UC Irvine
UC Irvine Electronic Theses and Dissertations

Title
Cheap Talk, Trust, and Cooperation in Networked Global Software Engineering: Game Theory Model and Empirical Evidence

Permalink
https://escholarship.org/uc/item/99z4314v

Author
Wang, Yi

Publication Date
2015

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, availalbe at https://creativecommons.org/licenses/by/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org
Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Cheap Talk, Trust, and Cooperation in Networked Global Software Engineering: Game Theory Model and Empirical Evidence

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Yi Wang

Dissertation Committee:
Professor David F. Redmiles, Chair
Professor Debra J. Richardson
Professor Brian Skyrms

2015

Portion of Chapter 4 © 2013 IEEE.
All other materials © 2015 Yi Wang

DEDICATION

To Natural & Intellectual Beauty

TABLE OF CONTENTS

Page

LIST OF FIGURES **vii**

LIST OF TABLES **ix**

LIST OF ALGORITHMS **x**

ACKNOWLEDGMENTS **xi**

CURRICULUM VITAE **xiii**

ABSTRACT OF THE DISSERTATION **xvi**

1 Introduction **1**
1.1 Motivating Observations 3
1.1.1 Empirical Observations 3
1.1.2 Summary 6
1.2 Dissertation Outline 7
1.2.1 Overview of Each Chapter 7
1.2.2 How to Read This Dissertation 9

2 Research Overview and Approach **10**
2.1 Overall Research Questions 10
2.2 Research Approach 11
2.3 Overview of Potential Contributions 12

3 Backgrounds **14**
3.1 Related Work 14
3.1.1 Trust in Globally Distributed Collaboration 14
3.1.2 Informal, Non-work-related Communication in SE and CSCW 16
3.1.3 Summary of Related Work 17
3.2 Game Theory and Human behavior 17
3.3 Stag Hunt Game and Its Evolution 20
3.4 Collaborations in GSE are Stag Hunt 22
3.5 Summary 23

4 Study I: Basic Model and Results 25
4.1 Introduction . . . 25
4.2 Stag hunt with e-Cheap Talk . . . 27
4.2.1 Talk is Still Cheap, but NOT Free . . . 27
4.2.2 The Stag Hunt Game with Cheap Talk . . . 28
4.3 Theoretical Analyses And Results . . . 29
4.3.1 Analysis Method . . . 29
4.3.2 Analysis Results . . . 31
4.3.3 Analytical Answers to Research Questions . . . 34
4.4 Case Study I: Apache Lucene's IRC Discussions (#Lucene-dev) . . . 36
4.4.1 Case Study Design . . . 36
4.4.2 Data Preparation: Message Classification . . . 37
4.4.3 Empirical Results and Findings . . . 39
4.5 Case Study II: Chromium OS' IRC Discussions (#chromium-os) . . . 42
4.5.1 Case Study Design and Data Preparation . . . 43
4.5.2 Empirical Results and Findings . . . 44
4.6 Discussions . . . 47
4.6.1 Implications to Trust and Collaboration Research . . . 47
4.6.2 Methodological Implications . . . 48
4.6.3 Implication to Collaboration Management . . . 50
4.6.4 Implications to Tools Design & Usage . . . 51
4.6.5 Threats to Validity . . . 52
4.7 Summary . . . 53

5 Study II: Simulation in Pseudo Scale-free Network Setting 55
5.1 Motivation . . . 55
5.2 Cheap Talk and Social Network Analysis . . . 58
5.2.1 Social Network Analysis of Distributed Collaboration . . . 58
5.2.2 Linking Cheap Talk with Social Network Analysis . . . 59
5.3 Research Approach . . . 60
5.3.1 Agent Based Modeling and Simulation . . . 60
5.3.2 Applying the Research Method . . . 61
5.4 Virtual Experiment Design . . . 62
5.4.1 Preliminaries: Network and Game . . . 62
5.4.2 Individual-level Strategy Change . . . 66
5.4.3 Group-level Strategy Dynamics . . . 67
5.5 Model Implementation and Virtual Experiment Design . . . 69
5.5.1 Model Implementation . . . 69
5.5.2 Virtual Experiment Design . . . 70
5.6 Results and Findings . . . 71
5.6.1 RQ2-1: Cheap Talk's Positive Impact . . . 72
5.6.2 RQ2-2: Impacts of Degree Distribution . . . 73
5.6.3 RQ2-3: Impacts of Seeding Strategies . . . 75
5.6.4 Sensitivity Analysis . . . 78
5.7 Discussion . . . 80

5.7.1 Implications to Research 80
5.7.2 Implications to Distributed Collaboration Practice 81
5.7.3 Limitations and Future Opportunities 82
5.8 Summary 83

6 Study III: The Role of Baseline Trust 86
6.1 Introduction 86
6.1.1 Empirical Networks vs. Artificially Generated Networks 87
6.1.2 Why Individual Variations on Baseline Trust Are Important? 88
6.1.3 Research Statement 89
6.2 Research Procedures 90
6.2.1 Data Collection and Clean 90
6.2.2 Study Procedure and Main Task 92
6.2.3 The Game Structure 92
6.3 Technical Challenges and Corresponding Solutions 93
6.3.1 Challenge I: Building Empirical Network 93
6.3.2 Challenge II: Extracting Individual's Baseline Trust 97
6.3.3 Challenge III: Specifying Individual's Decision Dynamics 100
6.3.4 Summary 104
6.4 Virtual Experiment Design 106
6.5 Results and Findings 107
6.5.1 Overview of Results 107
6.5.2 Diffusion Trajectories 109
6.5.3 Seeding Strategies 113
6.6 Discussion 116
6.6.1 Implications 116
6.6.2 Design Implications 118
6.6.3 Threats to Validity 119
6.7 Summary 119

7 Summary and Conclusions 121
7.1 Summary 121
7.2 Implications 123
7.2.1 Implications to Theory 123
7.2.2 Implications to Collaboration Management Practice 124
7.2.3 Implications to The Design of Collaboration Tools 125
7.2.4 Implications to Research Methods 126
7.3 Limitations and Future Work 128

Bibliography 131

A Dictionaries for IRC Message Classification in Study I 146
A.1 Lucene 146
A.2 Chromium-OS 147
A.2.1 Reused Keywords 147

A.2.2 New Keywords 147

LIST OF FIGURES

Page

1.1 Collaborators engaging in cheap talk express higher trust levels compared to those who do not. The horizontal bold line in each box represents the median value. 5

3.1 A visual illustration of stag hunt game, License: CC BY-SA3.0, Credits to C.Jensen & G. Riestenberg 2012. 21
3.2 Stag hunt game's payoff matrix and a numeric example. 22

4.1 The payoff structure of the stag hunt game with *e-cheap talk* and a numerical example ($e = 0.2$, and $g = 1$). 29
4.2 Strategy switch process and fixation probability. 30
4.3 The simplex describes the fixation probability between three homogeneous states and their stationary distribution in the long run ($N = 100, \mu = 1, \rho = 1/N$). Only transitions stronger than ρ are shown. A possible route for a non-homogeneous state (represented by the red point in the middle) to reach the stable state is depicted. 32
4.4 The heuristics based decision tree for classifying *cheap talk* messages and corresponding rules. 37
4.5 The dynamics of *cheap talk* and work-related message over time (left axis: *cheap talk*, right axis: work message). 41
4.6 The dynamics of *cheap talk* and work-related message over time. 46

5.1 An example of scale free network: the link structure of a wiki. License: CC BY-SA3.0, Credits to Chris Davis. 64
5.2 The payoff structure and a numeric example of different strategies. 65
5.3 An example of applying the decision model to make decision ($\beta \rightarrow +\infty$). In this example, A changes her strategy after the reviewing process. 67
5.4 A simple example of individual-level strategy changes that lead to the whole group achieving a homogenous "all-cooperative" state (all agents using same strategy) when $\beta \rightarrow +\infty$. The real simulated process may be much more complex than this example. In addition, it is also possible that the group fails to achieve the "all-cooperate" state. 68
5.5 The relationship between degree exponential (x-axis) and success rate (y-axis). 72
5.6 The relationship between degree exponential (x-axis) and average number of strategy reviews (y-axis) before reaching "all-cooperate" state. 74

5.7 The differences between two seeding strategies (Hub seeding vs. Random seeding) on success rate. 76
5.8 The difference in influence on the average number of strategy reviews occurring before reaching an "all-cooperation" state between two seeding strategies (Hub seeding vs. Random seeding). 77
5.9 Frequency of achieving all-cooperate as a function of cost (e) and punishment to defector (g), $\lambda = 2.0$. 79
5.10 Frequency of achieving all-cooperate as a function of cost (e) and punishment to defector (g), $\lambda = 2.6$. 79

6.1 An example of baseline trust's influences on trust and cooperation development in a simple 4-node network. 89
6.2 The Process of Performing Three Major Research Tasks. 92
6.3 The payoff structure and a numeric example of different strategies. 93
6.4 The dynamics of a developer's de-trended trust inferred from their word use from 06/2009-12/2014. The line indicates the average trust over this period. 99
6.5 An comparison of applying the decision model to make decision ($\beta \to +\infty$). In this example, A changes her strategy after the review process. 104
6.6 The developers' social network of LUCENE. The gray-scale indicates each individual's baseline trust. 105
6.7 Different possible full diffusion trajectories on LUCENE network ($c = 0.6$). . . 108
6.8 The change of frequency of full diffusion under different c. 111
6.9 The change of the number of normalized periods to reach full diffusion under different c. 111
6.10 Comparisons of between random seeding and seeding from the hubs (frequency of full diffusion). 113
6.11 Comparisons of between random seeding and seeding from the hubs (periods to reach full diffusion). 113
6.12 Comparisons of between random seeding and seeding from the distrustful (frequency of full diffusion). 114
6.13 Comparisons of between random seeding and seeding from the distrustful (periods to reach full diffusion). 115

7.1 Two research approaches that integrates the abstract modeling and simulation with empirical, observational study. 127

LIST OF TABLES

Page

2.1 Brief Summary of Projects and Collected Data. 12

4.1 Summary and mapping of research questions, theoretical propositions, and empirical findings 48

5.1 Summary of model conditions and parameters. 70
5.2 Summary of findings and corresponding research questions. 72

6.1 An Example of Using Multiple Names in Project Repositories. For privacy concerns, the names and emails are not real. 94
6.2 The basic static of baseline trust. Means are not shown in this table because both means are exactly "0." For some individuals, we cannot resolve the names even using manual mapping, so the sample size is slight smaller than section 6.3.1. 99
6.3 Summary of findings and corresponding research questions. 107

List of Algorithms

Page

1 An Simple Algorithm for Mapping Names and Emails of the Same Developer. 96

ACKNOWLEDGMENTS

A great number of individuals have given me their support and encouragement over my Ph.D career. I would like to express the most sincere gratitude to my faculty advisor and committee chair, Professor David Redmiles. Without his guidance and patience, this dissertation would not have been possible. I have been incredibly lucky to have him as an advisor and a friend in last five years.

I want to thank my dissertation committee of Professor Debra Richardson, and Professor Brian Skyrms for their support over the past years as I developed my research ideas into my dissertation. The discussions with Prof. Richardson helped me better frame the research ideas and this dissertation. Prof. Skyrms' research has inspired me to look into evolutionary game theory. I benefit a lot from Prof. Skyrms' social dynamics seminars in last three years. His encouragement was a true source of inspiration and motivation. I would like to thank my advancement committee of Professor Judith Olson, and Doctor Cory Knobel.

I spent two summers at IBM T. J. Watson Research Center as a research intern. I want to thank Doctor Patrick Wagstrom for being my mentor in both two internships. He is an amazing mentor who is knowledgeable and always ready to help other team members. I also want to thank other colleagues I worked with: Doctors Evelyn Deusterwald, John Richards, Ching-Hua Chen, Steve Abrams, as well as many other friends met in Yorktown Heights.

It is amazing to have a fantastic collection of mentors throughout the years in UCI. Specifically, Doctors Matthew Bietz, and Ban Al-Ani were always there to help me refine the research idea, critique my work, and advise me in many other aspects.

My lab mates and friends at the Department of Informatics have been with me during last five years. Although I cannot mention everyone here, I want particularly to thank my two academic siblings: Erik Trainer and Benjamin Koehne. Many amazing friends have accompanied me on my journey.

The early ideas of this dissertation were also presented and discussed in NSF-funded SCALE meetings. I would like to thank all participants. Particularly, Doctor Anita Sarma in UNL helped a lot in developing and presenting the first study. Many researchers in different institutions contributed many valuable suggestions. Although I cannot mention everyone here, I want particularly to thank Professor Arie van Deursen in TUDelft for sharing a sample of coded LUCENE mailing list data. I also want to thank the feedbacks from the USER group in IBM Almaden Lab, and their generosity for allowing me to play with the Watson personality analytic APIs.

I would not have been able to complete my Ph.D. without the support of my family. My parents Xiuying Zhang and Yudong Wang, who showed me unconditional comfort and love during my entire Ph.D life. They support me to pursue my adventure over 6,000 miles away from home. I would like to thank two important ladies in my life: Shelley Li and Claire Zhang.

The work in this dissertation is partially supported by National Science Foundation under the grants: 0943262 and 1111446.

Part of the text in chapter 4 of this dissertation is a reprint of the material as it appears in (Wang and Redmiles, 2013). The co-author listed in this publication directed and supervised research which forms the basis for the thesis/dissertation.

I want to thank Doctor Colleen Jankovic for her help in editing this dissertation and fixing many writing and language problems.

CURRICULUM VITAE

Yi Wang

EDUCATION

Doctor of Philosophy in Information and Computer Science **2015**
University of California, Irvine *Irvine, CA*

Master of Science in Information and Computer Science **2013**
University of California, Irvine *Irvine, CA*

Master of Science in Computer Science and Technology **2008**
Shanghai Jiao Tong University *Shanghai, China*

Bachelor of Engineering in Computer Science and Technology **2004**
Beijing Language and Culture University *Beijing, China*

RESEARCH EXPERIENCE

Graduate Research Assistant **09/2010–05/2015**
University of California, Irvine *Irvine, CA*

Research Intern **06/2014–09/2014**
IBM Watson *Yorktown Heights, NY*

Research Intern **06/2013–09/2013**
IBM T. J. Watson Research Center *Yorktown Heights, NY*

TEACHING EXPERIENCE

Reader **2010–2011, 2014–2015**
University of California, Irvine *Irvine, CA*

SELECTED PUBLICATIONS

1. Wang, Y., Wagstrom, P., Duesterwald, E. & Redmiles, D. (2014). New Opportunities for Extracting Insights from Cloud Based IDEs. *Proc. of ICSE'14*, NIER Track, **Innovation and Future Impact Award**.

2. Wang, Y., & Redmiles, D. (2013). Understanding Cheap Talk and the Emergence of Trust in Global Software Engineering: An Evolutionary Game Theory Perspective. *Proc. of CHASE'13*, held in conjunction with *ICSE 2013*, pp. 149-152.

3. Al-Ani, B., Bietz, M., Wang, Y., et al. (2013). Globally Distributed System Developers: Their Trust Expectations and Processes, *Proc. of CSCW'13*, pp. 563-573.

4. Al-Ani, B., Wang, Y., Marczak, S., Trainer, E., & Redmiles, D. (2012). Distributed Developers and the Non-Use of Web 2.0 Technologies: A Proclivity Model. *Proc. of ICGSE'12*, pp. 104-113.

5. Wang, Y., Trainer, E., Al-Ani, B., Redmiles, D., & Marczak, S. (2012). Attitude and Usage of Collaboration Tools in GSE: A Practitioner Oriented Theory. *Proc. of CHASE'12*, held in conjunction with *ICSE 2012*.

6. Wang, Y., & Zhang, M. (2010). Penalty Policies in Professional Software Development Practice: A Multi-method Field Study. *Proc. of ICSE'10*, pp. 39-47.

7. Wang, Y. (2009). Building the Linkage between Project Managers' Personality and Success of Software Projects. *Proc. of ESEM'09*, pp. 410-413.

8. Wang, Y., & Shi, H. (2009). Software Outsourcing Subcontracting and Its Impacts: An Exploratory Investigation. *Proc. COMPSAC'09*, pp. 263-270.

MANUSCRIPT IN PROGRESS

1. Wang, Y., & Redmiles, D. (2015). Trust and Coordination in Software Development: An Extensible Evolutionary Model. In Preparation.

2. Wang, Y., & Redmiles, D. (2014). Cheap Talk, Cooperation, and Trust in Global Software Engineering: An Evolutionary Game Theory Model with Empirical Support. Submitted to Empirical Software Engineering Journal.

ACADEMIC SERVICE

Student member: ACM (incl. sigsoft, sigchi), APA, IEEE, Informs.
Reviewer (Conference): CHI, CSCW, ICGSE, ICSM, INTERACT, SCAM, VL/HCC.
Reviewer (Journal): Journal of Mixed Method Research.

PROFESSIONAL CERTIFICATE

06/2014: Data Scientist Certification, John Hopkins University
03/2013: Social and Economic Network Analysis, Prof. Matthew O. Jackson, Stanford University
06/2009: Chartered Statistician, State Statistics Bureau of P. R. China

ABSTRACT OF THE DISSERTATION

Cheap Talk, Trust, and Cooperation in Networked Global Software Engineering: Game Theory Model and Empirical Evidence

By

Yi Wang

Doctor of Philosophy in Information and Computer Science

University of California, Irvine, 2015

Professor David F. Redmiles, Chair

Prior research indicates that trust and cooperation amongst unfamiliar collaborators in remote locations is crucial to Global Software Engineering (GSE) practices. Furthermore, trust and cooperation development requires proper social or technical mediations. Our prior empirical research demonstrates that informal, non-work-related conversation (a.k.a, "cheap talk") over the internet correlates with higher trust and better cooperation. This empirical observation inspired us to hypothesize that cheap talk may also bring about better trust and cooperation. We conducted three related studies to investigate how cheap talk influenced the emergence and diffusion of trust and cooperation in GSE teams. Using game theory, we performed theoretical analyses and agent-based modeling and simulation in which we abstracted GSE collaboration to a variation of the classic stag hunt game. We empirically studied the project communication records of Apache Lucene and Google Chromium OS to validate our approach and to provide a context for our three studies.

The results of the three studies revealed the following. First, all three studies revealed that cheap talk over the internet positively influences trust and cooperation development among GSE practitioners. Second, cheap talk over the internet often functioned as "**catalyst**" and might disappear once trust and cooperation become a team norm. Third, proper seeding

strategies would improve the effectiveness and efficiency of trust and cooperation development when considering the social network's effect. Last, but not least, individuals' baseline trust significantly impacted the dynamics of trust and cooperation development. Our results and findings have important implications for theory development, GSE practices, and research methodology. Theoretically, we developed descriptions and explanations. Practically, the results and findings led to implications for collaboration management, and potential data-driven tools that support game theory analytics. Methodologically, we demonstrated the feasibility of using together, game theory, social network analysis, and agent-based modeling simulation to investigate software engineering's human and social aspects.

Chapter 1

Introduction

Cooperation is one of the foundations of human society (Fehr, Fischbacher and Gächter, 2002; Jones and George, 1998). Yet, cooperation requires a modest level of trust, specifically if one is to overcome the fear of potential losses that could result from others' non-cooperative behaviors (Das and Teng, 1998; McAllister, 1995; Skyrms, 2008). Cooperative behavior is also often viewed as an indicator of trust (Riegelsberger, Sasse and McCarthy, 2003). However, building trust in newly formed relationships, such as globally distributed teams, could be difficult (Jarvenpaa and Leidner, 1998). In most social situations, trusting and cooperating with strangers requires face-to-face interaction (Camera, Casari and Bigoni, 2013).

In the field of global software engineering, however, collaborating with unfamiliar remote colleagues is unavoidable. Given its popularity, globally distributed software developers must collaborate with colleagues they will never meet face-to-face. Interacting with unfamiliar collaborators over the internet can be risky (Olson and Olson, 2013; Wagstrom and Datta, 2014), particularly since there is no way to form accurate expectations to build and maintain trust (Al-Ani et al., 2013). Without trust, software engineers may be unwilling to cooperate with their remote colleagues; yet, their restrictive working circumstances pose significant challenges to building trust. Therefore, we ask: is there an effective and efficient way to support trust development, and hence improve cooperation amongst global software engineers?

Researchers have established the importance of communication in globally distributed teams for coordinating within and across teams, e.g. (Ehrlich and Cataldo, 2012; Herbsleb and Mockus, 2003; Wagstrom, Herbsleb and Carley, 2010; Datta, Sindhgatta and Sengupta, 2012), sharing knowledge, finding expertise, and team and individual performance improvement. Obviously, communication brings work-related benefits such as performance improvements and quality assurance.

Although GSE team members discuss work-related topics, they also spend substantial time having non-work-related conversations (Giuffrida and Dittrich, 2015; Wang and Redmiles, 2013); for example, they make jokes about Google's stock price changes after the release of ChromeBook, showing off their multilingual abilities. This type of conversations' benefits are indirect, and may not bring immediate work-related performance improvement. Conventional wisdom tells us that such behavior might lower someone's performance by occupying work time. Talk is cheap, they say, and it means nothing to your work! Economists Farrell and Rabin (Farrell and Rabin, 1996) claim that cheap talk "does not generally lead to efficiency."

Practitioners tend to hold similar views. For instance, if a manager sees a team member using Facebook messenger during working hours to share funny pictures with colleagues in another country, his or her natural reaction might be to stop it. Most managers perceive this behavior as a kind of cyberloafing, or an unnecessary interruption of normal work, and hence take actions to sanction them (Vitak, Crouse and LaRose, 2011; Ugrin and Pearson, 2013). For example, a study by Robert Half Technology shows that 54% of U.S. multinational companies indicated they have banned workers from using social networking sites, and less than 10% guarantee full access[1].

However, this widespread skepticism about non-work-related talk may be unwarranted, especially when it comes to projects where cooperation is important. Although informal, non-work-related communication does not directly lead to immediate work performance improvement, it may help enhance factors related to teamwork,such as trust and cooperation. Since 2009, the author's research group has performed intensively empirical studies of global software engineering teams (e.g., Al-Ani and Redmiles (2009); Al-Ani et al. (2013)). During this process, we gradually noticed that informal, non-work-related conversations have some association with trust. The next section provides more details.

1.1 Motivating Observations

1.1.1 Empirical Observations

Investigating trust in distributed teams (e.g., Al-Ani et al. (2013)), we found rich evidence of users' favorable attitudes toward, and frequent use of, *cheap talk* during interactions with remote colleagues. Moreover, we noticed an association between *cheap talk* and trust. When we interviewed software developers in distributed teams, some of them reported that non-work-related *cheap talk* helped build trust. According to the interviewees' narratives, we observed *cheap talk's* potential to, for instance, build common ground, develop close interpersonal relationships, and deal with cultural differences. For example, one interviewee emphasized *cheap talk's* trust-building role as such:

> *Yes, I think it's critical* [*for building trust*]. *We do try to bring people over at key planning junctures or transfers of technology. So some of the Poland engineers*

[1] `http://www.wired.com/2009/10/study-54-of-companies-ban-facebook-twitter-at-work/`, accessed in 03/27/2015.

were here in early Q1 for cross training. A couple of them are here this week. I took them out for beer [and] laughs last week. One thing we do is we have a quarterly, we call it off site, where we just go out and do a social activity for team building. So that happens every quarter. When we have visitors we try to do something social in addition to the planning.

In a GSE setting, it is difficult to maintain a "real world" version of face-to-face *cheap talk*. However, collaboration tools allow for the adaptation of *cheap talk* over the Internet. The interviewees mentioned that they spent extra effort and time on non-work-related activities via various tools, such as casual talk over IM, sharing personal pictures, and discussing hot topics. For example, one interviewee reported a preference for instant messaging because "*you can put in little characters like a smiley face or a wink.*" Another interviewee described the type of *cheap talk* afforded through remote online communication–"*Where are you calling from?..what's the weather like? is it hard to have to work at home in the evenings?*"–and how it can be used "*to maybe build up some sympathy.*"

Example 1. *...but I like instant messaging. Mainly because with instant messaging you can put in little characters like a smiley face or a wink or something like that. I like that.*

Example 2. *Well, first time I am talking with someone, well, "Where are you calling from? Oh." Because maybe we are in different time zones, at home or whatever. "What's the weather like?" I mean, "Oh, is it hard to have to work at home in the evenings?" to maybe build up some sympathy. "Is it hard for you because you have to call in, in the evening to talk with us?"*

However, some interviewees did not express any interest in engaging in *cheap talk* with remote colleagues, and even considered *cheap talk* valueless. Based on their opinions on *cheap talk*, we coded interviewees into two categories: (*non-Cheap talk* and *Cheap talk*).

We also computed each individual's average trust towards their collaborators. During the interview, each interviewee was asked to locate his or her collaborators on a trust spectrum, which was then coded into a 5-level interval scale to produce an aggregated trust score. In total, 43 interviews were collected. However, 2 of these were incomplete and did not include a trust score. Thus, in the end, we coded and analyzed 41 of these, classifying 9 as *non-Cheap talk* and 32 as *Cheap talk*.

We performed a simple *independent sample t-test* that revealed significant differences between these two groups' trust: *P-value: 0.013*, *Effect Size: 0.921 (Cohen's d)*, and hence significance at the 0.05 level while the *effect size* indicates the sample size is sufficient. The interviewees in the *Cheap talk* category exhibited higher trust (mean: 4.152) than those in *non-Cheap talk* category (mean: 3.607). Figure 1.1 illustrates the results.

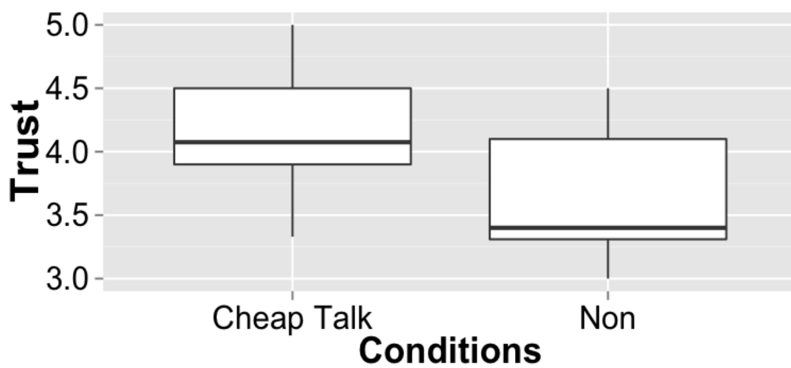

Figure 1.1: Collaborators engaging in cheap talk express higher trust levels compared to those who do not. The horizontal bold line in each box represents the median value.

Does this mean these interviewees build some kind of "friendship" such that they trust each other and converse about their lives? Probably not. One interviewee commented on sharing his personal life stores:

> *I like to share something about my hobby, favorite food,* [*with remote team members*]*. But they could never be my friends. You cannot develop friendships with a guy you have never seen in physical world. .... I want them to know I am not a*

crazy nerd.

1.1.2 Summary

The observations shows three key points:

1. Cheap talk is likely to positively influence trust and cooperation development among GSE practitioners.

2. GSE practitioners are usually willing to use cheap talk over the internet when communicating with remote collaborators, although they may need to pay a small cost for it.

3. GSE practitioners seem to view cheap talk as a strategy for demonstrating their interest in cooperation, rather than as a step toward developing a close interpersonal relationship.

The above key findings lead us to wonder whether low-cost cheap talk over Internet is an efficient mediation for trust and cooperation development. If it is, we next wonder how it functions within complex social and technical team structures. We decided to view it as a developer's strategic choice, which enables us to analytically and empirically investigate a GSE team's cooperation and trust dynamics. The following chapters detail the backgrounds, designs, and results of a series studies seeking to achieve this goal.

1.2 Dissertation Outline

1.2.1 Overview of Each Chapter

This chapter explains our research study's motivation and its basic scope. The remainder of the dissertation is organized as follows.

Chapter 2 presents the overall research research questions and a high-level description of the methodology we employed to tackle the research problem, including the empirical datasets we used. We also briefly discussed the potential contributions of answers to our research questions.

Chapter 3 describes previous research on trust, non-work-related talk, and cooperation in the areas of global software engineering, computer supported collaborative work, among others. Chapter 3 also explains why we need different perspectives for conceptualizing and exploring cheap talk in global software engineering teams. It also briefly discusses using game theory as a lens for studying human and social behaviors. Chapter 3 also provides a brief introduction to the stag hunt game, which is the fundamental game structure for this dissertation. Moreover, we explain the rationale of abstracting global software engineering collaborations as stag hunt.

Chapter 4 introduces the first study. In this chapter, we modify the classic stag hunt game by allowing the association of "cheap talk" and "cooperation" as a new concrete strategy. Then, we derive a set of theoretical results through reasoning and simulation with the new game model and with standard evolutionary game theory techniques. We validate these results with two empirical case studies on open source projects: Apache LUCENE and Google CHROME-OS. We found general support for our theoretical results[2].

[2]A portion of this chapter (section 4.2-4.3) is © 2013 IEEE and reprinted, with permission, from [Wang, Yi, and D. Redmiles, "Understanding Cheap Talk and the Emergence of Trust in Global Software Engineering: An Evolutionary Game Theory Perspective." Proc. CHASE, 149-152, 2013].

Chapter 5 addresses one of the first study's limitation, which is that we assume that team members are well-connected without considering the potential influence of the social network formed by individual team members. This chapter also describes the second study, aiming to tackle this problem with agent-based simulation, and details a virtual simulation experiment design and its results. It builds relationships between trust and cooperation development at the team level and the degree characteristics of networks, as well as demonstrates the importance of selecting the proper set of "hub" nodes as seeds for accelerating the process of trust and cooperation development.

Chapter 6 further extend the first and second study, reporting a study that utilizes empirical networks rather than artificially generated networks for simulating a team's trust and cooperation dynamics. Using empirical networks also enables us to incorporate important individual characteristics, such as an individual's baseline trust[3], into the investigation. We employed Natural Language Processing techniques (NLP) to extract network and individual characteristics. The results show: (1) that considering baseline trust yields much richer dynamics, (2) the trajectories of non-traditional trust and cooperation diffusion, and (3) that proper seeding strategies would help the trust and cooperation development process.

Chapter 7 summarizes this dissertation's contribution and concludes with a set of theoretical, practical, methodological, and design implications. These implications derive from the theoretical and empirical results of our three major studies. Chapter 7 also discusses some of the work's limitations. We also discussed some future directions, in order to push this research further.

[3] In this dissertation, we use term "baseline trust" to refer an individual's general, global tendency to perceive the trustworthiness of other individuals (or other entities, such as organizations) (Driscoll, 1978)

1.2.2 How to Read This Dissertation

Chapter 2 discusses the background of the research described in this dissertation. Section 3 includes the backgrounds shared by all three studies. In this chapter, section 3.3 and 3.4 may have special importance because it introduces the classical stag hunt game and why software development activities can be abstracted as stag hunt. The chapters 4 to 6 contain this thesis' main contributions. Each of them presents one of three independent, yet closely interrelated studies. Every later chapter can be viewed as an incremental extension to the prior one. Nevertheless, each chapter is almost self-contained and can be independently read.

We use several abbreviations throughout this thesis. These abbreviations are usually well-adopted in literature from several research communities. If there is no special notation when they appear, the terms and corresponding abbreviations are: *global software engineering (GSE)*, *software engineering (SE)*, *computer supported collaborative work (CSCW)*, *human-computer interaction(HCI)*, *computer mediated communication (CMC)*, *evolutionary game theory (EGT)*, *evolutionarily stable state (ESS)*, Natural Language Processing (NLP), and *social network analysis (SNA).*

Chapter 2

Research Overview and Approach

2.1 Overall Research Questions

Prior literature and empirical observations lead us to consider whether online cheap talk could facilitate rapid trust and cooperation development. We seek to investigate this question from a novel strategic interaction perspective (rather than, for instance, a social relationship perspective) in order to precisely and dynamically describe individuals' strategic behaviors and group dynamics. We adopted evolutionary game theory as our major theoretical tool for this purpose, modifying classic stag hunt game (Skyrms, 2001) to describe GSE collaboration and cheap talk. Since software development teams are also networked organizations (Bird et al., 2006; Toral, Martínez-Torres and Barrero, 2010; Wolf et al., 2009), we aim to identify how global network characteristics influence a trust and cooperation development process that involves cheap talk. Moreover, if we view trust as a relatively stable personality trait, we see that individuals significantly vary in their baseline trust. Consequently, it makes sense to ask whether an individual's baseline trust affects group-level trust and cooperation dynamics. Thus, our research questions are as follows:

RQ 1: Can trust and cooperation emerge as a GSE team norm with the presence of cheap talk over the internet?

RQ 2: If yes, how does cheap talk over the internet support the development of trust and cooperation?

RQ 3: Since GSE teams are networked groups and team members have different baseline trust levels, how do these factors influence trust and cooperation development with cheap talk over the internet?

2.2 Research Approach

We designed three studies that incrementally progress toward answering these research questions and form the main body of this dissertation. Study I focuses on a well-mixed population without considering a network effect. Study II adds basic network settings through simulations on artificially-generated pseudo-scale free networks. Study III introduces individual variances, while using empirical networks derived from observational data. From Study I to Study III, the research becomes progressively more realistic and relevant to GSE practices.

We plan to take a mixed-method approach to tackling the research questions. Three major methods will be applied in this research: Game Theory Modeling, Agent-Based Simulation, and Empirical Studies. Game theory modeling and agent-based simulation are used to model individual and group strategies and dynamics. Empirical studies not only validate the modeling and simulation results, but also provide simulation infrastructures (Study III). We will introduce the details of each study's research approach in corresponding chapters. For our empirical study, we collected project data from three projects repositories: Lucene, Chromium-OS, and Asterix. The first two studies are large open source projects and serve as the main empirical datasets used in this research, the third dataset is relative small,

Project	Domain	Data Utilized in This Research
Apache Lucene	Information Retrieval	IRC messages, Mailing lists, Bug tracking, Commit and Build records (only for identifying user), Source code (only for reference).
Google Chromium-OS	Linux-based Operating System	IRC messages (unofficial), Mailing lists, Discussion Group, Bug tracking, Commit and Build records (only for identifying user), Source code (only for reference).
Asterix	Data Management	Mailing lists, Bug tracking, Building messages, Source code.

Table 2.1: Brief Summary of Projects and Collected Data.

and is only used for a preliminary proof of concept exploration. Table 1 briefly introduces these datasets.

2.3 Overview of Potential Contributions

The research in this dissertation investigates the influence of cheap talk on trust and cooperation development in the global software engineering context. Using game theory as a theoretical lens, the proposed dissertation combines various research techniques and offers the following potential contributions to related research fields (e.g., Software Engineering, Computer-Supported Collaborative Work):

- Explains the role of online cheap talk in trust and cooperation development, focusing in particular on network and individual baseline trust variations.
- Refines the theoretical explanations by validating and supporting them with empirical evidence from real world global software engineering practices.

- Develops implications for trust and cooperation improvements in collaboration management and intelligent decision tool design.
- Demonstrates the feasibility of employing game theory in data-driven, agent-based modeling and simulation to study social and technical problems in GSE.

The next chapter introduces the details of the three studies. Each study is driven by its own research questions, which are derived from the three main research questions proposed in section 2.1.

Chapter 3

Backgrounds

Coders will survive an Ice Age,

because they know how to hunt.

Anonymous, *stackoverflow.com*

3.1 Related Work

3.1.1 Trust in Globally Distributed Collaboration

Trust is important for teamwork, leadership, and organizational process in globally distributed collaboration (Jarvenpaa, Shaw and Staples, 2004), and it has been intensively investigated in SE and CSCW literature. Due to the lack of shared context and common ground, trust development among globally distributed teams differs from co-located ones (Olson and Olson, 2000).

The global software engineering (GSE) research on trust reflects researchers' broad interests. First, researchers try to understand why there is a lack of trust in globally distributed teams by observing real global software development practices. For example, Moe and Šmite (2008) conducted a multiple case study using data collected from four projects, finding many contributing factors. With this knowledge of factors that influence trust, CSCW researchers investigated how to utilize and design proper communication technology to improve its development, for example: Bos et al. (2002); Trainer and Redmiles (2012), etc. Social mechanisms and management practices have also been used to enable trust and cooperation's development, e.g., via the initial interactions that kick off the project (Zolin et al., 2004), individual's knowledge (Mortensen and Neeley, 2012), and so on.

Trust influences team performance in various ways. Charness and Dufwenberg (2006) used a trust game in a lab experiment setting to study the association between trust and reciprocal behaviors. In (Jarvenpaa, Shaw and Staples, 2004), the authors showed that early trust directly influences team cohesiveness at a collaboration's beginning. However, in later phases, trust comes to play a moderating role in the relationship between team communication and perceptual outcomes. Our prior work 2015*b* developed an extensible evolution game theory model to explore the co-evolution between trust and coordination. There is also research integrating the construct of "trust" with other factors in order to investigate their joint influence on team performance; for instance, Chua, Morris and Mor (2012) studied the relationship between cultural intelligence and affect-based trust, and their combined effect on creative collaboration performance.

We admittedly exclude a wealth of literature in this short introduction[1].However, as reflected in the work we discuss above, we conclude that the research on trust in globally distributed collaboration is very diverse. Moreover, trust's importance as a foundation for successful collaborations is well acknowledged by researchers in many areas.

[1] A more comprehensive, yet more general review of trust appears in Rousseau et al. (1998).

3.1.2 Informal, Non-work-related Communication in SE and CSCW

Building on the literature on non-work social interactions that occur in the workplace, in a previous paper we first introduced the formal concept of *cheap talk* in software engineering(Wang and Redmiles, 2013). Workplace social interactions are usually conceptualized in terms of "socialization" (Dittrich and Giuffrida, 2011), "small talk" (Cassell and Bickmore, 2003; Steed et al., 2003), and "water cooler" (Herbsleb et al., 2002), and their importance has been noted in (Ducheneaut, 2005). For instance, Dittrich and Giuffrida (Dittrich and Giuffrida, 2011; Giuffrida and Dittrich, 2015) identified socialization as one of four usage dimensions of IM in their qualitative study of a Danish/Indian global software team. They found that a few of the *Skype* chats were purely social, such as everyday chats around the water cooler in co-located settings, and argued that informal communications provide a channel for building trust and social relationships. However, they neither explained why the social chats helped to build trust, nor did they assess the measurable influence on cooperation. Similar results were also reported by Cramton and Hinds (Cramton and Hinds, 2007), who argued that casual interactions eased cooperation among individuals from three different cultures.

Guzzi et al (Guzzi et al., 2013) reported on a qualitative study of the LUCENE mailing list, focusing on the communication patterns of open source developers. Three of their four "social interaction" categories were work-related, and only one ("social norm": 3 in 506) had nothing to do with work. The other three ("acknowledgement of effort," "coordination," and "new contributors") were at least partially related to work, as their labels suggest. In general, SE researchers are beginning to pay more attention to how personal and affective factors influence software development collaboration. Some preliminary work by Calefato et al. (Calefato, Lanubile and Sportelli, 2013) demonstrates that informal information from social media can augment social awareness and improve trust.

3.1.3 Summary of Related Work

It is fair to claim that both trust and informal, non-work-related communication in GSE have attracted intensive interest among researchers in different areas. However, there are few studies that investigate them together; rather, they are often only informally discussed together, or discussed as an aspect of a broader concept. For example, sharing informal messages was discussed as one option amongst a set of communication channels that jump-start trust building (Zheng et al., 2002). Moreover, prior studies reviewed above emphasize the role of a "*social relationship*" in informal interactions (Mislin, Campagna and Bottom, 2011). However, the concept of social relationships cannot fully explain an individual's strategic choices, nor how a communication strategy evolves into a norm at the team level. Indeed, our motivating example in chapter 1 provides empirical evidence supporting the idea that informal communications may not necessarily lead to "social relationships."

To further investigate the interplay between informal, non-work-related communication and trust, we argue that researchers need to look at the issue from another perspective. In this dissertation, we view informal, non-work-related conversation among GSE practitioners as a strategic behavior emerging from their perception and evaluation of its *cost and benefit*. This not only allows us to link two research streams, but also gives us an opportunity to leverage an analytical framework (namely, evolutionary game theory) to precisely describe, explain, and predict GSE teams' trust and cooperation dynamics. In fact, the concept of "cheap talk" more or less reflects the "cost and benefit" view of strategic behaviors.

3.2 Game Theory and Human behavior

Game theory provides a framework to analytically model and investigate decision-maker strategic interaction situations, in which each participant's utility for the outcome depends on both individual and overall team decisions and relative positions (Easley and Kleinberg, 2010). Game theory has been successfully applied in various research areas (Bowles and Gintis, 2011), including Political Science, Management, Sociology, Computer Science, Network Science, and even Professional Sports management. Combined with Network Science, game theory can powerfully assess, predict, and improve individual behavior and group dynamics within an organization, which can be abstracted as a social network. For example, Kreindler and Young (2014) and Young (2011) studied the spread of innovation in social networks with a coordination game where individuals play 2×2. Game theory is also used to design social mechanisms and policies. For instance, game theory analysis of electronic commerce and communication networks helps design reliable market mechanisms (Easley and Kleinberg, 2010), such as reputation systems (Friedman, Resnick and Sami, 2007) and decentralized trust aggregation systems based on individual-level behaviors and interactions (Bachrach et al., 2009). Skyrms, Avise and Ayala (2014) summarized the representative progress of applying evolutionary game theory in various social science domains.

Organizations are coordinated action systems comprising individuals and groups with differing preferences, information, interest, and knowledge (March and Simon, 1993). In essence, an organization is (1) a multi-agent system, with (2) identifiable boundaries and structures, as well as (3) system-level goals, toward which (4) the constituent agents are expected to individually or collectively contribute by interacting with others (Puranam, Alexy and Reitzig, 2013). Game theory social network models reflect these organizational essences, only requiring a few assumptions. Researchers can easily validate its conditions and evaluations in experimental and natural settings (Gintis, 2000). For example, Banerjee et al. (2013) modeled the behaviors and relationships among individuals in a microfinance network and empirically validated it with data from a rural Indian village. The results directly modeled the characteristics of a strong microfinance network, and, moreover, provided insight into how one might design a sustainable one. At the individual level, game theory can help determine optimal interpersonal interaction strategies. Jackson, Rodriguez-Barraquer and Tan

(2012) employed similar techniques to investigate the robustness of the informal exchange of favors among a society's members.

Some studies have already applied (evolutionary) game theory to research social networks' characteristics and dynamic evolutions, validating their results with real network data. For example, using a dynamic network model, Skyrms (2009) studied the evolution of interaction structures and the co-evolution of structure and strategy. Young (2011) extended his adaptive play (Young 1993) to fixed networks in order to describe and explain the failure and success of technology/policy diffusion in a social network. Furthermore, Nowark and Sigmund investigated the evolution of indirect reciprocity in social network configuration, and Jackson and Zenou (2013) and Szabó and Fath (2007) provided a detailed networked games review.

Game theory is particularly well-suited to investigate cheap talk in GSE practices. It opens up new possibilities, allowing researchers to: (1) abstract complex behaviors (cheap talk, cooperate) to analytical tractable models without making excessive assumptions, (2) assess and predict individual level strategic dynamics, (3) assess and predict group level dynamics, such as the proportion of team members adopting a specific strategy, and (4) integrate with other social theories, such as social network theory. Since we sought to analytically investigate cheap talk's influence on trust and cooperation development in GSE settings, game theory provides a solid framework.

Of course, game theory's applications for studying social and technical issues in software engineering go beyond cheap talk. For example, signaling game Skyrms 2010 has made significant contributions. In the professional software development world, signals are widely present (e.g., impression formation: Marlow and Dabbish (2013), communication among locations: Matthiesen, Bjørn and Petersen (2014)). Moreover, people rely on signal systems to cooperate and achieve team goals, especially in global software engineering, since remote developers lack face-to-face interaction and close monitoring (Crowston et al., 2012). In current practices, many signals are easily manipulated. For instance, commit history may be interpreted as a "signal" of work progress; however, people can produce a more appealing commit history by retaining trivial commit changes. In many cases, their collaborator may not be able to detect these distortions e.g., Pentland and Heibeck (2008), Marlow, Dabbish and Herbsleb (2013), and thus it may be necessary to diagnose existing signal systems and design more authentic, manipulation-resistant ones. However, this research direction is beyond the scope of this dissertation.

3.3 Stag Hunt Game and Its Evolution

Stag hunt originated in Jean Jacques Rousseau's A Dissertation on the Origin of the Inequality of Mankind:

> If a deer was to be taken, every one saw that, in order to succeed, he must abide faithfully by his post: but if a hare happened to come within the reach of any one of them, it is not to be doubted that he pursued it without scruple, and, having seized his prey, cared very little, if by so doing he caused his companions to miss theirs.
>
> -Jean Jacques Rousseau: *A Dissertation on the On the Origin of the Inequality of Mankind*, The Second Part, 1754, translated by G.D.H Cole (Rousseau, 1950)[2].

In his A Treatise of Human Nature, Hume expressed a similar idea using a different scenario:

> Two neighbors may agree to drain a meadow, which they possess in common, because 'tis easy for them to know each others mind; and each must perceive that the immediate consequence of his failing in his part is the abandoning of

[2]Available online: `http://goo.gl/ZaeQ5X`.

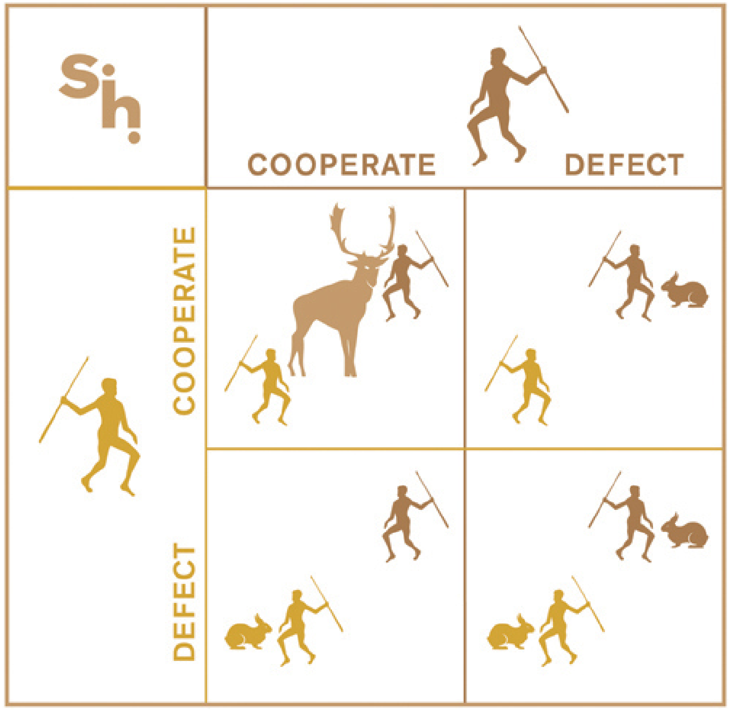

Figure 3.1: A visual illustration of stag hunt game, License: CC BY-SA3.0, Credits to C.Jensen & G. Riestenberg 2012.

> the whole project. But 'tis very difficult, and indeed impossible, that a thousand persons should agree in any such action.
>
> -David Hume: *A Treatise of Human Nature*, Book I, Part II Section VII, 1738, as cited in (Skyrms, 2008).

The classic stag hunt game is a non-zero-sum, 2 by 2 game in which each player has two strategic choices: *cooperate* or *defect* (see fig. 3.1). In ancient times, two men hunt for food. If both defect, they would hunt individually, and each would get a hare. If both cooperate, they could kill a stag, and each would receive one half stag. If one cooperates while the other defects, the *cooperator* would receive nothing and the *defector* would receive a hare. Formally, the stag hunt game can be represented by the first payoff matrix in fig. 3.2 if $R > T > P > S$[3]. Compared to the Prisoner's dilemma (if $T > R > P > S$), "*cooperate*" is not strictly dominated by "*defect*" in this game. The state of (*cooperate*, *cooperate*) is a

[3] T may also equal to P.

$$\begin{array}{ccc} & C & D \\ \begin{array}{c} C \\ D \end{array} & \multicolumn{2}{c}{\begin{pmatrix} R & S \\ T & P \end{pmatrix}} \end{array} \qquad \begin{array}{ccc} & C & D \\ \begin{array}{c} C \\ D \end{array} & \multicolumn{2}{c}{\begin{pmatrix} 2 & 0.5 \\ 1.5 & 1 \end{pmatrix}} \end{array}$$

Figure 3.2: Stag hunt game's payoff matrix and a numeric example.

payoff-dominated equilibrium, while (*defect*, *defect*) is a risk-dominated equilibrium. Which one can be achieved is determined by players' belief in their opponent: namely, their trust (Skyrms, 2001, 2004).

Even in economics literature, the stag hunt game is generally less used for investigating the evolution of human cooperation. However, it is more natural representation of real world cooperation (Skyrms, 2001). The most significant advantage of the stag hunt game is that it allows "*cooperation*" to be achieved by rational individuals. In stag hunt, (*cooperate*, *cooperate*) is an equilibrium, since once both parties agree on "cooperate", neither intends to defect. But (*cooperate*, *cooperate*) is not the only equilibrium; (*defect*, *defect*) is also an equilibrium. (*defect*, *defect*) is risk-dominance equilibrium, and even more probable to reach in the long run with mutation (Kandori, Mailath and Rob, 1993; Young, 2001).

3.4 Collaborations in GSE are Stag Hunt

The stag hunt game is a natural metaphor of dyadic (one-to-one) collaborations[4] in software engineering activities. In many cases, developers do not necessarily need to cooperate with others to complete their jobs ("receive a *Hare* as payoff"), even when their work items are highly interdependent. However, low cooperation may influence their work's quality. Cataldo et al. (Cataldo, Herbsleb and Carley, 2008) pointed out that the communication amongst

[4]In the scope of this dissertation, cooperation is restricted to dyadic interactions to simplify the analysis and discussion. In fact, multi-person cooperation can be conceptualized as a series of dyadic ones. That is, if m people cooperate, one can assume there are at most C_m^2 pairs of dyadic cooperators.

developers can significantly influence the quality of a software system, even if work items can be independently completed. Thus, collaboration can produce higher quality work ("receive a half *Stag* as payoff"). In some cases, a software engineer may believe that her colleague will cooperate, but things do not go as she expects. Thus she may experience some "unfavorable" results (e.g., fail to deliver a commitment on time) due to the other's "defect," whereas the other can still achieve the utility of individual action ("receive a *Hare* as payoff"). Hence, dyadic collaboration in software development can be analogous to stag hunt, allowing us to use standard EGT techniques to investigate software development collaborations.

Some empirical evidence indirectly demonstrates the feasibility of abstracting GSE collaborations as stag hunt game. In (Wagstrom, 2009), the author found the social technical congruence[5] (STC) of the project they surveyed is either high or fairly low; few projects have medium-level STC. The distribution of STCs shows twin-peak pattern, which indicates that some teams can eventually turn all members into "cooperators," whereas others are almost entirely comprised of "defectors." This fits stag hunt game's two possible long-term states (all-cooperate and all-defect).

3.5 Summary

This chapter first briefly reviewed related work, which confirmed that we need an alternative perspective that views cheap talk as a strategic behavior. The new perspective enables us to utilize game theory as an analytical tool to understand, assess, and predict how cheap talk influences trust and cooperation development. Next, we introduced the classic stag hunt game, as well as its dynamic characteristics. We developed an argument that collaboration in GSE practices can be abstracted as stag hunt game. The discussions in this chapter form the foundation for the following three concrete studies, which are detailed in the next

[5]social technical congruence is a measure of how well the collaboration and communication fits the source code dependency.

chapters.

In the next chapter and beyond, we presents the main contributions of this dissertation. Chapter 4 introduces a modified classic stag hunt game that conceptualizes cheap talk over the internet as part of a new defined strategy. We analytically demonstrate how low-cost cheap talk supports trust and cooperation's development, and we validate these theoretical results with empirical evidence from two case studies.

Chapter 4

Study I: Basic Model and Results

> Cheap talk matters!
>
> Brian Skyrms, *Social Dynamics*, p.295

4.1 Introduction

In chapter 3, we demonstrated that stag hunt is a natural metaphor for GSE collaborations. However, the classic stag hunt game cannot guarantee that the "cooperate" strategy will become a group-level norm. In chapter 1, the motivating empirical study shows cost-incurring *cheap talk* correlates with higher trust and better cooperation in GSE practices. Hence, we can hypothesize that it may also bring about improved trust and cooperation. Can we analytically demonstrate the role of low cost cheap talk as an effective way to develop trust and cooperation?

We seek to investigate this question under the framework of stag hunt game. To do so, we modify its classic form and add a new strategy "*cheap talk-cooperate*" by associating the concrete "cooperate" action with *cheap but not costless talk*. The modified game is played by a fixed number of players, simulating a team with fixed number of developers, allowing us to describe the dyadic interactions among team members, and enabling players to update their strategies through social learning. Using EGT analytical techniques such as Nowak (2006), we can explore individual strategic behaviors and social group interaction trends in term of the proportions of members using different strategies.

More specifically, this study investigates the following research questions:

RQ1-1: Can *cheap talk-cooperate*[1] self-reinforce if it is secured by situation-intrinsic incentives?

RQ1-2: Can trust emerge and eventually become a cooperation-ensuring team convention if defectors' punishment compares to the cost of *cheap talk*?

RQ1-3: What are the long-term dynamics (including frequency) of using *cheap talk*, particularly as cooperation and trust are established over time?

To validate the our theoretical model's results, we performed two empirical case studies on Apache Lucene and Chromium-OS by mining their logged IRC discussions. Our analyses provided general support for our theoretical results and predictions. For example, we observed: consistency between cheap talk and cooperation's development; their precedence relationship; and cheap talk's disappearance over time. From a methodological perspective, this study demonstrates the feasibility of our novel approach, which integrates a theoretical game theory model with an empirical study to develop generalizable, rigorous GSE knowledge.

The rest of this chapter is organized as follows. Section 4.2 introduces the extended stag hunt game, and Section 4.3 presents the theoretical analyses and results respectively. Two empiri-

[1] we use "C-C" to refer this strategy in this chapter.

cal case studies on the development IRC discussions of Apache Lucene and Chromium-OS are presented in section 4.4 and 4.5 to validate the model and its analytical results. Finally, sections 4.6 and 4.7 discuss related issues and conclusions.

4.2 Stag hunt with e-Cheap Talk

We extend the classic stag hunt game to analytically explore the role of *cheap talk* in trust and cooperation development. In this section, we develop a new game that associates *cheap talk* with a new interpersonal interaction strategy.

4.2.1 Talk is Still Cheap, but NOT Free

A typical way to model *cheap talk* is to treat it as a cost free *signal* with no predefined meaning. Imagine that you talk about your dog with your officemates; you pay nothing for this type of conversation. Although irrelevant to your work, cheap talk conveys signals, which accrue meaning with the progress of interaction, and thereby increase the possibility and frequency of collaboration (Santos, Pacheco and Skyrms, 2011). However, in GSE, *cheap talk* often occurs over the Internet via various collaboration tools like IM. *Cheap talk* requires individuals to expend extra time and effort; thus, although it is not free, this type of interaction is *cheap* [2] when compared to the cooperation's benefits. In this sense, *signal* fails as an accurate abstraction for cheap talk; although it is still cheap, *cheap talk* is more accurately understood as a concrete action, rather than a costfree signal. We denote this type of *cheap talk* as "*e-cheap talk*."

[2] In economics literature, the cost of cheap talk is zero. However, in the scope of this paper, we assume the cost is "minimal" instead of "zero".

4.2.2 The Stag Hunt Game with Cheap Talk

To capture the essence of *e-cheap talk*, we slightly alter the classic stag hunt game. The new game has three strategies instead of two. To initiate an *e-cheap talk* over the Internet, the proposer must pay a small management cost (e.g., spend extra time to upload pictures). We reasonably assume that this cost is small compared to cooperation's benefits ($e << R$). It may not be constant, but it is likely that the fluctuations do not span different orders of magnitude. Furthermore, to simplify the discussion, we suppose that players' preferences remain consistent (i.e., there is no execution noise). If a player decided to start *e-cheap talk*, she would use the "*cooperate*" strategy in the following interaction. Thus, we add a new strategy called "C-C" whereby the new game contains three strategies $\{C\text{-}C, C, D\}$.

If two players choose C-C in their interaction, they share the cost of *e-cheap talk*, and thereby each potentially receive R-$e/2$ payoff. Note that, even in cases in which only one player pays the cost, the **average** payoff of playing (C-C, C-C) is still R-$e/2$ in the long run. If one plays C-C while the other plays C, the first will pay the cost all by herself. Her payoff is (R-e, R). If a player of C-C meets another player of D, the second player may be punished for refusing to cooperate.[3] We assume that the punishment equals g and the C-C player receives a compensation (the same amount as the punishment), and thus the payoff of this interaction is (S+g-e, T-g). Retaining this part of the classic payoff structure,[4] Figure 4.1 illustrates a modified stag hunt game with *e-cheap talk* and a numerical example.

[3]Punishment may take many forms, e.g., reputation loss.

[4]It is reasonable to assume that, in the classic game, the player who plays *defect* would not be punished since she has no reason to give up a risk dominated strategy without any hint of her opponent's action.

$$
\begin{array}{c}
\\
C\text{-}C \\
C \\
D
\end{array}
\begin{array}{c}
\begin{array}{ccc} C\text{-}C & C & D \end{array} \\
\begin{pmatrix}
R\text{-}e/2 & R\text{-}e & S\text{-}e\text{+}g \\
R & R & S \\
T\text{-}g & T & P
\end{pmatrix}
\end{array}
\qquad
\begin{array}{c}
\\
C\text{-}C \\
C \\
D
\end{array}
\begin{array}{c}
\begin{array}{ccc} C\text{-}C & C & D \end{array} \\
\begin{pmatrix}
1.9 & 1.8 & 1.3 \\
2 & 2 & 0.5 \\
0.5 & 1.5 & 1
\end{pmatrix}
\end{array}
$$

e=0.2, g=1

Figure 4.1: The payoff structure of the stag hunt game with *e-cheap talk* and a numerical example ($e = 0.2$, and $g = 1$).

4.3 Theoretical Analyses And Results

To simulate the team setting, the new stag hunt game will be played by a fixed number (N) of players. Using the analysis technique we will introduce in section 5.1, we reveal the long-term dynamics of cooperation with a 100-member team as an example, and the conditions that enable long-term cooperation (see section 5.2). The findings provide answers to the research questions highlighted in the introduction.

4.3.1 Analysis Method

Our analysis is based on EGT methods for finite populations (e.g., (Nowak, 2006; Nowak et al., 2004)) and stochastic process analysis techniques (e.g., (Fudenberg and Imhof, 2006; Traulsen, Nowak and Pacheco, 2006; Fudenberg and Imhof, 2008)). In EGT, the individuals' payoff represents their *fitness* or *social success* (Nowak, Tarnita and Wilson, 2010). A population's strategy change is followed by learning dynamics; i.e., the most successful individual will tend to be imitated by the others. In our discussion, we use simple standard birth and death chains[5] to describe the switch between strategies. We follow the method in (Nowak, 2006) to assume that in any period there are at most two coexisting strategies (for convenience: i and j). Suppose one player decides to try i in a state in which the whole

[5]Informally, for a fixed population, a single individual strategy switch from j to i can be viewed as the "*death*" of a j-player and the "*birth*" of an i-player.

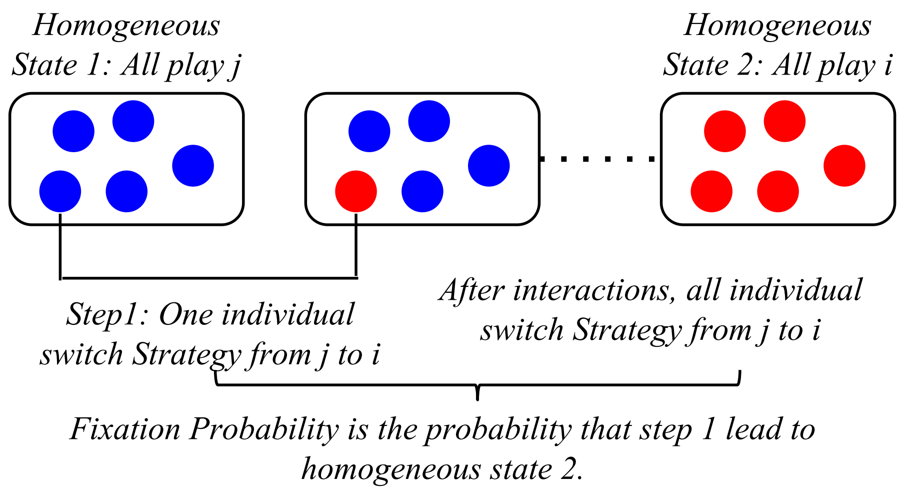

Figure 4.2: Strategy switch process and fixation probability.

population takes strategy j, this would eventually lead to two situations: first, this single i-player may cause the entire population to play i; second, the i-player may return to play j. The probability of the first situation is called a fixation probability: ρ_{ji}, whereas the probability of the second situation is 1-ρ_{ji} (see figure 4.2 for a more intuitive description).

$\rho_{ji} = 1/N$, i has no evolutionary advantage over j, since $1/N$ represents pure neutral selection (Fudenberg and Imhof, 2006). The switches from all-j and all-i, resulting in a Markov Chain. With these assumptions, we can write the fixation probability ρ_{ji} in the following form if $j \neq i$:

$$\rho_{ji} = \frac{1}{1 + \sum_{k=1}^{N-1} \prod_{l=1}^{k} \frac{g(l)}{f(l)}} \tag{4.1}$$

Here, $g(l)$ and $f(l)$ refer to the fitness of playing j or i when there are l individuals playing strategy i. Using π_{ij} to denote payoff in an i-j interaction, they are given by (2) and (3):

$$g(l) = \frac{\pi_{ji} l + \pi_{jj}(N - l - 1)}{N - 1} \tag{4.2}$$

$$f(l) = \frac{\pi_{ii}(l-1) + \pi_{ij}(N-l)}{N-1} \tag{4.3}$$

After calculating all ρ_{ij}, and assuming that the mutation limit is $\mu > 0$, we can form an irreducible Markov process to describe the transition between strategies. The diagonal element of the transition matrix is $1 - \mu \sum_{j \neq i} \rho_{ij}$. We can calculate the long-term stationary distribution for all homogeneous states ("all-cooperate," "all-defect," and "all-C-C").

4.3.2 Analysis Results

4.3.2.1 Evolutionarily Stable State (ESS)

For this numerical example, "all-cooperate" is the only evolutionarily stable state. However, how "all-cooperate" is achieved with *e-cheap talk* is still unclear, and the result does not fit the fixed population setting.

4.3.2.2 Dynamic Long-Term Analysis

In this section, we show the analytical results of the specific numerical example in figure 4.1 by using the method introduced in section 5.1. Supposing in a 100-member team (N=100), the stationary distribution and fixation probabilities (those that are stronger than neutral, i.e., $\rho_{ji} > 1/N$) of the three homogeneous states are described in Figure 4.3[6]. In this example, the result is clear. "all-defect" and "all-C-C" would disappear in the long run, which indicates that almost all individuals eventually learn cooperation and build trust. In

[6] The arrows between the three homogeneous states are neither necessary nor possible for indicating how the actual transition happens, they only indicate the pairwise transitions with probability (calculated using equation 1, 2, and 3) stronger than neutral. For example, it is possible that while some "defectors" are still in the process of becoming "*C-C*" players, some "*C-C*" players have already become "cooperators."

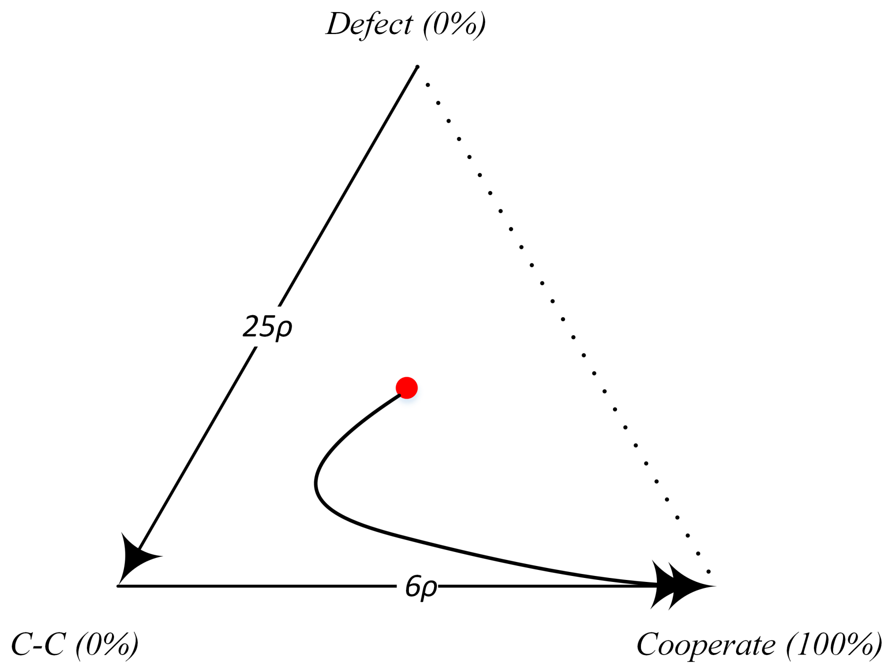

Figure 4.3: The simplex describes the fixation probability between three homogeneous states and their stationary distribution in the long run ($N = 100, \mu = 1, \rho = 1/N$). Only transitions stronger than ρ are shown. A possible route for a non-homogeneous state (represented by the red point in the middle) to reach the stable state is depicted.

this example, the punishment (g in figure 4.1) is large enough such that "*defect*" strategy is strictly dominated by "*C-C*," which leads to the disappearance of "all-defect." Besides, "cooperate" also strictly dominates "*C-C*." However, if the punishment (incentive) was small enough, e.g., $S - e + g < P$, we could expect the "all-defect" might survive in the long run.

The strengths of the transitions differ, which helps us to identify precisely how transition occurs. As figure 4.3 indicates, the transition from "all-defect" to "all-C-C" is rather strong while the transition from "all-C-C" to "all-cooperate" is not so strong. The other transitions are relatively weak, for example, the transition probability from "all-cooperate" to "all-C-C" is only 2.62×10^{-5}. This indicates the important *catalytic effect* of *C-C* in moving the individuals from *defect* to *cooperate*, although it would disappear in the long run.

For hybrid states consisting of populations with differing strategies, there is a similar way to reach a final "all-cooperate" state. At an individual level, a "defector" will first become a "C-C" player and then become a "cooperator." A hybrid state (indicated by the red point in the central of the simplex) may follow the transition patten described in figure 4.3. Some defectors change to "C-C" players, while some "C-C" players change to cooperators in parallel. However, in the long run, all hybrid states cannot survive, therefore the transition is temporary. In our analysis, "C-C" disappears eventually, but it plays a central role in bringing the system from the unfavorable state full of defectors to satisfying "all-cooperate."

In some cases, maintaining"C-C" is also an acceptable and possible alternative. When compared to the cost of setting up or managing *e-cheap talk*, C-C player can avoid being exploited by the defectors (Defect), while maintaining approximately equal good payoff against the pure cooperators (Cooperate) for $\epsilon << R$.

4.3.2.3 Conditions for Long-term Cooperation

Independent of the above numerical example, what conditions generally ensure the transition from *defect* to C-C, and from C-C to *cooperate*? If these conditions can be identified, they could be applied to high-level organization or computer supported collaborative systems design to straightforwardly promote cooperation. To answer this question, we use an important analytical measurement to identify pairwise evolutionary advantageousness in the long run (Nowak, 2006). For the two strategies i and j, if the following condition holds, the transition from j to i is more probable:

$$(N-2)\pi_{ii} + (2N-1)\pi_{ij} > (N+1)\pi_{ji} + (2N-4)\pi_{jj} \tag{4.4}$$

If N is large enough, the above condition can be reduced to:

$$\pi_{ii} + 2\pi_{ij} > \pi_{ji} + 2\pi_{jj} \tag{4.5}$$

The transition from *C-C* to *cooperate* is guaranteed to be more probable if:

$$R + 2R > R - e + 2(R - \frac{e}{2}) \ \Rightarrow -2e < 0 \tag{4.6}$$

which automatically holds when *e¿0*. Obviously, this transition is guaranteed by the payoff structure in this model. The transition from *defect* to *C-C* is guaranteed to be more probable if:

$$R - \frac{e}{2} + 2(S + g - e) > T - g + 2P \ \Rightarrow \tag{4.7}$$

$$g > \frac{T + 2P - R - 2S}{3} + \frac{5e}{6} \tag{4.8}$$

So far, we have presented analytical and numerical results. Both indicate that *e-cheap talk* sufficiently improves the probability of cooperation, so long as the punishment to *D-player* compares to her opponent's cost of using *C-C* and the transition from *C-C* to *cooperate* is almost destined. Moreover, this result is independent of team size for "N" has been eliminated in equation (8).

4.3.3 Analytical Answers to Research Questions

The main analytical findings are summarized in three propositions:

- Proposition 1. *e-cheap talk sufficiently promotes cooperation if punishment to a defector is comparable to his opponent's cost of using C-C.*
- Proposition 2. *Trust is developed implicitly with the explicit improvement of cooperation, and further ensures the cooperation.*

- Proposition 3. *e-cheap talk decreases or even disappears once trust and cooperation are fully developed, and it functions as a catalyst in this process.*

These three propositions provide answers to the research questions in section 1. Proposition 1 and 2 confirm RQ1-1 and RQ1-2. Proposition 3 answers the long-term dynamics of *e-cheap talk* (RQ1-3). Each proposition can be validated empirically; and we can expect the following empirical observations: (1) *e-cheap talk* and cooperation strongly correlate, such that if a pair of individuals use *cheap talk*, they are likely to also cooperate on work-related issues; (2) *e-cheap talk* should occur between individuals prior to work-related talk; and, (3) *e-cheap talk* will gradually decrease with the cooperation's progress.

For a real world GSE team, the theoretical model provides a possible explanation for how trust develops within the team using *e-cheap talk*. At the beginning of GSE cooperation, team members have less confidence about whether or not their remote colleagues will behave cooperatively. Therefore, they may prefer to use *e-cheap talk* to exhibit their willingness to cooperate. The collaboration then establishes whether remote colleagues respond cooperatively (either using *e-cheap talk* or directly cooperate). The individuals who initiate *e-cheap talk* would avoid significant loss even if the others decide not to cooperate, since the defectors' punishment ensures the cost of *e-cheap talk* will be compensated. However, once cooperation with remote colleagues has been established, people may feel they can devote all their effort toward collaborating, without using *e-cheap talk* as a costly "probe." Although others may still defect, they are willing to take this risk, and they begin to trust that others will not defect. In GSE practices, we can expect that team members will use less *e-cheap talk* as cooperation progresses.

4.4 Case Study I: Apache Lucene's IRC Discussions (#Lucene-dev)

To validate the analytical model's results, we performed a case study of Apache LUCENE's IRC discussion. The case study design follows well-established guidelines in empirical software engineering literature such as (Easterbrook et al., 2008; Kitchenham et al., 2008; Runeson and Hst, 2009), etc. The results of this empirical study provided general support to the theoretical propositions, thus supporting the analytical model's validity and its real world GSE setting results.

4.4.1 Case Study Design

LUCENE is an open source information retrieval framework and API. We chose LUCENE and its IRC channel #lucene-dev because: (1) LUCENE is a mature project with a stable core development team, enabling us to study long-term interactions; (2) LUCENE has two IRCs: the general IRC channel #lucene, and the logged channel #lucene-dev, and since #lucene-dev exists solely for developers to discuss and record issues related to development, this allows us to be more focused as well as reduces "noisy" general user message interference; and, (3) compared with email archives, the interactive, near-synchronous IRC chats are more similar to day-to-day offline conversations. According to (Guzzi et al., 2013), pure *cheap talk* email threads only account for 0.6% of all emails (see section 3.3). An email may contain both *cheap talk* and work-related talk, making it difficult to classify. Besides, developers participate in less than 75% of all email threads, while development related threads only account for 35% (Guzzi et al., 2013).

Our study sample comprised all logged messages from 04/15/2010 (the start date of #lucene-dev) to 12/31/2012, which we collected from #lucene-dev's plain text file online archive. We

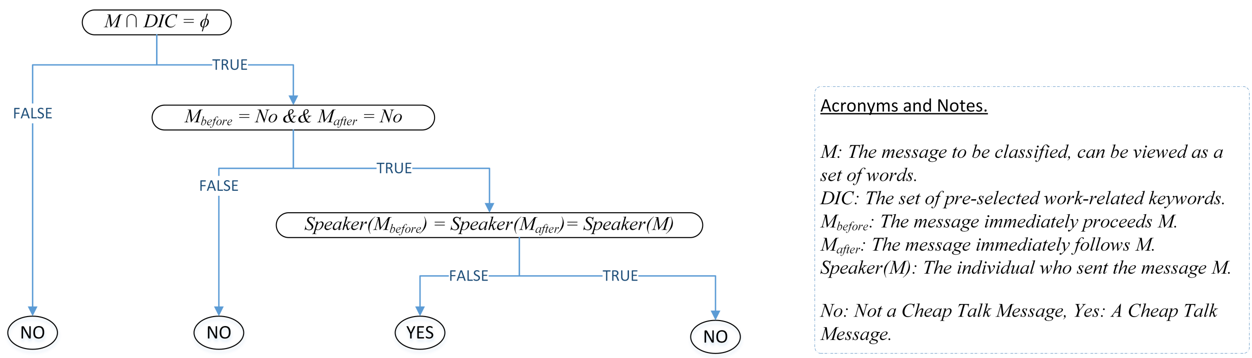

Figure 4.4: The heuristics based decision tree for classifying *cheap talk* messages and corresponding rules.

downloaded and extracted messages with three elements: time, sender, and content, excluding the auto-generated login (out) messages. In total, we have 18216 messages (excluding the simple greeting or confirmation message, e.g., yes/no, "thanks" or "good morning"). Using the three expectations (see section 5.3) derived from theoretical propositions as working hypotheses, our empirical analysis consisted of the following steps: (1) we classified all messages into two categories: *cheap talk* messages and work-related messages. A lightweight qualitative analysis was performed on all *cheap talk* messages in order to gain a basic understanding of *cheap talk* in LUCENE. In later analysis, we open coded and merged them; (2) we identified all work-related dyadic cooperation pairs and *cheap talk* pairs to explore the relationships between *cheap talk* and cooperation (Proposition 1 and 2); and, (3) we applied statistical methods to study the dynamic pattern of *cheap talk* (Proposition 3). In general, the empirical data provides moderate level support to the theoretical predictions, and no contradiction was found.

4.4.2 Data Preparation: Message Classification

We developed a simple, rule-based, decision-tree classifier to automatically classify the messages into two categories: *cheap talk* messages and work-related messages. Three heuristics determined the classification: (1) *Cheap talk* should not contain a set of specific keywords (dictionary[7]) such as: issue, bug, error, commit, Java, etc.; (2) if messages immediately before and after an unclassified message are work-related, there is a near impossibility the message is *cheap talk*; (3) *cheap talk* is usually more interactive than work-related talk, whereas the latter may comprise several messages clarifying a problem. Therefore, if a message is surrounded by messages from the same sender, it is less likely to be *cheap talk*. Simple greeting or confirmation messages (e.g., "yes/ok,") were not classified, but they were retained for the rule, requiring their presence as context. Figure 4.4 describes the decision process and the rule applied to each decision step. The classification resulted in two sets of labelled messages. The number of *cheap talk* messages (n=1296) is far less than the number of work-related ones (n=16920).

Given that the number of *cheap talk* messages is much lower than work-related talk, we simply performed a manual post-classification results check. The decision rules ensured that it is almost impossible for a *cheap talk* message to be classified as "work-related." Specifically, the classification method almost only generates *false positive cheap talks*. We randomly sampled 500 messages classified as "work-related" and found only one instance of *cheap talk* message (*precision* = 99.8%). After the classification finished, we manually excluded all false positives (n = 38). Compared with more sophisticated techniques (e.g., *D-tree* induction), our simple approach avoided the tricks in constructing proper training sets, and works well for this specific classification. The dictionary of pre-selected work-related keywords integrated common and domain knowledge into the classifier and is highly extensible. We compared the results with two Python classification algorithms (Naive Bayes, Decision Tree) from the scikit-learn library in terms of Precision and Recall. The heuristics-based decision tree classifier outperformed both.

[7] This study's dictionary contains 66 keywords, see Appendix A.1 for details.

4.4.3 Empirical Results and Findings

4.4.3.1 e-Cheap Talk in #lucene-dev

e-cheap talk covers various topics, including weather, food, hometowns, politics, or even personal life[8]. Compared with mailing list social interaction messages (Guzzi et al., 2013), the topics are much more diverse. Furthermore, *cheap talk* can be intentional or impromptu. Intentional *cheap talk* is usually used to show one's friendliness and minimize social distance (Cassell and Bickmore, 2003); for example, presenting information that may be unfamiliar and interesting for others (example 1), or introducing one's background (example 2). The latter is more situational and often triggered by an event. For instance, example 3 was triggered when a Russian developer joined the #lucene-dev. In general, intentional *cheap talk* is more common than impromptu.

> Example 1: *state vs federal rights is still a big topic in the US. do you know about the health care legislation that recently passed the US congress? (American Politics)*
>
> Example 2: *hamburg is near but its also a separate region. and hanover but it does not share. hannover is the capitol of lower Saxony, so bremen is an island :-) (Hometown)*
>
> Example 3: *my girlfriend can speak Russian, she is a Estonian. (Personal Life)*

As our model predicts, ignoring *cheap talk* is not a proper behavior. Punishment to uncooperative responses to *cheap talk* does exist and takes various forms. The most common punishment is rejecting a *cheap talk* defector's future cooperation request. In our sample, there are 6 instances of explicit punishments via direct refusals to defector's cooperation request. In one extreme case, the defector left #lucene-dev since no one was interested in

[8] We quote three representative examples (example 1-3) literally except for some formatting.

cooperating or talking with him any longer.

4.4.3.2 e-Cheap Talk and Cooperation

Our study found strong correlation between *cheap talk* and cooperation. We extracted all dyadic cooperation pairs by identifying "send-reply" patterns from work-related messages. For *cheap talk* messages, we identified dyadic pairs using the same method. In total, there are 136 pairs of cooperators engaging in work-related discussions, and 101 pairs of *cheap talks*. Almost 70% work-related pairs (95 in 136) are also in the set of dyadic *cheap talk* pairs. Only 6 *cheap talk* pairs did not have any work-related talk. We further explored the temporal relationship between a *cheap talk* pair and its corresponding work-related pair. 90.5% (86 in 95) *cheap talk* pairs appeared before the corresponding work-related pairs. These results at least partially indicate: (1) *Cheap talk* correlates to cooperations. If two individuals use *cheap talk*, it is probable that they also cooperate. (2) A cooperative relationship may result from *cheap talk*. *Cheap talk* is not caused by cooperation, for most cooperations follow from the appearance of *cheap talk*. These form some support to Propositions 1 & 2. Although we cannot directly measure trust in this study, the literature on virtual collaboration often uses the establishment of cooperation as a good indicator of trust (Bos et al., 2002).

4.4.3.3 The Dynamics of Cheap Talk over Time

The frequency of *cheap talk* drastically changed since the launch of #Lucene-dev. From 04-15-2010 to 04-30-2011, #lucene-dev was very inactive except for the first two weeks, in which time there was little *cheap talk* and few work-related messages. Specifically, there were zero chats in 90 consecutive days (12-04-2010 to 03-03-2011). In May 2011, the LUCENE project decided to better utilize the #lecune-dev to ensure the discussions among code contributors would be logged; subsequently, discussions increased in frequency.

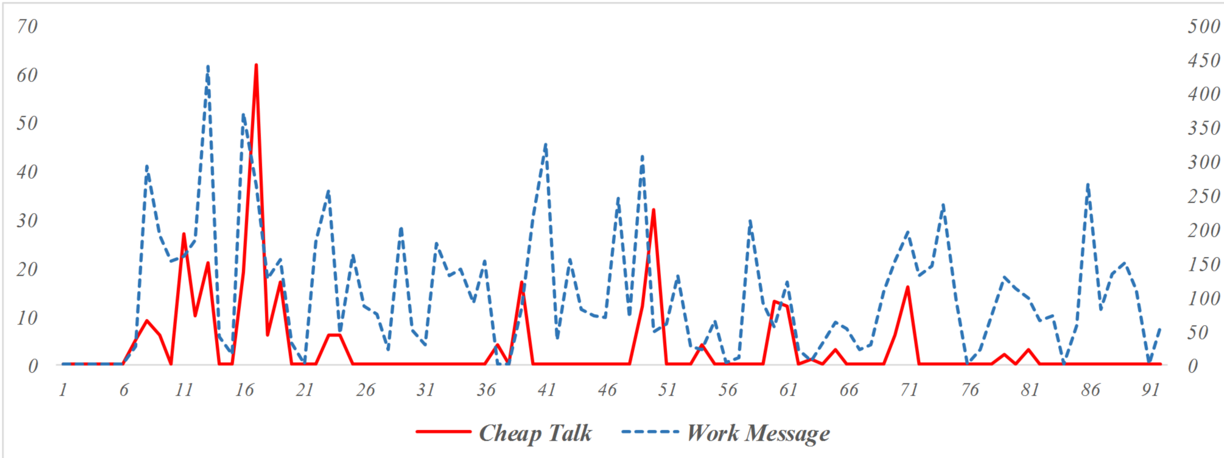

Figure 4.5: The dynamics of *cheap talk* and work-related message over time (left axis: *cheap talk*, right axis: work message).

However, the dynamic patterns of *cheap talk* and work-related message are quite different. The number of *cheap talks* continued to decrease. *Cheap talk* also almost disappeared in about three months after the migration of development activities to #lucene-dev. Figure 4.5 visualizes the differing dynamics of both *cheap talk* and work-related messages from 05-01-2011 to 07-31-2011. Unlike the decreasing trend of *cheap talk*, there are no significant variations for work-related talk except for a weekly cyclical effect. There were several boosts in *cheap talk*, but those were mainly due to several core developers joining the new IRC channel later than usual.

By modeling the dynamics of *cheap talk* and work-related message as two time series (from 05-01-2011 to 07-31-2011), we statistically tested their cointegration[9] to examine their distinctiveness. We followed the *Engle-Granger* two-step method: we first constructed the spread through Ordinary Least Square (OLS) regression and then tested the unit root with the *Augmented Dickey-Fuller* (ADF) test. The test was performed using `R` package ***fUnitRoots***. The ADF *p-value* is 0.162, hence these two dynamics are not cointegrated and they do not exhibit similar change patterns.

[9]Two time series are cointegrated if they share a common stochastic drift, hence are mutually predictable, see (Pfaff, 2008) for more details.

Exactly as the model predicted, the *e-cheap talk* plays an important role in developing cooperation and trust. Once cooperation and trust becomes stable, an individual may not be willing to pay the cost of *e-cheap talk*, directly switching to "cooperate" mode since she can expect (and "*trust*") the others will have similar expectation or behavioral choices. This supports the theoretical proposition 3's prediction of the decreasing trend of *cheap talk* in the long run.

4.5 Case Study II: Chromium OS' IRC Discussions (#chromium-os)

The analyses of LUCENE's IRC discussions provide general support to the theoretical results. To further validate theoretical results, we performed another case study with CHROMIUM OS' IRC discussions (#chromium-os). It is the project behind Chrome OS, which is a cloud-based operating system based on the Linux kernel and designed to work with web applications and installed applications.

Although Chromium OS is an open source project, it differs from LUCENE due to the heavy involvement of a commercial organization (i.e., Google). Many developers work at the same company, hence their participation may be quite different from voluntary behaviors in #lucene-dev (Wagstrom, 2009). For example, some developers may know each other fairly well offline, and they may be not so "free" to leave the community. This allows us to examine our results' generalizability in a slightly different setting. Although Google provides an internal IRC channel, many Google developers still maintain their presence in #chromium-os[10]. One disadvantage is that #chromium-os does not exclusively contain mes-

[10]We retrieved the "tree" (the sum of the various source repositories used to build the project) status history on http://chromiumos-status.appspot.com/. Then ,we manually matched the usernames in tree status (associated with email address) with #Chromium-OS users, finding that the majority of #chromium-os active code contributors have email addresses in the form of "XXX@google.com", which indicates they

sages from developers, even though it is the designated development IRC channel. Due to the popularity of Chromebook with the general public, many end users who are supposed to use #chromium-os-users and #chromium-os-discuss do post something in it. We had to restrict the time frame when constructing the study sample.

4.5.1 Case Study Design and Data Preparation

The case study design is almost the same as the prior one except for several small differences. First, the study sample consists of the first 18 months (11/19/2009 - 04/30/2011) of IRC messages to ensure data quality (Prikladnicki et al., 2014). There were a substantial amount of messages from end users since 05/2011 after the debut of Chromebook to mass consumer electronics market in June/July 2011. Since then, messages from end users account for a nontrivial proportion ($\geq$ 5%) of all messages, while almost all discussions were among developers in the first 18 months. Second, #chromium-os is not an officially logged IRC channel, so we utilized an unofficial log on echelog[11].

The study sample contains 93617 messages. We adapted the "dictionary" used in message classification while keeping the same classification rules. 19 LUCENE specific keywords were dropped, and 71 new CHROMIUM OS specific keywords (most of them are the names of tool and Linux package) were added[12]. Using the same data processing steps, we got 3280 *cheap talk* messages and 90337 *work related* messages. We read through all *cheap talk* messages; 3127 messages remained as "cheap talk" messages after excluding 153 false positives. We ran similar precision tested for work-related messages. In 500 randomly sampled messages classified as "work-related," no cheap talk message was identified ($\approx$ 100% *precision*). The classification results achieve acceptable precision although we cannot rule out the possibility that some cheap talk messages were mislabeled. The number of mislabeled messages should

are Google employees.

[11] http://echelog.com/logs/browse/chromium-os/.

[12] See Appendix A.2 for details.

be kept at a relatively low level.

4.5.2 Empirical Results and Findings

4.5.2.1 e-Cheap Talk in #chromium-os

The percentage of "cheap talk" in #chromium-os is much lower (3.45% vs. 6.96%) than that in #lucene-dev. However, the topic is even more diverse due to the larger number. Besides the topics we mentioned before, we noticed there are a specific set of cheap talk messages that poke fun at competitors and their products. These messages usually arouse interesting and funny conversations. For example, a contributor shared a link to Google's stock price, and another one commented: "*that's because shareholders probably expected a microsoft killer...like journalists*". Since developers work at the same company (even the same department or team), using this strategy in cheap talk may help them quickly regain their offline common ground. Another noticeable fact is that the cheap talks did not occur in a one-to-one conversation pattern, which is prevalent in #lucene-dev. Rather, more than half of the cheap talks (1632 in 3127) took the "meeting" style in #chromium-os. That is, an individual posts a cheap talk message without explicitly specifying who is the receiver in #chromium-OS, and then some other people talk about it when they see it. It more closely replicates of a group of people come together and chat funny things during their lunch rather than have a one-to-one conversation. In fact, the many-to-many strategy is more cost-effective. In #chromium-os, many contributors may already know each other quite well in their offline lives and be confident with their peers, so they are confident using the many-to-many strategy to broadcast their willingness to cooperate. Conversely, an individual may have to painstakingly talk to everyone to ensure their willingness to cooperate will be delivered to unfamiliar collaborators properly in #lucene-dev.

4.5.2.2 e-Cheap Talk and Cooperation

Although the majority of cheap talks took "meeting" form, there are still some correlations between *cheap talk* and cooperation. Using the same method, we extracted all dyadic pairs of "work-related" cooperation and *cheap talk* relationships. In total, we identified 239 pairs of cooperators who explicitly cooperated (one mentioned the other in at least one message), and 125 pairs of individuals who explicitly engage in cheap talk. 46% of work-related pairs (110 in 239) are also in the set of dyadic *cheap talk* pairs. 15 *cheap talk* pairs did not engage in any explicit work-related talk. 42.7% *cheap talk* pairs (47 in 110) appeared before the corresponding work-related pairs. Compared with the results from Case Study I, the effects are not so strong, e.g., less than half of the work-related pairs were developed after the occurrence of *cheap talk* pairs. However, most individuals participated in the meeting style and many-to-many cheap talks happened during the project kick-off. Even when counting these cheap talks, the majority of work-related pairs followed *cheap talk*.

4.5.2.3 Cooperation Reparation with e-Cheap Talk

While individuals simply quit their participations if they were explicitly punished by others in #lucene-dev, developers proactively took efforts to repair trust and cooperation even before the punishment become explicitly in #chromium-os. For instance, one developer (Alpha) insisted he had fixed a bug (actually he did not) and behaved in a very uncooperative manner when another contributor (Beta) attempted to use a joke about "bug" to persuade him to review the code together. Beta then stopped the conversation and left. However, once Alpha realized his misbehaviors, he made a joke about himself and mentioned Beta as the receiver without explicit apology. Beta accepted this, and they started to work together to review Alpha's code. In this example, both Alpha and Beta seem to accept *cheap talk* as a proper and sufficient action to repair the cooperation. Since contributors likely work in the same

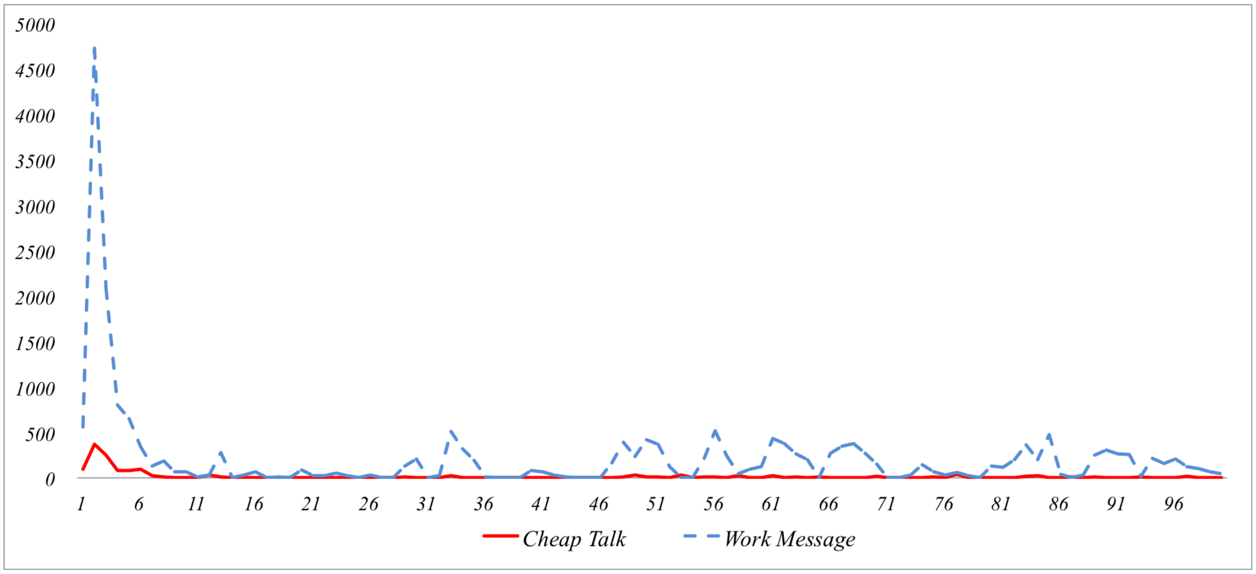

Figure 4.6: The dynamics of *cheap talk* and work-related message over time.

company and the contribution to Chromium OS project may be part of their regular jobs, they are not free to quit their participation in #chromium-os without any negative effect on their career. Therefore, they tend to switch from "defect" to "C-C" once they realize that "defect" is improper. In this scenario, we find that an organization's formal offline social realities (e.g., co-worker) may facilitate cooperation development in virtual collaboration.

4.5.2.4 The Dynamics of Cheap Talk over Time

Figure 4.6 plots the differing change patterns of both cheap talk and work-related message from 11-21-2009[13] to 02-28-2010. At the beginning, both the number of *cheap talk* messages and work-related messages (most of them are about setting up the development environment) are high. However, the number of *cheap talk* messages dropped very fast and almost disappeared after two weeks. In total, the cheap talk in the first month accounts for over 1/3 of all cheap talks in the whole sample (18 months). Although there was less cheap talk, work-related messages also dropped after reaching their peak at the second day, and their number retains a relatively stable level. We also utilized cointegration analysis to statisti-

[13] No message in 11/19 and 11/20.

cally examine their distinctiveness. The ADF *p-value* is 0.379, hence these two dynamics are not cointegrated and do not exhibit similar change patterns.

4.6 Discussions

4.6.1 Implications to Trust and Collaboration Research

Table 4.1 summarized the research questions, theoretical propositions, and corresponding empirical evidences. Combining game theory deduction and empirical study, this study provides an alternative explanation of how, in the presence of punishment, *e-cheap talk* can create trust. In the stag hunt game, cooperation requires at least moderate trust. Without trust, individuals may not expect others to choose the cooperate strategy in their interaction, and thereby defect to avoid risk. With *e-cheap talk* acting as a **catalyst**, trust emerges from and ensures cooperation. The *e-cheap talk* guarantees that any loss in a failed "cooperation trial" can be offset by punishment/incentive enforcement, leading to increased willingness to cooperate. In this process, trust toward others, or more precisely, trust of others' "rationality," eventually develops and "cooperation" becomes the mainstream choice. The improvement of cooperation can be observed, while trust development is implicit (hard to be directly observed in behavioral level); they are essentially two faces of the same process. Even when people may have already known each other very well (Case Study II), they still utilized cheap talk at the early phase of virtual collaboration. The model we developed employs cost-benefit perspectives to study "social interactions" among software developers at the strategic level. Our model offers innovative insights; for instance, it predicts that *e-cheap talk* will gradually disappear with the cooperation's progress, which is rarely mentioned in prior literature. And we furthermore demonstrate *e-cheap talk's* positive impact on GSE collaborations.

Table 4.1: Summary and mapping of research questions, theoretical propositions, and empirical findings

RQs	Theoretical Propositions	**Empirical Findings** Study I: #lucene-dev *(Total: 18216 messages)* *(Cheap talk: 1296 messages)*	Study II: #chromium-os *(Total: 93617 messages)* *(Cheap talk: 3280 messages)*
RQ1-1	Prop. 1 & 2	Section 4.4.3.1, 4.4.3.2	No direct evidences
RQ1-2	Prop. 1 & 2	Section 4.4.3.1, 4.4.3.2	Section 4.5.2.2, 4.5.2.3
RQ1-3	Prop. 3	Section 4.4.3.3	Section 4.5.2.4

The results presented in this chapter also reinforce social chat's role in jump-starting trust and cooperation (Zheng et al., 2002) in the early phase of collaboration, as well as the quick decrease of using this "jump-starter" after the establishment of trust and cooperation. The empirical results show that the jump-starter (cheap talk) is not only used in the early phase of collaboration, but also applied to repair trust (see section 7.2.3). These findings provide more comprehensive and direct support to the "jump-starter" metaphor in real distributed collaboration settings. Also, the theoretical model provides a plausible explanation to the mechanism underlying the "jump-starter" media effect.

4.6.2 Methodological Implications

The human and social aspects of software engineering are increasingly well studied in software engineering(Harper et al., 2013). A large body of empirical literature has focused on many aspects of human interactions in software development, including coordination and trust. Empirical studies contribute rich evidence of real world practice, but are relatively void of explicit theory. While greatly informing research, approaches that can combine empirical studies and predictive theories have obvious advantages. When assumptions are clearly articulated, theory can provide predictive power and avoid costly "trial and error" decisions.

Particularly in studying human and social aspects of software engineering, we argue that game theory provides an ideal analytical framework for developing theoretical knowledge. Software engineering process consists of human decision making activities. Developers have to find proper strategies for team member cooperation. Game theory has demonstrated its capacity for studying human decision making processes and the long-term attributes of social systems since the 1950s, and continues to offer new perspectives to cooperation research (Nowak, 2013). Via game theory model reasoning, we can deductively develop propositions characterizing real world phenomena and provide generative causality and explanation (Cederman, 2005).

Game theory models' generative approach is well-suited to current SE research on empirical practices. All theoretical models and reasoning, no matter how sophisticated, are imperfect abstractions of the real world. Their external validity must be validated through comparison to empirical/experimental results. This process forms a feedback loop between empirical and theoretical work. SE research may benefit from the combination of theoretical and empirical study, which helps to build generalizable and rigorous knowledge while maintaining strong connections to reality. Leveraging EGT, we made some preliminary attempts in this direction. The study was motivated by real world phenomena (section 2), and presented theoretical explanations for it (section 3, 4, & 5). We validated the theoretical propositions through two empirical case studies (section 6, & 7). Our study demonstrates the feasibility of this approach and shows how to use it to develop sense-making theory and generalizable empirical knowledge. However, we admit a limitation of this methodology is that the theoretical model and its propositions can be difficult to validate using direct empirical evidences. Researchers may need to pay extra attention to empirical study design or employ several different empirical methods.

4.6.3 Implication to Collaboration Management

Rapid development of trust and cooperation in globally distributed software teams has been a long lasting problem (Olson and Olson, 2000; Al-Ani and Redmiles, 2009). Cheap talk over the Internet may provide a sound solution. Sharing non-work-related information should be encouraged, especially in the early phase of cooperation and trust development (see our propositions in section 5.3). Prior studies (Schumann et al., 2012) show that even a personal picture may be very helpful in building trust. Furthermore, organizational communication structure affects collaboration, work performance, and the quality of software deliverables (Herbsleb and Grinter, 1999; Kwan, Cataldo and Damian, 2012). Specific communication structures promote or hinder trust and collaboration development among developers (Layman et al., 2006). As we mentioned in section 8.2, the top-down, dictator-style communication is generally disadvantageous to the emergence of cooperation and trust. Therefore, cooperation and trust would benefit from increased participations of general developers who are not in central positions, which may eventually speed up collaborations at the global level (Herbsleb and Mockus, 2003). Furthermore, although some communication network structures are very efficient, they are not stable enough for the imbalance of cost. For example, the individuals in the center (often leaders of project) have to pay most of the cost since all communications rely on her as a "repeater." If the cost is high, the network may break down before cooperation's development. Hence, reducing the cost is crucial. Open and transparent communication will help to reduce the cost and mitigate the imbalance of cost distribution among members (Dabbish et al., 2012), and thus, should be encouraged in management practices (e.g., logging all conversations). The model developed in this chapter provides a framework to enable agent-based simulation for examining mechanism design alternatives (Ren and Kraut, 2014*a*).

4.6.4 Implications to Tools Design & Usage

Both software engineering and CSCW communities agree that interactions among software developers should be encouraged (Wagstrom, Herbsleb and Carley, 2010); and many collaboration tools have been designed for this purpose. However, few intelligent tools can tell software engineers how to interact with their remote collaborators who may be quite different from them. Using game theory as the analytical engine for such intelligent tools may be a good alternative. By simulating and summarizing interaction scenarios with pre-defined dynamics and individual characteristics, it would be able to identify which strategy brings the best payoff to users. We are currently working on a decision support tool to suggest the proper strategies about interacting with an unfamiliar collaborator. It extracts information from an individuals past work traces. Using the game theory modeling and simulation as the core engine, it can help a developers to decide which is a proper strategy (i.e., trust without any reservation or cheap talk first) at the initial stage of collaboration.

In current practices, Web 2.0 and social media tools (Storey et al., 2014) are primary *e-cheap talk* channels. However, (Al-Ani et al., 2012*b*) pointed out that their utilization is not encouraging, for instance, managers often believe these tools cause interruptions to normal work, hence do not want to support the use of them at the organizational policy level. Our analysis shows that *e-cheap talks* over Web 2.0 tools are worth a try. Managers may need to be more open to the using Web 2.0 tools in order to promote *cheap talk*, and a more proactive/aggressive alternative may be integrating *cheap talk* into normal workflow. A developer does not need to intentionally interrupt work to initiate *cheap talk*, for instance an automatically-captured photo of the developer shared on a group page offers a form of cheap information. This approach can serve as an alternative to integrating IDE with social media, e.g., internal "Facebook" plug-ins for Eclipse IDE (Calefato et al., 2011). These tools would also help to reduce the cost of *cheap talk*. In some situations, although *cheap talk's* cost is small, its distribution may be very imbalanced, i.e., a few people always pay the cost (usually several core members). A simple tool that balances individual effort may solve this problem. For example, the tool can identify the members less frequently initiating *e-cheap talk* by analyzing the logs, and then suggest they take *e-cheap talk* initiative.

4.6.5 Threats to Validity

Our research method helps us to achieve high internal and external validity. Game theory modeling requires only minimal assumptions that do not involve too much subjective bias. The empirical studies of Lucene and Chromium-OS rely on open human communication records, and hence suffer less from the validity issues raised in data collection practices. The message classification was automatically executed with relatively little subjective intervention, further preventing significant internal validity threats. The theoretical reasonings also help to solve the generalizability problem of empirical study. However, a potential threat to validity is that the team is fixed in its life cycle, whereas OSS teams are usually very flexible. In its life cycle, new comers may join and some familiar faces may leave. However, this does not mean the model's results are essentially wrong. Lucene's core team is very stable and the timeline for achieving universal cooperation is relative short (at most several months). Chromium OS' project team is even more stable, for many contributors are from Google. Nevertheless, this issue needs to be further investigated in more stable software development teams. It is also possible that the results may be generalizable to other types of virtual collaborations (e.g., collaborative editing or knowledge sharing) if these collaboration could be abstracted into the form of the extended stag hunt game. However, this generalizability also needs empirical validations in the specific settings to fulfill the requirements of the research method we employed in this study.

Regarding construct validity, the major threats comes from the construct "trust". As we know, "trust" is a very complex, vague defined, multi-facet concept (Hosmer, 1995; Corritore, Kracher and Wiedenbeck, 2003). In this study, we assume the emergence of cooperative behavior indicates the development of trust. Although this assumption has widely adopted by researchers in behavioral experiments, it may have some validity concerns such as lack of direct relationship (Riegelsberger, Sasse and McCarthy, 2003). One way to overcome this is performing longitudinal, observational study to capture the dynamics of trust with more direct measurements of trust such as Rotter (1967). Doing so will not only help us to evaluate the validity of this research but also provide a reference for other researchers to select proper measurements of trust. However, according to the state-of-art literature, associating cooperative behavior and trust is still valid.

4.7 Summary

In this chapter, we investigated how informal non-work related conversations over the Internet (*e-cheap talk*) help to promote cooperation and trust in software development teams. We employ a unique cost-benefit perspective using a novel EGT model inspired by a previous field study. We performed a subsequent assessment by data mining the development IRC channels of Lucene and Chromium-OS. Together, our model and empirical studies lead us to conclude that e-cheap talk can help distributed teams build cooperation and trust. However, e-cheap talk works as catalyst in trust and cooperation development process, and tends to gradually disappear in the long run. The study also has implications to GSE practice and collaboration tool design.

The model we proposed does not consider a network effect. All players are equally important with respect to their positions in a fully connected network. However, in the real world, team members are often located in their social network with different positions (Gharehyazie et al., 2014). The network structure influences their communication and interaction (Damian et al., 2007), hence affects the evolution of cooperation and trust development. In next chapter (study II), we discuss our attempt to investigate the trust and cooperation development within the context of a pseudo scale free network. The main research method employed in study II is agent-base modeling and simulation.

Chapter 5

Study II: Simulation in Pseudo Scale-free Network Setting

> they observe their friends, neighbors, or colleagues doing so.
>
> Jon Kleinberg (2007)

5.1 Motivation

Cooperation and trust develop in the context of a social network. However, researchers often do not pay enough explicit attention to social networks' impacts on people's behavioral adaptation in distributed collaboration (Keegan, Gergle and Contractor, 2012*a*). Social network analysis researchers tend to focus more on extracting structural features of empirical networks, such as online communities (e.g., (Yang, Adamic and Ackerman, 2008)) or social media networks (e.g., (Leskovec, Huttenlocher and Kleinberg, 2010)), and adapting them to network models. Although CSCW researchers widely acknowledge digital social networks' power to spread behavioral adaptations, they rarely investigate how specific networks differently affect individuals' behavior (Aral, 2012). Consequently, how informal communication and network features together shape cooperation and trust development remains unclear. Therefore, it is necessary to examine, in a unified framework, the relationships amongst a network's topological features, cooperation and trust development, and informal communication (cheap talk).

Study II aims to develop theoretical insights into network features' role in cooperation and trust development with the presence of cheap talk. More specifically, we focus on a scale-free network[1] (Barabási, Ravasz and Vicsek, 2001), which is perhaps the most widely reported real world network structure (Clauset, Shalizi and Newman, 2009*a*; Sundararajan et al., 2013) and allows the emergence of cooperation (Santos and Pacheco, 2005). To our knowledge, no other study investigates scale-free network cooperation and trust development, particularly accounting for the effects of cheap talk. Leveraging game theory, we developed a decision model to describe agents' strategic choices, and then simulated their strategic decision-making dynamics. In the simulation experiment, we controlled the scale-free network degree exponentials in order to establish the relationship between a scale-free network's degree distribution, and cooperation and trust development with cheap talk. This is the first attempt to link degree distribution with dynamic cooperation and trust development processes. Literature has shown that "seeding" (i.e., initiating behavioral changes) amongst a small fraction of individuals at hub positions may trigger behavioral change cascades (e.g., (Aral, Muchnik and Sundararajan, 2013)). Examining our simulation data would help us identify whether any specific seeding strategy promotes scale-free network cooperation and trust development.

Since cheap talk proves to be a useful way to promote cooperation and trust development (see chapter 4) in general cases, our first research question follows:

[1]In this study, we actually focus on "pseudo scale-free network," the detailed explanation is in 5.4.1.2.

RQ2-1: Does cheap talk positively impact a scale-free network's cooperation and trust development?

If cheap talk's positive role could be confirmed, we would then want to examine two specific aspects of network impacts: degree distribution and seeding strategy. Hence, we express the second research question as the following two sub-questions:

RQ2-2: Given the presence of cheap talk, how do scale-free networks' degree distributions influence cooperation and trust development?

RQ2-3: Given the presence of cheap talk, what are the different seeding strategies' impacts on cooperation and trust development?

Three main results emerge from our modeling and simulations. First, we find that cheap talk has positive impacts on cooperation and trust development. Although there are simulation instances that fail to achieve cooperation, group cooperation was established in the majority of simulation instances (72.62%). Second, scale-free networks of different degree distributions yield differing effectiveness and efficiency when it comes to cooperation and trust development. Third, hub position seeding leads to improved cooperation and trust development effectiveness and efficiency. This effect's significance increases when degree distribution is more uneven (i.e. a few individuals have many friends, while most have one friend).

Summarizing our contributions, our study represents the first attempt to apply agent-based modeling and simulation to link collaboration and network research. Understanding network features' cooperation and trust development role, with the presence of cheap talk, provides both theoretical and practical implications. Theoretically, it helps to develop both micro- and macro-level understandings of distributed collaboration when there are multiple social factors. Practically, it would provide guidelines for designing and engineering an organizational network that facilitates cooperation and trust development with limited resources.

For example, our results indicate "seeding" a cooperative strategy in specific hub positions would help to quicken the establishment of group-wide cooperation norms. Policymakers looking for low-cost efficiency and effectiveness improvement could leverage this knowledge by identifying key individuals for initial cooperation and trust development.

5.2 Cheap Talk and Social Network Analysis

5.2.1 Social Network Analysis of Distributed Collaboration

Social networks are vital precisely because "no man is an island." Social networks support various distributed collaboration activities, such as information sharing, identifying potential collaborators, and team coordination. There are many studies in CSCW applying social network analysis to the study of different distributed collaborations. For example, Farcooq et al. Farooq et al. (2007) suggested that displaying network diagrams of researchers who use similar queries in Citeseer helps identify potential collaborators and research communities. Another example applied structural social network metrics to "breaking-news" Wikipedia article revisions, characterizing article editorial pattern Keegan, Gergle and Contractor (2012*b*). These studies greatly improve our understanding of distributed collaborations. For a more comprehensive review, please refer to Hansen and Smith (2014).

Several studies use controlled experiments to study how cooperation develops given certain specific network structural features Goggins, Mascaro and Mascaro (2012). Similar to our work, Suri et al. Suri and Watts (2011) focused on cooperation and trust development dynamics. They reported a series of web-based experiment in which subjects repetitively played public goods game[2], challenging long-held research beliefs by demonstrating that seeding cooperation did not significantly improve cooperation development. However, their

[2] http://en.wikipedia.org/wiki/Public_goods_game.

study was limited to a relatively small network (24 nodes), as well as their use of public goods game. Public goods game is an N-person prisoner's dilemma that prohibits spontaneous cooperation emergence Barclay (2004). We contend that it is necessary to re-examine the results with different networks and game structures.

5.2.2 Linking Cheap Talk with Social Network Analysis

In chapters 3 and 4, we show that rich literature exists on informal communications among GSE practitioners. However, combining it with SNA has not received enough attention. For example, we built a reasoning on the role of cheap talk in chapter 4, but implicitly assumed all individuals are fully connected, which is almost impossible in the real world. There might be two reasons for this lack of attention to the combined research streams. First of all, it is difficult to observe and infer behavioral changes from empirical social network data, such as identifying whether a person becomes "more cooperative" or "less cooperative" from the traces of his or her activities. In the few research studies on this topic, the analysis uses a static snapshot of individuals' behaviors rather than allowing people to dynamically adapt their behaviors (e.g., Yang et al. (2011)). Secondly, researchers in the two streams prefer different research methods. Small sample experiments, observational study, interviews, and surveys dominate the first stream (e.g., Gao, Hinds and Zhao (2013)), whereas the second stream often relies on large-scale network empirical data. In spite of these difficulties, it would be valuable to link these two streams to further develop understanding of cheap talk's role in social network cooperation and trust development. We determined agent-based modeling and simulation would adequately suit this research combination and we will introduce how we built the agent-based model and simulation in the following sections.

5.3 Research Approach

5.3.1 Agent Based Modeling and Simulation

In this study, we adopted agent-based modeling and simulation (Macal and North, 2010) to explore cooperation and trust development in a special class of networks (scale-free networks). Leveraging evolutionary game theory, the model expresses strategic behavioral learning and adaptation at both individual and team levels. By simulating an individual's behavior, we capture from the ground-up a complex dynamic systems' behaviors and properties Ren and Kraut (2014*b*). Agent-based modeling and simulation has been increasingly applied in HCI and CSCW to develop theoretical knowledge and practical design implications Olson and Kellogg (2014); Ren and Kraut (2014*c*).

Computer-modeled agent-based simulation is a particularly useful tool for developing new theoretical insights and linking well-defined yet separated research streams Etzion (2014). As mentioned in the literature review above, there are well-established theories about cooperation and trust in distributed collaboration, as well as social network analysis on various collaboration systems. Traditionally, though, these theories tend to only consider a single perspective. For instance, social interaction theories suggest informal communication may help to reduce social distance and ease cooperation Cramton and Hinds (2007). Studies such as Suri and Watts (2011) demonstrate that strategy decisions are greatly influenced by social network members who are closely related to the decision maker, yet these studies do not investigate the decisions' dynamics. Combining them in an agent-based model is feasible and may bring about better understanding. Moreover, the dynamic, adaptive processes at the core of agent-based models make them particularly suitable for analyzing longitudinal, chronically reproduced processes such as behavioral changes and social norm development Etzion (2014), such as cooperation and trust development.

From an epistemological perspective, modeling and simulation is a generative knowledge development approach that suitably answers "what-if" questions. Agent-based simulation is similarly appropriate for investigating social phenomena that are influenced by complex and inter-related factors. These factors are difficult, if not impossible, to directly observe and assess in real-world settings, and existing theories about them are often too abstract for use in mechanism design. Thus, agent-based modeling and simulation integrate and concretize abstract theory in a virtual experiment environment, which enables researchers to identify places where theories agree, disagree, or are independent of each other, and to pin down factors that organization policymakers could manipulate to produce desirable outcomes Ren and Kraut (2014*c*).

Moreover, using agent-based modeling and simulation as the research method does not mean we exclude or ignore empiricism. In (Sugden, 2000), the author argued that abstract modeling and simulation are not abstractions from, or simplifications of, the real world. They represent the counterfactual worlds constructed by the researcher. Of course, there are some gaps between model world and real world, but these gaps can be filled only by inductive inference. Models describe how the world could, and can, be linked to the real world through implicit inductive reasoning. If we can see the relevant models as instances of some category, some of whose instances actually exist in the real world, we could achieve some degree of empiricism.

5.3.2 Applying the Research Method

To apply agent-based modeling and simulation as a research method, we followed the 7-step guidelines in Ren and Kraut (2014*c*). The key step in this process is building a conceptual model. The main difficulty in developing conceptual model is specifying how agents make behavioral decisions under the influence of their social network position. Based on the game theory model described in Wang and Redmiles (2013), we developed a decision model in which an agent's direct neighbors influence his or her behavioral decisions (Kleinberg, 2007). This model allows individual agents to form and adapt their strategies according to their social network positions, and also considers the restriction of individuals' "bounded rationality" Kahneman (2003). We translated the conceptual model to computational representations, and implemented it. In this step, since most current scale-free network generators are based on stochastic models that cannot provide precise degree distribution control, we generated our own networks. In the next two sections, we will further introduce how we built the model's conceptual framework, implementation, and virtual experiment design.

5.4 Virtual Experiment Design

5.4.1 Preliminaries: Network and Game

5.4.1.1 Scale-free Network

Consider a finite set of agents $N = \{1, 2, ..., i, ..., n\}$. The relationship between each individual determines a social network. Let $\Gamma \equiv (N, L)$ denotes the social network where $L \subseteq N \times N$ is the set of dyadic interactions among the individuals. Γ is undirected, i.e., $(i, j) \in L$ indicates $(j, i) \in L$. Besides, links to oneself are not allowed, i.e., $(i, i) \notin L$. We further define N_i as the set of individuals who have direct links with i. Formally, $N_i = \{j \in N, such\ that(i, j) \in L\}$, while $k_i \equiv |N_i|$ is the number of neighbors of i, often referred as his or her degree.

A network is characterized by its degree distribution (López-Pintado, 2006). By convention, we use $P(k)$ to denote the fraction of the nodes with degree $k \geq 1$. Obviously, $k = 0$ indicates a node has no link with others. We assume no isolated individuals exist, hence

$P(0) = 0$. There are different types of networks such as *homogeneous* network (all nodes have same degree), *exponential* network (degree follows exponential distribution), and *scale-free* network. In this article, we are particularly interested in *scale-free* network, which exhibits a power-law degree distribution. Formally,

$$P(k) \propto k^{-\lambda}, \lambda \in [2, 3] \tag{5.1}$$

For a *scale-free* network, the degree variance can be very large, making it a powerful descriptor of real-world social relationships. Empirical evidence has shown that many paradigmatic examples of social networks are characterized by scale-free connectivity distributions Sundararajan et al. (2013). In particular, geographically-distributed teams often use some key members and clusters of co-located team members as cross-site communication hubs, which are two major features of a scale-free network. In general, scale-free networks provide a relatively good abstraction for distributed collaboration.

The λ in (5.1) is the degree exponential. λ is neither directly related to the average degree nor to the number of total edges. However, it does reflect the degree distribution of a scale-free network. In general, the smaller it is, the more even its degree distribution. Intuitively, if λ is smaller, there would be more nodes with more than one link (i.e., more people have more than one "friends" in network). Figure 5.1 provides an example of a scale-free network. In figure 5.1, clusters are often connected by hub nodes which play an important role in maintaining connectivity and facilitating social learning.

5.4.1.2 Pseudo Scale-free Network

Scale-free networks usually require a relatively large number of nodes ($\geq 10^6$) to avoid the random noise of the small network. However, no real world software engineering team would

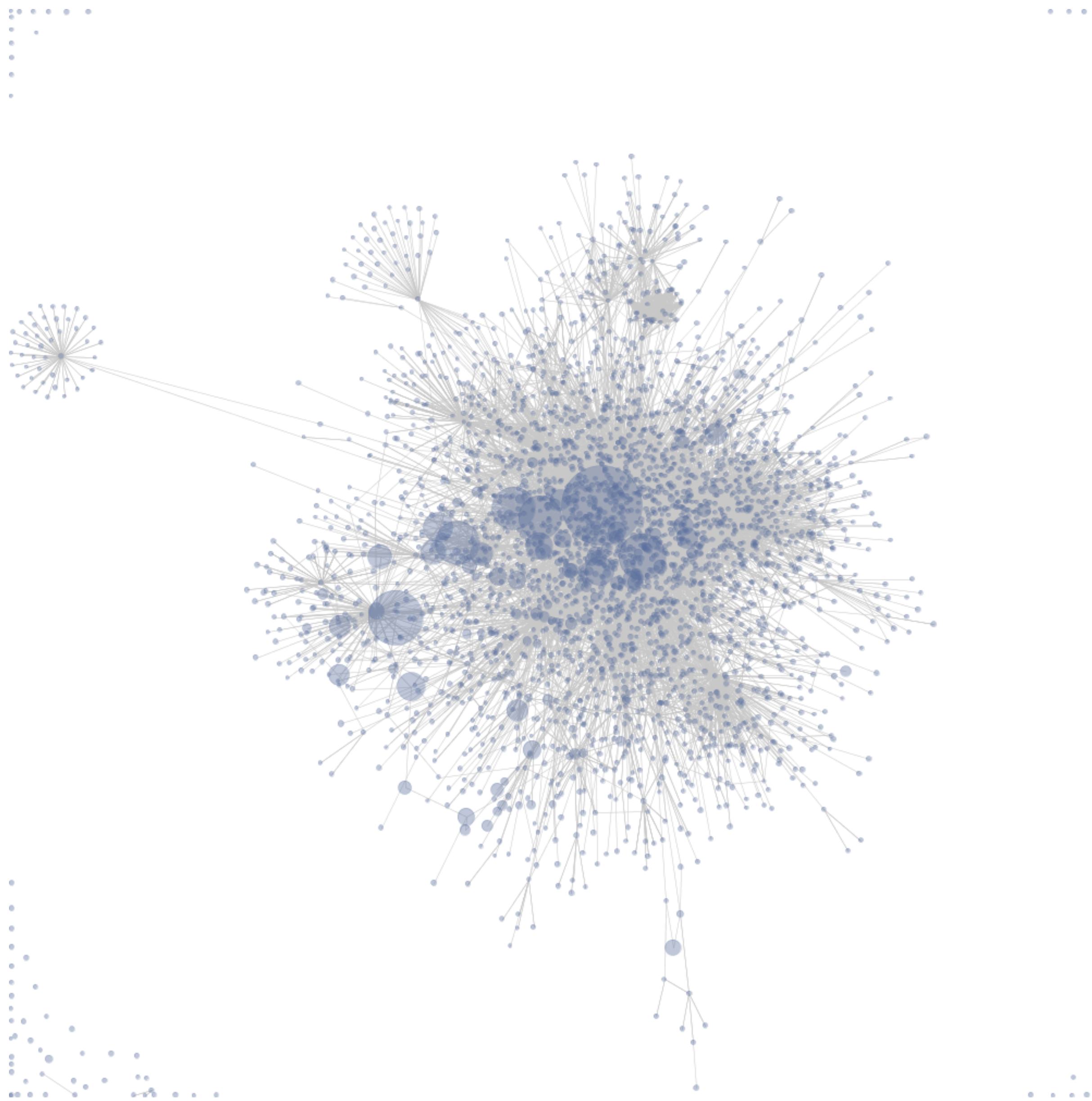

Figure 5.1: An example of scale free network: the link structure of a wiki. License: CC BY-SA3.0, Credits to Chris Davis.

be that large. Most software project teams have, on average, no more than 150 project members. In a study using ISBSG industry dataset[3], the authors found that the largest team in the ISBSG dataset has 468 members, and most projects have less than 100 members (Pendharkar and Rodger, 2009). In this study, we used a pseudo scale-free network. The

[3]ISBSG: `http://www.isbsg.org/`.. ISBSG (release 7) is a large industrial software development dataset containing project information of 1238 projects.

term "pseudo" indicates the network size is small, while its degree distribution still fits power law. To generate a pseudo scale-free network, we first generated a sequence of degrees that follows power distribution, and used the configuration model (Chung and Lu, 2004; Molloy and Reed, 1995) to generate a network which has an exactly prescribed degree distribution with `Python` NetworkX package.

5.4.1.3 Game Structure

We adopted the payoff structure defined in chapter 4 (see figure 5.2), in which we changed the classical stag hunt game to describe cheap talk strategies in globally distributed collaboration. There are three strategies an agent may use. To set up an *cheap talk* over the Internet, the proposer must pay a small management cost (e.g., spend extra time to talk about something irrelevant to work). The game structure assumes that the cost is small compared to the benefits from the cooperation ($e << R$). It may not be constant, but it is likely that the fluctuations do not span different orders of magnitude. Furthermore, the game assumes that players' preferences remain consistent (i.e., there is no execution noise). If a player decided to start *cheap talk*, she would use the "*cooperate*" strategy in the following interaction. Therefore, we add a new strategy called "*C-C*" and make the new game contain three strategies {*C-C*, *C*, *D*}. For a more detailed explanation, please refer to the relevant content in chapter 4.

$$
\begin{array}{c} \\ C\text{-}C \\ C \\ D \end{array}
\begin{array}{c}
\begin{array}{ccc} C\text{-}C & C & D \end{array} \\
\begin{pmatrix} R\text{-}e/2 & R\text{-}e & S\text{-}e\text{+}g \\ R & R & S \\ T\text{-}g & T & P \end{pmatrix}
\end{array}
\qquad
\begin{array}{c} \\ C\text{-}C \\ C \\ D \end{array}
\begin{array}{c}
\begin{array}{ccc} C\text{-}C & C & D \end{array} \\
\begin{pmatrix} 1.9 & 1.8 & 1.3 \\ 2 & 2 & 0.5 \\ 0.5 & 1.5 & 1 \end{pmatrix}
\end{array}
$$

e=0.2, g=1

Figure 5.2: The payoff structure and a numeric example of different strategies.

5.4.2 Individual-level Strategy Change

During each period, one agent will be selected to interact with her neighbors in the network. Before the interaction, she needs to review her strategy and decide whether or not to switch to a new strategy. For example, she might be a defector, but she wants to review whether she needs to be more cooperative to maximize her interaction benefits. She checks all of her neighbors' strategies, and myopically selects a strategy to maximize her expected benefit in the next interaction. The adaptation of her strategy is probabilistic (log-linear discrete choice model Train (2009)), and related to the expected payoff of playing strategy s (i.e. $U_i(s)$ at time t according her neighbors' strategies at time $t-1$). Formally, the probability of her strategy choice s in strategy space S will satisfy the following conditions:

$$\begin{aligned} &\frac{e^{\beta U_i(s)}}{\sum^{S} e^{\beta U_i(s)}} \\ &\beta \geq 0 \end{aligned} \tag{5.2}$$

In 5.2, β refers to the sensitivity towards payoff change: the larger β is, the more likely it is that she chooses a best reply given the actions of his neighbors. Figure 5.3 presents an example of how people in a network apply this decision rule when $\beta \rightarrow +\infty$ (deterministic best-response). In this scenario, A is connected with three individuals: two (red dots) are cooperators and the other (blue dot) uses C-C. Her last strategy is C-C. When reviewing her strategy, she simply uses equation (2) to determine which strategy is the best reply to her neighbors' strategies. According to the numeric payoff structure in figure 1, if she plays C, the expected payoff is: $1\times\frac{2}{3}+1\times\frac{1}{3}=1$, which is more beneficial than playing C-C $(0.9\times\frac{2}{3}+0.95\times\frac{1}{3}=0.916)$. Consequently, she would change her strategy from C-C to C after reviewing her strategy. The game structure has her exclusively consider her direct neighbor's strategy during the review process because prior empirical and experimental studies such as Suri

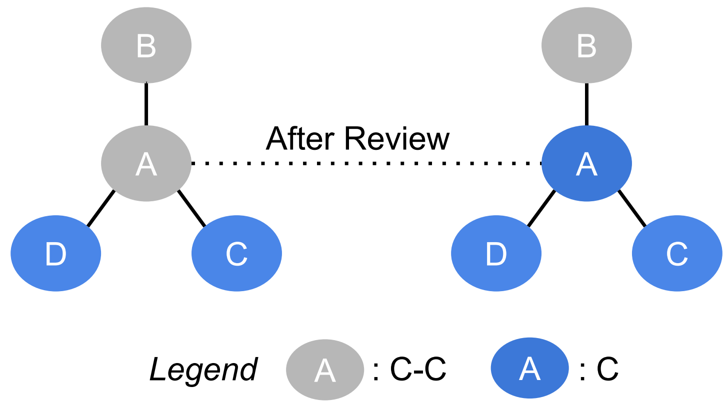

Figure 5.3: An example of applying the decision model to make decision ($\beta \to +\infty$). In this example, A changes her strategy after the reviewing process.

and Watts (2011) have reported that individuals' decisions are overwhelmingly influenced by their direct neighbors.

To introduce stochasticity to the dynamic system and reflect agents' "bounded rationality" in decision-making, we allowed agents to make mistakes in strategy formation. In our simulation, we include a small mistake probability in which agents randomly select, instead of carefully calculate, a strategy. This increases the model's real-world applicability, given that people may not form the best reply due to practical restrictions, such as the lack of information about their neighbors, or limited information-processing ability.

5.4.3 Group-level Strategy Dynamics

Individual-level strategy changes may eventually lead to one specific strategy becoming a group-wide norm. As an example, Figure 5.4 demonstrates one possibility for how individual-level strategy can lead to the adoption of group-wide cooperation. In the example, there are four agents connected with three links. At the beginning, all individuals defect (we use "all-defect" to denote this state). Next, individual A begins to review her strategy and finds she will receive improved results if she changes to C-C. In the next round, B is selected for

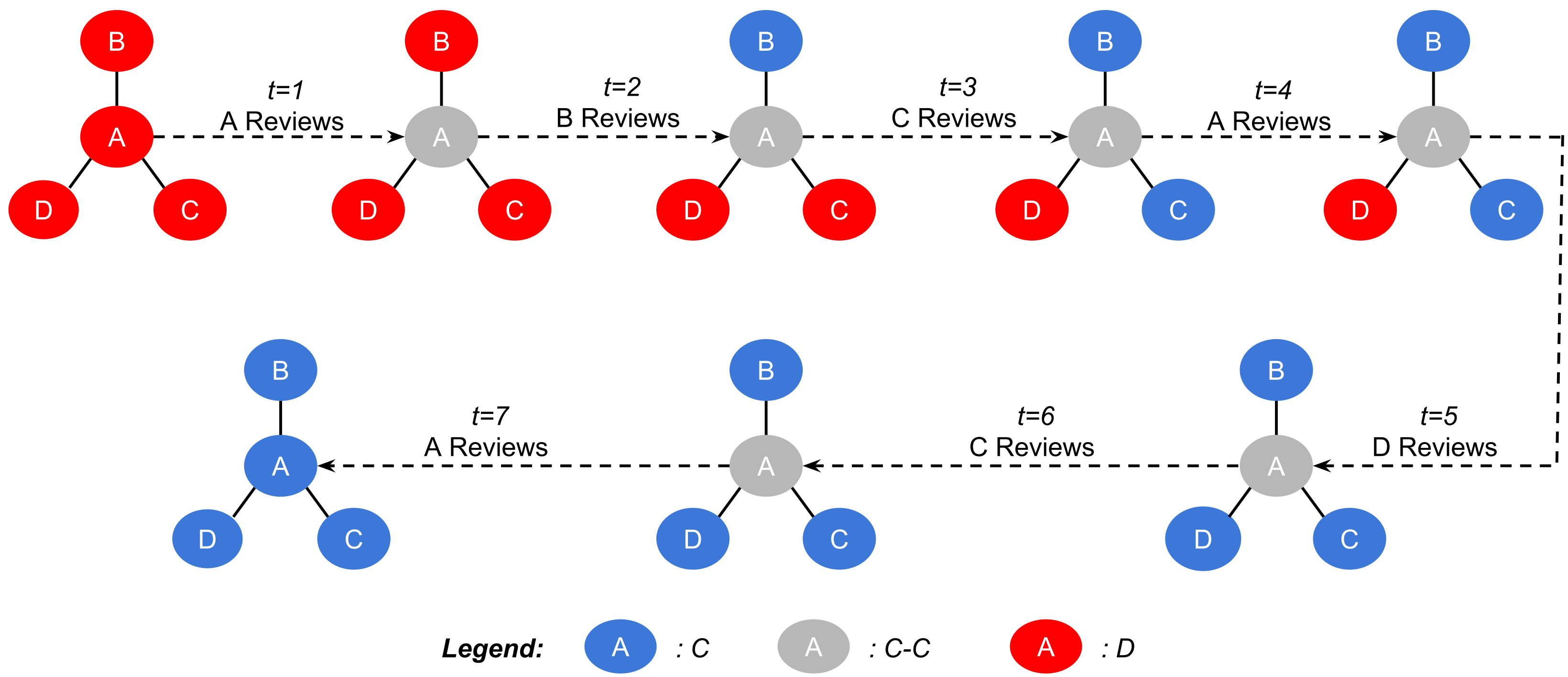

Figure 5.4: A simple example of individual-level strategy changes that lead to the whole group achieving a homogenous "all-cooperative" state (all agents using same strategy) when $\beta \to +\infty$. The real simulated process may be much more complex than this example. In addition, it is also possible that the group fails to achieve the "all-cooperate" state.

review. Since agent B only connects to agent A, her best reply is C when agent A uses C-C. The same situation applies to agent C and D. Continuing this process, the system becomes stable when all agents use C (we use "all-cooperate" to refer this state). In real-world cases, this process may be much more complex; indeed, the whole process may take thousands of behavioral change instances. However, this example illustrates the main idea behind our simulation. If a simulation does reach "all-cooperate" state, we assume cooperation and trust has been successfully developed among group members. In determining who will be the next individual to review his or her strategy, we incorporate a basic social learning (DiMaggio and Garip, 2012) mechanism. That is, if A has changed her strategy in time t, one of her neighbors will be selected to review in time $t + 1$; otherwise, the selection of players in time $t+1$ is random. This reflects social reality well, since if your neighbor changes their behavior, you might want to observe it and decide whether or not you need to adapt to those changes.

5.5 Model Implementation and Virtual Experiment Design

5.5.1 Model Implementation

Our integration of social network and behavioral decision-making is unique, and therefore it is difficult to customize existing simulation platforms such as Vensim DSS or NetLogo in order to implement our model. Consequently, we used `Python` and `NetworkX`[4] to build the simulation and NetworkX to generate the scale-free networks. However, we did not use its preference-attachment-based graph generator API[5] because it cannot control the degree exponential λ. Instead, we first created a degree sequence with special degree exponential, and

[4]NetworkX: `https://networkx.github.io/`

[5]i.e. barabasi_albert_graph(n, m[, seed]), please refer to the document at `http://goo.gl/i6lbvk` for more information.

Parameter	Definition	Value	Related to Theory Development
N	Population Size	100	N
Payoff	Benefits of different strategies	See figure 2.	N
λ	Degree Exponential	2.0-3.0, with 0.2 interval	Y (main experiment conditions)
M	Probability of mistake in decision making	0.1	N

Table 5.1: Summary of model conditions and parameters.

then from this sequence we stochastically generated network instances, ensuring the degree exponential approximately equals the desired value Clauset, Shalizi and Newman (2009*b*). The decision model essentially comprises a discrete-time model of difference equations, which can be implemented and run with any language, even on a spreadsheet. We chose `Python` for convenience because `NwtworkX` is a `Python` package and because it is also easily extended to incorporate additional factors in future research.

5.5.2 Virtual Experiment Design

To answer the research questions proposed in section 1, we designed a virtual simulation experiment. Table 1 summarizes the experiment's conditions and parameters. RQ2-1 does not require direct manipulation to the experiment conditions. Since we are primarily interested in evaluating degree distribution's impact (RQ2-2), the main condition is *degree exponential*, which ranges from 2.0 to 3.0. We set the interval as 0.2, thereby generating 6 conditions: $\lambda = 2.0, \lambda = 2.2, \lambda = 2.4, \lambda = 2.6, \lambda = 2.8, \lambda = 3.0$. This interval is not arbitrarily determined. Since the degree sequence generator in NetworkX cannot provide a sequence that fits special degree exponential precisely (in fact, it is not possible for any random sequence generator), there are usually ± 0.1 approximate errors. Consequently, we set the interval as 0.2 to avoid these errors as well as to ensure the generated sequence's exponential of a smaller condition is actually smaller than that of a larger condition.

For each condition, 10 scale-free network instances were generated. For each instance, the model was independently run 100 times, each with 10% node as unique seeding point to start the "trust" strategy. Therefore, we have $6 \times 10 \times 100 = 6000$ simulation trials in total. At the beginning of each simulation, every individuals' default strategy is "D". As we mentioned before, agents' rationality is bounded, and they naturally will make mistakes. We set the mistake probability as 0.1, meaning that 1 out of 10 decisions will be a random mistake. For this parameter, sensitivity analyses suggest the main results remain robust when this probability varies between 0 to 0.20%. The same payoff structures were used as the numeric examples in figure 5.2, enabling us to compare the results in chapter 4. We also performed sensitivity analyses to the payoff parameters and found general support for robustness. The payoff structure also satisfies the conditions specified in Wang and Redmiles (2013), making the results comparable. The team size was set at 100: a typical number for mid-scale distributed collaboration according to prior empirical study Al-Ani et al. (2013); Pendharkar and Rodger (2009), and a large enough sample to provide rich dynamics.

5.6 Results and Findings

In this section, we report the results of running virtual simulation experiments. We used `R3.0.2` for Mac to perform all statistical analyses reported in this section. All diagrams and charts are generated with `ggplot2`. The analyses yield five propositions, which provide answers to the three main research questions. Table 1 summarized the correspondence between research questions and propositions. The remainder of this section provides more detail.

RQs	Proposition	Key Points
RQ2-1	Proposition I	Cheap talk is positive.
RQ2-2	Proposition II & III	More even degree distribution is usually better.
RQ2-3	Proposition IV & V	Seeding at hubs is better.

Table 5.2: Summary of findings and corresponding research questions.

5.6.1 RQ2-1: Cheap Talk's Positive Impact

First, we wanted to know whether or not cheap talk positively impacts cooperation and trust development. According to Wang and Redmiles (2013), without cheap talk the "all-defect" state (all individuals are defectors) remains stable and almost impossible to transfer to the "all-cooperate" state. Figure 5.5 depicts the percentage of simulation runs that successfully reached the "all-cooperate" state in each degree exponential condition.

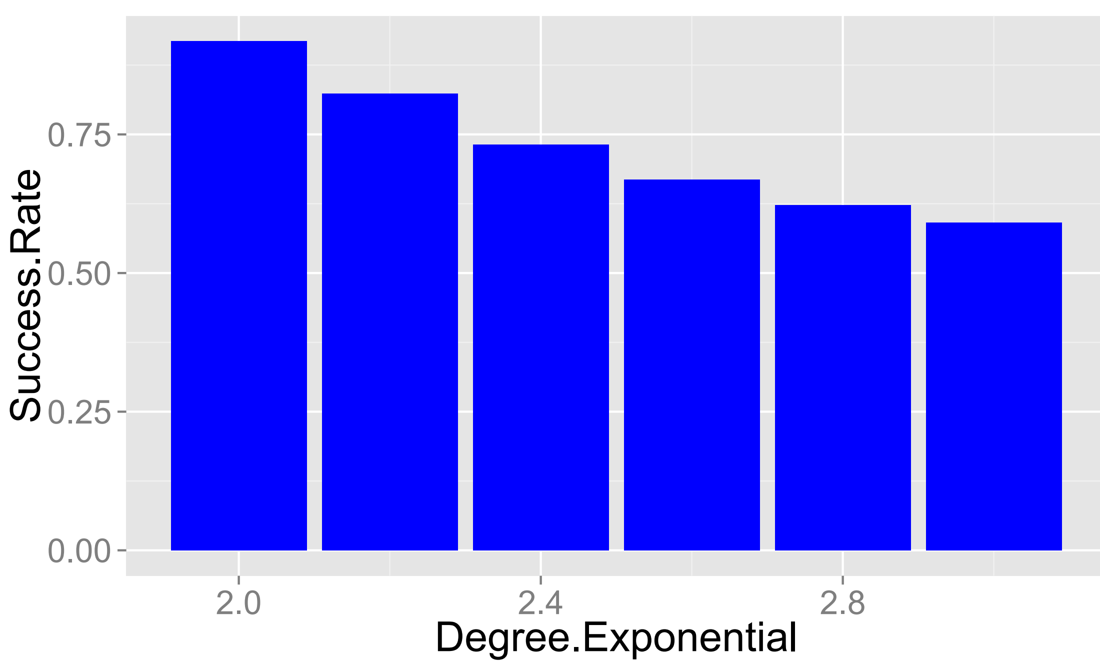

Figure 5.5: The relationship between degree exponential (x-axis) and success rate (y-axis).

For all 6 conditions, the majority of independent simulation runs successfully reached the stable "cooperation" state. In all 6000 runs, 4357 runs (72.62%) successfully developed cooperation. In all 6 conditions, even the least successful one achieved over 60% success rate. The results at least partially support the claim that cheap talk promotes cooperation and trust when group members form a pseudo scale-free network. In all the successful simulation runs, the "all-cooperate" state actually indicates the disappearance of "cheap talk," which confirms the role of "cheap talk" as a "catalyst." Hence, we have following proposition:

PROPOSITION I: *If individuals' relationships form a pseudo scale-free network, cheap talk with proper punishment to defectors remains an effective way to promote cooperation and trust development. In this process, it still largely functions as catalyst.*

5.6.2 RQ2-2: Impacts of Degree Distribution

The impacts of degree distribution are evaluated from two perspectives: effectiveness and efficiency. *Effectiveness* is measured by the success rate of reaching the "all-cooperate" state in each condition. *Efficiency* refers to how quickly the "all-cooperate" state is achieved, and therefore it can be measured by the number of total strategy reviews that occurred before reaching the final state.

5.6.2.1 Effectiveness

Figure 5.6 provides intuitive effectiveness illustration. An obvious pattern emerges whereby the success rate decreases with the degree exponential increase. As mentioned in section 3, the larger the degree exponential, the more individuals who have only one connection. Therefore, cooperation and trust development more easily succeeds when "average" people have more friends. This observation leads to:

PROPOSITION II: *Cheap talk's positive impact o is more significant when the degrees are more evenly distributed. Intuitively, if more people have more than one "friend" in the network, cooperation and trust are more likely to develop.*

5.6.2.2 Efficiency

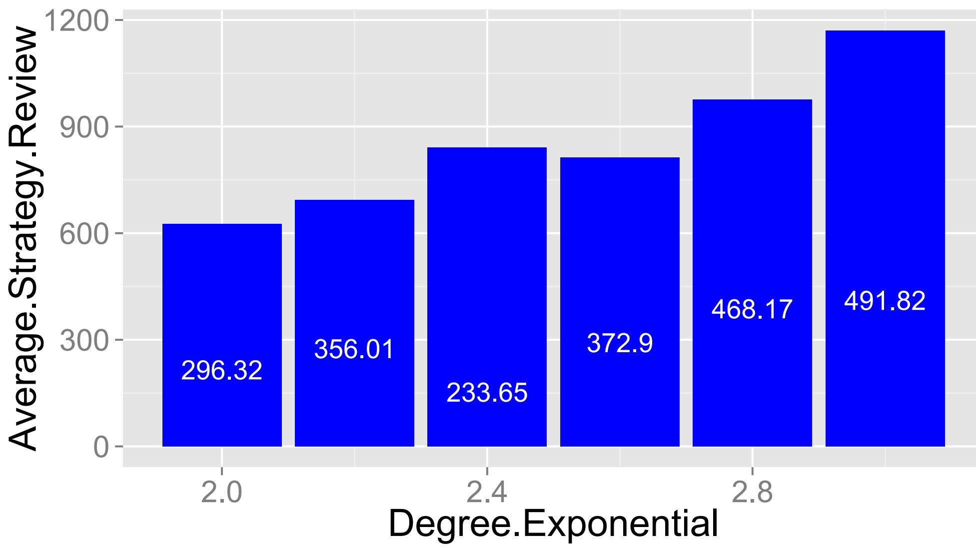

Figure 5.6: The relationship between degree exponential (x-axis) and average number of strategy reviews (y-axis) before reaching "all-cooperate" state.

We considered whether or not different degree distributions yield different efficiencies in achieving a homogenous "cooperation" state. Figure 7 shows the average number of strategy reviews that occur before reaching stable cooperation state. We only consider the 4357 cases that successfully reached group-level cooperation. The number displayed on each bar is the standard deviation. Obviously, with the increase of degree exponential, it usually takes more steps to achieve the final state. A one-way ANOVA test was performed to identify whether significant differences exist across different conditions, yielding significant results (*p-value* < 0.001, $F(5, 4351) = 963.9$). We then performed *pairwise t-test* to determine the differences between any two groups. The results show all differences significant at 0.01 level except for two groups: "Degree Exponential = 2.4" and "Degree Exponential = 2.6." However, the *p-value* is 0.015, which is still significant at the 0.05 level. Hence, our third proposition follows:

PROPOSITION III: *If the degrees are more evenly distributed, cooperation and trust's development will be more efficient. Intuitively, if more people have more than one "friend" in the network, it will take less time to develop cooperation and trust.*

5.6.3 RQ2-3: Impacts of Seeding Strategies

A network position is considered a "hub" when it satisfies both of the following two heuristic conditions: (1) its degree centrality[6] is in top 10% of all nodes; (2) it has at least three direct links: if a node has only two links, it is more likely a connector than a hub.

Whether or not seeding from hub positions promotes cooperation and trust development is a hot topic in networked collaboration researchSuri and Watts (2011), and there are some contradictory results (see section 2.2). We used our simulation-generated data to examine the specific settings (pseudo scale-free network, stag hunt game with cheap talk, and best-reply decision model). We first identified and separated the simulations that begin from hubs. Then, we compared their results with the whole sample to test the existence of significant differences. In the remainder of this subsection, we present the comparisons of effectiveness and efficiency.

5.6.3.1 Effectiveness

Figure 5.7 shows the seeding strategies' influence on the effective development of cooperation and trust. When the degree exponential is small, the differences are not very apparent. However, the difference becomes significant when the degree exponential approaches 3.0.

[6]Here, to determine the hubs we used simple degree centrality rather than betweenness centrality because the betweenness centrality distributions are alike for nearly all scale-free network Goh et al. (2002).

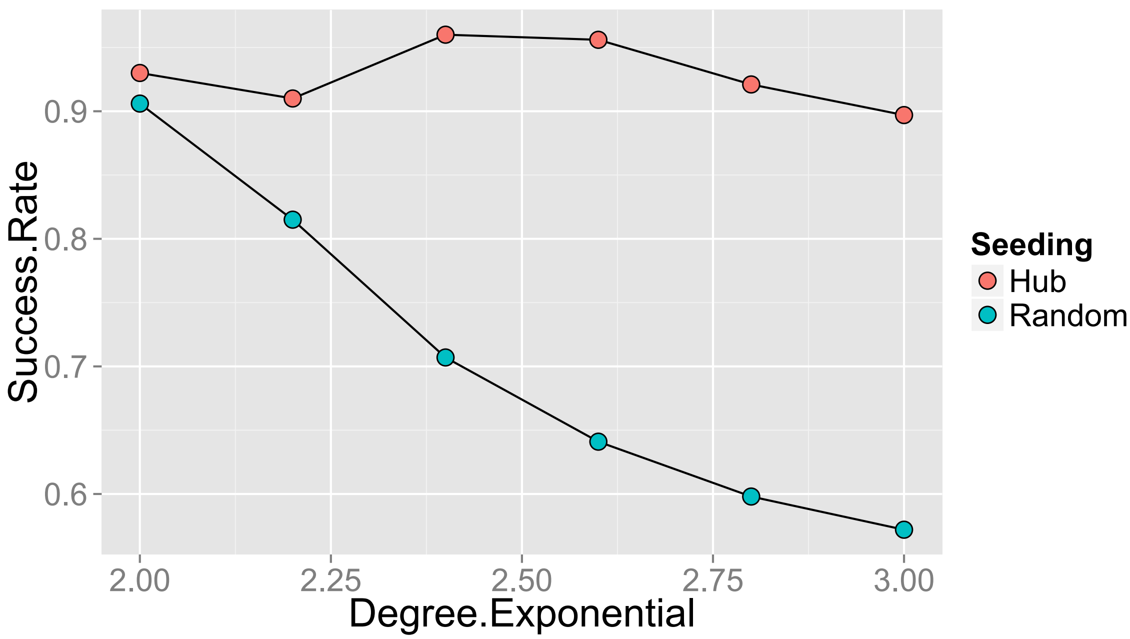

Figure 5.7: The differences between two seeding strategies (Hub seeding vs. Random seeding) on success rate.

For example, the success rate of hub seeding remains around 90% at a 3.0 level, whereas random seeding's success rate barely exceeds 50%. A possible explanation for why the seeding position becomes sensitive with a larger degree exponential is that the group relies more on several key hubs to establish the "cooperation" and transmit "cooperation" to the rest of the group. Since these nodes only account for a small proportion of the whole group, it is possible that some barriers restrict them from becoming cooperators. Hence, the whole group may not successfully develop cooperation and trust. Therefore, we can claim:

PROPOSITION IV: *Seeding cooperation and trust in the hubs during the early stages will increase the possibility for whole group to build cooperation and trust, especially when there are many individuals with only one connection.*

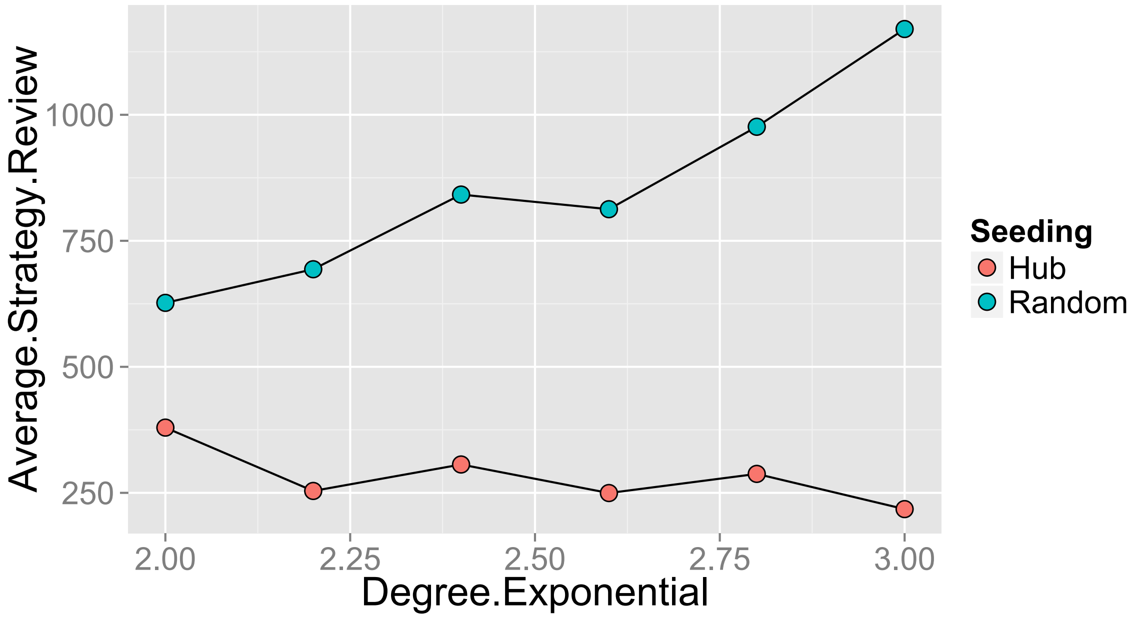

Figure 5.8: The difference in influence on the average number of strategy reviews occurring before reaching an "all-cooperation" state between two seeding strategies (Hub seeding vs. Random seeding).

5.6.3.2 Efficiency

From figure 5.8, it is easy to conclude that seeding from the hubs will greatly improve the efficiency of cooperation and trust development. In general, hub seeding generates at least 60% in efficiency improvements. Although the differences are rather straightforward and intuitive from figure 5.8, we still performed *t-test* to statistically examine the differences. As expected, all differences are significant at 0.001 level for all 6 conditions[7].

This improvement is more apparent with a larger degree exponential, which is unsurprising given its similar effectiveness. When the degree exponential is small, there are fewer differences between hubs and other nodes. However, when degree exponential is large, it is more likely to form a structure in which one big hub links to many nodes with no other connections (similar to the relationship between node A and nodes {B, C, D}. In this sit-

[7] We performed 6 independent *t-tests*. In each test, we performed *t-test* for the number of average strategy reviews in two groups (Hub Seeding & Random Seeding) at a given degree exponential.

uation, the central hub's behavioral changes will immediately influence his or her satellites' behavior. If behavior changes occur in common nodes first, it requires extra time for the hub to be influenced, and then for the hub to "dispatch" to other nodes. Therefore, the seeding strategy is more efficient when most individuals only have one connection. Based on the above discussion, we have:

PROPOSITION V: *Seeding cooperation and trust in the hubs in early stages will shorten (quicken) the process of cooperation and trust development, especially when there are many individuals with only one connection.*

5.6.4 Sensitivity Analysis

In this section, we present a simple sensitivity analysis (Harrison et al., 2007) to evaluate the results' robustness. In this experiment, the main threat to robustness is the payoff structure. It is necessary to test whether the results are associated with the specific numeric payoff structure we used in the simulation experiment. From chapter 4 (equation 4.8), we know that the key feature for the game structure to ensure the emerge of cooperation is the punishment (g) and cost of online cheap talk (e). Thus, we parameterized them in the sensitivity analysis. Here, we reuse the network structures generated in the experiment. To simplify the discussion, we only take two conditions of degree exponential (λ), which are 2.0 and 2.6.

Figure 5.9 and 5.10 plots the sensitivity analysis results. When punishment/compensation is comparable to cost, cheap talk can encourage the majority of individual simulation processes to reach homogeneous cooperation state. This shows the robustness of PROPOSITION I. It is obvious that the PROPOSITION II holds for most situations. For nearly all combinations of (e, g), more individual simulation processes reach "all-cooperate" states in the condition of $\lambda = 2$ than the condition of $\lambda = 2.6$. By employing the similar method, we also tested

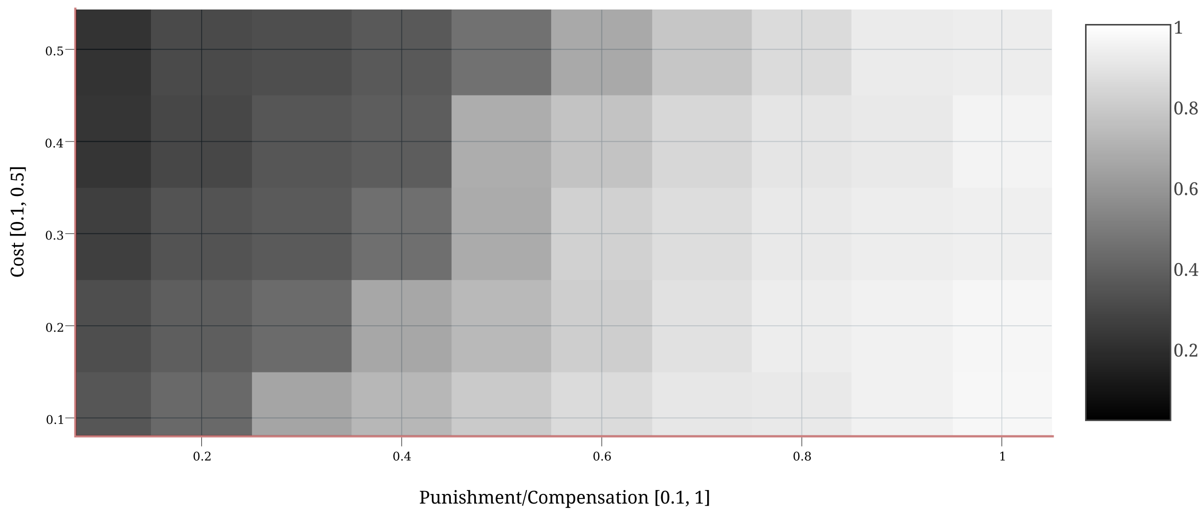

Figure 5.9: Frequency of achieving all-cooperate as a function of cost (e) and punishment to defector (g), $\lambda = 2.0$.

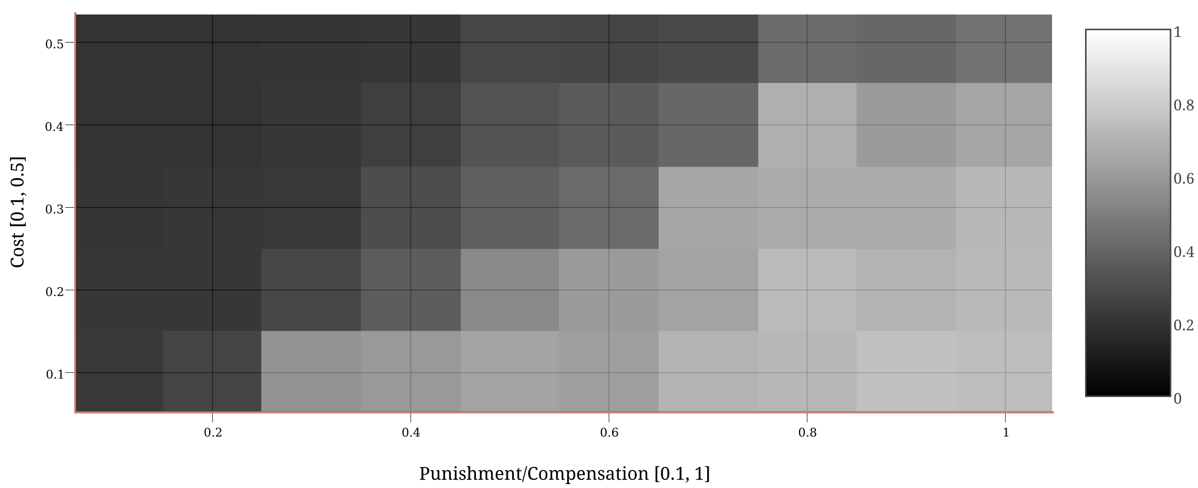

Figure 5.10: Frequency of achieving all-cooperate as a function of cost (e) and punishment to defector (g), $\lambda = 2.6$.

the sensitivity of Proposition III through Proposition V. The results provide general support for robustness. To keep this chapter concise, we will not introduce the details in this section.

The social learning coefficient β is another factor influencing the process. If it is too small, the process may be very slow, because the probability of switching from a low to high payoff strategy may be very low. Given that people may make mistakes at a specific rate (5% in this study), it is even possible that they cannot escape from the low payoff strategy's basin of attraction (Ellison, 2000), which may ultimately lead to the failure of trust and cooperation development.

5.7 Discussion

5.7.1 Implications to Research

This work has implications for future distributed collaboration and trust research. First, the results summarized in PROPOSITION I expand the findings in chapter 4 by incorporating the influence of social networks. We find that cheap talk not only positively impacts cooperation and trust development when individual agents are fully connected, but also helps to promote cooperation and trust development when agents form a pseudo scale-free network. Moreover, we found that cheap talk functions like a "catalyst" even in the new settings. Second, PROPOSITION II & III characterize the impact of a pseudo scale-free network's degree distribution on cooperation and trust development. Compared to prior social network analysis work in CSCW, ours is the first study relating degree distribution to cooperation and trust development, whereas prior research often focuses on a network's micro-level local features, such as whether or not an individual is located in a specific position (McDonald, 2003; Xiao and Tsui, 2007). Our study thus helps develop knowledge about how network features influence collaboration. Thirdly, we tested different seeding strategies' impacts on cooperation and trust development (PROPOSITION IV & V). Although there is rich literature on seeding strategies in marketing and information diffusion (Sundararajan et al., 2013), our study focuses on a less investigated area: seeding strategy's influence on cooperation and trust. Furthermore, we explored the specific conditions in which seeding at the hubs becomes more effective and efficient. “Seeding” for behavior change also extends the traditional application of “seeding” for technology adoption in CSCW literature (Mark and Poltrock, 2001). In developing our model, we viewed cooperation and trust development as a dynamic, adaptive process, whereas existing CSCW literature usually views “cooperation” as a one-shot behavior when studying it in social network setting (Gao, Hinds and Zhao, 2013).

Our main research method was agent-based modeling and simulation; we demonstrated its powerful research potential when combined with theoretical knowledge crossing different domains, and used to investigate complex distributed collaboration systems. For example, the agent-based model described in this chapter focused on the much-research CSCW research topic of distributed collaboration. In developing it, we applied game theory from Economics, and network neighbors' influences from Social Network Analysis. To reiterate, we believe the agent-based modeling and simulation will help researchers develop theory that links traditionally less-related research streams, especially when the research targets are complex and dynamic social and technical systems. The modeling and simulation approach performs well with empirical approaches, and therefore can be combined to form a full research cycle (Ren and Kraut, 2014*c*).

5.7.2 Implications to Distributed Collaboration Practice

Some findings, especially our seeding strategy insights, can be directly applied to distributed collaboration practices. Policymakers may identify team hubs and invest more resources to help them adopt cooperation first. They can also encourage individuals in hub positions to be more open to the changes of those who are not in hubs. In this way, the common nodes' cooperative behavior may be better transmitted to the public by the hubs.

Another possible implication lies in the design of organizational communication networks, considering that our findings indicate that a network with smaller degree exponentials may be more effective and efficient in cooperation and trust development. Policymakers may intentionally manipulate the relationship-building process among group members, for example by creating more opportunities for "common" members to develop relationships. Thus, the degree could be more evenly distributed, and the cooperation and trust development would rely less on those in key positions. The demonstrated positive impact of cheap talk in networked organization settings justifies its practice in cooperation and trust-building.

Although we did not explicitly examine the social learning factor (β), it is important to remove the social learning barriers within the team. If learning is slow, the overall results may be unfavorable. It is necessary to take some action to improve social learning among the team members.

5.7.3 Limitations and Future Opportunities

There are several limitations to this study. First, we only considered degree distribution, which is one of many social network features. Even for networks with the same degree distribution, their actual topologies may vary a great deal. We plan to utilize the simulation-generated data to perform more fine-grained analysis, e.g., to study how local network features influence cooperation and trust development. We will also employ more techniques (see (Jackson, 2010) for reference) to provide further analyses, and we expect more refined findings to emerge . In fact, it is possible that some distributed collaboration networks are not pseudo scale-free networks, and therefore we will try to replicate the virtual experiment in other network models. Additionally, this chapter focuses on the overall simulation results and does not consider certain details about how individual simulation runs dynamically to reach the final state. Studying these detailed simulation records may also lead to useful insights and provide further explanation for the results, including: how C-C strategy is replaced by C strategy; why seeding strategies are more efficient when λ is large; and whether or not there is any dynamic pattern of behavioral change.

Another limitation results from this study's research method. The game theory-based decision model is an abstraction of real world decision processes. It only considered one factor: whether behavioral changes will lead to added benefits. However, many other factors may influence people's decisions. For example, conformity may make an individual hesitate to change, even if new behavior yields better payoff. Incrementally including these social factors may further enhance our understanding of social networks' impact on cooperation development. In addition, people's decisions may also be influenced by those not directly connected to them. More complex decision models can be directly applied to replace the current decision model, which makes the simulation implementation highly reusable. The results can also be viewed as a benchmark for assessing other decision models. Finally, we are also uncertain as to how well the simulated sample represents the whole state space.

To improve this work's relevance to practice, a possible direction is to replace the artificially generated network with an empirical, observational network. We can also introduce the individual characteristics into the agent model and run simulations to study the system's dynamics. The next chapter follows this direction to investigate how individual characteristics, especially baseline trust, influence the trust and cooperation development process.

5.8 Summary

In this chapter, we extended the work described in chapter 4. We introduced an agent-based modeling and simulation study of pseudo scale-free network cooperation and trust development with cheap talk. We applied game theory to develop a decision model that describes how people change their strategies, and then simulated how agents using the decision model form and adapt their behaviors in a pseudo scale-free network. This allowed us to explore the dynamics of cooperation and trust development, and our results confirmed the positive effect of cheap talk in cooperation and trust development. We also identified degree distribution's influence on cooperation and trust development in both effectiveness and efficiency. We then compared two seeding strategies using simulation-generated data and found seeding from hubs is generally more effective and efficient. This work establishes links among informal communication (cheap talk), network features, and cooperation and trust development, and further suggests rich opportunities for future work, and for designing and improving distributed collaboration systems.

Furthermore, our study demonstrates the feasibility of using agent-based modeling and simulation to link traditionally separated research streams. Our study could potentially evolve into a multi-perspective "testbed" for future research, a type of investigation that CSCW practices could benefit from. For instance, practitioners can leverage this method to evaluate the mechanisms they want to introduce into distributed, collaborative groups, and hence may avoid costly yet common "trial and error."

Finally, theoretical knowledge embedded in our model can be combined with creative design intuition to generate effective mechanisms in promoting cooperation and trust. The decision and interaction model presented here was primarily based on game theory, but social science (including economics) literature offers a wide range of theories explaining individual and group decision-making that could be exploited in the future. This argument also applies to social network theories; for example, homophily (McPherson, Smith-Lovin and Cook, 2001) among agents and its influence on non-directly linked agents' decisions may be an important feature that researchers (including ourselves) need to pay more attention to.

The next chapter presents another agent-based modeling and simulation study as an incremental extension of this study. We used an empirical, observational network in the new study, and considered the influences of "baseline trust" in cooperation and trust development with cheap talk. Compared with this study, the new study is more relevant to real world practices. The methods and results are discussed in turn.

Chapter 6

Study III: The Role of Baseline Trust

> Me and Janet really are two different people.
>
> — Michael Jackson

6.1 Introduction

In the prior two chapters, we studied how trust and cooperation emerge with the presence of online cheap talk. Chapter 4 developed the basic model, and chapter 5 investigated the influence of specific network structures with agent based simulation. However, there are two limitations: first, we used an artificially generated network; and, second, we did not consider the influence of individual variation on baseline trust[1].

Most current collaboration simulations typically consider social network structures generated from random or small world graphs (as we did in study II), rather than real world

[1] As a reminder, in this dissertation, we use "baseline trust" to refer an individual's general, global tendency to perceive the trustworthiness of other individuals (or other entities, such as organizations) (Driscoll, 1978). Hence, it is the personality trait aspect of "trust."

interpersonal networks, and do not incorporate empirically observed distributions of trust propensities or correlations between individuals' personal and social network characteristics. These factors are known to profoundly impact the dynamics and causal mechanisms driving collaboration and trust development. Without empirical data to constrain them, under diverse, unrealistic assumptions the simulations may yield different results. Furthermore, due to the lack of basis in reality, these results may not be relevant to, or applicable in, practice.

6.1.1 Empirical Networks vs. Artificially Generated Networks

An artificially generated network provides great convenience and flexibility for investigating a network's social interactions. As already mentioned, though, this flexibility may lack reality and relevance. On the other hand, although using observational (or surveyed) data and randomized experiments is more realistic, it usually fails to assess global characteristics and the effects of multiple strategies due to the need to maintain experimental control and precision (Taylor, Bakshy and Aral, 2013). Besides, investigating the influences of individual variations on baseline trust requires an empirical network. Although we can arbitrarily assign any baseline trust to any node of an artificially generated network, there is no way to assess how much this method distorts reality.

This motivated us to use empirical networks as the infrastructure of our agent-based modeling and simulation. This method has not yet been well adopted, but does provide several benefits (Aral, Muchnik and Sundararajan, 2013). Using real world observations of an empirical network in a simulation immediately leads to ready-to-use practical implications, and avoids the troublesome mismatch between artificially generated and real world networks.

6.1.2 Why Individual Variations on Baseline Trust Are Important?

Individuals differ in many aspects, holding various beliefs, preferences, and corresponding behaviors (Gintis, 2014). An individual's baseline trust, understood as the global belief of other individual's trustworthiness, may strongly influence his or her behavioral choices (Driscoll, 1978). Baseline trust is evident in statements such as: "most people are trustworthy" or "most people are basically good and kind" (Yamagishi and Yamagishi, 1994; Delhey, Newton and Welzel, 2011).

Our prior interviews confirm that global software engineering practitioners vary when it comes to baseline trust. A few interviewees tended to trust everyone without any reservation. One interviewee's comment represents this attitude: "*I trust everyone. Even if they do something wrong, I still believe people are generally trustworthy.*" There are also many interviewees who prefer to "*give others the benefit of doubt.*" In other words, they trust until proven otherwise. By contrast, some individuals need their remote colleagues to prove they are trustworthy. He or she trusts others only when there is a reason. There are still a few people who believe they can never trust any of their collaborators, and may always prefer to be distrustful.

In a networked team, an individual's baseline trust will not only influence his or her own behavioral choices, but also others in the same network. For example, figure 6.1 abstractly depicts a team in which the node "B" has a differing baseline trust. Without considering baseline trust, there are no structural differences between the two networks, and one might draw the conclusion that the pattern of trust and cooperation development are similar. However, when considering baseline trust, the situation changes. For instance, trust and cooperation development may fail if B strongly prefers to be uncooperative. Hence, C and D may not switch to "cooperate." Although baseline trust is important for developing more *situational* trust and cooperation, it is often neglected in literature, especially when considering it in a social network context. To our current knowledge, there is no such work in CSCW or SE literature.

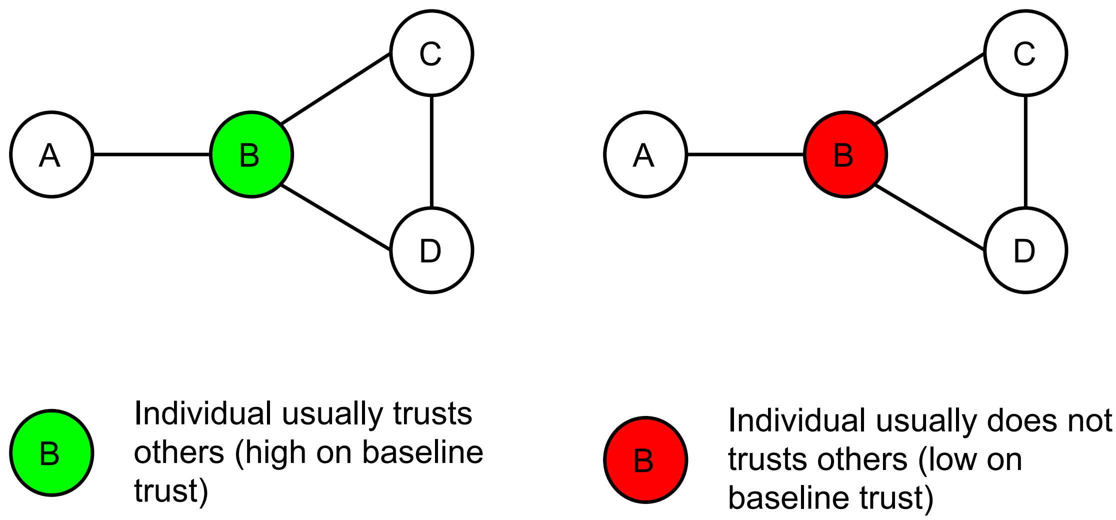

Figure 6.1: An example of baseline trust's influences on trust and cooperation development in a simple 4-node network.

Incorporating baseline trust also enables a new seeding strategy. In chapter 5, we discussed seeding from the hubs. With baseline trust, it is possible to identify the small fraction of agents with the lowest trust. Triggering the diffusion of trust and cooperation from them may be another alternative.

6.1.3 Research Statement

Given the important role of baseline trust, it is necessary to investigate its influence in the diffusion of trust and cooperation, with online cheap talk moderating the process. We also sought to examine whether specific seeding strategies would be helpful when considering baseline trust. Our research merges the flexible, but abstract, simulation-based approach with the more realistic, yet limited, empirical approach, aiming to answer following questions:

With the presence of cheap talk over the internet,

RQ3-1: *how does baseline trust influence the diffusion process of trust and cooperation in the empirical network context?*

RQ3-2: *what are the different seeding strategies' impacts on trust diffusion and cooperation in the empirical network context?*

The remainder of this chapter proceeds as follows. Section 6.2 presents a high-level overview of the research process. Section 6.3 discusses how we solve our major research tasks' key challenges. Section 6.4 introduces our virtual simulation experiment design. Section 6.5 presents the results and findings. Section 6.6 discusses related issues, and section 6.7 summarizes this chapter.

6.2 Research Procedures

6.2.1 Data Collection and Clean

6.2.1.1 Collecting Communication Records

There are some off-the-shelf tools available for collecting online data like mailing lists. However, these tools may have some drawbacks, such as obscuring email addresses for privacy reasons (Bettenburg, Shihab and Hassan, 2009). Moreover, most of them cannot support multi-data sources. We wrote a crawler in `Python` to download the public communication records from various sources on our own. The implementation of crawler utilized and customized crawler4py[2], which is an open source crawler framework developed by the UCI Mondego research group[3]. We tested it with a relatively small project (Asterix) first before applying it to crawl data for this study. In total, we collected 83,627 HTML documents.

6.2.1.2 Data Extraction and Clean

The original downloaded data was not ready for use. Since all crawled documents are HTML files, we leveraged `Python` BeautifulSoup[4] to analyze the HTML files and extract the information we needed. During this step, we also excluded all auto-generated information, such as the commit and build messages automatically sent to mailing list subscribers. To ensure the reliability of this process, we manually examined 100 crawled records for each type of data in each project (total: 700). Then, we compared them with the automatically extracted and cleaned data. Overall, the automatic process provided satisfied results (Precision: 98% 100%, Recall: 97% 100%, varies over different categories). We labelled every cleaned piece of information according to its category, and stored it in a MongoDB database as a JSON Object. Each JSON object represents one of four information types (*IRC message, Discussion, Email, and Issue Discussion*). A JSON object records message content, authors, time, original `URL`, and any other necessary information (e.g., who was mentioned in a message). As a NoSQL database, MongoDB enabled us to query and manipulate the less-structured data. In total, we have 121,539 JSON objects. Then, in order to improve text search and query performance, we built a simple index of the author names to associate users with the text content they produced.

[2] crawler4py: `https://github.com/Mondego/crawler4py`.
[3] `http://mondego.ics.uci.edu`.
[4] BeautifulSoup 4.3.2 `http://www.crummy.com/software/BeautifulSoup/`.

6.2.2 Study Procedure and Main Task

The study design follows standard agent-based modeling and simulation procedures(Macal and North, 2010). Before running the virtual experiment, we built the environment (the networks) for the agent to interact with, and specified their characteristics and decision rule. These map to the three main components shown in figure 6.2. After finishing these tasks, we performed the virtual experiment, analyzed the data collected from the experiment, and then summarized our results and findings.

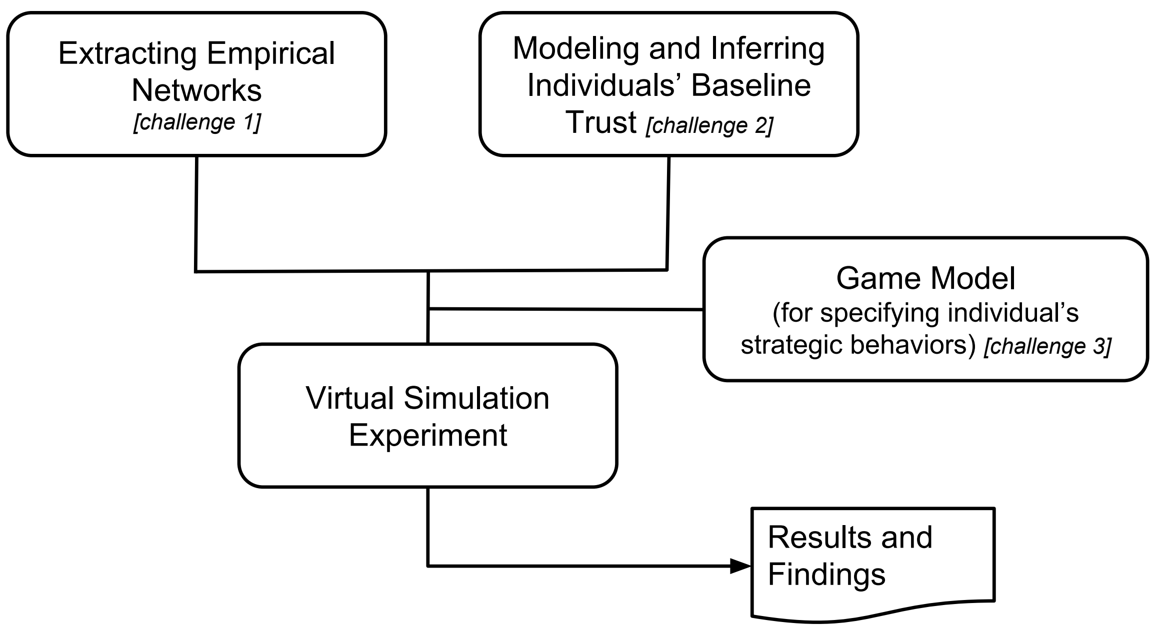

Figure 6.2: The Process of Performing Three Major Research Tasks.

6.2.3 The Game Structure

As we did before, we adopted the payoff structure (see figure 6.3 defined in chapter 4 and modified the classical stag hunt game to describe cheap talk strategies in global distributed collaboration. There are three strategies an agent may use in the new game: $\{C\text{-}C, C, D\}$.

For a more detailed explanation, please refer to the relevant content in chapter 4.

$$
\begin{array}{c|ccc}
 & C\text{-}C & C & D \\
\hline
C\text{-}C & R\text{-}e/2 & R\text{-}e & S\text{-}e+g \\
C & R & R & S \\
D & T\text{-}g & T & P
\end{array}
\qquad
\begin{array}{c|ccc}
 & C\text{-}C & C & D \\
\hline
C\text{-}C & 1.9 & 1.8 & 1.3 \\
C & 2 & 2 & 0.5 \\
D & 0.5 & 1.5 & 1
\end{array}
$$

$e=0.2,\ g=1$

Figure 6.3: The payoff structure and a numeric example of different strategies.

6.3 Technical Challenges and Corresponding Solutions

There are some technical challenges associated with the three major tasks (as we marked in figure 6.2. In this section, we introduce these challenges and corresponding solutions.

6.3.1 Challenge I: Building Empirical Network

6.3.1.1 Challenges

The main challenge for building the empirical network is defining the network structure. Software engineer social network development has been well studied, e.g., (Bird et al., 2006; Hong et al., 2011), etc. In this study, we only build a social network for those who contributed source code to the repositories, but excluded peripheral contributors who may report bugs, but have not contributed code. Following Bird et al. (2006), we only assume those who have exchanged more than 150 messages (in all repositories such as mailing list, IRC, bug report, discussion) are linked developers. We reasonably assume that people who have exchanged an unsubstantial number of messages over a few years can hardly have influenced each other and made significant contribution to the projects. This also ensures that we can precisely figure out people's baseline trust for the relative large amount of records for each individuals. The

IRC	Mailing List	Building System
The Imp	tlann@XXX.com	lannister@google.com

Table 6.1: An Example of Using Multiple Names in Project Repositories. For privacy concerns, the names and emails are not real.

major challenge comes from how to re-establish the individual's identity, because individuals may use different names for different repositories, even in a single project.

Take the example in table 6.1, in which a developer named Tyrion Lannister[5] may use the name "The Imp" in IRC chatting, while using an email address "tlann@xxx.com" in the bug-tracking system. It is almost impossible to automatically infer whether or not these two identifiers represent the same person. This process requires human judgment. In this example, a person who watches the popular television show Game of Thrones likely understands the link between "The Imp" and "tlan@xxx.com." However, the computer will not have enough information to do so. Although many developers tend to consistently use all or part of their email address as their usernames, a substantial proportion use multiple names (Lucene: 31 in 82, Chrome-OS: 35 in 129).

6.3.1.2 Solution

Although we can manually identify this study's mapping, manual identification is not scalable for very large networks. Consider this simple heuristic: if an individual contributor uses the combination of $< username, email >$ in an open source project, he or she might use this combination in other online occasions (e,g., community Q&A platform such as stackoverflow.com, a game community, etc). For a specific username, the corresponding email is probably the email address that has the highest co-occurrence with the name on the Internet.

[5]To protect the privacy of individual information, we use a fake name here.

We developed a method that leverages Google search[6] to retrieve the number of search results for the different combinations of $< username, email >< username, andemail >$. We wrote a script that automatically sends search requests to Google.com by manipulating the search `URL`. Then, we used `Python` BeautifulSoup to analyze the returned HTML. The number of search results can be retrieved from the corresponding HTML elements.

Suppose there are two sets: $Name$ of cardinality N and $Email$ of of cardinality M. The first denotes that the set consists of all usernames $Name[i]$, while the second denotes the set consists of all local parts of email addresses[7] $Email[j]$. Two lists $Name[]$ and $Email[]$ refer to them respectively. Using the local part only provides two major benefits. First, it removes the potential problem of using the special character "@" in the search. Second, in many cases people may only use the local part rather than the full email address to refer themselves, therefore, it actually improves the name mapping algorithm's performance.

This process' formal, step by step description appears in algorithm 1 in next page.

Although it is quite simple, the algorithm yields strong results. We performed a simple experiment using manually matched pairs of *name* and *email* as the ground truth. We first randomly selected 100 pairs of manually matched $< username, email >$. Then, we used algorithm 1 to map over the these pairs' two sets of username and email. The results were encouraging, as it returned 96 pairs of mapping, 93 of which were correct ($precision = 0.97, recall = 0.93$).

[6] We did not use Google Search API, for it returns search results in JSON rather than the number of search results.

[7] "Local part" refers to the string before the @ in an email address, for example, lannister is the **local part** of lannister@google.com.

Algorithm 1 An Simple Algorithm for Mapping Names and Emails of the Same Developer.

```
procedure NameMapping(Name[], Email[])
    new mapping[N]                    ▷ Create a list to store the mapping results.
    new likelihood[N]                 ▷ Create a list to store the mapping probabilities.
    for i ← 1, i ≤ N, i++ do
        p ← no. of Google search results on term "Name[i]"
        likelihood[i] ← 0
        for j ← 1, i ≤ M, j++ do
            q ← no. of Google search results on term "Name[i] Email[j]"
            if q/p > likelihood[i] then
                likelihood[i] ← q/p
                mapping[i] ← Email[j]
            else
                Pass
            end if
        end for
    end for
    return mapping, likelihood
end procedure
```

For this specific study, we also manually matched names and emails while experimenting with the above automatic method. If a developers' social networks are very large, the automatic method is a good option for avoiding time-consuming and costly manual efforts. It could also be applied in developing automated tools in future study.

6.3.2 Challenge II: Extracting Individual's Baseline Trust

6.3.2.1 Challenge

Conventionally, standard questionnaire surveys based on mature psychometric models are the typical method used to infer an individual's trust, e.g., (Delhey, Newton and Welzel, 2011). However, it is very difficult to ensure open source project members' participation, especially considering that most surveys' response rates fall below 20%. It is highly likely that we would be unable to assess the baseline trust of the majority of developers' social networks, which would inevitably lead to highly distorted results. Moreover, a questionnaire-based survey is difficult to automate, meaning it would not likely be used to develop automated decision support tools for GSE practitioners, which thus would thus limit this study's potential practical value.

6.3.2.2 Solution

For each individual, we collected his or her communication records from the IRC channel, mailing list, and issue tracking system. Then, we organized the communication records by month. Using adapted NLP methods proposed in Kanavos et al. (2014) and Kempter et al. (2014), we then calculated the trust score for each month. For each month, we required a substantial number of total messages (¿100) to ensure the reliability the trust score. Otherwise, we simply assigned a "0" to this month. The two-tuple $< month, trust >$ form a time series. We performed a de-trending transformation on the time series using `pracma` package[8] in `R` (R Development Core Team, 2008). The de-trending pre-process is necessary because trust may exhibit an increasing trend that results from continual interaction with other team members. Obviously, the increasing trend is irrelevant to "baseline" trust which is a stable personality trait.

[8] `http://www.inside-r.org/packages/cran/pracma/docs/detrend`.

To ensure the reliability of inferring trust through word count, we used two linguistic lexicons: LIWC (Linguistic Inquiry and Word Count: LIWC 2007 Pennebaker, Booth and Francis (2007); Tausczik and Pennebaker (2010)) and NRC Emotion Lexicon (Mohammad and Turney, 2013). Each contains multiple dimensions for a single word; we only used the dimensions related to "trust" and ignored others, such as "joy."

To optimize the results' reliability, we took two measures. First, we compared the level of precision using *unigram*, *bigram*, and *trigram*. Using *unigram* is obviously problematic. For instance, in the sentence "*I do not believe his commitment*," if one only uses *unigram*, we would miss the negative descriptor "*not*," and incorrectly label the statement as an indicator of *high* trust. We experimented with different combinations, and found that combining *unigram&bigram* almost always returns the best results. This is consistent with the prior research such as (Pang and Lee, 2008; Kouloumpis, Wilson and Moore, 2011).

Second, we computed trust with the LIWC and NRC Emotion Lexicon, compared the results, and found them quite consistent. We computed the correlation amongst two trust score sequences for each individual, and found most of the pairs were significantly correlated. We also compared their means, and identified no significant differences. Hence, we used the average trust value of both lexicons as the final value. Formally, for individual i in month j, his or her trust is:

$$trust(i, j) = \frac{trust_{LIWC}(i, j) + trust_{NRC}(i, j)}{2} \tag{6.1}$$

Figure 6.1 describes the dynamics of a developer's de-trended trust inferred from their word use from 06/2009-12/2014. Although trust changes over the time, it fluctuates either way in relation to the average (the horizontal line in figure 6.1). The average of trust hence approximates baseline trust.

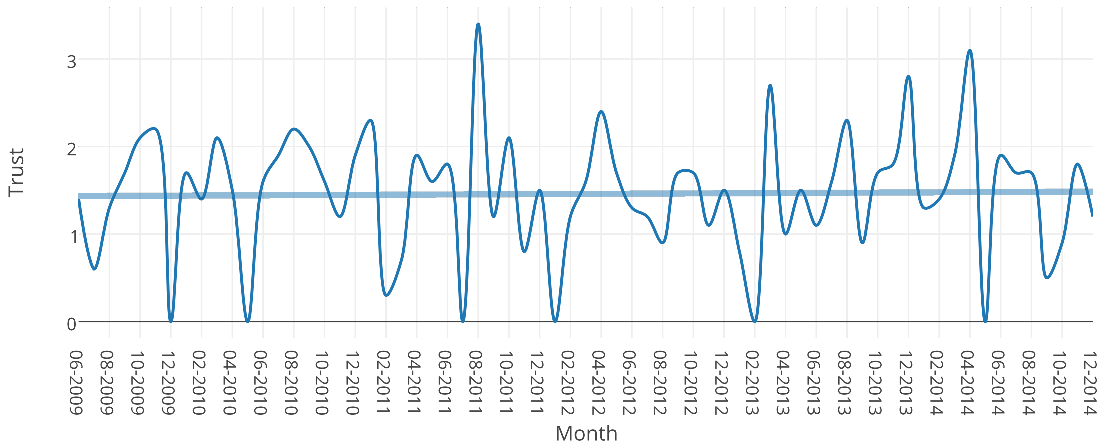

Figure 6.4: The dynamics of a developer's de-trended trust inferred from their word use from 06/2009-12/2014. The line indicates the average trust over this period.

We normalized individual trust to the closed interval [-1, 1] using the sequential combination of *z-score* normalization and *feature scaling*. This transformation is mainly for the convenience of specifying how the baseline trust influences individual behaviors (see the next section for details). Table 6.2 shows the basic statistics of baseline trust in LUCENE and CHROMIUM-OS.

Project	**Sample Size**	**Median**	**Standard Deviation**
LUCENE	81	0.329	0.380
CHROMIUM-OS	126	0.261	0.517

Table 6.2: The basic static of baseline trust. Means are not shown in this table because both means are exactly "0." For some individuals, we cannot resolve the names even using manual mapping, so the sample size is slight smaller than section 6.3.1.

This approach of inferring an individual's baseline trust is still only an approximation of his or her true baseline trust. In fact, even the standard trust measurements' validity is not fully guaranteed, as it is influenced by various factors, including respondent's interpretation, survey execution, and so on (Sturgis and Smith, 2010). Explicit and implicit self-reporting biases may further undermine the survey's validity, since people may prefer to show they trust rather than distrust others. Literature such as (Gou, Zhou and Yang, 2014) established the acceptable level reliability of analyzing an individual's communication records to infer his or her personality traits. The approach is also supported by psycholinguistic models, for example, Tausczik and Pennebaker (2010) and Shen, Brdiczka and Liu (2013). However, it is no doubt worthwhile to further explore and evaluate the approach when deploying it in other domains and scenarios.

6.3.3 Challenge III: Specifying Individual's Decision Dynamics

6.3.3.1 Challenge

Since we can extract an individual's baseline trust, we face another challenge; that is, how can we specify how an individual's baseline trust influences their strategic behaviors? As we did in study II, we assume an agent's decision-making is probabilistic rather than deterministic. A specific strategy's resulting higher payoff does not guarantee the agent will switch to it.

Therefore, we need to figure out how to properly reflect baseline trust's influence in decision models. Should baseline trust directly influence the probability of an individual's strategy selection, or only their subjective judgment of utility (and then indirectly their behavior)? And what is this influence's extent? This should be properly specified.

6.3.3.2 Solution

Obviously, one can define a mechanism that allows baseline trust (as a belief of the world: whether people are generally trustworthy) to directly alter the probability of behaviors (Feltovich, 2000). However, arbitrarily defining such a mechanism[9] is risky, because different mappings may yield quite different dynamics, and there is no easy way to evaluate the results' sensitivity. Performing sensitive analysis over a series of functions is very difficult.

We take a more conservative approach to baseline trust's influence by applying the Belief-Preference-Constraints (BPC) model (Gintis, 2014) to treat "baseline trust" as a type of constraint that influences an individual's subjective evaluation of his or her payoff. In the payoff structure, baseline trust's influence will be expressed by an idiosyncratic payoff. We assume the utility functions in 6.2 to satisfy von Neumann-Morgenstern's utility theorem (Von Neumann and Morgenstern, 2007), which allows us to avoid changing the decision dynamics or arbitrarily changing the probability of a specific strategy, which may lead to unfavorable noise in global level dynamics.

Let S be the set of all possible strategies, and $s \in S$ be a specific strategy. In this study, there are three possible strategies: $\{$*cooperate (C), cheap talk-cooperate (*C-C*), or defect (*D*)*$\}$. $U_i(s)$ denotes the overall value an individual i received by using strategy s. In accordance with the above discussions, $U_i(s)$ is determined by two parts as follows:

$$U_i(s) = \begin{cases} P_{interaction}(s, \_s) + P_{trust} & \text{if} \quad s = C \\ P_{interaction}(s, \_s) & \text{if} \quad s = C - C \\ P_{interaction}(s, \_s) - P_{trust} & \text{if} \quad s = D \end{cases} \tag{6.2}$$

In equation 6.2, $P_{interaction}(s, \_s)$ refers to the (*expected*) payoff received from interacting with one's neighbors. In this study, we only consider the direct influence, i.e., two directly connected individuals, A and B. Let's suppose he or she has M neighbors, and (m_C, m_{C-C}, m_D)

[9]Here, "mechanism" refers a function mapping baseline trust to a specific value of increasing or decreasing probability.

denotes the numbers of his or her neighbors who choose three possible strategies at period $t-1$. $P_{interaction}(s, \_s)$ can be written in the following form:

$$P_{interaction}(s, \_s) = \frac{m_C}{M} \times p(s, C) + \frac{m_{C-C}}{M} \times p(s, C-C) + \frac{m_D}{M} \times p(s, D) \qquad (6.3)$$

In 6.2, P_{trust} refers to the idiosyncratic payoff resulting from different baseline trust levels. We simply use a linear function to describe it:

$$P_{trust} = c \times \textit{baseline trust}. \qquad (6.4)$$

The constant c is in the range of [0, 1] where "0" means baseline trust has no influence, and "1" indicates baseline trust has full influence. This structure has been used in literature such as (Skyrms, 2005; Wang and Redmiles, 2015*b*) to address individual subjective bias or preferences' influence on payoff evaluation. We can simply parameterize the constant c to examine the results' sensitivity.

Now let's take a closer look at equation 6.2. As we mentioned in section 6.5.2, baseline trust ranges from [-1, 1]. If an individuals' baseline trust is positive, they receive extra idiosyncratic payoff from using "cooperate" strategy. This is intuitive because the "cooperate" strategy fits their personality (they tend to trust others). If they select "defect," the overall value will be less than the value they could get from interacting with their neighbors, because they may feel unhappy for selecting a strategy that does not fit their personality. Individuals with a negative baseline trust tend to distrust others and work independently. Therefore, the overall value of using "defect" will increase, whereas they may feel uneasy using the "cooperate" strategy. For "*C-C*," we assume neither population has a special preference for it. Therefore, their payoffs are solely determined by the interactions.

Since baseline trust's influence is only reflected by payoff changes, we do not need to change the logistic learning rule we used in chapter 5. As in chapter 5, parameter β represents the sensitivity towards payoff change: the larger β is, the more likely it is that he or she will choose a best reply, given the actions of his or her neighbors.

$$\begin{gathered} \frac{e^{\beta U_i(s)}}{\sum^S e^{\beta U_i(s)}} \\ \beta \geq 0 \end{gathered} \tag{6.5}$$

As in chapter 5, figure 6.5 shows an example of an individual's decision-making process. In the original example (figure 6.3.a), A changes her strategy from "*C*-*C*" to "*C*," since the latter results in a better payoff (1 vs 0.916, see chapter 5 for details). But in figure 6.3.b, if we assume the baseline trust is -0.2 and $c = 0.5$, the expected payoff of playing "*C*" becomes $1 - 0.5 \times -0.2 = 0.9$. However, the expected payoff of playing "*C*-*C*" is still 0.916 according to equation 6.2. Obviously, $0.9 < 0.916$, so the probabilities of using "*C*" and"*C*-*C*" are specified in equation 6.6:

$$\begin{aligned} P(C) &= \frac{e^{0.9\beta}}{e^{0.9\beta} + e^{0.916\beta}} < \frac{1}{2} \\ P(C - C) &= \frac{e^{0.916\beta}}{e^{0.9\beta} + e^{0.916\beta}} > \frac{1}{2} \end{aligned} \tag{6.6}$$

If $\beta \to +\infty$, the learning process is deterministic, and A will definitely keep using *C*-*C*.

A minor challenge is to determine who will be the next individual to review their strategy. We adopted the same rule we used in chapter 5 to enable simple social learning in a network. That is: (1) if an individual is selected to review her strategy and changes it in period t, the next selected individual should be one of her neighbors; (2) otherwise, randomly pick one from all individuals. As we did before, we also allow individuals to make a small rate of mistakes to reflect their bounded rationality. This also ensures the process will eventually reach absorbing states (Young, 1998).

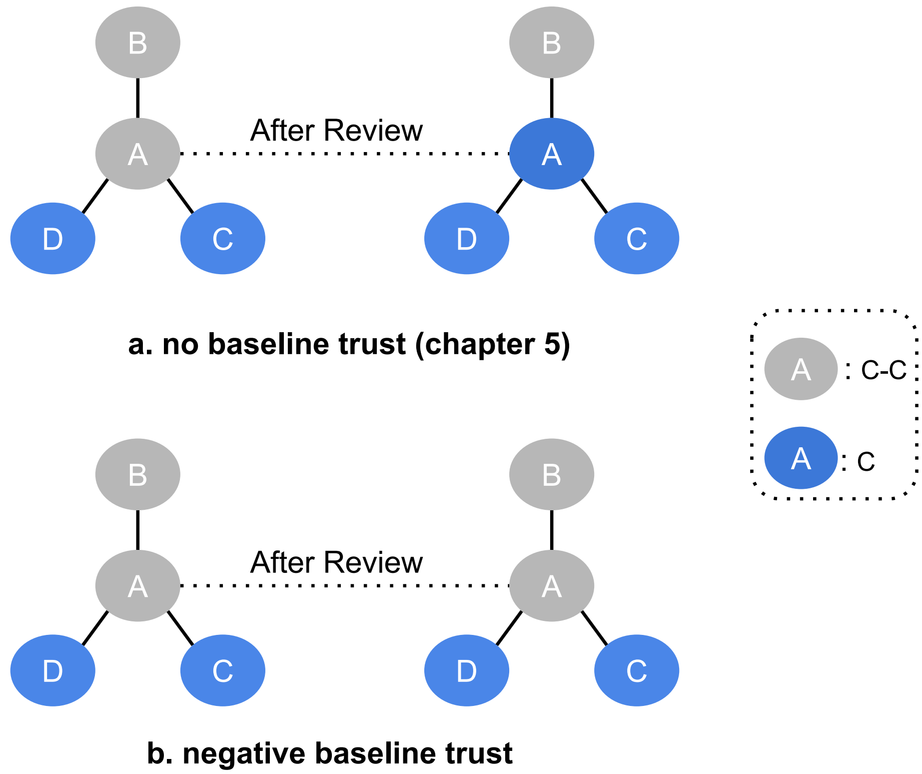

Figure 6.5: An comparison of applying the decision model to make decision ($\beta \rightarrow +\infty$). In this example, A changes her strategy after the review process.

6.3.4 Summary

By solving the above challenges, we can build infrastructures (Networks and Individuals' Baseline Trust) for the virtual simulation experiment. Figure 6.6 visualizes the LUCENE's network with individual's baseline trust depicted in different levels of grayscale. The CHROMIUM-OS's network is similar. In this study, we only considered the largest connected component[10] of each network, and removed those individuals who do not belong to that component. Please note that, although the average of baseline trust in each group is exactly "0" due to the *z-score* normalization, both groups have more members with positive baseline trust than those with negative baseline trust (see table 6.2, both medians are positive).

[10] Informally, the largest connected component is the largest set of nodes and edges in which there is a path formed by edges between every pair of nodes.

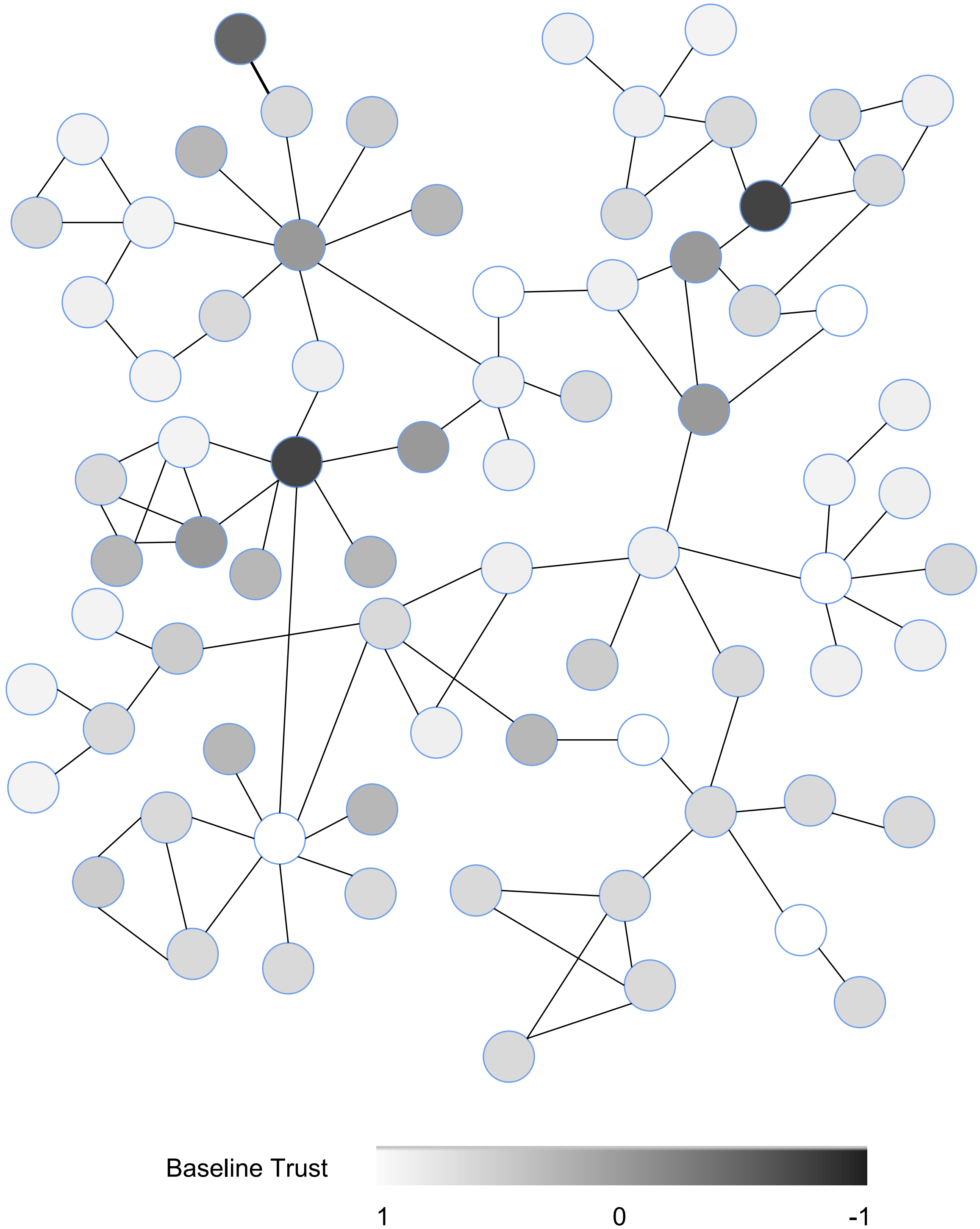

Figure 6.6: The developers' social network of LUCENE. The gray-scale indicates each individual's baseline trust.

6.4 Virtual Experiment Design

The virtual experiment contains two parts, which correspond to this study's two foci (baseline trust and seeding strategy) and two research questions.

Obviously, if $c = 0$, the baseline trust will have no influence. Therefore, it serves as a benchmark for assessing the influence of different degrees of baseline trust. We sequentially manipulated baseline trust's influence (c) with the interval of 0.1 from 0.1 to 1. Thereby, we have 10 conditions: $c = 0.1, c = 0.2, ..., c = 1$. For each condition, we perform 1,000 independent trials for each of LUCENE and CHROMIUM-OS networks. In total, we have $10 \times 2 \times 1,000 = 20,000$ independent simulation trials. At the beginning of each trial, 10% individuals are randomly selected as seeds who use the "cooperate" strategy. To reduce the complexity, we set all individual's learning factor $\beta = 10$ for all simulations [11].

We tested two seeding strategies: **seeding from the hubs**, and **seeding from the distrustful**. As opposed to the first part, seeds are not randomly assigned. We rank all individuals according to their hub score[12] and their baseline trust. Then, for seeding from the hubs, we choose 10% of individuals with highest hub score, and begin simulation with them. For seeding from the distrustful, we choose 10% individuals who are lowest in baseline trust, and begin simulation with them. All other settings are kept intact. As in the first part, for each seeding strategy in each baseline trust condition (c), we perform 1,000 independent trials for each network. We reuse the discrete event simulator developed in study II to manage the simulation process. We keep detailed records of every simulation trial's state in every period.

[11] The distribution of learning factors may have some correlation with the distribution of baseline trust, future research may need to address this point.

[12] The hub score was calculated with the method described in (Manning, Raghavan and Schütze, 2008), we made slight changes.

6.5 Results and Findings

6.5.1 Overview of Results

In this section, we reported the results of running virtual simulation experiments. The analyses yield six propositions, which provide answers to the two main research questions. Table 6.3 summarizes the correspondence between the research questions and these propositions. The results are aggregated from all 1,000 trials in a given experiment condition. We used `R3.0.2` for Mac to perform all statistical analyses reported in this section. The remainder of this section provides more detail.

RQs	Proposition	Key Point
RQ3-1	PROPOSITION I, II, & III	When considering baseline trust, the simulation results show: (a) *C-C* is still important at the beginning of diffusion and possible to be a long term stable strategy, (b) more diverse diffusion trajectories appear in later phases, (c) diffusion is more limited when baseline trust's influence becomes substantial, and cheap talk becomes a stable strategy in the long run, (d) the average speed of diffusion improves, while it varies more significantly.
RQ3-2	PROPOSITION IV, V & VI	Both seeding strategies (seeding from the hubs and seeding from distrust) positively influence the effectiveness and speed of diffusion. Using them together provides even better performances.

Table 6.3: Summary of findings and corresponding research questions.

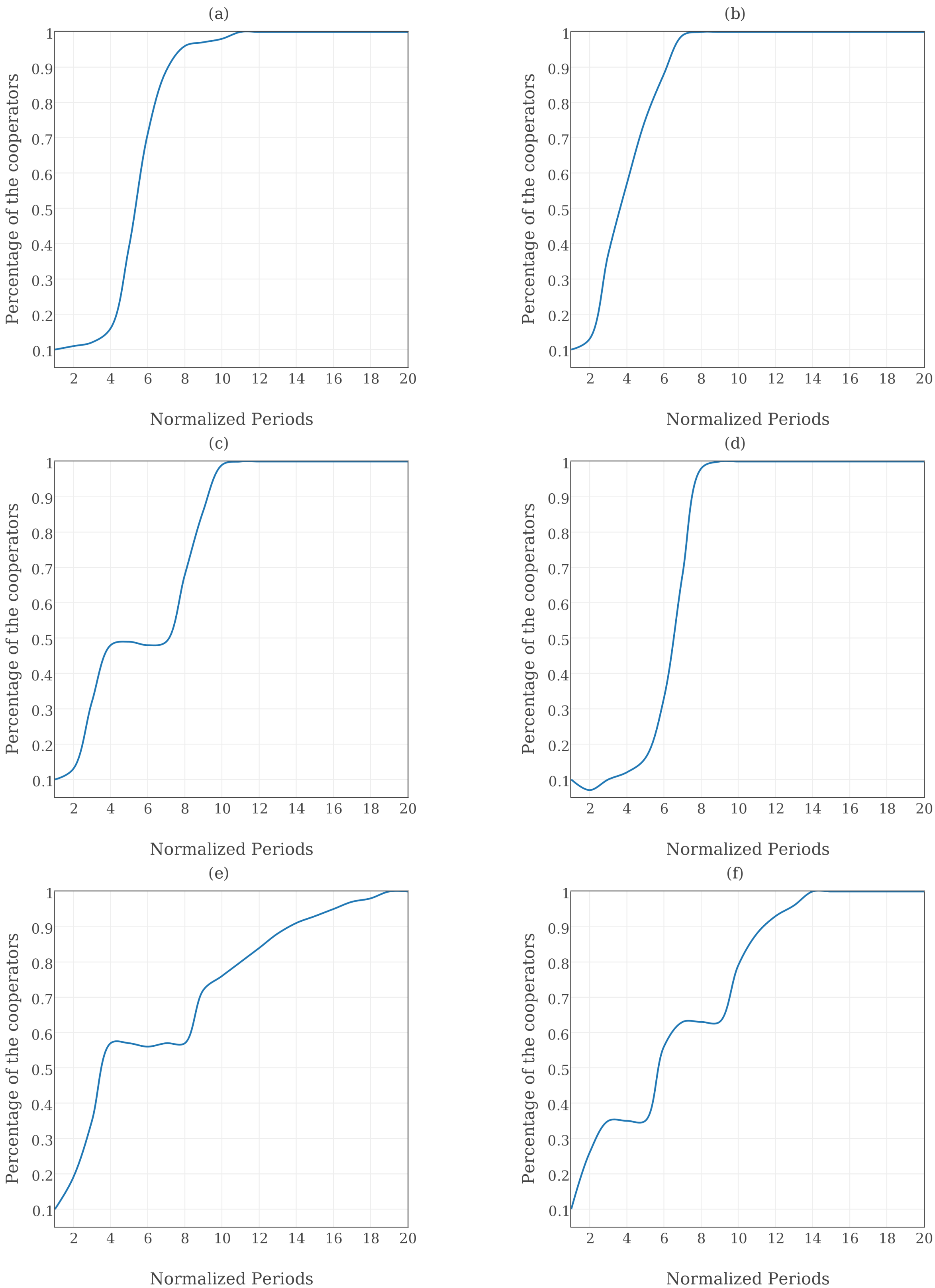

Figure 6.7: Different possible full diffusion trajectories on LUCENE network ($c = 0.6$).

6.5.2 Diffusion Trajectories

We then examined the influence of coefficient c, which determines baseline trust's different degrees of influence. If $c = 0$, baseline trust has no influence (see equation6.2. With the increase of c, the influence of baseline trust becomes more significant.

6.5.2.1 Diverse Trajectories of Diffusion

Figure 6.7 (a through f) shows the the six trajectories of full diffusion of trust and cooperation in LUCENE's network. Six similar trajectories were also observed in CHROMIUM-OS. There are no significant differences except the total periods in diffusion process. In order to keep this chapter concise, we will not plot them again. The diffusions are generally quicker in LUCENE's network since it is smaller. We cannot rule out the possibility that different network topologies and baseline trust distributions also contribute to the difference of the speed of diffusion, but they are beyond our interest.

First of all, chap talk is still important. Almost all trajectories start with a relatively flat (or even slightly low) part. During these periods, agents are most likely to switch to cheap talk, since it is natural after realizing your neighbors have not been cooperative. Then, the trajectories become fairly diverse and non-classic with the increase of baseline trust's influence. When $c = 0$, the majority of diffusions exhibit the classic S-curve (Abrahamson and Rosenkopf, 1997). However, more diffusion trajectories appear with the increase of c, which indicates that baseline trust diversifies trust and cooperation's diffusion. It is reasonable for introducing the influence of baseline trust to make the payoff structure personalized and no longer static (see equation 6.2).

Perhaps the most interesting trajectory is that which exhibits a "staged" pattern (figure 6.7.f). We examined the detailed process behind these patterns and noticed a few highly "distrustful" individuals cause the "platforms" in the diffusion trajectories. The process is stuck and only moves forward after they "mistakenly" change their behavior; it becomes exhaustively long. There are not many processes that demonstrate this trajectory; however, its frequency becomes non-trivial when c is large ($c \to 1$). Compared with other patterns, this pattern is less investigated. Even in Rossman, Chiu and Mol (2008) and Rossman (2012), which document several non-classic diffusion trajectories, the "staged" trajectory is not covered.

For the empirical network of LUCENE, all trajectories are observed when $c \geq 0.5$. For the empirical network of CHROMIUM-OS, all trajectories were observed when $c \geq 0.4$. This suggests that the critical value for the diffusion process expresses that all trajectories may depend on the profile of baseline trust distribution.

6.5.2.2 The Effectiveness of Diffusion

Figure 6.8 shows the frequency change of the individual simulation process reaches a stable homogeneous trust state (full diffusion) under different c. Both LUCENE and CHROMIUM-OS show similar patterns. An apparent patterns is that there are fewer processes reaching full diffusion with the increase of baseline trust's influence. In $c = 0$ situation, the majority of simulations achieve full diffusion. However, when $c = 1$, almost 29% simulations reach limited diffusion for LUCENE network, and around 37% for CHROMIUM-OS. The decrease of the diffusion's effectiveness may be non-linear (see figure 6.8). The full diffusion rate drops faster when c is between 0.4 and 0.6, which indicates that there may be some qualitative change when c falls in this interval. Among those limited diffusion processes, we observed the existence of *Cheap talk-Cooperate (C-C)* as a long run stable state. That is because switching to cooperator becomes less attractive for some individuals when the extra payoff is offset by their idiosyncratic payoff related to baseline trust.

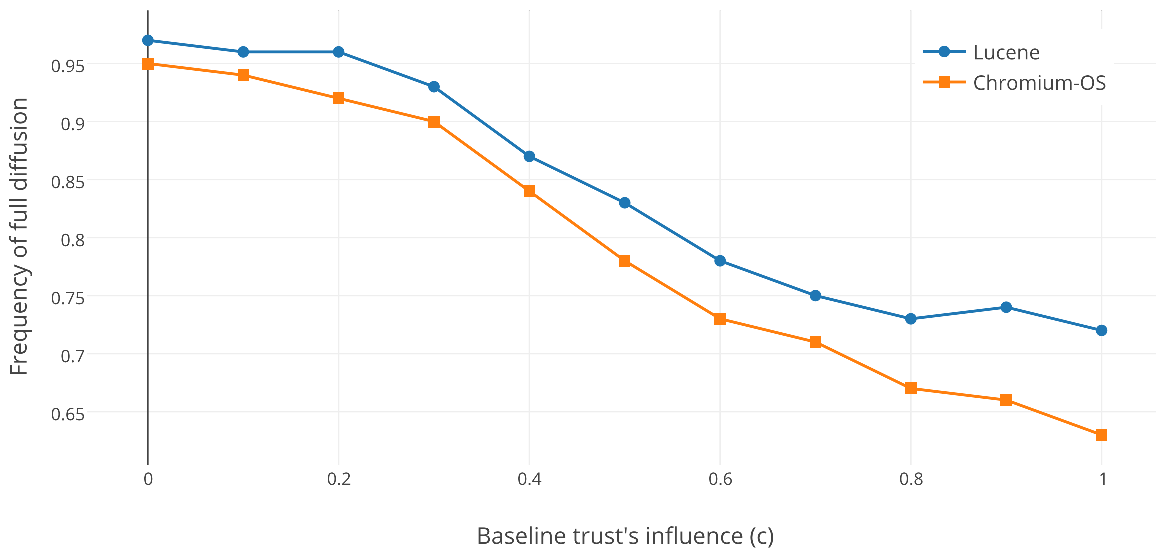

Figure 6.8: The change of frequency of full diffusion under different c.

6.5.2.3 The Speed of Diffusion

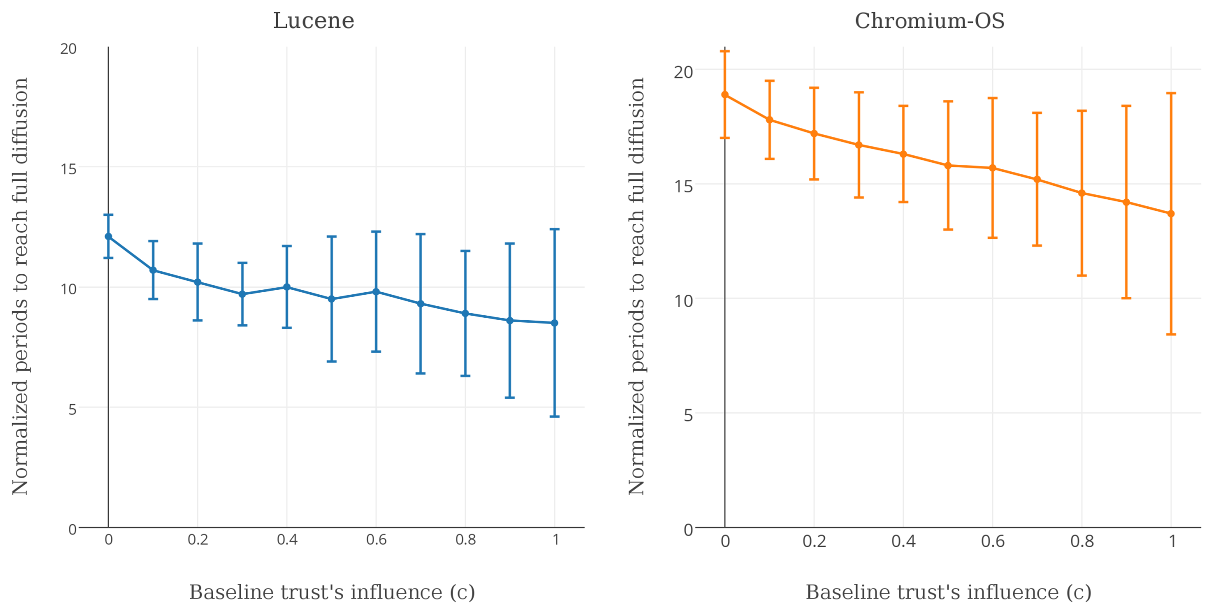

Figure 6.9: The change of the number of normalized periods to reach full diffusion under different c.

Figure 6.9 shows the average number of periods (normalized) to reach full diffusion[13] in each condition. In figure 6.9, the average speed of diffusions is faster with the influence of baseline

[13] We ignored all limited diffusion simulations in calculating the averages.

trust, although the improvements are not very significant. This may result from the fact that the majority of individuals have positive baseline trust in two projects. However, the speed of diffusion varies much more significantly. We performed a simple ANOVA test on the sample of full diffusions in three conditions ($c = 0$, $c = 0.5$, and $c = 1$), and the results suggest there are significant differences in the variances.

6.5.2.4 Summary of Findings

The main findings can be summarized as the following three PROPOSITIONS:

PROPOSITION I: *C-C is important at the diffusion process' outset. The diffusion of trust and cooperation exhibits non-standard trajectories when baseline trust has substantial influence on an individual's subjective payoff evaluation.*

PROPOSITION II: *The probability of a limited diffusion of trust and cooperation becomes greater when baseline trust substantially influences an individual's subjective payoff evaluation. Also, strategy C-C may become a long-term stable strategy.*

PROPOSITION III: *Suppose the baseline trust has substantial influence, then the average speed of diffusion improves if the majority of individuals have positive baseline trust; however, the speed of diffusion varies more significantly.*

Baseline trust matters! PROPOSITION I, II, & III not only re-confirm the importance of cheap talk, but also illustrate how baseline trust shapes the diffusion of trust and cooperation. As we did in chapter 4, we performed sensitivity analysis on payoff structures. Under the condition that the payoff from interaction is at the same level of baseline trust, the results are robust enough when punishment/compensation is comparable to cost.

6.5.3 Seeding Strategies

6.5.3.1 Seeding from The Hubs

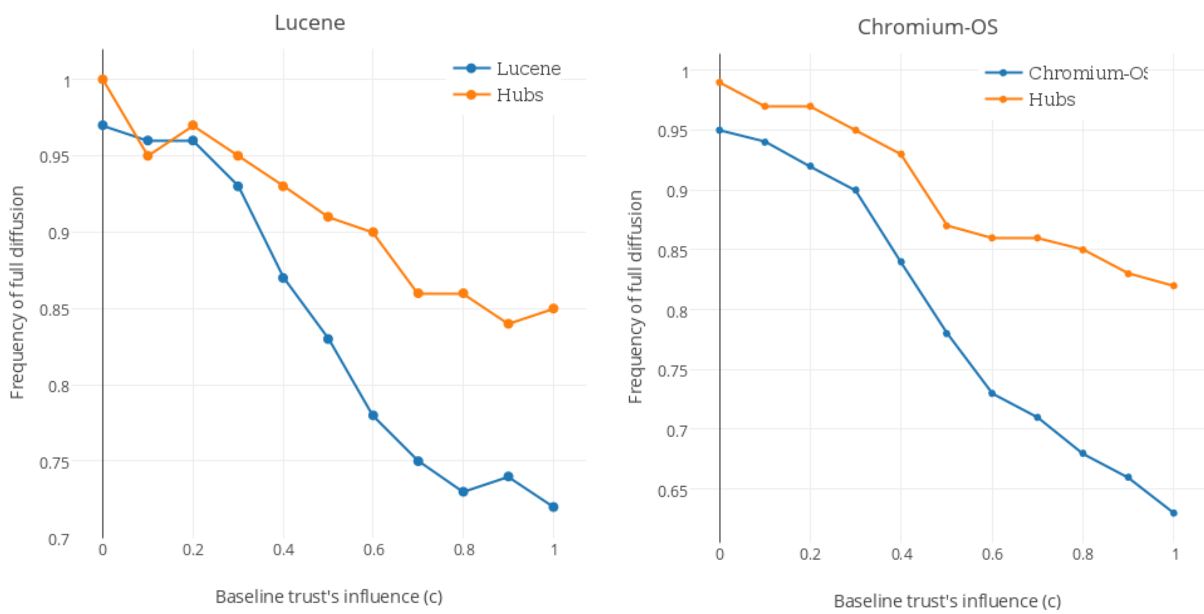

Figure 6.10: Comparisons of between random seeding and seeding from the hubs (frequency of full diffusion).

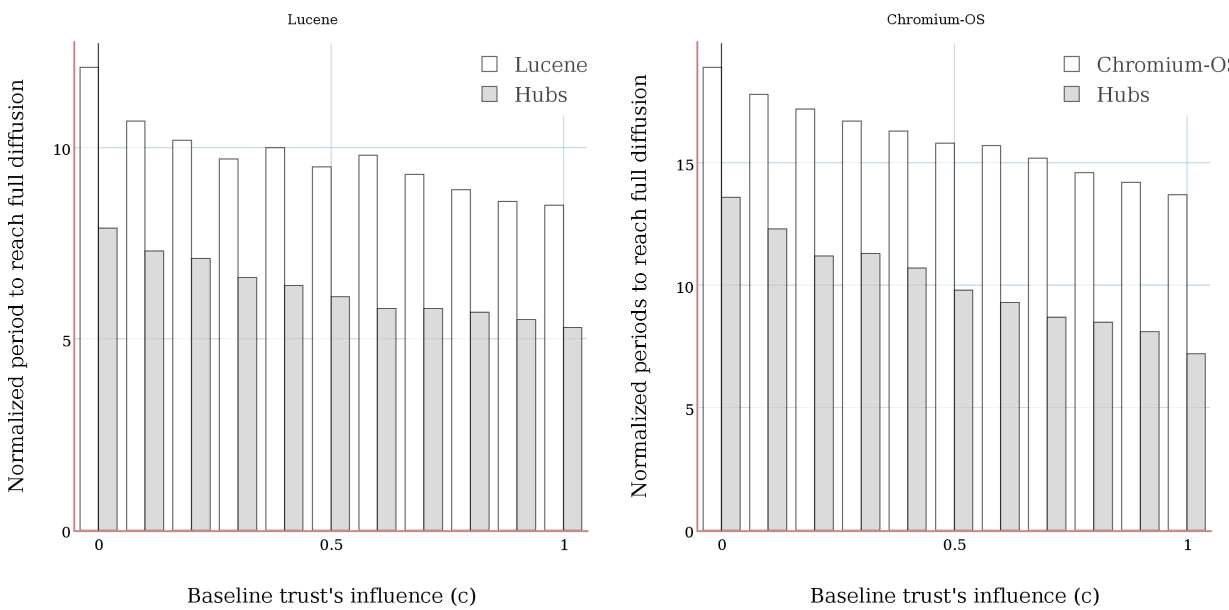

Figure 6.11: Comparisons of between random seeding and seeding from the hubs (periods to reach full diffusion).

Figure 6.10 shows the influence of seeding from the hubs on the effective development of co-operation and trust. Obviously, this effect is more significant when baseline trust's influence is large ($c \to 1$). Similarly, the speed of diffusion also improves by using this strategy (see figure 6.11, and simulation results from both LUCENE and CHROMIUM-OS networks show the same patterns. Seeding from the hubs also helps to reduce the uncertainty about how long it takes to reach full diffusion in the worst case.

6.5.3.2 Seeding from The Distrustful

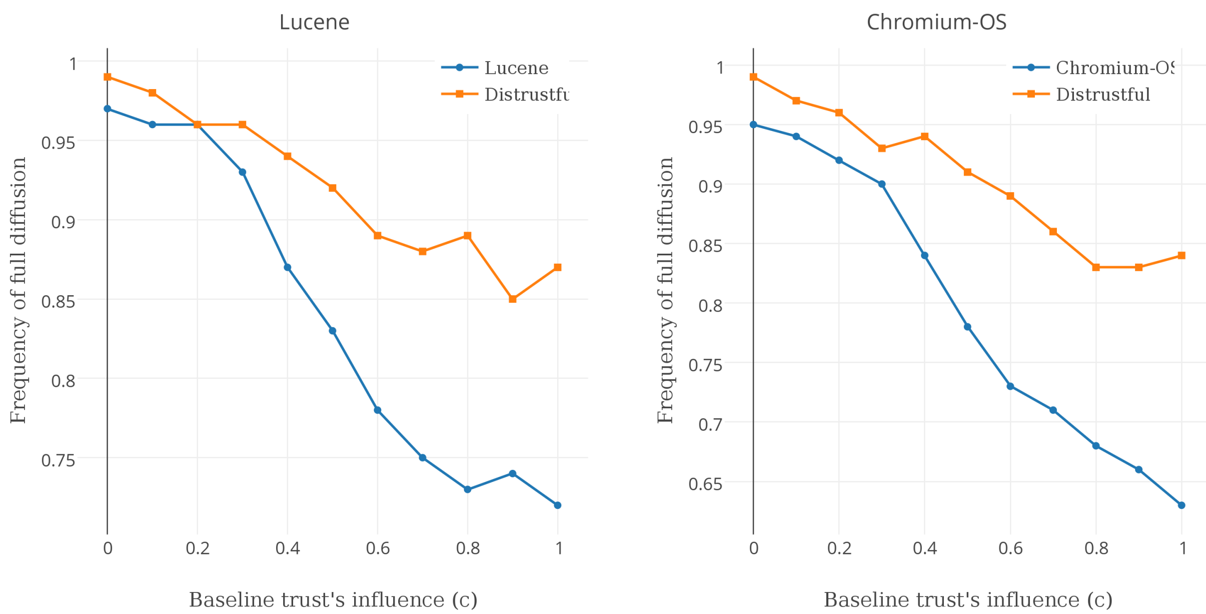

Figure 6.12: Comparisons of between random seeding and seeding from the distrustful (frequency of full diffusion).

The second seeding strategy examined in this study is seeding from the distrustful. The simulation results suggest it is also an effective and efficient way to improve the diffusion of trust and cooperation. Figure 6.12 shows that for both networks, seeding from the distrustful always brings better than random results for almost all conditions. The only exception is $c = 0.2$ for LUCENE network. For the speed of diffusion, figure 6.13 indicates the exact same patterns.

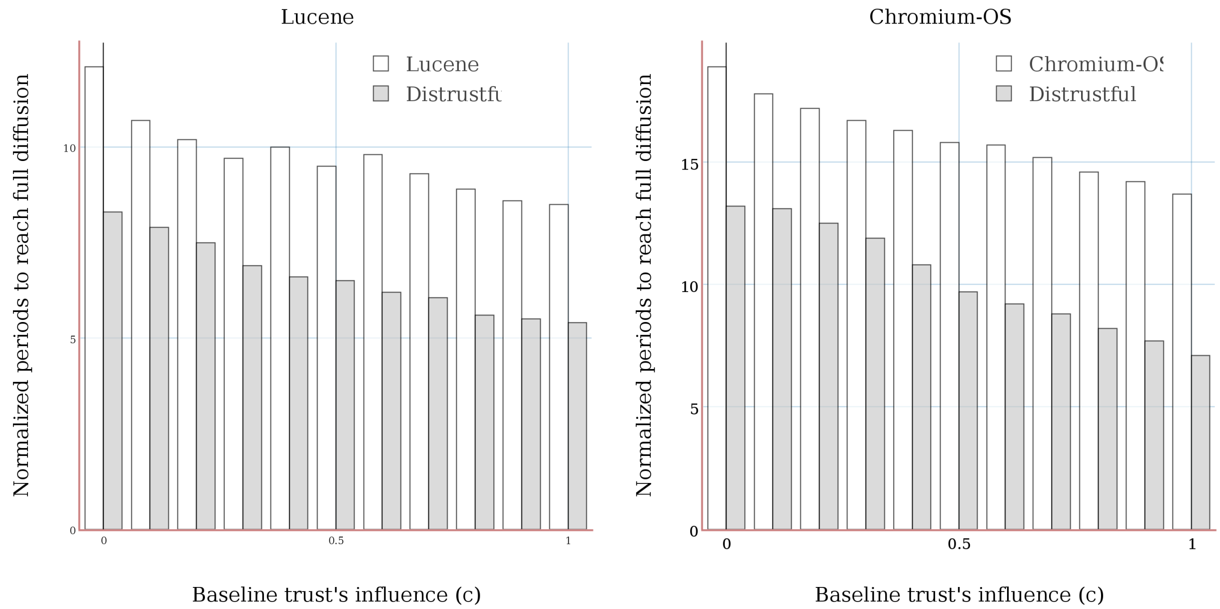

Figure 6.13: Comparisons of between random seeding and seeding from the distrustful (periods to reach full diffusion).

6.5.3.3 Joint and Independent Effect

Please note that there may be some correlations between "hubs" and "the distrustful." Indeed, in the two empirical social networks used in this study this correlation exists. Those who are distrustful are slightly more likely to appear in the hub positions, which is why we do not simply put the term "Ceteris Paribus" in Proposition IV and Proposition V. Due to the restrictive empirical network structure, we cannot fully evaluate their effects independently. However, it is reasonable to assume that "seeding from the distrustful" has at least a moderate level of independent positive effects. Intuitively, for example, when $c = 1$, seeding from the distrustful yields better results on both the effectiveness and speed of diffusion. Therefore, there must be an effect resulting from the independent influence of seeding from the distrustful strategy. The independent effect of seeding from the hubs can be established by similar arguments.

6.5.3.4 Summary of Findings

PROPOSITION IV: *The effectiveness and the speed of diffusion improves when seeding from those in the hub positions.*

PROPOSITION V: *The effectiveness and the speed of diffusion improves when seeding from those who are distrustful. The effect becomes increasingly significant with the higher influence of baseline trust.*

PROPOSITION VI: *The combination of both strategies yields better results, although they both have independent positive effects on the speed of diffusion.*

Seeding strategy matters! PROPOSITION IV, V, & VI show that using proper seeding strategy would help improve the effectiveness and speed of trust and cooperation development. Although there are some correlations between the hubs and the distrustful, both of them have their own impact. Again we performed sensitivity analysis on payoff structures. Under the condition that the payoff from interaction is at the same level of baseline trust, the results are robust enough when punishment/compensation is comparable to cost.

6.6 Discussion

6.6.1 Implications

6.6.1.1 Implications to Research

This work has implications for future distributed collaboration and trust research. First, we explore how baseline trust, as an important individual characteristic, shapes the diffusion of trust and cooperation in a social network setting. The state of the art social network research to focus on the aspects resulting from different proximities of individuals, while assuming individual nodes are as same as each other (Sundararajan et al., 2013). Our work thus demonstrates the importance of considering individual characteristics.

We devised a new way to conduct data-driven, agent-based simulation study, which combines abstract, flexible simulation and empirical, observational data collected from real world software projects. We demonstrate that it achieves both methods' advantages, allowing us to explore dynamics in individual and team level while keeping the results and findings high relevant to practical application. This method has great potential in CSCW and SE research. Researchers may utilize it in developing theory or guiding empirical study, since it produces empirically testable proposition/hypothesis, especially for research targeting complex, dynamic social and technical systems in which individual differences cannot be ignored.

Another contribution is the model itself, which represents an extensible theoretical knowledge for understanding both how people interact and influence each other in a social network, and how their baseline trust influences their behavioral decisions. Integrating these social theories into a consistent and comprehensive model enables us to examine a rich set of factors together in a unified platform. Moreover, the model can be extended by incorporating more complete theories, which can be easily achieved through coding these theories to the decision rules.

In developing the infrastructure for the simulation experiment, we adapted, developed, and invented several methods to extract social structure and individual's baseline trust from team communication records. These methods can be applied to other theoretical and empirical studies.

6.6.1.2 Implications for Practice

Some findings, especially our seeding strategy insights, can be directly applied to distributed collaboration practices. In chapter 5 , we demonstrate how identifying team hubs and investing more resources to help them adopt cooperation first (seeding from the hubs) may be an effective and efficient way to improve the diffusion process. This study confirms the usefulness of this strategy in empirical network settings. Moreover, we show that "seeding from the distrustful" is also a effective strategy; one might even combine them to achieve better results.

Through our study it also becomes clear that some individuals with lower baseline trust may potentially block the progress of diffusion. Therefore, another possible implication lies in designing organizational communication networks to minimize such individuals' influence. This can be achieved by adding new links between unconnected people; for example, connecting A to C in figure 6.1 may avoid the negative influence of B, and may eventually force B to switch to cooperate.

6.6.2 Design Implications

This study opens possibilities for designing tools that support team collaboration. The agent-based model can be expanded and augmented with a rich user interface to serve as a decision-making tool for GSE practitioners. By changing the model's parameters (e.g., payoff structure, social learning factors, etc), project managers and team leaders can run "what-if" experiments to navigate the mechanism design space and explore different scenarios, matching decisions to the team's context. In this way, the team may be able to select the best mechanism, such as a combination of different seeding strategies, to proactively facilitate the team work process. The agent-based model is dynamic, which enables tools based on it to identify the influence of a specific scenario, as well as develop insights into the long-term consequences of complex social technical processes.

6.6.3 Threats to Validity

Construct Validity The main construct in this study is "baseline trust," which refers to an individual's general, global tendency in perceiving the trustworthiness of other individuals (or other entities, such as organizations) (Driscoll, 1978). It is a dimension of personality, yet we did not use traditional psychometric approaches to measure it due to practical restrictions. We adopted an unconventional method similar to (Gou, Zhou and Yang, 2014), for which there are some early experimental evidences to confirm its validity (Shen, Brdiczka and Liu, 2013; Tausczik and Pennebaker, 2010). To establish full confidence in this method, more evaluations must occur in future.

External Validity This study utilized empirical data from two open source projects. We cannot guarantee that the results and findings of this study are generalizable to other projects. However, replicating this study with different empirical data sets would inductively develop solid knowledge on the research topic. Researchers would become more confident in a theory when similar findings emerge in different contexts (Basili, Shull and Lanubile, 1999).

Internal Validity There is no significant threat to internal validity. The empirical data used in this study are public communication records collected by computer programs, and thus there is almost no human judgement involved in the data collection, extraction, and cleaning process. The agent-based simulation is autonomous, and we only specify the rules that are applied to all agents without any differences.

6.7 Summary

In this chapter, we presented a study that investigates the influence of baseline trust in the diffusion of trust and cooperation. We used empirical data from two real world global software engineering projects to inform, build, and analyze data-driven simulations. A set of main results emerged from our simulations. First, baseline trust yields more diverse non-classic diffusion trajectories, and may in the long run make "*Cheap talk-Cooperate (C-C)*" become a stable state. Second, with the increase of baseline trust's influence, the speed of diffusion may improve if the majority of individuals have positive trust; however, it may vary more significantly. Third, seeding is more effective when combining both hubs and the distrustful. The findings call into question some conventional wisdom about diffusion trajectories when individuals' payoffs become subjective and heterogeneous, and suggest new seeding strategies that integrate both network and individual characteristics.

The next chapter concludes the dissertation. It summarizes the contribution of each chapter, and discusses general implications, limitations, and future work.

Chapter 7

Summary and Conclusions

This chapter summarizes the dissertation and presents clear statements about answers to its central research questions. Furthermore, it discusses this research's implications from four perspectives, as well as its limitations and promising future directions.

7.1 Summary

Chapter 1 presented the idea of informal, non-work-related conversation's role in the emergence of trust and collaboration in globally distributed software engineering activities. The chapter looked at several scenarios and empirical results grounded in real-world observations of GSE teams, in which team members are usually unfamiliar with each other and have limited opportunities for face-to-face interaction.

Chapter 2 formally presented the research question we explore in this dissertation, and = briefly introduced the overview of research approach and data sets. We also pointed out potential contributions by answering the research questions.

Chapter 3 reviewed related work on trust and informal, non-work-related communication in distributed teams. We argued that the perspective of social interactions may not sufficiently develop a full understanding of cheap talk. Hence, we posit that cheap talk may need to be investigated from the perspective of strategic behavior. In this chapter, we also argued why stag hunt offers a good abstraction of collaborations in global software engineering.

Chapter 4 described our first study, which investigated the role of cheap talk over the internet in the emergence of trust and cooperation. We introduced "cheap talk and cooperate (C-C) as a new concrete strategy for the classic stag hunt game, and conducted theoretical analyses on its dynamics in a fixed team setting. Together with two empirical case studies on LUCENE and CHROMIUM-OS, we identified cheap talk's positive effects in enabling trust and cooperation's development. Moreover, we explain why and how cheap talk could serve as a *catalyst* for establishing team trust and cooperation.

Chapter 5 presented study II, which incorporates the influence of social network structural characteristics. This study reveals that cheap talk remains an important mediation for trust and cooperation development in a pseudo scale-free network. We investigated how a network's degree of distribution influences the trust and cooperation development with the presence of cheap talk. We also demonstrated that seeding trust and cooperation from the hubs could indeed be effective and efficient.

Chapter 6 further explored the influence of individual baseline trust's influence within the context of an empirical network. Cheap talk's role was ultimately re-confirmed. Study III highlights the importance of baseline trust in trust and cooperation development. We showed moderator level of influence would yields very diverse diffusion trajectories. Again, we demonstrated seeding trust and cooperation from the hubs could be effective and efficient. Moreover, we found the new seeding strategy (seeding from the distrustful), which becomes possible when considering baseline trust, is also effective and efficient.

Chapter 4, **5** and **6** provide answers to the three research questions of the research. For **RQ1** and **RQ2**, all three studies revealed that cheap talk over the internet positively influences trust and cooperation development among GSE practitioners, and explained why cheap talk over the internet helps the trust and cooperation development. For **RQ3**, study II and III developed basic understanding on the influences of network structures and individual characteristics through simulations over artificially-generated and empirical networks.

7.2 Implications

7.2.1 Implications to Theory

The research described in this dissertation offers significant theoretical contributions. Using game theory as the theoretical lens, our research takes three steps to establish an understanding of cheap talk over the internet and its role in developing trust and cooperation in global software engineering. The research contributes to an alternative view of cheap talk, since we understand it not merely as a symbolic externalization of social relationships, but also as a deliberate strategic behavior. Combining theoretical modeling with empirical evidence, our research offers the following key theoretical insights:

> *With cheap talk acting as a catalyst, trust emerges from and ensures cooperation. Cheap talk guarantees that loss in a failed "cooperation trial" can be offset by punishment/incentive enforcement, leading to increased willingness to cooperate. In this process, trust toward others, or more precisely, trust of others' rationality, eventually develops, and "cooperation" becomes the mainstream choice. The improvement of cooperation can be observed while trust development is implicit, but they are essentially the same process. Furthermore, the role of cheap talk is also significant when considering social network effects. When in-*

corporating baseline trust, cheap talk can become more than a catalyst, and serve as a stable choice for some team members. Specific seeding strategies could be used to promote trust and cooperation with the presence of cheap talk.

The research presented in this dissertation not only identifies the correlation between cheap talk over the Internet and the emergence of trust and cooperation in globally distributed software engineering teams, but also establishes causation between them by answering the "why" and "how" questions. For example, we explains why cheap talk is effective to jump-start cooperation, which has been repeatedly observed since a decade ago (Zheng et al., 2002; Aragon et al., 2009; Mitchell and Zigurs, 2009).

Our research could shed light on investigating collaboration and coordination behaviors in complex social technical systems through analytical modeling and simulation. Traditional research often only focuses on a few specific constructs, and does not necessarily integrate them together into a comprehensive view or readily offer insights for prescriptive purposes. Our model avoids these disadvantages by requiring minimal assumptions for integrating complex social theories, and by its capacity to study both individual and group level dynamics. The model itself is dynamic, which enables researchers to not only study a static snapshot of collaboration, but also a long-term, evolving picture of the collaboration process in both quantitative and qualitative terms.

7.2.2 Implications to Collaboration Management Practice

The results of this dissertation can be directly applied in global software engineering practices, particularly collaboration management. We introduce them in detail in each chapter. Here, we briefly summarized their key points.

- Cheap talk with remote collaborators over the internet may be encouraged in the team, especially in the early phases of collaboration. [Study I]

- Reducing and balancing the cost for cheap talk, and naturally integrating it into daily workflow would help develop trust and cooperation. [Study I]
- Seeding from the team members in the hub position will improve the effectiveness and efficiency of building the team's trust and cooperation. [Study II and Study III]
- Seeding from the team members with lowest baseline trust will improve the effectiveness and efficiency of building the team's trust and cooperation. [Study III]
- Designing a communication structure should avoid over-reliance on specific individuals in order to enable trust and cooperation diffusion [Study II and Study III]
- While developing shared team identity is important for trust development, understanding team member's personal differences is also important for managers, team leaders, and policymakers [Study III].

7.2.3 Implications to The Design of Collaboration Tools

By augmenting well-designed user interfaces and utilizing project data, the models and simulations have the potential to be the infrastructures of intelligent decision supporting tools for GSE practitioners. Many researchers have used various forms of data-mining to summarize and abstract information into digestible end-user tools. Most of these tools deal with problems such as finding the expertise, identify the dependency of the code, predicting software quality. However, it might be equal important to help practitioners figure out *how to make wise decisions in their interaction with other team members.* For GSE policymakers such as project managers and team leaders, they often want to implement strategies to enhance the collaboration in their teams. Evaluating these strategies before actual implementation is critical for their decisions. Without proper tools, they may make a judgement according to their intuition or relatively limited information.

The models and simulations provide a way to connect the data generated in GSE process with game theory analytics to provide insightful decision support to GSE practitioners. Theoretically grounded models will be integrated with real world information-rich environments to specify individual team member and groups' decision-making processes. With simulations, practitioners can navigate the possible solution space and explore different scenarios, matching potential decisions to the context of their collaborators, their team, and foreseeing the immediate and long-term consequences of their decisions. For example, a project manager may utilize her project data and try to assess a seeding strategy with the model and simulation developed in this research before putting effort on it. It would help reducing the unexpected cost of making wrong decisions.

7.2.4 Implications to Research Methods

The human and social aspects of software engineering are increasingly well studied in the software engineering research community. A large body of empirical literature has focused on many aspects of human interactions in software development, including coordination and trust. Empirical studies contribute rich real world practice evidence, but are relatively void of explicit theory. Although empirical studies greatly inform research, approaches that combine empirical studies and predictive theories have obvious advantages. When assumptions are clearly articulated, theory can provide predictive power and avoid costly "trial and error" decisions.

The research methods developed in this dissertation are innovative and have potentials to be applied in a broader research area in CSCW and SE. In particular, we argue that game theory provides an ideal analytical framework for developing theoretical knowledge. Software engineering processes consist of human decision-making activities, and developers must determine the proper strategies for supporting team member cooperation. Game theory has demonstrated its capacity for studying human decision-making processes and the long-term attributes of social systems, and continues to bring new perspectives to cooperation research (Nowak 2006). Via game theory model reasoning, we can deductively develop propositions characterizing real world phenomena, which provide generative causality and explanation (Cederman 2005).

Two studies (Study I and Study III) described in this dissertation explore two different possibilities of combining game theory modeling and simulation with empirical studies. Figure 7.2 summarizes the methods we applied in this paper.

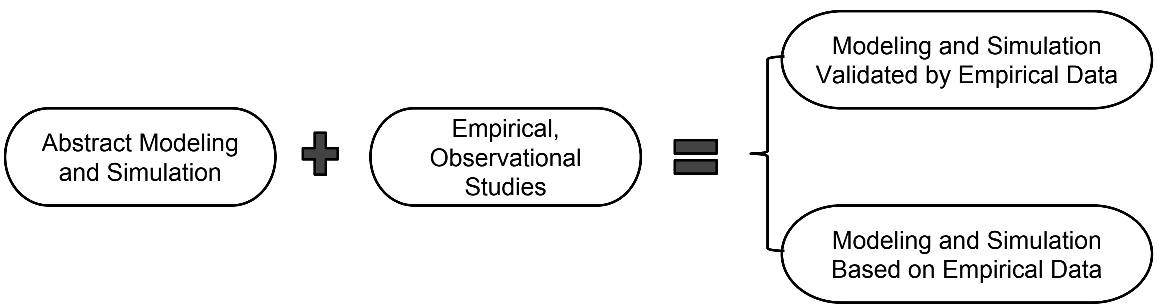

Figure 7.1: Two research approaches that integrates the abstract modeling and simulation with empirical, observational study.

Study I (chapter 4) first builds a theory model inspired by empirical evidence, then analyzes its dynamics theoretically. The theoretical propositions is examined by two empirical case studies (see the upper right part of figure 3). This combination can be used for theory development. The theoretical model and analysis is similar to "hypothesis development" while empirical study functions as "hypothesis testing". Together, they form a completed process of developing scientific explanations (Cooper, Schindler and Sun, 2006). Study III (chapter 6) explores another possibility for integrating abstract modeling and simulation with empirical methods, which serves as an alternative to simply using empirical results to motivate and validate theoretical modeling and simulation results. This integration (see the lower right part of figure 3) enables researchers to utilize both methods' advantages toward exploring a richer set of research problems, which are hard to investigate with pure empirical or abstract methods.

7.3 Limitations and Future Work

Collaboration in global software engineering practices is complex. High-quality collaboration in GSE practices depends on many social and technical factors (Olson and Olson, 2000; Bradner and Mark, 2002). In this research, we focus on one of these factors: trust. Although trust is a fundamental factor for enabling cooperation, there are still other important factors. For instance, individuals' adaptivity and learning (Al-Ani et al., 2012*a*; Erev and Roth, 2014), social identity (Stets and Burke, 2000; Watts, Dodds and Newman, 2002), intra- or inter-team signaling systems (Huttegger et al., 2014; Fowler and Christakis, 2010; Skyrms, 2010), social diversity (Santos, Santos and Pacheco, 2008), information exchange (Skyrms, 2014) and many other factors, all have significant influence on globally distributed collaboration. Moreover, how these factors interplay with trust to shape globally distributed collaborations is still largely unanswered, and these topics require a great deal of future research efforts.

In this dissertation, we discussed the social network structure and individual characteristics' influences on trust and cooperation diffusion. However, the social network we studied is static, and static networks only approximate real world social networks, which are essentially dynamic. For instance, Alice and Bob are friends and talk to each other frequently, but this does not guarantee they will be friends forever. Our research shows the relationships among global software engineering practitioners are often not stable (Al-Ani et al., 2013). Trust relationship may be damaged and repaired for many reasons (Vangen and Huxham, 2003). Therefore, adding dynamics to the network structure is necessary for future research. Study III shows that some individuals may significantly hinder the process of trust and cooperation diffusion. It is possible that individuals may create new connections with another set of nodes rather than stick to leveraging those "pertinacious" individuals to spread belief or behavior contagion. For instance, in figure 6.1, A may directly build relationships with C and D instead of trying to influence them through B. Investigating the unwire/rewire phenomena also require dynamic network structures. Literature such as (Skyrms and Pemantle, 2000; Jin, Girvan and Newman, 2001) provides theoretical foundations for guiding our future research in this direction. Furthermore, peer influence is also far more complex in the real world(Aral and Walker, 2014) and could be very dynamic (Pinheiro et al., 2014). The future research may wish to consider it.

Our research approaches are innovative for computer-supported collaborative work and software engineering areas. The generative nature of these approaches ensures they will achieve satisfiable levels of scientific rigor. At least for the three studies on trust and cooperation, these approaches exhibit reasonable levels of validity and we were able to draw relevant conclusions for real world application. It is reasonable to assume that the approaches are applicable to other similar human factors, such as social identity. However, like many other empirical or theoretical methods, the approaches utilized in this dissertation are not, and can never be, universal solutions for GSE collaboration research problems. The confidence of integrating abstract modeling and simulation with empirical methods would be inductively established if researchers (including ourselves) could continuously obtain reliable results through these approaches (Friedman, 1974) on a variety of research topics. Moreover, researchers who are interested in research methodology and epistemology may try to identify the conditions, scopes, and other issues related to applying this approach in studying various globally distributed collaborations.

Utilizing game theory analytics, this research opens possibilities for developing data-driven, algorithmic software tools that support the decision-making of GSE team members and project managers. Decision systems are able to help software developer make proper decisions. It goes beyond data mining which mainly focuses on identifying patterns (Menzies, 2013). In section 7.2.2, we outlined a few scenarios of using this type of tool. We are working on early prototypes. An exemplary prototype we have developed can suggest the proper way to interact with unfamiliar remote team members (Wang and Redmiles, 2015*a*). We are also trying to build decision support tools leveraging the method and results from study II and III to help managers and teams better organize team collaboration. We plan to further pursue this direction in the future, and evaluate these tools through controlled laboratory experiment and field deployment in real world GSE teams.

The ultimate goal of our research is to understand, theorize, and prescriptively improve collaboration in GSE teams. We believe studies of globally distributed collaboration research will greatly benefit from rigorous models that allow for analytical study, computer simulation, and, in particular, integrating them with empirical studies for better examining complex social-technical systems. Doing so will help to overcome the barriers between different disciplines, and inform organizational concerns as well as software tool development. It will undoubtedly help us to achieve the ultimate goal in future. In a broader sense, our approach and results could be applied in other socially meaningful areas after careful evaluation, for instance collaboration among multiple or diverse stakeholders in sustainable software engineering (Penzenstadler, Femmer and Richardson, 2013), or mass collaboration in citizen science (Bos et al., 2007; Wiggins and Crowston, 2010) .

Bibliography

Abrahamson, Eric, and Lori Rosenkopf. 1997. "Social network effects on the extent of innovation diffusion: A computer simulation." *Organization science*, 8(3): 289–309.

Al-Ani, Ban, and David Redmiles. 2009. "In strangers we trust? Findings of an empirical study of distributed teams." *Proceeding of the Fourth IEEE International Conference on Global Software Engineering*, 121–130.

Al-Ani, Ban, Erik Trainer, David Redmiles, and Erik Simmons. 2012*a*. "Trust and surprise in distributed teams: towards an understanding of expectations and adaptations." *Proceedings of the Fourth international conference on Intercultural Collaboration (ICIC)*, 97–106.

Al-Ani, Ban, Matthew J. Bietz, Yi Wang, Erik Trainer, Benjamin Koehne, Sabrina Marczak, David F. Redmiles, and Rafael Prikladnicki. 2013. "Globally distributed system developers: their trust expectations and processes." *Proceedings of the ACM Conference on Computer-Supported Colaborative Work and Social Computing (CSCW)*, 563–574.

Al-Ani, Ban, Yi Wang, Sabrina Marczak, Erik Trainer, and David Redmiles. 2012*b*. "Distributed Developers and the Non-use of Web 2.0 Technologies: A Proclivity Model." *Proceedings of the International Conference on Global Software Engineering*, 104–113.

Aragon, Cecilia R, Sarah S Poon, Andrés Monroy-Hernández, and Diana Aragon. 2009. "A tale of two online communities: Fostering collaboration and creativity in scientists and children." *Proceedings of the Seventh ACM conference on Creativity and cognition (C&C)*, 9–18.

Aral, Sinan. 2012. "Poked to vote." *Nature*, 489: 212–214.

Aral, Sinan, and Dylan Walker. 2014. "Tie strength, embeddedness, and social influence: A large-scale networked experiment." *Management Science*, 60(6): 1352–1370.

Aral, Sinan, Lev Muchnik, and Arun Sundararajan. 2013. "Engineering social contagions: Optimal network seeding in the presence of homophily." *Network Science*, 1: 125–153.

Bachrach, Yoram, Ariel Parnes, Ariel D Procaccia, and Jeffrey S Rosenschein. 2009. "Gossip-based aggregation of trust in decentralized reputation systems." *Autonomous Agents and Multi-Agent Systems*, 19(2): 153–172.

Banerjee, Abhijit, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. 2013. "The diffusion of microfinance." *Science*, 341(6144): 1236498.

Barabási, AL, E Ravasz, and T Vicsek. 2001. "Deterministic scale-free networks." *Physica A: Statistical Mechanics and its Applications*, 299: 559–564.

Barclay, Pat. 2004. "Trustworthiness and competitive altruism can also solve the ?tragedy of the commons?" *Evolution and Human Behavior*, 25(4): 209 – 220.

Basili, Victor R, Forrest Shull, and Filippo Lanubile. 1999. "Building knowledge through families of experiments." *Software Engineering, IEEE Transactions on*, 25(4): 456–473.

Bettenburg, Nicolas, Emad Shihab, and Ahmed E Hassan. 2009. "An empirical study on the risks of using off-the-shelf techniques for processing mailing list data." *Proceedings of IEEE International Conference on Software Maintenance (ICSM)*, 539–542.

Bird, Christian, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. 2006. "Mining email social networks." *Proceedings of the Mining Software Repositories (MSR)*, 137–143.

Bos, Nathan, Ann Zimmerman, Judith Olson, Jude Yew, Jason Yerkie, Erik Dahl, and Gary Olson. 2007. "From shared databases to communities of practice: A taxonomy of collaboratories." *Journal of Computer-Mediated Communication*, 12(2): 652–672.

Bos, Nathan, Judy Olson, Darren Gergle, Gary Olson, and Zach Wright. 2002. "Effects of Four Computer-mediated Communications Channels on Trust Development." *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 135–140.

Bowles, Samuel, and Herbert Gintis. 2011. *A cooperative species: Human reciprocity and its evolution.* Princeton University Press.

Bradner, Erin, and Gloria Mark. 2002. "Why distance matters: effects on cooperation, persuasion and deception." *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 226–235.

Calefato, Fabio, Filippo Lanubile, and Francesco Sportelli. 2013. "Can social awareness foster trust building in global software teams?" *Proceedings of the Social Software Engineering*, 13–16.

Calefato, Fabio, Filippo Lanubile, Nicola Sanitate, and Giuseppe Santoro. 2011. "Augmenting social awareness in a collaborative development environment." 39–42, ACM.

Camera, Gabriele, Marco Casari, and Maria Bigoni. 2013. "Money and trust among strangers." *Proceedings of the National Academy of Sciences*, 110(37): 14889–14893.

Cassell, Justine, and Timothy Bickmore. 2003. "Negotiated Collusion: Modeling Social Language and its Relationship Effects in Intelligent Agents." *User Modeling and User-Adapted Interaction*, 13(1-2): 89–132.

Cataldo, Marcelo, James D. Herbsleb, and Kathleen M. Carley. 2008. "Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity." *Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2–11.

Cederman, Lars Erik. 2005. "Computational Models of Social Forms: Advancing Generative Process Theory." *American Journal of Sociology*, 110(4): pp. 864–893.

Charness, Gary, and Martin Dufwenberg. 2006. "Promises and partnership." *Econometrica*, 74(6): 1579–1601.

Chua, Roy YJ, Michael W Morris, and Shira Mor. 2012. "Collaborating across cultures: Cultural metacognition and affect-based trust in creative collaboration." *Organizational Behavior and Human Decision Processes*, 118(2): 116–131.

Chung, Fan, and Linyuan Lu. 2004. "The average distance in a random graph with given expected degrees." *Internet Mathematics*, 1(1): 91–113.

Clauset, Aaron, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009*a*. "Power-law distributions in empirical data." *SIAM review*, 51(4): 661–703.

Clauset, Aaron, Cosma Rohilla Shalizi, and M. E. J. Newman. 2009*b*. "Power-Law Distributions in Empirical Data." *SIAM Review*, 51(4): 661–703.

Cooper, Donald R, Pamela S Schindler, and Jianmin Sun. 2006. *Business research methods.* Vol. 9, McGraw-hill New York.

Corritore, Cynthia L, Beverly Kracher, and Susan Wiedenbeck. 2003. "On-line trust: concepts, evolving themes, a model." *International Journal of Human-Computer Studies*, 58(6): 737–758.

Cramton, Catherine D, and Pamela J. Hinds. 2007. "Intercultural Interaction in Distributed Teams: Salience of and Adaptations to Cultural Differences." *Proceedings of the AOM Annual Meeting*, 1 – 6.

Crowston, Kevin, Kangning Wei, James Howison, and Andrea Wiggins. 2012. "Free/Libre open-source software development: What we know and what we do not know." *ACM Computing Surveys (CSUR)*, 44(2): 7.

Dabbish, Laura, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. "Social coding in GitHub: transparency and collaboration in an open software repository." *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 1277–1286.

Damian, Daniela, Luis Izquierdo, Janice Singer, and Irwin Kwan. 2007. “Awareness in the wild: Why communication breakdowns occur.” *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, 81–90.

Das, Tarun K, and Bing-Sheng Teng. 1998. “Between trust and control: Developing confidence in partner cooperation in alliances.” *Academy of management review*, 23(3): 491–512.

Datta, Subhajit, Renuka Sindhgatta, and Bikram Sengupta. 2012. “Talk versus work: characteristics of developer collaboration on the jazz platform.” *Proceedings of the Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)*, 655–668.

Delhey, Jan, Kenneth Newton, and Christian Welzel. 2011. “How general is trust in “most people” Solving the radius of trust problem.” *American Sociological Review*, 76(5): 786–807.

DiMaggio, Paul, and Filiz Garip. 2012. “Network effects and social inequality.” *Annual Review of Sociology*, 38: 93–118.

Dittrich, Y., and R. Giuffrida. 2011. “Exploring the Role of Instant Messaging in a Global Software Development Project.” *Proceedings of the International Conferences on Global Software Engineering*, 103–112.

Driscoll, James W. 1978. “Trust and participation in organizational decision making as predictors of satisfaction.” *Academy of management journal*, 21(1): 44–56.

Ducheneaut, Nicolas. 2005. “Socialization in an Open Source Software Community: A Socio-Technical Analysis.” *Computer Supported Cooperative Work (CSCW)*, 14(4): 323–368.

Easley, David, and Jon Kleinberg. 2010. *Networks, crowds, and markets: Reasoning about a highly connected world.* Cambridge University Press.

Easterbrook, Steve, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. “Selecting empirical methods for software engineering research.” In *Guide to advanced empirical software engineering.* 285–311. Springer.

Ehrlich, Kate, and Marcelo Cataldo. 2012. “All-for-one and one-for-all?: a multi-level analysis of communication patterns and individual performance in geographically distributed software development.” *Proceedings of the ACM Conferences on Computer Supported Cooperative Work and Social Computing (CSCW)*, 945–954.

Ellison, Glenn. 2000. “Basins of attraction, long-run stochastic stability, and the speed of step-by-step evolution.” *The Review of Economic Studies*, 67(1): 17–45.

Erev, Ido, and Alvin E Roth. 2014. “Maximization, learning, and economic behavior.” *Proceedings of the National Academy of Sciences*, 111(Supplement 3): 10818–10825.

Etzion, Dror. 2014. “Diffusion as Classification.” *Organization Science*, 25(2): 420–437.

Farooq, U., C.H. Ganoe, J.M. Carroll, and C.L. Giles. 2007. "Supporting distributed scientific collaboration: Implications for designing the CiteSeer collaboratory." *Proceedings of the HICSS*, 26–26.

Farrell, Joseph, and Matthew Rabin. 1996. "Cheap Talk." *The Journal of Economic Perspectives*, 10(3): 103–118.

Fehr, Ernst, Urs Fischbacher, and Simon Gächter. 2002. "Strong reciprocity, human cooperation, and the enforcement of social norms." *Human nature*, 13(1): 1–25.

Feltovich, Nick. 2000. "Reinforcement-based vs. Belief-based Learning Models in Experimental Asymmetric-information Games." *Econometrica*, 68(3): 605–641.

Fowler, James H, and Nicholas A Christakis. 2010. "Cooperative behavior cascades in human social networks." *Proceedings of the National Academy of Sciences*, 107(12): 5334–5338.

Friedman, Eric, Paul Resnick, and Rahul Sami. 2007. "Manipulation-resistant reputation systems." *Algorithmic Game Theory*, 677–697.

Friedman, Michael. 1974. "Explanation and scientific understanding." *The Journal of Philosophy*, 5–19.

Fudenberg, Drew, and Lorens A. Imhof. 2006. "Imitation processes with small mutations." *Journal of Economic Theory*, 131(1): 251 – 262.

Fudenberg, Drew, and Lorens A. Imhof. 2008. "Monotone imitation dynamics in large populations." *Journal of Economic Theory*, 140(1): 229 – 245.

Gao, Ge, Pamela Hinds, and Chen Zhao. 2013. "Closure vs. Structural Holes: How Social Network Information and Culture Affect Choice of Collaborators." *Proceedings of the ACM Conference on Computer-Supported Collaborative Work and Social Computing (CSCW)*, 5–18.

Gharehyazie, Mohammad, Daryl Posnett, Bogdan Vasilescu, and Vladimir Filkov. 2014. "Developer initiation and social interactions in OSS: A case study of the Apache Software Foundation." *Empirical Software Engineering*, 1–36.

Gintis, Herbert. 2000. *Game theory evolving: A problem-centered introduction to modeling strategic behavior.* Princeton University Press.

Gintis, Herbert. 2014. *The bounds of reason: game theory and the unification of the behavioral sciences.* Princeton University Press.

Giuffrida, Rosalba, and Yvonne Dittrich. 2015. "A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams." *Information and Software Technology.*

Goggins, Sean, Christopher Mascaro, and Stephanie Mascaro. 2012. "Relief Work After the 2010 Haiti Earthquake: Leadership in an Online Resource Coordination Network." *Proceedings of the ACM Conference on Computer-Supported Colaborative Work and Social Computing (CSCW)*, 57–66.

Goh, Kwang-Il, Eulsik Oh, Hawoong Jeong, Byungnam Kahng, and Doochul Kim. 2002. "Classification of scale-free networks." *Proceedings of the National Academy of Sciences*, 99(20): 12583–12588.

Gou, Liang, Michelle X. Zhou, and Huahai Yang. 2014. "KnowMe and ShareMe: Understanding Automatically Discovered Personality Traits from Social Media and User Sharing Preferences." *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 955–964.

Guzzi, Anja, Alberto Bacchelli, Michele Lanza, Martin Pinzger, and Arie van Deursen. 2013. "Communication in Open Source Software Development Mailing Lists." *Proceedings of Mining Software Repositories (MSR)*, 277–286.

Hansen, DerekL., and MarcA. Smith. 2014. "Social Network Analysis in HCI." In *Ways of Knowing in HCI.* , ed. Judith S. Olson and Wendy A. Kellogg, 421–447. Springer New York.

Harper, Richard, Christian Bird, Thomas Zimmermann, and Brendan Murphy. 2013. "Dwelling in Software: Aspects of the felt-life of engineers in large software projects." 163–180, Springer.

Harrison, J Richard, Zhiang Lin, Glenn R Carroll, and Kathleen M Carley. 2007. "Simulation modeling in organizational and management research." *Academy of Management Review*, 32(4): 1229–1245.

Herbsleb, James D., and Audris Mockus. 2003. "An empirical study of speed and communication in globally distributed software development." *Software Engineering, IEEE Transactions on*, 29(6): 481–494.

Herbsleb, James D., and Rebecca E. Grinter. 1999. "Splitting the organization and integrating the code: Conway's law revisited." 85–95. ACM.

Herbsleb, James D., David L. Atkins, David G. Boyer, Mark Handel, and Thomas A. Finholt. 2002. "Introducing instant messaging and chat in the workplace." *Proceedings of the ACM Conferences on Human Factors in Computing Systems (CHI)*, 171–178.

Hong, Qiaona, Sunghun Kim, S. C. Cheung, and Christian Bird. 2011. "Understanding a Developer Social Network and Its Evolution." *ICSM '11*, 323–332. Washington, DC, USA:IEEE Computer Society.

Hosmer, Larue Tone. 1995. "Trust: The connecting link between organizational theory and philosophical ethics." *Academy of management Review*, 20(2): 379–403.

Huttegger, Simon, Brian Skyrms, Pierre Tarrès, and Elliott Wagner. 2014. "Some dynamics of signaling games." *Proceedings of the National Academy of Sciences*, 111(Supplement 3): 10873–10880.

Jackson, Matt. 2010. *Social and Economic Network.* Princeton University Press.

Jackson, Matthew O, Tomas Rodriguez-Barraquer, and Xu Tan. 2012. "Social capital and social quilts: Network patterns of favor exchange." *The American Economic Review*, 102(5): 1857–1897.

Jackson, M.O., and Y. Zenou. 2013. *Economic Analyses of Social Networks. International library of critical writings in economics*, Elgar, Edward Publishing.

Jarvenpaa, Sirkka L, and Dorothy E Leidner. 1998. "Communication and trust in global virtual teams." *Journal of Computer-Mediated Communication*, 3(4).

Jarvenpaa, Sirkka L, Thomas R Shaw, and D Sandy Staples. 2004. "Toward contextualized theories of trust: The role of trust in global virtual teams." *Information systems research*, 15(3): 250–267.

Jensen, C, and G Riestenberg. 2012. "Stag-hunt Game." *http://goo.gl/ieiLmX*, [Online; accessed 12-12-2014].

Jin, Emily M, Michelle Girvan, and Mark EJ Newman. 2001. "Structure of growing social networks." *Physical review E*, 64(4): 046132.

Jones, Gareth R, and Jennifer M George. 1998. "The experience and evolution of trust: Implications for cooperation and teamwork." *Academy of management review*, 23(3): 531–546.

Kahneman, Daniel. 2003. "Maps of Bounded Rationality: Psychology for Behavioral Economics." *American Economic Review*, 93(5): 1449–1475.

Kanavos, A., I. Perikos, P. Vikatos, I. Hatzilygeroudis, C. Makris, and A. Tsakalidis. 2014. "Conversation Emotional Modeling in Social Networks." *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, 478–484.

Kandori, Michihiro, George J Mailath, and Rafael Rob. 1993. "Learning, mutation, and long run equilibria in games." *Econometrica: Journal of the Econometric Society*, 29–56.

Keegan, Brian, Darren Gergle, and Noshir Contractor. 2012*a*. "Do Editors or Articles Drive Collaboration?: Multilevel Statistical Network Analysis of Wikipedia Coauthorship." *Proceedings of the ACM Conference on Computer-Supported Collaborative Work and Social Computing (CSCW)*, 427–436.

Keegan, Brian, Darren Gergle, and Noshir Contractor. 2012*b*. "Staying in the Loop: Structure and Dynamics of Wikipedia's Breaking News Collaborations." *Proceedings of the WikiSym*, 1:1–1:10.

Kempter, Renato, Valentina Sintsova, Claudiu Musat, and Pearl Pu. 2014. "EmotionWatch: Visualizing Fine-Grained Emotions in Event-Related Tweets." *Eighth International AAAI Conference on Weblogs and Social Media.*

Kitchenham, Barbara, Hiyam Al-Khilidar, Muhammed Ali Babar, Mike Berry, Karl Cox, Jacky Keung, Felicia Kurniawati, Mark Staples, He Zhang, and Liming Zhu. 2008. "Evaluating guidelines for reporting empirical software engineering studies." *Empirical Software Engineering*, 13(1): 97–121.

Kleinberg, Jon. 2007. "Cascading behavior in networks: Algorithmic and economic issues." *Algorithmic Game Theory*, 24: 613–632.

Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. 2011. "Twitter sentiment analysis: The good the bad and the omg!" *ICWSM*, 11: 538–541.

Kreindler, Gabriel E, and H Peyton Young. 2014. "Rapid innovation diffusion in social networks." *Proceedings of the National Academy of Sciences*, 111(Supplement 3): 10881–10888.

Kwan, I., M. Cataldo, and D. Damian. 2012. "Conway's Law Revisited: The Evidence for a Task-Based Perspective." *Software, IEEE*, 29(1): 90–93.

Layman, Lucas, Laurie Williams, Daniela Damian, and Hynek Bures. 2006. "Essential communication practices for Extreme Programming in a global software development team." *Information and software technology*, 48(9): 781–794.

Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg. 2010. "Signed Networks in Social Media." *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 1361–1370.

López-Pintado, Dunia. 2006. "Contagion and coordination in random networks." *International Journal of Game Theory*, 34(3): 371–381.

Macal, Charles M, and Michael J North. 2010. "Tutorial on agent-based modelling and simulation." *Journal of simulation*, 4(3): 151–162.

Manning, Christopher D, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval.* Vol. 1, Cambridge university press Cambridge.

March, James G, and Herbert Alexander Simon. 1993. *Organizations. Second Edition.* Wiley.

Mark, Gloria, and Steven Poltrock. 2001. "Diffusion of a Collaborative Technology Cross Distance." *Proceedings of the ACM Conference on Supporting Groupwork (Group)*, 232–241.

Marlow, Jennifer, and Laura Dabbish. 2013. "Activity traces and signals in software developer recruitment and hiring." *Proceedings of the ACM conference on Computer supported cooperative work (CSCW)*, 145–156.

Marlow, Jennifer, Laura Dabbish, and Jim Herbsleb. 2013. "Impression formation in online peer production: activity traces and personal profiles in github." 117–128, ACM.

Matthiesen, Stina, Pernille Bjørn, and Lise Møller Petersen. 2014. "Figure out how to code with the hands of others: Recognizing cultural blind spots in global software development." *Proceedings of the ACM conference on Computer supported cooperative work & social computing (CSCW)*, 1107–1119.

McAllister, Daniel J. 1995. "Affect-and cognition-based trust as foundations for interpersonal cooperation in organizations." *Academy of management journal*, 38(1): 24–59.

McDonald, David W. 2003. "Recommending Collaboration with Social Networks: A Comparative Evaluation." *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 593–600.

McPherson, Miller, Lynn Smith-Lovin, and James M Cook. 2001. "Birds of a feather: Homophily in social networks." *Annual review of sociology*, 415–444.

Menzies, Tim. 2013. "Beyond data mining; towards idea engineering." *Proceedings of the 9th International Conference on Predictive Models in Software Engineering*, 11.

Mislin, Alexandra A., Rachel L. Campagna, and William P. Bottom. 2011. "After the deal: Talk, trust building and the implementation of negotiated agreements." *Organizational Behavior and Human Decision Processes*, 115(1): 55 – 68.

Mitchell, Alanah, and Ilze Zigurs. 2009. "Trust in virtual teams: solved or still a mystery?" *ACM SIGMIS Database*, 40(3): 61–83.

Moe, Nils Brede, and Darja Šmite. 2008. "Understanding a lack of trust in Global Software Teams: a multiple-case study." *Software Process: Improvement and Practice*, 13(3): 217–231.

Mohammad, Saif M., and Peter D. Turney. 2013. "Crowdsourcing a Word-Emotion Association Lexicon." *Computational Intelligence*, 29(3): 436–465.

Molloy, Michael, and Bruce Reed. 1995. "A critical point for random graphs with a given degree sequence." *Random structures & algorithms*, 6(2-3): 161–180.

Mortensen, Mark, and Tsedal B Neeley. 2012. "Reflected knowledge and trust in global collaboration." *Management Science*, 58(12): 2207–2224.

Nowak, M. A. 2006. *Evolutionary Dynamic: Exploring the Equations of Life.* The Belknap Press of Harvard University.

Nowak, M.A. 2013. *Evolution, Games, and God.* Harvard University Press.

Nowak, Martin A., Akira Sasaki, Christine Taylor, and Drew Fudenberg. 2004. "Emergence of cooperation and evolutionary stability in finite populations." *Nature*, 428(6983): 646–650.

Nowak, Martin A, Corina E Tarnita, and Edward O Wilson. 2010. "The evolution of eusociality." *Nature*, 466: 1057–1062.

Olson, Gary M, and Judith S Olson. 2000. "Distance matters." *Human-computer interaction*, 15(2): 139–178.

Olson, Judith S, and Gary M Olson. 2013. "Working together apart: Collaboration over the internet." *Synthesis Lectures on Human-Centered Informatics*, 6(5): 1–151.

Olson, Judith S., and Wendy A. Kellogg, ed. 2014. *Ways of Knowing in HCI.* Springer New York.

Pang, Bo, and Lillian Lee. 2008. "Opinion mining and sentiment analysis." *Foundations and trends in information retrieval*, 2(1-2): 1–135.

Pendharkar, Parag C., and James A. Rodger. 2009. "The Relationship Between Software Development Team Size and Software Development Cost." *Commun. ACM*, 52(1): 141–144.

Pennebaker, James W, Roger J Booth, and Martha E Francis. 2007. "Linguistic inquiry and word count: LIWC 2007." *LIWC.net, Published Online*, 1–10.

Pentland, Alex, and Tracy Heibeck. 2008. *Honest signals.* MIT press Cambridge, MA.

Penzenstadler, Birgit, Henning Femmer, and Debra Richardson. 2013. "Who is the advocate? stakeholders for sustainability." *Green and Sustainable Software (GREENS), 2013 2nd International Workshop on*, 70–77.

Pfaff, Bernhard. 2008. *Analysis of Integrated and Cointegrated Time Series with R.* Springer.

Pinheiro, Flavio L, Marta D Santos, Francisco C Santos, and Jorge M Pacheco. 2014. "Origin of peer influence in social networks." *Physical review letters*, 112(9): 098702.

Prikladnicki, Rafael, Alexander Boden, Gabriela Avram, Cleidson RB de Souza, and Volker Wulf. 2014. "Data collection in global software engineering research: learning from past experience." *Empirical Software Engineering*, 19(4): 822–856.

Puranam, Phanish, Oliver Alexy, and Markus Reitzig. 2013. "What's" New" about New Forms of Organizing?" *Academy of Management Review*, amr–2011.

R Development Core Team. 2008. "R: A Language and Environment for Statistical Computing." Vienna, Austria, R Foundation for Statistical Computing, ISBN 3-900051-07-0.

Ren, Yuqing, and Robert E. Kraut. 2014*a*. "Agent-Based Modeling to Inform Online Community Design: Impact of Topical Breadth, Message Volume, and Discussion Moderation on Member Commitment and Contribution." *Human-Computer Interaction*, 29(4): 351–389.

Ren, Yuqing, and Robert E. Kraut. 2014*b*. "Agent-Based Modeling to Inform Online Community Design: Impact of Topical Breadth, Message Volume, and Discussion Moderation on Member Commitment and Contribution." *Human-Computer Interaction*, 29(4): 351–389.

Ren, Yuqing, and RobertE. Kraut. 2014*c*. "Agent Based Modeling to Inform the Design of Multiuser Systems." In *Ways of Knowing in HCI.* , ed. Judith S. Olson and Wendy A. Kellogg, 395–419. Springer New York.

Riegelsberger, Jens, M Angela Sasse, and John D McCarthy. 2003. "The researcher's dilemma: evaluating trust in computer-mediated communication." *International Journal of Human-Computer Studies*, 58(6): 759–781.

Rossman, Gabriel. 2012. *Climbing the charts: What radio airplay tells us about the diffusion of innovation.* Princeton University Press.

Rossman, Gabriel, Ming Ming Chiu, and Joeri M Mol. 2008. "Modeling diffusion of multiple innovations via multilevel diffusion curves: Payola in pop music radio." *Sociological Methodology*, 38(1): 201–230.

Rotter, Julian B. 1967. "A new scale for the measurement of interpersonal trust." *Journal of Personality*, 35(4): 651–665.

Rousseau, Denise M, Sim B Sitkin, Ronald S Burt, and Colin Camerer. 1998. "Not so different after all: A cross-discipline view of trust." *Academy of management review*, 23(3): 393–404.

Rousseau, Jean Jacques. 1950. "A Dissertation on the Origin and Foundation of the Inequality of Mankind." *The social contract and discourses (G.D.H Cole, Trans.). EP Dutton, New York*, 196–282.

Runeson, Per, and Martin Hst. 2009. "Guidelines for conducting and reporting case study research in software engineering." *Empirical Software Engineering*, 14(2): 131–164.

Santos, Francisco C, and Jorge M Pacheco. 2005. "Scale-free networks provide a unifying framework for the emergence of cooperation." *Physical Review Letters*, 95(9): 098104.

Santos, Francisco C., Jorge M. Pacheco, and Brian Skyrms. 2011. "Co-evolution of pre-play signaling and cooperation." *Journal of Theoretical Biology*, 274(1): 30 – 35.

Santos, Francisco C, Marta D Santos, and Jorge M Pacheco. 2008. "Social diversity promotes the emergence of cooperation in public goods games." *Nature*, 454(7201): 213–216.

Schumann, Jana, Patrick C. Shih, David F. Redmiles, and Graham Horton. 2012. "Supporting Initial Trust in Distributed Idea Generation and Idea Evaluation." 199–208.

Shen, Jianqiang, Oliver Brdiczka, and Juan Liu. 2013. "Understanding email writers: Personality prediction from email messages." In *User Modeling, Adaptation, and Personalization*. 318–330. Springer.

Skyrms, Brian. 2001. "The Stag hunt." *Presendential Address of the Pacific Division of the APA, Proceedings and Addresses of the APA 75*, 31–41.

Skyrms, Brian. 2004. *The stag hunt and the evolution of social structure.* Cambridge University Press.

Skyrms, Brian. 2005. "Dynamics of conformist bias." *The Monist*, 88(2): 260–269.

Skyrms, Brian. 2008. "Trust, risk, and the social contract." *Synthese*, 160(1): 21–25.

Skyrms, Brian. 2010. *Signals: Evolution, learning, and information.* Oxford University Press.

Skyrms, Brian. 2014. *Social Dynamics.* Oxford University Press.

Skyrms, Brian, and Robin Pemantle. 2000. "A Dynamic Model of Social Network Formation." *Proceedings of the National Academy of Science*, 97: 9340–9346.

Skyrms, Brian, John C Avise, and Francisco J Ayala. 2014. "In the light of evolution VIII: Darwinian thinking in the social sciences." *Proceedings of the National Academy of Sciences*, 111(Supplement 3): 10781–10784.

Steed, Anthony, Maria Spante, Ilona Heldal, Ann-Sofie Axelsson, and Ralph Schroeder. 2003. "Strangers and friends in caves: an exploratory study of collaboration in networked IPT systems for extended periods of time." *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 51–54.

Stets, Jan E, and Peter J Burke. 2000. "Identity theory and social identity theory." *Social psychology quarterly*, 224–237.

Storey, Margaret-Anne, Leif Singer, Brendan Cleary, Fernando Figueira Filho, and Alexey Zagalsky. 2014. "The (R) Evolution of social media in software engineering." 100–116, ACM.

Sturgis, Patrick, and Patten Smith. 2010. "Assessing the validity of generalized trust questions: What kind of trust are we measuring?" *International Journal of Public Opinion Research*, 22(1): 74–92.

Sugden, Robert. 2000. "Credible worlds: the status of theoretical models in economics." *Journal of Economic Methodology*, 7(1): 1–31.

Sundararajan, Arun, Foster Provost, Gal Oestreicher-singer, and Sinan Aral. 2013. "Information in Digital , Economic , and Social Networks." *Information Systems Research*, 24(4): 883–905.

Suri, Siddharth, and Duncan J Watts. 2011. "Cooperation and contagion in web-based, networked public goods experiments." *PloS one*, 6(3): e16836.

Szabó, György, and Gabor Fath. 2007. "Evolutionary games on graphs." *Physics Reports*, 446(4): 97–216.

Tausczik, Yla R, and James W Pennebaker. 2010. "The psychological meaning of words: LIWC and computerized text analysis methods." *Journal of language and social psychology*, 29(1): 24–54.

Taylor, Sean J, Eytan Bakshy, and Sinan Aral. 2013. "Selection effects in online sharing: Consequences for peer adoption." *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 821–836.

Toral, SL, MR Martínez-Torres, and Federico Barrero. 2010. "Analysis of virtual communities supporting OSS projects using social network analysis." *Information and Software Technology*, 52(3): 296–303.

Trainer, Erik H, and David Redmiles. 2012. "Foundations for the design of visualizations that support trust in distributed teams." *Proceedings of the the International Working Conference on Advanced Visual Interfaces (AVI)*, 34–41.

Train, Kenneth E. 2009. *Discrete choice methods with simulation.* Cambridge university press.

Traulsen, Arne, Martin A. Nowak, and Jorge M. Pacheco. 2006. "Stochastic dynamics of invasion and fixation." *Phys. Rev. E*, 74: 011909.

Ugrin, Joseph C, and J Michael Pearson. 2013. "The effects of sanctions and stigmas on cyberloafing." *Computers in Human Behavior*, 29(3): 812–820.

Vangen, Siv, and Chris Huxham. 2003. "Nurturing collaborative relations Building trust in interorganizational collaboration." *The Journal of Applied Behavioral Science*, 39(1): 5–31.

Vitak, Jessica, Julia Crouse, and Robert LaRose. 2011. "Personal Internet use at work: Understanding cyberslacking." *Computers in Human Behavior*, 27(5): 1751–1759.

Von Neumann, John, and Oskar Morgenstern. 2007. *Theory of games and economic behavior (60th Anniversary Commemorative Edition).* Princeton university press.

Wagstrom, Patrick. 2009. "Vertical Interaction in Open Software Engineering Communities." PhD diss. Carnegie Mellon University.

Wagstrom, Patrick, and Subhajit Datta. 2014. "Does latitude hurt while longitude kills? geographical and temporal separation in a large scale software development project." *Proceedings of the International Conference on Software Engineering (ICSE)*, 199–210.

Wagstrom, Patrick, James D Herbsleb, and Kathleen M Carley. 2010. "Communication, Team Performance, and The Individual: Bringing Technical Dependencies." Vol. 2010, 1–7, Academy of Management.

Wang, Yi, and David Redmiles. 2015*a*. "Knowing how to interact with unfamiliar remote collaborators." *Submitted to the 30th IEEE/ACM International Conference on Automated Software Engineering (ASE'15).*

Wang, Yi, and David Redmiles. 2015*b*. "Trust and Coordination in Software Development: An Extensible Evolutionary Model." *Working Paper.*

Wang, Yi, and D. Redmiles. 2013. "Understanding Cheap Talk and the Emergence of Trust in Global Software Engineering: An Evolutionary Game Theory Perspective." *Proceedings of the CHASE*, 149–152, © IEEE, **Reprinted, with permission**, from [Wang, Yi, and D. Redmiles, "Understanding Cheap Talk and the Emergence of Trust in Global Software Engineering: An Evolutionary Game Theory Perspective." Proceedings of the CHASE, 149–152, 2013].

Watts, Duncan J, Peter Sheridan Dodds, and Mark EJ Newman. 2002. "Identity and search in social networks." *science*, 296(5571): 1302–1305.

Wiggins, Andrea, and Kevin Crowston. 2010. "Distributed scientific collaboration: research opportunities in citizen science." *CSCW 2010 workshop on The Changing Dynamics of Scientific Collaboration.*

Wolf, Timo, Adrian Schroter, Daniela Damian, Lucas D Panjer, and Thanh HD Nguyen. 2009. "Mining task-based social networks to explore collaboration in software teams." *Software, IEEE*, 26(1): 58–66.

Xiao, Zhixing, and Anne S Tsui. 2007. "When brokers may not work: The cultural contingency of social capital in Chinese high-tech firms." *Administrative Science Quarterly*, 52(1): 1–31.

Yamagishi, Toshio, and Midori Yamagishi. 1994. "Trust and commitment in the United States and Japan." *Motivation and Emotion*, 18(2): 129–166.

Yang, Jiang, Lada A Adamic, and Mark S Ackerman. 2008. "Competing to Share Expertise: The Taskcn Knowledge Sharing Community." *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM).*

Yang, Jiang, Zhen Wen, Lada A Adamic, Mark S Ackerman, and Ching-Yung Lin. 2011. "Collaborating Globally: Culture and Organizational Computer-Mediated Communications." *Proceedings of the Internation Conference on Information Systems.*

Young, H. Peyton. 1998. *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions.* Princeton University Press.

Young, H Peyton. 2001. *Individual strategy and social structure: An evolutionary theory of institutions.* Princeton University Press.

Young, H Peyton. 2011. "The dynamics of social innovation." *Proceedings of the National Academy of Sciences*, 108(Supplement 4): 21285–21291.

Zheng, Jun, Elizabeth Veinott, Nathan Bos, Judith S. Olson, and Gary M. Olson. 2002. "Trust Without Touch: Jumpstarting Long-distance Trust with Initial Social Activities." *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 141–146.

Zolin, Roxanne, Pamela J Hinds, Renate Fruchter, and Raymond E Levitt. 2004. "Interpersonal trust in cross-functional, geographically distributed work: A longitudinal study." *Information and Organization*, 14(1): 1–26.

Appendix A

Dictionaries for IRC Message Classification in Study I

A.1 Lucene

No. Total Keywords = 66.

information retrieval, ranking, index caching, caches, bounded indices, language detecting(detection), token(ize), word break, Kuromoji, schema, encoding scheme, highlighter, distance measure(s), nightly, bug, issue, compatibility, configuration, JIRA, contention, build #[1], parse(r), test(s), benchmark, trunk, commit, merge policy, error, exception, regenerate, initialized, branch, intermittent, truncated, subprocess, inter-process, manifest file, real(-)time, load-sharing, load-balancing, proxy, main memory, concurrent, enumerate, logging, Java, boolean, comparator, decouple, synchronization, sequentialization, invalid, memory leak, JFlex, FNFE, parameters, character escaping, concatenation, stacked segments, dis-

[1] The word "build" is not necessary to be a keyword for detecting work-related message, but "build #" always refers a build failure.

crepancy(ies), lexicon, unstored, vector, fuzzy query, collection api, wildcard.

A.2 Chromium-OS

No. Total Keywords = 118.

A.2.1 Reused Keywords

bug, issue, compatibility, configuration, JIRA, contention, build #, parse(r), test(s), benchmark, trunk, commit, merge policy, error, exception, regenerate, initialized, branch, intermittent, truncated, subprocess, inter-process, manifest file, real(-)time, load-sharing, load-balancing, proxy, main memory, concurrent, enumerate, logging, Java, boolean, comparator, decouple, synchronization, sequentialization, invalid, memory leak, parameters, concatenation, stacked segments, wildcard.

A.2.2 New Keywords

keyboard layout, tree status, gizmo-paladin, boot, firmware, register, kernel, cmdline, native android, GNU/Linux distro, Valgrind test, vanilla linux, autotest, InitSDK, ChromeSDK, parallel, merge conflict(s), chroot, sysroot, syslinux, config param, libdevmapper, librpm, crbug.com, CQ, CL[2], git push, patch(es), deprecated, clang, x86-generic, amd64-generic, module-init, setup_board, resynced, driver, partition, mount, hash, SyncChrome, canary(ies), gpu, symbolizing, prefix, 0x...[3], buffer, userspace, device_tree, KVM, UI image(s), simplechrome, build_image, test image, dependencies, waterfall, code-review, developer(dev)

[2] CQ and CL must be capitalized.

[3] This refers any word start with "0x" which indicates a hexadecimal number. Most of them are used to represent a memory address.

mode, recovery mode, hard-reset, interpreter, use case(s), modeset, dev(-)server, xserver, flash image, ARM, libxml, rootfs, tryserver, Ubuntu, IO[4].

[4]IO must be a separated word when considering as a keyword.