing. The pin boat is responsible for setting up the starting line. The mark boat is responsible for setting up the windward mark. The mark boat should also report to the PRO the wind strength and direction because the wind might be slightly different at the windward mark.

To setup the race, the PRO measures the wind strength and direction. The PRO decides on the course length and direction; this information is transmitted to the mark boat. The mark boat sets the mark at the distance and compass heading. Meanwhile, the

pin boat sets the starting line. Usually the starting line is setup visually, with the PRO calling in directions to the pin boat. The racing sailboats check-in to the race by sailing by the stern of the signal boat, where a crew member records their sail number. Once the course is setup, the PRO and the mark boat continue to take wind readings. If there is a wind shift the PRO needs to decide whether to reset the course. If the decision is to reset the course, then he must instruct the mark boat on a new direction and possibly a new distance. Sometimes, if the correction is large enough they will also reset the starting line to keep it square (perpendicular) to the wind. Once everything is good to go, the PRO will go into the starting sequence. During the starting sequence every crew member on the signal boat is occupied. A timer calls out the time, the flag person raises and lowers the appropriate flags, and a line sighter watches the starting line to see if any racers go over early.

The Race Committee scenario is modeled as a team with 4 members working on a single job with 16 possible tasks. The team consists of the PRO, the signal boat, the mark boat and the pin boat. The job structure, shown in Figure 20, consists of 18 dependencies including two sets of conditional dependencies and a single conditional dependency. The job structure in Figure 20 includes the parameters for the task duration distribution for each task, estimated from a sample of various races' setup jobs. The average completion time of the setup jobs by the committee is 46 minutes.

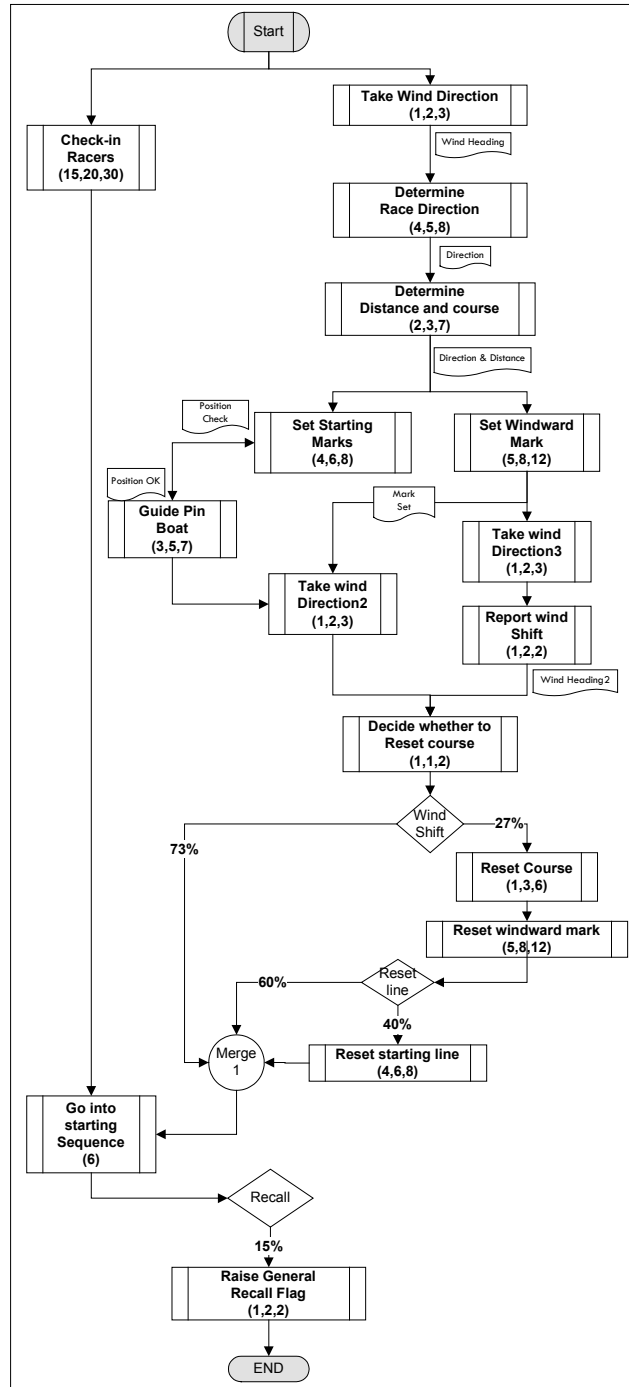


Figure 20: Job Structure for the Race Committee Scenario

4.2 Model Verification and Validation Strategy

The validity of a model indicates the degree of confidence the users have that the model can adequately represent the actual system to satisfy its purpose (Giachetti 2010).

Validation increases user confidence in the analysis and decisions made based on the model results.

Sargent (2007) points that a model validation process consists of data validity, conceptual model validation, computerized model verification, and operational validation. Data validation assures that the data used to build, validate, and experiment with the model is adequate and correct. Model verification ensures that the computer programming and conceptual model implementation are correct. Conceptual model validation determines that the theories and assumptions upon which the model was built are correct and the model representation of the problem entity is adequate for the purpose of the study. Operational validation determines if the model output has sufficient accuracy to use the model for its intended purpose. Table 13 summarizes the strategy used to validate the model.

Table 13: Summary of Model Validation & Verification

| Technique/ Test | Validation Dimension | Scenario Used | Results Comparison |
|---|---|------------------------|---------------------------|
| Job Structure Verification | Data Validity | Race Committee | With Real System |
| Debugging | Model Verification | Both & test bed models | N/A |
| Execution Tracing | Model Verification | Race Committee | Model Results |
| Internal Validity | Model Verification Operational Validity | Both | Model Results |
| Comparison with Actual System | Conceptual Validity Operational Validity Model Verification | Race Committee | With Real System |
| Reasonable Output – Change parameters one at the time | Conceptual Validity | Both | With Literature |

To validate the data, we checked that the job structure defined for the race committee job is plausible when compare with the total job completion time observed

from the real system. Although the Race Committee scenario is based on a real team, data obtained was limited due to number of races available for data gathering and due to the fact that this research do not had any control on the job structure or team composition. Nonetheless, the objective of the analysis described in this chapter is to verify and validate the conceptual model and its implementation as a simulation tool to study and design teams; for this purpose a realistic scenario will suffice.

Sargent (2007) describes two basic approaches to model verification: static testing and dynamic testing. Static testing involves checking the program code, while dynamic testing requires running the model and checking the output. This research utilized both approaches to verify the correct implementation of the Team Coordination Model and to verify the simulation application. Both the dynamic and the static approaches required the used of several logs built within the simulation application. The logs show information about the events taking place during the simulation and the respective changes in the state of the agents and its objects. Two of the verification techniques employed, *debugging* and *execution tracing*, required the use of these logs. The third verification technique employed, internal validity, analyzes only the results of the simulation.

Proving both, conceptual validity and operational validity are required before perform further experimentation with the model. This research employs two main approaches to test the validity of the Team Coordination Model. First, the results of the simulation are compared with the data compiled from the actual system and the scenario job structured of the Race Committee scenario. Second, the behavior of the model is compared with the results documented in the literature. Table 13 summarizes the

process used to validate the *Team Coordination Model*. It is worthy to note that these approaches are intended to assess the degree of validity of the model behavior and results to analyze and study teams. The validation of the model for predictive purposes will be limited by the small amount of data compiled from a real team scenario.

4.3 Data Analysis and Model Verification

4.3.1 Job Structure Verification

The purpose of this analysis is to check the feasibility of the job structure defined for the Race Committee scenario. This analysis determines the range and most likely value of the job completion time according to the job structure, and compares these results with the average completion time observed from the real system. The average observed from the real team should be higher than the most likely value calculated here since this analysis only considers tasks duration, and does not includes the duration of coordination activities. Nonetheless, the analysis would reveal if the job structure is not feasible. For example, a value for the calculated minimum completion time larger than the observed average would imply that either the task duration distributions are wrong and/or the dependencies are wrong. Also, significantly smaller calculated values draw similar conclusions, for example if the observed values double the calculated values.

This analysis is similar to finding the critical path of a network, but considering that the critical path can change due the probabilistic dependencies between tasks and the variability on tasks duration. All the possible paths in the network are considered to determine the range and most likely values of the job completion. The alternative paths

generated by conditional dependencies are average based on their probabilities. It is worth mentioning that this analysis did not involved the simulation model, and the calculations were made using MS Excel.

Table 14 shows the alternative sequence for the possible paths with its respective probabilities of occurrence, and the minimum, most likely, and maximum completion times. Figure 21 shows a simplified network representation of the Race Committee's job structure with the number outside the parenthesis representing the task number used to define the sequences on Table 14. The numbers inside the parenthesis indicate the minimum, most likely, and maximum durations for the task. A path is defined by the deterministic dependencies. The alternate sequences are generated by the distinct routes a path could follow depending on the probabilistic dependencies. The path including the sequence of task 3, task 4, and task 8 is not shown in the analysis because the sequence of task 3, 6, 7, and 8 is clearly dominant (have a larger minimum, most likely, and maximum values). The path averages are calculated by summing the products of each alternate sequence value by its probability. All the sequences in Table 14 starts in the "Start" node and finalize in the "End" node.

Table 14 shows that the averages for the paths B and C are similar, with C's values being slightly larger. Therefore we conclude that path C is the critical path. The average completion time observed from the actual team was 46 minutes, which is within the range resulted for critical path C. Also, the calculated most likely value is 26.3% smaller than the observed average completion from the actual team. The duration of the coordination activities could account for the difference. These results shows that the job

structure defined for the Race Committee scenario is a feasible representation of the actual job perform by this team.

Table 14: Results of the Job Structure Verification

| Path | Alternate Sequences | Seq. Prob. | Sequence Estimated Completion Times | | | Path Average Completion Times | | |
|------|--------------------------------------|------------|-------------------------------------|----|-----|-------------------------------|------|------|
| | | | MIN | ML | Max | MIN | ML | Max |
| A | S,5,15, E | 0.85 | 21 | 26 | 36 | 21.2 | 26.3 | 36.3 |
| | S,5,15,16, E | 0.15 | 22 | 28 | 38 | | | |
| B | S,1,2,3,4,9,10,11,15, E | 0.6205 | 21 | 29 | 43 | 23.2 | 33.5 | 49.6 |
| | S,1,2,3,4,9,10,11,15,16,E | 0.1095 | 22 | 31 | 45 | | | |
| | S,1,2,3,4,9,10,11,12,13,14,15,E | 0.0918 | 31 | 48 | 71 | | | |
| | S,1,2,3,4,9,10,11,12,13,14,15, 16,E | 0.0162 | 32 | 50 | 73 | | | |
| | S,1,2,3,4,9,10,11,12, 13, 15,E | 0.1377 | 27 | 42 | 63 | | | |
| | S,1,2,3,4,9,10,11,12,13,15,16,E | 0.0243 | 28 | 44 | 65 | | | |
| C | S,1,2,3,6,7,8,11,15,E | 0.6205 | 22 | 30 | 44 | 24.2 | 33.9 | 50.0 |
| | S,1,2,3,6,7,8,11,15,16,E | 0.1095 | 23 | 32 | 46 | | | |
| | S,1,2,3,6,7,8,11,12, 13, 14,15,E | 0.0918 | 32 | 47 | 70 | | | |
| | S,1,2,3,6,7,8,11,12, 13, 14,15, 16,E | 0.0162 | 33 | 49 | 72 | | | |
| | S,1,2,3,6,7,8,11,12, 13, 15,E | 0.1377 | 28 | 41 | 62 | | | |
| | S,1,2,3,6,7,8,11,12, 13, 15,16,E | 0.0243 | 29 | 43 | 64 | | | |

Table 15: Functions and Features Checked During Model Verification

| Function or Feature Checked | Verification Approach |
|--|--|
| User interface and correct formation of job and team | Verified using the application logs generated by the IMT scenario |
| Formation of mental model of each agent based on the share mental model factor | Verified using the application logs generated by the IMT scenario |
| Correct execution of task interdependencies | Verified using the application logs and simulation results report for the IMT scenario and various toy problems scenarios |
| Both, asynchronous and synchronous communication between agents (team members) | Verified using the application logs generated by the IMT scenario |
| Decisions made by the agents and the leader | Verified using the application logs generated by the IMT and the Race Committee scenarios to check that decisions were communicated and that the following actions were as expected |
| Simulation Results | Verified using the application logs and simulation results report for the IMT scenario and various toy problems scenarios to check correctness and consistency of results. (See sections 4.3.2 and 4.3.3 for more details) |

Throughout the development of the simulation model application, some simulation runs end up in an infinite loop or deadlock situation. This behavior occurs when one or more of the tasks were not executed by any of the agents in the team. Some causes identified for this behavior are:

- None of the agents has the task(s) on their assigned tasks list. This mainly occurs when the team does not use a Task Assignment Plan (TAP). Also, it might occur if the team leader assign a task to another member, and this member did not received the message from the leader.
- The agent assigned to the task(s) never receives the message with the information from one or more predecessors.

Although many modifications made to the model decrease the frequency of this behavior, the problem has not been completely eliminated. Since some probabilistic behaviors included in the model, such as the possibility of a message failing to reach the

intended agent, seems to contribute to the problem, its complete eradication implies radical changes to the model. These changes were left for future versions of the model. Section 4.5 provides further discussion about this problem.

4.3.3 Execution Tracing

One of the activities performed towards the end of the verification process involves checking the simulation reports. One problem detected with the statistics in the report was that negative values kept appearing in the idle column when running the Race Committee scenario. The idle times are calculated by subtracting the processing time, rework time, and communication time from the total time of the simulation. If an idle time is negative is because the simulation must be over estimating the processing, rework, or communication times.

Execution tracing requires the analyst to follow the line-by-line execution of the model to reveal errors (Balci 1998). A time table of the simulation was developed for the Race Committee scenario showing the log of activities performed by the team leader, which in that particular replication had a negative value in the idle time. The original time table is shown in Figure 22. The analysis of this time table served to discover various remaining bugs in the programming, which are summarized in Table 16 and Table 17. The bugs were solved in various steps and a new time table analysis was developed after each step to verify the bugs were corrected and check for remaining bugs.

| Minute | Task Processing | | Asynch Comm | Synch Comm | | |
|--------|-----------------|--------------------------|-------------|------------|-------------|---|
| | Status | Task | | Status | Other Agent | |
| 0 | Pause | TakeWindDirection | Create | | | Synch Comm Legend: Comm - Synchronous communication requested and accepted ReqComm - Request for an Synch Comm was received and accepted Wait - A synch comm was requested but ignore or denied by the other agent |
| 1 | Process | TakeWindDirection | | | | |
| 2 | Pause | DetermineRaceDirection | Create | | | |
| 3 | Process | DetermineRaceDirection | | | | |
| 4 | Process | DetermineRaceDirection | | | | |
| 5 | Process | DetermineRaceDirection | | | | |
| 6 | Process | DetermineRaceDirection | | | | |
| 7 | Pause | DetermineDistance&Course | Create | | | |
| 8 | Process | DetermineDistance&Course | | | | |
| 9 | Process | DetermineDistance&Course | | | | |
| 10 | Idle | | | Comm | Mark | |
| 11 | Process | GuidePinBoat | Create | Comm | Pin | *Synch Comm at same time than Task Processing |
| 12 | Pause | GuidePinBoat | | | | |
| 13 | Process | GuidePinBoat | | | | |
| 14 | Pause | GuidePinBoat | | Comm | Pin | |
| 15 | Pause | GuidePinBoat | | Comm | Pin | |
| 16 | Process | GuidePinBoat | | | | |
| 17 | Pause | GuidePinBoat | | Comm | Pin | |
| 18 | Process | GuidePinBoat | | | | |
| 19 | Idle | | | ReqComm | Pin | |
| 20 | Idle | | | ReqComm | Mark | |
| 21 | Process | TakeWindDirection2 | Create | Comm | Mark | *Synch Comm at same time than Task Processing |
| 22 | Idle | | | | Comm | Pro |
| 23 | Idle | | | Comm | Signal | |
| 24 | Idle | | | Comm | Signal | |
| 25 | Idle | | | ReqComm | Pin | |
| 26 | Idle | | | ReqComm | Mark | *Two way communication |
| 27 | Idle | | Create | | | |
| 28 | Idle | | | | Comm | Pro |
| 29 | Idle | | | Comm | Signal | *A comm accepted from Mark, and a comm sented to Signal |
| 30 | Idle | | | ReqComm | Mark/Signal | |
| 31 | Idle | | | Comm | Mark | |
| 32 | Idle | | | | | |
| 33 | Idle | | | ReqComm | Mark | |
| 34 | Pause | DecideToResetCourse | Create | | | |
| 35 | Pause | DecideToResetCourse | | | ReqComm | Mark |
| 36 | Process | DecideToResetCourse | | | | |
| 37 | Idle | | | Comm | Signal | |
| 38 | Idle | | | Comm | Mark | |
| 39 | Idle | | | Comm | Pro | *Self message |
| 40 | Idle | | | Comm | Signal | |
| 41 | Idle | | | Wait | Signal | |
| 42 | Idle | | | Comm | Signal | |
| 43 | Idle | | | | | |
| 44 | Idle | | | | | |
| 45 | Idle | | | | | |
| 46 | Idle | | | | | |
| 47 | Idle | | | | | |
| 48 | Idle | | | | | |
| 49 | Idle | | | | | |

Figure 22: Time Table for the Leader Agent of the Race Committee Scenario

Table 16: Problems detected during the verification of the model

| | Problem Description | Causes | Solution |
|---|--|---|---|
| 1 | Negatives in the simulation stats | Some communication activities were occurring at the same time, causing the time to be counted double | The way the communication time is recorded was modified to count the time elapsed doing communication activities instead of the sum up the time of each activity individually |
| 2 | A communication activity perform at the same time as processing a task | When the communication is pick up first there is no way to stop the agent from choosing a task afterward during the same minute | An event was added to the <i>Communicator</i> activity that is trigger by the <i>Task Processor</i> activity every time a new task is started. The event pause the task process if the agent is performing a communication task |
| 3 | Synchronous communication at the same time as an asynchronous communication | sometimes the model may allow the preemption of asynchronous communication preparation for urgent synch messages | Changes were made to the conditions, in the <i>RespondToSynchRequest</i> event in the <i>DecisionMaker</i> activity, to ensure that no synchronous activity is considered while another communication activity is being setup |
| 4 | Self Synchronous communication by the leader | Occurs when sending the <i>NoTaskToProcess</i> messages | Condition added in the <i>ActionCheckEnd</i> event to avoid adding self messages to the send list |
| 5 | One synchronous request sent at the same time that another one was accepted by the agent | The model allows two-ways synchronous communications if established between same pair of agents (see problem 7) | Same solution than in problem 2 with the exception of two ways communications between same agents |
| 6 | A synchronous message sent to a <i>null</i> agent | Occurs when: message = <i>PerformTask</i> , task = <i>ReqNotMet</i> | A condition was added to the <i>AddToSendList</i> event in the <i>DecisionMaker</i> activity that not allow a message to be added to the send list with a <i>null</i> Message receiver |
| 7 | Asynchronous communication preparation that is suppose to be preempted, is not being so, therefore the communication time is count double. | | The mechanism was in place, but the event was not identified in the logs. |
| 8 | Two ways synchronous communication time count double | | The way to collect statistics by the <i>Communicator</i> activity was re-design |
| 9 | Two synch communications accepted at the same time | The <i>currentCalling</i> variable was not being actualize correctly | Correct actualization of <i>CurrentCalling</i> variable in the <i>DecisionMaker</i> activity |

Table 17: Problems detected during the verification of the model (Continuation)

| | Problem Description | Causes | Solution |
|----|---|--|--|
| 10 | Synchronous communication at the same time as an asynchronous communication | Problem when re-sending synch messages | Correct the <i>ProcessSynchRequest</i> event in the <i>DecisionMaker</i> activity to avoid the coordination status to change to 0 when a agent is re-sending a synch message |
| 12 | Two synch communications accepted at the same time | Same cause than problem 10 | Same solution than problem 10 |
| 13 | Two Synch request sent at the same time | Same cause than problem 10 | Same solution than problem 10 |
| 14 | Rework minutes not being correctly count | | The whole method of collecting stats was modified in the <i>TaskProcessor</i> activity |
| 15 | <i>ReqTaskStatus</i> messages for task <i>ReqNotMet</i> | | When a requirement is <i>ReqNotMet</i> is not added to the Missing Requirements |
| 16 | Attempting to set a timer after an activity is dead | | Terminate activities first |

4.3.4 Internal Validity

The internal validity test determines if the stochastic variability present in a simulation model does not affect the consistency of its results (Sargent 2007). The lack of consistency in the model results resulting from its stochastic variability would make the model results questionable.

Both scenarios, Race Committee and IMT, are used to test the internal validity of the simulation model. Each scenario is run using two different sets of random number seeds. The means of the completion times and coordination loads obtained from each set of random number seeds are compared using a two-sided t-test (assuming equal but unknown variances). The research hypothesis is that different random numbers do not produce different distributions of results. The test hypotheses are stated as follow:

H_{0a} : Mean Completion Time_{random set #1} = Mean Completion Time_{random set #2}

H_{0b} : Mean Coordination Load_{random set #1} = Mean Coordination Load_{random set #2}

In both cases, the alternative hypothesis is that the means are not equal. The results are shown in Table 18 and Table 19.

Table 18: Results for the Internal Validity test

| Race Committee Scenario | | | | |
|------------------------------------|------------------------|---------------|--------------------------|---------------|
| | Completion Time | | Coordination Load | |
| | Set #1 | Set #2 | Set #1 | Set #2 |
| Average | 66.2 | 67.2 | 60.43% | 62.83% |
| Standard Deviation | 11.35 | 10.58 | 5.6% | 5.3% |
| N | 15 | 15 | 15 | 15 |
| S²_{n-1} | 1803.52 | 1567.11 | 439.04 | 393.26 |
| v | 28 | | 28 | |
| T | -0.2496 | | -1.2055 | |
| t_{v,α/2} | 2.084 | | 2.084 | |
| Result | Do not Reject | | Do not Reject | |
| P-value | 0.8047 | | 0.2381 | |

Table 19: Results for the Internal Validity test

| IMT Scenario | | | | |
|------------------------------------|------------------------|---------------|--------------------------|---------------|
| | Completion Time | | Coordination Load | |
| | Set #1 | Set #2 | Set #1 | Set #2 |
| Average | 1488.4 | 1576 | 54.49% | 56.11% |
| Standard Deviation | 161.52 | 146.62 | 3.2% | 4.7% |
| N | 10 | 9 | 10 | 9 |
| S²_{n-1} | 234,798 | 171,979 | 90.44 | 177.47 |
| V | 17 | | 17 | |
| T | -1.2325 | | -0.8882 | |
| t_{v,α/2} | 2.1098 | | 2.1098 | |
| Result | Do not Reject | | Do not Reject | |
| P-value | 0.2345 | | 0.3869 | |

None of the test performed show statistical evidence to reject that the two set of random number seeds produce different completion times or coordination loads for the same scenario. These results are positive toward the verification and validation of the TCM, but not conclusive since the tests do not prove conclusively that the means are equal.

One further analysis to assess if these results are evidence that the research hypothesis is true is to examine β , the probability of failing to reject H_0 when it is not true. The β of a statistical test is not easy to calculate, but we can assume to be high for this tests since the sample sizes are small. This implies that these tests do not have much discriminating power to assess differences in the means.

A sensitivity test was performed using the IMT case to assess the discriminating power of the simulation results to detect differences in the results, particularly the completion times (CT). Ten replications of six scenarios based on the IMT case were run. The only difference between the scenarios is that all the task duration distribution parameters were changed by the same percentage with respect to the original parameters. The variations were made from reducing the parameters 10% to increasing the parameters 15%, in 5% increments. Table 20 shows the results of the replications for each scenario.

One-tail t-tests were performed to test the following hypothesis for all scenarios i and j differing in the task durations by 5%, 10%, and 15%:

$$H_0: CT_{\text{scenario } i} = CT_{\text{scenario } j}$$

$$H_1: CT_{\text{scenario } i} < CT_{\text{scenario } j}$$

Assuming that a difference in the task duration distribution parameters should produce similar differences in the completion times, then all the H_0 in the tests are false.

Hence, the H_0 rejections provide an idea of the discriminating power of the model. Table 20 shows the p-values for the t-tests.

Table 20: Results from Sensitivity Analysis to Variations in Task Durations

| Run # | Completion Times (minutes) | | | | | |
|-------------------------------|----------------------------|----------------|-------------------|----------------|----------------|----------------|
| | 5% Decrease | 10% Decrease | Original Scenario | 5% Increase | 10% Increase | 15% Increase |
| 1 | 1520 | 1445 | 1525 | 1587 | 1599 | 1546 |
| 2 | 1153 | 1661 | 1501 | 1740 | 1614 | 1607 |
| 3 | 1280 | 1504 | 1382 | 1705 | 1498 | 1655 |
| 4 | 1521 | 1371 | 1135 | 1437 | 1581 | 1628 |
| 5 | 1680 | 1509 | 1406 | 1409 | 1628 | 1787 |
| 6 | 1574 | 1302 | 1612 | 1623 | 1641 | 1650 |
| 7 | 1387 | 1595 | 1647 | 1644 | 1615 | 1655 |
| 8 | 1553 | 1122 | 1591 | 1533 | 1604 | 1684 |
| 9 | 1214 | 1335 | 1201 | 1607 | 1576 | 1462 |
| 10 | 1270 | 1467 | 1853 | 1669 | 1556 | 1692 |
| <i>Average</i> | <i>1415.20</i> | <i>1431.10</i> | <i>1485.30</i> | <i>1595.40</i> | <i>1591.20</i> | <i>1636.60</i> |
| <i>Std. Dev.</i> | <i>178.23</i> | <i>155.51</i> | <i>213.67</i> | <i>108.22</i> | <i>41.37</i> | <i>87.20</i> |
| <i>p-value 5% difference</i> | <i>0.4170</i> | <i>0.2628</i> | <i>0.0846</i> | <i>0.4553</i> | <i>0.0805</i> | |
| <i>p-value 10% difference</i> | <i>0.2182</i> | <i>0.0072</i> | <i>0.0780</i> | <i>0.1807</i> | | |
| <i>p-value 15% difference</i> | <i>0.0078</i> | <i>0.0050</i> | <i>0.0303</i> | | | |

All the tests between scenarios with 15% difference in tasks duration resulted with p-values below the 5% significance level. Only one of four tests between scenarios with 10% difference in task duration resulted with a p-value below the 5% significance level, and one resulted in a p-value below the 10% significance level. Only two of the five tests between scenarios with 5% difference in task duration resulted with p-values below the 10% significance level, and none of the p-values felt below the 5% significance level. These results place the discriminating power of the model around the 10% difference for the completion time.

Combining the results from the sensitivity analysis with the results from the internal validity, this author concludes that the stochastic variation in the TCM does not produce a difference in the results, at least in the completion times, larger than the 10%. If the differences in the means due to the random numbers would be larger than 10%, the p-values of the internal validity tests would have been close or lower than the 10% significance level. All the p-values in the validity tests are greater than 0.20.

4.4 Model Validation

4.4.1 Comparison with Actual System

The comparison analysis used the results of 50 replications of the Race Committee scenario. Table 21 summarizes the results for this test. The results are compared with some statistics taken from the real system by a member of the real team.

The mean completion time resulted from the simulation is 21.2 minutes higher (46.1%) than the mean completion time measured when the real team performed the job. This difference should be analyzed using other statistics to provide a real sense of how well the simulation represents the real system.

First, the job has three decisions that change the structure of the job: *if the course should be reset, if the starting line should be reset* (occurring around 40% of the times the course is reset), and *if start flag should be recall*. A two-side hypothesis test was performed to determine if the percentages from the simulation are equal to the input percentages. The null and alternate hypotheses are:

$$H_0: p_{\text{simulation}} = p_{\text{input}} \quad \text{and} \quad H_1: p_{\text{simulation}} \neq p_{\text{input}}$$

Table 21: Comparison with Actual System Results

| Statistic | Real System Value | Simulation Results (50 replications) | | |
|---|-------------------|--------------------------------------|--|-------------------------|
| | | Average | Standard Deviation | 90% Confidence Interval |
| Completion Time | 46 minutes | 67.2 | 12.16 | (63.7,70.7) |
| Total Processing Time | (45,66,94) | 64.12 | 7.67 | (61.94, 66.30) |
| Coordination Load | Not available | 61.7% | 5.15% | (60.23%, 63.15%) |
| Leader percentage of Communication time | 50 a 70% | 58.6% | 6.82% | (56.6%, 60.5%) |
| Number of Communications between members | Not available | 113.5 | 24.8 | (106,120.5) |
| % of Replications which require Reset Course | 27% | 26% | Do not reject Ho: Average Simulation = Average from system (p-value = 0.873) | |
| % of Replications which require Reset Starting Line | 11% | 14% | Do not reject Ho: Average Simulation = Average from system (p-value = 0.497) | |
| % of Replications which require Recall Flag | 15% | 18% | Do not reject Ho: Average Simulation = Average from system (p-value = 0.552) | |

All the tests, results shown in Table 21, resulted in do not reject the null hypothesis with p-values larger than 0.49. Therefore, there is no statistical evidence to reject that the percentages estimated by the simulation are different than the percentages the tasks are required on the actual job. Since the actual system's values are input of the simulation, these results just verify that the model is simulating correctly the dynamic aspects of the job structures.

The total processing time is the time spent by all the team members performing tasks of the job. This time could be higher than the completion time since multiple tasks could be executed simultaneously reducing the time the team finishes the job. Since the processing times of the tasks are defined as random variables following a triangular distribution (with a minimum value, a most likely value, and a maximum value), the average total processing time of the simulation should be close to the aggregate most

likely value of the tasks. The aggregate minimum, most likely, and maximum values are calculated considering that some tasks are not required 100% of the time the job is performed. When comparing the average processing time from the simulation, 64.12 minutes, is close to the expected aggregated most likely value of the tasks processing times, which is 66 minutes. This verifies that the model is generating correctly the task processing times, and the team member agents are taking the correct amount of time to process the tasks.

Since the model is simulating well the dynamic job structure and the processing of tasks by the team members, the deviation in the completion times should be caused by the communication activities of the team member agents. There is no data in terms of the coordination load for the whole Race Committee Team, but a rough estimate of the percentage of time the leader spent communicating with teammates is between 50% and 70% of the time. The simulation yields a 58.6% average which is close to the value of the actual system.

A possible source of the deviation in the completion time could be the duration of the communications. The CybelePro infrastructure do not allow activities duration of less than one time unit when using a discrete clock. The Race Committee's communication media is walkie-talkies, and the duration of the communications between members usually last less than one minute. Since the model has been using the *minute* as the time unit, the simulation is adding some fractions of a minute in excess to each communication. An alternative approach is to define the time units as fractions of minutes. The 50 runs were repeated using 0.25 of a minute as the time unit and adjusting all the time distributions parameters appropriately; the distributions for the

communication durations were setup at half of their previous value. Results are shown on Table 22. These results yield an average completion time of 50.4 minutes, still statistically larger than the real team average at a 10% significance level. However, the average completion time is 24.5% smaller than the average from previous sample, although the total processing time increases by 10.9%.

Table 22: Comparison with Actual System Results with adjusted time units

| Statistic | Real System Value | Simulation Results (50 replications) | | |
|---|-------------------|--------------------------------------|--|-------------------------|
| | | Average | Standard Deviation | 90% Confidence Interval |
| Completion Time | 46 minutes | 50.35 minutes | 9.85 min | (48.0, 53.6) |
| Total Processing Time | (45,66,94) | 71.1 min | 8.18 min | (68.8, 73.4) |
| Coordination Load | Not Available | 34.5% | 3.8% | (33.4%, 35.6%) |
| Leader percentage of Communication time | 50 a 70% | 38.4% | 4.8% | (37.0%, 39.7%) |
| Number of Communications between members | Not available | 125 | 27.8 | (117.2, 132.8) |
| % of Replications which require Reset Course | 27% | 35.3% | Do not reject Ho: Average Simulation = Average from system (p-value =0.182) | |
| % of Replications which require Reset Starting Line | 11% | 11.8% | Do not reject Ho: Average Simulation = Average from system (p-value = 0.861) | |
| % of Replications which require Recall Flag | 15% | 9.8% | Do not reject Ho: Average Simulation = Average from system (p-value = 0.299) | |

Another possible source of discrepancies between the simulated results and the real team results is that the model assumes that a team member stops momentarily the execution of a task while executing a communication time. In the actual scenario, it is observed that a race committee member can and frequently does do a task while communicating. The model does not allow for multi-tasking. Thus, in tasks where multi-tasking is common the model will over-estimate the duration.

4.4.2 Comparison with Literature Results

The knowledge about team dynamic available in the existing literature was used to examine the simulation results. Table 23 summarizes the effects and relations examined for the TCM.

Table 23: Parameters Tested for Expected Effects

| Parameters | Expected Effect | Reference |
|--|---|---------------------------|
| Task Assignment Plan | The availability of a task assignment plan should reduce the amount of communication required | Espinosa et al. (2004) |
| Formalization & Centralization vs. Job interdependency | Low formalization and low centralization works best for jobs with high task interdependence, while high formalization and high centralization works best for jobs with low task interdependence | Andres and Zmud (2002) |
| Task Complexity | Task complexity influence the probability of individuals making mistakes while executing a task | Christiansen et al.(1999) |

The Task Assignment Plan (TAP) was evaluated with both the Race Committee and the IMT scenarios. 28 replications without using the TAP were run for the Race Committee scenario. The results from these 28 replications were compared with the results from the 50 replication previously run, in which the TAP was used. The sample with TAP had an average completion time of 67.20 minutes and an average coordination load of 61.7%; while the sample without TAP had an average completion time of 154 minutes (a 129% increase) and an average coordination load of 81.8% (32.5% increase). It is worth to point that the average values for the completion time and the coordination load without using the TAP are much larger than the upper limit of the 90% confidence interval for the averages using the TAP as shown in Table 21. Hence, the absence of a TAP in the model increase the completion time and the amount of coordination required as it was expected from the literature results.

The results using the IMT scenario shown a similar behavior of the model, but were not compelling as the differences for the Race Committee scenario. A sample of 17 replications was run for both, using the TAP and not using it. The sample with TAP had an average completion time of 1521 minutes and an average coordination load of 54.7%; while the sample without the TAP had an average completion time of 1594 minutes (around 5% increase) and an average coordination load of 57.3% (also a 5% increase). The difference in the completion time average is not statistically significant with a p-value of 0.1510 for the null hypothesis that the means are equal (one-tail test). The difference between the coordination load averages is significant for a 10% confidence level (p-value of 0.0625). The results for the effect of the TAP in the model showed that it significantly decrease the amount of coordination required.

According to Andres and Zmud (2002), jobs with high interdependence between tasks benefit by having a coordination strategy with low centralization and low formalization. Jobs with low levels of interdependence benefit from high levels of centralization and formalization. To test the effect of the centralization and formalization on the performance measures, 45 replications were run for each of the three centralization and formalization levels. Figure 23 and Figure 24 show the results for the average completion times and for the coordination loads of both scenarios.

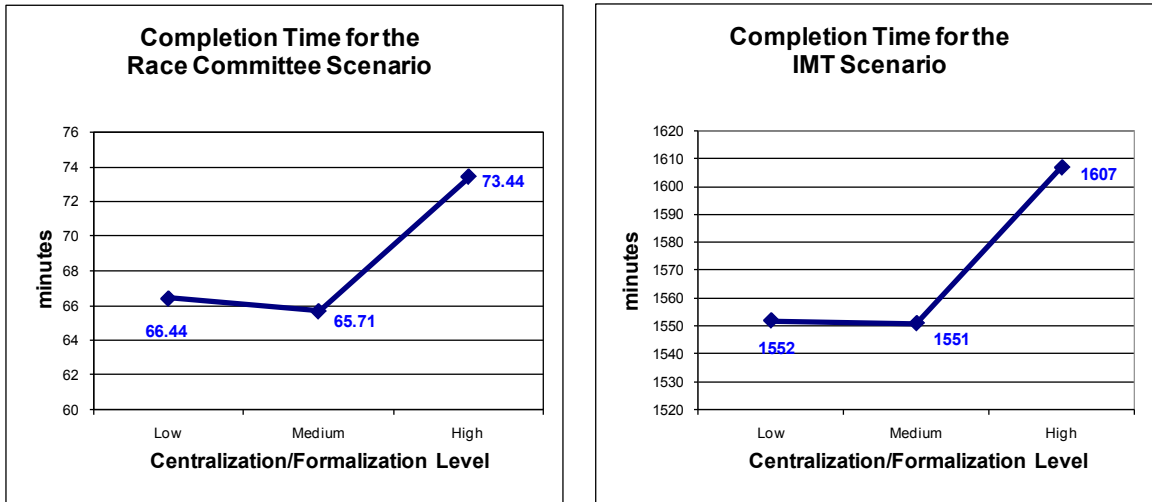


Figure 23: Completion Time vs Centralization/Formalization

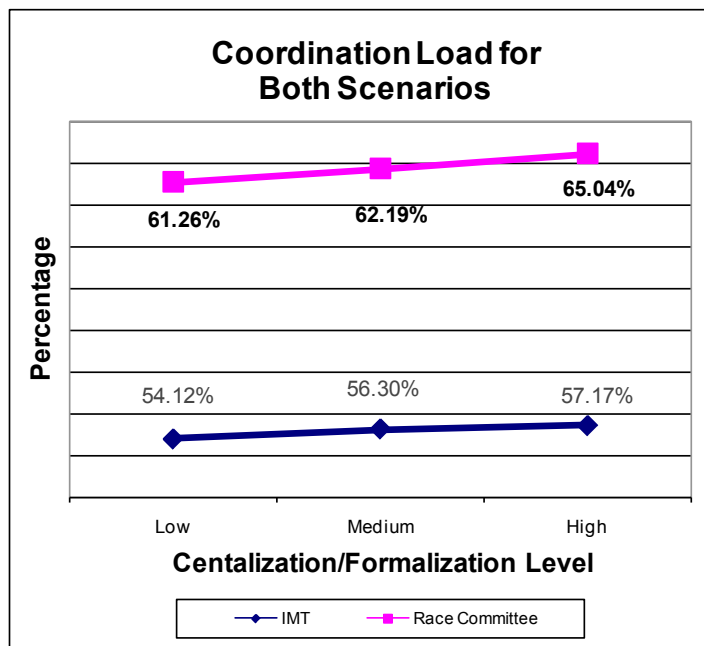


Figure 24: Results of Coordination Load vs. Centralization/Formalization

The effects of the centralization and formalization levels are similar for both scenarios. A one-sided T-test ($\alpha = 0.1$), assuming unequal variances, is used to compare the average of the samples (H_0 : means are equal) for completion times and coordination load. Results are the following:

- No significant difference was observed between the completion times of the low and medium levels (p-value = 0.3950 for Race Committee scenario, p-value = 0.4916 for IMT scenario).
- High levels of centralization and formalization produce a statistically significant increase in the completion times compare with medium levels (p-value = 0.0031 for Race Committee scenario, p-value = 0.0312 for IMT scenario).
- There is a statistically significant increase in the coordination load when the centralization and formalization levels were increase from low to medium (p-value = 0.0209 for Race Committee scenario, p-value = 0.0919 for IMT scenario) and then from medium to high (p-value = 0.0001 for Race Committee scenario, p-value = 0.0258 for IMT scenario).

The results of this test are consistent with the literature, assuming a high degree of interdependency in the jobs, which predicts better results for low levels of formalization and centralization.

Task Complexity affects the probability of make mistakes by individuals executing the task (Christiansen et al. 1999). The model was tested to assess the effect of task complexity while leaving everything else the same. The test evaluated the results of 20 replications of the race committee scenario for each level of task complexity, setting the complexity level of every task on the job to the level being run. The same experiment was repeated with 10 replications of the IMT model per complexity level. A one-sided T-test ($\alpha = 0.1$), assuming unequal variances, was employed to compare the average of

the samples (H_0 : means are equal) for completion times, coordination load, and total rework minutes. Results are the following:

- For the race committee, low and medium task complexity levels do not produced significant differences in completion times (p-value = 0.1618) or coordination load (p-value = 0.3317), but there were a significant difference in total rework minutes (p-value = 0.0106). The IMT scenario produced similar results.
- When changing the task complexity level from medium to high, the Race Committee scenario showed significant increases in all three performance measures (p-value = 0.0059 for completion times, p-value = 0.0004 for coordination load, and p-value = 0.0406 for the total rework minutes). The IMT scenario showed no significant increases in completion times (p-value = 0.3911) and coordination load (p-value = 0.1819), but the results for the total rework minutes was marginally no-significant increase (p-value = 0.1033).

These results show that the total rework minutes increase when the task complexity level is increased in the Team Coordination Model.

4.5 Verification and Validation Conclusion

According to North and Macal (2007), no agent-based model can be guaranteed to be 100% free of coding errors because there are too many scenarios that would need to be tested. The implementation of the Team Coordination Model is no exception.

Nonetheless, the TCM application was subjected to extensive debugging and verification process to test that every function of the simulation works as expected, and the output of the simulation is consistent and correctly calculated. At this point we conclude that, besides the problem with some runs ending in deadlock, the verification process has identified and corrected most of the problems in the TCM. Any implementation bug remaining is not affecting significantly the functioning or the results of the simulation model.

Concerning the deadlock problem, its frequency increases as the complexity of the simulated scenario increases (the problem was not observed with very simple test scenarios). Factors like team size, job size, duration of the communications, and amount of interdependencies seem to increase the likelihood of a run end up in a deadlock. However, for a specific scenario, the runs are affected randomly since none of the stochastic events in the model (results of conditional dependencies, tasks duration, etc.) seems to increase the occurrence of the deadlocks. For example, in section 4.4.1, the observed frequencies for the results of conditional dependencies were consistent with those defined in the job structure. If any of the stochastic events in the job structures would be causing the deadlock, the resulting frequencies of the results would be significantly different than the ones input to the model. Therefore, we can conclude the problem does not affect the validity of the results since the sample of runs represent population. The only setback cause by the deadlocks is that they only oblige to make more runs to obtain the number of samples required.

The validation process shows that the model results seem to be reasonable and the assumptions build up into the model seem to agree with the literature on team. In

conclusion, considering the results of the tests discussed in this chapter, the model shows the capability to simulate adequately a team for the purpose of analyzing it, but has some limitations that might limit predicting accurately the absolute value of a team's performance. For example, the analysis in Section 4.3.4 estimates that the model can distinguish appropriately differences in the completion time of 10% in magnitude. The next chapter discussed the application of the Team Coordination Model to its intended use of designing teams and study team coordination, which could render more insight on how well the model accomplishes the goals of this research.

CHAPTER 5:

EXPERIMENTATION

The Contingency Team Design Methodology proposed in this research consists of four basic steps: gather data about the team and the job, develop the job structure and input the scenario into the TCM, validate the input data and preliminary results, and run a factorial experiment to study the team and recommend the best team design. The application of the first three steps to the Race Committee case was described on the previous chapter. This chapter describes the application of the TCM to the design of the Race Committee team through a design of experiments.

Although the verification and validation process was described in the previous chapter, the experimentation described in this chapter will provide further insight on the adequacy of the TCM to meet the goals of this research. Section 5.1 describes the experimental design, while Section 5.2 discusses the results in the context of the behavior of the model. Section 5.3 discussed the best team design based on the experimental results.

5.1 Design of Experiments

The objective of the experiment is to determine what levels of the team design factors will optimize the performance of the team running the Sailboat Race. It is assumed that the characteristics of the individual Race Committee jobs are difficult to

modify, therefore the experiment included only team design factors. This problem is consistent with the sample formulation shown in Figure 2.

Table 24 shows the six factors chosen for the experiment and their respective operationalization for the factorial experiment. The Race Committee team is basically a centralized team with the Principal Race Officer (PRO) making most of the decisions, including the centralization factor in this experiment will tell if this is the best approach for the team. Formalization was included in the experiment to study the interaction with other factors, but the Race Committee is in nature non-formal. In practice, the Race Committee might have a decision on the other four factors selected: members' experience, members' teamwork skills, team size, and TAP.

Table 24: Operationalization of Experimental Factors

| Factor Name | Factor ID | Low Value | High Value |
|-----------------------------|------------------|--------------------------|---------------------------|
| Team Centralization | A | Low Level Setting | High Level Setting |
| Team Formalization | B | Low Level Setting | High Level Setting |
| Team Member Experience | C | Low Level for all Agents | High Level for all Agents |
| Team Member Teamwork Skills | D | Low Level for all Agents | High Level for all Agents |
| Team Size | E | 4 members team | 6 members team |
| Task Assignment Plan (TAP) | F | NO, TAP not used. | YES, TAP used |

Every other factor or input, including all the tasks characteristics and job structure, are kept constant during the experiment. All the behavioral and communication parameters of the simulation model are also kept constant during the experimentation. The performance measures evaluated during the experiment are the completion time of the job, the coordination load, and the total amount of rework minutes required to complete the job. This last measure is used as a surrogate for the quality of the team job.

The chosen experimental design is a 2^{6-1} fractional factorial design. This design requires half the runs than a full factorial, but allows assessing the effect of the

interaction between each pair of factors. The interactions between four factors were assumed to be insignificant, while the effect of three factors interaction will be confounded with one another. Any significant effect of three-factor interactions was evaluated cautiously to determine which of the confounded interactions is more likely to be significant.

5.2 Experimental Results

Ten replications were run for each of the experimental treatments. The average of the ten replications for each performance measure was the result recorded as the response on each of the treatments; therefore the ANOVA was calculated with just one run per treatment. The multiple replications per treatment help to reduce the effect of the simulation variability due to the randomness in the team members' behavior, task duration, and job structure.

Table 25 shows the experimental design and the results for each experimental run. The results were analyzed using the Minitab software. Figure 25, Figure 30, and Figure 33 show the Pareto of the treatments effects on the completion time, coordination load, and rework time respectively. The effects that extend beyond the reference line are significant at a 10% significance level. The remainder of this section discusses the results of the experiment and analyzes the treatments with the most significant effects. The purpose of this analysis is to gain more insight on the behavior of the simulation model and its adequacy to simulate teams.

Table 25: Results for Each Experimental Treatment

| Experimental | Experimental Factors Levels | | | | | | Completion | Coordination | Rework |
|--------------|-----------------------------|------|------|------|---|-----|------------|--------------|------------|
| Run Number | A | B | C | D | E | F | Time (min) | Load (%) | Time (min) |
| 1 | Low | low | low | low | 4 | NO | 106.70 | 64.21 | 2.60 |
| 2 | high | low | low | low | 4 | YES | 71.00 | 63.39 | 0.60 |
| 3 | Low | high | low | low | 4 | YES | 60.30 | 55.79 | 0.50 |
| 4 | high | high | low | low | 4 | NO | 117.90 | 66.18 | 0.30 |
| 5 | Low | low | high | low | 4 | YES | 59.40 | 56.97 | 1.00 |
| 6 | high | low | high | low | 4 | NO | 90.60 | 58.35 | 1.00 |
| 7 | Low | high | high | low | 4 | NO | 83.00 | 50.54 | 1.08 |
| 8 | high | high | high | low | 4 | YES | 59.40 | 61.13 | 0.30 |
| 9 | Low | low | low | high | 4 | YES | 59.40 | 56.97 | 1.00 |
| 10 | high | low | low | high | 4 | NO | 110.80 | 61.58 | 0.60 |
| 11 | Low | high | low | high | 4 | NO | 86.60 | 54.89 | 1.00 |
| 12 | high | high | low | high | 4 | YES | 59.40 | 61.13 | 0.30 |
| 13 | Low | low | high | high | 4 | NO | 60.25 | 58.50 | 0.67 |
| 14 | high | low | high | high | 4 | YES | 72.10 | 64.84 | 1.90 |
| 15 | Low | high | high | high | 4 | YES | 63.30 | 57.61 | 1.20 |
| 16 | high | high | high | high | 4 | NO | 73.25 | 64.52 | 0.75 |
| 17 | Low | low | low | low | 6 | YES | 77.60 | 66.84 | 0.90 |
| 18 | high | low | low | low | 6 | NO | 129.70 | 81.07 | 1.60 |
| 19 | Low | high | low | low | 6 | NO | 97.40 | 65.18 | 0.60 |
| 20 | high | high | low | low | 6 | YES | 89.00 | 71.50 | 1.40 |
| 21 | Low | low | high | low | 6 | NO | 98.13 | 71.59 | 1.00 |
| 22 | high | low | high | low | 6 | YES | 94.50 | 73.54 | 2.60 |
| 23 | Low | high | high | low | 6 | YES | 74.40 | 62.73 | 1.50 |
| 24 | high | high | high | low | 6 | NO | 124.70 | 80.24 | 0.90 |
| 25 | Low | low | low | high | 6 | NO | 95.50 | 63.79 | 1.20 |
| 26 | high | low | low | high | 6 | YES | 94.50 | 73.55 | 2.60 |
| 27 | Low | high | low | high | 6 | YES | 74.50 | 62.73 | 1.50 |
| 28 | high | high | low | high | 6 | NO | 120.89 | 84.07 | 3.44 |
| 29 | Low | low | high | high | 6 | YES | 69.90 | 63.60 | 0.90 |
| 30 | high | low | high | high | 6 | NO | 101.91 | 74.45 | 2.69 |
| 31 | Low | high | high | high | 6 | NO | 83.25 | 65.61 | 1.25 |
| 32 | high | high | high | high | 6 | YES | 95.80 | 73.62 | 1.90 |

5.2.1 Results for Completion Time

All the experimental factors but Formalization (B) demonstrated a significant effect on the completion time. Also, five interactions end up having a significant effect on the completion time, including one three-factor interaction. Figure 25 summarizes the results for the completion time.

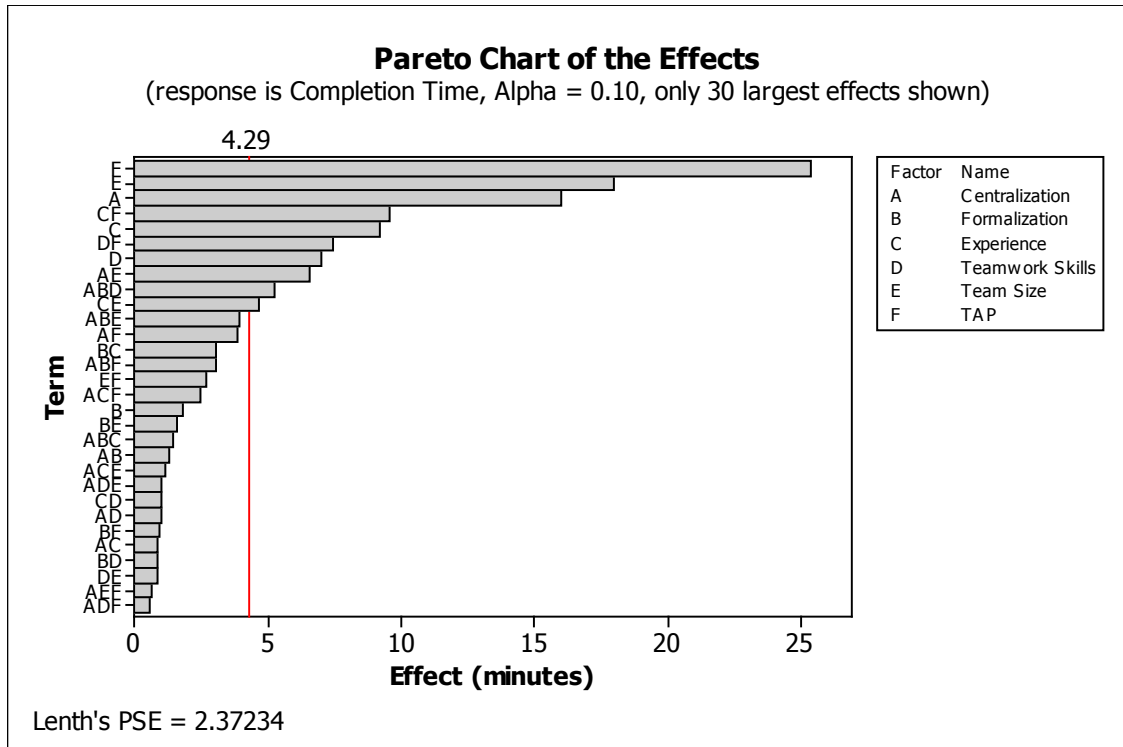


Figure 25: Pareto of Effects for Completion Time

The fact that the formalization factor did not result in a significant effect on the completion time is easily explained due to the factor's effect embedded in the model and the communication parameters of the scenario. In the model, the level of formalization affects the probability of choosing asynchronous versus synchronous communication media. The race scenario requires only short synchronous communication; therefore the influence of the formalization factor on this scenario is minimal.

The TAP factor showed (F) a negative effect on the completion time as expected since a low value (absence of TAP) is expected to increase the amount of coordination required and the completion time (Espinosa et al. 2004). Interesting is the effect of the interaction between the TAP (F) and the team member experience (C), shown in Figure 26, which effect is significant and positive for the completion time. A further analysis of

this effect shows that when the team is using the TAP, the team member experience does not cause a difference on the completion time. However, a high level of team members experience reduces the completion time when a TAP is not used. The same behavior is observed for the interaction between the TAP and teamwork skill level. These results are what you would have expected to occur in the actual scenario, where a team members experience and teamwork skills should be more helpful in the absence of a predetermined action plan.

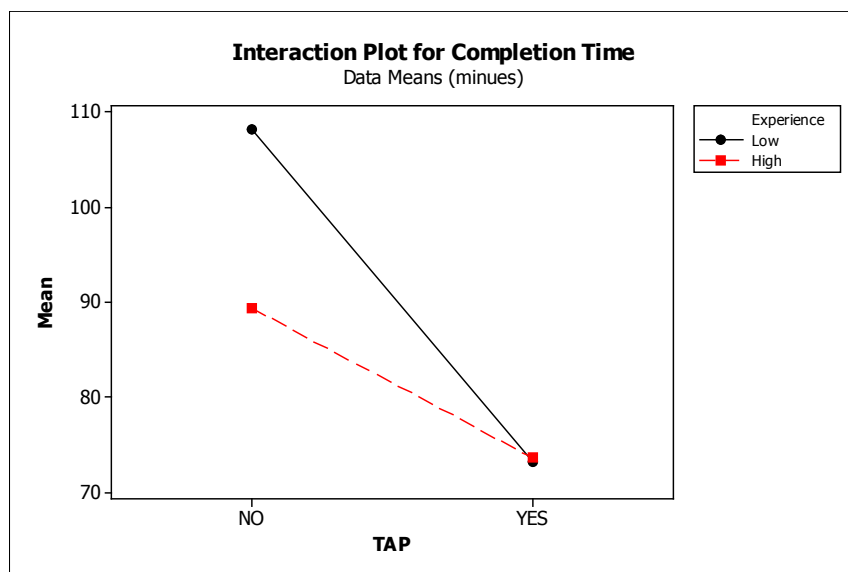


Figure 26: Interaction Plots for Tap vs. Experience

Both, the team size (E) and the centralization level (A) resulted with positive significant effects for the completion time. The team size alone does not yield much insight about the model. The results show that four members are better than six for the Race Committee team, but we don't know if five or three members would be better than four. The centralization behavior was tested and analyzed in section 4.4.2 . The results of this experiment confirm that low levels of centralization reduce the completion time.

Nonetheless, the interaction between team size and centralization, shown in Figure 27, has a significant effect on completion time. The plot for this interaction shows that, although for any size the preferred centralization level is low, the difference is more noticeable when the team size is greater. The model produces this behavior since a high centralization level makes the communication with the leader to be more frequent; therefore a larger number of members will required more communication with the leader, and consequently increase the completion time.

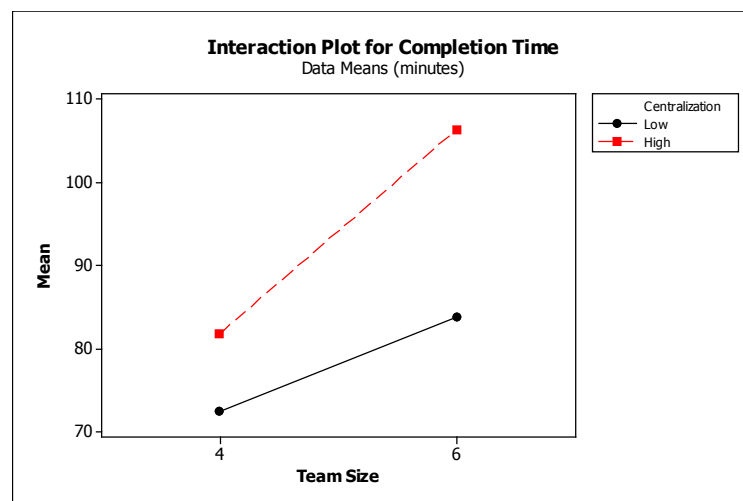


Figure 27: Interaction Plot for Team Size vs. Centralization

Figure 28 shows the interaction between the team size and team experience which presents an interesting result. Although a high experience level produces lower completion times regardless team size, the difference is greater when the team size is smaller. This behavior of the model makes sense since a smaller team has less available time to perform the job tasks, therefore a higher level of experience should translate into a greater efficiency.

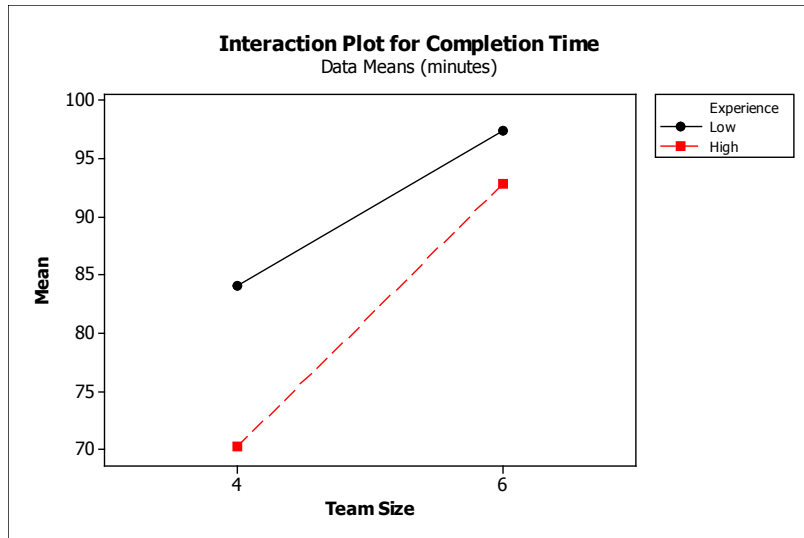


Figure 28: Interaction Plot for Team Size vs. Experience

The ABD interaction also resulted significant, but this interaction is confounded with the CEF interaction, after examine the cube plots for both, the researcher inclines for the later. The cube plot for the CEF is shown in

Figure 29. The results show that difference in completion time caused by the absence of a TAP is more noticeable when the experience level is low and the team size is smaller. However, when the experience level is high, the larger difference in the completion time caused by the absence of TAP is observed for the largest team.

5.2.2 Results for Coordination Load

The results for the coordination load in Figure 30 show that only the team size (E), and the centralization level (A) have a significant effect. These effects are as expected since by increasing the size of a team or the level of centralization you will expect the team to require more communication to coordinate the job. An unexpected result was that the TAP's effect on Coordination Load, although close ($|TAP's\ effect| =$

2.427, effect reference line for 0.10 significance = 2.55), was not significant. It is expected the team to required significantly more coordination if a TAP is not used.

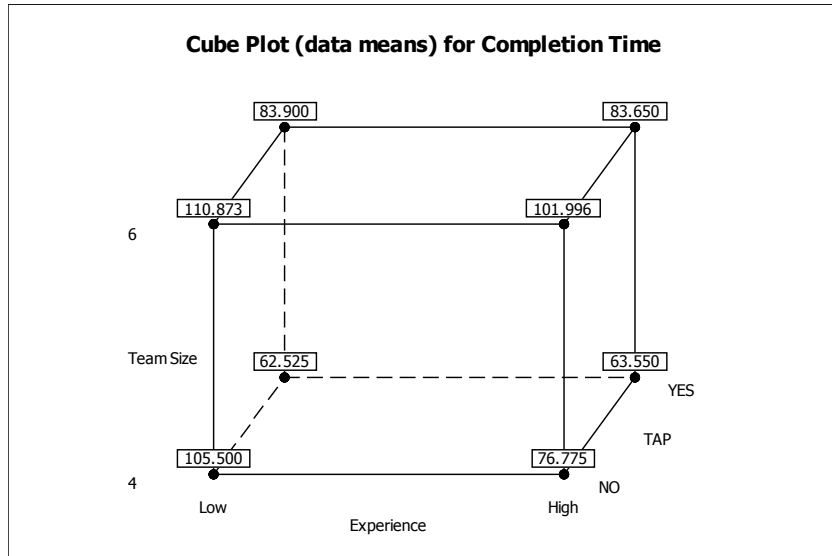


Figure 29: Cube Plot for the ABD Interaction

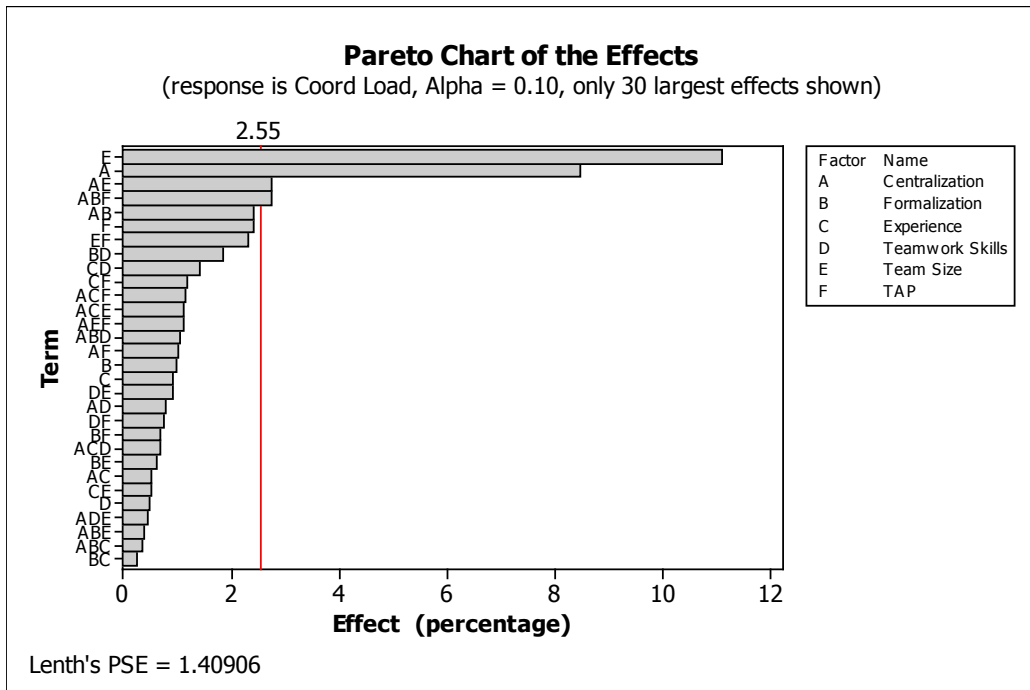


Figure 30: Pareto of Effects for the Coordination Load

Two interactions, including one three-factor interaction, have a significant effect on the coordination load. The interaction between team size (E) and centralization (A) shows that a high level of centralization produces higher percentage of coordination load regardless the team size, but the difference is more noticeable for larger teams. Figure 31 illustrates this interaction. The effect of this interaction on the coordination load is similar than the effect on the Completion time which implies that the output performance (completion time) is influence by the coordination performance (coordination load).

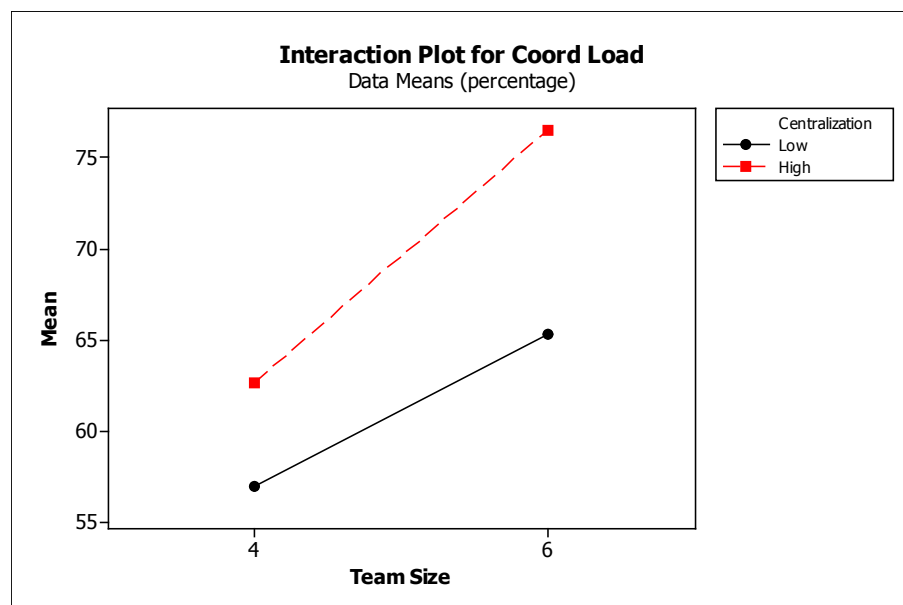


Figure 31: Interaction Plot for Team Size vs. Centralization

Another interesting result is the interaction between the TAP, formalization, and centralization factors (ABF) illustrated in Figure 32. This interaction shows that if the TAP is used, the formalization level should be high regardless the level of centralization. However, if a TAP is not used, the level of formalization should be the opposite of the level of centralization to reduce the coordination load. However, considering that the effect of the formalization should be small and that this interaction might be confound

with the another 3-factor interaction (skill level, teamwork skill level, and team size), this result should be interpret cautiously.

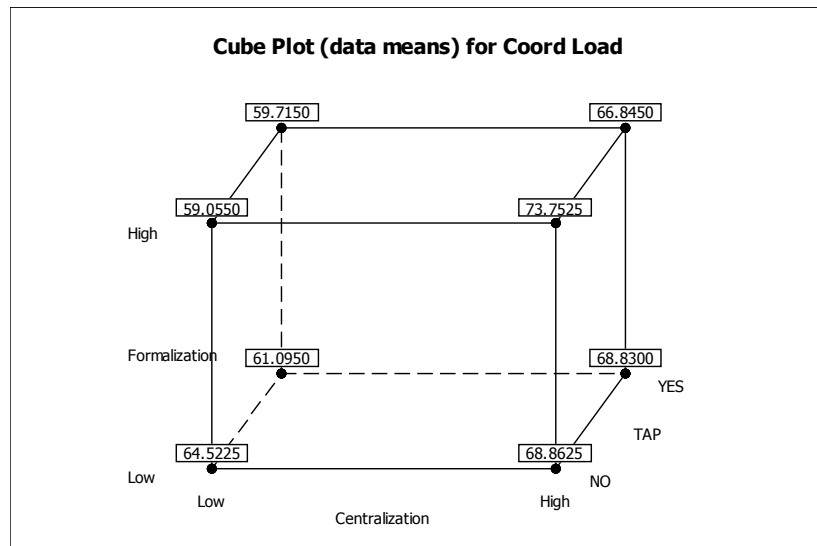


Figure 32: Interaction between TAP, Centralization, and Formalization

5.2.3 Results for Rework Time

The results for the rework time in Figure 33 show that only the team size (E) has a significant impact on team quality by itself. Nonetheless, four interactions, including two three-factor interactions, have a significant effect on rework time.

The interaction between the centralization level (A) and the team size (E) has the largest effect on rework time. The interaction plot on Figure 34 shows that the team size do not make a difference when the centralization level is low, but when the centralization level is high, a larger team size affects the quality of the results. This result is unforeseen since the model did not include a direct effect on the rework time for these factors. One possible explanation is that when the centralization is high, a larger team increases significantly the communications with the leader provoking many interruptions

in the leader task processing. The higher rate of interruptions increases the opportunities for the leader to make mistakes. Since the ABC and the DEF three-factor interactions are confounded, it is difficult to determine which one has a significant effect on the quality.

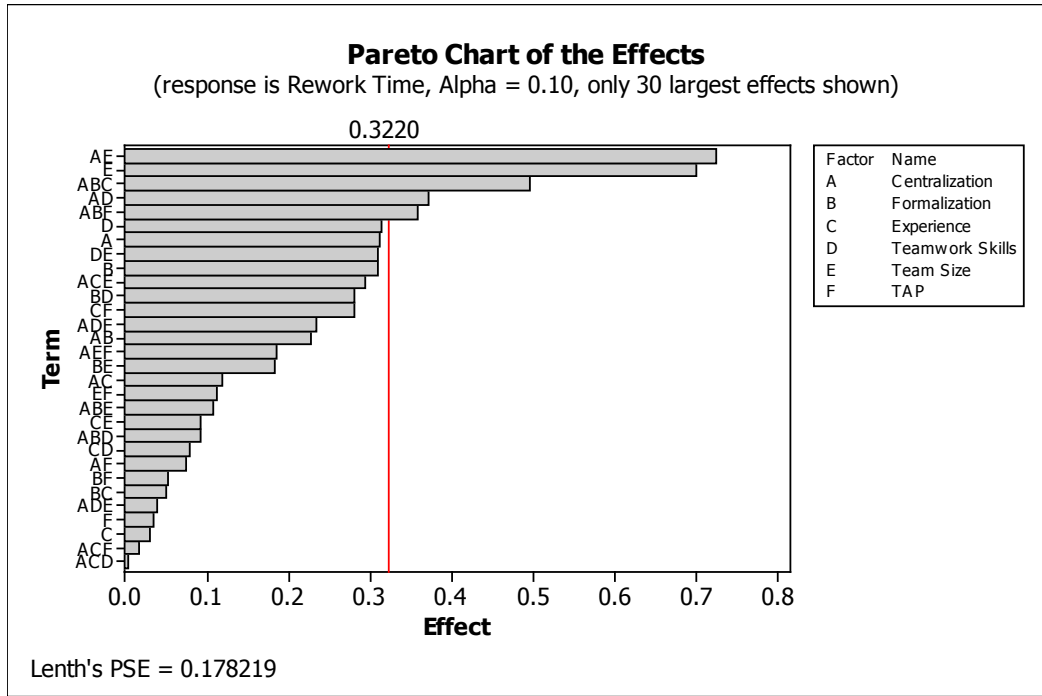


Figure 33: Pareto of Effects for the Rework Time

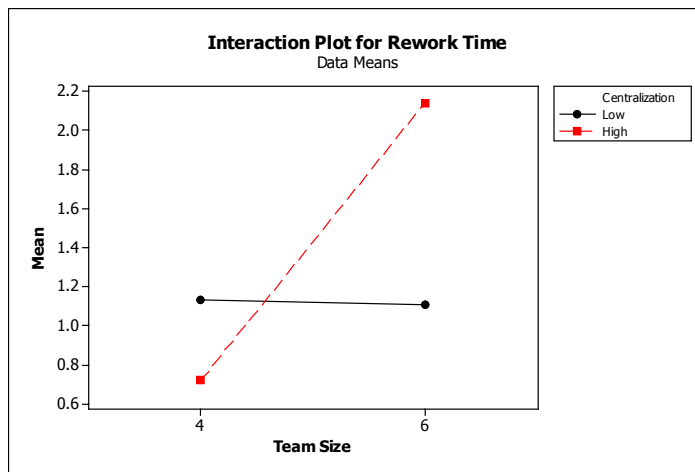


Figure 34: Interaction Plot for Team size vs. Centralization

5.2.4 Final Remarks on the Experimental Results

This experiment shows some interesting results particularly from the interaction between the factors. The behavior of the model resulting from the individual factors was intended and a consequence of the model assumptions. Nonetheless, it is worth to highlight that the behavior resulting from the interaction between the factors, although most are logical from a literature and/or common sense perspective, are a consequences of the execution of the agent-based model.

These interactions between factors demonstrate the contingency nature of the model. For example, the interaction between the TAP and the experience shown that when the team member experience makes a difference when the team do not have a task assignment plan and needs to rely more on coordination. Similarly, the experience is more decisive of the outcome when the team size is smaller. Another example of the contingency of the model is shown by the interaction between formalization, centralization and the TAP. When the TAP is employed by the team, the team benefits from having the centralization and formalization at the same level, but when the TAP is not used, the centralization and formalization levels should be opposites.

5.3 Team Optimization

The objective of the team design analysis is to determine the best combination of design factors that will yield the better performance. The main performance measure for the Race Committee is the completion time since the objective of the team is to set up the race as quickly as possible. Table 26 shows the result of the ANOVA analysis for the

experimental factors with its recommended values to minimize the completion time of the job. Based on these results we can recommend a team with low centralization level, high formalization level, where its members have a high degree of experience and teamwork skills, keeping the size of the team in four members, and using a Task Assignment Plan as a coordination mechanism. These results are expected for each factor, probably with the exception of the formalization level. The formalization factor is included in the recommendations since its p-value of 0.1005 is close to the significance level of 0.10 used in the Anova test and, as discussed in the previous section, some of its interactions were significant.

Table 26: Recommended Values for the Individual Experimental Factors

| Factor Name | Factor ID | Anova P-value | Estimated Effect | Recommended Value |
|-----------------------------|------------------|----------------------|-------------------------|--------------------------|
| Team Centralization | A | < 0.0001 | 7.99 | Low |
| Team Formalization | B | 0.1005 | -0.90 | High |
| Team Member Experience | C | < 0.0001 | - 4. 60 | High |
| Team Member Teamwork Skills | D | < 0.0001 | - 3.51 | High |
| Team Size | E | < 0.0001 | 9.01 | 4 members |
| Task Assignment Plan (TAP) | F | < 0.0001 | -12.69 | YES |

It is necessary to include the effect of the interactions in the analysis before giving the final recommendations about team design. The following analysis uses the model resulting from the ANOVA analysis to determine the optimal values of the experimental factors considering all the significant treatments (including single factors and interactions). The experimental factor B (formalization) and some two-factor interactions were included in the model to make the model hierarchical. The model was optimized as an integer programming model using Excel Solver shown in equations 2 to 4.

(2) *Min Completion Tim*

$$\begin{aligned} &= 86.09 + 7.99A - 0.90B - 4.60C - 3.51D + 9.01 E - 12.69F \\ &- 0.65AB - 0.45AC + 0.50AD + 3.28AE - 1.94AF + 1.55BC \\ &+ 0.44BD + 0.79BE - 0.49BF + 2.32CE + 4.80CF + 3.72DF \\ &+ 1.36EF - 2.64ABD + 1.98ABE - 1.52ABF + 1.24ACF \end{aligned}$$

Subject to:

$$(3) \ A, B, C, D, E, F = \text{integer}$$

$$(4) \ -1 \leq A, B, C, D, E, F \leq 1$$

Where:

A = level of centralization

B = level of formalization

C = level of members experience

D = level of members' teamwork skills

E = Team Size

F = use of TAP

The objective function minimizes the completion time using the estimated coefficients from the ANOVA analysis as the coefficient for each term (significant treatment). The decision variables are the value of the experimental factors. The only constraints are that the value of the decision variables should either -1 or 1.

The results from the optimization, shown in Table 27, recommend values for the experimental factors that are the same than the ones shown in Table 26 with the exception of the formalization factor which should have a Low level. This difference was cause because the effect of the interactions containing the formalization factor is

larger than the effect of the factor by itself. The expected completion time of the job using this design for the Race Committee team is 50.4 minutes.

Table 27: Optimization of the Completion Time

| Treatment | Coefficient Estimate | Value of Experimental Factors | | | | | | Treatment Value | Resulting Effect |
|---------------------|----------------------|-------------------------------|----|----|---|---|---|-----------------|------------------|
| | | A | B | C | D | E | F | | |
| Intercept | 86.0961 | | | | | | | 1 | 86.0961 |
| A-Centralization | 7.9945 | | -1 | | | | | -1 | -7.9945 |
| B-Formalization | -0.9030 | | | -1 | | | | -1 | 0.9030 |
| C-Member Experience | -4.6033 | | | | 1 | | | 1 | -4.6033 |
| D-Teamwork Skill | -3.5117 | | | | | 1 | | 1 | -3.5117 |
| E-Size | 9.0086 | | | | | | 1 | -1 | -9.0086 |
| F-TAP | -12.6898 | | | | | | | 1 | -12.6898 |
| AB | -0.6452 | | -1 | -1 | | | | 1 | -0.6452 |
| AC | -0.4548 | | -1 | | 1 | | | -1 | 0.4548 |
| AD | 0.5023 | | -1 | | | 1 | | -1 | -0.5023 |
| AE | 3.2758 | | -1 | | | | 1 | -1 | 3.2758 |
| AF | -1.9383 | | -1 | | | | | 1 | 1.9383 |
| BC | 1.5477 | | | -1 | 1 | | | -1 | -1.5477 |
| BD | 0.4423 | | | -1 | | 1 | | -1 | -0.4423 |
| BE | 0.7908 | | | -1 | | | 1 | -1 | 0.7908 |
| BF | -0.4908 | | | -1 | | | | 1 | 0.4908 |
| CE | 2.3217 | | | | 1 | | 1 | -1 | -2.3217 |
| CF | 4.7970 | | | | 1 | | | 1 | 4.7970 |
| DF | 3.7180 | | | | | 1 | | 1 | 3.7180 |
| EF | 1.3602 | | | | | | 1 | -1 | -1.3602 |
| ABD | -2.6405 | | -1 | -1 | | 1 | | 1 | -2.6405 |
| ABE | 1.9798 | | -1 | -1 | | | 1 | -1 | -1.9798 |
| ABF | -1.5236 | | -1 | -1 | | | | 1 | -1.5236 |
| ACF | 1.2486 | | -1 | | 1 | | | 1 | -1.2486 |
| Total | | | | | | | | | 50.4447 |

A similar analysis was made to optimize the rework time. The results from this analysis shown that the optimal design for a team is to have high centralization and formalization levels, high degree of experience and teamwork skills, a team size of 4 members; the TAP resulted irrelevant for the quality measure. The resulting expected rework time is 0.25 minutes. The values of the centralization and formalization recommended to minimize the rework time contrast with the values recommended to minimize the completion time, which indicate a conflict in the objectives.

The same analysis was made using the response optimizer from Minitab, but optimizing the three response variables at the same time. This analysis requires targets for each response variables, which were set at 50 minutes for the completion time, 50% for

the coordination load, and 0.25 minutes for the rework time. The recommended levels for the factors were the same obtained previously to minimize the completion time.

In conclusion, this chapter demonstrates the application of the Team Coordination Model to the design teams. The result of the analysis is the combination of values for each of the design factors that likely will produce the best performance by the team. The results of this chapter also demonstrate the contingency nature of the TCM.

CHAPTER 6:

CONCLUSIONS

This chapter summarizes the contribution of this research effort and points directions to complement or continue it. The chapter starts with an overview of the research motivation and goals, an overview of the Team Coordination Model's noteworthy characteristics, followed by with a summary of the research contributions. Finally, the chapter addresses this research limitations and future research directions.

6.1 Summary of Research

This research created the *Team Coordination Model* (TCM), an agent-based model of teams performing jobs with a stochastic structure, and developed the *Contingency Team Design Methodology* (CTDM), a methodology to apply the model to the design of teams. The TCM estimates the coordination load and performance of a team based on its composition, coordination mechanisms, and job's structural characteristics. The CTDM uses the TCM to execute a factorial design of experiments in order to determine the team design characteristics that most likely lead the team to achieve optimal performance.

Model conceptualization was the first step toward the development of the agent-based, discrete-event simulation model. The conceptual TCM, shown in Figure 35, summarizes the contingency relation between the design factors and the performance measures. During this conceptualization phase, this research developed other models,

such as the job structure model and the communications taxonomy, that were significant in the development of the TCM. These models were discussed in Chapter 3.

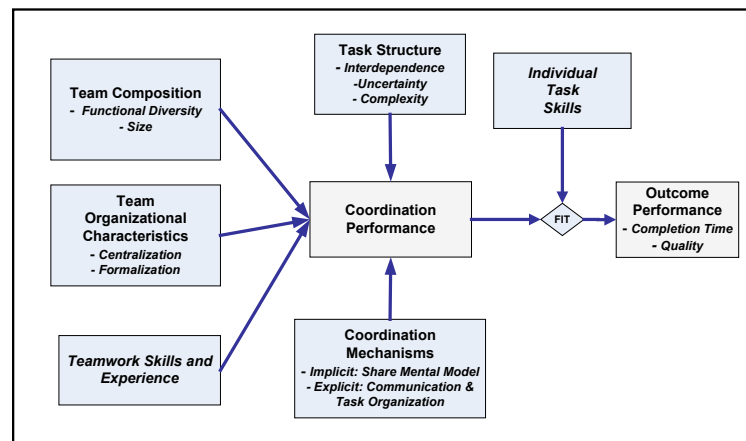


Figure 35: Conceptual Team Coordination Model

The TCM was implemented as an agent-based, discrete-event simulation model. The simulation application is programmed in Java, while Cybele Pro provides the architecture for the creation of the agents and the communications between them. The TCM provides to the team member agents with the capabilities to communicate synchronously and asynchronously, and to make decisions regarding the task and coordination activities. The TCM also implements the concept of share mental model as a mechanism of implicit coordination.

This research developed two main scenarios to verify and validate the TCM. One of the scenarios is based on a team of university officials managing a hurricane emergency. The second scenario is based on a committee that runs sailboat races. The verification and validation phase of the research include several tests that compares the simulation results with actual team results and with the results predicted by the team performance literature.

A 2^{6-1} fractional factorial design of experiments was designed and run with the purpose of determining the best team configuration for the team in the Race Committee scenario. This research analyzed the effect on the output performance measures of the individual experimental factors and their interactions. The results show consistency with the team coordination and performance literature. The experiment ended with recommended design for the Race Committee team. This experiment culminated the validation of the TCM and completed the application of the TCM and the CTDM to analyzed and design teams.

6.2 Research Contributions

The TCM is the main contribution of this research to the team modeling and simulation literature. Teamwork simulation models can be divided in two main categories according to the purpose of the model: one that seek to develop artificial intelligence agent to interact with or substitute for human teams, and the second type of models seek to simulate human teams with the purpose of analyzing or improving them. The development of models for the second category, although gaining importance, is still in its infancy (Salas et al. 2005). Current models in this category lack the capability of simulate teams working on complex job environments that cannot be model by a static structure of tasks and dependencies. The models that address this complexity to some extent (Yilmaz and Philips 2007; Dong and Hu 2008) are limited in their focus of applicability. This research provides an agent-based model capable to simulate a wide variety of teams working in stochastic job environments. The TCM is capable of

simulate a wider set of teams than current models when the purpose of the simulation is to analyze and improve team performance.

Another important contribution of this work is the job structure model implemented in the TCM. This research borrows the use of an activity network to model the job structure from the project management literature and previous team modeling research (Jin et al. 1995; Kunz et al. 1998). However, job models found on the literature includes probabilistic tasks durations, but not stochastic job structures.

The activity network is composed of tasks (nodes) and their dependencies (arcs), and uses the dependencies types defined by Giachetti (2006). Additionally, this research defined three new types of dependencies between tasks: conditional dependencies, single conditional dependencies, and merge dependencies. These types of dependencies provide the capability of modeling stochastic job structures. The stochastic job structure model allows more flexibility on the teams that can be analyzed utilizing the TCM and the CTDM.

This research borrows the symbols used to represent the job structures from the Specification and Description Language (SDL) (IEC 2007). Contrary to other type of diagrams used to describe the flow of processes, the SDL is suitable to represent stochastic job structures and to document graphically the processes implemented in an agent-based application.

The taxonomy of communications events, discussed in Section 3.2.2, was significant in the development of the TCM. This taxonomy simplifies the classification and processing of messages between agents implemented in the simulation. Messages in

the same category are processed by the same function of the team member agents; this improve the tractability and the scalability of the simulation model.

Another noteworthy characteristic of the TCM is the simple-to-define data model required to build a scenario. The data required to model teams and their respective jobs is numerous, and frequently the data available to the team analyst is scarce since the team under study might not be assembled on regular basis. This research takes these issues in consideration and the TCM makes easier for the analyst to gather the required data. For example, most of the parameters are either binary or to be set at one of three levels (low, medium, high). This allow the analyst to assess the levels of these parameters, such as team member skills and experience on a qualitative basis, rather than quantitative. The analyst still need to define task duration, but the TCM uses a triangular distribution. The triangular distribution allows the analyst to model the duration in absent of data by defining the minimum, most likely, and maximum duration for the tasks.

6.3 Limitations and Future Work

This research creates the TCM to simulate a wider set of teams than other teamwork simulation models. Nonetheless, the applicability of the TCM has limitations. First, although the stochastic job structure allows complex jobs to be modeled, the structure stills needs to be well-defined. This means that all possible tasks, dependencies, and conditions should be known by the analyst. Second, the model does not consider possible delays in task execution or coordination activities caused by other team member obligations not related to the job being performed by the team. Therefore, the model applies to teams which team members are completely focus on the job at hand or at least

the team members have the team work as top priority. Finally, the model assumes that all synchronous communication occurs between two individual team members. Therefore coordination activities such as meetings or conference calls are not supported by the model.

The last two limitations should be overcome in a future version of the model. Future expansions of this research should include as an input factor the percentage of time each team member dedicates to the team job. The percentage might be used to determine at any given point during the simulation the probability the team member is busy with other responsibilities and the magnitude of the delay. The simulation application interface already includes the capability of defining other type of communication types such as meetings, but its implementation was delayed to a future version due to time constraints.

Another feature that will expand the applicability of the TCM is the capability to simulate teams working on more than one job at the same time. The implications of defining various separate jobs versus defining one comprehensive job is that the starting and ending times could be kept separated. For example, a team can start working on a job and, after specific amount of time, receives another job to work on. In this case, the team will have to consider two separate due dates which might alter its priorities. Another implication of having two different jobs defined is that the performance measures corresponding to each job will be collected separately. Marks et al. (2001) argued that many teams work on multiple goals at the same time, and researchers should take this in consideration when studying team effectiveness. The current implementation

of the TCM allows the definition and execution of more than one model, but this feature was not verified nor validated.

Another opportunity to improve the TCM is to expand the number of factors that affect team performance included in the model. A review of the team modeling and team performance literature is part of this research. Most of the factors that have an effect on coordination and team performance are included in the TCM, either explicitly as an input factor or implicitly in the model processes. However, the list of factors considered in the TCM is not exhaustive. The most comprehensive list of factors that influence team performance has been compiled by Salas et al. (2005), whom identified 29 important factors that influence team performance. These authors categorized each factor in three categories: “must be modeled factors”, “should be modeled” factors, and “would like to model” factors. The TCM considers to some extent 20 of these 29 factors.

Building a non-specialized agent-based simulation model that implements all of the 29 factors described by Salas et al. (2005) will be unpractical. North and Macal (2007) recommend an incremental approach to build agent-based applications. Increasing the complexity of the model increases the time and cost of development the agent-based application, and might cause the model to become intractable. Nonetheless, there are opportunities to improve the TCM. For example, two key assumptions of the model are that the motivation and attitudes of the individual team members are good, and the members always work to complete their tasks and make decisions with the intention of improving team performance. These assumptions are not necessarily true in many teams.

This research identified as a problem the limitations of traditional research methods to develop encompassing theories to advance the study of team coordination and performance. This dissertation argues that computational tools, particularly agent-based simulation models, have the potential to overcome those limitations as Kim and Burton (2002) demonstrate to some degree using the VDT model. The TCM represents a step toward solving this problem. However, a simulation model requires extensive validation, particularly comparing its results with the results of many actual human teams, before it could be used to test comprehensive theories about team performance. This validation process requires a long term research effort. This dissertation could be the founding of such research effort.

An expansion of this research in the future should include additional collection of data from teams working in a controlled environment. Although the Race Committee scenario data was collected from an actual team, this research had no control over the team design factors or job structure. This research feels confident on the validity of the model, but having a controlled data set will expand the validation analysis, particularly the predictive validity. For example, the results of the various simulated scenarios can be compared with the results of an actual team when all the input data factors have been controlled or are known.

In conclusion, this research contributes to the advancement of the field of team modeling, particularly the modeling of human teams for the purpose of improving their performance. The resulting TCM is a computational tool capable of modeling team coordination and performance for a wide set of teams. The TCM could be used, according to the CTDM methodology, to determine the team configuration that most

likely provides the best result performing a specific job. Nonetheless, this research has limitations to be addressed in future research efforts.

REFERENCES

- Albino, V., P. Pontrandolfo, et al. (2002). "Analysis of Information Flows to Enhance the coordination of production processes." International Journal of Production Economics **75**(1): 7-19.
- Altman-Klein, H. and A. Pongonis-Michugh (2005). National Differences in Teamwork. Organizational Simulation. W. B. Rouse and K. R. Boff. New Jersey, Wiley: 229-251.
- Ancona, D. G. and D. F. Caldwell (1992). "Demography and Design: Predictors of New Product Team Performance." Organization Science **3**(3): 321-341.
- Andres, H. P. and R. W. Zmud (2002). "A Contingency Approach to Software Project Coordination." Journal of Management Information Systems **18**(3): 41-70.
- Baker, D. P. and E. Salas (1997). Principles for Measuring Teamwork: A summary and Look Toward the Future. Team Performance Assessment and Measurement: Theory, Methods, and Applications M. T. Brannick, E. Salas and C. Prince. Mahwah, N.J., Lawrence Erlbaum Associates 331-355.
- Balci, O. (1998). Verification, Validation, and Testing. Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice J. Banks. New York, Wiley: 335 - 396.
- Beavers, G. and H. Hexmoor (2001). Teams of agents. IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ.
- Bonabeau, E. (2002). "Agent-based modeling: Methods and techniques for simulating human systems." Proceedings of the National Academy of Sciences **99**(May 14, 2002): 7280-7287.
- Brannick, M. T., A. Prince, et al. (1995). "The Measurement of Team Process." Human Factors **37**(3): 641-651.
- Brannick, M. T. and C. Prince (1997). An Overview of Team Performance Measurement. Team Performance Assessment and Measurement: Theory, Methods, and Applications M. T. Brannick, E. Salas and C. Prince. Mahwah, N.J., Lawrence Erlbaum Associates 3-16.
- Burke, C. S., E. Salas, et al. (2004). "How to Turn a Team of Experts into an Expert Medical Team: Guidance From the Aviation and Military Communities." Quality and Safety in HealthCare **13**(Suppl 1): i96-i104.

Burton, R. M. and B. Obel (1995). "The Validity of Computational Models in Organizational Science: From Model Realism to Purpose of the Model." Computational and Mathematical Organization Theory **1**(1): 57-71.

Campbell, D. J. (1988). "Task Complexity: A Review Analysis." Academy of Management Review **13**(1): 40-52.

Cannon-Bowers, J. A. and E. Salas (1997). A Framework for Developing Team Performance Measures in Training. Team Performance Assessment and Measurement: Theory, Methods, and Applications M. T. Brannick, E. Salas and C. Prince. Mahwah, N.J., Lawrence Erlbaum Associates 45-62.

Cannon-Bowers, J. A., S. I. Tannenbaum, et al. (1995). Defining Team Competencies and Establishing Team Training Requirements. Team Effectiveness and Decision Making in Organizations. R. Guzzo and E. Salas. San Francisco, CA, Jossey-Bass: 333-380.

Carley, K. M. (1995). "Computational and Mathematical Organization Theory: Perspective and Directions." Computational & Mathematical Organization Theory **1**(1): 39-56.

Cheng, C. E. (1992). "Computer Simulation and Its Management Applications." Computer in Industry **20**: 229-238.

Christiansen, T. R., L. C. Christiansen, et al. (1997). "Object-Oriented Enterprise Modeling and Simulation of AEC Projects." Microcomputers in Civil Engineering **12**(3): 157-170.

Christiansen, T. R., L. C. Christiansen, et al. (1999). "Modeling and Simulating Coordination in Projects." Journal of Organizational Computing and Electronic Commerce **9**(1): 33-56.

Cohen, S. G. and D. E. Bailey (1997). "What Makes Teams Work: Group Effectiveness Research from the Shop Floor to the Executive Suite." Journal of Management **23**(3): 239-290.

Daft, R. L. (1995). Organization Theory & Design. St. Paul, MN, West Publishing Company.

Dong, S. and B. Hu (2008). "Multi-agent based simulation of team effectiveness in team's task process: a member-task interaction perspective." International Journal of Simulation and Process Modelling **4**(1): 54-68.

- Doolen, T. L., M. E. Hacker, et al. (2003). "The Impact of Organizational Context on Work Team Effectiveness: A Study of Production Team." IEEE Transactions on Engineering Management **50**(3): 285-296.
- Espinosa, A., J. Lerch, et al. (2004). Explicit vs. Implicit Coordination Mechanisms and Task Dependencies: One Size Does Not Fit All. Team Cognition: Understanding the Factors That Drive Process and Performance. E. Salas and S. M. Fiore, American Psychological Association
- Fan, X. and J. Yen (2007). R-CAST: Integrating Team Intelligence for Human-Centered Teamwork. Twenty-Second AAAI Conference on Artificial Intelligence
- Fiore, S. M., E. Salas, et al. (2003). "Distributed Coordination Space: Toward a theory of Distributed Team Process and Performance." Theoretical Issues in Ergonomics Science **4**(3-4): 340-364.
- Fussell, S. R., R. E. Kraut, et al. (1998). Coordination, Overload and Team Performance: Effects of Team Communications Strategies. Proceedings of the 1998 ACM conference on Computer supported cooperative work Seattle, WS, ACM Press.
- Galbraith, J. R. (1977). Organization Design. Reading, MA, Addison-Wesley Publishing Company.
- Garrido, J. M. (2001). Object-oriented discrete-event simulation with Java: a practical introduction. New York, Kluwer Academic/Plenum Publishers.
- George, M. L. (2002). Lean Six Sigma : Combining Six Sigma Quality with Lean Production Speed. New York, NY, McGraw-Hill.
- Giachetti, R. (2010). Design of Enterprise Systems: Theory, Architecture, and Methods Boca Raton, FL, CRC Press.
- Giachetti, R. E. (2004). "A Framework to Review the Information Integration of the Enterprise." International Journal of Production Research **42**(6): 1147-1166.
- Giachetti, R. E. (2006). "Understanding Interdependence in Enterprise Systems: A Model and Measurement Formalisms." Proceedings of the Business Process Management 2006 Workshops, J. Eder and S. Dustdar (Eds.), LNCS 4103: 257 – 268, Springer-Verlag
- Hexmoor, H. and G. Beavers (2002). Measuring Team Effectiveness. IASTED International Multi-Conference: Applied Informatics 2002, Innsbruck, Austria.

Hirschfeld, R. R., M. H. Jordan, et al. (2006). "Becoming Team Players: Team Members' Mastery of Teamwork Knowledge as a Predictor of Team Proficiency and Observed Teamwork Effectiveness." Journal of Applied Psychology **9**(2): 467-474.

Hoegl, M. and H. G. Gemuenden (2001). "Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence " Organization Science **12**(4): 435-449.

Howell, D., C. Windahl, et al. (2010). "A project Contingency Framework based on Uncertainty and its Consequences " International Journal of Project Management **28**(3): 256-264.

IAI (2002). Open Cybele Agent Infrastructure Users Guide, Intelligent Automation Incorporated.

IAI (2006). CybelePro Agent Infrastructure User's Guide 3.0. Rockville, MD, Intelligent Automation, Inc.

Jin, Y., R. E. Levitt, et al. (1995). "The Virtual Design Team: A Computer Simulation Framework for Studying Organizational Aspects of Concurrent Design " Simulation **64**(3): 160-173.

Kiel, L. D. (2005). "A Primer for Agent-Based Modeling in Public Administration: Exploring Complexity in "Would Be" Administrative Worlds." Public Administration Quarterly **29**(3-4): 268-296.

Kim, J. and R. M. Burton (2003). "The Effect of Task Uncertainty and Decentralization on Project Team Performance." Computational & Mathematical Organization Theory **8**(4): 365-384.

Kim, S. C., S. H. Chang, and G. Heo (2006). "Team Crystallization (SIO₂) : Dynamic Model of Team Effectiveness Evaluation Under the Dynamic and Tactical Environment at Nuclear Installation." Safety Science **44**: 701-721.

Komaki, J. L. (1997). Behind the Scenes: Fieldtesting a Measure of Effectiveness for Theater Teams. Team Performance Assessment and Measurement: Theory, Methods, and Applications M. T. Brannick, E. Salas and C. Prince. Mahwah, N.J., Lawrence Erlbaum Associates 63-84.

Kraiger, K. and L. H. Wenzel (1997). Conceptual Development and Empirical Evaluation of Measures of Shared Mental Models as Indicators of Team Effectiveness. Team Performance Assessment and Measurement: Theory, Methods, and Applications M. T. Brannick, E. Salas and C. Prince. Mahwah, N.J., Lawrence Erlbaum Associates 63-84.

- Kunz, J. C., T. R. Christiansen, et al. (1998). "The Virtual Design Team." Communications of the ACM **41**(11): 84 - 91.
- Levitt, R. E. (2003) VDT Computational Emulation Models of Organizations: State of the Art and Practice. **Volume**, 1-5 DOI:
- Levitt, R. E., J. Thomsen, et al. (1999). "Simulation Project Work Processes and Organizations: Toward a Micro-Contingency Theory of Organizational Design." Management Science **45**(11): 1479- 1495.
- Loper, M. L. and B. Presnell (2005). Modeling an Emergency Operations Center with Agents. Proceedings of the 2005 Winter Simulation Conference, Orlando, FL.
- Macal, C. M. and M. J. North (2005). Tutorial on Agent-Based Modeling and Simulation. Proceedings of the 2005 Winter Simulation Conference, Orlando, FL.
- Malone, T. W. (1987). "Modeling Coordination in Organizations and Markets." Management Science **33**(10): 1317-1332.
- Malone, T. W. and K. Crowston (2001). The Interdisciplinary Study of Coordination. Coordination Theory and Collaboration Technology. G. M. Olson, T. W. Malone and J. B. Smith, LEA, Inc.: 7-50.
- Manganelli, R. L. and M. M. Klein (1996). The Reengineering Handbook: A Step-By-Step Guide to Business Transformation New York, NY, American Management Association.
- Marks, M. A. (2000). "A Critical Analysis of Computer Simulations for Conducting Team Research." Small Group Research **31**(6): 653-675.
- Marks, M. A., J. E. Mathieu, et al. (2001). "A Temporally Based Framework and Taxonomy of Team Processes." Academy of Management Review **26**(3): 356-376.
- Mathieu, J. E., L. L. Gilson, et al. (2006). "Empowerment and Team Effectiveness: An Empirical Test of an Integrated Model." Journal of Applied Psychology **91**(1): 97-108.
- Mitchell, D. K. (2000). Mental Workload and ARL Workload Modeling Tools. A. R. Laboratory, Human Research & Engineering Directorate.
- Morgan, B. B., E. Salas, et al. (1993). "An Analysis of Team Evolution and Maturation." The Journal of General Psychology **120**(3): 277-291.
- North, M. J. and C. M. Macal (2007). Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation. New York, NY, Oxford University Press.

- Nuñez, A. N. (2006). A Methodology to Quantify Coordination of a Business Process Industrial and Systems Engineering. Miami, FL, Florida International University. **Ph.D.**
- Pagell, M. and J. A. LePine (2002). "Multiple Case Studies of Team Effectiveness in Manufacturing Organizations." Journal of Operations Management **20**(5): 619-639.
- Paris, C. R., E. Salas, et al. (2000). "Teamwork in Multi-person Systems: a Review and Analysis." Ergonomics **43**(8): 1052-1075.
- Petre, M. (2004). "Team Coordination through Externalized Mental Imagery " International Journal of Human-Computer Studies **61**: 205-218.
- Piccoli, G., A. Powell, et al. (2004). "Virtual Teams: Team Control Structure, Work Processes, and Team Effectiveness." Information Technology & People **17**(4): 359-379.
- Rathnam, A., V. Mahajan, et al. (1995). "Facilitating Coordination in Customer Support Teams: A Framework and Its Implications for the Design of Information Technology." Management Science **41**(12): 1900-1921.
- Salas, E., J. W. Guthrie, et al. (2005). Modeling Team Performance: The Basic Ingredients and Research Needs. Organizational Simulation. W. B. Rouse and K. R. Boff. New Jersey, John Wiley: 185-228.
- Salas, E., D. E. Sims, et al. (2005). "Is there a "Big Five" in Teamwork?" Small Group Research **36**(5): 555-599.
- Sargent, R. G. (2007). Simulation Model Verification and Validation. 2007 Winter Simulation Conference.
- Stewart, G. L. and M. Barrick (2000). "Team Structure and Performance: Assessing the Mediating Role of Intrateam Process and the Moderating Role of Task Type." Academy of Management Journal **43** (2) 135 - 148.
- Stewart, G. L. (2006). "A Meta-Analytic Review of Relationships Between Team Design Features and Team Performance." Journal of Management **32**(1): 29-55.
- Tambe, M. (1998). "Implementing Agent Teams in Dynamic Multiagent Environments." Applied Artificial Intelligence **12**(2-3): 189-210.
- Umanath, N. S. (2003). "The Concept of Contingency Beyond "It depends": illustrations from IS Research Stream." Information & Management **40**: 551-562.

Wang, W. P., D. L. Kleinman, et al. (2001). Modeling Team Coordination and Decisions in a Distributed Dynamic Environment. Coordination Theory and Collaboration Technology. G. M. Olson, T. W. Malone and J. B. Smith, LEA, Inc.: 7-50.

Wong, S. and R. M. Burton (2000). "Virtual Teams: What are their Characteristics, and Impact on Team Performance? ." Computational & Mathematical Organization Theory **6**(4): 339-360.

Yen, J., X. Fan, et al. (2006). "Agents with Shared Mental Models for Enhancing Team Decision Makings." Decision Support Systems **41**: 634-653.

Yilmaz, L. and J. Philips (2007). "Team-RUP: agent-based simulation of team behavior in software development organizations." International Journal of Simulation and Process Modelling **3**(3): 170-179.

VITA

JOSE A. ROJAS-VILLAFANE

EDUCATION

Florida International University, Miami, FL, Industrial & Systems Engineering,

Doctoral Candidate, Industrial & Systems Engineering, summer 2010

Dissertation: An Agent-Based Model of Team Coordination and Performance

University of Michigan, Ann Arbor, MI, Industrial & Operations Engineering, M.S. 1997

University of Puerto Rico, Mayaguez, PR, Industrial Engineering, B.S. 1996

RELEVANT EXPERIENCE

August 2004 – Dec 2005 Teaching Assistant, Florida International University,
Miami, FL

Jan 2004 – present Instructor, Universidad del Turabo, Gurabo, PR.

Jan 2003 – Dec 2003 Lecturer, Universidad del Turabo, Gurabo, PR.

Jan 2002 – Dec 2003 Lecturer, Colegio Universitario de San Juan, San Juan, PR

Apr 1999 – Nov 2001 Project Engineer, Cutler Hammer- Eaton, Las Piedras, PR

RESEARCH PROJECTS

Rojas J., J. Gershenson, D. Villegas, A. Rincón, C. Alvarado, *Multi-disciplinary Systems Engineering: Engaging MSI STEM Students through Space-based Capstone Design*, NASA MUREP Small Programs Grant (October 2009 – September 2012)

Giachetti R., M. Perry, and J. Rojas, *Simulation Modeling and Statistical Network Tools for Improving Collaboration in Military Logistics*, US Air Force Research Lab (August 2007 – December 2008).

PUBLICATIONS

Conference Papers:

Giachetti, R., and J. Rojas (2007), Simulating Coordination of Human-Robot Teams for Military Operations, *Proceedings of the 2007 Industrial Engineering Research Conference*

Rojas, J., and R. Giachetti (2008), An agent-based simulation model to explore collaboration for unstructured tasks, *Proceedings of the 2008 Industrial Engineering Research Conference*

Rojas, J., and R. Giachetti (2009), An agent-based simulation model to Analyze Team Performance On Jobs with a Stochastic Structure, *Proceedings of the First International Conference on Advances in System Simulation (SIMUL 2009)*.

HONORS

2005 Omega Rho International Honor Society of Operations Research
1996 – 1997 Rackham Merit Fellowship, University of Michigan, Ann Arbor, MI
1995 Tau Beta Pi Engineering Honor Society
1995 Alpha Pi Mu Industrial Engineering Honor Society