

実用アプリケーション作成における F# と他言語の比較

200812010 小柳 勇貴

本研究は、マイクロソフトリサーチが開発している F# 言語の他言語に対する優位性を明らかにするものである。F# と他のプログラミング言語を比較するために、Office アプリケーションの操作、データベースの操作、Web アプリケーションの動作を行うプログラムを作成した。

Office アプリケーションの操作を行うプログラムでは、Excel の起動、値のセット、グラフの作成を行った。比較対象として、VBS を用いた。F# は VBS と比べて、関数の宣言と使用を同時に行うことが出来るため行数が抑えられる。配列データの準備が容易といった利点があることが分かった。

データベースの操作を行うプログラムでは、O/R マッピングを行い、MySQL のデータベースに対して基本的な操作を行った。比較対象として C# を用いた。F# は C# と比べて、型推論による型付けのためにクラス定義が不要になり、行数が抑えられる利点があることが分かった。

Web アプリケーションの作成では、Visual Studio 2010 で ASP.NET プログラミングを行った。Web ページのデザインは F# と C# で共通のものを使い、Web アプリケーションの動作部分を作成し比較した。動作部分のプログラムは、MySQL 内のテーブルを利用して、簡単な欠取を取るプログラムを作成した。その結果、F# は C# と比べて Web アプリケーションを作成するためのサポートが少なく、現時点では C# の方が Web アプリケーションを作成しやすいことが分かった。

オンライン地図サービスを用いたバス情報のデータベース化

200812022 小室 翔平

バスの路線情報は、普段バスを利用しない人にとっては分かりづらい点がある。日付や時間帯によっては運行しない時間帯が存在し、普段から利用しない人にとっては、バスは利用しづらいと考えられる。

また、バスを利用する為の情報が少ないということも考えられる。地元の路線バス会社のホームページには、すべてのバス停の停車時刻を表示した時刻表がない。バス会社によっては、ホームページの時刻表ですべてのバス停の停車時刻を知ることができない場合もある。バス路線情報、時刻を確認できる Web サービスが少くないということがあげられる。移動経路を調べるサービスの中には、移動手段が徒歩、車、電車しか選べない。

以上のように、複雑な運行やバス停車時刻情報の不足により、バスを利用したくても利用できないということがあると考えられる。

昨年度はわかりやすいバス情報を提供する為のアプリケーションを開発した。本年度では、アプリケーションをさらに便利に利用するための機能を作成した。バス路線表示画面では、運行している路線を日付指定して表示する機能を追加した。日付の指定は画面上に表示されたカレンダーを用いて行う。カレンダーの日付をクリックすることで、その日に運行するバス路線を表示する。最初に画面を表示したときは、当日の運行路線を表示する。運行する路線のみを表示することで、路線の確認をしやすくなる。

バス路線の登録についても機能を追加した。運行日の情報を扱うため、データベースに新しくテーブルを作成した。路線作成の際に、登録されているポイントを利用する機能を作成した。既存のポイントを利用することで、路線作成にかかる時間を削減することができる。

自立走行型全方位移動台車の開発

200812050 古川 翔

本研究の目的は、自立走行全方位移動台車の開発である。本研究は、機械学科の中嶋研究室との共同研究である。本研究において、自立走行型全方位移動台車の制御プログラムを担当した。

本研究の開発対象となる台車は、遠隔地に設置した計算機より制御する。計算機からの指令は、無線を介して送信する。台車の特徴として、メカナムホイールを採用しており、全方位移動が可能となる。また、動作は通常と低速動作が用意されている。開発言語は VC++ を用いた。

台車の自立制御には、デッドレコニング法を採用している。デッドレコニング法とは、ルートや行動時間などを、事前に理論を元に設定する方式である。この方式を用いることにより、自立走行を実現している。しかし、事前に動作を設定するため、スリップなどによる誤差の影響は考慮できない。そのため、今回は走行経路上にチェックポイントを設け、その都度誤差修正を行う方法を採用した。

チェックポイントでは、天井に設置したカメラにて、台車の位置を認識する。台車の位置を認識するために、台車に 2 つの LED を搭載している。LED はカメラ画像を二値化し、輝度値を元に検出する。検出された LED 位置より、台車の座標及び、台車の傾き角度を求めると、算出された数値を元に、角度修正及び、位置修正を行う。修正時の動作は、低速モードにて行う。

論理式表現と妥当性評価に関する研究

201112049 藤沢 駿

コンピュータはプログラムで指定された手順通りに実行される。このため、プログラムは手順をそれが一意的であるか否かにかかわらず、常に意識することを強要される。現実の事象や問題を処理するアプリケーション作成では、手順と同等またはそれ以上にシステムの論理的妥当性について熟考する必要がある。プログラミングにおいて最初に論理的妥当性が検証できていれば、手順設計で不要な迷いを避けることができる。そこで、本研究では自然言語で書かれた課題の論理表現と妥当性の確認方法の検討を行った。発表では分析例として〇×ゲームを紹介する。ゲームは先手・後手の順で交互にコマを置き、先に 1 列並んだ方が勝ちというものである。妥当性確認には線形時相論理 (LTL) を用いる。LTL では、先手が 1 列に並ぶことを P、後手が 1 列に並ぶことを Q とおくと、先手が常に勝つという条件は $\Box(\neg Q \cup P)$ で表される。論理が妥当でないとき、すべての場合に条件が成り立つことを検証するよりも、その否定形が計算できるためである。ここでは、先手に戦略を与えることで必勝法の有無を明らかにした。

論文では、他にも算数の論理表現についても検討した。例えば、単純な鶴亀算では、各条件を述語論理で考える。述語は実モデルでは集合に対応するので、答えはすべての述語の積集合で求められることが分かった。今後の課題は、位置または時間によって状態が変化するような複雑な問題について、可能世界間の遷移と各世界で成り立つ述語でモデル化することである。