Exploring the Perception and Behavior of Software Engineers about Computer Software Patent

SHANN-BIN CHANG [1*] SHU-MIN CHANG [2,3]

[1.] Department of Information Management, Ling Tung University
No.1, Ling Tung Road, Nan Tun, Taichung, Taiwan.
[2.] Department of Industry Engineering & Management, Nan Kai University of Technology
No.568, Chung-Cheng Road, Tsao Tun, Nan Tou, Taiwan.
[3.] Department of Business Administration, National Yunlin University of Science & Technology
No.123, Section 3, University Road, Douliou, Yunlin, Taiwan.
[*] Corresponding Author email address: changsb@teamail.ltu.edu.tw

Abstract: -Patent is one kind of the intellectual property, and the nature of patent institution is to accelerate the speed of innovation and invention. It is likely to the patent of computer software. But the opinion voiced by exponents of open source and free software is extremely opposed to patenting computer software. They are concerned the software patent acts as a barrier to technology improvement. This study employs a modified technological acceptance model (TAM) to explore the perception and behavior of software engineers, and examines the relationship between perception and management behavior about patent. The findings of this study are two-fold: 1. Most of the relations of this research model are positively significant, including knowledge of patent, dependency on patent, perceived usefulness of patent, perceived ease of use of patent, intention to use patent, and actual behavior of patent management; 2. The scales of firm and number of patents owned by firms have a significantly different effect from the above constructs, but the individual attributes of engineers are not significantly different from those constructs. Finally, this study provides two suggestions for future study: one is concerned with the willingness and capability of computer software inventors to codify invention; another concern the competency of absorbing the patent knowledge from computer software engineer. Based on these two aspects, researchers can achieve a deeper understanding of the knowledge innovation and technology diffusion.

Key-words: -Patent, computer software, software engineer, technological acceptance model (TAM)

1 Introduction

Patent is one kind of the intellectual property, and the nature of patent institution is to accelerate the speed of innovation and invention. It is likely to the patent of computer software. But another voice from the exponents of open source and free software is extremely opposed to the software patent. These exponents are concerned that patenting computer software is a barrier to technology improvement. Before 1996, the computer software was protected by copyright. Since ‘Examination Guidelines for Computer-related Inventions’ was announced by US Patent and Trademark Office in 1996 [1], and several legal precedents of U.S. Court of Appeals for the Federal Circuit (CAFC), such as State Street Bank & Trust Co. v. Signature Financial Group, Inc. and Amazon.com v. Barnes & Noble, illustrated more clearly that the computer software and business methods that can indeed be patented. These phenomena encouraged firms to apply more actively for patents of computer software and business methods. However, another opinion from the exponents of open source and free software is extremely opposed to the software patenting process. They are concerned the software patent acts as a barrier of technology improvement [2]. It is an interesting issue to find out which one is the primary concept of software engineers. Because this issue is a perception and behavior problem, this study explores the attitude and behavior of software engineers by means of a modified technology acceptance model (TAM) and examines the relationships of perception and managing behavior of patent. Furthermore, this study examines the differences of perception and behavior by reviewing the size of the firms, whether any patent is owned, and the demographics of engineers, including gender, age, salary, experience and education.

2 Literature review

The literature of this study can be divided into four parts: 1. The first part introduces the nature of patent. 2. Describes the development of software patents. 3. Discussed a contrary opinion about software patent. 4. Introduces the TAM: the base of our research model.

2.1 Nature of patent

There are two characteristics of patent. One, it is technological document; it discloses new invention or innovation, that can advance technology development quickly. The second characteristic of patent is that of a legal document providing an exclusive right to the inventor or assignee, and protecting the inventor's new technology from being stolen by other firms. The propose of patent institution is to encourage the inventor to disclose the invention quickly, to prevent other firms from spending on R&D and minimize time used for advanced research; simultaneously, government awards the assignee the exclusive right to use, make, and sell the technology during a limited time.

There are three basic requirements of a patent: usefulness in some industry, novelty for the whole world, and non-obviousness from the prior art [3]. However, not all invention for which a patent application might be considers fit these requirements; there are many terms and conditions to limit patents application. For example: nature rules, natural phenomenon and abstract conception are not patentable in the US; science theory, mathematical methods and game rules are not patentable in Taiwan. Whether computer software can be patented or not is a controversial and confused issue. The reason for this is computer software is just only a code or notation, and is not concrete or touchable. This study discusses these aspects below.

2.2 Computer software patent

In an earlier time, computer software was protected by copyright, and business methods were protected by trade secret; both of these were not protected by patent. Since the time Internet technology has swept across the world, every company has come to view it as a new stage on which to compete in the 21st century. Business methods through computer and network technologies have become the weapons in this battle for success. Therefore, most of the countries began to discuss how to protect the computer software and business methods by patent. Germany was the first country to issue the guidelines for examining computer software related inventions in 1995, and US issued the examination guidelines for computer-related inventions in 1996, then, Japan in 1997, Taiwan in 1998, EU in 1999 sequentially issued the related guidelines. At the same time, several legal precedents of U.S. Court of Appeals for the Federal Circuit (CAFC), such as State Street Bank & Trust Co. v. Signature Financial Group, Inc. and Amazon.com v. Barnes & Noble, illustrated more clearly the business methods that can be patented. These phenomena encouraged firms to apply patents for computer software and business methods more actively [4, 5].

Intellectual Property Office of MOE issued examination guidelines for computer-related inventions in Taiwan in 1998. This document defined computer software invention as both apparatus invention and method invention. Apparatus invention comprises the hardware to create a concrete structure; method invention includes pre-computer process activity, post computer process activity and process activity within the computer. All the inventions must fit with the basic requirements of a patent, i.e. usefulness in some industry, novelty in the whole world, and non-obviousness from the prior art; and the inventions must comply with further terms and conditions.

As regards how to represent a of computer software patent, there is no defined concrete format. The only requirements are that the patent should describe clearly, in writing, the prior art, the objective of the invention, the details of the technology, and its characteristics and functioning. The patent should allow a specialist in that field to understand and practice the technology based on these disclosures. The representations of disclosure can consist of text description, data flow diagrams, pseudo code, block diagram, flow chart, time chart…and so on [6].

2.3 Different Opinion

As in above analysis, patent institution should stimulate an inventor to disclose his invention, and so accelerate the speed of innovation. Computer software invention is no exception. Nevertheless, another voice from exponents of open source and free software is extremely opposed to the software patent. They are concerned the software patent acts as a barrier to technology improvement, and even copyright is inhibiting progress. Stallman was the first one to advocate free software; he created a new word "copyleft" to set against copyright [7]. That means all software source code should be open for using, modifying and sharing. Torvalds was another key man for open source; he invented the Linux operation system; and opened the source code for engineers and others to use, modify, innovate and elaborate. He absolutely believed that if we want to accumulate knowledge the only way is to open up all knowledge [8].

Although many countries provided the related acts to protect computer software invention, many of adherents of open source were extremely opposed to the software patent. First, they provided some evidence to argue the utilities of patent institution, such as Apache html for Behlendorf, Sendmail program for Allman, Bind DNS for Vixie, Perl web interaction program for Wall. These inventions have not applied for any patents protection, but helped advance the rapid development of the Internet. All of these open source projects are still running [9]. Then, they organized a virtual community on the Web to find the evidence of prior existence, to show the patent was invalid [2]. Besides, many studies discuss why OSS can succeed [10, 11, 12] or how to use OSS to enhance the R&D capabilities of software [13, 14, 15, 16] in recent years. Has patenting computer software reinforced innovation, or prevented invention? There is no absolute answer, and to attempt to find ones is one objective of this study.

2.4 Technology Acceptance Model

Davis [17] developed the Technology Acceptance Model (TAM) based on TRA (Theory of Reasoned Action) [18] and TPB (Theory of Planned Behavior) [19]. The difference between TAM and TRA lies in two perceptual constructs: perceived usefulness and perceived ease of use in TAM. Perceived usefulness means that a person subjectively believes that using a particular technology system would enhance his/her performance. On the other hand, perceived ease of use refers to the degree to which a user believes that using a particular technology system would be easy. Another difference is that perceived usefulness and perceived ease of use can totally replace subjective norm. Nowadays, TAM has been widely applied in research to measure user behavior concerning information technology. However, followed by the high development tempo of information technology, the model has been adjusted in order to meet the need of different research purposes or hypotheses [20, 21, 22, 23, 24, 25].

This study views a computer software patent as an institutional technology system, users must have some knowledge or technique to use, apply and manage patents. Therefore, we modified TAM to explore the adoption by software engineers and software companies of this technology system, expecting to understand the perception and behavior of patent management of software engineers during the software innovation process.

3 Methodology

3.1 Population and sample

The population of this study is the software engineers in Taiwan. We selected five occupational categories in the Chinese Yellow Page Website database (http://hipage.hinet.net) as our sample list [26], to include: computer software, computer program design, computer system analysis and design, information system integration service, and internet software design. 1000 companies were randomly sampled from this list and were investigated by mail questionnaire.

3.2 Research model

The research model of this study was modified from TAM; there remained perceived ease of use, perceived usefulness, behavior intention, actual behavior four constructs, and added were knowledge of patent, dependency of patent two new constructs. The variables of every construct are measured by Likert seven-point scales. The research model is shown in Fig. 1, and the descriptions of every construct are shown in Table 1.

3.3 Hypotheses

This study proposes five hypotheses based on the research model:

H_1: The software engineer’s knowledge of patent of will have a positive effect on his dependency on the patent.

H_2: The software engineer’s knowledge of patent, his dependency on patent, and the perceived ease of use patent, will have a positive effect on the perceived usefulness of patent.

H_3: The software engineer’s knowledge of patent will have a positive effect on the perceived ease of use patent.

H_4: The software engineer’s dependency on patent, perceived ease of use patent, and perceived usefulness of patent will have a positive effect on the intention of use patent.

H_5: The intention of use patent by a software engineer will have a positive effect on the actual behavior of patent management.

Furthermore, this study is interested to differentiate the scale of firms, patents owned, and respondent attributes. These differences could cause or not cause significant divergence of constructs in the research model, such as KP (Knowledge of Patent), DP (Dependency on Patent), ABP (Actual Behavior of Patent Management) and so on. Hypotheses H_6 to H_8 are proposed.

H_6: There are significant differences in the perception and behavior of patent, because of the difference of firms' scales.

H_7: There are significant differences in the perception and behavior of patent, because of the difference of firms owned patents.

H_8: There are significant differences in the perception and behavior of patent, because of the difference of engineer attribute.

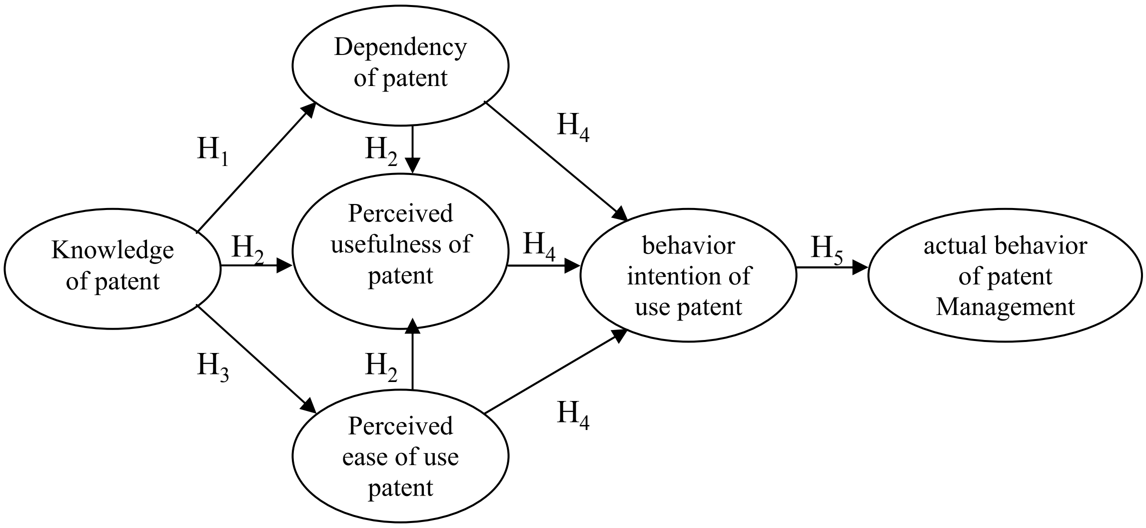

Fig. 1 Research Model

Table 1 Description of Construct

Construct	Description	Variable
Knowledge of Patent (KP)	To measure the level of software engineers understanding of the difference between copyright and patent, and understand the information and value of patents	7 items were developed by this study
Dependency on Patent (DP)	To measure the level of software engineers dependence on patent information when he works on software R&D.	4 items were developed by this study
Perceived Ease of Use patent(PEU)	To measure the perceived ease of use of software engineers when he search, read, and apply patents	4 items were modified by TAM
Perceived Usefulness of patent (PU)	To measure the perceived usefulness of software engineers when they search, read, and apply for patents during their works	4 items were modified by TAM
Intention of Use Patent (IUP)	To measure the level of software engineers willingness to search, read, apply for patents	4 items were modified by TAM
Actual Behavior of Patent Management (ABP)	To measure the actual level of patent management behavior by software engineers	4 items were modified by TAM

4 Analysis and Findings

The population of this study is software engineers in Taiwan. We mailed 1,000 questionnaires and received returns from 105 respondents, deleted the invalid 7, the effective samples are 98, (9.8%). The statistic method of this study ought to have been SEM (Structural Equation Model) to test the research model, but the sample size is not large enough. Therefore, hypotheses H_1 to H_5 were tested by multi-regression, and hypotheses H_6 to H_8 were tested by t-test and ANOVA. The analysis is separated into four parts: profiles of respondents, variables description and reliability of constructs, relational analysis and differential analysis.

4.1 Respondents' profile

The demographic statistics of respondents include gender, age, salary, experience and education; these are listed in Table 2. About the firm's profile, this study classifies by total number of employees, software engineers and the number of patents owned, these are listed in Table 3.

4.2 Variables description and constructs reliability

This sector calculated the mean and standard deviation of 27 variables within six constructs, and computed the reliability coefficient cronbach α of every construct. The results were listing in Table 4. All the cronbach α were over 0.8, indicating the reliability of construct was good. About the means, most variables of KP (Knowledge of Patent) and IUP (Intention of Use Patent) are higher (>5.0), indicated the perceptions of computer software engineers are similar to the nature of patent. However, the lowest mean of variables belonged to PEU (Perceived Ease of Use patent), implying the software engineers were insufficiently capable of managing patents. As for standard deviation, the variables of ABP (Actual Behavior of Patent Management) were highest, the phenomenon might be concerned with firms' scale and number of patents owned, as this study will discuss below.

Table 2 Demographic Statistics of Respondent

gender		age		Month salary	
male	70	20-29	26	30,000 ↓	10
female	27	30-39	43	30,001~50,000	41
		40-49	25	50,001~100,000	44
		50 ↑	3	100,001↑	3
Experience		**Education**			
1 year↓	6	High school	2		
1-3 years	16	college	28		
3-5 years	17	university	41		
5-10 years	27	master↑	27		
10-20 years	31				
21↑	1				

Table 3 Profile of Respondent Firms

Total employees		Software engineers		Number of patents	
9 ↓	24	4 ↓	30	none	50
10-29	18	5-10	24	1~5	33
20-49	29	11-20	18	6~10	8
50-149	16	21-40	14	11~20	3
150 ↑	9	41 ↑	5	21~50	2
				51 ↑	2

Table 4 Description and Reliability of Construct

construct	variables	mean	s.d.	cronbach α
KP	I understand the computer software can be protected by copyright	6.4592	0.9966	0.8272
	I understand the computer software can be protected by patent	6.0408	1.4353	
	I understand how to use copyright to protect the invention	5.2245	1.5830	
	I understand how to use patent to protect the invention	5.2245	1.5232	
	I understand the difference of copyright and patent	4.6327	1.8014	
	I understand what information are inside the patent document	3.8469	1.8851	
	I understand what information can retrieve from patent analysis	3.9388	1.8931	
DP	Patent information is important for me to develop the software	5.0714	1.8006	0.8754
	I feel studying patents can help me to develop software	5.0102	1.7080	
	Studying patents can stimulate my innovation inspiration	4.8980	1.6149	
	When I develop software, patent information is more important the other documents (eg. Textbook, report, journal paper….)	4.7813	1.9040	

PEU	I can find the necessary patent easily.	3.7041	1.7003	0.8765
	I can understand the content when I read the patent.	3.9898	1.7439	
	I feel codifying the software invention into a patent document is easy.	3.4286	1.7469	
	I feel applying for the patent is easy.	3.1340	1.6049	
PU	I feel it is usefulness for my work when I study patents	4.9175	1.5184	0.8674
	I can find the useful information when I study patents	4.7143	1.6807	
	I feel applying for software patents that is usefulness for protect new invention	5.2551	1.5486	
	I feel apply software patents can disclose my invention clearly	5.2959	1.6573	
IUP	I am willing to apply for patent to disclose my software invention	5.0918	1.6751	0.9043
	I am willing to apply for patents to protect my software invention	5.3265	1.5975	
	I am willing to spend time to research patents for understanding technology development	4.9082	1.7116	
	I am willing to spend time to research patents for understanding the details and characteristics of technology	4.8163	1.6889	
ABP	I will research similar patents before I design software	4.1633	2.0544	0.8401
	I will search similar patents before I sale the software product	4.2857	2.0358	
	My company will apply the patent to protect the invention when the software was completed	4.1959	2.0343	
	My company has fulltime staffs to handle intellectual property	3.5000	2.3031	

4.3 Relational Analysis
Because of the sample size is not large for SEM (Structural Equation Model), this study used multi-regression to test the hypotheses H_1 to H_5. The statistics summaries of multi-regression were listing in Table 5.

1. H_1: The software engineer's knowledge of patent of will have a positive effect on his dependency on the patent.

Table 5 shows a significant positive effect is existed between KP and DP, the coefficient β=0.434, p-value<0.001 and adj-R^2=18.0%.

2. H_2: The software engineer's knowledge of patent, his dependency on patent, and the perceived ease of use patent, will have a positive effect on the perceived usefulness of patent.

For this hypotheses the enter method was used to test the regression of three independent variables. There is no collinear because all tolerance > 0.1, and the adj-R^2 = 48.9% indicating the explanation power is good. However, only DP and PEU are significant among three independent variables, the other one variable KP is not significant for PU.

3. H_3: The software engineer's knowledge of patent will have a positive effect on the perceived ease of use patent.

There are significant positive effects between KP and PEU, the coefficient β=0.415, p-value<0.001 and adj-R^2=16.4%.

4. H_4: The software engineer's dependency on patent, perceived ease of use patent, and perceived usefulness of patent will have a positive effect on the intention of use patent.

For this hypotheses the enter method was also used to test the regression for three independent variables. There is no collinear because all tolerance > 0.1, and the adj-R^2 = 53.8% indicated the explanation power is well. However, only PEU and PU are significant among three independent variables, another one variables DP is not significant for IUP.

5. H_5: The intention of use patent by a software engineer will have a positive effect on the actual behavior of patent management.

There are significant positive effects between ABP and IUP, the coefficient β=0.610, p-value<0.001 and adj-R^2=36.5%.

The test results of H_1 to H_5 were shown in Fig. 2. Significant relationships are represented by bold arrows, coefficients are indicated beside the arrows; non-significant relationships are represented by dotted arrows, coefficient was omitted. Every dependent construct contains adj-R^2 represent the explanation power by independent variables.

Table 5 Statistics Summaries of H_1 to H_5

H	Depend.Var.	F	p-value	Adj-R^2	Independ. Var	β	t	p-value	TOL
H_1	DP	22.337	0.000	0.180	Intercept	2.101	3.404	0.001*	
					KP	0.434	4.726	0.000*	
H_2	PU	31.961	0.000	0.489	Intercept	1.516	3.248	0.002*	
					KP	0.047	0.557	0.579	0.741
					DP	0.597	7.148	0.000*	0.755
					PEU	0.177	2.137	0.035*	0.770
H_3	PEU	20.012	0.000	0.164	Intercept	0.922	1.524	0.131	
					KP	0.415	4.473	0.000*	
H_4	IUP	38.703	0.000	0.538	Intercept	0.732	1.768	0.080	
					DP	0.132	1.378	0.171	0.515
					PU	0.217	2.796	0.006*	0.794
					PEU	0.522	5.332	0.000*	0.497
H_5	ABP	56.766	0.000	0.365	Intercept	0.450	0.907	0.367	
					IUP	0.610	7.534	0.000*	

Note: * represents significance at α=0.05

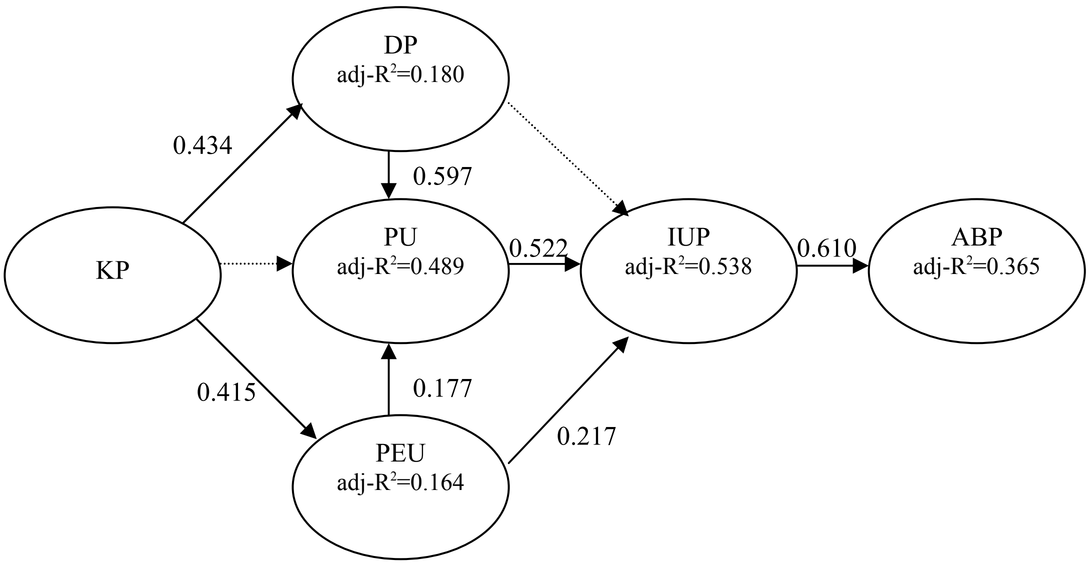

Fig. 2 Result of Research Model

4.4 Differential Analysis

On the differential analysis, this study is concerned with the difference of scale of the firms, number of patents owned, and respondent attributes. Use was made of these differences to test divergence of six constructs in research model, such as KP (Knowledge of Patent), DP (Dependency on Patent), ABP (Actual Behavior of Patent Management) and so on.

1. H_6: There are significant differences in the perception and behavior of patent, because of the difference of firms' scales.

This study used the number of employees to represent the firms' scale. Of the 76 respondents, this study select 50 employees as a cut point, means the number of employees over than 50 employees as a large firms, and less than 50 as a small firms. This hypothesis uses t-test and we list the statistics in Table 6.

On table 6, there are significant differences in DP, PEU and ABP; the p-values are 0.009, 0.033 and 0.005, respectively. These results indicate that large firms developing software products have more dependency of patent, feel more at ease about the use of patent, and are doing more patent management activities than small firms.

Table 6 t-test of Firms' Scale

Construct	Large Firms			Small Firms			Variance	t	p-value
	sample	mean	s.d.	sample	mean	s.d.	test		
KP	25	5.3771	1.1700	71	4.9477	1.1470	=	1.602	0.113
DP	25	5.6367	1.479	71	4.7394	1.5280	=	2.678	0.009*
PEU	25	4.1200	1.5328	71	3.4049	1.3851	=	2.159	0.033*
PU	25	5.4800	1.2498	71	4.9531	1.3260	=	1.734	0.086
IUP	25	5.3300	1.2263	71	4.9648	1.5094	=	1.089	0.279
ABP	25	4.8900	1.4180	71	3.8063	1.7078	=	2.844	0.005*

Note: * represents significance at α=0.05

2. H_7: There are significant differences in the perception and behavior of patent, because of the difference of firms owned patents.

Profile of respondent firms was listing in Table 3. This hypothesis separates the firms into two groups by patents owned. The firms that have at least one patent fall into one group; another group is the firms having no patent. T-test was used to test hypotheses H_7 and results was listing in Table 7. All constructs of this research model are significant differences between these two groups. These phenomena indicate that firms of patent owned have higher level of patent knowledge, more dependent on patent, feel more ease about the of use of patents, and are undertaking more patent management activities than firms having no patent.

Table 7 t-test of Owned Patent

contruct	Owned Patent			No Patent			Variance	t	p-value
	sample	mean	s.d.	sample	mean	s.d.	test		
KP	48	5.4613	1.1246	50	4.6600	1.0446	=	3.657	0.000*
DP	48	5.3628	1.2919	50	4.5450	1.5728	=	2.807	0.006*
PEU	48	4.1875	1.4463	50	2.9650	1.1856	=	4.584	0.000*
PU	48	5.3698	1.1973	50	4.7333	1.4200	=	2.394	0.019*
IUP	48	5.4323	1.3176	50	4.6550	1.5219	=	2.698	0.008*
ABP	48	5.0729	1.4412	50	3.0450	1.3436	=	7.208	0.000*

Note: * represents significance at α=0.05

3. H_8: There are significant differences in the perception and behavior of patent, because of the difference of engineer attributes.

The demographic statistics of respondents include gender, age, salary, experience and education; these are listing in Table 2. This hypothesis uses the above attributes of engineers to test the differences in KP, DP, PEU, PU, IUP and ABP. The results indicate that none of attributes could significantly affect the constructs of this study. It is means that there are no significant differences in the perception and behavior of patent, because of the difference of engineers' attributes.

5 Conclusion and Suggestion

5.1 Conclusion

From the above analysis, this study draws three conclusions:

1. This study regards patent institution as a technology system, and employs a modified TAM to verify the perception and behaviors of computer software engineers. The results indicate most of the paths of the research model are significant; it means that the perceptions and behaviors of software engineers are similar to the nature of patent in substance, and have a good fit with of modified TAM model.

2. The differences of patents owned cause significant divergence of all constructs in this research model, and the differences of firms' scale cause significant divergence of DP, PEU and ABP. Large firms and patent owning firms are stronger than small firms and no-patent firms in above constructs. These results indicate that the perception and capabilities of patents of small firms are insufficient, that will influence the actual behaviors of patents management.

3. There are no significant differences in the perception and behavior about patent because of the difference of engineers' attributes. The implication is that the demographics of software engineers are not important in term of the usage or management of patents. Comparing with above two conclusions, we can propose that a scale, system and culture of firms are more important factors for the perception and behavior of software engineers in using or managing patents.

5.2 Suggestions

This study discusses the perception and behavior of computer software engineers about patents. However, the effect of patent institution influence on knowledge innovation and technology diffusion is another important issue. That is also concerned with the perception and behavior of computer software engineers. Therefore, this study provides two suggestions for future study.

1. The willingness and capability of codifying patent. Patent is a carriage of knowledge and technology. If one engineer creates a new technology, he can decide for what scope of the invention to apply for patent or not to apply at all. In another hand, the capability of codifying technology is a critical problem; that capability directly influences the details and clarities of patent. Therefore, willingness and capability of codifying patent is an issue of major interest for future study.

2. The absorptive capacity of software engineers for patent knowledge. The above suggestions focus on codification, that is, the transfer of tacit knowledge to explicit knowledge [27]. This suggestion is the inverse of that issue. Software engineers get the explicit knowledge from a patent, and transfer that to his tacit knowledge. Therefore, the absorptive capacity will influence the knowledge innovation; that is another interesting issue of future study.

References

[1]. USPTO, *Examination Guidelines for Computer-Related Inventions*, 1996.
[2]. Kevin G. R., David K., *Rembrandts in the Attic: Unlocking the Hidden Value of Patents,* Harvard Business School Press, 2000.
[3]. Mansfield, E., Patents and Innovation: an Empirical Study, *Management Science*, Vol.32, 1986, pp.173-181.
[4]. Lai, K.-K., Chang, S.-B., Wu, S.-C., Exploring Technology Competencies and Technology Position Relative to Business Method from US Patent Database, *Journal of Technology Management*, Vol.8, No.2, 2003, pp.91-122, Taiwan, ROC.
[5]. Chang, S.-B., Lai, K.-K., Chang, S.-M., Exploring Technology Diffusion and Classification of Business Methods: Using the Patent Citation Network, *Technological Forecasting and Social Change*, Vol.76, 2009, pp.107-117.
[6]. IPO, *Examination Guidelines for Computer Software Related Inventions*, MOE in Taiwan, 1998.
[7]. http://www.gnu.org/..., The website of an organization of free software. Visited in 2008.

[8]. Wayner P., *Free for all: How Linux and the free software movement undercut the high-tech titans*, Business Weekly Publications, 2000.
[9]. Laat, Paul B. de, Copyright or copyleft? an Analysis of Property Regimes for Software Development, *Research Policy*, Vol.34, 2005, pp.1511-1532.
[10]. Bonaccorsi, A., Cristina, R., Why Open Source Software can Succeed, *Research Policy*, Vol.32, 2003, pp.1243-1258.
[11]. Comino, S., Manenti, F. M., Parisi, M. L., From Planning to Mature: On the Success of Open Source Projects, *Research Policy*, Vol.36, 2007, pp.1575-1586.
[12]. Sohn, S. Y., Mok, M. S., A Strategic Analysis for Successful Open Source Software Utilization based on a Structural Equation Model, *The Journal of Systems and Software*, Vol.81, 2008, pp.1014-1024.
[13]. Zhao L., Elbaum, S., Quality Assurance under the Open Source Development Model, The *Journal of Systems and Software*, Vol.66, 2003, pp.65-75.
[14]. West, J., How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies, *Research Policy*, Vol.32, 2003, pp.1259-1285.
[15]. Cerri, D., Fuggetta, A., Open standards, open formats, and open source, *The Journal of Systems and Software*, Vol.80, 2007, pp.1930-1937.
[16]. Sowe, S. K., Stamelos, I., Angelis, L., Understanding knowledge sharing activities in free/open source software projects: an empirical study, *The Journal of Systems and Software*, Vol.81, 2008, pp.431-446.
[17]. Davis, F.D., Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, *MIS Quarterly*, Vol.13, No.3, 1989, pp.319-340.
[18]. Fishbein, M. and I. Ajzen, *Belief, Attitude, Intentions and Behavior: An Introduction to Theory and Research*, Addison-Wesley, Boston, MA., 1975.
[19]. Ajzen, I., *From Intention to Actions: A Theory of Planned Behavior*, in J. Huhl and J. Bechmann (Eds), Action-Control: From Cognition to Behavior, Springer, Heidelberg, 1985, pp.11-39.
[20]. VenKatesh, V. and F.D. Davis, A Model of the Antecedents of Perceived Ease of Use: Development and Test, *Decision Sciences*, Vol.27, No.3, 1996, pp.451-481.
[21]. Agarwal, R. and J. Prasad, Are Individual Differences Germane to the Acceptance of New Information Technologies, *Decision Sciences*, Vol.30, No.2, 1999, pp.361-391.
[22]. Hung, S.-Y., Chang, C.-M., User acceptance of WAP services: test of competing theories, *Computer Standards & Interfaces*, Vol.27, 2005, pp.359-370.
[23]. Shin, D.-H., User acceptance of mobile Internet: Implication for convergence technologies, *Interacting with Computers*, Vol.19, 2007, pp.472-483.
[24]. Kim, T. G., Lee, J. H., Law, R., 2008, An empirical examination of the acceptance behaviour of hotel front office systems: An extended technology acceptance model, *Tourism Management*, Vol.29, 2008, pp.500-513.
[25]. Polancic, G., Hericko, M., Rozman, I., An empirical examination of application frameworks success based on technology acceptance model, *The Journal of Systems and Software*, Vol.83, 2010, pp.574-584.
[26]. http://hipage.hinet.net/...., The website of Chinese yellow page. Visited in 2008.
[27]. Nonaka, I. & H. Takeuchi, *The Knowledge-Creating Company*, Oxford University Press, 1995.