Research Journal of Applied Sciences, Engineering and Technology 4(5): 518-520, 2012
ISSN: 2040-7467

Submitted: September 23, 2011 Accepted: November 16, 2011 Published: March 01, 2012

Quantitative Analysis of Programming Efficiency in Mixed Programming

Wen Lin-Tao, Zhi Feng-Hu and Chang Xiong-Zhou
Electronic and Information Engineering Deptartment Su Zhou Vocational University, Su Zhou, Jiang Su, China

Abstract: Prolog language is used for mixed programming in a example of Class Scheduling System based on multi-agent, in order to enhance the programming efficiency. The Language Suitability Model is introduced to quantify the programming efficiency of the programming. Finally, experiment is carried out for sample codes.

Key words: Efficiency of programming introduction, programming languages, prolog

INTRODUCTION

The main design languages of agent system: The language used to develop Agent is mostly Java, followed by C language. Java is an object-oriented programming language, and is very suitable for the development of Internet-related applications. It is easy to learn Java, especially for the programmers who are familiar with the similar C++ language, because its syntax is very similar to C++, but many low-level, difficult, confusing, prone-to-error or not frequently used functions in the C++ have been abandoned. In addition, Java has the following characteristics, making it become the best programming language for the development of Agent systems (Lange and Oshima, 1998; Intelli One Technologies, 2001; George and Luger, 1992) Platform Independence, Secure Execution, Dynamic Class Loading, Multithread Programming and Reflection.

Prolog suitable for design of agent: Prolog is the most famous logic programming language in the areas of artificial intelligence and expert system. Relying on inference mechanism, it solves problems through unity, replacement, digestion, recalling, matching and other mechanisms. Agent has some logical reasoning ability, which should be also one of its basic functions. Prolog language is easier to understand, with the stronger ability to express natural languages.

Principle and method of the mixed programming of java/C++ with prolog: A lot of mixed programming techniques have been applied in the development of modern software. Mixed programming techniques can well integrate the expertise of different programming languages, to complement each other. In the course of mixed programming of Prolog language with other languages, prolog program segments can be packaged into DLL, which is used to be called for other languages, (Ha *et al*., 2002) mainly used for mixed programming with C++, and also for the Java environment.

In the Java language, JNI (Java Native Interface) is built in Java Virtual Machine; JNI defines a standard naming and call standard, and allows users to use native codes, so that users can use the Amzi! Prolog Logic Server provided by Amzi! Prolog to achieve integration with Java (Zhang and Cui, 2004). In this study, Prolog language is used for mixed programming in a example of Class Scheduling System based on multi-agent, in order to enhance the programming efficiency. The Language Suitability Model is introduced to quantify the programming efficiency of the programming. Finally, experiment is carried out for sample codes.

EFFICIENCY ANALYSIS OF MIXED-LANGUAGE PROGRAMMING

Language suitability model of mixed-language programming: In this study, the concept of "Language suitability" is put forward a language closer to natural language, the more accurate to express the programmer's intents, the easier to write and understand, and the higher the language suitability, so the programming efficiency is surely high, which can shorten the development cycle and save development costs and maintenance costs. Language suitability and program complexity are closely related and have own different emphases, that is, the program complexity focuses on program maintainability and operational efficiency, and the language suitability

Corresponding Author: Wen Lin-Tao, Electronic and Information Engineering Department Su Zhou Vocational University, Su Zhou, Jiang Su, China

Table 1: Weight of words and characters

Word and character	Grammatical work	Constant and variable word	Basic arithmetic	operator language-related
Weight	1.5	1.0	1.0	1.2

focuses on the simple use of languages and it easy to write programs. It emphasizes the simplicity of operation, and provides the choice of host language and embedded language in the mixed programming with a rapidly quantitative reference standard.

- If a program segment is considered as a character set, the variable words, grammatical words, constant words and other ASCII characters are included; the less the quantity of characters needed to complete the same functions, the less the character types, indicating that the expression ability of the language is strong. The fewer the number of grammatical words, the closer the corresponding statements to a natural language, such as prolog program. But when variable words are named by the standard naming principle, it is rather difficult to understand constant words and variable words. Other ASCII characters can be classified into basic arithmetic symbols, such as: + - * / ( ) ; such symbols are common mathematical symbols, are independent of programming languages and are the closest to natural language, and their contribution to language suitability is high; language-related symbols: the other symbols in addition to basic arithmetic operation symbols, such as & and |, the specific meanings of these symbols have different definitions in different languages, and the more such symbols in use, the lower language suitability. The number of kinds of basic arithmetic operators and constant variable words does not influence the language suitability. Table 1 gives the weight of words and characters to show the arithmetic operators.

  Let n1 = The number of grammatical words
  n2 = The number of constant and variable words
  n3 = The number of basic arithmetic operators
  n4 = The number of language-related operators
  N1 = The number of grammatical kinds
  N2 = The number of language-related operator kinds

  Refer to Halstead method (Yun-fei *et al*., 2004), then:

$$S_1 = \frac{1}{(1.5 \times n_1 + n_2 + n_3 + 1.2 \times n_4) \times \log_2(N_1 + N_2)} \quad (1)$$

- Considering the logic meaning of a program, in the same size of program segments, the less the circular structure and the judgment branch structure, the easier to understand the program, and the higher language suitability. Let N = Circulating cycle, P = The number of branch judgments, C = total number of characters, then:

$$s = \frac{c}{N + p + 1} \quad (2)$$

  The denominatorplus 1 is to prevent N+P = 0 from emerging when testing program segments of a pure sequence structure, thus to lead to error when it is divided by zero.
- As the model focuses on the choice of different l by pro.0grammer, according to "90-10 test method" proposed by Shneiderman, choose a programmer with certain C language (common language)-based programmer to test for the intelligibility measurement method; if the programmer can properly write 90% of the programs written with a language, the value is 100 as the correction factor U. Let l = the number of code rows written well; L = the total number of code rows, then:

$$U = \frac{1000 * I}{9 * L} \quad (3)$$

- It can be drawn from the three aspects above:

$$S = Y * (a_1 * s_1 + a_2 * s_2) \quad (4)$$

  where, S means language suitability, α1 and α2 are the degree of emphasis of program characters and logic relationship suitability (weighted coefficient) evaluated by the programmer; they can be determined according to the programmer's experience, or derived from the survey, and α1+α2 = 1 is ensured.

EXAMPLE OF MODEL APPLICATION

In this section, a university's class scheduling system is used as the research example, its original business logic is regarded as the base model, the technologies of Agent are used to solve class scheduling problems, and the multi-Agent technology is used to facilitate the mechanism of communication and coordination among various participants in the class scheduling system, and to reach the goal of scheduling all the classes. This is a Prolog logic module ProLogicModule, which encapsulates a set of Horn clauses, and describes special requirements for class scheduling of a teacher:

Table 2: Test results

Parameter	Sample 1	Sample 2
n1	0	6
n2	29	23
n3	28	6
n4	14	40
N1	0	2
N2	3	4
N	0	0
P	0	3
C	275	261
U	100	100
α1	1	1
α2	1	1
S	13750	3263

Teacher Accept (lesson,10): Weekday (wednesday) noon (forenoon), previousclass (No_PE class), classroom (multimedia room).
TeacherAccept (lesson, 9):
weekday (Thursday), noon (forenoon), section (third).
TeacherAccept(lesson,8):
Noon (aternoon), nextclass (No_Mathclass),classroom (m ultimediaroom)
:
?- teacherAccept (#LessonPara#, AccPara)

In the module, a set of the simplest Horn clauses is used to clearly express subjective preferences of a teacher in terms of scheduling its own 3 classes.

A program segment of prolog in the above example is selected for simulation testing as sample 1, and java code equivalent to its function is used as sample 2:

```
If
    (weekday == "Wednesday" && previousclass == "No_PEclass" && Noon == "forenoon" && classroom == "multimediaroom") return 10;
If
    (weekday == "Thursday" && noon == "forenoon" && section == "third") return 9;
if
    (noon == "afternoon" && nextclass == "No_Mathclass" && classroom == "multimediaroom")  return 8;
```

where, α1 and α2 are 0.5, considered in a balanced manner, and meanwhile the sizes of the samples are very small and they can be completely re-written by the programmer, so U = 100. Table 2 gives the test results.

CONCLUSION

In this study, the mixed Prolog method is introduced into a Class Scheduling System in order to enhance the programming efficiency. Taking into account the Agent features, the Prolog language is used the Language Suitability Model. Experiment result has shown that the Language Suitability Model can reflect the difficulty level of programming with different languages, so as to provide a good quantified measure for the choice of programming languages in mixed programming.

REFERENCES

George, F. and C.S. Luger, 1992. A Translation of the History of China Machine Press. Artificial Intelligence-Structures and Strategies for Complex Problem Solving, pp: 499.

Ha, M., X. Wang and B. Niu, 2002. Multi-agent system based on VIC & VC. J. Hebei University (Natural Science Edition), 22(2): 167.

Intelli One Technologies, 2001. Agent Construction Tools. Retrieved from: http://www.agentbuilder.com/AgentTools/index.html.

Lange, D.B. and M. Oshima, 1998. Programming and Deploying Java Mobile Agents with Aglets. Addison Wesley Longman, Inc.

Yun-fei, Y.I.N., Z. Zhong and S.C. Zhang, 2004. Improvements about halstead model in software science. Comp. Appl., 24(10).

Zhang, B. and S. Cui, 2004. Realizes the java and the prolog complementary of the advantages using the JNI. Transport. Comp., 22(6): 120.