

佐賀大学でのソフトウェア開発実験における 単体テスト演習とその評価支援環境

白田 元^{*†} 次富 俊輔^{**†} 大月 美佳^{*‡}

*佐賀大学工学部 〒840-8502 佐賀県佐賀市本庄町 1

**佐賀大学大学院工学系研究科 〒840-8502 佐賀県佐賀市本庄町 1

E-mail: † {gengen, bsc}@cs.is.saga-u.ac.jp, ‡mika@is.saga-u.ac.jp

あらまし 佐賀大学工学部知能情報システム学科では、平成 15 年度より新カリキュラムおよびそれに基づく JABEE コースを制定し、最新の知識を持つソフトウェア技術者の育成を目指している。平成 17 年度から 3 年生を対象としたソフトウェア協同開発を体験学習するための実験が開講される。この実験を運用する計画を調整するため、先行して演習主体の講義を行ってきた。また、教官の定型作業を自動化し指導に注力するための支援システム CODELESS の設計開発を行ってきた。本論文では、特に CppUnit を利用した単体テスト演習と、この演習の評価支援について述べる。

キーワード 技術教育, 教育支援, 単体テスト, テスト自動化

Unit Test Workouts and Support of Its Evaluation in Development Experiment at Saga University

Gen SHIRATA[†] Shunsuke TSUGITOMI[†] and Mika OHTSUKI[‡]

† Faculty of Science and Engineering, Saga University 1 Honjo-Machi, Saga-shi, Saga, 840-8502 Japan

E-mail: † {gengen, bsc}@cs.is.saga-u.ac.jp, ‡mika@is.saga-u.ac.jp,

Abstract Department of Information Science, Saga University, established a new curriculum and the JABEE course based on it, to bring up software engineers having the latest knowledge. Since 2005, an experiment to learn a collaborative software development will be open for junior students. We are holding a lecture containing experiments to prepare for the experiment. Concurrently, we are designing and developing a support system CODELESS for a teacher to concentrate his/her attention on instruction by automating routine works. In this paper, we describe about unit test workouts and automatic test execution for evaluating them.

Keyword Technical Education, Education Support, Unit Test, Test Automation

1. はじめに

現在のソフトウェア開発は単独ではなく複数人での協同開発が主流となっている。協同開発のマネジメントの失敗は、システムの破綻を招く。納期の遅れ・プロジェクトの失敗・品質の低下について、旧来のプロセスモデルの問題点が指摘されており、eXtreme Programming (XP)[1]を含む俊敏(Agile)な開発手法が提案されてきている[2]。また、オープンソースを中心として、協同でのソフトウェア開発を支援するための優れたツールが提供されてきている。このような新しい技術に基づいた協同開発の体験、ツールに関する知識、マネジメントの必要性に対する認識などを早期に学ぶ必要がある。

佐賀大学工学部知能情報システム学科では平成 15 年度より新カリキュラムおよびそれに基づく JABEE コースを制定し、最新の知識を持つソフトウェ

ア技術者の育成を目指している。平成 17 年度から 3 年生を対象としたソフトウェア協同開発を体験学習するための実験が開講される。

この実験では、XP に基づき、小規模なチームで密接なコミュニケーションをとりながら、小規模なゴールを多数設定して、インクリメンタルな開発を行う。実装にあたっては、共有可能なバージョン管理システム CVS[3]、自動構築支援ツール Make[4]、単体テスト支援ツール CppUnit[5]などの使用方法についても学習する。

筆者らは、この実験を運用する計画を調整するため、先行して演習主体の講義を行い、同時に教官の定型作業を自動化し指導に注力するための支援システム CODELESS(COoperative DEvelopment LEarning Support System)の設計開発を行ってきた[6]。本稿では、特に先行講義における CppUnit による単体テスト演習と、

CODELESS システムのうち、CppUnit 演習での学生評価を支援するための「自動テスト結果表示ツール」について述べる。

2. 協同開発体験実験と先行講義

本節では、予定している協同開発体験実験の目的と構成、またこの実験を目指した準備として行ってきた先行講義についてその概略を述べる。

2.1. 協同開発体験実験の目的と構成

本実験の正式な科目名は「システム開発実験」であり、平成 17 年度前期から開講される 3 年生の必修科目である。本実験は、平成 16 年度後期に開講される座学型の講義である「オブジェクト指向開発」に対応する演習科目でもある。

このことから、本実験の第一の目的はオブジェクト指向(OO)技術の実体験である。OO 技術を実体験するにあたっては、色々なアプローチが考えられるが、限られたインターフェースで相互作用するオブジェクトからなるプログラムを独立に実装して組み上げることで、OO 技術の有効性を学習することを目指す。

また、現実社会の多くのソフトウェア開発では多人数からなる協同開発が一般的であるにも関わらず、従来の実験・演習では個別かつむしろ非協力的な開発作業が推奨されている。これは成績評価が個人単位であるためやむを得ない面もあるが、全く体験しないのは望ましくない。

このような理由から、本実験では OO 技術に基づいた協同開発体験演習を行う。ただし、企業で一般的に行われる協同開発をそのまま実現することは、時間的制約からも難しい。そこで、段階的に目的を定めて短期間にリリースを繰り返す開発プロセスである eXtreme Programming(XP)[1]を採用した。XP にはこの短期間のリリースの他にも、コードの共同所有やテスト駆動型開発など、近年重要視されている開発者の実践項目が多く含まれており、学ぶ価値は高いと考えられる。

構成としては、要求・分析・設計を行う「計画ゲーム」の実体験と、単体テストと実装を行う「イテレーション(反復)」を 2 回程度行う予定である。この過程で、CVS、Make、CppUnit などの開発ツールの使用法を学ぶ。

2.2. 先行講義

本実験の平成 17 年度の実施に向けて、平成 14 年度および平成 15 年度の選択科目「情報システム構築」にて、実験的な実施を行ってみた。以下でそれぞれの構成について説明する。

平成 14 年度

当初の計画では、最終的な構成を踏まえて、初回で実力判定・グループ分け。続く 2 回で計画ゲームを行い、残りを 2 回から 3 回のイテレーションとする予定であった。X 端末にログインして FreeBSD サーバ上で、GNU C++、GNU Make、CppUnit を使用して開発した。開発対象は、CGI プログラムとして動くゲームならば何でもよい、ということにした。GNU Make、CppUnit については簡単な演習を用意した。結果としては、設計段階で多くの時間をとられ、ツールが活用されないなどの課題を残した。

平成 15 年度

学科の計算機環境が大幅に変更され、X 端末がなくなり、学生個人の PC を使用するようになった。また、MSDN Academic Alliance を導入し、Windows OS 上で Visual C++ 6.0 を使用することになった。平成 15 年度講義では、前年度の経験を踏まえ、計画プロセスを削除(計画は教官側で用意)、14 回中 5 回で重点的にツール演習を行い、残り 9 回で 2 回のイテレーションと発表会の予定とした。しかしながら、実際には 9 回がツール演習に延び、4 回で 1 イテレーションのみに変更した。

このうち、CppUnit 演習について 4 節で述べる。

3. 支援システム CODELESS

本実験は全学生必修であり、かつ個々の学生の課題が多様になるため、教師・TA のみでは指導が困難になることが予測された。このため、自動化できる部分はなるべく自動化し、指導の負荷を軽減することを目的とした演習支援システム CODELESS(COoperative DEvelopment LEarning Support System)の設計・実装を進めてきた[6]。CODELESS は以下のようなツール群によって構成される。

- 「CVS 作業状況表示ツール」
CVS の利用状況を、教官および学生がネットワーク上で閲覧できるようにする
- 「学生の登録・変更ツール」
CVS への登録・変更などを、認証された学生が自分でネットワーク上から行えるようにする
- 「計画ゲーム作成支援ツール」
計画ゲームでの成果物を電子的に管理し、教官および学生がネットワーク上で閲覧できるようにする
- 「CppUnit による自動テスト結果表示ツール」
CVS コミット時に単体テストの実装状況を出力保存し、教官および学生がネットワーク上で閲覧できるようにする

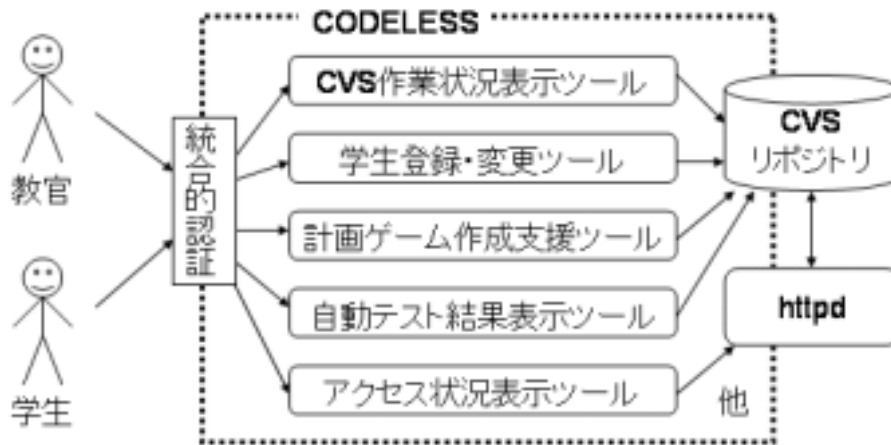


図1 CODELESS 全体構成

- 「資料ページへのアクセス状況表示ツール」
講義資料ページの利用状況を、教官および学生がネットワーク上で閲覧できるようにする
このうち「CppUnit による自動テスト結果表示ツール」について、5 節で述べる。

4. CppUnit 演習

本節では、先行講義中に行った CppUnit 演習について説明する。CppUnit は、単体テストの定型作業をライブラリ化したフレームワークである。Java 言語における単体テストフレームワークとして提案された JUnit を C++ に移植したもののひとつである。

CppUnit 演習は、インターフェースを先に書くテスト駆動型開発で行う。インターフェースを先に書くことで、「そのクラスがどういう機能を持つべきか」ということを強く意識するようになる、と期待される。

平成 14 年度は、カウンタを例題とした簡単な演習を 2 回行った。

平成 15 年度は、前年度の経験を踏まえ、具体的な成果物に近い例題を選び、3 回+ の演習を行った。その

後、実際の成果物の作成時に前もって作るべきテスト項目を具体的に指示して開発を進めている。

5. CppUnit による自動テスト結果表示ツール

このツールは、学生がリポジトリにコミットしたテストの実装状況を、Web ページで閲覧できるようにするためのツールである。教官は主に成績評価・指導のために、学生は自分の進捗状況を把握するために閲覧できるようにする。課題提供からテスト結果を Web で表示するまでの手順を図 2 に示す。

まず、教師が CVS のリポジトリに課題としてモジュールを登録 (import, インポート) する。学生はそこから自分のところへ課題のコピーを取得 (checkout, チェックアウト) して、そこにテストを実装する。課題終了後、学生はリポジトリへ登録 (commit, コミット) する。

この時点で、CVS の外部プログラム実行機能により本ツールが起動される。起動された本ツールは、コミットされたモジュールをチェックアウトしてきて、Makefile に記述されたビルド・テストを実行する。本

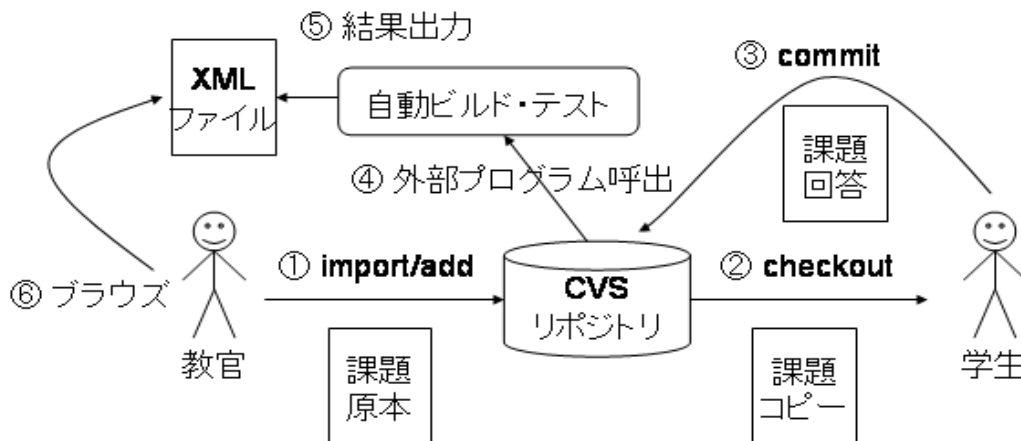


図2 CppUnit による自動テスト結果表示ツールの流れ

ツールでは、CppUnitの「テスト結果をXML形式ファイルとして出力する機能」により、Webサーバの領域に出力して終了する。このファイルを教官・学生はWebサーバを介して、ブラウザ上で結果を見ることができる。

6. 評価と課題

本節では、先行講義でのCppUnit演習について、および自動テスト結果表示ツールについて、評価と課題それぞれ述べる。

6.1. CppUnit 演習の評価と課題

CppUnit 演習の主目的は、クラステストを書くことでインターフェースを意識してもらうことであった。しかしながら、平成14年度は、実際の開発でどう用いるかを詳細に指示しなかったため、最終的な成果物には適用されていなかった。

以下のような問題点が学生の理解を妨げる原因となっていると思われる。

- 本年度までの先行講義では、学生が背景知識としてのオブジェクト指向概念を持っていない。
- ほとんどの学生が、他人のコードを読む習慣を持たない。サンプルコードがあってもその内容を理解しようとせず、単にその場しのぎにコピー&ペーストしている。
- テストの必要性を感じるほど大量のコードを書いたことがない。また、このためテストケースを考えることができない。そもそも何のためにテストを行うのか説明しても良く理解できていない。
- ツールの表示メッセージを読まない。特に英語のメッセージに拒絶反応を示す。CppUnitは結果表示が英語であるため、理解が困難であったようである。なお、他のツール(Visual C++, WinCvs ごった煮版)のメッセージ日本語であったが、その機構を理解してないためエラーメッセージに対処できない例も頻発していた。

オブジェクト指向概念の欠如から来る問題については、平成16年度から「オブジェクト指向開発」が開講されるため、軽減が期待できる。

習慣および経験不足から来るものについては、本実験中での演習を強化するとともに、FAQの充実を図る。また、平成15年度のやり方では最終ゴールが見えにくく、モチベーションの低下が見られたため、授業計画で計画の提示を前倒しするといった変更を検討する。

6.2. 現ツールの課題

現時点での自動テスト結果表示ツールは、設計段階にあり、今学期内での実装完了を目指している。平成

14年度にはUNIX OS上で開発を進めていたが、平成15年度には学生の環境がWindows OS上のVisual C++に変更された。一方、リポジトリは従来どおりUNIX OS上で運用されており、学生の開発環境とのずれが生じている。リポジトリをWindows環境へ移行するなど、実験環境の変更を検討すると共に、ツールの設計変更も検討する必要がある。

6.3. 全体的な課題

協同開発という形をとると、個人ベースの評価は難しくなる。基礎知識・技術の学生間のバラつきが大きく、学生間に積極的な協力が見られない。このような状況に対して、積極的な介入指導を行うためにCODELESSに進捗状況表示ツールの導入を行う予定である。進捗状況表示ツールについては、プロジェクト管理のための既存ツールを組み込むか、新規に開発するか、現在評価中である。

7. まとめと今後の予定

本稿では、平成17年度の協同開発体験実験のための先行講義におけるCppUnitを使用した単体テスト演習と、それと並行して開発を進めている自動テスト結果表示ツールについてその設計を示した。

今後は、演習内容の改善を続けていくとともに、ツールの実現と運用テストを行っていく予定である。

文 献

- [1] R. ジェフリーズ, 他, XPエクストリーム・プログラミング導入編, ピアソン・エデュケーション, 東京, 2001
- [2] A. コバーン, アジャイルソフトウェア開発, ピアソン・エデュケーション, 東京, 2002
- [3] 大月 美佳, 入門 CVS 第2版, 秀和システム, 東京, 2002.
- [4] R. M. ストールマン, 他, GNU Make, アスキー, 東京, 2001.
- [5] 大月 美佳, CppUnit による XP エクストリームプログラミング 実践テスト技法, 秀和システム, 東京, 2003.
- [6] 矢田 忠大, 長津 慶成, 龍野 哲平, 浦 光輝, 大月 美佳, "プログラム協同開発演習の支援環境CODELESSの構想," 第55回電気関係学会九州支部連合大会講演論文集, P477, 長崎, 日本, Sept. 2002