

本研究は昨年度の卒業研究で作成した時代別画像投稿サイトを引き継ぐものである。昨年度は、テストツールや可視化ツール(ER 図、コンテナー配置図、画面遷移図)といった開発支援ツールを用いていなかった。このため、大きなソースコード群を引き継いだものすぐに派生開発に取り掛かることができなかった。そこで、研究の第1段階として Rails アプリケーションの可視化ツールを調査した。その結果 3 つのツール(Rails ERD, Railroadry, RSpec + capybara + Roundabout)を有と判断した。これらは gem 形式の Ruby ライブラリで提供されており Linux Debian 7.1 に簡単にインストール可能であった(Railroadry)は Windows にはインストール不可)。これらのツールを用いて、昨年度開発した Web アプリケーションの ER 図、コンテナー配置図を自動生成した。ただし、画面遷移図に関しては、あらかじめテストシナリオを記述した spec ファイルが必要となるためテスト用の単純な Web アプリケーションで確認した。

これらの調査の結果、Rails アプリケーション開発では、期待される振舞を書き、支援ツールの図表で確認しながらコード開発を進める振舞駆動開発が有効であることが確認できた。これまでの成果は、(1)Debian 7.1 上での Rails 環境構築、(2)Rails ERD によるモデル可視化、(3)Railroadry によるコンテローラ可視化、(4)RSpec+Roundabout による画面遷移の可視化である。研究の第2段階では、時代別画像投稿サイトの完全な spec ファイルを作成し正確な図表付きドキュメントを作成する予定である。

SourceGraph を用いた Haskell コードの派生開発方法に関する研究

201012016 春日 大地

本研究は、Web 上に C 言語プログラムのビルド・実行環境を構築するプロジェクトの最重要部分である。Web 上で任意のコードにビルド・実行許可を与えれば、すべてのセキュリティを放棄することになる。幸いにも、Haskell で記述された"geordi - C++ eval bot"が GitHub 上でオープンソースとして公開されている。geordi は Linux OS の jail 機能と chroot 機能を用いて安全な C++言語のビルドと実行を可能にしている。しかし、geordi はインターネットリレーチャット(IRC)用に設計されており、改行の扱い及びそれに伴うエラー箇所の通知方法の解決が必要である。このため、geordi をベースに派生開発を行うが、パッケージサイズ 517kB のソースコードに目を通すのは困難である。Haskell コードの解析ツールに SourceGraph がある。SourceGraph はソースコードを読み込み、関数の呼び出し関係、構造、複雑さの指標を图示するものである。研究では、SourceGraph を用いることで、全体構造の把握、必要な変更箇所がピンポイントで見えるかについて調査をする。現時点の成果は、(1)C++言語ビルド・実行 Web サイト(セキュリティ対策なし)の構築、(2)irc サーバー構築、(3)geordi のビルドと評価、(4)SourceGraph の簡単な評価である。

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 はサーバーサイド 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、サーバー間の双方向処理を可能にする。本研究では、これまでに Node.js を用いて画像チャットサーバーを構築した。これにより、研究室内の会話で他者の作業を妨害することが防げると期待している。一方、開発の過程で、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ことが分かった。一部対応するのが、Doxgen(ドキュメンテーションジェネレータ)によるコールグラフ生成、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、コールバッ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、複雑なアプリケーションにも適用可能かを調査する。これまでの成果は、(1)express を用いた Node.js アプリケーションのサーバーコードのひな形作成、(2)WebSocket の導入、(3)クライアントコードの作成、(4)画像アップロード機能の実装、(5)JavaScript コード可視化ツールの調査、(6)Node.js コード可視化記法の提案である。

機械学習によるオブジェクト検出

201012070 吉川 哲平

本研究の目的は、Web カメラを用いてさまざま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、空き缶など物体表面の塗装や撮影時の角度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 関数を用いた。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。多数の正解画像と非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段階として空き缶を対象に、正解画像 4000 枚、非正解画像 1494 枚で学習させた。ここで、4000 枚の正

解画像は 1 枚の画像(17×63 ピクセル)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。今回は、検出精度の数値化ができなかった。これは、人の目による検出判断で処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。本研究では、対象物に赤外線フィルムを貼り、赤外線画像とカラー画像から検出精度を自動で計算する方法も検討中である。

ディジタル変復調実験のためのソフトウェア受信機の開発

201012038 田中 駿

本研究では安価かつ手軽なディジタル変復調の実験機材を開発する。PSK、QAM 等の変調機能を用意した信号発生器は各社から販売されている。しかし、受信機は信号のフォーマットごとに異なることから、機器選択が限られて高額である。このような状況の中、近年 RTL2832U チップを用いたソフトウェア受信機が注目を集めている。RTL2832U はワンセグチューナー用チップであるが IQ 出力モードがあり、これを PC で処理することでソフトウェア受信機が構築できる。受信用ソフトウェアにはオープンソースの SDR#があり、AM、FM 放送、各種アナログ無線を受信することができる。そこで、我々は SDR#にディジタル復調機能(PSK、QAM)を追加することを計画している。研究の第1段階として Visual Studio 2012 Ultimate を用いて SDR#のソースコード解析を行い、AM 及び FM 変調部分のコードを確認した。現時点での成果は、(1)USB ワンセグチューナーのソフトウェア受信機化、(2)SDR#のソースコードビルド、(3)主要部分のシーケンス図生成、コードマップ作成、(4)ソースコード上での AM 及び FM 復調部分の確認である。

Haskell による人工無脳機の実装

201012013 大口 雄也

本研究では、関数型言語である Haskell を用いて人工無脳機の実装をめざす。Haskell とは、リサーチ・ソフトウェア社によって 1985 年に発表された純粋型プログラミング言語であり、特徴として、遅延評価を採用している。関数の引数や式が必要になるまで評価されない、型推論によって関数の型を明確に指示しなくても自動的に型が推論される等があげられる。また他の言語で利用される配列と同じ役割を持つ list があり、Haskell では list 内のすべての値に適用するアクションや一部だけに適用、抜き出すアクションがあるので今回はそれを利用していきたい。

人工無脳機とは人間の入力したキーワードを拾って、内部のデータベースとのマッチングによって知的に入力を解釈して応答しているように見えるプログラムである。

人工無脳機作成の上でもっとも大事な点としてユーザへの入力に対して返答できることがあげられる。Haskell ではユーザ-インターフェースライブラリ readline を導入することによりユーザへの入力と返答をすることが簡単に実行できるようになった。現段階では、適当な言葉に対してデータ中の語句をランダムに返す。ある特定の言葉により感情値が増減し、その数値によりデータ内から返答するプログラムを作成することに成功した。今後の研究内容としては(1)入力に対して対応できる言葉を増やす(2)データベースの内容を増やし返答できる語句を多くする(3)可能であれば自己学習できるようにする。これらのことを実行し、最終的にはより人間の入力に対してプログラムではなく人間が返答しているかのように錯覚するプログラムを目指していく方針である。

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ウェアキーボード、フリック入力、音声入力などが主体である。しかし、ソフトウェアキーボードは機器の小型化を制約し、また、音声入力はインターネット接続の制約がある。そこで、本研究ではモータースイッチの On、Off をシリアル通信で PC に送信し、スイッチ切り替え時間を固定値で判別し、モータースイッチの長短および文字区切りを検出している。

現在までの成果は、(1)タクトスイッチ基板の製作、(2)Arduino シリアル通信の実装、(3)C# WPF アプリケーションの作成(シリアル通信、描画、スイッチ切り替え時間判別)である。

開発では、手書きによる DFD の他、Visual Studio 2010 Ultimate のシーケンス図作成機能、クラス図機能が大いに役立ったので合わせて報告する。