

第 1 章

序論

1.1 研究の背景

図形は情報の表現や伝達さらには思考の道具として広い範囲で用いられている。図形の中で図形要素や図形要素の配置規則がはっきりしており、文章や数式と同じように、意味が一義的に導き出せるものを特に図形言語と呼ぶ [1]。図形言語は、テキスト言語より情報を理解しやすくする。情報を表すとき、テキスト言語では文字を一次元に並べるが、図形言語では図形や立体を二次元あるいは三次元に配置することで情報を視覚的に表すことができる。こうした図形言語は、図形要素間になり立っている関係を表す空間的な文法を持っていると考えることができる。

空間パーサ生成系とは、図形の空間的な文法を定義することで図形の空間パーサを生成するシステムのことである [2] [3] [4] [5] [6]。

空間パーサ生成系についての研究として、SPARGEN[2] や Penguins[3] [4] などの研究がある。Golin らにより提案されている SPARGEN は、OOPLG (Object-Oriented Picture Layout Grammars) を用いて図形言語の文法を定義することで空間パーサを生成するシステムである。OOPLG では、図形の属性や制約を C++ を用いて定義している。また、Marriot らの Penguins では、図形の文法として Constraint Multiset Grammars (CMG) [3] [7] を用いて空間パーサを生成している。

しかしながら、これらの空間パーサ生成系は、テキストを用いて図形言語の文法を定義するので、ユーザは文法を理解している必要があり、一般のユーザにとって使いやすいものとはいえないという問題があった。

そこで、我々が研究を行っている恵比寿 [5] [6] [8] では、ユーザが入力した図形

を用いて大まかな図形の CMG の文法を自動的に生成するようにしている。そのあとユーザは、生成された文法を見ながら、制約の追加または削除などをおこない修正する。また、VIC[9] では視覚的な制約入力インターフェイスを恵比寿上に実装し、テキスト編集を行わずに CMG の文法を生成している。

図形を処理するビジュアルシステムは、空間パーサを用いて図形を解釈し、その図形を描き変えることが多い。ここで、図形を描き変えるということは、図形を生成、削除、移動したり、図形の属性の値（色や文字列の値、線の太さなど）を変更するアクションである。

図形が描き変えられるとき、図形の生成により図形の数が増えて重なることや図形の削除により図形間の位置関係が分かりにくくなることなどの問題が生じる。この問題の解決方法としては、ユーザが直接図形をレイアウトする方法がある。しかし、図形を処理するシステムはもっとインタラクティブであるべきだと考える。

本研究では、ビジュアルシステム生成系にレイアウト制約を扱えるようにし、図形の一部または全体を解釈しながらバランスよくレイアウトすることができるビジュアルシステムの研究を行った [10]。その研究に基づいて、我々はビジュアルシステム生成系「Rainbow」を開発した。「Rainbow」は、我々がこれまで研究を行ってきた恵比寿にレイアウト制約を追加することにより実現している。

1.2 研究の動機

我々は、オブジェクト指向に基づくソフトウェア開発方法論である OMT (Object Modeling Technique) [11] のモデルから実行可能なオブジェクト指向コードに変換を行うための新しい手法について研究してきた。特に、設計と実装フェーズにおいて、オブジェクトモデル中のクラスの振舞いを分かるため最も重要な動的モデルの仕様から実行可能な Java コードを生成する方法について研究を行った [13] [14] [15]。

また、クラスの振舞いを表す動的モデルの活動 (activity)、動作 (action) に「アイコン変形 (icon transformation)」を定義し、それからシステムをアニメーションさせるコードを生成する方法を研究した [16] [17]。

これらの研究で用いられたオブジェクトモデル、動的モデルを表現するのに使わ

れる図形表記のオブジェクト図、状態遷移図は、分析フェーズから設計フェーズ、実装フェーズまでを続き目なく同一の表記で表すことにより、ある開発フェーズで追加された情報を、次のフェーズで失ったり翻訳し直したりする必要がなくなる。また、これらの図形はシステムを開発する複数の開発者により利用することが可能であり、開発者間のコミュニケーションの道具になる。

今まで研究されている空間パーサでは、図形の一部または全体を解釈しながら分かりやすくレイアウトすることができない。ビジュアルシステム生成系にレイアウト制約を導入することにより、これらの問題を扱うことが可能ではないかと考えたのが本研究の動機である。

1.3 研究の目的

本研究の目的は、ビジュアルシステム生成系にレイアウト制約を導入し、扱えるビジュアルシステムの範囲を広げることである。

「Rainbow」を用いることにより、ソフトウェア開発において重要視される各種の設計図、工程図、データの流れ図、回路図、E-R ダイアグラム、組織図、家系図、および、テレビドラマの登場人物の関係を表す図などの様々な図形をもっとインタラクティブに扱うビジュアルシステムを作ることが可能になる。すなわち、空間パーサにレイアウト制約を追加することにより、パーシング中に図形を自動描画することで図形をよりインタラクティブに処理するビジュアルシステムを作ることが可能である。

特に、従来のオブジェクト指向 CASE ツール [18] [19] は、モデルを表現するのに使われる図形表記を描くための図形エディタを備えている。しかしながら、それらの図形エディタでは、図形表記を構成する基本部品をユーザがすべて配置するといった操作環境となっているため、ユーザが分かりやすいレイアウトを考慮しながら配置していなければならない。ここで、「Rainbow」はインタラクティブな CASE ツールを作るための重要な要素技術の一つを提供すると考えられる。