

COLLECTIVE CREATIVITY IN SCIENTIFIC COMMUNITIES

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Guangyu Zou

Certificate of Approval:

Jeffrey Smith
Professor
Industrial and Systems Engineering

Levent Yilmaz, Chair
Associate Professor
Computer Science and Software
Engineering

Saeed Maghsoodloo
Professor
Industrial and Systems Engineering

George T. Flowers
Dean
Graduate School

COLLECTIVE CREATIVITY IN SCIENTIFIC COMMUNITIES

Guangyu Zou

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of

the Requirements for the

Degree of

Master of Science

Auburn, Alabama

August 10, 2009

COLLECTIVE CREATIVITY IN SCIENTIFIC COMMUNITIES

Guangyu Zou

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Guangyu Zou, son of Baoyong Zou and Xiuzhi Kou, was born on May 12, 1979 in Liaoyang, P. R. China. He graduated from Liaoyang No. 1 High School in 1997. He entered Northeastern University, Shenyang, P. R. China and graduated with a Bachelor of Science degree in Automation in July 2001. He was admitted into the Graduate School of Northeastern University upon recommendation with the entry examination waived. In March 2004 he graduated with a Master of Science in Systems Engineering. From April 2004 to June 2007, he worked at ZTE CO., Ltd as a software engineer. From June 2007 to December 2007, he worked at Alcatel-Lucent as a UMTS OAM Software Developer. After that, he entered Graduate School at Auburn University majoring in Industrial and Systems Engineering in January, 2008. During his graduate study at Auburn University, he performed well on both his academic study and research work. From January 2008 to December 2008, he was a graduate assistant in charge of keeping the department website updated and computer maintenance.

THESIS ABSTRACT
COLLECTIVE CREATIVITY IN SCIENTIFIC COMMUNITIES

Guangyu Zou

Master of Science, August 10, 2009
(M.S., Northeastern University, March 2004)
(B.S., Northeastern University, July 2001)

131 Typed Pages

Directed by Levent Yilmaz

Innovation is the driving force of personal growth, national wealth and social progress. Significant attention has been given to advancing cyber infrastructures, but little is known about their factors, as well as their interaction in producing the context that contributes to creating innovation. It is widely accepted that in open scientific communities, organizational creativity and innovation rate is high. So it is significantly important to analyze such communities in order to better understand their mode of operation. Our objective in this study is to use agent simulation as a computational laboratory to understand the innovation potential of scientific communities. The simulation model serves as a useful thinking tool for policy analysis to foster innovation in scientific communities. The simulation results show that centrality, as a measure of

degree of connectedness, exhibits positive correlation with innovation in exploration-oriented and utility-oriented community but negative correlation in service-oriented community. Additionally utility-oriented communities have social network with low density and high centrality, which suggest high potential for innovation.

ACKNOWLEDGEMENTS

I would like to express special gratitude to my advisor Dr. Levent Yilmaz, associate professor, Department of Computer Science and Software Engineering at Auburn University, for his instruction, guidance, encouragement and patience in completion of the research and thesis. In particular, his suggestions, criticisms and materials greatly contributed to this thesis.

Thanks also my advisory committee members, Dr. Jeffrey Smith, Dr. Saeed Maghsoodloo and the professors and staff members in the Department of Industrial and Systems Engineering at Auburn University for their kindness and help through these two years.

Finally, sincere thanks to my wife Ying Zhao. She gave her greatest support and encouragement to help me succeed in finishing all the research work. Also, I thank my parents, who poured enormous effort into supporting my study during these years.

Style manual or journal used: Guide to Preparation and Submission of Thesis and
Dissertation

Computer software used: Microsoft Word 2007

TABLE OF CONTENTS

LIST OF FIGURES	xii
LIST OF TABLES	xv
CHAPTER 1 Introduction.....	1
1.1 Problem.....	1
1.2 Importance	2
1.3 Methodology.....	2
1.4 Organization of the Thesis	6
CHAPTER 2 Literature Review.....	7
2.1 Scientific Communities.....	7
2.2 Complex Adaptive Systems Perspective.....	10
2.3 Exploration of Scientific Communities Using Agent Based Modeling.....	11
CHAPTER 3 Design Concepts and Details	13
3.1 Purpose.....	13
3.2 State Variables and Scales.....	13
3.3 Process Overview and Scheduling.....	15
3.3.1 Entry and Enculturation	16
3.3.2 Innovation and Generators.....	18
3.3.3 Sub-Domain	24
3.3.4 Evaluator	25
3.3.5 Turnover.....	26
3.3.6 Scheduling.....	27
3.4 Framework for Communicating Individual Agent.....	27
3.4.1 Relationships.....	29
3.4.2 Fitness	30
3.4.3 Stochasticity.....	30
3.4.4 Observation.....	30

3.5	Details	30
3.5.1	Initialization	31
3.5.2	Types of Scientific Communities	31
CHAPTER 4	Implementation of Simulation Model	38
4.1	Introduction of Repast	38
4.1.1	Contexts	39
4.1.2	Projections	39
4.2	Implementation of Agents	41
4.2.1	Basic Agent	42
4.2.2	Individuals	42
4.2.3	Evaluators	43
4.2.4	Kenes	44
4.3	Implementation of Projection	44
4.3.1	Continuous Space	44
4.3.2	Network	45
4.4	Scheduler	46
4.4.1	Directly Schedule an Action	46
4.4.2	Schedule with Annotations	47
4.4.3	Global Method	48
4.5	Output	49
CHAPTER 5	Verification, Validation and Evaluation	50
5.1	Verification	51
5.1.1	Unit Testing	51
5.1.2	Integration Testing	53
5.2	Validation	53
5.3	Experiments	55
5.3.1	Innovation Metrics	55
5.3.2	Network Metrics	57
5.3.3	Sensitivity Analysis	59
5.3.4	Experiments between Different Types of Communities	70
5.3.5	What Distinguishes Innovative Communities?	94

CHAPTER 6	Conclusions.....	100
6.1	Extension.....	101
6.2	Future Research	102
REFERENCES	103
APPENDIX	107

LIST OF FIGURES

Figure 1.1 Complexity in terms of randomness.....	3
Figure 1.2 Systems model of creativity	4
Figure 3.1 Flow chart of the life circle of agents.....	16
Figure 3.2 Susceptibility to influence	17
Figure 3.3 Demonstration for elaboration.....	20
Figure 3.4 Location of new genes.....	21
Figure 3.5 Demonstration for combination.....	22
Figure 3.6 Demonstration for gene selection.....	23
Figure 3.7 The pattern of genes due to sub-domains	25
Figure 3.8 Three kinds of relationships	30
Figure 3.9 Flowchart of decision process of exploration-oriented community	33
Figure 3.10 Flowchart of decision process of utility-oriented community.....	35
Figure 3.11 Flow chart of decision type of service-oriented community	37
Figure 4.1 Contexts and projections	40
Figure 4.2 Class view of systems model of creativity	41
Figure 5.1 Overview of simulation model development	50
Figure 5.2 The number of gene over the time.....	54
Figure 5.3 Percentage of genes	54

Figure 5.4 The pattern of kenes with the size of 10	60
Figure 5.5 The pattern of kenes with the size of 20	61
Figure 5.6 The pattern of kenes with the length of 10	62
Figure 5.7 The pattern of kenes with mainly using elaboration	63
Figure 5.8 The pattern of kenes with balance	64
Figure 5.9 Impacts of growth rate on density.	66
Figure 5.10 Impacts of growth rate on centrality.....	67
Figure 5.11 Impacts of recruitment on density	67
Figure 5.12 Impacts of recruitment on centrality.....	68
Figure 5.13 Impacts of turnover on density	69
Figure 5.14 Impacts of turnover on centrality	69
Figure 5.15 Histogram of the number of agents who create the same number of kenes..	72
Figure 5.16 Plot of the number of kenes created by each agent	73
Figure 5.17 Emergent pattern in GEM layout	74
Figure 5.18 Impact factor over time on exploration-oriented community.....	75
Figure 5.19 Histogram of the number of agents who create the same number of kenes..	77
Figure 5.20 Plot of the number of kenes created by each agents.....	77
Figure 5.21 Emergent pattern in GEM layout	78
Figure 5.22 Impact factor over time on utility-oriented community	78
Figure 5.23 Histogram of the number of agents who create the same number of kenes..	80
Figure 5.24 Plot of the number of kenes created by each agents.....	81
Figure 5.25 Emergent pattern in GEM style.....	82
Figure 5.26 Impact factor over time on utility-oriented community	82

Figure 5.27 Histogram of the number of agents who create the same number of kenes..	84
Figure 5.28 Plot of the number of kenes created by each agents.....	84
Figure 5.29 Emergent pattern in GEM style.....	85
Figure 5.30 Impact factor over time on service-oriented community	86
Figure 5.31 Number of accepted kenes.....	87
Figure 5.32 Average kene fitness.....	88
Figure 5.33 Average diffusion for kenes with different types of communities.....	89
Figure 5.34 Density with different types of communities	90
Figure 5.35 Centrality with different types of communities.....	91
Figure 5.36 Proportion of strong ties with different communities.....	92
Figure 5.37 Clustering coefficient for agents with different types of communities	93
Figure 5.38 Total clustering coefficient with different types of communities.....	93
Figure 5.39 Network metrics of exploration-oriented community grouped by average kene fitness.....	95
Figure 5.40 Network metrics of utility-oriented community grouped by average kene fitness.....	97
Figure 5.41 Network metrics of service-oriented community grouped by average kene fitness.....	98

LIST OF TABLES

Table 3.1 State variables and scales	14
Table 3.2 Parameters of three kinds of communities	32
Table 3.3 Parameters of exploration-oriented community	33
Table 3.4 Parameters of utility-oriented community	34
Table 3.5 Parameters of service-oriented community	36
Table 4.1 Definition of basic agent	42
Table 4.2 Definition of individual	43
Table 4.3 Definition of evaluator	43
Table 4.4 Definition of kene	44
Table 5.1 Range of parameters for communities	65
Table 5.2 Predefined parameters for exploration-oriented community	71
Table 5.3 Results on exploration-oriented community	71
Table 5.4 Predefined parameters for utility-oriented community	75
Table 5.5 Results on utility-oriented community with equal probabilities	76
Table 5.6 Results on utility-oriented community with unequal probabilities	80
Table 5.7 Predefined parameters for service-oriented community	83
Table 5.8 Results on service-oriented community	83
Table 5.9 Experiment results	93

Table 5.10 Network metrics of exploration-oriented community grouped by average kene fitness	95
Table 5.11 Network metrics of utility-oriented community grouped by average kene fitness	96
Table 5.12 Network metrics of service-oriented community grouped by average kene fitness	98

CHAPTER 1

Introduction

1.1 Problem

Innovation is the driving force of personal growth [1], national wealth [2] and social progress [3]. Creativity is the ability to produce work that is both novel and appropriate. Creativity is a topic of wide scope that is important at both the individual and societal levels for a wide range of task domains [4]. From a theoretical standpoint, understanding creative capacity is integral to a complete account of human cognition for the simple reason that human thought is so essentially generative [5]. Significant attention has been given to advancing cyber infrastructures, but little is known about if virtual open science communities are capable of producing context that contributes to creating innovation [6]. The proposition being that virtual organizations can more efficiently and effectively leverage the combination of diverse information and knowledge, skills and resources from different locations and thereby enhance the individual opportunity to learn and the organizational capacity to innovate. To date, however, these claims remain largely untested. Furthermore, the knowledge creation process in such communities is poorly understood [7].

The problem this thesis focuses on is how innovation emerges based on the connections among members in the scientific community and how to determine whether or not a specific configuration has potential to lead to innovation.

1.2 Importance

Scientific communities consist of members that not only work on a common product, but also work together and adjust their actions to new information. Since in such community forms organizational creativity and innovation rates are high, the study on scientific communities is very important to solving problems and sharing knowledge. However, we know little about how social collectives govern and coordinate the actions of individuals to produce innovation output in an effective and efficient way. This study aims to explore alternative forms of organizing and governance to improve innovation and sustainability of such community forms as Exploration-oriented, Utility-oriented and Service-oriented [6].

1.3 Methodology

The strategy adopted here is to explore how and why do communities of innovation form and evolve using agent based modeling and by viewing such communities as a complex adaptive system.

Complex system is a system composed of interconnected parts that as a whole exhibit one or more properties (behavior among the possible properties) not obvious from the properties of the individual parts. In essence, complexity is concerned with

emergence that is the process where the global behavior of system results from the actions and interactions of agents. [8]

A number of scientists have been working on producing several candidate measures of effective complexity. Most of the proposed measures differ from each other but share at least one important characteristic, in that those strictly regular things as well as strictly irregular things are simple, while things that are neither regular nor irregular things are complex. Figure 1.1 illustrates how information, compressibility and randomness relate to any of these useful notions of complexity [9].

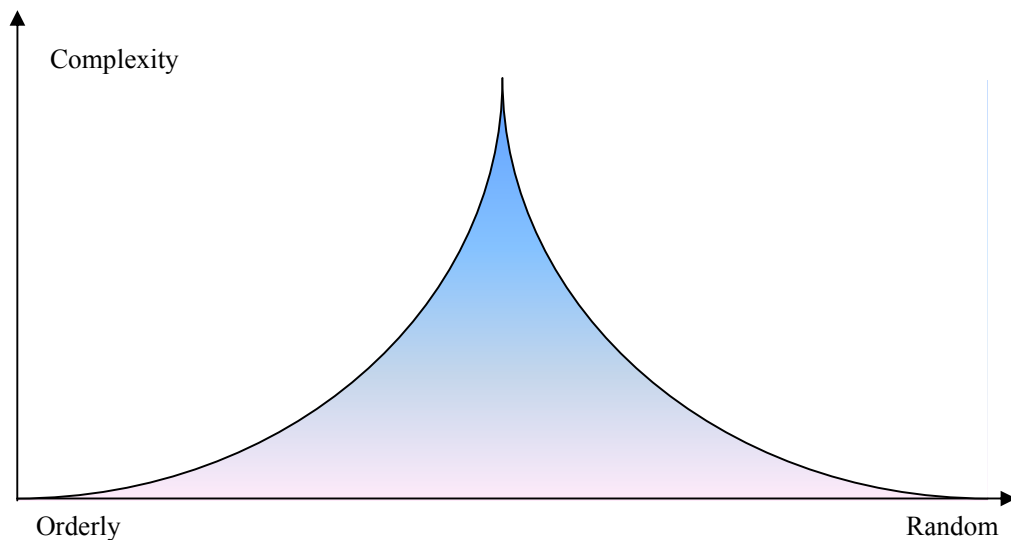


Figure 1.1 Complexity in terms of randomness

So it is noteworthy that complexity is between orderly and random.

Scientific communities have properties of a complex system, such as unpredictable creativity. At the same time, even a simple agent-based model (ABM) can exhibit complex behavior patterns and provide valuable information about the dynamics of the real-world system that it emulates [10]. In order to study a scientific community, agent

based modeling is used here. ABMs provide theoretical leverage where global patterns of interest are more than the aggregation of individual attributes, but at the same time, the emergent pattern cannot be understood without a bottom up dynamic model of the micro foundations at the relational level [11].

Agents with their own states are independent of one another, which indicates the agent based model is heterogeneous, and each agent behaves according to a uniform rule predefined to be imposed on the system.

In a scientific community, there are three main components: individual, evaluator and knowledge units (i.e., kenos) such as articles, experiments and various other artifacts produced during the scientific process whose relationships are shown in Figure 1.2.

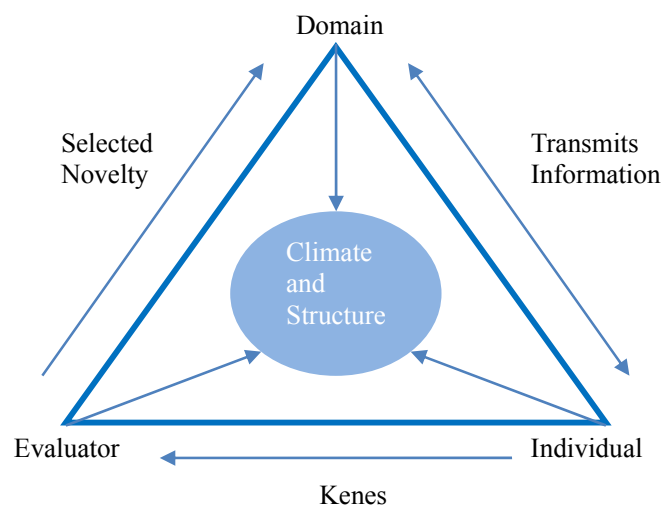


Figure 1.2 Systems model of creativity [adopted and extended [12]]

The components shown in Figure 1.2 make up the systems model of creativity, which is useful to explain the innovation process in the open scientific society. The first component is called “individual”, and it plays the role of generator, which can make

contributions to the society. The second component is “evaluator” whose function is to judge if the contributions created by individuals are appropriate. The last part is the domain that contains the knowledge the members of the society created and are interested in, and domain is composed of kenes that represent the knowledge units.

As shown in Figure 1.2, there are also three major components in the simulation model presented in this thesis, which are individual, evaluator and kene. Kenes are created by individuals and are evaluated by evaluators. So kenes don't have any behaviors except that they own some state variables. In addition, both individual and evaluator have a common super class that is called basic agent, because both of them have some of the same behaviors such as moving in the grid. Therefore, there are two types of agents i.e. individual and evaluator. In addition our purpose is to analyze how these two kinds of agents interact with each other to generate patterns of kenes and socio-technical networks.

Repast is used as a computational laboratory to simulate the activities of individuals in the scientific community, where two kinds of agents represent individual and evaluator respectively, and three sorts of networks correspond to the relationships among agents. Meantime agents with different states follow the same rule to move independently. The simulation results show that centrality has significantly effects on innovation over all communities. Additionally a utility-oriented community has low density and high centrality for social network, which yield more potential for innovation. Based on the analysis of impact factors changing over time, the research topics in the utility-oriented community lose the interests of public in a shorter period than other two types of communities. At the same time there is the greatest variation of average kene

fitness in utility-oriented communities. Finally elaboration plays an important role on innovation. However, extensive divergence may mean lack of coherence, so balance is needed.

1.4 Organization of the Thesis

The additional sections of this thesis are organized as follows. The next chapter is a literature review that is composed of a comprehensive review of all background knowledge and circumstances pertinent to creativity in scientific communities. Chapter 3 is based on the ODD (Overview, Design concepts, Details) template, which describes the overall design of the simulation model. And chapter 4 focuses on the implementation of a simulation model using Repast, followed by chapter 5 that verifies, validates and evaluates the simulation model. The last chapter summarizes some conclusions about using Repast to simulate creativity in scientific communities.

CHAPTER 2

Literature Review

2.1 Scientific Communities

The scientific community consists of scientists, domain knowledge as well as their relationships and interactions. It is normally divided into "sub-communities" each of which works on a particular field within science, and objectivity is expected to be achieved by the scientific method. Peer review, through discussion and debate within journals and conferences, assists in this objectivity by maintaining the quality of research methodology and interpretation of results.

Promoting affiliation between scientists is relatively easy, but creating larger organizational structures is more difficult, due to traditions of scientific independence, difficulties of sharing implicit knowledge, and formal organizational barriers. [13]

An Open Community System is an open project that aggregates efforts of many geographically separate individuals toward a common research problem. [13] Open scientific community forms of organization, which have emerged in recent years, are based on open research conducted in the spirit of free and open source. Much like open source schemes are built around a source code that is made public, the central theme of open research is to make clear accounts of the methodology, along with data and results extracted from the internet. This permits a massively distributed collaboration. Such a

scientific form has affected almost every aspect of scientific communities including

Social sciences: Anthropology, Economics, Psychology, Geography, Linguistics, Philosophy, Political science, Sociology, History, Education, Law, Management and

Applied sciences: Architecture, Cognitive sciences, Engineering, Health sciences (Medicine), Military Science etc.

Open source software is the most prominent example of open source development. The success of Linux, an open source operating system, is currently receiving much attention by software developers and software users alike. Linux is touted as highly stable and reliable. It has steadily increased its market share and has led to a consolidation among UNIX operating systems. Typically, open source software is developed by an internet-based community of programmers. Participation is voluntary and participants do not receive direct compensation for their work. In addition, the full source code is made available to the public [14].

Open source systems are built by potentially large numbers (i.e., hundreds or even thousands) of volunteers. Work is not assigned; people undertake the work they choose to undertake. There is no explicit system-level design, or even detailed design. There is no project plan, schedule, or list of deliverables [15]. So work on the open source projects can be summarized as

- a creative exercise
- leading to useful output
- where the creativity is a lead driver of individual effort.[16]

The value of any kind of data is greatly enhanced when it exists in a form that allows it to be integrated with other data. Unfortunately this has led to a proliferation of

ontologies when using common controlled vocabularies or ontologies. In biomedical domain, for instance, the Open Biomedical Ontologies (OBO) consortium is pursuing a strategy to overcome this problem of data integrated with other data. Existing OBO ontologies, including the Gene Ontology, are undergoing coordinated reform, and new ontologies are being created on the basis of an evolving set of shared principles governing ontology development. The result is an expanding family of ontologies designed to be interoperable and logically well formed and to incorporate accurate representations of biological reality. [17]

Any individual who wishes to join the OBO needs to follow these steps.

1. First join one or more mailing lists in salient areas as a way to become familiar with the collaborative.
2. This will be followed by a period in which compliance with the principles is addressed, especially as it concerns potential conflicts in areas of overlap.
3. By joining the initiative, the authors of ontology commit to working with other members to ensure that, for any particular domain, there is convergence on a single ontology.

There are two central features of collective invention [18]: firms release to their competitors information about the design and efficiency of new plants or technologies; and individual firms devote very little resources explicitly to the discovery of new knowledge. Thus the key to understanding collective invention is in the exchange and free circulation of knowledge and information within groups rather than in the inventive efforts of particular firms or individuals.

Our model discussed in Chapter 3 has a similar life circle of individuals as above.

Prior research suggests that open scientific communities can be considered a complex, self-organizing system that is typically comprised of large numbers of locally interacting elements. Although the rules describing those local interactions may be few and simple, often global properties emerge that are unexpected and difficult to predict. [19] The next section discusses the complex adaptive system in details.

2.2 Complex Adaptive Systems Perspective

Human societies are complex in that there are many, non-linear interactions between their units, that is between people. The interactions involve the transmission of knowledge and materials that often affect the behavior of the recipients. The result is that it becomes impossible to analyze a society as a whole by studying the individuals within it, one at a time. The behavior of the society is said to emerge from the actions of its units [20]. It is indeed precisely suggested in [21] that the emerging architecture of complex systems tended to often be spontaneously such, because complex systems were born out of simple ones, and because simple systems then tend to be somehow included in more complex ones. As for our modeling of scientific community development, the rationale for the emergence of a hierarchy of modules is strongly similar: a complex system is dynamically born out of a simple one; new modules are created out of existing ones to supplement them by developing existing functionalities or adding new ones; and these new modules can be included in higher ones during the compilation process or at least are called as sub-systems. [22]

2.3 Exploration of Scientific Communities Using Agent Based Modeling

In order to study open science communities, agent based modeling (ABM) is used here. ABMs provide theoretical leverage where the global patterns of interest are more than the aggregation of individual attributes, but at the same time, the emergent pattern cannot be understood without a bottom up dynamic model of the micro foundations at the relational level. [11]

Agent-based modeling relies on a novel view of the creation of structure in social systems. Traditional social science generally assumes that social facts such as markets or cooperative behavior exist, and that they produce various forms of social organization and structure. Agent-based modeling assumes that both social structure and such social facts as markets or cooperative behavior are created from the bottom up via the interactions of individual agents. Rather than examining how social structure shapes behavior, agent-based modeling focuses on how local interactions among agents serve to create larger and perhaps global social structures and patterns of behavior. [23] Currently there are a lot of successful applications using ABM. For instance, agent-based computational economics (ACE) is the computational study of economies modeled as evolving systems of autonomous interacting agents. Thus, ACE is a specialization of economics to the basic complex adaptive systems paradigm [24].

An understanding of organizational creativity will necessarily involve understanding the creative process, the creative product, the creative person, the creative situation and the way in which each of these components interacts with the others [25].

The following parts will be organized based on the pattern of ODD which stands for overview, design concept and details [26].

CHAPTER 3

Design Concepts and Details

3.1 Purpose

Applications-oriented social scientific simulation models are characterized by relatively complex agents, as they contain as a rule several social scientific theories, and by the fact that, at the same time, the simulations work with large populations of tens of thousands of agents. The simulations may be grounded in empirical data and are utilized to model real social systems, whereby social networks must also be modeled in a functionally equivalent way to the real social world. The goal of computer modeling is the optimization of social interventions. [27]

There are two goals to simulate the process of knowledge production. One is to use agent based simulation as a computational laboratory to discover the innovation of a scientific community. The other is to explore innovation output of the simulated different communities under a variety of configurations and parameter settings.

3.2 State Variables and Scales

Table 3.1 presents the attributes of the individual and evaluator. An individual has seven attributes associated with it. The first three are probabilities that indicate the frequency to invoke the three kinds of generators for contribution that will be discussed

in section 3.3.2. Also the sum of these three probabilities is one. Motivation represents the probability for an agent to be activated in each time interval. Motivation is based on a reputation that corresponds to the agent's contribution. In general, the more genes an agent creates, the higher reputation he has. The tenured attribute denotes whether or not an agent has passed the enculturation process. Only those agents who are tenured can make contributions to the community where they exist. In addition, the time when the individual is created is used to find an appropriate time in the future to judge if he has reached the requirements to be tenured. The last attribute is the sub domain the agent is most interested in.

As far as the evaluator is concerned, there are three state variables associated with it. The first two weight factors are used to compute the fitness of a specific gene. If its fitness is greater or equal to the threshold, it will be retained in the domain. Otherwise it has to be removed.

Table 3.1 State variables and scales

Components	State Variables
Individual	The probability to elaboration
	The probability to create
	The probability to combination
	Reputation
	Motivation
	Tenured
	The time when the individual is created
	Sub domain
Evaluator	Weight of input links of genes
	Weight of output links of genes
	Threshold for a gene to be retained

There are also some predefined scales for this simulation model.

- The size of grid

This parameter is used to define the grid where all the agents are.

- The length of kene

This parameter denotes the size of bit vector which is used to present the kenes.

- The stop tick

It is to define the time when the simulation will end.

3.3 Process Overview and Scheduling

There are three novel factors in Neil Smith's model: the complexity of software modules as a limiting factor in productivity, the fitness of the software to its requirements, and the motivation of the developers. [28] In the model, the kene's bit vector, kene's fitness and motivation of individual are corresponding to these three factors respectively.

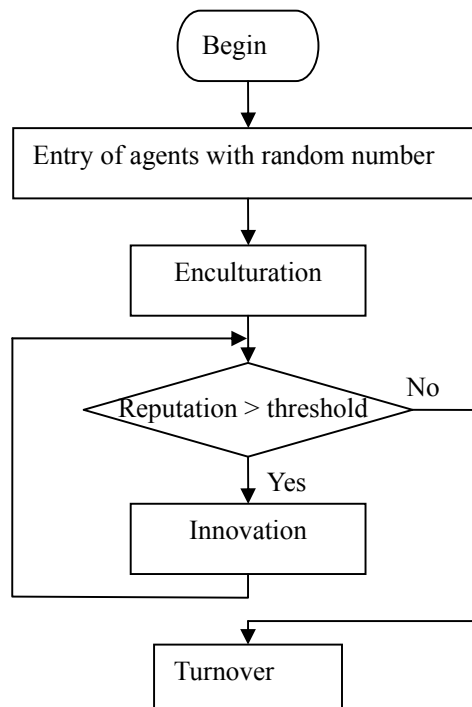


Figure 3.1 Flow chart of the life circle of agents

Figure 3.1 shows the flow chart of an agent's life circle.

At each time interval, a random number of new agents enter the community. The max number of new agents is different with the types of societies, such as exploration-oriented, utility-oriented and service-oriented, where a utility-oriented community has the max growth rate, a service-oriented has a moderate value of growth rate and an exploration-oriented community has the minimum value.

After a new agent moves into the society, he begins the enculturation process during which he becomes gradually familiar with the society. Then a constant time interval later, an evaluation process occurs to determine whether or not the new agent has enough enculturation to do contribution to this domain. If and only if his enculturation level is greater than a threshold, the new agent is qualified. Otherwise, the new agent will leave the society. The threshold to evaluate enculturation level is also different with the types of communities, where an exploration-oriented community has a higher level than a utility-oriented community whose threshold is greater than a service-oriented community.

The following sections will discuss the entry, enculturation, innovation and turnover processes in details.

3.3.1 Entry and Enculturation

At every time interval, a random number of new agents enter the community and begin the enculturation process. Agents go through the enculturation process to become familiar with the domain before making contributions.

During the enculturation process, agents move randomly within the knowledge space. As they move, at each time interval they select a random number of community members in their neighborhood to interact with. The change in fitness level of the agent is a function of its susceptibility to influence and the intensity of influence it receives from the agents that it interacts with.

The first parameter is the susceptibility of agents, which is defined as follows.

$$S_i(t) = \beta_0 + e^{-\beta_1 - \beta_2 \gamma_i(t-1)} \quad 3.1$$

Where, $\gamma_i(t-1)$ is the time period during which the agent has been in the community. β_0 , β_1 and β_2 make sure the trend is that initial susceptibility is high and it decreases exponentially.

The curve of this function is like Figure 3.2.

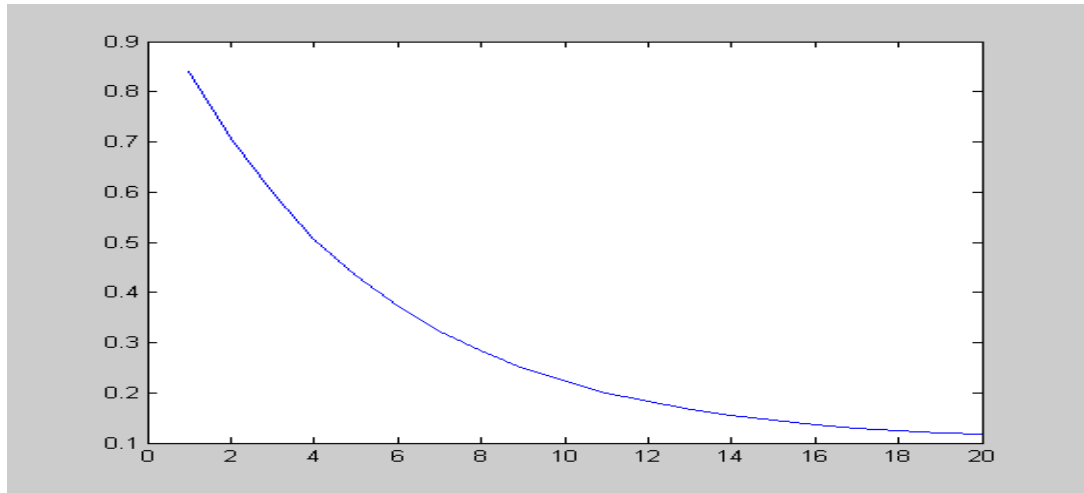


Figure 3.2 Susceptibility to influence

The second parameter is the inclination of agents, which is defined as equation 3.2.

$$I_i(t) = \sum_{j=1}^n I_{ij}(t) \quad 3.2$$

$$I_{ij}(t) = (F_j(t-1) - F_i(t-1)) \alpha_{\text{peer}}$$

Where, α_{peer} is the rate at which the enculturation level is pulled toward the peer. $I_{ij}(t)$ indicates the influence of peer j to i . The influence that an agent receives is the sum of that from all the peers.

The enculturation level of an agent is then specified as a function of the susceptibility and inclination.

$$F_i(t) = F_i(t - 1) + S_i(t)I_i(t) \quad 3.3$$

3.3.2 Innovation and Generators

Evolutionary theory of technical change often contains the following components.

1. The point of departure is the existence and reproduction of entities like genotypes in biology or a certain set-up of technologies and organizational forms in innovation studies.

2. There are mechanisms that introduce novelties in the system. These include significant random elements, but may also produce predictable novelties.

3. There are mechanisms that select from among the entities present in the system. The selection process reduces diversity and the mechanism operating may be the natural selection of biology. [29] In this model, all three characters are included, which indicates the potential to create innovation to some extent.

At each time interval, the motivation of agents determines whether or not this agent will make a contribution. The agents with higher motivation are more likely to gain the opportunity of innovation. There are three types of generators including creation, elaboration and combination.

3.3.2.1 Kene Creation

The creation operator is to create a new kene independently. The steps are as follows.

1. Randomly select a number n between 0 and the length of kene.
2. Randomly select n bits from a kene and assign every selected bit as 1.
3. Calculate the distance between the new kene and reference point, which is the number of bits with the value 1.
4. Put the new kene at one of the four possible locations.

Knowledge becomes highly idiosyncratic, does not diffuse automatically and freely moves among agents and it has to be absorbed by agents through their differential abilities accumulated over time [30]. Therefore, every kene has its location and can be learned by other individuals.

3.3.2.2 Kene Elaboration

Production system based agents have the potential to learn about their environment and about other agents through adding to the knowledge held in their working memories. The agents' rules themselves, however, always remain unchanged. For some problems, it is desirable to create agents that are capable of more fundamental learning: where the internal structure and processing of the agents adapt to changing circumstances. There are two techniques commonly used for this: neural networks and evolutionary algorithms such as the genetic algorithm (GA). [31] In my model an evolution mechanism similar to GA is used to create new kenes.

The elaboration is to generate a new kene based on an already existing kene. The steps are as follows.

1. Select a kene retained in the domain randomly and copy it to a new kene.
2. Randomly select the beginning position and end position in the new kene to mutate.
3. Randomly change the value of selected bits in the kene between the beginning position and the end position, as is illustrated in Figure 3.3.
4. Calculate the distance from new kene to original kene.
5. Put the new kene at one of the four possible locations.

The equation to calculate the location of new kene is as following.

$$b_x = a_x + D_x \text{ or } b_x = a_x - D_x \quad 3.4$$

$$b_y = a_y + D_y \text{ or } b_y = a_y - D_y \quad 3.5$$

Where $[a_x, a_y]$ is the coordination of kene a and $[b_x, b_y]$ is the coordination of kene b. D_x is equal to the number of x-dimensional bits different between kene a and kene b. Similarly, D_y is equal to the number of y-dimensional bits difference between kene a and kene b. So the position of the new kene will be any one of these four possible positions, which is depicted in Figure 3.4.

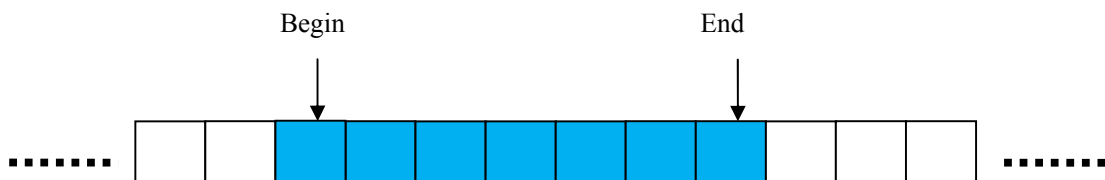


Figure 3.3 Demonstration for elaboration

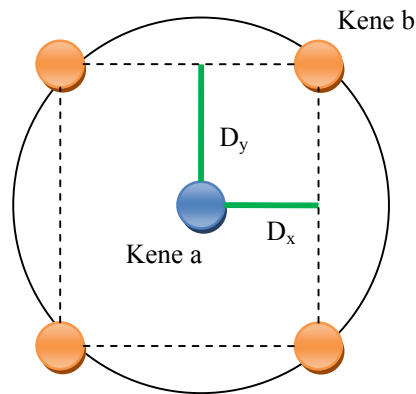


Figure 3.4 Location of new kenes

3.3.2.3 Kene Combination

The combination is to generate a new kene based on more than two existing kenes.

The steps are as follows.

1. Randomly select a number of existing kenes. In this model, the number is selected arbitrarily as 3.
2. Create a new kene and randomly select two positions in the bit vector of this kene.
3. Copy the bits before the first position from the first kene. Copy the bits between the first position and the second position from the second kene. Copy the bits after the second position from the third kene. The process is demonstrated in Figure 3.5.
4. Calculate the position of the new kene based on equation 3.6, which is also repeated for the y dimension.

$$x = w_1x_1 + w_2x_2 + w_3x_3 \quad 3.6$$

The x_1 is the x -position of the first kene. The w_1 is the weight of the first kene, which is the ratio of the number of bits contributing to new kene to the length of kene. Also it is held that $w_1 + w_2 + w_3 = 1$.

- Put the new kene at the location calculated using the method above.

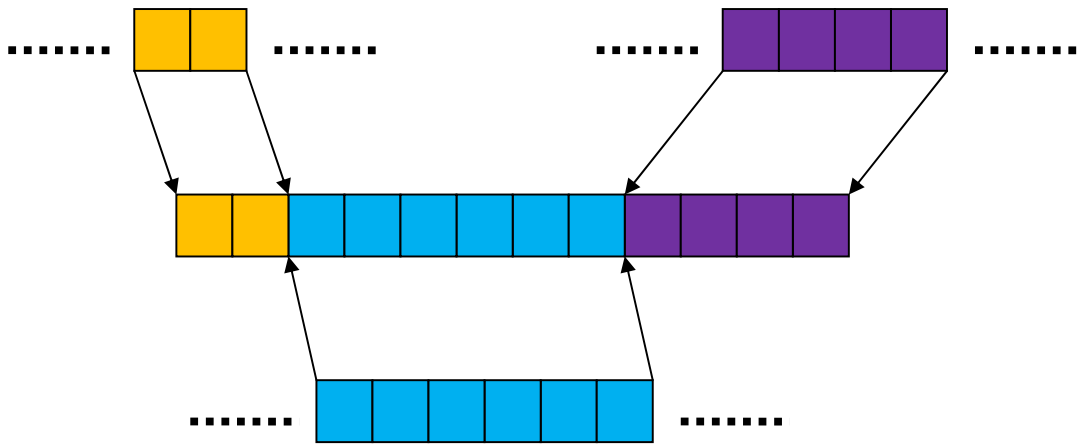


Figure 3.5 Demonstration for combination

3.3.2.4 Principle of Kene Selection

Kenes with higher fitness are cited more frequently than those with lower fitness. In order to implement this, we define distribution to indicate the probability of kenes chosen to combine or elaborate, which is based on the fitness of kene. Assume that the fitness of kene i is f_i and the total number of kenes is N . Then the probability of the kene i to be chosen is calculated using the following equation.

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad 3.7$$

At each time interval, a random number is created. If this random number is smaller than the probability of kene to be chosen, then that kene will be selected to perform combine or elaborate. In Figure 3.6, there are four kenes whose probabilities to be selected are 0.1, 0.2, 0.3 and 0.4 respectively. Further the kene in blue covers the range from [0, 0.1); the kene in dark red covers the range [0.1, 0.3); the kene in green covers the range [0.3, 0.6), and the kene in violet covers the range [0.6, 1). If the generated number is 0.4, then the green kene will be selected.

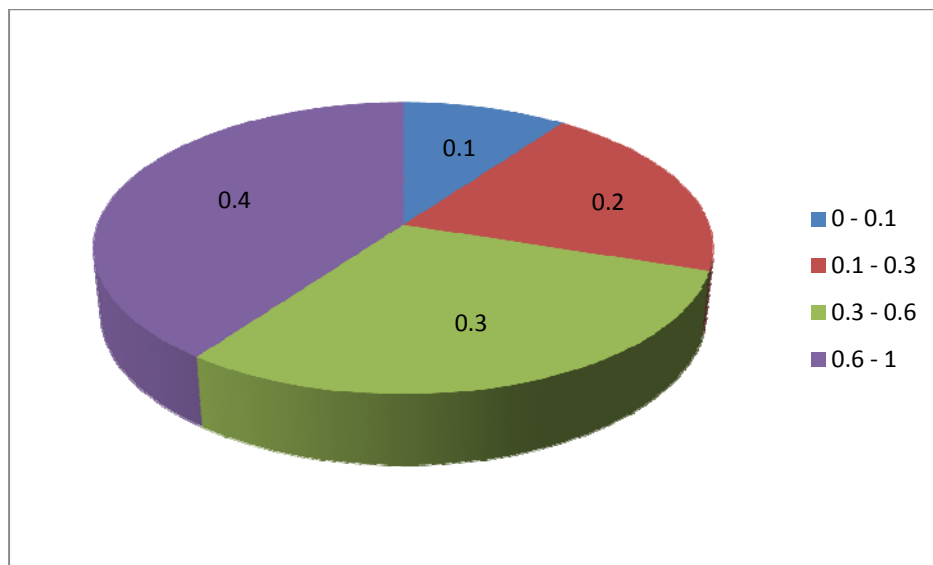


Figure 3.6 Demonstration for kene selection

Additionally, at every time interval, a similar mechanism is used to determine whether or not a specific individual will contribute to the domain knowledge. The probability of individual to make a contribution is based on his/her reputation that is assigned randomly in the initialization process. The reputation will change dynamically during the simulation process, which means the reputation of an individual will increase if the kene published by him/her is accepted. On the other hand, the reputation will

decrease if his/her kene is declined. We will discuss the reputation rule in details in the following section.

3.3.2.5 Credits on Contributions

If the kene created by an agent is accepted by the evaluator, the reputation of this agent will increase.

$$R_i(t) = R_i(t - 1) + (1 - \alpha(t)) * (1 - R_i(t - 1)) \quad 3.8$$

Where, $R_i(t)$ is the reputation of agent i at time t . And $R_i(t - 1)$ is the reputation of agent i at time $t - 1$.

On the other hand, if the kene is declined by the domain, the corresponding reputation level will decrease.

$$R_i(t) = R_i(t - 1) - \alpha(t) * R_i(t - 1) \quad 3.9$$

Where, α is the proportion that the reputation of agent i changes, and the proportion monotonically increases with successive acceptance or refusal, which trend is as equation 3.10. Once the successive acceptance or refusal sequence is interrupted, α is set to $\alpha(0)$.

$$\begin{aligned} \alpha(t) &= \alpha(t - 1) + 0.5 * (1 - \alpha(t - 1)) \\ \alpha(0) &= 0.1 \end{aligned} \quad 3.10$$

3.3.3 Sub-Domain

Each individual has its own sub domain that indicates what fields that individual is interested in. In this model the spaces of different sub domains have different reference points. All the kenes created by the individuals with the same domain are based on the unique reference point. The sub domain is a kind of trait [32] which has a set of distinct values. For examples, there are several sub domains in OBO (Open Biomedical

Ontologies) such as anatomy, biochemistry and taxonomy etc. At the same time anatomy also has its own sub-domains such as amphibian gross anatomy, fungal gross anatomy, cell type and cell component etc.

Figure 3.7 depicts the situation where there are four sub domains. Every individual belongs to one of these four domains so that the locations of their contributions are based on the corresponding reference point.

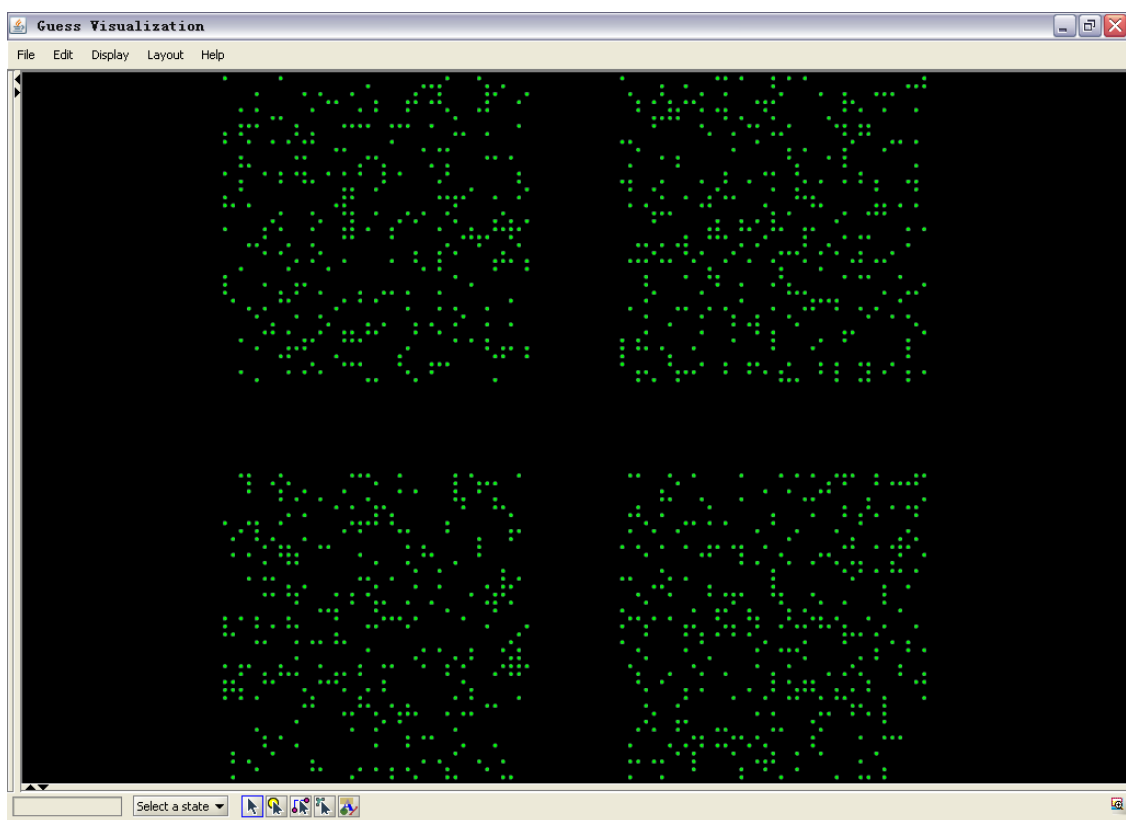


Figure 3.7 The pattern of kenes due to sub-domains

3.3.4 Evaluator

Each evaluator selects a kene randomly and judges whether or not it can be retained based on its fitness. One kind of fitness, called individual fitness, is assigned to a kene with a random number between 0 and 1. The other fitness is relational fitness and it

is calculated by the number of input and output links of a kene. If the total fitness of a kene is less than a particular threshold, the kene will be discarded. On the contrary the kene with a fitness greater than the threshold, will be retained in the domain in order to be reused by other individuals.

$$\text{Relational Fitness} = w_{\text{out}} * N_{\text{out}} + w_{\text{in}} * N_{\text{in}} \quad 3.11$$

$$\text{And } w_{\text{out}} + w_{\text{in}} = 1$$

$$\text{Ratio of Relational Fitness} = \text{Relational Fitness} / \text{Max Relational Fitness} \quad 3.12$$

Where w_{out} and w_{in} are the weights associated with influence and dependence respectively, and N_{out} and N_{in} are the number of links outward and inward respectively. Additionally, the max relational fitness is the max value of all the kenes' relational fitness.

So the final fitness equation is as below.

$$F(k) = a * I(k) + (1-a) * R(k) \quad 3.13$$

Where a is the weight of individual fitness. $F(k)$ is the final fitness of kene k . $I(k)$ is the individual fitness of kene k . $R(k)$ is the ratio of relational fitness of kene k .

3.3.5 Turnover

If the motivation of an agent is less than the exit threshold, it leaves the community or transfers to another community in which it has more potential to increase motivation and make contributions.

The exit threshold is different with the types of communities including exploration-oriented, utility-oriented and service-oriented. Here service-oriented community has the maximum threshold, utility-oriented has a moderate value and exploration-oriented

community has the minimum threshold. And the higher the threshold for turnover is, the easier an agent will leave the community.

3.3.6 Scheduling

The scheduling deals with the order of processes and in turn the order in which the state variables are updated. In my model the characters of scheduling are as follows.

- Use discrete time steps.
- A global method to calculate the rank of each individual which determines the probability of making contributions. The rank of individual is based on his or her reputation which reflects the contribution and familiarity towards the domain.
- At each time interval the execution sequence for individual is random and every individual updates its state asynchronously according to this order.
- Every individual is based on the current context, i.e. asynchronous updating [33].

3.4 Framework for Communicating Individual Agent

The activities of scientific communities are simulated by Repast integrated simulation framework that is a kind of software for creating agent based simulations [10] using the Java language.

All the kenes and agents including individuals and evaluators exist in a two dimension grid. At each time interval agents move randomly in the grid.

Three staged models of scientific society are developed with increasing levels of sophistication to study innovation and sustainability.

The first model considers the creation and development of knowledge in the scientific society, including the contribution of new knowledge, growth of the domain, citation behavior, and the clustering of knowledge into specialties. For the contribution of new knowledge three kinds of generators are available to use including creation, combination and elaboration. In addition, the generator of combination and elaboration will lead to citation behavior that includes not only the connection among genes but also the relationship involving the associated owners of genes.

The second model views a scientific community as an autonomous system through the introduction of new individuals and the interconnection between normal individuals and evaluators. There are three life stages for an individual including enculturation, innovation and turnover process, where the innovation process is implemented in the first model. So the activities of agents in the enculturation and turnover process will be the key points of research at this stage. Different parameters will be investigated such as susceptibility for influence of other agents, decay rate and enculturation rate.

The next model will extend the original model to simulate the different types of societies based on a variety of cultural parameters such as exploration-oriented, utility-oriented and service-oriented which are different from one another in the following aspects: recruitment selectivity, growth rate, turnover rate and decision making style.

3.4.1 Relationships

There are three types of relationships including kene and kene, individual and individual, kene and individual.

Figure 3.8 shows these three kinds of relationships.

i. Relation among kenes

The relationship between kenes indicates that a kene is elaborated or combined with others.

ii. Relation among individuals

The owner individual of new kene is related to that owner of kene used to derive the new kene. This relation captures the influence between individuals.

iii. Relation between kenes and individuals

That an individual is related to the kene created by him indicates the individual is the owner of the kene.

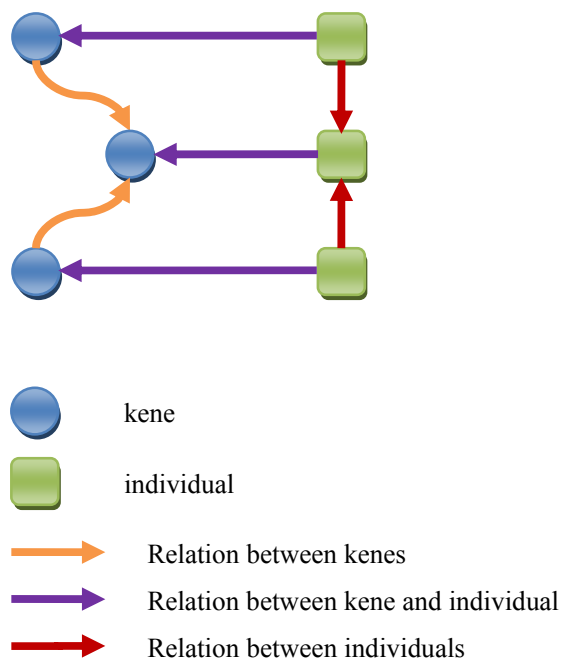


Figure 3.8 Three kinds of relationships

3.4.2 Fitness

There are two simple rules for the fitness of agents.

If the kene created by an agent is accepted by the evaluator, the fitness of this agent will increase. On the other hand, if the kene is declined by the domain, the corresponding fitness level will decrease. In the model, fitness is equivalent to reputation.

3.4.3 Stochasticity

In this model there are various stochastic aspects as described below.

- Individual randomly selects three kinds of generators.
- The kenes to combine and elaborate are selected randomly.
- Individuals and evaluators move randomly in the context.
- The location of new kene is chosen randomly in the four possible locations.

3.4.4 Observation

This section is about how data are collected from the agents based model for testing, understanding, and analyzing. In the model, there are three metrics observed over the time periods such as the number of kenes increases over time, the input and output links of kenes, how many kenes are created by each individual.

3.5 Details

In this section we discuss model elements (initialization, input, sub-models) that present the details that were not discussed in the overview.

3.5.1 Initialization

This part deals with how the environment and the individuals are created at the start of a simulation run.

- The fitness of each individual is set randomly.
- The sub domain of each individual is randomly selected.
- The fitness of each gene is also set randomly.
- The length of gene is 10.
- The width and height of grids are 200 respectively.
- The probability for three generators is the same.
- The initial location of individual and evaluator is random.
- The reference point is the center of grid.

3.5.2 Types of Scientific Communities

To test the different creative output of an open scientific community based on different evaluation strategies, we considered three types of open source society: Exploration-oriented, Utility-oriented and Service-oriented. The objective of exploration-oriented is to share innovations and knowledge. One example of exploration-oriented is OBO foundry whose goal is to create a suite of orthogonal interoperable reference ontologies in the biomedical domain, thereby enabling scientists and their instruments to communicate with minimum ambiguity. In this way the data generated in the course of biomedical research will form a single, consistent, cumulatively expanding whole. The objective of utility-oriented is to satisfy an individual need. The example of utility-oriented is nanoHUB organization whose vision is to pioneer the development of

nanotechnology. The members in nanoHUB are developing resources to help others learn about nanotechnology while making use of it in their own research and education. The purpose of service-oriented is to provide stable services, an example of which is ontology lookup service (OLS). The OLS provides a web service interface to query multiple ontologies from a single location with a unified output format.

These three community cultures differ from each other in terms of recruitment selectivity, growth rate, turnover and decision making style.

The table below defines the three kinds of communities.

Table 3.2 Parameters of three kinds of communities

Community Type	Characters	
Exploration-oriented	Recruitment Selectivity	High
	Growth rate	Low
	Turn over	Low
	Decision-making Style	Centralized
Utility-oriented	Recruitment Selectivity	Moderate
	Growth rate	High
	Turn over	Moderate
	Decision-making Style	Emergent selection
Service-oriented	Recruitment Selectivity	Low
	Growth rate	Moderate
	Turn over	High
	Decision-making Style	Council

Recruitment selectivity indicates the threshold that determines whether an agent will begin to contribute or leave the community after the process of enculturation. If the enculturation level is greater than the threshold, the agent begins to contribute. Otherwise the agent will leave the community.

Growth rate indicates the number of new individuals entering the community at each time interval i.e. the number belongs to $U(0, \text{Growth Rate})$.

Turnover indicates the threshold that determines whether or not an agent will leave the community. If the motivation level is less than the threshold, the agent will leave the community. Otherwise, the agent will stay in the community and continue to make contributions to the domain.

Decision making style involves the process that determines whether to accept or reject a contribution.

3.5.2.1 Exploration-oriented

Table 3.3 Parameters of exploration-oriented community

Community Type	Characters	
Exploration-oriented	Recruitment Selectivity	High
	Growth Rate	Low
	Turnover	Low
	Decision-making Style	Centralized

Centralized indicates that each kene created by agents is judged by a single evaluator, which process is shown in Figure 3.9.

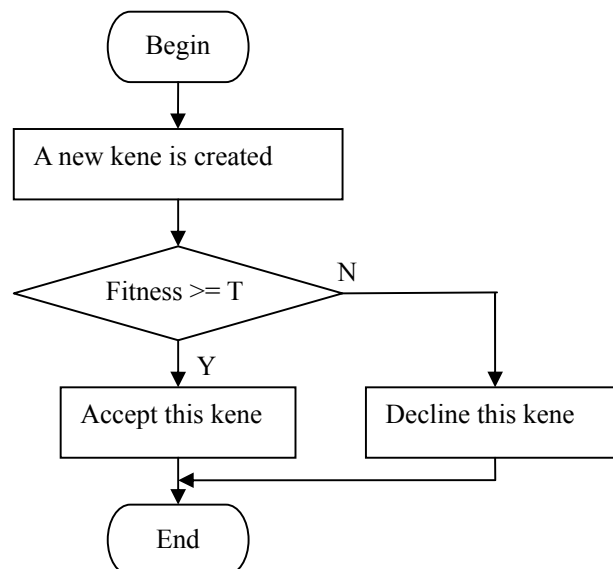


Figure 3.9 Flowchart of decision process of exploration-oriented community

Once a new gene is created by an agent, the evaluation process takes place to determine whether to accept it or reject it. The evaluation is based on the gene's fitness value that is a random float number belonging to uniform distribution from 0 to 1. In general, the threshold T for a gene to be selected is 0.5. If the fitness of the gene is greater than 0.5, it will be retained in the domain and would also be referenced by other agents in the future. On the other hand, the new created gene has to be removed if its fitness value is less than 0.5.

3.5.2.2 Utility-oriented

Table 3.4 Parameters of utility-oriented community

Community Type	Characters	
Utility-oriented	Recruitment Selectivity	Moderate
	Growth Rate	High
	Turnover	Moderate
	Decision-making Style	Emergent selection

Emergent selection decision is implemented as following. Because it is impossible for the new gene to be used by other individuals when it is just created, the evaluation of it will be postponed to a definite number of time periods later, such as 50 ticks. Then if references to this gene are greater than a threshold, it is accepted. Otherwise the gene will be removed from the domain. This process is shown in the flow chart below.

T_1 is a threshold to determine when the created gene will be evaluated and T_2 is another threshold to determine the minimum number of out lines for a gene to be accepted.

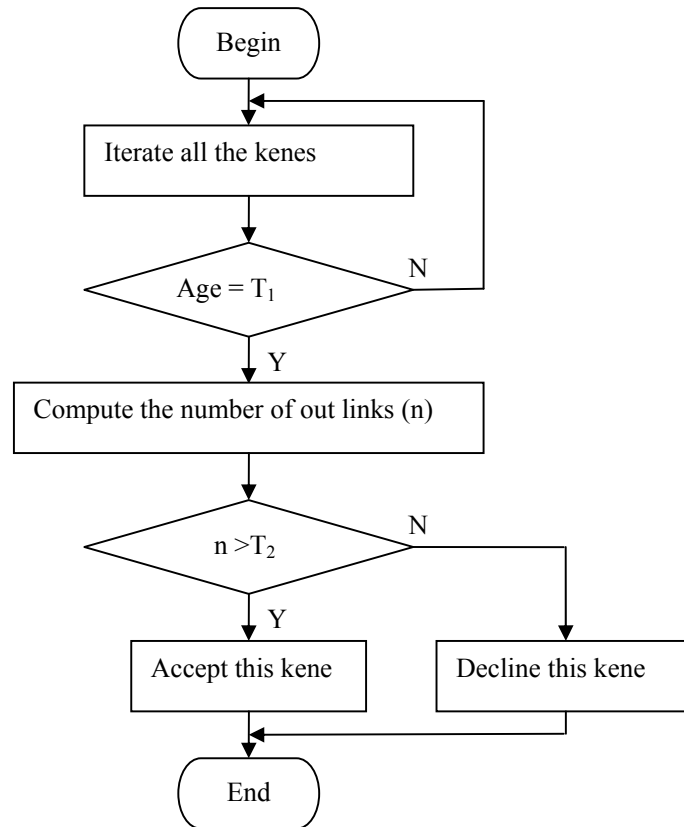


Figure 3.10 Flowchart of decision process of utility-oriented community

As far as utility-oriented community is concerned, only one evaluator exists. At each time interval, the evaluator iterates all the kenes in the domain to find out those kenes that exist in the domain for just a predefined time. Then the evaluator calculates the number of out links for every kene that is T_1 years old. If the number is greater than T_2 that is also a predefined value, the kene will be retained in the domain so that this kene could be used to create new kenes by other agents in the future. On the other hand, the kene with the number of out links less than T_2 will be removed from the domain.

There is one thing that needs to be noticed. The out links of the kene are built when others kenes cite this kene. So the greater the number of out links of a kene, the more impact the kene has. In other words, the kene is more important in this domain.

3.5.2.3 Service-oriented

Table 3.5 Parameters of service-oriented community

Community Type	Characters	
Service-oriented	Recruitment Selectivity	Low
	Growth Rate	Moderate
	Turnover	High
	Decision-making Style	Council

Under the decision style of council, whether or not to accept a new kene is determined by several independent evaluators. If majority of the evaluators accept this new kene, then it is retained in the domain. Otherwise the kene is declined. The flow chart in Figure 3.11 describes the process.

In future versions, the evaluator can be members of the community, and probability for acceptance can be a function of the path length between the two kinds of agents.

In a service-oriented community, there exist multiple evaluators who are in charge of the task to determine whether to accept or reject a new kene. When a new kene is introduced by an agent, all these evaluators will give their own judgments independently. The process of each evaluator to evaluate the new kene is the same as that in the exploration-oriented community. After all the evaluators finish the process of evaluation, the new kene is accepted by the domain if more than half of the evaluators agree to accept it. Otherwise the kene is removed.

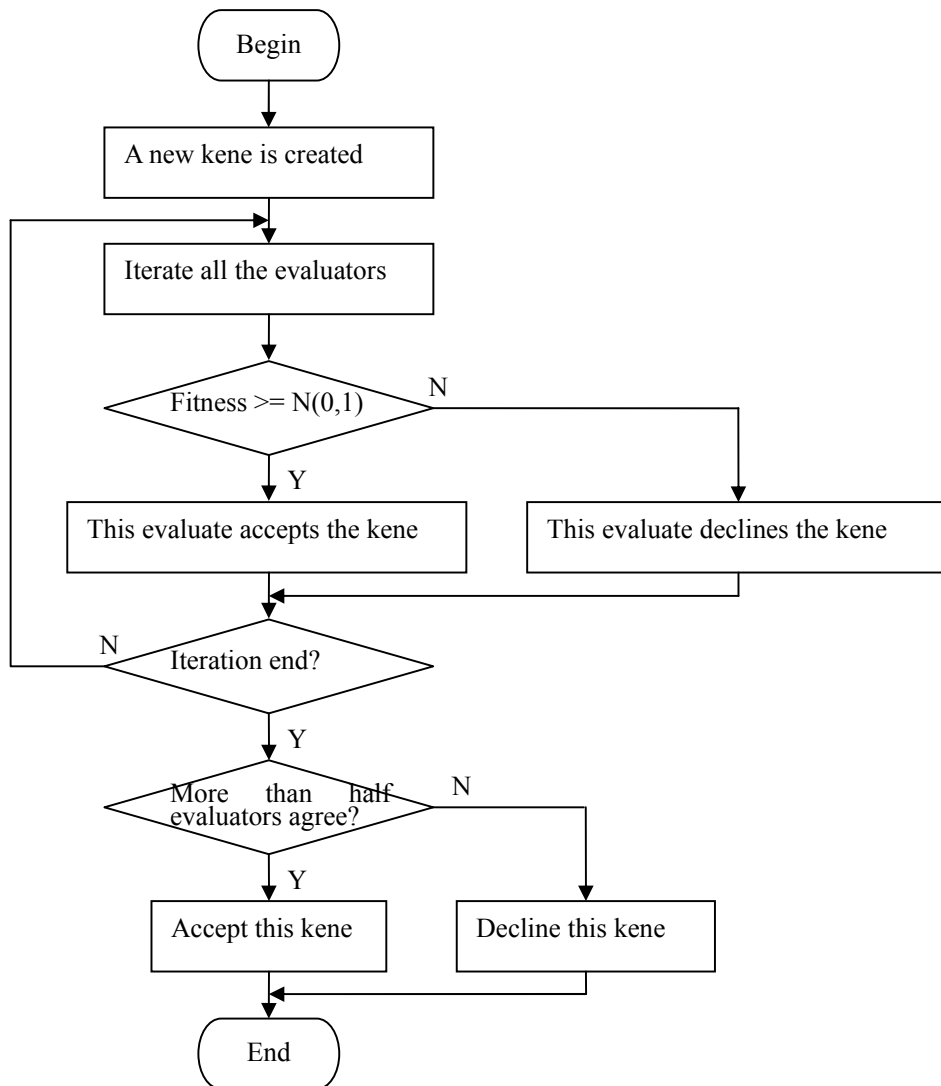


Figure 3.11 Flow chart of decision type of service-oriented community

CHAPTER 4

Implementation of Simulation Model

4.1 Introduction of Repast

According to the website of Repast, Repast is an acronym for the Recursive Porous Agent Simulation Toolkit that is a free and open source agent-based modeling toolkit that simplifies model creation and use. Repast Symphony provides a rich variety of features including the following.

- The model development can use pure Java, Groovy, flowcharts and any mixture of them.
- A pure Java model execution environment includes built-in results logging and graphing tools that make it easy to change the appearances of agents.
- The context is based on a flexible hierarchy that can realize the modeling and visualization of 2D environments and 3D environments.
- The discrete event scheduler is fully concurrent multithreaded.
- All the models developed by Repast are object-oriented.

In general, the standard model using Repast is based on contexts and projections [34].

4.1.1 Contexts

The core data structure in Repast is called a Context and all agents must be in a context. From a modeling perspective, the Context represents an abstract population. The objects in a Context form the population of a model. Although the context doesn't provide mechanisms to implement relationships between agents, it is an infrastructure to define the interactions of the populations.

In addition to maintaining the collection of proto-agents, the Context also holds its own internal states that can consist of multiple types of data. This provides a way in which the agents can interact with the context and exchange information. In order to maintain these states, context also has behaviors associated with it.

Context is a hierarchical structure which means a parent context can have some sub contexts. Different sub contexts hold different internal states and the same agent could have different behaviors when it exists in different sub contexts. Also if an agent is in a sub context, it is certain it is in the parent context. On the other hand, the reverse is not true i.e. an agent in the parent context can not be in a sub context.

4.1.2 Projections

Projections are kinds of data structures used to define relationships between agents within a context. From a practical view, that Projections are added to a Context is to allow the agents to interact with each other. Projections have a many-to-one relationship with Contexts, which means each Context can have an arbitrary number of Projections associated with it. In other words, within each Context, the agents can have more than one type of relationship with one another.

There are some frequently used projections including grid, continuous space, network and geography. The following figure shows how context, sub context and projection interact.

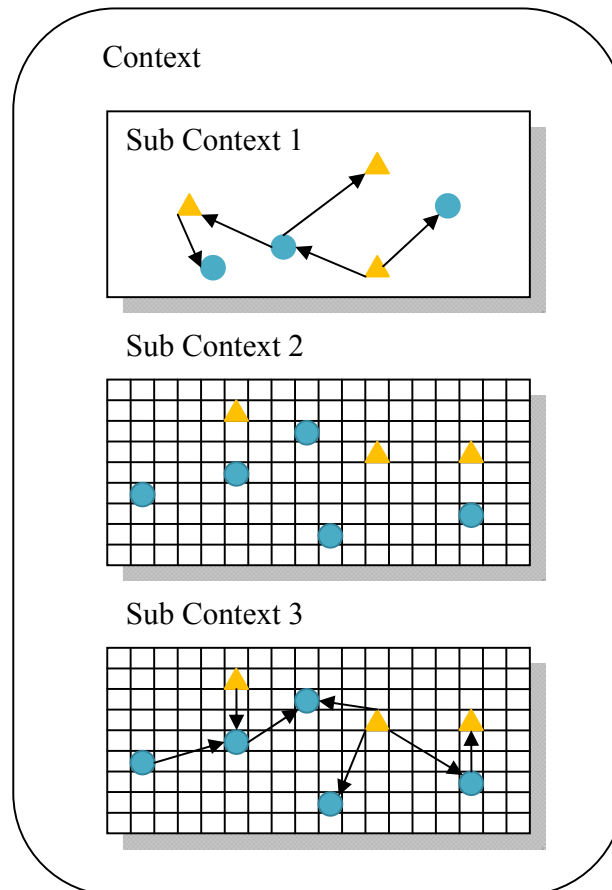


Figure 4.1 Contexts and projections

In Figure 4.1, the context has three sub contexts where sub context 1 only has network projection which indicates the relationship between agents and in sub context 2 the projection used is grid in which every agent occupies one cell that can be represented by a pair of coordinates. Furthermore, sub context 3 consists of the mixture of grid and network. As an agent in sub context 3, it can have two kinds of projections associated

with it. Finally, it is noticed that any agents existing in a sub context also belong to the parent context.

4.2 Implementation of Agents

According to Figure 1.2, there are three major objects: individual, evaluator and kene respectively. Kenes are created by individuals and are evaluated by evaluator. So kenes do not have any behaviors except that they own some state variables. In addition, individual and evaluator have a common super class that is called basic agent, because both of them have some same behaviors such as moving in the grid.

The Figure 4.2 presents the class view of these three major components in the model.

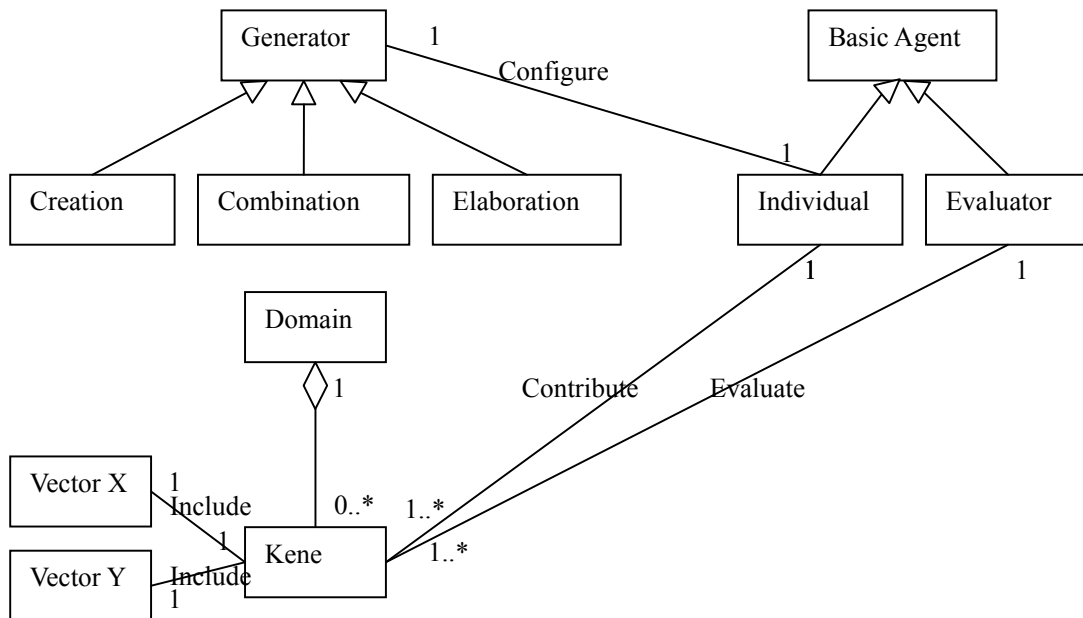


Figure 4.2 Class view of systems model of creativity

The following is the corresponding implementation of these four kinds of objects.

4.2.1 Basic Agent

The basic agent does not have any state variables, but it has two behavior methods including move and isValidPosition. Here the move method defines how an agent moves in the context at each time interval. The method of isValidPosition is used to judge whether or not a coordinate is in the range of the current context and it is useful for agents to move and for new kenes to be located in the context.

Table 4.1 Definition of basic agent

Type	Behaviors
Basic Agent	Move
	isValidPosition

4.2.2 Individuals

Three kinds of probabilities in Table 4.2 are corresponding to the frequency of using creation, combination and elaboration generators. Reputation represents the level of contribution of agents. When a kene created by an agent is accepted by the community, the agent's reputation will be up. Otherwise the reputation value would be down if his/her kene is rejected. The motivation is generalize reputation, which means the probability to make contributions. In other words, the higher an agent's reputation is, the more likely to provide his/her kenes. The time when the kene is created is useful to evaluate the result of enculturation. After a constant amount of time, a new agent needs to be assessed based on his/her enculturation level. Only those agents who reach a minimum threshold can stay in the community. Otherwise they have to leave. Tenure variable indicates if an agent has

passed the enculturation process successfully. Major represents what sub area an agent is interested in.

All these behaviors listed in Table 4.2 form the whole life cycle of an agent including entry and enculturation, innovation and turnover.

Table 4.2 Definition of individual

Type	State Variables	Behaviors
Individual	Probability to use creation generator	Create
	Probability to use combination generator	Combination
	Probability to use elaboration generator	Elaboration
	Reputation	Enculturation
	Motivation	Entry and turnover
	Time to be created	
	Tenure	
	Major	

4.2.3 Evaluators

In this model, three kinds of open scientific societies are studied, which includes exploration-oriented, utility-oriented and service-oriented. As far as a special society is concerned, it has the particular different decision style as discussed in Chapter 2.5.2. Weight for out links is used to calculate the fitness of the new kene. More links a kene has, more effects it does. Additionally the threshold for new kenes to be accepted is the minimum fitness value. Finally, the evaluator has only one behavior i.e. evaluation.

Table 4.3 Definition of evaluator

Type	State Variables	Behaviors
Evaluator	Decision style	Evaluate
	Weight for out links	
	Threshold for new kenes to be accepted	

4.2.4 Kenes

Although kenes are not active agents, they make up domain that is one of the three main components of this model. In addition the pattern of kenes determines the creativity of one society to some extent.

Table 4.4 Definition of kene

Type	State Variables	Behaviors
Kene	Length of kene	None
	Vector x	
	Vector y	
	Fitness	

The length of kene determines the complexity of a kene. Generally, the longer a kene is, the more complicated the kene is. Also the length of kene is consistent with the size of vector x and vector y, both of which are bases to calculate the location of kene. Fitness refers to the degree to which the kene is suitable for the society.

4.3 Implementation of Projection

In this model, two kinds of projections are used to represent the relationship between agents, which are continuous space and network.

4.3.1 Continuous Space

The continuous space is very similar as grid. The main difference between continuous space and grid is that the location of an agent is represented by floating point coordinates in continuous space rather than by integer coordinates as in grid.

In this model the codes to implement continuous space is as follows.

ContinuousSpace space =

ContinuousSpaceFactoryFinder.createContinuousSpaceFactory(null).createContinuousSp

```
ace("Simple Space", context, new RandomCartesianAdder(), new  
repast.simphony.space.continuous.StickyBorders(), gridWidth, gridHeight);
```

There are some major arguments needing user to set including name, context and the size of continuous space. In addition, users also can define their own functions how to set the initial location of agents.

At each time interval, both individuals and evaluators will move in the continuous space. In order to implement this motion, only one simple method is needed to invoke.

```
moveTo(T object, double... newLocation)
```

4.3.2 Network

As discussed in Chapter 2.4.1, there are three kinds of relationships between agents including the relationship between kenes, the relationship between individuals and the relationship between kenes and individuals. And these relationships are implemented by network in Repast.

The program to define a network is like below.

```
NetworkBuilder builder = new NetworkBuilder("RelationOfKenes", context, true);  
Network network = builder.buildNetwork();
```

Since there are three kinds of relationship, three independent networks are needed to represent them. The other two are RelationOfIndividuals and RelationBetweenKenesAndIndividuals besides RelationOfKenes.

For a developer, only three arguments are needed to be taken into considerations, which are name, context and directed. Because the relationships in this model are directed, the third argument is set to true.

When an individual creates a kene, a relationship will be build between them, which indicates the individual is the owner of this new kene. The program codes are like below.

```
network.addEdge(individual, kene);
```

If a new kene are combined from three other existed kenes, then the new kene will build a relationship with these three kenes. The program is like below.

```
network.addEdge(old kene, new kene);
```

At the same time, the owners of these related kenes are also needed to build a kind of relationship. It indicates that an individual cites others' products as reference.

```
network.addEdge(cited individual, current individual);
```

In addition, it is easy to get all the links associated with an agent.

```
network.getEdges(agent);
```

Sometime users may want only out or in links associated with a specific agent.

```
network. getInEdges(agent);
```

```
network. getOutEdges(agent);
```

4.4 Scheduler

There are basically two ways to work with the Repast Symphony Scheduler which determines the behavior of agents at each time interval.

4.4.1 Directly Schedule an Action

In this scenario, the user need get a schedule and tell it when and what to run. An example of adding an item to the schedule as follows:

```
ScheduleParameters params = ScheduleParameters.createRepeating(1, 2);  
schedule.schedule(params, agent, "move");
```

Firstly, an object of parameter for schedule is created, which defines the begin time and interval time. The example above initializes such a parameter that runs at the time 1 and runs once every two time units.

Secondly, the parameter defined will be sent to the scheduler of system along with the method that will be invoked when parameters are satisfied and the object in which the invoked method exists. Here, the method named as move in the object will be called per two time intervals from the time 1.

4.4.2 Schedule with Annotations

The Java 5 introduces several new and exciting features, one of which is annotation. For Repast, the typical case where the annotation is used is where action is defined. And the annotation tells the scheduler when and how often to invoke a method. This mechanism to define schedule is used in the model of this article.

The model in this article has two kinds of agents who will do something at each time interval. So in each of them, there is a method named step with annotation associated with it.

```
@ScheduledMethod(start = 1, interval = 1)  
public void step() {...}
```

The codes above let the scheduler of system know the step method will be called per 1 time interval from the beginning of time 1.

There is one thing worth notice that the method called by scheduler must be a public function. Otherwise a Java runtime exception will prompt.

4.4.3 Global Method

In the model of this article, it is necessary to update the individuals' rank based on their reputation. The purpose is to make sure that the individual with higher rank will get more opportunities to do contribution. So a global method has to be created, which will run before all the agents and will run only once at each time interval. And this method can't be in any agents. Otherwise this global method may be run as many times as the number of all agents. Thus such a global method is defined in the context builder. There are two reasons. One is that the context builder has only one handler associated with it. The other is that the context builder is in charge of initialization of continuous space, networks and agents, which makes it be at the position over all agents.

The codes to define the global method is similar as the directly schedule method discussed in the Chapter 3.4.1. The only difference from that is the argument of priority that is one parameter of class ScheduleParameters.

```
ScheduleParameters params = ScheduleParameters.createRepeating  
(1, 1, ScheduleParameters.FIRST_PRIORITY);  
schedule.schedule(params, this, "update");
```

The argument of FIRST_PRIORITY guarantees that the global method will run before all the scheduled method.

4.5 Output

Although Repast integrated simulation environment provides many useful and excellent tool kits to help developer implement their own purposes, general output of functions are not included in it. So we have to write some codes to reach the goal to show some useful information after the simulation model is end.

In order to finish this task, it is separated into two steps.

Firstly, we must know when the model will stop. Since the length to run the model is defined in the parameters panel of the Repast, we can get the information using codes like below.

```
endTick =
```

```
(Integer) RunEnvironment.getInstance().getParameters().getValue("stopTick");
```

Once the current time is equal to the stop tick, the model can be stopped by the command below.

```
RunEnvironment.getInstance().endRun();
```

And then we can invoke a method that does some analysis tasks. In our model, a dialog will be popped up to show some information like density and clustering coefficient. In this dialog window, there is a menu item linking to Guess software that can expertly analyze the pattern of nodes in the aspect of cluster.

CHAPTER 5

Verification, Validation and Evaluation

According to handbook of simulation, the evaluation for a simulation model consists of two levels, one is verification, and the other is validation that includes conceptual validation and operational validation. Conceptual validation means that the conceptual model is consistent with the real world. Operational validation refers to a test protocol to demonstrate the model outputs meet the requirement of real world. [35]

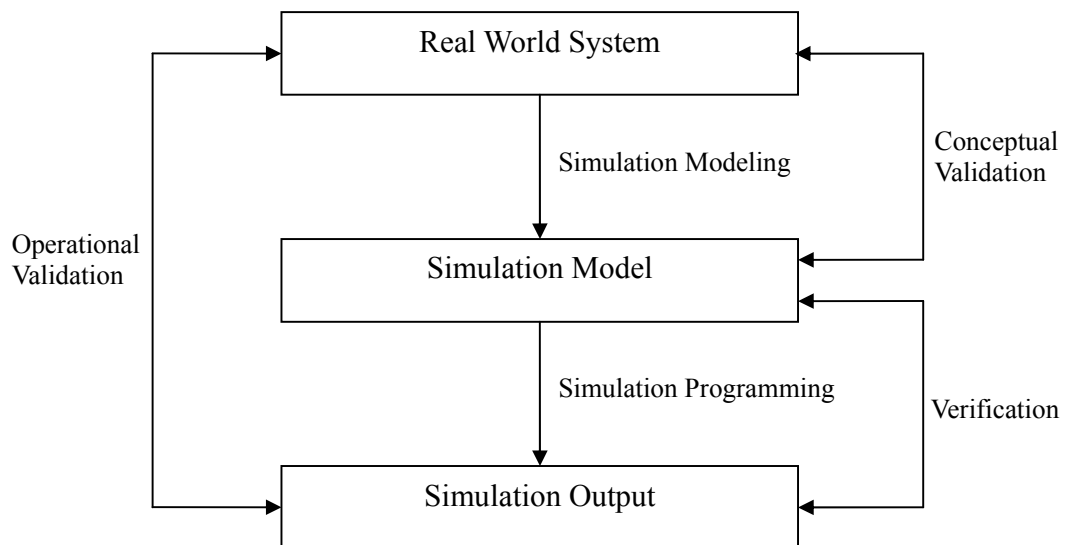


Figure 5.1 Overview of simulation model development [36]

Simulation modeling refers to the activity of driving the theoretical model from the real-world system. And the simulation programming refers to the activity that the computer based representation is derived from the model. There are two steps to check

whether or not the simulation program reflects the real world truly and fully, which are verification and validation.

5.1 Verification

Verification is the process of determining that a computer model, simulation, or federation of models and simulations implementations and their associated data accurately represents the developer's conceptual description and specifications. [37] In order to achieve the goal, unit test and integration test will be used.

5.1.1 Unit Testing

According to the Wikipedia, unit testing in computer programming is a software design and development method where the programmer gains confidence that individual units of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class.

Since our simulation uses object oriented programming language, Java, the unit testing is to assess the correctness of the method. In the simulation model there are three main classes, context builder, individual and evaluator. And each of them has a main entrance respectively. So unit testing focuses on these three methods.

1. Build function in the class of ContextBuilder

The build function is used to build the whole context of this simulation model such as continuous space and networks etc. So we check if the output is the same as our

expectation by variance of input arguments. For example, change the size of continuous space; change the initial location of agents and build relationships between different agents. Through these tests, we gain the confidence about the program itself.

2. Step function in the class of Individual

The step function in the class of Individual consists of move, enculturation and innovation. For the move function, we can change the coordinates of the next step to see if the function works well. And for the enculturation function, we can check the enculturation level when the individual meets different neighbors. In addition, for the innovation function, it is appropriate to change the probability of three kinds of generators to see if corresponding generator is invoked.

3. Step function in the class of Evaluator

Step function in the class of evaluator only has one role that is to evaluate whether or not to accept a new gene. So we can give it many genes with different fitness value and check if the evaluator responds the correct decision.

4. Results Dialog

The results dialog is used to show some useful information after the simulation mode is end, and it is independent from the framework of Repast. So we can use JUnit to finish this test. For each function of the dialog, we can use assert to check the return value is right. Also as far as the layout of dialog is concerned, we can insert a main function to see if it is elegant.

In addition, for each of tests above, debug is a good idea to track the flow process of single function step by step.

5.1.2 Integration Testing

Integration testing (sometimes called Integration and Testing, abbreviated I&T) is the activity [38] of software testing in which individual software modules are combined and tested as a group. It is between unit testing and system testing.

Integration testing takes what have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

In this scenario, we check the correctness of the simulation model by analysis of output data. The core class of this model is Individual. So we pay much attention on it. At each time interval, the program can out put some information into a log file, such as the used generator, whether or not his kene is accepted, and the total number of his contributions. Firstly, the number of every individual's contribution is none-decreasing function. Secondly, the new kene must be accepted when the number of contributions increases.

Therefore, we are confident with the simulation program through all the tests above.

5.2 Validation

Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended uses. [34]

There are two expected regularities about the scientific community: one is the slope of curve of kene number over time will be smaller and smaller with the increasing of

maturity of this community; the other is most of kenes are created by a small numbers of individuals.

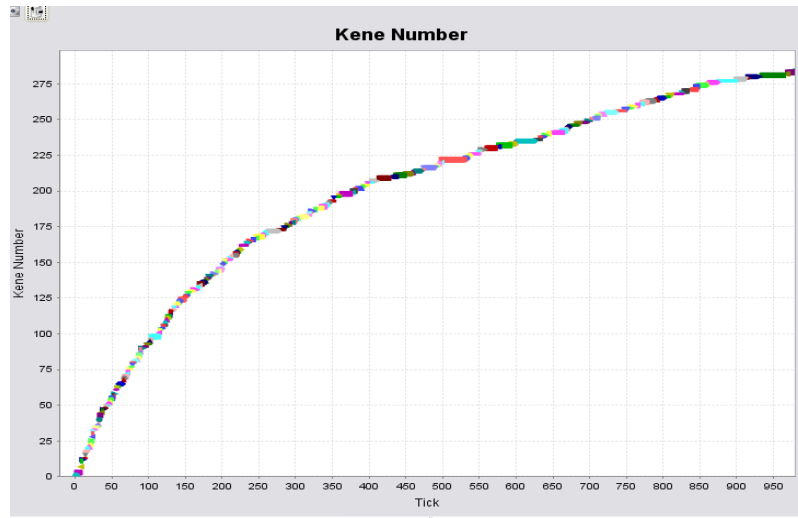


Figure 5.2 The number of kene over the time

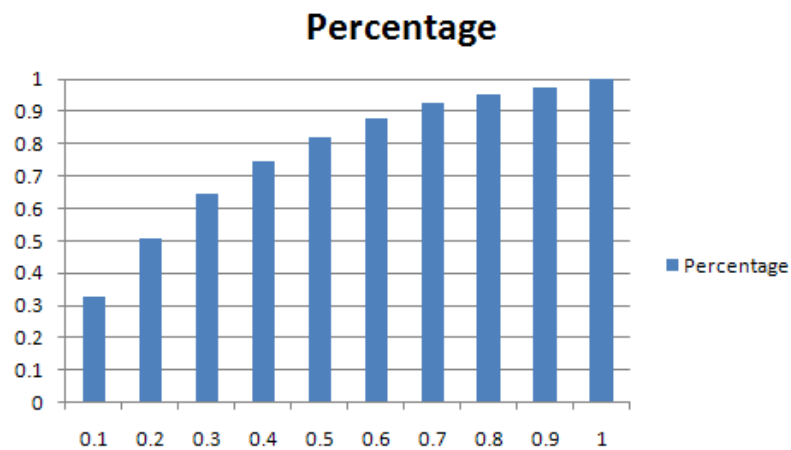


Figure 5.3 Percentage of kenes

In Figure 5.3, the x-axis is the percentage of individuals ordered by their ranks, and the y-axis is the corresponding percentage of kenes created by individuals. The top 20% individuals create more than 50% kenes. From Figure 5.2 and Figure 5.3, we can see the model is consistent with these two common senses above.

5.3 Experiments

The motivation for pattern-oriented modeling (POM) is that, for complex systems, a single pattern observed at a specific scale and hierarchical level is not sufficient to reduce uncertainty in model structure and parameters [39]. So we use batch method to get the mean value. The next section is about some metrics used to measure creativity.

5.3.1 Innovation Metrics

5.3.1.1 Impact Factor

In general, Impact Factor (IF) is frequently used as a metric for the importance of a sub-domain to its field. Impact Factor has the advantage over raw citation count that it is situated in time and accounts for changes in sub-domain importance over time [40]. Equation 5.1 represents the impact factor of sub-domain s for a given time frame t , which is 100 time steps in the model.

$$IF(s, t) = \frac{\text{\#citations from } D^t \text{ to } D_s^{t-1} \text{ or } D_s^{t-2}}{|D_s^{t-1}| + |D_s^{t-2}|} \quad 5.1$$

Where, D^t is the set of genes in time t , and D_s is the set of genes in sub-domain s .

5.3.1.2 Diffusion

However Impact Factor can't accurately reveal the importance of a specific gene. One of drawbacks of Impact Factor is whether a particular gene has broad or narrow impact. Does a highly cited gene dominate a prolific sub-field or does it have broad appeal and utility across many fields? In this section, we use sub-domain based impact measures that reveal more than citation count alone.

The metric for evaluating broad-based impact is Diffusion [40], defined for a given sub-domain s:

$$\text{Diffusion}(s) = H(P_s) = - \sum_{s'} P_s(s') \log P_s(s') \quad 5.2$$

$$\text{Where, } P_s(s') = \frac{\# \text{ citation from } D_{s'} \text{ to } D_s}{\# \text{ citations to } D_s}$$

Where, D_s is the kene set for a sub-domain s, and $D_{s'}$ is the kene set for a sub-domain s' .

5.3.1.3 Average Kene Fitness

The metric of Average Kene Fitness is used to evaluate the overall quality of kenes created by individuals. It is very important to assess the innovation of a community with combined use of the total number of accepted kenes. In general, the higher average kene fitness and the number of accepted kenes are, the more innovative a community is. Assume the total number of accepted kenes is N, then the equation to calculate the average kene fitness is as follows.

$$\text{Average Kene Fitness} = \frac{1}{N} \sum_{i=1}^N F_i \quad 5.3$$

Where, F_i is the fitness of kene i that includes individual fitness and relational fitness calculated in equation 3.13.

5.3.2 Network Metrics

5.3.2.1 Density

The density of the open science community network is defined as the ratio of the number of edges between individuals to the maximum number of possible edges, which indicates the cohesiveness of the community. The greater the density is, the higher the cohesiveness degree is. The equation of density of social network is equal to the number of edges between individuals divided by the number of all possible edges.

$$\rho = \frac{\# \text{ edges}}{\# \text{ all possible edges}} \quad 5.4$$

5.3.2.2 Clustering Coefficient

The clustering coefficient of a vertex in a graph quantifies how close the vertex and its neighbors are to being a clique (complete graph) [41].

A graph $G = (V, E)$ formally consists of a set of vertices V and a set of edges E between them. An edge e_{ij} connects vertex i with vertex j .

The neighborhood N_i for a vertex v_i is defined as its immediately connected neighbors as follows:

$$N_i = \{v_j : e_{ij} \in E \wedge e_{ji} \in E\} \quad 5.5$$

Thus, the clustering coefficient of vertex i for directed graphs is given as

$$C_i = \frac{|\{e_{jk}\}|}{k_i(k_i - 1)} : v_j, v_k \in N_i, e_{jk} \in E \quad 5.6$$

Where, k_i is the total degree of the vertex i . $|\{e_{jk}\}|$ is the number of edges among all the neighbors of vertex i .

The clustering coefficient for the whole system is given by Watts and Strogatz as the average of the clustering coefficient for each vertex:

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n C_i \quad 5.7$$

Where, n is the total number of the vertices in the graph.

The clustering coefficient is used to denote the degree of kenos and individuals who connect with each other. Also there are three different clustering coefficients which are related to kenos, individuals and all the agents respectively.

Clustering coefficient for technical network is only based on the network built on the kenos. And clustering coefficient for social network is in terms of the network of individuals. The last clustering coefficient regards total network as a whole.

5.3.2.3 Centrality

Within network analysis, the measure of centrality of a vertex determines the relative importance of a vertex in the graph. In this thesis, degree centrality is used to assess the social network. Degree centrality is defined as the number of links associated with a node. Then the average degree centrality is calculated as follows.

$$\bar{C} = \frac{1}{N} \sum_{i=1}^N C_i \quad 5.8$$

Where, N is the total number of vertices. C_i is the degree centrality of vertex i.

It is noteworthy that there may be multiple links among individuals because an individual can cite the kenos created by the same individual for several times. Under this situation, the weight of link between individuals reflects the citation times.

5.3.2.4 Proportion of Strong Ties

The difference between strong ties and weak ties is based on the number of links between vertices of the tie. In general, the more the number of links between two vertices is, the stronger the tie is. In this model, the strong ties are defined as those ties with the number of links greater than or equal to 2. The ties with the number of links equal to 1 are defined as weak ties. Proportion of strong ties a vertex is equal to the number of strong ties divided by the total number of ties associated with it. The average proportion of strong ties over the whole population of agents is equal to the total number of strong ties over the total number of ties in the network, which is shown in Equation 5.9.

$$\bar{R} = \frac{\# \text{ Strong Ties}}{\# \text{ Strong Ties} + \# \text{ Weak Ties}} \quad 5.9$$

5.3.3 Sensitivity Analysis

Sensitivity analysis (SA) is the study of how the variation (uncertainty) in the output of a mathematical model can be apportioned, qualitatively or quantitatively, to different sources of variation in the input of a model [42].

5.3.3.1 Experiments with Concept Creation Operator

In the section we present the influence of creation operator. Figure 5.4 represents the situation with the length of kene at 10.

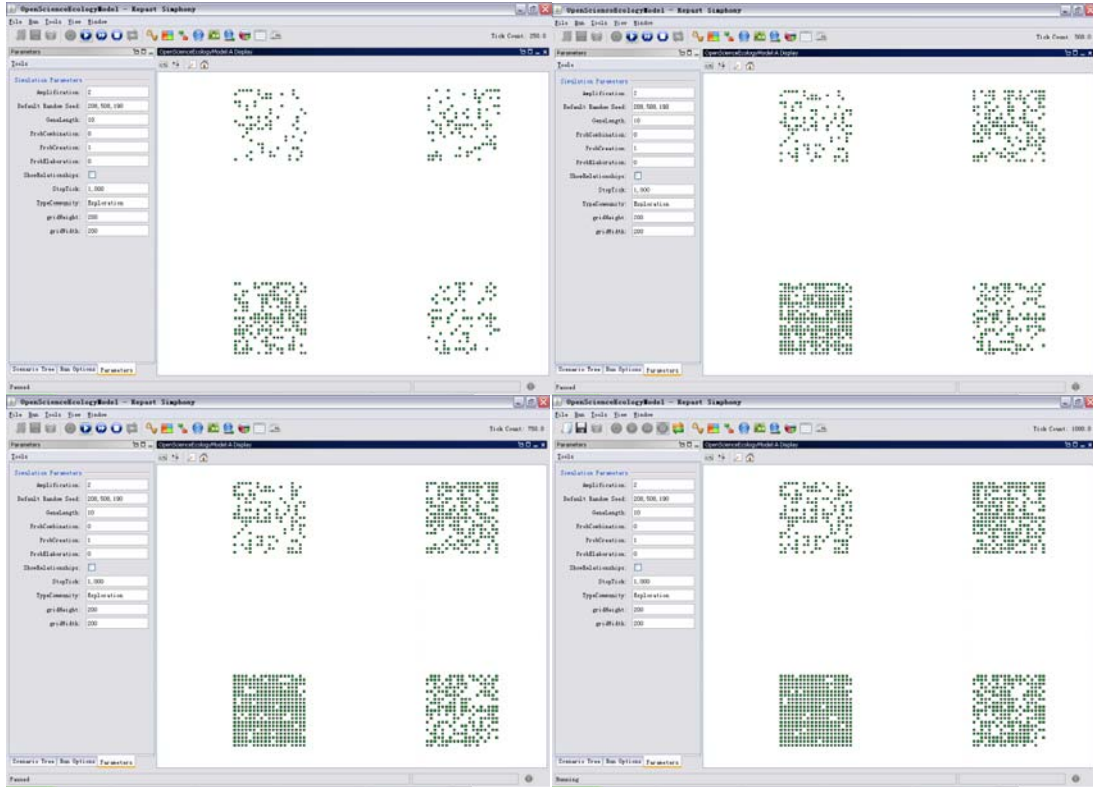
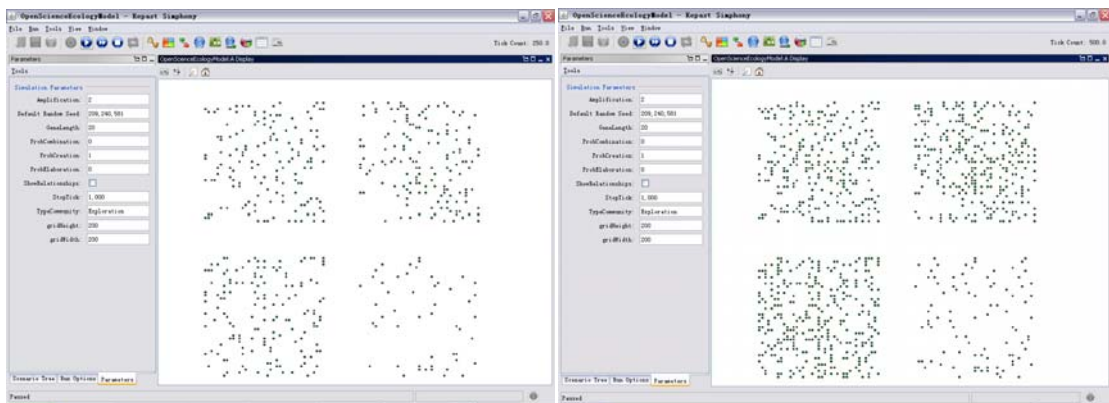


Figure 5.4 The pattern of kenes with the size of 10
 In this graph, there are four clusters due to the four sub domains for a new kene.

Each cluster is a square whose length of side is $2 * 10 = 20$.

If the length of kene is set to 20, the situation of kene pattern is like the figure below.



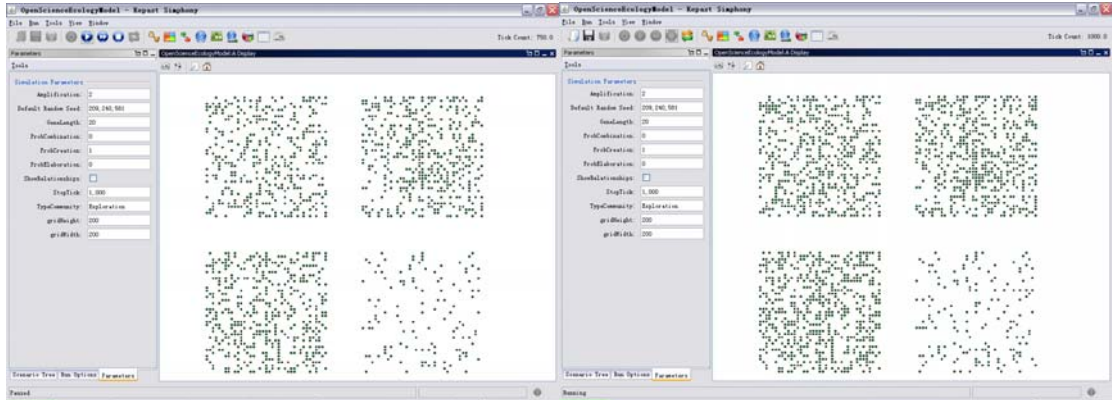
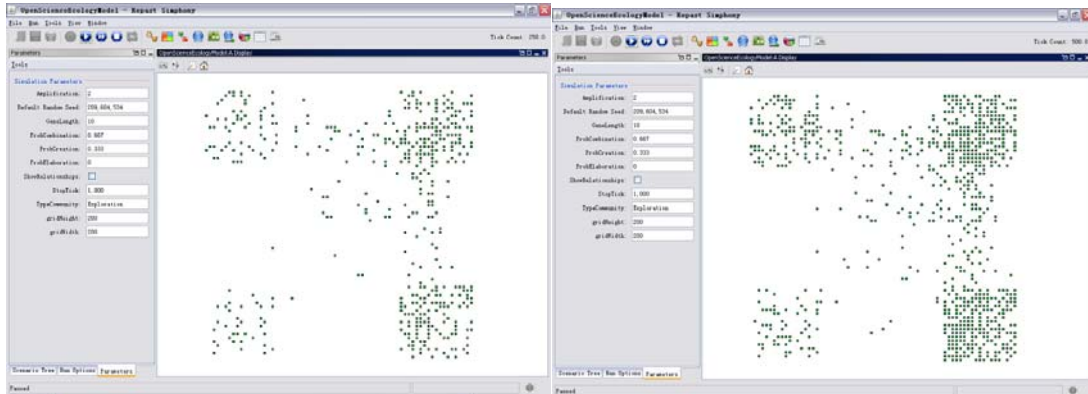


Figure 5.5 The pattern of kenes with the size of 20

From Figure 5.5, we can see that the space covered by kenes is only associated with the length of kene when only creation operator is used.

5.3.3.2 Experiments with Combination Operator as the Main Generator

When combination generator is used mainly and the length of kene is 10, the pattern of kenes looks like Figure 5.6.



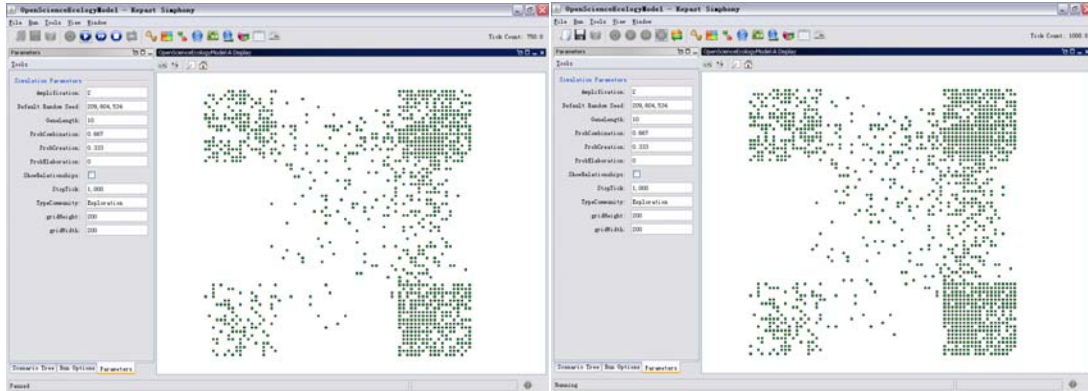


Figure 5.6 The pattern of genes with the length of 10

The combination operator is to generate new gene combined with some other existed genes. And the location of new gene is in the middle of these genes. So only using combination operator can not expand the occupied area.

Figure 5.6 is mainly using combination operator. Compared with only using creation operator, the basic characters of these two situations are similar. However the interspaces among the four clusters are occupied by genes, which shows the combination operator builds the bridge over different sub domains.

5.3.3.3 Experiments with Elaboration as the Main Generator

When mainly using elaboration operator, genes occupy more areas than that of only using one of other two operators.

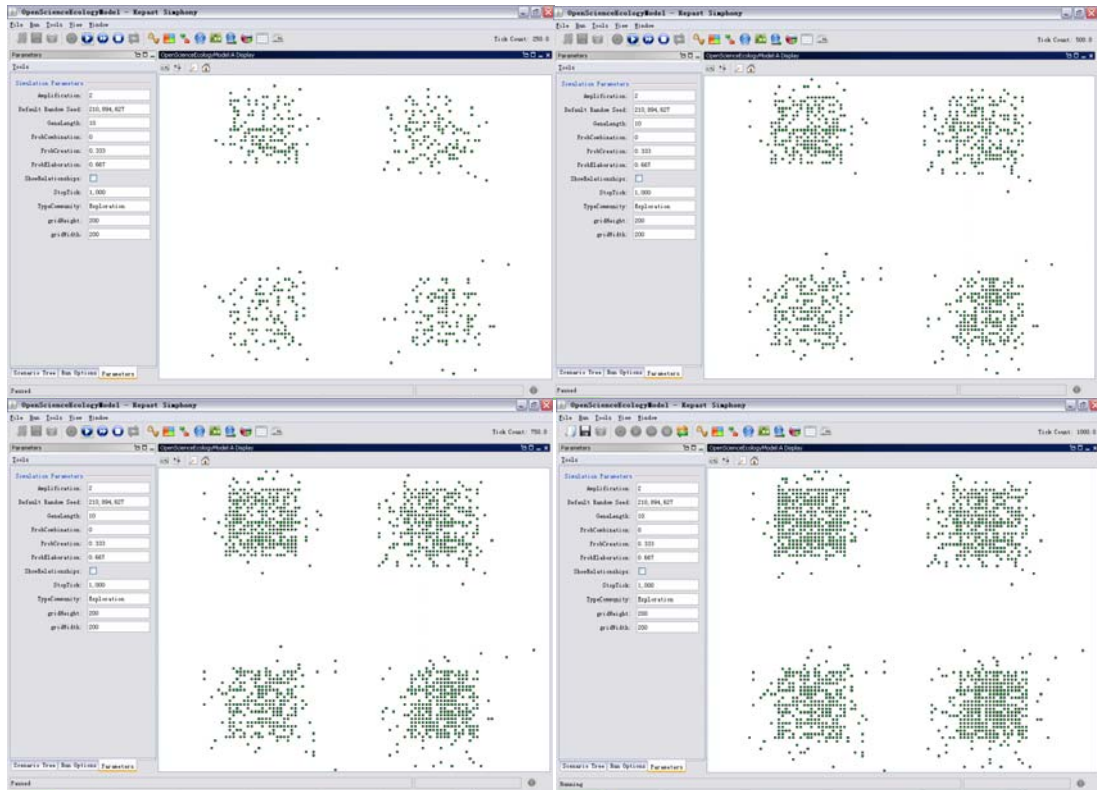


Figure 5.7 The pattern of kenes with mainly using elaboration

From Figure 5.7 we can conclude the elaboration operator makes great contribution to extend the covered space. In another word, elaboration plays an important role on innovation. However, extensive divergence may mean lack of coherence, so balance is needed.

5.3.3.4 Experiments with Equivalent Probability for Three Generators

The experiment is to use three generators with equivalent probabilities, i.e. the probabilities of using each of them is 0.33333. Figure 5.8 depicts the situation when probabilities of using three generators are equivalent with each other.

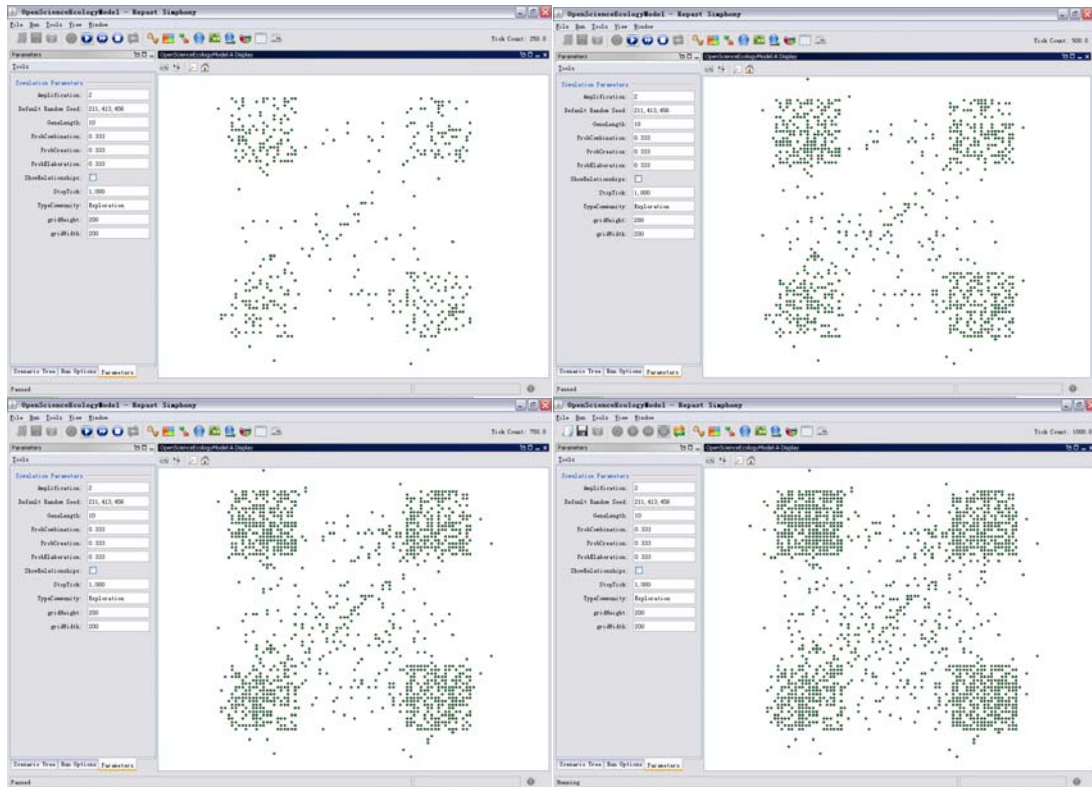


Figure 5.8 The pattern of kynes with balance

From Figure 5.8, we can see the scattering degree is moderate compared with other experiments.

5.3.3.5 Experiments with Different Predefined Parameters of Communities

Characters that distinguish three types of communities with each other are growth rate, recruitment selectivity, turn over and decision style. Three of them are parameterized, which are growth rate, recruitment selectivity and turnover. The experiment is to analyze the impacts on network metrics (i.e. density, clustering coefficient and centrality) with respect to different combinations of input parameters.

Table 5.1 summarizes the range of parameters for three kinds of communities. Growth rate indicates a random number of agents belonging to $U(0, \text{growth rate})$ enter the community at each time interval. Recruitment selectivity represents the threshold for a new agent to pass enculturation process. If the enculturation level is greater than the threshold, the new agent becomes tenured. Otherwise the agent leaves the community. Turnover is the threshold for an agent to leave the community. If the reputation is less than the threshold, the tenured agent leaves the community. In addition, the relative relationship among these three kinds of communities stays unchanged, which means the growth rate of utility-oriented community is equal to the growth rate of exploration-oriented community plus 2, and the growth rate of service-oriented community is equal to the growth rate of exploration-oriented community plus 1. Also the recruitment selectivity of utility-oriented community is equal to the recruitment selectivity of exploration-oriented community minus 0.1, and the recruitment selectivity of service-oriented community is equal to the recruitment selectivity of exploration-oriented community minus 0.2. Moreover the turnover of utility-oriented community is equal to the turnover of exploration-oriented community plus 0.1, and the turnover of service-oriented community is equal to the turnover of exploration-oriented community plus 0.2.

Table 5.1 Range of parameters for communities

	Exploration	Utility	Service
Growth Rate	[1.5, 5.5]	[3.5, 7.5]	[2.5, 6.5]
Recruitment Selectivity	[0.3, 0.5]	[0.2, 0.4]	[0.1, 0.3]
Turnover	[0.1, 0.3]	[0.2, 0.4]	[0.3, 0.5]

Through experiments the impacts of these three predefined parameters are summarized below. The impacts on outputs consist of two components: density and

centrality in that these two parameters play a very important role on the potential for creativity.

- Impacts of growth rate

Impacts of growth rate on density are shown in Figure 5.9.

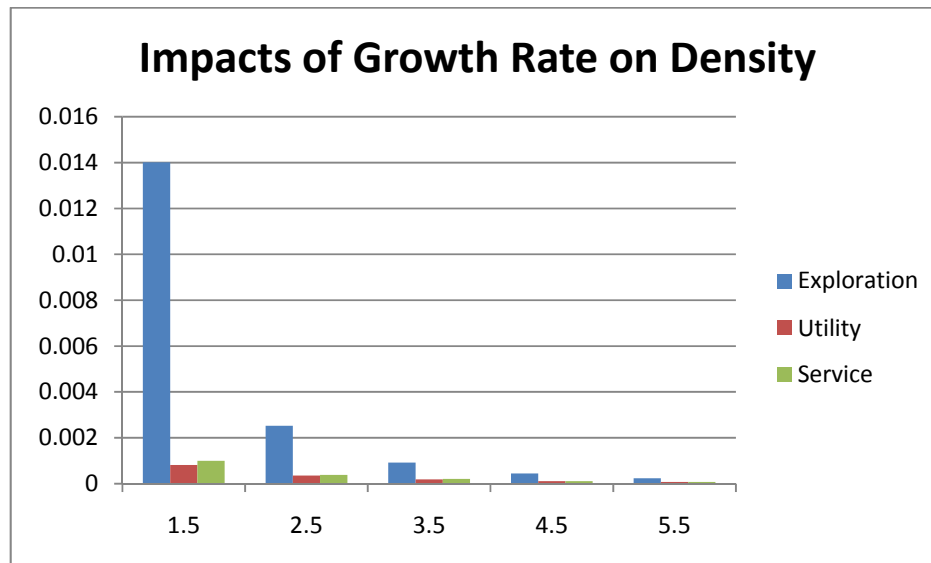


Figure 5.9 Impacts of growth rate on density.

From Figure 5.9, we can see the density of all three kinds of communities decreases with growth rate increasing.

Impacts of growth rate on centrality are shown in Figure 5.10. The x-axis of the figure is the parameter values of exploration-oriented community. The growth rate of utility-oriented community and service-oriented community are based on that of exploration-oriented community with the relationship discussed in the second paragraph of the section. The similar policy is applied in all the figures of comparisons of impacts of predefined parameters for communities.

From Figure 5.10, we can see the centrality of all three kinds of communities decreases with growth rate increasing.

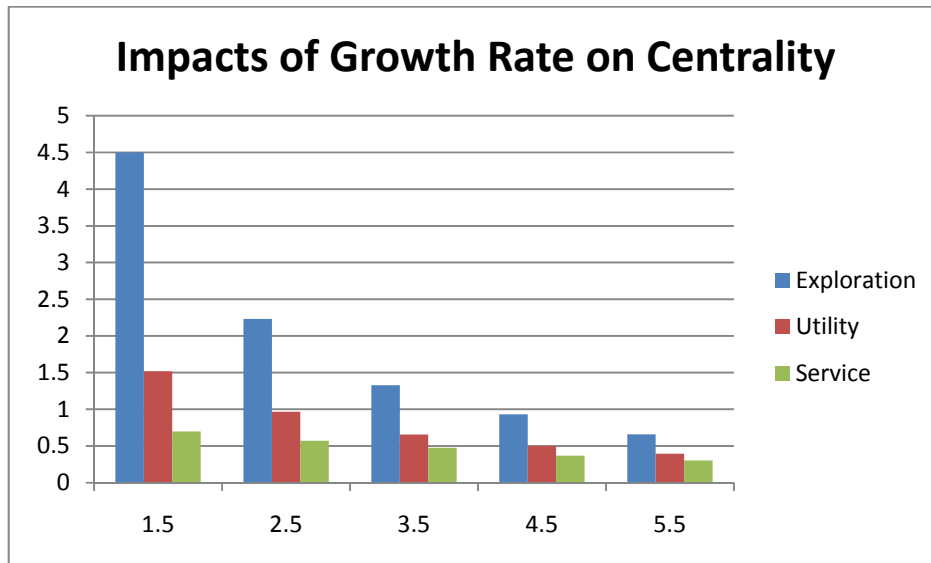


Figure 5.10 Impacts of growth rate on centrality

- Impacts of recruitment selectivity

Impacts of recruitment selectivity on density are shown in Figure 5.11.

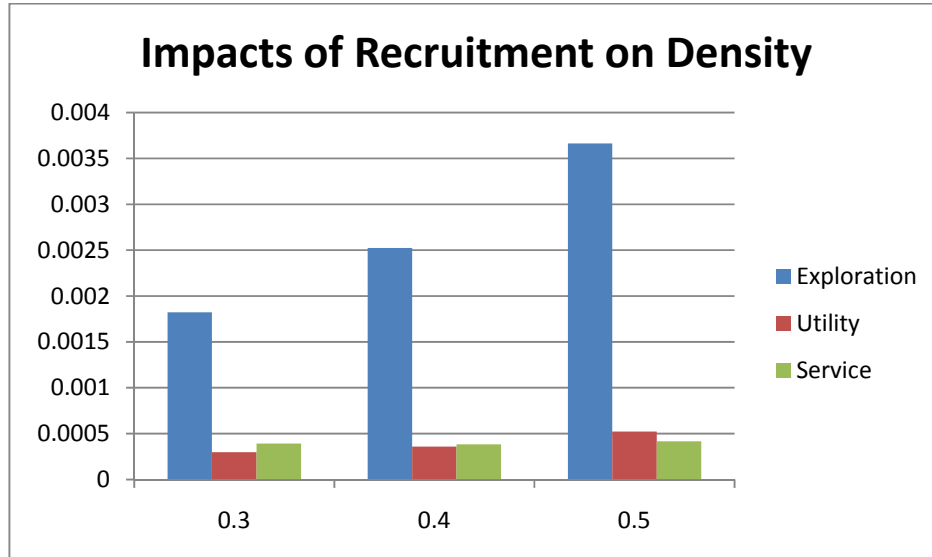


Figure 5.11 Impacts of recruitment on density

From Figure 5.11, we can see the density of all three kinds of communities increases with recruitment selectivity increasing.

Impacts of recruitment selectivity on centrality are shown in Figure 5.12.

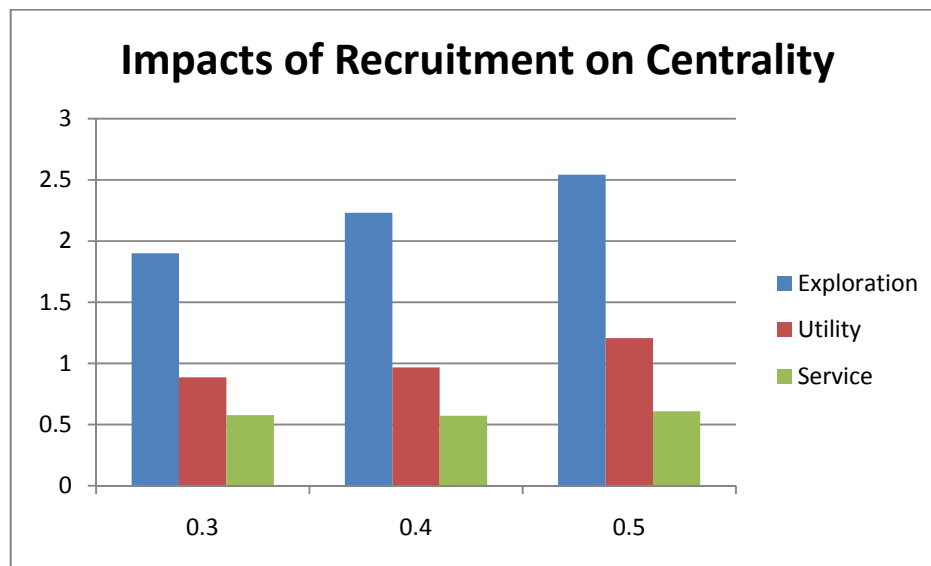


Figure 5.12 Impacts of recruitment on centrality

From Figure 5.12, we can see the centrality of exploration-oriented and utility-oriented community increases with recruitment selectivity increasing. However the centrality of service-oriented community almost stays unchanged with recruitment selectivity.

- Impacts of turnover

Impacts of turnover on density are shown in Figure 5.13.

From Figure 5.13, we can see the density of exploration-oriented community decreases with turnover increasing. However the density of utility-oriented community increases with turnover increasing. Additionally the density of service-oriented community vibrates with turnover.

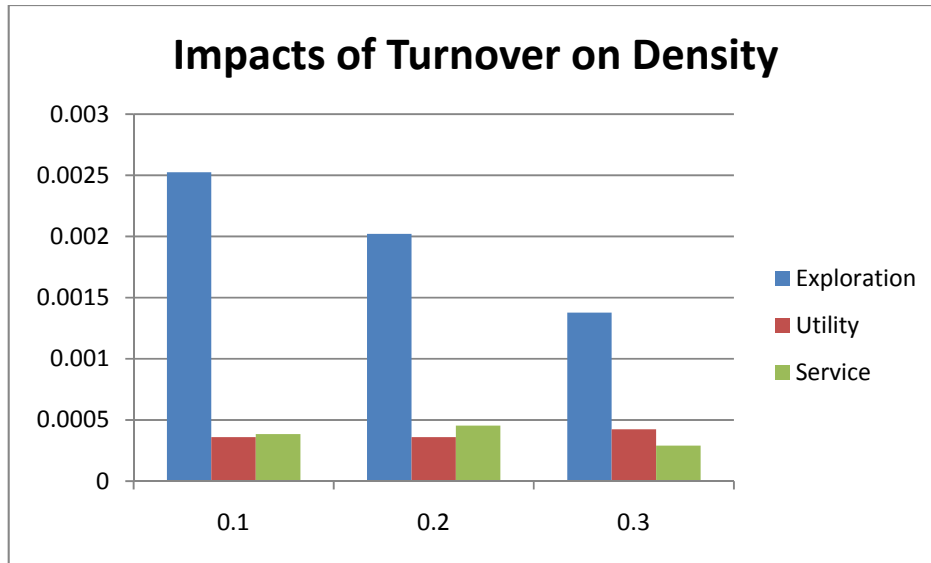


Figure 5.13 Impacts of turnover on density

Impacts of turnover on centrality are shown in Figure 5.14.

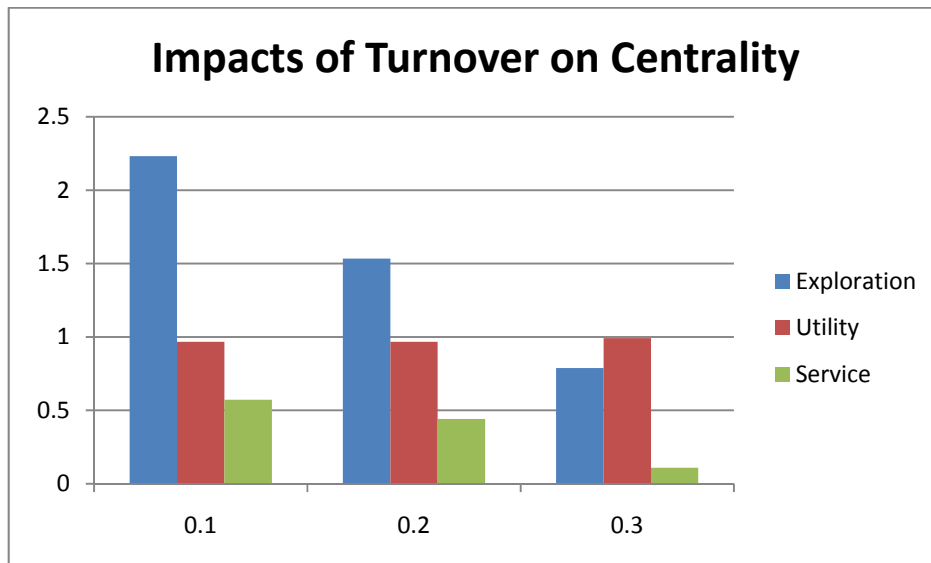


Figure 5.14 Impacts of turnover on centrality

From Figure 5.14, we can see the centrality of exploration-oriented and service-oriented decreases with turnover increasing. On the other hand the centrality of utility-

oriented community slightly increases with turnover increasing. There is an interesting phenomena emergent that is the centrality of utility-oriented community higher than exploration-oriented community when the turnover is 0.3.

Based on the sense that a community with low density and high centrality for social network yields more potential for innovation, much attention is paid in this situation in the model. Therefore growth rate is set to 2.5, and recruitment selectivity is set to 0.4. Additionally turnover is selected as 0.3 in that it leads to a kind of interesting phenomena as discussed in above paragraph.

5.3.4 Experiments between Different Types of Communities

Metrics including gene growth rates, clustering coefficient, diffusion and citation impacts are used to judge whether or not this simulation model can reflect the activities of real scientific societies.

When the simulation is completed, a results dialog pops up. This dialog shows the number of output links associated with a specific individual or gene. Through this analysis, we can see the genes published earlier have higher probability to be chosen to combination or elaboration. Similarly, the individual with higher motivation will produce more genes. This situation is the same as the higher participants perceived personal rewards for their Linux engagement; the more they were willing to be involved in Linux-related activities in the future [43]. It is also reflected by a case study of the development of the Apache web server showed that the top 15 programmers added 88% of the LOCs (Mockus et al., 2000). In contrast, the top 15 programmers for the GNOME project were only responsible for 48%, whereas the top 52 persons were necessary to reach 80%. A

clustering of the programmers based on the LOCs hinted the existence of a smaller group of 11 programmers within this larger group, who were still active. [44]

5.3.4.1 Exploration-oriented

1. Predefined parameters

The following are predefined parameters for exploration-oriented community, and three operators probabilities are equal with each other i.e. 0.333.

Table 5.2 Predefined parameters for exploration-oriented community

Parameters	Value	Description
Recruitment selectivity	0.4	If the enculturation level is greater than 0.4, the new agent becomes tenured. Otherwise the agent leaves the community.
Growth rate	2.5	At each time interval, a random number of agents belonging to $U(0, 2.5)$ enter the community.
Turnover	0.3	If the reputation is less than 0.3, the tenured agent leaves the community.

2. Results

Table 5.3 includes experimental results such as density, diffusion, centrality and clustering coefficients.

Table 5.3 Results on exploration-oriented community

Metrics	Value
Accepted Kenes	1387
Average Kene Fitness	0.656
Average Diffusion for Kenes	1.038
Density for Social Network	0.001
Ratio of Strong Ties to Weak Ties	0.043
Centrality for Social Network	0.709
Clustering Coefficient for Social Network	0.092
Clustering Coefficient for Social-Technical Network	0.384

Figure 5.15 indicates the total number of individuals who create the same amount of kenos. The x-axis represents the number of kenos, and the y-axis represents the total

number of individuals producing the corresponding number of kenes. From Figure 5.15, we can see most of individuals only generate few kenes. In addition, the number of agents decreases with the number of kenes increases.

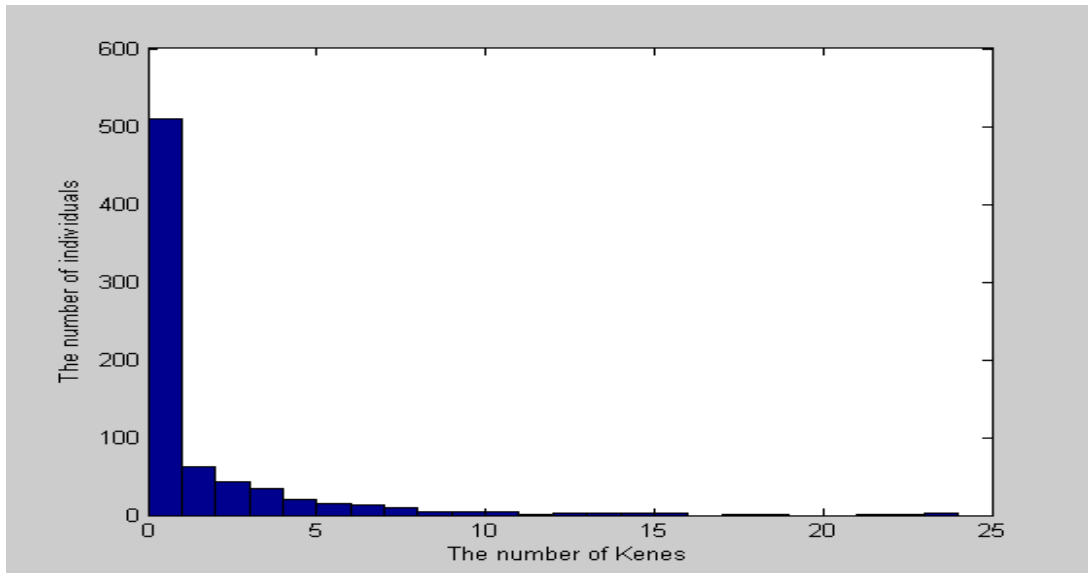


Figure 5.15 Histogram of the number of agents who create the same number of kenes

Figure 5.16 indicates the number of kenes created by each individual ordered by the rank of individuals. And the rank of individuals depends on their motivation levels. The x-axis of histogram below is the No. of individuals and the y-axis is the number of corresponding kenes created by individuals.

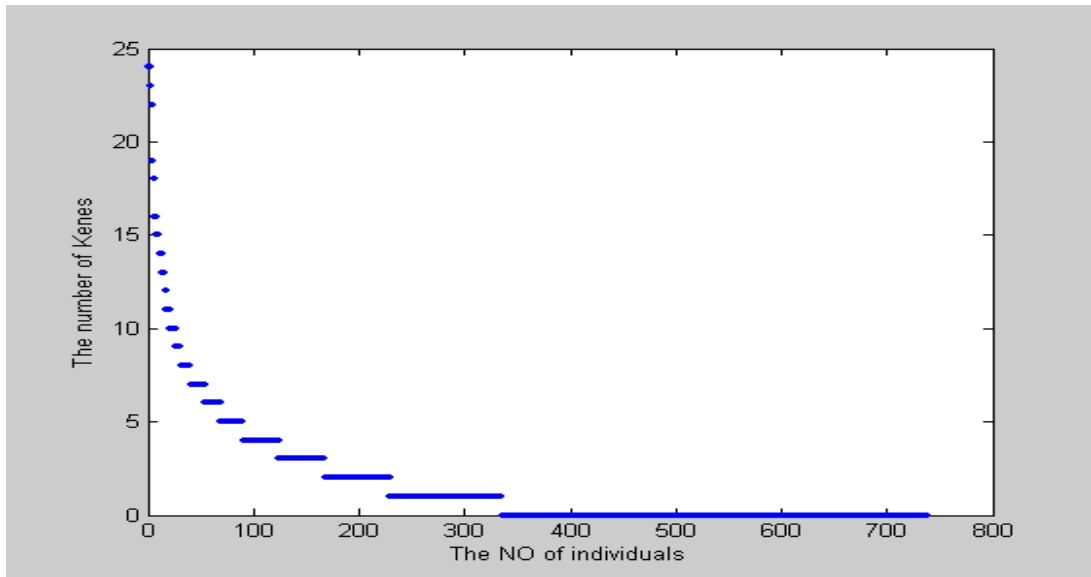


Figure 5.16 Plot of the number of kenes created by each agent

From Figure 5.16 we can easily find the core members make very greater contributions to this community than those common members do.

When GEM (Graph Embedding) style is selected, the graph is shown in Figure 5.17.

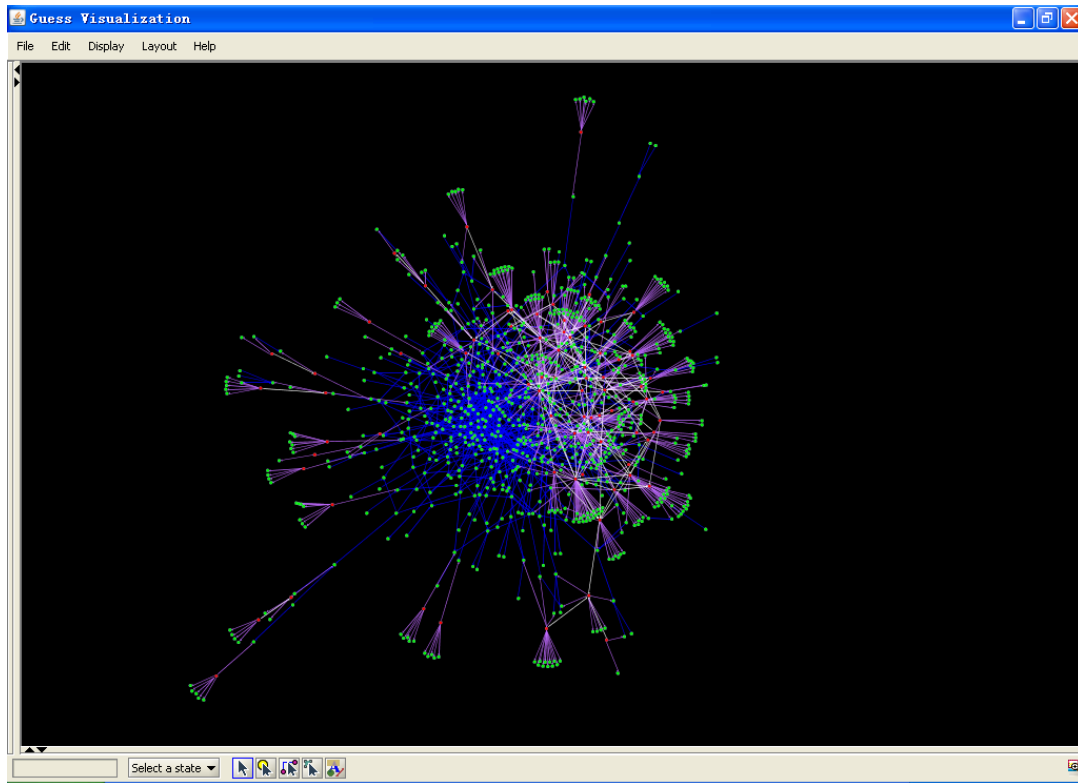


Figure 5.17 Emergent pattern in GEM layout

Figure 5.17 shows a pattern of cluster, where green nodes are kenens and red nodes are agents. In addition, purple lines represent the relationship who creates which kenens. Blue lines represent the citation relationship between kenens. Also the relationships among agents are denoted by white line.

Figure 5.18 shows the situation of accumulated impact factor changing over time where the unit scale of x-axis is 100 time ticks. The equation of impact factor is presented in equation 5.1.

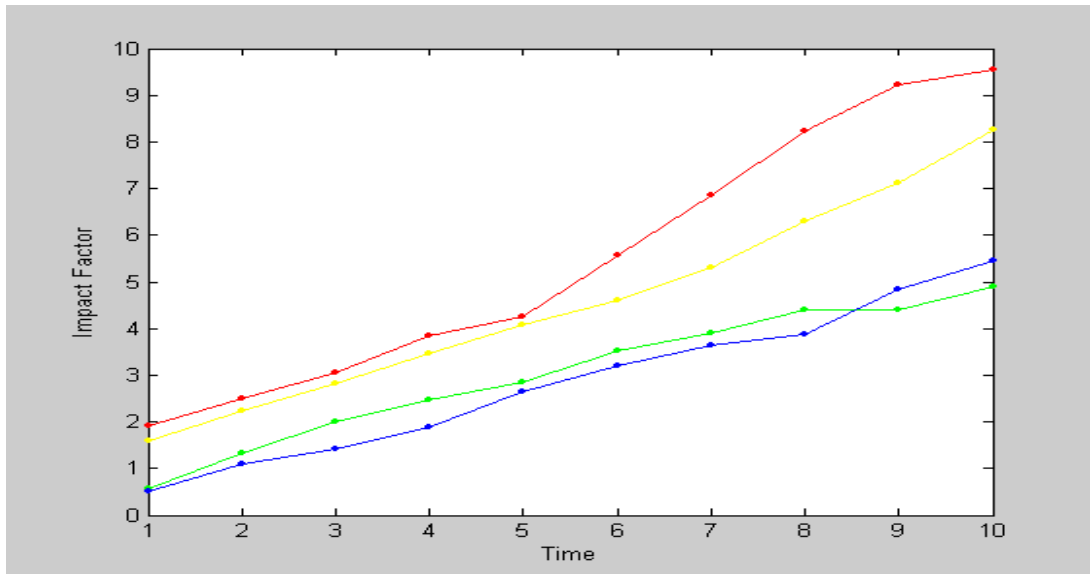


Figure 5.18 Impact factor over time on exploration-oriented community

The slope of trend of impact factor almost keeps the same over time. It shows the degree to which the people pay attention on this field doesn't change with the time.

5.3.4.2 Utility-oriented

1. Predefined parameters

The following are predefined parameters for utility-oriented community, and three operators probabilities are equal with each other (i.e. 0.333).

Table 5.4 Predefined parameters for utility-oriented community

Parameters	Value	Description
Recruitment selectivity	0.3	If the enculturation level is greater than 0.3, the new agent becomes tenured. Otherwise the agent leaves the community.
Growth rate	3.5	At each time interval, a random number of agents belonging to $U(0, 3.5)$ enter the community.
Turnover	0.4	If the reputation is less than 0.4, the tenured agent leaves the community.

2. Results

Table 5.5 records the results on utility-oriented community. Compared with exploration-oriented community, the total knowledge and the degree of clustering are much fewer. The comparison of these three kinds of communities will be discussed in detail in section 5.3.4.

Table 5.5 Results on utility-oriented community with equal probabilities

Metrics	Value
Accepted Kenes	748
Average Kene Fitness	0.829
Average Diffusion for Kenes	0.542
Density for Social Network	0.0004
Ratio of Strong Ties to Weak Ties	0.556
Centrality for Social Network	0.901
Clustering Coefficient for Social Network	0.057
Clustering Coefficient for Social-Technical Network	0.207

Figure 5.19 indicates the total number of individuals who create the same amount of kenos. From this graph, we can see that there is a huge gap between the individual who creates the most kenos and other individuals. In my opinion, a super core member emerges under this situation.

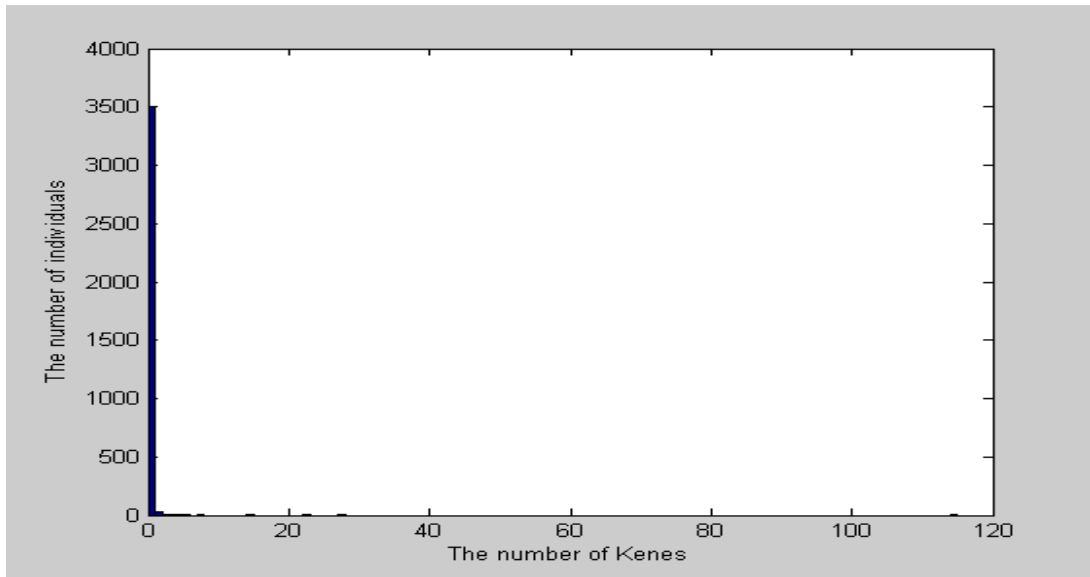


Figure 5.19 Histogram of the number of agents who create the same number of kenes

Figure 5.20 indicates the number of kenes created by each individual ordered by the rank of individuals. From it we also can see the No. 1 individual creates much more kenes than any of others.

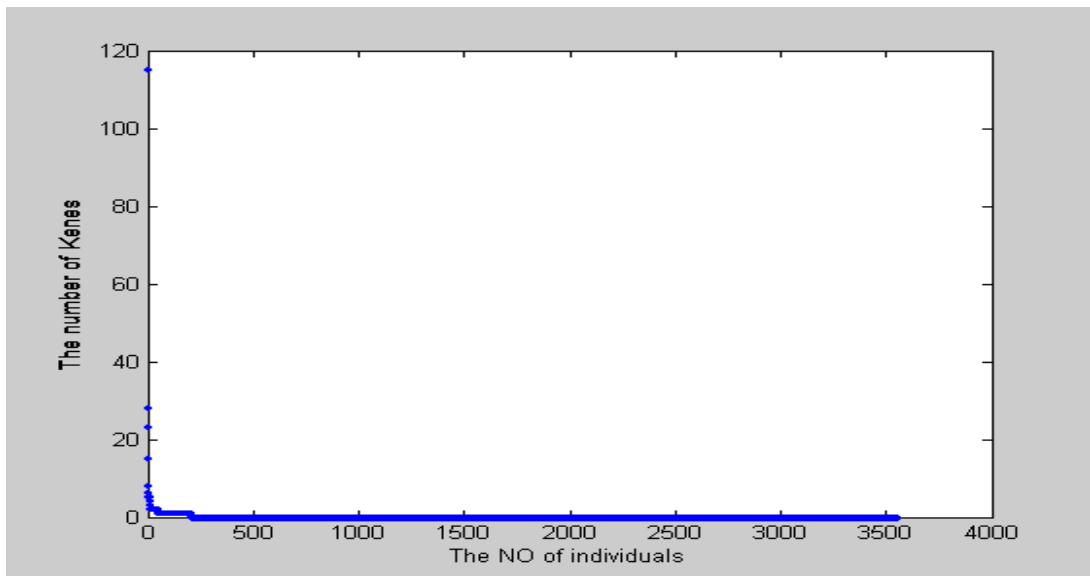


Figure 5.20 Plot of the number of kenes created by each agents

Figure 5.21 is under the style of GEM using Guess software. In this graph there are one chief member around whom other agents and kenes are.

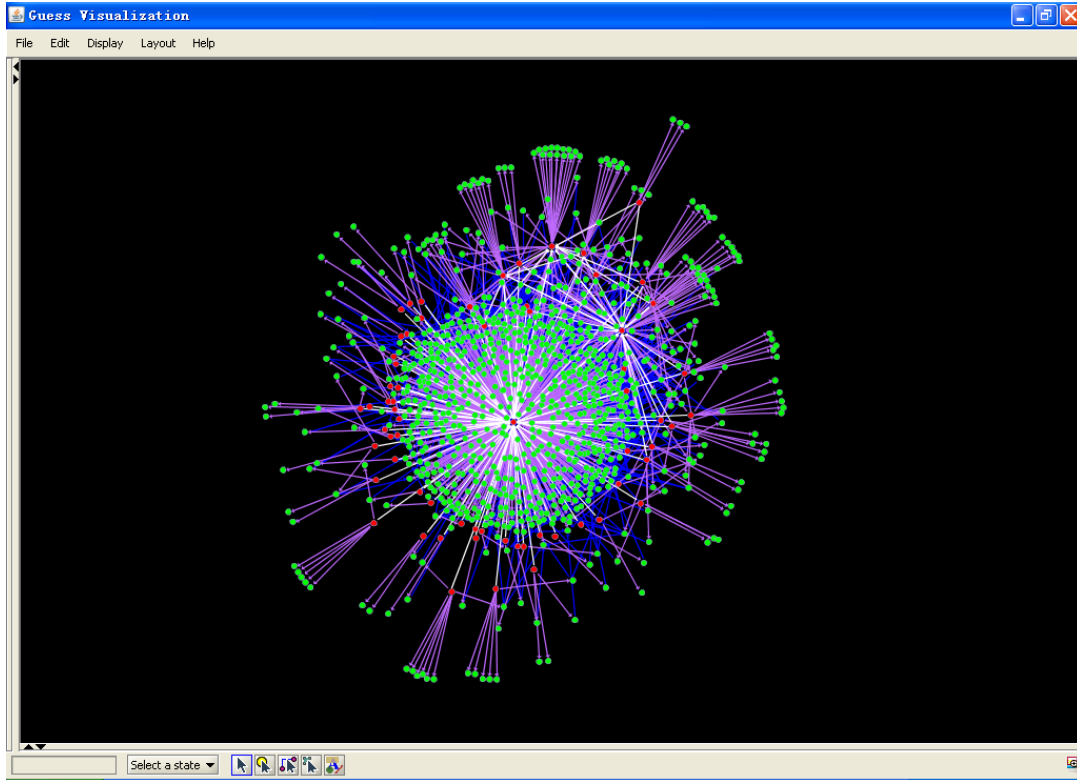


Figure 5.21 Emergent pattern in GEM layout

Figure 5.22 is the trend of accumulated impact factor changing over the time.

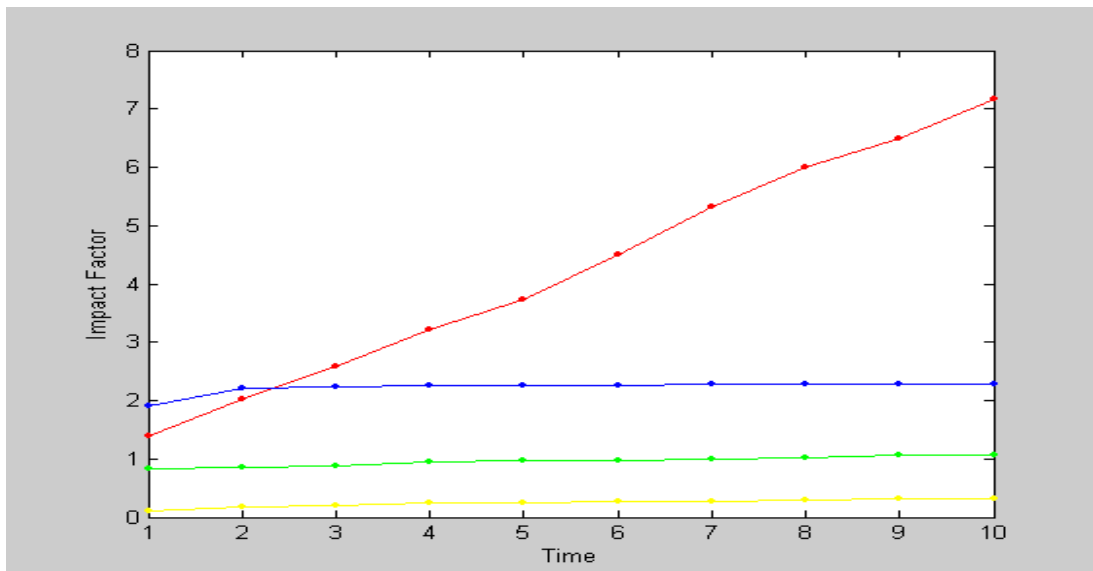


Figure 5.22 Impact factor over time on utility-oriented community

In the utility-oriented community with the predefined parameters, the impact factor decreases dramatically. Just after three hundred time intervals, the impact factors of three sub domains decrease to 0, which means that the research topics in this community have lost the interests of public.

Because the decision style in the utility oriented community is emergent selection which means the determination for kenes to accept is based on the references of kenes. The more references a kene has, the more likely this kene will be accepted. But any kenes can't be cited when it is just created. So the decision for a kene is postponed after a constant number of ticks that is 100 in this model.

In addition, the overall reference circumstance is highly related to three kinds of operators including creation, combination and elaboration. Thus the increase of probability of using combination and elaboration will lead to the corresponding increase of clustering. The experiment below is to increase the probability of combination and elaboration so that the sum of probabilities choosing these two operators is equal to 90% i.e. the probability of creation is 0.1, the probability of combination is 0.45 and the probability of elaboration is 0.45.

Table 5.6 is the summary of the simulation result after the probabilities of using three kinds of generators change. From this table, we can see all the metrics increase compared with Table 5.5. It shows that the probabilities of generators have great effects on the utility-oriented community. In other words, the operators of combination and elaboration play an important role on the expansion and sustainment of this community.

Table 5.6 Results on utility-oriented community with unequal probabilities

Metrics	Value
Accepted Kenes	1459
Average Kene Fitness	0.985
Average Diffusion for Kenes	0.494
Density for Social Network	0.001
Ratio of Strong Ties to Weak Ties	0.457
Centrality for Social Network	3.061
Clustering Coefficient for Social Network	0.066
Clustering Coefficient for Social-Technical Network	0.23

Figure 5.23 indicates the total number of individuals who create the same amount of kenes. From this figure, it is still very clear that some chief members emerge in this community.

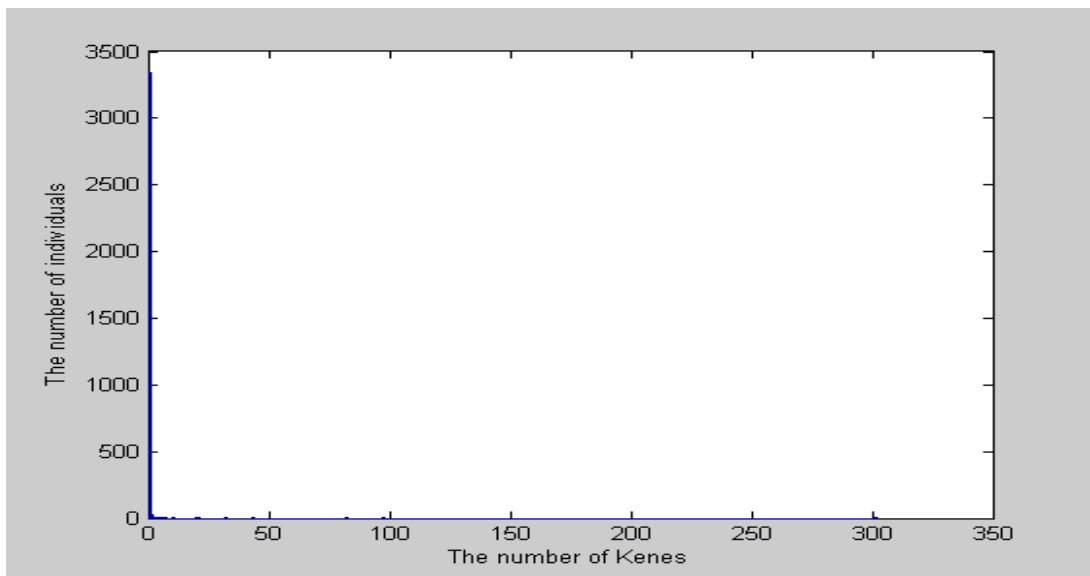


Figure 5.23 Histogram of the number of agents who create the same number of kenes

Figure 5.24 indicates the number of kenes created by each individual ordered by the rank of individuals. From it we also can see the No. 1 individual creates much more kenes than any of others.

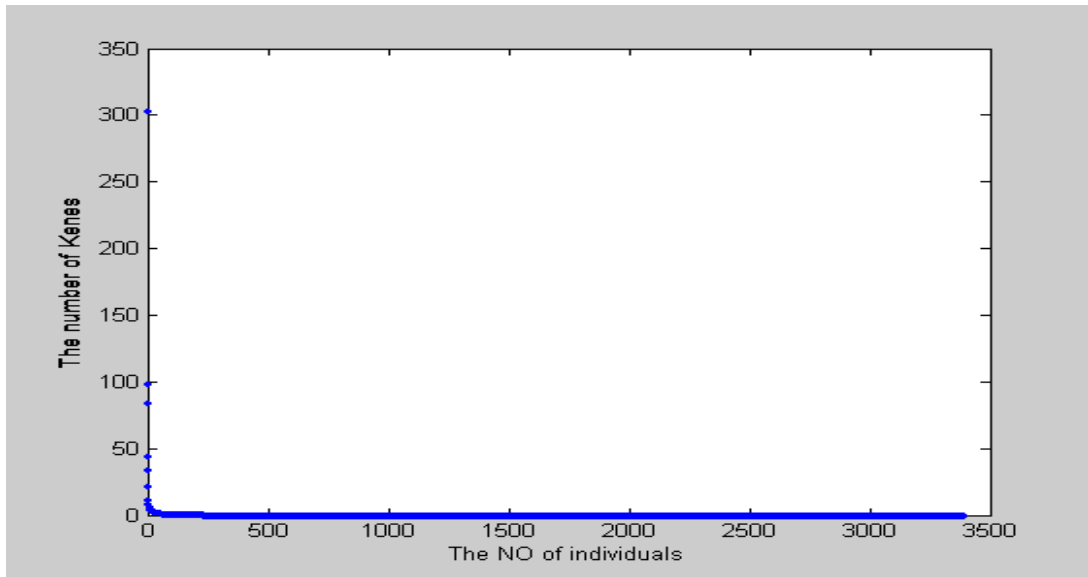


Figure 5.24 Plot of the number of kenes created by each agents

Figure 5.25 shows the pattern under the style of GEM. In this image there are three clear core members who make great contributions to the project.

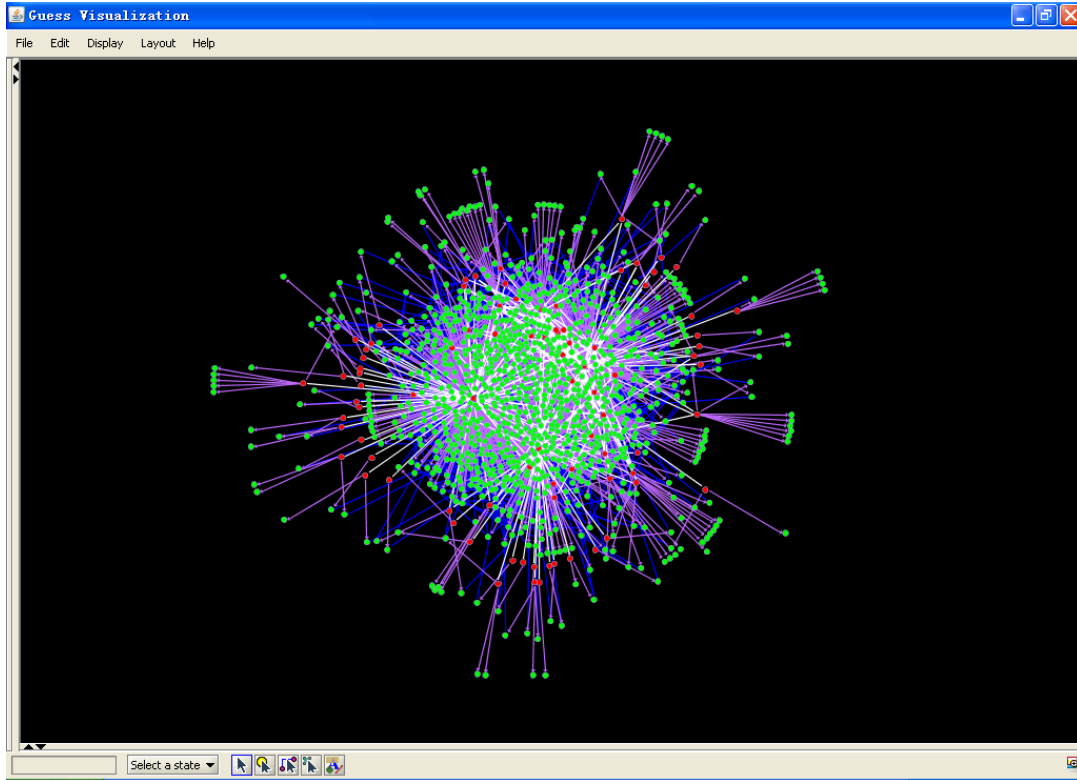


Figure 5.25 Emergent pattern in GEM style

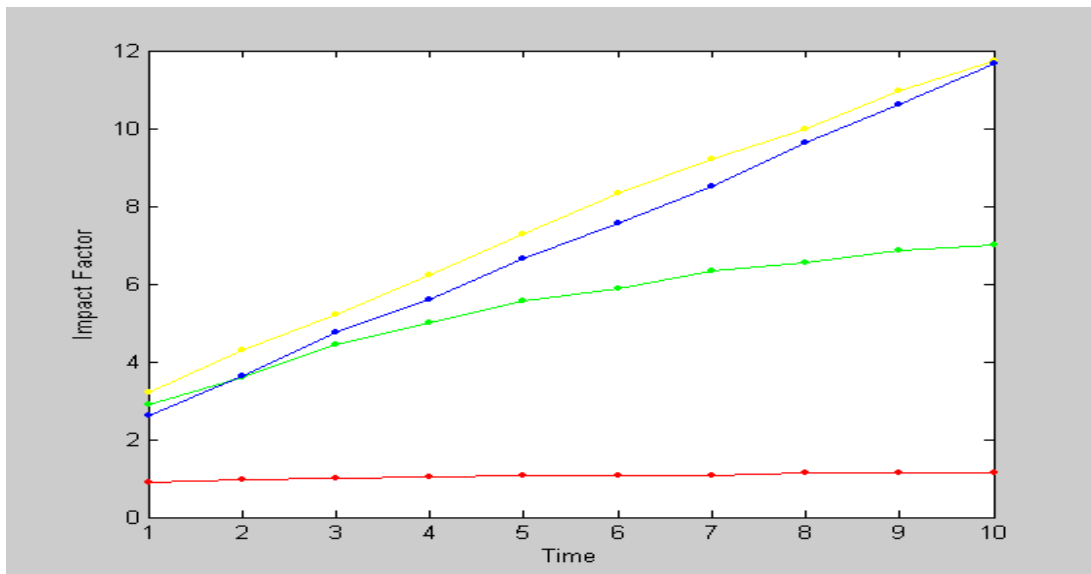


Figure 5.26 Impact factor over time on utility-oriented community

Figure 5.26 shows the trend of impact factor over time. Compared with Figure 5.22, there are still three sub domains that can attract the publics although one sub domain has the zero impact factors. It proves that the appropriate selection of generators can keep the scientific community attractive for a longer time.

5.3.4.3 Service-oriented

1. Predefined parameters

The following are predefined parameters for service-oriented community, and three operators probabilities are equal with each other i.e. 0.333.

Table 5.7 Predefined parameters for service-oriented community

Parameters	Value	Description
Recruitment selectivity	0.2	If the enculturation level is greater than 0.2, the new agent becomes tenured. Otherwise the agent leaves the community.
Growth rate	3.5	At each time interval, a random number of agents belonging to $U(0, 3.5)$ enter the community.
Turnover	0.5	If the reputation is less than 0.5, the tenured agent leaves the community.

2. Results

Table 5.8 summarizes the simulation results on service-oriented community.

Table 5.8 Results on service-oriented community

Metrics	Value
Accepted Kenes	1397
Average Kene Fitness	0.69
Average Diffusion for Kenes	1.038
Density for Social Network	0.0004
Ratio of Strong Ties to Weak Ties	0.031
Centrality for Social Network	0.196
Clustering Coefficient for Social Network	0.084
Clustering Coefficient for Social-Technical Network	0.271

Figure 5.27 indicates the total number of individuals who create the same amount of kenes.

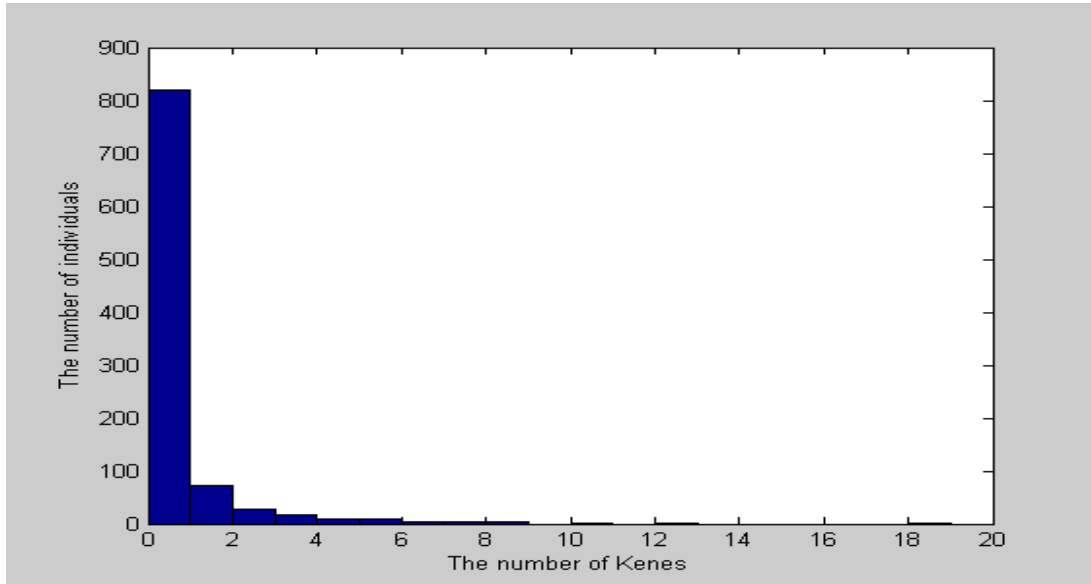


Figure 5.27 Histogram of the number of agents who create the same number of kenes

Figure 5.28 indicates the number of kenes created by each individual ordered by the rank of individuals.

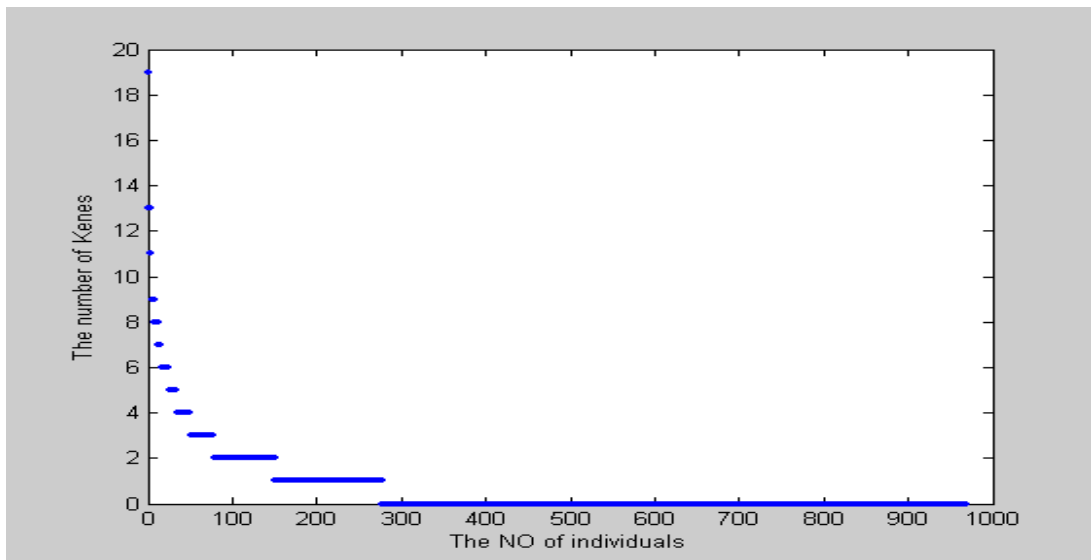


Figure 5.28 Plot of the number of kenes created by each agents

When GEM style is selected, the graph of agents' distribution will be like Figure 5.29. From this image, it is clear that the community has a center. It is different from the exploration-oriented community and the utility-oriented community.

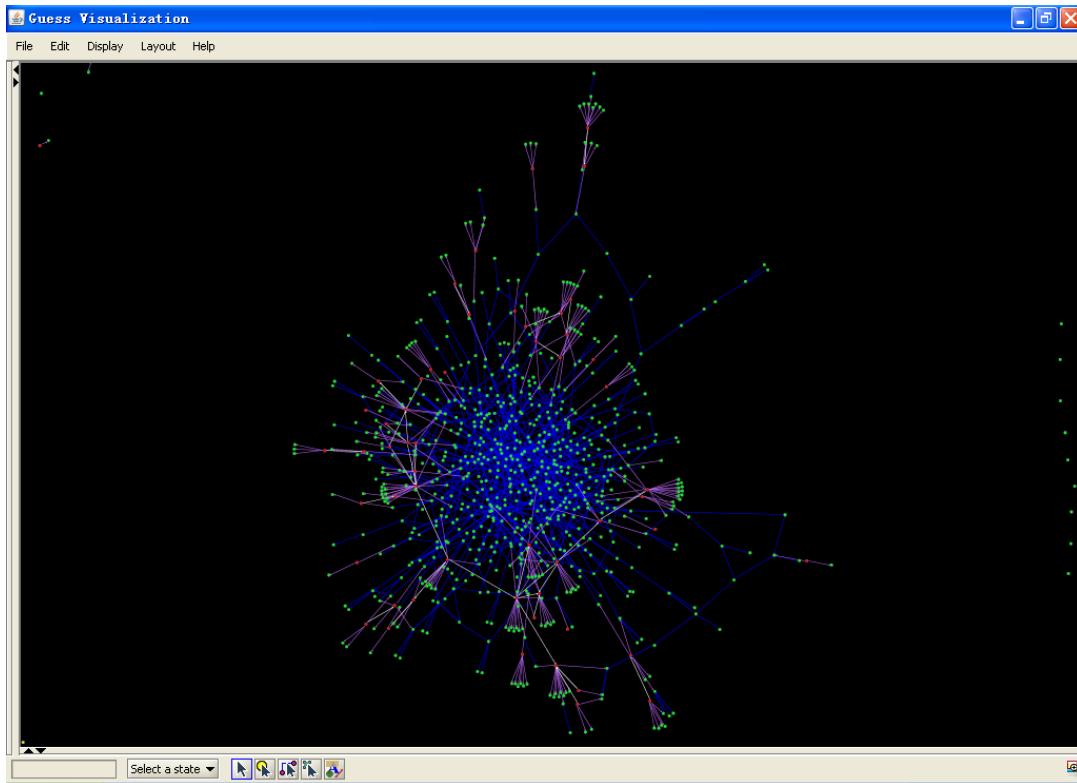


Figure 5.29 Emergent pattern in GEM style

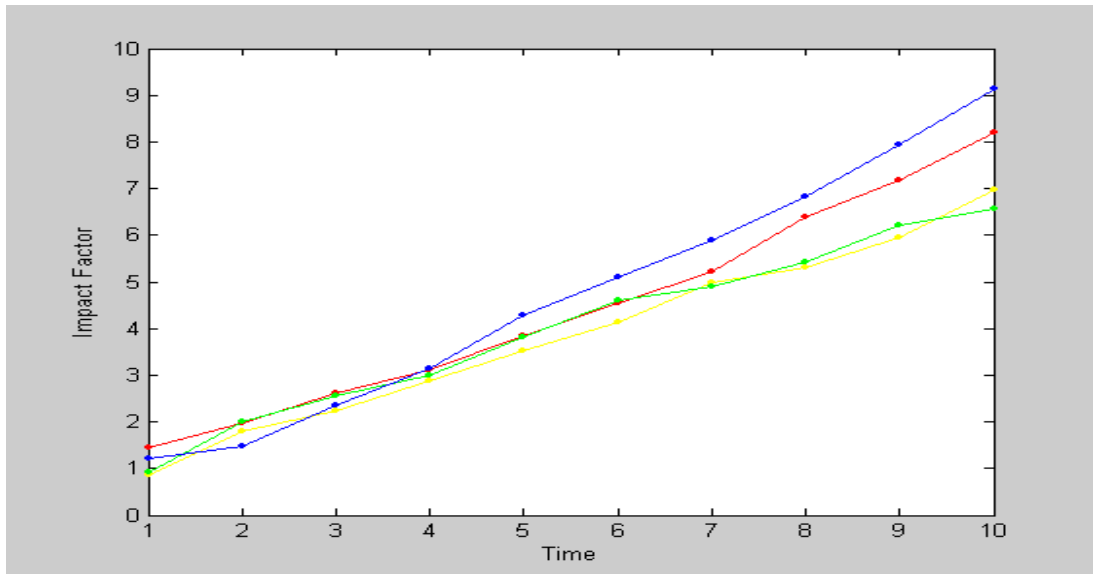


Figure 5.30 Impact factor over time on service-oriented community

Figure 5.30 shows that the accumulated impact factor of service oriented community is also decreasing with the time. However the slope is larger than utility oriented community, which means the service oriented community has longer attractions than utility community.

5.3.4.4 Emergent Social Organization Structures under Alternative Communities

The purpose of this experiment is to compare communities with respect to innovation metrics and network metrics. So the experiment can be divided into two sections. One is to compare communities against number of accepted kenes, diffusion and average kene fitness. The other is to perform comparison of communities in terms of density, centrality and clustering coefficient.

- **Comparison of Communities with Respect to Innovation Metrics**

Innovation metrics include number of accepted kenes, diffusion and average kene fitness.

Firstly, the comparison of communities with respect to the number of accepted kenes is performed. Figure 5.31 represents that number of accepted kenes varies with the type of community. From this figure, we can conclude that exploration-oriented community has the similar number of accepted kenes to service-oriented community. And the number of accepted kenes of both of them is much larger than utility-oriented community.

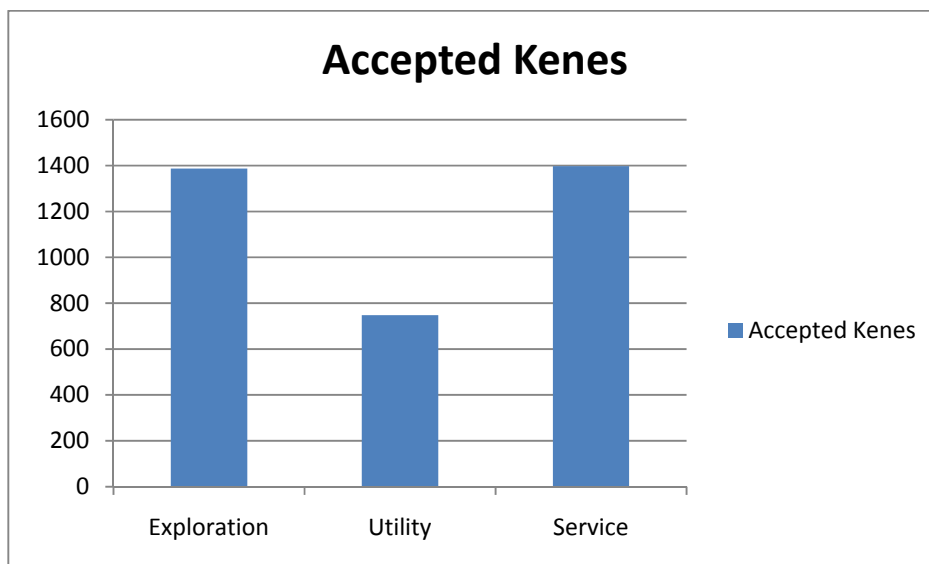


Figure 5.31 Number of accepted kenes

Secondly, the comparison of communities with respect to the average kene fitness is performed. Figure 5.32 represents average kene fitness varying with the types of communities. Here utility-oriented community has the highest average kene fitness, and exploration-oriented community has the lowest average kene fitness. The reason is due to

the difference of decision-making style in different communities. The decision-making style in utility-oriented community is emergent selection under which evaluation of genes is based on the references associated with the gene. Only those genes with many enough references can be retained in the domain. So the genes in the utility-oriented community has higher relational fitness so as to have higher average gene fitness further.

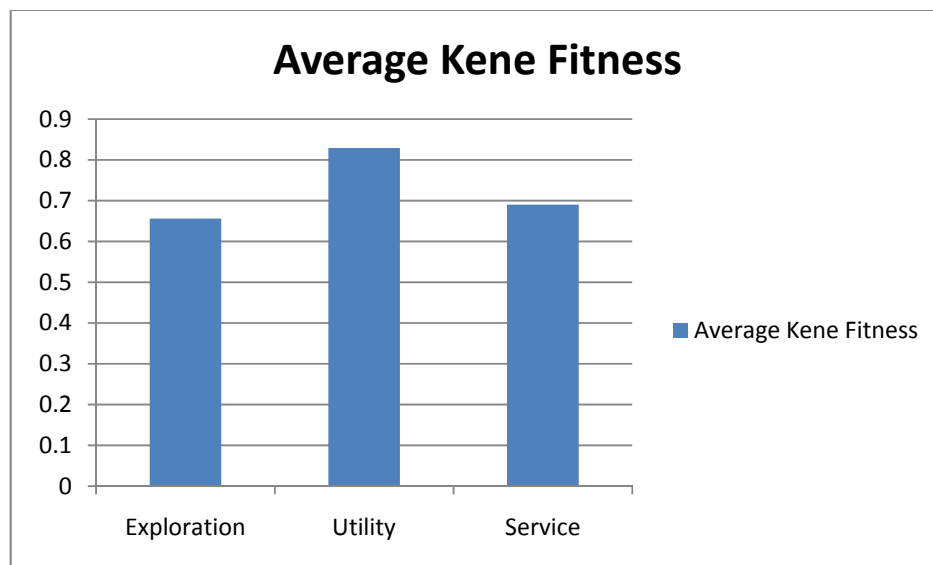


Figure 5.32 Average kene fitness

The situation that the average diffusion for genes changes with the type of community is shown in Figure 5.33. It shows that the average diffusion of exploration oriented community is slightly higher than service oriented community that is also much higher than utility oriented community. It also means that the influence across sub domains in the utility oriented community is lower than other two types of communities.

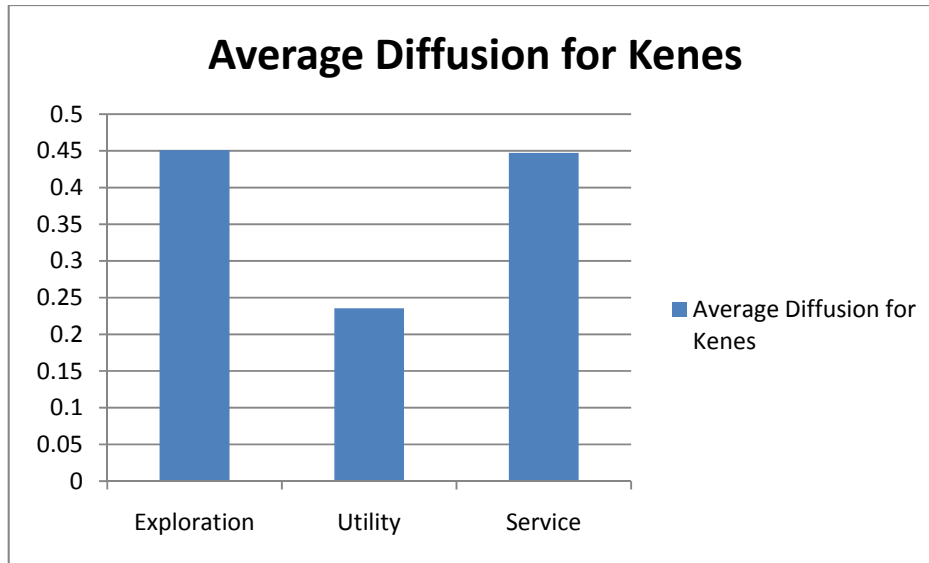


Figure 5.33 Average diffusion for kenes with different types of communities

- **Comparison of Communities with Respect to Social Network Metrics**

This section compares the three types of communities against social network metrics including density, centrality, clustering coefficient of social network.

Firstly, the comparison of communities with respect to density of social network is made. Figure 5.34 represents that density of individuals varies with the type of community. From this figure, we can conclude that utility-oriented community has the similar density to service-oriented community. And the density of both of them is much lower than exploration oriented community.

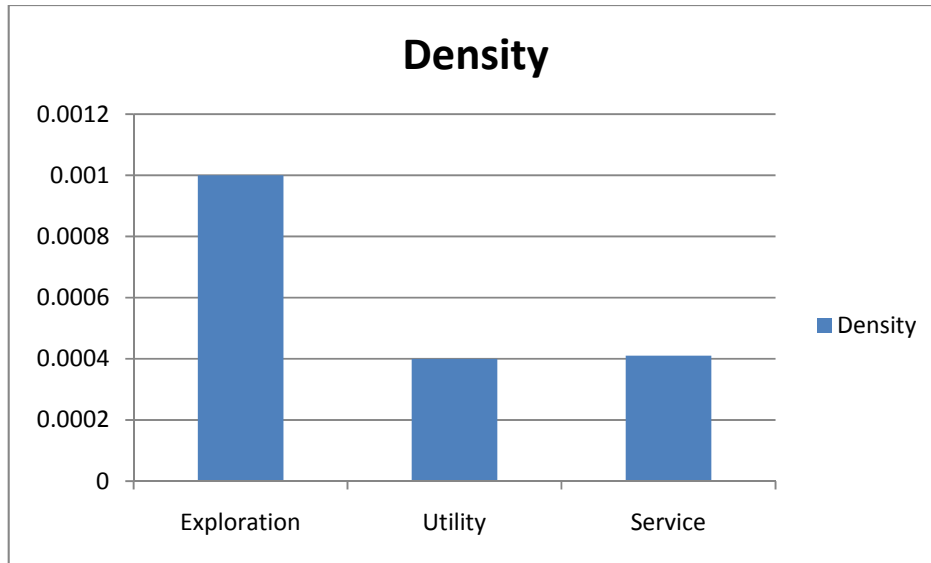


Figure 5.34 Density with different types of communities

Secondly, the comparison of communities with respect to centrality of social network is performed. Figure 5.35 shows the centrality with different types of communities. Here service-oriented community has much lower centrality than utility-oriented and exploration-oriented community, which is consistent with the character of them. The purpose of service-oriented is to provide stable services. So the members in the service-oriented community consider co-operation less than those in utility-oriented community and in exploration-oriented communities.

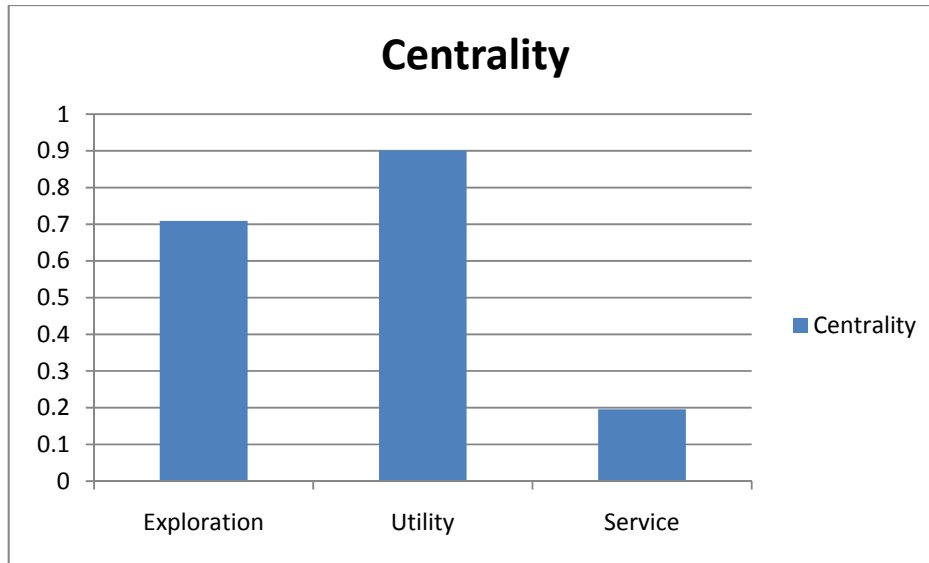


Figure 5.35 Centrality with different types of communities

The situation of proportion of strong ties varying with type of community is similar to that of centrality. Utility-oriented community has significantly higher proportion of strong ties than other two types of communities. Also the proportion of strong ties in exploration-oriented community is larger than that in service-oriented community. The result is consistent with the definition for centrality and proportion of strong ties. Centrality is based on the degree of vertices, i.e., the number of links that a node has. Also the definition for strong tie is related to the number of links in the tie. Comparison of Figure 5.36 and Figure 5.35 proves there are some kinds of correlations between centrality and proportion of strong ties.

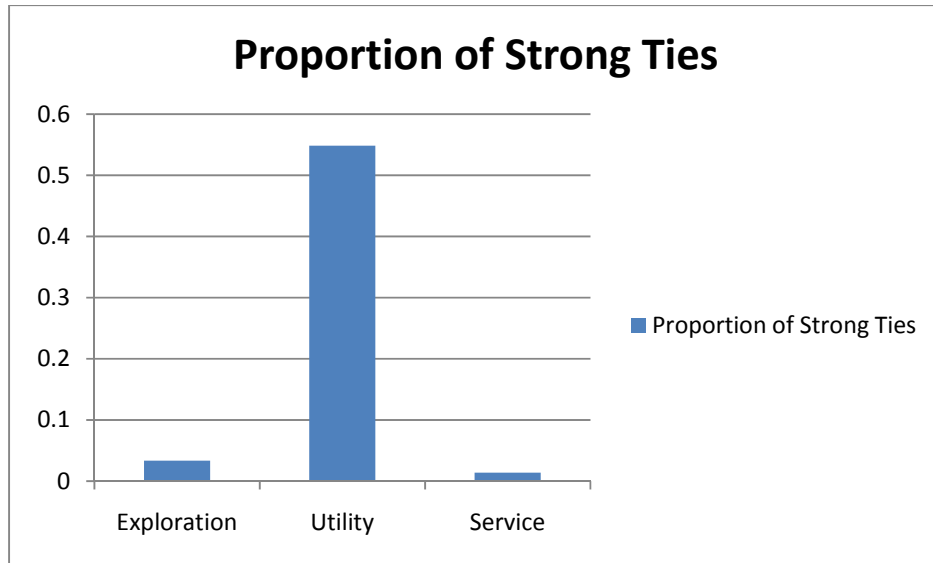


Figure 5.36 Proportion of strong ties with different communities

The situation that the clustering coefficient for social network changes with the type of community is shown in Figure 5.37, and Figure 5.38 reflects that the clustering coefficient for social-technical network changes with the type of community. Both of these figures have the same conclusions about the clustering on different scientific communities. Here exploration oriented community has the strongest intention for clique. And utility oriented community has the loosest organizational structure. Service oriented community is between exploration oriented community and utility oriented community.

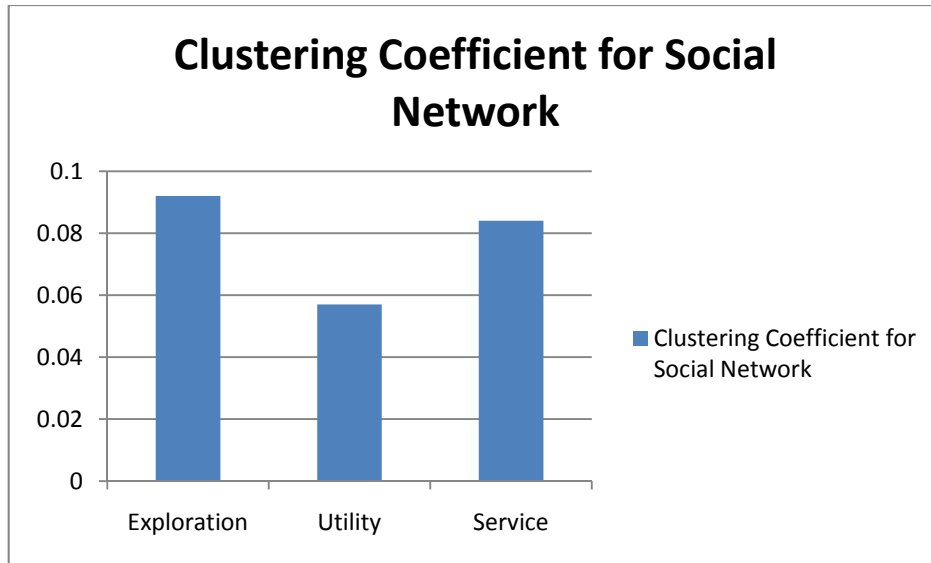


Figure 5.37 Clustering coefficient for agents with different types of communities

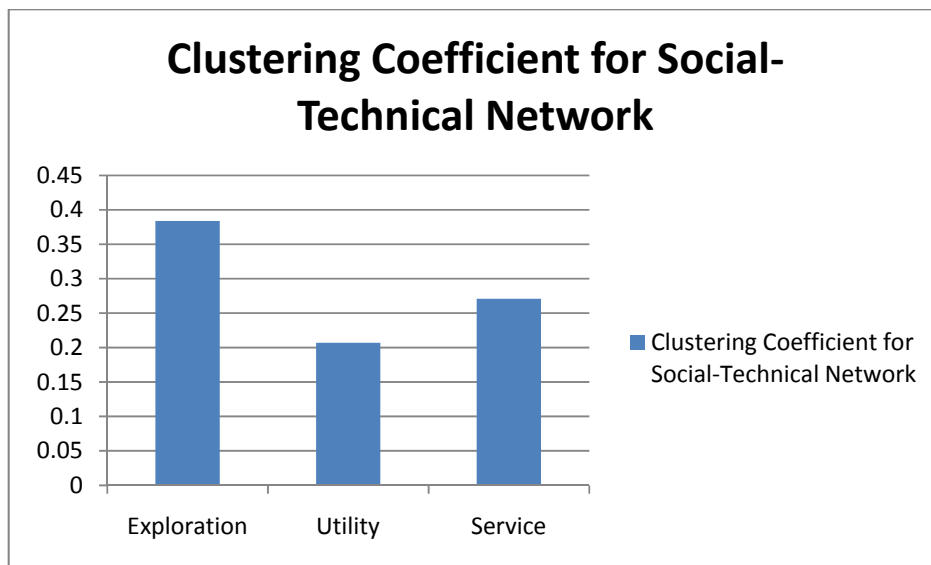


Figure 5.38 Total clustering coefficient with different types of communities

Table 5.9 summarizes all of the experiment results above in a table.

Table 5.9 Experiment results

Metrics	Results
Density for Social Network	Exploration > Service ≈ Utility

Accepted Kenes	Exploration \approx Service > Utility
Average Diffusion for Kenes	Exploration > Service > Utility
Centrality for Social Network	Utility > Exploration > Service
Clustering Coefficient for Social Network	Exploration > Service > Utility

Another thing worth paying attention is absolutely most of kenes are created by very few individuals in utility-oriented community.

5.3.5 What Distinguishes Innovative Communities?

The purpose of this experiment is to explore the characters that distinguish the scientific communities. Here relationship between average kene fitness and social network metrics in a specific type of community is shown. To analyze the relationship batch run of the simulation model is needed where the replication times is 100. The experiment consists of two steps: the first step is to divide the collective data into three groups (i.e. low, moderate and high innovation) based on the average kene fitness; the second step is to compute the average network metrics for each group.

5.3.5.1 Innovation in Exploration-oriented Community

In all the one hundred replications, the minimum average kene fitness is 0.630202741, and the maximum average kene fitness is 0.7205807. The replications with average kene fitness between 0.630202741 and 0.660328727 are classified as low average kene fitness. The replications whose average kene fitness are in [0.660328727, 0.690454714) are classified as moderate kene fitness. The rest of replications belong to high kene fitness. Table 5.10 summarizes the network metrics grouped by average kene fitness.

Table 5.10 Network metrics of exploration-oriented community grouped by average kene fitness

	Low	Moderate	High
Density	0.001278	0.00138	0.001296
Centrality	0.764141	0.771677	0.794853
Clustering Coefficient	0.089459	0.091548	0.087866
Proportion of Strong Ties	0.043825	0.035967	0.031939
Number of Replications	10	68	22

From Table 5.10, we can see the replications with moderate average kene fitness occupy the majority of all the collective data. Figure 5.39 depicts the comparison of the results.

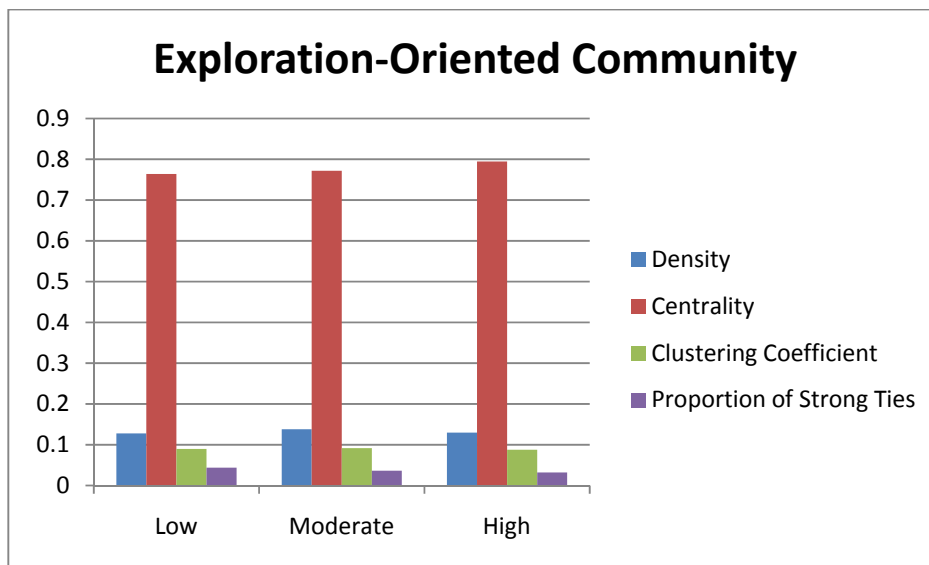


Figure 5.39 Network metrics of exploration-oriented community grouped by average kene fitness

In Figure 5.39, density and clustering coefficient almost keep the same in different group of average kene fitness. However, the centrality increases with the increase of average kene fitness. It shows centrality has positive effects on the quality of kenes in exploration-oriented community. Additionally, proportion of strong ties decreases with average kene fitness increasing.

5.3.5.2 Innovation in Utility-oriented Community

In all the one hundred replications for utility-oriented community, the minimum average kene fitness is 0.708175474, and the maximum average kene fitness is 0.955391252. The replications with average kene fitness between 0.708175474 and 0.790580734 are classified as low average kene fitness. The replications whose average kene fitness are in [0.790580734, 0.872985993) are classified as moderate kene fitness. The rest of replications belong to high kene fitness. Table 5.11 summarizes the network metrics grouped by average kene fitness.

Table 5.11 Network metrics of utility-oriented community grouped by average kene fitness

	Low	Moderate	High
Density	3.08E-04	4.43E-04	5.61E-04
Centrality	0.730454	1.030052	1.310602
Clustering Coefficient	0.04008	0.050025	0.054477
Proportion of Strong Ties	0.530242	0.530113	0.565781
Number of Replications	12	37	51

From Table 5.11, we can see the replications with high average kene fitness occupy the majority of all the collective data, which is opposed to exploration-oriented community. Figure 5.40 depicts the comparison of the results.

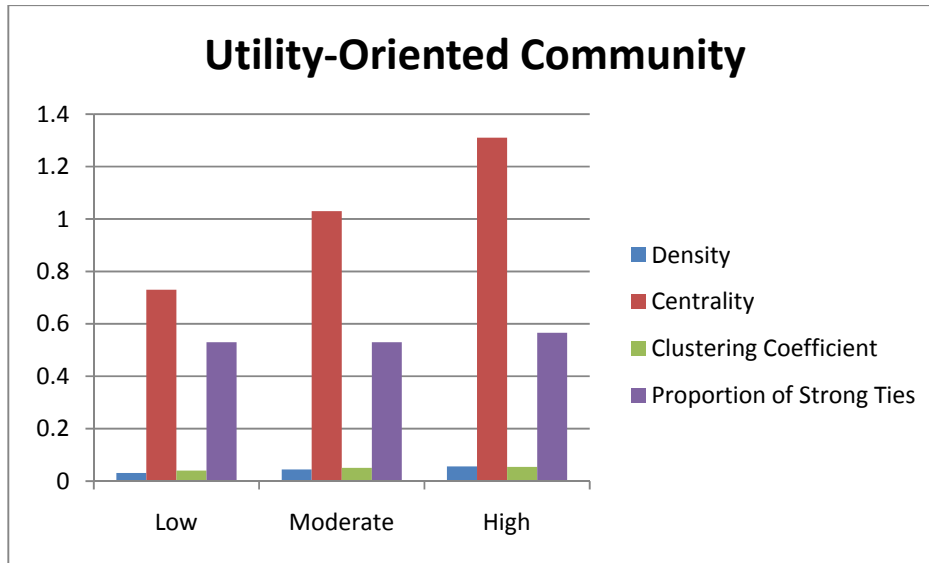


Figure 5.40 Network metrics of utility-oriented community grouped by average kene fitness

In Figure 5.40, clustering coefficient is not significantly different across different innovation performance categories. However, density and proportion of strong ties, especially centrality, increase with the increase of average kene fitness. It shows centrality has positive effects on the quality of kenes in utility-oriented community.

5.3.5.3 Innovation in Service-oriented Community

In all the one hundred replications, the minimum average kene fitness is 0.644617998, and the maximum average kene fitness is 0.728058495. The replications with average kene fitness between 0.644617998 and 0.672431497 are classified as low average kene fitness. The replications whose average kene fitness are in [0.672431497, 0.700244996) are classified as moderate kene fitness. The rest of replications belong to high kene fitness. Table 5.12 summarizes the network metrics grouped by average kene fitness.

Table 5.12 Network metrics of service-oriented community grouped by average kene fitness

	Low	Moderate	High
Density	2.97E-04	2.96E-04	2.97E-04
Centrality	0.115405	0.113757	0.107917
Clustering Coefficient	0.052939	0.05424	0.054654
Proportion of Strong Ties	0.076199	0.07704	0.070723
Number of Replications	9	69	22

From Table 5.12, we can see the replications with moderate average kene fitness occupy the majority of all the collective data, which is similar to exploration-oriented community. Figure 5.41 depicts the comparison of the results.

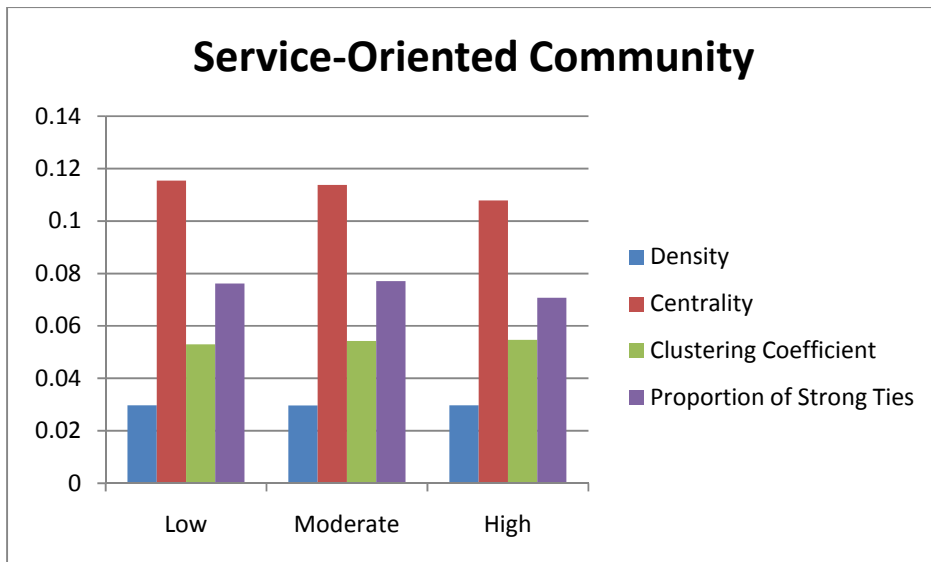


Figure 5.41 Network metrics of service-oriented community grouped by average kene fitness

In Figure 5.41, density and clustering coefficient almost stay the same in different group of average kene fitness. However, both centrality and proportion of strong ties decrease with the increase of average kene fitness. It shows centrality and proportion of strong ties has negative effects on the quality of kenes in service-oriented community.

From all the experiments above, the common points are that the centrality has influences over the quality of kenes. Additionally the proportion of moderate average

kene fitness is larger than low and high average kene fitness in exploration-oriented community and service-oriented community. On the contrary, in the utility-oriented community the replications with high average kene fitness occupy the majority. Combined with the fact that utility-oriented community has the largest variety range of average kene fitness, we can conclude that there is the greatest variation of quality of kenes in the utility-oriented community.

CHAPTER 6

Conclusions

From what has been discussed above, we can safely reach the conclusion as follows.

Firstly, the open science community model provides an existence proof that it is possible to use simple local rules to generate higher levels of organization. In particular, it shows that the three generators can lead to clusters of genes and individuals that behave according to their independent rules.

Secondly, the simulation model is a useful abstraction and simplification of the real world as discussed in the verification, validation and evaluation section. Therefore the model presented in this thesis provides a basis to perform further research by changing the value of parameters.

Next, the simulation model can lead to the insights into where there might be policy leverage in the real world [45]. For example, one might identify which configuration of open science community has more potential to innovate through setting the configuration onto the simulation model in this thesis. Based on the simulation results, the utility-oriented community has the maximum centrality and minimum density of agents, which means it has more potential to be creative.

Finally centrality has significantly effects on the quality of genes in all three kinds of scientific communities. At the same time there is the greatest variation of average gene fitness in utility-oriented community.

6.1 Extension

The key point of the model proposed in this thesis is the interconnection among agents. Thus the model can be used to simulate those communities which are organized based on the dynamic of relationship among members.

These types of communities are listed as follows. [18]

Shared Instrument: Its main function is to increase access to a scientific instrument. Shared Instrument laboratories often provide remote access to expensive scientific instruments such as telescopes. In this community our model can help figure out the strategy of improving the efficiency of expensive instruments.

Virtual Community of Practice: It is a network of individuals who share a research area and communicate about it online. Virtual Communities may share news of professional interest, advice, techniques, or pointers to other resources online.

Virtual Learning Community: Main goal is to increase the knowledge of participants but not necessarily to conduct original research.

Distributed Research Center: Its functions are similar as a university research center but at a distance. It is an attempt to aggregate scientific talent, effort, and resources beyond the level of individual researchers.

For these types of communities above, our model only needs to be slightly modified according to the specific characters of each type so that it can simulate the activities in these communities and analyze the data.

6.2 Future Research

In the current model, there are some simplifications. So we can enrich the model by adding additional attributes such as the knowledge level, which evolves over time as the agent innovates and receives information broadcast by other agents. Higher the level of knowledge is, more likely the agent provides an appropriate gene. Concomitantly the aggregated level of all agents in a community reflects the speed of knowledge diffusion.

In addition, one possible extension is to divide the context of Repast model from single one into two sub contexts, which means agents and genes are in one of these two sub contexts respectively. In the sub context in which agents exist the grid projection is used so that agents can move and communicate with neighbors. In other context, network projection is used to represent the relationship among genes. And the location of genes can be omitted so as to focus on the cliquishness and clustering analysis.

Finally, in the current model, the evaluator is fixed, which means the agent as evaluator can't do contribution to the community. However the reviewers of a journal, for instance, can also submit their own papers to be reviewed by other reviewers in the real world. Perhaps exchange of roles between evaluators and common members may lead to the emergence of other complex phenomena.

REFERENCES

- [1] Francisco Javier LLorens-Montes, Víctor J. Garcia-Morales, Antonio J. Verdu-Jover, “The influence on personal mastery, organisational learning and performance of the level of innovation: adaptive organisation versus innovator organization,” *International Journal of Innovation and Learning*, Volume 1, Number 2 / 2004, Pages: 101 – 114
- [2] Faiz Gallouj, Edward Elgar, “Innovation in the Service Economy: The New Wealth of Nations,” 2002
- [3] S Gopalakrishnan, “A Review of Innovation Research in Economics, Sociology and Technology Management,” *Omega*, Int. J. Mgmt Sci. Vol. 25, No. 1 pp. 15-28, 1997
- [4] M. Csikszentmihalyi, “Implications of a systems perspective for the study of creativity,” *Handbook of Creativity*, pages 3, 1999.
- [5] Thomas B. Ward, Steven M. Smith, and Jyotsna Valid. “Conceptual Structures and Processes in Creative Thought.”
- [6] Levent Yilmaz, “Dynamics of Collective Creativity and Open Innovation in Scientific Commons Complex Adaptive Systems Perspective,” *Computer Science and Software Engineering*, Auburn University.
- [7] Susan A. Mohrman, “The Dynamics of Knowledge Creation: Phase One Assessment of the Role and Contribution of the Department of Energy's Nanoscale Science Research Centers.” University of Southern California, Los Angeles, CA 90089.
- [8] Complex System. From Wikipedia website.
- [9] Gary William Flake, “The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos,” *Complex Systems, and Adaptation*, Chapter 9. MIT Press, 2000.
- [10] Eric Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” Icosystem Corporation, 545 Concord Avenue, Cambridge, MA 02138.

- [11] Michael W. Macy and Robert Willer, "FROM FACTORS TO ACTORS: Computational Sociology and Agent-Based Modeling," Department of Sociology, Cornell University, Ithaca, New York 84153.
- [12] Levent Yilmaz, "On the synergy of conflict and collective creativity in open source software communities," Computer Science and Software Engineering, Auburn University.
- [13] Nathan Bos, Applied Physics Laboratory, Johns Hopkins, Ann Zimmerman, "From Shared Databases to Communities of Practice: A Taxonomy of Collaboratories," School of Information, University of Michigan. *Journal of Computer-Mediated Communication*, 12 (2007) 318–338.
- [14] Alexander Hars, "Working for Free? – Motivations of Participating in Open Source Projects," IOM Department, Marshall School of Business, University of Southern California.
- [15] AUDRIS MOCKUS, "Two Case Studies of Open Source Software Development: Apache and Mozilla," Avaya Labs Research, Carnegie Mellon University.
- [16] Karim R. Lakhani and Robert G Wolf, "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," MIT Sloan School of Management, The Boston Consulting Group.
- [17] Barry Smith, "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration," *Nature Biotechnology* 25, 1251 - 1255 (2007).
- [18] R. Cowan, N. Jonard, "The dynamics of collective invention," *Journal of Economic Behavior & Organization*, Vol. 52 (2003) 513–532.
- [19] Greg Madey, "Agent-Based Modeling of Open Source using Swarm," Computer Science & Engineering, University of Notre Dame.
- [20] Nigel Gilbert, "Agent-based social simulation: dealing with complexity," Centre for Research on Social Simulation, University of Surrey, Guildford, UK.
- [21] Simon, Herbert A. "The Architecture of Complexity." *Proceedings of the American Philosophical Society*, 106 (December): 467-482. 1962.
- [22] Jean-Michel, Paul A. David, "SimCode: Agent-based Simulation Modelling of Open-Source Software Development," Dalle University Pierre-et-Marie-Curie & IMRI-Dauphine and Stanford University & Oxford Internet Institute.

- [23] Brian J. L. Berry, L. Douglas Kiel, and Euel Elliott, "Adaptive agents, intelligence, and emergent human organization: Capturing complexity through agent-based modeling," School of Social Sciences, University of Texas at Dallas, Richardson, TX 75080.
- [24] Leigh Tesfatsion, "Agent-based computational economics: modeling economies as complex adaptive systems," Department of Economics, Iowa State University, Ames, IA 50011-1070, USA.
- [25] Richard W. Woodman, "Toward a Theory of Organizational Creativity," Texas A&M University.
- [26] Volker Grimma, Uta Bergerb, "A standard protocol for describing individual-based and agent-based models," Department of Okologische Systemanalyse, Permoserstr. 15, 04318 Leipzig, Germany.
- [27] Robert Tobias and Carole Hofmann, "Evaluation of free Java-libraries for social-scientific agent based simulation," *Journal of Artificial Societies and Social Simulation* vol. 7, no. 1.
- [28] Neil Smith, Andrea Capiluppi, Juan Fernández Ramil, "Agent-based Simulation of Open Source Evolution," Computing Department, The Open University, Walton Hall, Milton Keynes MK7 6AA, U.K.
- [29] Charles Edquist, "Systems of Innovation, Technologies, Institutions and organizations," Pages 6-7.
- [30] Franco Malerba, CESPRI, "Sectoral Systems of Innovation and Production," Bocconi University, Via Sarfatti 25, 20136 Milan, Italy.
- [31] Nigel Gilbert, "How to Build and Use Agent-Based Models in Social Science," Centre for Research on Simulation for the Social Sciences, School of Human Sciences, University of Surrey, Guildford, GU2 5XH, UK.
- [32] Robert Axelrod, "The dissemination of Culture: A Model with Local Convergence and Global Polarization," *Journal of Conflict Resolution* 41 (1997): 203-26.
- [33] John H. Miller and Scott E. Page., "The Standing Ovation Problem," April 12, 2004.
- [34] Turtlebender, Etatar, "Getting started for Repast Symphony," Confluence July 24, 2008.

- [35] Edward J. Rykiel Jr., "Testing ecological models : the meaning of validation," *Ecological Modeling* 90 (1996) 229-244.
- [36] Stephen Vincent, "Input Data Analysis Chapter 3," Compuware Corporation. 1998.
- [37] Validation & Accreditation (VV&A) for Models and Simulations, Department of Defense Documentation of Verification, Missile Defense Agency, 2008.
- [38] SoftwareTestingClub.com, 2009, "Is Integration A Phase?", <http://www.softwaretestingclub.com/forum/topics/is-integration-a-phase> Integration a phase.
- [39] Volker Grimm, "Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology," *Science* 310, 987 (2005).
- [40] Gideon S. Mann, David Mimno, Andrew McCallum, "Bibliometric Impact Measures Leveraging Topic Analysis," Department of Computer Science, University of Massachusetts Amherst, Amherst MA 01003.
- [41] D. J. Watts and Steven Strogatz, "Collective dynamics of 'small-world' networks," *Nature* 393: 440–442. (June 1998).
- [42] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D. Saisana, M., and Tarantola, S., "Global Sensitivity Analysis," *The Primer*, John Wiley & Sons, 2008.
- [43] Guido Hertel, Sven Niedner, Stefanie Herrmann, Institut fuer Psychologie, "Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel," University of Kiel, Olshausenstr. 40, D-24 098 Kiel, Germany.
- [44] Stefan Koch & Georg Schneider, "Effort, co-operation and co-ordination in an open source software project: GNOME," Department of Information Business, Vienna University of Economics and Business Administration, Augasse 2–6, A-1090 Vienna, Austria
- [45] Robert Axelrod, "Building New Political Actors: A Model for the Emergence of New Political Actors," *Artificial Societies: The Computer Simulation of Social Life*, 1995, 19-39.

APPENDIX

Pseudo Code for the Model

- ContextCreator

The class of ContextCreator is used to build the context in which all agents must be.

It has two major parts. One is constructor; the other is a global method.

```
public class ContextCreator implements ContextBuilder
{
    public Context build(Context context)
    {
        // Read parameters and build context according to these parameters
        int N = readParameters("geneLength");
        int gridWidth = readParameters("gridWidth");
        int gridHeight = readParameters("gridHeight");
        int amplification = readParameters("amplification");
        ContinuousSpace space = buildSpace("Simple Grid", gridWidth,
gridHeight);
        // Build the network that represents the relation between genes.
```

```

NetworkBuilder("RelationOfKenes", context, true);

// Build the network that represents the relation between individuals.

NetworkBuilder("RelationOfIndividuals", context, true);

// Build the network that represents the relation between individuals and
kenes.

NetworkBuilder("RelationBetweenKenesAndIndividuals", context, true);

//Set a global method named "update" that runs every time tick.

setGlobalMethod("update", 1, 1);

return context;
}

public void update()
{

Context context = m_context;

// Invoke the function that is in charge of entry and turnover.

entryAndTurnover(context);

// Calculate the ranks for every individuals, which is based on their
reputations.

calRanksForIndividuals();

// Judge if the end time reaches.

```

```

        if (currentTick == endTick)
        {
            // Calculate some metrics when the simulation is end.
            calMetrics();
        }
        //Record some history data
        recordHistory(currentTick);
    }
}

```

- Individual

The class Individual represents the member of scientific community, who makes contributions to the community. The class has two major functions. One is constructor; the other is named as “step” which is invoked every time tick.

```

public class Individual
{
    //The argument means if the individual is tenured when he is created
    public Individual(boolean tenure)
    {
        // Read parameters and assign them to state variables
        m_iAmplification = readParameters("amplification");
    }
}

```

```

    m_dProbCombination = readParameters("probCombination");

    m_dProbCreation = readParameters("probCreation");

    m_dProbElaboration = readParameters("probElaboration");

    m_dReputation = Math.random();

    m_bTenure = tenure;

    m_iCreatedTick = getCurrentTickCount();

    m_subDomain = (int)(Math.random()*4);

    m_ID = ++s_ID;
}

@ScheduledMethod(start = 1, interval = 1)

public void step()

{

    // Agents move in the grid randomly.

    move();

    if(m_bTenure)

    {

        // Innovation only if his motivation is greater than a random
number.

        double rand = Math.random();

        if (rand > this.m_dProbMotivation)

```

```

    {
        return;
    }

    // Select one of three generators based on the random number.
    if (rand < m_dProbCreation)
    {
        create();
    }

    else if (rand < m_dProbCreation + m_dProbCombination)
    {
        combination();
    }

    else
    {
        elaboration();
    }
}

else
{

    //enculturation

```

```

        enculturation();
    }
}

```

- Evaluator

The class Evaluator represents the arbitrator of scientific community, who determines whether or not a gene is appropriate to be retained in the domain. The class has three major functions. One is constructor; the second is named as “step” which is invoked every time tick; the third is named as “evaluate” which is used to determine the fitness of a specific gene.

```

public class Evaluator
{
    // Constructor
    public Evaluator()
    {
        // Read parameters and assign them to state variables.
        m_iAmplification = readParameters("amplification");
        m_strDecisionType = readParameters("typeCommunity");
    }
}

```



```

@ScheduledMethod(start = 1, interval = 1)

public void step()
{
    move();

    // Emergent selection corresponding to utility-oriented community
    if (m_strDecisionType.equals("Utility"))
    {
        Context context = ContextUtils.getContext(this);

        // Get the iterator of kenes
        Iterator<Kene> iter = context.getObjects(Kene.class).iterator();

        // Get the network for the relationship of kenes
        Network network = (Network)
context.getProjection("RelationOfKenes");

        int tenureDuration = 100;

        int iCurrentTick = getCurrentTickCount();

        Vector<Kene> removedKene = new Vector<Kene>();

        // Iterate all the kenes
        while (iter.hasNext())
        {
            Kene kene = iter.next();

```

```

// Only those kenes that pass the duration of tenure.
if (iCurrentTick == kene.getCreatedTick() +
tenureDuration)
{
// The kenes that never are cited.
if (!network.getEdges(kene).iterator().hasNext())
{
removedKene.add(kene);
}
}
}
// Remove those kenes that never are cited.
for (Kene kene : removedKene)
{
context.remove(kene);
}
}
// Determine whether or not a specific kene is qualified.
public boolean evaluate(Kene kene)

```

```

{
    boolean result = false;

    Context context = ContextUtils.getContext(kene);

    if (context == null)
    {
        // Calculate individual fitness of the kene.

        result = kene.getM_individualFitness() >= 0.5;
    }
    else
    {
        // Calculate the sum of individual fitness and relational fitness.

        result = kene.getM_individualFitness() + calFitness(context,
kene)/g_dMaxFitness >= 0.5;
    }

    return result;
}
}

```