

路径修饰学 Rhetorics of Path

吴玉军
Wu Yujun

(广西计算中心广西软件新技术实验室 南宁市星湖路 32 号 530022)
(Guangxi New Software Technology Lab., Guangxi Computing Center,
32 Xinghu Road, Nanning, Guangxi, 530022)

摘要 介绍一种所谓的“路径修饰学”^[1],从符号学的角度来阐述超文本应用模型的原理,并提供了一些设想。

关键词 超文本模型 创作系统 符号学

Abstract Introduces somewhat “rhetorics of paths”, It expresses the principle of design model of hypertext in semiotic views, and give some ideas.

Key words hypertext model, authoring system, semiotic

1 超文本技术与符号学

超文本技术是人们在处理庞杂的计算机信息数据过程中发展壮大起来的。超文本的基本特征是链—节点方式。阅读路径就是一组有序的节点—链—节点—链……。所谓的“有序”,指的是阅读路径有它的内在逻辑关系,在这一点上,它与符号学和语言学方面有一定的相通之处。

2 例子·说明

为了比较清楚地说明超文本技术与符号学方面的相通之处,我们举个实例。要建立一个小的超文本应用系统,目的是介绍“几何透视”这个概念。一般方法是,先为读者提供理论材料,然后让读者选择是否接着看,这些描述“几何透视”的一些实例。这样,已经明白的读者就可以跳过这些实例,而有兴趣的读者则可以通过标有“图形例子”的按钮(button),直接进入有透视类型实例的卡片。

这里面存在一个问题。假如说,例子不止一个,那么作者应该如何组织例子和说明,例子和例子间的相互关系呢?

3 序列·语段

顺序在超文本应用中占据着非常重要的地位。对

于教学软件来说,说明往往放在例子的前面,而例子和例子之间可以直接互访,读者在看完第一个例子后,他可以要求直接看第二个例子,也可以要求直接返回说明,然后再进行下一步的选择。

这里,我们引用“序列结构”的概念。在前面提到的例子中,读者需要的阅读结构是(说明+(例子+例子)),其中并行的例子组成一个子成份,它赋予读者更大的灵活性。它的反面结构是((说明+例子)+(说明+例子)),在这个结构中,例子之间要靠说明来连接,这对读者来说是很不方便的。

“序列”是语言学的一个基本概念,序列模式中有一个概念叫“语段”。文本就是由不同层次的语段构成。结构是阅读的前提,所以在超文本系统中,选择适当的语段是成功的关键。如果要让超文本阅读与平常的书本阅读一样,那么仅仅让读者通过链从一张卡片到另一张卡片中去是远远不行的。

直观地,“超文本路径”可以对应“语言序列”,它可以这么表示:

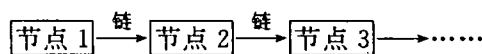


图1 超文本路径

Fig.1 Hypertext path

其中的链我们称之为“序列链”。它代表着从源节点指向目标节点这么一种前后关系。

同时,在每个节点里(对于超文本应用系统来说,就是一页文字卡片),单词间是相邻的,它们一个挨一个。这种序列关系是“并发关系”,它们同时出现在读者的面前,单词间的链我们称之为“并发链”。

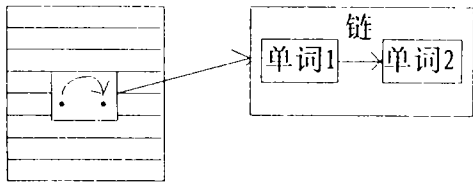


图2 超文本卡片
Fig. 2 Hypertext card

进一步的说明如下:

- 序列链 连接的信号或信号成分,在不同时间里依次发生。序列链的信号在屏幕上交替出现。

- 并发链 连接的信号或信号成分,在同一时间里同时发生。并发链的信号在屏幕上同时出现。

有了这两种链后,我们就可以定义相应的语段:序列语段和并发语段。在超文本中,序列语段描述的是单张卡片的表现形式(或者是屏幕上成组卡片的表现形式)。

这样,我们现在可以用这几个新词汇来重新描述前面讨论过的东西。

在介绍“几何透视”的超文本应用系统中,我们该选择什么样的序列语段呢,是(说明+例子)语段,亦或是(例子+说明)语段?在显示“例子卡片”和“说明卡片”时,我们该选择哪一种语段,究竟是用并发语段好呢(例子显示在另外一张子序列卡片上),还是用序列语段更简洁(例子和说明共用同一张卡片)?因此,通过这些语段我们从此可以用一个新的角度来分析超文本应用的结构。在超文本中,结构是最重要的,而卡片仅仅是用来填充结构的材料。我们现在所讨论的超文本应用基本上是一种叙事结构,如何保证这种叙事结构既合理又充满效率呢?

形式上,语段可以描述成一序列的槽。这些槽根据一定的规则进行填充。规则包括:

- 什么样的材料才可以用来做填充物
- 哪个槽必须被填充,而哪个槽又可以是空的一般,我们需要三种主要语段:群集(constellations),子关系(subordination),和相互依存(interdependences)。

- 群集语段,槽可以填充或者清空
- 子关系语段,关键槽(head)必须被填充,同时,修饰槽(modifier)可以填充或不填充(可选按钮表示的就是子关系)

- 相互依存语段,所有的槽必须被填充

这三种语段在逻辑学上各自有对应的关系。群集对应拆散,子关系对应包含,相互依存对应相等。但它们又各有不同的语义:

- 群集,每个独立部分都可以象整体结构那样,演示相同的语义。如,“我看见一幢房子和一辆轿车”中,“房子”和“轿车”都归属于“看”,所以可以这么说“我看见一幢房子”或“我看见一辆轿车”。

- 子关系,只有关键项(head)和文本的其他部分发生直接关系,而修饰项(modifier)并不直接和子关系之外的文本相连。如,“我看见一棵大树”中,“大”仅仅属于“树”,它既不属于“我”也不属于“看见”。

- 相互依存,整体的效果不能分解成其中某一部分的效果。如,“我说他来了”中,“他来了”做为一个整体,属于“说”,仅仅说“我说他”或“我说来了”都没有原来的意义。

群集表达可能的关系,如“我看见一幢房子,一辆轿车,一列火车,……”。与此相反,相互依存表达必要的关系,“他来了”必须包含两个元素,而“他给我一本书”需要三个元素,我们不能象群集那样随机地添加元素。

子关系中,只有关键项在生成文本的叙述结构中是活动的。例如:

<描述性句子>向劫犯射击的警察,是一个年轻黑人。

<叙述性句子>一个年轻的黑人警察,向劫犯射击。

所以,在超文本应用中,应该保证,作为形式关键项的元素在语义上也必须是关键项,否则容易引起读者的迷茫。

4 超文本的叙述模型

对于前面的“几何透视”应用,可作如下分析。

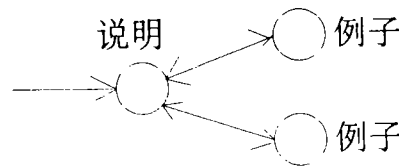


图3 错误·节点/链表示(箭=链,圆=节点)

Fig. 3 Error. Node/link representation.
(arrow=link, circle=node)

合理的结构应该在例子之间加上链。

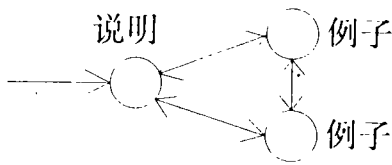


图 4 正确·节点/链表示

Fig. 4 Correction. Node/link representation

如果用语段表示来代替传统的节点/链表示,则:

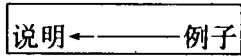


图 5 错误·语段表示 (方框=语段, 单箭头=修饰项指向关键项)

Fig. 5 Error. syntagmatic representation (Boxes= Syntagma. single arrow=points from modifier to head)

图 5 里因为仅仅有一个例子槽,所以例子之间不能生成一个序列语段,这对读者来说显然是不方便的。如果我们选用较自然的阅读结构(说明+(例子+例子))来替换“非自然”的((说明+例子)+(说明+例子))结构,那么最后的槽应该可以包含一群元素,而不仅仅是单个元素。

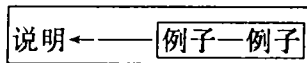


图 6 正确·语段表示 (无方向箭头=连接群集的槽)

Fig. 6 Correction. Syntagmatic representation (non-directed arrow connects slots in constellation)

从构造传统文本阅读的词素模式、句法模式、修辞模式和叙述模式的过程中,我们可以相应地构造超文本阅读进程的语段。比如,我们可以构造一个序列相互依存语段,这样当读者读了当前卡片后将马上转向另一张卡片;我们还应该将修饰项放在关键项之前,就好象从句放在主句之前一样。

5 创作工具

作者不应该直接和链打交道。超文本的创作工具应该赋予作者随意定义所需模式的功能。

例如“比较”是图形分析的一个重要技术,如果你想知道图形中的某一部分在总体信息中的作用,那么你可以改变这部分,或者干脆移掉它,以期观察它的效果。用语段表示如图 7。图 7 中,圆角方框表示卡片,一般方框表示语段,箭头表示句法关系。单箭

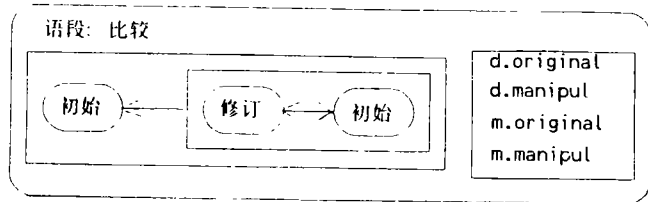


图 7 定义“比较模式”的创作卡片

Fig. 7 Authoring card defining the comparison pattern

头表示子关系语段,双箭头表示相互依存语段,水平箭头表示序列语段。这张卡片定义了一个子语段,它的修饰槽包含一个修订图形和初始图形的相互依存。这样,当显示初始图形显示时,就产生一条指向相互依存的链,如果读者沿着这条链前进,那么就显示几秒钟修订图形,然后由于序列相互依存语段的原因,又重新显示初始图形。

因为存在初始图形和修订图形这两个元素,所以在传递它们之间的差别信息时就必须要在语义上激活相互依存语段。

创作卡片还包含一个当前可用卡片的清单(显示在图 7 的右边)。当定义了句法模式后,就可以生成它的一个程序,然后就可以将清单中的填充卡片拖拉到三个槽之一去。

编程方式设想为面向约束的,一般规则是:根据约束条件可以打开的任意卡片,用户都可以经由链来存取。约束是在创作卡片中定义的,链则显示在当前卡片上。链是自动生成的。

因此,如果相对于“说明”来说,它的“例子”是一个序列修饰项,只有在显示“说明”之后才能存取“例子”。如果“例子”是一个并发修饰项,那么,只有在显示“说明”期间,才能存取这个“例子”。

6 连贯性

超文本需要的一个基本特性就是某种连贯性。

文本的连贯性常常通过“主题-引子”(theme-rheme)结构生成。首先,句子采用某个元素作为“引子”。例如,童话故事的典型开头就是这样:

“很久很久以前,有一位公主。”

其中,“公主”这个元素,被用这种方法说明过后,它的状态就是“已知项”,就可以作为一个“主题”来引用。我们可以使用这个“已知项”或代词,并将它放在句子的初始位置,用它来介绍一个新的“引子”。例如:

“她长得很美,但是脾气却非常坏。”

通过这种方法,文本就可以生成一系列已知的和新的元素。即,引子将转换成主题,而这个主题又将成为介绍新引子的基础。

主题-引子结构在任何文本或电影中都是必不可少的,但是,在超文本中必须解决的问题是,如何标识主题和引子。因为在某张卡片中引用了某项元素,读者或许并没有阅读过那张卡片,就好像读者一下子就读到“她长得很美,但是脾气却非常坏”,而没有读过“很久很久以前,有一位公主”那样。所以,在创作过程中,作者必须对所有的媒体信息(卡片、文本、图形等)做上标记,并附加普遍约束:只有当某个卡片元素被作为引子介绍过后,读者才能存取那些用这个卡片元素作为主题的卡片。

7 一般组织

前面讨论的叙述结构相对较小,下面我们介绍一个相对较大的。

有一个学生系统,包括历史介绍(Introduction)、说明(Explanations)和结论(Conclusion),在语义上组成一个序列子关系,E修饰I,C修饰E。

在叙事结构中,应该运用约束这个观念。我们给根据语义生成的约束分配不同的强度,让读者选择一个阈值,只有这个阈值之上的约束才是合法的。

假设子关系的强度为100,而群集的强度为50。

如果读者希望系统象一本书,阈值就要高些,这样合法的约束就较少,作者所起的作用就很大。例如,如果选择75作阈值,那么将只存在从I到E和从E到C的链。如果读者希望系统象一本词典,那么阈值就要低些。因此很多卡片都将可以存取,读者可以随意地阅读。这样,如果我们选择25作为阈值,那么系统将会在所有的三部分间生成链。

8 转换语段·视点

我们还可以从“电影语段”中吸取经验。比如蒙太奇,这是一种“转换语段”,它变换地提供两个以上的一系列事件。用这种方法,每个系列中的时间关系都是连贯的,而在系列之间,这种时间关系就是并发的了。在超文本中,“转换语段”可以用四个槽来定义,每个系列用两个槽。如图8:

其中,系列1槽和系列2槽是序列相互依存关系,当系列1槽打开时,系列2槽也将打开,并在一定时间后替换系列1,而每个槽同时又是修饰槽的说明对象,修饰槽显示的是同一系列中的下一个事件。当转换语义中的系列1被打开时,它在系列2的对应

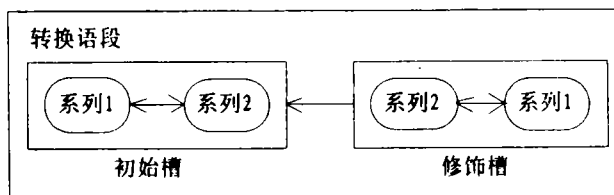


图8 转换语段

Fig. 8 Alternate syntagma

者将会取代它,并显示可以用来打开系列2中的后继项的按钮,如果这条链确实存在,那么相应卡片被打开,它将在屏幕上保留一段时间,接着它在系列1中的对应项也将打开。这种模式不断重复下去。当然,序列相互依存关系也可以是并发相互依存关系,此时,屏幕上将同时出现两张卡片,每张卡片都描述这个并发事件中的相应系列。

“转换语段”中的两个系列是叙述并发的不同事件。而如果从两个不同的人的视角来看待同一个事件时,我们称之为“视点语段”。在超文本应用中,对于同一个事件,我们可以对其中一个系列用文字,而对另一个系列用图片或者动画。此时,各种媒体信息表达的是同样的内容,但却运用了丰富的表达形式,以期表达效果更加全面、深刻。

超文本技术还可以从符号学和电影学中借鉴更多的内容,这期待着相关软硬件技术的进一步发展。有了相似的模型,还需要在实践中不断进行补充和完善,超文本技术毕竟有着自己不同的特点。

超文本系统面临的巨大挑战,事实上就是存取信息方面的挑战。如何配置合理的信息结构,如何制订高效的检索机制,如何在尽量降低信息冗余的前提下发挥超文本迷人的非线性特征,如何让超文本应用系统始终走在用户思维发展的前面等,所有的这些问题,不仅是超文本设计模型所必须解决的,而且也是作者们在创作时必须要注意的。

超文本技术,将人们强大而灵活的思维处理能力,同计算机高效、迅速、准确的信息存取能力和惊人的信息储备能力有机地结合在一起。超文本放射出的迷人魅力将会令全社会各阶层的人都受益。信息时代需要有效的信息处理技术,超文本技术作为这一领域的代表,起到极其重要的作用。

参考文献

- 1 Anderson P B. Towards an aesthetics of hypertext systems, A semiotic approach. Hypertext, 1990: 224~237.