基于神经网络的软件外包过程模式决策研究[①]

蒋严冰[1] 周 欣[2] 刘 英[2] 胡敬坤[1]

(1.北京大学 软件与微电子学院 100871；2. IBM 中国研究院 100094)

Study on Neural Network-Based Software Outsourcing Process Pattern Decision

Yanbing Jiang (School of Software and Microelectronics, Peking University, Beijing 100871)

Xin Zhou, Ying Liu (IBM China Research Lab, Beijing 100094)

Jingkun Hu (School of Software and Microelectronics, Peking University, Beijing 100871)

Abstract: Software outsourcing has became a common method used in software development. To guide the software outsourcing process pattern decision, this paper puts up a neural network based decision model which is made up of a group of orthogonal and concurrent input factors, a three-tier neural network decision architecture and a group of output process patterns. The model is verified with a real software outsourcing process pattern decision case. This model provides the theory for the development of outsourcing decision tools and complete decision supporting mechanism for various outsourcing model in every phase of software engineering.

Key words: software outsourcing; neural network; decision model; outsourcing process pattern

1 Introduction

In recent years, software outsourcing has become an effective way to reduce cost, improve software productivity, and concentrate limited resource on main business in most international software companies. With more and more companies being aware of the potential of software outsourcing, the market of software outsourcing grows rapidly. According to the prediction of IDC, the increase rate of global application software outsourcing market would be 29.2% per year[1]. About one third of global software production value is achieved by outsourcing. Software outsourcing has become one of important trends of international software industry.

With the development of software outsourcing, studies on the classification, management, decision-making method of software outsourcing has been mentioned in many references. In summary, the studies can be classified as follows.

① Studies on the types of software outsourcing behavior conducted by software enterprises. For example, reference [2] believes that companies in America, Japan and the Union of Europe apply reverse T-type, pyramid-type and olivary-type outsourcing behavior pattern respectively by comparing their organization models.

② Studies on the factors and strategies that determine whether to choose outsourcing or not. References [3-5] put forward the decision strategies that determine whether the whole system or some parts of it need to be outsourced and whether a candidate vendor is competent for the outsourcing work.

③ Studies on the software outsourcing process. Related references try to establish the process model of

① Supported by IBM University Cooperation Program of 2008

software outsourcing to guide the practice of outsourcing. For example,Ref.[6] discusses the software products, the phases, and major activities in the process of outsourcing. Ref.[7] presents a management model for the whole outsourcing process. Ref.[8] presents three models that describe the cooperation and the division of works within each software engineering phase and highlight the communication pattern between clients and vendors. Ref.[1] discusses Japanese outsourcing model of software engineering. All above presented software outsourcing process models focus on the work division between clients and vendors within each software outsourcing engineering phase, and are instructive in engineering practice.

④Studies on the technique problems in software outsourcing, such as the communication and collaboration among teams from different areas, how to deal with the zone difference, how to deal with the cultural difference, how to keep the confidentiality and etc[9].

⑤Studies on the influential factors of software outsourcing. The Refs.[10-12] believe that critical factors which influence outsourcing decision are difficult to be completely found out because many aspects need to be taken into account, such as management, economics, and software engineering, etc. Ref.[10] describes fourteen factors that influence the success or failure of a special software outsourcing project. Moreover, this paper presents a special decision model based on Structural Equation Modeling. Ref.[12] applies fuzzy decision method to decision model according to Project Related (PR), Technology Related (TR) and Strategy Related (SR) factors and Onsite (ON), Offshore (OF) and in-house (IH) patterns.

The reference review above reveals that studies on software outsourcing focus on the pre-outsourcing stage and investigation stage, but little work can be found on the decision in software outsourcing engineering stage. Moreover, the outcome decision result is very simple, the input factors are missing and the decision process heavily relies on data mining technology.

Based on the existing studies and experiences, this paper presents a set of input factors. Based on the real cases study and classification of software process model, this paper presents complete output model of decision.

As input factors are orthogonal and concurrent, we select neural network as the base to establish a decision model that maps the input factors to the recommended software outsourcing process pattern.

We verified this model with practical cases. This model provides the theory base for the development of outsourcing decision tools and complete decision supporting mechanism for various outsourcing model in every phase of software engineering.

2 An Overview of Proposed Approach

2.1 A brief introduction to neural network

Neural network is a decision model that simulates the behaviors of neural cells of biological system. The early study on neural network can be traced back to the 20th century. Since 1986, large numbers of neural network papers have been published. Compared to other decision models, neural network has lots of advantages such as orthogonal feature, concurrent processing, self-learning etc.

Artificial Neural Network, which is made up of many interconnected artificial neurons, simulates the way that biological nervous systems process input the information. An artificial neuron is a cell which has many inputs and one output and deduces an output decision from these inputs. The key component of the neuron is the stimulation function which can transfer the inputs into a single output, 0 or 1. To deduce complicated decision, many neurons can be interconnected to form an artificial neural network in which the output from some neuron can be used as an input factor to other neuron.

2.2 Framework of proposed approach

This paper applies the theory of artificial neural network to constructing the decision model of software outsourcing. The inputs are factors that influence the software outsourcing process pattern. The output is the recommended process model that presents the division of work in the software outsourcing process.

Fig.1 shows a neural network-based software outsourcing process pattern decision model which consists of three-tie neural cells. The inputs of the model are influential factors. The neural cells in the first layer independently process input factors to form a preliminary process pattern decision, which briefly describes division of work between clients and vendors in software development life circle. The second level takes the outputs from the first layer and original inputs as input factors to deduce a more detailed process pattern decision for a certain phase in software life circle. The third layer sums up the final decision for the all phases in software life circle on the basis of each phase decisions from the second layer.

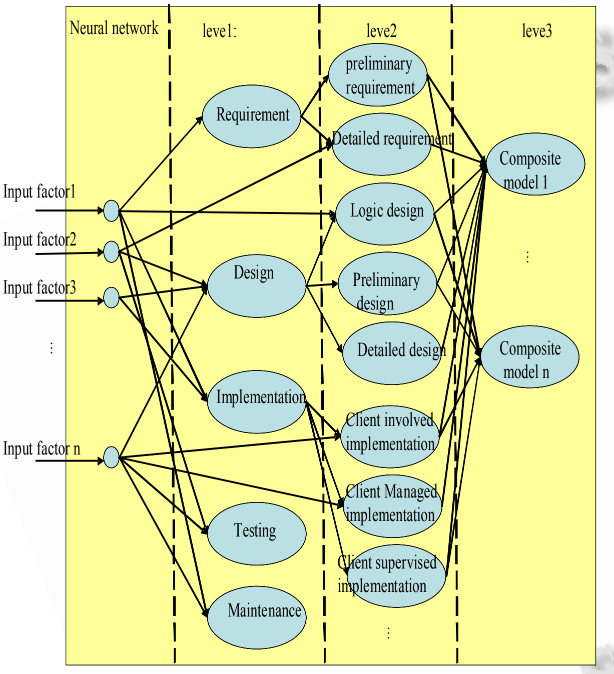

Fig.1　Overview of neural network based software outsourcing process pattern decision model

3　The Influential Factors Set of Software Outsourcing Process Decision

Related references have different definitions on the input factors in the decision of software outsourcing process. This paper presents a set of input factors, which are listed in related references or considered as a real influential factor by our practice and analysis, as the inputs of software outsourcing process pattern decision model. Table1 lists these 24 factors, which can be divided into three categories.

Table 1　The influential factors set of software outsourcing process decision

Type	NO.	Factor	Comment
Technical factors	f1	Software complexity	[10][12]
	f2	Software scale	[10][12]
	f3	Origin of requirement	client/salesman /expert.
	f4	Comprehensible of requirement	[10]
	f5	Variability of requirement	[10]
	f6	Reuse of software	
	f7	Reusability of software	
	f8	Type of software	Product or system.
	f9	Dependency on the private platform	Depending on the private-owned software platform
	f10	Independence	integrating problems with client or other systems
Manage-ment factors	f11	Time limit	[10]
	f12	Relative cost advantage	[10]
	f13	Capability of monitoring vendors	[10][12]
	f14	Confidentiality	
	f15	Communication Capability	[10][12]
The Client's capability of develop-ment[15]	f16	Capability of business modeling	Can be described in terms of person, tool, technology, method and experience, etc.
	f17	Capability of requirement analysis	
	f18	Capability of system analysis	
	f19	Capability of building system architecture	
	f20	Capability of preliminary design	
	f21	Capability of detailed design	
	f22	Capability of implementation	
	f23	Testing Capability	
	f24	System maintenance Capability	

4 Software Outsourcing Process Patterns

The outcome of the decision model is software outsourcing process pattern, which describes the division of work in software life circle between client and vendor. In detail, the process mode should include personnel, techniques, working flows, and artifacts provided by one side of clients and vendors or by the cooperation of the two sides. The detailed description of software out-sourcing process model is out of the scope of this paper. The brief software outsourcing process patterns is descryibed in Table 2. According to the phases in software life circle that the two sides of clients and vendors might be involved in, software outsourcing process pattern framework can be divided into five first class patterns, which can be divided into many sub patterns as shown in Table 2. Because the outsourcing decision of each phase in software life circle has little interrelationship with each other, each pattern is regard as orthogonal to others and can be decided by a neural cell.

Table 2 The brief Software outsourcing process patterns

Pattern name	Description	Sub- pattern	Sub-pattern description
Requirement Outsourcing	It is necessary for clients to distribute all or part of requirement analysis to vendors in the requirement phase	Preliminary requirement	Client put up the preliminary requirement, leaving the detailed requirement to vendors.
		Complete requirement outsourcing	All the work of requirement is given to vendors.
Design Outsourcing	It is necessary for clients to distribute all or part of design to vendors in the design phase	Business logic design	Client designs business logic
		Architecture Design	Client designs the overall technique architecture
		Complete design outsourcing	All the work of design is given to vendors.
Implementation Outsourcing	It is necessary for clients to distribute all or part of implementation to vendors in the phase of implementation	Client invoved implementation	Client and vendor implement together.
		Client managed implementation	Client assigns duties, monitors quality and schedule
		Client supervised implementation	Client monitors the progress of implementation periodically, and gives advice
		Complete implementation outsourcing	All the work of implementation is given to vendors.
Test Outsourcing	It is necessary for clients to distribute all or part of testing work to vendors in the phase of testing	Client managed testing	Client make test schedule and cases
		Complete testing outsourcing	All the work of testing is given to vendors.
Maintenance Outsourcing	It is necessary for clients to distribute outsource all or part of maintaining work to vendors in the phase of Maintenance	Client involved maintenance	Client maintains simple parts，Vender maintains the difficult parts
		Complete maintenance outsourcing	All the work of maintenance is given to vendors.

5 The Neural Network-Based Software Outsourcing Process Pattern Decision Model

5.1 The building method of the neural network-based decision model

5.1.1 Establishing stimulation function of a neuron

The key problem of building neural network based decision model is to establish the stimulation function which can transfer the values of inputs into a single value, 0 (con) or 1 (pro). The implementation idea of the stimulation function can be described as follows:

At first, each factor related to some decision is considered to decide if it has pro or con impact to the decision. Then some integration pattern is used to form a general decision. The integration pattern might be one of the following:

① Sum of weight. Each of the related factors is given a weight, which is often decided by statistics of many questionnaires. If the weighted superposition of all the factor values exceed some pre-defined threshold, the final outcome is 1, otherwise it is 0.

② Vote down. If one of the related factor values is 0, then the final outcome is 0. If and only if all of the related factor values are 1, then the final outcome is 1. This pattern is suitable for the context where all the factors are essential to make the pro decision.

③ Election. The final outcome is determined by the balance between the pro factors and the con ones. This pattern is simply form of the first pattern. In this paper, because all the neurons are established by means of the deduction and summary of real practice in software outsourcing, the weight of each factor is hard to be estimated without data mining, we choose this pattern as the main integration pattern.

5.1.2 Establishing the model of a single neuron

The steps that establish the model of a single neuron are listed as follows:

① Deciding the related factors that should be taken into account as the input of the neuron.

② Considering if the related factors are con or pro to the decision neuron.

③ Choosing the suitable integration pattern in the stimulation function.

5.1.3 Establishing the model of a neural network

A neural network is a set of neurons. Neurons in a neural network can have two relationships:

① Concurrent. The two neurons have no sequence relationship, and can be deduced concurrently.

② Sequence. The output of one neuron is just an input of other one.

The three-tie neural network in Fig.1 can be established by the two relationships.

The rest of the chapter describes the part of the three-tie neural network.

5.2 The neural cells in the first layer

The first layer in the three-tie neural network consists of five neural cells. The paper only takes the requirement and design neuron as examples for the limitation of space.

Table 3　Related input factors of requirement decision neuron

NO.	Influential factor	val -ue	Pro or con	Type	Reason
FR1	Easy expression of requirement	1	+	Promotion	It is easy to transfer requirement information between client and vendor
FR2	Variability of requirement	1	-	Promotion	Increasing the outsourcing cost
FR3	Short development period	1	+	Promotion	Decreasing the Development period by distributing requirement to vendors.
FR4	Communication capability	1	+	Promotion	Improving efficiency
FR5	Relative cost advantage	1	+	Promotion	Saving cost
FR6	Capability of monitoring vendor	1	+	Promotion	Ensuring the quality of software system
FR7	Confidentia-lity	1	-	Necessary	Can not let vendors know the requirement information
FR8	Requirement analysis capability	0	+	Necessary	Need to acquire req-uirement with the help of vendors

The stimulation function of the requirement decision neuron is listed as follows:

$R=\neg FR7\wedge(FR8\vee vote(FR1-\ FR2+\ FR3+\ FR4+\ FR5+\ FR6))$, where

$$vote(x)=\begin{cases}1,\ x>0\\ 0,x<=0\end{cases}$$

Table 4 Related input factors of design decision neuron

NO.	Influence factor	val ue	Pro or con	Type	Reason
FD1	Software complexity	1	-	Promo-tion	Leading to the difficulty of contro-lling system de sign.
FD2	Software scale	1	+	Promo-tion	Saving cost
FD3	Software reuse	1	-	Promo-tion	Vendor may not know about client's reuse technology
FD4	Software reusability	1	-	Promo-tion	It is complicated to transfer the consideration of software reusability to venders.
FD5	Software product development	1	-	Promo-tion	The quality of software products is hard to control.
FD6	Dependency on privateowned platform	1	-	Promo-tion	Vendor may not know much about the client` private-owned platform
FD7	Short development period	1	+	Promo-tion	To shorten project time by outsourcing
FD8	Communica tion capability	1	+	Promo-tion	Improving efficiency
FD9	Relative cost advantage	1	+	Promo-tion	Cost down
FD10	Capability of monitoring vendor	1	+	Promo-tion	Ensuring the quality of software system
FD11	Confiden-tiality	1	-	Neces-sary	Can not let vendors know the requirement information
FD12	Design capability	0	+	Neces-sary	Need to acquire design with the help of vendors

The stimulation function of the design decision neuron is listed as follows:

$D=\neg FD11\wedge(vote(-FD1+\ FD2-\ FD3-\ FD4-\ FD5-\ FD6+FD7+FD8+FD9+FD10+FD12))$, where

$$vote(x)=\begin{cases}1,\ x>0\\ 0,x<=0\end{cases}$$

5.3 The neural cells in the second layer

The second layer in the three-tie neural network includes about ten neural cells. The paper only lists the detailed requirement and design neuron as showed in table5 for the sake of space limitation.

Table 5 The detailed requirement and design neurons

Preliminary requirement decision neural cell					
NO.	Influence factor	value	Pro or con	Type	Reason
R	Requirement outsourcing	1	+	Neces-sary	Necessary condition
R1	Detailed requirement Capability	0	+	Neces-sary	Client without Detailed requirement Capability would resort to vendor
stimulation function: $R'=R\wedge\neg R1$					
Detailed requirement decision neuron					
R	Requirement outsourcing	1	+	Necessary	Necessary condition
R1	Requirement Capability	0	+	Necessary	Client without requirement Capability would resort to vendor
stimulation function: $R'=R\wedge\neg R1$					
Logic design decision neural cell					
D	Design outsourcing	1	+	Necessary	Necessary condition
D1	Familiarity to business	1	+	Promotion	-
stimulation function: $D'=D\wedge D1$					

Architecture design decision neuron					
D	Design outsourcing	1	+	Necessary	Necessary condition
D1	Familiarity to technique	1	+	Promotion	-
stimulation function: D'= D∧D1					
Complete design decision neuron					
D	Design outsourcing	1	+	Necessary	Necessary condition
D1	Familiarity to business	1	+	Promotion	
D2	Familiarity to architecture	0	+	Promotion	
stimulation function: D'= D∧D1∧¬D2					

5.4 The neural cells in the third layer

The third layer draws out the final decision suggestion which sum up all the outsourcing decision of each software phase from the second layer. The detailed description is omitted.

6 Example

The decision model presented in this paper can be verified with a practical outsourcing decision case described below:

A government IT agent planned to develop a security management software platform. The related decision factors are listed in Table 6.

Table 6 The related decision factors in developing the security management software platform

Type	NO.	factor	value
Technical factors	f1	Software complexity	1
	f2	Software scale	1
	f4	comprehensible of requirement	1
	f5	variability of requirement	0
	f8	Type of software	0
	f9	Dependency on private-owned platform	0
	f10	Independence	1
Management factors	f11	Time limit	1
	f12	Relative cost advantage	0
	f13	Capability of monitoring vendors	1
	f14	Confidentiality	1
	f15	Communication Capability	1
The Client's capability of development[	f16	Capability of business modeling	1
	f17	Capability of requirement analysis	1
	f18	Capability of system analysis	1
	f19	Capability of building system architecture	1
	f20	Capability of preliminary design	1
	f21	Capability of detailed design	0
	f22	Capability of implementation	1
	f23	Testing Capability	0
	f24	System maintenance Capability	1

We deduce from these inputs by using the decision model described in Section 5.

In the first layer, the judgment of requirement outsourcing:

R=¬FR7∧(FR8∨vote(FR1- FR2+ FR3+ FR4+ FR5+ FR6))=0;

Because the system belongs to security area, all the requirement information should be kept secret. Although several other factors promote the conclusion of requirement outsourcing, the requirement should not be distributed to vendors because the confidentiality of the system is a negative necessary factor.

The judgment on the design outsourcing：D=¬FD11 ∧ (vote(-FD1+FD2-FD3-FD4-FD5-FD6+FD7+ FD8+FD9+FD10+FD12))=0, As the same reason in the judgment of requirement outsourcing, it is impossible to implement design outsourcing

The judgment of testing outsourcing and maintenance outsourcing can be carried through in the same way.

In the second layer, because the client have capability of software development, it is necessary for the client to know the technical details. So we select the Client involved implementation pattern.

In the third layer, we draw out decision from the outcome decisions of the first layer and the second layer.

The final outcome decisions are listed in Table 7.

Table 7 The final outcome decisions

Phase	Recommended software outsourcing process pattern
Requirement	None
Design	None
Implementing	Client involved implementation
Testing	Complete testing outsourcing
maintenance	Client involved maintenance

In fact, the software outsourcing process pattern deduced from the decision model is basically the same as the pattern that come into being after long term of discussion and experiment.

7 Conclusion and Expectation

In this section, we put forward a neural network -based decision model which maps input factors to software outsourcing process pattern and verified the model with a real case. This paper provides complete decision supporting mechanism for software outsourcing process decision in every phase of software life circle.

The succeeding work will be concentrated on how to improve the accuracy of the decision model and to develop software outsourcing process decision tools. We will do lots of questionnaires from the practical project to mine out useful information on the really influential factors and the experience of the design of decision model. We will also use data mining and machine learning technique to optimize the decision model.

Acknowledgment This paper is supported by IBM University cooperation program.

References

1 Ma R. China Software Outsourcing white book. Software Engineer, 2004(1):32－33.

2 Tan LW, Tian BF. Comparative Research on Offshore Service Outsourcing Pattern for Transnational Corporations in U. S., Japan & E. U. China Soft Science, 2006,20(5):128－134.

3 Klepper R, Jones W. Outsourcing Information Technology Systems and Services. Upper Saddle River, New Jersey: Prentice Hall, 1997:5－6.

4 Halvey JK, Melby BM. Information Technology Outsourcing Transactions Process, Strategies, and contracts.Pap/Dsk ed. Hoboken: Wiley press, 1996:23－24.

5 Ian Tho. Managing the Risks of IT Outsourcing, First ed.Linacre House, Jordan Hill, Oxford: Elsevier Butterworth-Heinemann, 2005:8－10.

6 Yalaho A, Nahar N. The ICT-Supported Unified Process Model of Offshore Outsourcing of Software Production: Exploratory Examination and Validation.In:PICMET 2007 Proceedings, CD, 2007: 1466－1483.

7 Meng BG, Su Q. Research on the Management Processes of Software Outsourcing. China Soft Science, 2004,18(3):90－93.

8 Nissen HW. Designing the Inter-Organizational Software Engineering Cooperation: An Experience Report. Proc. of the Int'l Workshop on Global Software Development, International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotl- and, 2004:24－27.

9 Hohenberger S, Lysyanskaya A. How to Securely Outsource Cryptographic Computations. Kilian J, Ed. Proceedings of Theory of Cryptography, Second Theory of Cryptography Conference. Springer-Verlag Berlin Heidelberg, 2005:264－282.

10 Wada Y, Nakahigashi D, Hiroshi Tsuji. An Evaluation Method for Offshore Software Development by Structural Equation Modeling. Proceedings of SEAF OOD 2007. Springer-Verlag Berlin Heidelberg, 2007, 128－140.

11 Oza N, Hall T, Rainer A, Susan Grey. Critical Factors in Software Outsourcing–A Pilot Study. Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research, ACM, USA, 2004:67－71.

12 Mikhailov L, Tsvetinov P. Fuzzy Approach to Outsourcing of Information Technology Services. Proceedings of the 2005 ACM Symposium on Applied Computing, ACM, USA, 2005:94－98.