

平成 25 年度卒業研究論文概要 渡辺研究室
シクの表現が困難であった。また、UML シーケンス図では、JavaScript 言語には明確なクラスが無いことから構造を正確に表現することが困難であった。そこで、本研究は Node.js に最適な可視化記法を提案し、適用可能かを調査を実施した。

Rails テスト駆動開発環境の構築と調査

201012029 下田 悅介
Cucumber は Ruby on rails のテスト駆動開発ツールとして有名で、自然言語でテストを記述できるテスト言語である。テストを開発者と共に作業する時に便利で、記述したテストは専門知識のない人でも読める。テストの結果は、成功すれば緑色、失敗すれば赤色に表示されるため、実行結果は一見でわかる失敗した実行結果は、どのファイルの何行目でエラーが起きているか表示されるので、即座に改善することができます。

開発ヒストリの二手に別れ作業することと、開発側は作成中にエラーを気にせず集中できる。その後、開発したプログラムをテストし結果を修正すれば正常に動作するか気づき、すぐに開発に取り掛かることができる。これにより、開発側はプログラム開発のみを行うことができるので、一人作業するより効率よく作業出来ることが期待される。

本研究では、テスト駆動開発ツール Cucumber を実際に動かせるよう環境開発の構築を行い、Ruby,rails,RubyGems のインストールを行った。

Cucumber は、まずファイルにテストする振る舞いを記述することから始まる。その後、ステップファイルにプログラムを作成し、テストを実行して、完全なステップ定義を作り上げる。ファイルには日本語で内容を記述した。Cucumber を用いることで容易に把握できることを確認した。

User Mode Linux を用いた Web 要求の隔離処理に関する研究

201012016 春日 大地

本研究は、Web 上でユーザーからのリクエストを、隔離された安全な環境で処理することを目的としている。作成したプロジェクトは Ruby on Rails で作成し、ユーザーから C 言語のプログラムのコードを受け取り、その処理結果をユーザーに返すものである。Web 上でのコードにビルド・実行許可を与えるれば、すべてのセキュリティを放棄することにする。現時点での対策として、実行環境を Linux の Debian 10 とし、C 言語の処理の部分を User Mode Linux の遠隔操作に SSH コマンドで実行した。データの受け渡しには SCP コマンドと User Mode Linux の遠隔操作に SSH コマンドを用いた。

このプロジェクトの問題点は、1.ユーザーが同時にアクセスしたとき、ファイルが改変されることにより出力結果が違ったものになってしまう。2. 同時アクセスやビルド実行を行わせることにした。データの受け渡しには SCP コマンドを実行したときにコントローラが終了しないことである。

同時にアクセスによりファイルが意図せざり改変される恐れがある問題は、対策としてプロセス間の排他制御を行なうことにした。最初は Ruby の Thread クラスを使用しようとしたが、Ruby on Rails では一つのコントローラが終了するまでのリクエストを処理できなかった。その他にもセマフォを用いた排他制御や、デーモンプログラムに常にリソースの管理を行わせようとしたが、このプロジェクトの内容には合わなかつた。そこで Ruby の File モジュールにある flock を使用することにした。プログラムを記入するファイルを複数用意し、そのうちの一つを使用中のみロックする方法で、このプロジェクトのプロセス間の排他制御を行うことにした。ビルドと実行に最大 5 秒の時間がかかるため、先にページを表示し、実行結果は jQuery の ajax を用いて非同期に表示した。

無限ループに陥った時の処理には、Linux の timeout コマンドを使うことにする。この際、SSH 自体に timeout コマンドを用いると C 言語の実行ファイルの方に timeout コマンドを使用する。

Node.js アプリケーションの作成とソースコード可視化に関する研究

201012057 本田 真一郎

Node.js はサーバサイド JavaScript フレームワークの 1 つである。特徴はイベント駆動型でコールバック先を指定してイベントを発生させることでブラウザ、サーバ間の双方処理を可能にする。本研究では講義支援ツールを作成した。講義の監修教員のパソコンの画面を部屋にいる生徒全員が見るためには、スクリーンに映し出すなどしなければならない。しかしそれでは視力の悪い人に見えにくいくらい、教師の画面を生徒に配信する web アプリケーションを見る角度でスクリーンを作成しようとした。生徒のパソコンに教授のパソコンの画面のスクリーンショットを直接配信することで上記の問題が解決され、講義の効率化が期待される。web アプリケーションにはスクリーンショット配信機能のほか、画像送信機能やメッセージ送信機能、スクリーンショットの画像の上から、線を描画する機能を追加した。一方、開発過程で、JavaScript 言語ではプログラム構造を可視化するツールが少ないので、一部対応する機能を追加した。Doxxygen (ドキュメントジェネレータ) によるコード生成、Visual Studio 2012 によるデバッグ機能 (上位言語である TypeScript に対応) のみであった。このため、Node.js を知らない開発者にアプリケーションの構造を伝えることが困難であった。

研究では、フローチャートによる表現を試みたが、イベント発生やイベントに付随するデータの流れ、コードバ

Arduino を用いたワンキー入力装置の開発

201012060 水沢 弘

現在の携帯機器の入力はソフトキーボード、リック入力、音声入力が主体である。そこで、本研究ではモールス信号をベースとしたワンキー入力装置を開発する。ワンキー入力装置では、小型化や端末を見ない操作が期待できる。

研究は、Arduino を用了した学習機能の実装の 2 段階で行う。Arduino は AVR マイコンと USB ポートを備えた基板で、C++ 開発環境 Arduino、USB ケーブル、PC で構成している。現時点の試作機は、タクトスイッチの ON/OFF の時間とシリアル通信で PC に送信し、モールス信号の長短および文字区切りを検出している。その後、記録されているバーバンヒー式すなれば、PC に書き換えて出力する。手始めに、WPF のスライダーで長音短音の調節と文字の区切り時間を変更した。次に、スイッチ入力でも長音短音の調節と文字の区切り時間の変更した。その結果、ワンキーで登録されている文字の入力が可能になった。また、何かに試してもらい、入力ミスや、入力にかかる時間から改善点を調査した。

Haskell によるチャットボットの作成

201012013 大口 雄也

本研究は、純粹閑散型言語 Haskell で特定分野の質問に対して役に立つ返答をするチャットボットを作成することを目指す。今回は基礎となるチャットボットを作成したので報告する。

Haskell は、リサーチ・ソフトウェアが 1985 年に発表した純粹閑散型プログラミング言語である。特徴としては、遅延評価と強型付けがあげられる。

遅延評価とは、必要になつた時のみ関数を呼び出すものであり、無限の数列を参照し必要な部分のみ利用することができる。また、強型付けとは、ある処理・演算が限った型の引数をとることを禁止するというものである。

しかし、今回は、Haskell の強みである遅延評価を生かすことできず、また強型付けのために C 言語では簡単にかける参照も場所を変えるだけできなくなる仕様に苦しめられた。

今回はユーザーからの入力に対して返答するプログラムまで作成することができた。またチャットボットを利用して IRC サーバに投稿されたメッセージを表示するクライアントアプリケーションも Haskell で作成したので報告する。

機械学習による物体認識と AR マーカーによる検出精度の検討

201012070 吉川哲平

本研究の目的是、Web カメラを用いて様々な物体を認識することである。現在、顔認識技術に関しては完成度が高く広く普及している。しかし、任意の物体を学習させて検出する能力は、必要な学習時間の長さ、検出精度とともに不十分である。研究では、物体の形状と検出精度の関係、物体表面の塗装や撮影時の距離の違いと検出精度の関係を調査する。xml 形式の学習ファイル作成には、OpenCV2.2 に含まれる opencv_createsamples.exe,

opencv_haartraining.exe を用いた。また、検出には haarcascade 開数を用いた。opencv_createsamples.exe は、正解画像をコンピュータ上で仮想的に回転させて正解画像のパターンを増やすものである。現在、顔認識技術に関しては完成度が高く広く普及している。しかし、任意の物体を学習させて検出する能力は、必要な学習時間の長さ、検出精度とともに不十分である。研究では、物体の形状と検出精度の関係、物体表面の塗装や撮影時の距離の違いと検出精度の関係を調査する。xml 形式の学習ファイル作成には、OpenCV2.2 に含まれる opencv_createsamples.exe,opencv_haartraining.exe を用いた。また、検出には haarcascade 開数を用いた。opencv_createsamples.exe は、正解画像を別々のフォルダに格納し opencv_haartraining.exe を用いて学習する。研究の第 1 段階としてぬいぐるみを対象に、正解画像は 1 枚の画像 (53 × 138) から opencv_createsamples で作成した。なお、CPU: Core i5 - 2400, 3.10GHz, RAM: 4GB, OS: Windows 7 Professional 64 ビットの PC で xml ファイルを作成するのに 43 分を要した。また、さまざまなお体の検出精度を数値化することは難しい。これは、人の目による検出判断で處理に手間がかかること。また、撮影環境に大きく依存するため結果の再現性が確保できないためである。そこで、ARToolkit を使ってマーカーを検出し、マーカーをさまざまな物体に張り付け、マーカーと物体を一致させることで検出精度を測ることを検討した。