Spring 5-6-2011

MULTIAGENT COALITION FORMATION IN UNCERTAIN ENVIRONMENTS WITH TYPE-CHANGING INFLUENCES AND ITS APPLICATION TOWARDS FORMING HUMAN COALITIONS

Nobel A. Khandaker

University of Nebraska-Lincoln, nobel@huskers.unl.edu

MULTIAGENT COALITION FORMATION IN UNCERTAIN ENVIRONMENTS

WITH TYPE-CHANGING INFLUENCES AND ITS APPLICATION TOWARDS

FORMING HUMAN COALITIONS

by

Nobel Khandaker

A DISSERTATION

Presented to the Faculty of

The Graduate College at University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Computer Science

Under the Supervision of Professor Leen-Kiat Soh

Lincoln, Nebraska

May, 2011

MULTIAGENT COALITION FORMATION IN UNCERTAIN ENVIRONMENTS WITH TYPE-CHANGING INFLUENCES AND ITS APPLICATION TOWARDS FORMING HUMAN COALITIONS

Nobel Khandaker, Ph.D.

University of Nebraska, 2011

Adviser: Leen-Kiat Soh

We aim to solve the problem forming multiagent coalitions in uncertain environments where the coalition members' capability of solving tasks change due to their learning. The MCFP-M problem for the agents refers to forming or joining coalitions on behalf of a set of human users so that those human users can solve tasks and improve their types (expertise) to improve their performances over time. MCFP-A problem for a set of agents refers to their forming or joining coalitions so that they are able to solve a set of assigned tasks while optimize their performance over time. We propose the Integrated Human Coalition Formation and Scaffolding (*i*HUCOFS) framework for solving MCFP-M. *i*HUCOFS agents balance the tradeoff between solving the current task well and improving the human users' types to solve future tasks better by facilitating learning and teaching. We have verified *i*HUCOFS' assumptions using simulation experiments and implemented the framework in ClassroomWiki–a Wiki environment for collaborative learning. Our deployment results show that *i*HUCOFS' agents can model the students accurately and form student groups to enhance collaboration and learning. We have proposed the Agents' Dyadic Learning Influenced Tradeoff (ADLIT) framework that consists of a coalition formation protocol and approximation strategies to solve MCFP-A. ADLIT agents can form coalition to solve the current task well and improve their performance over time by improving their types with learning. Our empirical studies show that the ADLIT agents' local learning interactions lead to a scalable and robust mechanism for improvement in the long term.

Acknowledgements

I would like to acknowledge the contributions of many people who helped me directly and indirectly to complete this work. First I would like to thank my advisor Dr. Leen-Kiat Soh who guided and supported me throughout the PhD process, always providing me with insightful advice. I am also thankful for the guidance of my supervisory committee members: Dr. Hong Jiang, Dr. Stephen Scott, Dr. Ashok Samal, Dr. David Brooks, and Dr. Victor Lesser. Finally, I would like to thank my parents, Khandaker Mozammel Haque and Mitali Hossain and my wife Shakila Azmal for their love and support. This thesis is *dedicated to* my parents.

My research work was completed utilizing the *Holland Computing Center* of the University of Nebraska - Lincoln and the research work was funded by the following grants:

- National Science Foundation grant: NSF SBIR DMI-0441249

- National Science Foundation grant: NSF grant CNS-0829647

- National Science Foundation grant: NSF Grant DMI-044129

- Microsoft Research Grant

Contents

Chapter 1

Introduction

Multiagent coalition formation is a process that allows a set of agents to work together to solve tasks that they cannot solve themselves alone effectively and efficiently. Here effectiveness refers to an agent's ability to solve a task well i.e., generate a solution for a task with high quality. Furthermore, efficiency denotes the reward-to-effort ratio incurred by the agents who solve a task. Multiagent coalition formation techniques can be used to solve a variety of real-world problems, e.g., task allocation (Shehory and Kraus, 1995), bargain buying in business transactions (Yamamoto and Sycara, 2001), and forming student groups in computer-supported collaborative learning environments (Vassileva et al., 2003). In our research, we aim to solve a problem that we call the *Multiagent Coalition Formation Problem* or MCFP where a set of agents form coalitions in an uncertain environment and their types (i.e., the agents ability to solve tasks) change (positively *or* negatively) over time due to their learning interactions with other agents. Furthermore, if we consider human users as intelligent agents capable of making autonomous agents, MCFP can be extended to a variety of real world problems that involve forming human coalitions.

Consider a set of students in a classroom who are about to work together in teams or coalitions on an assignment (e.g., design a software) to earn some reward (e.g., grade). This upcoming assignment includes a set of subtasks (e.g., coding, interface design) which are

to be solved by the members. From their previous experience, the students know their own capabilities (i.e., the nature of the subtasks he/she can solve) and their expertise of solving those subtasks (i.e., how well they can solve the subtasks). This expertise of the student members in solving the subtasks as well as other uncertain factors (e.g., a coalition member falling ill) determines *how well* a coalition of students is able to solve the upcoming assignment. In addition, since solving the problems involved in the assignment requires interactions among the members of a coalition, it is possible that the coalition members would be able to learn from one another and as a result would learn new capabilities and/or change their expertise (increase or decrease) related to those capabilities. This change in the students' capabilities and expertise has the potential to impact the performance of the coalitions they form in the long term (i.e., future assignments after this upcoming assignment). The students in this class are thus faced with the following problem:

> Which coalition should I form that would allow me to solve this new assignment well to earn rewards and to learn new capabilities or gain expertise to solve future assignments better, leading to a better overall course grade?

This coalition formation problem has the following distinct characteristics:

- If we denote a student's capabilities and the associated expertise as that student's *type*, the types of the coalition members help determine how well they are able to solve a problem and thus their rewards.

- A student can learn by interacting with other students and the students' *types may change* (positively or negatively) due to their learning.

- There is a *tradeoff* between the current task and future task rewards. A coalition that provides the maximal reward now (short term) may not provide the optimal learning opportunities that would result in the type changes that could lead to better future task rewards (long term) and vice versa.

- The environment in which the students operate is *uncertain*. The uncertainty in the environment arises from the fact that: (1) the quality of the solution of a task solved by a coalition and (2) the type change that can be achieved by an agent by learning *cannot* be accurately calculated beforehand but only *estimated*.

One way we can solve this coalition formation problem is by using a set of intelligent agents who would find the right coalition for the students. In this case, the agents work as mediators for the human users to form coalitions for a chosen task where the human users become the actors to solve that chosen task. We thus define this coalition formation problem as MCFP-M coalition formation subproblem under MCFP as:

> **MCFP-M** - The MCFP-M coalition formation problem for the mediator agents refers to forming or joining a coalition for the participating actor agents or human users (who they represent) such that those actor agents are able to solve the chosen task effectively and efficiently and optimally improve their types over time.[1]

Similar to MCFP-M, there are several multiagent coalition formation scenarios where the agents form coalitions in an uncertain environment to solve tasks and the agents' type (skills, knowledge, etc.) can *change* because of their learning interactions with other agents in the environment. Examples of this type of coalition formation can be found in electronic marketplaces (Lerman and Shehory, 2000), trust-based coalition formations (Griffiths and Luck, 2003), etc. All of the distinct characteristics of MCFP-M also exist in these coalition formation problems but with the difference that the agents *themselves* now act in the coalitions in MCFP-A. We define the agents' coalition formation in the uncertain environments with type-changing influences as the MCFP-A subproblem under MCFP as:

---

[1]Notice that, in our definition of the MCFP-M, we use the term "actor agents" to denote "human users". That is because, in our agent-mediated coalition formation problem, we assume human users as agents who are able to perceive the environment and take autonomous actions and who employ other agents as mediators to find the right coalition for them.

MCFP-A - The MCFP-A coalition formation problem for the actor agents refers to forming or joining coalitions so that they are able to solve the current tasks effectively and efficiently and optimally improve their types over time.

Comparing the coalition formation problem for the actor agents and the coalition formation problem for the mediator agents, we see that there are significant differences between these two subproblems which require different approaches while we design the solutions for them. The three main categories of differences are:

- **Control** - Since the agents who form coalitions cannot act in them in the mediator-formed coalition formation problems, the mediating agents have *less control* over how the actors would behave once they join the coalition. This lack of control leads to additional uncertainties in the mediator agents' coalition formation- and learning-related decision making process.

- **Interactions & Type Change** - Unlike the agent-actor case, the interactions in the agent-mediator coalition formation problems occur among the agents who do not form coalitions. As a result, the mediators who make the coalition-forming, or learning decisions experience *additional tracking and modeling responsibilities* that are not incurred by the coalition-forming actors in the agent-actor-based coalition formation problems.

- **Imposed Constraints** - If the perceptions or goals of the actor and the mediator agents are different, that difference may often *impose a set of constraints* on the coalition-forming mediators. As a result, the mediators would have to accommodate those imposed constraints in its learning and coalition formation process.

Due to these significant differences, the critical issues in a multiagent coalition formation problem, e.g., reward distribution scheme and social welfare (Sandholm et al., 1999),

would have to be accommodated in the agent reasoning process differently in the MCFP-A and MCFP-M problems. Our division of MCFP into two subproblems will allow us to design separate solutions that cater to the specific needs of each of the subproblems.

1.1   Motivation

The motivation for solving the MCFP arises from its potential of providing us critical insights into the various real-world coalition formation problems in uncertain environments with type-changing influences. These insights can then be used to understand our limitations (e.g., the computational complexities) in solving those problems and to design frameworks and algorithms that improve the state of the art of the solution techniques of such problems. Due to their ability to communicate, cooperate, and make autonomous decisions, multi-agent systems are gaining popularity as a tool for designing real-world software solutions (Fischer et al., 2007). Many of those real-world problems e.g., task allocation, bargain buying in business transactions, resource allocation, and human coalition formation require formation of coalitions in uncertain environments where the types or capabilities of the actors change as they collaborate with their coalition members. One way these real-world coalition formation problems can be solved is by mapping them onto our multiagent coalition formation subproblems (MCFP-A, MCFP-M) and then by using our developed solution techniques. For example, the buyer coalition formation problem where buyer coalitions are acting together to buy bulk items together over the internet can be mapped to MCFP-A. In this case, the agents forming the coalitions are the actors, they form coalitions in an uncertain environment since the availability of an agent with specific type (defined by buying need and capability) for a specific coalitional buying action is not guaranteed, and as the agents successfully negotiate bulk buying deals and make profits, their capability as a buyer (e.g., the amount of capital they have) is likely to increase i.e., their types will change over time. Furthermore, the student coalition formation problem in

collaborative learning environments can be mapped onto MCFP-M, where the uncertainty in the environment arises from, (1) the agents' limited observability - the agents forming student coalitions cannot observe all interactions of the students in the environment and (2) the inherent inaccuracy in measuring students' attributes that define their types - i.e., the inaccuracy in the measurement of students' attributes using tests or surveys (Cortina, 1993). Also, the participating students' types (e.g., knowledge and ability to solve problems) change due to their interactions with their coalition members.

Analyzing the state of the art of MCFP-M type coalition formation problems, i.e., problems where the agents mediate the coalitions formed by human users, we see that researchers have tried a number of different solution approaches. For example, (1) Gogoulou et al. (2007), Christos and Kyparisia (2007) and Graf and Bekele (2006) used optimization algorithms to form student groups, (2) Diebel (2005) used educational heuristics to form student groups, (3) Greer et al. (2001) used a multiagent system to match learners (i.e., two-member coalitions) who need help with other knowledgeable peers, and (4) Pechoucek et al. (2002) used a multiagent system to form coalitions of relief organizations to aid in the disaster management scenario. However, these coalition formation research approaches do not accommodate the fact that the human actors in real-world coalition formation scenarios work in an uncertain environment where the measured attribute values are not accurate (i.e., the values vary due to unmeasured environmental factors (Cortina, 1993)) and the impact of those attributes on the effectiveness and efficiency of a coalition may not be accurately identified but can only be estimated through tracking and modeling. As a result, these coalition formation techniques often rely on manually specified and fixed (i.e., *not* learning enabled) coalition formation strategies. So, a multiagent coalition formation framework that uses agent-mediated coalition formation techniques and is able to learn (1) the models of the actors (i.e., the students in this case) and (2) the probabilistic mapping of those models on the effectiveness and efficiency of the coalitions would be able to form better coalitions that optimize their effectiveness and efficiency for a set of future tasks by

improving the types of the participating actors.

Furthermore, a variety of real-world coalition formation problems occur in uncertain environments with type-changing influences, but the current state of the art of multiagent coalition formation techniques that are designed to be applied in the real world *do not* adequately accommodate the uncertainty and the type-changing influences in the environment into account. For example, Shehory and Kraus (1995); Yamamoto and Sycara (2001); Tsvetovat and Sycara (2000); Klusch and Gerber (2002); Pechoucek et al. (2002); Soh and Li (2003); Cornforth et al. (2004); Huhns et al. (2005); and Kiam et al. (2007) do not consider the uncertainty in the environment. Furthermore, the recently published coalition formation research approaches that do take the environmental uncertainty into account to form coalitions e.g., Kraus et al. (2003); Chalkiadakis and Boutilier (2004); Hosam and Khaldoun (2006); Chalkiadakis (2007) do not adequately exploit the agents' type-changing influences into account in their framework or solution approach. For example, the agents in the solutions proposed in (Kraus et al., 2003); (Chalkiadakis and Boutilier, 2004); (Hosam and Khaldoun, 2006); (Chalkiadakis, 2007) do not consider the impact of an agent *influencing type changes* in its coalition members by learning actions. Due to the necessary collaboration of the agents in the coalition formation problems, if an agent is able to collaborate again with its current coalition's members in future, a positive type change in those members' types can potentially improve that agent's performances significantly. So, our discussions indicate that there is a need for multiagent coalition formation techniques that can take the uncertainty in the environment and the type-changing influences into account to form effective and efficient coalitions for MCFP-A and MCFP-M coalitions.

To summarize, applying the solution techniques of MCFP-A and MCFP-M on these problems would allow us to (1) understand the theoretical limitations of finding such coalition formation problems, (2) solve the real-world coalition formation problems that are being solved using other existing techniques, and (3) apply the devised solution techniques to solve important but yet to be automatically solved real-world coalition formation prob-

lems e.g., businesses forming coalitions to improve their earnings or to solve difficult tasks ((Khanna and Yafeh, 2007); (Iacobucci and Rosa, 2005)) and political coalition formation to pass legislature (Kreppel and Tsebelis, 1999)) *well*.

1.2 Solution Approach

To solve MCFP-M, we propose the Integrated Human Coalition Formation and Scaffolding Framework (*i*HUCOFS). *i*HUCOFS is a multiagent-based coalition formation framework in which a set of intelligent agents assist a set of human users form coalitions and help (scaffold) those human users to optimize their effectiveness and efficiency in the coalitions over a set of tasks. The *i*HUCOFS framework is designed to form and scaffold coalitions, trading off expected utility of solving the current task and the potential utility of better coalitions in the future in the hope that such *sacrifices in the present* would yield *positive type changes* in the participating human users so that they are able to earn better rewards in the future.

We have realized *i*HUCOFS (Figure 1.1) in two different multiagent instantiations that solve the MCFP-M for student coalition formation in the collaborative learning domain. Our first instantiation of *i*HUCOFS is an educational heuristic-based coalition formation algorithm called VALCAM *Vickrey Auction-Based Learning Enabled Coalition Formation Algorithm* (VALCAM) that improves student learning by forming competent student groups with students who are also compatible. VALCAM has been implemented in the asynchronous CSCL (Computer-Supported Collaborative Learning) tool I-MINDS. Although VALCAM's implementation in the asynchronous version of I-MINDS is a significant part of our *i*HUCOFS framework, we are *not* going to discuss its design or implementation in I-MINDS in this dissertation. For a summary of our achievements regarding VALCAM and its implementation in the asynchronous version of I-MINDS, see (Khandaker and Soh, 2011*b*; Khandaker et al., 2011).

To understand the validity and implications of *i*HUCOFS framework, we have designed SimCoL (Khandaker and Soh, 2010*d*) - a simulation environment in which a set of agents, guided by the published collaborative learning theories and observations, mimic the collaborative learning interactions of the students in a classroom. SimCoL allows us to conduct *low-cost replicable* experiments that can be used to validate the assumptions and test the impact of our group formation techniques.

One deficiency of VALCAM is that it primarily relies on provided heuristics to form student coalitions. To improve our instantiation of *i*HUCOFS for solving student coalition formation problems in collaborative learning environments, we have designed and implemented MHCF (Multiagent Human Coalition Formation) algorithm. In MHCF, agents work as *mediators* for the students and use a negotiation-based coalition formation algorithm to form student coalitions. The novelty of MHCF lies in its multiagent-based multidimensional modeling of the participating students and its probabilistic modeling of the environment that allows it to learn to form effective and efficient coalitions that: (1) improve the current-task performance of the students and (2) improve the behavior of the students through learning so that, they are able to earn higher rewards for future tasks.

To solve MCFP-A, our proposed ADLIT framework allows an agent to make a series of *sequentially rational* (Chalkiadakis and Boutilier, 2008) decisions to improve the tradeoff between the current task and future task to maximize its rewards over time with respect to its type. In our design and implementation of ADLIT, we formulate the type change of the agents as a Partially Observable Markov Decision Process (Kaelbling et al., 1998) and design a coalition formation protocol and approximation strategies. The coalition formation protocol specifies how a set of agents coordinate among themselves to pick (1) the types of the members for their coalition and (2) the implicit and explicit learning actions for their coalitions. In the coalition formation protocol, the agents have to evaluate the different coalition formation proposals presented to them in the light of the environment uncertainty and openness. Since the exact evaluation of a coalition in terms of the current task vs.
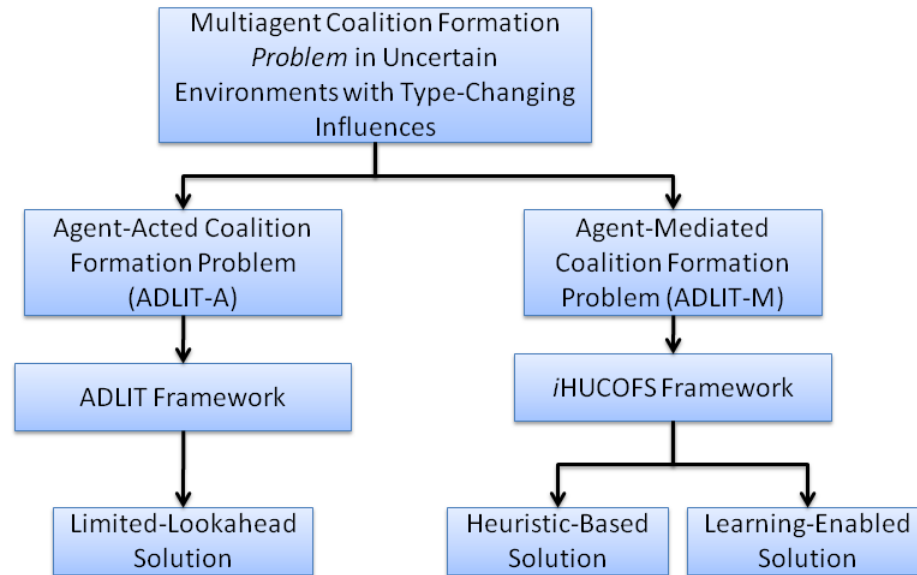
Figure 1.1: Problem-Solution Framework

future task reward tradeoff under partial observability and uncertainty is computationally intractable (Chalkiadakis, 2007), we have designed three approximation strategies. First, in the Myopic Current Task (MCT) strategy, the agents reasoning focus only on the current task reward *without* making deliberate effort towards improving their types through implicit and explicit learning. Second, in the One-Step Lookahead (OSL), the agents *look ahead* into the next round of the repeated coalition formation process to approximate the future-task reward gain in addition to considering current task rewards. Finally, as an improvement over our One-Step Lookahead, in the Learning by Exploration (LEA) method, the agents use a *simulated annealing* (Russel and Norvig, 2003, p. 115) approach to improve their choice of learning actions by balancing its exploration and exploitation.

1.3 Contributions

Our research work contributes to the advancements of the solution techniques for the of multiagent coalition formation problems in uncertain environments with type-changing in-

fluences. To be specific, we have designed and implemented the following (Figure 1.2):
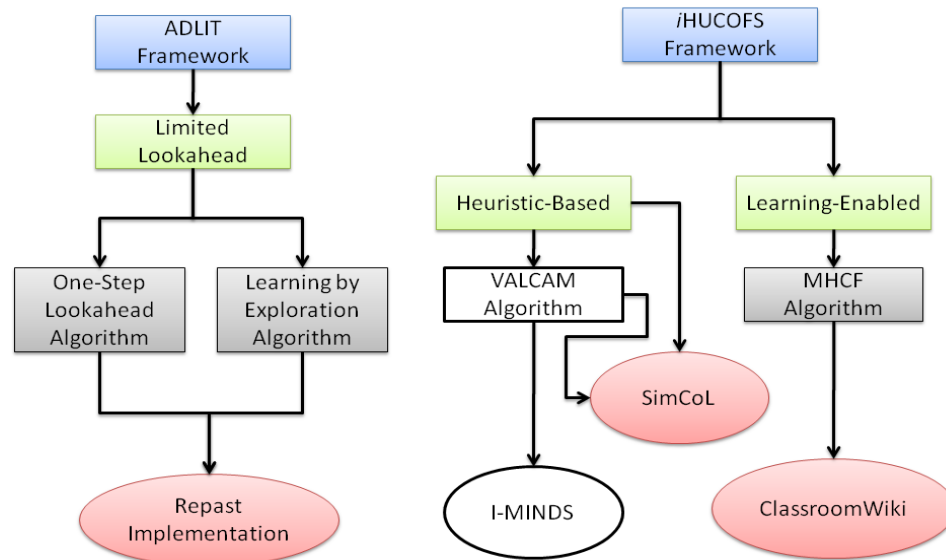
Figure 1.2: Implementation of Our Solution

- **iHUCOFS**: We have designed *i*HUCOFS (Soh and Khandaker, 2007; Khandaker and Soh, 2008)—a framework that describes the MCFP-M with a set of assumptions and provides design principles for solving MCFP-M effectively. We have instantiated *i*HUCOFS using two algorithms - VALCAM and MHCF and designed and implemented the *Simulation Tool for Collaborative Learning* (SimCoL) - a multiagent toolkit that simulates the interactions of the actors and mediators in the *i*HUCOFS environment.

  - **VALCAM**: VALCAM: According to the principles described in *i*HUCOFS, we have designed the VALCAM algorithm (Soh, Khandaker, Liu and Jiang, 2006; Soh and Khandaker, 2007; Khandaker et al., 2011; Soh, Khandaker and Jiang, 2006; Khandaker et al., 2006; Soh et al., 2008; Khandaker and Soh, 2011*b*) and investigated its impact by implementing it in the *asynchronous* version of I-MINDS (Intelligent Multiagent Infrastructure for Distributed Systems in Education) (Khandaker and Soh, 2011*b*; Khandaker et al., 2011). Our studies show

that VALCAM is able to improve student learning and performances by forming heterogeneous student groups that contain competent and compatible students.

- **SimCoL**: While designing *i*HUCOFS framework, we wanted to understand the role of competence and compatibility on the collaborative learning outcome of the students. Considering the high time and cost requirements of conducting experiments with human subjects, we have designed and implemented SimCoL (Khandaker and Soh, 2010*d*)—a multiagent tool for simulating student learning interactions in a CSCL environments. SimCoL uses a set of observations and theories published in the collaborative learning research domain (Shell and Brooks, 2007) to simulate the *student-student* and *student-teacher* interactions in a CSCL classroom. The analysis of the results of our conducted simulation experiments (Khandaker and Soh, 2010*d*) shows that SimCoL is able to very closely portray the collaborative learning behavior of the students in a CSCL classroom.

- **MHCF**: To improve upon the design of the VALCAM algorithm, we have designed the MHCF algorithm that uses a negotiation-based multiagent coalition formation algorithm and a Bayesian network (Cowell, 1998, pp. 9-27) to learn the composition of student groups that would allow the members to solve the current task well as well as improve their behavior to earn higher rewards for future tasks. To investigate the impact of MHCF algorithm, we have designed and implemented ClassroomWiki—a multiagent-based asynchronous online Wiki environment.

    * **ClassroomWiki**: In ClassroomWiki (Khandaker and Soh, 2010*e*,*a*,*c*; Khandaker et al., 2011), the students are able to collaboratively prepare Wikis on teacher-assigned topics where the mediator intelligent agents track and model their learning behavior and activities. The mediator agents then use a negotiation-based algorithm and a Bayesian network to form stu-

dent groups that allow the students to solve the current-task reward and improve the current task rewards for the future tasks as well. We have conducted experiments in several courses at the University of Nebraska-Lincoln, where the students collaborated using ClassroomWiki to collaboratively create Wikis on teacher-assigned topics and the ClassroomWiki agents used MHCF to form student groups. The analysis of the results of our two semester-long experiments indicate that the MHCF-formed groups allowed the participants to solve the teacher-assigned tasks effectively as well as helped them improve their behavior to increase their effectiveness and efficiency over time.

- **MCFP-A**: We have designed ADLIT (Khandaker and Soh, 2011*a*)—a multiagent coalition formation framework for forming agent coalitions in an uncertain environment for agent actor cases with type-changing influences. While designing ADLIT, we have provided the necessary theoretical formulations to design a stable coalition formation protocol and two approximation algorithms (OSL and LEA) that allow the agents to improve their rewards by improving their current-task and future-task reward tradeoff. To test our design of the ADLIT framework, we have implemented it in Repast (Repast, 2009)—a multiagent simulation toolkit. The analysis of the results of our experiment in Repast shows that ADLIT improves the effectiveness and efficiency by exploiting the optimal current-task vs. future-task reward tradeoffs.

    - To the best of our knowledge, the ADLIT framework is the first that we know of that utilizes observations reported by researchers working in the human learning domain to strengthen the agents' decision-making process in multiagent coalition formation scenarios. Furthermore, current multiagent systems literature has *mostly focused on* how agents learn from each other when they interact or negotiate or collaborate (Chalkiadakis, 2007; Kraus et al., 2003; Chan and Le-

ung, 2008; Tsvetovat and Sycara, 2000; Klusch and Gerber, 2002; Kraus et al., 2004) *but not on* what learning can do to an agent's type when forming or joining a coalition, and parlaying that into the future. Our solution, on the other hand, integrally considers that, allowing agents to maximize their performance with respect to their type over time.

- We also consider the tradeoff balancing the current task and future task rewards. Though this tradeoff is *not new*, embedding the agents' learning influence into the process *is*. Here, that influence allows the agents to more comprehensively model their current task rewards and more accurately estimate their future task rewards. On the implementation viability side, since we have to deal with the uncertainty, openness, and partial observability, we also propose approximation strategies to lessen the computational burden on the agents. Our three sets of reported empirical analyses also provide insights into how the ADLIT framework behaves and adapts to changes in the multiagent environment. Other contributions of our research regarding MCFP-A include the Repast-based software implementation of the MCFP-A simulation and empirical data.

1.4   Outline

The rest of this dissertation organized as follows: first, in Chapter 2, we describe the recently published research works related to the solutions of MCFP-A and MCFP-M to motivate the need for a better solution approach to both of them. In Chapter 3, we present the *i*HUCOFS framework with a set of observations and assumptions that provide a guideline for developing algorithms for solving the MCFP-M problem and discuss our implementation of the *i*HUCOFS framework in the negotiation-based MHCF algorithm. In Chapter 4, we describe the design and implementation of SimCoL—a multiagent-based simulation environment for simulating the student activities in the computer-supported collaborative

learning classroom. In Chapter 5, we discuss our implementation of MHCF algorithm in ClassroomWiki - a multiagent- and web-based collaborative Wiki tool for classroom use. Chapter 6, discusses the ADLIT framework for multiagent coalition formation that is designed to solve the MCFP-A problem and presents the results of our experiments. Finally, Chapter 7 concludes and discusses the future directions of the continuations of our work.

Chapter 2

Related Work

In this chapter, we discuss the research works related to the the MCFP-A and the MCFP-M subproblems. First, in Section 2.1, we informally define the environments of our MCFP-A and MCFP-M problems, and from those definitions, identify critical aspects of our problems that distinguish our research approach from the other coalition formation research approaches. In Section 2.2, we discuss the problems addressed by the agent-mediated coalition formation research approaches in a variety of domains (e.g., collaborative learning, business), summarize their solutions, and discuss how our solution approach may better solve the MCFP-M problem. In Section 2.3, we summarize the problems addressed by the agent-acted coalition formation research approaches, present their solutions, and discuss how our solution approach is required to better solve the MCFP-A subproblem.

2.1   Critical Aspects & Problem Environment

As discussed in the definitions of our problems MCFP-A and MCFP-M in Section 1, we aim to solve the multiagent coalition formation problem in an uncertain environment with type-changing influences. To identify the critical aspects that define our problems and guide our solution approaches, we informally define the MCFP-M environment (see Section 3.3.2 for a formal definition) and the MCFP-A environment (see Section 6.1 for a

formal definition) as the following:

MCFP-M Environment—We assume that we have a set of actor agents *and* a set of mediator agents who need to form coalitions to solve a set of tasks. *The mediator agents work as assistants to the actor agents and help them form coalitions for the tasks the actor agents would like to solve*. Once the actor agents join their respective coalitions for a task, they interact (e.g., communicate, learn, scaffold) with their coalition members to solve that chosen task. The types of the coalition members determine (1) the cost they incur while solving the task, (2) the quality of solution they are able to produce through their interactions, and (3) the *change* (positive or negative) in their types they are able to achieve through their interactions.

MCFP-A Environment—We assume that we have a set of actor agents who need to form coalitions to solve a set of tasks. The tasks require the expertise of the actor agents where the expertise of an agent is determined by its type. Once the actor agents join their respective coalitions for a task, they interact (e.g., communicate, learn, teach) with their coalition members to solve that chosen task. The types of the coalition members determine (1) the cost they incur while solving the task, (2) the quality of solution they are able to produce through their interactions, and (3) the *change* (positive or negative) in their types they are able to achieve through their interactions.

Notice that, the uncertainty faced by the actor and mediator agents during the coalition formation process in MCFP-A and MCFP-M environments arises from the agents' incomplete information about (1) the *impact* of a coalition members' types on the effectiveness and efficiency of that coalitions and (2) the impact of a set of learning and scaffolding interactions on the types of the interacting members of a coalition.

The uncertainty and the type change in the environment are related to the following critical aspects that need to be accommodated to produce effective and efficient solutions of the MCFP-A *and* the MCFP-M problem:

- **Uncertainty of Coalitional Outcome** ($ca1$)—Since the environment is uncertain,

the effectiveness and efficiency of a potential coalition according to its members' types cannot be accurately calculated, but only estimated. As a result, during coalition formation, the agents need to *learn* or *model* the type-related uncertainty in the environment and use its learned uncertainty values to choose the most effective and efficient coalition that it can. That implies, to improve its decision-making ability over time, an agent needs to *probabilistically explore* its environment by learning or refining its estimate of the type-related uncertainties and *exploit* what it knows by joining the most effective and efficient coalition.

- **Tradeoffs Associated with Type-Change** ($ca2$)—In the MCFP-A and the MCFP-M environments, the agents are able to change the types of their own *and* the types of their coalition members through learning interactions. So, if the actor agents in MCFP-A and the mediator agents in MCFP-M are able to learn the dynamics of type change, they could improve the performances of the coalitions further by exploiting the current task vs. future task reward tradeoff. Exploiting this tradeoff, the agents in MCFP-A and MCFP-M may improve their rewards over time by sacrificing optimal reward for the current task, improving the coalition members' types, and then earning higher rewards in the future.

- **Probabilistic User Model and its Impact** ($ca3$)—In MCFP-M problem, it is critical for the mediator agents to model the actors it is assisting to form coalitions. Due to the uncertainties in the real-world, the mediator agents need to consider the fact that (1) its prepared model of the actors, (2) the impact of the actors' types on the effectiveness and the efficiency of a coalition, (3) and the change in their types they observe cannot be accurately calculated. These inaccuracies are produced by the additional difficulties (Section 1) faced by the mediator agents while helping (observing the actions of the actors, calculating the impact of the models of a coalition on its effectiveness and efficiency, etc.) the actor agents form coalitions. So, while modeling the actor agents, and calculating the impact of those models on the effectiveness and

efficiency of a coalition, a mediator agent needs to accommodate the environmental uncertainties that affect its models.

2.2   MCFP-M Coalition Formation Problem

Here we divide our discussions regarding the related works of the agent-mediated coalition formation research approaches according to the *characteristics* of the problem domains the researchers address. First, we discuss the existing *agent-mediated* coalition formation techniques for forming coalitions of human users and human organizations and distinguish their research approaches from ours according to the three critical aspects discussed in Section 2.1. Based on our discussed distinctions, we also present how our solution approach (Section 2.2) would solve MCFP-M better. Then, we discuss the existing multiagent coalition formation solutions in which the agents *act as representatives* to form buyer coalitions *on behalf of* the human users. Furthermore, we discuss how these research approaches do not adequately address the three critical aspects of MCFP-M and how our solution approach (Section 2.2) would yield better solutions for the MCFP-M problem.

Figure 2.1 summarizes our categorization of the relevant research approaches discussed in this subsection. Notice that in the coalition formation research approaches where agents act as representatives for solving buyer coalition formation problems (Section 2.3), the role of the actors in the coalitions is *minimal* as opposed to the agent roles in the other (Section 2.2) agent mediated problems of coalition formation. So, to better organize our discussions, we divide these two classes of problems into two different subsections.

There were several research approaches that utilized agents to form human coalitions. Table 2.1 presents the published research approaches and describes how the critical aspects of MCFP-M environment ($ca2, ca3$ Section 2.1) were addressed in those research works. Furthermore, Figure 2.1 categorizes these research approaches according to their respective considerations of the critical aspects ($ca2, ca3$ in Section 2.1) of MCFP-M. In Table 2.1, we
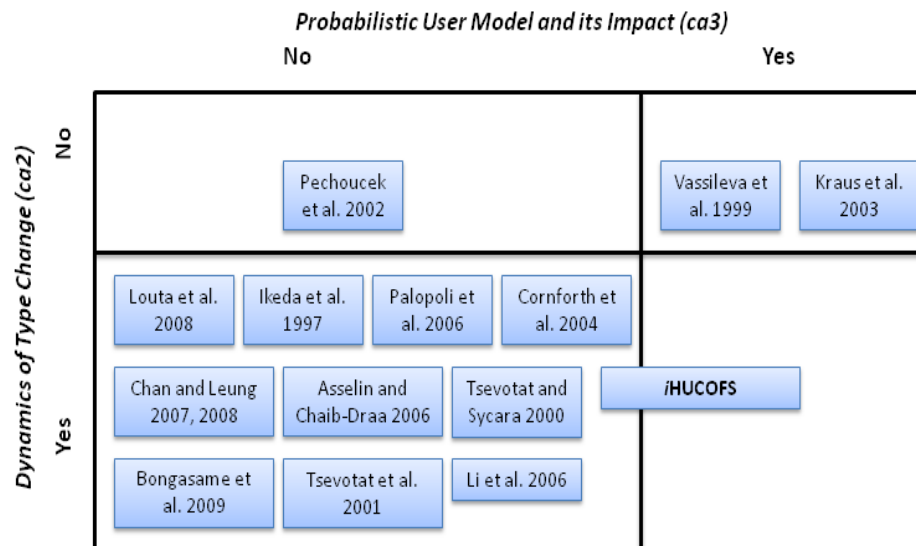
Figure 2.1: Categorization of Agent-Mediated Coalition Formation Research

see that these multiagent coalition formation research approaches were designed to form human coalitions with the mediated help of multiagent systems. Although their problem was similar to the problem of MCFP-M, these research works were not designed to address both of the aspects ($ca2,\ ca3$) critical to the solution of MCFP-M.

Table 2.1: Agent-Mediated Coalition Formation Research I

(Research): Problem. Solution.	ca2	ca3
(Ikeda et al., 1997; Inaba et al., 2000): Learner coalition formation in collaborative learning environments when they cannot solve the task individually. Software system that detects opportunities to form learner groups and a single agent that negotiates learner groups is designed.	*Yes*	*na*
(Pechoucek et al., 2002): Forming coalitions for the human users or organizations for solving large, complicated missions. A negotiation-based distributed coalition formation based on the agents' social knowledge of acquaintance models was provided.	*na*	*na*
(Vassileva et al., 1999; Bull et al., 2001; Greer et al., 2001; Vassileva et al., 2003): Learner coalition formation for providing 1-to-1 peer help in collaborative learning environment. A framework where agents watch and develop probabilistic models of the users and communicate among themselves to find the best 2-member coalition was developed.	*na*	*Yes*

(Research): Problem. Solution.	ca2	ca3
(Li et al., 2006): Forming coalition of learners with similar expertise and preferences. A fuzzy set theory based framework forms coalitions based on the agents' model of the learners is developed.	*Yes*	*na*

na: Not addressed

First, these research works in Table 2.1 did not explicitly utilize or address the dynamics of type change ($ca2$, $ca3$ in Section 2.1) present in the MCFP-M problem. In typical collaborative learning environments, the type change among the participating learners *always exists* due to the *inherent learning capability* of the human users. Nevertheless, designing algorithms and frameworks that *actively* pursues to accommodate and optimize the critical type changing aspects would be able to better improve the effectiveness and efficiency of the formed coalitions. For example, by distributing the expert members among the coalitions in the coalition structure, it is possible that not all expert members are able to perform optimally (loss of effectiveness for some coalitions) in their coalitions due to the other not-so-expert members. However, those expert students' interactions with their coalition members are likely to improve the non-expert group members' expertise (as discussed by Cress and Kimmerle (2008) in the context of Wikis) allowing them to form more effective and efficient coalitions over time.

Second, these research works (Table 2.1) were *not* designed to consider the probabilistic user models and their impacts ($ca3$ in Section 2.1) on the effectiveness or efficiency of the coalitions except (Vassileva et al., 1999; Bull et al., 2001; Greer et al., 2001; Vassileva et al., 2003) where the researchers focused on forming *1-to-1* groups for providing just in time peer help. However, MCFP-M is a different type of problem where its environment contains multi-person coalitions that last longer and provide more *frequent* and *complex* (e.g., one-to-many as opposed to one-to-one) interaction (learning and scaffolding) opportunities for the members than the 1-to-1 peer groups. As a result, in I-HELP, the environment dynamics that defines a coalition's ability to improve the types of its members by learning and teaching were not accommodated. For example, if a *helper-helpee*

(Vassileva et al., 1999; Bull et al., 2001; Greer et al., 2001; Vassileva et al., 2003) group is not able to solve the problem, the system could find another willing helper from the set of participating students to provide the necessary help. However, in a multi-person coalition environment like MCFP-M, members of a group are usually engaged in a longer period of interactions and such prompt solutions are not appropriate in case a coalition fails to provide the desired learning outcome for its members. To deal with such problems, and to keep up with the changing dynamics (user models, changing relationships), the multi-person coalition formation solution for MCFP-M needs to *learn* the students' models *as well as* the combination of those models (in a coalition) that would improve the effectiveness and efficiency of the formed coalitions over time. However, such learning and re-structuring of the formed coalitions considering the probabilistic impact of the models of the students and the dynamics of type change in the environment to improve the effectiveness and the efficiency of multi-person coalitions are not discussed in I-HELP. So, for such differences between the 1-to-1 peer groups and multi-person groups, the probabilistic user modeling, its considered impact, and the group formation techniques discussed in (Vassileva et al., 1999; Bull et al., 2001; Vassileva et al., 2003) cannot be readily applied to solve MCFP-M.

In electronic markets, *buyer coalition formation* is an important purchasing strategy for people who (1) need to buy small amount of goods and have limited bargaining power or (2) are interested in increasing their profit margin by obtaining volume discounts. To solve this problem, several researchers (Boongasame et al., 2009; Louta et al., 2008; Chan and Leung, 2008, 2007; Asselin and Chaib-Draa, 2006; Palopoli et al., 2006; Cornforth et al., 2004; Li et al., 2003; Tsvetovat and Sycara, 2000) investigated the buyer coalition formation in the business domain and proposed multiagent frameworks and algorithms where the agents could form coalitions *on behalf of* the human buyers they represent. Table 2.2 briefly describes these research approaches and Figure 2.1 categorizes and compares these mentioned research approaches according to their respective considerations of the critical aspects ($ca2$, $ca3$ in Section 2.1) of the MCFP-M environment.

Table 2.2: Agent-Mediated Coalition Formation Research II

(Research)	Solved Problem
(Boongasame et al., 2009)	Presented an algorithm for forming a buyer coalition with bundles of items, called the GroupBuyPackage scheme, in order to maximize the total discount. Their simulation results showed that the total discount of the coalitions in this scheme are close to that in the optimal scheme.
(Louta et al., 2008)	Proposed a dynamic multi-lateral negotiation model and construct an efficient negotiation strategy based on a ranking mechanism that does not require a complicated rationale on behalf of the buyer agents. Their proposed framework considered both contract and decision issues, is based on real market conditions, and has been empirically evaluated.
(Chan and Leung, 2008, 2007)	Proposed a distributed mechanism where agents propose incremental improvements (using private-belief based non-transferable utility) toward a stable solution. Their experiments showed that the mechanism is able to reach core stable solutions in over 97% of the cases, and b-core stable solution in almost of all cases.
(Asselin and Chaib-Draa, 2006)	Proposed a negotiation protocol for software to find the pareto-optimal buyer coalitions and minimizes worst distance to ideal among all software agents given strict preference ordering.The authors also investigated the buying behavior changes of the agents.
(Palopoli et al., 2006)	Proposed a Consumer Buying Behavior model, called $E^2 - CBB$, that considers new emergent issues, as the capability to solve semantic heterogeneity, and the adaptive presentations of Web stores to classify and compare a number of agent-based approaches for managing B2C e-commerce.
(Cornforth et al., 2004)	Proposed a novel market-based communication protocol, which governs the aggregate behavior of individual agents and subsequent emergent properties of the system. In the solution, the agents used the contract net protocol and bid to join coalitions that are the solution to the given problem.
(Li et al., 2003)	Studied the mechanism design problem of coalition formation and cost sharing in an electronic marketplace, where buyers can form coalitions to take advantage of discounts based on volume. The authors propose a coalition formation mechanism that reaches stable core of the game.
(Tsvetovat and Sycara, 2000)	Discussed a negotiation-based coalition formation as a means to formation of groups of customers coming together to procure goods at a volume discount ("buying clubs") and economic incentives for creation of such groups.

Notice that Although designed to form coalitions of human users by a set of mediating agents, these proposed solutions were not designed to *exploit* the dynamics of type change (*ca*2 in Section 2.1) or *consider* the probabilistic models of the actor agents and those

models' impact ($ca3$ in Section 2.1) on the effectiveness and efficiency of the coalitions. For example, the agents in these solutions do not have any option of engaging in the *tradeoffs* in their current task and future task rewards stemming from the improvements in their types (i.e., buyer capabilities) due to their interactions (i.e., business transactions) with the coalition members. Furthermore, in (Asselin and Chaib-Draa, 2006) and (Chan and Leung, 2008), the researchers assumed that the buying capability or available capital, which is equivalent to the *type* of an agent in MCFP-M, is accurately known by the buyer agent and there is no uncertainty in a buyer agent's ability to utilize its type during a buying transaction. Although this assumption is valid in the business-related problem domains addressed by the researchers, it does not hold for MCFP-M.

Our discussions in Section 2.3 and Section 2.2 indicate that these multiagent coalition formation research works were not designed to address the critical aspects ($ca2$, $ca3$ in Section 2.1) of MCFP-M. So, as our solution for MCFP-M (Section 1.2), we design and implement *i*HUCOFS—a multiagent-based coalition formation framework in which a set of intelligent mediator agents assist a set of human users form coalitions and scaffold those human users to optimize their effectiveness and efficiency in the coalitions over a set of tasks. The *i*HUCOFS framework's mediator agents probabilistically model the human users and consider the impact of their models on the effectiveness and efficiency of the coalitions. The mediator agents also trade off the *expected* reward of solving the current task and the *potential* reward of better coalitions in the future with the hope that such sacrifices in the present would yield *positive* type changes in the participating human users leading to more effective and efficient coalitions over time. Notice that we position *i*HUCOFS in *both* bottom-right and bottom-left quadrants of Figure 2.1. That is because of the two *different types* of implementations (Section 1.3) of *i*HUCOFS framework. *i*HUCOFS framework's ideas have been implemented in two versions: (1) VALCAM—a heuristics- and auction-based algorithm and (2) MHCF—a negotiation-based algorithm for forming student coalitions in the CSCL environment. In VALCAM, the mediator agents utilize a heuristic-based algorithm

to utilize the dynamics of type change of the participating students and *do not* consider the probabilistic models of the students and those models' probabilistic impact. Furthermore, in MHCF, the mediator agents utilize the probabilistic models of the participating students and those models' probabilistic impact on the performance of the coalitions (using a Bayesian network) to exploit the type-changing influences of the students (i.e., dynamics of type change). So, we place part of the *i*HUCOFS framework in the (1) *bottom-left* quadrant of Figure 2.1 to acknowledge its VALCAM implementation and (2) *bottom-right* quadrant of Figure 2.1 to acknowledge its MHCF implementation.

2.3   MCFP-A Coalition Formation Problem

Agent-acted coalition formation is a well-researched area in multiagent systems domain. Researchers working in the deterministic domain (i.e., without considering uncertainties) investigated the agent-acted coalition formation research approaches from a variety of perspectives to use it as a technique to solve problems and to understand the theoretical underpinnings of the related to those problems. The coalition formation research works related to the MCFP-A problem are described in Table 2.3.

Table 2.3: Agent-Acted Coalition Formation Research I

(Research)	Solved Problem
(Shehory and Kraus, 1995)	Coalition formation as a method of task allocation
(Sandholm and Lesser, 1997)	Coalition formation for computationally bounded agents
(Dieckmann and Schwalbe, 1998)	Coalition formation in dynamic environments
(Sandholm et al., 1999)	Computational complexity of coalition formation
(Vassileva et al., 2002)	Coalition formation for long term coalitions
(Breban and Vassileva, 2002)	Coalition formation in business environments (coalitions between customers and vendors)
(Chalkiadakis and Boutilier, 2003)	Coalition formation for agent coordination
(Dang and Jennings, 2004)	Optimization of social payoff in coalition formation

(Research)	Solved Problem
(Abdallah and Lesser, 2004)	Coalition formation as a method of agent organization
(Yokoo et al., 2005)	Solution properties of the coalition formation problem in open anonymous environments (e.g., internet)
(Bachrach and Rosenschein, 2008)	Coalition formation in heterogeneous task solution environment
(Sombattheera and Ghose, 2008)	Anytime coalition structure generation for coalition formation problems

Although these research works discussed in Table 2.3 were designed to solve the agent-acted coalition formation problem in a variety of environments, they were not designed to address the critical aspects ($ca1, ca2$ in Section 2.1) of MCFP-A. For example, these research approaches assume that once a set of agents with specified types form a coalition, the outcome of that coalition is *guaranteed*. However, as we have discussed for the first critical aspect ($ca1$ in Section 2.1) the outcome of a coalition is *not guaranteed* but uncertain due to the environmental influences and thus need to be considered by the agents. Furthermore, these research works do not *explicitly* consider or utilize the tradeoffs arising from the dynamics of type change ($ca2$ in Section 2.1). Due to the lack of adequate consideration of these two factors, we place these research works in the *top-left quadrant* of Figure 2.2.

Notice that the researchers do not accommodate these two critical factors mainly because they *focus* on finding the coalition structure for a given set of agents with different types for an environment to optimize some other aspects (e.g., reward) of the described problem. As a result, (1) the agents in their environments *do consider the types* of the other agents in their decision-making, (2) the agents' *types change* as a result of their interactions, but the agents *do not explicitly utilize* the dynamics of type change as a *tool* or *technique* for improving the effectiveness or efficiency of the formed coalitions. Due to the uncertainty and the dynamicity (due to the task and agent openness) of the environment, the computational complexity of the coalition formation problem is even more magnified and it is important for the agents to utilize the dynamics of type change as a *tool* or *tech-*

nique for improving the effectiveness or efficiency of the formed coalitions. Thus, although their approaches do solve their problems well, those approaches cannot be directly applied to solve MCFP-A without extending them to consider the critical aspects of the MCFP-A problem.

Researchers working in the multiagent systems domains also studied various types of agent-acted coalition formation problems in *uncertain* environments. Table 2.4 outlines the published solutions of the agent-acted coalition formation problems in uncertain environments and identifies how each solution approach addresses the critical aspects ($ca1, ca2$ in Section 2.1) of MCFP-A.

Table 2.4: Agent-Acted Coalition Formation Research II

(Research): Problem, Solution.	ca1	ca2
(Ketchpel, 1994): Coalition formation for tasks that require heterogeneous agent expertise and where the agents are uncertain about the value of a coalition. An auction-based coalition formation algorithm is proposed.	*Yes*	*na*
(Dieckmann and Schwalbe, 1998): Coalition formation for tasks that require heterogeneous agent expertise and where the agents are uncertain about the value of a coalition. Agents use non-cooperative best-reply rule to join the best reward-yielding coalition.	*Yes*	*na*
(Suijs et al., 1999; Suijs and Borm, 1999): Coalition formation for a set of agents with random finite expected reward for a set of tasks. Theoretical foundations for games with this restricted form of uncertainty i.e., the existence of core was discussed.	*na*	*Yes*
(Klusch and Gerber, 2002): Coalition structure generation for dynamic environments where tasks and agents change. Negotiation-based distributed coalition formation scheme was provided for developing efficient coalition formation algorithms in dynamic environments.	*Yes*	*na*
(Kraus et al., 2003): Coalition formation for tasks in heterogeneous, time-constrained tasks with incomplete information. Heuristic-based coalition formation protocol and strategies was designed.	*Yes*	*na*
(Soh and Li, 2003): Coalition formation in dynamic, uncertain, and noisy environments. Multi-phase (planning, instantiation, and evaluation) and negotiation-based coalition formation algorithm was designed.	*Yes*	*No*
(Griffiths and Luck, 2003): Coalition formation for agents with heterogeneous objectives in a dynamic, uncertain, and noisy environment. Clan-(trusted agent group) based coalition formation mechanism was proposed.	*na*	*Yes*

(Research): Problem. Solution.	ca1	ca2
(Blankenburg and Klusch, 2004): Forming coalition of the most reliable agents in repeated coalitional games. A trusted kernel-based, encryption-enabled coalition formation protocol was designed.	*Yes*	*na*
(Gaston and DesJardins, 2005): Finding effective and efficient social network in dynamic environment. Distributed coalition formation algorithm for the agents was provided.	*na*	*na*
(Chalkiadakis, 2007; Chalkiadakis and Boutilier, 2008): Repeated Agent coalition formation under uncertainty. Negotiation-based and Bayesian learning enabled distributed coalition formation protocol was provided.	*Yes*	*Yes*
(Wang and Singh, 2007; Erete et al., 2008; Hang et al., 2008, 2009; Hendrix et al., 2009) Modeling the trust of other agents. Various social and communication-strategy-based distributed algorithms are discussed that could capture an agent's local estimate of the other agents' performance or behavior for a specific task or tasks in the form of trust.	*na*	*Yes*
(Stone et al., 2000; Kuhlmann et al., 2006; Rettinger et al., 2008). Modeling teammates in the robocup soccer. Learning-enabled and statistical modeling of teammates are designed.	*na*	*Yes*

na: not addressed
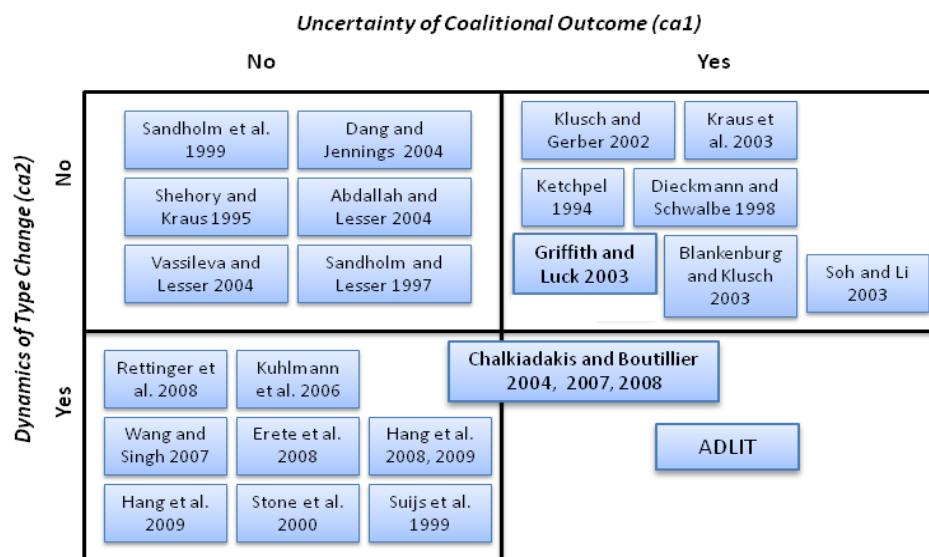
Figure 2.2: Categorization of Agent-Acted Coalition Formation Research

Although there were several significant research approaches towards solving the agent-acted coalition formation problem in various settings (Table 2.4), none of these research approaches were designed to adequately address *both* of the critical aspects of the MCFP-

A problem. Figure 2.2 also categorizes these research approaches discussed in Table 2.4 in four quadrants according to their considerations of the critical aspects of MCFP-A.

In the MCFP-A environment (Section 2.1), an agent has the opportunity to exploit the dynamics of type change to improve its own rewards over time. That means the agents have another new dimension besides the basic *exploration vs. exploitation* tradeoff (Kaelbling et al., 1996) common in typical reinforcement learning scenarios to improve its own rewards over time. Notice that, the change in the types of the agents *do occur* in a variety of coalition formation environments as presented in Table 2.4 and Figure 2.2. Notice that the solutions proposed by some of the multiagent researchers (Griffiths and Luck, 2003; Chalkiadakis, 2007; Chalkiadakis and Boutilier, 2008) have considered the impact of joining a coalition on the types of the actor agents and the tradeoffs arising from such an impact. Although not an exact match, this consideration is similar to the notion of agent types in our research. However, an agent's ability of *deliberately influencing type change* in its coalition members' types and the tradeoffs associated with such choices have *not* been considered.

To summarize, the state of the art in multiagent coalition formation have *not* exploited the tradeoffs associated with the situation where an agent can influence its coalition members to change their types by working in the same coalition due to implicit and/or explicit learning. There are several repercussions of not considering the impacts and the tradeoffs associated with such influences of the agents. First, not considering these two types of learning and the associated tradeoffs would mean that the agents will not be able to learn and improve their types efficiently. Second, note that forming coalitions in an uncertain environment is an intractable problem that requires at least exponential cost to solve it (Chalkiadakis, 2007) and that the improved types of the agents result in improved performances of coalitions. Thus, the ability to improve the types of the agents efficiently through the use of implicit and explicit learning in this computationally demanding environment is a *potential leverage* for the system designer that can provide higher performance without a high computational cost. As a result, not taking advantage of this leverage would result in

inefficient solutions of the ADLIT problem. Third, in an agent-human hybrid work environment, the human users learn from their interactions and change their types (e.g., become more skilled in solving tasks by interacting with other group members (Stahl et al., 2006)) over time. Failure to exploit that type-changing nature of the human users while forming coalitions in such a hybrid system may result in coalitions that have poor performances. Finally, the agents whose decision making processes do not exploit efficiently or reap the benefits from the learning or improvement of types naturally occurring in from the coalitions will be deemed unhelpful by the human users making the agent-human cooperation or coordination quite difficult.

To adequately address both these two critical aspects $(ca1, ca2)$, we propose the ADLIT framework (Section 6) as a solution to the MCFP-A coalition formation problem. Our overall approach to solve the MCFP-A coalition formation problem is based on the influence or consideration of agent learning on its choice of coalition (from the viewpoint of short-term vs. long-term rewards tradeoff) to form better coalitions (in terms of earned rewards) in an uncertain environment where the agents' types change due to their learning experiences. And thus, when an agent considers which coalition to join, it considers how its own learning and teaching with its coalition members may change its own type *and* the types of its coalition members. Based on those type changes, the agents estimate the tradeoff between the reward for the current task and the reward for future tasks and choose their coalitions and learning interactions according to that tradeoff value calculation.

Chapter 3

MCFP-M: *i*HUCOFS Framework

In this chapter we describe the Integrated Human Coalition Formation and Scaffolding (*i*HUCOFS) framework (Soh and Khandaker, 2007; Khandaker and Soh, 2008) and its implementation in the MHCF algorithm for human coalition formation. First, in Section 3.1, we describe the *i*HUCOFS framework environment using a set of assumptions. Then using those assumptions, we derive a set of design principles in Section 3.2 for our solution. Then, using those design principles, we discuss the realization of our Multiagent Human Coalition Formation (MHCF) algorithm for forming human coalitions in Section 3.3.

3.1   Assumptions

Assumption 1 (Task)

There is a set of tasks in the environment that needs to be solved. Each task contains some reward that is available upon solving it. Not all tasks are available or known to the human users at the creation of the environment.

Assumption 2 (Human Coalitions)

There is a set of human users in the environment who form disjoint coalitions to solve the tasks to earn rewards as a coalition. That coalitional reward is then distributed among the members according to some distribution scheme agreed upon by the members while

forming the coalition. How well a coalition is able to solve an assigned task depends on the *skills* or *types* of its members.

Assumption 3 (Agents)

There are a set of mediator agents in the environment who help the human users form or join coalitions. There is also a system agent that assign tasks to the human coalitions and provides rewards to the members of the formed coalitions when they solve that assigned task. Figure 3.1 shows the conceptual setting of the *i*HUCOFS framework.
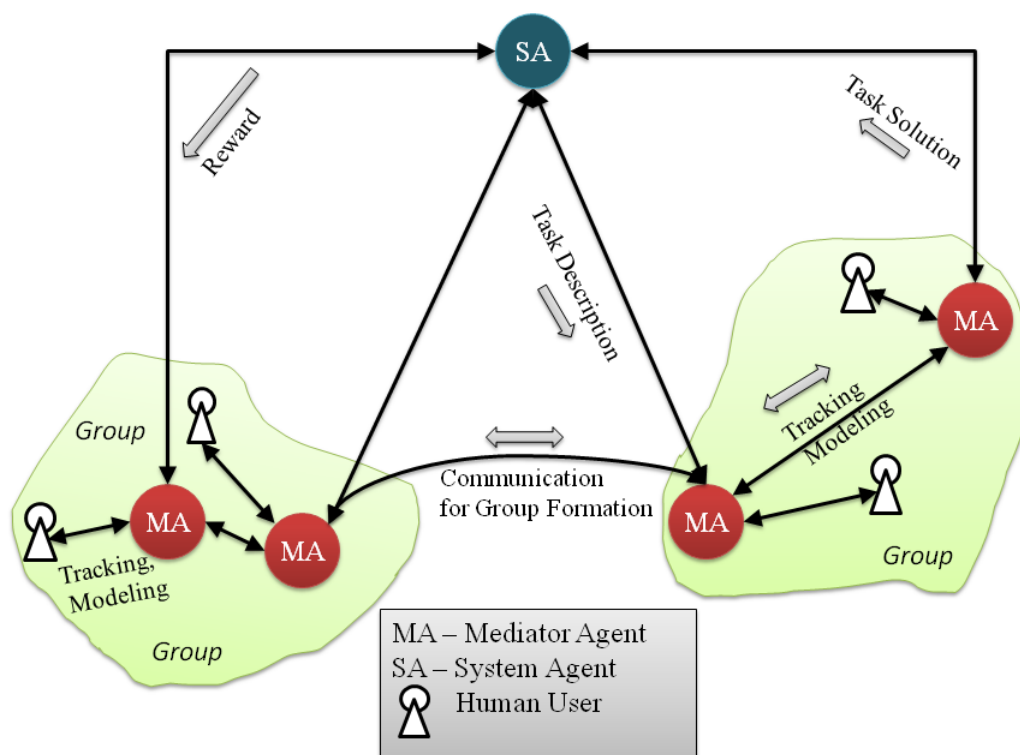
Figure 3.1: Conceptual Architecture

Assumption 4 (Learning)

The human user participating in the task solving activities of their respective coalitions are capable of learning from their experiences and may *change* their behavior over time due to their learning. Human learning in a collaborative setting can come in various shapes and forms (Inaba et al., 2000):

- **Learning by Observation**. The users learn indirectly by observing other learners' learning process. This type of learning can be facilitated by the mediator agent by putting the human user in a group that contains users with similar deficiency of knowledge about a certain task.

- **Learning by Teaching/Guiding**. Learning by teaching occurs when a human user learns or refines his or her own knowledge by teaching other group members. This type of learning is particularly useful in CSCL settings where the students learn by teaching each other.

- **Learning by being Taught**. This is the simplest type of learning where a human user learns when he or she is being taught by someone else. This type of learning is particularly useful in CSCL settings where the teacher and students have differences in their knowledge about the task they are solving.

- **Learning by Reflection/Self-Expression**. This type of learning occurs when a human user *rethinks* his or her own solution and analyzes his or her self-thinking process (Schon, 1987, p. 28).

- **Learning by Apprenticeship**. In learning by apprenticeship, the expert teacher shows the apprentice (student) how to do a task, watches as the apprentice practices portions of the task, and then turns over *more and more* responsibility until the apprentice is proficient enough to accomplish the task independently (Collins et al., 1991).

- **Learning by Practice**. This type of learning occurs when a human user applies his or her existing knowledge to solve an assigned problem. This type of learning is very common in situations where each human user contributes to the solution of the assigned problem by working on it. However, there may be human users who are *free-riding* i.e., depending on the competent and the knowledgeable users to solve

the assigned problem. As a result, these users *do not* learn by practice.

- **Learning by Discussion**. This type of learning occurs when the human users discuss a topic with each other. Notice that this type of learning is basically a sequence of Learning by Observation, Learning by Teaching, Learning by being Taught, Learning by Reflection/Self-Expression, Learning by Practice with the exception that the roles of the human users are *dynamic* in Learning by Discussion.

Assumption 5 (Scaffolding)

We refer to scaffolding as a form of *assistance* provided to a human user by a teacher or peer where that assistance helps that human user perform a task that he or she would normally not be able to accomplish alone. We assume that the system agent and the mediator agents try to scaffold the human users to change their behaviors to improve their performance as individuals and as coalition members.[1] This scaffolding can be of two types:

- **Explicit**. A human user is guided by the assigned mediator agent explicitly to help him learn how to change his or her behavior for the current task in the current coalition.

- **Implicit**. The system agent or the assigned mediator agent constructs environment states (implicit help) that allow the human user learn how to improve his or her behavior in the future coalitions. Here, the implicit help can be provided in the form of putting the human user in a coalition where the other coalition members can engage in teaching and learning interactions with that that human user to improve his or her behavior.

Assumption 6 (Uncertainty)

The behaviors of the human human users i.e., their individual actions in a coalition, are

---

[1]The agent-mediator instance of coalition formation problem may also involve a *changing* agent autonomy over time. Switching between the role of an adviser and a representative. In our research, we focus only on the coalition formation- and scaffolding-related aspects of the agent-mediator cases of the ADLIT. A research approach that investigates the issue of adjustable autonomy could be found in (Screrri et al., 2002; Maheswaran et al., 2004).

stochastic–meaning the behavior of the human human users *cannot* be accurately modeled; only modeled with probability.

Assumption 7 (Reward)

The rewards achievable by the human coalition can be divided into two generic categories: (1) current-task rewards and (2) future-task rewards. The current-task reward is a human human user's share in the coalitional reward his or her coalition has earned by solving an assigned task and can be *exactly* calculated as soon as the coalition completes the assigned task. The future-task reward is an *estimation* of the current-task rewards of the future tasks that is expected to be available due to the improvement in a human human user's behavior due to his or her learning. The value of the future-task reward for a human human user's participation in his or her coalition's effort to solve an assigned problem can be calculated as an expected value.

Assumption 8 (Attribute Categorization)

We assume that the skills or knowledge of the human users can be divided into a finite set of discrete categories. For example, the knowledge of the human users can then be divided into high, medium, and low. This assumption arises from our observation that the human skills regarding various types of tasks can often be identified by domain experts and measured by the psychometric measurement techniques. For instance, the knowledge of a human user regarding a particular topic can be determined by pre-tests, his or her motivation can be determined by psychometric tests (Vallerand et al., 1992). However, these psychometric tests themselves often suffer from measurement errors which reduces the accuracy of their measurements (Carmines and Zeller, 1979).

3.2 Design Principles

According to our assumptions of the *i*HUCOFS framework (Section 3.1) we use the following principles while designing the MHCF algorithm for coalition formation:

User Modeling

The mediator agents must be able to model different user motivations, behaviors, and utilities and should be able to capture the change in their behavior over time due to learning. This design principle addresses Assumption 2, Assumption 3, and Assumption 4 (Section 3.1).

Satisficing Solution for the Current Task

The system agent and the mediator agents must be able to take decisions with incomplete information or noise. Further, since outcomes are uncertain, it could be costly for the agents to devise an optimal solution only to find out that it does not lead to the expected outcome. Thus, this motivates the agents to make do with what they know, and sub-optimal but satisficing solutions may be preferable. This design principle addresses Assumption 2 (Section 3.1).

Learning Mechanism

To overcome the *noisy* environment and *incompleteness* of the available information, the mediator agents should use a learning mechanism to *filter out* the necessary information to achieve the required level of accuracy. The learning mechanism could include typical agent learning (e.g., reinforcement learning) and also the multiagent learning where the mediator agents learn from each other's experience (e.g., learning by discussion and learning by observing). This design principle addresses This design principle addresses Assumption 6 (Section 3.1).

Scaffolding

The proposed *i*HUCOFS environment is noisy and has incomplete information and uncertain outcomes. These characteristics imply that the mediator agents may not be able to collect accurate data to form the most suitable coalition. However, we know that human users may learn and improve their behavior when scaffolding is provided. Therefore, the mediator agents should spend more time and computational resources for scaffolding. Since the mediator agents' beliefs about the environment may contain inaccuracies, spend-

ing resources for forming the *perfect* coalition *may not* yield the best outcome in terms of utilities for the human users. On the other hand, spending more resources for scaffolding would mean that the human users would be able to improve their behavior and in turn improve the outcome for the current and future coalitions. This design principle addresses Assumption 4 and Assumption 5 (Section 3.1).

Exploiting the Formation Vs. Scaffolding Tradeoff

Say the system agent is forming a coalition to solve a task. When the task is completed, a mediator agent is able to collect the rewards for its assigned human user. To optimize the reward, that mediator agent could decide to spend more time and computational resources to *form* or *join* the perfect coalition, i.e., find a coalition that would yield its human user the optimal reward for the current task. On the other hand, that mediator agent may choose to spend more for *scaffolding* the formed coalition with the hope that its assigned human user is able to improve its capabilities and skills so that he or she is able to earn higher rewards for future tasks. This tradeoff arises from the fact that the coalitions that optimize the reward for the current task for a human user *may not* be the coalition that provides the optimal learning opportunities to increase the rewards for the future tasks. This design principle addresses Assumption 4, Assumption 5, Assumption 6, and Assumption 7 (Section 3.1).

3.3 MHCF Algorithm

Based on the design principles (Section 3.2) of the *i*HUCOFS framework, we have designed the *Multiagent Human Coalition Formation* (MHCF) algorithm (Khandaker, 2009; Khandaker and Soh, 2010*b,c*) for forming human coalitions.

3.3.1 Environment

The MHCF environment E is denoted as a 5-tuple $\langle S, A, G, T, R \rangle$. Here,

- $S = \{s_1, \ldots, s_{ns}\}$ is the set of human users

- $A = \{a_1, \ldots, a_{ns}\}$ is the set of agents where each agent a_i is assigned to a human user s_i

- $G = \{g_1, \ldots, g_{ng}\}$ is the set of human coalitions

- $T = \{t_1, \ldots, t_{nt}\}$ is a set of tasks which the human coalitions collaborate to solve, and

- R is a 2-tuple $R = \langle R_{ct}, R_{ft} \rangle$ where R_{ct} and R_{ft} are two real-valued functions that estimate the probability of a human user's current-task and future-task rewards when he or she joins a coalition

Here R_{ct} is defined as a function:

$$R_{ct} : f(sm_{g,t}, t_j) \rightarrow \mathbb{R} \tag{3.1}$$

furthermore, R_{ft} is defined as a function:

$$R_{ft} : f(sm_{g,t}, t_j) \rightarrow \mathbb{R} \tag{3.2}$$

In Equation 3.1 and Equation 3.2, $sm_{g,t}$ is a set of the models of the members of the potential human coalition g at time t where the coalition g is being formed to solve the task t_j, and we define,

$$sm_{g,t} = \{sm_{k,t} | s_k \in g\} \tag{3.3}$$

Note that the functions R_{ct} and R_{ft} use the model of the members of a potential coalition to calculate the *expected* current-task and future-task rewards for a human user to if he or she joins that potential coalition to solve a task. While the current-task reward allows

an agent to estimate what its human user will receive after participating in a coalition, the future-task reward is what the agent expects the human user will gain in the future, taking into account the likelihood of the student working with some of the same coalition members on similar assignments again. This allows MHCF to *look ahead* while encouraging students to perform sufficiently well in their current tasks.

3.3.2 MCFP-M Problem

We formally define the MCFP-M problem as:

> Given a set of human users S and a set of tasks T, the human coalition formation problem for the mediating agents A refers to forming or joining coalitions *on behalf of* their human users so that, (1) those human users S are able to solve task T effectively and (2) the human users are able to improve their types ($sm_{i,t}$) to earn higher rewards for the future tasks.

The solution to the MCFP-M problem comes from the MHCF algorithm for coalition formation where the mediator agents negotiate to find a coalition for their assigned human users.

3.3.3 MHCF Algorithm for Coalition Formation

Coalition formation in MHCF occurs in a set of negotiation rounds where in each round, one agent is *randomly* selected to act as a proposer who negotiates with other agents in the framework to form a coalition for its assigned human user. The negotiation of an agent is carried out in the *Proposition*, *Consideration*, and *Notification* steps. In the following list, we describe these steps in details and Algorithm 3.1 summarizes the coalition formation process.

- **Proposition** - In the proposition step, the proposer agent chooses $(n_{sg} - 1)$ other agents (n_{sg} is the minimum coalition size) and proposes a coalition which includes

the human users assigned to those chosen agents. The proposal from an agent a_i to agent a_j is: $P = \langle sm_{g,t}, r_{ct}, r_{ft}, t_j \rangle$ where $sm_{g,t} = \{sm_{k,t} | s_k \in g\}$ is a set of models (*not the ids*) of the students in the proposed coalition g, $r_{ct}, r_{ft} \in \mathbb{R}$ are the expected current-task and future-task rewards (Equation 3.1, Equation 3.2) for the task t_j calculated from the *perspective* of a_i.

- **Consideration** - In this step, the proposed-to agent a_j first compares its model sm_j stored by the proposer agent a_i with its own model of human user s_j. If that model is not updated, in other words, if agent a_i is *unaware* of the recent changes in the model of the human user s_j, the responding agent rejects the proposal and sends the updated model of s_j to the proposer. Note that this notification from the responding agent allows a proposer to have updated view of the other potential members during the coalition formation round. If the proposer has the updated view of the responding agent's assigned student, the responding agent compares the expected current-task and future-task reward values of the proposed coalition to its current coalition. The responding agent leaves its current coalition to join the proposed coalition if the *weighted sum* of current-task and future-task rewards is larger for the proposed coalition g, i.e.,

$$r_{g,t} = w_{ct} \cdot r_{ct,t} + w_{ft} \cdot r_{ft,t} \tag{3.4}$$

In Equation 3.4, $r_{ct,t} = R_{ct}(sm_{g,t}) \in \mathbb{R}$ and $r_{ft,t} = R_{ct}(sm_{g,t}) \in \mathbb{R}$

- Notification - If all of the chosen agents agree to join the proposed coalition, the proposer sends out a confirmation message to them notifying that they are now in the newly formed coalition. Otherwise, if *any* of the responding agent disagrees, the proposer stops the negotiation process and waits for some other agent's proposal or its next turn to join a coalition.

Once the negotiation rounds end, the agents notify their assigned human users about their respective newly formed coalitions and the details of the task they will collaborate to solve and the collaboration process begins.

Notice that in the proposition step of the negotiation process, the proposer does not reveal its preference of the coalition members-as it only reveals the types or models of those coalition members without identifying them-to its counterpart, the responding agent. Such a revelation would have put the proposer agent at a disadvantage since the responding agent could, after getting the proposal, propose its own coalition with the members mentioned in the proposal without including the original proposer.

Algorithm 3.1 MHCF Algorithm for Coalition Formation

1: **if** Chosen as a proposer **then**
2:     Randomly choose $S_{ne} \subseteq S$ (as neighborhood agents)
3:     Receive updated member models from neighborhood agents S_{ne}
4:     Choose coalition $SM \subseteq S_{ne}$ where sm are models members SM and $\underset{SM}{\arg\max} R_f t(sm_{g,t}) + R_f t(sm_{g,t})$
5:     **for all** $a \in A$ **do**
6:         Send proposal $P = \langle sm_{g,t}, r_{ct}, r_{ft}, t_j \rangle$ to a
7:     **end for**
8:     **if** All agents $a \in A$ accept **then**
9:         Form coalition SM
10:     **end if**
11: **else if** Received proposal P **then**
12:     **if** $r_{g,t}$ is higher for coalition $sm_{g,t} \in P$ than current coalition **then**
13:         Join $sm_{g,t}$
14:     **end if**
15: **end if**

3.4   Learning for Coalition Formation

In this section we describe the multiagent learning techniques used by the human user agents in the MHCF environment. To describe the agent learning, we first discuss the properties of the MHCF environment that makes the learning employed by the agents a difficult problem to solve. Then we discuss the learning goal of the agents in the MHCF

environment and provide details of the processes with which the agents try to achieve those goals.

3.4.1 Learning Problem

Our goal is to design and develop a multiagent framework that is applicable in real-world human coalition formation environments. Since we do not aim to find the exact solution of the coalition formation problem, our aim is to design the framework in such a way that, once deployed, the agents are able to interact with the environment through their actions, learn by observing the consequences of their actions and improve their actions over time. Our focus is to develop an environment where the agents work together as a cooperative multiagent system and help each other learn. One way to implement such learning technique is reinforcement learning. However, the agents in such a real-world environment face the challenges like multiagent learning designers face. One of those problems is that the state and action spaces are continuous. As a result, designing and implementing a multiagent reinforcement learning environment becomes difficult and sometimes impossible (Busoniu et al., 2008). Furthermore, real-world human coalition formation contains incomplete and uncertain information which complicates the learning problem for the agents further. In our research work, we tackle the learning problem for coalition formation using: (1) domain knowledge and heuristics regarding the environment of a generic coalition formation problem and (2) a multiagent systems' ability to utilize cooperative learning. We include the domain knowledge of the human users working in the coalition formation problem by the following assumptions about the problem domain. According to our setup of the MHCF environment, the mediator agent's learning goal is to estimate the probability that a human user with a given model (i.e., a set of attribute values) and a given scaffolding action is able to complete an assigned task in the joint coalitional action.

3.4.2   Bayesian Cooperative Learning

Notice that, to form coalitions in MHCF, in other words, to decide who to form the coalition with, each mediator agent must learn: (1) the model of other mediator agent's human users, (2) the probabilities that describe given a human user's model, his or her agent's scaffolding, e.g., what is the probability he or she is able to complete the negotiated assigned individual task, and (3) the probability that a human user's model would change after working in a coalition. In the MHCF environment, one mediator agent is assigned to a human user and may not be able to observe the probabilities regarding the values of human user models or scaffolding that is not represented by its assigned human user. That is because, not all human users would have the same model, not all tasks would have the same required joint coalitional actionset, and finally, not all human users would have the opportunity of working with every other possible type of human users. For example, in a CSCL environment, a mediator agent assigned to a student with medium knowledge category may not be able to observe the behavior of a human user with a model containing high knowledge category. This implies, it will be difficult for the mediator agents to learn those probabilities through their own experience/observation only. However, if we try to frame the problem of learning these probability values as a cooperative learning problem, then it becomes easier for the agents to solve since they share burden of learning among themselves. For example, in the CSCL case, if the agent with high knowledge category human user and the agent with medium knowledge category human user share each other's observations, both agents would be able to refine their probability estimates regarding human users with high and medium knowledge category. So, with our approach of cooperative learning, two questions arise: (1) how the probabilities regarding the MHCF environment would be represented by individual agents and (2) how would the agents share their learned probability values with each other. For modeling the probabilities of the environment, the mediator agents in MHCF environment use Bayesian networks, and for sharing the learned probability values, the mediator agents use cooperative learning.

As defined in (Xiang, 2002, p. 28), a Bayesian network is a triplet $\langle V, G, \mathcal{P} \rangle$ where V is a set of variables, G is a connected directed acyclic graph whose nodes correspond one-to-one to members of V such that each variable is *conditionally independent* of its *non-descendants* given its parents and $\mathcal{P}$ is a set of probability distributions. Say the parent of nodes $v \in V$ is denoted by $\pi(v)$, then,

$$\mathcal{P} = \{P(v|\pi(v))|\forall v \in V\} \tag{3.5}$$

Furthermore, the following holds:

$$P(V) = \prod_{v \in V} P(v|\pi(v)) \tag{3.6}$$

The use of Bayesian network allows the agents to represent the probabilities of the environment in a compact way. First, using Bayesian network, the agents are able to learn the *likelihood of the events* that happen in the coalition. For example, observing the assigned human user's behavior, an agent is able to answer the question: "what is the likelihood that given a user model and the coalition, the assigned human user would be able to complete the assigned part of the coalitional task?" Second, a Bayesian network provides a compact representation of *relevance* of the various elements to the agents directly and qualitatively before the agents make any observations or numerical assessments. For example, it allows the agent designers to *encode* the dependence relationship among the elements of the MHCF environment. Finally, the *causation* encoded in a Bayesian network allows the agent designers to define the *structure* of the relevance relationships among the elements of the MHCF environment. That encoded causation allows the Bayesian network to eliminate superfluous information and represent the non-transitive and induced dependencies. For example, in MHCF, causation would allow the mediator agents to not care about information that is not related to its assigned human user's current-task and future-task rewards (e.g., the current-task and future-task reward of human users in other coalitions). Furthermore,

the encoded causation can allow the mediator agents to reason by combining independent elements, (e.g. the coalitional task and its assigned human user's model) to the dependent elements (e.g., the current-task reward of its assigned human user) of the MHCF environment.

3.4.3   Use of a Bayesian network

Figure 3.2 shows an example Bayesian network that could be used by the agents in MHCF to represent/learn the current task and future task functions.
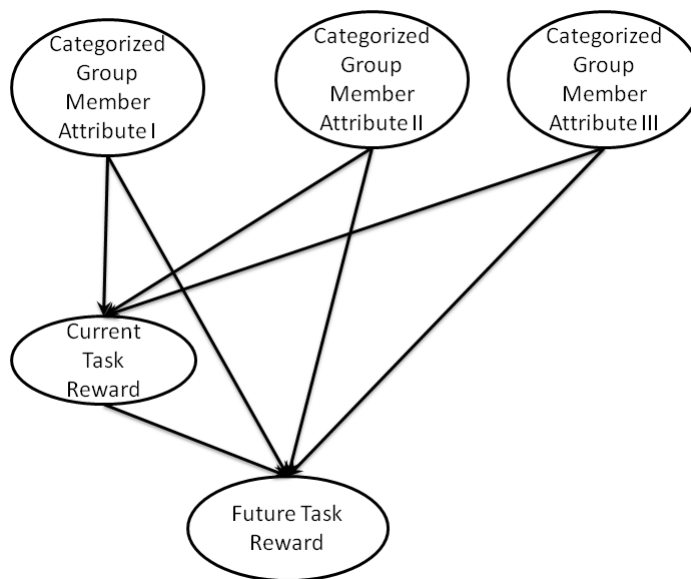
Figure 3.2: Example of Bayesian network Structure

In the MHCF environment, we assume that the mediator agents are able to observe and learn from its assigned human user's actions in the coalition. However, these agents may not be able to observe the actions of the other human users in the coalition. As a result, mediator agents would not be able to learn how the model and scaffolding affect their individual actions. Also, to improve the current-task and future-task rewards of the assigned human user, a mediator agent needs to learn the Bayesian network for all values of the models, tasks, and scaffolding. One way the agents could learn this Bayesian network

more efficiently and *more accurately* is by: cooperative or divided learning. Using this strategy, the mediator agents *cooperatively* build a Bayesian network that maps a potential human user groups' model to their collaborative learning outcome, i.e., represents R_{ct}, R_{ft}. In other words, through communication, all the participating mediator agents maintain a *single* Bayesian network and together learn the probabilistic model of the environment. Figure 3.3 shows the structure of the cooperatively built Bayesian network.
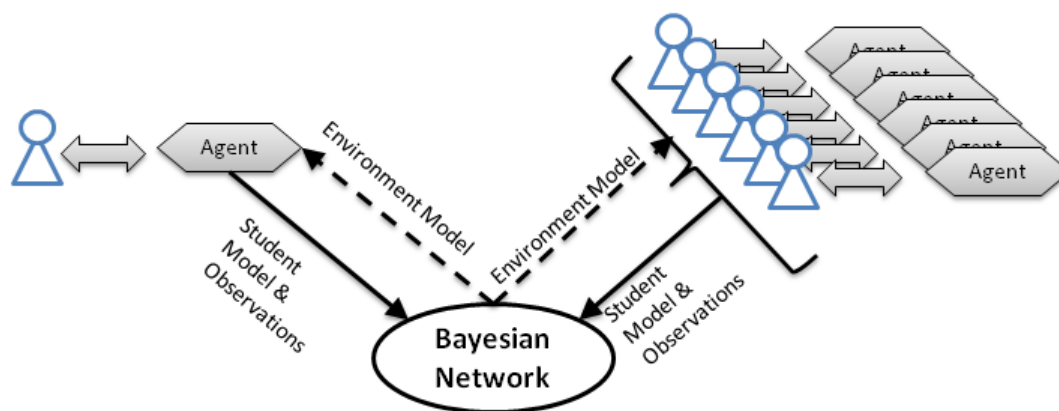
Figure 3.3: Cooperative-Built Bayesian network

The Cooperatively built Bayesian network is built and maintained using the following steps (Figure 3.4):

- **Initialization**—Initialization is done by (a) setting uniform priors - i.e., setting equal current-task and future-task outcomes for all inputs, or (b) using previously earned rewards by the human users.

- **Update**—When the coalition completes the task, the group receives the reward which is used to update the Bayesian network probabilities.

Notice that each mediator agent is able to observe different human user models (i.e., the model of their assigned human users) simultaneously and thus together divide the task (cf. division learning in (Dillenbourg, 1999) of observing the entire set of possible model-
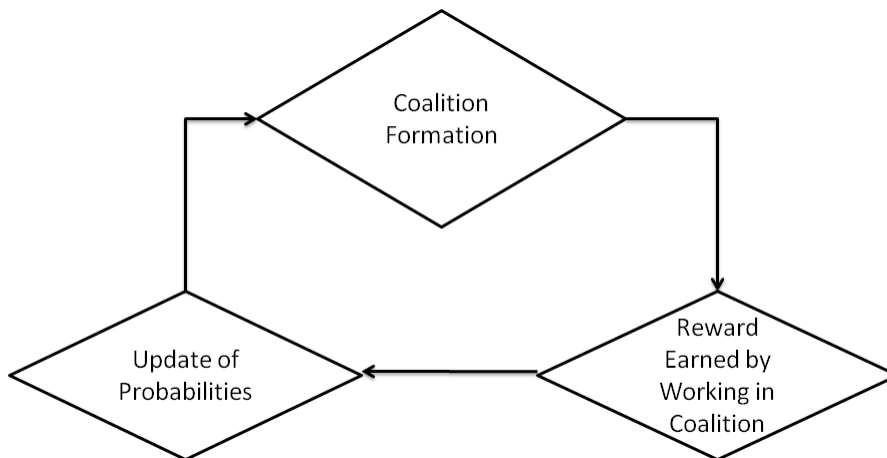
Figure 3.4: Update of Bayesian network

output combinations. This learning strategy is designed to allow the agents to learn the dynamics of the environment better.

3.5   Implementation in SimCoL and ClassroomWiki

Our implementation of *i*HUCOFS consists of our development of the collaborative learning simulation toolkit SimCoL (Chapter 4) and the instantiation of the MHCF algorithm for forming student coalitions in ClassroomWiki (Chapter 5)—a web-based, multiagent-enabled collaborative writing tool. Notice that, our goal of implementing SimCoL is to investigate and validate our assumptions (e.g., impact of group formation and scaffolding on student performances) in the *i*HUCOFS framework. Furthermore, our instantiation of the MHCF algorithm in ClassroomWiki validates MHCF's impact in terms of improving the performances of the formed student coalitions.

Chapter 4

MCFP-M: SimCoL

In this chapter, we describe *Simulation Tool for Collaborative Learning* SimCoL-a multia-gent application for simulating the collaborative learning of a set of students in the CSCL environment. The inspiration source of our research work here is CSCL environments that combine research ideas from psychology (especially educational psychology), education, and computer science to create an online collaborative learning environment for students. The rest of this chapter is organized as follows. Section 4.1 presents a set of learning theo-ries and observations based on the individual and collaborative human learning process and define the scaffolding of students in a collaborative learning environment. In Section 4.2, we use the observations in Section 4.1 to design the agent that represents the teacher, the agents that represent the students, and the tasks in the SimCoL environment. Then in Section 4.3, we describe how the SimCoL environment was realized using Repast—a mul-tiagent simulation tool. Section 4.4 describes our experiment setup and results.

Notice our use of the following terminologies in this chapter: an agent that simulates the teacher's behavior in SimCoL is denoted as a *simulated teacher* or *teacher* while an agent that simulates a student's behavior is called a *simulated student* or *student*. An agent that *assists* the students in forming groups is called a *student assistant agent*, while one that *assists* the teacher is called a *teacher assistant agent*.

4.1 Categories of Student Learning

In this section, we discuss definitions, theories, and empirical observations regarding three different aspects of students' learning processes: (1) individual learning, (2) peer-based learning, and (3) collaborative learning in student groups. Using these learning theories, we derive a set of observations that are used in Section 4.2 to build agents to simulate the student collaborative learning behavior in a CSCL classroom.

4.1.1 Individual Learning

We use "learning" to refer to the improvement in a student's knowledge or expertise on a topic or skill, which could be topic-specific, e.g., learning how to solve differential equation, or topic-independent, e.g., teamwork or communication skills. According to learning theories (Ellis et al., 1997; Shell and Brooks, 2007) the four main elements that affect how a person learns are: (1) what the student already knows (knowledge), (2) how able/intelligent the student is (ability), (3) how motivated the student is (motivation), and (4) the emotional state of that student (emotion). The cognitive components that represent these factors are: (1) the crystallized intelligence as accumulated knowledge stored in long-term memory, (2) fluid intelligence as represented by working memory capacity, and (3) motivation as represented by working memory allocation (Shell and Brooks, 2007), and (4) emotional state (Ellis et al., 1997). Next, we define these elements in greater detail. Shell and Brooks (Shell and Brooks, 2007) use the term knowledge to refer to the accumulated knowledge in a student's long-term memory.

The ultimate result of learning would occur as the improvement of the knowledge of the students. Shell and Brooks (Shell and Brooks, 2007) use ability to represent the cognitive ability or intelligence of a person. They suggest that there are two different parts of ability: fluid intelligence and crystallized intelligence. The fluid intelligence is a fixed entity that deals with general cognitive capacity and crystallized intelligence represents the

accumulated knowledge of the student. Furthermore, the fluid intelligence is basically the working memory of a student (Shell and Brooks, 2007). However, there is a difference between the *absolute* working memory capacity a person has and the amount of working memory capacity he or she *has available* at a particular time for a particular task. The behavior of a person while working on a task and the improvement in his or her knowledge due to learning by working on that task depend upon the amount of working memory that person has available at that time. Further, the amount of working memory available to any person at a time is determined by: (1) his or her existing knowledge for that task, (2) his or her motivation to work on that task, and (3) emotion (Shell and Brooks, 2007). Motivation determines why we do what we do (Shell and Brooks, 2007). In other words, motivation is the process whereby goal directed behavior is instigated and sustained. Finally, the emotion of a student determines whether the students are feeling happy or sad. So, we write our first observations as:

Observation 1: A student's improvement of knowledge of a topic is mainly affected by: (1) his or her existing knowledge, (2) ability, (3) motivation, and (4) emotion.

Observation 2: The amount of working memory available to a student determines how much he or she can learn.

Observation 3: The working memory of a student interacts with his or her prior knowledge and new information (regarding a task) to produce learning and behavior.

Observation 4: A student's available working memory for a task can be described as his or her ability for that task.

Although the aforementioned four components that affect learning are cognitively distinct from one another, there are *combinatorial* effects (Shell and Brooks, 2007): (1) the prior knowledge stored in the long-term memory *interacts* with the working memory to produce learning, (2) available amount of working memory *limits* how much prior knowledge and information can be used/activated at any time, (3) the amount of working memory *is determined* by motivation, extent of prior knowledge, and emotion, and (4) as knowledge

increases, it *increases* the effective working memory capacity allowing acceleration of future learning processes. Finally, according to the recent research work on perceptual and motor acquisitions (Fitts, 1964; Anderson, 1982) the pace of skill acquisition for a learner *accelerates* in the beginning and *slows down* to a stable state, leading to:

Observation 5: A student's available working memory for a topic is proportional to his or her: (1) knowledge on that topic, and (2) motivation to learn that topic. Furthermore, this available working memory is inversely proportional to the emotional state of that student.

Observation 6: As the knowledge of a student on a particular topic increases, his or her learning outcome for that topic would increase at the beginning and slow down to a steady state after a certain amount of time.

4.1.2   Peer-Based Learning

When a student is working with his or her peer to solve some assigned task, the student and the peer may learn from each other about that task. The possible learning scenarios between two interacting peers are described by (Inaba et al., 2000)(Table 4.1): learning by observation, learning by teaching/guiding, learning by being taught, learning by reflection/self-expression, learning by apprenticeship, learning by practice, and learning by discussion. From these peer-based learning scenarios, we observe that the prior knowledge of the participating students plays an important role in deciding what type of learning scenarios may occur. For example, learning by teaching (and learning by being taught) is more common among two students where one student with more prior knowledge teaches his or her peer who has less prior knowledge. Furthermore, a large difference between two interacting students' prior knowledge about how to solve a certain task can hinder their learning. This effect is described in Vygotsky's *zone of proximal development* (ZPD) theory (Vygotsky, 1978). So we write our next observation as:

Observation 10: Two students may learn about a topic from their interactions (Table 4.1) when the content of prior knowledge they have are not too different from one

another.

Table 4.1: Possible Learning Scenarios among Peers

Observation	Student-Peer Knowledge	Learning by
7	High-High	Observation, Reflection, Practice, & Discussion
8	High-Low or Low-High	Observation, Teaching, Being Taught, Reflection, Practice, & Discussion
9	Low-Low	Observations

4.1.3 Collaborative Learning

The term "collaborative learning" is an instruction method in which students at various performance levels work together in small groups toward a common goal (Dillenbourg, 1999). Stahl (2004) describes the following properties of collaborative learning:

Observation 11: The collaborative knowledge building is a cyclic process that feeds on itself.

Observation 12: This collaborative knowledge building cycle is a hermeneutic cycle, meaning, "one can only interpret what one already has an interpretation of".

Observation 13: Individual knowledge of a student is gained from collaborative knowledge of his or her group members through interaction. That collaborative knowledge is in turn produced by individual knowledge of the interacting group members.

Kreijns (Kreijns et al., 2003) describe the interaction between students as the key to collaboration among group members. Furthermore, researchers (der Linden and Renshaw, 2001) suggest that collaborative learning occurs from the exchange of dialogues among the students.

Observation 14: The collaboration among the members of a group of students occurs due to their interaction/discourse with each other.

Zumbach (Zumbach et al., 2005) describes a collection of dyadic (between two students) interactions for a group of students which were reported by researchers in the CSCL community. An example of interactions mentioned in (Zumbach et al., 2005) is: (a) student a proposes a solution for the assigned task, (b) student b accepts or proposes another solution to the task. Thus:

Observation 15: The compilation of discourse/interaction patterns presented by Zumbach (Zumbach et al., 2005) describes a typical *dyadic* (e.g., between two students) learning scenario in terms of a chain of action-reaction patterns.

The quality the discourse/interactions within a group depends on the *affective* state of a student (Issrof and Soldato, 1996) and his or her social relationship with other students in the group. Jones and Issroff (Issroff and Jones, 2005) and Vass (Vass, 2002) report that, students who are friends have established ways of working which are implicitly understood rather than explicitly discussed. In addition, (Kreijns et al., 2003) mentions that social relationships contribute to common understanding, an orientation towards cooperation, and the desire to remain as a group. Finally, as reported in (Khandaker et al., 2006), the students form their view of other students due to the type and extent of collaboration they receive from their peers. Clear and Kassabova (Clear and Kassabova, 2005) further report that in collaborative learning settings it is common to have students whose motivation is affected by the motivation of other group members. When the other group members are motivated to learn and to collaborate, it increases the motivation of a student who had low motivation when he or she joined the group, and vice versa. We derive from the above the following observations:

Observation 16: Good social relationship improves the quantity and quality of interactions among group members.

Observation 17: The quantity and quality (i.e., learning outcome) of interactions among a group of students vary over time due to factors internal and external to the classroom environment. Improvement in social relationship among the members of a group improves the

quality of collaborations among them. On the other hand, when a student group member experiences distracting factors, that experience reduces the quality of his or her collaboration with other members.

Observation 18: A group member's motivation is impacted (*positively* or *negatively*) by his or her group members' motivations.

Observation 19: Social relationship between a student and his or her peer (as perceived by the student) change according to the frequency, extent, and quality of collaboration (e.g., how many times did my peer helped me).

4.1.4 Scaffolding

Bruner (Bruner, 1985) and Cazden (Cazden, 1983) define scaffolding as the act of providing assistance to a child so that he or she is able to carry out a task (e.g., solve a problem) that he or she cannot do by herself. Over time, the concept of scaffolding has been introduced into traditional classrooms to aid learners to achieve difficult learning objectives and complete difficult tasks (Khandaker et al., 2006) where tools and software are used to (1) offer structure and support for completing a task and (2) promote peer interactions to enable peers to support each other's learning. In the first type of scaffolding, the students are provided information about how to better approach to solve the task that they are having difficulty with. In the second type of scaffolding, the peer support of a student is enhanced in the hope that those peers would provide guidance and information for that student to help him or her solve that task. Researchers in the CSCL community are now utilizing scaffolding in the form of incorporating structure of learning activities (e.g.,(Harrer et al., 2008)) and improving peer support (e.g.(Liu et al., 2008)). As CSCL researchers (e.g., (Khandaker et al., 2006; Liu et al., 2008) note that due to being in different zones of proximal development, the learners benefit most when the scaffolding is targeted toward their zone of development. So, one of the recommendations provided to the CSCL practitioners is to customize the scaffolding to specific learners' needs. Hence:

Observation 20: Scaffolding in the CSCL environment can be provided by: (1) providing structure and support for completing tasks and (2) improving of peer support.

Observation 21: Scaffolding in the CSCL environment may be used to improve the knowledge of the learners regarding the assigned task.

Observation 22: Learners in a CSCL environment benefit more when the provided scaffolding is targeted to their zone of proximal development.

4.2 Simulation Environment & Algorithms

The SimCoL environment is defined as a 3-tuple: $E = \langle T, I, H \rangle$. Where T is a set of tasks, I is an agent who simulates the teacher, and $H = \{h_1, \ldots, h_{ns}\}$ is a set of agents who simulates the students in a collaborative classroom environment. In this section, we first define the tasks T. Then, based on the observations presented in Section 4.1, we describe the attributes and the behavior of agents H who represent the students in SimCoL. Furthermore, we describe how the simulated teacher I forms groups of simulated students and carries out CSCL classroom sessions in the SimCoL environment using a set of simulation steps. Finally, we describe the collaboration process of the simulated students H in a group in SimCoL using a set of simulation steps and discuss how their attributes change.

4.2.1 Task

The tasks in SimCoL represent the problems and exercises that are solved by the students in a CSCL environment. The set of tasks is denoted by, $T = \{T_1, \ldots, T_{nt}\}$ where,

$$T_j = \langle ct_j, di_j, tl_j, \overrightarrow{sq_j} \rangle \tag{4.1}$$

Here, ct_j denotes the concept of the task. This concept represents the subjective knowledge required to solve the task. $di_j \in \mathbb{R}$, is the difficulty of the task as determined by the simulated teacher. tl_j is the time limit within which the task is to be completed. $\overrightarrow{sq_j}$ is a

vector representing the simulated student groups' (who are working on the task) view of the solution quality of the task T_j at time t.

4.2.2 Simulated Student

We represent the model $hm_{i,t}$ of each simulated student $h_i \in H$ in SimCoL by a 6-tuple:

$$hm_{i,t} = \langle K_{i,t}, AB_{i,t}, MO_{i,t}, EM_{i,t}, SR_{i,t}, TSQ_{i,t} \rangle \tag{4.2}$$

where, $K_{i,t} = \{\langle ct_j, ex_{i,j,t} \rangle | \forall T_j \in T\}$ is the knowledge of simulated student h_i at time t with ct_j representing the concept of T_j and $ex_{i,j,t} \in \mathbb{R}$ is the expertise, i.e., the amount of knowledge the simulated student has about the concept. The goal of simulated student collaboration is to increase the value of this expertise. $AB_{i,t} = \{AB_{i,j,t} | \forall T_j \in T\}$ with $AB_{i,j,t} \in \mathbb{R}$, is the ability of h_i at time t for task T_j. $MO_{i,t} \in \mathbb{R}$, is the motivation of h_i at time t. $EM_{i,t} \in \mathbb{R}$, is the emotional state of simulated student h_i at time t. $SR_{i,t} = \{sr_{i,k,t} | h_k \in H - h_i\}$ where $sr_{i,k,t} \in \mathbb{R}$ is the social relationship between h_i and h_k at time t as perceived by h_i. $TSQ_{i,j} \in \mathbb{R}$ denotes the target solution quality of T_j of h_i at time t.

We have included $K_{i,t}$, $AB_{i,t}$, $MO_{i,t}$, and $EM_{i,t}$ in the model according to Observation 1 and included $SR_{i,t}$ according to Observations 16 and 17. Also, combining Observations 4 and 5, we assume that the ability of a simulated student is related to his or her knowledge, motivation, and emotion in the following way:

$$AB_{i,j,t} \propto w_{abx} \cdot ex_{i,j,t} + w_{abm} \cdot MO_{i,t} - w_{abe} \cdot |EM_{i,t}| \tag{4.3}$$

where w_{abx}, w_{abm}, and w_{abe} are weights. According to Equation 4.3, the ability of a simulated student for a particular task at any time is *proportional* to the sum of his or her expertise on the concept of that task and motivation minus the absolute value of his or her

emotional state. We also define the target solution quality of a simulated student with:

$$TSQ_{i,j,t} \propto AB_{i,t} \tag{4.4}$$

So, a simulated student's target of the quality of the solution of the assigned task is proportional to his or her ability for that task. According to Observations 3 and 4, the ability of a simulated student determines how much of his or her existing knowledge can be *activated* to produce behavior (i.e., effort to solve the task) and learning. Therefore, given the same time limit tl_j for a task T_j, a simulated student with higher ability would be able to solve the assigned task better than a simulated student with lower ability. So, we assume that the simulated students have targets of the final solution quality according to their own abilities.

4.2.3 Teacher

The teacher I in SimCoL acts as the coordinator of the CSCL sessions. The teacher delivers instructions, forms groups, and assigns collaborative tasks. In SimCoL, we have implemented three different group formation methods: random, Hete-A (Gogoulou et al., 2007), and VALCAM (Soh, Khandaker and Jiang, 2006) group formation method.

Algorithm 4.1 shows how the teacher carries out the CSCL session through a set of simulation steps. First, the teacher initializes the classroom (tasks, group formation scheme, how often scaffolding should be provided, and how many groups would receive scaffolding). Then, for each initialized task, the teacher: (1) initializes a collaborative session (Step 2-4), forms simulated student groups (Step 8-12), and announces the start of the collaborative session to all simulated students (Step 13). Then until the collaborative session is over, the teacher periodically sorts the groups according to their current achieved solution quality of the task (Step 16) and then selects the groups who have the lowest solution quality. Those selected groups are then provided scaffolding (Step 18). Finally, the teacher

announces the end of the collaborative session when the time limit for the current task is over (Step 23).

Algorithm 4.1 Simulation Steps of Teacher

Require: i) $T \leftarrow \{T_1, \ldots, T_{nt}\}$, ii) $G_{fs} \leftarrow$ group formation scheme, iii) $t_{sc} \leftarrow$ scaffolding period, $n_{sc} \leftarrow |G_{sc}| \cdot r_{sc}$, iv) students $H \leftarrow \{h_1, \ldots, h_n\}$, and v) agents $Sa \leftarrow \{Sa_1, \ldots, Sa_n\}$

  1: **for all** $T_j \in T$ **do**
  2:     Initialize collaborative session s_j $t \leftarrow 0$
  3:     $G \leftarrow \{G_1, \ldots, G_m\}$
  4:     $n_g \leftarrow \lfloor H_m/m \rfloor$
  5:     Announce task T_j to H
  6:     **if** $G_{fs} = R_n$ **then**
  7:         Form random group for H
  8:     **else if** $G_{fs} = H_a$ **then**
  9:         Form Hete-A group for H
 10:     **else if** $G_{fs} = H_v$ **then**
 11:         Form VALCAM group for H
 12:     **end if**
 13:     Announce start of session s_j to H
 14:     **while** $t < tl_j$ **do**
 15:         **if** $mod(t, t_sc) = 0$ **then**
 16:             Sort(ASC) G according to $tsa_{j,t}$
 17:             **for** $t \leftarrow 0$ to n_{sc} **do**
 18:                 Provide scaffolding to G_i
 19:             **end for**
 20:         **end if**
 21:         $t \leftarrow t + 1$
 22:     **end while**
 23:     Announce end of collaborative session s_j to H
 24: **end for**

4.2.4 Assistant Agents

The student-assistant and teacher-assistant agents have been incorporated in SimCoL to implement various agent-based coalition formation algorithms. Each student-assistant agent in SimCoL is assigned to a simulated student and it monitors the change in that assigned simulated student's: (1) expertise gain and (2) social relationship with other students. The teacher-assistant agent is assigned to the teacher to (1) assign and monitor student collab-

orative performances and assign them virtual currency according to that performance and (2) communicate with the student-assistant agents to form groups using VALCAM (Soh, Khandaker and Jiang, 2006).

4.2.5   Collaboration and Scaffolding

Following Observations 14 and 15, in SimCoL, we simulate the collaborative behavior (i.e., collaboration to solve the assigned task and to improve expertise) of a group of simulated students using a series of *dyadic* interactions among the group members. Here, we describe how those interactions occur in SimCoL. First, we define the following functions that dictate the behavior of the simulated students simulating the collaborative learning in SimCoL. Here, we assume that two simulated students h_i and h_k with models $hm_{i,t}$ and $hm_{k,t}$ are working in a group G_m to solve task T_j and all variables $w_{x,y,z}$ are weights:

Motivation Update (based on Observation 18):

$$MSU(MO_{i,t}, G_m) = \left[w_{omo} \cdot MO_{i,t} + w_{gmo} \cdot \sum_{k \in G_m - h_i} \frac{MO_{k,t}}{|G_m| - 1} \right] \qquad (4.5)$$

where w_{omo} and w_{gmo} are weights, $MO_{i,t} \in hm_{i,t}$

Collaboration Probability (based on Observation 16):

$$CP(hm_{i,t}, hm_{k,t}, T_{j,t}) = w_{csr} \cdot sr_{i,k,t} + w_{csq} \cdot (sq_{j,t} - TSQ_{i,j,t}) \qquad (4.6)$$

where w_{csr} and w_{csq} are weights, $sr_{i,k,t} \in SR_{i,t}$, $TSQ_{i,j,t} \in hm_{i,t}$, and $sq_{j,t} \in T_j$.

Collaboration Cycle (based on Observation 15): $cc_{i,k,t,j} = \{act_{i,k,t,j}, rct_{i,k,t,j}, lct_{i,k,t,j}\} \subseteq CC_{i,k,t,j}$ denotes a collaboration cycle completed by h_i with h_k at time t for task T_j. Here, $act_{i,k,t,j}$ denotes an utterance of action, $rct_{i,k,t,j}$ denotes an utterance of reaction in reply to the action $act_{i,k,t,j}$, and $lct_{i,k,t,j}$ denotes the reaction in reply to the reaction

$rct_{i,k,t,j}$. $ci_{i,k,t,j} \in CI_{i,k,t,j}$ denotes a collaboration cycle initiated by h_i but declined by h_k. $CY_{i,k,t,j} = \{CC_{i,k,t,j}, \ldots, CC_{k,i,t,j}, \ldots, CI_{k,i,t,j}\}$ denotes the set of all collaboration cycles between h_i and h_k regarding T_j.

Solution Quality Update:

$$SQU(hm_{i,t}, hm_{k,t}, cc_{i,k,t,j}) = \begin{cases} 0 & \text{If } p_{sq} \geq k_{sq} \\ \frac{ab_{i,j,t} + ab_{k,j,t}}{di_j} & \text{Otherwise} \end{cases} \tag{4.7}$$

where $cc_{i,k,t,j}$ is a collaboration cycle, $\kappa_{sq}, p_{sq} \in \mathbb{R}$ denotes the solution quality update probability *threshold* and a random number that is drawn from a uniform random distribution respectively. $ab_{i,j,t} \in AB_{i,t} \in hm_{i,t}, ab_{k,j,t} \in AB(k,t) \in hm_{k,t}$, and $d_j \in T_j$.

Human Expertise Update (based on Observation 3,4,7-10 and 13):

$$HEU(hm_{i,t}, hm_{k,t}, cc_{i,k,t,j}) = \begin{cases} 0 & \text{If } DE(hm_i, hm_k, T_j) > \kappa_{zone} \\ w_{hab} \cdot ab_{i,j,t} + \\ w_{hde} \cdot DE(hm_{i,t}, hm_{k,t}), T_j) & \text{otherwise} \end{cases}$$
$$\tag{4.8}$$

where,

$$DE(hm_{i,t}, hm_{k,t}, T_j) = |ex_{i,j,t} - ex_{k,j,t}| \tag{4.9}$$

In Equation 4.8, $cc_{i,k,t,j}$ is a collaboration cycle, κ_{zone} is the zone of proximal development constant (Section 4.1.3), w_{hab} and w_{hde} are weights, $ab_{i,j,t} \in AB_{i,t}, ex_{i,j,t} \in K_{i,t} \in hm_{i,t}$, and $ex_{i,k,t} \in K_{k,t} \in hm_{k,t}$.

Social Relationship Update (based on Observation 19):

$$SRU(CY_{i,k,t,j}) \propto [|CC_{i,k,t,j}| - |CI_{i,k,t,j}|] / [|CC_{i,k,t,j}| + |CI_{i,k,t,j}|] \tag{4.10}$$

where $CC_{i,k,t,j}, CI_{i,k,t,j} \in CY_{i,k,t,j}$ are collaboration cycle sets.

Scaffolding Effect (based on Observation 20-22):

$$SEU(hm_{i,t}, ct_j, SO_j) \propto \begin{cases} \frac{1}{1+|ex_{i,j,t}-sol_{i,j}|} & \text{If } p_{sca} \geq \kappa_{sc} \\ \\ 0 \text{ otherwise} \end{cases} \quad (4.11)$$

where $SO_{i,j} = \langle ct_j, sol_j, csc_j \rangle$ is the scaffolding object, $ct_j \in T_j$, sol_j denotes the level of expertise for the student the scaffolding is designed for, csc_j denotes the cost (e.g., time and effort required to design the object) of the scaffolding, p_{sca} is a probability value drawn from a uniform distribution, and κ_{sc} is the scaffolding threshold.

Algorithm 4.2 shows the simulation steps of a student in SimCoL with the various formulas that are used by the agents in parenthesis. During initialization, the student receives its group assignment and the task from the teacher (Algorithm 4.1). Then the student updates its own motivation according to other group member's motivations, and its ability. During the session, the student tries to collaborate with its group members if the quality of the solution is less than its expected solution quality (Step 2) or if someone else in the group wants to collaborate (Step 9). In both of these cases, whether the collaboration is successful or not depends on the collaboration probability (Step 10). During the collaborative session, if the student receives scaffolding from the teacher (Step 13) in the form of a scaffolding object, it updates its expertise. Finally, when the collaborative session ends, the student updates its own view of its social relationship with all its group members (Step 19).

Algorithm 4.2 Simulation Steps of Student h_i

Require: i) Initialize group G_m, T_j, update motivation and ability

 1: **while** Session s_j is not over **do**

 2:    **if** $sq_{j,t} < esq_{i,k,j}$ **then**

 3:      Propose collaboration to randomly chosen $h_k \in G_m - h_i$

 4:      **if** h_k agrees **then**

 5:        Complete and store collaboration cycle in $CC_{i,k,t,j}$, update solution quality and expertise

 6:      **else**

 7:        Store failed collaboration cycle in $CI_{i,k,t,j}$

 8:      **end if**

 9:      **if** Received collaboration request from h_k **then**

10:        **if** $CP(hm_{i,t}, hm_{k,t}, T_{j,t}) > \kappa_{ch}$ **then**

11:          Complete and store collaboration cycle in $CC_{k,I,t,j}$, update solution quality and expertise

12:        **end if**

13:        **if** Received Scaffolding SO_j **then**

14:          Update Expertise

15:        **end if**

16:      **end if**

17:    **end if**

18: **end while**

19: Update social relationship

4.3 Implementation

The SimCoL environment was implemented using the Java version of the Repast (Repast, 2009)—a multiagent simulation toolkit. Table 4.2 describes: (a) the categorizations and the ranges of the randomly generated values in SimCoL, i.e., the student attributes and the weights and constants used in the equations in Section 4.2. Figure 4.1 shows the deployment diagram and Table 4.2 shows the input, output, and control parameters of SimCoL.
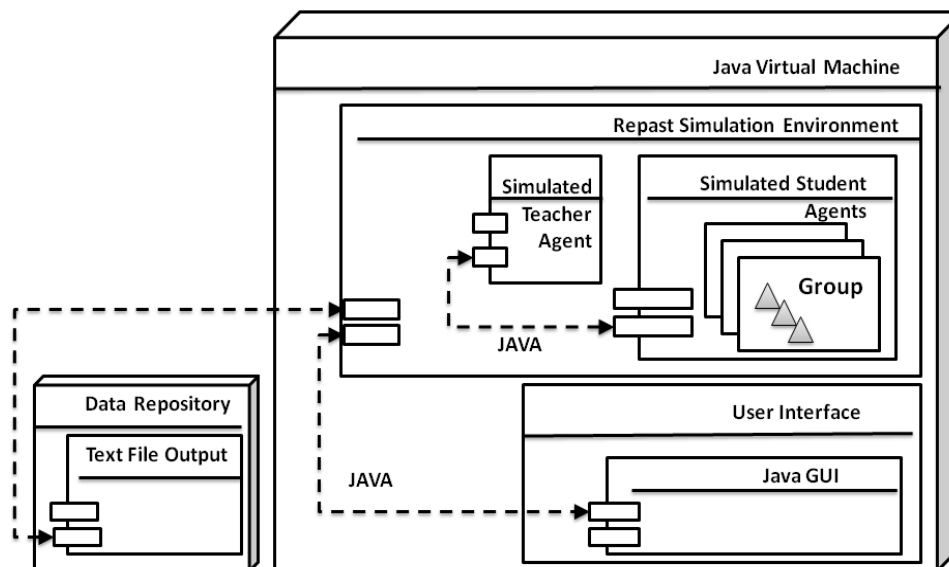
Figure 4.1: Deployment Diagram of SimCoL

Table 4.2: Categorizations, Distributions, Weights, and Constants

Attribute	Category	Generated from Normal Distribution with
Task Difficulty (Equation 4.1)	Low$[0.0, 0.3)$, moderate$[0.3, 0.6)$, and high $[0.6, 1.0)$	$\mu_{di} = 0.5, \sigma_{di} = 0.1$, and range $[0, 1]$
Expertise (Equation 4.2)	Low $[0, 0.3)$, moderate $[0.3, 0.6)$, and high $[0.6, 1.0]$	$\mu_{ex} = 0.3, \sigma_{ex} = 0.25$, and range $[0, 1]$
Ability (Equation 4.2)	Low $[0, 0.3)$, moderate $[0.3, 0.6)$, and high $[0.6, 1.0]$	Calculated using 0 with range $[0, 1]$
Motivation (Equation 4.2)	Low$[0, 0.2)$, moderate$[0.2, 0.8)$, and high $[0.8, 1.0]$	$\mu_{mo} = 0.4, \sigma_{mo} = 0.25$, and range $[0, 1]$
Emotion (Equation 4.2)	Sad $[-1.0, -0.5)$, neutral $[-0.5, 0.5)$, Happy $[0.5, 1.0]$	$\mu_{em} = 0.0, \sigma_{em} = 0.5$, and range $[0, 1.0]$
Social Relationship (Equation 4.2)	Unknown $[0, 0.2)$, familiar $[0.2, 0.8)$, and friend $[0.8, 1.0]$	$\mu_{sr} = 0.4, \sigma_{sr} = 0.25$, and range $[0, 1]$

Attribute	Category	Generated from Normal Distribution with
	Equation 4.3	Weights: $w_{abx} = 0.4, w_{abm} = 0.4,$ and $w_{abe} = 0.2$
	Equation 4.4	Proportionality constant: 0.9
	Equation 4.2.5	Weights: $w_{omo} = 0.8$ and $w_{gmo} = 0.2$
	Equation 4.6	Weights: $w_{csr} = 0.5$ and $w_{csq} = 0.5$
	Equation 4.7	Proportionality constant: 0.001
	Equation 4.8	Weights: $w_{hab} = 0.8$ and $w_{hde} = 0.2$ and proportionality constant: 0.001
	Equation 4.9	Proportionality constant: 0.001
	Collaboration threshold	$\kappa_{ch} = 0.2$
	Zone of proximal development threshold	$\kappa_{zone} = 0.99$

4.4 Results

The goal of our experiment is four-fold: (1) discussing how SimCoL is able to identify and reveal the complex relationship between the variables (i.e., student attributes) of a computer-supported collaborative learning environment, (2) comparing the emergent phenomenon of student performance in SimCoL with that of the published CSCL results, and (3) providing evidence of the validity of SimCoL simulation environment, and (4) relate our findings to validate the assumptions (Section 3.1 in Chapter 3) and the design principles (Section 3.2 in Chapter 3) discussed in the *i*HUCOFS framework. In Section 4.4.1, we discuss how the social relationships among the students in SimCoL impact their collaborations. In Section 4.4.2, we describe the experiment that shows the inter-dependence of the students' attributes on their collaborative learning outcome. In Section 4.4.3, we study the
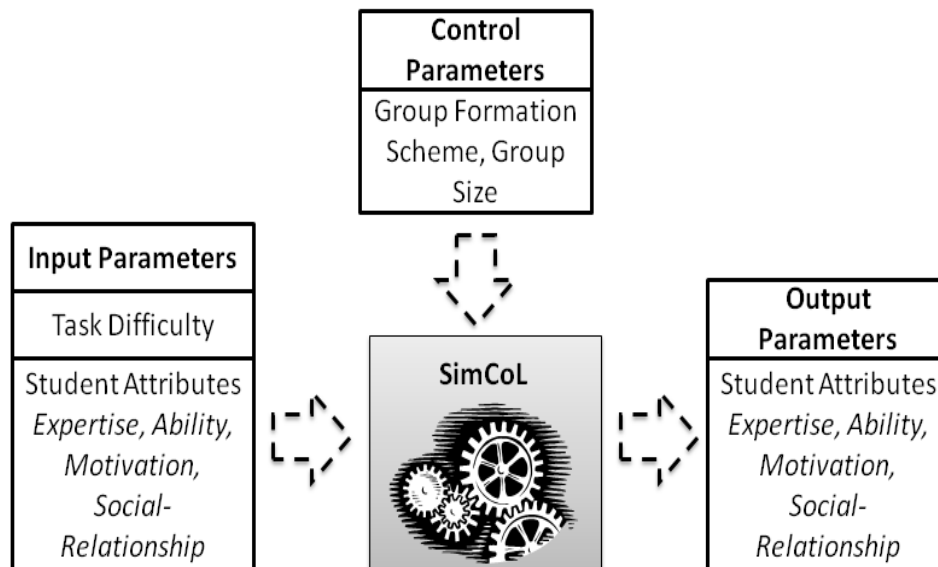
Figure 4.2: Input, Output, and Control Parameters of SimCoL

impact of group formation and group size on student learning.

This allows us to: understand the usefulness of SimCoL in carrying out what-if scenarios in CSCL environments and correlate the observed patterns of student behavior in SimCoL with that of the reported CSCL studies. In Section 4.4.5, we compare and validate the emergent patterns of student behavior in SimCoL with that of the observed student behaviors in the reported CSCL studies. Notice that, all of our experiments are replicated for 10 simulation seeds.

4.4.1 Collaboration among Learners

Here, we ran the simulation for 100 students for 2000 simulation ticks for each run by varying the values of two attributes at a time. We then plotted the successful collaborations of the students against their changing attribute values. Among all attributes, we have found that the social relationship among the group impacts students' collaboration efforts the most. Figure 4.3 shows the results and Table 4.3 shows the skewness and kurtosis values.
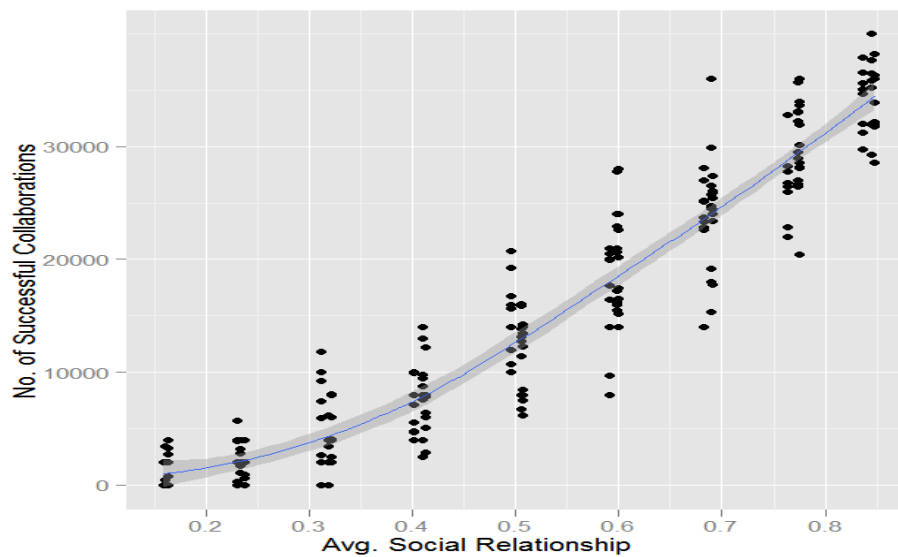
Figure 4.3: Successful student collaborative cycles vs. average student social relationship

Table 4.3: Skewness and Kurtosis of Distributions of Collaboration Cycles

Social Relationship	0.2	0.4	0.6	0.8	1.0
Skewness	1.2	1.8	0.2	-0.1	-0.8
Kurtosis	3.2	6.4	2.1	2.3	3.3

This indicates that as the collaborative learning researchers (Clear and Kassabova, 2005) mention, social relationship among the students is a critical factor in improving the collaborations among them. Furthermore, the lack of the strong relationship between the other attributes like expertise can be explained by our formulation of collaboration probability (Equation 4.6). The two key factors that determine a student's participation in a collaboration cycle is the target solution quality (Equation 4.7) and social relationship. However, if the task solution quality is high (due to other members' contributions), a student's expected solution quality is then mainly determined by his or her social relationship with other group members. This result portrays a common scenario where students often refuse to collaborate/contribute when they see other members solving the task (Roberts and McInnerney, 2007). This inspires us to include social relationship of the agents in the *i*HUCOFS framework's (Section 3.3 in Chapter 3) group formation mechanism.

4.4.2 Compound Impact Analysis

This compound impact analysis allows us to: (a) investigate how the students belonging to the different categories of an attribute respond to the changes in another attribute, e.g., how do the student with low expertise react to a change in their motivation, and (b) investigate whether a student's lower value in an attribute can be compensated by a higher value. For this experiment, we ran the simulation for 100 students for 2000 simulation ticks for each run by varying the values of two attributes at a time. Figure 4.4 and Figure 4.5 show the average and standard deviation of student expertise gain for students with *low*, *medium*, and *high* expertise against changing motivation. Figure 4.6 shows the average and standard deviation of student expertise gain for students with *low*, *medium*, and *high* expertise against changing social relationship. Table 4.4 and Table 4.5 show the skewness and kurtosis of the students with *low*, *medium*, and *high* expertise in Figure 4.4 and Figure 4.5.

According to Figure 4.4, we see when the average motivation of the students is increased, the students of all categories (low, medium, and high) of expertise are able to

improve their expertise gain and there are students who fall behind (unchanged standard deviation). This is to be expected as dictated by expertise update equation Equation 4.8 where the expertise increase is determined by the motivation and difference in expertise. Furthermore, the unchanged standard deviation indicates that there are students in all three cases (*low to high* motivation) who cannot gain expertise due to the increased motivation.

Figure 4.6 shows that as the social relationship of students improve, their expertise gain improves at first, and then that rate of improvement slows down to zero. Furthermore, Figure 4.6 shows that the standard deviation of the students expertise gain remains somewhat unchanged with the increasing social relationship. This occurs due to our use of student social relationship while calculating the collaboration probability among two students (Equation 4.6). The expertise gain of the students in the group depends on how well they collaborate. As the social relationship among the students starts to increase from initial lower value, the probability of them collaborating increases. As a result, they are able to gain more expertise. However, when their social relationship values are near maximum and all students in every group are collaborating, increase in the social relationship further, does not impact their expertise.

Figure 4.4: Avg. student expertise gain vs. average student motivation for low, medium, and high expertise (*left to right*) students.
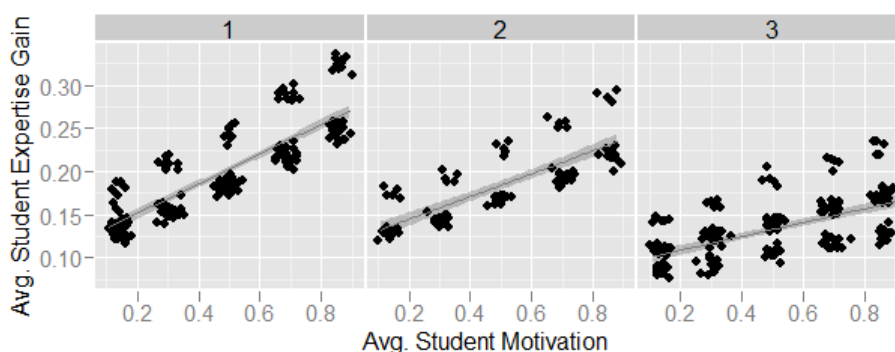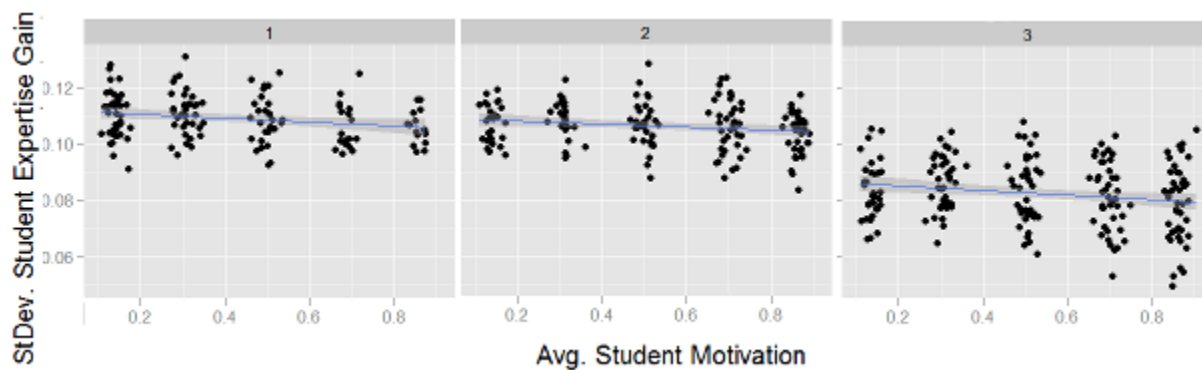
Figure 4.5: StDev. of Student Expertise Gain vs. Average Student Motivation for Low, Medium, and High Expertise (*left to right*) Students.

Table 4.4: Skewness and Kurtosis of the Distributions of Expertise Gain (Figure 4.4)

Social Relationship	Low	Medium	High
Skewness	0.6	0.5	1.3
Kurtosis	2.3	3.0	4.9

Table 4.5: Skewness and Kurtosis of the Distributions of Expertise Gain (Figure 4.5)

Social Relationship	Low	Medium	High
Skewness	0.6	0.5	1.3
Kurtosis	2.3	3.0	4.9
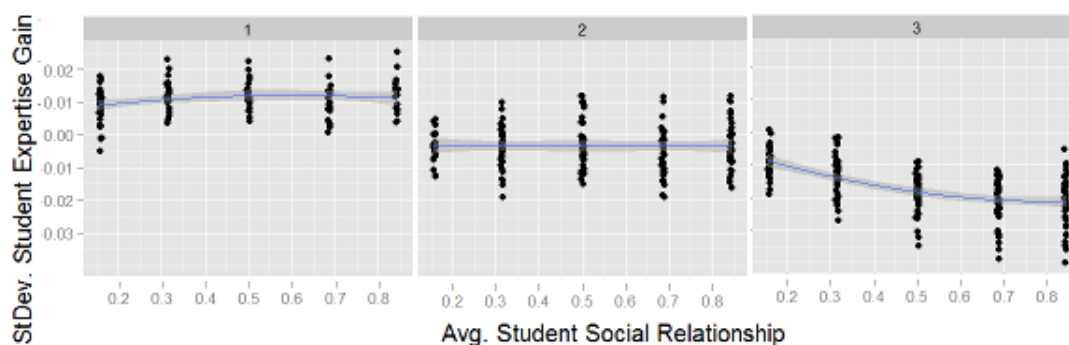
Figure 4.6: Standard Deviation of Student Expertise Gain vs. Average Student Social Relationship for Low, Medium and High Expertise (*left to right*) students.

Our observations here provide us the insight that, the critical student attributes in a CSCL setting often impact (negatively and positively) one another's contributions to a student's collaboration and learning. This observation is *in sync* with the current theories that describe the collaborative learning mechanism being affected by a variety of student attributes like motivation (Shell and Brooks, 2007; Issrof and Soldato, 1996; Clear and Kassabova, 2005). Thus, while setting up the collaborative learning environment, or when evaluating the outcome, it is important to look at all of those critical attributes together instead of in isolation as often discussed in the results of current CSCL research (Kreijns et al., 2003; Issroff and Jones, 2005; Gogoulou et al., 2007; Soh, Khandaker and Jiang, 2006). In other words, while determining the impact of a collaboration script, group formation scheme, or other CSCL tool, the students' learning outcome *alone* may not be a sufficient indicator. Instead, we should also look at factors like motivation and social relationship that could have influenced the students' expertise gain.

4.4.3 Impact of Group Formation Method

In this section, we study the effect of two teacher-controlled aspects of a typical CSCL environment, i.e., (1) the group size and (2) the group formation scheme, on the average expertise gained by the students. During the simulation, the student groups in this experiment were formed using Random, Hete-A, and VALCAM group formation methods with the group size selected from the range $[2, 4]$. VALCAM is an agent-based algorithm of group formation which uses a multiagent system to form student groups that brings together experts with non-expert students where the members have high social relationships. Hete-A algorithm is a non-agent-based algorithm that forms heterogeneous groups. In Hete-A, the students are first categorized by assigning them to a matrix whose dimensions represent the attributes of a student. Once the students are categorized, the Hete-A algorithm builds heterogeneous groups by selecting students with the highest difference of attribute values according to their position in the matrix. Here, the Hete-A algorithm was used with the motivation and expertise as the two matrix dimensions. We first ran the simulation with the parameters described in Table 4.2 for 30 students for 2000 ticks with expertise distribution mean $\mu_{ex} = 0.8$, expertise distribution standard deviation $\sigma_{ex} = 0.8$, and collaboration threshold $\kappa_{ch} = 0.5$, for a set of 30 students, for 10 different tasks and for 2000 simulation ticks, where the students mean expertise and social relationship was set to the mean initial values reported in (Soh, Khandaker and Jiang, 2006). Figure 4.7 shows that the students in the VALCAM-formed groups performed better than the randomly formed and HETE-A formed groups.

The improvement in student performance in VALCAM-formed groups was reported in (Soh, Khandaker and Jiang, 2006), so this result *reproduces* those observations. This improvement of student performance in VALCAM-formed groups can be explained by the way VALCAM forms student groups that contain expert and non-experts who have high social relationships amongst themselves. Since, the collaboration probability (Equation 4.6) and therefore the collaborative learning in SimCoL is determined by the expertise differ-

ence (Equation 4.8) and social relationship (Equation 4.2), VALCAM-formed groups in SimCoL were able to collaborate better (i.e., higher number of successful collaborative cycles) yielding higher collaborative learning outcome. These results suggest that by setting the initial classroom conditions (e.g., student attributes) in SimCoL like a CSCL classroom, we could execute *what-if* scenarios by running simulations and compare the performances of group formation mechanisms.
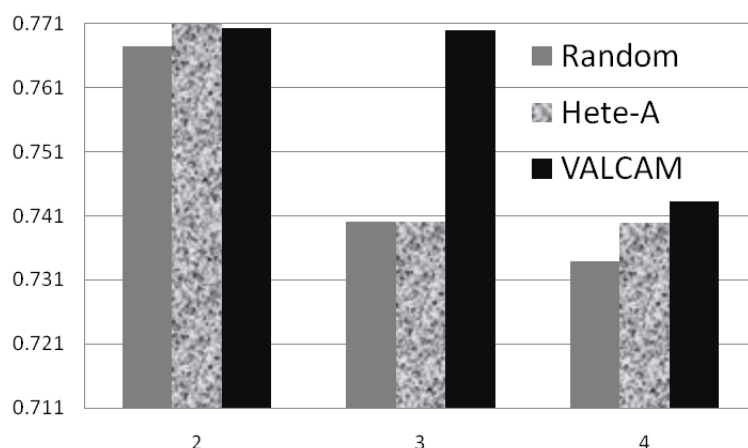
Figure 4.7: Average expertise gain (y-axis) for varying group sizes (x-axis).

4.4.4   Cost and Impact of Scaffolding

In this experiment, we investigate how the individual and group scaffolding improves the expertise of the students when they are collaborating in various types of groups. To collect data, we ran the simulation with the same default set of parameters Table 4.2 for 180 students for 2000 simulation ticks. We calculated: (1) the average improvement in the expertise gain of the students and (2) the cost incurred for providing scaffolding for individuals and groups. For a group in this experiment, one scaffolding object is used per group for group scaffolding (i.e., scaffolding cost is required for one scaffolding object) and one scaffolding object per group member (i.e., scaffolding cost is equal to the sum of all generated scaffolding objects) is used for individual scaffolding. Figure 4.8(a) shows

that the students in all groups are able to improve their expertise more from the individual scaffolding than from the group scaffolding. This is expected, since: (1) individual scaffolding addresses individual students' needs, and (2) according to our design of scaffolding (Equation 4.11), a student's expertise is improved most when the scaffolding is targeted towards his or her expertise level.

Figure 4.8(b) shows that for all three types of groups, the group scaffolding yielded more expertise gain per unit cost than the individual scaffolding. The cost of scaffolding denotes the time and effort required for providing scaffolding to the students. Providing individual scaffolding requires more cost since each individual student has to be modeled and different types of scaffolding have to be provided to the students according to their expertise level. On the other hand, group scaffolding requires less cost since the scaffolding action is more generic and only one type of scaffolding is provided to the entire group. But unexpectedly, the group scaffolding is shown to be more economical in terms of expertise improvement per unit cost. Upon closer analysis, this can be explained by the cyclic and convergent nature of the collaborative knowledge building process (Observation 11). Due to this cyclic nature, collaborative knowledge is transferred among the group members due to their interactions throughout the collaborative session. Furthermore, our non-adaptive scaffolding process periodically provides scaffolding to a fixed number of student groups by first sorting them according to their performances. However, near the end of the collaborative cycle, due to the heterogeneous nature of groups, there are some students who have already reached near-maximum expertise level. So, scaffolding for such group members is no longer effective. Thus, individual and group scaffolding do not yield any expertise improvement for those high-expertise group members. But, for those high-expertise group members, the individual scaffolding incurs a much higher cost than would the group scaffolding. As a result, the improvement of expertise per unit cost for individual scaffolding is smaller than the group scaffolding. These results indicate that although targeted individual scaffolding may improve the expertise gain of a set of students more than group-based

scaffolding, the former is less-economical when applied in a non-adaptive manner.
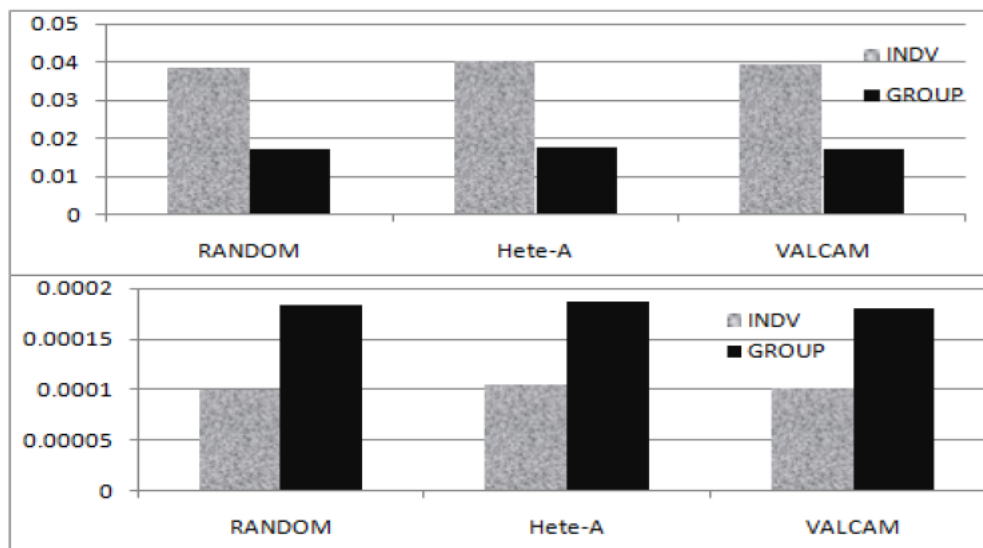
Figure 4.8: (a)(top) Average expertise gain for individual and group scaffolding (b) (bottom) Average expertise gain per unit cost for individual and group scaffolding.

4.4.5 Validity and Correlation with CSCL Results

Here we validate SimCoL's design by comparing its simulation results with published collaborative learning patterns.

Variance in Learning Rate. As reported in (Spoelstra and Sklar, 2008), high-ability students have higher learning rates than low-ability students because they are able to grasp, process, and internalize information received during the collaboration process. When we compared the learning rates of high/low ability learners in SimCoL, we found that the high-ability students learn at a faster rate than the low-ability students (0.1 vs. 0.3). Convergence of Learning Rates. CSCL researchers (Cazden, 1983) described the collaborative knowledge building as a cyclic process that converges to a final value. Researchers (Spoelstra and Sklar, 2008; Stahl, 2004) also described that the rate at which the students gain expertise is faster in the beginning and then slows down over time. The total expertise gain curve shown in Figure 4.9 has two properties: (1) the total expertise gain of the students converges to

a final value and (2) the rate of change of the curves is higher in the beginning and slows down at the end. Furthermore, the same convergence pattern is observed when the simulation run is repeated with Hete-A group formation method. So, the knowledge gain of the students in SimCoL follows patterns described by other CSCL researchers (Roberts and McInnerney, 2007; Clancey, 2004).
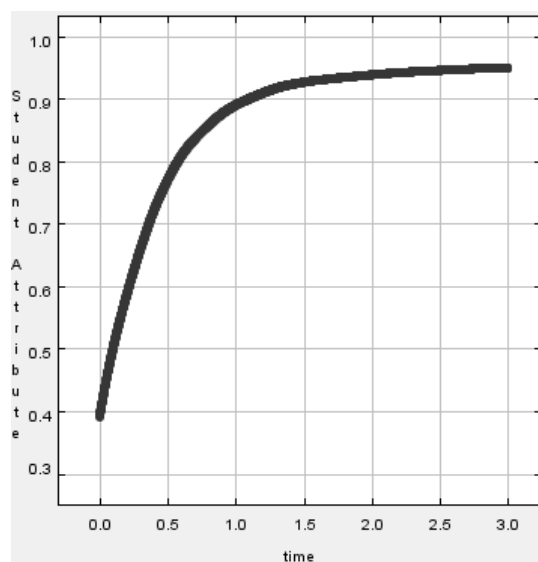
Figure 4.9: Total expertise gain of students collaborating in groups formed by random group formation method.

Correlation with Observed CSCL Results. Here we try to compare our simulation results with CSCL results published in (Soh, Khandaker and Jiang, 2006). For this comparison, we have first mimicked a simulated environment as the CSCL classroom (Soh, Khandaker and Jiang, 2006) by setting the parameters of SimCoL equal to the parameters of the CSCL classroom (Soh, Khandaker and Jiang, 2006), i.e., we set: (1) the mean expertise of the students in SimCoL as 0.7, (2) number of tasks as 5 for each collaborative learning session, (3) number of students as 11, and (4) mean social relationship of the students as 0.9. Then similar to the CSCL classroom, we have simulated 4 collaborative sessions in SimCoL. Then we have calculated the correlation between the actual CSCL results and simulated results in SimCoL. Table 4.6 shows that for both expertise gain and

social relationship change, the correlation was significant and high. However, as we have discussed in Section 4.1.1, student attributes like motivation may also impact the student expertise gain which we have not collected data upon. So, this correlation can be made stronger with the consideration of those factors which is in our future plan (Chapter 7).

Table 4.6: Correlation between Simulation Data and Observed CSCL (Soh, Khandaker and Jiang, 2006) Results

Attribute	Correlation
Student Expertise	0.83
Student Social Relationship	0.97

4.5 Conclusions

The evolving domain of learning theories and CSCL systems (Stahl et al., 2006) indicate that a simulation environment could provide a low-cost tool to the researchers and teachers to better understand the impact of instructional approaches. Here we have proposed Sim-CoL, an agent-based tool for simulating the collaborative learning in a CSCL system. We have described the design and implementation of the SimCoL environment and its agents using observations reported by the researchers working in the individual, peer-based and collaborative learning domains. The overall simulation results of the SimCoL environment is *consistent* with previously reported collaborative learning patterns. Furthermore, our results hint that the SimCoL environment allow the researchers to gain better insights into the impact of: (1) individual student attributes, (2) various agent-based and non-agent based group formation algorithms, (3) different types of scaffolding processes on the collaborative learning outcome of students, and (4) CSCL and collaborative learning on real classrooms in particular, and any human-computer environments where online collaborative activities take place among users with diverse behaviors. When compared with the

assumptions (Section 3.1 in Chapter 3) and the design principles (Section 3.2 in Chapter 3) discussed in the *i*HUCOFS framework, our results provide us the following insights. First, the group formation method has a significant impact on the performances of the students which inspires us to focus our efforts toward designing and implementing a sophisticated group formation method that improve student performances. Second, a student group's performance and learning can be significantly *enhanced* by providing scaffolding to the group or the individual students. This encourages us to use implicit or explicit methods to scaffold the student coalitions so that they can improve their performances over time. Third, the group formation methods' ability to form groups largely depends upon the students' attributes. Since these attributes can be difficult to measure accurately in the real world, it is important to take the uncertainties arising from that inaccurate measurements while forming student groups. Finally, due to the significant impact of the social relationship, we may have to accommodate the social relationship of students in our future instantiations of the *i*HUCOFS framework.

4.6   Future Work

Our future work involves running a *what-for* simulation scenario that would allow us to gain valuable insights into the environment dynamics (e.g., which of the student attributes was the dominant factor in determining the CSCL outcome) of that CSCL setting. In future, we also plan to incorporate the MHCF algorithm (Section 3.3) for coalition formation in SimCoL to compare its performance against VALCAM and study how the student attributes (e.g., social relationship) impact the performance of MHCF algorithm in the simulation environment. Based on those simulation results, we will then be able to improve our design of the MHCF algorithm further.

Chapter 5

MCFP-M: ClassroomWiki

This chapter is organized as follows. In Section 5.1, we describe the ClassroomWiki architecture with detail descriptions of its components. Section 5.2 outlines our implementation of MHCF algorithm (Section 3.3) for student group formation in ClassroomWiki. Then Section 5.3 describes our experiment setup and Section 5.4 presents our results. Finally, Section 5.5 concludes and Section 5.6 presents our future work.

5.1 ClassroomWiki Architecture

ClassroomWiki is composed of four conceptual modules (Figure 5.1): (1) Wiki (WIM), (2) Communication (COM), (3) Tracking and Modeling (TAM), and (4) Group Formation (GFM). First, the WIM allows the teacher to create and assign Wiki assignments to the students. For students, the WIM allows: (1) revision and (2) versioning of their Wiki assignment text. Second, COM facilitates student and teacher communications through: (1) assignment-specific topic-based forums used by the teacher and the student groups and (2) announcements and emails from the teacher to the individual students or student groups. Third, TAM tracks students' interactions with their group members and with the modules of ClassroomWiki to build a detailed student model. That model is then used to: (1) better assess students' individual contributions towards their groups' Wiki-related work leading

to: (a) detection and prevention of free-riding behavior and (b), precise and specific interventions from the teacher to improve collaboration, and (2) better group formation. Finally, the GFM allows the teacher to automatically form student groups *randomly* or by using the tracked student models and the MHCF algorithm.

5.1.1 Wiki Module (WIM)

The WIM allows the teacher to create and assign a topic to the student groups in the course. Once assigned by the teacher, the student groups collaborate to create a Wiki on that topic which is evaluated by the teacher after the due date of the assignment. The WIM consists of the assignment and the versioning component.
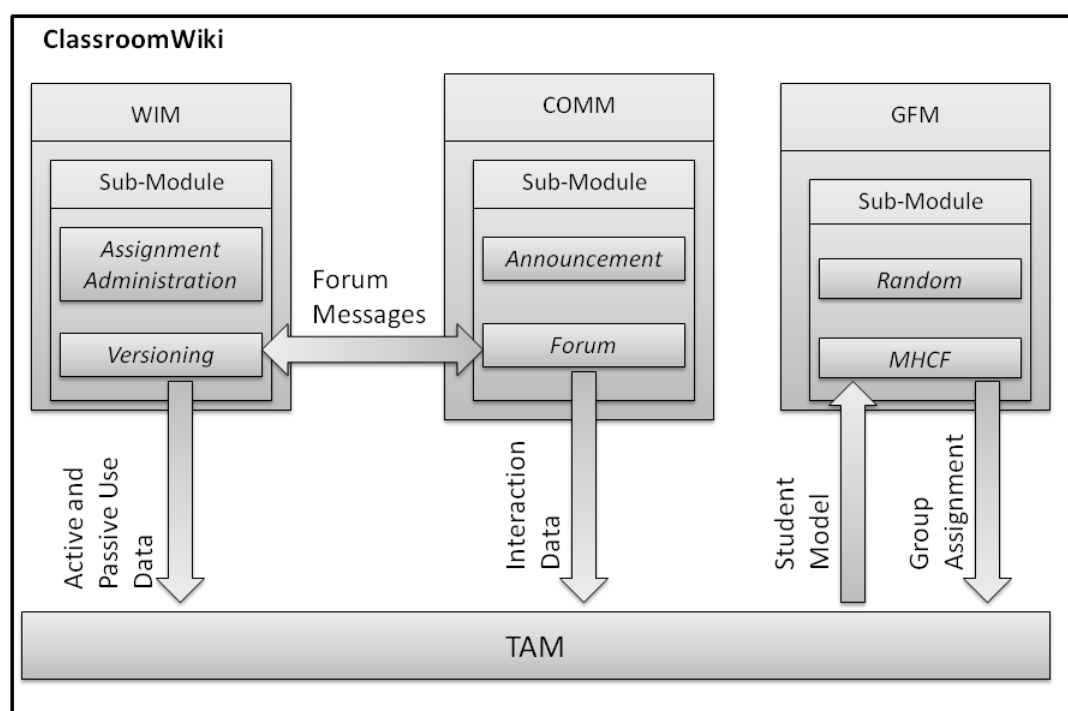

Figure 5.1: ClassroomWiki Architecture

The assignment component of WIM allows the Wiki teacher to create Wiki assignments for the participating students. The Wiki assignment specifies the topic, the requirements for the final submitted version (e.g., required sections, word limit, due date), and minimum

size of the student groups. Once created by the teacher, the assignment component stores this specification which can then be accessed by the students (while they are collaborating) and by other modules (e.g., the group-size is used by the group formation module). The versioning component tracks and stores all changes (e.g., addition, deletion) made to the Wiki by all members of each student group. This tracking allows a student to view a color-coded (e.g., different colors for additions, deletions, and unchanged text) history of changes of the Wiki made by his or her group members.

5.1.2 Communication Module (COM)

ClassroomWiki's Communication Module(COM) consists of two components: (1) a topic-based forum and (2) an announcement system. The topic-based forum in the COM facilitates the collaboration process of the students in two ways. First, while collaborating, the members of a student group can discuss their plan or approach of writing the Wiki, their revisions, and other Wiki-related questions and comments in the forum. Second, the forum allows the teacher to respond to questions posed by the members of a student group for their Wiki. The announcement system allows the teacher to notify the students about changes or other assignment-specific matters. Furthermore, the forum component supports the four forms of learning mentioned in Section 2.1 by allowing the students to discuss the Wiki assignments with their group members using a topic-based forum. This COM allows the students to discuss any ideas or concepts contained in the Wiki and thus assimilate and accommodate new knowledge from the forum while they are collaborating. Further, while the students exchange knowledge in the forum and resolve their cognitive conflicts through discussions in the forum, the forum itself transforms (i.e., external assimilation and accommodation in Section 2.1) due to those posted discussions. Note that the teacher is also able to participate in a student group's forum where that participation impacts the group's knowledge on the Wiki-topic and thereby enhances the assimilation and accommodation process of that group's members.

5.1.3 Tracking and Modeling Module (TAM)

The goal of the Tracking and Modeling Module (TAM) in ClassroomWiki is to create and maintain a model of every participating student. The student model in ClassroomWiki is built using information regarding student activities that can be divided into the following five categories:

- **Active Use** - the actions of a student that push information onto his or her group's Wiki and changes the content of that Wiki, e.g., the number of words (1) added, (2) deleted, and (3) rearranged.

- **Passive Use** - student activities in ClassroomWiki that pull information from his or her group's Wiki and do not result in a change in the contents of that Wiki. For example, the number of times a student views (1) the revision history of their group's Wiki, (2) the topics posted by other group members, and (3) the messages in his or her posted topics.

- **Interaction** - a student's interactions with his or her group members while collaborating, e.g., the total number of topics created, the total number of replies posted, the size of his or her messages in words, and the average number of other group members who replied to a student's posted topic.

- **Survey Response** - a student's responses to the various surveys or questionnaires posted by the teacher. These surveys can be designed to capture a student's opinion about the effectiveness of his or her group, peers, or the ClassroomWiki itself. For example, a student may be asked to evaluate the contribution of another group member toward their group's Wiki.

- **Evaluation** - the evaluation scores received by a student for all Wiki-related activities, e.g., a teacher's evaluation of a student for his or her contribution for the group

Wiki Assuming $S = \{s_1, \ldots, s_{ns}\}$ is the set of all students in ClassroomWiki, Section 5.1.4 summarizes the information tracked by the Tracking and Modeling module for a student $s_i \in S$ in ClassroomWiki. The tracked information is used in ClassroomWiki to build student models that: (1) are used by the Group Formation module to realize the MHCF group formation mechanism and (2) allow the teacher to assess the individual contribution levels of students facilitating specific and precise teacher interventions. The details of the use of this stored information are provided in Section 5.1.

5.1.4 Information tracked by Agent a_i for Student s_i

1. **Active Use**

   (a) Number of words added au_i^1

   (b) Number of words deleted au_i^2

   (c) Number of *content-related phrases* added au_i^3

   (d) Number of ***new*** *content-related phrases* added au_i^4

   (e) Min distance between the whole sentences added by s_i and the *whole sentences* added by s_i's group members au_i^{10}

   (f) Min distance between the whole sentences deleted by s_i and the whole sentences added by s_i's group members au_i^{11}

   (g) Number of forum messages posted by s_i to other members' forum topics au_i^7

   (h) Number of group members' editions the content related phrase added by s_i survives au_i^8

   (i) Number of group members' editions the whole sentences added by s_i survives au_i^9

   (j) Number of forum topics created au_i^5

(k) Number of forum messages posted by other group members to s_i's forum au_i^6

2. **Passive Use**—Number of times logged in to the ClassroomWiki, length of each ClassroomWiki session, number of times a student views:

   (a) Wiki assignment specification pu_i^1

   (b) Details of other group members; e.g. email pu_i^2

   (c) Other group member's revisions pu_i^3

   (d) Revision history i.e. list of all revisions and authors of a Wiki pu_i^4

   (e) Other group's revisions if allowed by the teacher pu_i^5

   (f) Forum topics (a) posted by the student pu_i^6 i.e. to check the messages by other group members, and (b) posted by other group members pu_i^7

   (g) Forum messages posted by other group members pu_i^8

3. **Interaction**

   (a) Number of topics created

   (b) Number of messages posted for own topics ir_i^1 and other group member's topics ir_i^2

   (c) Length of the posted topics ir_i^3 and messages ir_i^4 in words

   (d) Number of days the user changed (a) posted forum topics ir_i^5 or messages ir_i^6, (b) posted Wiki revisions ir_i^7

4. **Survey Response**

   (a) Student's evaluation of his or her Peers, i.e., peer-rating sr_i^1

   (b) Student's evaluation of his or her group, i.e., team-rating sr_i^2

   (c) Student's evaluation of ClassroomWiki, i.e., Wiki-rating sr_i^3

5. **Evaluation**

   (a) Teacher's evaluation of a student's a. contributions toward his or her group's Wiki, i.e., Wiki evaluation ev_i^1

   (b) Average performance in other classroom activities or assignments ev_i^2

5.1.5   Student Model in ClassroomWiki

Using the tracked information described in Section 5.1.4, ClassroomWiki builds a student model $hm_{i,t}$ of a student $s_i \in S$ at time t as:

$$hm_{i,t} = \langle cn_{i,t}, co_{i,t}, ev_{i,t} \rangle \tag{5.1}$$

where (1) $cn_{i,t} \in \mathbb{R}$ denotes the average amount of contribution of a student tracked from his or her Wiki assignments, (2) $co_{i,t} \in \mathbb{R}$ represents the collaborative effort of a student calculated from the summary of that student's collaborative or interactive activities, and (3) $ev_{i,t} \in \mathbb{R}$ represents the average evaluation of a student based on the teacher-assigned grades and peer evaluations.

Furthermore, we collapse the different groups of tracked information (Section 5.1.4) by averaging:

$$\widehat{au_{i,t}} = \frac{\displaystyle\sum_{k=1}^{11} w_{aun} \cdot au_{i,t}^k}{\displaystyle\sum_{s_i \in g}\sum_{k=1}^{11} w_{aun} \cdot au_{i,t}^k} \tag{5.2}$$

$$\widehat{pu_{i,t}} = \frac{\displaystyle\sum_{k=1}^{3} wpun_k \cdot pu_{i,t}^k}{\displaystyle\sum_{s_i \in g}\sum_{k=1}^{3} wpun_k \cdot pu_{i,t}^k} \tag{5.3}$$

$$\widehat{ir_{i,t}} = \frac{\displaystyle\sum_{k=1}^{7} wir_k \cdot au_{i,t}^k}{\displaystyle\sum_{s_i \in g}\sum_{k=1}^{7} wir_k \cdot au_{i,t}^k} \qquad (5.4)$$

$$\widehat{sr_{i,t}} = \frac{sr_{k,t}^1}{\displaystyle\sum_{s_i \in g} sr_{i,t}^k} \qquad (5.5)$$

$$\widehat{ev_{i,t}} = \frac{\displaystyle\sum_{k=1}^{2} wev_k \cdot au_{i,t}^k}{\displaystyle\sum_{s_i \in g}\sum_{k=1}^{2} wev_k \cdot au_{i,t}^k} \qquad (5.6)$$

$$cn_{i,t} \propto \widehat{au_{i,t}} \qquad (5.7)$$

$$co_{i,t} \propto w_{irg} \cdot \widehat{ir_{i,t}} + w_{sr} \cdot \widehat{sr_{i,t}} \qquad (5.8)$$

$$ev_{i,t} \propto \widehat{ev_{i,t}} \qquad (5.9)$$

Here, $waun_k, wpun_k, w_{ir}, w_{sr}, wir_k, wsr_k,$ and wev_k in Equation 5.2-5.9 are weights. Notice that Equations 5.2-5.9: (1) capture the time-averaged performance (e.g., the relative values of the active or passive use) of a student with respect to his or her group and (2) allow the teacher to customize the model of a student to better capture his or her performance.

5.2   Implementation

We have used the open-source, Java-based Spring framework (www.springsource.org) to implement ClassroomWiki as a Web-based system. This implementation allows the teachers and students to participate in collaborative Wiki writing assignments from any com-

puter that has an Internet connection and a Web browser. In our current implementation, the ClassroomWiki modules (WIM, COM, TAM, and GFM) are programmed as plain Java objects that reside in the Spring framework. Here, the Spring framework acts as a container that (1) provides a repository (MySQL, `dev.mysql.com`) for the ClassroomWiki modules to store and retrieve information described in Section 5.1.4 and (2) stores the html pages those act as the GUI for the modules. The Spring framework (`www.springsource.org`) and the ClassroomWiki modules are hosted on a Java Glassfish Application Server (`glassfish.dev.java.net`) which serves the online user-requests (e.g., access requests from teachers, students) by providing them the html pages generated by the ClassroomWiki modules in the Spring framework. The agents in ClassroomWiki also use: (1) LingPipe (Natural Language Processing Tool, (`alias-i.com/lingpipe`)) to calculate the values au_i^k for $k = 1, \ldots, 9$ in Section 5.1.4 and Netica-J (`www.norsys.com/netica-j`) to create and maintain the Bayesian network (Figure 5.6). Furthermore, we have used Repast (`repast.sourceforge.net`), an agent-based simulation framework to realize the agents for the MHCF group formation mechanism in the group formation module. Deployment diagram in Figure 5.2 shows the implementation of ClassroomWiki modules. Furthermore, Figure 5.3 shows the typical sequence of steps a student group and the teacher while they are interacting with ClassroomWiki to revise an assignment.

5.2.1   Wiki Module (WIM)

The current implementation of Classroomwiki in Spring framework provides Web interfaces (html pages) for (a) the teacher to create and assign collaborative Wikis to the students and (b) the students to view their assignment and collaboratively build a Wiki for their group according to the teacher's assignment specification. To facilitate the student's revision of a Wiki, we have embedded TinyMCE (`tinymce.moxiecode.com`) - a JavaScript word processor into the ClassroomWiki website. The participating students in
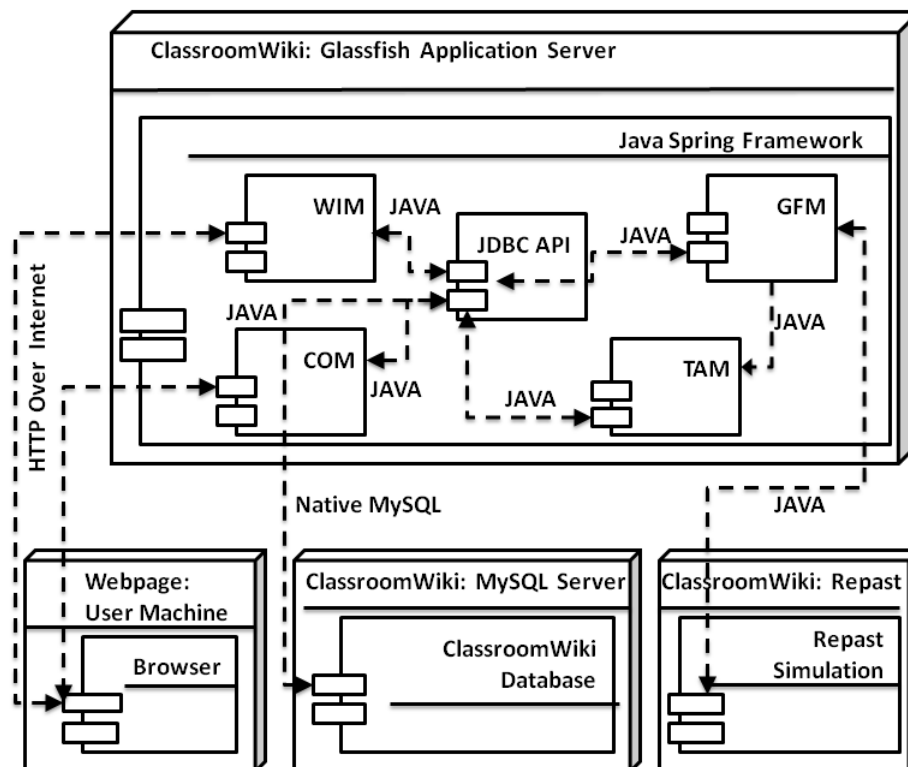
Figure 5.2: Deployment Diagram of ClassroomWiki

ClassroomWiki can use this TinyMCE's Microsoft-word-like interface to write and revise their group's Wiki (see Figure 5.4)

To implement the versioning functionality, we have used an open-source Java library called DaisyDiff (code.google.com/p/daisydiff) to identify the added, deleted, and reorganized text by comparing the two versions of a given html file. For example, for a given original html file, its modified version, and a *configurable* CSS (Cascading Style Sheet) color code specification file (i.e., the colors of added, deleted, and unchanged words), DaisyDiff can determine the differences between them and generate a diff html file. This diff html file marks all changes (added, deleted, and unchanged words) according to the colors specified in the CSS file making it easy for the students to visualize and comprehend the changes made by their group members (see Figure 5.5). Furthermore, while generating that diff file, DaisyDiff allows the TAM to track the words that were
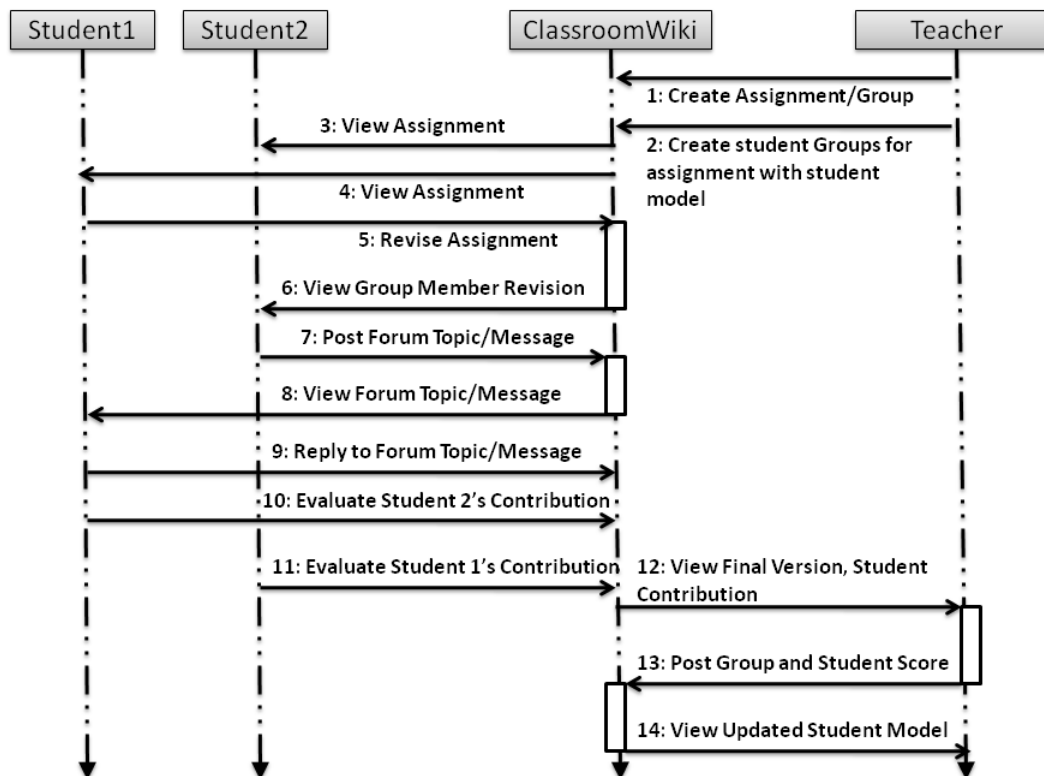
Figure 5.3: Interaction of Students and Teacher in ClassroomWiki

added, deleted, or left unchanged by a revising student.

5.2.2   Communication Module (COM)

The topic-based forum and the announcement system in the Communication module are also implemented as html webpages using Spring in our current implementation of ClassroomWiki. The webpage for the announcement system (Section 3.2) provides a form to the teacher which allows him or her to write and submit announcements in html which is displayed to all students when they log in to the ClassroomWiki website. In the topic-based forum webpage, the forum messages are categorized according to their topics allowing the students to easily search, read, and respond to the topics/messages for each collaborative Wiki assignment.

Figure 5.4: Editor Interface in ClassroomWiki

5.2.3 Tracking and Modeling Module (TAM)

The TAM is also implemented in Java to collect the student activity-related information (i.e., active use, passive use, survey response, evaluation, and interaction in Section 5.1.4) and store this collected information in a MySQL database for other ClassroomWiki modules to use. TAM collects the student-activity related information by using Spring framework's request-driven Model-View-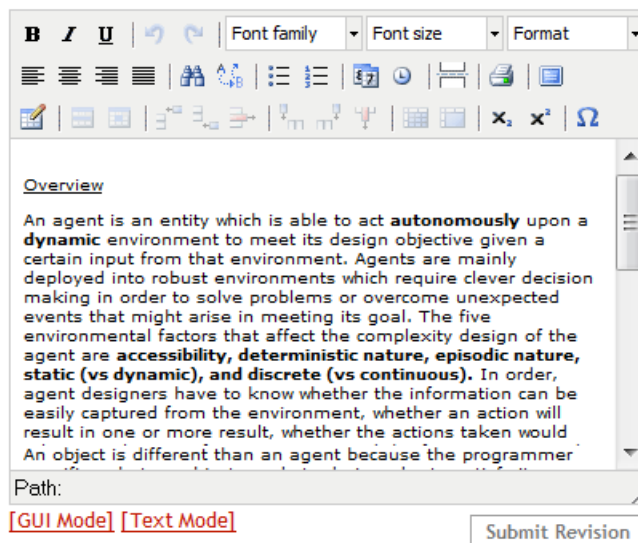Control (MVC) architecture. Spring's MVC architecture is designed around a central servlet called the dispatcher servlet that acts as a gateway that passes the user requests to the ClassroomWiki modules, collects the result of the processing of the user request from the modules, and then generates the html pages that displays the results to the requesting user. To track all user interactions in the ClassroomWiki website, we have implemented the HandlerInterceptorAdapter, a Java Class written in Spring, that intercepts all incoming user requests processed by the dispatcher servlet. Each request contains (1) the time and name of the requested webpage and (2) all variables associated with the requested webpage and the modules that process that user request. So, using the *HandlerInterceptorAdapter*, the TAM is able to collect all user activity-related information (tracked information as described in Section 5.1.4) in ClassroomWiki. This
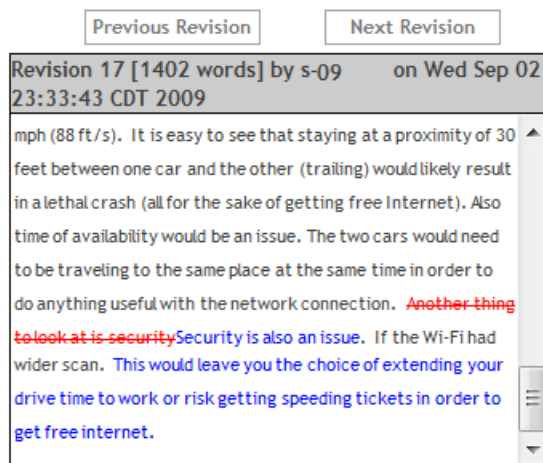
Figure 5.5: Versioning Interface in ClassroomWiki

collected information is then used by the module to build, store, and update the student models in the MySQL database.

5.2.4 Group Formation Module (GFM)

ClassroomWiki's Group Formation module allows the teacher to form student groups either randomly or using the Multiagent Human Coalition Formation (MHCF) framework. Since a detailed description of the MHCF algorithm is discussed in Section 3.3, here we only provide a summary.

Given the model of a student, derived from the tracked information discussed in the previous section, a student agent uses the model to first estimate probabilistically the contribution of a student towards his or her group's Wiki i.e., his or her performance as a group member. Note that the MHCF algorithm assumes a probabilistic environment where a student's average performance (as an individual Equation 5.7 and as a group member (Equation 5.8)) can be estimated but not accurately predicted. Then, based on this probabilistic view of the environment, the agent, on behalf of its user, negotiates with others (1) to collaborate to solve the current task well (improving the current-task reward or score) as well as (2) to increase his or her knowledge learned from the collaboration to solve future

tasks well (improving the future-task reward or score). This is where the tradeoff between the current- and future-task rewards comes into play and the assumption that users learn to improve their problem solving skills from collaborative activities. We further elaborate on MHCF in the following paragraph:

Adopted MHCF Environment

In our adoption of the MHCF algorithm, its environment E is denoted as a 5-tuple $\langle S, A, G, T, R \rangle$. Here, $S = \{s_1, \ldots, s_{ns}\}$ is the set of students, $A = \{a_1, \ldots, a_{ns}\}$ is the set of agents where each agent a_i is assigned to a student s_i, $G = \{g_1, \ldots, g_{ng}\}$ is the set of student groups, $T = \{t_1, \ldots, t_{nt}\}$ is a set of tasks which the student groups collaborate to solve, and reward R is a 2-tuple $\langle R_{ct}, R_{ft} \rangle$ where R_{ft} and R_{ft} are two real-valued functions that estimate the probability of a student's current-task and future-task rewards when he or she joins a coalition with,

$$R_{ct} : f(sm_{g,t}, t_j) \to \mathbb{R} \tag{5.10}$$

$$R_{ft} : f(sm_{g,t}, t_j) \to \mathbb{R} \tag{5.11}$$

$$sm_{g,t} = \{sm_{k,t} | s_k \in g\} \tag{5.12}$$

where $sm_{g,t}$ is a set of the models of the members of the potential group g at time t where that group g is being formed to solve the task $t_j \in T$. Note that the functions R_{ct} and R_{ft} use the model of the members of a potential group to calculate the expected current-task and future-task rewards (in terms of evaluations) for a student to if he or she joins that potential group to solve a task. While the current-task reward allows an agent to estimate what its student will receive after participating in a group, the future-task reward is what the agent expects the student will gain in the future, due to their improvement in expertise

or behavior through learning. This allows MHCF to look ahead while encouraging students to perform sufficiently well in their current tasks.

Notice that the structure of the formed groups depend upon functions R_{ct} and R_{ft} that help the student agents understand how the negotiated composition of a group of students would (1) solve the chosen task and (2) collaborate to learn to solve future tasks better. In MHCF, the student agents *cooperatively* build a Bayesian network Figure 5.6 that maps a potential student groups' model to their collaborative learning outcome, i.e., *represents R_{ct} and R_{ft}*. In this Bayesian network,

$$co_{g,t} = f(co|s_k \in g) \to \mathbb{R} \tag{5.13}$$

$$cn_{g,t} = f(cn_{k,t}|s_k \in g) \to \mathbb{R} \tag{5.14}$$

$$ev_{g,t} = f(ev_{k,t}|s_k \in g) \to \mathbb{R} \tag{5.15}$$

for a student group $g = \{s_k|k = 1, 2, \ldots\}$. The Bayesian network is built and maintained using the following steps:
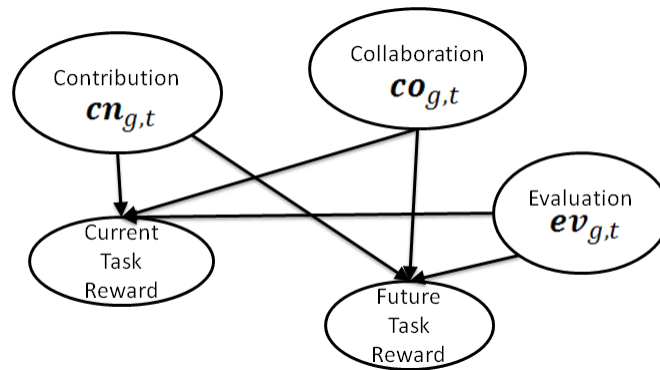

Figure 5.6: Bayesian network Structure

Learning for Group Formation

- **Initialization**—Initialization is done by (a) setting uniform priors - i.e., setting equal current-task and future-task outcomes for all inputs, or (b) using previous performance scores - allowing the teacher to choose previous test scores to initialize the Bayesian network. For example, as will be discussed later when we discuss the deployment results, the teacher can choose previous test scores that represent the contribution, collaboration, and evaluation aspects of a student and use their average performance scores in the classroom to set the current-task and future-task reward.

- **Update**—When the teacher assigns a grade to a group's completed written assignment, that grade becomes the current-task reward node value that group's tracked collaborative interactions (contribution, collaboration, and peer evaluations) are used to calculate the contribution, collaboration, and evaluation nodes. Furthermore, the evaluation scores are inserted as future-task reward node values for all previous assignment scores that had the same contribution, collaboration, and evaluation values.

Notice that each student agent is able to observe different student models and thus together divide the task (cf. division learning in (Weiss and Dillenbourg, 1998)) of observing the entire set of possible model-output combinations. This learning strategy is designed to allow the agents to learn the dynamics of the environment better.

5.3 Experiment Setup

To investigate the impact of MHCF algorithm and ClassroomWiki in formation and scaffolding student groups, we have deployed it in four university-level courses. Appendix C and Appendix C show the internal review board-reviewed format and the student and the parent consent forms that were used to get informed consent from the students and their parents for our deployments. Furthermore, Table 5.1 summarizes the use of ClassroomWiki

in three university-level courses and one community outreach activity. For our HIST 202 and CSCE 475 experiments, we adopted a control-treatment protocol. We divided the students in each deployment into the control and treatment sets based on prior individual student assessment scores-based on assignments assigned to the students prior to the ClassroomWiki activities-such that the average score for each set of students was similar. Note that *all* of our experiment setup was *double-blind*, i.e., neither the participating students nor the teacher who graded the student groups' Wikis and provided interventions knew which students belonged to the control or the treatment set.

Table 5.1: Deployments of ClassroomWiki

Deployment	Period	Experiment
HIST 202 America after 1877	(1) 3 Weeks 03/09-04/09 (2) 10/09-05/10	Asynchronous Collaborative Writing
CSCE 475 Multiagent Systems	3 Months 08/09-11/09	Asynchronous Collaborative Writing
ENGL 180 Intro. to Literature	1 Day 10/09	Synchronous Collaborative Writing
GEM Project	10/09–05/10	Mentoring Collaborative Writing
WMNS 101 Introduction to Women and Gender Studies	10/09-05/10	Asynchronous Collaborative Writing

5.3.1 HIST 202 Deployment

In the first deployment, 145 participating students were divided into control 72 and treatment 73 sets by following the aforementioned process. The control set students were further divided into 14 groups ($5 - 6$ members) randomly while the treatment set students into 14 groups ($5 - 6$ members) using the MHCF algorithm. Also, the student models were initialized with previous tests and assignments (Equation 5.1). The students then collaborated with their group members to prepare Wiki on "US as a super power" topic for three weeks.

Then the teacher reviewed each group's Wiki essay and scored each $[0 - 100]$ and converted each group's Wiki grade to the student members' individual grades by multiplying that group's grade with the relative contribution of that student; i.e., ev_i^1 for a student $s_i \in S$ member of group g is:

$$ev_i^1 \propto gr_i \times \Big[\frac{au_i^1}{\sum\limits_{s_i \in S} au_i^1} + \frac{au_i^2}{\sum\limits_{s_i \in S} au_i^2} \frac{ir_i^3}{\sum\limits_{s_i \in S} ir_i^3} + \frac{ir_i^4}{\sum\limits_{s_i \in S} ir_i^4} + \frac{ir_i^5}{\sum\limits_{s_i \in S} ir_i^5} +$$
$$\frac{ir_i^6}{\sum\limits_{s_i \in S} ir_i^6} + \frac{ir_i^7}{\sum\limits_{s_i \in S} ir_i^7} \Big] \quad (5.16)$$

where, gr_i is s_i's group grade, au_i^j for $j = 1, 2$ and ir_i^k for $k = [3, 7]$ are the tracked student activity (Section 5.1.4). At the request of the instructor, we are redeploying ClassroomWiki in the Spring semester of 2010 and are expecting to gather more data on ClassroomWiki.

5.3.2 CSCE 475 Deployment

In this deployment, the 17 participating students were divided into control (8) and treatment (9) sets. For initializing the Bayesian network, the course pre-requisite test scores were used. Then for each collaborative writing assignment, the control set students were further divided into 2 groups randomly while the treatment set students were divided into 3 groups using the MHCF algorithm. Each student then collaborated with his or her group members on their Wiki assignment writing up on a particular Multiagent Systems topic. After the due date, the teacher reviewed each group's Wiki essay and scored each $(0 - 100)$. Then the teacher reviewed each group's Wiki essay and scored each $[0 - 100]$ and converted each group's Wiki grade to the student members' individual grades by multiplying that group's grade with the relative contribution of that student; i.e., ev_i^1 for a student $s_i \in S$ member of

group g is:

$$sc_i \propto \sum_{k=1}^{12} \left[\frac{au_i^k}{\sum_{s_i \in g} au_i^k} \right] + \left[\frac{au_i^{10} + au_i^{11}}{\sum_{s_i \in g} au_i^{10} + au_i^{11}} \right] \tag{5.17}$$

$$ev_i^1 \propto gr_i \times [sc_i - median(SC_g)] \tag{5.18}$$

where, gr_i is s_i's group grade, au_i^j for $j = 1, 2$ and ir_i^k for $k = [3, 7]$ are the tracked student activity (Section 5.1.4), and SC_g is the set of all student contribution values in s_i's group g.

5.3.3 ENGL 180 Deployment

In this deployment, we used ClassroomWiki to allow the students to collaborate in a synchronous environment to: (1) build Wiki pages and (2) cross link their ideas and concepts on the assigned topics as they built those pages. For the ENGL 180 classroom, the students were assigned to write individual tropes on a variety of literary topics (e.g., Shakespeare - Sonnet 130). As they were writing their individual tropes, they referenced each others' work and cross linked the relevant literary ideas.

5.3.4 GEM Deployment

The Girl Empowerment and Mentoring (GEM) for Computing Project aims to inspire middle and high school girls towards computing in post-secondary education by (1) empowering them with skills and interest for IT and (2) mentoring them to improve their self-efficacy and motivation for a career in IT. To help the mentors involved in the GEM project to closely monitor the performance and contribution of the participating students, ClassroomWiki is being used for collaborative writing. The purpose of our GEM deployment was to further test the software and usability of ClassroomWiki software.

5.3.5 WMNS 101 Deployment

In this Women and Gender Studies course, we are using ClassroomWiki to investigate the collaborative knowledge building process over time in an asynchronous collaborative writing environment. In this classroom, the students work in groups to define a set of terms specified by the instructor over a semester. As they learn the subject matter and attend lectures from distinguished speakers, they are going to modify the definitions of those terms over time. ClassroomWiki is here used by the instructor to (1) form groups that encourage collaboration among their members and (2) track and model the members' activities to gain insights into the collaborative knowledge building process in the groups over time. However, over the semester, student dropouts, and inconsistent grading schedule our collected data did not yield any meaningful observations.

5.4 Results

5.4.1 User Acceptance

We have conducted a Wiki-rating survey (Appendix A) among the students to estimate students' view of ClassroomWiki as a collaborative writing tool. Table 5.2 shows the students' evaluation of ClassroomWiki environments. The analysis of the students' evaluations reflects that, on average, the students found ClassroomWiki useful. Also, when asked to compare, the students in the HIST 202 course rated ClassroomWiki to be better (average $3.4/5.0$) than the Blackboard's Wiki. The instructors who used ClassroomWiki in HIST 202 and CSCE 475 also expressed their approval of ClassroomWiki as a useful tool for implementing collaborative writing.

Table 5.2: User Rating Scores of ClassroomWiki (Likert Scale $[1, 5]$)

HIST 202 Mean	HIST 202 StDev	ENGL 180 Mean	ENGL 180 StDev	CSCE 475 Mean	CSCE 475 StDev
23.4	5.4	23.9	6.65	24.0	4.7

Teacher's Comments for HIST 202

I found ClassroomWiki's automated group formation and reliable web-interface to be useful for the students. The contribution tracking was able to provide accurate student contribution information and the captured contribution reflected the improvement effort of the students across all grade levels.

Teacher's Comments for CSCE 475

Besides automatic formation of student groups, I liked the individual student contribution assessment ability of ClassroomWiki. Using the detailed student activity summaries for an assignment, I was able to accurately gauge a students' contribution to his or her group and confidently assign grades to the students and proactively intervene in case of the non-contributing students.

5.4.2 Overall Student Performance and Collaboration

Analyzing *All* rows of Table 5.3, Table 5.4, and Table 5.5. To investigate ClassroomWiki's impact on the collaborative learning outcome of the students, we compare their scores in the ClassroomWiki assignment (All students' evaluation scores in Table 5.4) with other similar essay assignments (Table 5.5) and observe that the mean and median student scores in the three essay assignments do not show any clear trends/patterns. For example, there were no clear indications that the treatment students were better students to begin with. Furthermore, in the ClassroomWiki experiment, the students achieved lower mean and

higher median scores than the other assignments except Blackboard's Wiki assignment. However, in Blackboard's Wiki assignment, students' scores were likely to be inflated since Blackboard does not allow to the teacher to track individual student contributions (non-contributing students). This comparison suggests that, although some students have always achieved low scores in the tests (and thus produced low averages for the class), while using ClassroomWiki, some of those low-performing students have performed better when they collaborated with their group members. The improvement of those low-performing students' performances then raised the median score of the students in the ClassroomWiki assignment. In our experiment: (1) ClassroomWiki's collaborative tools (e.g., versioning in WIM, assignment-specific forum in COM), (2) teacher's periodic reminders, and (2) enforced accountability of each student's contribution due to our use of TAM could have motivated the students to collaborate with each other to improve the quality of their group's final essay. That improved participation then led to the improved median score.

Table 5.3: Student Evaluations in ClassroomWiki

Set	Min	Median	Mean	Max	StDev
Control	0.00	85.00	70.38	97.00	32.90
Treatment	0.00	83.00	74.84	95.00	24.69
All	0.00	83.00	72.62	97.00	29.05

Table 5.4: Standard Deviation of Group Members' Evaluation Scores

Set	Min	Median	Mean	Max
Control	0.00	34.00	27.40	41.64
Treatment	0.80	9.12	15.51	44.63

Table 5.5: Student Evaluations in Other Tests

Set	SEG	CRE	CLK	MID	BLK
Control	$mn = 77.0$ $md = 72.1$	$mn = 70.3$ $md = 75.0$	$mn = 41.2$ $md = 50.0$	$mn = 76.3$ $md = 83.0$	$mn = 75.0$ $md = 85.0$

Set	SEG	CRE	CLK	MID	BLK
Treatment	$mn = 75.5, md = 78.0$	$mn = 69.3, md = 77.0$	$mn = 38.2, md = 50.0$	$mn = 76.5, md = 77.0$	$mn = 79.2, md = 87.0$
All	$mn = 73.8, md = 77.0$	$mn = 69.8, md = 76.0$	$mn = 39.7, md = 50.0$	$mn = 76.4, md = 77.0$	$mn = 77.1, md = 85.0$

SEG - Segregation Essay 1/13/09, CRE - Civil Rights Essay 3/13/09, CLK - Clicker 4/30/09 MID - Midterm 2/27/09 , BLK - Blackboard Wiki 2/27/09
mn =Mean and md =median

5.4.3 Impact of MHCF Group Formation

Improvement in student scores and Collaborative Interactions One way to measure how well the agents were able to model the students and form student groups is to compare the performance and collaboration of control and treatment sets. Table 5.6 shows that the treatment set students achieved better scores (higher mean and lower standard deviations) than did the control set students (statistically significantly for HIST 202 with $p < 0.05$). Table 5.6 and Table 5.7, respectively, indicate that the treatment set students collaborated more (in terms of revisions and forum discussions) than the control set students in HIST 202 and CSCE 475 courses. The ability of the treatment set students' ability to better collaborate and learn could be attributed to MHCF's ability to form better groups using the Bayesian network. In HIST 202, the student agents in MHCF formed heterogeneous student groups and resulted in student groups that allowed them to write better quality collaborative essays. In CSCE 475, the student agents were able to observe the students' interactions to iteratively learn how to form better student groups yielding improved student performance and collaboration.

Table 5.6: Individual and Group Evaluation Scores

Set	HIST 202 Mean	HIST 202 StDev	CSCE 475 Mean	CSCE 475 StDev
Control	70.40	32.90	68.35	30.00
Treatment	74.80	24.70	88.21	13.90

Table 5.7: Average Revision Count per Wiki Assignment

Set	HIST 202 Mean	HIST 202 StDev	CSCE 475 Mean	CSCE 475 StDev
Control	3.0	2.7	7.0	6.6
Treatment	3.8	3.2	9.1	4.0

Table 5.8: Average Forum Topic Count per Wiki Assignment

Set	HIST 202 Mean	HIST 202 StDev	CSCE 475 Mean	CSCE 475 StDev
Control	2.1	1.4	4.8	2.5
Treatment	2.4	1.8	6.0	5.0

Table 5.9: Average Forum Message Count per Wiki Assignment

Set	HIST 202 Mean	HIST 202 StDev	CSCE 475 Mean	CSCE 475 StDev
Control	2.3	2.0	11.6	7.3
Treatment	2.4	1.9	17.5	11.8

5.4.4 Composition of Formed Student Groups

The comparison of the group composition - a key attribute in a collaborative learning setting as discussed in (Roberts and McInnerney, 2007)-of the control and treatment set students helps us explain how and why the treatment set students performed better in the course deployments. First, in HIST 202 experiment, because the MHCF algorithm did not have learning opportunity as there was only one session of group formation, we have boot-

strapped the Bayesian network to form heterogeneous groups containing students of a variety of levels of competence, i.e., an introductory classroom, as recommended in (Cress and Kimmerle, 2008). The analysis of our results shows that, the average standard deviation of the treatment students' total prior scores were higher (13.9 vs. 8.3 with $p < 0.05$) than the control set students. This implies that our bootstrapping of the Bayesian network allowed the MHCF agents to form treatment set student groups that had a higher level of heterogeneity than the control set. With everything else being roughly equal, this increased level of heterogeneity is thus considered to be contributing to the increased collaboration and improved performance of the students in HIST 202 (Table 5.6, Table 5.8, and Table 5.9).

Unlike HIST 202, in the CSCE 475 deployment, the Bayesian network did have 6 iterations of group formation sessions to learn the composition of the groups. The comparison of the standard deviation of the control and treatment set students' (Figure **??**) evaluation scores shows that the randomly formed student groups were heterogeneous and the MHCF-formed student set groups were homogeneous in nature. Furthermore, a t-test confirms ($p < 0.05$) that the control set student groups had higher standard deviation on average than the treatment set students. Homogeneous student groups have been suggested by the collaborative learning researchers and practitioners (Johnson and Johnson, 1999) as a strategy to form student groups especially for settings where students on average have high expertise on the task they are collaborating to solve. Being an advanced course in computer science, CSCE 475 consists of students most of whom possess the prior knowledge and expertise to solve the task at hand. After the first round of group formation in CSCE 475, the Bayesian network was updated with the students' collaborations and evaluations encouraging a homogeneous composition of student groups. As a result, when the randomly formed student groups were heterogeneous in terms of their effort, evaluation of each other, and teacher's evaluation, MHCF agents cooperatively learned to form homogeneous student groups. Over time, that homogeneous composition of student groups allowed the treatment set students to collaborate and learn better than the control set students.

The results here thus suggest that ClassroomWiki's Bayesian network-enabled multiagent-based group formation method improves the students' performance by: (1) providing the teacher the option to bootstrap the Bayesian network to form students groups with a chosen composition (e.g., heterogeneous) and (2) learning the appropriate group composition (e.g., homogeneous groups for an advanced learner setting) for a given collaboration setting.
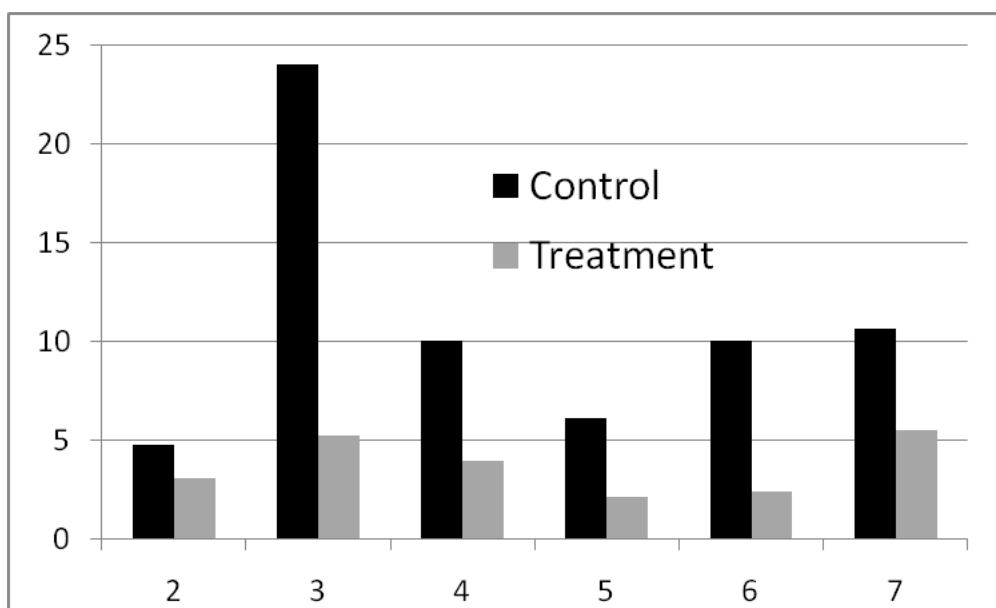

Figure 5.7: Average Standard Deviation of the Group Members' Scores in Control and Treatment Sets

5.4.5   Impact of Multiagent Tracking and Modeling

Accurate and Detailed Tracking of Student Activities

ClassroomWiki's Tracking and Modeling module (TAM) allowed us to track all student activities in the ClassroomWiki in details. Figure 5.8 and Figure 5.9 show the student activity counts over the entire period of our experiment. This tracked information was (1) used by the teacher at the end of the experiment to assess the individual contribution of the students and detect free-riding, (2) periodically checked by the teacher during the experiment to provide specific and precise guidance to the student groups to improve their

collaboration, and (3) analyzed to discover hidden trends and patterns (Section 5.4.5) in student behavior that could be utilized to improve the design of the collaborative writing assignment or environment (i.e., ClassroomWiki).
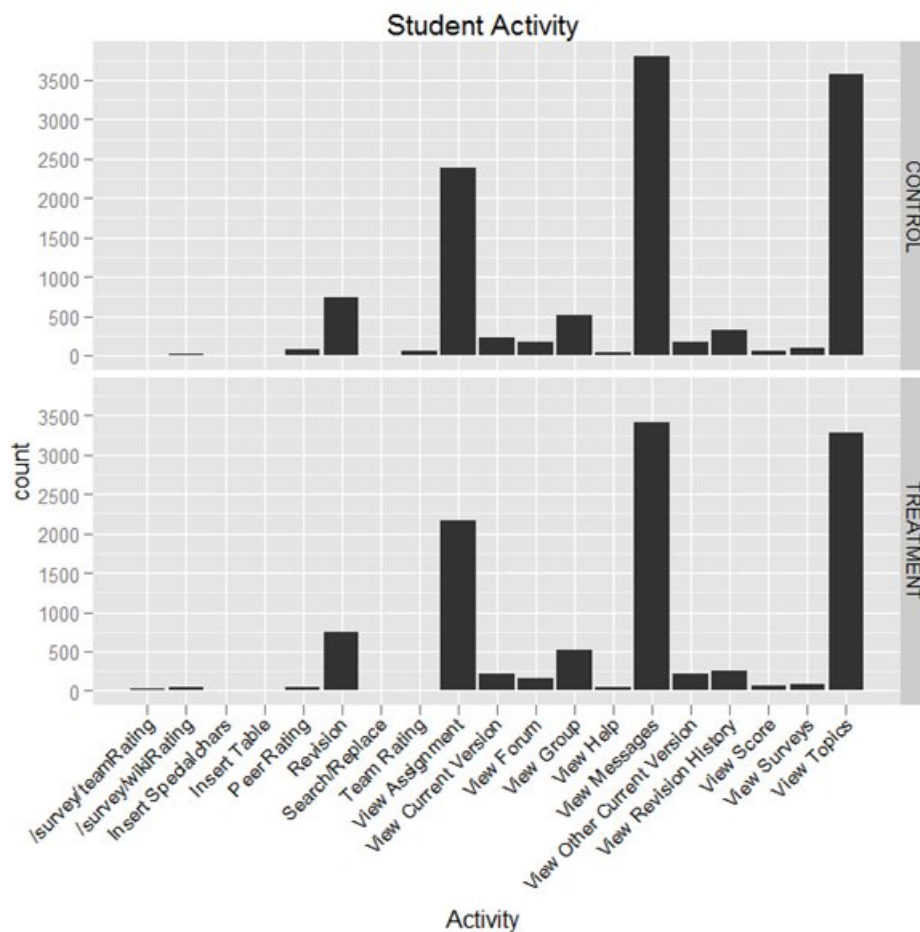
Figure 5.8: Tracked Student Activity Summary I

Identifying, Penalizing, and Alleviating Free-Riding

Although it is difficult to accurately verify in our experiment setup, there are *positive indications* that the student assessment done with the data collected by the TAM was able to accurately capture the level of individual contributions in our deployments. In HIST 202 Deployment, there were 13 students who did not revise their group's Wiki or post any fo-
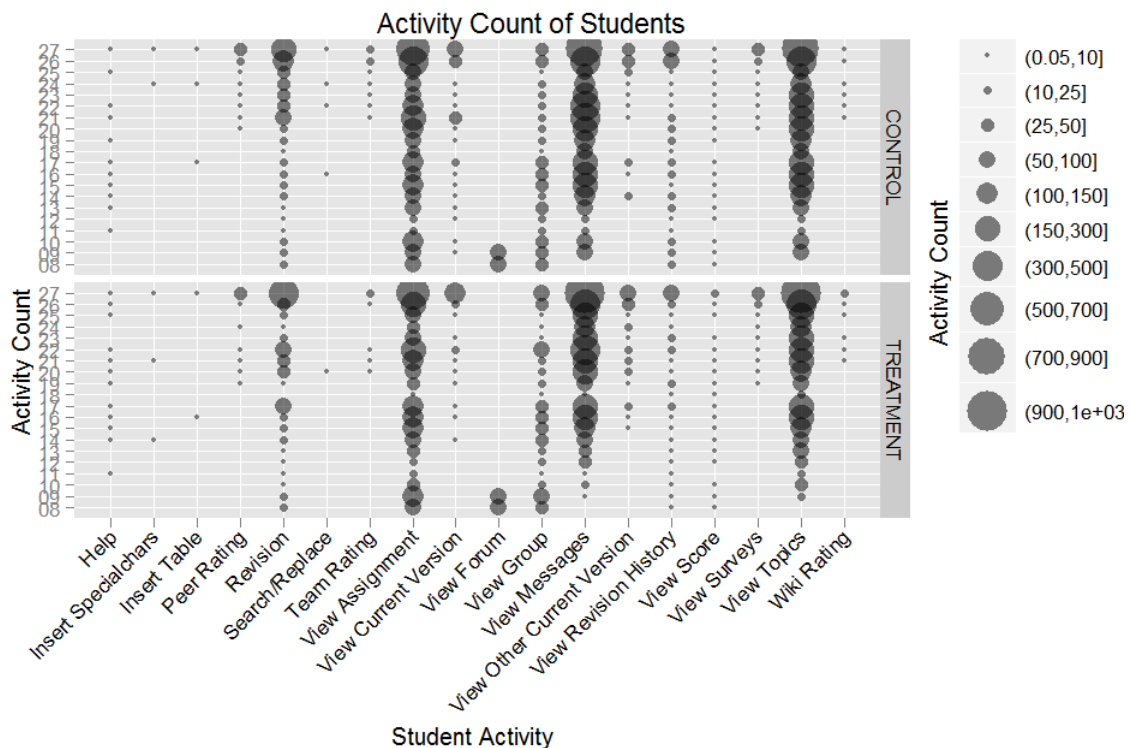
Figure 5.9: Tracked Student Activity Summary II

rum topics or messages). As a result, their individual student scores were 0. In CSCE 475 Deployment, there were also four cases of free-riding students. Though technically still contributing, these students posted trivial or next-to-meaningless contributions to the wiki or merely added or subtracted a few sentences moments before the due date. These types of student contributions resulted in: (1) yielded low contribution scores (for the students who added them at the last minute) due to our use of the number of group members' editions a particular edition survives (i.e., au_i^8 and au_i^9 in Equation 5.16, Equation 5.17) and (2) *annoyed* the other group members who could not review those last-minute editions. The instructor of the course in CSCE 475 deployment received timely reports from ClassroomWiki and was able to penalize and then e-mail or meet with the free-riding students to discuss their actions. As a result of these meetings, the 4 *different* non-contributing students in CSCE 475 *did not* repeat their free-riding behavior in future topic summary experiments. Therefore, these results suggests that the data collected by Researchers (Ebner et al., 2008)

mention that the main problem regarding free-riding is two-fold: first, often collaborative learning tools do not allow the teacher to accurately capture student's contributions toward his or her group. Second, if the students perceive that their group members are not held accountable for free-riding, they feel aversion toward collaborative work. So, the teacher's ability to identify and penalize free-riding students in the ClassroomWiki deployments suggests that the micro-level detailed tracking and modeling was able to alleviate these issues associated with free-riding.

Improved Assessment of Student Performance

Table 5.10 and Table 5.11 below show the correlations between the students' *scores in the Wiki assignments* (ClassroomWiki and Blackboard's Wiki) and their *scores in the other tests/assignments* in HIST 202 and CSCE 475 deployments. Using these correlation values, we were able to conduct a baseline comparison of ClassroomWiki's performance with respect to a Wiki that does not provide any tracking/modeling of student activities for individual assessment. This Wiki system provided by Blackboard had an interface and functionality similar to ClassroomWiki's but did not provide any tracking/modeling of students' interactions for assessing their contributions toward their groups. Table 5.10 and Table 5.11 indicate that the correlation between the students' exam scores in the class and their scores in Wiki were *higher* for ClassroomWiki than Blackboard Wiki's. Since these exam scores represent the knowledge and understanding of the students that they gained in the class, the higher correlation suggest that the detailed micro-level tracking and modeling of student behavior in ClassroomWiki allowed it to capture the performance of the students more accurately than Blackboard's Wiki.

Table 5.10: Correlation between the Student Scores in ClassroomWiki and Other Tests for HIST 202

Test/Assignment	ClassroomWiki	Blackboard Wiki
Final 05/01/09	0.69	0.54
Midterm Essay Exam 03/02/09	0.52	0.67
Civil Rights Essay 03/13/09	0.51	0.39
Origins of Segregation Document Analysis 1/13/09	0.30	0.18

Table 5.11: Correlation between the Student Scores in ClassroomWiki and Other Tests for CSCE 475

Test/Assignment	ClassroomWiki	Blackboard Wiki
Final 12/01/09	0.64	N/A

Furthermore, Table 5.11 shows that in CSCE 475 deployment, the students' evaluations for the topic summary scores were *highly* (> 0.6) correlated with their final exam evaluations. The values in Table 5.10 indicate that, except the first document analysis assignment, the scores the students received in the ClassroomWiki assignments were well correlated with their scores in the other assignments/exams. These moderately high correlation values suggest that individual student scores that were calculated based on ClassroomWiki's student contribution summary (e.g., number of words added/deleted, number of forum messages posted, etc.) closely represented the actual performance of the students in the other tests and assignments in the class.

Hidden Patterns of Student Behavior

After the ClassroomWiki experiment was over, we have analyzed the ClassroomWiki data for hidden trends and patterns in student behavior which could help us improve the design of the collaborative writing assignment and the environment. We have found an interesting periodicity pattern in student activities. When we counted the student activities in the

ClassroomWiki over the entire course of the assignment, we saw that the students' interaction count rose and fell with a period of seven days. Figure 5.10 shows the student activity count over the three-week period in the HIST 202 deployment and that the ClassroomWiki students worked less on Fridays and more near the middle of the week (i.e., Tuesdays and Thursdays). Furthermore, the students' activity count increased every week for the three weeks of the assignment to rise from 1000 during the first week to ≥ 6000 on the third week. Since we assume that the knowledge gain results from the students' participation in the collaborative activities (i.e., Wiki revisions, posting messages, topics), one way we could improve the collaborative learning outcome of the students is by changing the design of the assignment so that the students are more motivated to collaborate *early* rather than *near the due date*. One way the impact of due date could be utilized to motivate the students is by dividing the Wiki assignment into segments and assigning separate intermediate due dates for those segments. For example, the "US as a super power" assignment used in ClassroomWiki's experiment could be divided into three stages (topic choice, evidence gathering, and final Wiki) where the stages have three different due dates, the Wiki sections submitted at each stage are graded separately for each student group.This result suggests the potential of ClassroomWiki's multiagent tracking and modeling for providing valuable insights into the collaborative activities of the students.

Identifying Students' Trend of Improvement

We have also categorized the students according to the change in their scores from the midterm to the final exam. Table 5.12 shows that the students whose performances (i.e., their scores) *improved* from the midterm to the final (second row of Table 5.12) had achieved *higher* mean (difference statistically significant $p < 0.05$) and *higher* median scores in the ClassroomWiki assignment than the students whose performances (1) did not change or (2) deteriorated. This demonstrates that our *micro-level* detailed tracking and modeling in ClassroomWiki may identify the students who are trying harder and improving their per-
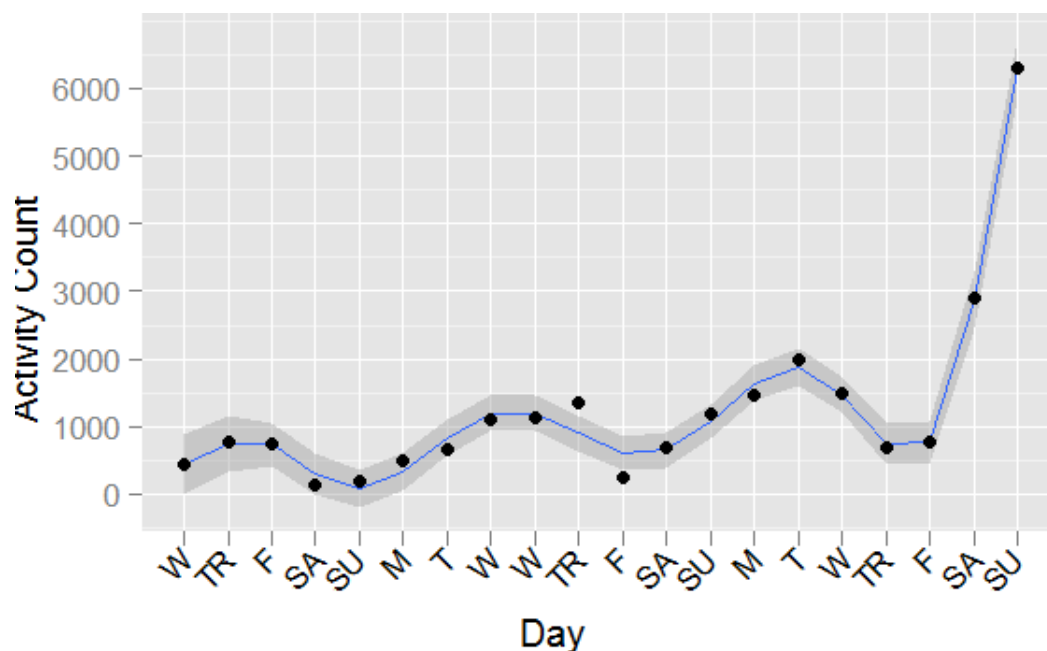
Figure 5.10: Weekly Student Activity Pattern in HIST 202

formance in the classroom and suggests our micro-level model's potential as a metric for capturing the students' learning progress.

Table 5.12: Student Scores in ClassroomWiki HIST 202 Deployment

Category	Median	Mean
Students with *No* change in the scores (Final → Midterm)	69.0	50.0
Students with *Positive* Change (Final → Midterm)	85.0	81.16
Students with *Negative* Change (Final → Midterm)	78.50	65.77

5.5 Conclusions

We have presented ClassroomWiki, designed based on the educational research on modeling the collaborative learning process, to improve typical Wiki's functionalities in two aspects: (1) individual student contributions and (2) group formation. We have also discussed how the MHCF algorithm was realized in the ClassroomWiki environment to solve the MCFP-M problem in the asynchronous collaborative writing setting. We have reported on our experiments regarding collaborative Wiki assignment in a university-level history course. Although not all results were statistically significant, our analysis suggests that ClassroomWiki and MHCF may (1) improve the collaborative learning outcome of the students by its group formation framework, (2) help the teacher identify and penalize free-riding students, and (3) facilitate specific and precise teacher interventions based on the tracked student activities.

5.6 Future Work

The future work regarding our implementation of *i*HUCOFS framework and the MHCF algorithm in ClassroomWiki involves the following. First, we plan to enrich the scaffolding capability of the mediator or student agents by adding an implicit scaffolding machine. Providing of implicit scaffolding to the students using agents is a well-researched area in *Intelligent Tutoring* systems. We plan to incorporate the existing intelligent tutoring systems research (Jung and VanLehn, 2010; Arroyo et al., 2010) in the mediator agents' reasoning so that they are able to explicitly scaffold the human users. Furthermore, to improve the agent learning capability in the MHCF algorithm, we plan to study the impact of choosing different Bayesian network structures on the mediator agents' ability to form better groups. Finally, we plan to deploy ClassroomWiki to larger classrooms in future to collect more data and better validate the performance of MHCF algorithm in terms of forming and scaffolding student groups.

Chapter 6

MCFP-A: ADLIT Framework

Here in Section 6.1, we describe the environment of the ADLIT framework. Then in Section 6.2, we describe a coalition formation protocol that is used by the ADLIT agents while forming coalitions. In Section 6.3, we describe the MCT, OSL, and LEA approximation strategies that are used by the agents to evaluate the coalition proposals in terms of current-task vs. future-task reward tradeoff for coalition formation. Section 6.4 describes how we implement the ADLIT framework in an agent-based simulation environment and a previously published coalition formation problem scenario. We then report our experimental setup in Section 6.5 and assess the performance of our ADLIT framework in Section 6.6. Finally, we conclude in Section 6.7 and discuss our future work in Section 6.8.

6.1 ADLIT Framework

In this section, we first present a set of assumptions that define the agents, the tasks the agents solve, and the learning actions that the agents use to change their types. Based on these assumptions, we describe ADLIT's environment, the coalitions, and the dynamics of the environment (i.e., what determines the solution quality of the tasks and the type changes in the coalition members). We then define the current task vs. future task tradeoff in the ADLIT environment and formally define the coalition formation problem.

6.1.1 Assumptions

Assumption 1 (Task)

There is a set of tasks in the environment that needs to be solved. Each task contains a set of subtasks and some divisible reward that is available upon solving it.

Assumption 2 (Agents and Coalitions)

There is a set of agents in the ADLIT framework's environment. Although not every agent is required to do so, these agents can form or join disjoint coalitions to solve the tasks in the environment and earn rewards. The reward earned by each coalition is distributed among its members according to some distribution scheme agreed upon by the members while forming the coalition.

Assumption 3 (Agent Type)

Each agent in the environment has a type that is composed of a set of *capability-expertise* pairs. The capability specifies the nature of the subtask that agent is able to solve and the expertise value denotes how well that agent is able to solve that particular subtask.

Assumption 4 (Task Solution Quality and Associated Uncertainty)

There is a probabilistic mapping between the types of the coalition members and the solution quality they can achieve. Here, the probabilistic nature of the mapping is due to the uncertainty in the environment. Furthermore, the agents are not aware of this mapping and must learn to estimate this mapping by observing the outcome of a coalition (in terms of task solution quality) when they solve a task.

Assumption 5 (Type Change through Implicit and Explicit Learning)

The agents in ADLIT are able to change their types only by learning from their experience in a coalition. Here, the change in the type of an agent means: (1) that agent learning a new capability or (2) the positive or negative changes in that agent's expertise values of its capabilities. Furthermore, the change in the type of an agent may result from implicit and explicit learning where these two types of learning are defined as:

- **Implicit Learning**: While working in a coalition, an agent could act as a learner and observe the other coalition members' actions in the coalition (e.g., learning by observation (Inaba et al., 2000)) to learn to solve a task better without requiring any interactions from those coalition members, we call this implicit learning.

- **Explicit Learning**: An agent may also act as a teacher agent and provide explicit guidance to a learner agent (i.e., a member of its own coalition) so that the chosen learner agent can improve its type by learning from that guidance to solve future tasks better. An example of explicit learning is Learning by Being Taught (Inaba et al., 2000) where the teacher agent provides explicit guidance to a learner agent to teach it how to solve some subtask better.

Assumption 6 (Type Change and Associated Uncertainty)

Due to the uncertainty in the environment, the mapping from the (1) the types of a coalition's members, (2) the task which they are solving, and (3) the specific implicit or explicit learning action being used, to the change in the coalition members' types that occur is probabilistic.

Assumption 7 (Cost of Implicit and Explicit Learning)

There is a cost associated with the implicit and explicit learning actions. In the explicit learning scenario between a learner agent and a teacher agent, the learner agent incurs the

learning cost and the teacher agent incurs the teaching cost. Furthermore, if the learner agent is implicitly learning from a coalition member without that coalition member providing any explicit guidance, the learner agent pays the learning cost.

Assumption 8 (Effectiveness of Learning Actions and Emergent Global Performance Improvement)

We assume that there is an order of effectiveness (in terms of improving the types of the agents) of the implicit and explicit learning interactions performed by the agents. Our assumption is driven by the learning theory research (Hoppe et al., 2003) that describe the variances among the impact of different learning actions (learning by being taught, learning by apprenticeship) and the relative difference in the teacher's and learner's expertise in improving the learner's expertise (i.e., type in our case).

Notice that there is an implicit connection between the types (i.e., expertise) of the teacher and the learners in the ADLIT environment that may potentially impact the performance of all agents in the systems. For example, if the teacher agent has high expertise compared to the learner, it may help the learner agents to earn higher expertise from their chosen implicit and explicit learning actions (Hoppe et al., 2003). So a resultant type change that yields a high expertise value for an agent may potentially impact a large number of future coalition members. That is, the improvement in the type of an individual (i.e., the learner agent) helps the system improve the types of other agents (i.e., the potential future learners) when that individual opts to become a teacher agent in the future, resulting in the improvement in overall system performance. This "emergent" impact of individual local decision (i.e., an agent's decision to improve its own type or the types of its coalition members) on the global performance (i.e., the overall performance of all agents working in the coalitions in the ADLIT environment) is one of the key aspects of our research in this domain.

Assumption 9 (Current and Future Task Reward)

The rewards achievable by an agent in a coalition can be divided into two generic categories: (1) current-task rewards and (2) future-task rewards. An agent's current-task reward is composed of its share in the reward its coalition has earned by solving a task. This current task reward can be exactly calculated by an agent when a coalition completes the assigned task. The future-task reward is the change (positive or negative) in the rewards of the future tasks that is expected to be available due to the change in an agent's own type and the types of its coalition members as a result of implicit and explicit learning.

Assumption 10 (Reward Distribution)

Since we assume that the solution quality of the task of a coalition primarily depends upon the types of its members, we assume that after a coalition receives a reward after completing a task, that reward is distributed to the members according to a distribution scheme in which the agents that have better types receive higher reward.

Assumption 11 (Agent Type Revelation)

Since the reward distribution scheme distributes the coalitional reward based on its members' types (Assumption 10), we assume that, while negotiating to form a coalition, the negotiating agents know each other's types.

Assumption 12 (Time)

There is a notion of discrete time in the ADLIT environment. All events (agents forming coalitions, agents earning rewards, etc.) occur at discrete time quanta.

Assumption 13 (Task Openness)

The set of tasks in the environment changes over time, i.e., new tasks are added to this set and existing tasks are removed from this set. Exactly when and which task is added to or

removed from the set of tasks is unknown to the agents.

Assumption 14 (Agent Openness)

The set of agents in the environment changes over time, i.e., new agents are added and existing agents are removed from the environment. Exactly when and which agent is added to or removed is unknown to the agents.

Assumption 15 (Partial Observability)

The environment is partially observable. In particular, each agent is only able to observe its own type change. Further, each agent has only access to information such as coalition rewards for its own coalitions. For example, an agent A who has observed another agent B receiving rewards when working with B will not be able to observe how many other rewards that B has received from coalitions of which A is not a member.

6.1.2 Environment

The environment of the ADLIT framework is defined as:

$$E = \langle \mathcal{T}, \Theta, \mathcal{N}, \mathcal{I}, \mathcal{O}, \mathcal{K}, \nu, \rho, \psi, \zeta, \varphi, \mu \rangle \tag{6.1}$$

In Equation 6.1:

- $\mathcal{T} = \{T_1, \ldots, T_{|\mathcal{T}|}\}$ is a set of tasks. Here, a task is defined as $T_k = \langle S_k, R_{max,k} \rangle$ where S_k is a set of subtasks, $R_{max,k} \in \mathbb{Z}$ is the maximum and divisible reward for solving task T_k.

- $\Theta = \{\theta | \theta = \langle \alpha, \beta \rangle, \alpha \in A, \beta \in B\}$ is a set of all possible agent types in E where θ's are the possible types of the agents $\mathcal{N}$ defined by capabilities A and expertise B. Here, A, B contains nominal values. For example, if the environment contains software engineering tasks, A could contain $\{coding, \ interface \ design\}$ and B

could be $\{Good,\ Bad\}$. Notice that there is a *one-to-one* correspondence between the capabilities and the subtasks that are in the tasks $\mathcal{T}$.

- $\mathcal{N} = \{i = 1, \ldots, |\mathcal{N}|\}$ is a set of agents where agent i has type θ_i.

- $I = \{ic_{i,j}^l, sc_{i,j}^l | l = 1, 2, \ldots, n_{|I|}, i, j \in N_C\}$ is a set of implicit learning- and explicit-learning- related actions where,

  - $ic_{i,j}^l \in I$ is the implicit learning action of agent i to learn (by observation) from coalition member j for a subtask $s_l \in T$.

  - $sc_{i,j}^l = \langle sc_i^l, sc_j^l \rangle \in I$ is the explicit learning action between agent i and agent j for a subtask $s_l \in T$. Here, agent i is the teacher agent and agent j is the learner agent. The teacher agent's action here is sc_i^l and the learning agent's action here is sc_j^l.

$O = \{\langle \kappa, \theta_C \rangle | \kappa_C \in \mathcal{K}\}$ is the set of possible outcomes when any coalition C solves task $T_C \in \mathcal{T}$ where,

  - $\kappa_C \in \mathcal{K}$ is the quality of the solution of task T_C.

  - $\theta_C = \{\theta_1, \ldots \theta_{|\mathcal{N}_C|}\}$ is the vector of resulting or updated type of the members $\mathcal{N}_C$ in coalition C.

- $\nu : \Pr(\kappa | \theta_C, T_C)$ is the task solution quality dynamics which probabilistically determines the task solution quality of a task T_C solved by members of coalition C who have the type θ_C.

- $\rho : \kappa_k \to R_k \forall\, T_k \in \mathcal{T}$ is the reward function which is *known* to the agents.

- $\psi : \Pr(\theta_C' | \theta_C, ic_C, sc_C)$ is the type change dynamics which probabilistically determines the new types of the members of a coalition C for: (1) given types of coalition members θ_C, (2) the coalition members' implicit learning actions (ic_C), and (3) explicit learning actions (sc_C). This function is *unknown* to the agents.

- $\zeta : \{ic, \langle sc_i^l, sc_j^l \rangle \in \mathcal{I}\} \to \mathbb{Z}$ is the learning cost function which is known to the agents.

- $\varphi : \Pr(T_C|t)$ denotes the task openness, i.e., the probability of encountering a task T at a time t. This probability is *unknown* to the agents.

- $\mu : \Pr(\theta_s|t)$ denotes the agent openness, i.e., the probability of encountering a set of agents S of types θ_s at a time t while forming coalitions. This probability is unknown to the agents.

6.1.3   Coalition

A coalition C is defined as:

$$C = \langle \mathcal{N}_C, \theta_C, T_C, d_C, ic_C, sc_C \rangle \tag{6.2}$$

where in Equation 6.2

- $\mathcal{N}_C \subseteq \mathcal{N}$

- $\theta_C = \{\theta_i | \, \forall i \, \mathcal{N}_C\}$

- $ic_C = \{ic_{i,j}^l | \, l = 1, 2, \ldots, n_{|\mathcal{I}|}, i, j \in \mathcal{N}_C\}$ is the set of implicit learning actions of all coalition members

- $sc_C = \{sc_{i,j}^l | \, l = 1, 2, \ldots, n_{|\mathcal{I}|}, i, j \in \mathcal{N}_C\}$ is the set of explicit learning actions of all coalition members

6.1.4   Individual Reward and Utility

Say several agents form a coalition C with a reward demand vector $\boldsymbol{d}_C = \langle d_1, \ldots, d_{\mathcal{N}|C|} \rangle$, $d_i \in \mathbb{Z}$. Then, the relative demand of an agent i is $r_i(\theta_C) = \dfrac{d_i}{\sum_i d_i}$.

So, if C receives achieves a task solution quality κ, an agent $i \in \mathcal{N}_C$ receives an individual reward:

$$R_i(\kappa, \theta_C) = r_i(\theta_C) \cdot \rho(\kappa) \tag{6.3}$$

In Equation 6.3, ρ is the reward function known by all agents 6.1. Furthermore, if agent i has engaged in implicit and explicit learning actions $ic_{i,C} = \{ic_{i,j}^l, l = 1, 2, \ldots, n_{|\mathcal{I}|}, j \in \mathcal{N}_C - \{i\}\}$ and $sc_{i,C} = \{sc_{i,j}^l, l = 1, 2, \ldots, n_{|\mathcal{I}|}, j \in \mathcal{N}_C - \{i\}\}$ with its coalition members and incurred the learning cost $\zeta(ic_{i,C}, sc_{i,C})$, its earned utility for joining a coalition is

$$U_i(\kappa, \theta_C, ic_{i,C}, sc_{i,C}) = R_i(\kappa, \theta_C) - \zeta(ic_{i,C}, sc_{i,C}) \tag{6.4}$$

As discussed in Section 6.1, the coalition members' types determine the solution quality for a task and, in turn, that solution quality determines the coalition's reward. Due to the impact of a coalition member's type on its coalition's reward, we assume that an agent's reward demand is proportional to its contribution to the quality of that solution. In other words, for an agent $i \in C$, its reward demand is:

$$d_i \propto \sum_{\beta_i \in \theta_i} \frac{\beta_i}{\beta_{i,max}} \tag{6.5}$$

where $\beta_{i,max}$ is the *maximum expertise* possible for the chosen task T_C's subtasks, and β_i is the expertise of i for T_C's subtasks. So, an agent is able to demand the most reward or reward share when it is an expert on the subtasks included in their chosen task. Notice that this reward distribution scheme is similar to the impact of an agent's action on its individual reward in the ACE strategy (Weiss, 1993).

6.1.5 Reward Tradeoff

The current-task vs. future-task reward tradeoff encountered by an agent i while forming a coalition C (Equation 6.2) at time t is denoted by the following tradeoff:

$$
\begin{aligned}
Tradeoff_{C,t} &= \sum_{\kappa} [\Pr(\kappa|\theta_C, T_C) \cdot R_i(\kappa, \theta_C)] - [\zeta(ic_{i,C}, sc_{i,C})] \\
&+ \sum_{\Delta t} \gamma^{\Delta t} \sum_{T_f \in \mathcal{T}} \Pr(T_{C,\Delta t}) \\
&\cdot \sum_{\theta_{cf} \in \Theta} \Pr(\theta_{Cf}|ic_{i,C}, sc_{i,C}, T_f, \theta_C) \cdot \Pr(\theta_{Cf,\Delta t}) \\
&\cdot \sum_{\kappa \in \mathcal{K}} \Delta R(T_{f,\Delta t}, \kappa, \theta_C) \quad\quad\quad (6.6)
\end{aligned}
$$

where in Equation 6.6 $\Delta R(T_{f,\Delta t}, \kappa, \theta_C)$ denotes the improvements in the rewards of i for solving the task T_C at the future time $t + \Delta t$ and,

$$
\Delta R(T_{f,\Delta t}, \kappa, \theta_C) = \Pr(\kappa|\theta_{Cf}) \cdot R_i(\kappa, \theta_{Cf}) - \Pr(\kappa|\theta_C, T_C)R_i(\kappa, \theta_C) \quad\quad (6.7)
$$

6.1.6 ADLIT Coalition Formation Problem

From the microscopic perspective of coalition formation (Vassileva et al., 2002), we formally define the ADLIT coalition formation problem as:

The ADLIT coalition formation problem for an agent $i \in E$ refers to the joining or forming coalitions and choosing implicit and explicit learning actions for those coalitions while solving tasks in $\mathcal{T}$ so that its current task vs. future task reward tradeoffs incurred during coalition formations allow it to (1) optimally improve its types and (2) maximize its rewards with respect to the improved types θ_i over time.

6.2 Coalition Formation Protocol

Here we propose a protocol that allows the agents in the ADLIT framework to communicate with the other agents in the environment to form or join coalitions and select its implicit and explicit learning actions.

The coalition formation protocol in ADLIT is composed of multiple stages where at any stage of coalition formation, there exists a set of coalitions CS in E. Notice that, in those coalitions, it is possible that there are agents who are in *singleton* coalitions since they have not been able to form coalitions. For those singleton coalitions, demand vectors and the implicit and explicit learning actions would be null. With this described setup, the coalition formation protocol allows the agents to form coalitions in the following way. During a coalition formation stage, an agent is selected at random as a proposer where each agent has an equal probability of being selected and this coalition formation is repeated n times (n is a multiple of $\mathcal{N}$) allowing each agent to act as a proposer equal number of times. Algorithm 6.1 shows the details of the protocol followed by a proposer agent while forming coalitions. During the initialization step (Steps 2-6 in Algorithm 6.1), the proposer agent selects its neighborhood and then communicates with those neighbors (Step 7 in Algorithm 6.1) to learn their types. When the responding agents provide their types to the proposer agent (Step 2 in Algorithm 6.2), in Step 9 (Algorithm 6.1), that proposer agent prepares a coalition formation proposition $\sigma = C$ from its list of neighboring agents. In this step, the proposer chooses the members of the coalition and the implicit and explicit learning actions for those members of the coalition in such a way that the expected value of that coalition is maximum from its own perspective. In Steps 11-12 (Algorithm 6.1), the proposer agent communicates with each of the potential coalition members (Step 9 in Algorithm 6.1) to get their choice of explicit learning actions for the proposed coalition. Then in Step 15 (Algorithm 6.1), the proposer agent announces the preliminary choice of implicit and explicit learning actions of the proposed coalition members to all members. In Steps 12-14 (Algorithm 6.2), the responding coalition members finalize their explicit

learning actions. Notice that this announcement in Step 15 (Algorithm 6.1) enables a potential coalition member to coordinate their learning actions according to the implicit and explicit learning actions of its other coalition members. That means, a coalition member can choose to participate and learn from a teacher agent (i.e., an agent willing to teach) or choose to ignore that explicit learning offer and select some other implicit learning action. Finally, in Step 22 (Algorithm 6.1) if the value of the proposed coalition is higher than its current coalition, the proposer agent communicates with all responding agents with the final coalition formation proposal to check whether they agree to join the proposed coalition or not. If all responding agents agree (Step 18 Algorithm 6.2), the proposer (Step 24 Algorithm 6.1) forms the coalition. Once the coalition is formed, the proposer (Step 24 Algorithm 6.1) and the responder (Step 18 Algorithm 6.2) notify their respective previous coalitions that they are leaving the coalition.

Algorithm 6.1 Coalition Formation Protocol for the Proposer Agent $i_p \in C_p$.

1: **Initialization**:
2: **if** $|\mathcal{N}_{ne}| = 0$ **then**
3:     Randomly select $|\mathcal{N}_{ne}| \in \mathcal{N}$ agents to build neighborhood
4: **else**
5:     Replace agents who declined to join coalition in the last round with randomly selected agents
6: **end if**
7: Get types θ_{ne} from $\mathcal{N}_{ne}$ agents
8: **Proposal Preparation**:
9: Prepare proposal $\sigma = C$ that has the maximum value from neighborhood agents $\mathcal{N}_{ne}$
10: **Implicit and Explicit Learning Action Selection**:
11: **for all** Potential Coalition Members $j \in \sigma$ **do**
12:     Receive potential implicit, explicit learning actions: $sc_C \leftarrow sc_C \cup sc_{j,C}$, $ic_C \leftarrow ic_C \cup ic_{j,C}$
13: **end for**
14: **for all** Potential Coalition Members $j \in \sigma$ **do**
15:     Announce ic_C, sc_C
16: **end for**
17: **Explicit Learning Action Finalization**:
18: **for all** For potentials coalition members $j \in C$ **do**
19:     Receive explicit learning actions: $sc_C \leftarrow sc_C \cup sc_{j,C}$
20: **end for**
21: **Coalition Formation**:
22: **if** Value of coalition $C_p <$ value of Coalition C **then**
23:     **if** If all members of the coalition C agree **then**
24:         Form Coalition C
25:         Notify members of coalition C_p about leaving coalition
26:     **end if**
27: **end if**

Algorithm 6.2 Coalition Formation Protocol for the Responder agent $j \in C_j$ in response to i_p's proposal.

1: **Initialization**:
2: When requested, send type θ_j to proposer i_p
3: Receive coalition formation proposal $\sigma = C$ from proposer agent i_p
4: **Implicit and Explicit Learning Action Selection**:
5: **for all** Potential Coalition Members $i \in \sigma$ **do**
6:     Select implicit and explicit learning actions: $sc_C \leftarrow sc_C \cup sc_{j,i}, ic_C \leftarrow ic_C \cup ic_{j,i}$
7: **end for**
8: **for all** Potential Coalition Members $j \in \sigma$ **do**
9:     Announce $ic_C, \ sc_C$
10: **end for**
11: **Explicit Learning Action Revision**:
12: $ic_C \leftarrow$ implicit learning actions of the members of C, $sc_C \leftarrow$ explicit learning actions of the members of C (sent by i_p)
13: Choose explicit learning actions for members: $sc_{j,C} = \{sc_{j,i} | i \in C - j\}$ s.t. the expected value of coalition $C \in \sigma$ is maximum
14: Send modified coalition formation proposal σ' to i_p
15: **Coalition Joining**:
16: $\sigma \leftarrow$ finalized proposal from proposer i_p
17: **if** Value of coalition $C >$ value of coalition C_j **then**
18:     Join coalition $C \in \sigma$
19:     Notify members of coalition C_j about leaving coalition
20: **end if**

Whether an agent agrees to join a coalition proposed by a proposer agent depends on its evaluation of that coalition formation proposal (in Algorithm 6.1 and Algorithm 6.2). The value of a proposition σ for forming coalition $C = \langle \mathcal{N}_C, \theta_C, T_C, d_C, ic_C, sc_C \rangle$ from the perspective of an agent i is defined by the current-task vs. future-task reward tradeoff (re-writing Equation 6.6):

$$
\begin{aligned}
Q_i(C, B_i) &= \sum_\kappa [\Pr(\kappa | \theta_C, T_C) \cdot R_i(\kappa, \theta_C)] - [\zeta(ic_{i,C}, sc_{i,C})] \\
&+ \gamma V_i
\end{aligned}
\tag{6.8}
$$

$$V_i(B_i) = \sum_{T_f \in \mathcal{T}} \Pr(T_{C,\Delta t}) \cdot \sum_{\theta_{cf} \in \Theta} \Pr(\theta_{Cf} | ic_{i,C}, sc_{i,C}, T_f, \theta_C) \cdot \Pr(\theta_{Cf,\Delta t})$$
$$\cdot \sum_{\kappa \in \mathcal{K}} \Delta R(T_{f,\Delta t}, \kappa, \theta_C) \tag{6.9}$$

Comparing Equations 6.6, 6.8, 6.9, we see that the first term in Equation 6.8, is the current task reward in Equation 6.6. Furthermore, the second term in Equation 6.9, i.e., $V_i(B_i)$, is defined as the expected future task reward increase in the current task future task reward tradeoff in Equation 6.6 measured from the perspective of agent i.

6.3 Computational Approximations

It is quite impractical for the agents in the ADLIT's environment to evaluate the coalition formation proposals by solving the POMDP described by Equation 6.8 and 6.9, for a variety of reasons. First, the outcomes (in terms of current-task vs. future-task reward tradeoff) of the agents' decisions (choice of coalition and learning actions) are uncertain due to the change in the agent types (Assumption 8 in Section 6.1.1). So, spending resources for finding the maximum expected reward yielding solution (i.e., by solving the POMDP) may prove to be inefficient for an agent in terms of its reward-to-effort ratio. In addition, due to the partial observability (Assumption 12 in Section 6.1.1) and the openness (Assumption 13 and 14 in Section 6.1.1), an agent cannot accurately predict when (1) the task for which it is training (by implicit or explicit learning actions) and (2) the coalition members it is teaching will be available. Furthermore, the agents in the environment have their own beliefs and are not constrained by any central coordination mechanism. The *differing beliefs* mean that the agents are likely to have *differing evaluations* (especially the newcomer agents that are introduced due to agent openness) of the same coalition formation proposal. This difference in the evaluations makes it impossible for the ADLIT agents to

predict whether a coalition formation proposal will be accepted by the potential members (even if that proposal is the best according to its belief). As a result, there is *no guarantee* that an agent can accurately evaluate how its decisions at present (e.g., choice of coalition members and learning actions) will impact its rewards in the future by solving the POMDP in Equation 6.8 and 6.9. To summarize, the impact of the openness, the uncertainty, and the partial observability in the environment on (1) an agent's ability to form or join a coalition and (2) a coalition's ability to achieve a particular task solution quality indicate that spending resources for solving the POMDP will not be efficient. So, our ADLIT agents use approximation strategies instead of solving Equation 12, to ballpark the value of a coalition while forming or joining coalitions. These approximation strategies will allow the ADLIT agents to calculate their reward tradeoffs sufficiently accurately and with minimal computation to optimize their type changes and maximize their rewards with respect to their types over time.

We provide three strategies for the agents to approximate the value of a proposed coalition during coalition formation protocol (Algorithm 6.1 and 6.2). First, in the Myopic Current Task (MCT) Strategy (Chalkiadakis and Boutilier, 2008), the agents choose the coalition that provides the maximal current-task reward without taking advantage of the type-changing impact of the implicit and explicit learning actions. In the One-Step Lookahead (OSL) strategy (Emery-Montemerlo et al., 2004), the agents *look ahead* into the immediate round of coalition formation to approximate the reward gain in the future. Finally, in the Learning by Exploration (LEA) strategy, the agents utilize the exploration-exploitation issue to interact with its environment to maximize their current-task vs. future-task reward tradeoff. Notice that, we use the MCT strategy as a baseline to investigate the performances of our OSL and LEA strategies that are designed around the agents' exploitation of the implicit and explicit learning actions.

6.3.1 Myopic Current Task Strategy (MCT)

In this strategy, an agent *myopically* reasons about the value of a coalition and form coalitions such that each yields the maximum reward for the current task only, without engaging in any implicit and explicit learning actions. In other words, the MCT strategy using agents do not engage in the current task vs. future task reward tradeoff which is key to our ADLIT problem. We write the Myopic form of Equation 6.8 as:

$$Q_i(C, B_i) = \sum_\kappa [\Pr(\kappa|\theta_C, T_C) \cdot R_i(\kappa, \theta_C)] - [\zeta(ic_{i,C}, sc_{i,C})] \tag{6.10}$$

6.3.2 One-Step Lookahead Strategy (OSL)

In the One-Step Lookahead strategy, an agent focuses only on the next step of the coalition formation round to calculate the future task reward (third term in Equation 6.8) when determining the value of a proposed coalition. So, suppose that θ_{Cf} is the resulting coalition members' types due to the implicit and explicit learning provided by the current coalition C, and T_f is the future task for the next round of coalition formation, we write Equation 6.9 as:

$$
\begin{aligned}
V_i(B_i^0) &= \sum_{T_f \in \mathcal{T}} \Pr(T_f) \cdot \sum_{\theta_{Cf} \in \Theta} \Pr(\theta_{Cf}|ic_{i,C}, sc_{i,C}, T_f, \theta_C) \\
&\quad \cdot \sum_\kappa [\Pr(\kappa|\theta_{Cf}, T_f) \cdot R_i(\kappa, \theta_{Cf}) - \Pr(\kappa|\theta_C, T_f) \cdot R_i(\kappa, \theta_C)]
\end{aligned} \tag{6.11}
$$

where in Equation 6.11,

- $\Pr(T_f)$ is the probability of seeing the Task T_f in a future coalition formation round (captures the task openness);

- $\Pr(\theta_{Cf})$ is the probability of seeing the coalition members with types θ_{Cf} in a future coalition formation round (captures the agent openness);

- $\Pr(\theta_{Cf}|ic_{i,C}, sc_{i,C}, T_f, \theta_C)$ is the probability of achieving updated coalition member type θ_{Cf} for (1) task T_C, (2) types θ_C, (3) implicit learning actions ic_C, and (4) explicit learning actions sc_C;

- $\Pr(\kappa|\theta_C, T_C)$ is the probability of achieving task solution quality κ for coalition member types θ_C and current task T_C;

- $R_i(\kappa, \theta_C)$ is the reward of an agent i in coalition C when they achieve solution quality κ (Equation6.3).

So, the future task reward gain component in Equation 12 for an agent is then the expected increase in the individual reward of that agent over (1) the set of all possible tasks, (2) the set of all possible types of the coalition members, and (3) the set of possible task solution qualities.

6.3.3 Learning by Exploration Strategy (LEA)

The key component of the current task vs. future task reward tradeoff in the ADLIT environment lies in the agents' ability improve their types considering the openness and uncertainty of the environment. However, to improve their types efficiently (e.g., with low cost of learning), the agents have to learn the *type change dynamics* (Section 6.1) of the environment. Since the agents have neither prior knowledge nor any central coordination mechanism, they have to observe the impact of their participation in the varying learning actions to learn the type change dynamics. Furthermore, each round of coalition formation provides an agent a single opportunity to observe the result of its choice of the learning action. So, during a coalition formation round, an agent may choose to exploit: i.e., form coalitions based on the type change dynamics knowledge it has acquired, or explore: i.e., form coalitions and choose a new learning action (or a learning action whose impact it has not observed enough) to better understand the impact of the chosen action on the type change of the agents. Here, this balance between the exploration and exploitation is critical

to the agents' ability of utilizing the current task vs. future task reward tradeoff. Notice that, the possible number of implicit and explicit learning actions in the ADLIT environment (Section 6.1) could be large. So, if the learning actions *vary* in terms of changing the types of the agents and the learning cost, an agent can *improve* its reward tradeoff (Equation 6.6) even further by improving its own type and its coalition members' types with less cost. However, this exploration of the learning actions must be done with caution for two reasons. First, while exploring, it is possible that an agent can actually hurt its reward tradeoff by choosing the wrong learning actions. Second, due to the uncertainty associated with the type change dynamics of the agents (Section 6.1), an agent may have to engage in a learning action multiple times to closely approximate its effect on the types of the participating learner agents. Once an agent has sufficiently explored its learning actions, it should choose the learning action that yields the maximum reward tradeoff values for itself and its coalition members. So, we need a strategy that would take all of these factors into account and allow the agents to balance its exploration and exploitation effort regarding the learning actions in ADLIT environment.

To balance the agents' exploration and exploitation, we have designed the LEA strategy as a refinement of the One-Step Lookahead strategy. The LEA strategy uses an idea similar to simulated annealing (Russel and Norvig, 2003, p. 115) to provide a mechanism that could be used by the agents to balance its exploration vs. exploitation of the environment to choose more efficient learning actions over time. To define the LEA strategy, we modify our One-Step Lookahead strategy (Equation6.9) as:

$$
\begin{aligned}
V_i(B_i^0) &= \sum_{T_f \in \mathcal{T}} \Pr(T_f) \cdot \sum_{\theta_{Cf} \in \Theta} \Pr(\theta_{Cf} | ic_{i,C}, sc_{i,C}, T_f, \theta_C) \\
&\quad \cdot \sum_{\kappa} [\Pr(\kappa | \theta_{Cf}, T_f) \cdot R_i(\kappa, \theta_{Cf}) - \Pr(\kappa | \theta_C, T_f) \cdot R_i(\kappa, \theta_C) \\
&\quad + e^{\frac{En(ic_C, sc_C, T_C, \theta_C)}{Tm(ic_C, sc_C, T_C, \theta_C)}}]
\end{aligned}
\tag{6.12}
$$

Say we define Confidence $conf(ic_C, sc_C, T_C, \theta_C) : ic_C \times sc_C \to \mathbb{Z}$ to represent an agent's experience regarding the implicit and explicit learning actions ic_C, sc_C. This confidence is proportional to the number of times the agent has observed the type change θ_{Cf} for given task T_C, coalition member type θ_C, implicit learning action ic_C, explicit learning actions sc_C. Then in Equation 6.12:

$$En(ic_C, sc_C, T_C, \theta_C) \propto \frac{1}{\frac{conf(ic_C, sc_C, T_C, \theta_C)}{\sum_{ic_C, sc_C \in \mathcal{I}} conf(ic_C, sc_C, T_C, \theta_C)}} \qquad (6.13)$$

represents the energy of the system, which is inversely proportional to the average confidence of the agent over all possible implicit and explicit learning actions $\mathcal{I}$.

$$Tm(ic_C, sc_C, T_C, \theta_C) \propto \frac{1}{conf(ic_C, sc_C, T_C, \theta_C)} \qquad (6.14)$$

represents that the temperature of the system is inversely proportional to the confidence $conf(ic_C, sc_C, T_C, \theta_C)$.

The key idea behind our design of the LEA strategy lies in the use of the energy and temperature to *bias* an agent's estimated value of a coalition based on what an agent knows about the type change dynamics (Section 6.1.2) in the environment. The first component of Equation 6.12 is the same as Equation 6.11. As an improvement over the OSL strategy in Equation 6.11, we have added an exponential term $e^{\frac{En(ic_C, sc_C, T_C, \theta_C)}{Tm(ic_C, sc_C, T_C, \theta_C)}}$ inside the summation. Our goal of adding this term to the components in Equation 6.12 is that the relative differences in the values of the first and the second term (inside the braces) would enable an agent to balance between explorative and exploitative behavior. In other words, we aim to bias an agent's estimated tradeoff values calculated using the first term of Equation 6.12 (similar to the OSL strategy in Equation 6.11) for a chosen implicit and explicit learning action by the energy and temperature in the second term. During the initial rounds of coalition formation, an agent would not have a high number of observations for some of the implicit and explicit learning actions. This would yield a high value of the energy (due to

the inverse proportionality in Equation 6.13) and then for those implicit and explicit learning actions ic_C, sc_C that have relatively smaller number of observations, the temperature (Equation 6.14) would be low. Those high energy and low temperature values would yield a high value for the $e^{\frac{En(ic_C,sc_C,T_C,\theta_C)}{Tm(ic_C,sc_C,T_C,\theta_C)}}$ term. As a result, the agent using the LEA strategy would be more likely to explore the learning actions ic_C, sc_C whose impact it knows *less* about. In that case, the agent's evaluation of a coalition formation proposal will depend *more* on the exploration of the impacts of the relatively unknown learning actions and *less* on the exploitation of the learning actions it knows about.

Over time, as the agent observes the impacts of available learning actions, the opposite scenario would occur. As an agent observes the impacts of the available learning actions in the environment, the energy value $En(ic_C, sc_C, T_C, \theta_C)$ would decrease (due to the inverse proportionality in Equation 6.13). Then the temperature values for the implicit and explicit learning actions ic_C, sc_C that an agent has high confidence about, will be high. Those high energy and high temperature values would yield a lower value for the $e^{\frac{En(ic_C,sc_C,T_C,\theta_C)}{Tm(ic_C,sc_C,T_C,\theta_C)}}$ term thereby diminishing its influence on the agent's evaluation of a coalition (Equation 6.8). In that case, the agent's evaluation of a coalition formation proposal will depend *more* on exploiting the implicit and explicit learning actions the agent knows about and *less* on exploring new learning actions. Finally, notice that, if after the agent interacts with the environment sufficiently long (yielding a low energy of the system), and the temperature for a given implicit and explicit learning action remains low, that would mean, due to its *not-so-good* impact on the types of the agents, the agent chose not to exploit those learning actions. In that case, due to the low energy value (i.e., a lower value of the term $e^{\frac{En(ic_C,sc_C,T_C,\theta_C)}{Tm(ic_C,sc_C,T_C,\theta_C)}}$, the agent using the LEA strategy would not choose those low-type-change-yielding implicit and explicit learning actions.

To summarize, using the LEA strategy, an agent is able to balance its exploration of new learning actions and exploitation of the learning actions it is confident about. Since not all learning actions may change the types of the agents positively and since the environment

is uncertain, the agents in the LEA strategy may perform poorly in the beginning due to its exploration. However, as an agent gathers experience regarding the available learning actions, it should be able to learn to choose the learning actions that yield optimal type change but incur the lowest cost. That efficient use of the learning actions is likely to improve the LEA agents' performances over time. Later in our Results section (Section 6.6), we investigate the impact of the LEA agents' exploration and exploitation behavior.

6.3.4 Frequentist Learning of Environment Dynamics

To design the agent strategies, we formulate the type change of the agents due to the implicit and explicit learning in the repeated coalition formation environment of ADLIT as a partially observable MDP (POMDP) (a.k.a. a belief state MDP) (Dearden et al., 1999). Let us assume that the agents $i \in \mathcal{N}$ have initial belief B_i that contains the estimations regarding the dynamics and the openness of the environment. As an agent joins coalitions, observes the result of its choice of coalition members and the learning actions, it updates its beliefs and uses that updated belief to form coalitions in the next round. Say the set of coalitions in ADLIT at time t is CS^t and the belief of an agent i is B_i^t. If an agent i is in a coalition C at time t and i observes coalitional outcome $o \in \mathcal{O}$, then i updates its belief at time $t + 1$ as:

$$B_i^{t+1}(\nu) = \frac{Obs^t(\kappa | \theta_C, T_C)}{\sum_{\kappa \in \mathcal{K}} Obs^t(\kappa | \theta_C, T_C)} \tag{6.15}$$

$$B_i^{t+1}(\theta_{Cf}) = \frac{Obs^t(\theta_{Cf})}{\sum_{\theta \in \Theta} Obs^t(\theta_{Cf})} \tag{6.16}$$

$$B_i^{t+1}(\varphi) = \frac{Obs^t(T_C)}{\sum_{\kappa \in \mathcal{T} \in \mathcal{T}} Obs^t(T_C)} \tag{6.17}$$

$$B_i^{t+1}(\psi) = \frac{Obs^t(\Delta\theta_i|T_C, \theta_i, ic_C, sc_C)}{\sum_{\Delta\theta} Obs^t(\Delta\theta_i|T_C, \theta_i, ic_C, sc_C)} \tag{6.18}$$

In Equation 6.15-Equation 6.18:

- $Obs^t(\kappa|\theta_C, T_C)$ denotes the frequency of observing task solution quality κ when the coalition members type was θ_C and the task was T_C

- $Obs^t(T_C)$ denotes the frequency of observing task T_C

- $Obs^t(\Delta\theta_i|T_C, \theta_i, ic_C, sc_C)$ denotes the frequency of observing the individual type after the agent of type θ_i have participated in a coalition C for task T_C, implicit learning action ic_C, and explicit learning action sc_C

Notice that in Equation 6.15-Equation 6.18, the agents use a frequentist (Amati, 2006) approach where they use their observations to update their estimates of the probabilities.

6.4   Implementation

In this section, we first describe how we have adopted the coalition formation problem described by (Chalkiadakis and Boutilier, 2008) for the ADLIT framework. Then we describe how the ADLIT framework is implemented in the agent-based simulation environment Repast.

6.4.1   Coalition Formation Environment Overview

The coalition formation environment in our implementation is an extension of the coalition formation environment described in (Chalkiadakis and Boutilier, 2008) with the added possibility that the agents may improve their types by learning. According to our Assumption 3 (Section 6.1.1), in the adopted coalition formation environment, each agent has *three* different capabilities (Section 6.1.2): interface design, programming, and system engineering.

For each of these capabilities, there are four possible expertise levels (Section 6.1.2): *bad*, *average*, *good*, and *expert*. The numeric values associated with the expertise of the agents are $bad = 1$, $average = 2$, $good = 3$, and $expert = 4$. So, for example, an agent i's type could be:

$$\theta_i = \{\langle interface\ design, 1\rangle, \langle programming, 2\rangle, \langle systems\ engineering, 3\rangle\} \quad (6.19)$$

Notice that as previously discussed in the ADLIT Environment Subsection (Section 6.1.2), we use a finite set of discrete values to represent the expertise of the agents, which is similar to the original coalition formation scenario described in (Chalkiadakis and Boutilier, 2008), for three reasons. First, we have chosen a finite set of discrete values since a continuous range of values would rapidly (and infinitely) increase the number of possible agent types (θ_i Section 6.1.2) in the environment. That increased number of possible agent types would render the agent reasoning too complicated or even intractable to be used in a practical and or scalable coalition formation scenario. Second, a continuous range of values $(0, 100.0)$ can be categorized or discretized into a finite set of discrete ranges, e.g., $(0, 10), (10, 20.0)$ or bins, i.e., that continuous range can be converted to a set of discrete values. Although such discretization may reduce the resolution of the ability of observation of an agent, it would increase the efficiency of the reasoning of the agents' (i.e., who use that discretization) and would allow the agents' (i.e., who use that discretization) to better deal with the noise in environment. Finally, since we evaluate our coalition formation protocol (Algorithm 6.1 and Algorithm 6.2) and the approximation strategies (Section 6.3) by comparing their performances with an approximation strategy (MCT Section 6.3.1) similar to the Myopic Current Task strategy described in (Chalkiadakis and Boutilier, 2008), we decided to have our coalition formation scenario to be similar to the original scenario described in (Chalkiadakis and Boutilier, 2008) i.e., use discrete expertise values. In our setup, the tasks are defined with required expertise values for each of the capabilities for the member

agents. For example, a task T_i may be defined as:

$$T_i = \{\langle ID, 13 \rangle, \langle PR, 2 \rangle, \langle SE, 8 \rangle\} \tag{6.20}$$

Here, ID denotes the required expertise for the interface designer, PR denotes the required expertise for the programmers, and SE is the required expertise for the software engineer.

The agents in our implemented environment participate in the coalition formation rounds where the coalitions solve software development tasks. After solving each task, a coalition earns rewards based on the quality of the solution prepared by them. This quality of the solution of the software development task is determined by the task solution quality dynamics (Section 6.1.2). The reward earned by the coalition is divided among the agents and each agent receives an amount equal to its individual reward less its cost of implicit and explicit learning actions. Furthermore, due to their participation in the learning actions, the coalition members' types change according to the type change dynamics (Section 6.1.2) of the environment. Then the coalition formation process is repeated with the updated types of the agents.

6.4.2 Randomization in the Simulation Environment

Since the ADLIT framework represents an uncertain environment, there is a need for randomization based on certain distributions. For example, when the proposer agent is chosen during the coalition formation protocol (Section 3), our implemented environment needs to make a discrete random choice among the agents. To keep the choices of our simulation random and our experiments replicable, in the beginning of the simulation run, we initialize a uniform random number generator with a specified simulation seed. The uniform random number generator is then used to make the random choices in the environment. We define

the uniform random number generator as:

$$\iota(i,j) = [i,j] \,\forall i,j \in \mathbb{R}^+ \tag{6.21}$$

This uniform random number generator defined in Equation 6.21 is then used in our simulation environment whenever the simulation needs to make a random choice among the agents (e.g., choosing the proposer in the coalition protocol as described in Algorithm 6.1) and tasks (e.g., choosing the task that the agents would solve at a certain time Assumption 12 in Section 6.1.1). Notice that our random number generator can generate both integer and real numbers.

6.4.3 Task Solution Quality Dynamics

The task solution quality κ (Equation 6.1.2) for a task in the implementation of ADLIT is divided into four levels bad, $average$, $good$, $perfect$. As we have discussed in Assumption 4 (Section 6.1.1), when a coalition solves a task, the quality of their solution is primarily determined by the expertise of the members of that coalition and is affected by the uncertainty in the environment. The quality of the solution of a task by a coalition C is thus determined by the λ_1 function which is unknown to the agents in ADLIT:

$$\lambda_1(\theta_C, T_C) \propto \begin{cases} \sum_{i \in C} \beta_i - \sum_{i \in T_C} \beta_i & \text{If } \iota(0,1) < \aleph_1 \\ \sum_{i \in C} \beta_i - \sum_{i \in T_C} \beta_i + \epsilon_T & \text{Otherwise} \end{cases} \tag{6.22}$$

In Equation 6.22, $\sum_{i \in T_C} \beta_i$ is the sum of the required expertise for solving task T_C. Furthermore, $\aleph \in [0,1]$ denotes the task solution quality uncertainty threshold. So, Equation 6.22 shows that the task solution quality is equal to the ratio of (1) the difference between the sum of required expertise for the task and the sum of expertise of the members of the coalition and (2) the total required expertise of the task when the uncertainty value drawn from the random normal distribution falls below the uncertainty threshold. If that

random value falls above the uncertainty threshold, the task solution quality is varied (in the randomly chosen positive or negative direction) with the chosen value of ϵ_T.

6.4.4 Coalitional and Individual Reward

Depending on the quality of the solution κ_C, the reward of a coalition C for a task T_C is determined by the function ρ (Section 6.1.2):

$$\rho(\kappa) : \frac{\kappa_C}{\kappa_{max}} \tag{6.23}$$

Here, $\kappa_C \in \mathbb{Z}$ and $\kappa_{max} \in \mathbb{Z}$ is the maximum possible solution quality of the task. So, according to Equation 6.23, the reward of a coalition is proportional to the ratio of the solution quality of the task they solved and the maximum possible quality of that task. As a result, a coalition is able to achieve maximum reward if the quality of their solution for the task is maximum. Furthermore, the individual reward of an agent i after its coalition C solves a task T_C is:

$$R_i(C) \propto R_{max,C} \times \frac{\kappa_C}{\kappa_{max}} \times \frac{\beta_i}{\beta_{max}} - \zeta(ic_{i,C}, sc_{i,C}) \tag{6.24}$$

In Equation 6.24, $R_{max,C} \in \mathbb{R}$ is the maximum task reward of T_C and $\zeta(ic_{i,C}, sc_{i,C})$ is the total cost of implicit $ic_{i,C}$ and explicit $sc_{i,C}$ learning actions of agent i. In this manner, each agent receives rewards based on its expertise. This is a fair distribution of the rewards to the individual agents since the expertise of an individual agent directly contributes to the quality of the solution of a task (Assumption 4 in Section 6.1.1). Since the coalitional reward depends on the solution quality (Assumption 10 in Section 6.1.1), each member's expertise directly impacts the total amount of reward its coalition receives. In other words, higher expertise (i.e., higher β_i) of an agent expectedly (due to the task solution uncertainty) leads to higher rewards for its coalition after solving a task. Thus, it is only fair that an expert member receives a higher reward than a member who has low expertise after solving

a task.

6.4.5 Implicit and Explicit Learning Actions

As alluded to in Assumption 5 (Section 6.1.1) the agents in ADLIT are able to utilize implicit and explicit learning actions to change their types. In our implementation, we have chosen the following set of implicit and explicit learning actions described in (Inaba et al., 2000). Table 6.1 shows the implicit and explicit learning actions and our notations.

Table 6.1: Chosen Implicit and Explicit Learning Actions

Learning Action	**Type of Learning**	**Denoted by**
None	No Learning	ic^0
Observation	Learning by Observation	ic^1
Discussion	Learning by Communication	sc^1
Teaching	Learning by being Taught	sc^2
Apprenticeship	Learning by Apprenticeship	sc^3

6.4.6 Impact and Cost of Learning

Educational psychology researchers (e.g., (Hoppe et al., 2003)) discuss that the impact of a learning action on the expertise of the participating agents may vary depending on a variety of factors in the problem domain. This implies that, for a given environment, some learning actions could be more effective than others. So, in our current implementation of ADLIT (following Assumption 2.1 in Section 8), we have used the following *order* of effectiveness of the learning actions:

$$ic^1 < sc^1 < sc^2 < sc^3 \tag{6.25}$$

Furthermore, Equation 6.25 also describes the order of the learning cost incurred by the agents who participate in these learning actions. So, the cost of engaging in a learning

action is proportional to its effectiveness in changing the type of the participating agent(s). In other words, the more effective a learning action is in terms of changing the type of the agent, the higher the cost incurred by the agent who uses that learning action. If we use $lo(sc^l)$ to denote the order of effectiveness of the learning action sc^l, the cost of the learning action sc^l is:

$$\zeta(sc^l) \propto lo(sc^l) \tag{6.26}$$

6.4.7 Type Change Dynamics

As described in the Assumption 5 (Section 6.1.1), the type change of an agent i due to its implicit and explicit learning actions with agent j is calculated using the following λ_2 function which is unknown to the agents:

$$\lambda_2(\theta_i, \theta_j, ic^l_{i,j}, sc^l_{j,i}) \propto \begin{cases} (\beta_i - \beta_j) + lo(ic^l) + lo(sc^l) & \text{If } lo(0,1) < \aleph_2 \\ (\beta_i - \beta_j) + lo(ic^l) + lo(sc^l) \pm \epsilon_\theta & \text{Otherwise} \end{cases} \tag{6.27}$$

In Equation 6.27, $\aleph_2 \in [0, 1]$ is the learning probability uncertainty threshold. In Equation 6.27, we also assume that the randomly chosen $\epsilon_\theta \in [\beta_{min}, \beta_{max}]$. Notice that, as described in the Coalition Formation Overview subsection above, we assume that the expertise of the agents, i.e., β_is, and consequently the difference in their expertise $(\beta_i - \beta_j)$s, are discretized and those values are in the range $[0, 3]$. Furthermore, we have assumed an order of effectiveness of the implicit and explicit learning actions which ranges from $[1, 4]$. So, according to our assumed discretization, the values of ϵ_θ will be $\epsilon_\theta \approx (\beta_j - \beta_i)$, $\epsilon_\theta \approx lo(ic^l)$, and $\epsilon_\theta \approx lo(sc^l)$.

According to Equation 6.27, the change in a learner's expertise is equal to the difference in the interacting agents' expertise plus the learning order factor when the probability

value drawn from the random normal distribution falls below the uncertainty threshold. Otherwise-if the probability value falls above the uncertainty threshold, the change in the learner's expertise is slightly perturbed from the actual value by the amount ϵ_θ.

This design of the learning function implies that, if the difference between the expertise values of the agents' is *high* and the order of the learning action is *high*, then the improvement in expertise will be *high*. Furthermore, if the learner agent's expertise is *higher* than the agent it is learning from, the learning function's outcome could be *negative* depending on the value of the learning order factor—i.e., it is possible to learn the wrong thing!. Notice that our consideration of the impact of the difference in the interacting agent's expertise on the learner's learning outcome is derived from the reported educational psychology and collaborative learning research (Stahl, 2004; Cress and Kimmerle, 2008) that suggest that the amount of expertise gain through successful collaborative interactions depends upon the difference in the expertise of the learners.

Equation 6.27 represents a key component of the uncertainty in the ADLIT environment. By choosing the different values of the threshold ϵ_θ in Equation 6.27, we will be able to change the agents' ability to use implicit and explicit learning for type change. For example, if the threshold value is set high (e.g., 0.9, or 90% of the time), the agents' implicit and explicit learning actions would be predictable allowing the agents to more easily learn the impacts of their choices of learning actions. That knowledge regarding that impact would allow the agents to choose the most cost-effective learning action leading to higher type change and higher utility (reward-cost) over time. On the other hand, if the threshold value is chosen to be very low (e.g., 0.1), the agents would observe different outcomes for the same chosen implicit or explicit learning action in a coalition. This would lower the agents' ability to effectively use the type change to improve their utility over time. Although the exact value of an environment parameter is not important in multiagent simulation environments (Bonabeau, 2002) choosing a very low value would mean that for the same input values of Equation 25, the agents would observe different outcomes. Thus,

those low values would make it difficult for the agents to learn the type change dynamics of the environment negatively affecting their performances. We discuss how the chosen uncertainty values impact the ADLIT agents' performances in our results (cf. Section 6.6.2).

6.4.8 Implementation in Repast Simulation Environment

To investigate the impact and effectiveness of our proposed strategies, we have implemented the ADLIT framework in the JAVA-version of the agent-based simulation environment Repast (Repast, 2009). Inside the simulation environment of Repast, we have developed 32 Java classes for the interconnected elements (e.g., agent, task, model, reward) of the ADLIT environment. The developed source code (size 500 KB), the executable program (size 18MB), and the documentation (in Javadoc format) are available in the author's website (`http://cse.unl.edu/~knobel/research`). Figure 6.1 shows the GUI of our simulation tool implemented in Repast.

The GUI of Repast is designed to run the simulation for a specified set of values of the parameters for debugging and testing purposes. To run long experiments that involve varying the values of multiple parameters of the ADLIT environment, we have used a parameter file and run Repast in the batch simulation mode on the Prairiefire cluster (`http://hcc.unl.edu`). Prairiefire is a 90 Node Production-mode LINUX cluster where each node contains (1) 2 Opteron 248, (2) 2 Opteron 275, and (3) 8 Opteron 870 processors and ($2.2GHz$/64 bit) per node. Each node provides a minimum of 4 GB $PC2700$ of memory per node. The Prairifire cluster is connected to the internet with an Infiniband Myrinet (2 Gb/s) Gigabit Ethernet. Furthermore, the Prairifire cluster contains (1) 1 TB SCSI RAID (XFS over NFS) and (2) 6 TB SATA RAID (ReiserFS over NFS) storage system providing 20 GB storage space per node. Prairifire provides 8 nodes per user at one time allowing us to run 8 experiments simultaneously. Since our simulation involves reasoning and interaction of a large number of agents over a large parameter space (e.g., the total number of possible combinations of varying number of agents, tasks, uncertainties,
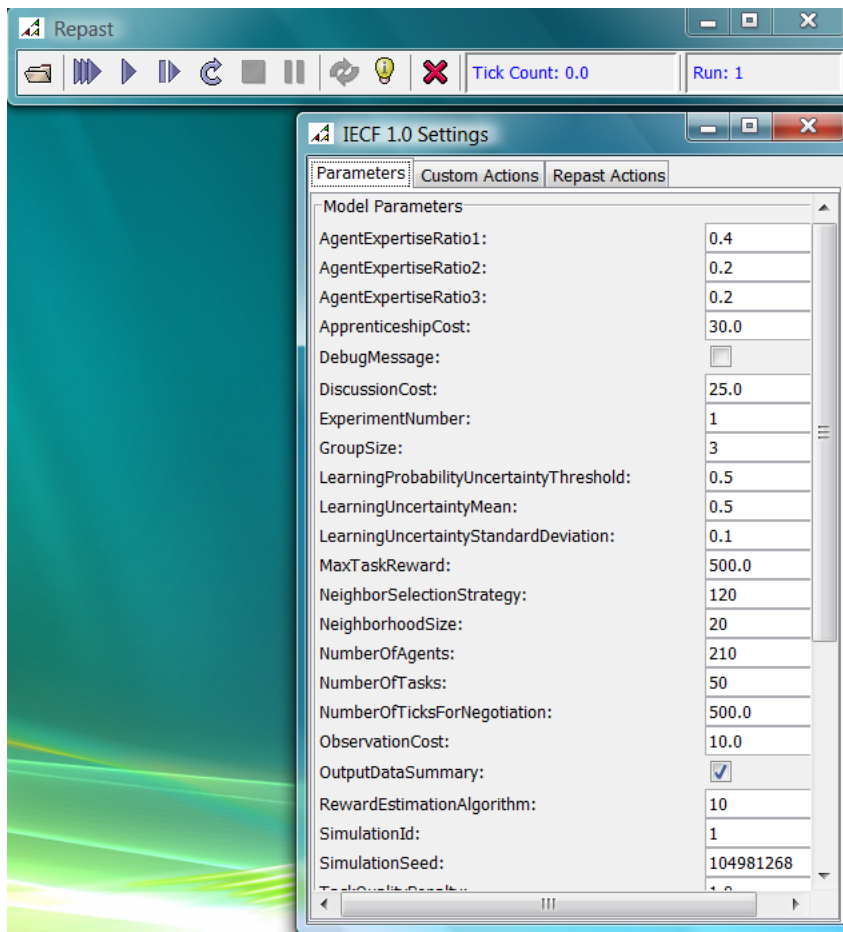
Figure 6.1: Repast GUI for ADLIT

etc.), the computational power and the ability to run multiple simultaneous experiments in Prairifire have provided us the opportunity to complete our experiments and collect results in a relatively short amount of time. For our experiments to be discussed in the next section, we regularly used 8 nodes at a time, and each run lasted about 96 hours.

6.4.9 Openness of the Environment

In ADLIT environment, we simulate the task openness by (1) generating a pool of potential tasks by creating tasks with different expertise requirements, (2) replicating the set of potential task list to meet the total number of tasks in the environment, and (3) randomly choosing the sequence of tasks are available for the agent coalitions to solve. Furthermore,

we simulate agent openness by randomly selecting and replacing one or more agents during the coalition formation round. Each selected agent is replaced with a new agent with randomly selected expertise for its capabilities. Furthermore, the ratio of agents that are replaced by newcomers is equal to the agent openness of the environment.

6.5   Experiment Setup

We have run a set of simulation experiments in the Repast environment to understand the impact of the implicit and explicit learning interactions in the environment and the performance of our approximation strategies. To be specific, using the default values shown in Table 2, we have run simulations in our Repast environment and collected detailed data on the performances of the agents, their coalitions, and their type changes. To reduce the randomness in our collected results, we have *replicated* all experiments with the same set of 10 different simulation seeds. Table 6.2 summarizes the data collected in the simulation runs of our experiment. This data is then analyzed and presented in our Results (Section 6.6).

Table 6.2: Default Values of the Simulation Parameters used in Experiments

Parameter	Default Values		
Number of Agents $|\mathcal{N}|$ (Section 6.1.2)	60		
Number of Tasks $|\mathcal{T}|$ (Section 6.1.2)	200		
Neighborhood Size $|\mathcal{N}_{\setminus}|$(Algorithm 6.1)	9		
Reward Estimation Strategies	MCT, OSL, LEA		
Randomly Chosen Simulation Seeds for Replication	352495638, 296241537, 734806509, 893883484	133653767, 35543328, 547081818,	515637740, 791347241, 376005653,
Agent Expertise Ratio Array (Assumption 6 in Section 6.1.1)	$\{0.6, 0.1, 0.1, 0.1, 0.1\}$		
Maximum Task Reward (Assumption 1 in Section 6.1.1)	300.0		
Length of Coalition Formation Protocol Round (Section 6.2)	360		

Parameter	Default Values
Task Solution Quality Uncertainty Threshold ($\aleph_1$ in Equation 6.23)	0.7
Task Solution Quality Variation Constant (ϵ_T in Equation 6.23)	1
Task Openness (Assumption 13 Section 6.1.1)	0.1
Task Difficulty (Assumption 13 Section 6.1.1)	1
Agent Openness (Assumption 14 Section 6.1.1)	0.1
Type Change Uncertainty Threshold ($\aleph_2$ Equation 6.27)	0.7
Task Solution Quality Variation Constant (ϵ_T Equation 6.27)	1
No Learning Cost Constant (Assumption 7 Section 6.1.1)	0.0
Observation Cost Constant (Assumption 7 Section 6.1.1)	1.0
Teaching Cost Constant (Assumption 7 Section 6.1.1)	2.0
Discussion Cost Constant (Assumption 7 Section 6.1.1)	2.5
Apprenticeship Cost Constant (Assumption 7 Section 6.1.1)	3.0
Learning Effectiveness and Cost Order (Equation 6.25)	$ic^1 = 1, sc^1 = 2, sc^2 = 3, sc^3 = 4$
Learning Cost Proportionality Constant (Equation 6.26)	10
Learning Variation Constant (ϵ_θ Equation 6.27)	1

Table 6.3: Simulation Outcome Data Collected After Running Our Experiment.

Tracked Variable	Description
totalUtility (Equation 6.4)	Total Reward Minus the Total Learning Cost of the Agents
totalLearningCost (Equation 6.4)	Total Learning Cost Paid by the Agents
totalTaskSolutionQuality (Equation 6.4)	Total Task Solution Quality Achieved by the Agents

Tracked Variable	Description
NONE_LEARNING_COUNT, TEACH-ING_LEARNING_COUNT, APPREN-TICESHIP_LEARNING_COUNT, OBSERVATION_LEARNING_COUNT, DISCUSSION_LEARNING_COUNT	Total learning action counts
PROGRAMMER_BAD, PRO-GRAMMER_AVERAGE, PRO-GRAMMER_GOOD, PROGRAM-MER_EXPERT	Number of Agents with different exper-tise values for $programmer$ capability
INTERFACE_DESIGNER_BAD, IN-TERFACE_DESIGNER_AVERAGE, INTERFACE_DESIGNER_GOOD, INTERFACE_DESIGNER_EXPERT	Number of Agents with different exper-tise values for $interfacedesigner$ capa-bility
SYSTEMS_ENGINEER_BAD, SYS-TEMS_ENGINEER_AVERAGE, SYS-TEMS_ENGINEER_GOOD, SYS-TEMS_ENGINEER_EXPERT	Number of Agents with different exper-tise values for $systemsengineer$ capabil-ity

To understand how the ADLIT agents are learning the type change and task open-ness dynamics of the environment, we also record the probability values observed (Sec-tion 6.3.4) by a *randomly* selected agent. Since all agents use the *same* frequentist learn-ing, this randomly selected agent's learned probability values represent how the other agents are learning task solution dynamics (Equation 6.23) and the type change dynam-ics (Equation 6.27) in ADLIT. Figure 6.2 shows a snapshot of how the selected agent's estimation of the probability of encountering the tasks changes over time. Figure 6.2 uses three visual dimensions to depict the selected agent's estimation of the task open-ness probability values. The x-axis represents the different tasks in the ADLIT environ-ment, the y-axis represents the coalition formation rounds in which the agent encoun-ters and solves a variety of tasks, and the darkness of the cell represents the probability value where a darker cell indicates a higher probability value. The marked area in Fig-ure 6.2 shows the selected agent's estimation of the probability of encountering a task $(Task = \langle Interface\ Designer = 5, Programmer = 3, Software\ Engineer = 8 \rangle$ in

the next round of coalition formation. As an agent encounters the randomly distributed tasks, it updates the probability of encountering that task again in the next round. So, the change in the darkness of the cells over time shows how our chosen agent captures the change in the probability of encountering a particular task. Later in Section 6.6 we will investigate how the agents' frequentist learning impacts their performances in terms of their earned utility.
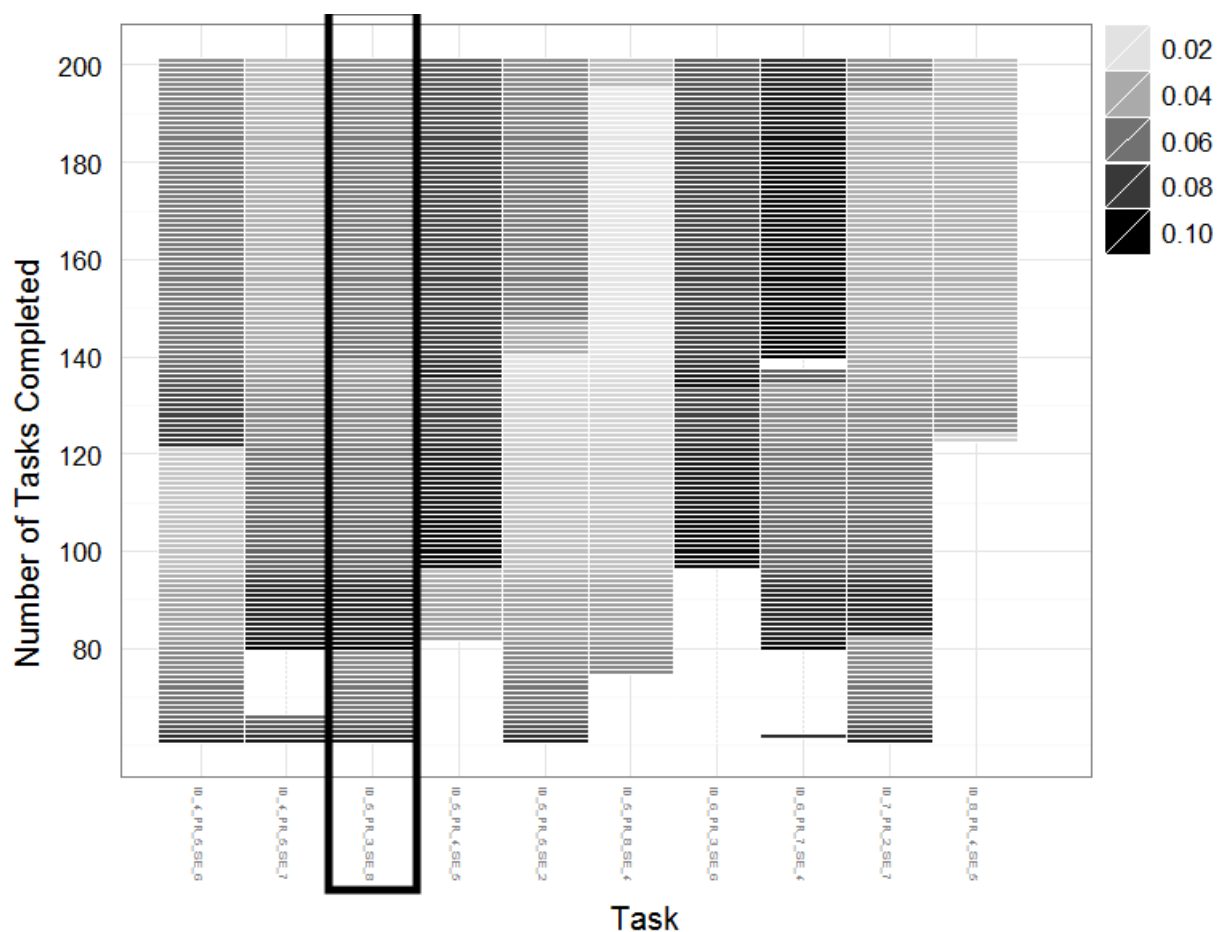

Figure 6.2: Change in the Probability of Encountering a Task as Learned by a Randomly Selected Agent.

6.6 Results

In this section, we discuss our results of our experiments that are designed to investigate the performances of our teaching and learning-based coalition formation protocol and the approximation strategies in terms of total utility, learning cost, and the type changes achieved by the agents. In our results, we compare the performance of the agent coalitions formed using implicit and explicit learning based approximation strategies (OSL and LEA in Section 6.3) against the coalitions formed using two variations of the MCT strategy (Section 6.3) — i.e., a strategy that aims to maximize the current-task reward of the agent coalitions without taking advantage of the learning-induced type-change and the current-task vs. future-task reward tradeoff. In the first variation MCT-TSL, the agents use a frequentist learning approach to *learn* the task solution quality dynamics (Section 6.1.2), i.e., the agents learn to estimate the function λ_1 (Equation 6.23). In the second variation MCT+TSL, the agents *know* the task solution quality probability function λ_1 in the environment. Notice that the agents using OSL and LEA strategies learn the probability of (1) achieving a task solution quality for given types of coalition members and (2) achieving a type change for their chosen implicit and explicit learning actions while participating in the coalition formation protocol. So, the consideration of these two variations of the MCT strategy, i.e., strategies that contain agents who learn the task solution quality probability and who do not, will allow us to isolate and better understand the impact of the OSL and LEA agents' strategy of type change through implicit and explicit learning actions. For organizing our findings, we have divided our discussions of the results into the following subsections: Feasibility and Impact (Section 6.6.1), Robustness (Section 6.6.2), Scalability (Section 6.6.3), and Summary (Section 6.6.4):

- **Feasibility and Impact** — The first objective of this subsection is to validate the feasibility of exploiting the current task vs. future task reward tradeoff arising from the type change of the agents in improving the utility of the agents in ADLIT. Our second

objective is to compare and contrast the utilities of the agents using the MCT+TSL, MCT-TSL, OSL, or LEA strategies to understand which of these strategies allow the agents achieve the best performance (in terms of utility) and why. Finally, our last objective is to identify whether or how our design of the strategies of the agents lead to any emergent behavior of the agents.

- **Robustness** — Although our approximation strategies are designed to take advantage of the type-changing ability of the agents' implicit and explicit learning actions, there are possible environmental aspects that may negatively impact the effectiveness of those strategies. The objective of our investigations in this subsection is to stress test the performance of our coalition formation protocol and the approximation strategies OSL and LEA in the midst of a variety of detrimental environmental factors, i.e., the (1) task and agent openness (Section 6.1.2), (2) the uncertainty of (a) task solution quality outcome for a chosen set of coalition members (Equation 6.23) and (b) type change outcome for a chosen learner and teacher agents and implicit and explicit learning actions (Equation 6.27), and (3) the learning cost (Equation 6.26).

- **Scalability** — Researchers define *scalability* of multiagent systems as the overall effect of processing times on performance (Rana and Stout, 2000) when the size and scale of the system increases. In this subsection, we aim to establish that, the performance improvement (in terms of the utility) that can be gained by using ADLIT's coalition formation protocol and the approximation strategies are independent of, to some extent, the amount of processing time spent by the agents. Closer inspection of the coalition formation protocol and the approximation strategies in ADLIT reveals that, the *length* of the coalition formation round (Section 6.2) and the *size* of the neighborhood ($|\mathcal{N}_{ne}|$ Algorithm 6.1) *together* determine the total computational time and resources spent by the agents. So, for our empirical scalability investigation, we first investigate the impact of the length of the coalition formation round (i.e.,

the time spent by the agents forming or joining coalitions and choosing implicit and explicit learning actions) on the earned utility of the agents using the OSL and LEA strategies. Then, we investigate the impact of the size of the neighborhood on (1) the total utility earned and (2) the time required for reasoning and coalition forming by the agents using the OSL and LEA strategies.

- **Summary** — Here we summarize our findings discussed in the feasibility, robustness, and scalability subsections.

6.6.1   Feasibility and Impact

Here we have run the simulation experiment with the default values described in Table 6.2 and collected the results for comparing the utility and type change achieved by the agents using the different approximation strategies. Figure 6.3 shows the utility earned by the agents in the ADLIT environment over all tasks for varying approximation strategies. As described in Equation 6.24, the total utility earned by an agent after solving a set of tasks in ADLIT is the total individual reward minus the cost of the implicit and explicit learning actions of that agent. Furthermore, Figure 6.4 shows the total task solution quality achieved by the agents in the ADLIT environment. Figures 6.3 and 6.4 show that the agents using the MCT+TSL strategy are able to earn higher total utility and higher task solution quality and higher utility than the agents using the MCT-TSL algorithm over time. This is due to the fact that the MCT+TSL agents know the task solution quality dynamics (Section 6.1.2) whereas the MCT-TSL agents have to learn over time to estimate that. So, initially when the MCT-TSL agents join coalitions, they do not often choose the set of agents (i.e., in the protocol shown in Algorithm 6.1 and 6.2) that would yield the maximal reward because of: (1) lack of information (e.g., when an agent is new) and (2) inaccurate estimation of the task solution quality dynamics (Equation 6.23) due to uncertainty. Such bad choices then negatively impact the solution quality of their solved task reducing their total reward earned by solving the specified set of tasks.
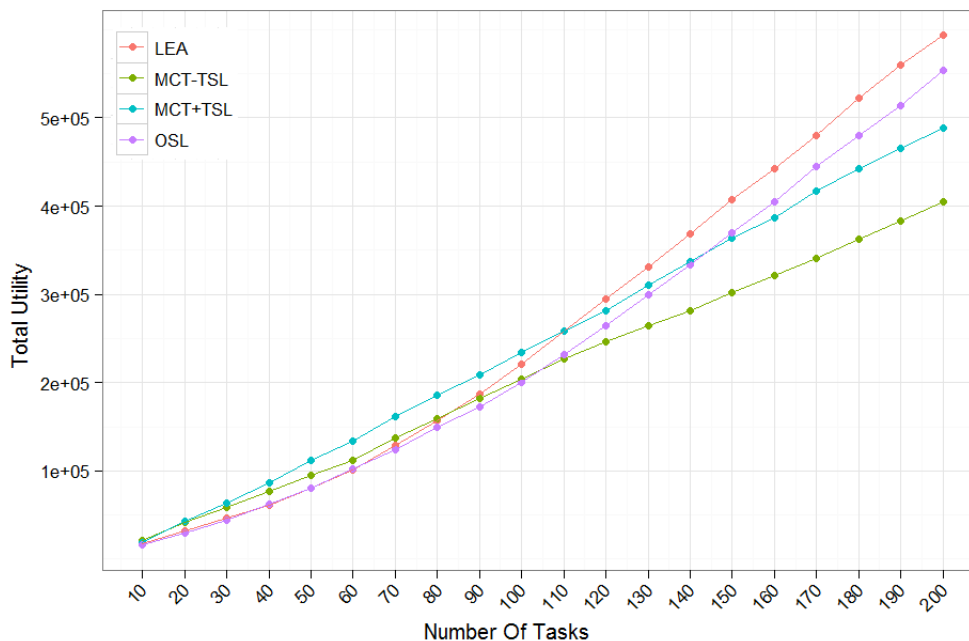
Figure 6.3: Total Utility Earned by the Agents using the MCT, OSL, and LEA Approximation Strategies over All Tasks.

Figures 6.4 and 6.5 also show that, over time, the agents that use the OSL and LEA strategies are able to earn statistically significantly higher utility compared to the MCT+TSL and MCT-TSL agents, which are our baseline agents. To be specific, when number of tasks > 131, (Figure 6.3), the total utility earned by the agents using OSL and LEA strategies are statistically significantly higher ($p < 0.05$) than the agents that use the MCT-TSL and MCT+TSL strategies. Figure 6.5 shows that, initially (number Of Tasks < 131), the OSL and LEA agents achieve lower task solution quality than the MCT+TSL and MCT-TSL agents. However, over time (number Of Tasks > 131), the OSL and LEA agents are able to solve the tasks with higher solution quality which lead to their higher utility. This ability of the OSL and LEA agents being able to solve the tasks better over time can be explained by their type change through implicit and explicit learning interactions. In the ADLIT environment, the agents' ability to solve a task better, i.e., achieve higher task solution quality, depend upon their types, or in other words, their expertise values for their capabilities. Figure 6.5 compares the weighted agent type values for the OSL, LEA, MCT+TSL, and
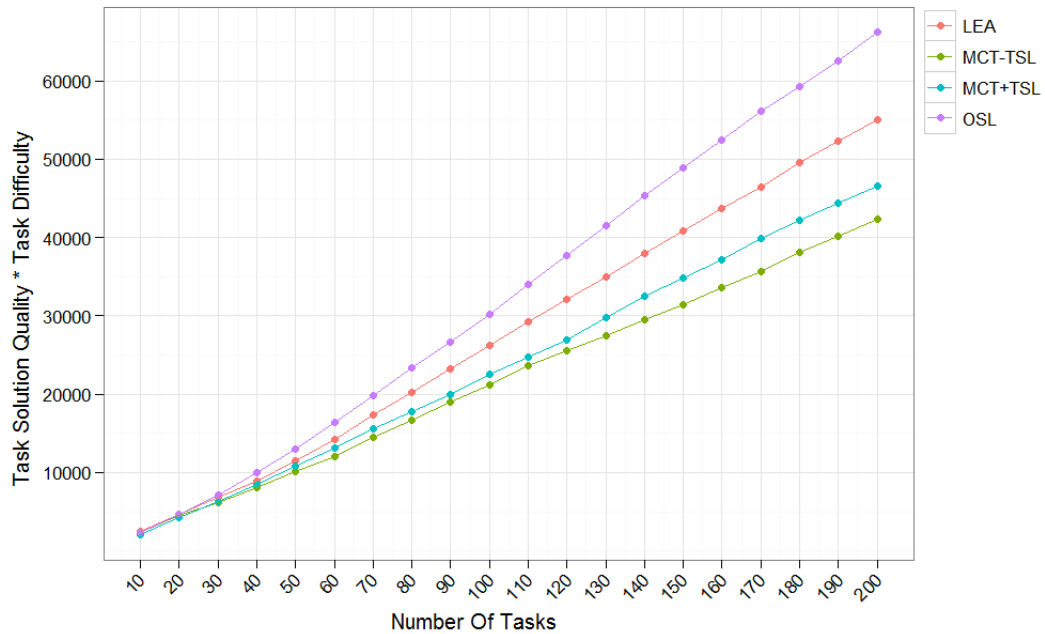
Figure 6.4: Total Weighted (by multiplying with Task Difficulty) Task Solution Quality Achieved by the Agents using the MCT, OSL, and LEA Approximation Strategies over All Tasks.

MCT-TSL strategies where the weighted agent type count is calculated for the (1) Interface Designer, (2) Programmer, and (3) System Engineer capabilities of agents using the Equation 6.28.

$$
\begin{aligned}
weighted\ agent\ type\ count \ =\ & 4 \times number\ of\ expert\ agents \\
+\ & 3 \times number\ of\ good\ agents \\
+\ & 2 \times number\ of\ average\ agents \\
+\ & 1 \times number\ of\ bad\ agents \quad (6.28)
\end{aligned}
$$

According to Equation 6.28, a group of agents who have *high* expertise values would have a *higher* agent type count for a chosen capability than than a group of agents who have *low* expertise values. Figure 6.5 shows that over time (i.e., when number of tasks

< 70 for LEA agents and number of tasks > 100 for OSL), the OSL and LEA agents can achieve statistically significantly higher ($p < 0.05$) weighted agent type counts compared to the MCT+TSL and MCT-TSL agents through implicit and explicit learning actions (Figure 6.6). That improvement of expertise then allow the agents to solve the assigned tasks better (e.g., achieve the higher task solution quality in Figure 6.4) and earn higher utilities over time (Figure 6.3). Since, the total utility of the agents is their reward minus their learning cost, the results here also indicate that the OSL and LEA agents' use of current task and future task reward tradeoff allow them to increase their earned reward enough to offset the incurred learning cost and obtain statistically significantly higher ($p < 0.05$) utility. Note that there is an observation of interest in Figure 6.5: from 0 task to 10 tasks, both OSL and LEA agents' performances *drop*. This is because initially the OSL and LEA agents do not have any knowledge about the impacts of their chosen implicit and explicit learning actions. As the agents learn from or teach other agents during coalition formation rounds, they update their probability using their observations to learn that impact. Furthermore, it is possible for the agents to actually degrade their own types by choosing the wrong teacher and learning action (cf. Equation 6.27). So, during the initial rounds of coalition formations, the OSL and LEA agents explore the various implicit and explicit learning actions and suffer the negative consequences (in terms or type degradation) of choosing the wrong learning actions.

Figure 6.6 shows the total number of learning actions used by the OSL and LEA agents over all tasks. We observe that in every round of coalition formation a *small* number of agents engage in some implicit and explicit learning actions. This is interesting since, due to the agents' consideration of the reward tradeoff, we might expect all OSL and LEA agents to start learning in the initial coalition formation rounds and then reduce their learning actions over time. Looking into this more closely, we realize that this pattern of OSL and LEA agents' participation in the learning actions is due to the dyadic nature of the explicit agent learning and the characteristic of the ADLIT environment. First, as discussed
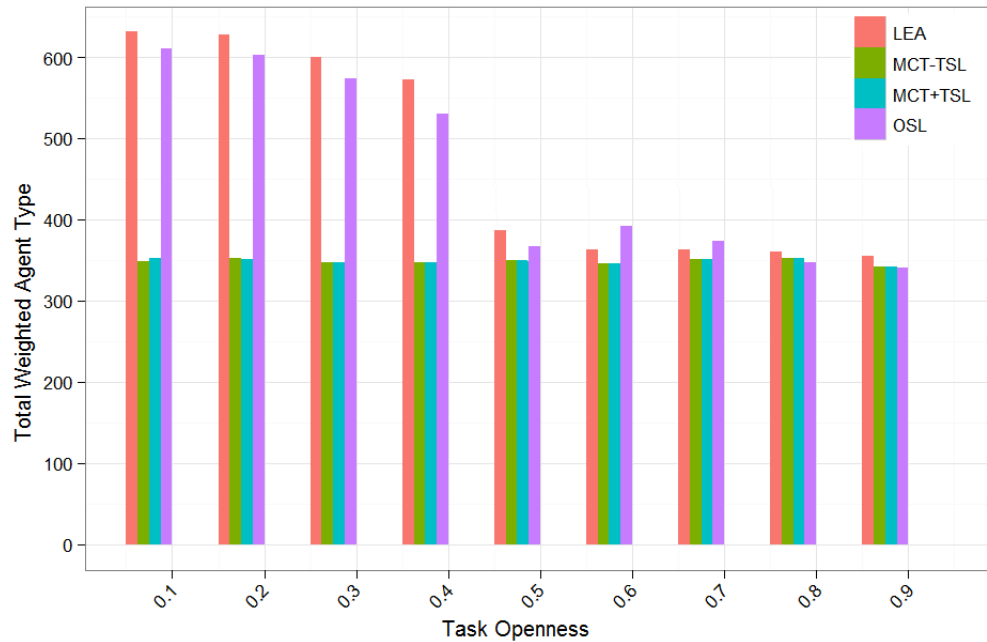
Figure 6.5: Total (Interface Designer+Programmer+Software Engineer) Weighted Agent Type Count of the Agents using the MCT, OSL, and LEA Approximation Strategies over All Tasks.

in Section 6.4.5, except Learning by Observation, all other learning actions require the participation of *two* agents in the ADLIT environment. However, the agents in the ADLIT environment have *differing* beliefs regarding the impacts of the learning actions on the types of the agents (Equation 6.3.4) due to (1) the learning uncertainty (Assumption 6 Section 6.1.1) and (2) partial observability (Assumption 15 Section 6.1.1). As a result, when joining or forming a coalition in a given round (Algorithm 6.1), not all agents are able to find suitable teachers or learners that they would like to teach to or learn from. Second, the OSL and LEA agents have to explore the uncertain environment to learn (Equation 6.3.4) the impact of their learning actions. So, it is possible that during exploration, some of the agents may end up *degrading* their types *instead of improving* while participating in the implicit and explicit learning. Finally, as we have discussed in Section 6.4.5, not all learning actions yield the same amount of type improvement. It is possible that an agent may have to participate in multiple learning actions over the tasks to achieve the type improvement

it wants (cf. Figure 6.5). All of these three scenarios thus explain the pattern of implicit and explicit learning shown in Figure 6.6. Finally, the OSL and LEA agents' participation pattern in the learning actions (Figure 6.6) along with their achieved type improvement (Figure 6.5) indicate that despite the uncertain and partially observable environment, those agents are able to improve their own types and the types of their coalition members, thereby allowing them to achieve statistically significantly higher utility over time.
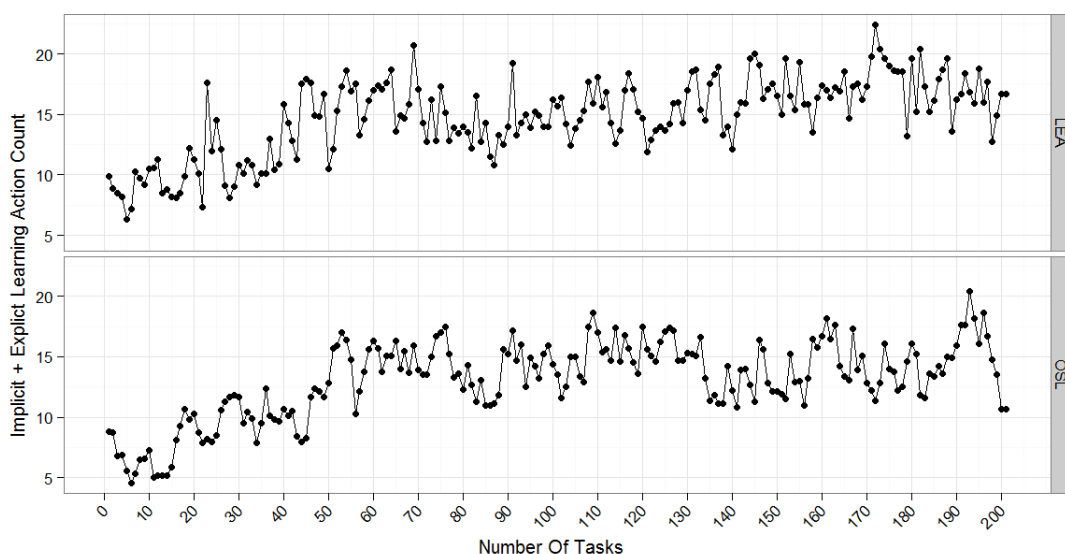
Figure 6.6: Total Number of Implicit and Explicit Learning Actions (TEACHING+OBSERVATION+DISCUSSION+APPRENTICESHIP-NONE) of the Agents over All Tasks.

If we compare the performances of the agents using the LEA and OSL strategies, Figures 6.3, 6.4, and 6.5 show that over time, the agents using the LEA strategy earn slightly lower task solution quality but achieve higher total utility. This result can be explained by the comparatively more efficient use of implicit and explicit learning actions of the LEA agents. Notice that, according to Equation 6.27, there are a large number of agent type combinations (e.g., an agent with BAD Expertise learning from an agent with Good expertise) and implicit and explicit learning actions that are possible in the ADLIT environment. However, as indicated in Equation 6.25, our implementation assumes there is an order of (1) the impact (in terms of how well it changes the types) and (2) the cost of the implicit

and explicit learning actions between a teacher and a learner. Given that the agents using the OSL and LEA strategies learn to estimate the λ_2 (Equation 6.27) function and that the agents' total utility is its total individual actual rewards after subtracting the learning cost, the agents who can learn the most efficient way of achieving the type change will achieve a higher utility. Figure 6.7 shows the differences in the total learning cost incurred by the agents using the LEA and OSL strategies. In Figure 6.7, we see that initially the LEA agents incur a higher learning cost but over time (number of tasks> 31), the LEA agents are able to learn more efficiently. This more efficient learning of the LEA agents is due to their exploration of the possible implicit and explicit learning actions. Since the agents using the OSL strategy (Equation 6.11) looks only *one* step ahead, focusing on improving their agent types for the tasks that are most likely to occur in the next step, which limits their exploration of the implicit and explicit learning actions. On the other hand, the LEA strategy (Equation 6.12) makes the agents *explore* their options (types of coalition members, implicit and explicit learning actions, etc.) more. For example, the OSL agents are more likely to accept a coalition proposal (Algorithm 6.1) if it means it can earn higher actual rewards for the next task, even if such acceptance would lead to less efficient implicit and explicit learning actions (i.e., provided by the coalition members). On the other hand, the energy and temperature terms in the LEA strategy (Equation 6.12) make the value of such a coalition low unless the coalition formation agent is equally confident about the impact of all available implicit and explicit learning actions. That means, even though the LEA agents may incur a *higher* cost initially (during exploration), *over time*, those agents will be able to choose more efficient implicit and explicit learning actions. That improved ability then eventually leads to significant reduction in the learning cost leading to higher utility despite the slightly lower task solution quality.
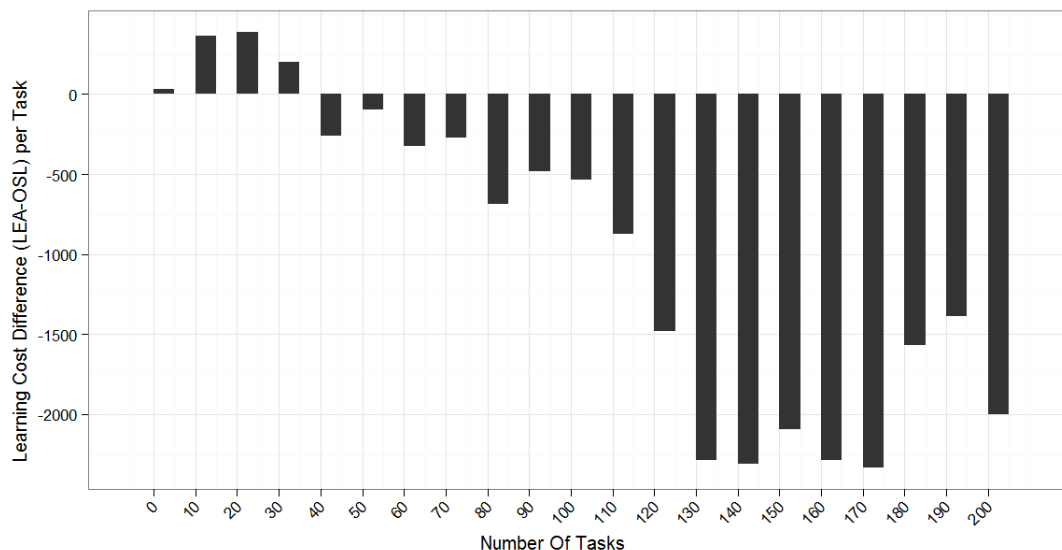
Figure 6.7: Difference in Total Learning Cost of the Agents Using LEA and OSL Approximation Strategies ($LEA - OSL$) over All Tasks.

6.6.2 Robustness

In this section, we investigate how the performances of the ADLIT agents are impacted by the task and agent openness, the learning cost, and the learning- and task solution quality- related uncertainty in the environment. Our goal here is to evaluate the robustness of our coalition formation protocol and the approximation strategies against these potentially detrimental environmental factors.

Task Openness

The task openness in the ADLIT environment denotes the percentage of new tasks encountered by the agents. Figure 6.8 shows the total utility earned by the agents over all tasks for varying task openness.

Figure 6.8 shows that when task openness is low (< 0.5), the agents using the OSL and LEA strategies are able to earn statistically significantly higher ($p < 0.05$) utility compared to the MCT+TSL and MCT-TSL agents. However, when the task openness is high, i.e., 0.5 or higher, the OSL and LEA agents cannot earn higher utility than the MCT+TSL
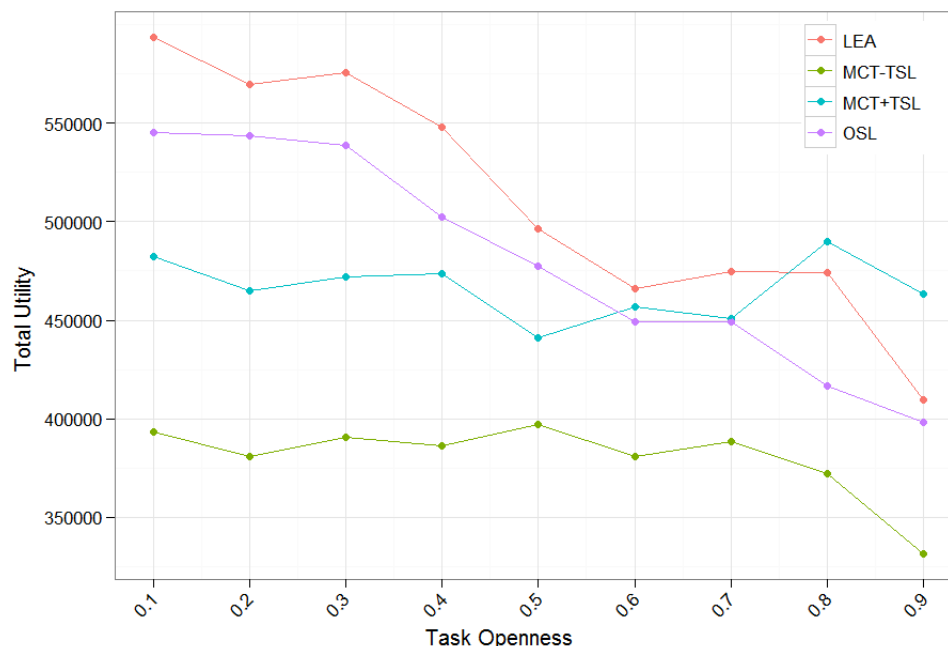
Figure 6.8: Total Utility Earned by the Agents Over All Tasks for Varying Task Openness.

agents. The task openness in ADLIT represents how often the agents encounter new tasks. Due to our design of the OSL and LEA agents' estimation values, if the task openness is high, the agents would reduce their learning actions. That is because, if the agents see that they have a very low probability to encounter a task more than once, the reward tradeoff value calculated in Equation 6.11, 6.12 would be low. In other words, they will realize that learning from other agents and/or training their coalition members would not improve their future task reward over time. Upon realizing that, the OSL and LEA agents will reduce their implicit and explicit learning actions. Further investigation into the agents' learning actions shows that (Figures Figure 6.9 and Figure 6.10) that, as the task openness is increased, the OSL and LEA agents reduce their implicit and explicit learning actions and are not able to improve their types (Figure 6.11). As a result, for high task openness, the OSL and LEA agents are not able to achieve higher task solution quality (task solution quality statistically not different $p < 0.05$) and their earned utility is similar to that of the MCT+TSL and MCT-TSL agents. The observations here show that ADLIT agents'

performance improvements achieved through type change be diminished for environments that have high task openness. This is expected since the current-task vs. future-task reward tradeoff is based on the idea that the agents improve their types only if they will be able to use that improvement to earn higher rewards in future.
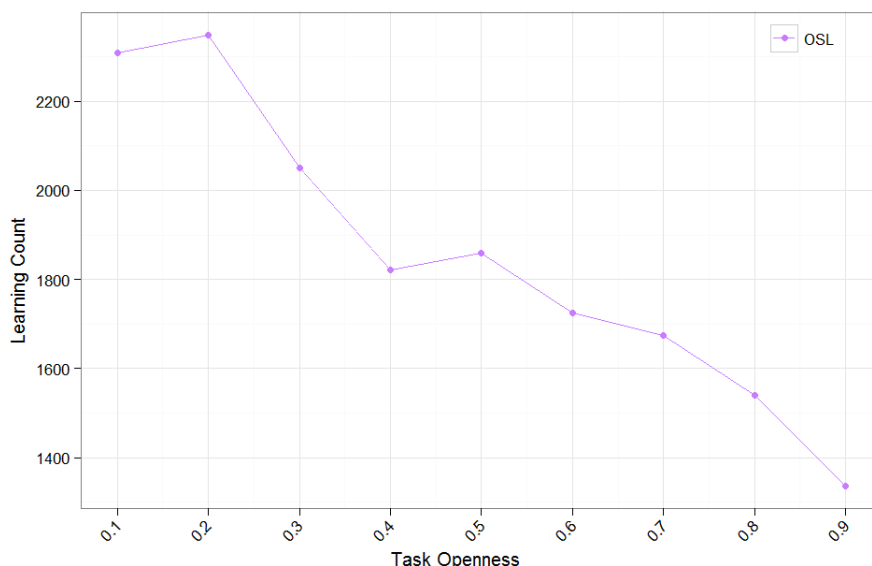

Figure 6.9: Total Implicit+Explicit Learning Action Count of OSL Agents for Varying Task Openness.

Agent Openness

Figure 6.12 shows the average total utility of the agents using the different strategies over all tasks for varying agent openness. Figure 6.12 shows that the agents using the OSL and LEA are able to earn statistically significantly higher ($p < 0.05$) utility compared to the agents using the MCT+TSL and MCT-TSL strategies. Furthermore, Figure 6.13 shows that the OSL and LEA agents are able to improve their types for all agent openness values, which explains their ability to solve the tasks better (Figure 6.14) and their higher utility. This result is unexpected since high agent openness means a large ratio of agents, who have improved their types and have learned the environment dynamics, are randomly replaced. When those trained agents are replaced by random agents (i.e., types are ran-
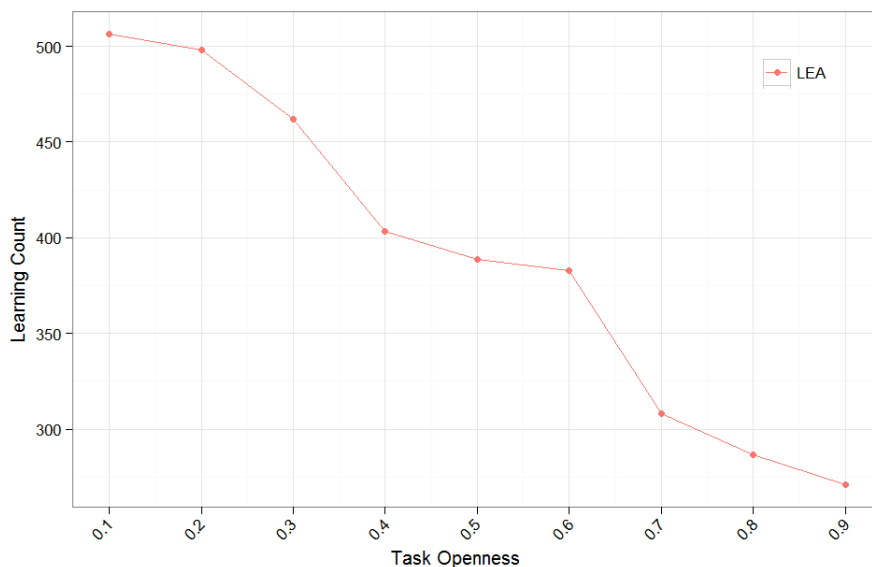
Figure 6.10: Total Implicit+Explicit Learning Action Count of LEA Agents for Varying Task Openness.

domly selected), it should negatively impact the overall performance of all OSL and LEA agents since the trained agents do not get to achieve the increased reward over time. Furthermore, the randomly selected agents now have to incur the cost of learning to improve their own types and have to learn the task solution and learning probabilities that enable them to improve their types over time. As a result, such random replacement is likely to lead to a diminished performance of the agents and one may expect the OSL and LEA agents to achieve a performance similar to the MCT+TSL and MCT-TSL agents who do not engage in type change. However, the performance and agent type improvements in Figure 6.12-6.13 of the OSL and LEA agents can be explained by the impact of the agents' local implicit and explicit learning actions on the globally emergent performance improvement of all agents in the environment (as discussed in Assumption 8 in Section 6.1.1).

Upon further investigation, we find the cause of the above unexpected observation. First, as discussed in Section 6.3, the OSL and LEA agents exploit the current task vs. future task reward tradeoff by improving their types through the use of learning actions. To implement agent openness, a randomly selected set of agents are replaced by random
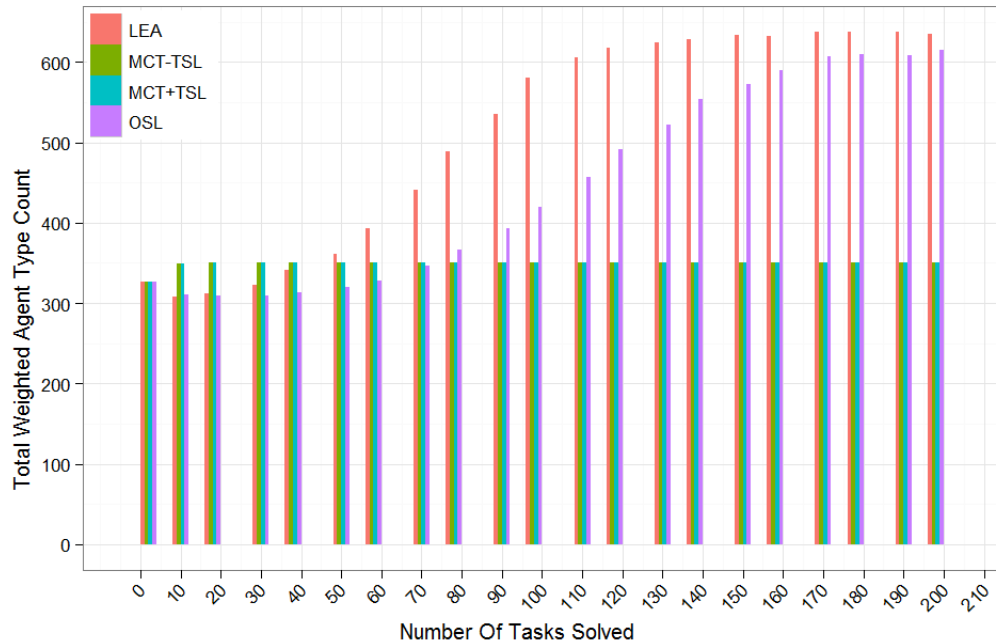
Figure 6.11: Total (Interface Designer+Programmer+Software Engineer) Weighted Agent Type Count of the Agents using the MCT, OSL, and LEA Approximation Strategies over All Tasks for Varying Task Openness.

agents over time. However, not all agents that have been selected to be replaced are replaced at the same time. When a set of agents are replaced, the remaining agents, who have improved their types and learned (frequentist learning) the task solution and learning probabilities can help the newly introduced agents improve their types. This is help is provided in two ways. First, due to our design of the coalition formation protocol and the dyadic nature of the agents' learning (Equation 6.4.6), the selection of a learning action of an agent requires the agreement of one of its coalition members. As a result, a newcomer agent that is forming coalitions with one or more experienced agents in its coalition formation proposal is likely to select the optimal learning action because those experienced members will not agree to a suboptimal action. In addition, the OSL and LEA agents' calculation of the value of a coalition allows the agents to take the probability of seeing the trained agents in future coalitions ($\Pr(\theta_{Cf})$ in Equation 6.11 and Equation 6.12) into account. This means, when a large number of agents are replaced in the environment, this probability
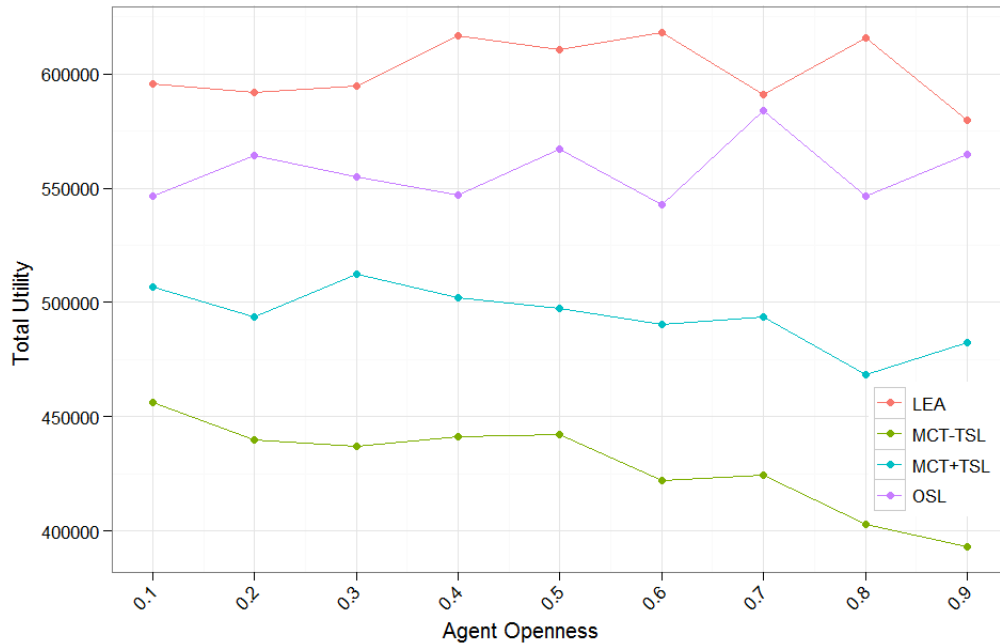
Figure 6.12: Total Utility of Agents for Varying Agent Openness.

value would be small and may lead to the agents reducing their implicit and explicit learning actions. However, the improved types of the agents in Figure 6.13 indicate that the OSL and LEA agents did not cease their learning actions for high agent openness. This effect can also be attributed to by the factor $[\Pr(\kappa|\theta_{Cf}) - \Pr(\kappa|\theta_C)]$ in Equation 6.11 and Equation 6.12 where that factor denotes the expected change in the task solution quality due to the coalition-forming agents' type change. When the agents already have improved types (i.e., the task solution quality improvement from type change is small), this factor's value would be small and the agents would *not* engage in implicit or explicit learning. On the other hand, if the coalition-forming agents' types are low, the value of this second factor would be high and compensate for the low probability of value $\Pr(\theta_{Cf})$. As a result, if the coalition-forming agents' types are low, the OSL and LEA agents would continue to improve their types by implicit and explicit learning actions. Then that type change would lead to statistically significantly ($p < 0.05$) higher utility over time. To summarize, due to the globally emergent type improvement of the OSL and LEA agents' local type-changing

decisions, the agents are still able to earn statistically significantly ($p < 0.05$) higher utility even when a large number of agents are replaced in the environment due to high agent openness.
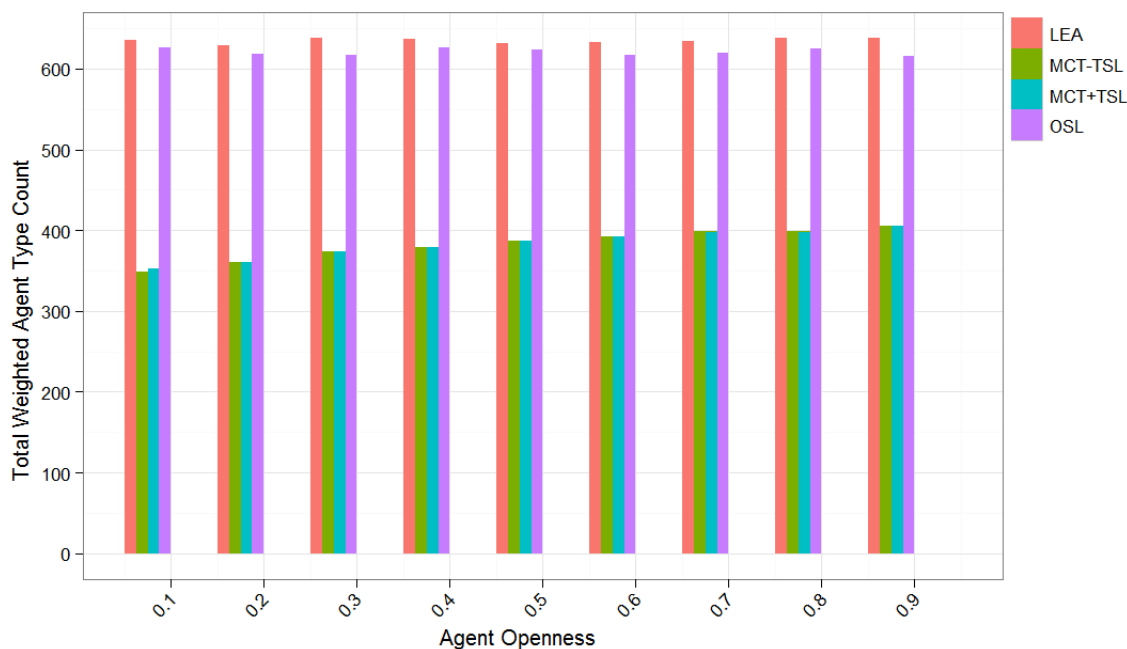

Figure 6.13: Type Change of Agents for Varying Agent Openness.

Learning Probability Uncertainty

Figure 6.15 shows the total utility of the agents using different approximation strategies over varying learning probability uncertainty. Furthermore, Figure 6.17 shows the agents' weighted agent type count over varying learning probability uncertainty.

Our results show that, even for high learning probability uncertainty values ($\aleph_2$ in Equation 6.27), the OSL and LEA agents are able to earn statistically significantly higher ($p < 0.05$) utility than the MCT+TSL and MCT-TSL agents. This is unexpected. As described in Equation 6.27, the learning probability uncertainty determines whether the output of the learning transition function is dependent upon the types of the agents and the learning action or whether output is generated randomly. The key difference between
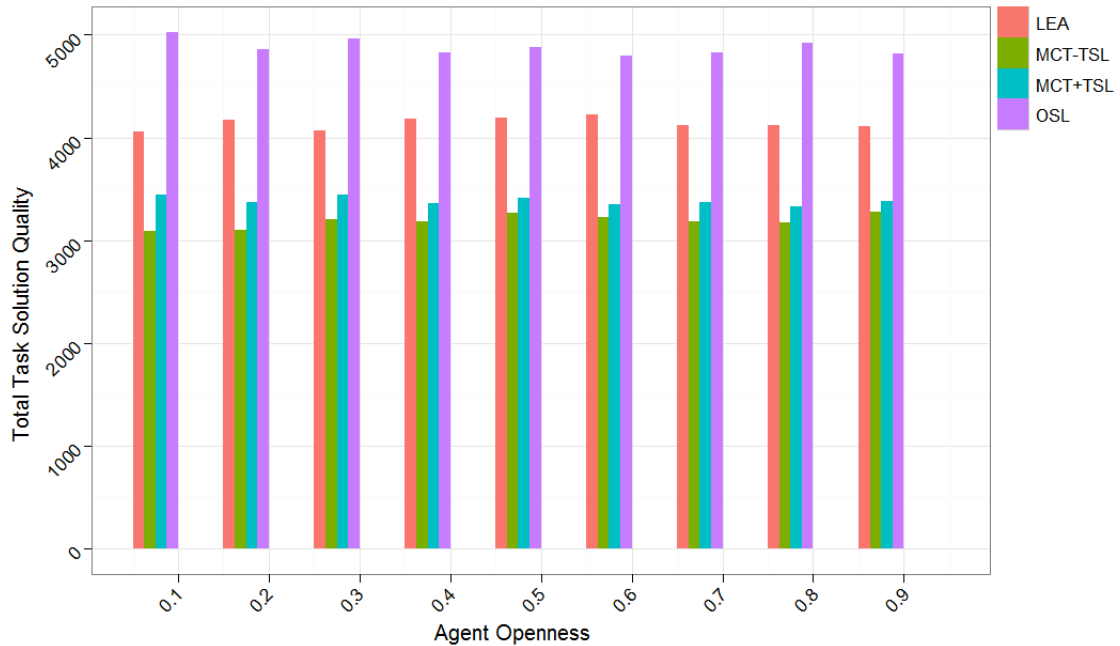
Figure 6.14: Total Task Solution Quality for All Tasks over Varying Agent Openness.

the OSL and LEA agents and the MCT agents (MCT+TSL, MCT-TSL) is that the former types of agents try to improve their earned utility values using implicit and explicit learning actions. For the OSL and LEA agents to be able to sufficiently accurately estimate the value of a coalition (Figure 6.1), they have to be able to estimate how a chosen learning action and the types of the learner and teacher agent are going to change its own type and the types of its coalition members. In our implementation, the agents' type change through implicit and explicit learning actions is determined by Equation 6.27. When the learning probability uncertainty ($\aleph_2$ in Equation 6.27) is high, the majority of the outcomes of the agents' learning actions are varied (*positively* or *negatively*) randomly with the learning variation constant (ϵ_θ).

This random variation would make it difficult for the agents to sufficiently accurately estimate the impacts of their learning actions. Without this estimation, the agents should not be able to improve their types since they will not be able to choose the appropriate learning action. By failing to improve their types, the OSL and the LEA agents would consequently

fail to improve their utility using the task vs. future task reward tradeoff. To understand how the OSL and LEA agents are able to achieve that type change, we further investigate the *type of learning* the OSL and LEA agents use. Figure 6.18 and Figure 6.19 show that over time, LEA and OSL agents learn to prefer the Discussion learning action and None learning action for high learning probability uncertainty. Note that the LEA agents switch to the Discussion and None learning more abruptly than the OSL agents. Closer analysis of our data reveals that the LEA agents are able to keep their utility gain high ($p < 0.05$) even in the face of higher learning probability uncertainty until the uncertainty reaches 0.8. Beyond the value of 0.8, a larger number of LEA agents (90% in LEA as opposed to 80% in OSL) switch to the Discussion and None learning actions. Whereas between $[0.7 - 0.9]$ learning probability uncertainty values, the OSL agents, who have limited exploration capabilities compared to LEA, gradually (30% of the agents over the learning probability uncertainty values $[0.7 - 0.9]$) increase their Discussion and None learning actions. Now, according to our chosen order of cost and effectiveness (Equation 6.25), the Discussion learning action has the lowest cost among all learning actions whose outcome (in terms expertise improvement) depend on the learner-teacher expertise difference (Equation 6.27). If an agent learns from a teacher agent who has higher expertise, even if the learning outcome is chosen randomly (when learning probability uncertainty is high), the overall learning outcome remains beneficial to the learner. That is because, in our implementation, our chosen Learning Variation Constant (ϵ_θ in Table 6.2) for random learning is 1 and the learning outcome is proportional to the expertise difference. So, a learner can improve its type by choosing a teacher whose expertise is higher (> 1) than its own even when the learning probability uncertainty in the environment is high. If the learner agents cannot find such a teacher (e.g., if the learner already has high expertise), they would not engage in learning. In that case, the number of agents not learning will increase which is also evident in Figure 6.18 and Figure 6.19. Our results show that, Learning Variation Constant (ϵ_θ in Table 6.2) is small, by finding and exploiting the underlying type change dynamics, the

OSL and LEA agents are able to improve their utility over time by overcoming the learning uncertainty in the environment.
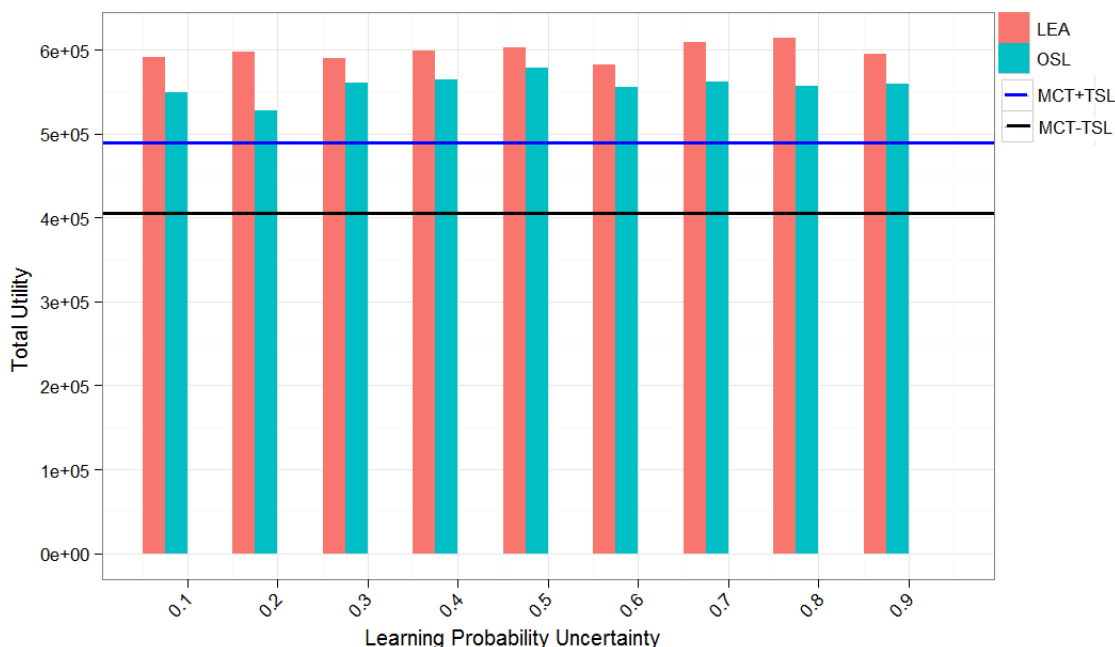

Figure 6.15: Total Utility of Agents for Varying Learning Probability Uncertainty Values.

Task Solution Quality Uncertainty

Figure 6.20 shows that as the task solution quality uncertainty is increased, the OSL, LEA, and the MCT-TSL agents' earned utility is reduced. We see that for medium or high task solution quality uncertainty values, i.e., if $\aleph_1 > 0.4$, the OSL, LEA, and the MCT-TSL agents achieve a lower utility than the baseline MCT+TSL agents. As described in Equation 6.23, the task solution quality uncertainty value $\aleph_1$ determines the percentage of tasks whose outcomes are determined at random. When the task solution quality uncertainty is determined randomly, the output of a task is and thus the reward earned by the agents is not dependent on the types (i.e., the expertise values of the relevant capabilities) of the coalition members. Due to such *random* task solution quality selections, the agents have difficulty learning the task solution quality dynamics (Equation 6.23), i.e., the mapping
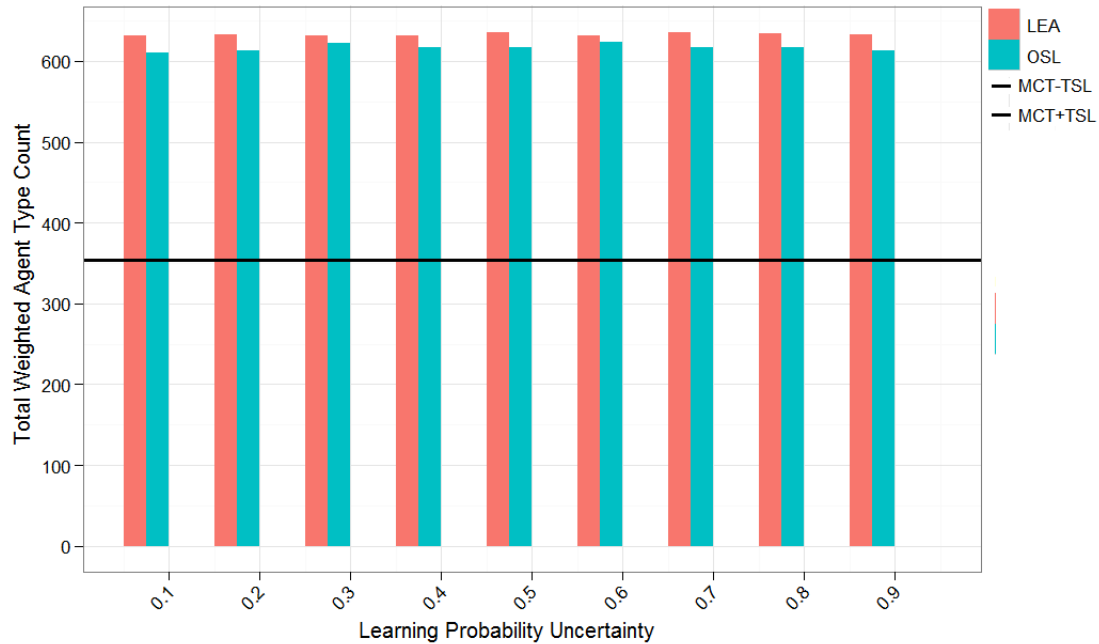
Figure 6.16: Total Weighted Agent Type Count (Interface Designer+Programmer+System Engineer) (Equation 6.28) for Varying Learning Probability Uncertainty Values.

of the coalition members' types to the task solution quality. As a result, when the agents determine the value of a coalition in the coalition formation round (Algorithm 6.1), the MCT-TSL (Equation 6.10), the OSL (Equation 6.11), and the LEA (Equation 6.12) agents *cannot* accurately calculate the value of a coalition in terms of the current-task reward it may provide. Consequently, those agents then choose to form coalitions that yield a low reward, e.g., choose agents with low expertise for a task that requires high expertise. This is evident for the OSL and LEA agents when we compare Figures 6.21-6.23. Figure 6.21 shows that the OSL and LEA agents are able to achieve statistically significantly ($p < 0.05$) higher types compared to the MCT-TSL agents and incur the cost of learning (Figure 6.22). However, Figure 6.21 shows that, despite having improved types, the OSL and LEA agents achieve statistically significantly ($p < 0.05$) lower task solution quality than the MCT-TSL agents for task solution quality uncertainty values > 0.4. This indicates that, for high task solution quality uncertainty values, the OSL and LEA agents improve their types, pay the
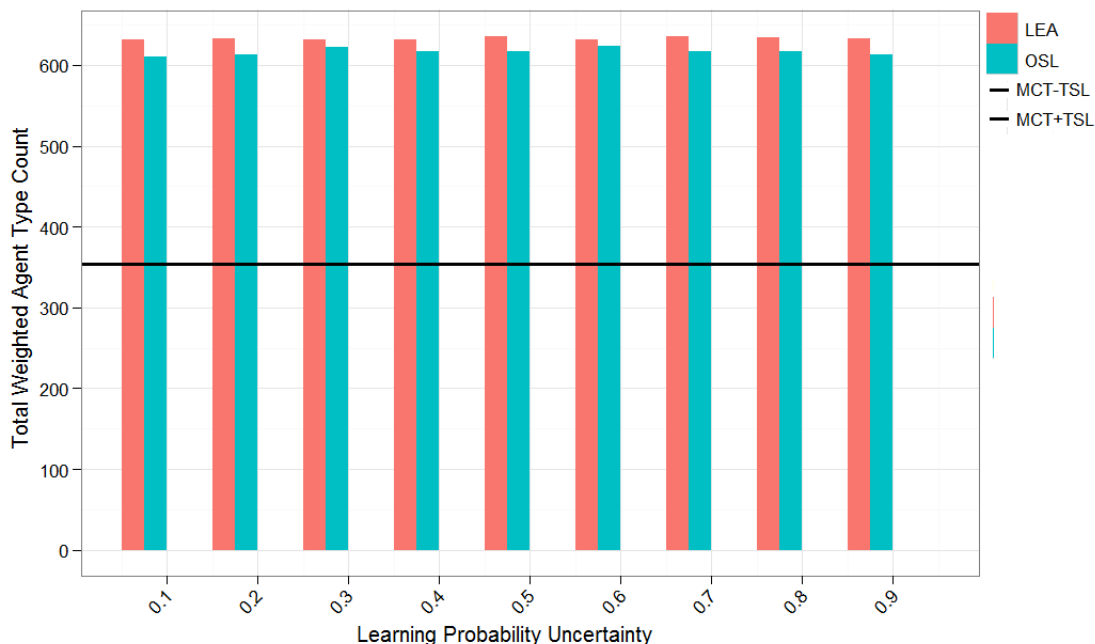
Figure 6.17: Total Weighted Agent Type Count (Interface Designer+Programmer+System Engineer) (Equation 6.28) for Varying Learning Probability Uncertainty Values.

cost of learning, but cannot get high rewards. Equation 6.23 shows that the task solution quality of a coalition in our implementation is determined by its members' types and the uncertainty of the environment. Since the OSL and LEA agents have improved types, their lower reward indicates that those agents could not choose coalitions that can provide good-enough rewards for the current task, i.e., those agents are not able to take advantage of the current-task vs. future-task reward tradeoff. This indicates a potential for improving the OSL and LEA agents' performances by improving those agents' learning of the task solution quality dynamics of the environment. If the OSL and LEA agents are able to more accurately estimate the task solution quality uncertainty value of the environment, they could be more careful about improving their types since such type changes are not going to lead to higher rewards in the future. We discuss how we are planning to improve the OSL and LEA agents' learning of the environment dynamics in our future works (Section 6.8).
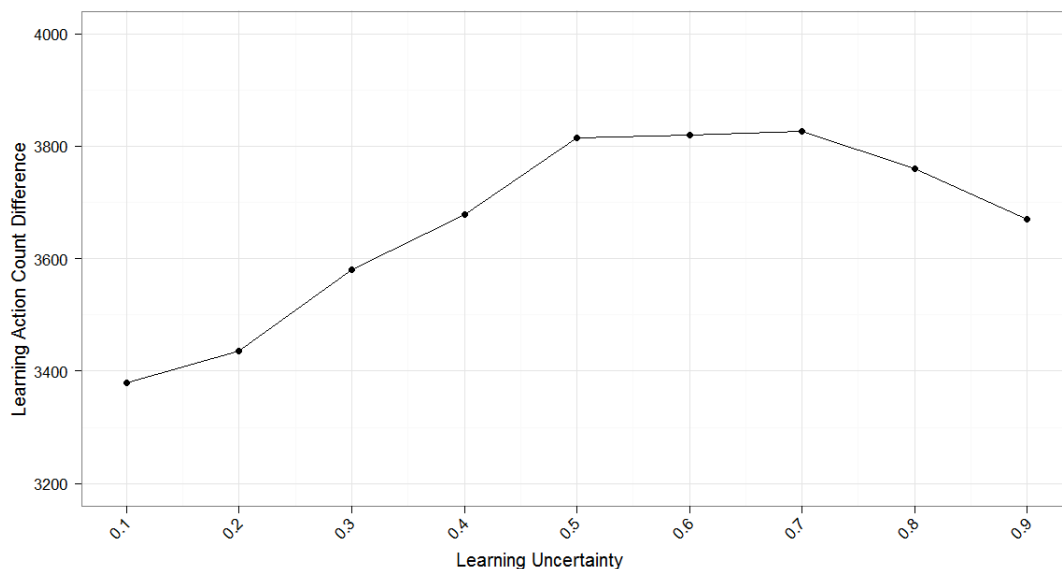
Figure 6.18: Learning Action Count Difference (Teaching+Apprenticeship+Observation-[Discussion+None]) of OSL Agents for Varying Learning Probability Uncertainty Values.

Learning Cost

Figure 6.24 shows the total utility earned by the agents over the tasks for varying learning cost proportionality constant (Equation 6.1). Here, by varying the learning cost constant (Equation 6.1), we increase the cost of learning proportionally maintaining the order of impact and cost presented in Equation 6.26. Figure 6.24 shows that for low learning costs (learning cost proportionality constant $[1, 15]$), the OSL and LEA agents are able to earn statistically significantly higher ($p < 0.05$) utility than the MCT+TSL and MCT-TSL agents. However, as the learning cost is increased, that utility advantage gained by the OSL and LEA agents diminishes and for learning costs proportionality constant 50, the LEA and OSL agents earn lower utility than the MCT+TSL and MCT-TSL agents. The OSL and LEA agents' failure to achieve higher utility can be explained from the point of view of their inability to optimize their type changes.

Figure 6.25 shows the agents' total weighted agent type count (Equation 6.28) for varying learning costs whereas Figure 6.26 shows the number of learning actions of the OSL and LEA agents for varying learning cost proportionality constants. Combining Figure 6.25
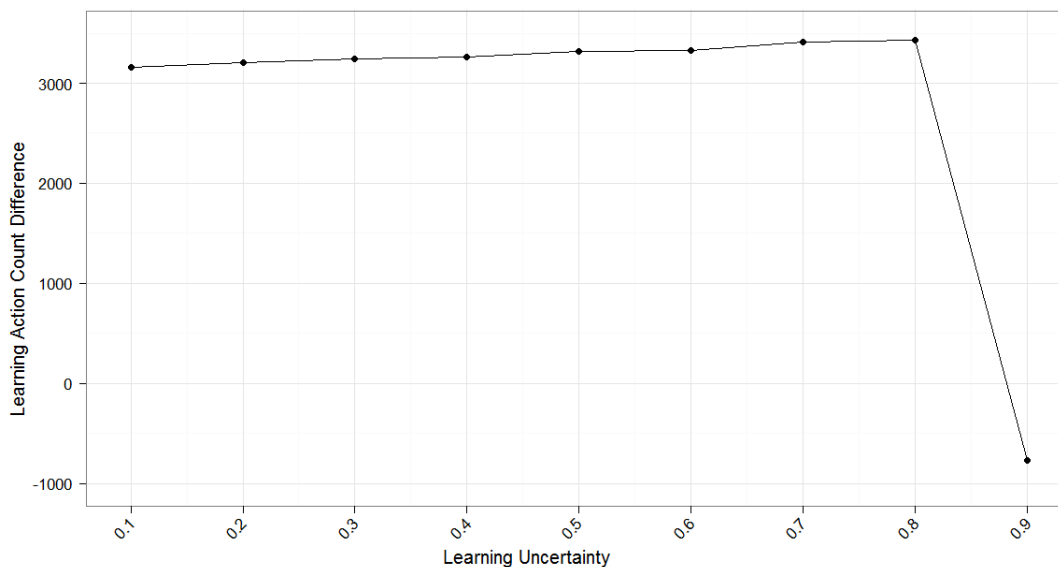
Figure 6.19: Learning Action Count Difference (Teaching+Apprenticeship+Observation-[Discussion+None]) of LEA Agents for Varying Learning Probability Uncertainty Values.

and Figure 6.26, we see that, as the cost of implicit and explicit learning increases, the OSL and LEA agents reduce their learning actions and consequently achieve lower types. This is expected, since the OSL and LEA agents' choice of the coalitions and their learning actions depend on their evaluation of the current task and future task reward tradeoff (Equation 6.11 and Equation 6.12). That is, if the learning cost is very high, the OSL and LEA agents see that the type improvement *does not justify* the high learning cost they are incurring. Consequently, those agents reduce their use of learning actions and achieve lower types. That reduced type then prevents the OSL and LEA agents from achieving higher utility than the MCT agents. Notice that the OSL and LEA agents' ability to reduce their learning actions in the face of high learning cost indicates that these agents can adapt to the high learning cost of the environment. However, without any learning actions, one might expect that the OSL and LEA agents would achieve utility values similar to the MCT agents. The fact that the OSL and LEA agents earn a somewhat lower utility (Figure 6.24) can be attributed to the agents' *exploration* behavior in the ADLIT environment. Since the OSL and LEA agents (especially the newcomer agents introduced due to agent openness) have
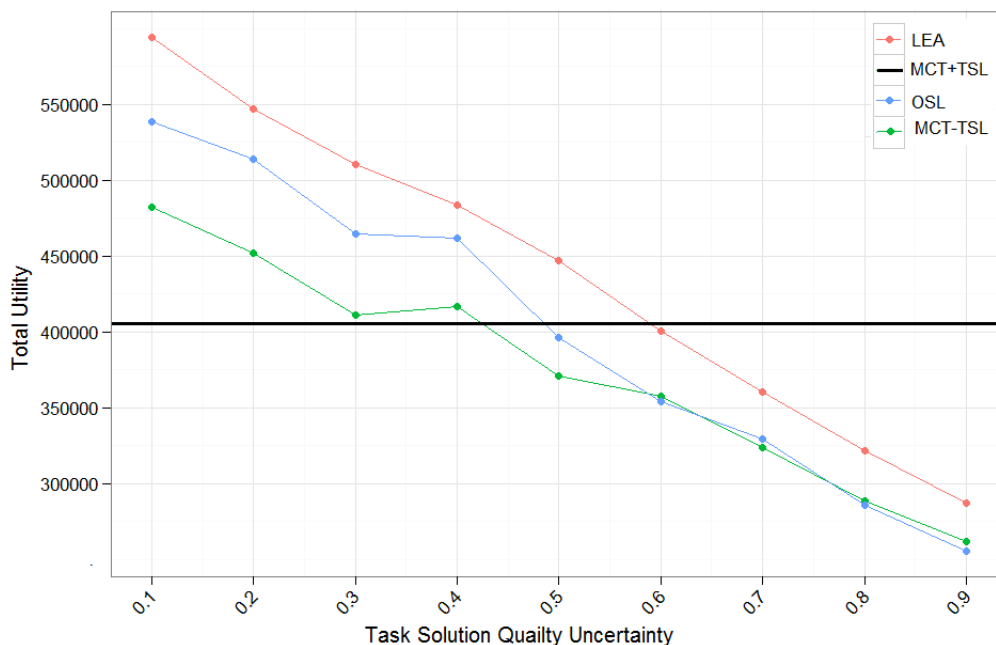
Figure 6.20: Total Utility of Agents for Varying Task Solution Quality Uncertainty.

to learn the type change dynamics (Equation 6.27) by engaging in implicit and explicit learning actions, their initial exploration-before realizing that it is not worthwhile to do so-of the impacts of their learning actions reduces their utility which manifests in the lower utility over all tasks. This observation thus encourages us to seek algorithms or strategies that would allow the OSL and LEA agents to learn the dynamics of the environment more efficiently (see Section 6.8).

6.6.3 Scalability

Here we investigate how the length of the coalition formation round and the neighborhood size (Section 6.2) impact the OSL and LEA agents' ability to earn higher utility.

Impact of Length of Coalition Formation Round on Agents' Utility

Figure 6.27 shows the average (over all tasks) total utility of the agents for varying length of the coalition formation round (in number of simulation ticks) for the chosen simulation
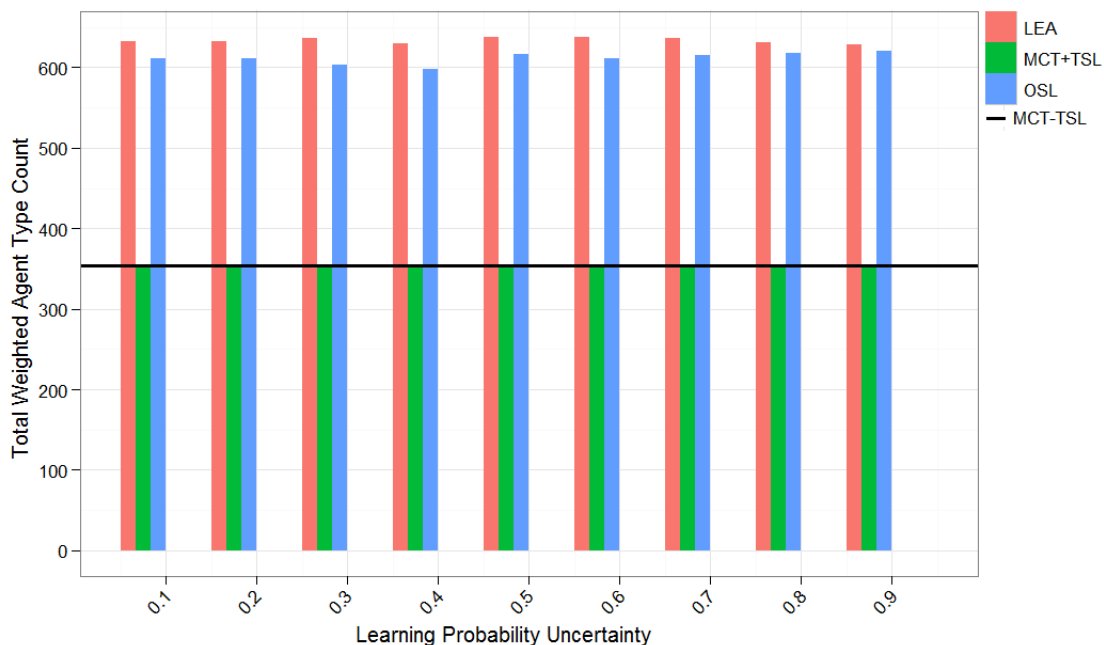
Figure 6.21: Total Weighted Agent Type Count (Interface Designer+Programmer+System Engineer) (Equation 6.28) for Varying Task Solution Quality Uncertainty.

seed values (Table 6.2). Figure 6.27 shows an absence of any statistically significant trend that would indicate the difference in time spent by the OSL and LEA agents while choosing coalitions and their total utility. This means that the OSL and LEA agents can retain the performance improvement-fueled by type changes via learning actions-by entertaining a relatively small number of coalition proposals while forming coalitions. Thus, the results here suggest that when the implicit and explicit learning based type change is used to improve the agents' performance; the agents do not have to rely on searching through the exponential number of coalition choices to improve their performances over time. This further implies that, even when we have a large number of agents in the ADLIT environment, it would be possible to limit the total time spent by the agents while they are deliberating which coalition to join. This lessened dependency on the time spent thus indicates the scalability of the coalition formation protocol and the approximation strategies (OSL and LEA) in ADLIT.
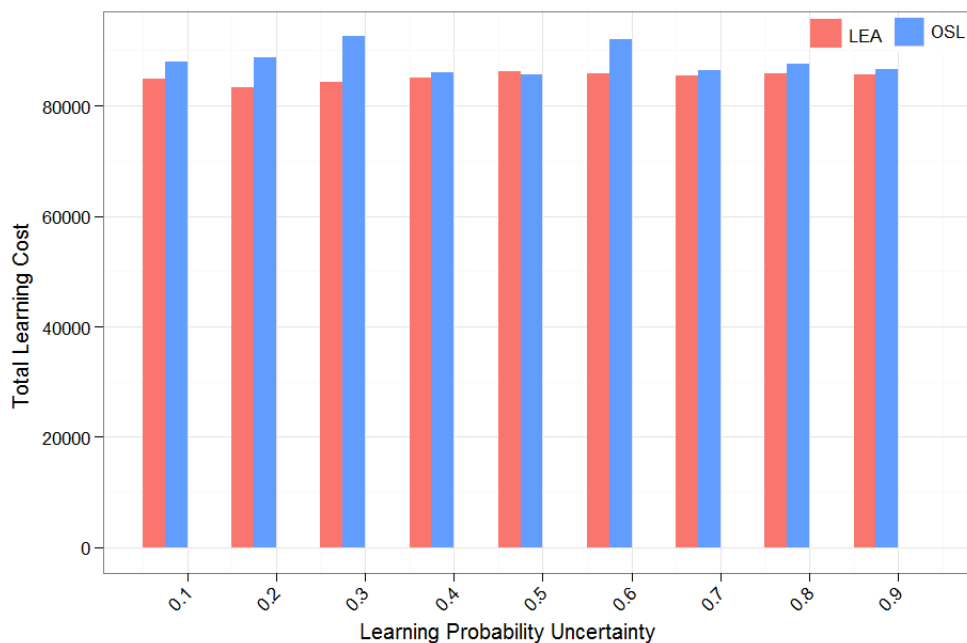
Figure 6.22: Total Learning Cost of the OSL and LEA Agents for Varying Task Solution Quality Uncertainty.

Impact of Neighborhood Size on Agents' Utility, Reasoning Time, and Memory Usage

Figure 6.28, 6.29, and 6.30 show the average (over all tasks) total utility earned, the total reasoning time spent, and the total memory (heap space in Java virtual machine) used, respectively, by the OSL and LEA agents for varying neighborhood sizes. We see that although increasing the neighborhood size increases the time and memory requirements, beyond the value of 6, the neighborhood size increase does not provide any statistically significant ($p < 0.05$) increase in their total utility. Notice that although Figure 6.28 shows a slightly negative trend in the decrease of the utility of the OSL agents as the neighborhood size increases, that decrease is not statistically significant ($p < 0.05$). So, Figure 6.28, 6.29, and 6.30 indicate that the OSL and LEA agents can earn the type changes and improvements in their utility over time by deliberating over a relatively small number of potential coalition-forming choices ($2^9 \times 360$ *as opposed to* 2^{60}). The explanation of the results here is again similar to our discussions in Section 6.6.3. Beyond neighborhoods of size 6, the
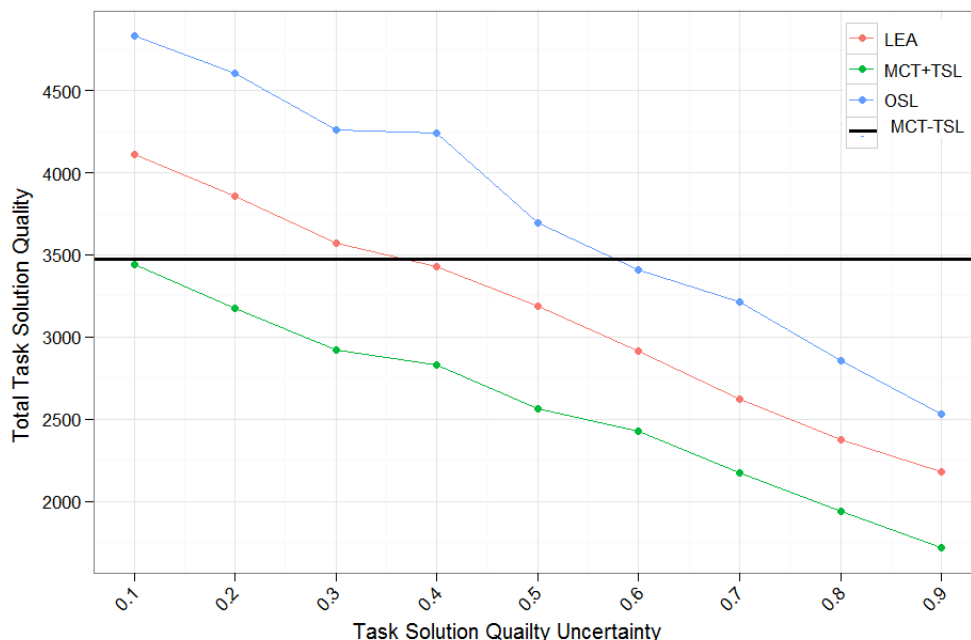
Figure 6.23: Total Task Solution Quality for Varying Task Solution Quality Uncertainty.

OSL and LEA agents are able to find the necessary implicit and explicit learning opportunities to obtain type changes that allow them to earn higher utility over time. Notice that, we do not claim that it is not possible to earn higher rewards for very large neighborhoods, e.g., 20. For such a large neighborhood, the agents would have to deliberate over an exponential number of coalitions and implicit and explicit learning choices ($2^{20} \times 360$ as opposed to $2^9 \times 360$). However, due to the task solution- and learning outcome- related uncertainties in the environment, deliberating over such a large number of choices *will not guarantee* that the agent will be able to achieve (1) a type change that leads to high task solution quality for future tasks or (2) a high task solution quality for the current task. Furthermore, the consideration of such a large number of choices would make the system not usable for any large-scale real-world coalition formation scenarios. So, our results here indicate that our OSL and LEA strategies may allow the agents to earn higher utility over time through type change without having to consider an exponential number of possible coalitions and implicit and explicit learning action choices. Although not definitive, the results here provide
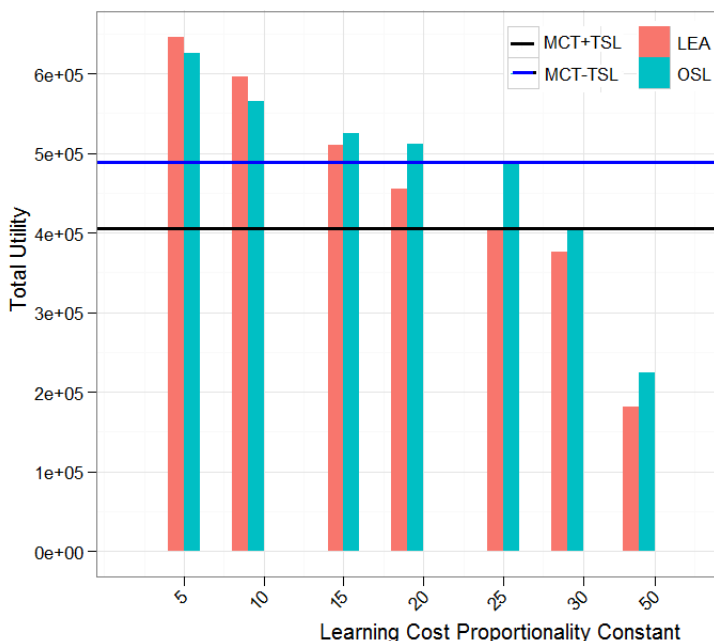
Figure 6.24: Total Utility of Agents for Varying Learning Cost Proportionality Constant.

evidence that the coalition formation protocol and the approximation strategies can be used for solving large-scale real world problems.

6.6.4 Summary

The analysis of our results can be summarized as the following:

Feasibility and Impact:

- The coalition formation protocol and the approximation strategies enable the agents to statistically significantly improve their types over time by choosing the implicit and explicit learning actions.

- Due to the agents' deliberate exploitation of the current-task vs. future-task reward tradeoff resulting from their type change, all agents using our OSL and LEA approximation strategies are able to improve their earned utility (statistically significantly) over time.
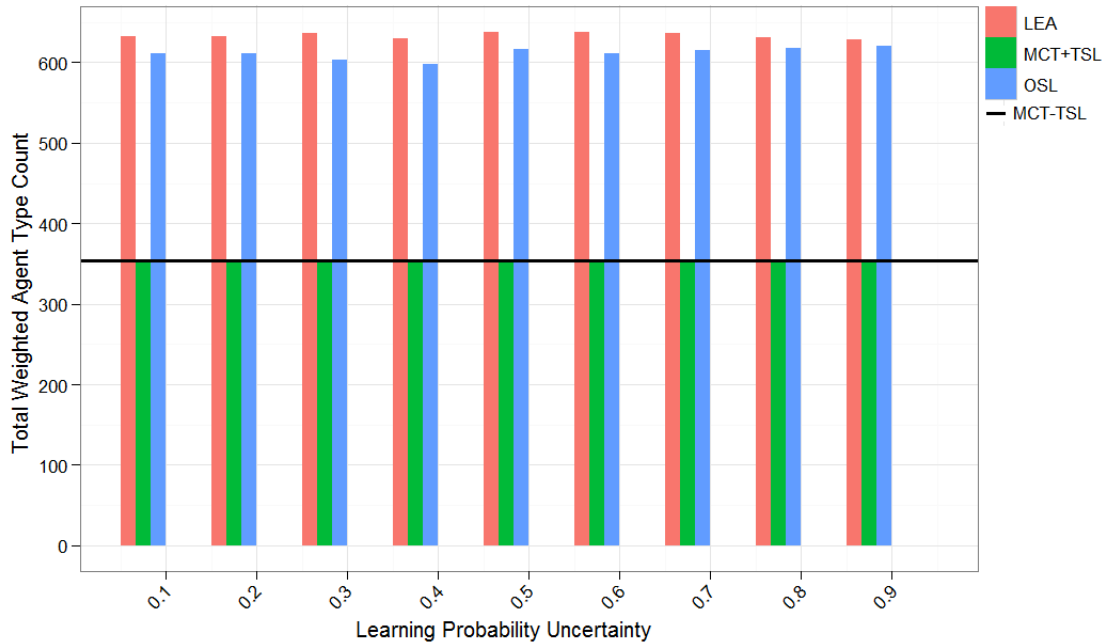
Figure 6.25: Total Weighted Agent Type Count (Interface Designer+Programmer+System Engineer) (Equation 6.28) for Varying Learning Cost.

- The frequentist learning approach allows the agents to estimate the dynamics of the environment sufficiently accurately. The agents' estimation of the dynamics then enables them to choose the appropriate coalition members and implicit and explicit learning actions that lead to the statistically significantly higher performance over time.

- An aggressive exploration of the implicit and explicit learning actions allow the agents to reduce the learning cost they incur and over time leading to more efficient type improvement.

Robustness:

- The agents using the OSL and LEA strategies are able to exploit the current task vs. future task reward tradeoff when the task openness is low < 0.5. For higher task openness, the agents' fail to achieve any performance improvement over the agents
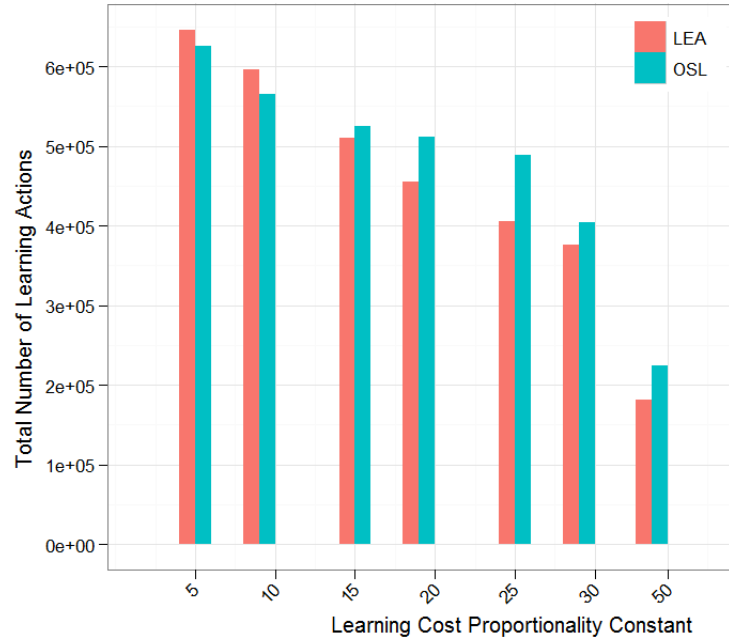
Figure 6.26: Total Number of Learning Actions for Varying Learning Cost Proportionality Constant.

who do not improve their types for exploiting the current task vs. future task reward tradeoff.

- The agents' local decision of improving their own types and the types of their coalition members leads to a globally emergent type improvement of the agents. That type improvement then allows the OSL and LEA agents to earn statistically significantly higher utility overcoming even very high agent openness values.

- If the task solution quality uncertainty is high, the ADLIT agents' use of implicit and explicit learning and type change fail to achieve any performance improvement over the agents who do not improve their types for exploiting the current task vs. future task reward tradeoff.

- ADLIT's coalition formation protocol and the approximation strategies enable the agents to overcome high learning probability uncertainty by finding the most efficient implicit and explicit learning actions for type change.
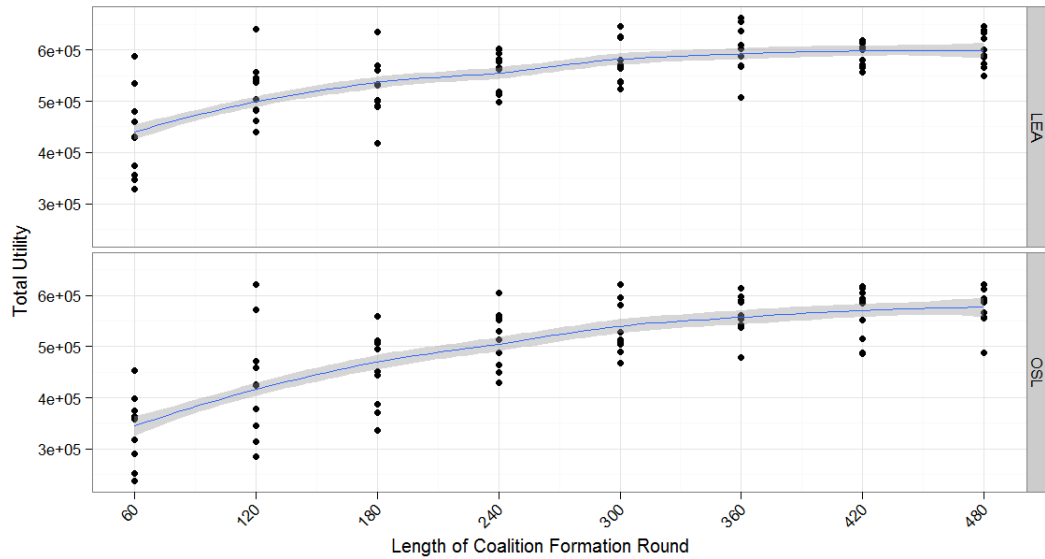
Figure 6.27: Total Utility of Agents for Varying Length of Coalition Formation Rounds.

- When the learning cost is high, the OSL and LEA agents reduce their learning actions where that reduction prevents them from optimizing their types. As a result, the OSL and LEA agents fail to improve their utilities over the MCT agents.

Scalability:

ADLIT agents' ability to exploit the current task vs. future task reward tradeoff by improving their types is not constrained with: (1) the number of potential coalitions and learning action choices they consider or (2) the amount of time they spend communicating or coordinating with other agents for choosing their coalition and learning action choices. The agents' independence from the potentially large number of coalition members and learning action choices suggest that our approach of agents' performance improvement through type change could be scaled up to large coalition formation problems.
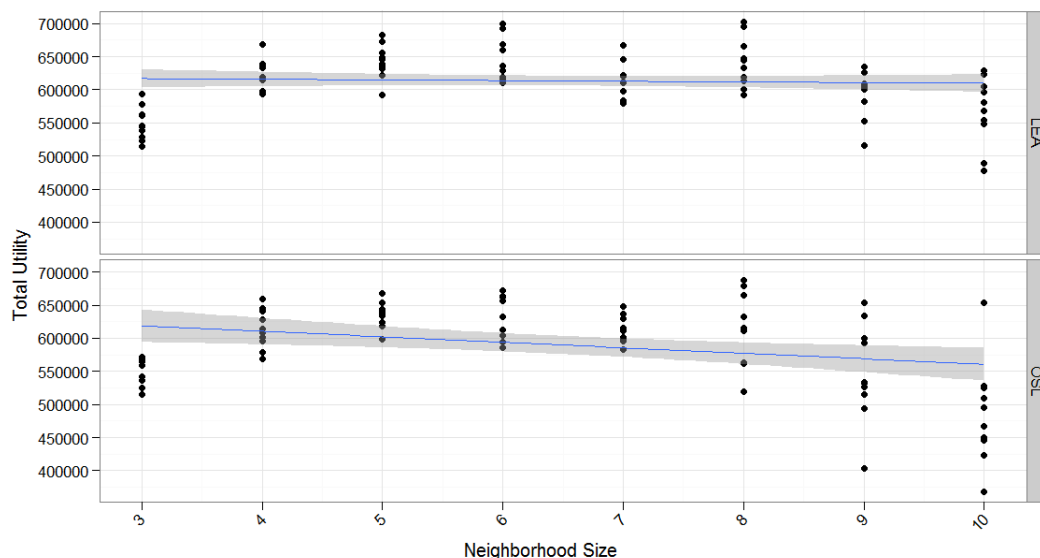
Figure 6.28: Total Utility Earned by the Agents during Coalition Formation for Varying Neighborhood Size.

6.7 Conclusions

We have defined and implemented the ADLIT framework for forming agent coalitions in uncertain environments where the agents' types change due to their participation in the implicit and explicit learning actions with the coalition members. We have described the ADLIT framework with a set of assumptions that formally define our environment's critical aspects: (1) the task and agent openness and (2) the dynamics of the environment that determines (a) the type change of the agents and (b) the reward earned by an agent coalition. For the agents in the ADLIT framework, we have defined a coalition formation protocol that allows the agents to join coalitions that provide them good-enough current task rewards as well as choose implicit and explicit learning actions that change their own types and more importantly, the types of their coalition members. That type change then increases the agents' utility of the future tasks over time. Furthermore, in the ADLIT framework, we have provided two approximation strategies: (1) One-Step Lookahead (OSL) strategy and (2) Learning by Exploration (LEA) strategy. The ADLIT agents use these approximation strategies to evaluate the values of the coalitions in terms of the current task vs. future task
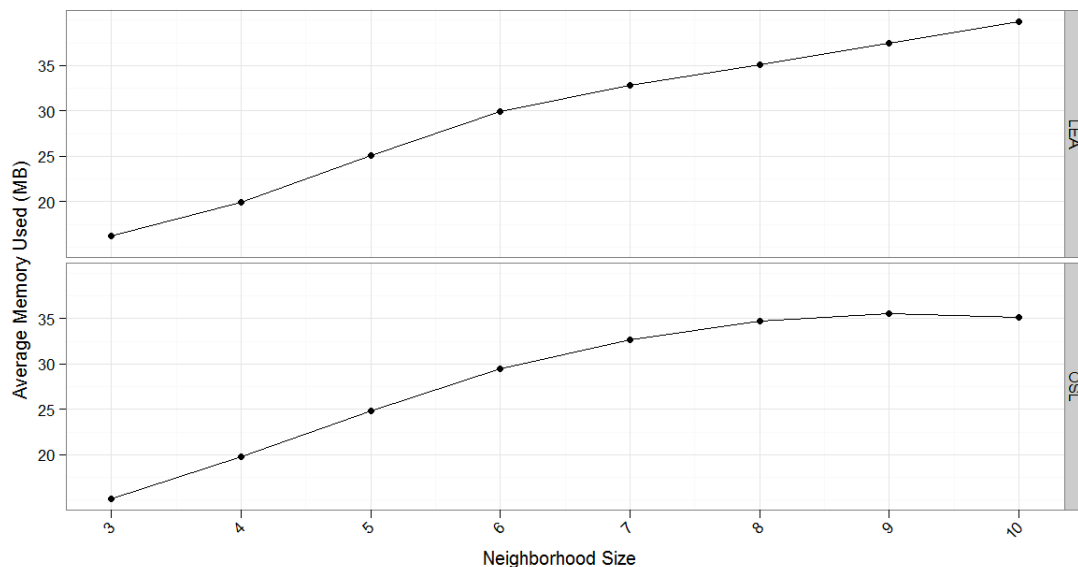
Figure 6.29: Average Memory Used by the Agents during Coalition Formation for Varying Neighborhood Size.

reward tradeoffs they are going to provide.

To test the validity of the coalition formation protocol and the approximation strategies of our ADLIT framework, we have implemented ADLIT in an agent-based simulation environment called Repast and have conducted an extensive set of experiments. The analysis of the results of our experiments shows the following. First, our results show that our designed coalition formation protocol and the OSL and LEA strategies enable the agents to deliberate over their current task vs. future task reward tradeoff and earn statistically significantly higher utility by changing their own types and the types of their coalition members. Second, our results show that the OSL and LEA strategies' effectiveness in improving the agents' utility is negatively impacted by the high learning cost and high uncertainty and openness associated with the task-solving ability of the agents. However, the OSL and LEA strategies allow the agents to withstand, to some extent, the uncertainty associated with the agents' ability to change their types and agent openness. Finally, our results indicate that while forming coalitions, it is possible for the OSL and LEA agents to achieve statistically significantly higher utility gain even when they deliberate over a relatively small number
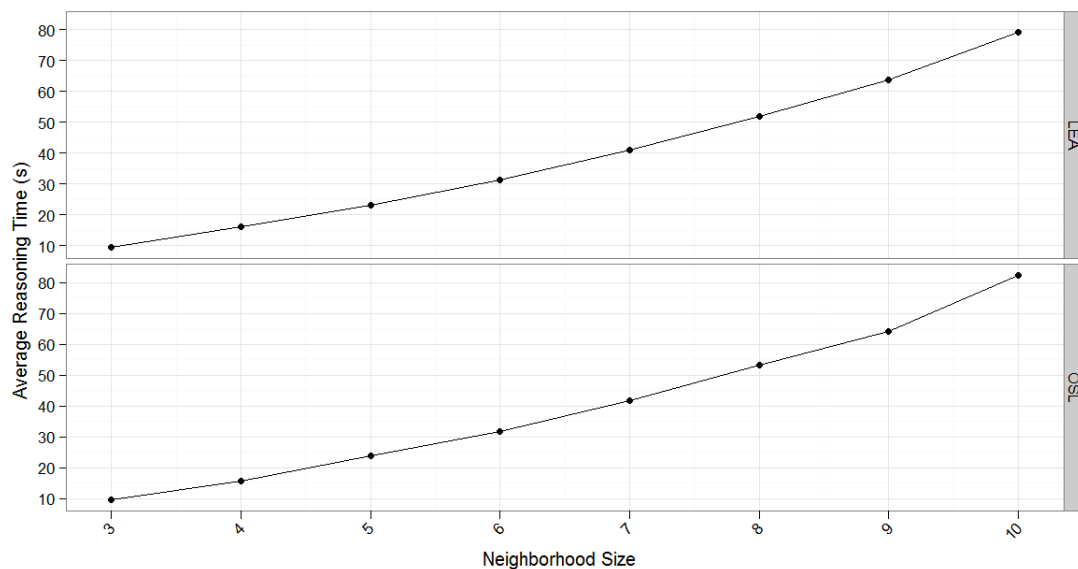
Figure 6.30: Average Reasoning Time Required by the Agents during Coalition Formation for Varying Neighborhood Size.

of potential coalition choices and implicit and explicit learning actions.

To conclude, our results indicate that the type change due to implicit and explicit learning of the agents can be used as an effective strategy to improve their performances in an open and uncertain environment. This is especially true when the agents not only change their own types but the types of their coalition members such as the learning effects that occur when humans work in a coalition. As the multiagent coalition formation research progresses and is used to solve real-world coalition formation problems where software and human agents work symbiotically, this power of implicit and explicit learning can be the key to achieving practical and useful solutions.

6.8 Future Work

We plan to extend the ADLIT coalition formation protocol and approximation strategies to consider the impact of a learner agent's learning on the teacher agents' type change. In the current formulation of the ADLIT framework, when a coalition member is learning

from one of its coalition members, only the learner agent's types are updated. In future, we would like to incorporate the possibility that when a learner and teacher agent engages in learning interactions, the type of the teacher agent may change too (e.g., learning by teaching described in (Inaba et al., 2000)). Furthermore, we plan to investigate the feasibility of using the existing research of Stone and Kraus (2010) in ADLIT Framework. To be specific, following the researchers' approach, we plan to formulate an ADLIT agent's deliberation over whether to engage in teaching and learning interactions with its coalition members as a *k-armed bandit* problem. Then, based on the expected utility improvement calculated from the solution of that problem, the ADLIT agents would decide which implicit or explicit learning action they would choose.

In addition, we plan to improve the learning strategies used by the agents in ADLIT to allow the ADLIT agents better overcome the detrimental aspects of the environment, e.g., openness of the environment, learning cost, and the uncertainty. First, notice that due to agent openness, some of the agents are replaced with new agents over time. These agents do not have any way of estimating the task solution quality dynamics or the agent type change dynamics (Section 6.4.5). So, we plan to change the coalition formation protocol so that an agent can use multiagent cooperative learning (Panait and Luke, 2005) to exchange their learned probability values (regarding the environment dynamics) and learn from one another. Such cooperative learning would enable the agents to improve their estimations more efficiently (i.e., with less number of observations) leading to higher utilities over time. Second, we plan to test the impact of using reinforcement learning mechanisms designed for uncertain environments on the ADLIT agents' learning capabilities. For example, Lauer and Riedmiller (2004) describe how a set of agents can use their reinforcement learning approach to learn the value of reward functions in uncertain environments. This reinforcement learning approach can be easily adopted by the ADLIT agents to further improve their estimations of the task solution quality dynamics and the agent type change dynamics. This improvement may then help the agents overcome the negative impact of the

uncertainties (e.g., associated with the task solution quality dynamics) in the environment better. Finally, in our future work, we plan to explore how the ADLIT agents' coalition formation protocol and the approximation strategies impact the agents' performances in high-risk environments. In high-risk environments, such a wrong choice may result in a bigger loss, e.g., a complete loss of their expertise or a very high penalty in the current task reward. We plan to investigate the performance of the ADLIT agents for increasing penalty (in terms of agents' type change and earned reward for the current task) for their choices of coalition member types and implicit and explicit learning actions.

Chapter 7

Conclusions & Future Work

Here we first summarize our conclusions discussed in Chapter 5, Chapter 4, and Chapter 6 in Section 7.1 and then summarize our future work presented in those chapters in Section 7.2.

7.1   Conclusions

In our research, we have proposed a solution for the MCFP problem where a set of intelligent agents form coalitions and interact among themselves to solve tasks. The environment of MCFP coalition formation problem is uncertain and the types of the agents (i.e., their skills in terms of solving the tasks) may change (positively *or* negatively) over time. Since intelligent agents are now being used to act as mediators to form human coalitions, we have divided our MCFP problem into two subproblems: MCFP-A and MCFP-M. In MCFP-M, a set of mediator agents form or join coalitions on behalf of their assigned human users (actors in the coalitions) so that those human users are able to solve the current task well and improve their types to solve future tasks better. In MCFP-A, a set of actor agents form or join coalitions to solve the current tasks well and improve their types optimally to maximize their rewards over time. To solve MCFP-M problem, we have proposed the *i*HUCOFS framework and to solve MCFP-A, we have proposed the ADLIT framework for

coalition formation.

*i*HUCOFS (Khandaker and Soh, 2008; Soh and Khandaker, 2007) is a multiagent-based coalition formation framework in which a set of intelligent agents assist a set of human users form coalitions and scaffold those human users to optimize their effectiveness and efficiency in the coalitions over a set of tasks. To understand the validity and implications of *i*HUCOFS framework, we have designed SimCoL (Khandaker and Soh, 2010*d*) - a simulation environment in which a set of agents, guided by the published collaborative learning theories and observations, mimic the collaborative learning interactions of the students in a classroom. To instantiate *i*HUCOFS, we have designed the MHCF algorithm (Chapter 3) that uses a negotiation-based multiagent coalition formation algorithm and a Bayesian network to learn the composition of student groups that would allow the members to solve the current task well as well as improve their behavior to earn higher rewards for future tasks. To investigate the impact of MHCF algorithm, we have designed and implemented ClassroomWiki (Chapter 5)—an asynchronous online Wiki environment. Using ClassroomWiki, we have conducted experiments in several University of Nebraska - Lincoln courses where the students collaborated using ClassroomWiki to prepare Wikis on advanced topics and the mediator agents used the MHCF algorithm to form student groups. Our findings in the SimCoL (Chapter 4) and ClassroomWiki experiments (Chapter 5) can be summarized in the following way:

- **SimCoL** (Khandaker and Soh, 2010*d*)

    - Although preliminary, our results indicate that by utilizing the critical learning-related attributes of the students and the collaborative learning interaction theories proposed by researchers, it is possible to closely mimic the collaborative learning interactions and outcomes of the students in a CSCL classroom. We envision that our findings with this simulation toolkit will enable other CSCL and multiagent researchers to bridge the gap between theory and applications with low-cost replicable simulation experiments.

- Our results indicate that, the composition of student groups have a *significant* impact on the students' learning outcomes. This result encourages us to further refine *i*HUCOFS' group formation algorithms which could then have a positive impact on the real-world CSCL systems.

- Our results also show that, using agent-based scaffolding, it is possible to positively influence the performances of the student coalitions in the CSCL environment. Since learning is present in a large variety of real-world problems (e.g., employee training in a business environment), we feel that our simulation results will encourage other researchers to use intelligent agent based scaffolding as a valid method of improving the performances of human users in collaborative settings in general.

- **ClassroomWiki** (Khandaker and Soh, 2010*a*,*b*,*c*)

  - Our results in ClassroomWiki show the strength of intelligent agents in terms of modeling the human users and overcoming the uncertainties in the environment with probabilistic reasoning. This performance of probabilistic modeling and reasoning of the agents will pave the way for future researchers who would apply multiagent coalition formation techniques to track and model human participant's performances in collaborative work settings.

  - Our results show that the *i*HUCOFS' realization in the MHCF algorithm is able to significantly improve the performances of human users by enhancing their learning abilities with better student groups. We envision that this improvement achieved by multiagent coalition formation techniques will further establish its usefulness in solving real-world coalition formation problems.

  - The positive impact of MHCF algorithm's group formation abilities validates the importance of the scaffolding (implicit) provided by the mediator agents. Although explicit scaffolding has been used by researchers (e.g., Intelligent

Tutoring Systems) we believe that in future, our findings here will encourage the multiagent and CSCL researchers to use implicit scaffolding as an efficient way of improving the performances of human coalitions.

To solve MCFP-A, we have proposed the *Agents' Dyadic Learning Influenced Tradeoff* (ADLIT) Framework (Chapter 6) in which a set of learning-enabled agents form coalitions in an uncertain environment and collaborate to solve tasks to earn rewards. In this setting, the agents in ADLIT take a good-enough, soon-enough approach and negotiate to join a coalition and select a set of learning and teaching actions that (1) improve the long-term reward gain by changing its type and its coalition members' types through learning and scaffolding and (2) allow it to earn good-enough short-term reward (including the cost of learning). In ADLIT, we have proposed a limited lookahead algorithm in which he agents take a myopic approach to obtain a limited lookahead in the future and choose the coalition and learning and scaffolding actions to form coalitions. Furthermore, we have proposed a learning by exploration algorithm in which the agents balance their exploration and exploitation to learn how to form coalitions that allow the agents to improve their performances and optimize their rewards. To understand the impact of our designed algorithms, we have implemented them in a simulation environment and conducted extensive simulation experiments. Our findings of those experiments can be summarized in the following way:

- **MCFP-A** (Khandaker and Soh, 2011*a*)

  - Our results indicate that by harnessing the type-changing influences of the agents' learning and teaching abilities, it is possible to optimize the types of the agents and maximize their performances (w.r.t. their types) over time. Here, the improvements in the agents' performances come not only from improving their own types but the types of their coalition members. Since it is possible for the agents to improve their own performances by influencing changes in

other members' types, we believe that, this technique will enhance the existing multiagent coalition formation solution techniques in the uncertain and type-changing environments.

– Our results show that by exploiting the learning and teaching abilities of the agents, it is possible to overcome the detrimental aspects of the environment (e.g., openness, uncertainty, partial observability) to a great extent. Since the real-world coalition formation problem environments often contain these detrimental aspects, we envision that our findings will significantly help the multiagent researchers who would like to apply theoretical multiagent coalition formation techniques to solve real-world agent coalition formation problems.

– Our results also indicate that the agents' use of learning and teaching actions for improving their types is not affected by the scale of the multiagent coalition formation problem it is used to solve. This finding is of great significance since it suggests the applicability of our exploitation of teaching and learning actions towards improving the solutions of large-scale multiagent coalition formation problems.

7.2 Future Work

Our future work regarding our proposed solution of the MCFP-M problem involves the following. First, we plan to improve the multiagent tracking and modeling capabilities of the mediator agents so that they are able to model the performances of the participating human users better. We also plan to improve our realization of the MHCF algorithm for human coalition formation by adding explicit scaffolding capabilities of the mediator agents. Furthermore, we plan to conduct several large-scale CSCL experiments using MHCF to better understand its ability of improving the performances of student groups. We also plan to expand the use of MHCF algorithms beyond just classroom collaboration scenarios. To be

specific, we plan to apply MHCF for human collaboration or coalition formation scenarios where the human users change their types through learning and where agent scaffolding and group formation is relevant. Examples of such collaboration environments include, just-in-time coalition formation for problem solving in social networking sites (e.g., fund raising for earthquake victims in Haiti through facebook), and effective coalition formation among the users for problem solving in crowdsourcing (Brabham, 2008)

Our future work regarding our proposed solution of the MCFP-A problem involves improving the agents' learning of the environment dynamics with more robust learning algorithms. Such improvements would allow the agents to better evaluate the tradeoff between current and future task performances and improve their types optimally in the face of detrimental environmental aspects. We plan to further study the performances of the MCFP-A framework with extensive empirical studies. In those future studies, we will implement ADLIT framework in other real-world multiagent coalition formation scenarios (e.g., uncertain coalition formation in large-scale electronic markets (Lerman and Shehory, 2000)) where the type change of the agents may play a significant role in improving the performances of the coalitions.

Bibliography

Abdallah, S. and Lesser, V. (2004), Organization-based cooperative coalition formation, *in* 'Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Techonology', pp. 162–168.

Amati, G. (2006), Frequentist and bayesian approach to information retrieval, *in* 'Advances in Information Retrieval', Vol. 3936/2006 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp. 13–24.

Anderson, J. (1982), 'Acquisition of cognitive skill', *Psychology Review* **89**, 369–406.

Arroyo, I., Woolf, B., Royer, J., Tai, M. and English, S. (2010), Improving math learning through intelligent tutoring and basic skills training, *in* V. Aleven, J. Kay and J. Mostow, eds, 'Intelligent Tutoring Systems', Vol. 6094 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 423–432.

Asselin, F. and Chaib-Draa, B. (2006), 'Performance of software agents in non-transferable payoff group buying', *Journal of Experimental and Theoretical Artificial Intelligence* **18**, 17–48.

Bachrach, Y. and Rosenschein, J. S. (2008), Coalitional skill games, *in* 'Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems', Vol. 2, Estoril, Portugal, pp. 1023–1030.

Blankenburg, B. and Klusch, M. (2004), On safe kernel stable coalition forming among agents, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', pp. 580–587.

Bonabeau, E. (2002), 'Agent-Based modeling: Methods and techniques for simulating human systems', *Proceedings of the National Academy of Sciences of the United States of America* **99**(3), 7280–7287.

Boongasame, L., Leung, H., Boonjing, V. and Chiu, D. (2009), Forming buyer coalitions with bundles of items, *in* 'Agent and Multi-Agent Systems: Technologies and Applications', pp. 714–723.

Brabham, D. (2008), 'Crowdsourcing as a model for problem solving', *Convergence: The International Journal of Research into New Media Technologies* **14**(1), 75–90.

Breban, S. and Vassileva, J. (2002), A coalition formation mechanism based on inter-agent trust relationships, *in* 'Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems', Bologna, Italy, pp. 306–307.

Bruner, J. (1985), Vygotsky: A historical and conceptual perspective., *in* 'Culture, communication, and cognition: Vygotskian perspectives', Cambridge University Press, Cambridge, England, pp. 21–34.

Bull, S., Greer, J., McCella, G., Kettel, L. and Bows, J. (2001), User modeling in I-Help: what, why, when and how, *in* 'International Conferernce on User Modeling', pp. 117–126.

Busoniu, L., Babuska, R. and Schutter, B. D. (2008), 'A comprehensive survey of multiagent reinforcement learning', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **38**(2), 156–172.

Carmines, E. and Zeller, R. (1979), *Reliability and Validity Assessment*, Sage Publications.

Cazden, C. (1983), 'Peekaboo as an instructional model: Discourse development at home and at school', *Stanford Papers and Reports in Child Language Development* **17**, 1–19.

Chalkiadakis, G. (2007), A bayesian approach to multiagent reinforcement learning and coalition formation under uncertainty, PhD dissertation, University of Toronto.

Chalkiadakis, G. and Boutilier, C. (2003), Coordination in multiagent reinforcement learning: A bayesian approach., *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', pp. 709–716.

Chalkiadakis, G. and Boutilier, C. (2004), Bayesian reinforcement learning for coalition formation under uncertainty, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', pp. 1090–1097.

Chalkiadakis, G. and Boutilier, C. (2008), Sequential decision making in repeated coalition formation under uncertainty, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', pp. 347–354.

Chan, C. and Leung, H. (2007), A distributed mechanism for non-transferable utility buyer coalition problem, *in* 'Proceedings of the IEEE International Conference on Tools with Artificial Intelligence', pp. 82–86.

Chan, C. and Leung, H. (2008), Belief based distributed buyer coalition formation with non-transferable utility, *in* 'Proceedings of the 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services', pp. 60–67.

Christos, E. and Kyparisia, A. (2007), A group formation tool in an E-Learning context, *in* 'Proceedings of the International Conference on Tools with Artificial Intelligence', pp. 117–123.

Clancey, W. (2004), 'Roles for agent assistants in field science: Understanding personal projects and collaboration', *IEEE-SMC* **34**(2), 12–137.

Clear, T. and Kassabova, D. (2005), Motivational patterns in virtual team collaboration, *in* 'Proceedings of the Australasian Conferenceon Computing Education', pp. 51–58.

Collins, A., Brown, J. and Holum, A. (1991), 'Cognitive apprenticeship: making thinking visible', *American Educator* **6**(11), 38–46.

Cornforth, D., Kirley, M. and Bossomaier, T. (2004), Agent heterogeneity and coalition formation: Investigating Market-Based cooperative problem solving, *in* 'Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems', New York, New York, pp. 556–563.

Cortina, J. (1993), 'What is coefficient alpha? an examination of theory and applications', *Journal of Applied Psychology* **78**(1), 98–104.

Cowell, R. (1998), Introduction to inference for bayesian networks, *in* 'Learning in Graphical Models', eds. jordan, M.I. edn, The MIT Press, pp. 9–27.

Cress, U. and Kimmerle, J. (2008), 'A systemic and cognitive view on collaborative knowledge building with wikis', *International Journal of Computer-Supported Collaborative Learning* **3**(2), 105–122.

Dang, V. and Jennings, N. (2004), Generating coalition structures with finite bound from the optimal guarantees, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', New York, New York, pp. 564–571.

Dearden, R., Friedman, N. and Andre, D. (1999), Model based bayesian exploration, *in* 'Proceedings of the Conference on Unertainty in Artificial Intelligence', pp. 150–159.

der Linden, J. V. and Renshaw, P. (2001), *Dialogic Learning*, Kluwer Academic Publishers, Dordrecht.

Diebel, K. (2005), Team formation methods for increasing interaction during in-class group work, *in* 'Proceedings of SIGCSE Conference on Innovation and Technology in Computer science Education', Vol. 37, pp. 291–295.

Dieckmann, T. and Schwalbe, U. (1998), Dynamic coalition formation and the core, *in* 'Proceedings of the Economics Department Working Paper Series, Department of Economics, National University of Ireland - Maynooth'.

Dillenbourg, P. (1999), What do you mean by collaborative learning?, *in* 'Collaborative learning: Cognitive Approaches', Elsevier, Oxford, pp. 1–19.

Ebner, M., Kickmeier-Rust, M. and Holzinger, A. (2008), 'Utilizing Wiki-Systems in higher education classes: a chance for universal access?', *Universal Access in the Information Society* **7**(4), 199–207.

Ellis, H., Ottaway, S., Varner, L., Becker, A. and Moore, B. (1997), 'Emotion, motivation, and text comprehension: The detection of contradictions in passages', *Journal of Experimental Psychology: General* **126**, 131–146.

Emery-Montemerlo, R., Gordon, G., Schneider, J. and Thrun, S. (2004), Approximate solutions for partially observable stochastic games with common payoffs, *in* 'Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems', Washington, DC, USA, pp. 136–143. ACM ID: 1018741.

Erete, I., Ferguson, E. and Sen, S. (2008), Learning Task-Specific trust decisions, *in* 'Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi Agent Systems', Estoril, Portugal.

Fischer, M., Bordini, R., Hirsch, B. and Torroni, P. (2007), 'Computational logics and agents', *Computational Intelligence* **23**(1), 61–91.

Fitts, P. (1964), perceptual-motor skill learning, *in* 'Categories of Human Learning', Academic, New York, pp. 243–285.

Gaston, M. and DesJardins, M. (2005), Agent-organized networks for dynamic team formation, *in* 'Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems', pp. 437–444.

Gogoulou, A., Gouli, E., Boas, G., Liakou, E. and Grigoriadou, M. (2007), 'Forming homogeneous, heterogeneous and mixed groups of learners', *ICUM* pp. 33–40.

Graf, S. and Bekele, R. (2006), 'Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization', *Intelligent Tutoring Systems* **4053/2006**, 217–226.

Greer, J., McCalla, G., Vassileva, J., Deters, R., Bull, S. and Kettel, L. (2001), Lessons learned in deploying a multi-agent learning support system: The I-Help experience, *in* 'Proceedings of AI in Education', pp. 410–421.

Griffiths, N. and Luck, M. (2003), Coalition formation through motivation and trust, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', Melbourne, Australia, pp. 17–24.

Hang, C., Wang, Y. and Singh, M. P. (2008), An adaptive probabilistic trust model and its evaluation, *in* 'Proceedings of the International Joint conference on Autonomous Agents and Multiagent Systems', Estoril, Portugal, pp. 1485–1488.

Hang, C., Wang, Y. and Singh, M. P. (2009), Operators for propagating trust and their evaluation in social networks, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', Budapest, Hungary, pp. 1025–1032.

Harrer, A., Malzahn, N. and Wichmann, A. (2008), 'The remote control approach - an architecture for adaptive scripting across collaborative learning environments', *Journal of Universal Computer Science* **14**(1), 148–173.

Hendrix, P., Gal, Y. and Pfeffer, A. (2009), Learning whom to trust: using graphical models for learning about information providers, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', Budapest, Hungary, pp. 1261–1262.

Hoppe, U., Vardejo, M. and Kay, J. (2003), What learning patterns are effective for a learner's growth?, *in* 'Artificial intelligence in education', H.U. hoppe et al. (Eds.) edn, IOS Press, pp. 219–226.

Hosam, H. and Khaldoun, Z. (2006), Planning coalition formation under uncertainty: Auction approach, *in* 'Proceedings of the Conference on Information and Communication Technologies', Vol. 2, pp. 3013–3017.

Huhns, M., Singh, M., Burstein, M., Decker, K., Durfee, E., Finin, T., Gasser, L., Goradia, H., Jennings, N., Lakkaraju, K., Nakashima, H., Parunak, H., Rosenschein, J., Rubinsky, A., Sukthankar, G., Swarup, S., Sycara, K., Tambe, M., Wagner, T. and Zavala, L. (2005), 'Research directions for service-oriented multiagent systems', *IEEE Internet Computing* (November-December), 65–70.

Iacobucci, D. and Rosa, P. (2005), 'Growth, diversification, and business group formation in entrepreneurial firms', *Small Business Economics* **25**(1), 65–82.

Ikeda, M., Go, S. and Mizoguchi, R. (1997), Opportunistic group formation, *in* 'Proceedings of the World Conference on Artificial Intelligence in Education', pp. 166–174.

Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R. and Toyoda, J. (2000), How can we form effective collaborative learning groups?, *in* 'Intelligent Tutoring Systems', Springer, Heidelburg, Berlin, pp. 282–291.

Issrof, K. and Soldato, T. D. (1996), Incorporating motivation into cooperative learning, *in* 'Proceedings of the European Conference on Artificial Intelligence in Education', pp. 284–290.

Issroff, K. and Jones, A. (2005), 'Learning technologies: Affective and social issues in Computer-Supported collaborative learning', *Computers and Education* **44**(4), 395–408.

Johnson, D. and Johnson, R. (1999), *Learning Together and Alone*, Allyn and Bacon, Needham Heights, MA.

Jung, S.-Y. and VanLehn, K. (2010), Developing an intelligent tutoring system using natural language for knowledge representation, *in* V. Aleven, J. Kay and J. Mostow, eds, 'Intelligent Tutoring Systems', Vol. 6095 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 355–358.

Kaelbling, L., Littman, M. and Cassandra, A. (1996), 'Reinforcement learning: a survey', *Journal of Artificial Intelligence Research* **4**, 237–285.

Kaelbling, L. P., Littman, M. L. and Cassandra, A. R. (1998), 'Planning and acting in partially observable stochastic domains', *Artificial Intelligence* **101**(1-2), 99–134.

Ketchpel, S. (1994), Forming coalitions in the face of uncertain rewards, *in* 'Proceedings of the National Conference on Artificial Intelligence', pp. 414–419.

Khandaker, N. (2009), 'ClassroomWiki website', http://cse.unl.edu/~knobel/classroomwiki/.
URL: *http://cse.unl.edu/ knobel/classroomwiki/*

Khandaker, N. and Soh, L. (2008), 'Forming and scaffolding human coalitions: A framework and an implementation for Computer-Supported collaborative learning environment', *International Transactions on Systems Science and Applications* **4**(2), 3–22.

Khandaker, N. and Soh, L. (2010*a*), Assessing individual contributions to groupwork by a summary of tracked Computer-Supported collaborative activities, *in* 'In Proceedings of Frontiers in Education Conference', pp. S1E–1 to S1E–6.

Khandaker, N. and Soh, L. (2010*b*), 'ClassroomWiki: a collaborative wiki for instructional use with multiagent group formation', *in IEEE Transactions on Learning Technologies* **3**(3), 190–202.
URL: *http://www.computer.org/portal/web/csdl/doi/10.1109/TLT.2009.50*

Khandaker, N. and Soh, L. (2010*c*), ClassroomWiki: a wiki for the classroom with multiagent tracking, modeling, and group formation, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10) (extended abstract)', pp. 1377–1378.

Khandaker, N. and Soh, L. (2010*d*), 'SimCoL - a simulation tool for computer-supported collaborative learning', *accepted for publication in IEEE Transactions on Systems, Man, and Cybernetics-C* **PP**(99), 1–11.

Khandaker, N. and Soh, L. (2010*e*), A wiki with multiagent tracking, modeling, and coalition formation, *in* 'In Proceedings of the Innovative Applications of Artificial Intelligence Conference (IAAI'10)', Atlanta, GA, pp. 1799–1806.

Khandaker, N. and Soh, L. (2011*a*), 'Implicit and explicit learning in repeated coalition formation under type change and uncertainty', *Journal of Autonomous Agents and Multiagent Systems (Under Review)* .

Khandaker, N. and Soh, L. (2011*b*), 'Improving group selection and assessment in an asynchronous collaborative writing application', *(To Appear) in International Journal of Artificial Intelligence in Education* .

Khandaker, N., Soh, L. and Jiang, H. (2006), Student learning and team formation in a structured CSCL environment, *in* 'Proceedings of the International Conference on Computers in Education (ICCE'06)', Beijing, China, pp. 185–192.

Khandaker, N., Soh, L., Miller, L., Eck, A. and Jiang, H. (2011), 'Lessons learned from deploying I-MINDS and ClassroomWiki âĂŞ collaborative learning and writing applications', *(to appear) IEEE Transactions on Learning Technologies* **4**(1), 47–57.

Khanna, T. and Yafeh, Y. (2007), 'Business groups in emerging markets: Paragons or parasites?', *Journal of Economic Literature* **45**(2), 331–372.

Kiam, T., Kwang, M. and Kwek, S. (2007), 'Coalition formation for resource coallocation using BDI assignment agents', *IEEE Transactions on Systems, Man, and Cybernatics* **37**(4), 682–693.

Klusch, M. and Gerber, A. (2002), 'Dynamic coalition formation among rational agents', *IEEE Intelligent Systems* **17**(3), 42–47.

Kraus, S., Shehory, O. and Taase, G. (2003), Coalition formation with uncertain heterogeneous information, *in* 'In Proceedings of the second international joint conference on Autonomous agents and multiagent systems', pp. 1–8.

Kraus, S., Shehory, O. and Taase, G. (2004), The advantages of compromising in coalition formation with incomplete information, *in* 'In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2', pp. 588–595.

Kreijns, K., Kirschner, P. A. and Jochems, W. (2003), 'Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research', *Computers in Human Behavior* **19**(3), 335–353.

Kreppel, A. and Tsebelis, G. (1999), 'Coalition formation in the european parliament', *Comparative Political Studies* **32**(8), 933–966.

Kuhlmann, G., Knox, W. and Stone, P. (2006), Know thine enemy: A champion RoboCup coach agent, *in* 'Proceedings of the National Conference on Artificial Intelligence', Menlo Park, CA, pp. 1463–1468.

Lauer, M. and Riedmiller, M. (2004), Reinforcement learning for stochastic cooperative Multi-Agent systems, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', pp. 1516–1517.

Lerman, K. and Shehory, O. (2000), Coalition formation for Large-Scale electronic markets, *in* 'Proceedings of the International Conference on Multiagent Systems', pp. 167–174.

Li, C., Rajan, U., Chawla, S. and Sycara, K. (2003), Mechanisms for coalition formation and cost sharing in an electronic marketplace, *in* 'Proceedings of the 5th international conference on Electronic commerce', ACM, Pittsburgh, Pennsylvania, pp. 68–77.

Li, X., Montazemi, A. and Yuan, Y. (2006), 'Agent-based buddy-finding methodology for knowledge sharing', *Information and Management* **43**(3), 283–296.

Liu, X., Doore, B. and Li, L. (2008), Scaffolding knowledge Co-Construction in web-based discussions through message labeling, *in* 'Proceedings of the International Conference on Society for Information Technology and Teacher Education', pp. 3041–3046.

Louta, M., Roussaki, I. and Pechlivanos, L. (2008), 'An intelligent agent negotiation strategy in the electronic marketplace environment', *European Journal of Operational Research* **187**(3), 1327–1345.

Maheswaran, R. T., Tambe, M., Varakantham, P. and Myers, K. (2004), Adjustable autonomy challenges in personal assistant agents: A position paper, *in* 'Agents and Computational Autonomy', pp. 1–44.

Palopoli, L., Rosaci, D. and Ursino, D. (2006), 'Agents' roles in B2C e-commerce', *AI Communications* **19**(2), 95–126.

Panait, L. and Luke, S. (2005), 'Cooperative Multi-Agent learning: The state of the art', *Autonomous Agents and Multi-Agent Systems* **11**(3), 387–434.

Pechoucek, M., Marik, V. and Barta, J. (2002), 'A knowledge-based approach to coalition formation', *Intelligent Systems* **17**(3), 17–25.

Rana, O. and Stout, K. (2000), What is scalability in Multi-Agent systems?, *in* 'Proceedings of Conference on Autonomous Agents and Multiagent Systems', pp. 56–63.

Repast (2009), 'Repast simulation toolkit', http://repast.sourceforge.net/.
URL: *http://repast.sourceforge.net/*

Rettinger, A., Nickles, M. and Tresp, V. (2008), A statistical relational model for trust learning, *in* 'Proceedings of the International Joint Conference on Autonomous agents and Multiagent Systems', Estoril, Portugal, pp. 763–770.

Roberts, T. S. and McInnerney, J. M. (2007), 'Seven problems of online group learning (and their solutions)', *Educational Technology and Society* **10**(4), 257–268.

Russel, S. and Norvig, P. (2003), *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey.

Sandholm, T., Larson, K., Andersson, M., Shehory, O. and TohmÃl', F. (1999), 'Coalition structure generation with worst case guarantees', *Artificial Intelligence* **111**(1-2), 209–238.

Sandholm, T. and Lesser, V. (1997), 'Coalitions among computationally bounded agents', *Artificial Intelligence* **94**(1), 99–137.

Schon, A. (1987), *Educating the Reflective Practitioner*, Jossey-Bass Publishers, San Francisco, CA.

Screrri, P., Pynadath, D. and Tambe, M. (2002), 'Towards adjustable autonomy for the real world', *Journal of Artificial Intelligence Research* **17**, 171–228.

Shehory, O. and Kraus, S. (1995), Task allocation via coalition formation among autonomous agents, *in* 'Proceedings of the International Conference on Artificial Intelligence and Multiagent Systems', pp. 655–661.

Shell, D. and Brooks, D. (2007), The unified learning model, *in* 'The Unified Learning Model: How Motivational, Cognitive, and Neurobiological Sciences Inform Best Teaching Practices', Springer, New York: N.Y., pp. 1–4.

Soh, L. and Khandaker, N. (2007), Formation and scaffolding human coalitions with a multi-agent framework, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)', Honolulu, HI, pp. 394–396.

Soh, L., Khandaker, N. and Jiang, H. (2006), Multiagent coalition formation for Computer-Supported cooperative learning, *in* 'Proceedings of the Conference on Innovative Applications of Artificial Intelligence Conference (IAAI'06)', Boston, MA, pp. 1844–1851.

Soh, L., Khandaker, N. and Jiang, H. (2008), 'I-MINDS: a multiagent system for intelligent Computer-Supported collaborative learning and classroom management', *International Journal of Artificial Intelligence in Education* **18**(2), 119–151.

Soh, L., Khandaker, N., Liu, X. and Jiang, H. (2006), A Computer-Supported learning system with multiagent intelligence, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)', Hakodate, Japan, pp. 1556–1563.

Soh, L. and Li, X. (2003), An integrated Multi-Level learning approach to multiagent coalition formation, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', pp. 619–624.

Sombattheera, C. and Ghose, A. (2008), A best-first anytime algorithm for computing optimal coalition structures, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', Estoril, Portugal, pp. 1425–1428.

Spoelstra, M. and Sklar, E. (2008), 'Using simulation to model and understand group learning', *International Transactions on Systems Science and Applications* **4**(1), 76–88.

Stahl, G. (2004), Building collaborative knowing: Elements of a social theory of CSCL, *in* 'What We Know about CSCL and Implementing It in Higher Education', Jossey-Bass Publishers, Norwell, MA., pp. 53–85.

Stahl, G., Anderson, T. and Suthers, D. (2006), *Computer-supported collaborative learning: An historical perspective*, R.K. sawyer (Ed.) edn, Cambridge University Press, Cambridge, UK.

Stone, P. and Kraus, S. (2010), To teach or not to teach? decision making under uncertainty in ad hoc team, *in* 'Proceedings of Autonomous Agents and Multiagent Systems Conference', Toronto, Canada, pp. 1504–1509.

Stone, P., Riley, P. and Veloso, M. (2000), Defining and using ideal teammate and opponent agent models: A case study in robotic soccer, *in* 'Proceedings of International Conference on Multi-Agent Systems', pp. 4–41.

Suijs, J. and Borm, P. (1999), 'Stochastic cooperative games: Superadditivity, convexity, and certainty equivalents', *Games and Economic Behavior* **27**(2), 331–345.

Suijs, J., Borm, P., Waegenaere, A. and Tijs, S. (1999), 'Cooperative games with stochastic payoffs', *European Journal of Operational Research* **113**(1), 193–205.

Tsvetovat, M. and Sycara, K. (2000), Customer coalitions in the electronic marketplace, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', ACM, Barcelona, Spain, pp. 263–264.

Vallerand, R., Pelletier, L., Blais, M., Briere, N., Senecal, C. and Vallieres, E. (1992), 'The academic motivation scale: A measure of intrinsic, extrinsic, and amotivation in education', *Educational and Psychological Measurement* **52**(4), 1003.

Vass, E. (2002), 'Friendship and collaborative creative writing in the primary classroom', *Journal of Computer Assisted Learning* **18**(1), 102–110.

Vassileva, J., Breban, S. and Horsch, M. (2002), 'Agent reasoning mechanism for Long-Term coalitions based on decision making and trust', *Computational Intelligence* **18**(4), 583–595.

Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C. and Grant, S. (1999), A multiagent design of a peer-help environment, IOS Press, pp. 38–45.

Vassileva, J., McCalla, G. and Greer, J. (2003), 'Multi-Agent Multi-User modeling in I-Help', *User Modeling and User Interaction* **13**(1), 179–210.

Vygotsky, L. (1978), *Mind in society*, Harvard University Press, Cambridge, MA.

Wang, Y. and Singh, M. P. (2007), Formal trust model for multiagent systems, *in* 'Proceedings of International Joint Conference on Artificial Intelligence', pp. 1551–1556.

Weiss, G. (1993), Learning to coordinate actions in multi-agent systems, *in* 'Proceedings of the 13th international joint conference on Artificial intelligence', Vol. 1, pp. 311–316.

Weiss, G. and Dillenbourg, P. (1998), What is 'multi' in multi-agent learning?, *in* 'Collaborative learning. Cognitive and computational approaches', Pergamon Press, Oxford, UK, pp. 64–80.

Xiang, Y. (2002), *Probabilistic Reasoning in Multiagent System - A graphical models approach*, Cambridge University Press, Cambridge, UK.

Yamamoto, J. and Sycara, K. (2001), A stable and efficient buyer coalition formation scheme for e-marketplaces, *in* 'Proceedings of the International Conference on Autonomous Agents and Multiagent Systems', Montreal, Quebec, Canada, pp. 576–583.

Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N. and Iwasaki, A. (2005), Coalitional games in open anonymous environments, *in* 'Proceedings of the National Conference on Artificial Intelligence', pp. 509–515.

Zumbach, J., Schonemann, J. and Reimann, P. (2005), Analyzing and supporting collaboration in cooperative computer-mediated communication, *in* 'Proceedings of the 2005 Conference on Computer Support for Collaborative Learning 2005: the Next 10 Years!', International Society of the Learning Sciences, Taipei, Taiwan, pp. 758–767.

Appendix A

Wiki Rating Survey

Questions are in Likert Scale $[1, 5]$ (Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree).

1. The interface of ClassroomWiki was user-friendly.

2. ClassroomWiki's functionalities (forum, revision) helped us work as a group better.

3. It was easy to revise our group's topic using ClassroomWiki's revision tool.

4. It was easy to keep track of our group members' revisions using ClassroomWiki's revision tool.

5. It was easy to communicate with my group members using ClassroomWiki's forum.

6. Help was available in the form of help documents/technical support when technical problems occurred.

7. ClassroomWiki is a better tool for group writing than Blackboard's wiki.

8. Additional comments/suggestions on the usefulness of ClassroomWiki as a group writing tool.

9. Additional comments/suggestions about how we can improve the functionalities ClassroomWiki in future.

Appendix B

IRB Student Consent Form

Informed Consent Form

Identification of Project: Online Collaborative Software Educational Evaluation (IRB#2003-05-281 EP)

Purpose of the Research: Computer-supported collaborative learning allows the participants to communicate and collaborate together to learn from each other using computer software (e.g. text chat, Wiki, webcams, whiteboards). The purpose of this study is to investigate how the various aspects of the behavior of a student and the behavior of his or her group members could improve student learning in a computer-supported cooperative learning environment.

Procedures: This evaluation will involve several interactive collaborative learning sessions. Each session will take approximately 2 hours. The students will be able to join these sessions online using computer-supported collaborative learning software. The students can join these sessions from any computer in the University of Nebraska Lincoln campus or from a computer with internet connection at home.

Risks and/or discomforts: There is little or no risk involved in the project.

Benefits: This should be an interesting learning experience in a computer-supported course as students get to interact with each other in task-based groups and perform structured cooperative learning activities. Some students will get to experience with an online tool for collaboration.

Confidentiality: Your name and identity will be coded on all software-related test forms. Your name will be kept as a basis of validating your participation in all sessions. Once the

sessions are complete, all records tying your code to your name will be destroyed from our online collaborative software project database.

The instructor will not know who is participating in the study until after the course is over and the grades are turned in.

Opportunity to Ask Questions: You have the right to ask questions prior to agreeing to participate and while participating. Please address questions to the CSE office, 402-472-6738, during office hours. If you have questions concerning your rights as a research subject that have not been answered by the investigator, you may contact the University of Nebraska-Lincoln Institutional Review Board, telephone (402) 472-6965.

Please initial here to indicate that you have read this page: ☐

Freedom to Withdraw: You are required to participate in the lab activities and the use of online collaborative software as part of the course requirement. However, you are free to choose whether to allow data that we collect specifically from you to be used in our analysis. Your decision to allow us to use the data will draw no other consequences or loss of benefits or privileges as a student at UNL. Your decision to allow us to use the data will in no way affect your grade in the course.

Consent, Right to Receive a Copy: You are voluntarily making a decision whether or not to participate in this research study. Your signature certifies that you have decided to participate having read and understood the information presented. Your will be given a copy of this consent form to keep.

Name: — Date: —

Signature: —

☐ Check here if you are under 19 years old. (If you check the above, we will give you a parental consent form for your parent to read and sign.)

Principal Investigator: Leen-Kiat Soh
Address: CSE, 256 Avery Hall
CSE Office Phone: 402-472-6738
E-mail: lksoh@cse.unl.edu

Appendix C

IRB Parental Consent Form

PARENTAL INFORMED CONSENT FORM
IRB # 2003-05-281 EP

Online Collaborative Software Educational Evaluation

You are invited to permit your child to participate in this research study. The following information is provided in order to help you to make an informed decision whether or not to allow your child to participate. If you have any questions please do not hesitate to ask.

Your child is eligible to participate in this study because your child enrolls in a UNL-offered course that uses technology to help teach the class. Your child will also be asked if he/she is willing to allow his/her participation data be used in the study.

Computer-supported collaborative learning allows the participants to communicate and collaborate together to learn from each other using computer software (e.g. text chat, webcams, whiteboards). The purpose of this study is to investigate how the various aspects of the behavior of a student and the behavior of his or her group members could improve student learning in a computer-supported collaborative learning environment. Specifically, this evaluation will involve several interactive collaborative learning sessions. Each synchronous session will take approximately 1-2 hours while an asynchronous session is open-ended. Your child will be able to participate in these sessions online using computer-supported collaborative learning software. Your child can join these sessions from any computer in the University of Nebraska Lincoln campus or from a computer with internet connection at home.

Your child, as a student of the course that participates in the online collaborative software educational evaluation, is required to attend and participate in these activities. However,

your child, and you as the parent, has the freedom to choose not to allow data that we collect specifically from your child (hence forth "participation data") to be used in our evaluation and analysis.

There are no known risks associated with this research.

This should be an interesting learning experience in a computer-supported course as students get to interact with each other in task-based groups and perform structured cooperative learning activities. Some students will get to experience with an online tool for collaboration.

Any information obtained during this study which could identify your child will be kept strictly confidential. The information obtained in this study may be published in scientific journals or presented at scientific meetings, but your child's identity will be kept strictly confidential.

☐ Parent's Initials (Page 1 of 2)

Your child's rights as a research subject have been explained to you. You have the right to ask questions prior to agreeing to participate and while participating. Please address questions to the CSE office, 402-472-6738, during office hours. If you have any questions concerning your child's rights as a research subject that have not been answered by the investigator, or to report any concerns about the study, you may contact the University of Nebraska-Lincoln Institutional Review Board, telephone (402) 472-6965.

You are free to decide not to allow us to use your child's participation data without adversely affecting their or your relationship with the investigator or the University of Nebraska-Lincoln. Your decision will not result in any loss of benefits to which your child is otherwise entitled.

The instructor will not know who is participating in the study until after the course is over and the grades are turned in.

DOCUMENTATION OF INFORMED CONSENT

YOU ARE VOLUNTARILY MAKING A DECISION WHETHER OR NOT TO ALLOW YOUR CHILD TO PARTICIPATE IN THE RESEARCH STUDY. YOUR SIGNATURE CERTIFIES THAT YOU HAVE DECIDED TO ALLOW YOUR CHILD TO PARTICIPATE HAVING READ AND UNDERSTOOD THE INFORMATION PRESENTED.

YOU WILL BE GIVEN A COPY OF THIS CONSENT FORM TO KEEP.

Child's Name —

Signature of Parent —- Date —

IDENTIFICATION OF INVESTIGATORS

PRIMARY INVESTIGATOR

Principal Investigator: Leen-Kiat Soh
Address: CSE, 256 Avery Hall
CSE Office Phone: 402-472-6738
E-mail: `lksoh@cse.unl.edu`   Page 2 of 2