

# Viola: 図式表現を用いたビジュアルシステムの 編集・実行システム\*

亀山 裕亮<sup>†</sup> 志築 文太郎<sup>†</sup> 田中 二郎<sup>†</sup>  
<sup>†</sup> 筑波大学大学院 システム情報工学研究科  
{kame, shizuki, jiro}@iplab.cs.tsukuba.ac.jp

## 1 はじめに

ビジュアルシステムにおいて、図形が決められた規則(ルール)のもとに組み合わせられている図式を図形言語と呼ぶ。ビジュアルシステムでは入力された図形に対して、ルールの適用を試みることで図形言語の解析を行なっている。このルールを、ビジュアルシステム上で図式表現を編集することで定義することができれば、ビジュアルシステムを容易に作成することが可能になる。

## 2 ビジュアルシステム Viola

我々は、図式表現を用いたルールの定義と、定義したルールに基づいた入力図形の解析、を行なうことができるビジュアルシステム Viola を作成した。

### 2.1 図式表現を用いたルールの定義

Viola ではルールの文法として拡張 CMG[1] を用いている。拡張 CMG は Marriott らにより提案された CMG[5] にアクションと呼ばれる図形の書き換え規則を追加した文法である。拡張 CMG を用いたルールの記述は構成要素、制約、属性、アクションにより構成されている。我々はこれらの要素を図式表現を用いて定義する手法を考案した [4]。Viola ではこの手法を用いることで、以下の手順により図形言語のルールをグラフィカルに定義することが可能である。

**ルールの定義範囲の指定** まず、画面上にルールを定義する範囲(図 2)を描画する。この枠の中で図式を編集することによりルールの定義を行なう。

**構成要素の定義** ルールにより解析される図形要素を構成要素と呼ぶ。構成要素については、枠内(図 2)の左側で直接図形を描くことにより定義を行う。

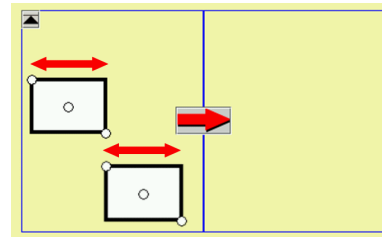


図 2: ルールの定義範囲

**制約の定義** ルールが適用されるために、構成要素となる図形要素が満たさなければならない条件を制約と呼ぶ。制約は、左側の枠の中に描いた構成要素の図形を操作し、満たさなければなら条件を例示することにより定義を行う。定義した制約は画面上に図示される。例えば幅が同じであるという制約であれば、図 2 のように赤い矢印を用いて制約が図示される。

**アクションの定義** 図形の削除や移動など、ルールが適用された時に実行される動作をアクションと呼ぶ。構成要素及び制約の定義を終えた後、中央にある矢印をクリックすることで、構成要素として描いた図形が右側の画面に複製される。複製された図形を編集し、アクションが実行された後の図式を作成することで、アクションの定義を行うことができる。

**ジェスチャの定義** ジェスチャーとはペンやマウスを用いて特定の軌跡を描くことにより、図形の削除や追加などの操作を行う機能のことである。Viola では手書きストロークを図形と同様に扱うことにより、ジェスチャーを定義し使用することができる。円の上でギザギザの手書きストロークを描いた時に円を削除するというジェスチャーを定義を図 3 に示す。

\*Viola: definition and execution system for defining the visual system. Hiroaki KAMEYAMA, Buntarou SHIZUKI, Jiro TANAKA. Department of Computer Science, University of Tsukuba.

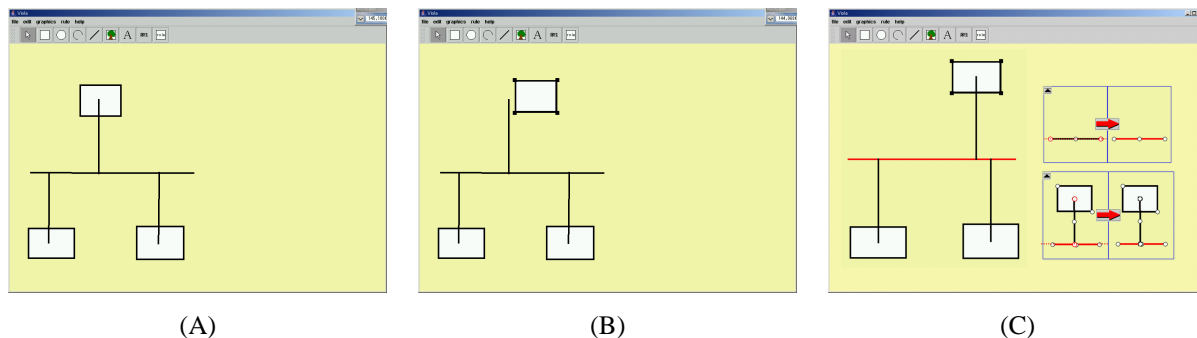


図 1: ネットワーク図の文法の定義と実行例

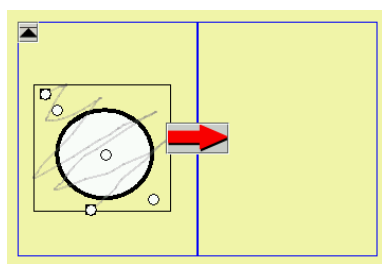


図 3: 円を削除するジェスチャーの定義

## 2.2 ビジュアルシステムの実行

Viola では描かれた図形に対しルールへの適応が可能かどうかインクリメンタルに解析を行う。ルールが適応可能な図形に対しては、図形間に制約を課し、アクションの実行を行なう。制約が課されることにより図形を編集しても図形間の制約関係が保持される。Viola では制約解消系に、幾何制約を扱うことができる Chorus[3] を用いている。

## 3 図形言語の定義とビジュアルシステムの実行の例

何もルールを定義していない状態でも図 1-(A) のようにネットワーク図を描くことは可能である。しかしネットワーク図としてもルールが何も定義されていないため、描いた図はネットワーク図としては認識されていない。この状態で図 1-(B) のように四角形を動かしてしまうと、ネットワーク図が崩れてしまう。

ここで、ネットワーク図の幹線となる線を定義し、次にネットワーク機器と幹線を結ぶ線を定義することにより、図 1-(A) の図式がネットワーク図として認識される。ルールの定義後では図 1-(C) のようにネットワーク機器を表わす四角形を動かしても、Viola により図形間の関係が保持されるため、ネットワーク図のレイアウトが崩れることはなくなる。

このように、ルールを定義してゆくたびに、始めに

描いたネットワーク図にはそのルールに従った制約が課せられる。ユーザは図が意図した通り動くかどうかにより、定義したルールが正しいかどうか容易に理解することができる。

## 4 関連研究

Viola では図形の書き換え前と書き換え後を表現する 2 つの定義インタフェースを用いて図形文法の定義を行う。このような図式表現を用いて書き換え規則を定義する研究として Visulan [6] や VISPATCH[2] などがある。書き換えが行われる前後の画面を使って文法を定義するという点ではこれらの研究と Viola は類似している。しかし、Viola では図形の書き換え規則 (アクション) 以外にも、規則の適用条件である制約や、属性についても図式表現を例示することにより、グラフィカルに文法を定義し、その文法に基づいてビジュアルシステムを生成することができるという点が異っている。

## 参考文献

- [1] Akihiro Baba and Jiro Tanaka. Evis: A visual system having a spatial parser generator. In *Proc. of Asia Pacific Computer Human Interaction*, pp. 158–164, July 1998.
- [2] Yasunori Harada, Kenji Miyamoto, and Rikio Onai. Vispatch: graphical rule-based language controlled by user event. In *Proc. of the 1997 IEEE Symposium on Visual Languages*, 1997.
- [3] Hiroshi Hosobe. A modular geometric constraint solver for user interface applications. In *Proc. of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST'01)*, Vol. 3, pp. 91–100, 2001.
- [4] 亀山裕亮, 飯塚和久, 志築文太郎, 田中二郎. GIGA: 空間解析器生成系におけるグラフィカルな文法編集システム. *情報処理学会論文誌*, Vol. 44, No. 11, pp. 2565–2574, November 2003.
- [5] Kim Marriott. Constraint multiset grammars. In *Proc. of the IEEE Symposium on Visual Languages*, pp. 118–125, 1994.
- [6] Kakuya Yamamoto. Visulan: A visual programming language for self-changing bitmap. In *Proc. of International Conference on Visual Information Systems, Victoria University of Tech. cooperation with IEEE*, pp. 88–96, 1996.