维普资讯 http://www.cqvip.com

转换成一个反应流，请求和反应是一一对应的。

一个对话是两个部分A，B之间的交互。A、B轮流向对方发出信号，每个部分都有一个包含交互历史信息的状态，A、B的状态分别叫α和β。每个部分都有自己的转换函数，其类型为：

f：：(state,input)→(state,output)

f从它前面的状态以及从另一部分接收的最后一个信号中计算新的状态和要向另一部分发送的信号。设A、B的转换函数分别为f_a、f_b，设A最初的状态为$\alpha_j, j\geq 0$，当建立B时，A发出最初的信号a_j给B，B的转换函数f_b就从最初状态β_0和从A接收来的信号a_l中计算其新状态β_1和要发送给A的信号b_0，这样就开始了A、B之间的对话。因此，转换函数f_a、f_b，初始状态α_j、β_0对每一部分分别定义了状态的序列和输出的序列。以下四个序列定义了对话的全部历史：

α：$(\alpha_0\ \alpha_1\ \alpha_2\cdots)$　a：$(a_0\ a_1\ a_2\cdots)$

β：$(\beta_0\ \beta_1\ \beta_2\cdots)$　b：$(b_0\ b_1\ b_2\cdots)$

由A初始化的A和B之间的对话是形式为(α a β b)的四元组，A和B之间的对话如下：

A	B
(initial)α_j	β_0(initial)
(send to B)$a_j\Rightarrow$	
	$(\beta_1, b_0)=f_b(\beta_0, a_j)$
	$\Leftarrow b_0$(send to A)
$(\alpha_{j+1}, a_{j+1})=f_a(\alpha_j, b_0)$	
(send to B)$a_{j+1}\Rightarrow$	
...	
$(\alpha_{j+k}, a_{j+k})=$ $f_a(\alpha_{j+k-1}, b_{k-1})$	
(send to B)$a_{j+k}\Rightarrow$	
	$(\beta_{k+1}, b_k)=f_b(\beta_k, a_{j+k})$
	$\Leftarrow b_k$(send to A)
...	

假设我们的对话的类型如下：

type Dialogue=STR Response Request

data Request=Get |Put Char

data Response=Got Char |Did

. Get：k：从键盘上输入字符n，然后执行k (Got,n)。

. (Put n)：k：将字符输出到打印机上，然后执行K Did。

下面就是用对话I/O写的echo程序：

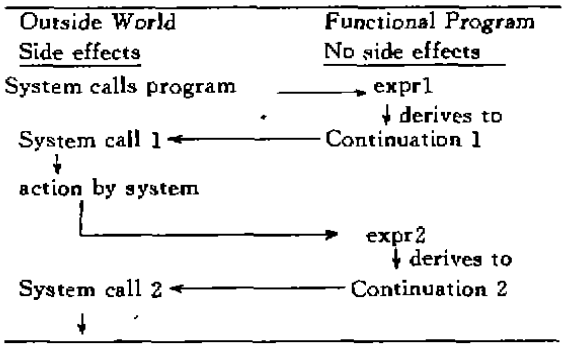
```
echo : : Dialogue
echo reps=Get : (\a-)
    if (a==eof)
    then[]
    else (Put a) : (echo (drop 2 reps))
    where Got a=reaps!! 1
```

\a→K是Haskell语言的记号，它表示一个λ表达式，即λa.k。

1.2 CPS I/O

后续是从中间值(中间值可能为空)到程序剩余部分的函数。因此，我们在定义函数的时候，不是直接返回函数的结果，而是将后续作为变元传递过去，作用在结果上。这样的程序设计风格常被叫做后续传递风格(Continuation passing style)或简称CPS。

假设中间值的类型为a，程序的剩余部分的类型为Answer，后续的类型则为a→Answer，那么怎样用CPS来处理I/O问题？让我们来看下图：



给定一个基于CPS的函数式I/O程序，假设程序从expr1开始运行，当程序希望执行一个I/O操作时，它发出一个信号给操作系统中断这个程序，程序剩余部分则保留下来。但是程序剩余部分的重新执行可能要依赖于这次I/O操作的结果，即上面提到过的中间值，中间值与程序剩余部分的映射则成为后续，系统执行完这次I/O操作之后，唤醒程序剩余部分，将后续作用到I/O的结果上，便得到一个新的程序，新程序从expr2处开始执行，重复上述过程，直至整个程序结束。

在CPS I/O中，可执行的类型是这样的代数类型，对每一种可表示的I/O行为有相应的类型构造子：

data CPS = INPUT (Char→CPS) | OUTPUT Char CPS |Done

CPS程序的意义能够容易地给出：

. INPUT k：从键盘输入n，然后执行k n。

. OUTPUT n p：将字符n输出到打印机上，然后执行p。

. Done：立即终止。

下面我们来看一个使用CPS I/O的程序：

```
echo : : CPS→CPS
echo c=INPUT(\a→)
    if (a==eof)
    then c
```

维普资讯 http://www.cqvip.com

else OUPTUT a(echo c))

由于在 echo 完成之后我们可能想做更多的 I/O，所以我们必须为 echo 提供一个后续 c，来表示当 echo 完成时还要做什么。

2 基于环境的 I/O 模式

顾名思义，基于环境的 I/O 模式是在一个特定的对象即环境(environment)上来直接进行 I/O 操作的、环境表示世界的状态。在基于环境的 I/O 模式中，对环境的处理分为隐式、显式两种方法；前者主要有 Monad I/O 方法，后者主要有 Clean I/O 方法。

2.1 Monad I/O

Monad I/O 方法是基于隐式环境的方法，环境构成了机器的状态，在环境上的操作有一个特定的类型 IO a，环境仅能出现在 IO 类型里。IO a 表示这样的动作：执行某些 I/O 操作，返回类型为 a 的值。IO Monad 实际上是一个抽象数据类型，因此称 Monad I/O 中的环境为隐式环境。例如，给定两个 I/O 操作：

```
getcIO : : IO char
putcIO : : char→IO()
```

getcIO 从标准输入设备上读取一个字符，putcIO 将一个字符写到标准输出上，getcIO 返回一个类型为字符的值，而 putcIO 则不返回任何值。这两个 I/O 操作都会导致环境发生变化，但是由于 IO Monad 是一个抽象数据类型，环境仅能出现在 IO 类型中，所以环境的任何变化对用户都是不可见的。

I/O 操作是怎样执行的呢？答案就是提供一个顶层的标识符 mainIO，其意义在于将整个程序的值看作是一个单一的 I/O 动作，叫 mainIO，通过执行这个动作来执行程序。譬如：

```
mainIO : : IO()
mainIO=putcIO"!"
```

以上定义的函数在屏幕上打印惊叹号"!"，这只是一个简单的 I/O 动作，怎样将一些简单 I/O 动作组合成一个比较大的 I/O 动作呢？这里我们引入两个组合子：

```
unitIO : : a→IO a
bindIO : : IO a→(a→IO b)→IO b
```

unitIO x 表示仅返回 x 的 I/O 动作。'bindIO' 是 Haskell 语言的记号，表示 bindIO 为中缀操作。m 'bindIO' k，其中 m : : IO a，k : : a→IO b，其意义为：先执行 m，它产生类型为 a 的值 x，再执行 k x，它产生类型为 b 的值 y。(IO，unitIO，bindIO)构成了一个 Monad。

下面再来看另外两个组合子：

```
doneIO : : IO()
seqIO : : IO a→IO b→IO b
```

done IO 什么 I/O 动作也不做，m 'seqIO' k，其中 m : : IO a，k : : IO b，首先执行 m，然后再执行 k，返回由 k 产生的值，doneIO、seqIO 可以由 unitIO、bindIO 定义出来：

```
doneIO=unitIO()
m 'seqIO' k=m 'bindIO' \a→k
```

下面我们来看如何用 doneIO、seqIO 定义 1.1.2，1.2 节的 echo 程序的：

```
echo : : IO()
echo=getcIO 'bindIO' \a→
    if(a = = =eof)then
        doneIO
    else putcIO a 'seqIO' echo
```

2.2 Clean I/O

Clean I/O[5]是基于显式环境的 I/O 方法。一个处理 I/O 的程序是这样的函数：给定初始环境，产生一个新的环境，在这个新环境里所有相继发生的变化都是由这个 I/O 程序引起的。程序能改变世界的状态，并通过对环境有访存权的函数获取有关信息，这样的函数由两个动作组成：立即改变世界的状态；当改变世界的状态反馈回来时，生成环境的新实例。每一个新 I/O 操作作用到前面操作所产生的环境结果上，我们来看下面的例子：

```
echo : : * World→ * World
echo world
  |c=eof=world2
    =echo world2
      where
      (c,world1)= : getchar world
      world2= : putchar c world1
```

* World 为 Clean I/O 系统中环境的类型，函数 echo 是环境 World 上的递归函数，它仅通过预先定义的函数 getchar 从环境中取回一个字符，使用预先定义的函数 putchar 将这个字符打印在屏幕上，如果取回一个"eof"，则 echo 的递归调用结束，getchar 的类型为 * World→(a，* World)，putchar 的类型为 * World→ * World。

显式环境传递的 I/O 方法可以从环境变量 world 的变化中看出来，程序的初始环境为 world，经过对环境有访存权的函数 getchar 作用后，环境变为 world1，并将 world1 传递给另一个对环境有访存权的函数 putchar，putchar 作用以后环境成为 world2，环境这一系列变化都是 I/O 操作的结果。

维普资讯 http://www.cqvip.com

参考文献

[1] P. Henderson, Purely functional operating systems, In J. Darlington et al. eds., Functional Programming and its Applications, Cambridge University Press, 1982

[2] JT O'Donnell, Dialogue: a basis for constructing programming envirnoments, In Proc ACM Symposium on Language Issues in Programming Environments, Seattle, ACM, 1985

[3] E Moggi, Notions of computation and monads, Information and Computation, 93, 1991

[4] SL Peyton Jones & PL Wadler, Imperative functional programming, In Proc. 20th ACM Symp. on Principles of Programming Languages. Charleston, 1993

[5] PM Achten & MJ Plasmeijer, The ins and outs of Clean I/O, J. of Functional Programming 5(1), 1995

[6] 袁华强，函数式 I/O 系统的研究与实现：一种 Monad 方法[博士学位论文]. 上海交通大学, 1996

（上接第 5 页）

D. Moon[20]提出了用散列表技术的查找结构，并将其应用于 Flavor 系统中。在散列表中要预先填充所有可能的分派情况，而在多方法环境下，各种分派情况的总数是组合爆炸的（复杂度为 $O(e^n)$[21]）。

G. Kicales[6]也提出了类似的方法应用于 CLOS 系统中。作为改进，P. H. Dussud[21]提出使用动态缓冲技术，动态缓冲中只保存被调用的方法。这种技术是非常量时间的，也没有解决组合爆炸的问题。

C. Lecluse 等[22]提出了结构子类化（sturctural subtyping）的概念。他们要求定义辅助的方法以满足仅仅通过子类型关系就可确定 MSA 方法。

W. G. Mugridge 等[23]定义参数序列 $(t_1, \cdots, t_n)$ 的覆盖（cover）为集合 $\{(S_1, \cdots, S_n) \mid S_1 \leqslant t_1, \cdots, S_n \leqslant t_n\}$，一个方法的可应用性定义为方法参数的覆盖与实际调用参数的覆盖的交集（可以认为集合越小的方法越具体）。对于一个类属函数调用，可通过比较相应覆盖的交集来确定 MSA 方法。这种解决方式应用于 Kea 中，当两个方法无法确定优先顺序时就声明调用是二义的。

总的说来，目前的研究工作都着重于提高各自算法所得到的目标代码的时间和空间性能。这些算法各有优劣之处，其中一部分已经应用于实际的多方法 OO 语言中。

我们认为，未来在多方法分派研究中的工作主要有几个方面：①对目前算法的改进和完善；②提出新的综合性能更佳的算法；③通过理论分析和实践检验，综合评价各种算法；④对算法进行综合。

对算法进行综合也有不同的方法。一方面，我们可以在编译系统中设置编译开关，由程序人员根据实际情况在时间与空间性能之间进行权衡和选择。另一方面，也是更重要的一方面，我们可以让编译系统检查分析程序的结构，自动选择最佳算法，甚至组合各种算法。

由于多方法语言强大功能所具有的吸引力，这方面的研究工作将会进一步深入下去。尽管存在着相当大的困难，我们仍然可以期望，在不久的将来，多方法语言将获得普遍的接受和应用。（参考文献共 22 篇略）

维普资讯 http://www.cqvip.com

计算机科学 1996 Vol. 23 №. 6

计算机网络的一种描述模型：一致网*)

王千祥 周兴社 康继昌

（西北工业大学计算机科学与工程系 西安 710072）

摘 要 This paper presents a model that can describe the character of computer network: Uniform Net (UN). Compared with the existed model, UN has benefits of intuitive, simple, and powerfull. With UN model, several typical systems such as CSCW are analyzed.

关键词 Computer network, Concurrent, UN, CSCW.

一、引 言

在计算机发展的初期，经常采用通用小型机作为巨型、大型机的组成部分，负责复杂的外设管理，形成了功能不同，但各部分之间可互相协同工作的体系。流水处理结构的出现则进一步从概念上推进了这一体系的发展，成为异构型处理网络的雏形。尔后，个人机的出现使计算机从单纯的数值计算领域走向了信息处理的广阔天地，更重要的是，个人机在地理上是高度分散的，而信息则只有在交流、综合的环境中才会产生，这两方面的矛盾只有通过共享资源才能解决。这种强劲的共享需求，决定性地推进了计算机向计算机网络的转变。

计算机网络的出现极大地丰富了计算机的研究内容。尽管从底层看，计算机之间只不过是在进行简单的消息传递操作，但这足以使计算机从孤立、封闭的状态向交流、开放的模式转变，相互连通的计算机之间可以共享各种资源。今天，尽管我们在网络上已经做了如此之多，但实际上，我们对计算机网络处理的本质的了解却并不十分深刻，诸如并发、不确定、多复本、延迟、不可靠等许多恼人的问题反复出现在不同的研究方向中，究其原因，在于我们至今对计算机网络处理过程的认识不充分。更进一步说，我们需要建立一种对应模型，它抽象而且直观地反映计算机网络处理过程的本质，借助于该模型，我们不仅可以很好地解释已有网络上各个方向的处理过程，还可以指导未来的研究与开发。

二、一致网模型

通常，我们总是将计算机看成是一种模拟演算工具，那么，从这种观点出发，计算机网络又是一种什么样的工具呢？计算机网络是信息社会的基础，是一种模拟信息处理工具。一致网（Uniform Net，或UniNet，或UN）就是从计算机网络出发，对计算机网络上多实体之间相互作用进行抽象得到的表达模型。我们希望UN能够象图灵机（TM）直观、简洁地描述了传统的计算机处理过程一样描述计算机网络上的处理过程，模拟信息社会中多实体协调地进行信息处理的相互作用。

2.1 UN的基本概念

不失一般性，我们可以想象计算机网络上存在一群实体，统称它们为对象，对象具有自治的行为、状态，并且相互之间可以传递消息，从而建立起交互作用关系。一个对象可以向一个或多个对象发送一条消息，一条消息可引起接收对象的特定行为，并可能改变其部分状态。UN是一个逻辑网，不直接与具体的物理计算机网络相对应。一个计算机网络上可存在多个UN，而一个UN也可以建立在几个相通的计算机网络之上。UN由三个核心概念组成：空间、行为、映象。

空间是指对象空间，更具体地讲是分布、共享的对象空间[7]，是一个规模可伸缩的对象群，对象由结构化的数据以及定义在其上的操作构成，分别反映对象的状态与变化规律，由对象群构成的空间是对现实世界较为准确的反映。对象可具有自己的特殊行为，这些行为不是由外部消息引起的，而是某些意义下的状态变化的要求，这种状态的变化可以部分地被外部对象所"预测"。

行为是指个体的行为，更具体地讲是在分布前提下，对空间状态或映象有影响的并发个体行为[8]。

*)本文得到国家自然科学基金资助，项目编号：69573023

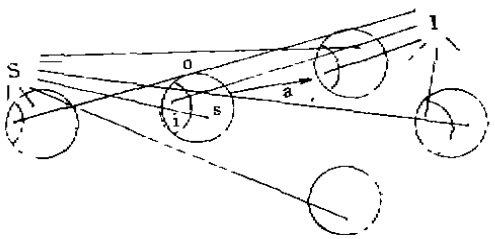
维普资讯 http://www.cqvip.com

大量的行为之间无因果关系，两个行为发出的先后次序是不确定的。然而，从总体上或从"事后"效果看，如果某段时间内空间中有许多行为发生，那么这些行为的集合一定可以按执行时间的先后排成一个全序，成为该时间段内的一个执行过程（尽管这与相对论的时空观不符，但在一定尺度范围内，对我们所描述的问题而言，这是足够的）。

映象是对象空间的部分状态在某一对象中的反映，并不是所有对象都有一个对外部空间的映象。空间中的任一状态，在任一对象中的反映都应与状态本体保持一致。映象表达了现实世界中实体之间的静态联系，而行为表达的是实体之间的动态相互作用。二者互为因果，映象是行为的结果，同时又与处理目标、对象状态一起决定本对象的下一行为，这在现实世界中也是常见的，例如：NFS 中 MOUNT 上的一个子目录，一次在线的检索结果等等，当然，最高级、最复杂的映象是人类的意识。

定义 1 UN 的一个形象为三元组(S,a,I)，其中 S 是分布共享对象空间的状态，a 是刚发出但还未执行发出的一个行为，I 是不同对象的映象。

空间状态与映象是不断变化的，但在两个连续行为发生的时间间隔内，可以认为它们的状态是确定的。当然，这种空间状态的确定性，或称之为空间状态在时间上的一致性是通过对行为的约束实现的。因此，假设在某一时刻，对象 o 即将发出行为 a，且它目前的状态为 s，映象为 i，则此刻 UN 的形象如下图所示。UN 的处理过程就是 UN 形象的变化过程。



我们之所以称该模型为一致网，主要是基于如下几方面的原因：

①一致性是交流的前提。我们无法想象一群分别面对不同环境（这里指的是广义环境，包含人们长期积累所得到的背景知识）的人们，应该怎样做才能互相交流；他们缺乏基本的相互理解，没有共同的交流基础。一致性从来就是一个完整体系的起点，UN 自然也不例外，如下几方面的一致性要求是 UN 存在的前提。

②对象空间需要一致性。如前所述，空间状态在某一时刻是确定的，不允许将不确定概念遗留在空间状态之中。另外，从对象空间的实现来看，为提高访问效率，往往采取复本机制，为此，就需要一定的实现机制，保证复本之间的一致性。

③并发操作需要一致性，与 TM 不同，在 UN 处理过程中，对未来可能出现的行为是完全不可知的，这种行为的不确定性是并发系统的重要特征。不仅如此，一个行为的完成通常需要一定的时间，在某一时刻，UN 中完全可能存在几个行为并行地执行，因此，如何保证执行时间上有交叠关系的并发行为结果有意义，即保证并发操作的一致性，是 UN 的主要研究内容之一。

④映象需要一致性。映象对一致性的要求比空间一致性的要求更为严格，基本上，内部的反映应与外部的状态保持同步，因为行为的作出通常取决于目标、内部状态与映象。如果不能保证映象的一致性，那么就不能保证行为的正确性，甚至会破坏空间状态的一致性与正确性。例如，在 CSCW 中，一个十分典型的需求是"你见即我见"(WYSIWIS)，这是协作的基础。

另外，从上面的介绍中我们不难看出，UN 的有效表达性依赖于如下三方面的假定：

①并发的交叠语义。PETRI NET 所代表的真并发语义与 Milner 等人坚持的交叠语义是对并发语义的两种主要理解与处理，它们的区别在于系统中是否存在一个统一的时钟。诚然，具体的物理上的统一时钟是不存在的，然而在牛顿定理可以起主要作用的时空范围内，我们有理由相信，从"事后"角度看，即便是两个不存在因果关系的行为也必然是一先一后发生的（爱因斯坦的手表一定是与统一的作息制度相协调的）。那么在 UN 的某段历史内，所有发生的行为必然可以排成一个全序执行队列。

②行为的原子语义。作为具有离散特性的活动，行为是一个纯过程，即一个行为引起的操作在对象状态的变换过程中是连续的，既不允许被中断，也不需要存在中间输入，所有的输入都是在操作开始时准备好的。从这一点看这正是传统图灵机所支持的，如果不是这样的话，即中间有 ORACLE 的输入，那我们应该将行为划分为前后两者及 ORACLE 总共三个原子行为来分别处理，以保证行为的纯过程。当然，在现实世界中，对于利用离散信号进行的计算过程而言这是容易做到的，而对于某些连续过程，则须

维普资讯 http://www.cqvip.com

采用阶段积累然后脉冲式产生外部行为的方法来实现。一个原子行为可以是一个数据库查询操作，可以是在屏幕上画一个点，还可以是一条指令的执行。

③对象的持续语义。任一对象的任一操作的执行都必须排除"永不停机"的可能性，即每个操作过程都具有良好的自我控制能力，在有限的时间内完成相应的操作，得到可利用的结果，或者报告异常情况的发生。"永不停机"实际上意味着"死机"，同时也意味着非正常"死亡"，会给系统带来不良的影响。

从下面起我们将重点分析 UN 的实现问题。

2.2 分布空间的构成方式

对象空间的一致性并没有束缚空间实现的多样化，相反，由于实际的网络通讯延迟与开销不可忽视，因而必须在逻辑一致性约束下，结合具体的应用领域及不同的计算机网络，研究相应的实现机制，以提高空间利用率，减少响应时间。我们依据对象空间的静态特性以及 UN 与具体计算机网络结点的对应关系，将空间结构分为四类：集中式、分布式、复本式及集成式。

(1)集中式结构。将所有对象驻留在网络同一结点上是最简单的实现途径，驻留了对象空间的结点作为主结点负责对空间的管理。对空间中任一对象的操作首先传递给主结点，由主结点实施操作。这种集中式结构的空间一致性易维护，而且统一的操作使所有的并发行为串行化，有效地简化了并发行为的一致性。其缺点在于所有的操作都需要主结点服务，当对象空间增大时，主结点将成为系统的瓶颈。另外，由于所有对象都在主结点上，系统的可靠性也更多地依赖于主结点，因此往往由高性能的结点来承担驻留对象空间的任务。

(2)分布式结构。分布式的方法可以克服集中式结构所固有的缺点，不同的对象可处在不同的结点上，对象空间的分布结构可充分利用网络上计算机结点的自治性，交互作用较多的对象将被分配到同一结点上，这样可大大减少网络开销，提高响应时间。与集中式相同，在分布式结构中每一对象只有一份内容，容易维护一致性，而且由于对象是分布的，所以不存在明显的瓶颈，可靠性也有所提高。其不足之处在于对象分布信息的维护，为实现分布对象空间对用户的透明性需要一个记录和维护每一个对象所处实际物理结点的机制。

(3)复本式结构。不论是集中式还是分布式结构，其共同点是对象无复本，空间的一致性容易维护，但是，单复本机制对于映象的支持不足。在 UN 中，映象变化的频率远远高于空间状态的变化频率，因此，如果所映象的对象状态在本地，则可大大减少映象的开销。复本式结构就是在每个网络结点上都设置相同的对象空间复本，这样，大多数映象过程可以在本地结点完成，表现出比分布结构更强的自治性。当然，对于改变空间状态的行为，则需要更新各个结点上对象空间的内容，以维护空间的一致性。复本式结构简化了映象的实现，增加了对象空间变化的开销，通常情况下，由于映象变化的频率远远高于空间状态的变化频率，所以从总体效果看，网络开销降低，访问时间减小。

(4)集成式结构。综合考虑上述几种对象空间结构可以发现，没有那种结构具有明显的优势。在实现难度、时空开销、一致性维护等情况方面，它们各有所长，因此，一种很自然的想法就是综合运用两种或三种不同的单一结构方式，构成一集成式结构，对不同的对象采用不同的空间结构来实现。

从对象空间的动态特性出发，我们还可以允许对象通过迁移、复制、加入以及退出等途径达到动态实现。

2.3 并发行为的约束

与空间、映象相比，在已有的涉及网络的应用系统中，关于并发行为的研究最多，比如：分布式数据库领域、分布共享存贮领域等。并发行为的约束即是规定行为之间的执行次序，以保证并发行为执行结果的正确性。在单机环境下，程序中每条指令的执行次序是十分明确的，顺序执行是串行程序的主要特征。同样，按照 UN 中关于并发的交叠语义规定，从总体上看，某一时间段内 UN 中所有发生的行为集可以按顺序排成一队列，问题在于如何控制这一全序队列的产生。

我们知道，行为是由处理目标、对象内状态以及映象所决定的，同样，行为的完成意味着向处理目标前进了一步，并且或者改变了某些对象的映象，或者改变了某些对象的状态。

定义 2 Aw 为改变对象状态的行为，Ar 为改变对象映象的行为。如果行为 Aw 与以前的若干个 ar，aw 相关，定义 $A=(a_1,a_2,\cdots,a_n)$ 为 Aw 的关联行为集，并台称 A 与 Aw 为复台行为。

定义 3 如果在 UN 的某一时间段内，所有发生的行为按发生的先后次序排列，得到一个全序队列：$(a_1,a_2,\cdots,a_n)$，则称该行为队列为一个执行，$E=(a_1,a_2,\cdots,a_n)$。

显然，如果对每个 Aw，A 与 Aw 在一个执行 E。

维普资讯 http://www.cqvip.com

中具有顺序：…，A，Aw，…，由于 A，Aw 的部分执行类似于一条指令的执行，因此 E 就相当于单机环境下指令的串行执行过程，自然能够满足 UN 的一致性，对并发行为的约束就是要保证系统中的真实并发执行过程 E 具有与执行 E_0 相同的最终效果，即 E 与 E_0 是等价的。

如果两个行为同时对一个对象状态进行操作(ar 或 aw)，且其中至少一个为 aw 则这两个行为是冲突的。显然，无冲突行为之间的相应次序是任意的。

不加证明地，我们引入如下两个推论：

推论 1 两个执行 E 与 E_0 是等价的，当且仅当它们包含相同的行为集，且在 E 与 E_0 中，冲突行为具有相同的部分顺序执行过程。

推论 2 两个执行 E 与 E_0 是等价的，当且仅当它们包含相同的行为集，且在 E 与 E_0 中，每一对冲突行为的相对次序相同。

利用上述分析结果，我们可以分别构造不同层次的并发行为约束策略：

(1)限制 E 与 E_0 的所有行为顺序一致，这是一种强一致性机制，易于实现，然而完全丢失并行性，效率低。

(2)限制有冲突行为所在的复合行为在 E 与 E_0 中的一致性。该机制是一种部分同步机制，部分利用并行性。

(3)只限制有冲突行为的相对次序。通常利用加锁机制来实现，充分利用了并行性、自治性，效率高，实现开销较大。

2.4 映象的实现

映象作为对象对外部部分状态的反映，与状态的复本具有相似之处：都与状态本体的内容保持着逻辑上的一致，但它们之间的区别也是明显的。首先，从功能上看，当前映象是决定本对象下一行为的重要因素，它是对象与外部交流的中间结果，因此在许多对象中，映象是必不可少的；而复本是为克服网络时间延迟与带宽有限的不足，为了提高执行效率而引入的，从一致重要性来看，并不是必不可少的成份。其次，映象内容必须与状态本体保持同步关系，而复本却允许在保证执行正确的前提下放宽一致性约束。另外，在实现结构上，复本与网络上的处理机结点相对应，而映象与 UN 中的对象相对应。

映象的反映区域与初始状态是通过对象建立的，表明对象当前所关注的空间状态区域，而其内容却是随着其本体的变化而改变，除非将其锁定，也就是说，对象空间中某一状态发生变化时，必须立即将新状态发给所有目前具有该状态映象的对象。因为这些对象其后的行为很可能就是依据该状态做出的，如果不能及时地更新映象内容，就不能保证 UN 中行为的正确性。为实现状态本体内容与映象内容的同步，实时广播行为是十分必要的。另外，同一状态在不同对象中映象的内容是相同的，但映象的形式却可以是多样的。同样一个数据，即可以用数字来表示，又可以用字符、高度、色彩等途径来表示。

在 UN 的核心概念中，映象是最难描述的，映象的加入导致了对 UN 描述的诸多限制，但是随着计算机网络的发展，以及人机交互成份的增多，映象将成为 UN 中的一个主要研究内容。

2.5 一致网的开放体系

计算机结点之间的互操作，构成 UN 赖以生存的逻辑对象空间，UN 将继续保持开放的特征，成为功能强大的模拟模型。UN 的开放主要体现在以下两个方面：

(1)UN 的演化。对象间的相互作用关系是构成 UN 的基础，因此 UN 的规模是随着对象间关系的变化而不断演化的。只要对象之间还保持有互操作关系，那么 UN 就仍然保持生存状态。在 UN 的生命期内，UN 维持一种动态的演化。一个 UN 之外的对象可以通过向 UN 中的某一对象发出行为而被动地加入 UN。当然被动加入 UN 的对象应首先具备加入 UN 的条件，必要时，加入 UN 的对象应首先在所属的结点上建立起复本，甚至创建起映象。与之相对的是，当某一对象退出 UN 时，必须首先申请，通知所有正在使用该对象状态的对象，包括复本与映象(主要是映象)，必要的话，对它们优先处理，然后将它们作废，或进行其它处理(如有必要，重建一个包含该状态的对象)。这样，申请退出 UN 的对象即可退出，从而实现 UN 的可靠的伸缩式演化。

(2)UN 之间的互操作。计算机网络资源是十分宝贵的，一个 UN 是建立在网络之上的具有相互作用关系的逻辑对象集。因此一个计算机网络上通常总是存在许多 UN，它们是为不同的目标而分别建立起来的对象集，甚至一个对象可能同时参与多个 UN 的演化，这就需要不同的 UN 在同一网络上协调地运行。从 UN 内部保证这种协调是有一定困难的，但是可以在网络上建立专门的监控对象，负责协调不同 UN 之间的互操作。

开放概念与异构密切相关。不同厂商的处理芯片，不同体系的处理结构，不同的操作系统以及不同

维普资讯 http://www.cqvip.com

的对象表示等等都是造成异构的原因。能够在异构的环境下使对象互操作，是UN开放体系的基本要求。

2.6 UN的表达能力问题

既然UN是一个抽象模型，象征着物理过程中多个实体之间的相互作用，我们自然要问：UN在表达能力方面有什么限制呢？即对于现实世界中的物理过程，哪些是可以用UN描述的，哪些是UN能描述，现实世界中不存在但可以虚拟构造的作用过程？

在2.1节中我们曾对UN作过三方面的假设：并发的交叠语义、行为的原子语义以及对象的持续语义。值得注意的是：三条假设不是我们为简化模型而提出的，我们是被迫的，换句话说，不满足这三条假设的过程，不能被归约到UN之中，"超出"了UN的表达能力。从目前发展状况看，这些"超越"UN表达能力的过程是暂时不必考虑的。

另外一个与实现相关的问题是对象空间结构的层次递归问题。作为一个模型，我们不必对层次递归的深度做约束，但计算机网络上的资源是有限的，无法表示一个无限递归的现实实体。而且，即便是递归深度较高的结构，由于存贮上的限制，也不能表示出来。因此，UN通常表示两种实体：一种是递归层次不深的现实实体，一种是具有无限递归层次的虚拟实体，采用类似"分形"思想表示。

对UN的研究，应将效率放在首位，也许UN的表达能力很重要，但目前更重要的是在已知的表达能力范围内，提供高效可行的实现方案，这是来自实践与经验的要求。

2.7 UN的基本框架实现

既然建立在计算机网络上的不同应用领域在UN上得到了统一，那么不同应用的基本框架的实现也可以统一起来，这可大大减少各个领域的实现开销，而且支持更大范围的互操作，更好地体现UN的开放特征。

对象是构造UN的砖石，对象技术的发展已有几十年，但在实际系统中采用的对象技术还不多，大多数是引入对象的思想，或者将数据结构作为对象，或者将过程作为对象。然而只有在对象与计算机网络结合之后，我们才真正体会到对象的重要性，状态与过程才真正被"封装"在一起了。

OMG的CORBA在构造对象空间方面已做出了卓越的努力，建立在CORBA之上的空间，可以使不同对象通过ORB建立起联系，并且由于动态对象池机制的引入，使空间的伸缩性十分容易实现。因此，借助于CORBA构造UN的基本框架是十分适宜的。

三、UN与典型系统

UN是计算机网络中多实体作用的抽象模型，那么它就应具有充分描述多种实际应用系统的能力。本节中我们就在UN与不同的实际应用系统之间建立起联系，从UN的观点结合具体系统的特点，描述并研究基于计算机网络的实际应用系统。我们以计算机支持的协同工作、分布式数据库及虚拟现实为典型分析目标。

3.1 计算机支持的协同工作(CSCW)

CSCW是指在信息社会中，人们在计算机与网络的支持下，为同一目标而协调地工作的处理模式。自1984年被提出以来，很快成为一门迅速发展的热门方向。基研究内容不仅涉及到了电子会议系统(EMS)、计算机辅助设计(CAD)、群体决策支持系统(GDSS)、多媒体(MM)、分布式交互仿真(DIS)等计算机学科，还与组织学、人类学、社会学、心理学等更广阔的科学领域息息相关，给传统的计算机网络的研究带来了巨大的冲击。

CSCW所具有的交互、协作、分布、共享、开放特征，以及其本身对计算机网络的高度依赖性，使之十分宜于用UN模型来描述。

CSCW所进行的工作目标是UN对象空间的主要组成。另外，各参与者的操作与所看到的屏幕显示，合起来可以用一类特殊的对象：代理机制来表示，这是一类与人相对应的对象，它们与组成工作目标的对象群一起构成了完整的对象空间，该空间的特征从属于一般UN的特征。

CSCW中的并发操作也是十分明显的，可以说，允许并发操作是提高工作效率的主要因素之一，也是CSCW蓬勃发展的重要原因。在EMS、GDSS等结构简单的系统中，已有基本的控制策略被提出，象FLOOR-CONTROL、TOKEN-PASS等方法，在许多群件的实现中采用上述控制方法的实例也十分普遍。显然，依据定理，适用于CSCW领域的策略，尤其是约束条件更宽松，处理效率更高的并发控制策略还有待深入研究。

最明显的映象是参与者所面对的屏幕所显示的内容。在许多场合下，CSCW需要"你见即我见"(WYSIWIS)，即保持映象与主体内容的一致性，只有这样才给参与者一种拉近的、面对相同目标的感觉。很显然，参与者将发出的行为是由当前屏幕的输

维普资讯 http://www.cqvip.com

出内容、自己的想法与处境、以及最终目标所决定的。

UN的开放性在CSCW中也有十分淋漓尽致的体现，参与者可以十分方便地创建、消除、移动对象，自身也可以十分灵活地加入或退出协作，而整个工作则一直保持在一种良好的运行状态之中。

CSCW系统充分利用计算机网络的优势，克服了空间障碍，使处于不同位置的人们不用聚集到一起便可以互相讨论、设计、工作。而且由于大家可以同时展开工作，避免了传统流水线工作模式下各环节之间本质牵扯较多，而实际过程中联系太少的缺陷，从而使工作效率大大提高，工作开销却大幅度下降。这也是CSCW得到计算机网络人员的高度重视的主要原因。

大型设计是一个十分复杂的过程，一方面工作量大，采用串行的方法将导致很大的时间开销，另一方面设计的各个环节之间互相约束：一个环节的变化将引起其它环节一系列的变动。因此，协同设计不仅存在横向的空间划分问题，还包含从市场调查、需求分析、…、一直到产品维护中不同环节的协调问题，而且设计过程中还将产生大量的数据，需要分布式数据库的支持。

尽管DIS是在军事领域发展起来的，但从计算机网络角度看，DIS代表的是一种新型的分布、实时、可视化，且可交互的人在回路中的仿真环境。在协作设计中，对象空间是被逐步完善起来的，工作的目标就是建立起一个由最终对象群表示的设计体，而DIS中，从一开始，对象空间的大部分结构就是确定的，DIS只是用来对一种过程或装置或人员进行验证、测试的仿真环境，尽管DIS的实现很大程度上取决于各种先进的传感、显示设备，从其内部结构及运行过程看，它是属于UN范畴的。

DIS的发展使我们更加相信，在现实世界多实体的模拟方面，基于计算机网络的UN具有强大的描述力量，不仅在军事训练方面，在教育、医疗以及其它更复杂的多实体活动方面，DIS都会大有用武之地，UN的概念也会对这些系统的研究与开发起到一定的启发作用。

3.2 分布式数据库

计算机网络与传统数据库技术的结合，产生了分布式数据库，由于大型系统组织与管理的需要，基于网络的数据库被广泛地研究和开发。迄今为止，分布式数据库的发展已取得了决定性的成果，解决了许多基本问题，并已有相当一部分成功的产品推出，这些工作对计算机网络的发展具有开拓性意义。

在分布式数据库中，最被强调的是数据，对应于UN中的对象空间。围绕着对象空间划分、分布与冗余展开的工作是分布数据库研究的主题。关系数据库是分布式数据库的主流，这从侧面证明了对象之间"关系"而非"函数"的互相作用方式。

分布式数据库也许是最早考虑并发操作控制的实际系统，并研究出两相锁、时间戳等构造一致性并发操作的算法。

设计数据库的最终目标是让人们方便地组织、管理、查询数据，在线事务处理(OLTP)是分布式数据库上的主要行为。在事务处理过程中，需参考的许多中间查询结果就是UN的部分映象，它们是后续行为——继续查询的依据。

3.3 虚拟现实(VR)

如果说CSCW中的对象空间过于被动，而行为主要是人的行为的话，那么在虚拟现实中的对象将彻底打破对象之间的区别(针对代理与一般对象而言)。一个先进的VR系统一定是基于计算机网络的，当你在VR中"漫游"时，你无法区分哪些对象实体是由人控制的，哪些对象实体是完全由计算机虚拟控制的。VR将充分发挥UN的表述能力，不仅可以模拟以往通过其它途径无法实现的实际过程，还可以模拟现实世界中不存在的虚拟过程，只要它在UN的表达能力范围之内。从VR角度来看，研究UN的表达能力是十分必要的：表达能力的研究不仅可以完善UN模型的体系，还可以对VR的设计与实现起指导作用，就象通过对TM的能力研究结果分析，警告人们不要企图编制一个测试任何程序的执行是否会导致死机的通用测试程序一样。

VR是模拟的至高境界，它的出现将象符号改变了我们的思维方式一样，令我们对社会、自我、时间、空间等基本概念重新进行思考。从现在到VR进入实用阶段将是一段艰苦又令人着迷的道路。

除上述几类应用外，UN对网络并行程序、分布共享存贮等系统也具有相同的抽象描述能力。象不同严格程度的一致性语义(严格一致性、顺序一致性、程序一致性以及释放一致性等)都是不同程度约束条件下控制并发操作的策略，只是由于这些系统中较少有人的参与，所以我们未将它们作为主要的分析系统。

四、待开展的工作

尽管关于UN的概念与实现在前面有了较多的

维普资讯 http://www.cqvip.com

分析，但与完善、抽象的模型系统相比，UN 仍有一定差距，尤其是作为一个理想的抽象模型，如果不能与实际经验系统相结合，提供比概念、功能更有效的支持，就必然缺乏持久的生命力，因此，必须继续进行以下几方面的研究与探讨：

UN 的进一步描述。这里指的是完善而不是十分精确的描述，尽管 UN 缺乏象数学函数那样的精确表达，但只要它是真实抽象的反映，就有理由继续其完整系统的描述，也许我们会找到一个类似于物理上的测不准原理或复杂系统的混沌理论一样的描述并发不精确的理论。

关于 UN 表达能力的研究。对 UN 的进一步系统描述一定能推动 UN 表达能力的研究。另一个推进 UN 表达能力研究的力量将来自实践方面，随着计算机网络在已有应用系统中的成熟，以及更多的模拟系统对计算机网络的求助，将逐渐解决这一问题。

不同领域中 UN 的有效实现策略。"能做"与"有效地做"是完全不同的两回事。利用 UN 可以支持许多领域，但我们同样关心 UN 框架如何为不同领域提供有效的支持。由于不同领域具有各自的特点，因此在 UN 基本框架基础上，要针对具体的领域进行有效实现策略的研究。

总结 算法与交互模型之间的区别对应着理性主义与经验主义的区别，理性主义从自身确信的假定出发，通过逻辑（算法）推理来解释世界；而经验主义从现实收集到的数据出发，以不完全的原则集（物理定律）解释世界[5]。在漫长的发展道路上跋涉了几十年的对象技术，一直以现实世界中的实体为基本反映对象，但只有在今天，计算机网络发展到了能触及社会各个角落的时候，才给了它充分展示其优越性的机会，在计算机网络上，分布的共享的对象群是主角，它们的交互行为是计算机网络传送的主要内容。

计算机本身的处理能力是超乎我们想象的，但经过这么多年的努力，随着其应用面越来越多，我们发现不仅我们对计算机的依赖性越来越大，计算机对人的依赖性也越来越大；更多的问题需要人机协同工作，各自发挥特有的优势才能解决，计算机发挥其对确定型问题的快速运算能力，人发挥其大脑灵活的分析思考能力。在计算机网络中，人的行为将成为不可忽视的一部分，而 UN 模型在反映人的特性方面（操作、观察分别与 UN 的行为、映象相对应），具有独特的优势。

从一个完整理论体系的角度来要求，UN 显得较为简单；然而，如果从指导分析与设计过程角度来看的话，UN 的功能还是比较充分的。限于篇幅，本文对 UN 的介绍很不完整，如果本文能够引发广大研究人员对计算机网络模型研究的注意，那也将是令笔者欣慰的。

参考文献

[1] Peter Wegner, Interactive Fundations of Object-Based Programming, IEEE, Computer, Oct, 1995

[2] Robin Milner, Processes: a mathematical model of computing agents, Technical Report, Edinburgh University

[3] 吴允曾，《吴允曾选集》，1987

[4] Robin Milner, The Base of interactive, CACM, Jan, 1993

[5] Peter Wegner, Interaction as a Base for Empirical Computer Science, ACM Computing Surveys, 27(1), 1995

[6] Michiel L. Dertouzos, Communication, computer and network, Scientific American, 1991

[7] 谢立、孙钟秀，分布式数据处理，国防工业出版社，1990

[8] 周兴社、王千祥，支持 CSCW 的模型：分布共享对象 DOS 研究：描述语义及对象结构，南京大学学报，1995, 10

[9] 王千祥、周兴社，支持 CSCW 的模型：分布共享对象 DOS 研究：并发控制及界面设计，同[8]

[10] 王千祥等，从图灵机到一致网，西北工业大学计算机系研究报告，1995 年 11 月

维普资讯 http://www.cqvip.com

计算机科学 1996 Vol. 23 №. 6

图:有待进一步开拓的人机通信媒介

刘正捷
(大连海事大学计算机系 大连 116026)

朱宗元
(中国科技信息研究所重庆分所)

摘 要 The role that pictures play in current human-computer communication is restricted in many ways in comparison with its counterpart in human communication. This difference is made mainly by being short of semantic information of pictures in the machine. Some representative work in this direction is introduced and discussed in this paper, including picture semantics oriented dialogue, semantic integration of pictures and natural language and visual context facilitated outline-detail separated communication. Finally, further research into the issues related to theories, metaphors and knowledge representations are suggested.

关键词 Human-computer communication, Picture, Semantics, Human communication.

1. 引 言

做为一种信息的记载形式和通信媒介,图(picture)的发明和使用远比符号语言为早,而且至今仍以其无可替代的价值在人类生活的各个方面被广泛使用着,如各种绘制的图解、工程图、地图、广告画、标志、美术绘画以及摄制的照片等等。随着计算机技术的进步,其处理的信息形式也从单纯的符号语言发展为越来越多地涉及到图,特别在人机界面上图已成为日益重要的通信媒介。有人甚至认为图将会取代符号语言在人机通信上占主导地位[1]。然而,与图在人际通信中的作用相比,人机通信中图的使用仍有许多方面存在很大局限性,成为阻碍人机通信效率提高的重要因素。因此,有必要认真审视一下图在目前人机通信中的作用,基于图的特性及人对图的使用方式,来思考如何在人机通信上进一步挖掘图的潜力。

本文首先从通信媒介的角度考察目前人机通信中图的使用并指出其不足,然后介绍和探讨发掘图的潜力的几种思路和方法。

2. 图在当前人机通信中的使用

这些年计算机技术的发展使得图在许多计算机应用中的使用成为可能,而且在一些领域中图已占据着重要地位,如 CAD、虚拟现实、信息直观化、CAI、仿真、图象处理和电子出版等领域。在这里,图或是作为计算机的输入和处理对象,或是作为面向用户的信息表现形式,或是两者兼而有之。然而,从人机交互作用的角度来考察图的使用时,我们不难发现,与人际通信中图的运用相比,在许多方面存在局限性。

人际通信中,图是十分重要的媒介。一方面图可做为直接媒介将信息直接传递给接收者,其特性使它对某些类型的信息有很高的表达能力和传递效率[2][3];另一方面图还可做为间接媒介在信息传递中起物理上下文[4]作用,通过简化信息的直接表达和间接传递隐含的信息来大大提高通信效率。然而在许多情况下,对于以图为媒介的人际通信实例来说,当把其中一个对话者换成计算机时,其通信的概念层次、表达手段、对话角色以及对话顺序等方面都不得不发生很大改变。这首先意味着通信将以一种人所不习惯的面向机器的方式进行,同时也意味着通信效率的降低和人的认知负担的增加。比如,当向一个受过职业训练的机械工人展示一幅机械装配图时,这幅图所传递的机器部件的几何特性、彼此的空间关系乃至装配顺序等丰富信息对于他来说一目了然。可是当向计算机展示(扫描输入)这幅图时,能指望机器从中得到什么?对它来说这只是一个信息量贫乏的位图(bitmap),目前的机器视觉技术还远不能实时地对现实世界的任意景物作出识别[5],因而从人机通信的角度还无法指望机器从图中自行获得人所能获得的同样多的信息,除非以极其繁琐的对话方式面面俱到地将这些信息传递给它,但这与人际通信意义上的对话已相去甚远了。目前人机通信中图的使用与人际通信相比在某些方面存在明显差距,由此导致的通信效率的降低也是显而易见的。

是什么原因造成了这种差距?解决这一问题的

维普资讯 http://www.cqvip.com

出路又在那里？通过对目前计算机应用中图的使用情况的分析，存在差距的主要原因在于以下几个方面。

(1)机器缺乏有关图的足够丰富的信息。在人机交互中有效地使用图的关键是让计算机具有基本图形信息之外的更丰富的关于图的信息[5]。从计算机图形系统的角度，有关图的数据可以分布在从面向机器处理到面向所表现对象这一范围内的若干层次上。目前的现实情况是，机器往往具有面向机器处理的信息，缺少面向所表现对象的信息，较典型的例子是做为位图来处理的图，它们对机器来说只是象素的阵列。机器中有关图的信息的匮乏使得用户与图的交流经常只限于"看图"。

(2)机器中关于图的信息不面向最终用户。在一些场合，虽然机器拥有一定的关于图的信息，比如图形系统中几何图元数据或场景描述，但它们只面向图的设计和生成，其作用往往只限于图的绘制过程，即使对用户开放，也只限定在图的模型设计者这样的特殊用户，一但图被生成，这些数据便无用武之地了，不能参加与图的最终用户的对话。

(3)机器中的信息与人具有的图的语义概念不相匹配。目前计算机具有的图的信息一般只限于图的几何属性，比如对象的形状、位置、颜色等，至多是与几何对象直接对应的有限的语义信息，如它们的标识符或对应现实世界对象的名字等。这些信息在图的语义中属于较具体的描绘信息[6]。然而，人的思维和通信往往横跨多个抽象层次[7]，人所具有的图的语义概念除了描绘信息之外，在许多情况下占主要地位的是有较高抽象度的描述信息，比如，地图上的"沿海省份"，发动机装配图上的"传动部件"等等，这些对人来说习以为常且在视觉上也显而易见的语义概念在抽象层次上超出了基本几何信息，往往正是机器所不具备的。人只得将自己习惯的概念转换成机器能接收的较低层概念来与之对话，这必然降低人机通信的效率。因此，在导致人机通信和人际通信在图的使用上的差距的诸多因素中，这是一个最重要的因素。

(4)机器缺乏与人际通信相适配的图的语义运用机制。在人际通信中，图一般只是同时采用的多种通信媒介中的一种。在这些媒介之间有着复杂的语义相互作用，比如，图在人际通信中往往起着物理上下文作用，在这种背景下对图的语义的运用能力是人的一项通信技能。类似的图的语义运用机制在目前的计算机中尚不具备，这必然对人的通信能力产生限制。

概括以上各方面的原因，可以认为问题的症结在于机器缺乏图的语义信息以及面向人机对话的语义运用能力。有关图的语义的研究及应用在目前的人机交互作用领域是一个被忽视了的问题[8]。因此，要从根本上缩小人机通信与人际通信在图的使用上的差距，就应当选择图的语义为突破点，找到让计算机获得图的语义并将其运用于人机对话的有效途径。

3. 开拓图的语义：几个研究方向

3.1 真实感图的语义的生成和使用

目前的虚拟现实系统多侧重于人的视觉感受，真实感(photorealistic)图形技术在其中起着关键作用。当前限制虚拟现实技术应用的一个主要障碍是人缺乏与虚拟环境的交互作用能力。重要原因之一是缺乏语音(语言)交互能力[5]，这除了语音识别方面的原因之外，一个更具根本性的原因便是机器缺乏所产生的真实感图的语义信息。虽然机器内部具有一些有关图的数据，如场景描述等，但由于其作用仅限于图的绘制(rendering)过程，并不用于人机对话，所以对话时机器根本不知道人此时看到的是什么，尤其在画面随人的运动而变化时，从而难以理解人对可视信息的引用，无法针对当时的视觉上下文与人进行对话。

为了解决这一问题，必须使场景描述数据除了用于画面的产生，还可用在人机对话中，使画面语义在为人所知的同时还为机器所了解。Hyper-Renderer[9]为此做了有益的探索。它的设计主要基于这样一个观点，即图形系统的可用性能通过使有关绘制过程的信息为其他过程所用而得以改善。具体说来，就是通过让系统在绘制画面的同时产生能为对话过程所用的画面语义信息来实现的。做为正常绘制过程的副产品，Hyaper-Renderer 产生关于所绘画面的画面描述(image description)，这是画面(图)的浅表语义，它包括画面上每个对象的可见性、阴影、光照、以及对象间的遮挡和内包容等空间关系。从画面描述中，系统能够知道画面上是什么，更重要的是知道人此刻看到了什么或是看不到什么，从而使人机对话从原来单向的"人看画面"提高到双方关于画面内容的交流。

3.2 示意图与自然语言的互补

多媒体人机通信技术近年来有了迅速发展，但对于怎样将不同媒体结合以设计出有效的人机界面仍然严重缺乏理论指导[10]。在图和正文的结合上也存在同样的问题。目前的人机界面设计在考虑图和正文的结合时通常只注重整个图的通信功能，并不详细考虑图本身的构成和各成分的语义以及在此基

维普资讯 http://www.cqvip.com

础上与正文的互补。然而，为了使图和正文的结合做为一个整体更有效地完成其通信功能，必须进一步考虑图中各成分的语义功能以及它们之间和它们与正文各片断之间的相互关系[13]。针对机器技术说明资料的生成，WIP 尝试了一条基于规划的方法来实现示意图与正文间有机结合的途径[13]。其设计者认为，多媒体信息表达可象正文表达一样被视为具有特定通信目标的一个通信作用序列，应当产生于欲表达信息的共同表示。基于这一观点，WIP 采用递进规划方法来规划和生成包括示意图和正文的多媒体信息表达。做为规划过程的算符(operator)，在知识库中设有一系列表达策略，用来决定如何使示意图与正文在基本语义成分级上彼此互补。这种互补关系取决于多种因素，除了用户任务和资源限制之外，应着重考虑欲表达信息的类型以及图中成分和正文片断的通信功能。根据对应用域及信息内容分类的研究，将欲表达的信息成分分为七种类型，即具体、抽象、空间、时间、协变、数量和否定信息，并对它们进一步划分了子类型，同时在这些信息类型与适用媒体(图或正文)之间建立了对应关系。针对示意图和正文的特点，WIP 将通信功能区分为十种，即吸引关注、比较、细化、使能操作、注解、标题、鼓动、证据，背景和概括。为了从示意图中分解出基本语义成分，WIP 将图视为基本影像的组合，又把影像进一步区分为三维描绘、二维图元和字符串，在此基础上考虑它们表达的信息种类和通信功能。图和正文在基本语义成分级上的内聚性就是基于这样的信息类型和通信功能划分通过规划来实现的。图 1(取自文[13])给出了 WIP 生成的一个信息表达实例，其通信目标是告诉读者怎样向咖啡壶内加水。

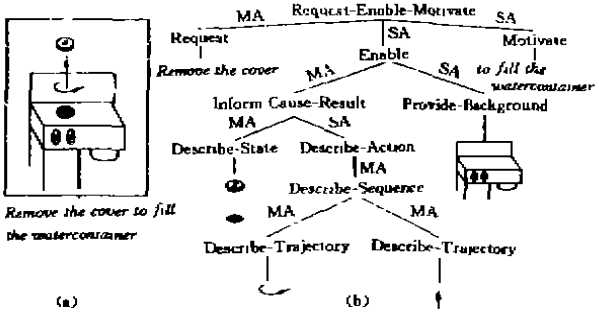


图 1 (a)WIP 生成的信息表达片段；
(b)相应的信息表达规划(MA：main act，
SA：subsidiary act)

VITRA GUIDE 系统[14]在车辆驾驶向导应用中针对空间语义的传递也采用了规划的方法来生成包括自然语言和图在内的多媒体信息表达。

3.3 赋予扫描图象以语义交互性

图象扫描仪的出现使得近年来扫描图象被大量用于计算机应用系统，这在一定程度上满足了人们对图的需求。然而，目前扫描图一般是以一幅图做为整体来处理的，用户与计算机之间不存在关于图的内容(语义)的可交互性。设想当屏幕上显示出一幅扫入的室内家具布置图时，如果用户用光标指向一个柜子并询问"这是什么?"、"它的用途是什么?"，机器对此只能是无动于衷。因为对机器来说这只是一个由象素组成的位图，它对图所表达的语义(特别是从人的角度所理解的语义)一无所知。对于上述查询，机器除了知道光标的二维座标之外，根本无从知道指向的是什么东西以及更深层的含义，因此无法针对图的内容与用户交流。实际上，这里计算机支持的只是图与观看者的关系，而不是面对图的两个对话者的关系，这使图的使用受到很大限制。

要解决这个问题，必须让计算机掌握扫描图象的语义，即具备图的语义模型及相应的语义交互能力。图的语义在这里指的是，从人的角度观察和理解到的图所表达的含义。基于对各种图的经验性分析，图的语义可以有三个层次，即视觉表现语义、视觉想象语义和关联语义。

(1)视觉表现语义：这是图在视觉上直接表现的信息，或者是人凭视觉观察到的信息。这包括图上直接表现的对象的外观及对象间的空间关系。这一语义又可进一步分为视觉感受信息和视觉表示信息两个子层次，如地图上的各种符号和它们代表的对象。而视觉表示信息又可分成基于约定的和面向应用域的表示信息。

(2)视觉想象语义：这部分信息存在于图表现的视觉场景中，但在图中并未在视觉上直接表现出来。比如在一幅汽车图片上，发动机、背向观察点那一侧的车轮和车门等对象虽然在视觉上存在于所表现的现实场景中，但在图上并不可见。这种信息可以很容易地在视觉概念模型(visual mental model)基础上靠想象获得。

(3)关联语义：这是在视觉上并不存在，但却与视觉表现语义和视觉想象语义相关联的信息。当人看到一幅图时，除了视觉上存在于所表现场景中的信息之外，还往往联系到更多的信息。比如对象的非可见属性(物理化学、时间、文化等)和对象间的非可见关系(控制、力学等)。

研究表明，信息空间具有层次抽象结构[15]。在图

维普资讯 http://www.cqvip.com

的语义空间中，视觉表现语义和视觉想象语义是人比较容易获得的、较少依赖人的概念模型，因而在不同人之间差异较小，尤其对视觉感受信息更是这样。相对来说，关联语义的获得较多地取决于对概念模型的引用，甚至要靠别人启发，所以有时在不同人之间存在差异。从这个意义上，可以把视觉表现语义和视觉想象语义称为浅表语义，而将关联语义称为深层语义。

基于对图的语义的上述理解，作者尝试了通过交互方式建立图的语义模型来赋予扫描图象以语义交互能力的途径。系统由经过图的语义模型相联系的语义建模工具和应用程序组成。在图的向量数据基础上，设计者使用建模工具以交互方式建立图的语义模型。在建立语义模型过程中，设计者要根据图的用途选择语义兴趣中心，而且以图的可见信息为起点按照语义层次由浅入深逐步进行。应用程序通过应用程序接口以与语义成分相适配的多粒度方式访问图的语义模型和进行语义处理，从而在人机间实现混合驱动的针对图的语义交流。目前，一个针对具有示意图视觉表现语义的系统原型正在设计实现过程中。值得指出的是，这种途径在原理上也同样可用来对计算机生成的静态、动态图实现语义交流。

3.4 以图为上下文的轮廓与细节相分离的人机通信

人际通信的高效率在很大程度上来自轮廓与细节相分离的特性，即只有欲传递信息的轮廓需经词语符号等直接表达方式传递，细节则由某些间接方式传递。上下文在这种通信方式中是一种十分重要的间接传递途径[16]。当前人机通信低效率的主要原因之一是对人际通信的这一基本特性的忽视。如果能在人机间重建类似的轮廓与细节相分离的通信机制，将会使人机通信从根本上得到改进[11]。在这里，上下文维护和利用机制的建立是关键问题。

在人际通信中，上下文有两种形式：语言(linguistic)上下文和物理(physical)上下文[4]。做为一种物理上下文，图在人际通信中一方面简化信息轮廓的直接表达，另一方面间接传递信息细节。然而，尽管图已较多地用于人机通信，但一般只是用来在预先设定的场合向人传递某特定信息，而不是用做通信上下文，更不是用做上下文来实现轮廓与细节的分别传递。之所以选择以图为上下文来研究轮廓与细节相分离的人机通信的实现，是由于以图为上下文的通信是常见的人际通信方式，人对此有高度的驾驭技能；同时图的直观性使之兼容于人的直觉能力，从而适合做为上下文展示来帮助人机间公共基础的形成[11]。

与上下文在人际通信中的作用类似，在相应的人机通信机制中，上下文模型是一个中心问题。这里首先要探讨的便是这个上下文模型应由什么信息来构成。正如俗话所说："一图顶千言"，图所提供的除了可见信息(视觉表现语义)之外，还有非可见信息(视觉想象语义和关联语义)。在上下文模型中维持这样大量的信息一般是难以实现的，也是没有必要的。可行的上下文模型应当基于核心上下文，即上下文中与当前通信主题联系最密切，而且最容易被通信双方引用的部分[11]。为此，必须找到在图的信息空间中对核心上下文进行界定的方法。

对于图所提供的信息可从语义和表达这两个侧面去加以分析[17]，它们对核心上下文的界定有着不同的作用。从图的语义方面来看，在图所提供的大量信息中，人所关注的部分往往取决于他当前的兴趣点[17]，这说明人在图上所关注的信息，即核心上下文与当前通信主题有密切关系，而且随其改变而改变。另一方面不同知识背景的人在同一幅图上会关注不同信息，而且获得信息的效率也各不相同[18]。这说明核心上下文的选取还取决于人的知识背景。因此，在以核心上下文为基础建立上下文模型时，必须注意核心上下文与当前通信主题和知识背景之间的关系。这要求在人机界面上采用强调手段使当前对话主题和目标对通信双方保持明确一致，同时通过沟通手段在通信双方建立必要的共同知识背景[17]。在此基础上，还应建立一种基于当前主题和共同知识背景对图的信息加以取舍来动态形成核心上下文的机制。

图所提供的信息在表达方面是由不同表达层次的信息组成的，核心上下文的组成同样需要考虑这一因素。在图所提供的信息中，相对来说可见信息对人比较浅显，因而不同人的认识倾向于一致。可见信息又可进一步分成描绘(depictive)信息和描述(descriptive)信息，它们对人有着不同的浅显程度[19]。描述信息涉及到符号系统，将符号系统映射到所表示对象的解释规则往往具有一定的随意性、须经学习获得。因此它的存在是外定的(extrinsic)。而描绘信息则是内定的(intrinsic)，它可以激发起人的与所表示信息类似的视觉经历，其解释不取决于外部定义的解释规则。二者相比较而言，描绘信息更易被人获得，而且对它的认识不同人有着更大的一致性。人在通信中习惯于引用当前容易获得的信息[20]。因而，核

维普资讯 http://www.cqvip.com

心上下文在表达方面更倾向于由可见信息，尤其是其中的描绘信息来组成。图所提供的核心上下文组成如图 2 所示。这样才能最大限度地保证通信双方对核心上下文认定的一致性，保证接收者依据信息轮廓从上下文获取的细节与传递者期望的相同，从而使人机通信顺利进行。

以图为上下文的轮廓与细节相分离的人机通信的实现除了这里对上下文模型所做的初步探讨之外，还涉及到许多有待深入研究的问题，这既需要对相应人际通信做更进一步的分析，也需要发展和完善系统建造特别是上下文模型和人机界面所需的理论和技术。但有理由相信，这种通信机制的最终建立将给人机通信带来根本性的改进。

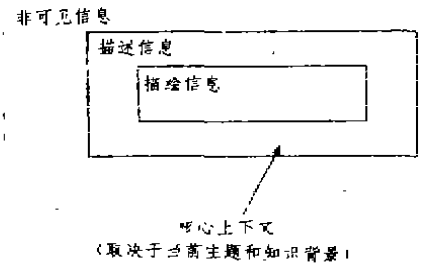


图 2　图所提供的核心上下文的组成

4. 结束语

图已经日益普遍地用在各种计算机应用系统中，成为人机通信的主要媒介。然而，当人们借助图与计算机进行通信时，却可以明显感到这种通信与相应人际通信的不同，图的作用在这里受到多方限制。原因何在？基于图在人际通信和人机通信中作用的比较，作者从不同层次和侧面揭示了造成这种情况的主要原因是机器缺乏与人相适应的语义信息及面向人机对话的运用能力，指出解决这一问题的关键是使机器具有获得图的语义并在人机对话中加以运用的能力。为此，本文介绍了有代表性的几个研究途径。

总的说来，关于图的语义及其在人机通信中作用的研究在人机交互作用领域还是一个薄弱环节，所开展的工作尚不多。这一方面是由于认知学、心理学和语言学等相关学科关于图的语义的研究尚不能以足够精确的形式向计算机科学提供系统建造所需要的理论指导；另一方面，系统建造所需的软件、硬件支撑技术上仍存在一些问题。

作者认为，为了最终实现基于图的语义的人机通信，目前应集中力量在三个方面取得突破。首先应该发展和完善有关图的语义及相应人机通信的理论基础。基于图的语义的人机通信的复杂性使得系统的设计实现不能仅凭感觉行事，而应有理论指导。目前相关学科提供的理论多是似是而非的，离系统建造要求的精确性还有不小的差距。弥补这种差距的一种可行办法是在现有理论基础上，通过对特定应用域典型实例的经验性研究来有针对性地发展和完善现有的理论，使之能对特定应用域的系统建造有实际指导作用。其次是发展与基于图的语义的人际通信相适应的人机通信隐喻技术[10]。目前人机通信隐喻所支持的是图与观看者的关系，为了实现基于图的语义的人机通信，需要能支持面对图的二个对话者关系的隐喻技术，以便使人际通信的知识和技能可以无障碍地移用于人机通信。这一方面要深入研究人际通信对图的语义的使用方式，从中找出能用于人机通信的隐喻；另一方面还要在现有技术条件下开发在概念和物理上与之适应的界面实现技术。第三是发展适合图的语义的知识表示和处理方法。象现实世界一样，作为它的一种映象的图（特别是象图）有着形式化和非形式化这二个侧面。然而，目前的知识表示及处理方法基本上是面向形式化的，这使它不足以全面准确地表示和处理图的语义。形式化表示可以被看作是非形式化表示的一种扩展，而非形式化表示则提供了形式化表示的上下文[5]，二者间的这种内在联系，也许可以为发展与图的语义相适应的知识表示和处理方法提供一条思路。（参考文献共 20 篇略）