

グラフィカルエディタ作成用ツールキット GIST (1)

1D-1

— 概要 —

Paul C. Brown Terry M. Topka

General Electric Company, Corporate Research and Development

井上 健 土田 崇 上原 みな 村田 なつめ ○渡邊 多恵子

横河電機株式会社 オープンシステム研究所

1 序

GIST(Graphical Interactive Systems Toolkit) は対話型、特にグラフィカル・ユーザインタフェースを持つアプリケーションの開発を支援するためのツールキットである。GISTはC++のクラスライブラリとして提供され、開発者は用意されているクラスのサブクラスを作成することにより、アプリケーション固有の機能を実現する。

GISTは米国GE中央研究所および横河電機オープンシステム研究所の共同研究による成果であり、同じく共同で開発されたC++OMToolを用いて開発された。

本稿ではGISTの機能およびその特徴を列挙し、ついで特徴のいくつかについて詳しく説明する。

2 機能と特徴

● 構造化グラフィクス

直線、曲線、多角形、円、テキストなどの図形プリミティブ(グラフィクス)が用意されており、これらの図形の移動や拡大/縮小、回転など任意の座標変換を施すことができる。

● 図形間の接続関係の表現

情報を視覚的に表現するイメージはグラフィクスから構成される。これらのイメージを連結したり包含させたりするための手段が用意されており、イメージ間の意味的な関係を表現することができる。図形の移動や変更による影響の計算や表示の更新は、GISTが自動的に行なう。

● アプリケーションモデル

アプリケーションが取り扱う(非視覚的な)情報(サブジェクト)とイメージの間の写像をつかさどるビューを導入することにより、階層的なモデル構築の枠組を提供している。これにより、データの変更時に表示の整合性を保つための機構を利用することができるので、ひとつのデータをさまざまな表現形態で表示し、ユーザの操作による変更に対応して常に無矛盾の状態を保つことが容易に実現できる。

● 状態遷移記述言語

対話処理を記述するための状態遷移記述言語が用意されている。この言語を用いてアプリケーションの動的な振舞いを定義する。コントローラがこの記述に基づいてイベントの解釈、すなわち状態遷移およびアクションの実行をつかさどる。

● デバイス依存部の分離

デバイス依存部とその他の部分は分離されており、デバイス依存部はXウィンドウおよびPostScriptに対応している。アプリケーションはXウィンドウのクライアントとして構築され、ボタンなどのウィジェットのレイアウトにはMotifのUILを使用する。またX11R5の国際化対応に準拠しており、日本語をはじめ多国語入力/表示が可能である。画面の情報はPostScript形式で出力できる。

2.1 図形間の接続関係の表現

アプリケーションでは、図形と図形の間に連結や包含といった幾何的な接続関係を与えることによって、意味的な関係を表すことが多い。このような接続関係が図形の移動や変形などの変化に対して保たれるようなアプリケーションを開発するには、煩雑な処理を実現することが必要になる。

GISTではイメージ(図形)相互の幾何的な接続関係を明示的に表現、操作するためにパッドおよびジョイ

GIST — A Toolkit for Graphical Editor Development
Paul C. Brown, Terry M. Topka
Corporate Research & Development, General Electric Company
P.O.Box 8, Schenectady, NY 12301, USA
Takeshi Inoue, Takamu Tsuchida, Mina Uehara,
Natsume Murata, Taeko Watanabe
Open Systems Laboratory, Yokogawa Electric Corporation
2-9-32 Nakacho, Musashino-shi, Tokyo, 180 Japan

ントというオブジェクトを用意している。パッドは、イメージ上に定義される他のイメージとの接続領域である。イメージは通常その前面、背面および周囲の辺上（領域図形の場合）または頂点（線分の場合）にパッドを持っている。一方ジョイントはパッドどうしの接続形態の種類を表すオブジェクトである。イメージとイメージを接続するにはそれぞれのイメージの持つパッドをひとつのジョイントに結び付ければよい。

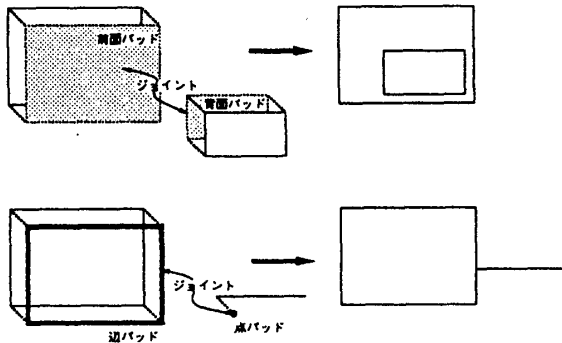


図1. イメージのパッドとジョイント

例えば四角形と円を直線で結ぶ場合、四角形と円それぞれの周囲の辺のパッドに対してジョイントを用意し、それらの辺のパッドと直線の端点のパッドをそれぞれのジョイントに結び付ける。また、四角形の中に別の四角形を入れて包含させる場合、外側の四角形の前面のパッドと、内側の四角形の背面のパッドをジョイントに結び付ける。いずれの場合にも、一部あるいは全部の図形の移動や変形に対してその接続関係は常に保たれる。すなわち、前者の例では四角形を移動しても四角形と円を結ぶ直線は連結したままで伸縮して追従し、また後者の例では、内側の四角形を移動しても外側の四角形からはみださないように外側の四角形が移動もしくは変形する。

関係を保つための図形の動作の仕方によっていくつかの異なる種類のジョイントが用意されている。領域図形が別の図形を包含している場合、内側の図形が外側の図形に接している状態で更に移動しても外側の図形からはみださないように、外側の図形が移動するジョイント、外側の図形が変形するジョイントがある。また、内側の図形が外側の図形をすきまなく覆うように外側もしくは内側の図形が変形するジョイントもある。

これらのジョイントを必要な動作に応じて使用し、またイメージの接続位置を明示的に制限したい場合にはイメージに新たにパッドを付加するなどにより、所望の形態で図形どうしを接続することを容易に実現することができる。

2.2 アプリケーションモデル

GIST では、同一の情報を複数の異なった場所にもしくは異なった形態で表示することを可能にしており、そのうちのいずれかがユーザの操作などにより変更された場合に、元の情報そのものと同時に残りの表示を更新するための機構が提供されている。

アプリケーションが取り扱う情報であるサブジェクトと、その情報の視覚的な表現であるイメージとの間を関連づけるのがビューである。InterViews [1] では、ビューは図形表示の定義と、その図形とサブジェクトの関係の両方を持っているため、あるアプリケーションのビューを修正せずに再利用することはむずかしい。

一方 GIST では InterViews のビューに相当するものを図形表示を定義するイメージと、単にそのイメージとサブジェクトとを対応づけるビューとに分離している。これにより、図形表示を取り扱うさまざまなイメージクラスはすべてのアプリケーションで修正せずに再利用することができる。

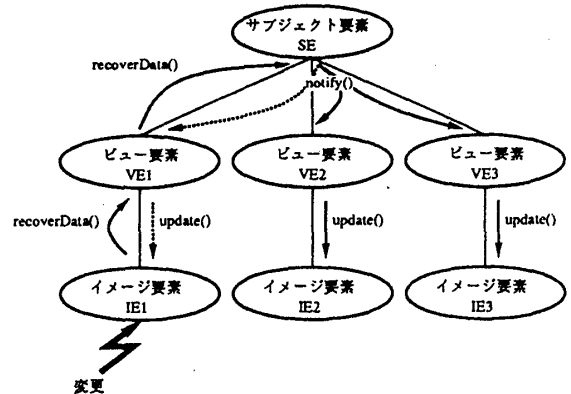


図2. サブジェクト-ビュー-イメージによるデータ管理

3 まとめ

以上に述べたような特徴により、GIST はグラフィクスを用いた対話的なエディタの開発に適している。チャートエディタやグラフィックビルダにとどまらず、情報を視覚的に表現し、情報の変化やユーザの操作に応じて図形を変化させるようなあらゆる分野のアプリケーションの開発に有用である。

今後は、より簡単に GIST アプリケーション開発ができるような環境を整えて、ユーザに提供していく。

[参考文献]

- [1] M. A. Linton, et al, "InterViews: A C++ Graphical Interface Toolkit", Technical Report CSL-TR-88-358, Stanford University, July 1988.