

GUI 直接構成オブジェクトのソースコードの合成¹

1N-7

畠山 正行² 西野 博史³茨城大学工学部情報工学科⁴

1. まえがき

流れ解析などの解析シミュレーションでは、プログラム構成やそれらの変更に柔軟なシステムが要求される。一方、オブジェクト指向パラダイムは、対象世界の要素とオブジェクトの自然な対応付けや、継承による再利用性、構成の変更の柔軟性などの点から見て大変有効である。しかし解析シミュレーションを行うドメインユーザが、必ずしもオブジェクト指向やプログラミングに慣れていないなど、種々の事情があってオブジェクト指向は未知現象の解析シミュレーション等にはほとんど用いられていない [1]。

そこでその様なドメインユーザを対象とした、GUI によるオブジェクトの直接合成及び生成支援環境を構築した。即ち、シミュレーション対象物体の幾何学的なオブジェクトの形状に対して属性やメソッドなどの他のオブジェクト部品を、GUI 操作により目で見ながら直接に張り付けることで形状オブジェクトを合成し、その後C++のソースコードを半自動生成するシステムを設計・実装した。

2. 設計方針

まず、流れの解析シミュレーションを対象世界と限定することにより実装上の汎用性の問題を軽減する。

ドメインユーザがオブジェクト指向プログラミングに消極的な理由の一つは、対象物とプログラム上の「もの」のイメージがうまく結び付かないということがある。これに対し、形状は「モノ」の属性の中でも、目に見えるという人間の分かり易い属性の一つであり、流れ解析では重要な役割を果たす。そこでまず形状データを作成し、可視化された形状オブジェクトを中心にシミュレーションシステムの構築を行えば、直感的にシミュレーション対象とオブジェクトの一対一対応のイメージを保ち易く、全体の設計が容易になると思われる。

解析シミュレーションでは、対象物の形状や離散化された方程式の交換をした後のデータ収集などが頻繁に行われる。対象物の交換毎にデータの作成やオブジェクトへの埋め込みを行っては大変手間がかかる。そこでこれら形状やメソッドをオブション部品とし、必要な部品をオブジェクトへ埋め込み合成及び変更・交換をするまでを半自動的に行う仕組みの実装を考えた。

また、この形状データの作成からソースコードの合成までは形状を目で見えて操作するということから、GUI で全て操作できることを設計方針とする。

形状エディタについては以前に当研究室で開発されたものを使用した [2]。

3. 実装

前記の設計方針に従い、次のものを設計・実装した。

1. オブジェクトの合成に必要な要素を格納するためのデータ構造。
2. 形状データから前記1. で定義したデータ構造上のオブジェクトを合成する仕組み。
3. 前記1. で定義したデータ構造上でオブジェクトを編集・合成する仕組み。
4. 前記1. で定義したデータ構造上のオブジェクトからC++ソースコードを再現する仕組み。
5. これらを統合的に操作するGUI。

以上のうち1.~4.の4個は個別にモジュール化され、5.のGUIにより統合されている。図1にそのGUI表現を示す。この各モジュールはオブジェクト指向に従って設計されている。

GUIは各モジュールの構成オブジェクトに対応させた操作インタフェース（ウインドウ）を提供する。

メソッド取り替えなどの合成にあたってのデータインタフェースなどの整合性の問題は現時点ではドメインユーザ自身が本システムに内蔵されたエディタを用いて編集することで解決するようにしてある。

¹Source Code Synthesis System of GUI Directly Composed Objects

²Masayuki Hatakeyama

³Hirohumi Nishino

⁴Faculty of Engineering, Ibaraki University, Hitachi, Ibaraki 316, Japan

4. システムの操作と合成

本システムでは以下の手順で目標システムを合成する。

1. 形状エディタにより対象の形状データ作成。
2. 形状データを持ったオブジェクトを合成・表現。
3. 形状オブジェクトに必要なメソッドや属性をGUIを使って部品リストから選択して張り付け、目的のオブジェクトを合成する。
4. 必要な部品が無ければ、本システム上で製作する。
5. 合成オブジェクトをC++ソースコードとして出力する。
6. メイン文に組み込み、修正、コンパイル、実行

この内、1から5まではGUIを使った本システム上で実行できる。

5. 実例

本システム上で実際に物体周辺の流れのシミュレーションをするオブジェクト指向の数値風洞システムのプログラムを合成した。出来上がったプログラムに多少の手を加えることにより完全動作する数値風洞プログラムが出来た。本研究で合成した風洞は2次元Burgers数値風洞である。

ユーザが手を加えるべき修正箇所は、メイン文の追加、ヘッダファイルの読み込み部の追加、一部メソッドのイニシャライズの設定、派生オブジェクトコンストラクタの基底オブジェクトコンストラクタ呼び出し部分などである。

6. システムの特徴と評価

本システムでは、形状を見ながらのオブジェクトの合成、オブジェクトとメソッドと属性の独立した編集・取扱い、形状作成からオブジェクトのソースコード合成まで一貫したGUI操作、などを実現した。これによりシミュレーション対象物とプログラム上のオブジェクトのイメージをかなり近付けることが出来た。

その一方、GUIが洗練されていない、全体的な機能不足、C++の仕様に対応仕切れていない、構文解析や整合性のチェック、などの課題を残した。

7. まとめ

解析シミュレーション分野を対象とした、GUIを使ったオブジェクトのソースコードの合成システムの設計と実装を試みてきた。

今の段階では本システム全体にC++言語の影響が強く現れていて、ドメインユーザに分かり易いシステムとは言いがたい。今後徐々にシステムをユーザ側に近付けて分かり易いシステムにして行きたいと思う。

参考文献

- [1] 加藤木和夫、畠山正行、小林秀行、「シミュレーション向けのオブジェクト生成支援環境の設計と実装」オブジェクト指向'95シンポジウム、情報処理学会シンポジウム論文集、Vol. 95、No. 3、pp. 205-212、1995年6月2日。
- [2] 長谷川好則、「オブジェクトベース世界の直接生成・接触・操作機構の研究」平成6年度茨城大学工学部情報工学科卒業論文、平成7年2月13日。



図1: ソースコード合成システムのGUIイメージ