Simulation Study of Knowledge Supply-Demand Systems in Collaborative Innovation (Postprint)

Authors: Wu Jiang, Chen Jun, Zhang Jingfan

Date: 2017-11-08T00:00:00+00:00

Abstract

Objective: To investigate the impact of interactions among knowledge-based teams on team performance in a collaborative innovation environment. Methods: Using multi-agent modeling and simulation methods, a knowledge supply-demand system is constructed from the micro-level perspective of knowledge management, with time cost and financial cost serving as evaluation metrics for work performance, and the system is implemented based on Python NetworkX. Results: Large-scale organizations have an advantage over small-scale organizations in reducing innovation costs; organizations with scale-free structures take longer to complete tasks and incur higher costs; increasing the connection neighborhood size of individuals in an organization does not monotonically improve the organization' s innovation efficiency, and innovation costs begin to increase when the average neighborhood size exceeds a certain threshold. Limitations: The optimization settings of human-to-human interactions in collaborative innovation were not considered. Conclusion: The knowledge supply-demand system based on multi-agent modeling simulates the knowledge integration process of knowledge-based teams from a micro-level perspective, which helps to understand the management of internal team knowledge and provides a new perspective for organizations to improve knowledge utilization efficiency and reduce innovation costs.

Full Text

Abstract

This study investigates how interactions within knowledge-based teams influence team performance in collaborative innovation environments. Using a multi-agent modeling and simulation approach, we constructed a Knowledge Supply & Demand System (KSDS) from a micro-level knowledge management perspective, with time cost and financial cost serving as evaluation metrics for work performance. The system was implemented using Python NetworkX. Our results demonstrate that large organizations possess advantages over smaller ones in reducing innovation costs. Scale-free network structures require longer completion times and incur higher costs. Increasing individuals' connection neighborhoods does not monotonically improve organizational innovation efficiency; once the average degree exceeds a certain threshold, innovation costs begin to rise. A limitation of this study is that it does not consider optimal configurations of human interaction in collaborative innovation. In conclusion, the multi-agent-based KSDS simulates the knowledge integration process of knowledge-based teams at the micro level, offering new insights for understanding internal team knowledge management and providing organizations with fresh perspectives on improving knowledge utilization efficiency and reducing innovation costs.

Keywords: Collaborative innovation; Knowledge Supply & Demand System; Multi-agent simulation; Complex networks; Computational experiments

Introduction

In the mobile internet era, service innovations delivered through mobile applications emerge continuously. As the number of apps grows rapidly, organizations must respond to market demands within increasingly short timeframes. To gain competitive advantages, knowledge-based teams should fully mobilize both internal and external resources while effectively integrating individuals with diverse knowledge backgrounds to reduce innovation time and financial costs. Achieving high-level collaborative innovation requires in-depth investigation.

This paper employs multi-agent simulation methods to model, at the micro level of knowledge management, the matching process between knowledge suppliers and demanders within organizations and the knowledge utilization during project implementation. Through computer simulation, we implement a knowledge supply-demand system that captures human-project interactions, embodying a cyclical and dynamic organizational collaborative process: "matching individuals and projects on the knowledge dimension → project implementation → knowledge re-matching → project re-implementation." Since an organization' s network topology fundamentally determines work performance [1], we situate individuals within four typical network structures—random networks, small-world networks, scale-free networks, and regular networks—to analyze how network average degree, network scale, and other factors affect organizational innovation performance, as well as the similarities and differences in achieving high-level knowledge collaboration across these four topological structures.

2.1 Collaborative Innovation

With the development of internet technologies, the innovation environment has evolved toward a 2.0 paradigm [2]. In the knowledge economy era, the complexity of knowledge itself (including structural complexity and carrier complexity) and the complexity of human factors (manifested in knowledge activities and social connections) have increased the difficulty of organizational knowledge management. Complex collaborative innovation systems cannot be fully analyzed through linear methods and require investigation from holistic and collaborative perspectives to understand knowledge innovation mechanisms.

Collaborative innovation represents the most dynamic and creative component of knowledge innovation [3], representing a resource matching process based on competition and cooperation that can be analyzed through knowledge integration and interaction dimensions [4]. Henderson et al. [5] first comprehensively elaborated the knowledge integration process: during product development, organizational knowledge forms structural knowledge when driven by external market demands, with integration dimensions encompassing knowledge, resources, behaviors, and performance. The interaction dimension refers to knowledge sharing among individuals, optimal resource allocation, and synchronization of knowledge activities.

Collaborative innovation can be categorized by scope into inter-organizational and intra-organizational forms. As a complex innovation organizational model, collaborative innovation’ s key lies in forming a network innovation model with universities, enterprises, and research institutions as core elements, and government agencies, financial institutions, intermediary organizations, innovation platforms, and non-profit organizations as auxiliary elements. Through deep cooperation and resource integration between knowledge creation and technological innovation entities, this model generates non-linear systemic effects [6]. Much inter-organizational collaborative innovation theory and practice builds upon industry-academia-research foundations. Intra-organizational collaborative innovation emphasizes optimizing internal resource allocation to maximize resource utilization benefits. Wu et al. [7] used multi-agent methods to simulate how transactive memory systems affect interactions between people and tasks in knowledge-based teams, treating the incentive environment of external human and material systems as an important influencing factor. Zhang Gupeng [8] explored how the periodic dynamic evolution of individuals’ interactive network environments affects individual innovation performance from a social network perspective. Xu Shenghua et al. [9] established a quantitative model of intra-organizational knowledge integration using system dynamics methods. While numerous computational experiments have used work performance and innovation performance as evaluation metrics for organizational individual behaviors, few studies have constructed simulation systems from a collaborative innovation perspective.

2.2 Analytical Foundation for Intra-Organizational Collaborative Innovation: Knowledge Supply and Demand

Knowledge supply-demand research involves two types of objects: knowledge demanders and knowledge suppliers. Effective matching between supply and demand parties based on knowledge dimensions is a prerequisite for rational knowledge utilization, a topic extensively studied in knowledge management and artificial intelligence research. Guo et al. [10] reviewed existing research on knowledge supply-demand matching from three perspectives: knowledge characteristics, information system construction, and knowledge services. Liu et al. [11] constructed an ontology-based matching model for tasks and knowledge individuals, proposing a comprehensively improved conceptual similarity calculation method to enhance accuracy in semantic similarity computation. Rather than exploring the matching mechanism itself, this study treats knowledge supply-demand matching as a crucial component within the KSDS model.

Focusing on intra-organizational collaborative innovation, we propose the "Knowledge Supply & Demand System" to analyze the knowledge interaction process between knowledge-supplying agents and knowledge-demanding agents within organizations. The main process includes "matching individuals and projects on the knowledge dimension → project implementation → knowledge re-matching → project re-implementation," representing a cyclical and dynamic collaborative workflow. As a complex socio-technical system, an organization comprises connected human agent sets along with various resources, equipment, and knowledge. From a knowledge dimension perspective, an organization is essentially a complex knowledge supply-demand system whose core components include knowledge-supplying agents (people), knowledge-demanding agents (projects/tasks), and knowledge agents (skills) that permeate all system processes.

3 Organizational Computational Model

Computational and mathematical methods have played important roles in organizational theory development [12], helping to define organizational forms, processes, collective decision-making, consensus negotiation, resource allocation, and organizational structures. Multi-agent simulation represents a bottom-up simulation paradigm that models individual characteristics and behaviors, treating individuals with specific attributes and behaviors as agents. Individual characteristics map to agent attributes, while individual behaviors map to methods [13]. A simulation engine then connects these agents to drive system operation. By leveraging agent autonomy, reasoning, communication, and collaboration mechanisms, this approach simulates phenomena where group members are both independent and interactive, thereby discovering overall group structures and functions [14]. This paper employs multi-agent modeling to simulate fundamental issues of knowledge integration under different network types, offering new perspectives for organizational knowledge management in collaborative innovation.

3.1 Knowledge Supply & Demand System

The Knowledge Supply & Demand System (KSDS) comprises four agent types: developers, projects, project teams, and network environments. [Figure 1: see original paper] illustrates the interactions among these four agent types.

Individuals represent collections of knowledge points, with their knowledge sets reflecting their knowledge backgrounds. Projects can be decomposed into multiple tasks, where each task is a collection of knowledge points. A task's knowledge combination represents knowledge demands, making the project a knowledge requirement document with a certain distribution pattern that is only completed when all requirements are fulfilled. At the micro level, individual knowledge sets interact with project knowledge demands. Based on predefined rules, individuals are selected to form project teams, which then complete corresponding projects as work units. The project implementation process in KSDS is a dynamic and continuous process of filling knowledge requirement documents through teamwork, starting from project knowledge demands.

Under realistic constraints, we simplified model representation with the following basic assumptions: (1) All projects are collaboratively developed by dynamically formed project teams; (2) Both developer skills and project skill requirements are represented by ordered sets of 10 elements; (3) Each project consists of 10 tasks, corresponding to the workload of 10 skills; (4) A developer can only participate in one task within a project per time unit but may join other tasks after completing their current one. Team members can only participate in other projects after all tasks in their current project are completed.

3.3 Rule Definitions

In KSDS, project allocation and implementation are entirely self-organized processes. Developers, projects, and teams interact based on the following rules:

Rule 1: Team Formation. During initial team formation and project implementation, personnel must be supplemented from the network environment. In the initial stage, developer groups and projects continuously interact. When a developer' s skills satisfy at least one project skill requirement (i.e., the developer' s skill level meets the project' s minimum requirement for that skill), the matching degree between the developer and project is greater than or equal to 1, allowing the individual to be incorporated into the project team. We propose the following matching degree formula:

$$MD = \sum_{i=0}^{9} \mathbb{I}(SK_{ji} \geq SK_{pi})$$

where SK represents an ordered skill list containing 10 skills. When j denotes a developer individual, SK_j indicates individual j' s mastery level of the 10 skills, with SK_{ji} representing individual j' s mastery level of skill i. When $SK_{ji} = 0$, individual j does not possess this skill. When p denotes a project, SK_p represents project p' s skill requirements during development, with SK_{pi} indicating project p' s requirement for skill i. When $SK_{pi} = 0$, project p does not require knowledge i during implementation. MD (Match Degree) represents the matching degree.

During project implementation, the matching degree calculation adjusts according to task completion status, only matching uncompleted tasks. We screen individuals from developers who can match the skill requirements of unfinished tasks, proposing Formula (2) as follows:

$$\text{if } SK_{ji} \geq SK_{pi} \text{ and } W_{pi} > 0 \ (i = 0, 1, \dots, 9) \text{ then } MD = MD + 1$$

where SK_{ji} indicates individual *j*' s mastery level of skill i, and W_p represents the workload corresponding to each skill in project p. W_{pi} denotes the workload for skill i in project p. When $W_{pi} > 0$, the task remains uncompleted; when $W_{pi} \leq 0$, the task is completed.

Rule 2: Task Assignment Within Teams. A project can be divided into 10 tasks, each jointly responsible by developer individuals. We assume that individuals can participate in multiple different tasks within a project but can only be responsible for one task per time unit. After completing their assigned task, individuals continue participating in other project tasks until all project tasks are completed. When assigning responsible persons for each task, each task corresponds to the workload of one skill. From a resource optimization perspective, each task always seeks the individual with the highest mastery level of that skill among project members.

Unlike the team formation process based on project skill requirements, the task matching process requires that the responsible person' s skill proficiency for the designated task be the highest among current team configurations. We propose Formula (3) as follows:

$$SK_{ji} = \max(SK_{mi}) \quad \forall m \in M_p$$

where SK_{ji} indicates individual *j*'s mastery level of skill i, and M_p is the member list of project p. If individual j is the responsible person for task i, they must satisfy Formula (3).

Rule 3: Project Implementation. During project implementation, the workload for each task in the project decreases daily. We propose Formula (4) as follows:

$$W_{pi}(t+1) = W_{pi}(t) - \sum_{j \in M_p} SK_{ji}$$

where $W_{pi}(t)$ indicates the remaining workload for task i of project p at time t, SK_{ji} indicates individual *j*' s mastery level of skill i and also represents the daily workload the developer can complete using this skill, and M_p is the member list of project p.

3.4 Environmental Factors

Individual behaviors are influenced by environmental factors. We assume the environment is relatively stable, providing fixed conditions. Here, environmental factors refer to network structure, which can be considered from two aspects: network node (individual) attributes and network properties. This paper focuses on network properties.

Network properties include: network scale represented by the number of actors, and network average degree reflecting the closeness of connections between actors. We assume that individuals in KSDS follow self-organizing behavioral rules, and that incentive mechanisms exist in the environment to make individual behaviors microscopically controllable. Repeating agent behaviors completes projects one after another.

3.5 Performance Metrics

We adopt average project development time cost and average financial cost as indicators to evaluate an organization' s collaborative innovation capability.

Average Time Cost ($T_{average}$): The average time for an organization to complete project development work, calculated as:

$$T_{average} = \frac{\sum_{p=1}^{NP}(ET_p - ST_p)}{NP}$$

where ET_p is project p' s completion time, ST_p is the time when project p first successfully matched members, and NP is the number of completed projects.

Average Financial Cost ($C_{average}$): The average financial expenditure for an organization to complete project development, calculated as:

$$C_{average} = \frac{\sum_{p=1}^{NP}\sum_{j\in M_p}(ET_{pj} - ST_{pj}) \times S_j}{NP}$$

where ET_{pj} is the time when individual j leaves project p, which is also the project completion time; ST_{pj} is the time when individual j joined project p; S_j represents individual j' s salary; M_p is project p' s member list; and NP is the number of completed projects.

3.6 Simulation Engine

The simulation engine integrates agents and the network environment, enabling the system to operate according to the established rules. The process steps are shown in [Figure 2: see original paper].

The simulation procedure is as follows: (1) Initialize the network environment: network average degree, number of developers and projects, and initialize developers (initial state: available) and projects (initial state: undone); (2) Form project teams: During initial team formation, traverse developers in the network environment and match individuals meeting project skill requirements based on Rule 1 to form project teams, changing project status to processing and team member status to occupied; continue matching qualified individuals from project members' neighboring nodes; (3) Assign tasks to team members based on Rule 2, changing member status to working; (4) During project implementation, update each task' s remaining workload based on Rule 3; (5) When a single project is completed, calculate its time and financial costs, and change project member status to available; (6) System termination condition: All projects are completed. Calculate the average time and financial costs for each project completion based on Formulas (5) and (6).

3.7 KSDS Implementation

We implemented the KSDS simulation system using Python, with the interactive network environment generated through the complex network modeling toolkit NetworkX [15]. By calling different NetworkX methods, we constructed specific network structures (random networks, small-world networks, scale-free networks, regular networks). We adjusted characteristics of specific interactive networks by setting network generation parameters. This paper analyzes how network average degree and network scale affect organizational innovation performance. For example, the average degree of random networks equals the product of node count and connection probability, allowing us to determine the interactive network environment's average degree by controlling node count and connection probability in experiments.

4 Experiments and Results Analysis

We randomly generated developer skill lists and project skill requirement lists to explore how the network environment of knowledge agents affects collaborative innovation performance, reflecting developer collaboration efficiency across four typical network structures through time and financial costs.

To investigate the impact of network average degree on project development time, we simulated 1,000 developers collaborating to complete 100 projects under four network types. By adjusting network initialization parameters, we gradually increased the network average degree from approximately 6 to 100 (random networks: 6-103; small-world networks: 6-104; scale-free networks: 6-98; regular networks: 6-104). Each parameter setting was repeated 100 times to calculate the average time and financial costs required to complete a single project under each network type. Results are shown in [Figure 3: see original paper] and [Figure 4: see original paper].

When the average degree exceeds 20, developers in small-world networks achieve lower time and financial costs for collaborative project development. When the average degree exceeds 10, scale-free networks require higher time and financial costs. In random, small-world, and regular networks, as the average degree increases from 6 to 20, project development time decreases sharply and financial costs also decline. [Figure 4: see original paper] shows that within the average degree range of 6 to 100, random, small-world, and regular networks exhibit local optima for financial costs, with optimal solutions between 15 and 30.

To investigate the impact of developer quantity on innovation performance, we set the number of projects at 100. Based on previous results showing local optima for financial costs within an average degree range of 15 to 30, we set the average degree to 16 for all four network types. [Figure 3: see original paper] and [Figure 4: see original paper] reveal that with an average degree of 16 and 1,000 developers, the time costs follow the order $T_{ba} > T_{er} > T_{ws} > T_{re}$ (where ba=scale-free network, er=random network, ws=small-world network, re=regular network), while financial costs follow $C_{ba} > C_{ws} > C_{er} > C_{re}$. In subsequent experiments, we initially set the number of individual developers at 1,020, gradually increasing to 2,000 in increments of 20, with each parameter setting repeated 100 times and results averaged. Experimental results are shown in [Figure 5: see original paper] and [Figure 6: see original paper].

As the average degree increases from 20 to 60, project development time costs continue to decrease steadily, while financial costs generally show an increasing trend. After the average degree exceeds 60, both time and financial costs generally stabilize. Scale-free networks exhibit different collaboration performance patterns compared to the other three network types. As the average degree varies from 6 to 100, multiple local optima exist for time costs. In the stage where average degree increases from 6 to 40, average financial costs continuously increase, stabilizing at approximately 65,000 yuan per project after reaching 40.

As node count increases with an average degree of 16, when the number of nodes exceeds 1,260, the time cost ordering shifts from $T_{ba} > T_{er} > T_{ws} > T_{re}$ to $T_{ba} > T_{ws} > T_{er} > T_{re}$, and the financial cost ordering shifts from $C_{ba} > C_{er} > C_{ws} > C_{re}$ to $C_{ba} > C_{ws} > C_{er} > C_{re}$. As developer quantity increases, the time cost per project development decreases across all four network types to varying degrees, indicating that organizational scale has a significant positive effect on reducing project development time costs. Additionally, developer quantity has minimal impact on financial costs in scale-free networks. Within the range of 1,020 to 1,300 developers, random and regular networks show a steady decline in financial costs.

[Figure 7: see original paper] shows the completion rate over time for 10,000 developers completing 2,000 projects under the four network types, with the network average degree still set at 16. All network types exhibit an initial stage of "0% completion rate." Subsequently, random, small-world, and regular networks experience a rapid increase in completed projects, reaching 50% completion (1,000 projects) in less than 50 days. Although completed projects continue increasing, the growth rate slows significantly, with all projects requiring over

140 days to complete. In contrast, scale-free networks enter a stage of steady increase after the initial "0% completion rate" phase: reaching 35% completion (700 projects) takes approximately 40 days, while 70% completion requires about 80 days. This indicates that organizations under scale-free interactive network environments maintain relatively constant daily output.

5 Summary and Outlook

This study, set against the backdrop of the mobile internet era, examines how individuals' interactive environments affect intra-organizational collaborative innovation from the perspectives of knowledge integration and the interaction mechanisms of innovation elements (individuals, knowledge, and innovation teams). Intra-organizational collaborative innovation represents the rational and efficient utilization of knowledge—effective matching between knowledge demand and supply parties along the knowledge dimension. Using multi-agent simulation methods, we constructed KSDS at the micro level and implemented it through Python NetworkX. Experimental results show that large organizations have advantages over small ones in reducing innovation costs, and that increasing individuals' neighborhood nodes does not monotonically improve organizational innovation efficiency; innovation costs begin to increase once the average degree exceeds a certain threshold. KSDS focuses on simulating the knowledge matching process between supply and demand parties and the knowledge collaboration process of individuals dynamically forming teams to complete projects, providing organizations with new perspectives for improving knowledge utilization efficiency.

Future research will validate and extend these conclusions using empirical data. Human interaction constitutes an important influencing factor in team collaboration processes. The next step will incorporate interaction factors into KSDS, where human interactions change the interactive network environment, which in turn affects collaboration processes—a dynamic and collaborative innovation process that will be the focus of future simulation research.

References

[1] Gao D, Deng X, Zhao Q, et al. Multi-Agent Based Simulation of Organizational Routines on Complex Networks [J]. Journal of Artificial Societies & Social Simulation, 2015, 18(3): Article No. 17.

[2] Song G, Zhang N. Innovation 2.0: Democratizing Innovation in the Knowledge Society [J]. Xinhua News Digest, 2009(10): 140-143.

[3] Practitioner P A K M. Knowledge Management: The Collaboration Thread [J]. Bulletin of the American Society for Information Science & Technology, 2005, 28(6): 8-11.

[4] Serrano V, Fischer T. Collaborative Innovation in Ubiquitous Systems [J]. Journal of Intelligent Manufacturing, 2007, 18(5): 599-615.

[5] Henderson R M, Clark K B. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms [J]. Administrative Science Quarterly, 1990, 35(1): 9-30.

[6] Chen J. Collaborative Innovation and the Construction of National Scientific Research Capacity [J]. Studies in Science of Science, 2011, 29(12): 2-3.

[7] Wu J, Hu B, Liu T. A Simulation Study of the Impact of Interactive Memory System on Group-Task Interaction [J]. Journal of Management Sciences, 2009, 22(1): 48-58.

[8] Zhang G. Dynamic Evolution of Small World Innovation Network and Its Effects [J]. Journal of Management Sciences in China, 2015, 18(6): 15-29.

[9] Xu S, Yin H. Model of Knowledge Integration Based on System Dynamic the Organization [J]. Chinese Journal of Management, 2013, 10(6): 890-897.

[10] Guo R, Chen F. Review on Matching Knowledge Supply and Demand [J]. Information Studies: Theory & Application, 2013, 36(12): 114-118.

[11] Liu J, Zou P, Zhang P. Research on the Matching Model Between Tasks and Knowledge Individuals Based on Ontology [J]. Computer Engineering and Science, 2010, 32(9): 119-121.

[12] Carley K M. Computational and Mathematical Organization Theory: Perspective and Directions [J]. Computational & Mathematical Organization Theory, 1995, 1(1): 39-56.

[13] Macal C M, North M J. Tutorial on Agent-based Modeling and Simulation [C]. In: Proceedings of the 37th Conference on Winter Simulation. 2005.

[14] Carley K M. A Comparison of Artificial and Human Organizations [J]. Journal of Economic Behavior & Organization, 1996, 31(2): 175-191.

[15] Hagberg A, Schult D, Swart P. Exploring Network Structure, Dynamics, and Function Using NetworkX [C]. In: Proceedings of the 7th Python in Science Conference. 2007.

Author Contributions

Wu Jiang: Conceptualized the research, designed the study and program, revised the manuscript. Chen Jun: Developed the program, conducted experiments, wrote the manuscript. Zhang Jinfan: Developed the program.

Conflict of Interest

All authors declare no conflict of interest.

Supporting Data

Supporting data is available in the online version of the journal at http://www.infotech.ac.cn.

[1] Chen J. Developer Quantity Impact.xlsx. Experimental results with developer quantity as independent variable.

[2] Chen J. Average Degree Impact.xls. Experimental results with network average degree as independent variable.

[3] Chen J. Completion Comparison.xls. Project completion rates under four network structures.

A Knowledge Supply-Demand Simulation System for Collaborative Innovation

Wu Jiang, Chen Jun, Zhang Jinfan
(School of Information Management, Wuhan University, Wuhan 430072, China)

Abstract: [Objective] This paper investigates the network environments facing knowledge-based teams and their impacts on job performance. [Methods] We constructed a Knowledge Supply & Demand System from a micro-level knowledge management perspective using multi-agent simulation technology, with time and financial costs as performance evaluation criteria, implemented using Python NetworkX. [Results] Large organizations reduce innovation costs more effectively than small ones. Scale-free network structures require longer completion times and higher costs. Increasing individuals' neighborhood nodes does not monotonically improve innovation efficiency; innovation costs begin rising once the average degree exceeds a certain threshold. [Limitations] The study does not optimize interactions among individuals for collaborative innovation. [Conclusions] The multi-agent-based KSDS simulates organizational knowledge integration processes at the micro level, enhancing understanding of team knowledge management and providing new perspectives for improving knowledge utilization efficiency and reducing innovation costs.

Keywords: Collaborative innovation; Knowledge Supply & Demand System; Multi-agent simulation; Complex networks; Computational experiments

Note: Figure translations are in progress. See original paper for figures.