

61-66

设计模型

ICAD

计算机科学1993Vol.20No.3

基于原型的设计模型^{*}

耿卫东 潘云鹤 何志均

TP391.72

(浙江大学人工智能研究所 杭州310027)

摘

要

In this paper, we mainly discuss design knowledge representation and reasoning in ICAD systems. Based on two hypotheses, a prototype-based design model is presented, the association method in design process is also discussed.

一、引言

Smithers 在 [1] 中提出了研究智能计算机辅助设计 (ICAD) 系统的三个途径: 1) 建立设计过程模型。2) 设计知识的表达及推理方法。3) 建立 ICAD 集成系统的体系结构。本文着重从 1), 2) 这两个途径来研究 ICAD 系统。

设计是一个面向目标的, 有约束的决策、探索和学习活动^[2]。设计知识主要包括设计对象知识和设计过程知识。从设计的全过程来看, 设计对象的描述经历了一个从概念到形状、从抽象到具体的过程。研究表明^[2,3,4], 设计师往往将他们的个人设计经验构作成具有不同抽象度的设计要素, 这些要素包括: 设计问题描述、设计对象的功能、结构、行为等。因此, 对设计知识的表示, 我们提出了如下两个假设:

1) 信息分解假设。设计知识可以被分解为不同的类和不同的抽象度, 而且可按不同的方式去处理。

2) 恒等对象假设。对于组成设计对象的部件来说, 无论在设计的哪个阶段, 只要没有对部件执行修改操作, 那么, 这些具有不同抽象度的部件就是等同的。

基于以上假设和认知心理学中的原型说^[5], 我们提出了一个基于原型的设计模型。下面分别从原型的表示、组织、基于原型的设计推理及 ICAD 模型等几个方面加以阐述。

二、原型的表示及组织

2.1 设计原型的表示

设计原型是一些经验性的设计要素的组合, 概括一类事物的基本特征或共同属性, 是对一类事物的抽象。在讨论设计原型的表示之前, 我们先来看一看设计对象的原型表示。

设计对象的描述包括该对象的组成部分、部件之间的关系、属性、功能等, 一类设计对象的原型知识可描述为如下一组方程^[6,7]:

原型知识 = 原型 + 变换规则 + 范例

原型 = 子原型 + 结构关系 + 特征

变换规则 = 一组操作规则, 用来限定对原型的操作,

使得操作结果仍留在原型范畴内

范例 = 一组对应于该原型的实例

子原型 = 原型或基本型

特征 = 属性 + 功能

结构关系 = 方位关系 (上、下、左、右、前、后)

连接方式 (固定、铰接、活动, ...)

大小关系 (<、>、=)

耿卫东: 博士生, 主要从事形象思维、智能 CAD 等领域的研究工作。潘云鹤: 教授, 主要从事人工智能、形象思维、智能 CAD 等领域的研究工作。何志均: 教授, 博士生导师, 主要从事人工智能、专家系统、CIMS 等领域的研究工作。*) 国家自然科学基金和国家 863 资助项目, 并得到浙江大学 CAD 和 CG 重点实验室与校基金支持。

在进行实际设计活动时,设计师往往需要从不同的角度来考察和处理同一个设计对象,这就要求 ICAD 系统中的设计知识必须是集成的^[10,12]。基于这样的思想,我们在设计原型中集成了定性知识、定量知识和控制知识。设计原型不仅包含设计对象的结构信息,而且还包含跟设计有关的其它信息,如环境特征、功能说明、性能指标、技术指标、经济指标等。除此而外,还提供了一套联想、选择、评价方法。这样,CAD系统可以方便地根据设计要求进行联想、选择、分析评价、方案调整等操作来推进设计过程。但设计推理过程不可能毫无止境,为此,我们引入了基本型的概念。当设计空间中的引用原型均为基本型时,设计推理空间就已基本确定,设计推理过程也就终止了。因此,我们给出了如下的设计原型和基本型描述:

```
原型={
  name:原型名.
  component:一组子原型.
  relationship:子原型间的结构关系
  feature:一组特征指标.
  method:{
    association:联想方法.
    selection:选择方法.
    evaluation:评价方法.
  }
  rule:变换规则.
  instance:一组实例.
},
基本型={
  name:基本型名.
  parameter:特征参数.
  method:生成该类对象几何模型的方法.
},
```

其中,原型名是访问原型的入口;组成设计对象的抽象实体(即一组子原型)为得到设计的不同层次的原型空间提供了基础;抽象实体间的结构关系提供了如何将抽象实体构作成设计对象的原则和操作序列;特征描述界定了原型的属性和功能,这些特征信息按其性质和作用可分为三类:功能特征、环境特征、特异性特征。功能特征给出了设计对象的内在功能属性描述。环境特征描述了作用于设计对象的可预料的外部因素和设计

对象在特定环境中所表现出的行为。特异性特征给出了设计对象的一些次要的描述性信息,如设计所要达到的技术指标、经济指标等;变换规则指明了不改变原型范畴所允许的操作范围,当产生了超出范围的操作时,则说明可能是产生了创造性设计,生成了新的原型,也可能是设计失败;一组联想、选择、分析评价方法构成了一种推理机制,根据这套推理机制,从设计要求出发,逐层引用原型知识来推进设计过程;系统中的基本型提供了一些原子对象的描述,最后的设计结果可看成是由这些原子对象“组装”而成的。例如:“椅子”的设计原型可描述如下:

```
原型={
  name:椅子.
  component:靠背,椅面,支架.
  relationship:靠背,椅面,支架的结构关系(略).
  feature:{
    功能特征:可坐等.
    环境特征:(略)
    特异性特征:重量,材料,颜色等.
  }
  method:{
    association:联想方法(略).
    selection:选择方法(略).
    evaluation:评价方法(略).
  }
  rule:变换规则(略).
  instance:转椅,沙发等.
}
```

假如子原型“支架”由诸如“长方块”这样的基本型组成时,那么各为“长方块”的基本型表示如下:

```
基本型={
  name:长方块.
  parameter:{
    length: X
    width: Y
    height: Z
    colour:
    .....
  }
  method:生成长方块的几何模型的方法.
}
```

2.2 原型的组织

设计是一个非常复杂的带有一定机遇性的探索活动,也是一个逐步获取更多信息的演变过程,设计师的知识库的动态性在其中起着重要作用^[13,20]。因此,ICAD系统的知

识库也必须是动态的,这样才能支持复杂的设计过程。

McLaughlin^[3]等根据设计的特点,采用了Kolodner提出的一个动态片断记忆模型(Dynamic Episodic Memory Model)来组织设计知识。

Burclasz和Zeid提出的认知记忆模型中设计知识的组织也基于这一思想。同样,Kolodner的动态建立知识结构的思想也深深影响着我们对原型知识的组织,我们采用了面向对象思想和联想机制,力图通过获取知识和使用知识的过程来动态建立设计原型间的联系。

在人们获取知识和使用知识时,联想起着重要作用^[23,24]。人们在获取知识时,往往表现为一种“智力”上的节省,例如,看到一只“狼”时,往往联想到“狗”,并对头脑中“狗”的信息作适当修改来获得有关“狼”的知识^[25]。在进行问题求解时,联想也是一种重要的从知识库中恢复出相关信息的手段。一般说来,人们的思维活动中的联想方法大体上可归纳为如下几类^[23,24]。

1) 时空相近联想。指从当前已知信息联想到在时间和空间上与其相近的其他信息。如由“椅子”想到“桌子”等,或由“星期六”联想到“星期天”等。

2) 特性相近联想。指从当前已知信息联想到与其性质上相似或接近的其他信息。如由“转椅”想到“躺椅”等。

3) 对比联想。指从当前已知信息联想到与其特点相反的其他信息,如由“冷色调”而想到“暖色调”等。

4) 定势联想。由于机遇和习惯而产生的观念上的联想。如由“垃圾”而想到“脏”。

基于上面的分析,我们在原型中增加了一个域——联想域,用它来记录联想的“痕迹”。在进行原型知识的组织时,首先建立设计领域的面向对象的概念模型,然后根据原型所代表的概念范畴间的语义关系,将原型纳入一种类体系之下,并根据上述四种联想

方法初步确定联想域中的内容,建立一个基本的原型知识结构。在使用原型知识进行设计问题求解时,首先根据原型中的联想痕迹来搜寻对当前问题有用的信息。如未找到,则激发原型中的联想方法来进行相应的搜寻,并根据一定的策略来调整联想域中的内容。实际上,这只是一个单步操作。对整个联想过程来说,它是“分布式”控制的,可从被联想到的原型继续按照上述步骤联想下去,逐步从知识库中恢复出更多的相关信息,并对原型间的联系进行一定的调整,从而形成一个具有动态知识结构的原型知识库。

三、基于原型的设计推理

3.1 原型的操作

从知识处理角度来说,对原型的操作分为两类:结构操作和属性操作。但对设计过程而言,原型的操作可分为如下四类:

1) 原型的落实。指一个设计对象的所有要素均已确定,这时根据原型知识库中的领域知识、基本型的描述及生成方法,逐步综合,从而得到设计对象的精确表示模型,即设计原型的一个具体实例。原型的落实不改变当前知识库中的知识。

2) 原型的完善。是将一个原型细化,得到一个完整的设计空间描述的过程。完善原型的操作是一种发生于多级原型空间里的递归活动,单步的操作一般包括这样三个步骤:提出要求(设计要求)、选择原型、评价模型。原型的完善不会改变知识库中各原型的内容。

3) 原型的修改。是指如果通过评价,发现系统中的原型有些不适用,就可以对原型作适当的修改,以适应新的设计状况。原型的修改操作必须在原型的变换规则允许范围内。一旦某个原型的修改完成后,余下的工作即为完善原型和落实原型。原型修改操作的结果是得到了一个改进的原型,但这种对原型的修改只是临时的或非实质性的,不影响知识库中旧原型的重用。

4) 原型的产生。新原型的产生意味着进

行了创造性设计。原型的产生操作可视为对原型修改操作的延伸,但这种修改已超出了旧原型的变换规则允许的范围,与旧原型已迥然不同,形成了质的差异,促使它被认为是一个全新的原型。

在进行设计推理时,根据当前的设计状态决定实施上述哪一类操作。

3.2 原型推理机制

在阐述基于原型的推理机制之前,我们首先引入设计语境 (Design Context) 的概念。在传统的CAD系统中,设计状态往往仅包含设计对象的描述^[12]。而实际上,设计状态不仅包括当前设计对象的已知信息,还包括设计方向的“暗示”信息,它指导着设计过程的推进。我们把设计过程中每一步设计状态称为一个设计语境。

一个设计语境中主要包括以下三类信息:设计说明、原型知识、控制知识。设计说明指明了当前的设计要求;原型知识指明了与当前设计问题相关的设计原型;控制知识指明当前操作属于哪一类(参见3.1节中给出的四类原型操作),这对联想方向的选择很重要。

对每一个设计语境,之所以要从它继续推理下去的原因有两类:1) 设计知识不充分。2) 当前设计原型中的部分要素不满足设计要求。对情况1),则引用原型中的子原型,形成更具体的设计原型空间;对情况2),我们则根据设计语境中的设计说明所“暗示”的信息,从原型中的联想域和联想方法出发,从知识库中恢复出足够多的跟当前设计问题相关的知识,并从联想到的知识中抽取对当前推进设计过程有用的信息,如果穷尽知识库也未能获得有用信息,则给出失败信息并回溯。如此一直循环下去,直到获得最后的设计结果为止。

四、基于原型的智能CAD模型

设计是一个将一组设计要求转化为满足该组要求的设计对象的过程。基于第一节提出的两个假设,设计要求总可以被分解为设

计对象和设计约束。设计对象对设计过程中设计原型选择起着决定性的作用;设计约束不仅影响着设计原型的选择,而且决定着设计过程中原型知识的变换方向。设计原型除了依据其中的子原型与其他原型发生作用外,另外一个相互联系的途径是通过其中的联想域和联想方法。在进行设计活动时,通过这几个途径来推进设计过程。图1给出了一个基于原型知识的ICAD工作模型。

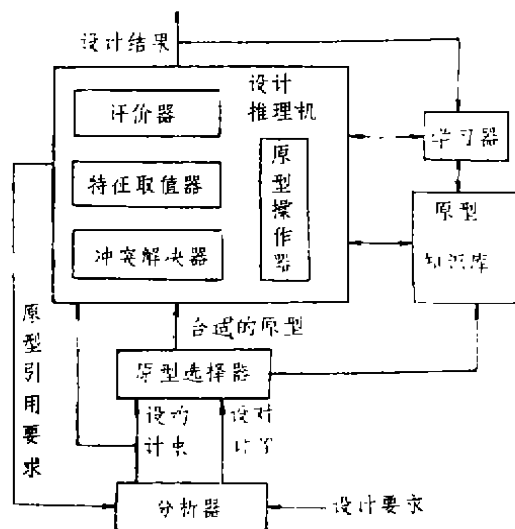


图1 基于原型的ICAD工作模型

设计要求经由分析器分解为设计约束及设计对象,由原型选择器选出适当的设计原型,推理机根据设计约束和所选出的原型来进行设计推理(参见第三节),最后的设计结果一方面作为系统输出给用户,另一方面作为学习器的输入,由学习器进行知识获取,并将所获得的知识自动加入到原型知识库中。

下面主要讨论学习器的工作机理和知识库的一致性维护。

4.1 原型知识的学习

上面已提到,设计是一个面向目标的不断探索和学习的活动。设计师通过自身的学习机制,从以前的设计活动中不断吸取经验,使得自己的领域知识变得日渐丰富。对基于原型知识的设计系统来说,每一次设计

活动的完成都要产生一个实例,有时可能还要产生一个新的设计原型。因此,基于原型知识的设计系统必须有以下两种学习能力:

1)从实例中进行学习。在实际的设计活动中,关于某一范畴的原型知识是逐渐演变的。对基于原型的设计系统来说,随着设计活动的进行,系统中的范例逐渐增多。当关于某一原型的范例足够多时,系统就从这些实例中进行学习,并调整相应的原型知识。

2)从类比中进行学习。当确认设计活动中产生的新原型时,要求新原型与系统中的其它原型有最少的相同知识。这时只有通过原型间的类比学习,才能获得新原型的知识。同时也通过新原型中的联想域建立与其它原型的联系。

当然,设计师的领域知识库也是一个开放系统,因此,基于原型的设计系统还应能够接受从外界直接输入的领域知识。

4.2 一致性维护

向原型知识库中加入新知识可能会引起系统中知识表示的不一致。设计过程中向原型知识库中加入的知识有两类——新实例和新原型。一般情况下,新实例的加入不会影响系统中知识表示的一致性。但一个新原型的加入可能会导致知识库中的原型知识有如下几种变化:

1)相关的原型和相应的范例停止使用,这种情况一般不会发生。

2)新原型取代了系统中某些相关原型的使用,尽管这些相关原型的范例依然存在,但在概念上的使用属于新原型。

3)新原型与系统中的其它原型相互独立,同时在系统中起作用。

对基于原型的设计系统来说,每一个新原型的产生都是由一个(或多个)相关原型演变而来的。在加入系统时一般是相对独立的,不会立即引起替换相关原型的活动。根据设计活动的这些特点,可采取如下的一致性维护策略:

1)新原型在加入系统时,都作为独立的

原型加入,但系统为新原型和它的相关原型保持一个特殊链表。通过这个链表,新原型与相关原型能够被当作同一范畴的不同原型而相互选择、引用。

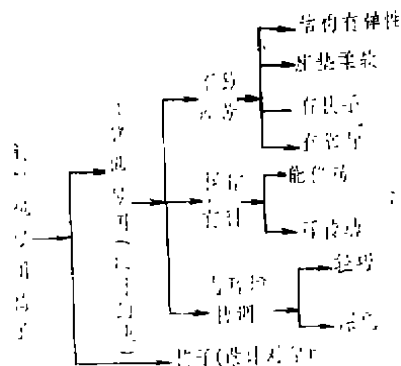
2)对每一个新加入系统中的原型及其相关原型的引用情况规定一个阈值。当对新原型的引用超出这个阈值时,新原型替换旧原型,并将旧原型中的相应范例加入到新原型中。

鉴于旧原型及相应的范例停止使用的情况一般不会发生,因此对这种情况可不予考虑。对于用户直接输入的知识,或由用户指定对知识库的变化,或由系统根据上述策略自动进行调整。

4.3 设计实例

为了更好地说明基于原型的ICAD模型的工作流程,下面给出一个示意性的设计实例。

假设用户要求设计一张供计算机房用的椅子,那么,设计要求首先经由分析器作如下的分解:



然后原型选择器从原型知识库中选出“椅子”原型(参见第二节给出的椅子原型),设计推理机将设计约束作用其中。很显然,系统中的“椅子”原型不满足设计约束,需修改,这时调用原型操作器执行原型修改操作。在设计语境中,把“扶手”这一子原型加入到“椅子”原型中的component域中,并在relationship域中加入相应的结构关系。“椅子”原型中的一部分特征信息由设计约束直接指定,如在当前状态下,椅子的环境特征被约束为可转动,可移动等,其余的特征信息则由特征取值器来推理确定。接下来的工作就是进行原型的完善操作。一方面引用component域中的子原型并通过method域中

的选择、评价方法来进行选择评价,另一方面则通过method域中的联想方法对可移动、可转动等功能特征进行联想,获取更深一层的设计约束和相关知识。重复上述过程,直到原型空间中所有节点均为基本型为止,这时系统中以设计语境的形式给出了设计结果的一个完整的原型描述。再调用原型落实现操作,得到最后的设计结果输出给用户,同时将新的实例加入原型知识库中的“椅子”原型中(这步操作由学习器完成)。

五、结 束 语

从人工智能角度看,设计也是一类特殊的问题求解。目前在人工智能领域中,比较有代表性的问题求解策略有两种——基于模型的推理(Model-Based Reasoning,简称MBR)和基于事例的推理(Case-Based Reasoning,简称CBR)。MBR使用由问题领域知识所构造的模型来进行推理,每一次求解都根据这个统一的模型从头做起,以前的求解经验不能为当前问题的求解所利用,因此适宜于解决新问题。CBR则基于以前的经验——事例,事例中包含有问题说明、解决方案等信息,在进行问题求解时,首先找出跟当前问题相关的事例,然后调整事例中的解决方案,使得它们能解决当前的问题^[22],因而适于求解一些频繁遇到的相似问题。然而,对于复杂的新问题,尤其是设计问题,MBR和CBR都无能为力,这是因为MBR随着问题复杂性的增加而变得不可控制,而CBR则因为事例的检索开销和方案调整的开销十分巨大,也是显得十分低效。原型中既包含有以前的设计经验——设计要素和设

计实例,又限定了设计模式。在进行基于原型的推理时,一方面,原型限定的设计模式降低了设计问题的复杂性,另一方面又可根据原型中的设计要素和设计实例来利用以前的设计经验。可以说,基于原型的推理综合了MBR和CBR的优点。

在实际生活中,设计思维方式是多种多样的,人们对客观世界的看法也是多种多样的。即使对同一设计要求,不同设计师的设计结果也各不相同。因此,不存在一个标准的设计思维过程。但是,设计思维过程与设计风格是紧密相关的。具有同一种设计风格的设计师对特定领域中的“设计世界”的看法是基本一致的,他们的设计思维方式也大体相同^[21]。而设计原型是对专业设计领域内种种设计要素的分类概括,原型中的联想、选择、评价方法所构成的设计推理机制描述了一类设计思维方式。因此,原型可以很好地模拟设计风格。

从知识工程角度来说,原型集知识表示及处理为一体,在结构形式上表现为一种面向对象的知识表示模式。随着面向对象技术的发展,常规性设计可望通过这样的系统实现完全自动化。而且,现有的AI技术如搜索、匹配、回溯、继承等可得到充分的应用。尽管在原型推理方法中隐含着联想、类比等正在探索之中的方法,但原型的引入明显地弥补了一直存在于常规性设计与创造性设计之间的断层。我们相信,随着对形象思维研究的深入^[21],在不久的将来一定能设计出一个真正具有创造性行为的ICAD系统。(参考文献共25篇略)

(接第71页)

- 利用V&V技术可对规格说明、概要设计及编码等正确性进行测试。

- 丰富的形式化支持工具。

形式化方法的使用对软件测试的自动化产生了直接的根本性影响。采用形式化方法,将使测试工具有如下特点:

- 测试用例自动生成,这一软件测试中最为重要的问题可望由形式化的方法的应用得以实现。

- 软件测试活动的自动化管理,包括测试定义,结果信息显示,测试库管理和回归

测试的控制等。

- 测试过程自动化。如测试用例的选择,期望结果与实际结果的比较等。

软件测试尚待解决的问题是:如何寻找具有强语义的形式化方法?如何在软件开发的前端应用形式化方法与技术?开发何种支持形式化的测试工具?形式化方法是否支持原有的特定方法及语义?如何寻找既形式化又易于理解的测试工具?解决了这些问题,软件测试才有希望达到可行性、经济性、健壮性及灵活性的目标。(参考文献共12篇略)