

PSP Practice Support System Using Defect Types based on phenomenon

Daisuke Yamaguchi¹, Ayahiko Niimi², Fumiyo Katayama¹ and Muneo Takahashi¹

1: Toin University of Yokohama, 1614 Kurogane-cho, Aoba-ku, Yokohama, JAPAN
2: Future University-Hakodate, 116-2 Kamedanakano-cho, Hakodate, Hokkaido, JAPAN
yamaguti@intl.toin.ac.jp

Abstract: In this paper, we propose the PSP Practice Support System using Defect Types based on Phenomenon. This system can transmit programming to specific human among many software processes using a Multiagent technology. The system is also synthesized to do parallel and cooperative proposing internally. Applying the proposed method to a personal process-removing task, a flexible programming for quality of software. Software developments depend on information, which is possible to collection of personal process. Agent planning has get use working data on user action and other communication. Therefore collection of all user data is necessary for agent learning. Agent studies the best transmission programming, planning and quality according to the makes planning in the personal process.

Keywords: Multi-Agent System, Personal Software Process, Software Engineering, Artificial Intelligence

I. INTRODUCTION

Software architecture has emerged as an important sub discipline of software engineering [1]. PSP support system is built using this. Moreover, We think that the data inputted can acquire software development process by sorting out using a user action record table [2].

In this paper, the PSP practice support system using Defect Types based on Phenomenon. This PSP practices Support System based on Multiagent Techniques [3]. Generally, software process data is complicated, and when building a support system using such data including some action time, the calculation with expression is difficult in many cases. Then, the PSP systems [2] configuration from a data pattern is effective using the Machine Learning who is excellent in pattern recognition to such a problem.

The system is also synthesized to do parallel and cooperative proposing internally. Applying the proposed method to a personal process-removing task, a flexible programming for quality of software. Software developments depend on information, which is possible to collection of personal process. Agent planning has get use working data on user action and other communication. Therefore collection of all user data is necessary for agent learning. Agent studies the best transmission programming, planning and quality according to the makes planning in the personal process.

II. Intelligent Agents and Multiagent System

Artificial Intelligence (AI) has made great strides in computational problem solving using explicitly represented knowledge extracted from the task. If we continue to use explicitly represented knowledge exclusively for computational problem solving, we may never computationally accomplish a level of problem solving performance equal to humans. From this idea, the paper describes the development of a multiagent system that can be used to support the assessment of design performance in the cellular automata model. Agents represent objects or people with their own behavior, and take the structure of cellular automata lattice.

Intelligent agents and multiagent systems are one of the most important emerging technologies in computer science today [4]. The advent of multiagent systems has brought together many disciplines in an effort to build distributed, intelligent, and robust applications. They have given us a new way to look at distributed systems and provided a path to more robust intelligent applications.

Multiagent systems deal with coordinating intelligent behavior among a collection of autonomous agents. Emphasis is placed on how the agents coordinate their knowledge, goals, skills, and plans jointly to take action or to solve problems. Constructing the multiagent systems is difficult [5,6]. They have all the problems of traditional distributed and concurrent systems plus the

additional difficulties that arise from flexibility requirements and sophisticated interactions.

III. Personal Software Process

The Personal Software Process (PSP) is a self-improvement process that helps you to control, manage, and improve the way you work. It is a structured framework of forms, guidelines, and procedures for developing software [2]. Properly used, the PSP provides the data you need to make and meet commitments, and it makes the routine elements of your job more predictable and efficient.

The PSP's sole purpose is to help you improve your software engineering skills. It is a powerful tool that you can use in many ways. For example, it will help you manage your work, assess your talents, and build your skills. It can help you to make better plans, to precisely track your performance, and to measure the quality of your products. Whether you design programs, develop requirements, write documentation, or maintain existing software, the PSP can help you to do better work.

Rather than using one approach for every job, you need an array of tools and methods and the practiced skills to use them properly. The PSP provides the data and analysis techniques you need to determine which technologies and methods work best for you. PSP write several program using the evolving process shown Figure 1.

The PSP also provides a framework for understanding why you make errors and how best to find, fix, and prevent them. You can determine the quality of your reviews, the defect types you typically miss, and the quality methods that are most effective for you.

After you have practiced the exercises in this book, you will be able to decide what methods to use and when to use them. You will also know how to define, measure, and analyze your own process. Then, as you gain experience, you can enhance your process to take advantage of any newly developed tools and methods.

The PSP is not a magical answer to all of your software engineering problems, but it can help you identify where and how you can improve. However, you must make the improvements yourself. PSP0 and PSP0.1 hierarchy include introduces process discipline and measurement. PSP1 and PSP1.1 hierarchy include introduces estimating and planning. PSP2 and PSP2.1 hierarchy include Introduces quality management and

design. Team Software Process exists over the PSP hierarchies.

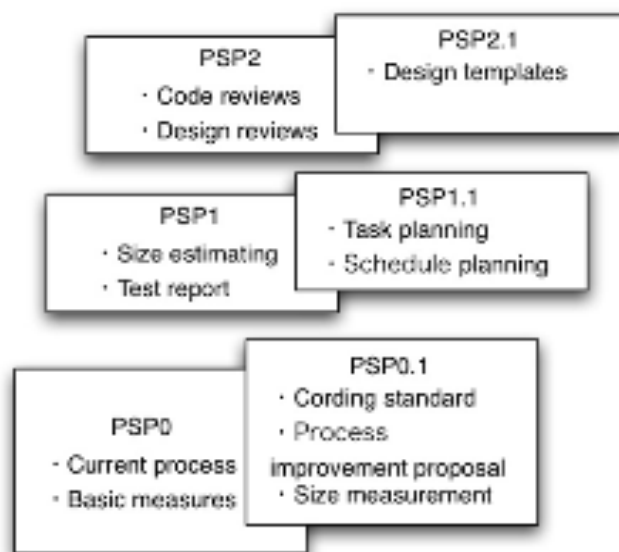


Fig.1. PSP Process Evolution

IV. The PSP Practice support system using Multiagent

In this section, we study combined as it occurs in genetic Techniques into agent learner [2]. We used as a tool for searching wide and complex solution space in Intelligent Agent learns data. Intelligent agent using complex techniques of related research. Multiagent is state in a filed shown Figure 2.

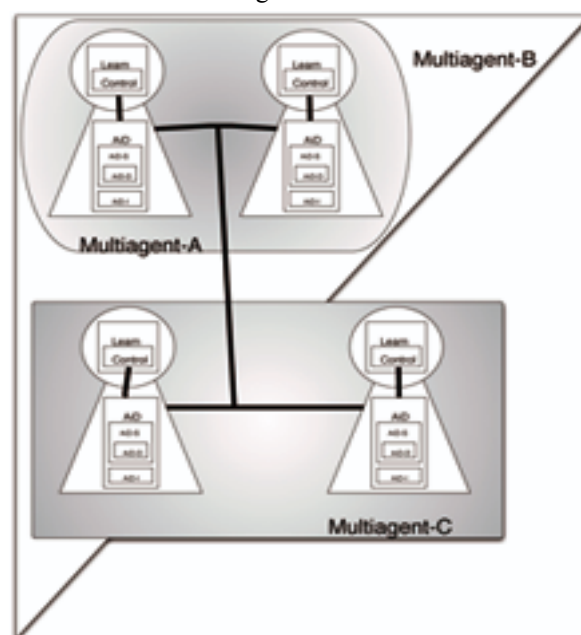


Fig.2. Support System Communication of Multiagent

Figure 2 depicts the Agent Communication Module and shared Information Data. The Agent make filed in order to share information data from Agent communication filed. These fields include other Learner kept in Intelligent Agent shown Figure 3.

Figure shows the Agent between communication modules in other communicate method. In this case, Intelligent Agent supports the PSP time and size measures record to user manipulation data. Intelligent Agent Controller selects Agent Information Data Share (AiD-S) or Agent Information Data Delivery (AiD-D) [3].

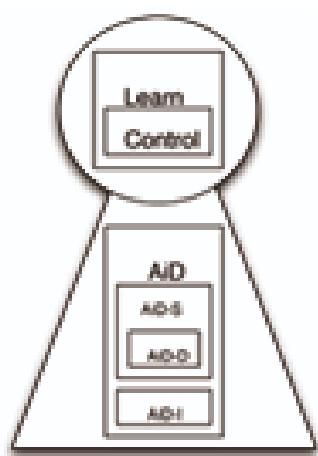


Fig.3. The Configuration of Agent Module

Other Learner support anything AI techniques of input data. Intelligent Agent has made combined these techniques into the Machine Learning. Machine Learning include same function of standard algorithm using user analyses data. These techniques supported by analysis data in time sheet that retrieval of start and end point.

Table 1 shows the PSP record form Time Measures and Size Measures [2].

Table 1. The Sample data scale of Program size categories No.X

	Plan	Results	Accumulation
Base	70	100	+100
Added	5	35	+35
Modified	0	3	+3
Deleted	0	8	+8
New and changed	0	0	0
Re-used	20	40	+40
New Re-used	0	20	+20

Total			
-------	--	--	--

In the PSP, engineers use the time recording log to measure the time spent in each process phase. In this log, they note the time they started working on a task, the time when they stopped the task, and any interruption time. For example, an interruption would be a phone call, a brief break, or someone interrupting to ask a question. By tracking time precisely, engineers track the effort actually spent on the project tasks. Since interruption time is essentially random, ignoring these times would add a large random error into the time data and reduce estimating accuracy.

Since the time it takes to develop a product is largely determined by the size of that product, when using the PSP, engineers first estimate the sizes of the products they plan to develop. Then, when they are done, they measure the sizes of the products they produced. This provides the engineers with the size data they need to make accurate size estimates. However, for these data to be useful, the size measure must correlate with the development time for the product. While a line of code (LOC) is the principal PSP size measure, any size measure can be used that provides a reasonable correlation between development time and product size. It should also permit automated measurement of actual product size.

So, This any measure record to support agent consider with using this Agent Learner expanded of PSP support. A person engaging in a person who experienced PSP and software development for many years is not very worried about a form record-keeping work.

Record keeping is vague, and what is performed of a person pressed by a work still increases. Necessity to perform automatically is important in a soldier, remission of an activity and process assay to record an activity precisely.

Therefore I record all activities, and a support system shares the documentary information, and Intelligent Agent examines to whether be content which documentary information to shows personal characteristic of difference with another person.

V. Improvement of Software Estimate Efficiency Centered Multiagent

In this section, explain improvement of Software Estimate used to Multiagent internal Agent Learner for

Intelligent Agent. Multiagent connects in other Intelligent Agents. Hence, that Intelligent Agent put the Agent Learner on necessary thoughts in Multiagent [7] .

1. Software Estimate Design of Agent Learner

The Software Design Estimate kept in Intelligent Agent. Figure 4 shows the Agent internal Data, PSP database and user logs connection modules in other communication method. In this case, Intelligent Agent supports the PSP time and size (LOC: Line of Code) measures record to user manipulation data. Intelligent Agent used to learning Control on internal database for AiD.

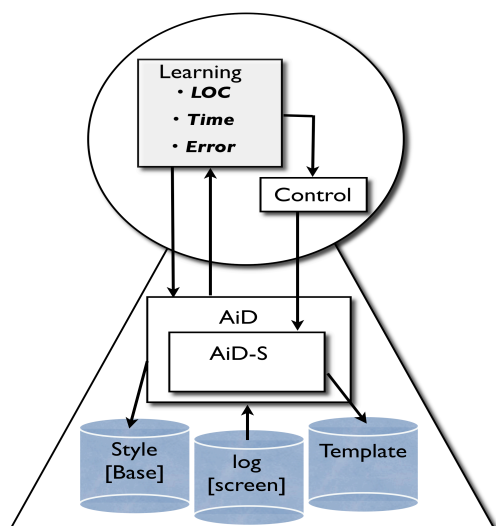


Fig.4. The AiD Data transferred from Agent Learner to action logs

2. The Estimating Probe Method of Agent Software Design

The Probe Method guides user in using historical data to make estimates. With estimated proxy size E , Intelligent Agent can calculate the projected program size P and did total estimate development time. The parameters β_0 and β_1 are used in the following equation to calculate projected added modified size:

$$\text{Projected Added \& Modified size } P = \beta_0 + \beta_1 * E \quad (1)$$

When two sets of data are strongly related, Intelligent Agent can use the linear regression method to represent that relationship. This means that linear regression is often appropriate. The parameters β_0 and β_1 are calculated from user historical data.

VI. CONCLUSION

In this research we were analyses PSP Practice Support System used to phenomenon of working data. We were able to viewpoint different searching user experience data. We create agent learner data in phenomenon data beside with user working analyses and wrote PSP Practice phenomenon error code type.

For future works, we will consider methods quick running of agent learner in data of error type phenomenon and user experience. We try to delete user missing work date filter on experience data. We consider to that delete missing work filter on error type phenomenon error type.

Future versions of PSP Practice Support System User need be conscious of rewrite error type code or fact process from error message of compiler and executor to test pattern data. But, This model will show at the system in a more natural, unscripted scenario, involving multiple parts in addition to other forms of process and error type phenomenon.

REFERENCES

- [1] Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, Judith Stafford (2010), Documenting Software Architectures SECONDD WDITION, Addison-Wesley
- [2] Watts S. Humphrey (2005), PSP - A Self Improvement Process for Software Engineers, Addison-Wesley
- [3] Daisuke Yamaguchi, Ayahiko Niimi and Muneo Takahashi (2010), Improvement of a Software Estimate Efficiency Centered PSP Practice Support System Using Multiagent Techniques, The Fifteenth International Symposium on Artificial Life and Robotics 2010, pp.857-860
- [4] Weiss, G. (1999), Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence, the MIT Press
- [5] Abul, O., Polat, F., and Alhadjj, R. (2000), Multi-agent Reinforcement Learning Using Function Approximation, IEEE Transaction on systems, man, and cybernetics-part c: application and reviews, Vol. 30, No. 4, pp. 485-497.
- [6] Khosla, R., and Dillon, T. (1997), Engineering Intelligent Hybrid Multi-Agent Systems, Kluwer Academic Publishers
- [7] Carles Sierra, JohnThangarajah, Lin Padgham and Michael Winikoff (2007), Designing Institutional Multi-Agent Systems, AOSE 2006, LNCS 4405, PP.84-103