

The House Keeping System of Automated Evaluation of Students' Programming Reports

Hikofumi SUZUKI, Katsumi WASAKI*, Tatsuo NAKAZAWA and Yasunari SHIDAMA*

Nagano National College of Technology
716 Tokuma, Nagano 381-8550 Japan
{*crown, pierre*}@ei.nagano-nct.ac.jp

*Faculty of Engineering, Shinshu University
4-17-1 Wakasato, Nagano 380-8553 Japan
{*wasaki, shidama*}@cs.shinshu-u.ac.jp

1 Introduction

There are three stages involved in the process of grading students' reports: first, the student submits a report; second, the instructor grades the report; and third, the evaluation of the graded report is returned to the student. The second stage of the process is problematic, because the work of careful grading is very time-consuming. However, it is possible to reduce the labour in this stage by using techniques of semi-automatic grading.

There are various techniques for evaluating students' reports automatically, and this research has identified two main categories:

- **House Keeping:** This technique includes automatic checking of report deadlines, report length, and grading of multiple choice type questions. After this process has been completed, the instructor grades and evaluates the student's report.
- **Evaluation Support:** This technique provides the instructor with a description of the experiment and aspects of grading, such as theory verification. It also provides a checklist to use for grading, and a chart for the report.

The following section describes research based on the House Keeping techniques used for a report submission system in programming education. This technology is the pre-process for creating informations that the score of the report and the cautionary note to the instructor when the case of the final grading in the afterwards. Furthermore, we consider that this becomes the aid of improvement of reliability for report's inspection system.

2 Basis of House Keeping

The term 'House Keeping' refers to the checking of small items in the report grading process[1]. This research used three House Keeping modules, designed to provide instructors with information on grading.

- **Composition Verification Module:** The Composition Verification Module compares the report submitted by the student with a standard report

form. Marks allocated reflect deviation from the standard. Table 1 shows the subjects that are necessary for programming reports. It is understood that students must describe a report by the composition of eight subjects.

- **Quantity Confirmation Module:** The Quantity Confirmation Module allocates marks on the basis of the submitted report's adequacy in relation to the required word count or length.
- **Keyword Scanning Module:** The Keyword Scanning Module allocates marks on the basis of the number of relevant key words that should be contained in the report.

The instructor can use the House Keeping mark information obtained by these three modules to assist the grading of students' reports.

Table 1: The necessary subjects for a programming report.

1. Problem Analysis	5. Program Explanation
2. Logic Model	6. Program List
3. Algorithm	7. Execution Result
4. Program Design	8. Consideration

3 House Keeping for Programming Report Evaluation

The target of the House Keeping system varies with the programming experiments. Therefore, the instructor must confirm the compilation of the attached program and a result[2][3]. More than one module becomes necessary to realize this support. A flow diagram of the House Keeping process is shown in Figure 1.

The Program Extract Module extracts the part of the program list from the report that is registered in the RDBS (Relational DataBase System) first. The Program List with the XML form is converted into plain from the XML form. After the program syntax is checked by the Syntactic Analysis Module, a program list is compiled automatically by the Compile Module. An executive result is then compared with an executive result, which is

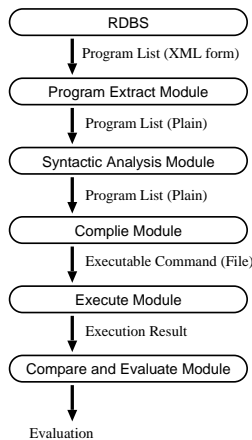


Figure 1: Flow diagram of the House Keeping process for the Program List.

attached to the student report and evaluated using the Compare and Evaluate Module. The evaluation processed at the end point is stored in the RDBS.

An outline of the House Keeping system in the Report Submit system is shown in Figure 2. To make House Keeping run smoothly, the RDBS is placed at the centre of the Report Submit system. The procedures for various evaluation outputs can be posted in the module units around the RDBS. In fact, it can separate such modules as the Composition Verification Module, the Quantity Confirmation Module, the Keyword Scanning Module, or other modules. Then, it is the same way that the executive confirmation of the program attached to the report as well.

4 Current Work and Future Research

The authors have constructed the system described above and prepared it for introduction to a beginners' class in C programming. In addition, the authors have analysed the programming reports of the 40-person class for a half-year period. Even if automatic submission is by e-mail, it is planned to build an exclusive registration system. The reports are already registered in an RDBS, and it should prove effective to arrange the modules so that a report can be evaluated in the surroundings of the RDBS.

To date, the Composition Verification Module, the Quantity Confirmation Module, the Syntactic Analysis Module, and the prototype of the Program Extract Module have been assembled, and experiments have been conducted using the proposed methods. Because the independence of each module is high, we can show that it is easy to evaluate the method, module by module. Moreover, the correspondence of plural every instructor, the subjects are also easy to construct.

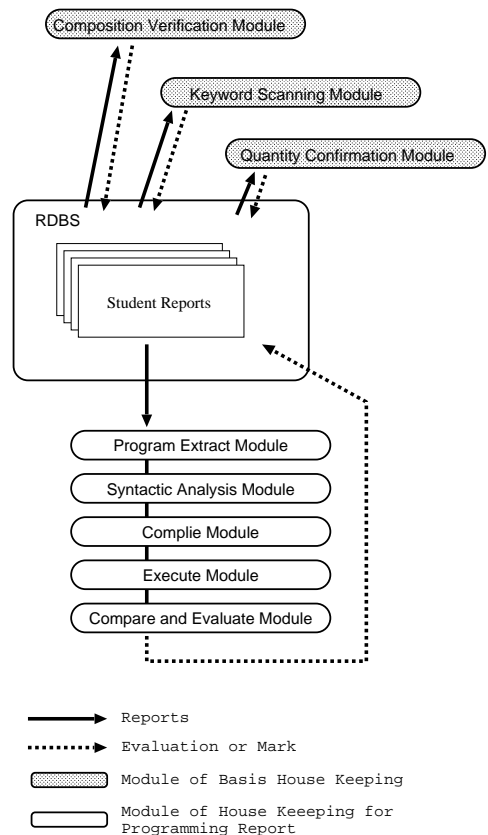


Figure 2: Outline of the House Keeping system in the Report Submit system.

By using multiple modules, we obtain the functions of the report inspection on the various sides. For this inspection being under all machinery process, this system can provide to reduce the work time of grading the report, and accurate outputs of report information. In addition, we expect to improve the reliability of this system by evaluating the correctness of the inspection modules.

Finally, further research is required to address the following issues: (1) how much support is necessary for an instructor; (2) what is the form of the figure used for the description of "Logic Model" and "Algorithm"; and (3) what method can be used to maintain contact with instructors supported by these modules.

References

- [1] Rika Sekimoto, Kenji Kaijiri, Masaya Yamagata: Development of the report preprocessing system using network for educational use, JSiSE, Vol.14, No.5, pp.217-222 (1998)
- [2] Tom Schorsch: CAP: An automated self-assessment tool to check Pascal programs for syntax, logic, and style errors, SIGCSE'95 (1995)
- [3] Charles Rich, Richard C.Waters: The Programmer's Apprentice, ACM Press (1990)