

オブジェクト指向設計図からの構造定義プログラムの 自動生成システムの開発研究

1 K-5

三宅 直矢* 原田 実**
 青山学院大学理工学研究科経営工学専攻*
 青山学院大学理工学部経営工学科**

1. はじめに

近年オブジェクト指向分析・設計のためのCASEツールとして、すでに分析・設計図の作図エディタが商品化されている。しかしこれらのCASEツールが一貫した支援系になるためには、システム構築の特に下流工程（コード生成）とつながる必要がある。本研究ではC++プログラミングの特に初期段階に必要である（プログラム）構造定義を自動生成するシステムを開発した。なおモデリングツールとしては構造化された設計図を表現できるSOME[1]を採用した。

2. プログラム構造定義の構成

本研究ではC++プログラム構築における構造定義を派生定義とクラス定義に分割した。C++の文法より、クラス名を定義する行には、そのクラスとそのクラスのスーパークラス、加えてクラステンプレート定義を記述する。本研究では、この定義群の総称を”派生定義”と呼ぶ。次にクラス定義は3つのアクセス権public、private、protectedに分けて記述することにした。これらの定義を特に”クラス内部private定義”、“クラス内部protected定義”、“クラス内部public定義”と呼ぶ。

3. 構造化オブジェクトモデリング技法SOMMと統合環境SOME

SOMMでは、①設計図を階層的に構造化し視覚的な向上を図り、②クラス単位で設計図を再利用でき、③分析からC++コード生成に至るまでの詳細な設計情報が表現可能である。

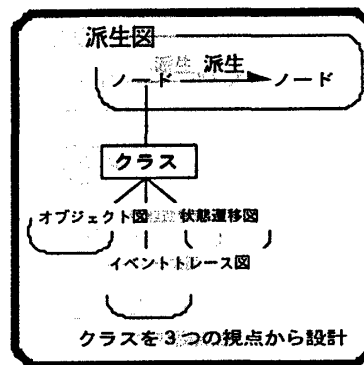


図1 SOMMの4モデル

さらにSOMEは分析法としてSOMMを採用した統合モデリング環境で、①設計対象を表現する全設計図を統合的に管理、②再帰グラフ理論を基にした形式性による設計対象/操作の厳密な意味定義などが可能である。このような利点があるので、SOMEを利用した。

4. 自動生成

本研究では、SOMMの各図の図形の持つ情報と複数図形間の接合関係をC++のプログラムに変換する知識をクリシェとしてルール化した。クリシェには、①ルール中の記述をそのままコード化する部分、②内容に応じて変更してコード化する部分（@で始まる）、③検索条件を記述している部分で構成されている。図2は派生定義を出力するまでの流れである。ここで、例えばクリシェの①は”class”、②は@Ccや@S、③は”[1.1]”以下の文章などとして記述されている。

5. 本システムの構成

本システムはSOME上で設計された図を入力とし

て、C++のクラス定義を出力する。

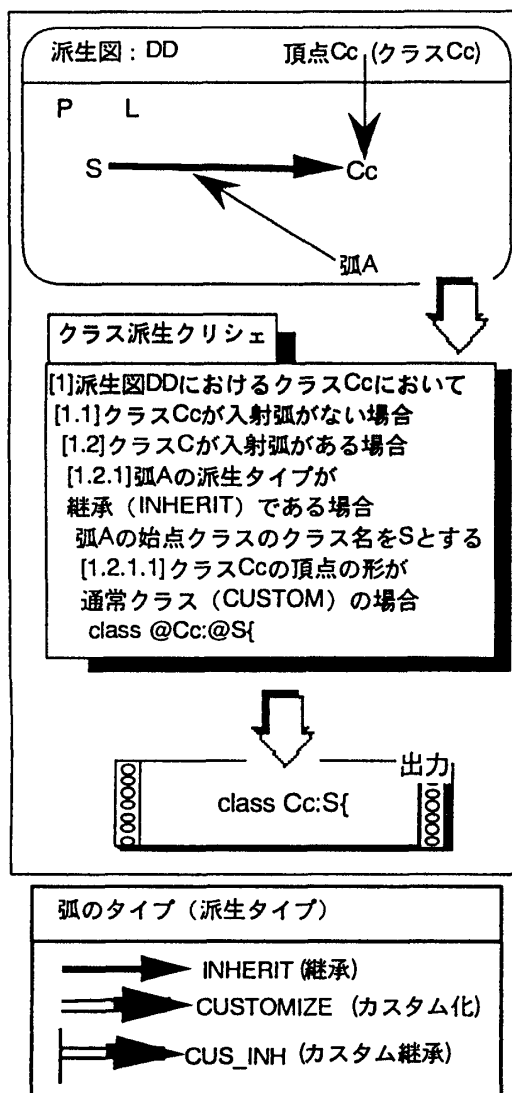


図2 図情報とクリシェ

(例：クラス派生定義クリシェ)

システムはまず、SOMEの持つ設計データベースからクリシェに埋め込むべき情報を検索する。定義順序は、クラス派生定義、クラス内部public定義、private定義、protected定義の順である。

例えば、クラス (頂点) Ccのスーパークラスを抽出するためには、頂点Ccが入射弧を持つかどうか注目に注目する。この場合頂点Ccは入射弧Aを持っている。そして弧Aは始点頂点Sを持っている。弧Aのタイプは「継承」であるからこれらの情報をまとめると「クラスCcはクラスSを継承している」と言える。よって、C++での意味は以下のようになる。

```
class Cc : S { . . . };
```

各クラス内部定義も同様にクリシェを利用してコード生成を行う。このようなクリシェが4個ある。また生成されたクラス定義には「#ifdef」、「#define」、「#endif」などのプリプロセッサ情報も付加されている。

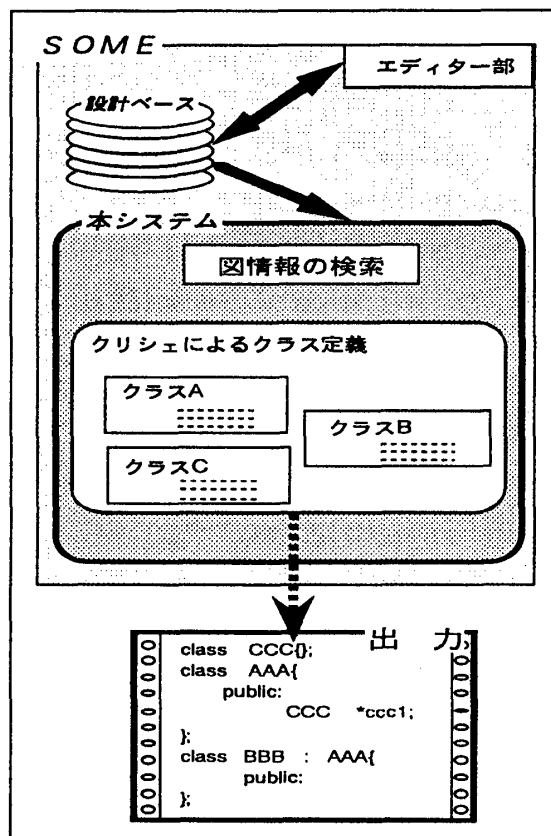


図3 本システムの構成

6. まとめ

以上、オブジェクト指向設計図からのクラス定義の自動生成システムについて述べた。

本システムから自動生成されたクラス定義ファイルは、C++のコードとして正確なものであった。これによりソフトウェア開発の作業量の減少が期待できる。今後、操作のインプリメントの部分などの自動生成が課題となる。

参考文献

[1][原田94]原田実,澤田隆史,藤沢照忠:"構造化オブジェクトモデリング環境SOME-SOMMに基づくOOA/OOD-",情報処理学会論文誌,94-SE-101,pp.1-8,(1994.11).環