

一种面向并行工程的产品开发 过程关系模型^[1]

清华大学自动化系, 北京 100084 常天庆 熊光楞 李敬逸

摘要 为了处理产品开发过程由串行到并行的转变而引起的问题, 全面地表示且维持产品开发过程中各开发小组及小组成员间应满足的关系, 保证开发过程满足顾客需求, 本文分析了并行产品开发过程的特点, 提出了一种适合这些特点的产品开发过程关系的层次表示模型, 介绍了此关系模型的实现方法和它与并行工程环境中其它系统的集成关系, 给出了此关系模型在产品开发过程中的应用。

关键词 并行工程 约束满足问题 冲突仲裁

引言

并行工程以其在缩短产品开发周期、提高产品质量及降低产品成本等方面的巨大效益引起了众多企业的重视。并行工程技术的关键思想之一是尽可能早地、全面地考虑产品开发过程下游对设计阶段的要求, 以避免或减少建些开发过程进行到相当晚期才发现的且又必须使开发过程返回到设计初期才能弥补的设计错误。为了适应这种新的设计思想, 并行工程采用了团队工作方式 (Teamwork): 产品开发有关人员被划分为数个多功能小组 (Cross functional team), 每个多功能小组的成员包括总体、设计、功能分析、工艺及制造等部门的有关人员。其结果, 设计人员与下游有关人员的交流必然增多, 设计时考虑下游问题的种类和数量明显增加, 表现为设计时必须处理多功能小组间及小组成员间存在的大量相互依赖关系, 这些关系在本文中用约束 (Constraint) 的方式表示。有效地维持约束以及当出现违约现象时及时地向有关小组及小组成员通告违约信息是保证产品开发过程顺利进行的基本要求。为此, 必须建立一种产品开发过程关系模型来形式化表示约束及约束间的层次关系, 并将其在线应用于产品开发过程, 使产品开发过程的实际执行情况符合此关系模型。

本文第一部分介绍了该关系模型所采用的基础知识, 简要总结了产品开发过程关系模型相关领域的研究现状, 指出了现有技术的不足。第二部分分析了产品开发过程中约束的层次关系, 提出了关系模型的组织结构, 给出了关系模型的实现方法。第三部分介绍了关系模型所能处理的约束范围以及它在产品开发过程中的应用。第四部分为结论。

1. 约束满足问题介绍

我们称关系模型中的关系为约束。这里, 约束指产品开发过程变量间应该满足的相互制约、相互依

国家 863/CIMS 资助课题

本文收稿日期: 1996 - 05 - 09

赖关系。约束是关系模型的基本元素之一，是本文讨论问题的出发点，目前，习惯上将约束有关问题统称为约束满足问题。约束满足问题是人工智能的一个相当活跃的研究领域，它的研究成果可用于解决一些产品开发过程中出现的问题。

1.1 约束及约束满足问题

约束问题通常基于约束网络来处理，约束及约束网络提供了一种过程、产品及资源等因素相互关系的表示与处理方法。

约束网络由 n 个变量 $X = \{ X_1, X_2, \dots, X_n \}$ 和 m 个约束 $C = \{ C_1, C_2, \dots, C_m \}$ 组成。变量 X_i 的值域为 D_i ，每个约束 C_i 由两部分组成：变量集、 $V(C_i) = \{ X_{i1}, \dots, X_{ij} \}$ 和关系 $R(C_i) = R(X_{i1}, \dots, X_{ij}) \subseteq D_{i1} \times \dots \times D_{ij}$ 。约束网络的全部解可表示为：

$$R(Z) = \{ (X_1 = x_1, \dots, X_n = x_n) \mid \bigwedge_{C_i \in V(Z)} R(C_i) \subseteq R(C_i) \} \quad (1)$$

其中 $\overline{X_U} = \{ X_U = (X_{U_1}, \dots, X_{U_m}) \mid \exists X_k, \overline{X}$ 是 X_U 的扩充 $\}$ 表示变量子集 $U = \{ U_1, \dots, U_m \}$ 在关系 k 上的投影影响^[1]。通过求解运算，可以确定约束网络是否有解、解决如何找出约束网络中单个或全部解等多方面的问题。

解决约束满足问题的一般方法是回溯搜索 (Backtracking)，即在求解过程中，以某种顺序遍历变量，在每次给一个变量赋值后，都检查该变量与已赋值变量间的约束满足性。若满足，则继续对下一个变量赋值；否则，选择该变量值域中其它的赋值，此时若发现值域中没有其它的赋值可选，则进行回溯。这种方法的效率很低，为此人们对它作了若干改进，并提出相应的算法，如“选择回溯”、“面向从属关系的回溯”等。另外，人们发现，在实际开始搜索之前，通过预处理可以消除回溯发生的许多因素。为此，人们也作了许多工作。

1.2 约束满足问题有关研究情况

约束满足问题是解决人工智能有关问题的有效方法之一。它在解决规划、调度、时序推理等问题中得到了广泛的应用。下面介绍一些约束满足问题用于产品开发过程方面的研究情况。

1.2.1 约束满足问题用于规划^[2]

Allen 提出用定性时间间隔处理规划问题^[3]。他认为，规划中的时间关系可用 $\{b, bi, m, mi, o, oi, s, si, d, di, f, fi, eq\}$ 13 种基本约束表示。基于这些基本约束可将约束问题用网络描述，其中，结点代表事件，有向边代表相关两事件之间的约束，它用基本约束的集合表示。在此网络的基础上，可将解规划问题看做是解约束满足问题。Vilain & Kautz 提出用事件端点之间的约束关系来处理规划问题^[4]。此时，端点之间所有的可能关系为： \sim 、 $<$ 、 $=$ 、 $>$ 、 \leq 、 \geq 及 \neq ，称为 7 种基本关系。此时约束网络中的结点是事件端点，有向边是 7 种基本关系的某种析取。一般的处理约束满足问题的算法原则上都适用于以上两种形式表示的网络，但具有效率不高的特点，算法的复杂度不低于 $O(n^3)$ 。peter van Beek 提出了一种改进算法^[5]：对于一个规划问题，当其用时间间隔表示的约束网络可转化为用端点表示的约束网络时，这种算法的复杂度为 $O(n^2)$ ，而后又进一步说明此算法也能提高一般规划问题的计算效率。清华大学的张钹等人提出了一种当约束是单成份时复杂度为 $O(n^2)$ 的算法^[6]，特别是此算法还可处理时间间隔限定的规划问题。

1.2.2 约束满足问题用于产品设计^{[7][8]}

设计人员将属于自己的设计领域的变量及变量之间关系表达成约束网络，利用此网络，可以计算出某些设计参数，判断某一设计结果正确与否并在不正确时给出修改建议。此时约束网络表示内容的重点只是属于设计者自身的设计问题，采用约束网络的目的只是为了提高设计工作效率，设计人员之间的约

束关系并未得到重视。

1.2.3 约束满足问题用于冲突仲裁

Sycara 在他的协商模型^[9]中将基于约束的仲裁作为冲突仲裁方案之一，他认为有些冲突只有通过改变或放松约束来解决。Qiang Yang 提出了一种基于约束的解决规划问题冲突的方法^[10]，将所有可能出现的冲突均表示成约束，从全局的角度考虑冲突，提出了一种有效算法，可通过搜索提前指出规划的不一致现象、发现死路、并有效地避免冲突发生。

1.2.4 CERC 的约束管理系统

West Virginia University 的 CERC 开发出一种旨在用于并行工程的通用约束管理系统^[11]，此约束管理系统由一组提供明确功能的模块组成。其中，一个模块负责约束的创建和修改；另一个用于约束管理，即保持约束网的一致性，估算约束和在必要时传播变量值。另外，还有一个约束传播规划器模块和一个用于执行区间计算 (interval mathematics) 的模块。以上四个模块与一个数学机 (Mathematics Engine) 相联，此数学机用于执行所有必要的计算。

这些技术在用于并行产品开发时均有不同程度的不足之处。首先，它们只从某个侧面解决问题 (规划问题、设计问题)；其次，对开发过程中某些约束缺乏有效的表示方式 (如复杂机电产品中电缆线的走向等)；再者，功能模块之间的约束关系未得到重视；最后，没有提供大线工作功能。

本文的目的旨在提出一种面向并行产品开发过程的关系模型以集成处理产品开发过程各个侧面的约束问题、全面地表示产品开发过程中出现的约束并实现在线监控产品开发过程的执行情况。

2. 关系模型的结构及实现

2.1 约束的表示

产品开发中遇到的变量有以下五种：连续变量 (如 18, 36E01)，矢量 (如：23, 12, 45)，离散变量 (如：3, 5, 20)，符号变量 (如：aluminum, red)，逻辑变量 (如：true, false)，定性变量 (如：某时间区间)。

为全面表示并行产品开发过程中的约束关系，约束应有以下形式：

方程：变量间的关系用等号或不等号表示。如： $L1 + L2 = L$ ， $X2 * Y = 5$

推理规则：表示一些符号约束。如：IF $X > 0$ THEN $Z = X$ ELSE $Z = Y$

定性约束：定性表示变量间的关系。如： $D1 \{b, m\} D2, D3 \{eq\} D4$

隐式约束：黑匣子约束。有确定的输入输出关系，但并不关心匣内的细节。

范围限制：变量的取值范围。如：某一材料；区间内，区间外。

2.2 约束网络的层次结构

2.2.1 层次划分

产品开发过程中的约束可划分为以下层次：

多功能小组间的约束：对于复杂产品的开发过程，多功能小组往往不只一个，这些小组可能跨地区分布，且采用异构计算机环境 (指硬件环境不同，工具软件不兼容等) 工作，如何建立、维持这些小组间的约束，以保证小组间协调工作是并行工程要解决的首要问题。

多功能小组内的约束：代表设计、工艺、制造及用户等部门的多功能小组成员间存在大量的约束关系，组建多功能小组是并行工程的核心内容，有效地处理这些约束关系是保证并行工程效益得以有效体

现的基本要求。

设计部门内部的约束：多功能小组内的设计人员可能各有分工。例如：机械产品设计时有几何造型人员、CAD 人员、产品功能分析人员等，尽早考虑这些约束也是减少大的设计返工的有效手段。

属于同一分工的设计人员之间的关系：实际工作中、同一分工可能需要多个设计人员共同完成。例如：动力学工程师与机械工程师的关系。

单个设计人员对应的约束：设计人员可将要设计的产品的参数间的关系用约束网络表示，此网络有利于提高设计效率。应该指出，此层本身又是一个多层约束网络。

2.2.2 约束集

将约束网络按照一定的规则划分为一系列约束集是实现约束网络层次结构的有效方法。约束集是一组有逻辑联系的约束。一个约束集可以对应一个多功能小组、一个 perspective (如：Design, Manufacturing)，一个设计人员等。如此，整个约束网络被划分为一系列相互之间有约束关系或继承关系的约束集。

形式化的层次约束网络如图 1 所示。模型的层次关系通过约束集的包含程度体现，总体组、小组 1、小组 2 及小组 3 对应的约束集在第一层，它们之间的约束用 $Cons. _ tx$ 表示；各组的设计部门、工艺部门、制造部门及用户对应的约束集在第二层，它们之间的约束用 $cons. _ px$ 表示；第一部门还可再细分形成第三层，如设计部门，可分为 CAD、造型、分析等对应的约束集，它们之间的约束用 $cons. _ dx$ 表示；CAD 对应的约束集分解为第四层 ...。

图 1 层次网络示意图

2.3 实现

我们采用面向对象的方法设计关系模型。使用面向对象的设计方法可以得到模块化程度高、可维护性和可复用性好的程序；另外，关系模型是并行工程支持软件的一部分，与它相关的软件均是面向对象的结构，为了使其与相关软件间有友好的接口，也应选择面向对象的设计方法。

2.3.1 类的划分

采用面向对象的方法设计此模型，模型核心部分的类为变量类、约束类及约束集类。图 2 为用 IDEF4 格式表示的类图。图 (b)、图 (c) 及图 (d) 分别为变量类图、约束类图及约束集类图。图 (a) 为 IDEF4 类图的符合说明，类图中双线上边的无素称为特性，特性可分为属性、例程、槽值、函数和过程，分别用符合 ?、%、@、\$ 及 # 标注在特性名的前面；双线下元素为类名。

2.3.2 关系模型管理系统

关系模型由关系模型管理系统管理，关系模型管理系统将关系模型的思想具体化，以一种软件的方式体现出来，使关系模型能应用于实际的产品开发过程。我们开发的关系模型管理系统具有如下功能：

创建、添加、修改、删除及检索等编辑功能。

约束网络预处理功能，以保证开发过程所对应的约束网络有解。

用适合并行工程特点的算法处理约束满足问题（包括一致性处理、估算变量值及约束传播等）的功能。

?	属性，可能是槽值，也可能是函数。
%	例程，可能是函数，也可能是过程。
@	槽值。
\$	函数，每次调用均有返回值。
#	过程，调用时无返回值，但有负作用。
<hr/>	
	; 类名

(a) 类图符号说明

@name	; 约束名。
@form	; 形式，如：方程、推理规则等。
@representation	; 约束表示。
%in-set	; 所属的约束集名。
@owner	; 拥有此约束的部门。如设计部门、制造部门等。
@variable	; 约束变量。
@description	; 描述。
<hr/>	
Constraint	; 约束类

(c) 约束类图

@name	; 变量名
@designr-value;	变量值。开发过程中开发人员确定的具体变量值，可能为空。
\$ net-value;	约束网络估算值。通过计算得出，随设计过程的推进而不断具体化。
\$ constrained-by	; 相关约束。
@owner;	拥有此变量的部门。如设计部门、制造部门等。
<hr/>	
Variable	; 变量类

(b) 变量类

@name	; 约束集名
@owner;	拥有此约束集的部门。如设计部门、制造部门等。
%parent	; 此约束集的父约束集名。
%chiden	; 此约束集拥有的子约束集名。
%contains __ constraints;	属于此约束集的约束。
%contains __ variables;	属于此约束集的变量。
# check	; 一致性检查过程。
\$ valve __ variable;	估算变量值。
<hr/>	
Constraint __ set;	约束集类

(d) 约束集类图

图 2 类图

2.3.3 在并行工程软件环境中的位置

关系模型管理系统是并行工程项目协调系统的组成模块之一，项目协调系统还含有过程规划与调度、过程监控、冲突仲裁等模块。其中，过程规划与调度模块和监控模块为关系模型提供过程的实时信息，判断实际过程是否向着顾客需求推进；冲突仲裁模块解决关系模型的不一致现象及设计结果违约等方面的问题。

关系模型管理系统集成在 PDM 框架上，框架通过其对象管理、产品结构管理、产品配置管理等功

能模块为产品及产品开发人员提供了透明的可视化的存储空间,分布式的网络通信环境,产品数据及设计版本管理工具。此框架为关系模型的创建、实际设计数据的抽取提供了良好的基础环境。

与关系模型管理系统有关的还有产品开发过程建模、仿真与优化模块、此模块为关系模型提供过程约束信息,关系模型为建模结果的顺利实施提供技术支持,同时又是产品开发过程动态建模闭环系统的执行机构之一。

3. 关系模型在产品开发过程中的应用

从应用的角度考虑,评价关系模型的准则应包括:能处理的产品开发过程中约束的范围,可解决的产品开发过程中问题的种类,处理问题的及时性以及使用的方便程度等方面,下面以功能为主线就这些方面对本文提出的关系模型作一说明。

3.1 关系模型可处理的约束范围

并行产品开发过程必须考虑的约束可分为三类。首先是时间规划约束:在产品开发过程中,将开发人员的工作划分为一系列活动,所有活动之间应满足规定的时间关系;其次是工作站所对应任务间的约束:设计人员在设计时不但满足总体方案的要求,而且还要考虑到设计出的产品应能满足下游生产制造部门所提出的有关工艺、可装配性、可加工性及可维护性等方面的要求。设计过程中的详细设计与性能校核间也存在大量的约束;最后是产品部件间的约束:机械产品设计时部件间的几何约束、重量约束、质心约束,电气产品设计时部件间的电气参数传递关系等约束均是设计应考虑的内容。

本关系模型可表示、处理所有以上三类约束。例如:用活动 1 {S} 活动 2 表示活动 1 与活动 2 同时开始,活动 1 {m} 活动 2 表示活动 2 在活动 1 完成后立即开始等时间规划约束;工件的长度 L_0 与机床的最大加工长度 L_{\max} 间的约束为 $L_0 \leq L_{\max}$, 孔轴线与孔所在平面之间的夹角 θ_{\max} 等工作站所对应任务间的约束;机体(部件 1)为电缆线(部件 2)预留的路径为 (P_1, P_2) , 约束 θ_{\min} 表示电缆线的拐角不能太小,即机体与电缆线间的约束。

3.2 用于方案可行性检查

方案可行性检查可通过网络预处理功能完成。网络预处理指在处理估算、传播等问题前对约束网络的一致性进行检查,找出不一致的原因,必要时调用冲突仲裁系统解决这些冲突现象,以保证约束网络有解。

此功能可用来解决许多开发过程中出现的问题。

产品开发过程中模型所确定的开发活动间的时间相互关系可能是一不实际要求,例如,既要求活动 1 与活动 2 同时结束又要求活动 2 使用活动 1 的输出结果,从而要求活动 2 在活动 1 后执行,类似问题在产品开发过程建模时可能未被发现。在模型执行前检测出它们并给出相应的提示是约束网络预处理的任务之一。

不同的开发人员可能从不同的角度对某一变量提出不同的相互矛盾的要求,当这些要求以约束的形式合并到全局约束网络时,网络预处理将会查出并以出错框的形式通告有关人员,提醒并督促有关人员及时解决这些问题。

3.3 协调产品开发过程

将解决约束满足问题的思想用于产品开发过程。利用经过预处理的约束网络,设计人员可计算出某些参数的大小或范围,发现某一决策对其它设计人员的影响,检查被提交的变量值是否正确,尽早发现

冲突。

3.3.1 估算

随着设计的细化，约束网络中的变量逐渐限定在更确切的范围甚至被赋予确定的值。通过约束网络运算，可以得出当前的变量值状态，设计人员可通过此功能来估算某些变量值的大小或区间。

3.3.2 约束传播

约束传播即推演出变量间在约束网络中所没有直接表示出的新约束。约束网络中某一变量的变化将会引起其它有关变量的变动，通过传播运算，可以确定出变化的受影响变量及判断出这些变量的受影响程度。利用此功能，可获得某一设计决策的作用范围及程度。例如，当某一设计人员对某一设计参数进行修改后，与此相关的参数所对应的设计人员均将得到此修改的通告信息，提示他们对这些参数作响应的修改。如此，避免了由于单方面修改而引起的大的返工。

3.3.3 冲突的检测及解决

冲突 (conflict) 是指设计人员在同一前提条件或同一目标要求下对某一事件作出相互矛盾的结论的现象。在产品开发过程中，冲突的起源主要有以下三种^[9]：

当某一开发人员提出的某一参数值使得另一开发人员不可能把其它参数值表示成一致集时，冲突产生。

对同一设计对象，属于不同部门的设计人员具有各自不同的设计目标或重点，目标或重点的不同经常引起冲突。

用不同的标准从不同的方面评价设计时，导致不一致的设计评价结论。这实际上是一个不同部门的局部最优不可能同时达到且有可能相互起负作用的问题。

对第一种冲突，可通过关系模型的实时检测功能来发现和解决。在产品开发过程中，当被评审的某一设计结果通过评审且被确认后，关系模型的冲突检测模块立即被触发，以检查此设计结果是否满足它对应的约束以及通过约束传播来检查它是否满足整个约束网络中的约束，即检查设计结果是否会引起不一致现象。如出现冲突，可通过修改与设计结果有关的参数来解决。当然，任一修改都可能又引起新的冲突，实际过程中，冲突的完美解决必须通过多次反复、权衡才能完成。

对第二种和第三种冲突，通过检测关系模型中各约束集间的约束关系是否满足来及时发现和解决。例如，设计飞机发动机叶片时，多功能小组中动力学工程师从叶片的功效角度考虑问题，认为叶片根部的厚度 (TH_a) 越薄越好，而机械工程师从叶片的强度角度考虑认为叶片根部的厚度 (TH_m) 越厚越好，约束网络中 TH_a 与 TH_m 的约束关系为 $TH_a = TH_m$ 。工作过程中，动力学工程师和机械工程师分别按各自的领域的规律设计有关参数，通过 CS_a 可以得出 TH_a 的一致取值范围为：

$$TH_{aL} \quad TH_a \quad TH_{aH} \quad (2)$$

通过 CS_m 可以得出 TH_m 的一致取值范围为：

$$TH_{mL} \quad TH_m \quad TH_{mH} \quad (3)$$

为保证 $TH_a = TH_m$ 成立，必须有：

$$TH_{aL} \quad TH_{mH} \quad \text{AND} \quad TH_{mL} \quad TH_{aH} \quad (4)$$

关系模型管理系统一旦发现 (4) 式不可能满足，立即通告有冲突发生。冲突解决可通过改变 CS_a 或 CS_m 中其它变量直到 (4) 式满足来实现，此过程需要反复地调用约束传播模块。

4. 结束语

综上所述, 本文提出的产品开发过程关系模型可有效地表示并行产品开发过程中的各类约束关系, 基于此关系模型开发出的关系模型管理系统可用于协调产品开发过程, 实时检测出过程中的冲突现象并解决某些冲突, 保证产品开发过程满足过程中的各类相互关系, 具有友好的人机接口并与并行工程其它软件系统有良好的集成关系, 是并行工程的一种关键技术。

参 考 文 献

1. Dechter R, Pearl J. Network-based heuristics for constraint-satisfication problem. *Artificial Intelligence*, 1988, 34: 1 ~ 38.
2. Stefik M. Planning with constraints (MOL GEN: Part 1). *Artificial Intelligence*, 1980, 16: 111 ~ 140.
3. Allen J F. Maintaining knowledge about temporal intervals. *Commun. ACM*, 1983, 26: 832 ~ 843.
4. Vilain M, Kautz H A. Constraint propagation algorithms for temporal reasoning. in *Processdings of AAAI*. Philadelphia, PA: 1986. 132 ~ 144.
5. Beek P V. Reasoning about qualitative temporal information. *Artificial Intelligence*, 1992, 58: 271 ~ 296.
6. 张钹. 问题求解理论及应用. 清化大学出版社, 1990.

A Concurrent Engineering-Oriented Relationship Model of Product Development Process

Chang Tianqing Xiong Guangleng Li Jingyi

Department of Automation, Tsinghua University, Beijing 100084

Abstract In order to deal with the problems caused by the change from serial product development process to concurrent product development process, represent completely and maintain the relationship among the cross function teams and the team members in product development process, assure the product development process of satisfying the requirements of customers, this paper analyzes the characteristics of concurrent product development process, presents a hierarchical relationship model of product development process which meets these characteristics, introduces the implementation of this relationship model and the integrative relation between the relationship model and the other systems in concurrent engineering environment, and describes the applications of the relationship model in product development process.

Keywords Concurrent engineering Constraint satisfaction problem Conflict resolution.