

オブジェクト指向によるインタラクティブ橋梁計画・設計システム のモデル化に関する研究

名古屋大学理工科学総合研究センター 正 会 員 〇井上 智正
名古屋大学理工科学総合研究センター フェロー 伊藤 義人

1. はじめに

近年のコンピュータの発展にともない、橋梁計画や橋梁設計を支援するプログラムが数多く開発されてきた。これらのシステムは橋梁計画・設計業務を飛躍的に効率化したが、その多くが単純な繰り返し作業をプログラム化しただけであり、橋梁計画・設計における本格的なコンピュータ支援は未だ実現したとはいえない。すなわち、橋梁上部構造、下部構造、および基礎構造の各構造物の計画・設計を行うプログラムは数多く実用化されたが、橋梁計画・設計全体を統合するインタラクティブなシステムは、実用化されていないのが現状である。

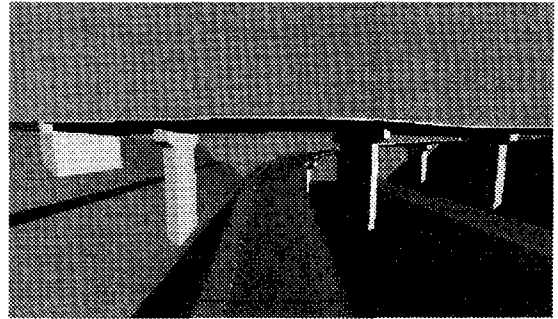


図-1 開発中のプロトタイプの3D表示

その原因は、橋梁計画・設計全体を1つのシステムとしてモデル化する手法が確立されていない点にある。すなわち、従来のシステム開発手法である構造化システム開発手法では、橋梁計画・設計全体をモデル化することが困難であるのだと考えられる。従来の開発手法の特徴は、システムに要求される複雑で抽象的な機能を単純で具体的な機能として階層的に細分化し、機能単位でシステム開発を行う機能中心のシステム開発手法である。しかし、この開発手法は、橋梁計画・設計のように複数の構造物が相互に影響を与えあう複雑な関係をモデル化することは難しい。

すなわち、橋梁計画・設計業務に従事する技術者の念頭に存在するのは、上部構造、下部構造、基礎構造、道路、地形などの「物」であり、橋梁計画・設計はこれらの物の構造および物同士の関係をインタラクティブに調整することにより進められる。もし、このような物中心のシステムをわざわざ機能中心の観点からモデル化すれば、本来の構造を大きく歪曲することになり、結果としてシステム開発が複雑化してしまう。したがって、橋梁計画・設計全体のモデル化には、物中心のシステム開発手法を採用する方が適していると考えられる。

そこで、本研究では物中心のシステム開発手法であるオブジェクト指向をインタラクティブな橋梁計画・設計システムの開発へ適用する手法を検討し、それにもとづきプロトタイプシステム（図-1）を開発することを目的としている。なお、本研究で採用したオブジェクト指向手法は、オブジェクト指向方法論OMT第1版に基づいている¹⁾。

2. オブジェクト指向による橋梁計画・設計のモデル化概要

オブジェクト指向の主な基礎概念を橋梁計画・設計のモデル化にどのように応用したか、例を挙げて簡単に説明する。ただし、大規模なシステムのオブジェクト構造を全て示すことはできないので、以下には説明のために大幅に簡略化したものを示す。

1) データ抽象と抽象データ型 (Object Based and Class Based)

最初にデータ抽象について、図-2に示すカプセル化とデータ隠蔽の2つに分けて説明する。カプセル化とは、物をデータとそのデータを処理する操作から構成されていると考えて、それらを1つのカプセルにまとめる仕組みである。例えば橋梁下部構造の橋台をカプセル化すると、橋台の構造寸法などデータと安定計算などの操作を1つのカプセルにまとめることである。このようにデータと関連のある操作をまとめることにより、複雑なものを単純化できる。またデータ隠蔽とは、カプセル化したオブジェクト内部のデータを外部から隔離し、データ変更は操作を介してのみ許可される仕組みのことである。これによりバグを含む不正なデータ変更による内部データの破壊を防止することができる。例えば橋台の底版寸法を不正

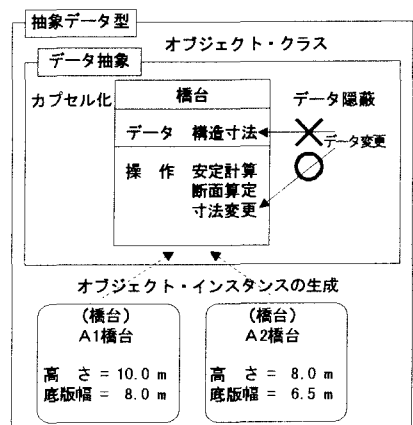


図-2 データ抽象と抽象型の例

に減じたことにより縁端杭が底版から外れてしまうような問題は底版寸法変更用の操作内部で未然に防止できる。

次に抽象データ型について説明する。抽象データ型とはデータ抽象化されたオブジェクトを型として、オブジェクト・インスタンスを生成する仕組みである。例えば、橋台という型から実際の計画・設計に用いられる寸法値を持った橋台のオブジェクト・インスタンスが生成される。

2) 継承 (Inheritance)

継承とは、親にあたる抽象データ型の持つデータや操作を、子にあたる抽象データ型が引き継いで自分のものとして扱える仕組みである。子の抽象データ型は親から継承したデータと操作に加えて、子独自のデータや操作を持つことができる。例えば、橋台は、逆T式橋台やラーメン式橋台など複数の構造形式に分類できるが、どのような構造形式の橋台でも必ず共通して持っているデータや操作がある。

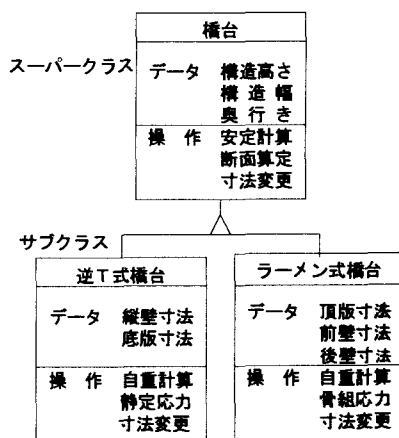


図-3 継承の例

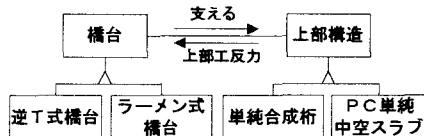


図-4 関連の例

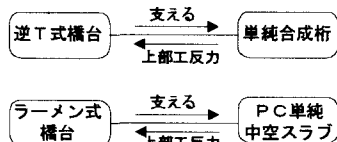


図4に示すようにスーパークラス間に関連があればそのサブクラスのオブジェクト・インスタンス間の組み合わせは自由に選択できる。

図-5 多態性の例

ある。例えば、図-5に示すようにスーパークラス橋台がスーパークラス橋梁上部構造を支えているという関連を持つていれば、スーパークラス橋台を継承するすべてのサブクラス橋台は同様に上部構造を支えるという関連を持つ。これにより、橋梁上部構造を支える橋台の構造形式を自由に変更できる動的結合という仕組みが実現される。

4) 集約 (Aggregation)

集約は関連の特殊形であり、あるオブジェクトクラスが他のオブジェクトクラスを部品として従属させる仕組みである。例えば、図-6に示すように道路橋梁の計画・設計オブジェクトクラスは、地形、道路および橋梁オブジェクトクラスを部品として従属させている。また、道路オブジェクトクラスは、平面線形や縦断線形などのオブジェクトクラスを従属させ、橋梁オブジェクトクラスは上部構造、下部構造および基礎構造オブジェクトクラスを部品として従属させている。このようなオブジェクトの階層構造をとることで、システムに要求される操作は基本的にトップダウン型で処理されることが強調され、システム内の処理の流れを明確にできる。

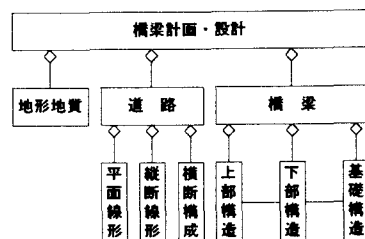


図-6 集約の例

3. さいごに

現在、本研究で作成中のプロトタイプシステムの上部構造や下部構造などのクラスはすべて構造形式を持たない最上位のスーパークラスのままである。今後は、上部構造、下部構造および基礎構造のスーパークラスを継承し、具体的な構造形式を持つサブクラスを複数作成し、その上で、本プロトタイプシステムによる実験と検討を重ね、本研究で開発した手法を本格的なシステムの実装に備えて洗練していく予定である。

【参考文献】1) J. Rumbough et. al. : Object-Oriented Modeling and Design, Prentice Hall, 1991.