

ロボットアームプログラミングにおける XML の活用

200312094 渡辺 雅介

昨年度の XML スキーマファイルを読み込んでのロボットアーム制御は、クライアント側で XML ファイル作成し、それを SOAP メッセージとしてサーバーに送ったのち、直接数値を与え、ロボットアームを起動させていた。ロボットアームに数値を与え、起動させるというものであった。しかし、その値が読み込まれ実際にロボットが動いたときに、どのくらいの範囲で動かせるのか、またどれほど動くのかがあやふやであったため、使用者には扱いづらい部分があった。今回私はそのロボットアームの操作をより簡単に実行できるプログラム作成の研究に取り組み、様々なプログラムの考察と実験、利用価値について研究を進めてきた。その中で XSL, XSLT, XPathNavigator と呼ばれる 3 つのプログラムについて着手した。それぞれのプログラムの特性として実行時または結果を表示した時の画面の見栄えや、読み取る値の順序や変換を自分なりに設定することが可能になることが分かり、「人間に扱いやすいロボットアーム」作成に大いに役立つのではないかと考えた。本研究では、XSL, XSLT, XPathNavigator の 3 つのプログラム特徴と、それらを実行した際の、画面の見易さや、操作性がどう変化するかを実行結果で示すとともに、ロボットアームを扱う上で、どのような利点として生かせるかをここに述べたものである。

防犯ブザーの高機能化に関する研究

200312078 牧口 誠也

近年、人々の防犯意識の高まりによりさまざまな防犯機具が開発されている。例えばスイッチやひもを引っ張って音を出すタイプの防犯ブザーや、子供向け携帯電話に用いられる GPS (Global Positioning System) の位置情報伝達機能を使った防犯器具である。前年度卒業生が製作していた赤外線を利用した防犯ブザーを基に、今度は無線を利用した防犯ブザーを作製しようと考えた。

無線を用いることによってできるだけ動作を必要とせず、かつ正確にブザーを鳴らせることはできないかと考えた。送信側から常に無線通信を行い、受信側にその信号が届かなくなった場合に音を鳴らせるというしくみである。現実的な例で言えば鞆の中などにブザーをいれておき、もう片方を手元に持っておけばひったくりにあったとき無線の通信範囲をこえた瞬間にブザーが鳴る。

そこでまず、ハードウェアに着目した無線送受信機の開発をした。

まずコンピュータ上でシミュレーションを行い、どのような動作をするのかあらかじめ検証した。その結果を反映し実際に無線送受信機の作成に取り掛かった。送信機はコイルの調整がうまくいかず周波数が安定しなかった。受信機は受け取った様子をオシロスコープの波形で観測すると波形に雑音の影響とみられるひずみがみられた。それぞれの回路の問題点の改善のためコイルの調整方法や雑音の解消のための原因の究明を行なった。本研究は無線送信機受信機のハードウェア面について述べたものである。

ゲノム解析における C++言語の Perl 言語に対する優位性の評価

200312086 村山 豊

ヒトゲノムの塩基配列の解読を目的としたヒトゲノム計画は 1991 年に始まり、2003 年にシーケンスの読み取り作業が終了した。ヒトゲノム計画に続いて現在様々な生物のシーケンシングが行われている。ヒトゲノムは 30 億個の塩基対からなっており、そのシーケンスデータを用いた遺伝子機能の解明が今後の課題である。

ゲノム解析とは塩基配列やアミノ酸の解析を行うことを指し、バイオインフォマティクスの一部として扱われる。ゲノム解析のプログラミングの手法としては解析を行うのに重要な文字列処理に適していて、プログラムの開発も容易といわれている Perl 言語や Perl の代替である Ruby が代表的である。広く使われている C 言語には文字列の型が存在しないので、簡単な操作でも配列や複雑な関数を使う必要があった。C 言語に標準ライブラリとして文字列を扱える型 `string` コンテナを追加した C++言語では文字列の処理の操作性が向上した。さらに C++と Boost という正規表現を含むライブラリを組み合わせると Perl などと同じ文字列処理が可能となる開発環境を構築できる。

本研究では、普及率と処理速度の面で有利といわれている C 言語とゲノム解析で行われる文字列処理に適しているといわれている Perl 言語による比較を目的とするものである。

比較する例題として塩基配列から塩基数を数える単純文字列処理、制限酵素による電気泳動パターンシミュレーションのプログラムの構築をおこなった。

以上の例題より、二つの言語によるプログラムの記述の違いによる作成の容易さ、プログラムの処理速度を比較、検討していく。