

一个面向大批量定制的重用配置方法

李妮娅^{1,2}, 张 健^{1,2}, 刘大有^{1,2}

(1. 吉林大学计算机科学与技术学院, 吉林长春 130012; 2. 吉林大学符号计算与知识工程教育部重点实验室, 吉林长春 130012)

摘 要: 为了提高产品配置的推理效率和降低企业内部变型产品的多样性,研究了面向大批量定制的重用配置技术. 本文针对用户定制需求在一定时空范围内呈现出的相似性和集中性特点,从已有的产品配置结果入手,完成个性化配置求解任务. 在基于广义产品结构的产品配置设计基础上,给出一种基于 XML 的配置结果描述方法;基于该描述方法,以用户定制页面为重用的最小处理单元,提出一种重用配置方法,并给出了算法描述. 实验结果表明,该方法能够随重用比例的提高而减少推理时间.

关键词: 产品配置; 广义产品结构; 重用配置; 配置推理

中图分类号: TP31 **文献标识码:** A **文章编号:** 0372-2112 (2011) 02-0383-06

A Product Reconfiguration Method Based on Mass Customization

LI Ni-ya^{1,2}, ZHANG Jian^{1,2}, LIU Da-you^{1,2}

(1. College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China;

2. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China)

Abstract: In order to improve the reasoning efficiency of PC (Product Configuration) and reduce the diversity of variant product in manufacture enterprise, the product reconfiguration technology in PC for manufacturing is studied. In view of the similarity and centrality of user's requirements in some conditions, the reconfiguration study begins from the existing configuration results to resolve the customized configuration problem. On the basis of PC design based on generic product structure, a description method of configuration results with XML format is proposed. Based on this work, a product reconfiguration method and its key algorithms are given, in which the configuration page is used as the minimal processing unit. The experiment result shows that the above reconfiguration method can reduce the reasoning time with the reconfiguration proportion increasing.

Key words: product configuration; generic product structure; product reconfiguration; configuration reasoning

1 引言

大批量定制生产方式因其能够适应客户需求多样的市场要求而被制造企业普遍认可和接受. 产品配置技术是实现这一生产方式的关键技术,它根据领域知识预先定义产品配置模型,在不创造出新的组件的前提下,快速形成满足用户定制要求的变型产品.

产品配置过程中,不同的用户可能对产品配置模型提出相同或近似的定制要求. 对于深度定制的产品来说,如果能够尽可能的重用已有的配置求解方案和变型产品,无需或只需部分改造/设计就获得一个满足用户定制要求的产品,其效率和成本都会优于重新配置一个变型产品,这就是重用配置所要研究的内容. 由于客户的个性化需求在一定时空约束下往往呈现出相似性和

集中性的特点,因此重用配置研究具有很好的应用价值,能够为企业带来减少资源冗余和提高生产效率等多方面的益处.

早期的重用配置研究主要集中在处理配置结果本身,只限于从配置结果原型中过滤出那些能够满足当前定制要求的组件和组件结构,而不能解决原型无法满足的那部分定制要求的配置问题^[1]. 近年来,研究转移到基于配置推理机制解决重用配置问题,求解能力大大提高,求解方法也逐渐丰富,如基于 Petri 网^[2,3]、基于机器学习^[4]、基于 Agent^[5,6]方法等. 文献[2,3]以制造领域为研究载体,给出了支持重用配置的零部件装配关系表达方式,但由于知识表示的局限性,对客户定制要求的支持能力有限;文献[4~7]以计算机软/硬件系统领域为研究载体,给出了支持重用配置的模型接口定义方法,

但不适用于制造领域复杂配置问题的重用配置任务。

本文旨在面向制造业领域,在前期产品配置研究工作^[8~10]基础上研究重用配置问题,给出一种基于广义产品结构的重用配置方法。理论分析和实验表明,重用配置求解与直接配置求解相比,能够大大提高推理效率。

2 基于广义产品结构的产品配置设计

在前期产品配置研究工作中,文献^[9,10]给出了一个基于广义产品结构的产品配置模型和针对这个模型的产品配置推理策略。

产品配置模型的结构信息采用广义产品结构树来表示。与产品结构树相比,广义产品结构树不是描述单一产品的装配关系,而是描述一组具有共同形状特征和功能、以一定市场为目标的产品集合的装配关系^[11,12]。广义产品结构树中那些用于表达一组零部件对象的似零部件节点被称为广义零部件。

产品配置模型的配置逻辑信息包括可定制信息、约束信息和定制接口信息。

配置参数 *param* 表示模型中的可定制信息, $param = (name, value, weight, default, scope, type, mapped)$ 。各要素分别表示参数名、参数取值、取值来源、取值的缺省表达式、取值范围、取值类型和参数对应的配置特征。推理过程中, *weight* 有四个有效的取值: -1 表示 *value* 为空; 0 表示 *value* 来自于 *default*; 1 表示 *value* 来自于配置规则的推理结果; 2 表示 *value* 来自于用户输入。规定配置参数的赋值遵循高权值优先的原则, 当且仅当数据的权值高于参数当前的权值时, 才对参数进行赋值操作。

配置规则 *rule* 表示模型中零部件之间的约束关系, $rule = (name, InParams, successful, logic, OutParams)$ 。各要素分别表示规则名, 规则的输入参数集合, 逻辑约束、执行结果和输出参数集合。规定 *InParams* 不能为空。如果逻辑约束执行成功, *successful* 的值为 *TRUE*; 否则为 *FALSE*。 $InParams \cap OutParams = \phi$ 。

配置接口 *portal* 表示模型的用户定制接口, $portal = (param, pageNum, siteNum, \dots)$ 。 *param* 表示可以直接接受用户定制要求的参数, 用户可以通过定制界面直接为其赋值。 *pageNum* 和 *siteNum* 是两个界面控制属性, 表示配置接口在第 *pageNum* 个定制页面的第 *siteNum* 个位置上显示。

用户的定制要求通过配置接口记录到 *param* 当中, 一个配置参数至多只能从属于一个配置接口。用户的定制活动被设计为在一页一页的定制页面上分阶段完成, 每个页面包含多个定制接口, 用户每提交一页,

推理计算就执行一次, 从而达到分解配置任务的目的。

产品配置推理策略根据广义产品结构树的先根顺序处理每个广义零部件, 根据用户的定制要求不断触发配置规则, 获得配置参数的取值, 最终给出配置方案。广义零部件的配置推理基于数据驱动的深度优先图搜索策略来完成, 推理算法 *ConfigureGPart* 如下所示:

算法 1 *ConfigureGPart* (*TriggParams*; *ChangedParams*)

// 输入参数 *TriggParams* 是当前能够触发规则的配置参数集合;
// 输出参数 *ChangedParams* 是被规则更新了取值的配置参数集合。

- ① [创建被触发的规则集合] 对于当前广义零部件对象的每个规则 *r*, 如果 $TriggParams \cap r.InParams \neq \phi$, 那么 *r* 添加到被触发的规则集合 *SatisRules* 中。
- ② [循环/递归出口] 如果 *SatisRules* = {}, 算法结束。
- ③ [选择过程] 根据评估函数 $f(r) = |r.InParams|$, 从规则集合 *SatisRules* 中选取出函数值最大的第一个规则 *r*, 符号“|”表示计算集合的元素个数。
- ④ [执行过程] 执行规则 *r*。
- ⑤ [深度递归调用] 被规则 *r* 更新的输出参数集合 *TempParams* 成为新的触发规则的参数集合, 递归调用该算法 *ConfigureGPart* (*TempParams*; *ChangedParams'*), $ChangedParams = ChangedParams \cup TempParams \cup ChangedParams'$ 。
- ⑥ [循环调用] 以 *r* 起始的递归处理完毕后, 转到步骤 2, 处理 *r* 的兄弟节点。

易见, 一个参数的取值发生变化时, 它就具备了触发规则的能力。如果这个参数是某条规则的输入参数, 那么这条规则就可能被触发, 这条规则的逻辑就能够被执行。如果规则的逻辑执行成功, 那么输出参数的值也可能被更新, 规则成功被触发。这时, 被更新的输出参数也具备触发规则的能力, 配置推理也就逐渐向后传递。

3 基于 XML 的变型规范设计

重用配置与配置相比, 其本质区别在于重用配置通过尽可能少的更新一个已经存在的配置结果来满足用户的定制要求, 因此重用配置获得的配置结果是满足用户定制要求的配置结果集中那些与重用原型最为相似的配置结果, 而不仅仅是满足用户定制要求的任意配置结果。

配置结果是基于用户定制要求和产品配置模型进行推理求解而产生的数据信息, 它的表达方式和信息容量对重用配置研究有很大的影响。配置结果一般包含模型的结构信息和基于用户定制要求获得的参数信息, 这些信息可以采用 XML 格式描述, 图 1 基于 UML 形式给出了配置结果中主要的 XML 元素 (XML Element) 和属性 (XML Attribute), 并规定了元素之间的嵌套关系。

由图 1 可见,一次配置任务(即 topfamily)可能产生多个配置结果(即 gpfamily),每个配置结果包含三种信息:当前节点(即 persistInfo)、配置参数(即 parameter)和子件节点(即 component).子件节点相当于一个子配置结果,它采用与父节点同样的方式描述详细信息.

基于该描述方式,图 2(b)给出了一个配置结果文档.图 2(a)是当前配置结果文档对应的产品配置模型.配置结果经过用户确认后会转化为产品结构的方式存储,即成为变型产品,因此配置结果也被称为变型规范.由图 2 可见,变型规范按照产品配置模型的树形结构分层组织数据,记录模型的结构信息和参数信息,参数的取值全部或部分被确定.用户/技术人员查看变型规范时,变型规范中的结构信息转化为树形结构显示,当前节点下的参数信息转化为列表形式显示.

4 重用配置设计

本文的重用配置设计基于这样的应用路线:假设当前用户已经选定了一个比较接近自己定制要求的产品原型,那么定制开始后,产品配置系统会首先根据产品原型对应的参数取值初始化定制页面,如果用户接受了这样的定制建议,则直接重用当前页面涉及的所有配置参数取值,否则,重新进行推理计算.在这个路线里,确定可重用的配置参数集合是设计的关键点,下面给出一种解决方法.

产品配置模型中,一个广义零部件可能被设计为包含多个用户定制页面,每个页面上的配置接口会触发一些配置规则执行业务逻辑,进而修改相关配置参数的取值,定制页面和这些配置参数之间的关系如表 1 所示.

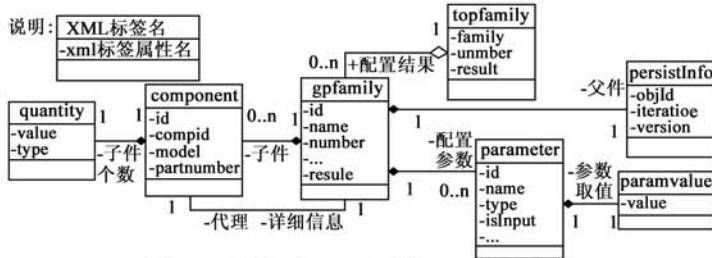


图 1 配置结果中XML元素的UML定义图

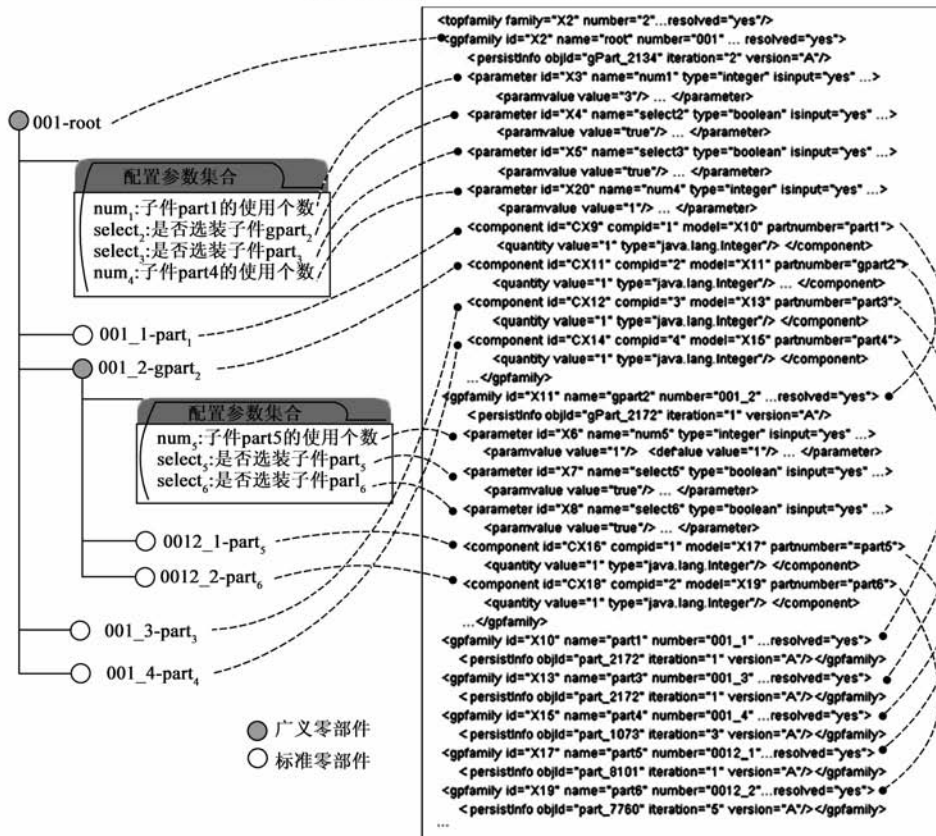


图 2 产品配置模型与变型规范文档的关系示意图

表 1 配置接口参数相关集合符号的含义与关系

页面顺序	当前定制页面的配置接口参数集合	当前定制页面赋值更新的配置参数集合之间的关系	备注
1	$P^1 = \{p_1^1, p_2^1, \dots, p_{n_1}^1\}$		— P^i :第 i 个定制页面的配置接口参数集合, $P^i \cap P^j = \phi, j = 1, \dots, k, j \neq i$.
2	$P^2 = \{p_1^2, p_2^2, \dots, p_{n_2}^2\}$		— n_i :第 i 个定制页面的接口个数;
...	...		— p_j^i :第 i 个定制页面的第 j 个位置显示的配置接口参数, $1 \leq j \leq n_i$;
k	$P^k = \{p_1^k, p_2^k, \dots, p_{n_k}^k\}$		— Q^i :第 i 个定制页面可能赋值更新的配置参数集合, $P^i \subseteq Q^i$; — $i = 1, \dots, k$.

定制页面 i 可能会更新的配置参数包括两种:配置接口参数和被配置接口参数触发的配置规则的输出参数,这两种参数构成定制页面可能更新的配置参数集合 Q^i ,它同时也是接口参数不变时可重用的配置参数集合.回顾配置推理算法 *ConfigureGPart* 易知,规则的评估函数值和值排序是稳定不变的,那么指定模型中任意配置接口参数 *param* 基于算法 *ConfigureGPart* 可能触发的配置规则和触发顺序也总是稳定不变的.因此,也就能够构造出可能被 *param* 触发的配置规则的输出参数集合,记作 Q_{param} ,简称为基于 *param* 被更新的配置参数集合.综上,定制页面 i 可能更新的配置参数集合 Q^i 的构造方法描述如式(1):

$$Q^i = \left(\bigcup_{j=1, \dots, n_i} Q_{p_j^i} \right) \cup P^i \quad \begin{cases} (1) P^i = \{p_1^i, p_2^i, \dots, p_{n_i}^i\}, \text{定制页面} \\ i \text{ 的所有配置接口参数集合.} \\ (2) Q_{p_j^i}, \text{基于配置接口参数 } p_j^i \text{ 被更} \\ \text{新的配置参数集合.} \end{cases} \quad (1)$$

从推理的上下文关系看, Q^i 中配置参数可以分为两个部分, $i = 1, \dots, k$:

(1) S_1^i :属于 Q^1, \dots, Q^{i-1} 且不在集合 P^i 中的配置参数集合,即:

$$S_1^i = \left(\bigcup_{j=1, \dots, i-1} Q^j \right) \cap Q^i - P^i$$

(2) S_2^i :不满足 S_1^i 条件的配置参数集合,即: $S_2^i = Q^i - S_1^i$.

易知, S_1^i 中的配置参数是 Q^i 中那些在第 i 个定制页面之前就可能已经被用户指定取值的定制接口参数和可能已经被推理计算过的非定制接口参数, S_2^i 中的配置参数是 Q^i 中那些在第 i 个定制页面时才可能被用户指定取值的定制接口参数和才可能第一次被推理计算的非定制接口参数.

基于 S_1^i 和 S_2^i 的定义,在重用配置过程中,当定制页面 i 上的定制要求发生变化时,遵循如下规则:

[重用规则] 如果当前准备更新 S_1^i 中的配置参数 p ,则基于 p 现有的数据进行配置推理;如果当前准备更新 S_2^i 中的配置参数 q ,则基于 q 定义的初始/缺省值进行配置推理.

显然,重用规则中的配置参数 $p \in S_1^i$ 现有的数据是来自于页面 $1, \dots, i-1$ 的配置推理结果/重用结果.

经过对 S_2^i 的定义进行公式推导和参数加权赋值检验, S_2^i 可以简化为式(2)的形式来构造,推导过程不予赘述.从而 S_1^i 亦容易求得.可见,若 Q 可求,则重用配置推理过程可以实现.

$$S_1^i = Q^1 \quad (2)$$

$$S_2^i = Q^i - \left(\bigcup_{j=1, \dots, i-1} S_2^j \right)$$

5 重用配置算法

由式(1)知,为构造集合 Q ,首先构造集合 Q_p .如下给出一种配置接口参数 p 的 Q_p 集合构造算法:

算法 2 *BuildCParamQ*($p; Q_p$)

// 输入参数 p 是当前准备处理的配置参数;输出参数 Q_p 是可能基于 p 而被更新的参数集合.

- [获取可能触发的规则集合] 对于当前广义零部件中没有被访问过的每个规则 r ,如果 $p \in r$. *InParams*, r 添加到规则集合 *QRules* 中.
- [循环/递归出口] 如果 *QRules* = $\{\}$,算法结束.
- [选择过程] 根据评估函数 $f(r) = |r$. *InParams* |,从规则集合 *QRules* 中选取函数值最大的第一个规则 r ,符号“|”表示计算集合的元素个数.
- [递归调用] $Q_p = Q_p \cup r$. *OutParams*,对每个 $p' \in r$. *OutParams*,依次调用算法 *BuildCParamQ*($p'; Q_p$), $Q_p = Q_p \cup Q_{p'}$.
- [循环调用] 转到步骤 2,处理 r 的兄弟节点.

基于算法 *BuildCParamQ* 和式(1),不难求得每个定制页面 i 的 Q^i 集合,进而基于式(2)求得 S_2^i 集合.

基于 S_2^i 和重用规则,广义零部件的重用配置算法可以描述为:

算法 3 *ReconfigureGPart*($i; ChangedParams$)

// 输入参数 i 表示定制页面的编号;输出参数 *ChangedParams* 是被规则更新了取值的配置参数集合.

- [重用原型数据] 如果 P^i 中的参数数据与配置规范原型相比没有发生变化, *ChangedParams* = $\{\}$,算法结束;否则,执行步骤 2.
- [重新推理计算] 将 S_2^i 中的配置参数恢复为模型的初始/缺省状态,基于 P^i 记录的用户定制要求构造用于触发规则的配置参数集合 *TriggParams*,调用算法 *ConfigureGPart*(*TriggParams*; *ChangedParams*),算法结束.

假设当前配置模型的定制页面数为 n ,根据算法

ReconfigureGPart 可知,若重用的定制页面数为 i ,那么算法 ConfigureGPart 被调用的次数为 $n - i$ 次.令 $x =$ 重用的定制页面数/页面总数, $y =$ 重用配置下 ConfigureGPart 的执行次数/全配置下 ConfigureGPart 的执行次数,则有 $y = 1 - x, 0 \leq x \leq 1$.

在当前知识表示和配置推理策略下,定制页面可能更新的配置参数集合 Q 是一个静态的数据集合,它只与产品配置模型有关,任何推理状态下都不会发生改变.因此,不妨在建模完成后,基于产品配置模型的树形结构自上而下先根遍历,依次访问每个广义零部件节点,然后基于算法 BuildCParamQ、式(1)和式(2)为节点下的每个定制页面构造集合 S_2 ,并保存结果,这样避免了以后重复工作.定制阶段时,当用户指定重用配置的产品原型后,配置推理引擎基于广义产品结构树的先根顺序依次为每个广义零部件构造定制页面,产品原型的结构信息和参数信息作为当前定制页面的初始信息显示给用户.用户一旦提交当前页面的定制要求,产品配置引擎就调用算法 ReconfigureGPart 进行处理,重用配置从而得以实现.

6 应用实例与实验结果分析

本节给出一个应用实例来解释重用配置推理的设计思想.图3给出一个广义零部件的配置逻辑关系描述图,它是根据配置规则与配置参数之间的输入输出关系而建立的一个虚拟有向图,简称为配置推理图,图中椭圆形节点是配置参数节点,矩形节点是配置规则节点^[9,10].配置推理策略和重用配置下集合 Q 的构造方法正是基于这个有向图进行的.

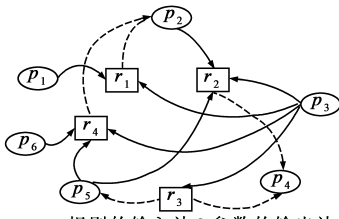


图3 某广义零部件的配置推理图

由图3可知,各个规则的输入输出参数及评估函数值如表2所示.

表2 规则静态信息表

规则	输入参数集合	输出参数集合	$f(r)$
r_1	p_1, p_3	p_2	2
r_2	p_2, p_3, p_5	p_4	3
r_3	p_3	p_4, p_5	1
r_4	p_3, p_5, p_6	p_2	3

设 p_1, p_3, p_6 是配置接口参数,由图3可知,经算法 BuildCParamQ, p_1 依次触发 r_1, r_2 ; p_3 依次触发 r_2, r_4, r_1, r_3 ; p_6 依次触发 r_4, r_2 . 于是得到 $Q_{p_1} = \{ p_2, p_4 \}$, $Q_{p_3} = \{ p_2, p_4, p_5 \}$, $Q_{p_6} = \{ p_2, p_4 \}$.

(1)假设 p_1, p_3, p_6 依序分布于三个定制页面上,那

么集合 P, Q, S_2 如表3所示.

表3 定制页面相关集合信息表

定制页面序号 i	P^i	Q^i	S_2^i
1	p_1	p_1, p_2, p_4	p_1, p_2, p_4
2	p_3	p_3, p_2, p_4, p_5	p_3, p_5
3	p_6	p_6, p_2, p_4	p_6

(2)假设 p_1 和 p_6 在第一个定制页面上, p_3 在第二个定制页面上,那么集合 P, Q, S_2 如表4所示.

表4 定制页面相关集合信息表

定制页面序号 i	P^i	Q^i	S_2^i
1	p_1, p_6	p_1, p_6, p_2, p_4	p_1, p_6, p_2, p_4
2	p_3	p_3, p_2, p_4, p_5	p_3, p_5

为了验证事前分析结果的正确性和说明重用配置在推理效率方面的优势,以一汽解放长春汽车厂某总成模型为例,给出算法 ReconfigureGPart 的实验结果.在配置参数(37个)和配置接口(22个)固定、配置约束基本稳定的情况下,该总成模型分别被设计为5、8和10个配置页面,然后逐个考察 i 个定制页面重用时算法 ReconfigureGPart 的执行时间之和.规定每个 i 值重复10次,每次重用的定制页面随机产生,取这10次结果的执行时间之和的平均值作为实验结果数据进行分析,分析结果如图4所示.

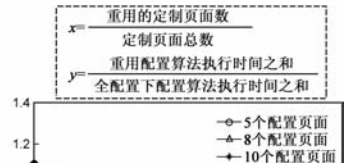


图4 重用配置实验结果分析图

图4中,数据基本沿函数 $y = 1 - x$ 的形态分布,这说明定制页面的重用比例越高,重用配置与全配置相比其执行时间就越短.此外,由于重用配置过程中配置接口参数与配置规范原型要发生比较(算法步骤1),因此与第5节算法分析的结论相比,函数存在一定的偏移.

7 结束语

一定时段和空间范围内,用户的定制要求往往会呈现出相似性和集中性的特点,表现为产品配置过程中不同用户对同一产品配置模型提出完全或大部分相同的定制要求.面对这些定制要求,重用已有的配置规范和变型产品比起重新配置推理来,显然更能够提高配置推理的效率和减小企业内部产品多样化.用户定制要求的相似性和重用性越高,重用配置的优点也就越突出.因此,重用配置研究具有很好的应用价值,是产品配置推理的一个有益补充.

本文主要研究了基于广义产品结构的产品配置中重用配置问题.设计了一种基于 XML 的配置结果描述方法,该方法将配置参数的推理结果和产品的结构信息结合在一起,具有平台无关和灵活性强的特点.给出了一种重用配置方法和它的关键算法描述,该方法以定制页面为重配置处理的最小单元,随用户定制页面重用程度的提高,按线性比例降低了产品配置的推理时间.下一步将基于已有的工作基础就变型结构的重用问题展开研究.

参考文献:

- [1] Stumptner M, Watawa F. Model-based reconfiguration [A]. Proceedings of Artificial Intelligence in Design [C/OL]. Lisbon, Portugal, 1998. <http://citeseerx.ist.psu.edu/view-doc/summary?doi=10.1.1.-47.9391>, 2008-08-22
- [2] Zhang L, Rodrigues B. Modelling reconfigurable manufacturing systems with coloured timed petri nets [J]. International Journal of Production Research, 2009, 47(16): 4569 - 4591.
- [3] 林春深, 唐晓强, 段广洪, 等. 基于 Petri 网的产品重构描述及规划 [J]. 清华大学学报(自然科学版), 2007, 47(5): 655 - 658.
Lin Chun-shen, Tang Xiao-qiang, Duan Guang-hong, et al. Product reconfiguration modeling and planning research using Petri nets [J]. J Tsinghua Univ (Sci & Tech), 2007, 47(5): 655 - 658. (in Chinese)
- [4] Wildstrom J, Stone P, Witchel E, Dahlin M. Machine Learning for on-Line Hardware Reconfiguration [C/OL]. Proceedings of the IJCAI '07 Workshop on Configuration, 1113 - 1118. <http://www.ijcai.org/papers07/-Papers/IJCAI07-180.pdf>, 2007.
- [5] Jokinen J, Jose L M L. Agent-based control of rapidly reconfigurable material handling system [A]. 12th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2007 Proceedings [C]. Patras, Greece, Sept. 25 - 28, 2007. 1217 - 1224.
- [6] Blanc P, Demongodin L, Castagna P. A holonic approach for manufacturing execution system design: An industrial application [J]. Engineering Applications of Artificial Intelligence, 2008, 21(3): 315 - 330.
- [7] 冯锦丹, 战德臣, 聂兰顺, 等. 基于模式的业务构件代码生成方法 [J]. 电子学报, 2008, 36(12A): 19 - 24.
Feng Jin-dan, Zhan De-chen, Nie Lan-shun, et al. Pattern based code generation method for the business component [J]. Acta Electronica Sinica, 2008, 36(12A): 19 - 24. (in Chinese)
- [8] Niya Li, Dayou Liu, Jian Zhang. A design of product configuration manager based on weight constraint parameter value in

manufacture [A]. Proceedings of the 5th International Conference on Control and Automation, ICCA05, vol. 6 [C/CD]. Budapest, Hungary, 2005. 1056 - 1061.

- [9] Niya Li, Dayou Liu, Jian Zhang, Yuhua Chen. A method of rule match conflict resolution for product configuration in manufacture [J]. International Journal of Computer Integrated Manufacturing, 2009, 22(3): 210 - 224.
- [10] 李妮娅, 刘大有, 张健. 制造业中基于 UML 的产品配置研究 [J]. 吉林大学学报(工学版), 2009, 39(3): 759 - 763.
Li Ni-ya, Liu Da-you, Zhang Jian. UML-based product configuration in manufacture [J]. Journal of Jilin University (Engineering and Technology Edition), 2009, 39(3): 759 - 763. (in Chinese)
- [11] Jaring M. Engineering reconfigurable product families-architecting the variability infrastructure of a product family on-chip [A]. Proceedings of COMPAAC'06, vol. 1 [C]. IEEE Computer Society, Washington, DC, USA, 2006. 95 - 102.
- [12] Jiao J, Simpson T W, Siddique Z. Product family design and platform-based product development: a state-of-the-art review [J]. Journal of Intelligent Manufacturing, 2007, 18(1): 5 - 29.

作者简介:



李妮娅 女, 1979 年 11 月生于黑龙江省五大连池市. 博士, 讲师. 主要研究方向为专家系统, 产品配置, 产品数据管理等.
E-mail: liny@jlu.edu.cn



张健(通讯作者) 男, 1977 年 11 月生于吉林省四平市. 博士, 讲师. 主要研究方向为工作流, 产品配置等.
E-mail: jzhang@jlu.edu.cn



刘大有 男, 1942 年 7 月生于河北省乐亭县. 教授, 博士生导师. 主要研究方向为知识工程与专家系统, 分布式人工智能, 多 Agent 系统等.
E-mail: dyliu@jlu.edu.cn