可视化语言文法形式化描述综述*)

许红霞 张 莉
(北京航空航天大学软件工程研究所 北京100083)

摘 要 可视化是人机交互的主要形式,可视化语言是计算机科学中一个重要研究领域,文法为可视化语言提供了一种有价值的形式化描述方法。本文基于可视化语言的特征,介绍了可视化语言文法形式化描述体系的基本理论,分析了几种典型形式模型,并探讨了当前的主要研究内容和面临的挑战。

关键词 可视化语言,形式化描述体系,图形文法,属性多重集合文法

A Survey on Visual Languages Formal Specification by Grammatical Approach

XU Hong-Xia ZHANG Li
(Software Engineering Institute, Beijing University of Aeronautics and Astronautics, Beijing 100083)

Abstract Visual languages is an important component of human-computer interaction. Grammar provides a useful formalism for specifying visual languages. This paper begins with the analysis of characterization of visual languages, then reviews theoretical aspects of grammar-like visual language formal specification, particularly of graph grammar and attributed multiset grammar, and also discusses several specification formalisms in the field. Several current research tasks and their challenge are pointed out finally.

Keywords Visual languages, Formal specification, Graph grammar, Attributed multiset grammar

1 引言

随着计算机技术的发展和大型、复杂软件项目的开发需求,需采取相应的技术手段来支持不同阶段、背景人员的交流。可视化是目前被广泛认可的一种方式,如可视化建模、可视化数据库查询、可视化编程、算法动画等方法,已应用于软件开发周期的不同阶段。

软件可视化分为两个领域:程序可视化(program visualization)和可视化编程(visual programming)[1]。前者是运用可视化技术显示数据结构、算法、程序控制流程等,后者指用可视对象及其空间排列来构成软件,即使用图形作为编程媒介来表示一种计算,从而需要应用一种可视化语言(visual language)来支持计算的有形表示。

可视化语言理论研究的基本问题是如何对其进行自然、简洁的描述。由于应用范围广泛,不可能规定一种统一的、适用于所有领域可视化语言的描述方法,从开始研究可视化语言以来,出现了许多不同的形式化描述方法,令人们难以适从。本文从可视化语言的基本特征出发,着重分析了基于文法的描述方法的研究状况,为可视化语言理论的研究、以及其它相关领域,如人机交互、空间数据库、形式语言的研究提供有价值的参考。

本文首先介绍了可视化语言的基本特征,然后讨论了基于文法的可视化语言描述方法的研究状况,并对当前几种典型的形式体系进行对比分析,最后探讨了可视化语言形式描述体系的主要研究内容和面临的挑战。

2 可视化语言的特征

2.1 可视化语言的定义

虽然可视化语言已经在很多领域得到广泛应用,但是,对于术语"可视化语言",至今还没有一个标准定义,通常,大家认为可视化语言就是系统地使用可视化表示方法来传递信息。可视化语句是由一组图符按照一定的规则在两维或多维空间组合而成[2]。鉴于本文的主要目的是分析可视化语言的文法形式描述方法,因此,将可视化语言视为在语言层为有效"语句"的一些图形集合,其中每个图形是位于多维空间的一组符号,语句的有效性依赖于符号之间的空间关系,因此,本文所指的可视化语言并不局限于可视化程序设计语言、或是可视化建模语言,而是泛指应用于计算机软件开发中的可视化语言。

2.2 可视化语言与文本语言的比较

可视化语言的特征往往是在和文本语言的比较中体现出来的。两者的相似之处在于都通过语法和语义来定义,由一种基础符号集组成、能被分解成一种明确的结构。不同之处在于:

(1)非线性次序。文本语言是由一系列符号顺序组成,而可视化语言是二维的,简单的线性次序不能表示所有的邻接关系。

(2)多种组合方式。文本语言的符号之间只有一维串连关系,因此,组成规则在语言描述中是隐式的,而可视化语言的图符之间存在许多不同的组合方式,如邻接、覆盖、相交等,这些用于图符空间排列的组成规则需要在语言规范中明确说明。

(3)基本图形结构。文本语言可以构造为树型结构,但可视化语言的基本结构通常是有向图。

因此,语素的图形化表示和多维化表示是可视化语言的特征。不同类型的可视化语言往往具有不同的图形对象和不

*)航空科学基金(00F51058)、北京市科技新星计划(H013610270112)。许红霞 在职博士,讲师,主要研究方向是可视化建模语言、软件工程环境。张 莉 博士,教授,博导,主要研究方向是软件工程、过程工程环境、软件体系结构、可视化建模语言。

同的图形组成方式。

3 可视化语言文法形式化描述研究状况

早在20世纪60年代就开始了可视化语言的理论基础研究，出现了许多不同的形式化描述方法，如文法的、逻辑的和代数的[3]。其中，逻辑方法和代数方法由于需要较深的数学基础，难以普及应用，主要分别用于人工智能领域和人机交互。而文法方法以计算机科学中的形式语言理论为基础，有大量成熟的理论和应用成果可利用，成为目前被广泛使用的一种方法。可视化语言的文法描述方法对字符串语言的传统重写机制进行了扩展，虽然仍通过产生式来描述语法，但基本结构不再是字符串，而是图形对象集合或多重集合、并规定了重写对象间的几何关系。本文主要分析各种不同的文法描述方法。

3.1 早期发展形式

早期的可视化语言文法描述方法是对文本语言短语结构文法的直接修改——将一维串连拓展为二维连接，产生式则是由一系列图形符号用二元连接操作符衔接而成的线性串。如形状文法(shape grammar)[4]、树型文法(tree grammar)[5]，规定每个图形符号都有两个连接点：一个“头”和一个“尾”，通过“头”和“尾”的不同连接方式构成二维图形。另一种扩展方式是将图形符号视为矩形块，位于网格或阵列中，包括水平串连和垂直串连两种组合关系，如早期的位置文法(positional grammar)[6]、阵列文法(array grammar)[7]。以上方法的优点是可以直接利用文本语言的文法理论、解析算法效率高，缺点是描述能力不强，只能描述有限的几类可视化语言。

近期发展的文法方法大致可划分为两种类型：图形文法(graph grammars)和属性多重集合文法(attributed multiset grammars)。下面将详细分析这两种方法的研究。

3.2 图形文法

1969年 Pfaltz 和 Rosenfeld[8]发表了题为“web grammars”的文章，揭开了图形文法的发展序幕，起初的图形文法主要用于处理图像，将一幅图像(image)用图(graph)来表示，其中带标记的结点表示原语对象，带标记的边表示几何关系。现在，图形文法已经广泛用于许多领域，如图形识别、图形生成、并行系统、数据库描述、可视化语言等。

图形文法为图形的局部转换提供了一种精确的数学模型，在给出图形文法的定义之前，有必要介绍几个基本概念，如图、同构、子图。

定义1 图形文法的基本数据结构——有向标记图 G 的形式定义为：

$$G=(NODES,\ EDGES,label_N,label_E,source,target) \quad (1)$$

其中：$NODES$ 和 $EDGES$，是结点和边的有穷非空集合；

$label_N$ 和 $label_E$，是结点和边的标识函数，用于从标识字母表中为每个结点和边分配标识符号；

$source$ 和 $target$，指定每个边 e 的源和目的。

在实际的图形化表示中，结点可以绘制为点、圆圈或矩形等，边可以绘制为从源到目的的箭头，标识是写在点和箭头旁边的字符串，如图1所示。通过带标识的边，可以描述在同一对结点之间存在多个边的情况(即 multigraph)，只需用不同的标识来区分它们即可。

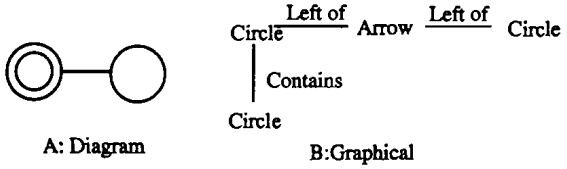

图1 edge-labelled 图形表示实例

定义2 两个有向标记图 $G_1(N_1,E_1,L_{N1},L_{E1},S_1,T_1)$，$G_2(N_2,E_2,L_{N2},L_{E2},S_2,T_2)$是同构的，记为：

$$G_1\cong G_2 \quad (2)$$

当且仅当存在一个双射函数 $h:G_1\to G_2$，$h=(h_N,h_E)$，$h_N:N_1\to N_2$，$h_E:E_1\to E_2$，且满足，

· $L_{N1}(n)=L_{N2}(h_N(n))$，$L_{E1}(e)=L_{E2}(h_E(e))$，对$\forall\ n\in N_1$，$\forall\ e\in E_1$

· $h_N(S_1(e))=S_2(h_E(e))$，$h_N(T_1(e))=T_2(h_E(e))$，对$\forall\ e\in E_1$

以自然语言描述，就是 G_1中的结点和边可以全部分别映射为 G_2的结点和边，同时保持源、目的、标识的分配关系。

定义3 若图 $G_1(N_1,E_1,L_{N1},L_{E1},S_1,T_1)$是图 $G_2(N_2,E_2,L_{N2},L_{E2},S_2,T_2)$的子图，则记为：

$$G_1\subseteq G_2 \quad (3)$$

当且仅当 $N_1\subseteq N_2$，$E_1\subseteq E_2$，且对$\forall\ n\in N_1$，$L_{N1}(n)=L_{N2}(n)$，对$\forall\ e\in E_1$，$S_1(e)=S_2(e)$，$T_1(e)=T_2(e)$。也就是说，G_1的结点集合和边集合分别是 G_2的结点集合、边集合的子集，并且两个图的源、目的、标识的分配关系一致。

在以上关于图的定义基础之上，图形文法可以用一个四元组来定义[9]，

$$GG=(\Sigma n,\Sigma t,S,P) \quad (4)$$

其中：Σn 是非终结结点符号集，Σt 是终结结点符号集，$\Sigma n\cup\Sigma t=\Sigma$ 符号表，$S\in\Sigma$ 为起始非终结符号，P 是一个有穷非空的产生式规则集，每个规则是一个三元组$(LHS,RHS,E)\in P$，其中：LHS、RHS，分别表示产生式规则的左图和右图；E，嵌入机制，描述图形 RHS 中的结点如何与主图中的其它部分连接。

图形文法本质上是一个图形重写系统，推导过程对应于图形的分解过程，因此，一个产生式的图形替换往往对应于一种图形操作。应用图形文法产生式进行图形替换的步骤是，如果一个图形 G(称为 host graph)包含了与产生式 P 的左图 LHS 同构的子图 ML(称为 redex)，则可将 ML 从 G 中移去，用产生式 P 的右图 RHS 替换 ML，然后运用嵌入机制 E 将图形 RHS 和图形 G 的剩余部分 G^-相连，从而将图形 G 重写为图形 G'，$G'=G\cup RHS/ML$，从起始图形开始、反复运用产生式最后生成的所有可能的图形就是由图形文法定义的语言，可记为，$L(GG)=\{G|S\stackrel{\cdot}{=}\!\!>G\}$。

3.3 属性多重集合文法

属性多重集合文法 AMG 是基于可视化语言符号多重集合的重写系统，符号具有属性，重写过程受到基于符号属性值的约束条件的控制，其产生式的右边虽然写成符号串的形式，但符号的顺序并不重要，因此实际上是无序集合，这点也是和文本语言文法的本质区别。AMG 可用一个六元组描述[10]，

$$AMG=(\Sigma n,\Sigma t,S,I,D,P) \quad (5)$$

其中：Σn 是非终结符号集，Σt 是终结符号集，$\Sigma n\cup\Sigma t=\Sigma$，$S\in\Sigma n$ 为初始符号，I 是属性标识符号集，D 是属性值域，三元组(Σ,I,D)表示属性字母表，

P 是产生式集合，每个产生式可用一个三元组(R,SF,C)定义，其中：R，形式为 $N\to M$ 的重写规则，N 表示非终结符，M 表示符号的多重集合；SF，语义函数集合；C，约束条件集合。

AMG 描述了由属性字母表生成的语言，字母表的每个符号 Z 都具备一系列属性$\{a_i|a_i\in attr(Z)\}$，语言的语法关系、语义关系都是通过符号的属性来确定，在应用产生式 p

时，也就是确定产生式中每个符号及其属性的匹配：

$$occur(p)=\bigcup_{Z_i\in\{N,M\}} occur(Z_i)$$

$$occur(Z)=\{Z.a_i \mid a_i\in attr(Z)\}$$

AMG 产生式将属性分为两种类型：定义属性和计算属性，其中，计算属性只限于产生式左边符号的属性。语义函数是定义属性到计算属性的映射，约束条件是定义属性到{True，False}的映射，约束条件隐含地表示了符号之间的几何位置关系，也是产生式应用的前提条件。一般，产生式格式如下：

$N\rightarrow\{M_1,\cdots,M_k\}$

$N.a_1=f_1(\langle$defined attributes$\rangle)$

…

$N.an=fn(\langle$defined attributes$\rangle)$

Where

*constraint*1(⟨defined attributes⟩)

…

constraintn(⟨defined attributes⟩)

由属性多重集合文法定义的语言就是由初始符号推导出的只包含终结符号的所有多重集合，记为 $L(AMG)=\{w \mid w\in\Sigma_t^*, S\stackrel{\cdot}{=}=>w\}$。

3.4 图形文法和属性多重集合文法的比较

图形文法以图论为数学分析手段，用标记图作为基本数据结构，而属性多重集合文法借助于集合理论，基本数据结构是带属性值的多重集合。下面，我们对这两种方法做一个对比分析。首先分析图形文法，用“图”的方式描述图形语言，直观、易于理解；不足之处是需要一个分析、识别终结符关系的阶段，以便用“边”的形式将其显示表示，并且分析算法比较复杂，尤其对于复杂图形而言，目前提出的大部分图形文法都属于上下文无关类型，不足以描述多种可视化语言，上下文相关图形文法虽然表现力强，但解析算法复杂度达到了指数级，实现困难。再来分析属性多重集合文法，通过对象属性的约束条件表示对象之间的关系，这种隐含方式虽不直观，但描述能力强，能够描述图形文法不能描述的语言，如递归组成的图形。缺点是其没有限制的计算也会导致分析复杂度增加，因此往往需要附加一些限制。

最后，我们看看两者对于图形嵌入问题的解决方法。由于图符之间的二维关系，必须确定被替换非终结符的周围环境和它的替换符号串之间的关系。*AMG* 的解决方法是隐式嵌入——通过产生式中的属性设定来创建同上下文的关系。缺点是不直观，用户不能立即看到属性赋值的结果，而且解析器也需花费大量的时间从属性和约束中提取隐含的关系信息。*GG* 有两种解决方法，一是提供上下文元素——就是在替换过程中保持不变的图形元素，用于建立替换图形和主图的关系。这种文法的可读性最好，但是无限制地运用上下文元素会导致图形重写规则非常复杂，而且无法描述有量词约束的关系，如完全图。二是提供独立的嵌入规则来定义替换图形的连接关系，即将被替换图形的射入/射出边转换成替换图形的射入/射出边。这种方法最易实现，只是嵌入规则难懂，并且无法在原先不相连的结点之间创建新关系。

4 相关成果分析

迄今为止，有许多学者基于文法形式体系，提出不同的可视化语言描述方法，其中一些已投入实际应用。表2对几个典型实例从数据模型、文法类型、嵌入方法、描述特点和描述能力几个方面进行了对比分析，它们分别是：LGG (Layered Graph Grammar)[11]、RGG (Reserved Graph Grammar)[12]、PLG (Picture Layout Grammar)[13]、CMG (Constraint Multiset Grammar)[14]、RG (Relational Grammar)[15]、PG (Positional Grammar)[16]、SRG (Symbol Relation Grammar)[17]。其中 LGG、RGG 是基于图形法的描述方法，PLG、CMG 是基于属性多重集合文法的描述方法，RG、SRG 是关系文法，SRG 实质上是图形文法的一种转化，而 RG 是属性多重集合文法的一种转化，只不过冠以关系文法的术语，PG 的早期版本是对文本语言文法进行直接拓展，由于描述能力极为有限，后来又进行了很大改进，表1是对改进后版本的分析。

通过以上成果的对比，可以看出，一种文法形式体系的描述能力越强，解析算法的有效性也就越差，因此，在提出一种文法形式体系时，都坚持这样的原则：有足够强的描述能力，同时支持有效解析。事实上，为了降低解析算法的复杂度，各类文法描述体系已经设置了一些限制，如 CMG 文法对负条件约束的支持、RGG 将简单的嵌入规则和少量的上下文元素相结合等等。

5 主要研究内容和挑战

由以上内容可知，可视化语言的多维特征使其形式描述不同于文本语言的传统方法，文[18]也曾讨论了可视化语言的研究内容，但那时并不明确其语法规范所应具备的性质。可视化语言作为一种语言，包括语素、语法、语义等基本要素，因此，其文法描述体系的研究内容也基于这三个要素，具体包括：

(1)*可视化语言的符号表* 可视化语言符号表中的元素，也就是语素，表现为一系列图形对象，语素的特征通过其属性来体现，一般分为三类属性：图形属性、语法属性和语义属性。图形属性反映了符号的图形特征，语法属性提供了形成语句的信息，语义属性则用于语句的逻辑解释和领域映射。为便于用户定制领域符号，符号表应提供相应的机制来支持用户利用基本图形对象，按一定的约束关系组成复合图形对象。

(2)*语法规范说明* 可视化语言的语法描述需要明确定义语素之间的空间关系，然后以此为基础用适当的文法描述各个语法单位。因此，可视化语言的语法规范包括两部分：具体语法和抽象语法。具体语法定义了图形对象的空间约束关系，是可视化语句图形布局的基础，抽象语法定义了对象之间的逻辑约束关系，为可视化语句的语义和所表达的含义提供基础。面临的挑战是这两部分的表示方法和同步更新，也就是所谓的 pretty print 问题。

(3)*语法分析* 可视化语句的语法分析结果一般是图形结构。由于图形符号之间不存在线性序关系，所以不可能直接应用下推自动机构造语法分析器，需要将二维图形结构转化成一维线性结构，再利用 LR 分析器，也可构造符号集合的偏序关系，以利用 Earely 算法、Cocke-Younger-Kasami 算法等，或以重写规则为驱动，进行组合、归约和回溯，直至分析成功或发现错误。再者，由于有许多可视化语言是上下文相关的，因此，构造分析器必须考虑解析算法的时间、空间复杂度。

(4)*语义的描述* 目前语义描述主要有三种方法：形式化、半形式化和非形式化。由于语义和领域知识的密切相关，语义的形式化描述非常困难，这也是目前许多学者极力要解决的问题。半形式化方式，如 OCL、约束图形在可视化建模语

言中已经得到应用，非形式化方式的说明语义(declarative semantics)也已应用。

表1 几种可视化语言文法形式体系的比较

类型	出处	嵌入方法	特点	描述能力
LGG	Rekers/Schürr 荷兰Leiden大学	上下文图形	● 抽象关系、空间关系分别使用两套图形文法来描述，并以图形方式显式表示 ● 分析算法复杂度高，达到指数级	● 可以描述多种可视化语言，如线形、树形、连通图形的语言 ● 引入上下文的重写规则，可读性好，但无法描述完全连通图之类的含有定量约束关系的语言
RGG	Da-Qian Zhang /Kang Zhang 澳大利亚 Macquarie 大学	嵌入规则+上下文元素	● 基于 LGG，进行改进：如引入带记号的结点标识符和禁止使用通配符 ● 简化了语法规则，提高描述能力 ● 分析算法复杂度降到多项式级	● 可以定义各种连通图形的可视化语言 ● 不支持产生式应用的负条件 NAC，无法描述包含多个孤立结点的图形
SRG	F.Ferrucci et al. 意大利Salerno大学	嵌入规则：上下文无关的关系重写规则	● 有两个相互依赖的重写系统：符号重写系统、关系重写系统 ● 关系是纯粹符号性质的、没有任何计算意义	描述能力很强，可以定义任何上下文相关语言
PLG	Golin/Reiss 美国Brown大学	隐式嵌入+remote符号	● 空间关系预定义为组成操作运算符、有限，扩展性差 ● 产生式右边至多包括两个符号，导致非终结符和语法规则数量增加 ● 引入"remote"符号指派生树中任意一个终结符，功能同上下文。	能够描述有向无环图结构的可视化语言，但文法可读性较差
CMG	Marriott/Helm 澳大利亚 Monash 大学	隐式嵌入＋上下文元素	● 空间关系、抽象关系没有显式表示，通过对符号属性的约束来定义 ● 存在定量符号使其具备上下文相关文法的描述能力 ● 产生式应用的负条件使其具有确定的自底向上分析能力	同 PLC 相比，可以定义、解析更多类型的可视化语言
RG	Wittenburg/Weitzman 美国麻省理工大学、贝尔实验室	嵌入规则	● 基于符号和关系的单集的重写系统 ● 关系是基于对符号或对符号属性的约束来定义 ● 通过非终结符属性推导来解决嵌入问题	处理能够进行逐层分解的可视化语言，如数学表达式、流程图等各类结构化图形语言，无法描述通用图形语言，如自动机、过程图等上下文相关语言
PG	Costagliola et al. 意大利Salerno大学	嵌入规则	● 用图形对象、组成关系的线性结构表示图形 ● 多维关系通过二维关系来描述 ● 根据符号的位置属性、可联接区域，定义空间组成关系	可以描述结构化、半结构化图形以及部分通用图形，如数据流图、petri 网

结束语 目前，可视化语言已经成为人机交互中的一个关键内容，而且，随着 World Wide Web 中可视化技术的应用，可视化语言对计算机科学领域的影响日益加深，国际上对可视化语言的研究已达到一定的深度和广度，但国内方面有关可视化语言的理论研究还比较少。本文主要分析了基于文法的可视化语言形式化描述方法的研究，其实可视化语言理论还包括许多其它内容，如分类标准、解析算法等，此外还需有可视化支撑环境来支持可视化语言的实现，因此，要真正推动可视化语言的发展，还有许多工作要做。

参考文献

1 Kranzlmuller D. Visualizing Program Behavior with the Event Graph. In: Zhang Kang, ed. Software Visualization: from theory to practice. Boston: Kluwer Academic Publishers, 2003. 29～57

2 Burnett, Margaret. Visual Programming. (In) John G. Webster. Encyclopedia of Electrical and Electronics Engineering. New York: John WIley & Sons Inc, 1999

3 Marriott K, Meyer B, Witterburg K. A survey of visual language specification and recognition. In: Marriott K, Meyer B, eds. Visual Language Theory. Berlin: Springe, 1998. 5～85

4 Stiny G. Introduction to shap and shape grammars. Environment and Planning, 1980, 7: 343～351

5 Fu K.-S. Tree systems for syntactic pattern recognition. IEEE Transactions on Computing, 1973, 22: 1087～1099

6 Costagliola G, Chang S-K. DR PARSERS: A generalization of LR parsers. (In) IEEE symposium on Visual languages. IEEE Computer Society Press, 1990. 174～180

7 Rosenfeld A. Picture Languages. Academic Press, 1979

8 Pfaltz J, Rosenfeld A. Web Grammars. In: Proc. First Intl. Joint Conf. on Artificial Intelligence. Washington, 1969. 609～619

9 Rozenberg G. Handbook on graph grammars and computing by graph transformation: Foundations, World Scientific, 1997, 1

10 Golin E J. A method for the specification and parsing of visual languages: [PhD thesis]. Dept. of computer science, Brown University, 1991

11 Rekers J, Schürr A. A graph grammar approach to graphical parsing. In: IEEE Symposium on Visual Languages. IEEE Press, 1995. 195～202

12 Zhang D-Q, Zhang K. Reserved graph grammar: a specification tool for diagrammatic VPLs. In: Proc. 1997 IEEE Symp. on visual languages. Capri, Italy: IEEE Press, 1997. 284～291

13 Golin E J, Reiss S P. The specification of visual language syntax. IEEE symposium on visual languages. IEEE Computer Society Press, 1989. 105～110

14 Marriott K. Constraint multiset grammars. IEEE symposium on visual languages. IEEE Press, 1994. 118～125

15 Wittenburg K. Relational Grammars: Theory and Practice in a Visual Language Interface for Process Modeling. In: 96 International Workshop on Theory of Visual Languages. Gubbio, Italy: IEEE Press, 1996

16 Costagliola G, Lucia A D, et al. Positional grammars: a formalism for LR-like parsing of visual language. In: Marriott K, Meyer B, eds. Visual Language Theory. Berlin: Springer, 1998. 171～191

17 Ferrucci F, Tortora G, et al. Relation Grammars: a formalism for syntactic and semantic analysis of visual languages. In: Marriott K, Meyer B, eds. Visual Language Theory. Berlin: Springer, 1998. 219～243

18 强军，王兵山，李舟军. 可视语言：把思想变成程序的工具. 计算机科学，1994，21(4)：6～10