

N-5 プログラミングレポート自動評価における事前処理 The Pre-Process for an Automated Report Evaluation System for Programming Report

鈴木 彦文 1) 中澤 達夫 1)
Hikofumi SUZUKI Tatsuo NAKAZAWA

和崎 克己 2) 師玉 康成 2)
Katsumi WASAKI Yasunari SHIDAMA

1. はじめに

学生が提出するレポートの処理は大きく次に示す3つの段階がある。(1) 学生からのレポートを受け付ける (2) 教官が提出されたレポートを採点する (3) 評価したレポートを学生に返却する。この中で、(2) のレポートの採点は慎重に行わなければならないが、非常に時間と労力が必要となる。本研究ではこの手間のかかるレポートの採点について自動的に評価を行うことにより、教官の負担を大きく軽減できると考え、プログラミングレポートの採点について事前処理と自動評価の2点から研究を行っている。本稿では、プログラミングレポートの事前処理について述べる。

2. レポート処理の流れとターゲット

学生レポートの自動評価の方法については様々な手法が考案されているが、本研究では次の点に絞り研究開発を行っている[1]。

(1) レポート受付・事前処理

レポート提出時や教官が採点を行う前に行うチェックの自動化[2]。締め切りの管理、レポートとして記述を指定された項目の有無、選択式問題の採点など、教官がレポート内容の採点を行う前に、機械的に確認できる作業の自動化を行う。プログラミング課題のレポートにおいては、プログラムリストの構文的な正しさの確認と、実行結果の確認を含む。採点にあたり些細ではあるが必要な項目を事前にチェックするシステムを構築することにより、教官にかかる負担を軽減する。

(2) 採点支援

レポートの記述内容、グラフ、図、プログラムリストの表示(視覚化)など、教官が採点する際の支援を行う。

(3) 自動評価

プログラムにおける正当性の定量的検証を行う。ここでは、PADにおける主張[3]を用い、プログラムにおける操作的検証を行う。

本研究では、これらの処理・支援・自動評価をプログラミングレポートに絞り開発を行っている。ただし、初級教育用ということで比較的小規模なプログラムを想定している。本稿では(1)レポート受付・事前処理におけるプログラムリストの取り扱いと処理について述べる。

3. レポートの事前処理

教官がプログラミングレポートを採点する際、内容の採点の前に確認を必要とする項目がある[1][2]。本研究では

- 1) 長野工業高等専門学校 電子情報工学科
- 2) 信州大学工学部 情報工学科

図1に示すようなモジュールを用意することにより、この処理を実現している。これらの確認・検査モジュールの結果は、教官がレポートの採点を行う際に参考にされる。

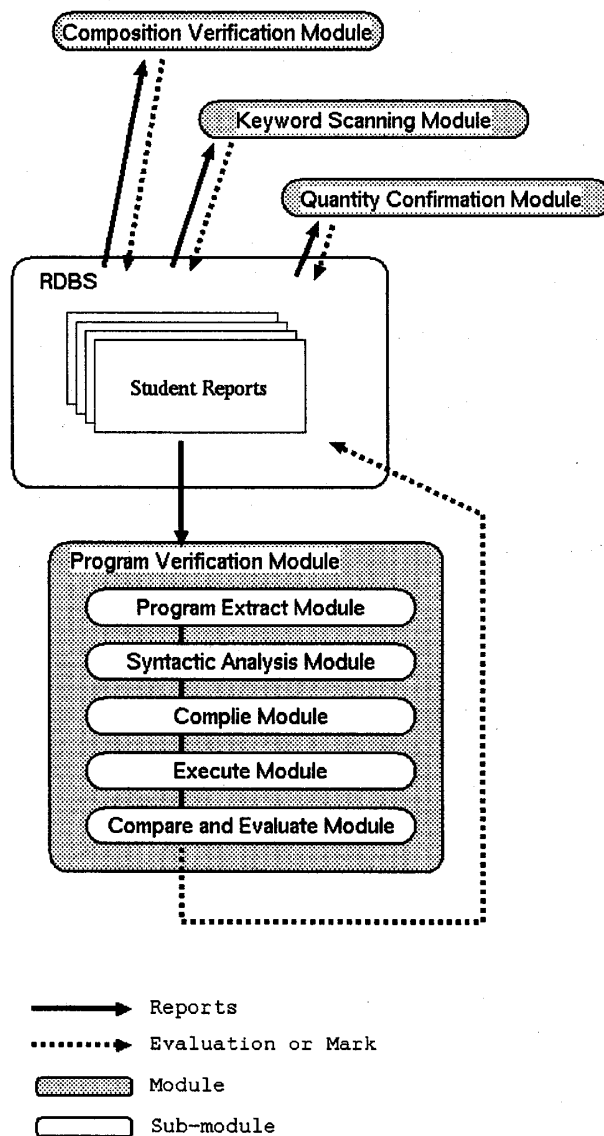


図1: レポート受付・事前処理システムにおける検査用モジュール

構造確認モジュール (Composition Verification Module)

学生が提出するレポートはXMLフォームで技術されている。例えば、表1では8つの項目についての記述を求めているが、この記述が構造的に問題ないか、また、必要な項目を満しているかを確認する。

表 1: プログラミングレポートを記述するために必要となる項目の例

1. 問題分析	5. プログラム説明
2. 論理モデル	6. プログラムリスト
3. アルゴリズムとデータ構造	7. 実行結果
4. プログラム設計	8. 評価・考察

分量確認モジュール (Quantity Confirmation Module)
 構造確認モジュールにて構造的な正当性が確認できた項目について、それぞれの記述量をカウントする。この際、例えば評価・考察など事前に設けた値以下の場合には教官に注意を促す結果を出力する。

キーワード走査モジュール (Keyword Scanning Module)
 レポート全体を走査しキーワードが利用されているかどうかを確認する。大きく 2 つのカテゴリに分類される。情報処理分野における広域キーワードと、特定の課題に対応した局所的なキーワードである。

プログラムリスト検査モジュール (Program Verification Module)
 プログラミングレポートには必ずプログラムリストと実行結果が添付されている。このプログラムリストについて動作確認をする必要がある。この作業はプログラムを切り出しコンパイルし実行するという機械的な作業によって実現出来る。そこで、プログラムリストについて (1) 構文検査 (2) 実行検査 (3) 添付された実行結果との比較を行う。

4. プログラムリストの事前検査
 プログラミングレポートの最も基本的な確認項目は (1) 提出されたプログラムリストの構文誤り検査 (2) 実行検査 (3) レポートに添付された実行結果との比較することである。図 2 はこの機能を実現する検査モジュールの処理の流れを示したものである。

まず、プログラムリスト抽出モジュール (Program Extract Module) が RDBS に格納されているレポートを取出しプログラムリストを抽出する。次に構文解析モジュール (Syntactic Analysis Module) にて抽出されたプログラムリストの構文 (文法) を確認し、コンパイルモジュール (Compile Module) にてコンパイルを行った後、実行結果比較モジュール (Compare and Evaluate Module) にて実行結果の確認を行う。

5. 検証
 事前検査モジュール検証のために、電子化されたレポートを 5 課題 81 通収集 (長野工業高等専門学校 電子情報工学科 3 年生) した。この結果、全ての課題においてプログラムリストの抽出と構文検査に成功した。しかしながら、実行検査に関しては 52 件で失敗。これはデータの入力形式を厳密に統一していなかったためであるので、課題の形式など指導方法の改善により解消する。また、実行結果の比較も同様に失敗するケースが多いが、実行結果の形式を統一することにより解消可能であるため、これも指導方法により解消する。

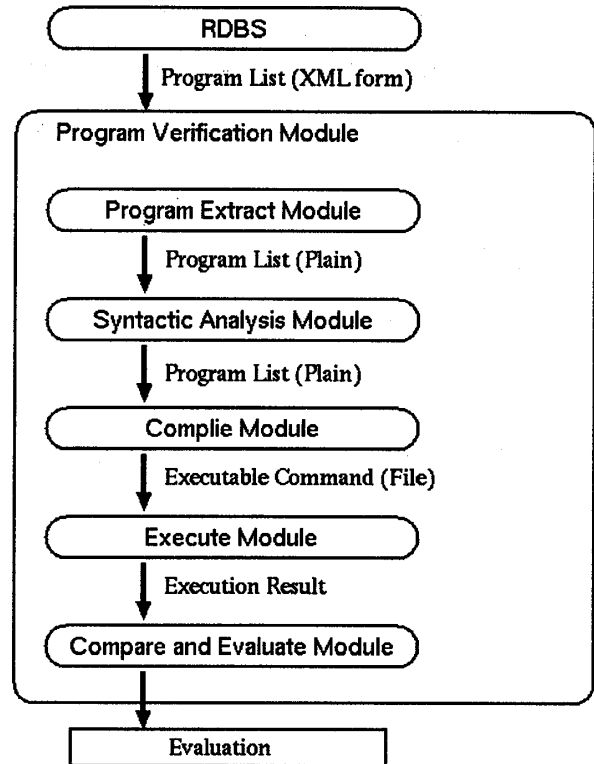


図 2: プログラム検査モジュールにおけるサブモジュールのトランザクション

6. まとめ
 以上のようにプログラミングレポートにおけるプログラムリストの検査について述べた。現在では対応していないが、コンパイル方法と実行方法 (入力データ) についてもフォーマットを定め、より柔軟な対応ができるようにする必要があるので、レポートそのものの記述形態を研究探る必要があるが、プロジェクト管理ファイル (Makefile) をコンパイル方法として指定する方法を採用し検証している。

検査を通過したプログラムのみに対し自動評価を適用させることにより、より正確な評価を行うことが可能となる。レポート受付・事前処理、採点支援、自動評価のシステムを組み合わせることにより、教官の採点にかかる労力が軽減できるだけでなく、採点の自動化により e-Learning や遠隔教育などにも応用が可能である。

参考文献

[1] Hikofumi SUZUKI, Katsumi WASAKI, Tatsuo NAKAZAWA and YasunariSHIDAMA : An Automated Report Evaluation System for Student's Programming Report by using House Keeping System ; International Symposium on Software Reliability Engineering (ISSRE2001), 19-20, 2001.

[2] 関本理佳, 海尻賢二, 山形昌也: ネットワークを利用したレポート受付・評価支援システムの実現, JSiSE, Vol. 14, No. 5, pp. 217-222 (1998)

[3] J.D. ウルマン: 情報処理シリーズ 1 プログラミングシステムの基礎, 培風館 (1981)