



生活の中でコンピュータの必要性が高まるにつれて、コンピュータを動かすためのソフトウェアもたくさん必要になってきます。ソフトウェアを作るとき、モジュール(部品)とモジュールとをくっつける役目を担うのが「グルーコード」と呼ばれるプログラム。荻原剛志先生は、グルーコードを簡略化したり、不要にしたりすることで、ソフトウェア開発の効率をもっと高められると考えて、研究に取り組んでいます。その具体的な方法についてお話を伺いました。

コンピュータサイエンス学科
荻原 剛志 教授



C言語のモジュールを簡単に つなげる新しい概念「coval」

モジュール同士を くっつけるプログラム

炊飯器や冷蔵庫といった一般家電にもコンピュータが組み込まれるようになり、私たちはたくさんのコンピュータに囲まれて生活しています。コンピュータが増えれば、ソフトウェアも大量に必要となりますが、それらを生み出すのは、システムエンジニアやプログラマーなど「人の手」です。人の手による作業ですから、当然ミスも起こり得ます。

現在のソフトウェアの開発現場では、すでに出来上がった「モジュール」と呼ばれるソフトウェアの部品を組み合わせることで、新しいソフトウェアを作ることが少なくありません。モジュール単体ではきちんと動くことが確認されているため、組み立てのミスさえなければ、正確かつ効率的に新しいソフトウェアを作ることができます。

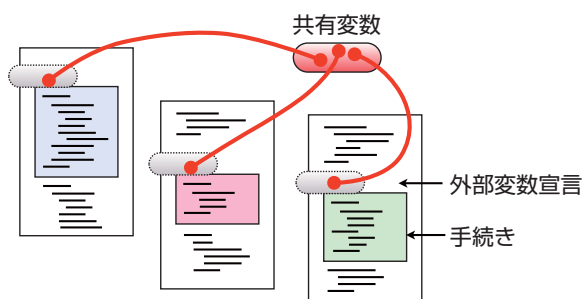
このとき、プログラマーが作るのはモジュール同士をつなぐための「グルーコード」と呼ばれるプログラムです。糊(グルー)の役割をするコードというわけです。

コンポーネント指向 プログラミング

現在のようにソフトウェアのプログラムが膨大な大きさになってくると、グルーコードを作ったり直したりすること自体が膨大な作業となり、そこにミスが起きる余地が出来てしまいます。

近年はグルーコードの記述を抑えるソフトウェア開発手法も考えられていますが、よく使われるC言語というプログラム言語には、そのような手法が使えません。特に日本が得意とする組み込みソフトウェアはC言語での開発が多く、モジュールがC言語で書かれています。

covalの概念 モジュール間で共有する変数と、共有変数に変更された時に起動する手続きから構成する。



C言語はそのまま使いたい、けれどもグルーコードは減らしたい、という相反する希望を叶えるため、私は「coval」という仕組みを提案しました(上図)。

covalは、モジュール間で共有する変数と、共有変数に変更された時に起動する手続きから成っています。各モジュールに設置され、あるモジュールが値を変更すると、共有しているモジュールに通知が送られます。通知を受け取るとcovalに記述されている手続きが実行されます(図1)。

図2のソフトウェアは、スライダーやテキストフィールドに入力された数値をグラフに変換するものです。逆に、グラフを変形させるとスライダーやテキストフィールドの数値も変わります。

このようなソフトウェアを作るとき、従来の方法では、スライダー、テキストフィールド、グラフ、それぞれのモジュールをつなぐため、数値を管理するグルーコードが必要でした。しかし、covalを導入すれば、グルーコードを減らしてシンプルに同じソフトウェアを作ることができます。

covalには、グルーコードによるミスを大きく減らし、モジュールやデバイスを後から付け加えて

も、グルーコードを作り直す必要がない、というメリットがあります。

このような開発思想のことを「コンポーネント指向」と呼びます。コンポーネントステレオの各ユニットのように、ユニットの追加があっても配線をつなぐだけで使えるようにしようという考えかたです。

新しい仕組みが ゴールではない

私は、C言語のグルーコードをなくすための研究で、文部科学省からの科学研究費の交付を受け、covalによるプログラム開発に取り組んでいます。現在は実際の組み込みシステムの環境でもちゃんと動く仕組みを実現するなど、より高度な改良を進めています。

とはいえ、私たちの研究では「こんな仕組みを作りました」ということが最終的なゴールではありません。今まで難しかったことがどれほど簡単になり世の中に役に立っているかを示す必要があるのです。さらに、新しい仕組みを用いるプログラミングの考え方や、開発の方法論を提案していきたいと思っています。

図1

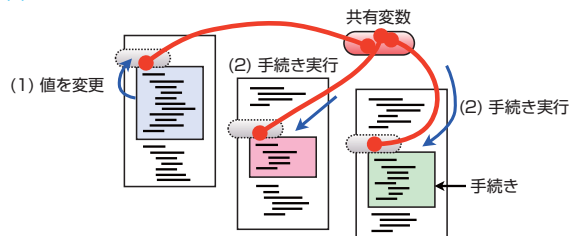


図2 ソフトウェアの部品をつなげる例

