# HIGH-SPEED COORDINATION IN GROUPWARE

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon

By

Mutasem Barjawi

# PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying, publication, or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science

176 Thorvaldson Building

110 Science Place

University of Saskatchewan

Saskatoon, SK, S7N 5C9

Canada

# ABSTRACT

Coordination is important in groupware because it helps users collaborate efficiently. However, groupware systems in which activities occur at a faster pace need faster coordination in order to keep up with the speed of the activity. Faster coordination is especially needed when actions are dependent on one another (i.e., they are tightly-coupled) and when each user can see and interact with other users' actions as they occur (i.e., real time). There is little information available about this type of fast coordination (also named high-speed coordination or HSC) in groupware. In this thesis, I addressed this problem by providing a body of principles and information about high-speed coordination. This solution was achieved by creating a groupware game called RTChess and then conducting an exploratory evaluation in which high-speed coordination was investigated. The results of this evaluation show that there were small amounts of high-speed coordination in the game and that high-speed coordination was difficult to achieve. In addition, HSC was affected by five main characteristics of the groupware environment: user experience, level of awareness of the partner's interactions, communication between partners, number of dependencies that affect the user's interactions, and pace of activities in the system.

# ACKNOWLEDGEMENTS

In the name of ALLAH, the most Gracious, the most Merciful

Praise all to ALLAH (God) who helped me complete this thesis; and peace be upon His prophet Mohammad son of Abdullah. Throughout this thesis, any shortage or incorrectness found is due to my shortage and lack of knowledge. Likewise, any completeness and good work found is all due to ALLAH alone.

I would like to thank my parents for their patience and support throughout my life and throughout my graduate studies. My utmost appreciation and thanks to my teacher and supervisor, Dr. Carl Gutwin. Thank you Carl for the knowledge and experience you taught me and for your smile that, during difficult times throughout my study, made me feel that it will be all right. Thanks also to my wife for her dedication, patience, and help throughout my study. To all my teachers: Regan Mandryk, Mark Eramian, and Ralph Deters, thank you for your help and knowledge that you taught me. Additionally, I would like to thank David Flatla, Samer Sallam, Brian de Alwis, and Miguel Nacenta for their help. Finally, I would like to thank everybody who helped or participated in my research.

Thank you all

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| CSCW | Computer-Supported Cooperative Work |
| HSC | High-Speed Coordination |
| RTChess | Real-Time Chess |

# CHAPTER 1

# INTRODUCTION

A *groupware* system is a computerized system that can be used by a group of users who work together to achieve a common goal. A *distributed groupware* system enables users to work with each other using different computers that are placed in different areas. For users to work with each other successfully, a distributed groupware system must communicate the interactions that occur on one side to designated users on other sides of the system. A *real-time groupware* system communicates user interactions at the time when they occur, which enables other users to see and interact with each other immediately.

*Tightly-coupled* interaction occurs when the interactions of each user depend on and/or influence the interactions of other users. For example, in a basketball game, the actions made by the player who has the ball strongly influence other players. For example, moving or passing the ball in one direction will at least change the direction of other players.

Users working with each other in a groupware system need to organize their interactions – for example, they can specify who does what and when. The act of organizing the interactions and making decisions about one's actions so that they work with other users' actions harmoniously is known as *coordination* (Malone and Crowston, 1990). This organization creates social protocols that people follow when using the system. The system itself can also enforce coordination policies between users. These social protocols and system policies can prevent conflicts and improve collaboration between the users and, thus, get the job done more efficiently.

This thesis is interested in the coordination that occurs in tightly-coupled real-time distributed groupware systems – specifically, those systems in which activities occur frequently and quickly. From now on, tightly-coupled real-time distributed groupware systems in which activities occur frequently and quickly will be referred to as *high-speed groupware*. In high-speed groupware systems, the users have only a small amount of time to coordinate and interact with others' activities occurring in the system because new activities will occur shortly and will also need coordination and interaction. In the basketball example, a player needs to react immediately and quickly to a ball thrown their way and, at the same time, be ready to react to new activities that might occur; such as opponents trying to intercept the ball. In this thesis, this type of coordination that occurs at a small time scale – where one user makes split-second decisions about their interactions based on others' activity – will be called *high-speed coordination* (HSC).

## 1.1 Research Problem

The problem addressed in this thesis is that there is little information available about HSC in high-speed groupware. High-speed activities have not succeeded in most groupware to date; in order to support high-speed activities (e.g., online team sports) we need to understand how HSC works and we need to answer the following basic questions:

- Does HSC occur in high-speed groupware environments, and how does it occur?
- Are the users aware of HSC?
- What in groupware makes HSC possible and what makes HSC more difficult?
- What is the speed at which users can coordinate best?

The importance of solving this research problem lies in the importance of understanding human behaviour when coordinating with other users working together in high-speed groupware. In groupware, relying on social protocols to coordinate between users' interactions is not sufficient in many situations such as when facing new types of conflicts that have not been experienced before (Morris et. al., 2004; Greenberg and Marwood, 1994). On the other hand, system policies which are created without a good understanding of how users interact with each other when working in high-speed groupware might handle interaction conflicts the wrong way. Developing a general understanding of HSC for high-speed groupware can help create a body of principles about HSC. This body should help in understanding human behaviour when working on high-speed groupware and, therefore, create coordination mechanisms such as social protocols and/or system policies that better accommodate this behaviour.

## 1.2 Solution

To solve the problem posed in the previous section, this thesis attempts to provide a body of principles and information about HSC. To achieve this goal, a high-speed groupware game – named *real-time chess* (*RTChess*) – was created and was used to conduct an exploratory evaluation. In this evaluation, HSC was investigated between the players of the game. The results of the evaluation were used to understand HSC and to answer the questions posed in the previous section.

RTChess is a game that is similar to the traditional chess game but with some different and new rules (see Chapter Three for detailed explanation about RTChess). These rules are the reason behind RTChess being considered a high-speed groupware game. Since the goal of this thesis is to explore HSC as it occurs naturally, RTChess did not impose any system policies that forced the players to either coordinate or not. Instead, RTChess allowed the players the freedom of using HSC.

## 1.3 Steps in the Solution

Four main steps have been carried out in this research: 1) Definition of HSC, 2) Development of a high-speed groupware system, 3) Determination of the procedure to be used for exploring HSC, and 4) Investigation of HSC in RTChess.

1- *Definition of HSC.* The first step in the solution was to define HSC. HSC is the type of coordination that occurs very quickly due to the high-speed activities that are continuously and frequently occurring in the system (see Section 5.2.3 for HSC definition).

2- *Development of a high-speed groupware system.* The second step was to build the system that was used to explore HSC between team partners. As a result, RTChess was built. For more information about RTChess see Chapter Three.

3- *Determination of the procedure to be used for exploring HSC.* The third step in the solution was to determine how HSC will be explored and investigated.

Since this was an exploratory study, any data recorded from the evaluation can help in the investigation. For this purpose, three channels of data were taken into consideration: observation of user interactions (through recorded material), user feedback (through questionnaires), and system recorded data. Three measures (operating distance, operating area, and switching control of chess pieces) were calculated and examined between the interactions of team partners. These measures were used to understand HSC between team partners.

4- *Investigation of HSC in RTChess.* The last step of this solution was to conduct the exploratory evaluation in order to explore HSC in RTChess and to answer the basic questions of the problem statement.

## 1.4 Evaluation

An exploratory evaluation was conducted in order to investigate HSC in the high-speed groupware game – RTChess. The goal of this evaluation was to investigate HSC between players of the game (more accurately, between team partners of each team separately), and to answer the basic questions posed in the problem statement of this thesis. Data gathered from this evaluation consisted of screen capture and voice recordings, log files that contained the users' interactions as recorded by the system, and user feedback in the answers of the questionnaires.

In general, the results of the study show that there were small amounts of HSC between team partners. Coordination was difficult at the speed of the game and, as a result, some of the players adapted to this difficulty by using new forms of coordination which were quick and did not require much attention (e.g., sending quick voice messages to the partner). Five characteristics of the groupware environment were found to affect HSC: the player's experience with the game, the level of awareness of the partner's interactions, communication between partners, the number of concurrent players on the chessboard, and the speed at which interactions occur in RTChess. For more details about the results of this evaluation and for detailed answers to the questions in the problem statement see Chapter Five.

## 1.5 Contributions

The main contribution of this thesis is the preliminary information gathered about HSC in groupware. This contribution adds to Coordination Theory and to the knowledge base about coordination in distributed environments. An understanding of human behaviour in high-speed groupware environments is also formed. This contribution will help developers build tools, systems, and/or system policies that accommodate human behaviour in high-speed groupware that supports HSC.

In addition to the main contribution, there are two other secondary contributions of this thesis. These secondary contributions are:

- The high-speed groupware game (RTChess) which can be used as a tool for exploring high-speed groupware environments, HSC, and fast activity.
- The initial measures of coordination that can be used to determine the degree of coordination between team members.

## 1.6 Thesis Outline

Chapter Two reviews previous work related to the main areas that HSC falls into: groupware, coupling, and coordination. This chapter also talks about how this research fits into each of these three main areas. In addition, work related to HSC from Kinesiology will be reviewed.

Chapter Three describes the computer game, RTChess, which was created to investigate HSC. First, the game and its user interface will be described. Second, the game's design and architecture will be described. Third, the game play will be described. Last, the play experience will be described to give the reader an idea about how it feels to play the game.

Chapter Four talks about the formal exploratory evaluation conducted to investigate HSC in the high-speed groupware environment (RTChess) and to answer basic questions about it. Details of the evaluation as well as the pilot and main studies and their results will be described.

Chapter Five analyzes the results of the evaluation in order to determine the contribution that has been gained. To do this, the results of the evaluation will be reviewed, and then the main research problem will be considered using these results and detailed answers to the evaluation questions will be given. After that, the evaluation process will be discussed in terms of factors that might have affected the results and in terms of the design issues that were considered throughout the evaluation. The chapter ends with a discussion regarding the lessons learned to improve the support for HSC in groupware.

Chapter Six summarizes the research conducted in this thesis by drawing conclusions. In addition, main as well as secondary contributions of this study will be pointed out. Last, this chapter discusses my vision of what needs to be done for this work in future.

# CHAPTER 2

# RELATED WORK

In this chapter, work related to HSC will be reviewed. In general, high-speed groupware systems are based on the following three areas of study: groupware, coupling, and coordination. In the end, work related to HSC from the sports field will be reviewed.

## 2.1 Groupware

Ellis et. al. (1991) defined *groupware* as a computer-based system that enables a group of people to work on a common task, and that provides an interface to a shared environment. Groupware is divided using two main dimensions: the time when people need to interact with the system, and the place from where people can use the system (Figure 2.1).

The first dimension used to divide groupware is *place*. This dimension divides groupware into co-located (working in the same place) and distributed (working in different places) groupware. *Co-located* groupware aims at enabling users to work effectively when gathered at the same place. Examples of co-located groupware are tabletop systems (e.g., Nacenta et. al., 2007), large display systems (e.g., Czerwinski et. al., 2006; Cao and Balakrishnan, 2003), and single display systems (e.g., Shoemaker, 2001). On the other hand, *distributed* groupware aims at enabling users to work effectively when they are at different places. Distributed groupware lacks the social cues and protocols that people are used to in face-to-face interaction (Hollan and Stornetta, 1992). Examples of distributed groupware are conferencing systems (e.g., Ahuja et. al., 1988), and view-sharing systems (e.g., Greenberg, 1990).

The second dimension used to divide groupware is *time*. This dimension divides groupware into synchronous and asynchronous groupware. *Synchronous* groupware – also known as *real-time* groupware – enables users to interact with each other at the same time. As a result, real-time interaction in groupware is highly disrupted by the presence of network delay (Stuckel and Gutwin, 2008). Examples of synchronous groupware are real-time drawing tools (e.g., Greenberg et. al., 1992;

Ishii et. al., 1992), shared text editors (e.g., Greenberg, 1996), chat systems (e.g., Bradner et. al., 1999), and real-time video games. *Asynchronous* groupware enables users to interact with each other but at different times. In other words, the users of an asynchronous system can choose the time when they want interact with the system. Examples of asynchronous groupware are cooperative hypertext (e.g., Lai et. al., 1988), collaborative data management (e.g., Preguiça et. al., 2000), collaborative writing (e.g., Fish et. al., 1988), and e-mail systems (e.g., Sproull and Kiesler, 1991).

| | **Same Time** | **Different Time** |
|---|---|---|
| **Same Place** | Face-to-face interaction<br>- Tabletop groupware (Nacenta et. al., 2007)<br>- Public displays (O'Hara et. al., 2003)<br>- Meeting Rooms (Rodden and Blair, 1991) | Asynchronous interaction<br>- Team rooms (Roseman and Greenberg, 1996)<br>- Semi-public displays (Huang and Mynatt, 2003)<br>- Message boards |
| **Different Place** | Synchronous distributed interaction<br>- Telephone<br>- Video conferencing<br>- Real-time video games<br>- Chat systems (Bradner et. al., 1999)<br>- Media spaces (Mantei et. al., 1991) | Asynchronous distributed interaction<br>- Electronic mail (Sproull and Kiesler, 1991)<br>- Blogs<br>- Forums |

**Figure 2.1.** Groupware time space matrix (Ellis et. al., 1991) with examples.

This thesis is concerned with distributed real-time groupware. Real-time interaction enables the users to handle each others' interactions as they occur and, therefore, increase the speed at which activities occur in the system. In turn, increasing the speed of activities in the system forms the basis of high-speed groupware.

## 2.2 Coupling

Generally speaking, *coupling* is the degree of dependency between two or more objects. This thesis is interested in coupling between users' interactions. Olsen and Teasley (1996) described interaction coupling as a unidimentional scale determined by the speed at which the response is needed and the amount of interaction required for either clarification or persuasion. Additionally, Gutwin (1998) described interaction coupling as the amount of work a user can do before they need to interact with other users. Depending on the task, interactions can be tightly-coupled, loosely-coupled, or some degree of coupling in between tightly-coupled and loosely-coupled.

*Loosely-coupled* interaction occurs when there is low dependency between users' interactions. Pinelle and Gutwin (2005) described loose coupling using three characteristics: low interdependency (i.e., users' interactions affect each other weakly and/or infrequently), high differentiation (i.e., users' interactions are distinct, separate, and self-contained), and low integration (i.e., managing interdependencies between interactions does not occur regularly). Additionally, Olsen and Teasley (1996) described loosely-coupled interaction as the type of interaction that requires people to be aware of each others' interactions without a need for immediate negotiation, and in which interactions can proceed in parallel. Moreover, Neale et. al. (2004) described loosely-coupled work in terms of few interactions and straightforward communication. For example, in a car manufacturing company, interaction between different departments could be loosely coupled in a sense that the marketing department and the production line department do not need to know details about each others' work. Instead, these departments might need to know only simple information such as how many cars are required and how many cars have been produced.

*Tightly-coupled* interaction occurs when there is high dependency between users' interactions. Olsen and Teasley (1996) described tightly-coupled interaction as the type of interaction where two or more people's work is directly dependent on each other. Immediate communication is required to negotiate some resolution. Neale et.

al. (2004) described tightly-coupled work as being highly dependent on frequent communication. This communication is demanding in a sense that highly interdependent tasks depend on the quality of the communication. Additionally, Stuckel and Gutwin (2008) described tightly-coupled interaction as the type of work in which each user's interactions immediately and continuously influence other users' interactions. For example, in a basketball team, all actions made by the player who is controlling the ball are closely monitored by the team members. If the ball is passed in a certain direction or the player controlling it runs in a specific direction, then other team members will act accordingly.

This thesis is concerned with tightly-coupled interaction. Tight coupling is more closely associated with real-time groupware because a single action made by one user could generate several other reactions made by those who were affected by that first action. These actions and reactions create a rapidly-changing environment which forms the basis of high-speed groupware. Additionally, in high-speed groupware, communication becomes more difficult due to the quick and rapid actions and reactions that occur in the system. For example, during a basketball game, there is little time for the player controlling the ball to talk to their teammates and form a plan.

## 2.3 Coordination

*Coordination* is the act of organizing the interdependencies between different activities that people engage in while working with each other towards common goals (Malone and Crowston, 1990). In other words, if user A coordinates with user B, then user A organizes their interactions with user B in a way that makes working together as effective as possible. According to Neale et. al. (2004), coordination requires people to coordinate both activities and communication. For example, when passing a ball in sports, two players are trying to coordinate their actions with each other. The passing player should take into consideration the position of their teammate, direction of movement, speed, and readiness to receive the ball. At the same time, the receiver needs to be ready, to know when the passing player will actually do the pass, to know the passing player's position, and to know direction of the pass. The faster and more accurately this information is communicated between both players, the better the pass will be. This kind of coordination is what makes "team play" possible in sports.

Coordination is affected by the type of interaction coupling found in the system. In loosely-coupled groupware, users engage in their own activities without the need for frequent communication or coordination with others (Pinelle and Gutwin, 2005). On the other hand, in tightly-coupled groupware, more interdependencies are introduced between users' interactions (Pinelle et. al., 2003) and, thus, more coordination is required.

Previous research in computer-supported cooperative work (CSCW) has not discussed the effects of the speed at which activities occur in the system on coordination. However, it is obvious that communication becomes more difficult between users when activities occur quickly and frequently in a tightly-coupled groupware. Therefore, since coordination is affected by the quality of communication (Neale et. al., 2004), then coordination is also affected by the speed at which activities occur in the system. In low-speed groupware, there is a lot of time which people could use to communicate and coordinate. However, in high-speed groupware less time is available for communication and thus, coordination is more difficult. For example, in a basketball game, if one player decides to apply a new manoeuvre, it would be difficult for the teammates to help in this manoeuvre because they do not know it yet. At the same time, it is difficult for the player to communicate with their teammates and explain how the manoeuvre is done because the game is running and there is no time. This type of coordination, which occurs in high-speed groupware, is the main focus of this thesis and is named here as high-speed coordination (HSC).

### 2.3.1 Coordination Theory

*Coordination theory* is the research area interested in the interdisciplinary study of coordination (Malone and Crowston, 1994). In more detail, Malone (1988) described coordination theory as the body of principles that describes how the activities of separate actors (i.e., those who are performing the activities) can be coordinated. However, these principles should be general enough to work on any kind of actors. For example, the actors could be a group of people, parts of an organization, or parts of a computer system. To facilitate the research of coordination theory, Malone and Crowston (1990) divided coordination into four generic components which are applicable to coordination in any discipline. These components are: goals, actors, activities, and interdependencies between the activities.

The main focus of coordination theory research is on the interdependencies between activities and the coordination mechanisms (also called coordination processes) undertaken to manage these interdependencies (Malone and Crowston, 1990; Malone and Crowston, 1991; Malone et. al., 1993; Malone and Crowston, 1994). The *interdependencies* between different activities form the elements that are unique to coordination (Malone and Crowston, 1990). Examples of common dependencies are: shared resources, producer/consumer relationships, simultaneity constraints, and task/subtask relationships (Malone et. al., 1993). These interdependencies are important to the extent that “if there is no interdependence, there is nothing to coordinate” – Malone and Crowston (1990), page 362.

*Coordination mechanisms* are tools created to manage the interdependencies that arise between activities. In terms of person-to-person coordination, coordination mechanisms are the means people use to coordinate their actions. For example, people may coordinate their actions by pre-arranged agreements (Stuckel, 2008), such as agreeing to divide the work into pieces manageable by each person. People may also use social protocols to coordinate their actions (Stuckel, 2008), such as walking or standing on the left- or right- hand side of the escalator respectively, or saying “hello” when answering a phone call and then listening for a reply from the other party. In addition to pre-arranged agreements and social protocols, the system could enforce a set of rules (also called coordination policies) that help manage the interdependencies (Morris et. al., 2004). A good example for system enforced rules is the car driving law.

Different coordination mechanisms can be created to manage the interdependencies that arise between different activities (Malone and Crowston, 1994). For example, to manage the ‘shared resources’ dependency, mechanisms such as first come/first serve and priority order could be used. However, Malone and Crowston (1994) put specific attention on two main coordination mechanisms: group decision making and communication. The *group decision making* mechanism is used by the actors to make decisions that affect the activities of the whole group. Baker et. al. (2003) describes decision making as the ability to perceive information from the environment, correctly interpret this information, and decide on the best reaction. For example, if two people want to pass through the same door at the same time, then one of them ‘decides’ to let the other person pass first. This decision – to allow the other person to pass first – was made after perceiving that another person was approaching the same door, interpreting that the other person wanted to actually pass through door and that it will happen at the same time, and then deciding to let the other person pass first.

The *communication* mechanism is one of the most common mechanisms used in coordination. A variety of communication options are available: for example, synchronous/asynchronous, paper, voice, or electronic. Generally, communication can be divided into two main groups: explicit and implicit communication. *Explicit communication* is the communication that takes place when one person consciously and deliberately communicates with another person (Pinelle et. al., 2003). For example, one person might call someone else’s name and then start talking with them. According to Pinelle et. al. (2003) there are three main types of explicit communication: spoken, written, and gestural. In tightly-coupled groupware, explicit communication is considered to be slow because it takes time to establish the communication channel and then transfer the information.

*Implicit communication* is the unintentional type of communication where information is observed from the environment (Pagello et. al., 1999). For example, the sound of keystrokes and the appearance of letters on the screen could tell others that someone is typing on the keyboard. In addition, Segal (1994) calls this type of implicit communication, where the user perceives the desired information by observing the environment as well as other users’ interactions, as *consequential communication*. In tightly-coupled groupware, implicit communication is used frequently to transfer information between the users; because either it is too slow to use explicit communication or it is too hard to explain the information explicitly (e.g., using speech, gestures, or written messages) (Stuckel, 2008). Implicit communication is often faster than explicit communication because, in implicit communication, the information is available in the environment unintentionally and can be used by anyone who needs it (Daft, 1986). This extra speed advantage that implicit communication has over explicit communication is what makes implicit communication important in high-speed groupware where many time-dependent tasks exist.

## 2.4 HSC in Kinesiology

High-speed environments, where interactions are tightly coupled and where interactions occur on a real-time basis, can be found in many team games such as football, basketball, netball, and hockey. Elements of high-speed environments can be easily recognized in such team sport games; for example, a group of people working for the same goal (the team) are interacting with each other at the same time (on a real-time basis) and their interactions affect each other (tightly-coupled). In addition, during game-play, team members keep interacting with each other frequently and quickly in an attempt to achieve victory. The interactions between team players need to be coordinated in order to have good "team play" because cohesive interaction between the team members is essential for the team performance (Baker et. al., 2003).

In the team sports field, research studies have examined team coordination and performance due to their importance for the team (e.g., Widmeyer et. al., 1990; Entin and Serfaty, 1999; Eccles et. al., 2004). For example, Eccles et. al. (2004) proposed a conceptual framework of coordination between team members in which coordination and communication were considered from a cognitive perspective (see Figure 2.2). In this framework, coordination was divided into three parts: pre-, in-, and post-process coordination. *Pre-process* coordination occurs when the team prepares itself for game-play by dividing roles and setting tasks. *In-process* coordination is the coordination undertaken during game-play as a response to changes in the task status. *Post-process* coordination occurs when the team reviews and evaluates its behaviour during the game-play and makes decisions and plans for future games. Eccles et. al. (2004) stated that, in many sports, in-process coordination – which is related to HSC – is difficult because of the scarcity of time, cognitive resources, and communication required for coordination. The main reason for this scarcity is the high taskwork (i.e., the elements of a team member's task that are not related to other team members' activities) demands.

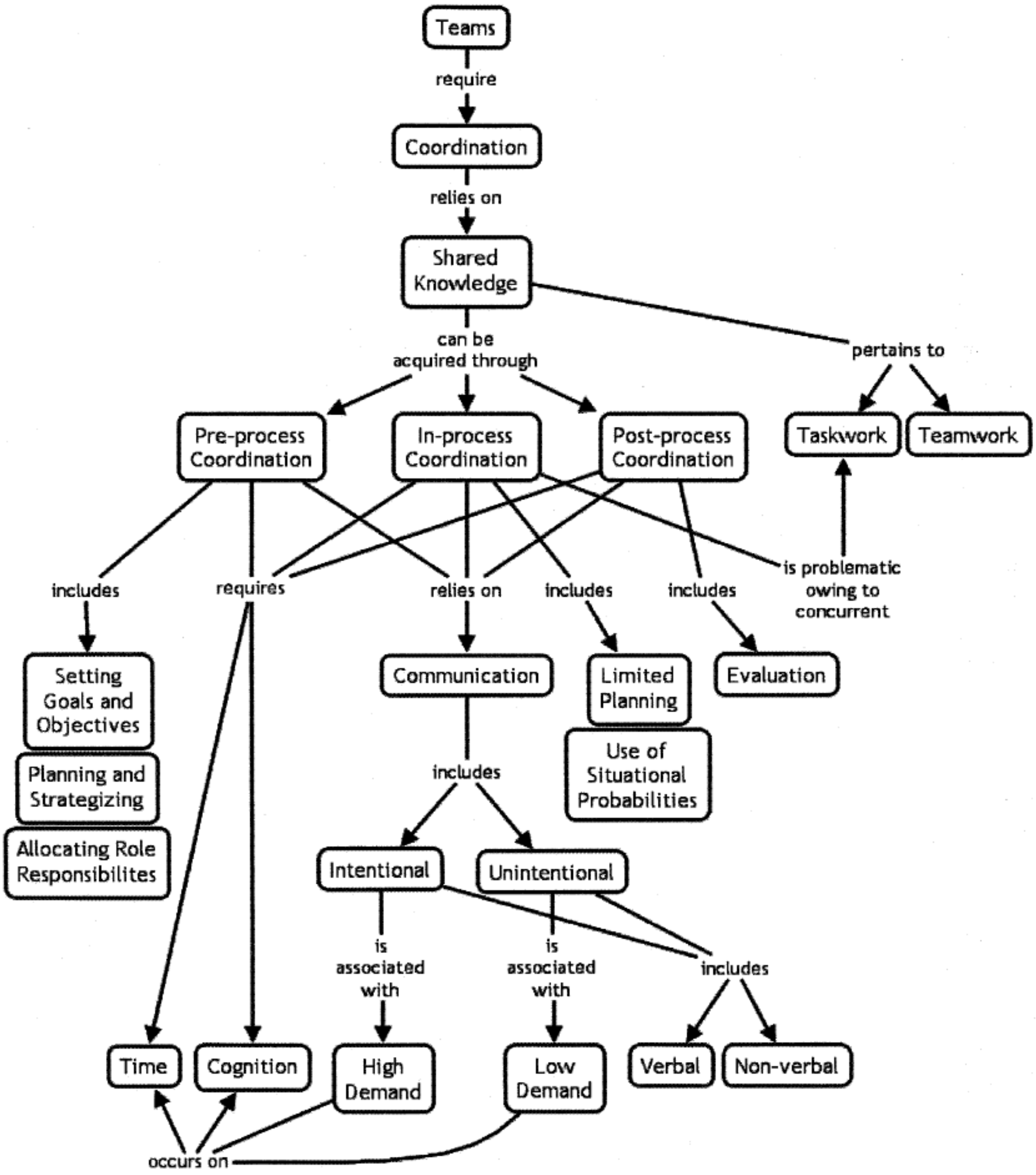

**Figure 2.2.** A conceptual framework of coordination in teams (Eccles et. al., 2004).

Additionally, in Eccles' framework, communication is divided into two parts: intentional and unintentional communication. These two parts are equivalent to the explicit and implicit communication types discussed earlier (see Section 2.3.1). Moreover, according to Eccles' framework, coordination relies mainly on the knowledge shared between the team members. Shared knowledge is divided into two parts: taskwork knowledge and teamwork knowledge. Taskwork knowledge is the knowledge a team member needs to be able to perform his or her task. Teamwork knowledge, on the other hand, is the knowledge a team member needs in order to be able to work as part of the team. The importance of the shared knowledge lies in that it enables team members to generate expectations about other team members' interactions. These expectations, in turn, help team members in deciding their own interactions more accurately and, eventually, in performing better as a team.

# CHAPTER 3

# REAL-TIME CHESS

This chapter describes the computer game, RTChess, that was created to investigate HSC. First, RTChess will be defined and its user interface will be described. Second, the game's design and architecture will be described. Third, the game play will be described. Last, the play experience will be described to give the reader an idea about how it feels to play the game. At the end of the chapter, the following terms should be clear to the reader because they will be used in the chapters to come: select action, ghost piece, select line, move action, move path, move step, and capture action.

## 3.1 Definition and User Interface

Real-Time Chess (RTChess) is a chess-like game that incorporates a high-speed groupware environment. Similar to the traditional chess game, RTChess has 32 chess pieces and an 8x8 chessboard (see Figure 3.1). Additionally, in RTChess the chess pieces move the same way they do in the traditional chess game.

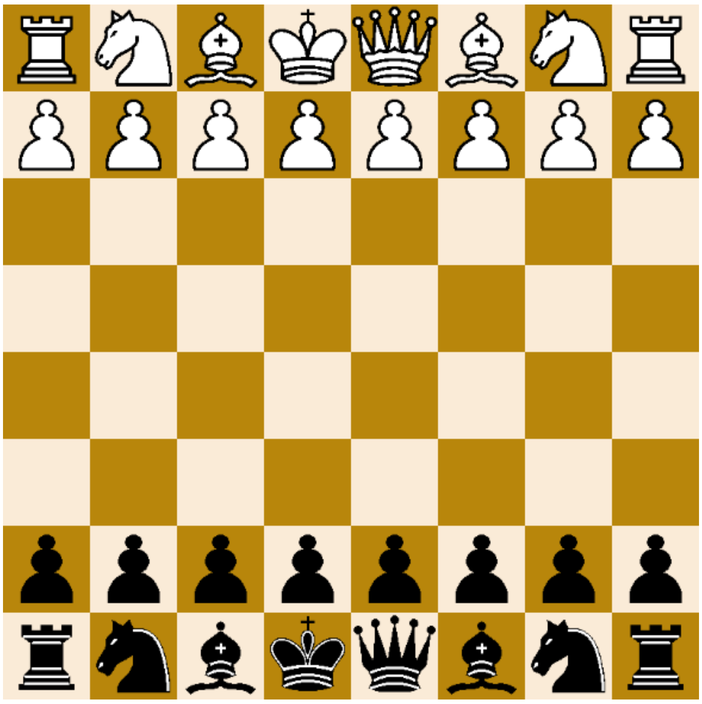

**Figure 3.1.** Screenshot of RTChess, the high-speed groupware game.

RTChess introduces two new rules: first, the number of players on each side is not limited to one player; which means that any number of players can play at the same time. Second, players do not take turns, which means that any player can make any move at any time they want (for further explanation on these two rules see Section 3.2).

The user interface of RTChess consists of an 8x8 chessboard as well as 16 white and 16 black chess pieces. In addition, the user interface contains grey and black telepointer cursors that represent the white and black team players (see Figure 3.2). Each telepointer represents only one player and has the name of this player written next to the cursor. Moreover, RTChess players can *select* and *move* friendly (i.e., from the same team) chess pieces as well as they can *capture* opponent chess pieces. Detailed explanations of these three interactions (select, move, and capture) will come later (see Section 3.3).

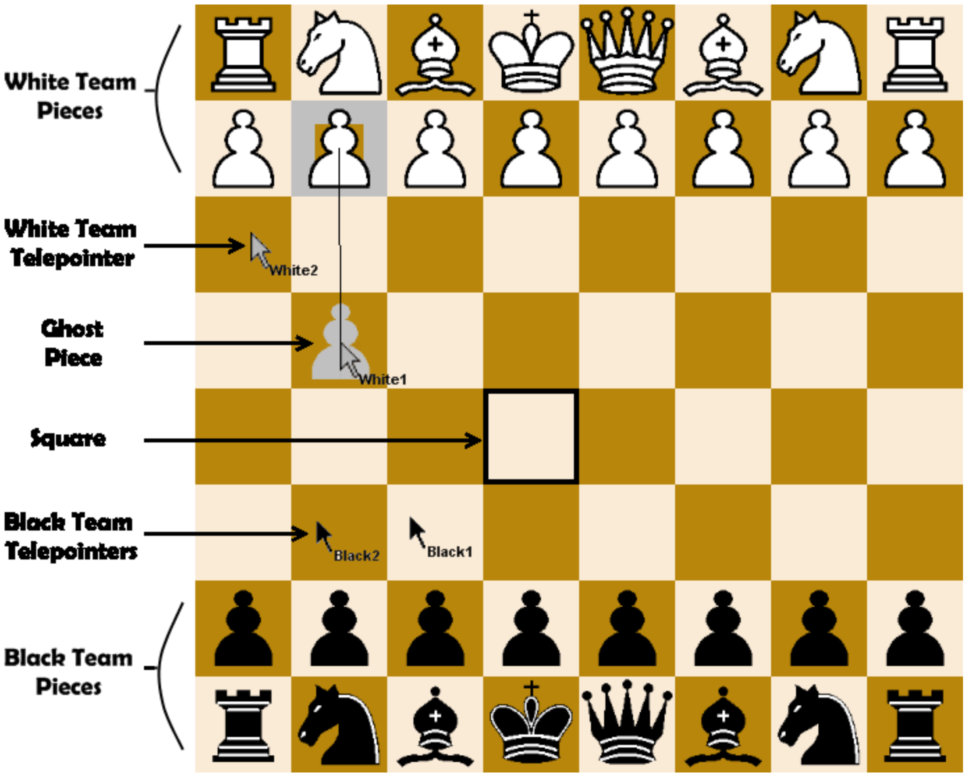

**Figure 3.2.** Screenshot of an RTChess board with annotations.

A player can select a chess piece by clicking and holding it. When a piece is selected, a *ghost piece* – a gray-coloured copy of the chess piece that is being selected (see Figure 3.2) – appears at the selecting player's telepointer to indicate the selection. In addition, a player can move a chess piece by releasing the ghost piece of an already selected chess piece into a new square where the chess piece is allowed to move. In a chess move action, the *from* and *to* squares between which the chess piece moves are called *start* and *end* squares. Last, a player can capture a chess piece by moving a friendly chess piece to a square that already contains an opponent chess piece.

## 3.2 System Design and Architecture

RTChess is a distributed groupware system built in Microsoft C# using the GT toolkit[1] and following the client-server system model. This model is based on having a server running and listening on a network for incoming client connections. The client initiates a connection with the server and opens a communication channel with it. RTChess is a centralized system in that the server holds the master game state and controls what other clients can or cannot do. All clients need to get permission from the server in order to be able to change the state of the game. The server is also the center for passing messages to other clients. For example, when a player makes a move, a request is sent to and checked by the server. If the move is legal, then the server will update its game state, and broadcast the state to all clients to update their game states.

Since all messages arrive at the server, a concurrency model had to be followed in order to help decide which message gets to be processed first. The solution was simply a first-come-first-served strategy. For example, if two players select the same piece at the same time, then the player whose request arrives at the server first is the one that gets to select the piece.

RTChess is similar to the traditional chess game in the way that the pieces move and in the way that the pieces capture opponent pieces. For example, the king moves one square in any direction, the rook moves vertically or horizontally any number of squares, and the knight moves in an L shape. Additionally, the pawn captures an opponent chess piece by attacking it diagonally. The goal in RTChess is to win the game by capturing the opponent team's king.

---

[1] For more information about GT, please visit: http://hci.usask.ca/research/gt/index.shtml.

On the other hand, some rules from the traditional chess game do not exist in RTChess. For example, the king is no longer affected by stalemate or checkmate. Additionally, special moves like castling and en passant are not available in this game. The only special move available is promotion, through which a pawn will be promoted when it reaches the opponent’s base. However, this special move – promotion – is limited to promoting pawns into queens only.

Moreover, RTChess has introduced two new rules: first, there are no turns. In other words, the players can make chess moves whenever they want without waiting for their own turn. Second, any number of players can play in each team. Each player joins the game through an RTChess client application on a separate machine which connects to the server. They can then specify a nickname and a team (white or black) which they want to join. After that, they can start interacting with the system and with other players.

In addition to being groupware, RTChess incorporates a high-speed activity environment. The high-speed activity is caused by two characteristics of RTChess’ groupware environment: real-time interaction and tightly-coupled interaction. Real-time interaction allows each player to see the actions of other players and, most importantly, to interact with these actions as they occur. In addition, to further increase the sense of real-time interaction, chess pieces in RTChess do not move instantaneously from the start square to the end square as they do in the traditional chess game. Instead, in RTChess, once a move action occurs, the chess piece starts travelling automatically from the start square to the end square. During this automatic movement (also called *auto-step movement*), the chess piece moves one square at a time, waiting 250ms at each square, until it reaches the end square (see Section 3.3 for further explanation on chess piece movement). During an auto-step movement, the player who initiated the move is not allowed to select or move any chess piece, including the currently moving one, until the current move ends. In addition, the auto-step movement allowed for further real-time interaction such as intercepting a chess piece while moving.

Tightly-coupled interaction occurs naturally in RTChess similar to the case found in the traditional chess game. In the traditional chess game, each player observes the chess moves made by the opponent and decides the next chess move accordingly. In RTChess, however, imagine the situation where a player needs to observe multiple opponents’ as well as multiple partners’ chess moves when, at the same time, all of these chess moves and interactions are occurring continuously on a real-time basis. In addition to these factors (real-time and tight-coupling), the game play itself has a role in making RTChess a high-speed environment; since one team is continuously moving and chasing the second team’s king, the second team needs to keep moving in order to protect the king and attack the opponent team.

## 3.3 Game Play

Players join the game by starting the client application and connecting to the server. After that, they choose a nickname and a team (white or black). A standard visual chessboard appears on the display through which the players can interact with the game. The title bar of the window shows the nickname of the player and the number of both white and black players that are currently joining the game. Each player is represented with a coloured telepointer (gray = white team, black = black team) and a nickname that appears next to the telepointer. The standard Microsoft Windows mouse pointer is used to represent each player at their own machines (see Figure 3.3).

**Figure 3.3.** Screenshot of the client window that appears once a player joins the game. The title bar shows the player's nickname (white1), the team (White), and the number of white and black players currently joined (W = white, B = black). The white cursor is the local player's cursor (white1 in this case).

After joining the game, the players can start interacting with the game immediately. The players are allowed three types of interactions: the *select* and *move* actions of friendly chess pieces, and the *capture* action of opponent chess pieces. In order to *select* a chess piece, the player is required to click and hold the left mouse button on the desired chess piece. Two conditions have to be met in order for the select action to be successful: first, the chess piece has to be of the same team as the selecting player (i.e., a friendly chess piece; for example a black team player can only select a black chess piece and vice versa). Second, the chess piece has to be free (i.e., no other player from the same team is selecting or moving it). If the conditions are not met, then the select action fails and nothing happens. On the other hand, if the conditions are met, then the select action succeeds and the following changes appear on the user interface: a gray outline appears underneath the selected piece; a *ghost* piece – a gray-coloured copy of the chess piece that is being selected (see Figure 3.4) – is created at the selecting player's telepointer; and a line (called the *select line*) is drawn between the ghost piece and the selected piece to enable the players to trace the ghost piece back to the selected piece. The selecting player can drag the ghost piece around the chessboard freely because the ghost piece is just a visual indicator which shows that a piece has been selected.

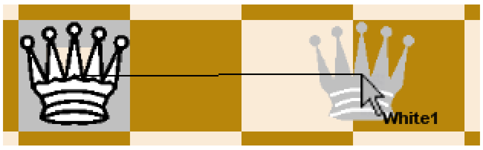

**Figure 3.4.** Simulation of a successful select action of the white queen by player white1.

In order to *move* a chess piece the player is required to select a piece, drag the ghost piece to the desired empty destination square, and then release the ghost piece. I will refer to the square on which the selected piece resides as the *start square* and the destination square over which the ghost piece was released as the *end square* of the move. Two conditions have to be met for the move to be valid: first, the move itself has to be valid for the selected chess piece following the movement rules of the traditional chess game. Second, except for the knight, the path from the start square to the end square (called the *move path*) must be empty (the reason for which will be explained shortly). If the conditions are not met, then the move action fails, the chess piece is deselected, all the visuals of the select action (i.e., the gray outline, the ghost piece, and the select line) disappear, and nothing happens (see Figure 3.5).

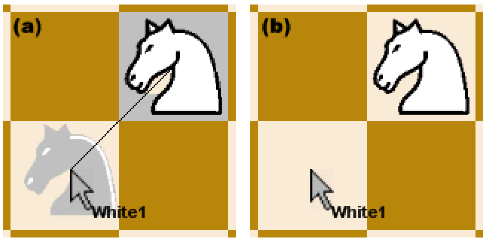

**Figure 3.5.** Simulation of a failed move action. (a) White1 selects and drags the knight. (b) White1 releases the knight. The move action is validated and found invalid. Therefore, the move is rejected and the visuals (gray outline, ghost piece, and select line) disappear.

On the other hand, if the conditions are met, then the move action succeeds, a gray outline appears on top of the end square, the ghost piece and the select line disappear, and the chess piece starts moving (see Figure 3.6).

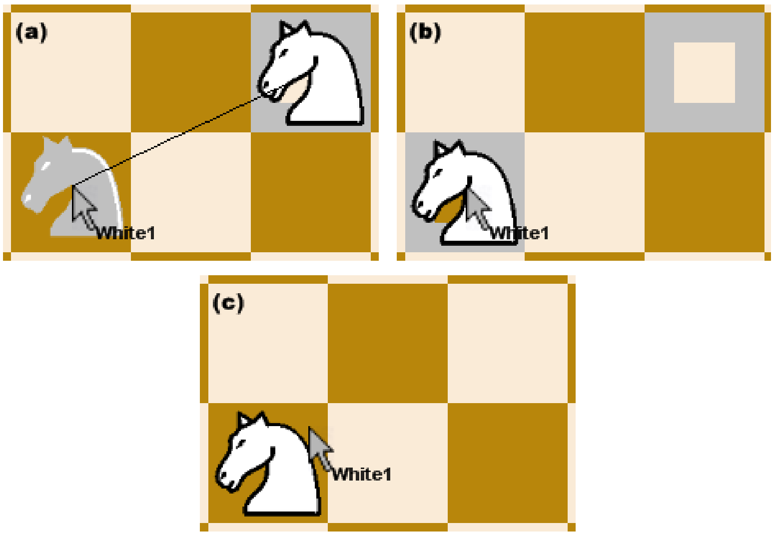

**Figure 3.6**. Simulation of a successful move action. (a) White1 selects and drags the knight. (b) White1 releases the knight. The move action is validated and found valid. Therefore, the knight starts automatic movement immediately. (c) The move action ended and the piece is ready to be picked up again.

In a traditional chess game, a chess piece moves from the start square to the end square instantaneously; in RTChess, however, things are different. In RTChess, once a move action is initiated, the chess piece starts travelling automatically from the start square to the end square. During this automatic movement (also called *auto-step movement*), the chess piece moves in steps, one square at a time, and waits 250ms before taking the next step (see Figure 3.7). Hence the empty move path condition required for a successful move action. The only exception to this rule is the knight which moves instantaneously from the start square to the end square (see Figure 3.6).

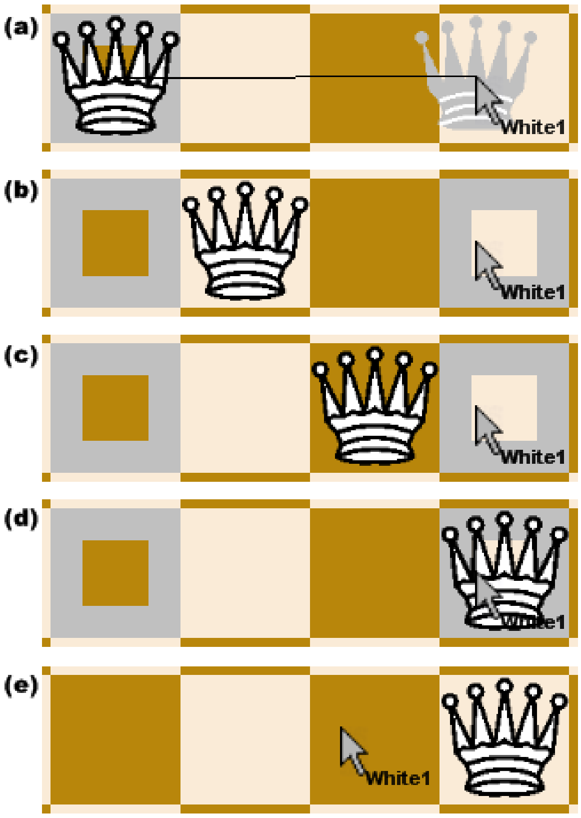

**Figure 3.7.** Simulation of a successful move action. (a) White1 selects and drags the queen. (b) White1 releases the queen. The move action is validated and found valid. Therefore, the queen starts auto-step movement immediately by taking one step. (c) After 250ms, the queen automatically takes another step. (d) After 250ms, the queen automatically takes another step and arrives at the end square. (e) After 250ms, the move action ends and the piece is ready to be picked up again.

In order to *capture* an opponent chess piece, the player is required to select a chess piece, drag the ghost piece to an end square on which an opponent resides, and then release the ghost piece. If the capture is invalid (i.e., does not follow the traditional chess game rules), then the capture action fails, the chess piece is deselected, all the visuals of the select action (i.e., the gray outline, the ghost piece, and the select line) disappear, and nothing happens (see Figure 3.8).

**Figure 3.8.** Simulation of a failed capture action. (a) White1 selects and drags the pawn. (b) White1 releases the pawn on top of the black knight. The capture action is validated and found invalid. Therefore, the capture is rejected and the visuals (gray outline, ghost piece, and selected line) disappear.

On the other hand, if the capture is valid, then the friendly chess piece start moving (auto-step movement) towards the opponent chess piece. Because of the auto-step movement pattern, the opponent chess piece does not get captured until the friendly chess piece actually steps on the square on which the opponent chess piece resides. If the opponent chess piece gets captured, then it disappears, the auto-step movement ends, and the capture action ends (see Figure 3.9).

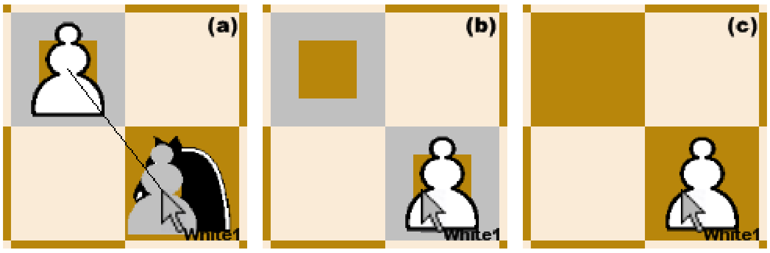

**Figure 3.9.** Simulation of a successful capture action. (a) White1 selects and drags the pawn. (b) White1 releases the pawn on top of the black knight. The capture action is validated and found valid. Therefore, the knight gets captured. (c) The capture action ended.

The auto-step movement pattern in RTChess allowed for more real-time interaction. Chess pieces can now enter and/or exit the move-path of a moving chess piece. The moving chess piece is not affected by these interactions (someone getting in or out of the move path) until it is about to step on a square on which another chess piece resides. Three cases exist in this scenario: first, the moving chess piece is about to step on a friendly chess piece which will cause the auto-step movement to stop because the move path is blocked (see Figure 3.10).

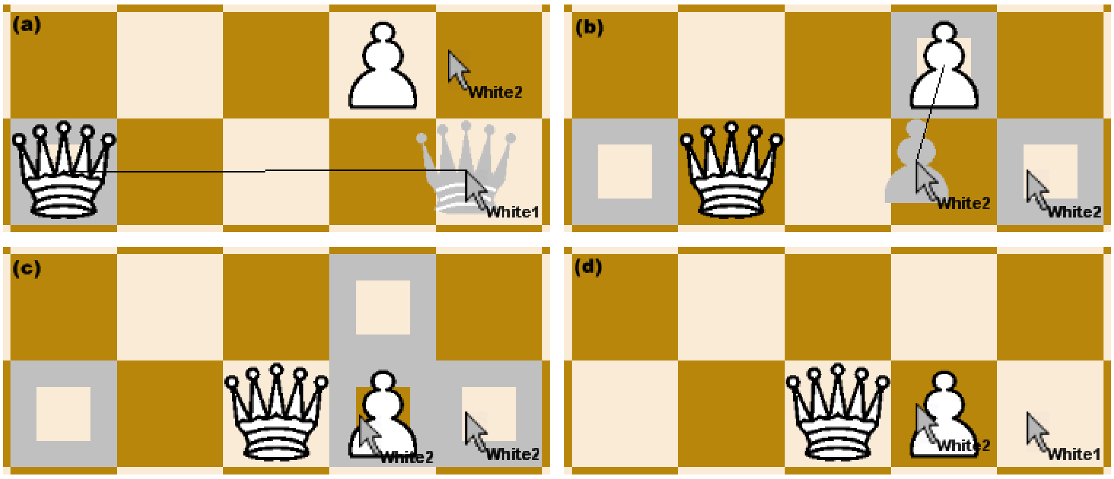

**Figure 3.10.** Simulation of an auto-step movement that got blocked by a friendly chess piece. (a) White1 selects and drags the queen. (b) White1 releases the queen. After successful validation, the queen starts auto-step movement. White2 selects and drags the pawn. (c) White2 releases the pawn into the move path of the queen. After successful validation, the pawn starts auto-step movement. The queen takes another step. (d) The pawn arrives at the end square and ends its movement. The queen attempts to take another step but fails because the pawn is blocking the way. The queen ends its movement.

Second, the moving chess piece is about to step on an opponent chess piece, in which case a capture action is validated. If the capture action was valid, then the opponent chess piece is captured and the moving chess piece stops (see Figure 3.11). Otherwise, if the capture action was not valid, then the auto-step movement stops because the move path is blocked.

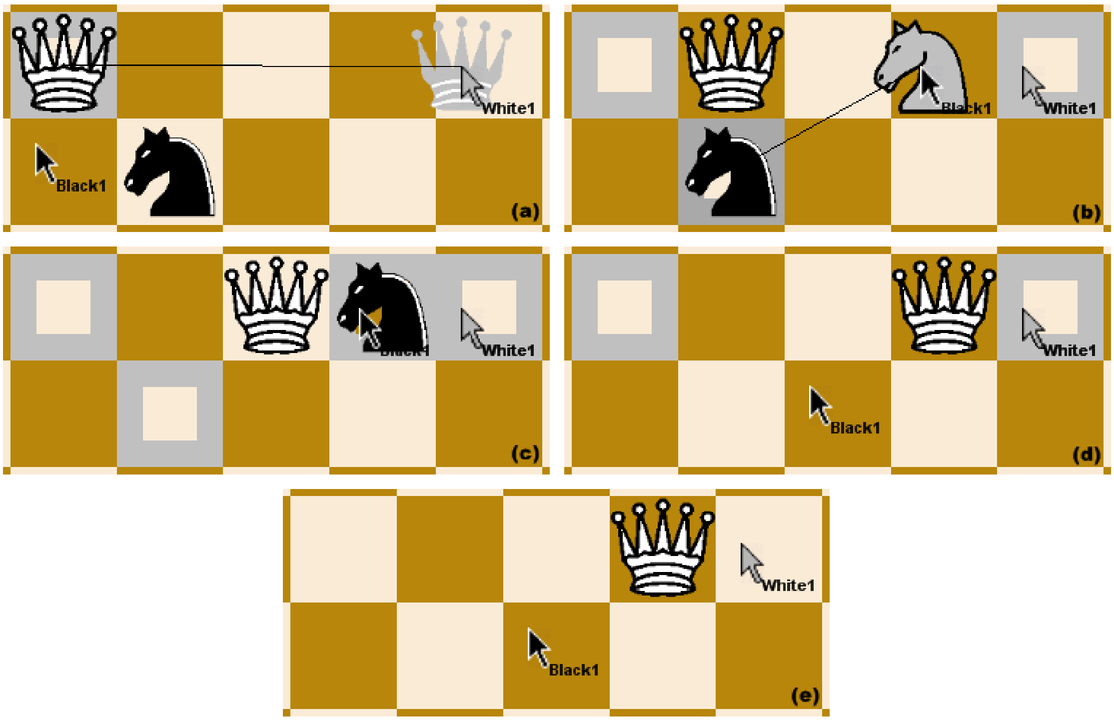

**Figure 3.11.** Simulation of an auto-step movement that got blocked by an opponent chess piece. (a) White1 selects and drags the queen. (b) White1 releases the queen. After successful validation, the queen starts auto-step movement. Black1 selects and drags the knight. (c) Black1 releases the knight into the move path of the queen. After successful validation, the knight moves instantaneously to the end square. The queen takes another step. (d) The queen attempts to take another step but finds an opponent chess piece (the knight). After successful validation, the queen captures the knight. (e) The queen ends its movement.

Third, a friendly or opponent chess piece gets out of the move path in which case the move will continue normally (see Figure 3.12).

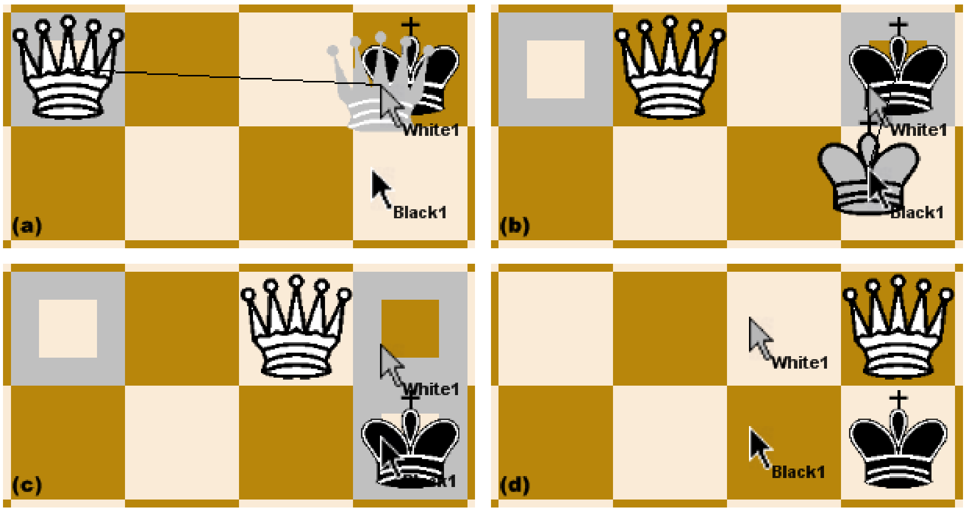

**Figure 3.12.** Simulation of a chess piece getting out the move path of another chess piece. (a) White1 selects and drags the queen to capture the king. (b) White1 releases the queen. After successful validation, the queen starts auto-step movement. Black1 selects and drags the king. (c) Black1 releases the king. After successful validation, the king starts auto-step movement and gets out of the queen’s move path. The queen takes another step. (d) Both the king and the queen reach their end squares and end their auto-step move.

Additionally, the moving chess piece can now be captured by an opponent chess piece while moving. In this scenario, the move is ended and the moving chess piece gets captured and disappears (see Figure 3.13).

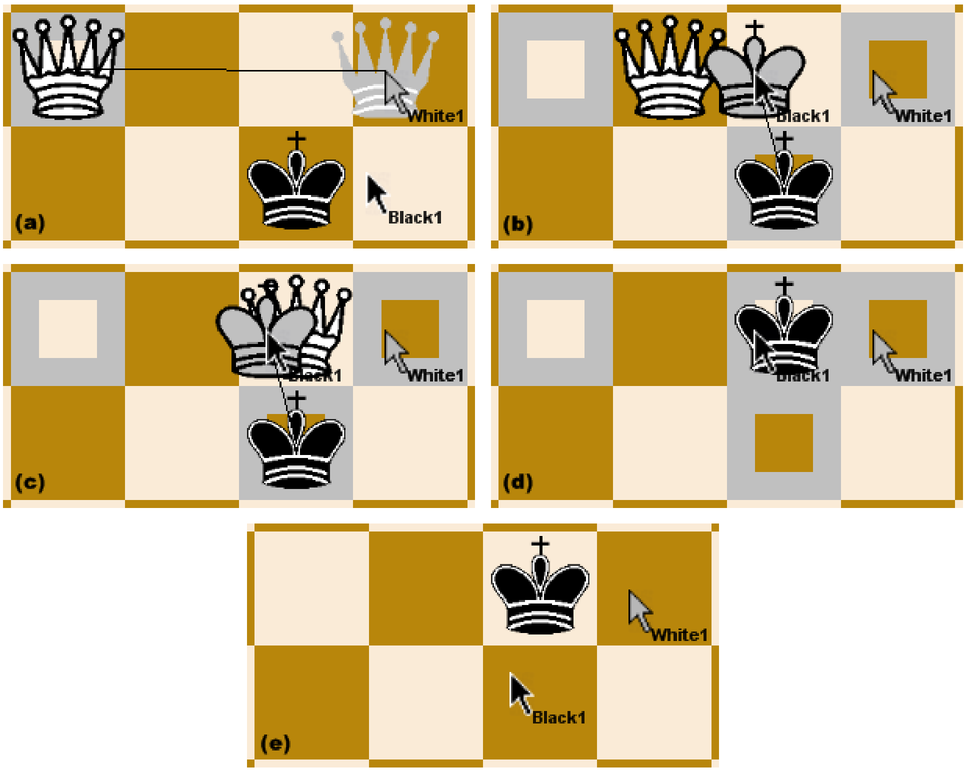

**Figure 3.13.** Simulation of a chess piece that got captured by an opponent chess piece while in the middle of an auto-step move. (a) White1 selects and drags the queen. (b) White1 releases the queen. After successful validation, the queen starts moving. Black1 selects and drags the king and waits for the right time in order to capture the queen. (c) The queen takes another step. The king has its target (the queen) in place. (d) Black1 releases the king. After successful validation, the king captures the queen. (e) The king reaches its end square and ends its move.

The goal in RTChess is to win as many games as possible. The team that has the highest winning score is the winner. To win a single game, one team is required to capture the king of the opponent team. If that happens, the game ends and a three-second break takes place. During the break, a screen appears showing the winner team and the time remaining for the next game to start (see Figure 3.14). After the break, the chessboard is reset and a new game is started.

White Team Wins!!

3

**Figure 3.14.** Screenshot of the screen that appears during the three-second break. It shows the team that won the game and the remaining time in seconds until the next game starts.

## 3.4 Play Experience

RTChess has a totally different play experience than the traditional chess game. This new play experience is caused by the shift from a two-player plan-and-think game (found in the traditional chess game) to a multi-player react-and-anticipate high-speed groupware game (found in RTChess). Several observations were made in nearly a year time of playing RTChess and will be listed in this section divided into two groups: game-play and strategies.

### 3.4.1 Game-Play

*RTChess is fun, fast, and addictive* - Games are extremely short (usually less than 30 seconds). Once the three-second break is over players get quickly engaged in a new game.

*High-volume amount of activity* – when people first start playing, they get overwhelmed by the amount of activity which occur in the game, especially when many people (more than five) play the game at the same time. People often adapt by focusing on the area around the piece they control. After gaining some experience, people start to take notice of the larger area of the chessboard.

*Teamwork is difficult at high speed* – complex team-based strategies have been slow to emerge. Teamwork does occur, but at a more general level. For example, some teams divided their players into offense and defence (see Chapter Four).

*The game is highly social* – game sessions were accompanied by a great deal of talk, both between team members (planning, encouragement, and commiseration) and across teams (mock trash talk, laughing, and congratulations on wins). Players reported that one of the reasons that the game is fun is that it is played with friends.

### 3.4.2 Strategies

*Short-range moves work better than long-range moves* – when a player performs a long-range move and the piece start the auto-step movement pattern, the player cannot select or move any other piece until the current move ends. This limitation made it possible for opponents to attack the ‘helpless’ moving piece while it is moving or at the end square (for example see Figure 3.13). In addition, this limitation locked the moving-player’s interaction for a while; in this high-speed environment, players preferred to have more frequent control over pieces in order to react to the rapid and sudden events occurring in the game.

*‘Rush’ strategies* – a rush strategy occurs when a player moves a chess piece quickly in an unpredicted pattern in order to attack the enemy. A team performing a rush strategy was able to win multiple games before the other team adapted to the new strategy and found a defensive plan. For example, ‘knight rush’ involves moving the

knight as quickly as possible in pursuit of the opponent’s king, using the harder-to-predict motion of the knight to confuse the opponents (see Figure 3.15).

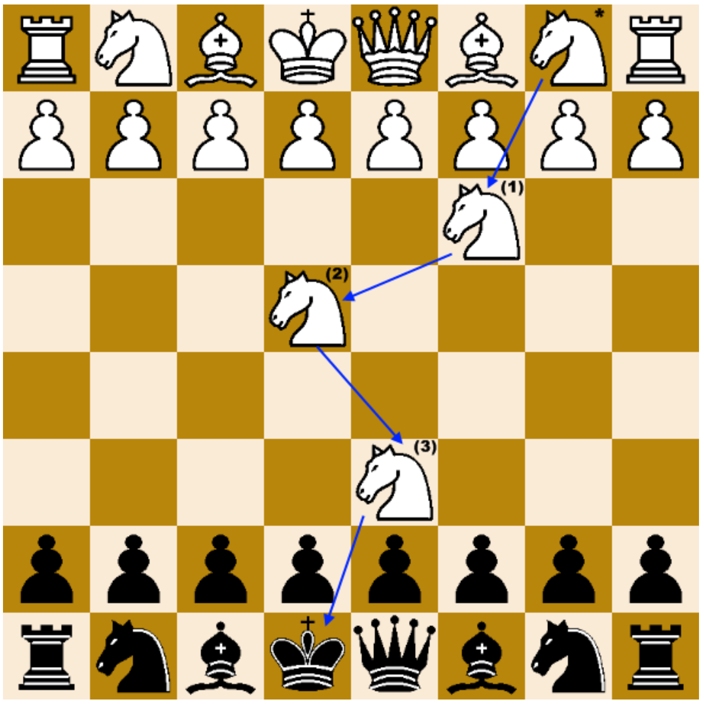

**Figure 3.15.** Simulation of the ‘knight rush’ strategy.

*Recognizable patterns emerge* – players have discovered particular patterns that are successful (e.g., knight rush), and these patterns become known to others over time. Teams then start to switch between the different patterns in order to confuse the opponents. Because of the high-speed activities going on in RTChess, the opponents might not have time to defend against the used pattern. For example, every now and then one team uses knight rush and get the opponents before they get a chance to defend against it.

# CHAPTER 4

# STUDIES OF COORDINATION

In this chapter, I will talk about the exploratory evaluation conducted to investigate HSC in the high-speed groupware environment – RTChess – and to answer basic questions about it. I will describe the details of the evaluation (goals, study design, system, and user task) as well as the details of the pilot and main studies and their results.

## 4.1 Goals

This evaluation aims to solve the research problem given in Chapter One by investigating HSC in a high-speed groupware environment and answering the following basic questions (also listed in Section 1.1) about HSC: (note that these questions will be answered in Chapter Five)

- Does HSC occur in high-speed groupware environments, and how does it occur?
- Are the users aware of HSC?
- What in groupware makes HSC possible and what makes HSC more difficult?
- What is the speed at which users can coordinate best?

## 4.2 Study Design

Different strategies could be used for exploring HSC in groupware such as field strategies, experimental strategies, respondent strategies, and theoretical strategies (McGrath, 1995). Each of these strategies has its own strengths and weaknesses. However, this evaluation – being one of the first steps in exploring HSC in groupware – was designed as an experimental simulation (one of the experimental strategies) where a system (RTChess) that simulates high-speed groupware systems was created and used to explore HSC as it occurs naturally (i.e., no system rules were included to force coordination) between the participants. This method will generate results that give general understanding of different aspects of HSC compared to focused understanding of a specific area of HSC generated in a more controlled experiment.

As this is an exploratory evaluation, many kinds of data were recorded to facilitate exploration of HSC. Four types of data were recorded: the game-play (using screen capture); game-play information such as user interactions, chess piece movements, and game results (saved by the system into log files); participant conversation (using an audio recorder); and participant reflections on the coordination task (using questionnaires).

## 4.3 The System: Modified RTChess

Many RTChess trials were conducted during the development of RTChess (see Chapter Three). In some of those trials when there were many players (up to seven), I noticed that the chessboard became crowded with players and became cluttered with chess pieces and player actions. Since this is an exploratory evaluation, it is more convenient to investigate HSC in its simplest form: HSC between two players.

To simplify RTChess, the number of players was limited to two players per team and, therefore, the number of chess pieces was reduced because two players can only control two pieces at a time and I did not want the chessboard to have many unused chess pieces that occupy important space. Four chess pieces were assigned to each team: king, queen, bishop, and knight (see Figure 4.1); two to be used by team members and two to serve as backup. No duplicate pieces were assigned in order to give players different choices to be used in accomplishing their task. Pawns were excluded because they are limited in movement. The king was assigned because it is essential in determining which team wins the game. The rest of the pieces were chosen because of their unique movement patterns: the knight jumps from start square to end square, the bishop is limited to one square color (I arbitrarily chose the square color on which the bishop can move), and the queen can move in all directions freely. The rook was excluded because its movement pattern is duplicated by the queen. Other than these two changes (limited number of players and reduced number of chess pieces), the simplified version is the same as the original version described in Chapter Three.

In the simplified version, the chess piece positions on the chessboard, the game-play rules, and the task are identical to those found in the original version. It is expected that limiting the number of players and pieces only affected the speed of the game-play in RTChess by becoming slower. However, it is assumed that if the game speed was still fast enough to keep things going on at high-speed, then – in terms of HSC – the results found using the simplified version should be applicable to the original game (this assumption was examined in the pilot study; see Section 4.5).

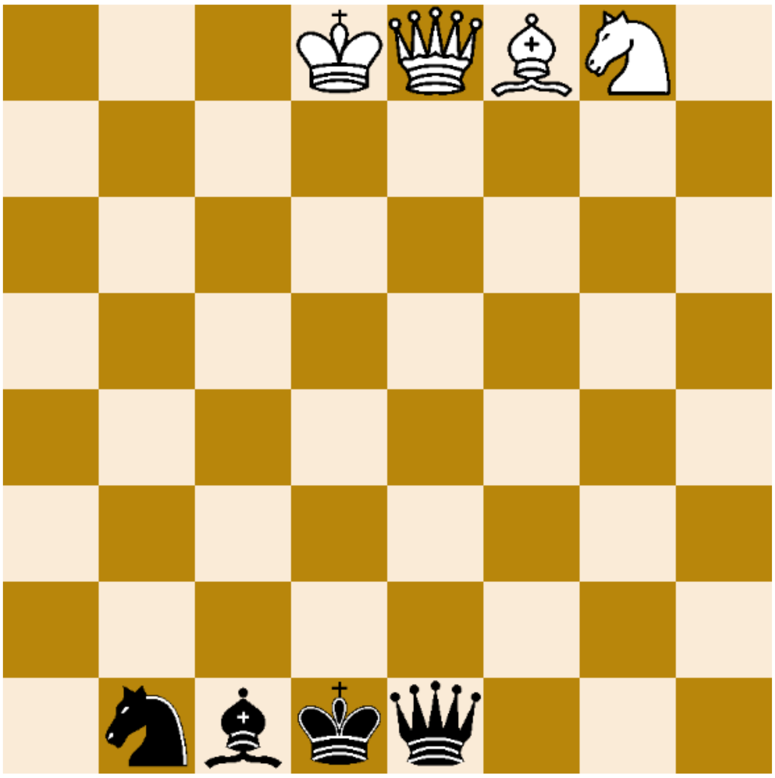

**Figure 4.1.** Screenshot of the modified RTChess chessboard.

## 4.4 Task

The participants were instructed to play the simplified version of RTChess, working together to achieve the highest score. The participants were divided into two teams of two players each. One team controlled the black pieces, one the white. In order to win the game, each team needed to capture the king of the other team. Team partners were instructed to accomplish their task while trying to work with each other as much as possible.

Partners of each team were informed that they had the choice of coordinating with each other; however, they were encouraged to use coordination. Coordination was not enforced because this thesis is interested in exploring HSC as it occurs naturally without any interference. It is hypothesized that using more coordination should result in winning more games. For example, on a basketball team, a player can coordinate their interactions with other teammates resulting in more organized team-play. On the other hand, a player can play without coordinating with others by never passing the ball or by passing the ball to a teammate who is not expecting it. Both cases are valid; however, coordinating with other teammates makes game-play more effective and even more enjoyable. A balanced combination of both is the key to a successful game.

## 4.5 Pilot Study

One thirty-minute pilot study was conducted to test out the system and get quick feedback that might be important in setting up the main study. Four participants, three male and one female, were recruited from the University of Saskatchewan. The participants were divided into two groups of two. The participants ranged in age from 23 to 30 years (mean of 26.75 years). All were regular computer users (minimum of 40 hours per week, mean of 57.5 hours per week), and three of them reported having multiplayer gaming experience (minimum of one hour per week, mean of 3.25 hours per week). All of them reported that they had played the original RTChess previously and that they were good at the game (minimum of 3/5 on a Likert scale).

All players were seated in the same room and could easily talk with each other. However, team members were not able to see each other because of office dividers or because they were facing away from each other.

### 4.5.1 Observations and Feedback

A total of 84 games were played. The maximum game length was 38 seconds and the minimum game length was 3.7 seconds with an average game length of 12.3 seconds and a standard deviation of 7.8 seconds.

First, during game-play, black team players explicitly agreed to stay away from each other because they did not want to get in each other's way. This behavior suggests that it was difficult for the black team players to keep track of each other so they used individual strategies instead. Since awareness is essential for coordination (Neale et. al., 2004), this behavior also suggests that HSC is difficult in the simplified RTChess. Another suggestion from this behavior is that if the system was fast enough to prevent partners from tracking each other, then the simplified version of RTChess is still fast enough to keep activities at high speed. This conclusion confirms that – in term of HSC – the results found using the simplified version are applicable to the original version.

Second, one player complained that they were not able to see their partner because their line of sight was blocked by office dividers. Hearing this complaint, I decided that this impediment should be removed in the main study, giving players an extra channel of communication by allowing them to see each other during game-play.

Third, in one of the games, when the black team only had a king remaining and the white team had two pieces, I noticed that the idle black team player started hovering their mouse pointer over the black king in an attempt to confuse the white team into thinking that they were going to take the piece. This behavior – and the one in which black team players decided to stay away from each other – gave me some ideas about the types of HSC that would be present in the main study.

Moreover, the system crashed towards the end of the pilot study. I had to restart the server as well as all the clients. However, since each game is a self-contained unit in the study (i.e., coordination can be explored between users' interactions made within each game separately), then this crash did not have much effect on my results. Additionally, since games are short (average of 12.3 seconds), then the data for the game in which the crash occurred is a small subset of the entire data set and therefore can easily be ignored without affecting the results.

During the time spent to restart the system, I noticed that the participants started talking and making strategies. This behavior suggested that the main study design be changed so that each session will contain two parts with a small break in-between instead of one continuous session. This break should encourage opportunity for planning.

Finally, I believe that limiting the number of players to two players per team will positively influence this study. The chessboard is a bit small; the more players on the board the more crowded it will become, leading to difficulties in distinguishing behaviors such as partners playing close together or far apart. Additionally, for an exploratory study, having two players per team will help keep things simple by examining only one-to-one partner coordination. Having more than two players per team will open the doors for more complicated coordination patterns such as one-to-many, many-to-one, and many-to-many.

## 4.6 Main Study

A total of five sessions were conducted. The total time to complete a session was approximately half an hour. Each session was divided into two 13 minute parts separated by a four minute break except for the third and fourth sessions. Towards the end of the first part of the third session the system crashed which resulted in a shorter session (20 minutes excluding break time). The fourth session, however, was a bit longer to make up for the time lost in session three (34 minutes excluding break time). As I explained in the pilot study section, removing one game’s data should not have a substantial impact on the results. It was unnecessary to train the participants on how to use the system because all of them already had experience playing the original RTChess. At the start of each session, participants were instructed to try and coordinate with their partners as much as possible. After each session, participants were asked to fill out a questionnaire.

### 4.6.1 Participants

Four participants, all males, were recruited from the University of Saskatchewan. Participants ranged in age from 25 to 35 years (mean of 29.75 years). All were regular computer users (minimum of ten hours per week, mean of 47.5 hours per week), and one reported playing multiplayer games (10 hours per week). All of them reported playing the original RTChess before and that they were good at the game (minimum of 3/5 on a Likert scale).

Participants were divided into two teams of two. The same teams were used for all of the five sessions to help increase experience with playing as a team. This increasing experience should help with identification of plans and HSC patterns.

### 4.6.2 Apparatus

Four computers were used, each with an optical mouse and a keyboard. All the machines were connected via Ethernet cable through a router. The four machines were Pentium 4 with 2GB of RAM. Processor speeds were 2.6, 2.5, 3, and 3 GHz. Since RTChess does not have high system requirements, all of these machines sufficiently executed RTChess. Participants were all seated in the same room. In the first and last sessions, white team participants were seated in front of each other (see Figure 4.2, chairs 1 and 2) and black team participants were seated next to each other (see Figure 4.2, chairs 3 and 4). However, in the second, third, and fourth sessions I switched both teams' positions because I wanted to examine if their positions had an effect on their coordination. We should note that those who were seated in front of each other were able to see and communicate with each other more easily than the others. Communication between those sitting next to each other (chairs 3 and 4) was more difficult because it required participants to do extra effort to be able to communicate with their partners (e.g., switching their visual focus to their partner).

The application was custom built in C# using the gt# Groupware Toolkit (http://hci.usask.ca/research/gt/index.shtml). All game interactions were recorded into log files on the server. CamStudio (http://camstudio.org/) was used to record the game display activities on the server. A Sony Digital Voice Recorder ICD-UX70 device was used to record participant voices in the room.

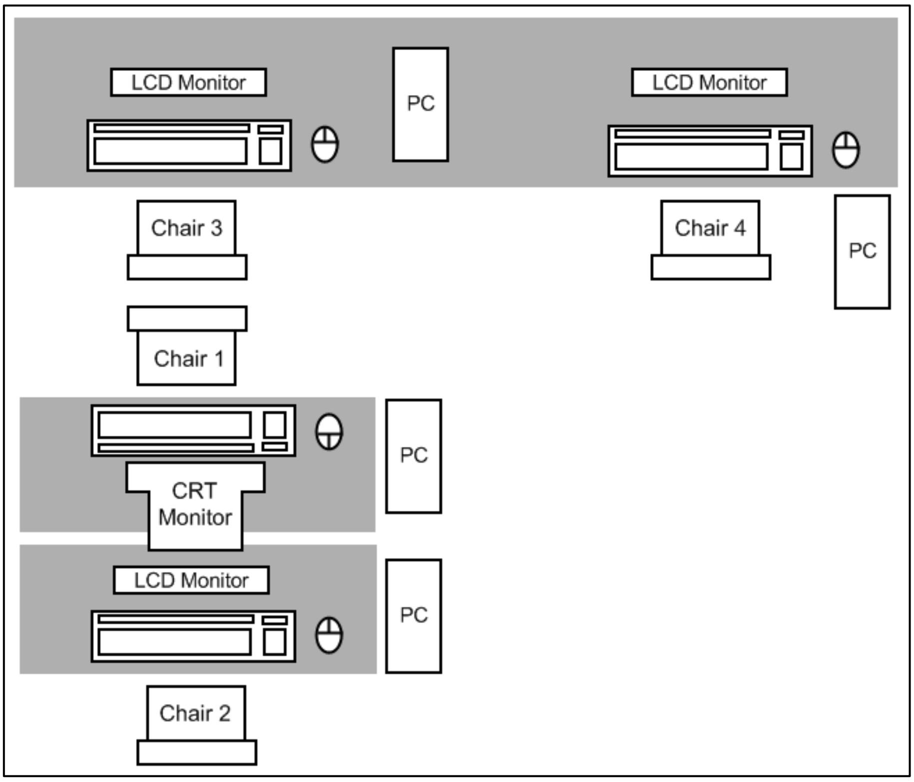

**Figure 4.2.** Diagram of apparatus. One team was seated in chairs 1 and 2, and the other in chairs 3 and 4.

### 4.6.3 Results

The results are organized into four main groups: general findings, monitoring results, log results, and questionnaire results. Monitoring results are those found by monitoring the screen capture and voice recordings for each session. Log results are those generated by analyzing the log files. Questionnaire results are those extracted from the questionnaires that were completed by each participant after each session.

#### *4.6.3.1 General Findings*

A total of 448 games were played in all of the sessions (see Figure 4.3 for number of games played in each session). The maximum game length was 89 seconds and the minimum game length was two seconds with an average game length of 14 seconds and a standard deviation of 11 seconds (note that the values are rounded) (see Figure 4.4). The most frequent game lengths were 6 and 8 seconds (Figure 4.5).

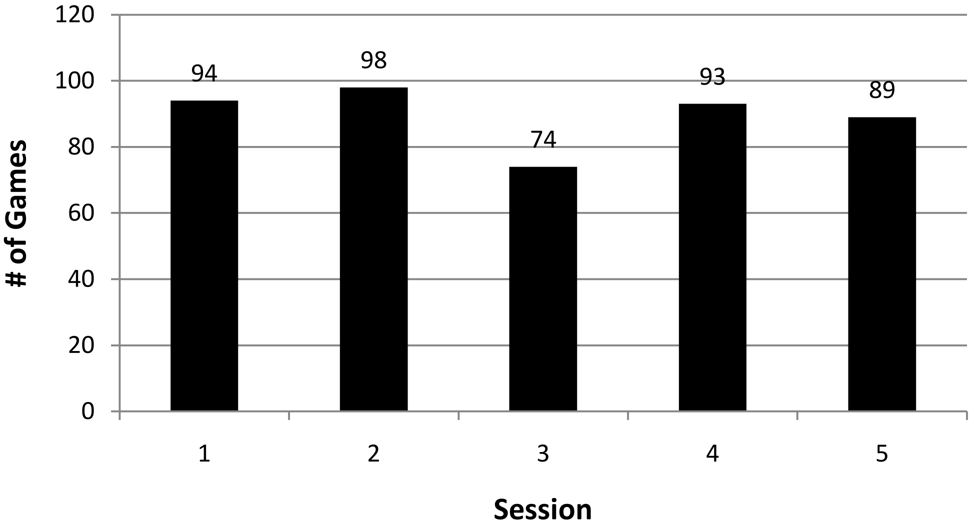

**Figure 4.3.** Number of games played in each session.

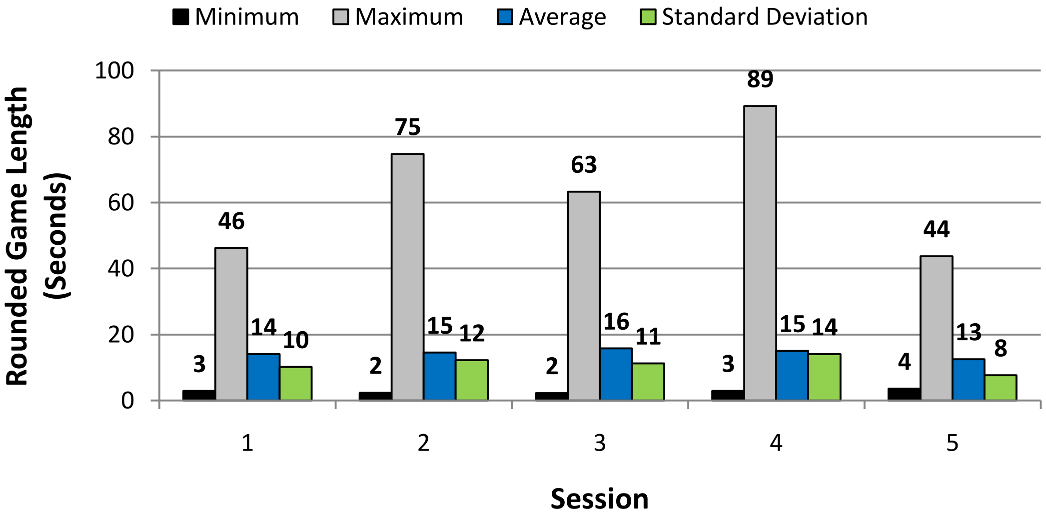

**Figure 4.4.** Game length information for each session.

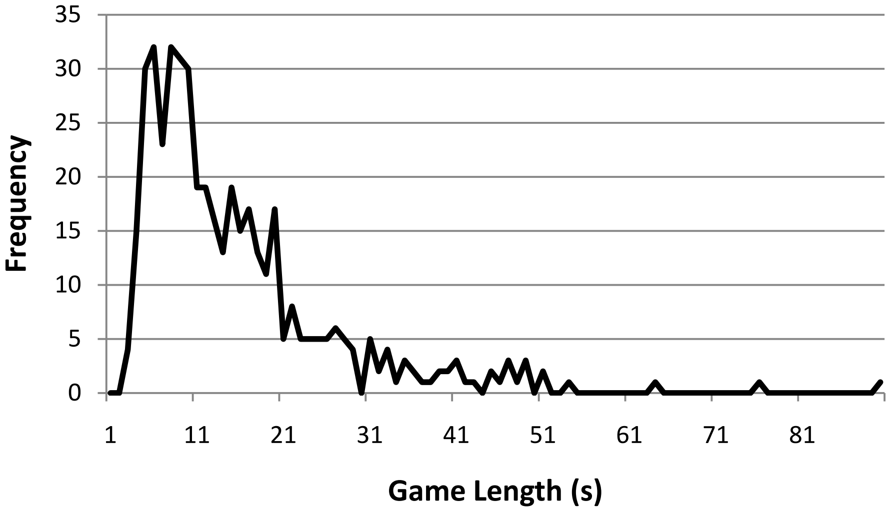

**Figure 4.5.** Frequency of game lengths throughout all the sessions.

In RTChess, many interactions occurred within a single game. Average number of moves made by each team per second, calculated throughout all the sessions, was 1.55 moves per second for the black team and 1.3 moves per second for the white team (see Figure 4.6). These numbers suggest that a game of an average length of 14 seconds has at least 18 move actions made by each team.

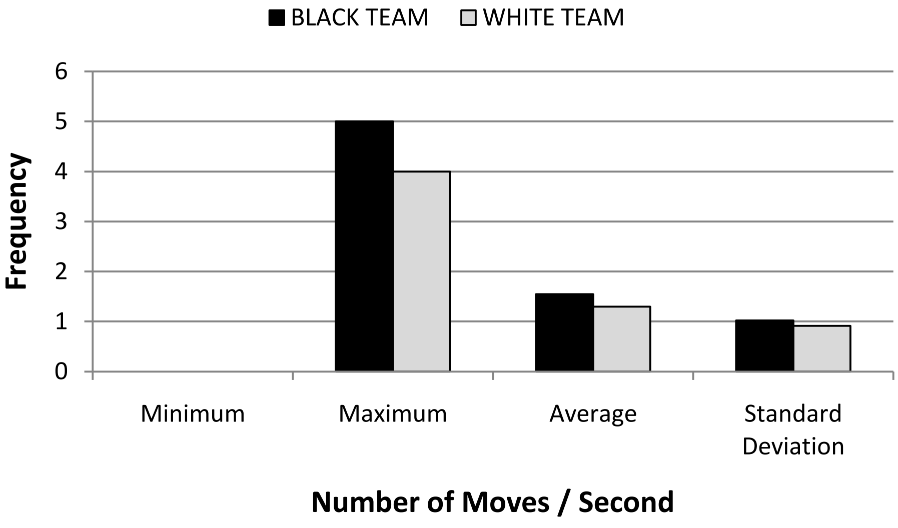

**Figure 4.6.** Statistics of the number of moves per second made by each team calculated throughout all the sessions.

Examining the frequency chart (see Figure 4.7), it is clear that two moves per second and one move per second were the most frequent moves per second performed by the black and white teams, respectively. The number of moves per second gives indication of the pace of the game-play. The more moves performed by both teams, the faster the pace of the game would be. As a result, it can be concluded that there were times during the game-play when the pace was fast (more than 40% of the time, the rate of play was greater than one move per second per team) and other times when the pace was slow (more than 16% of the time, the rate of play was less than one move per second per team).

The general game-play strategy was: at the start of a game, each player started moving one or two pieces, and engaged in one-on-one fighting with an opponent. Once their initial piece was captured, they looked for other pieces to control and engaged their opponent again. If no new pieces were found, then one player became idle and the game continued with the rest of the players. For the purposes of these results, I will refer to the white and black team players as white1, white2, black1, and black2. Offensive and defensive game-play strategies were established by each team. A player following an offensive strategy would not control the king and would attempt to attack the opponent's king. Alternatively, a player following a defensive strategy would protect the king and would stay away from attacking the opponent, unless there were no more pieces left for the offensive partner to control.

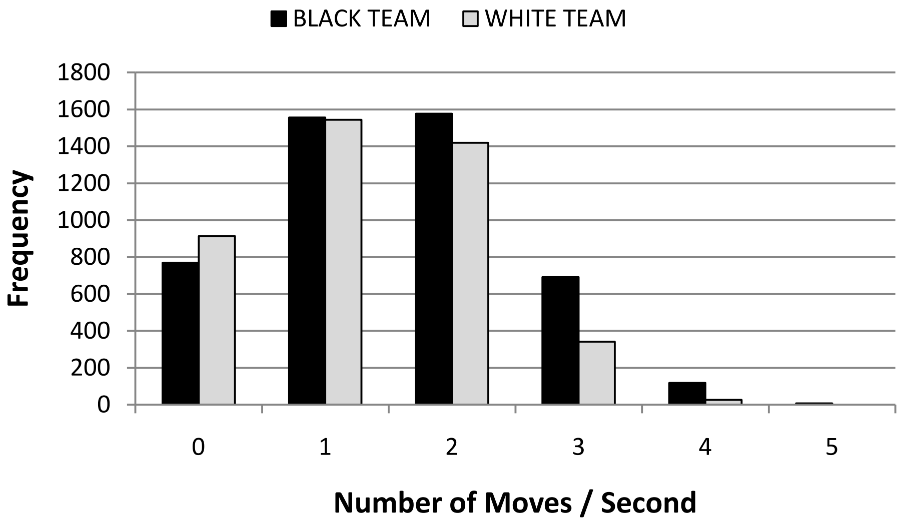

**Figure 4.7.** Frequency of the number of moves per second made by each team calculated throughout all the sessions.

The number of concurrent players on the chessboard was important in order to determine if a partnership was established between team members or not. For example, when two players were idle (one in each team), then only one player per team was playing and, thus, no partnership was established at that time. Such cases were ignored when looking for HSC because there were no partners to coordinate with. On the other hand, when one player was idle, then a partnership was established in one team only. Figure 4.8 summarizes, for all sessions, the percentage of the session time when two, three, or four participants were playing concurrently.

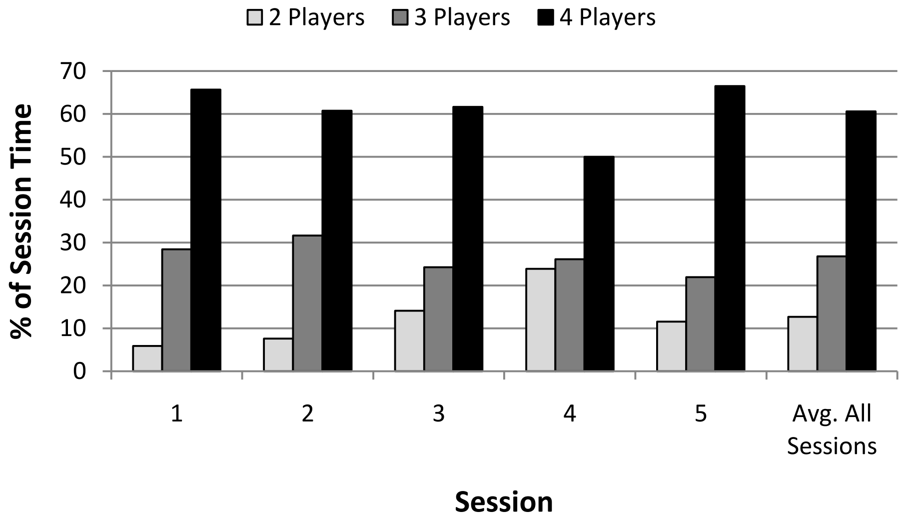

**Figure 4.8.** Percentage of session time when two, three, or four players were playing concurrently.

The black team won more games than the white team (see Figure 4.9). Player black1 reported having the highest multiplayer-game experience amongst all participants (10 hours per week); this might be one of the reasons for the black team's better performance. In addition, black1's comments about the black team's game-play show that they used more coordination than the white team. For example, black1 said that when they noticed an opponent chasing after their partner, they would attack that opponent, knowing that the opponent was busy with their partner and would not notice their attack.

It is notable that the white team reported negative feelings about their performance. White team players attributed these negative feelings to losing the game frequently. Player white2 even slapped their desk many times – presumably out of frustration. It is important to note that the white team's negative feelings might have affected their questionnaire responses. However, it is clear that the white team's performance was getting better through time and experience. In fact, by observing the game-play, I noticed that the white team showed clearly enhanced performance in the last two sessions compared to previous sessions.

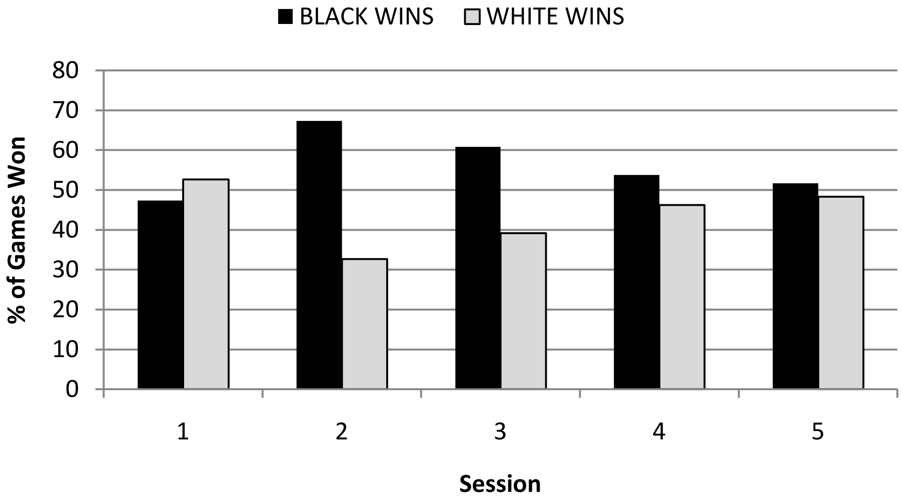

**Figure 4.9.** Percentage of games won by each team for each session.

#### *4.6.3.2 Monitoring Results*

Ten video recordings (two for each session) and five audio recordings (one per session) were generated in this study. After merging each video with its corresponding audio (using Microsoft Movie Maker), I reviewed each resulting recording two times: once to focus on the black team's actions (voice communication and game-play), and once to focus on the white team's actions.

For each session, I recorded every interaction that suggested HSC between partners (see Figure 4.10). More than 50% of these interactions occurred when there were only two or three concurrent players on the chessboard. Once they were recorded, the interactions were summarized into three categories: mouse movement, piece movement, and voice communication.

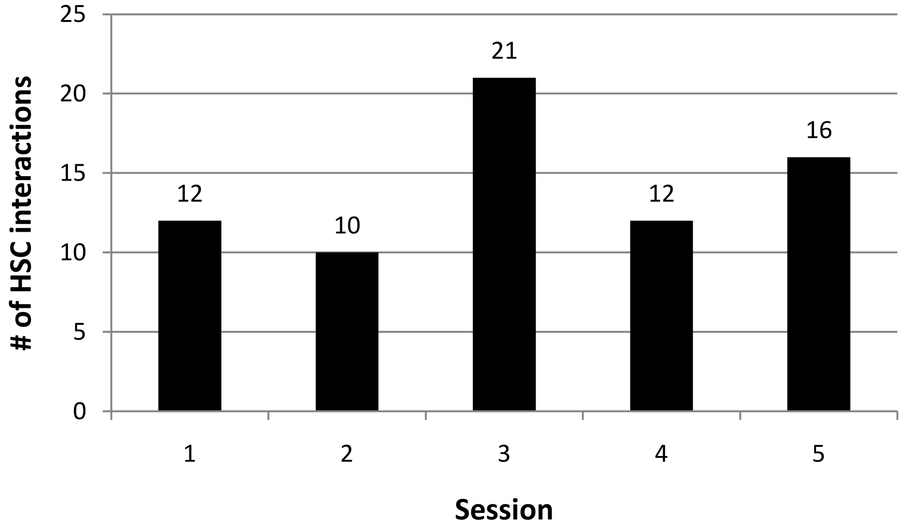

**Figure 4.10.** Number of interactions that suggest HSC which were found in each session by monitoring the screen capture and voice recordings.

- *Mouse Movement*

In many cases the game would reach a state with one piece in one team (king) and two pieces in the other (king and another piece). Specifically, in many of these cases the white king and the black king and bishop were left on the board (see Figure 4.11). Occasionally player white2 – who did not have any more pieces to move – attempted to help their partner by hovering their mouse over the black king in an attempt to distract their opponent. This behaviour suggests the existence of coordination between white team players because player white1 (controlling the white king) should have became alerted to any opportunities created by their partner's actions. In other words, the two partners were working together.

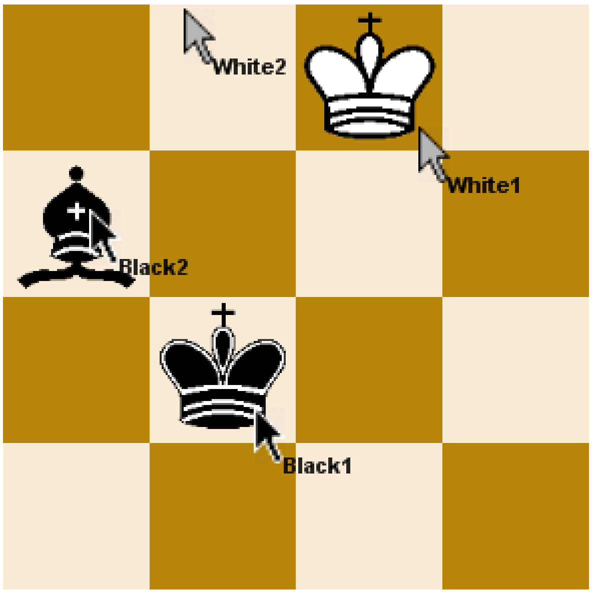

**Figure 4.11.** Example of the situation where players end up with two chess pieces on one side and one chess piece on the other (in this case black king and bishop as well as white king). One player (white2 in this case) has no more pieces to control.

- *Piece Movement*

Having the same situation – white king and black king and bishop – mentioned above (see Figure 4.11), black team players performed some interactions that suggested HSC between them. First, sometimes black1 and black2 chased the white king at the same time. They both moved in similar directions and then they moved closer together (see Figure 4.12). To succeed in this chase, they needed to keep track of each other in order not to collide and they needed to keep scanning for opportunities caused by their partner.

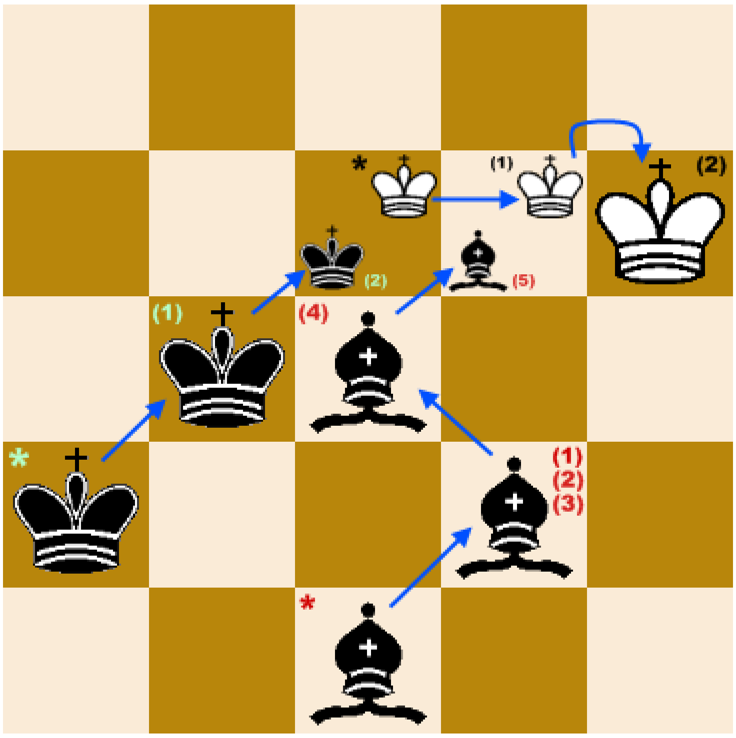

**Figure 4.12.** Black team chase. An example situation that occurred where the black team chased the white king. (*) is the start position. (n) is the timestamp at which the movement took place. Arrows show the direction of movement. Player's cursors are removed for clarity.

Another example of black team interaction that suggested HSC again occurred in the situation described above – with the white king, the black king, and the black bishop left on the board. The black bishop moves only on the light squares. The white king acknowledged this limitation by staying on dark squares rendering the black bishop's attacks useless. As a result, the black bishop started moving quickly around the white king in an attempt to distract them. While doing so, the black bishop made sure to quickly return to the square next to the black king in order to have protection and to serve as bait (see Figure 4.13). After staying for a short while between the black and white kings, the black bishop went for another round. The conversation that occurred in this situation also suggested HSC between black team partners. In this situation, black2 said to their partner: "Ready? Set the trap!" and then they said to the opponent: "go for it" – goading them to capture the black bishop, thereby falling into the "trap".

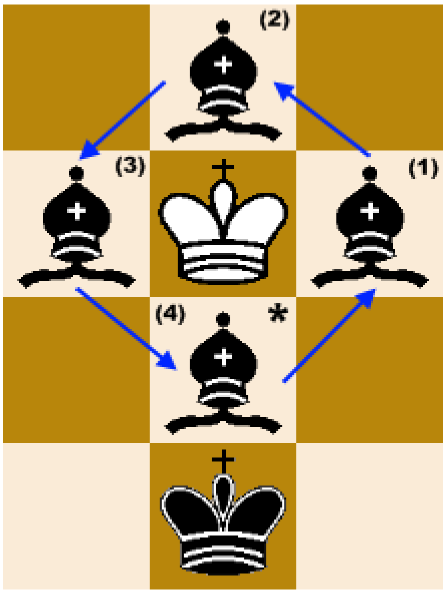

**Figure 4.13.** Black bishop's loop. An example situation that occurred in the study where the black bishop made loops around white king. (*) is the start position. (n) is the timestamp at which the movement took place. Arrows show the direction of movement. Player's cursors are removed for clarity.

Other situations and interactions occurred that also illustrate how team members used piece movements to coordinate their interactions. For example, in one situation the white team players were moving close to each other and they were about to move to the same square. White1 was moving the king and white2 was moving the knight. White1 saw the ghost piece of their partner and realized that they were trying to move to the same square, so white1 moved away from that square in order to avoid a collision. This behaviour shows how white1 used information about their partner's piece movement to decide their next move.

Another situation occurred in which player white2 saw that their partner was in a one-on-one battle with the black king. White2 realized that the black king would be busy focusing on their partner (white1), so they advanced from behind and successfully attacked the black king. Such behaviour shows that players try to take advantage of chances created by their partners and that they use visual information about their partners to plan their movement.

Finally, player black1 who had the highest experience in multiplayer-games (they reported playing 10 hours per week) was able to control two pieces simultaneously. They would move the first piece, then move the second one, and then go back to the first one. While moving one piece, they would keep an eye on the second. If any danger came close to the second piece, they would react accordingly. While this behaviour does not reveal HSC, it suggests that experience counts. It also suggests that even in a high-speed environment such as RTChess, players with higher experience are less likely to be overwhelmed by the fast pace of the system. It was not until the last session that white1 was able to control two pieces simultaneously.

- *Voice Communication*

Talking occurred at different times throughout a session: in game, between games, and between two parts of a session. In-game talk was usually short and the speaker did not usually wait for any reply from their partner. Examples are “don’t move in that spot” or “I’ll take the queen”. Sometimes partners agree on doing something and then they actually do it. For example, black team made a lot of switches – i.e., the players either swapped their pieces together, or they switched control over a single piece (see Section 4.6.3.3) – during the last two sessions. One partner would say something like “let’s switch king and queen” then their partner would reply with “ok go for it”.

In other situations, especially when one player did not have any piece to move, the idle player would aid their partner by giving directions and/or warnings. For example, a player would tell their partner “watch the knight” or “chase the knight”.

Between-games talk usually reflected on what just occurred in the last game. For example, one player said to their partner “I will be offensive” after having a difficult time being defensive in the previous game. Between-game talk was usually longer than in-game talk and would sometimes extend to the beginning of the next game.

Talk that occurred between parts of a session was usually longer and reflected on the part of the session just played and the strategy that was followed. Players, even opponents, engaged in conversations explaining why they moved in certain ways and how they used the information on the chess board. Partners also engaged in planning and strategizing conversations.

In terms of HSC, since coordination requires team members to coordinate both their activities and communication (Neale et. al., 2004), not being able to have longer conversations during game-play suggests that it was difficult to communicate and, therefore, difficult to do HSC in RTChess. This is supported by the fact that when the players had more time, they started talking in greater detail, reflecting, planning, and strategizing.

#### *4.6.3.3 Log Results*

Since this is an exploratory study, it was important to log all the actions and events that occurred during the game and then conduct a thorough investigation of these actions and events to look for signs of HSC. Specifically, the data logged consisted of player actions (mouse movement, select action, and move action) and game events (initial position of chess pieces, piece position change, piece capture, game start, and game win events).

In many cases, the game would reach a state where only one chess piece is left for either or both teams; which allowed for only one player of that team to be playing at that time. In other words, in some parts of the sessions the partners were playing concurrently using two different chess pieces while in other parts the partners were not. Even though when only one team member was playing, the other team member could still perform some actions in an attempt to help their partner such as hovering the mouse pointer (e.g., see Section 4.6.3.2) or talking across the room to give warnings. This analysis, however, is interested in those times when the partners were playing concurrently.

To address the concurrency issue, a percentage was calculated that represents the amount of time when both team members were playing concurrently (see Figure 4.14). From here on, any measures calculated between team members are bounded by these concurrency results. For example, if a certain HSC measure was found to occur 80% of the time between white team members in the first session, this means that the HSC measure occurred 80% of the time when both team members were playing concurrently – about 60% of the time as indicated in Figure 4.14 – in the first session.

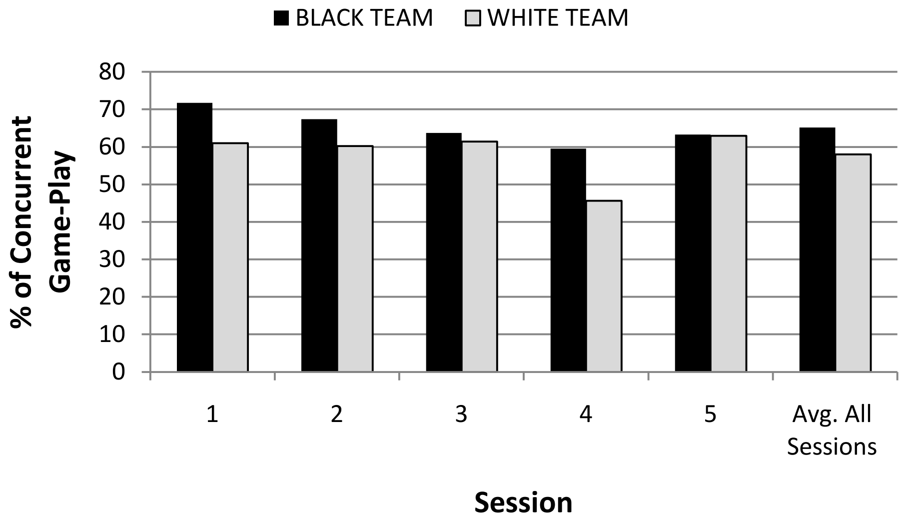

**Figure 4.14.** Percentage of time in each session when both team members were playing at the same time.

To examine HSC in RTChess, a set of three measures were examined between partners of each team. Each correlation will be defined, have its relationship to HSC presented, and have its study results presented. These correlations are:

- *Operating distance* – are partners close together or far apart?
- *Operating areas* – are partners playing in the same chessboard area or different areas?
- *Switching control of chess pieces* – are partners switching control of chess pieces during game-play or not?

- *Operating Distance*

This measure determines if partners were operating (moving pieces) close to each other or far apart. For the purposes of this measure, the chessboard rows and columns were numbered with indices so that each square has a row and column coordinates (see Figure 4.15). Using these coordinates, Euclidean[2] distance was measured between partners of each team during game-play. Given these indices, the minimum distance possible is 1.0 and the maximum is 9.899.

---

[2] Euclidean distance is measured between two positions (x1, y1) and (x2, y2) using the formula $\sqrt{(x2-x1)^2 + (y2-y1)^2}$.

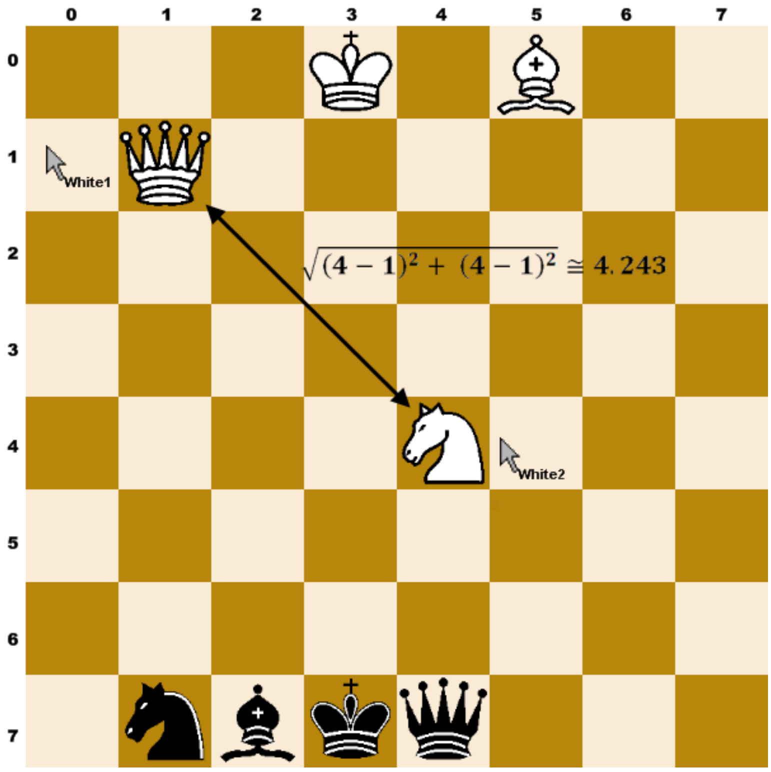

**Figure 4.15.** Chessboard showing the chessboard row and column indices used to calculate Euclidean distance between partners. An example distance was calculated between the white king and the white knight.

For the purposes of this measure, the position where a player was operating during game-play was considered to be at the position of the last piece the player had moved. For example, if player white1 moved a piece from (0,0) to (0,2) and the last piece moved by player white2 was at (1,1), then – considering the auto-step movement described in Chapter Three – the distance between white1 and white2 would be calculated between the following pairs of positions: (0,0) – (1,1), (0,1) – (1,1), and (0,2) – (1,1).

Figure 4.16 summarizes distance statistics between partners of each team calculated throughout all the sessions. Since the maximum distance possible is 9.899, the average distance found for both teams (~ three chessboard squares) indicates that partners had been operating close to each other. However, considering that the chessboard is only 8x8 squares and that the most frequent distance was four (see Figure 4.17), it can be concluded that partners of each team were not too close to each other. The next measure (operating area) will provide more information about whether team partners were operating close to each other in the same area or not.

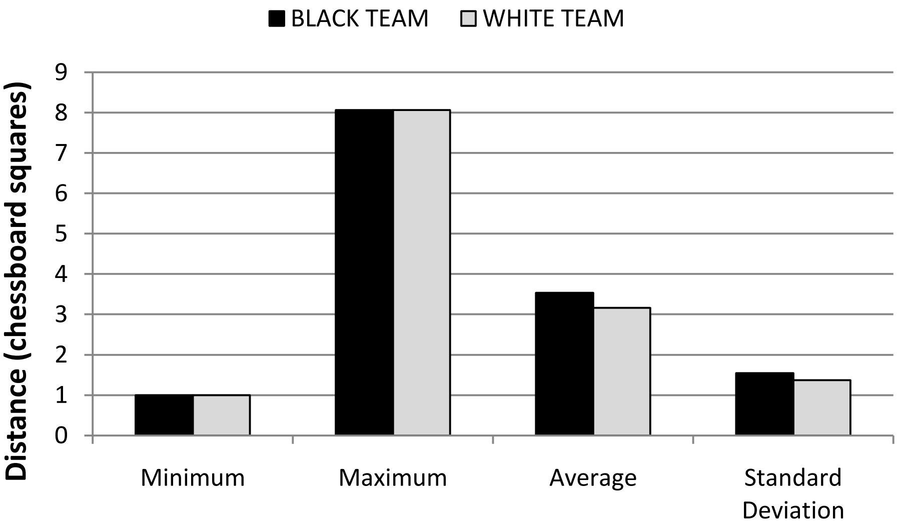

**Figure 4.16.** Distance statistics between partners of each team calculated throughout all the sessions.

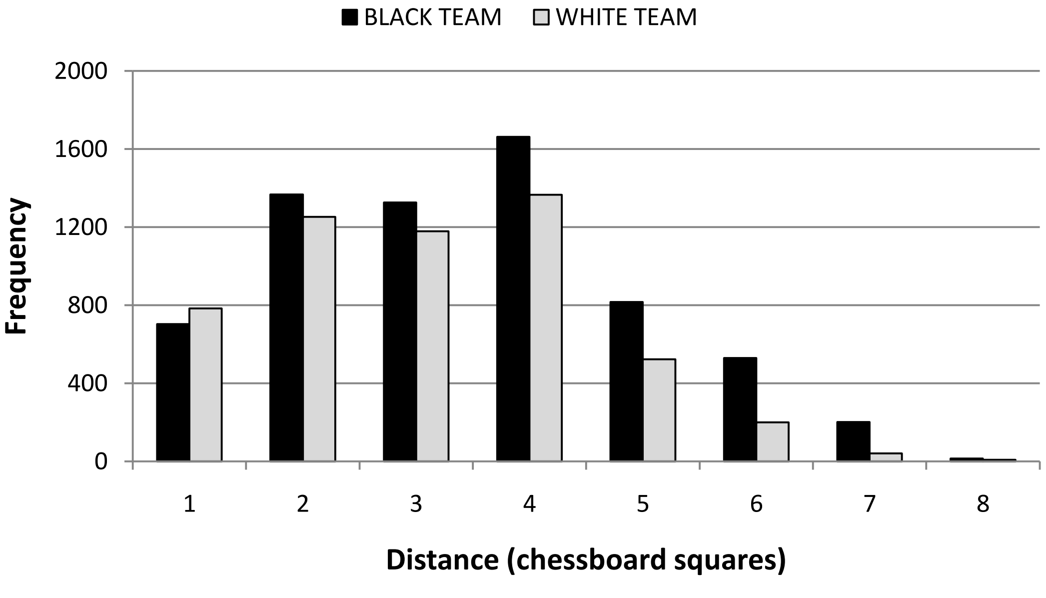

**Figure 4.17.** Frequency of the rounded distances between partners of each team calculated throughout all the sessions.

As discussed earlier, different numbers of concurrent players were operating at different times during the sessions. Therefore, it would be interesting to examine if the distance between team partners was affected by the number of concurrent players on the chessboard. The distance between team partners was measured again but this time it was grouped based on the number of concurrent players operating at the time when the distance was measured (see Figure 4.18). As it can be seen, the number of concurrent players on the chessboard had a slight effect on the distance between partners. When there were four concurrent players on the chessboard, partners played farther from each other (mean of 3.69 and 3.25 squares for black and white teams respectively) than when there were three concurrent players on the chessboard (mean of 3.10 and 2.64 squares for black and white teams respectively). To further support this result, the frequency chart also indicates that, for both teams, larger distances (such as distances of seven and eight chessboard squares) either did not occur during times when three concurrent players were on the chessboard or occurred but in less frequency (see Figure 4.19).

In terms of HSC, playing at very close distance (one or two chessboard squares) was difficult for both teams because it required each team member to do more HSC with their partner in order to avoid collisions and mistakes. By observation, it was clear that collisions and mistakes in moving a chess piece often causes a player to lose that piece. On the other hand, playing in average (three squares) or far (six squares) distance was easier for team members because players did not need to keep track of their partner’s exact actions; instead, players only needed to keep track of the general area where their partner was playing. Additionally, playing in close distance when three concurrent players were on the chessboard suggests that the partners were able to perform more HSC while attacking the last piece of the opponent team – the king. Moreover, playing in far distance when four concurrent players were on the chessboard suggests that each partner was busy in their own fight with opponent pieces.

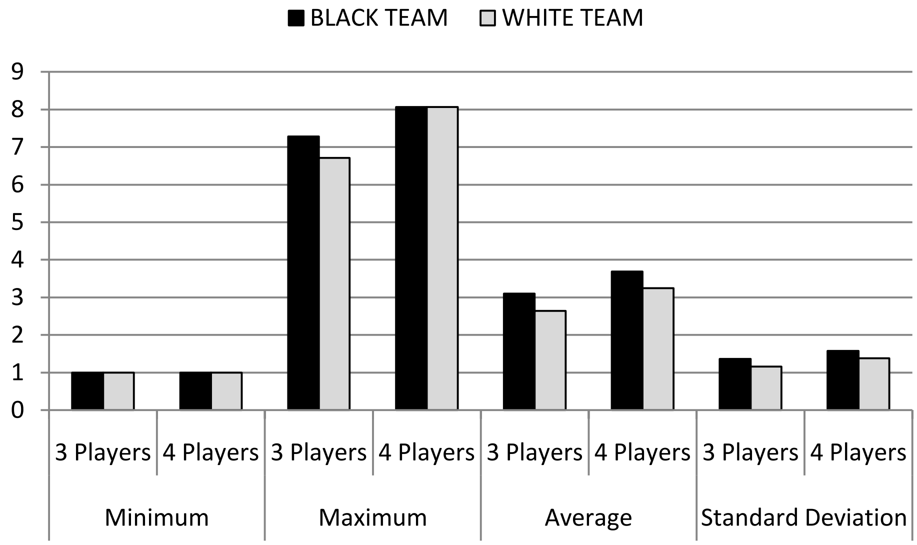

**Figure 4.18.** Distance statistics between partners of each team calculated throughout all the sessions and grouped by the number of concurrent players on the chessboard.

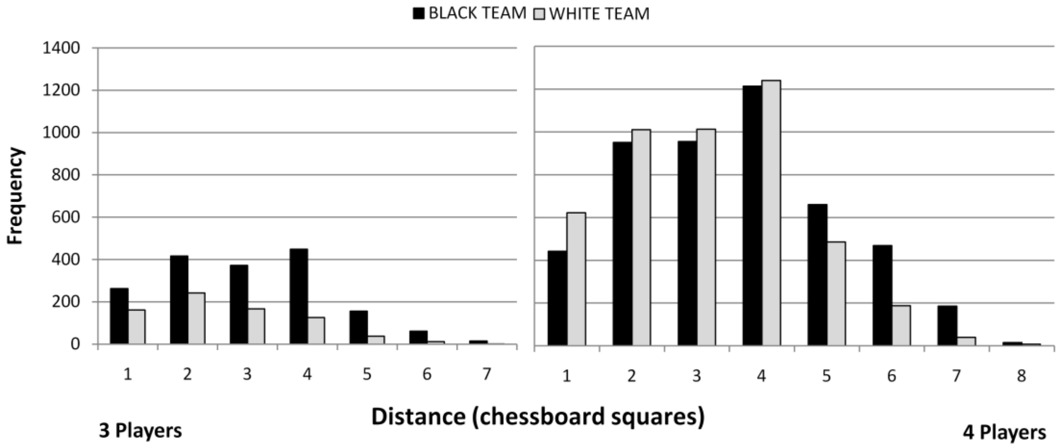

**Figure 4.19.** Frequency of the rounded distances between partners of each team calculated throughout all the sessions and grouped by the number of total concurrent players.

- *Operating Area*

This measure determines if partners were operating (making moves) in the same area or different areas of the chessboard. During the pilot study, black team partners mentioned that they wanted to stay away from each other in order to avoid collisions; which triggered me to include this "operating area" measure for the main study. In addition, this measure should work as a complement for the previous measure – operating distance – by clarifying whether the average distance of three squares meant that partners where operating close to each other in the same area or not.

For the purposes of this measure, the chessboard was divided into four virtual quadrants (see Figure 4.20). Given the average distance of three squares (see previous measure), a quadrant size of 4x4 seemed most appropriate. In addition, a 4x4 quadrant is the only size that covers the whole chessboard area while, at the same time, giving partners some room to move in the same area close to each other. Other sizes are either too small (e.g., 2x2), too big (e.g., 8x8), or do not cover the whole chessboard area (e.g., 3x3 or 5x5).

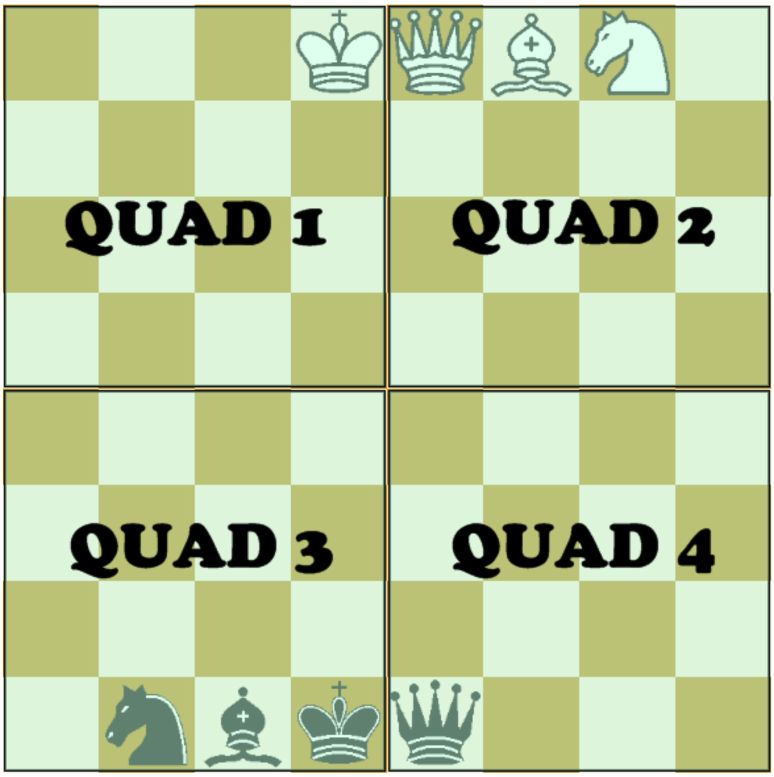

**Figure 4.20.** Chessboard showing the four virtual quadrants used in the "Operating Area" measure.

As it was described in the previous measure, each player's position was determined by the position of the last chess piece moved by that player. The player's position was used to determine the quad in which the player was operating. In the end, a percentage was calculated which indicates the amount of time partners operated in the same quad. A high percentage means that team partners generally operated in the same quad during the session, while a low one means that they were operating in different quads. Both of these cases suggest the existence of HSC between team members because in both cases partners needed to keep track of each other and needed to use information about their partner to plan their next move. An in-between percentage could indicate that either the partners were just playing without a strategy or they might have switched strategies during game-play.

Figure 4.21 summarizes, for all sessions, the percentage of time partners of each team played in the same quadrant. It can be seen that both teams had low percentages (average of 22% and 17% for black and white teams respectively) which indicates that, generally, partners of each team were operating in different quads. In addition, the average distance between the partners (three chessboard squares) eliminates the possibility that partners were playing on the inner edges of adjacent quadrants, in which case they would be operating close to each other. Moreover, these results confirm that the average distance between partners (found in the previous measure) did not mean that partners were considered close to each other.

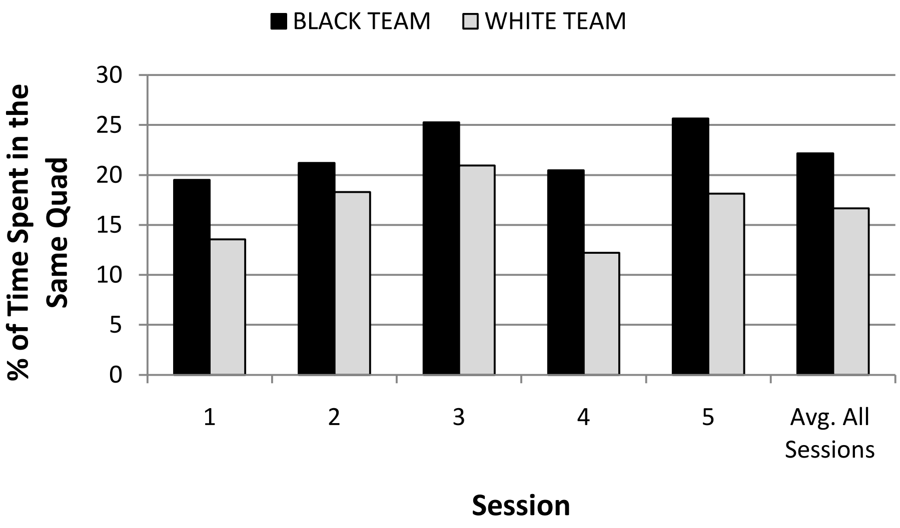

**Figure 4.21.** Percentage of session time partners of each team spent in the same quad.

In addition to the results so far, it would be interesting to see if the number of concurrent players on the chessboard had an effect on whether partners played in the same quad or different quads. Therefore, the percentage of partners operating in the same quad was calculated again and then grouped based on the number of concurrent players (see Figure 4.22). As it can be seen, partners did spend some more time playing in the same quad (24% for the black team and 30% for the white team) when less number of concurrent players were operating on the chessboard, the behaviour of which was also observed during game-play.

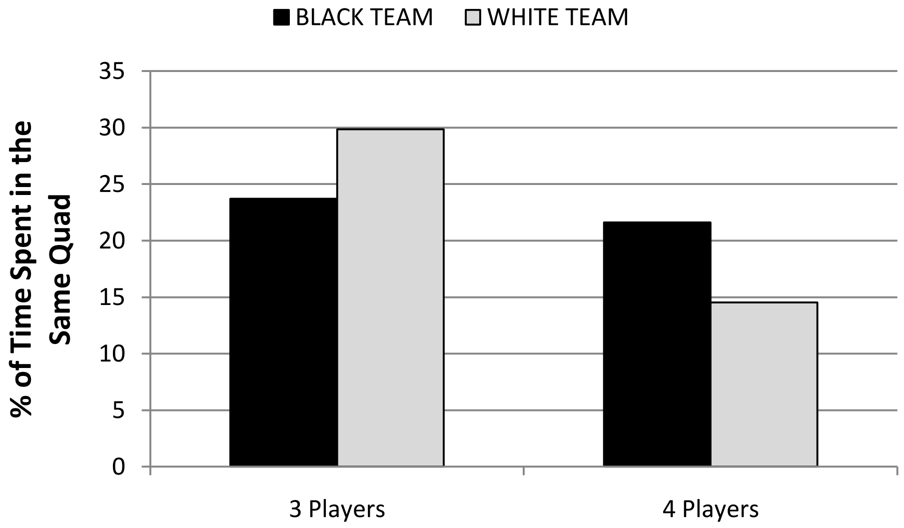

**Figure 4.22.** Percentage of time partners of each team spent in the same quad calculated throughout all the sessions and grouped by the number of concurrent players.

During the sessions, each team divided the game-play roles between the team members. As a result, players white1 and black1 played defensive most of the time, while players white2 and black2 played offensive most of the time. It would also be interesting to see whether partners divided the chessboard into two horizontal halves during game-play (e.g., as in football games where offensive player go to the front and defensive players stay in the back). In addition, since players move upwards or downwards in order to attack their opponents, partners might have divided the chessboard into left and right sides. As a result, it would also be interesting to see if the partners divided the chessboard into two vertical halves as well (see Figure 4.23).

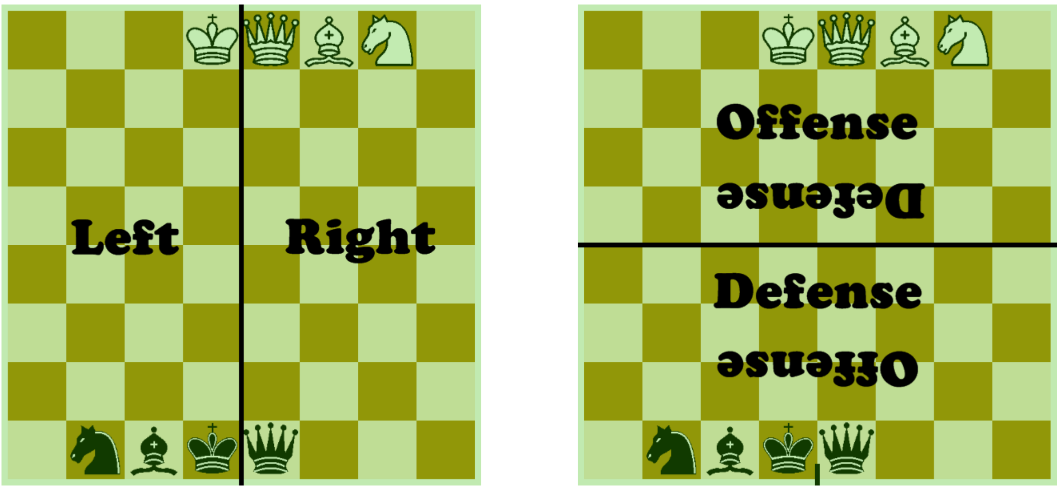

**Figure 4.23.** Chessboards showing the two vertical and the two horizontal halves used in further analysis for the "Operating Area" measure.

Figure 4.24 summarizes the percentage of time, from all the sessions, partners of each team spent operating in the same area (vertical or horizontal) grouped by the number of concurrent players. As it can be seen, all of the percentages are almost in the middle (about 50%), which does not give us any useful results. However, the only exception is the white team's percentage (30%) which suggests that white team players had some sort of vertical division of the chessboard when there were four concurrent players on the chessboard.

In terms of HSC, playing in different chessboard areas suggests that HSC at close proximity is difficult to achieve and, therefore, the partners played away from each other to prevent collisions and mistakes. In addition, playing in the same chessboard area when fewer players were operating concurrently suggests that partners engaged in more HSC interaction while attacking the last opponent piece – the king. The flip side of this result is that when more concurrent players were on the chessboard, partners were not able to perform much HSC and, therefore, stayed away from each other and either coordinated at a distance or just followed individual strategies.

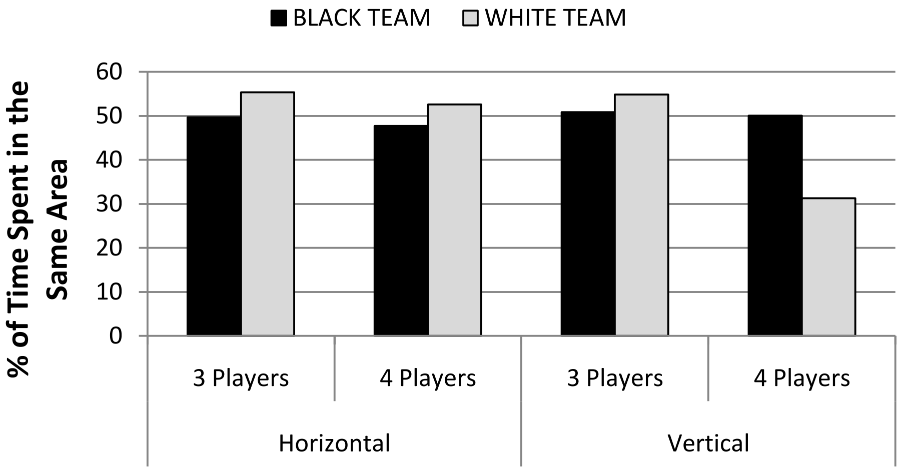

**Figure 4.24.** Percentage of time partners of each team spent in the same area (horizontal or vertical) calculated throughout all the sessions and grouped by the number of concurrent players.

- *Switching Control of Chess Pieces*

This measure indicates how frequently partners switched control of pieces during game-play. The idea for this measure was triggered by observing players switch control of pieces frequently during the game-play. Switching control of a piece occurs when one player releases control of a piece and then their partner takes over control of the same piece. This measure considered switching control of only a single piece between partners. The case of partners swapping control of two pieces around is actually formed by two single piece switches (one on each side of the swap). The importance of this measure lies in the fact that the existence of piece control switching is, by itself, an indication of the existence of HSC between the team members because partners would not be able to perform a switch without some sort of HSC with each other.

By observation, switching happened in three different methods: explicit switching, implicit switching, and switching-by-mistake. In the explicit method, one partner would ask for the switch and wait for confirmation, then the other partner confirms and initiates the switch (e.g., by saying "Okay, go!"). In the implicit method, one player would take control of a piece that was being controlled by their partner without informing the partner. For example, it was observed that player white2 took control of white1's piece after observing an opportunity to capture an opponent piece that moved next to white1's piece. After failing to capture the opponent piece, white2 returned back to their original piece and white1 continued normally. Additionally, in the implicit switching method, one player would let go of a piece knowing that their partner was seeking control of that piece. For example, it was observed only with the black team that player black2, who played offensively most of the time, would give an indication to their partner that they had lost their second piece (e.g., by saying "I lost my piece" or by making sounds like "Ooooh!"). When player black1, who played defensively most of the time, noticed that their partner had lost their second piece, player black1 would let go of their piece – an offensive piece – and would take control of the defensive piece – the king. At the same time, player black2 would take control of the last offensive piece. A switch-by-mistake occurred when partners played too close from each other and, at the same time, things got crowded because of opponent attacks. In such a situation, one partner could mistakenly pick up their partner's chess piece. Switching-by-mistake occurred rarely during the game-play and was observed in both teams. After a switch-by-mistake would occur, partners would laugh a lot, would ask questions such as "what happened?", and would continue playing with the new pieces they got.

To find piece control switches, partners' actions were examined within short-length timeframes. For example, suppose player white1 lets go of the white queen and then player white2 takes control of the white queen, shortly after. What is the timeframe within which both actions occur such that the actions are considered HSC? By observations, the timeframe length should be variable because it depends on the game's pace. When the pace was fast, players moved their pieces quickly. When the pace was slow, players moved slowly, even paused at times to think about what to do next. Consequently, the timeframe for the slower game pace should be longer than that for the faster game pace. In this study, high-speed actions are of most interest; therefore, short-length timeframes were used to find piece control switches. Specifically, the length of timeframes used was one second. This timeframe length (one second) was chosen considering the goal of this study in finding split-second coordination decisions made by the players of RTChess. In addition, to find piece control switches, two move actions (one per partner) were required. Once a player's action was found, a search for another action performed by the partner was started. The search process would start from the point where the first-partner's action was found up until one second had passed or until an action performed by the second-partner was found. To test for switching, the second-partner's action should move the same chess piece which was just moved by the first-partner's action.

Figure 4.25 summarizes, for all sessions, the number of piece control switches made by each team in each session. It can be seen that switching control of pieces took place, although the numbers are low. One reason for these low numbers of switches could be that partners used it to gain control of chess pieces that conform to their roles in the game (e.g., a defensive player would seek control of the king), the situation which occurred only a small number of times during the game-play. Another reason could be the tendency of players to keep controlling a piece until it was captured, thereby reducing the opportunities for switching.

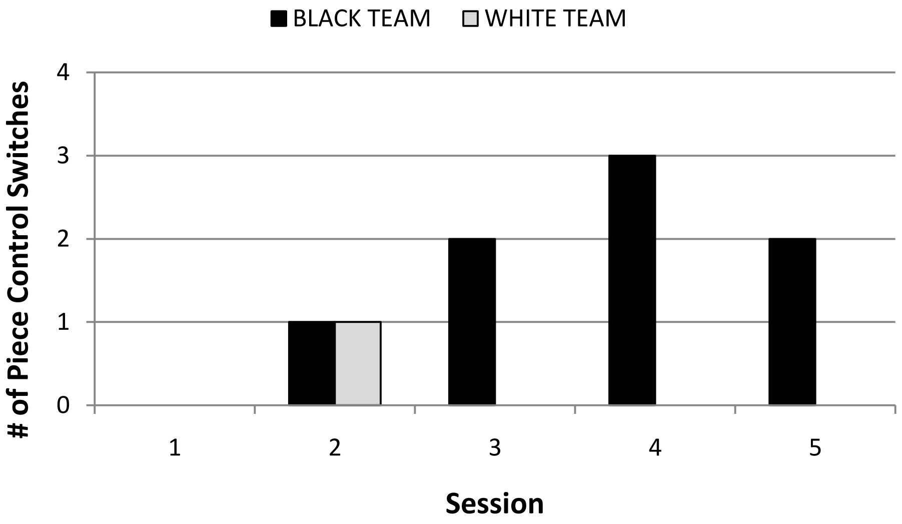

**Figure 4.25.** Number of piece control switches made by each team in each session.

As described above, the method used for finding piece control switches consisted of searching the logs for two consecutive move actions (one per partner) that occurred within a short timeframe length (one second). However, this method did not find all actual piece control switches because of two main reasons. Firstly, a move action consists of two steps: selecting the chess piece and then releasing it at the end square. These two steps do not have to occur within one second (e.g., a player could select the chess piece and then take a while deciding where to move it), thereby preventing the switch from being captured by the search algorithm used to find the piece control switches. Secondly, control switches could consist of different combination of consecutive partners' actions. For example, the current search algorithm looked for move-move combinations (i.e., two consecutive move actions, one per partner). However, other combinations are also possible such as move-select combination (when a player moves a piece and then their partner takes control by selecting it, but then decides not to move it), select-move combination (when a player selects a piece but then decides to give it to the partner, so they let go of the selected piece and the partner would move it), select-select combination (same as select-move combination, but in the end the second partner would not move the piece and, instead, they would let go of the piece). Nonetheless, these other combinations were not considered by the search algorithm because false select actions occurred many times. For example, it was observed that when partners get close to each other, one of them would mistakenly select their partner's piece and then let go of it. These situations appeared in the logs in the form of multiple consecutive select actions made by both partners (e.g., the same piece would be selected by the first partner, then by the second partner, then again by the first partner) and would be captured as multiple switch actions made by the partners (e.g., the mentioned three select actions would appear as two piece switches made by that team). Therefore, using only move-move combinations allowed for capturing accurate switches (i.e., no false switches were captured) and, at the same time, was sufficient to prove that piece control switching did occur in RTChess and that it occurred in low numbers.

Figure 4.26 summarizes, for all sessions, the number of explicit piece control switches that were found by monitoring the recorded material (see Section 4.6.3.2) and that were not captured by the search algorithm due to one of the two reasons mentioned above (the switch either occurred within more than one second or was not a move-move combination). As it can be seen, both results (the search algorithm results and the monitoring process results) show that the black team made more switches than the white team. This result suggests that the black team was able to perform more HSC interaction than the white team.

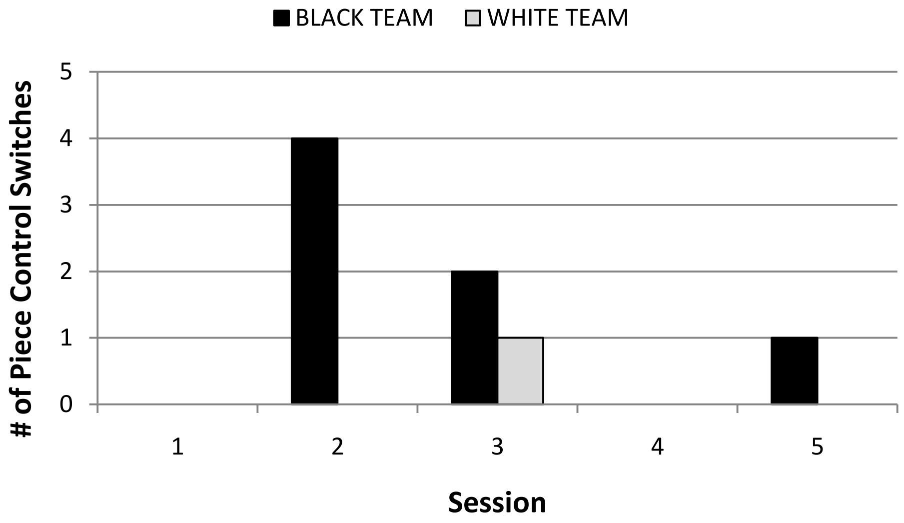

**Figure 4.26.** Number of piece control switches made by each team in each session that were found by monitoring the recorded material.

Similar to the previous measures, it would be interesting to see whether the number of concurrent players on the chessboard affected the number of piece control switches performed by the partners or not. Therefore, piece control switches (both found by the search algorithm and by the monitoring process) were counted again but this time they were grouped by the number of concurrent players operating at the time when the switch was found (see Figure 4.27). As it can be seen, both teams switched control of chess pieces when three as well as four concurrent players were on the chessboard. Additionally, when four concurrent players were on the chessboard, the black team members performed more switches than when fewer (three) concurrent players were operating on the chessboard due to the way the black team members played. As mentioned earlier, black1 played defensively most of the time. At the same time, whenever two or more black offensive pieces were available, black1 would control an offensive piece and would keep an eye on the black king. When the first two black offensive pieces were captured, black1 would give up the last offensive piece for their partner black2. These situations were more frequent when four players were concurrently operating on the chessboard because, at that time, it is most likely that more offensive black pieces were still available on the chess board, thereby allowing black1 to perform their manoeuvre.

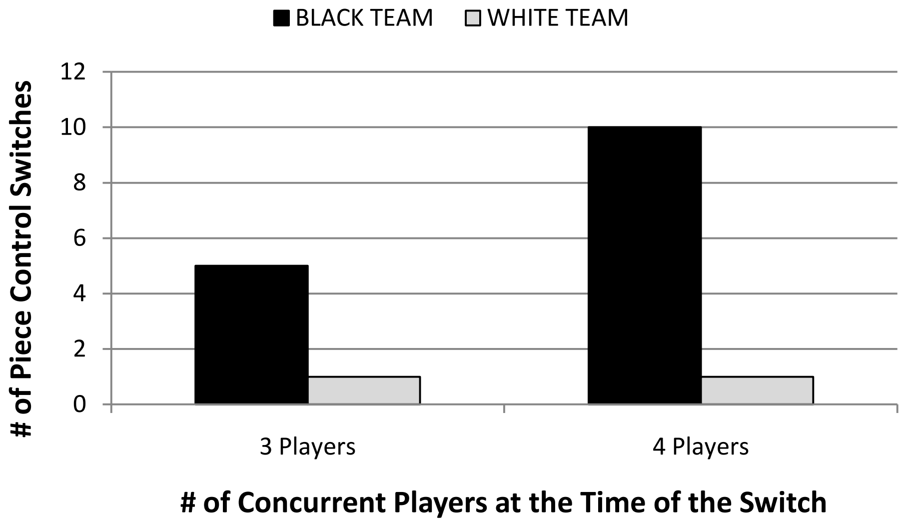

**Figure 4.27.** Number of piece control switches made by each team, counted throughout all the sessions and grouped by the number of concurrent players.

#### *4.6.3.4 Questionnaire Results*

The questionnaire consisted of two sections: demographic questions and RTChess questions (for demographic information see Section 4.6). The RTChess section had four ranking-style questions that used a 5-point scale with semantic anchors (see Appendix A for questionnaire material). Each question also had an empty space for participants to explain their responses. Note that the participants divided their playing strategies into defensive and offensive strategies which might have affected their responses (white1 and black1 were defensive, and white2 and black2 were offensive).

- *Coordination*

Participants were asked to rank the degree of coordination they had with their partners. The actual question was: “to what degree did you and your partner coordinate during the session”. Responses were marked on a one (very poor) to five (very good) scale and are summarized in Figure 4.28.

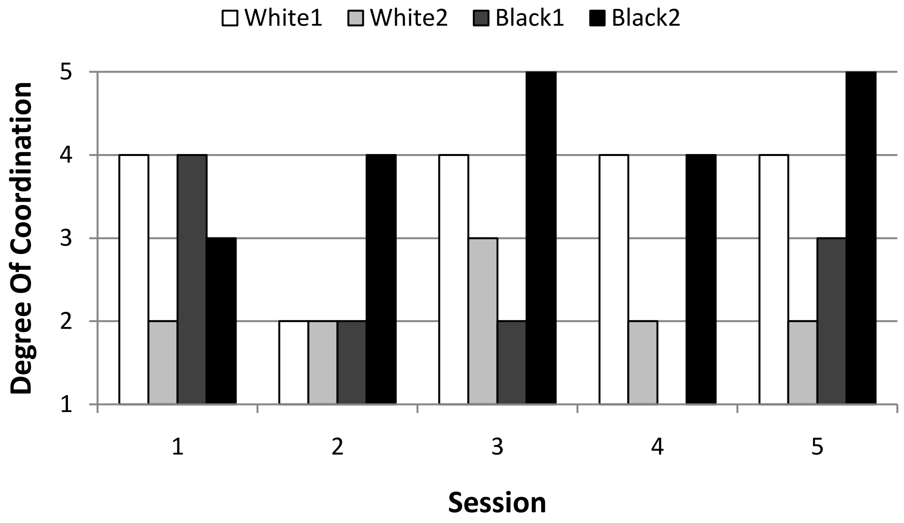

**Figure 4.28.** Participant report of degree of coordination for all sessions (one = very poor and five = very good).

Differences existed among team partners about whether they had coordinated or not. For example, in the fourth session, the black team rankings show such differences when player black1 thought that coordination with their partner was very poor and commented “changed the plan; this time black2 took the queen rather than the knight. Seemed to work well throwing off the other team”. On the other hand, player black2 thought that there was lots of coordination and commented “suggested small variation to our strategy to compensate for other team’s modifications”.

Rank differences amongst team members suggest that there were differences in perceived coordination. Examining the players’ comments revealed indications about what they considered as coordination. For example, player black2 considered creating strategies, sending verbal game states, and changing tactics as coordination with their partner. Black2’s actual comments were: “setup defence/offense strategy...”, “Using short words like ‘Distracting’ or ‘lost'em all’, etc. We could send game state updates to each other”, and “Changed tactics a few times...”. Additionally, player white1 considered trading off pieces as coordination with their partner as white1 commented “...There was at least one time where we traded off pieces, particularly the king piece”. Finally, player white2 considered discussing patterns and communicating with their partner as forms of coordination. White2’s actual comments were: “...sometimes discussed patterns we saw our opponents make” and “Bit more communication when all offensive pieces were gone to provide warning”.

Players’ feelings might have also affected their answers. For example, the white team’s negative feelings about their performance might have made them think that they were not coordinating well enough. In the second session, participant white1 commented “we seemed to do worse this time, we changed our strategy half way through but it didn’t seem to help…”.

Both teams recognized that high-speed activity was going on inside the game. Participants’ responses showed that the high-speed nature of RTChess made coordination difficult. For example, in the second session player white2 commented “too hard / too fast to coordinate…”. At the same time, player black1 commented “during the (games) there is too much going on to really coordinate...”. After that in the third session, player black1 commented “…the sessions (games) are so fast that there isn’t much time to do anything but play”. Later on, in the last session, player white2 commented “I think I was a bit more aware of (my partner’s) movement” which suggests that team-play experience gained from playing four sessions made a difference in the white team’s ability to coordinate.

When the game’s pace slowed (see Section 4.6.3.1 for information about the game’s pace), white team players indicated that they had more time to coordinate. For example, in the third session, player white2 commented “…bit more communication when all offensive pieces are gone to provide warnings”. Additionally, in the last session, player white1 commented “we switched players (pieces) more, although this happened when I didn’t have the king and it was the only player left”.

- *Forming Plans*

Participants were asked to rank how easy it was to form plans with their partners and to indicate whether they actually applied those plans or not. Participants were also asked to provide brief examples. The actual question was: “how easy was it to form plans with your partner and actually apply them? Can you give brief examples?”. Responses were marked on a one (very difficult) to five (very easy) scale and are summarized in Figure 4.29.

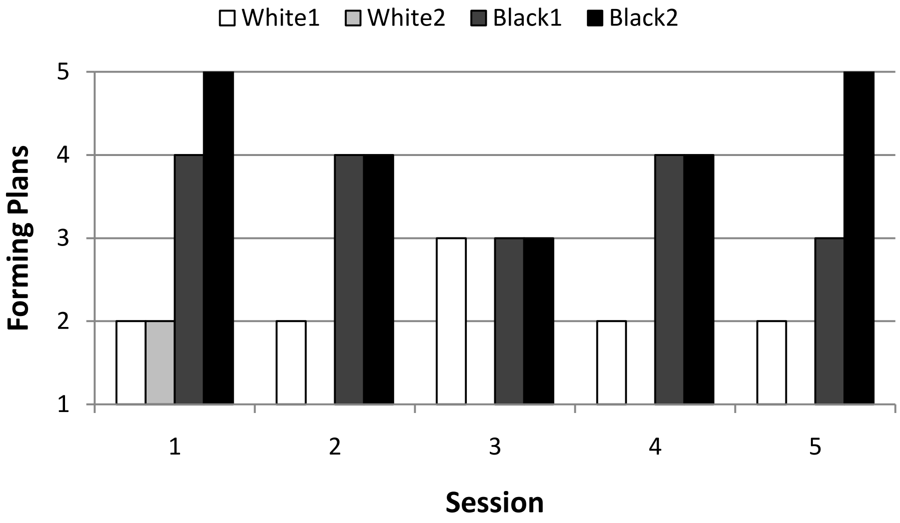

**Figure 4.29.** Participant report of easiness of plan forming (one = very difficult and five = very easy).

Players reported that it was difficult to form plans because of the high-speed nature of the game. Players commented on this difficulty by saying: “It was very difficult as there was not a lot of time to make strategies...” – white1, “No time for verbal communication. No opportunity for strategizing within game...” – white2, “most planning happened when the game pace slowed down” – black1, and “the game moves so quickly…” – black2.

Some partners were able to form plans in-between games (in the three second pause) and/or when one partner was idle. For example, player white1 commented “most strategic decisions were done between games or when my partner exhausted all (their) pieces”. Other players thought it was easy to form plans, as player black2 commented “as the game board is small and each piece has a name, verbal plan formation is easy...”. Player black2 adapted to the fast pace of the game by forming low-level plans (i.e., plans that do not contain much details) and then making small modification to them; in this regard player black2 commented “The game moves so quickly that plan creation happens very quickly, hence plans are not too detailed – more simple modifications to behaviour like ‘I will start using the bishop first’ or ‘I'll take the queen’”.

- *Communication*

Participants were asked how well they were able to communicate with their partner during game-play. The actual question was: “how well were you able to communicate with your partner during game-play?”. Responses were marked on a one (very poor) to five (very good) scale and are summarized in Figure 4.30.

Players found it difficult to communicate in this high-speed groupware environment. For example, player white2 commented “...the time to verbalize and process is too long...” and “no time to even move eyes off screen”. Player white1 also commented “again most communication happened once the other player lost all the pieces”. Additionally, player black1 commented “...game is too fast for on the fly communication”. On the other hand, one player found it was easy to communicate; as player black2 commented in the second session “verbal, small board, named pieces all made it easy to communicate...”. Later on, in the third session, player black2 commented “brief and quick messages but because these are easy to process, we can send them frequently, if necessary”. After that, in the fourth session, player black2 commented “verbal communication plus mouse gestures worked very effectively and quickly to communicate”. It is clear from player black2’s comments that through time they were able to develop their communication. Player black2 started first with verbal messages, then switched to quick verbal messages, and lastly verbal messages and mouse gestures.

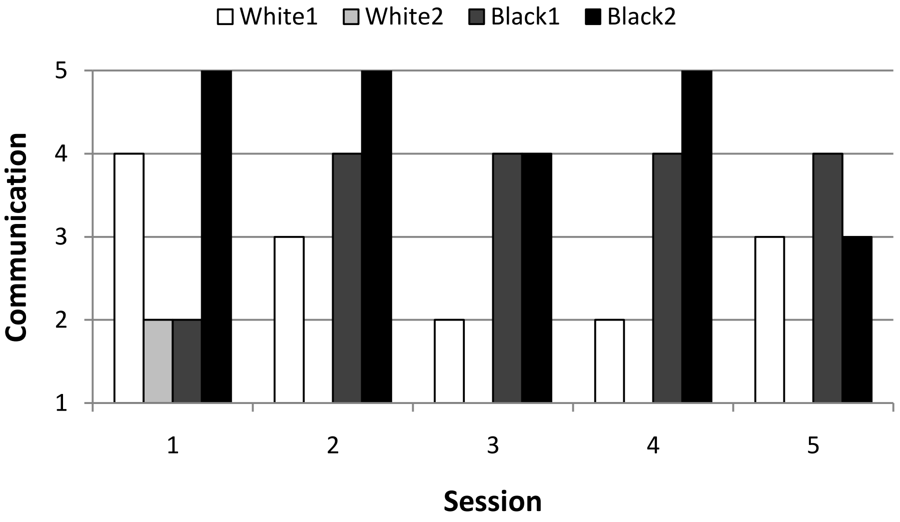

**Figure 4.30.** Participant report of degree of communication (one = very poor and five = very good).

It seems that visibility affected communication between the players. For example, in the 1st session when black team players were seated next to each other (see section 4.6.2 for apparatus details), player black2 commented “adjacency - so we could speak easily...”. However, when they were seated face-to-face, player black1 commented “the communication was easy because (my partner) was sitting across of me”. In the last session, when they were switched again to adjacent positions, player black2 commented “as we were not directly facing each other communication was a little more difficult”.

Again white team players associated their negative performance feelings with how well they communicated. For example, player white1 commented “I thought that it was getting a little easier to communicate this time when compared to last session just because it wasn't as "new" of a situation. However it seemed that we did worse”.

- *Communication Method*

Participants were asked to rank the frequency with which they used each of the communication methods: speaking, mouse gesturing, and hand gesturing during game-play. The actual question was: “for each of the following methods (speaking, mouse gesturing, and hand gesturing), please specify how frequent did you use it to communicate with your partner during the game-play. If you used other methods please write them too”. Responses of the three communication methods (speaking, mouse gesturing, and hand gesturing) were marked on a one (not used) to five (very frequently) scale and are summarized in Figure 4.31, Figure 4.32, and Figure 4.33 respectively. The results show that speaking was the most frequent way of communicating, followed by mouse gesturing, and then hand gesturing.

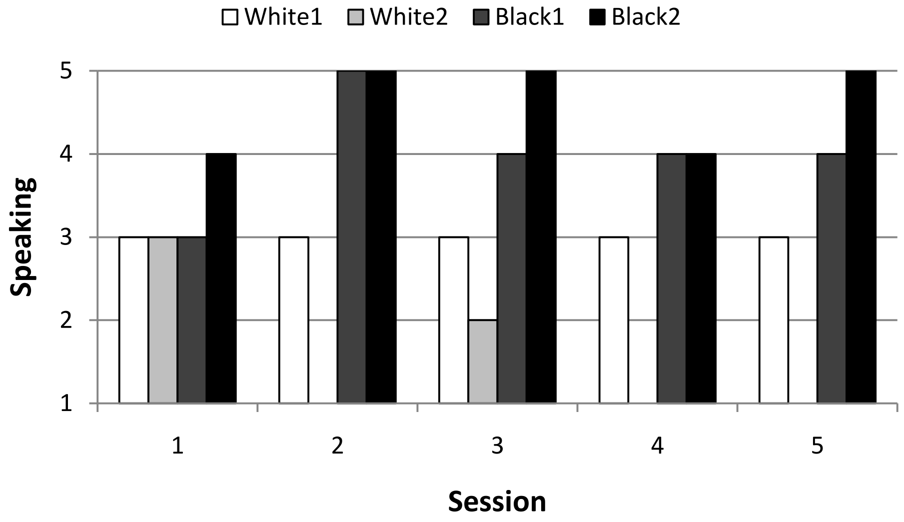

**Figure 4.31.** Participant report of frequency of 'speaking' usage as a communication method (one = not used and five = very frequently).

Amongst the three methods of communication, speaking required the least effort. It only required the player to talk and/or listen. Players' comments show that they thought speaking was the easiest and best method: "I choose to speak most of the time because that seemed to be the easiest method..." – white1, "we would talk mostly..." – black2, and "the best way to communicate was with voice" – black1.

Mouse gesturing was the next easiest method of communication. It did not require the players to divert their eyes from their screens, but it did require them to focus on their partner's actions. Player black2 commented "(I used) mouse gesture for taking over control of a piece from partner", "mouse gestures would be used to indicate that I (offense) had run out of pieces" and "(I used) mouse plus speech to indicate piece loss and piece takeover". From the comments, player black2 seemed to use this communication method only when there were no pieces left to control.

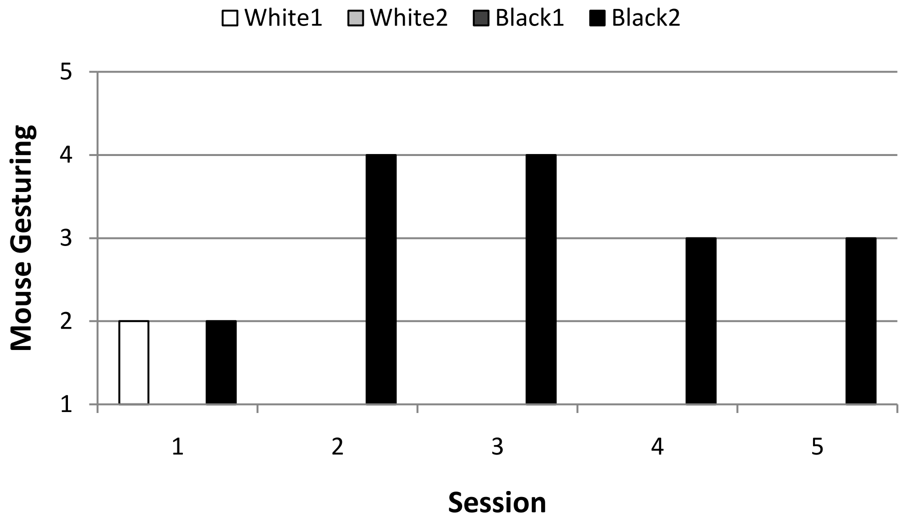

**Figure 4.32.** Participant report of frequency of ‘mouse gesturing’ usage as a communication method (one = not used and five = very frequently).

Hand gesturing was the most difficult method of communication. Hand gesturing required the most effort of the three communication methods as it required the players to switch their eyes from the screen, which was difficult because they did not have enough time to do so. No one used this method to communicate with their partners except for player black2 who commented “thumbs up at the end”.

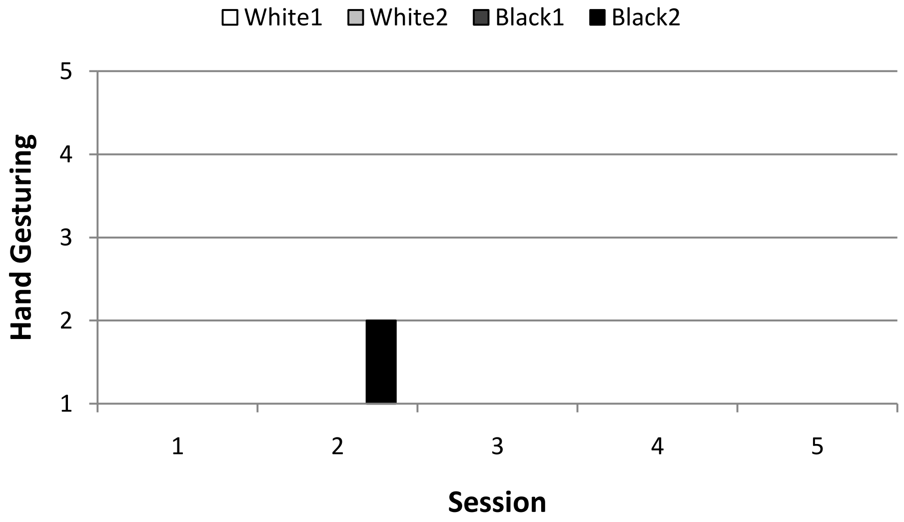

**Figure 4.33.** Participant report of frequency of ‘hand gesturing’ usage as a communication method (one = not used and five = very frequently).

# CHAPTER 5

# DISCUSSION

In this chapter, the results will be discussed to determine the contribution that has been gained. To do this, the results of the evaluation will be reviewed, and then the main research problem posed in Chapter One will be considered using these results and detailed answers to the evaluation questions will be given. After that, the evaluation process will be discussed in terms of factors that might have affected the results and in terms of the design issues that were considered throughout the evaluation. The chapter ends with a discussion regarding the lessons learned to improve the support for HSC in groupware.

## 5.1 Summary of Results

In Chapter Four, an evaluation was performed to investigate HSC in the high-speed groupware environment – RTChess. The results were divided into four categories: general findings, monitoring results, log results, and questionnaire results.

The general findings (see Section 4.6.3.1) showed that game lengths were very short (average length of 14 seconds and most frequent length of eight seconds) and that there were times during the game-play when the pace was fast (more than 40% of the time, the rate of play was greater than one move per second per team) and other times when the pace was slow (more than 16% of the time, the rate of play was less than one move per second per team). The game length information as well as the number of moves per second performed by each team gave a clear indication of the high-speed environment found in RTChess. In addition, the general findings showed that there were times during the game-play when two, three, or four concurrent players were operating (i.e., moving pieces) concurrently on the chessboard. It was clear that HSC did not occur when only two concurrent players (one in each team) were operating on the chessboard and that HSC occurred only in one team when three concurrent players were operating on the chessboard. Moreover, the general findings suggested that the better performance of the black team was due to the higher multiplayer-game experience player black1 had which allowed them to perform more coordinated interaction as it appeared in their comments. Moreover, the general findings showed that the white team's performance improved throughout the sessions.

Monitoring the screen capture and voice recordings resulted in finding interactions that suggested episodes of HSC between team partners (see Section 4.6.3.2). These interactions were divided into three categories: mouse movement (e.g., hovering the mouse over an enemy chess piece in an attempt to distract the opponent), chess piece movement (e.g., moving close to each other seeking protection and serving as bait), and voice communication (e.g., giving warnings and asking for chess pieces).

The log results examined three measures between partners of each team: operating distance, operating area, and switching control of chess pieces (see Section 4.6.3.3). The 'operating distance' measure found that on average partners were about three chessboard squares away from each other; which was not considered a very close distance given that the chessboard is only 8x8 squares and that the most frequent distance was four chessboard squares. In addition, the distance between team members was slightly affected by the number of concurrent players operating on the chessboard. When the four players were operating concurrently, partners of each team played farther from each other than when only three players were operating concurrently. This behaviour indicated that partners were able to perform more HSC interaction when they were chasing the last piece of the opponent team – the king.

The 'operating area' measure found that partners of each team were operating in the same chessboard area for an average of 22% and 17% of the time for the black and white teams, respectively. This result (operating in the same chessboard area for a small amount of the time) indicated that, most of the time, partners of each team were operating in different chessboard areas. In addition, the number of concurrent players affected whether partners operated in the same area or not. When the four players were operating concurrently, partners played in the same area (22% for the black team and 15% for the white team) less often than when only three players were operating concurrently (24% for the black team and 30% for the white team). Moreover, the operating area measure found that when the four players were operating concurrently, the white team partners operated 70% of the time of all the sessions in different vertical halves (left and right halves).

The ‘switching control of chess pieces’ measure found that partners were coordinating with each other by switching control of pieces even though the switches were in small numbers. In addition, the piece control switching measure found that the black team performed more switches (total of 15 switches) than the white team (total of two switches) which indicated that the black team performed more coordinated interaction during the game-play. Moreover, when the four players were operating concurrently, the black team performed more piece control switches than when three players were operating concurrently. The reason for this behaviour of the black team (having more switches when more players were operating concurrently) was caused by the way participant black1 played which made use of the more black chess pieces available on the chessboard when the four players were operating concurrently compared to the three concurrent players situation.

The questionnaire results (see Section 4.6.3.4) showed that the participants were able to perform HSC interactions during the game-play, but also that they found it difficult. The participants also reported doing more coordinated interaction when the game pace slowed down. In terms of planning, the participants found it was difficult to make long term plans as the environment changed frequently. Player black2 showed that they were able to make short plans and kept updating them during the game-play. Additionally, in terms of communication, the participants also found that it was difficult to communicate with their partners. Again, player black2 showed that they were able to communicate using short and quick status messages. Lastly, the participants showed that their favourite communication method was communicating by voice.

## 5.2 Implications of Results

The implications of the results will now be explored. This is done by examining the original research problem posed in Chapter One and exploring how these results address this problem.

### 5.2.1 Original Research Problem

The research problem presented at the beginning of this thesis was:

There is little information available about HSC in high-speed groupware. High-speed activities have not succeeded in groupware to date; in order to support high-speed activities (e.g., online team sports) we need to understand how HSC works and we need to answer the following basic questions:

- Does HSC occur in high-speed groupware environments, and how does it occur?
- Are the users aware of HSC?
- What in groupware makes HSC possible and what makes HSC more difficult?
- What is the speed at which users can coordinate best?

### 5.2.2 HSC Basic Questions

The four basic questions were designed to investigate and extract basic information about HSC from RTChess. The first question looks at whether HSC occurred in RTChess or not. The answer to the first question should give details about the way HSC occurred in RTChess and its different forms. The second question looks at the participants' feelings and whether they were aware of performing HSC or not. The answer to the second question should give details about how did the participants feel about HSC, what were the participants' comments on HSC, and how did the participants perceive HSC. The third question examines HSC in more depth in terms of the things that affected it and made it possible and/or more difficult. The last question examines the speed at which activities occurred in RTChess and tries to figure out the speed at which the participants could have coordinated best.

#### *5.2.2.1 Does HSC occur in high-speed groupware environments, and how does it occur?*

The results of this evaluation suggest that HSC did occur in RTChess, although not to a large degree. By monitoring the screen capture and voice recordings, different interactions were identified to suggest HSC between team members (see Section 4.6.3.2). Additionally, the log results showed indications of coordinated interactions between team members (e.g., not getting too close from each other, playing in different areas of the chessboard, and switching control of chess pieces).

The monitoring results (see Section 4.6.3.1) indicated that the maximum number of HSC interactions found in a session was 21 interactions. At the same time, a maximum of 15 piece control switches found in the log results (see Section 4.6.3.3). As a result, given the fact that on average a single RTChess game of length 14 seconds contains at least 18 move actions made by each team (see Section 4.6.3.1), it becomes clear that only small amounts of HSC occurred in RTChess.

It was difficult for partners to keep full awareness about each others' exact actions due to the high-speed activities going on in RTChess. As a result, partners stayed at a distance from each other and even played at different chessboard areas throughout the sessions in order to prevent collisions and mistakes (see Section 4.6.3.3). Since coordination requires awareness (Neale et. al., 2004), the awareness difficulty faced by the players made it difficult to achieve HSC between team members. To handle awareness difficulty, the partners appeared to use two levels of awareness: low-detail and high-detail levels of awareness (Gutwin and Greenberg, 2002). The low-detail level was used during the fast-paced game-play, especially when the four players were operating concurrently, to keep track of the general actions taken by the partner. For example, each team member kept track of the general area where their partner was interacting and, therefore, was able to stay in different chessboard areas during the game-play. The high-detail level of awareness was used during the slow-paced game-play, especially when three players were operating concurrently, to keep track of the exact actions taken by the partner. For example, when the black king, the black bishop, and the white king were left on the chessboard, the black king was following the black bishop's exact actions and was ready to take advantage of any opportunity created by the black bishop (see Section 4.6.3.2). Additionally, one player (black2) was able to develop alternative ways of accomplishing HSC in a fast manner by using quick verbal status messages and mouse gestures (see Section 4.6.3.4).

HSC occurred when the game was at full speed, no matter whether three or four players were operating concurrently (see Section 4.6.3). HSC occurred in three different ways: one-sided, two-sided, and one-sided that became two-sided. One-sided HSC occurs when a player uses information about their partner to decide the next interaction without involving the partner in this decision. A good example for this type of HSC is when player white2 attacked the black king, who was busy fighting white1, without involving white1 in this decision. Two-sided HSC occurs when two partners agree on something and then actually doing it. For example, black team members agreed on switching control of the black king and then actually did the switch (see Section 4.6.3.3). The last type of HSC (one-sided that becomes two-sided) occurs when a player keeps repeating a one-sided HSC pattern, then their partner would recognize that pattern and would actually involve it in future plans. For example, as the implicit switching of pieces, which occurred in the black team (see Section 6.4.3.3), occurred frequently, the black team members became familiar with this situation that whenever player black2 lost two pieces they would pick up the last piece directly and, at the same time, player black1 would leave that piece (which was picked up by black2) directly and head back for the king.

#### *5.2.2.2 Are the users aware of HSC?*

Players did state that they performed HSC with each other but they had differences in perceiving which actions should be considered as coordinated interactions. For example, in the fourth session, player black1 thought that they had no coordination at all while player black2 thought that they had lots of coordination (see Section 4.6.3.4). In addition, all the participants reported that coordinated interactions were difficult to accomplish especially when the game was at full speed (i.e., HSC was difficult). The participants' comments indicated that they coordinated at different times during a session: during the game-play which was reported as the most difficult time to perform coordination, in the three-second breaks between games, and in between the two parts of the each session (see Section 4.6.3.4).

#### *5.2.2.3 What in groupware makes HSC possible and what makes HSC more difficult?*

Five factors have been found to affect HSC in RTChess: the player's experience with the game, the level of awareness of the partner's interactions, communication between partners, the number of concurrent players on the chessboard, and the speed at which interactions occur in RTChess. First, all the participants had prior experience in playing RTChess (see Section 4.6). Player black1 was able to perform more coordinated interaction earlier than other players (e.g., controlling two pieces simultaneously; see Section 4.6.3.2). The reason for player black1's better performance was suggested to be their multiplayer gaming experience which was reported to be the highest amongst all the participants (10 hours per week). Additionally, by observation, white team players showed more coordinated interaction in the last session after gaining more game-play experience (see Section 4.6.3.2 and Section 4.6.3.4). As a result, increased experience should make HSC more possible and easier.

Second, in RTChess, it was difficult for partners to play in close proximity to each other (see Section 4.6.3.3). The main reason for this difficulty was the lack of high-detail awareness of the exact actions taken by the partner. This lack of awareness had led to difficulty in coordination which caused collisions between the partners playing in close distance. As a result, enhancing awareness should enable more coordinated interaction to occur between the partners especially when playing in close range.

Third, in RTChess, three channels of communication were available for the participants to utilize: voice, mouse gestures, and hand or body gestures. The participants reported that it was difficult to communicate during the game-play (see Section 4.6.3.4). One exception was player black2 who was able to develop short and quick status message and was able to use visual information available on the chessboard to communicate with their partner. Difficulty faced in communicating with partners is mainly caused by the high-speed environment which leaves "no time to even move eyes off screen" – white2. Coordination requires team members to coordinate both the communication and activities (Neale et. al., 2004), therefore, easy-to-use communication should make HSC easier to achieve.

Fourth, having more concurrent players on the chessboard meant that each player had to consider more dependencies when making decisions and performing interactions. It was clear in the results (see Section 4.6.3) that having three concurrent players operating on the chessboard allowed for more HSC interaction between the partners (such as operating in close distance and in the same area) than having four concurrent players operating on the chessboard. As a result, fewer dependencies taken into consideration when making split-second decisions should make HSC easier to achieve.

Last, considering ‘high-speed’ as a range of degrees (i.e., ranging from ‘barely high-speed’ to ‘very high-speed’), the speed at which interactions occurred in RTChess was measured by examining the number of moves performed per second (see Section 4.6.3.1). Performing an average of 1.55 and 1.3 moves per second by the black and white teams respectively was fast enough to make the players face coordination difficulties. In general, because of the high demands for the players to keep full attention to the pieces they were controlling, it was difficult for the players to utilize some of this attention in coordinating with their partners. As a result, HSC was more difficult during higher speed activity than at lower speeds. More details about this factor (the degree of high-speed at which interactions occur in RTChess) will be discussed in the next section (see Section 5.2.2.4).

#### *5.2.2.4 What is the speed at which users can coordinate best?*

All of the participants reported that it was very difficult to coordinate during game-play in RTChess (see Section 4.6.3.4). However, the participants noticed the existence of different speeds (i.e., ranging from ‘barely high-speed’ to ‘very high-speed’) and have reported performing more coordination during slower speeds. At the same time, player black1, who had the highest multiplayer-game experience, was able to do more coordinated interaction than other players even during large degrees of high-speed (see Section 4.6.3.1 and Section 4.6.3.2). In addition, examining the log results (see Section 4.6.3.3), it is clear that during large degrees of high-speed the participants were playing away from each other in different chessboard areas. In some cases, the players decoupled the task in order to cope with the speed. Such a behaviour (decoupling the task) was observed between partners by dividing themselves into defensive and offensive players or by agreeing to stay away from each other to avoid collisions. As a result, the slower the speed of interaction the easier it is for partners to perform coordinated interaction.

### 5.2.3 HSC Definition

Now that the four basic questions are answered, HSC can be better defined and compared to regular coordination. The main difference between HSC and regular coordination is that HSC is required to occur very quickly in order to keep up with the speed of activities in the system. However, this main difference causes other differences in the coordination mechanisms that should be utilized when coordinating in high-speed versus those needed for regular coordination (see Section 2.3.1 for more information about coordination mechanisms). For example, in regular coordination users could speak with each other as a method of communication; however, in HSC users might use implicit communication as a faster method of communication. In conclusion, HSC can be defined as coordination that needs to occur very quickly.

## 5.3 Limitations, Critical Reflection, and Lessons

In this section, limitations of the evaluation method will be explored. Critical reflection on the design decisions made throughout this study will be presented as well. In the end, generalization of the results presented in this study will be discussed.

### 5.3.1 Limitations

In this section, some limitations of the evaluation conducted in this thesis will be discussed. These limitations are: the small population (only four participants) that was used in the evaluation, HSC being monitored between only two people, and HSC being monitored only between team members.

*Small Number of Participants*

One of the issues under investigation in the evaluation was the effect of player’s experience on the emergence of HSC interaction in RTChess. Therefore, it was necessary to have the same participants play RTChess for several sessions in order to build up their experience. However, a larger population would give more accurate results than a small population because a behaviour found in the results would then be shared amongst many people which gives more validity for the results.

For example, using a small population, it is difficult to answer the question: if different participants were recruited, would the same results be found? However, due to limited resources and time limits, a small number of participants (only four participants) were recruited in this evaluation.

### *HSC between Only Two People*

Coordination is a prerequisite to successful teamwork (Neale et. al., 2004). Teams usually consist of two or more people. The more people in a team, the more dependencies each team member needs to take into consideration when coordinating with other team members. For example, in a soccer team, a player should keep track of the position of other nearby team members; just in case a need to pass the ball arises. However, it would be much easier for a football player to keep track of only one other team member.

In this evaluation each team consisted of only two players. Since this study is one of the first that explores HSC in high-speed groupware (RTChess in this case), simplicity was important in order to draw the base line for further studies. As was discussed in Section 4.5.1, having more than two players per team could allow for more complicated coordination patterns to occur. Having only two players per team helped keep this evaluation simple by exploring only one-to-one partner coordination.

### *HSC between Only Team Members*

Within a single team, all team members are working for the same overall goal. For example, in RTChess, white team members’ goal was to capture the black king, while the black team members’ goal was to capture the white king. It is expected that different results come out if HSC was examined between different teams. For example, it might be found that opponents operated in close proximity to each other and/or operated in similar chessboard areas especially for those opponents engaged in one-on-one fights. Of course, there will not be any piece switches between opponents. In this evaluation, however, HSC was examined only between team members for simplicity reasons and due to time limits.

### 5.3.2 Critical Reflection

In this section, some design decisions that were made during the evaluation will be examined. This includes some discussion of what could have been done differently, including reconsideration of the monitoring process, network delay, and white team's performance.

#### *The Monitoring Process*

For this study, I had to monitor the screen capture and voice recordings (see Section 4.6.3.2) in order to visually extract the interactions that suggested HSC between team partners. While this is sufficient for this thesis and its scope, it is important to note that the monitoring results are all from one point of view and, thus, might have bias. However, when the recordings were monitored, I made sure that only the interactions that showed clear HSC between the partners were noted. Alternatively, a team of observers could do the monitoring process. However, due to limited resources and time limits, it was not possible to recruit a team of observers.

#### *Network Delay*

The participants often blamed network delay when something went wrong during game-play. Two situations occurred often during the game-play in which network delay was blamed: first, when a player clicks on a chess piece to select it, the chess piece does not get selected right away because the server's permission which allows the selection to occur has not arrived at the client yet. Second, when two opponent chess pieces move into the same empty square attacking each other, it was unknown which chess piece will survive the attack because of the concurrency control model followed (first-come-first-serve). In these cases the participants became frustrated and said things like: "oh… the race condition" (i.e., the first-come-first-serve model) and "no... I clicked on it but it did not pick up". Interestingly, when the screen capture and voice recordings were monitored, it was found that most of these "delay problems" which caused chess piece not to get selected were actually miss-clicks by the players.

The network delay problem and its effects on HSC are out of the scope of this thesis. Nonetheless, the small amounts of network delay noticed by the players did not cause major problems. On the other hand, large amounts of network delay did not occur during game-play. However, since ‘real-time interaction’ is one of the important characteristics of high-speed groupware, large amounts of network delay are expected to hinder HSC.

*White Team’s Performance*

In the results, it was indicated that one of the reasons that might have affected the white team’s performance was the lack of experience found in the black team. In fact, there are two other factors that might have affected the white team’s performance as well: first, the chessboard layout. As shown in Section 4.3, the white chess pieces were placed at the top of the chessboard, while the black chess pieces were placed at the bottom of the chessboard. The white team players have reported that it would feel more natural if their pieces were at the bottom of the chessboard. However, as all players already have previous experience in playing RTChess, the ‘chessboard layout’ factor is not considered as a serious issue but is still a valid point.

The second factor that might have affected the white team’s performance is the seat positions of the team players. The white team players were seated next to each other on chairs three and four (see Section 4.6.2) for three consecutive sessions. Communication between players seated in these chair positions (chairs three and four) is expected to be more difficult than communication between players seated in chairs one and two. The main reason for this difficulty is that a player seated in chair three or four have to turn their head away from the screen in order to see their partner. However, since the communication used between the partners, even when seated in chairs one and two, was mainly through speech and mouse gestures (see Section 4.6.3.4), then sitting in chairs three and four should not have had any serious impact on the communication but might still be a valid point.

### 5.3.3 Lessons: Improving Support for HSC in Groupware

As has been discussed in Chapter Two, high-speed groupware falls into three areas of study: groupware, coupling, and coordination. This section will discuss how the results found in this study can be generalized for designers in these three different areas. In general, the results of this study show that HSC does occur in high-speed groupware but that it is often difficult to achieve by the users. To overcome the difficulties, several issues should be taken into consideration in regards to the five factors that affect HSC (user's experience, level of awareness of the partner's interactions, communication between partners, number of dependencies that affect the user's interactions, and degree of high-speed of activities in the system). First, in regards to the user's experience, tutorials and/or training systems and sessions could be created and used to increase user's experience. Increased user experience should enhance decision-making abilities by enabling the users to take fast real-time accurate decisions.

Second, in regards to level of awareness of the partner's interactions, audible and/or visual cues could be added to a system in order to increase awareness between the partners. However, these cues should be created in a way that enables the users to infer as much information as possible with the least effort possible.

Third, in regards to communication between partners, easy-to-use communication channels could be established to simplify and lower the effort of communication between the partners. One way to accomplish these easy-to-use communication channels is by using implicit communication. Implicit communication requires less effort from the user than explicit communication does. However, information provided by the implicit communication channels requires high user experience in order to be interpreted accurately.

Fourth, in regards to the number of dependencies that affect the user's interactions, having few dependencies and/or gradually increasing dependencies could help the user establish experience in handling earlier dependency and, thus, new dependencies would be easier to handle.

Last, in regards to the speed of activities in the system, lowering the speed at which interactions occur in the system could help partners perform more coordinated interactions. Alternatively, the system could automatically change speed based on the amount of activity going on, or the system could provide tools for the users to manually change interaction speed (e.g., Bullet Time[3]).

---

[3] Bullet Time is an ability found in some video games which allows the player to slow down the pace of the game to the extent that moving bullets can be seen with bare eyes (e.g., Max Payne).

# CHAPTER 6

# CONCLUSIONS, CONTRIBUTIONS, AND FUTURE WORK

## 6.1 Conclusions

A group of users working together on a tightly-coupled real-time groupware system, which incorporates a high-speed environment, often need to use HSC in order to avoid conflicts and to help achieve the goals. However, little information is available about HSC in groupware. In this thesis, a solution to this problem was achieved by building a high-speed groupware system (RTChess) – which incorporated high-speed activity and that supported tightly-coupled and real-time interaction – and by exploring HSC in this system. As a result, information about HSC was extracted from this system and was used to form a basic understanding about HSC. This basic understanding included characteristics of HSC and a list of five factors in groupware that affected HSC.

## 6.2 Contributions

The main contribution of this thesis is the initial information gathered about HSC in groupware. This contribution adds to Coordination Theory and builds up knowledge about coordination in distributed environments. In addition, this contribution allows for more understanding of human behaviour in high-speed groupware systems and enables developers to accommodate this behaviour in design. Finally, this contribution enables developers to build tools and systems that support this type of coordination.

In addition to the main contribution, there are two other secondary contributions of this thesis. These secondary contributions are:

- The high-speed groupware game (RTChess) which can be used as a tool for exploring high-speed groupware environments, HSC, and fast activity.
- The initial measures of coordination (operating distance, operating area, and switching control of items between the team members) which can be used to determine the degree of coordination between team members.

## 6.3 Future Work

This initial work in understanding HSC in groupware has led to a number of possibilities and issues for future work. In this section, some of these issues will be presented.

- *More and Different Participants*

A small number of participants were recruited for this study, they all had prior experience in using the system, and they all knew each other. Future studies should consider the following: first, recruit a larger number of participants in order to gain more validity for the results and to be able to confirm the results of this study; second, observe the effect of users' experience level on HSC by recruiting participants with high/low experience in using the system or with high/low experience of multiplayer games; last, observe the effect of having users who are strangers to each other on HSC.

- *Exploring Variations of HSC*

In this study, HSC was explored between team members (two members per team) who were working for the same goal. Future studies should consider HSC between team members of larger teams (more than two members per team) who are working towards variations of goals (similar goals, different goals, and/or conflicting goals). Moreover, this study was just a general exploration of HSC. Future studies should investigate more details about specific issues related to HSC which were found in this study. For example, future studies could look more into the five factors that affected HSC in groupware (user's experience, level of awareness of the partner's interactions, communication between partners, number of dependencies that affect the user's interactions, and pace of activities in the system).

- *Better Monitoring Process for the Recorded Material*

In this study, I monitored the recorded material by myself. Future studies should incorporate better systematic methods for monitoring the recorded material and extracting interactions that suggest HSC between the users.

- *Tracking the Eye Gaze of the Users*

Eye tracking systems could be installed to track the eye gaze the users. Tracking the eyes of the users could answer questions like how frequently do team members look at each other and where do team members focus their eye gaze at different times and paces of the game-play. In addition, eye tracking information could show different results for different types of users. For example, experienced users might look at different places of the chessboard area while inexperienced users might focus on the piece they are controlling.

- *Better measures of coordination*

In this study, three measures of coordination (operating distance, operating area, and switching control of chess pieces) were explored between the partners' interactions. Future studies could investigate better and/or different measures of coordination. For example, future studies could examine whether there is any correlation between the time of interaction performed by each partner (i.e., if an interaction of one partner leads (time-wise) to another interaction by the other partner).

- *Exploring Multi-Step HSC Patterns*

In the 'switching control of chess pieces' measure, consecutive partners' interactions were used to find whether partners performed switching or not. Future studies should consider HSC patterns that could happen in a non-consecutive pattern and/or could consist of multiple steps. For example, a switch could have happened in the following way (which was not captured by the measure in this study): one partner would let go of their piece and then the second partner could take some time to finish the move they were doing before grabbing the new piece.

- *Exploring Variations of Timeframe lengths*

In the ‘switching control of chess pieces’ measure, consecutive partners’ interactions were examined in order to find piece control switches. It was required that these consecutive actions occur within a one second timeframe in order to be considered as a HSC interaction. Future studies should consider other timeframe lengths when searching for HSC interaction. For example, experienced users might take less time to make decisions than inexperienced users. Therefore, inexperienced users might need a longer timeframe length to perform a HSC interaction than experienced users.

- *Exploring HSC in Other Tasks*

In this study, HSC was monitored as it occurred naturally between team members (i.e., without any system policies or game rules that forced team members to coordinate). Future studies could examine whether it would make any difference if the partners were forced into situations and tasks where they are required to coordinate with each other.

# REFERENCES

Ahuja, S. R., Ensor, J. R., and Horn, D. N. 1988. The rapport multimedia conferencing system. In *Proceedings of the ACM SIGOIS and IEEECS TC-OA 1988 Conference on office information Systems* (Palo Alto, California, United States, March 23 - 25, 1988). R. B. Allen, Ed. ACM, New York, NY, 1-8.

Baker, J., Cote, J., and Abernethy, B. 2003. "Sport-specific practice and the development of expert decision-making in team ball sports." *Journal of Applied Sport Psychology* 15, no. 1: 12-25.

Bradner, E., Kellogg, W. A., and Erickson, T. 1999. The adoption and use of “BABBLE”: a field study of chat in the workplace. In *Proceedings of the Sixth European Conference on Computer Supported Cooperative Work* (Copenghagen, Denmark). Kluwer Academic Publishers, Norwell, MA, 139-158.

Cao, X. and Balakrishnan, R. 2003. VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 173-182.

Czerwinski, M., Robertson, G., Meyers, B., Smith, G., Robbins, D., and Tan, D. 2006. Large display research overview. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). CHI '06. ACM, New York, NY, 69-74.

Daft, R. L. and Lengel, R. H. 1986. Organizational information requirements, media richness and structural design. *Manage. Sci.* 32, 5 (May. 1986), 554-571.

Eccles, D. W., and Tenenbaum, G. 2004. "Why an Expert Team is More Than a Team of Experts: A Social-Cognitive Conceptualization of Team Coordination and Communication in Sport." *Journal of Sport & Exercise Psychology* 26, no. 4: 542.

Ellis, C. A., Gibbs, S. J., and Rein, G. 1991. Groupware: some issues and experiences. *Commun. ACM* 34, 1 (Jan. 1991), 39-58.

Entin, E. and Serfaty, D. (1999). Adaptive team coordination. *Human Factors*, 41, 312-325.

Fish, R. S., Kraut, R. E., and Leland, M. D. 1988. Quilt: a collaborative tool for cooperative writing. *SIGOIS Bull.* 9, 2-3 (Apr. 1988), 30-37.

Greenberg, S. 1990. Sharing views and interactions with single-user applications. In *Proceedings of the ACM SIGOIS and IEEE CS TC-OA Conference on office information Systems* (Cambridge, Massachusetts, United States, April 25 - 27, 1990). F. H. Lochovsky and R. B. Allen, Eds. ACM, New York, NY, 227-237.

Greenberg, S. 1996. A fisheye text editor for relaxed-WYSIWIS groupware. In *Conference Companion on Human Factors in Computing Systems: Common Ground* (Vancouver, British Columbia, Canada, April 13 - 18, 1996). M. J. Tauber, Ed. CHI '96. ACM, New York, NY, 212-213.

Greenberg, S. and Marwood, D. 1994. Real time groupware as a distributed system: concurrency control and its effect on the interface. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (Chapel Hill, North Carolina, United States, October 22 - 26, 1994). CSCW '94. ACM, New York, NY, 207-217.

Greenberg, S., Roseman M., Webster D., and Bohnet R. 1992. Issues and experiences designing and implementing two group drawing tools. In *Proceedings of the Twenty-Fifth Hawaii International Conference on the System Sciences* (Hawaii, January 1992). Vol.4, pp. 139-150.

Gutwin, C. A. 1998 Workspace Awareness in Real-Time Distributed Groupware. *Doctoral Thesis*. UMI Order Number: AAINQ31029., University of Calgary.

Gutwin, C., Greenberg, S. 2002. A Framework of Awareness for Small Groups in Shared-Workspace Groupware. In *Computer-Supported Cooperative Work*, no. 3-4, 411-446.

Hollan, J. and Stornetta, S. 1992. Beyond being there. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Monterey, California, United States, May 03 - 07, 1992). P. Bauersfeld, J. Bennett, and G. Lynch, Eds. CHI '92. ACM, New York, NY, 119-125.

Huang, E. M. and Mynatt, E. D. 2003. Semi-public displays for small, co-located groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 49-56.

Ishii, H. and Kobayashi, M. 1992. ClearBoard: a seamless medium for shared drawing and conversation with eye contact. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Monterey, California, United States, May 03 - 07, 1992). P. Bauersfeld, J. Bennett, and G. Lynch, Eds. CHI '92. ACM, New York, NY, 525-532.

Lai, K., Malone, T. W., and Yu, K. 1988. Object lens: a “spreadsheet” for cooperative work. ACM Trans. Inf. Syst. 6, 4 (Oct. 1988), 332-353.

Malone, T. W. and Crowston, K. 1990. What is coordination theory and how can it help design cooperative work systems?. In *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work* (Los Angeles, California, United States, October 07 - 10, 1990). CSCW '90. ACM, New York, NY, 357-370.

Malone, T. W. and Crowston, K. 1994. The interdisciplinary study of coordination. *ACM Comput. Surv.* 26, 1 (Mar. 1994), 87-119.

Malone, T. W. 1988. What is coordination theory? Massachusetts Institute of Technology, Sloan School of Management. *Working Paper* #2051-88, February.

Malone, T. W., and Crowston, K. G. 1991. Toward an interdisciplinary theory of coordination. *Tech. Rep.* no. 120. Massachusetts Institute of Technology, Center for Coordination Science, Cambridge, Mass.

Malone, T. W., Crowston, K., Lee, J., and Pentland, B. 1993. Tools for inventing organizations: Toward a handbook of organizational processes. In *Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises*. IEEE, New York.

Mantei, M. M., Baecker, R. M., Sellen, A. J., Buxton, W. A., Milligan, T., and Wellman, B. 1991. Experiences in the use of a media space. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology* (New Orleans, Louisiana, United States, April 27 - May 02, 1991). S. P. Robertson, G. M. Olson, and J. S. Olson, Eds. CHI '91. ACM, New York, NY, 203-208.

McGrath, J. E. 1995. Methodology matters: doing research in the behavioral and social sciences. In *Human-Computer interaction: Toward the Year 2000,* R. M. Baecker, J. Grudin, W. A. Buxton, and S. Greenberg, Eds. Morgan Kaufmann Publishers, San Francisco, CA, 152-169.

Morris, M. R., Ryall, K., Shen, C., Forlines, C., and Vernier, F. 2004. Beyond "social protocols": multi-user coordination policies for co-located groupware. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work* (Chicago, Illinois, USA, November 06 - 10, 2004). CSCW '04. ACM, New York, NY, 262-265.

Nacenta, M. A., Pinelle, D., Stuckel, D., and Gutwin, C. 2007. The effects of interaction technique on coordination in tabletop groupware. In *Proceedings of Graphics interface 2007* (Montreal, Canada, May 28 - 30, 2007). GI '07, vol. 234. ACM, New York, NY, 191-198.

Neale, D. C., Carroll, J. M., and Rosson, M. B. 2004. Evaluating computer-supported cooperative work: models and frameworks. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work* (Chicago, Illinois, USA, November 06 - 10, 2004). CSCW '04. ACM, New York, NY, 112-121.

O'Hara, K., Perry, M., and Lewis, S. 2003. Social coordination around a situated display appliance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 65-72.

Olson, J.S. and Teasley, S. 1996. Groupware in the wild: lesson learned from a year of virtual collocation. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work* (Boston, Massachusetts, United States, November 16-20, 1996). M. S. Ackerman, Ed. CSCW ’96. ACM, New York, NY, 419-427.

Pagello, E., Angelo, A., Montesello, F., Garelli, F., Ferrari, C. 1999. Cooperative behavior in multi-robot systems through implicit communication. *Robotics and Autonomous Systems*. 29 (1999) 65-77.

Pinelle, D. and Gutwin, C. 2005. A groupware design framework for loosely coupled workgroups. In *Proceedings of the Ninth Conference on European Conference on Computer Supported Cooperative Work* (Paris, France, September 18 - 22, 2005).

Pinelle, D., Gutwin, C., and Greenberg, S. 2003. Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. *ACM Trans. Comput.-Hum. Interact*. 10, 4 (Dec. 2003), 281-311.

Preguiça, N., Martins, J. L., Domingos, H., and Duarte, S. 2000. Data management support for asynchronous groupware. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, Pennsylvania, United States). CSCW '00. ACM, New York, NY, 69-78.

Rodden, T. and Blair, G. 1991. CSCW and distributed systems: The problem of control. In *Proceedings of the ECSCW European Conference on Computer Supported Cooperative Work*, Amsterdam, Klewar Press, 49-64.

Roseman, M. and Greenberg, S. 1996. TeamRooms: network places for collaboration. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work* (Boston, Massachusetts, United States, November 16 - 20, 1996). M. S. Ackerman, Ed. CSCW '96. ACM, New York, NY, 325-333.

Segal, L. 1994. Effects of Checklist Interface on Non-Verbal Crew Communications, NASA Ames Research Center, *Contractor Report* 177639.

Shoemaker, G. 2001. Single Display Groupware Research in the Year 2000. *Technical Report* TR 2001-1. Simon Fraser University.

Sproull, L. and Kiesler, S. 1991. A Lesson in Electronic Mail. *Connections: New Ways of Working in the Networked Organization*. 177-184.

Stuckel, D. 2008. Reducing the effect of network delay on tightly coupled interaction. *Unpublished thesis (M.Sc.),* University of Saskatchewan.

Stuckel, D. and Gutwin, C. 2008. The effects of local lag on tightly-coupled interaction in distributed groupware. In *Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work* (San Diego, CA, USA, November 08 - 12, 2008). CSCW '08. ACM, New York, NY, 447-456.

Widmeyer, W., Bralwey, L., and Carron, A. 1990. The effects of group size in sport. *Journal of Sport & Exercise Psychology*, 12, 177-190.

# APPENDIX A

# EXPERIMENT MATERIAL

# Consent Forms

**DEPARTMENT OF COMPUTER SCIENCE**
**UNIVERSITY OF SASKATCHEWAN**
**INFORMED CONSENT FORM**

| | |
|---|---|
| Research Project: | **High-Speed Coordination in Groupware** |
| Investigators: | Dr. Carl Gutwin, Department of Computer Science (966-8646)<br>Mutasem Barjawi, Department of Computer Science |

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with group coordination in a fast-paced networked game. We will be recording actions in a multi-player chess game to determine what types of coordination occur.

There will be several sessions of **30 minutes** each, during which you will be asked **to play a game and answer a questionnaire**.

At the end of the study, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations, and will be made available at hci.usask.ca/publications

All personal and identifying data will be kept confidential. If explicit consent has been given, textual excerpts, photographs, or videorecordings may be used in the dissemination of research results in scholarly journals or at scholarly conferences. Anonymity will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

**You are free to withdraw from the study at any time without penalty and without losing any advertised benefits.** Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Carl Gutwin, Associate Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Carl Gutwin, Associate Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca
- Office of Research Services, University of Saskatchewan, (306) 966-4053

Participant's signature:______________________________________________________ Date:______________________

Investigator's signature:_____________________________________________________ Date:______________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Office of Research Services at the University of Saskatchewan.

# DEPARTMENT OF COMPUTER SCIENCE
# UNIVERSITY OF SASKATCHEWAN
# TRANSCRIPT / TEXTUAL EXCERPT CONSENT FORM

| | |
|---|---|
| Research Project: | **High-Speed Coordination in Groupware** |
| Investigators: | Dr. Carl Gutwin, Department of Computer Science (966-8646) |
| | Mutasem Barjawi, Department of Computer Science |

## TRANSCRIPTS

“I, ______________________________________, agree to allow transcripts of my conversations with the investigator or other participants during the experiment to be used for public presentation of the research results in the manner described in the consent form. However, I understand that I will be given the opportunity to read any transcript excerpts that are intended for public participation and to withdraw consent for them to be reported, if so desired. I also understand that I will receive a copy of any transcripts presented publically for my records. I understand that all identifying information will be removed from the transcripts and names will be changed prior to publication. ”

| | | | |
|---|---|---|---|
| **Participant** | | **Investigator** | |
| Name: | ___________________________________ | Name: | Mutasem Barjawi |
| Signature: | ___________________________________ | Signature: | ___________________________________ |
| Date: | ___________________________________ | Date: | ___________________________________ |

## TEXTUAL EXCERPTS

“I, ______________________________________, agree to allow excerpts of text that I wrote to be used for public presentation of the research results in the manner described in the consent form. However, I understand that I will be given the opportunity to read any excerpts that are intended for public participation and to withdraw consent for them to be reported, if so desired. I also understand that I will receive a copy of any textual excerpts presented publically for my records. I understand that all identifying information will be removed from the excerpts and names will be changed prior to publication.”

| | | | |
|---|---|---|---|
| **Participant** | | **Investigator** | |
| Name: | ___________________________________ | Name: | Mutasem Barjawi |
| Signature: | ___________________________________ | Signature: | ___________________________________ |
| Date: | ___________________________________ | Date: | __________________________________ |

# Post Experiment Questionnaire

**Session ID** : __________

**Date** : __________

**Player ID** : __________

**Time** : __________

## SECTION 1: GENERAL INFORMATION

1- **Gennder :** ☐ Male ☐ Female

2- **Age** : __________

3- **How many hours per week (average week) you spend on a computer:** __________

4- **How many hours per week (average week) you play multiplayer games:** _________

## SECTION 2: RTCHESS

**Please mark your answer on each of the following scales:**

5- **To what degree did you and your partner coordinate during the session:**

| | | | | |
|---|---|---|---|---|
| **Very Poor** | | | | **Very Good** |
| **1** | **2** | **3** | **4** | **5** |

**Please Explain:** ____________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

6- **How easy was it to form plans with your partner and actually apply them? Can you give brief examples?**

| | | | | |
|---|---|---|---|---|
| **Very Difficult** | | | | **Very Easy** |
| **1** | **2** | **3** | **4** | **5** |

**Please Explain:** ____________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

____________________________________________________________________________

**7- How well were you able to communicate with your partner during game play?**

| Very Poor | | | | Very Good |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

**Please Explain:** ____________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

**8- For each of the following methods please specify how frequently did you use it to communicate with your partner during game play. If you used other methods, please write them too:**

| | Not Used | | | | Very Frequently |
|---|---|---|---|---|---|
| **Speaking** | 1 | 2 | 3 | 4 | 5 |
| **Mouse Gesturing** | 1 | 2 | 3 | 4 | 5 |
| **Hand Gesturing** | 1 | 2 | 3 | 4 | 5 |

**Please Explain:** ____________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________