C言語によるデータバインディングの分散環境への適用実験

学生証番号 044273　　コンピュータサイエンス学科　　亀田 祐人（荻原研究室）

1 はじめに

オブジェクト指向ソフトウェア開発では、オブジェクトのプロパティ同士を動的に関連づけてソフトウェアを構築する、データバインディングと呼ばれる方法が使われている。

当研究室では、データバインディングの機能をC言語でも利用可能とするライブラリとしてcoval[1]を開発している。covalを利用することで、ソースコードの変更を少なく抑えつつ、モジュールの構成を変更できる。

従来のcovalは同一のプロセス内で値を共有し、モジュールを結びつける働きを持っていた（図1）。本研究では、異なるプロセスのcovalを相互に結合できるような機能の拡張を扱う。この拡張の利点を確かめ、実用上の問題を考察することが本研究の目的である。

2 remote covalによる値の共有

新たに提案するバインド機構をremote covalと呼ぶ。remote covalでは共有値を識別子（coval名）で管理するためのサーバを用意する。同じcoval名で登録されたcoval同士は、通信によって値を共有し、値が変更された時にはあらかじめ決めておいた手続きを起動できる（図2）。各プロセスで用いるcovalは従来のcovalと同じものである。

3 実験プログラム

タンクの水位調節をシミュレートするプログラムを作成した。流入バルブと給水バルブの状態をcovalで表し、水位によって流入バルブが開閉される。このような２つのタンクについて、図3のように流入バルブと給水バルブをremote covalで互いに接続する。２つのタンクのプロセスが、remote covalを介して協調動作することが確認できた。

4 評価とまとめ

remote covalによって、異なるプロセスが協調動作できることを示した。また、ソースプログラムの追加、変更が抑えられることも確認した。しかし、あらかじめcovalを用いてプログラムを記述しておく必要があり、この点が実用に向けての大きな問題と言える。

今後は具体的な事例での有用性を示すことが必要と考えられる。

参考文献

[1] 荻原剛志：手続き型言語におけるデータバインディング機構の提案と構造化設計への適用，情報処理学会論文誌，54巻，4号，pp. 1573-1580 (2013).

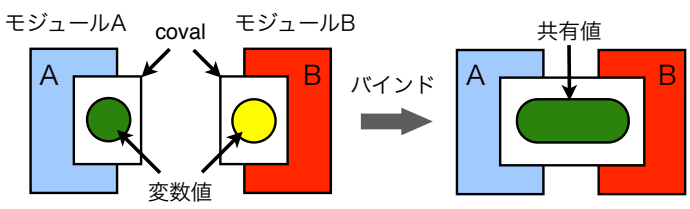

図1 covalとバインドの概念

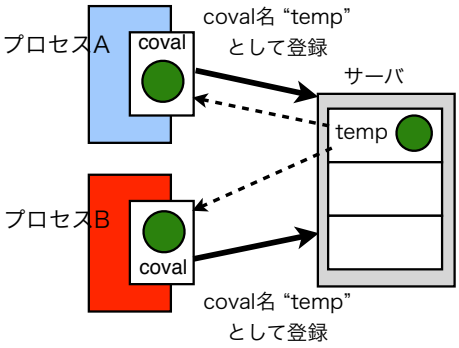

図2 remote covalの概念

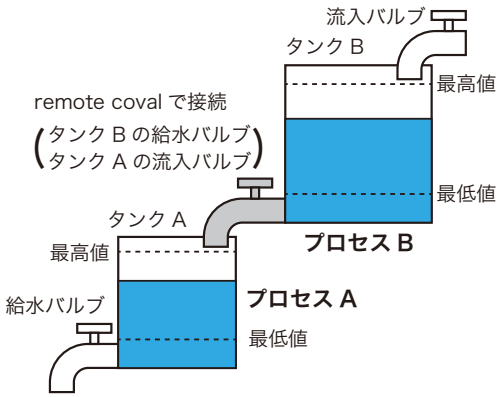

図3 タンクの水位調節

構文情報を利用したデバッグツールの作成

学生証番号 044525　ネットワークメディア学科　佐藤　雄太　(荻原研究室)

1. はじめに

XcodeやEclipseなどの統合開発環境が数多く出回っている。しかし、このようなツールは素人が一朝一夕に使いこなせるものではなく、使いこなせるまでには時間や学習コストが掛かる。そして、そういったツールでは対応出来ない状況もあるだろう。

デバッガを作成するには構文解析をする必要があるが、これには大変な手間がかかる。しかし、C言語の解析結果をXML形式で出力するXCI[1]というソフトウェアがある。本研究ではこれを用いて、独自のデバッグツールの作成を行い、有用性を示す。

2. XMLによるC言語の表現

XCIはANSI Cプログラムのインタプリタである。C言語の構文情報をXML形式で出力することができ、XMLファイルからCプログラムに復元することもできる。

XCIは、Cプログラムの抽象構文木の情報やそれに付随する型・変数値および参照・定義関数などの静的情報をXMLに出力する。このXMLを利用することで構文解析の手間を省くことができ、独自のデバッグツールを作成することができる。

3. printf関数挿入ツール

XCIの出力ファイルと本ツールの関係を図1に示す。本ツールはXMLファイルを書き換えて、別名のXMLファイルを作成するので、元のCプログラムを復元することができる。

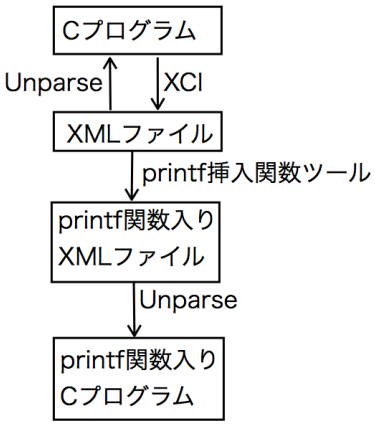

図1 ファイルとコマンドの流れ

このツールは指定した変数が変更されている箇所の次の行に、その変数値を出力するprintf関数を自動的に挿入する。指定された変数が式の左辺にあるときのみを検出し、printf関数を挿入していく。

```
1  : int main(void )
2  : {
3  :    int val, x, i;
4  :
5  :    val = 10;
6  :    printf("file : %s , line : %2d , val = %d\n",
                                __FILE__, __LINE__, val);
7  :
8  :    for (i = 0; i <= 5; i++ ) {
9  :       val++ ;
10 :       printf("file : %s , line : %2d , val = %d\n",
                                __FILE__, __LINE__, val);
11 :    }
12 :
13 :    x = val + 20;
14 :
15 :    return 0;
16 : }
```

図2 printf関数挿入ツールの実行結果

実行結果はUnparseを使用することにより、Cプログラムに復元することができ、「printfデバッグ」を行うことができる。(図2)

4. 検証

本ツールはコマンド1行で実行可能なため、使用時における学習コストや時間はかからない。

構文情報から変数の変更を特定し、printf関数を挿入しているので人が変数を目で追って行うより正確で素早い。手作業でprintfデバッグを行っていては、printf関数の書き間違えでバグの温床になりかねない。また、printf関数が多くなるとプログラムの可読性が低下する。しかしこのツールでは元のCプログラムに復元することが可能である。そして、プログラマが必要な機能を拡張していくこともできる。

本稿では省略したが、大域変数の変更箇所に警告を出していくツールも作成した。

5. おわりに

構文情報を利用することによって、デバッグツールを比較的簡単に作成できることが分かった。しかし、XMLの書き換えにJavaやDOMの知識が必要になる。そのためにそれらの知識が無くとも、GUIで直感的にデバッガを作成するためのツールがあれば良いと考えられる。またXCI自体にも潜在的なバグがあり、これも改善していく必要がある。

参考文献

[1] 川島 勇人, XCIで表現されたCプログラムの静的解析ツールの設計と表現, (https://dspace.jaist.ac.jp/dspace/bitstream/10119/1532/2/1568paper.pdf)

Bluetooth 通信を利用したカードゲームアプリの開発

学生証番号 044958　　コンピュータサイエンス学科　　呼　思楽 （荻原研究室）

1 はじめに

本研究では、昨年の卒業研究[1,2]を参考にして 、Bluetooth 通信を利用したゲームアプリケーションを開発した。人を選ばず誰にでも楽しんでもらうために、ゲームはルールが分かりやすく、なじみ深いトランプゲームの「ババ抜き」を題材として選んだ。インターネット接続が使えない状況であっても、Bluetooth 通信搭載のデバイスを利用すれば複数人で一緒にゲームをプレイできる。さらに、使う言語が違う人がいてもコミュニケーションをとれるような定型文のチャット機能を提案し、アプリケーションの一部として実現した。

2 アプリの流れ

iOS の GameKit Framework の中の GKSession クラスの機能[3]を使い、複数台の通信を実装する。

最初に、参加者のうちの 1 台がサーバ役となって接続作業を行う。カードを配布してペアのカードを取り除いてからゲームを開始する。カードのやり取りはすべてサーバ経由で行う。

ゲーム中は、全員の手札の枚数が分かる表示の画面と、自分と直接やりとりする相手の 2 人のカードだけが見える画面を遷移する。

3 考察

ゲーム自体は実物のトランプゲームと比べ、カードを配る手間や、複雑な操作がない。ルールの分かりやすいトランプゲームを題材とし、かつ多言語対応の選択式のチャットを搭載しているので、より多くの人に楽しんでもらえると思う。さらに、CPU 参戦機能や CPU によるチャット応答も備えているので、ゲームの面白みも増し、一人でも楽しく遊べる。

4 課題

ゲームの見栄えを良くするために、UI の改善が必要である。チャット機能のパターンも増やすべきであろう。また、ゲームの面白さや使いやすさについては、実際に何人もの人に利用してもらって具体的な評価をしてもらう必要がある。

図 1　ゲームプレイ画面

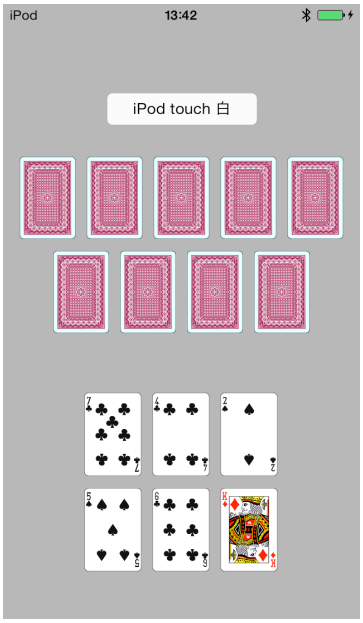

図 2　カードを引くときの画面

5 参考文献

[1] 今田達也, 機能のカスタマイズが可能な相互通信型モバイルアプリケーションの開発, 京都産業大学コンピュータ理工学部 卒業論文, 2013.

[2] 湯川隆介, 子どもと遊ぶ事を目的とした双六アプリケーションの開発, 京都産業大学コンピュータ理工学部 卒業論文, 2013.

[3]ぎぎねっと, GameKit を用いて複数の iOS デバイス間で P2P 通信, http://giginet.hateblo.jp/entry/2013/01/23/053050 (2013)