一种基于概念格的软件过程改进算法*)

强　宇[1,2]　胡运发[1]

(复旦大学信息工程与技术学院　上海 200433)[1]　(蚌埠坦克学院计算机室　蚌埠 233013)[2]

摘　要　概念格是一种优良的形式化分析工具,其具有的层次性和直观性,使之广泛应用于人工智能、数据挖掘、知识提取等领域。软件过程改善是软件工业化生产的一个关键,将概念格引入到了软件过程改善研究,提出了一种基于概念格的软件过程改进方法,并做了实验分析,证明了采用此方法对软件过程的改进具有推动作用。

关键词　概念格,软件过程改进,度量元,关联规则提取

Concept Lattice-based Algorithm for Improving Software Process

QIANG Yu[1,2]　HU Yun-Fa[1]

(College of Information Engineering and Technology, Fudan University, Shanghai 200433, China)[1]

(Computer Research Institute, Bengbu Tank College, Bongbu 233013, China)[2]

Abstract　As an kind of excellent FCA tool, concept lattice has been applied in AI, data mining, and knowledge extraction field because of its hierarchy and instinction. Software process improvement is key of software industry. Introduced the concept latttice into software proless research, presented a concept lattice method for improving software proless, and did an experimental analysis. Experiment demonstrates that the method is valid for propelling the software proless improvement.

Keywords　Concept lattice, Software process improvement, Metric, Association rule extraction

1　引言

1.1　软件开发过程的改进关键

目前在软件过程改进领域,国际上主要有三个流派,分别是(1)ISO9000,负责制订软件质量系统系列标准;(2)ISO-spice,ISO15504,是关于软件过程的一组标准,与第一种相比,它们不仅考虑软件过程,而且考虑人、技术、管理规程、客户支持和质量以及软件开发和维护规程等问题。(3)CMM/TSP/PSP4,是由美国卡内基·梅隆大学软件工程研究所组织开发的软件过程,包括侧重大中型软件企事业中有关软件过程的宏观管理和项目管理的 CMM(软件过程成熟度模型)[13-15],侧重于群组开发的 TSP(群体软件过程),侧重于小型企事业中有关软件过程的微观优化和个体技能的提高的 PSP(个体软件过程)。其中 CMM/TSP/PSP 在企业界和商业界得到了广泛认可和应用。

中小企业在企业文化、企业规模、软件生产的创新能力、动态变化能力、技术革新能力等方面具有不同于大型企业的企业特征,具体表现有几个以下方面[1]:

(1)企业规模

资金规模小。大型企业与中小企业进行过程改进的核心活动基本上一样,但中小企业必须精打细算,利用有限的资源做最需要的改进。

生产领域有限,产品用户有限。基于用户对象角度,软件开发分为两种类型:一是特性软件开发,为特定的用户定制软件产品;二是通用软件的开发,由于中小企业的人员和资金规模的限制,更多的企业选择第一种开发类型。

(2)企业文化

企业中任何一个人员均可能参与软件生产的各方面,如需求分析,设计、开发和测试等活动,是"通才"。在职责划分上,侧重以人员为中心而不是以角色为中心,缺乏层次性。中小企业在进行过程改进时就不太可能设立独立的软件过程组,而是软件工程人员参与到改进活动中,在工程项目开发的同时,并行进行改进。

(3)生产模式

项目的成功更多地依赖于少数人员的杰出技术能力和项目管理能力,企业缺乏明确定义的软件过程。成功项目的经验不能得到最大限度的继承,软件生产的可重复性差。规范化的软件过程不仅限制了软件的自由开发思想,同时也限制了在中小企业中,不同层次的软件人员对软件开发活动的支配力。对这类软件生产模式进行过程改进时,须平衡规范化和创造性的关系。

(4)人员变更

在中小企业中,人员变更相对频繁。软件过程改进就不能以具体的个人为中心,而是以软件过程及活动为中心。

基于以上特点,中小企业的软件过程改进应注意以下几方面:

(1)着重企业关心的目标,不一定完全依从 CMM 规范;

(2)在确立改进目标时,尽可能多地涉及各层次的项目组人员;

(3)尽量建立一般性的软件过程规范;

*)本文受国家自然科学基金和上海市青年基金资助。强　宇　博士后,研究方向为人工智能、知识工程;胡运发　博导,教授,研究方向为计算机网络、计算机通信。

(4)可以采用基于 TSP 的思想。

1.2 软件过程改进框架

在软件工程中,软件开发有多种方法,如瀑布法、原型法、增量迭代法等。结合更适合中小企业的具体特点,具体模型如下:

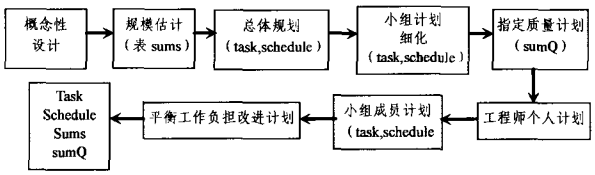

图 1 软件计划指定步骤图

Sums:规模小结表

Task:任务计划

Schedule:进度计划

sumQ:质量计划

软件过程改进可以贯穿到软件研发过程,如规模估计在需求分析阶段进行,任务、进度计划在概要设计和详细设计阶段进行,质量计划在代码阶段进行。在软件开发的各阶段,通过 Sums,Task,Schedule,sumQ 可以采集数据,例如表 sum 如下:

表 1 度量类别表

类别	度量数据
规模	需求文档数
	设计文档
	基本代码数
	增加代码
	删除代码
	被复用代码数
任务、进度	任务开始日期
	完成日期
	预计小时数
	每周工作小时数
质量计划	缺陷引入时间
	缺陷排除时间
	各构件的缺陷数

依据表 Sums(规模),sumQ(质量规模),Task(任务表),Schedule(计划表)和 ITL(风险分析管理表),SCI(配置表),peer(人员素质评价表),可以提取的基本度量元是规模、配置改变、风险、缺陷、项目进度、小组士气、用于软件过程改善。其中 peer 表如下所示。

表 2 人员素质表

属性	属性取值
全组士气	{1,2,3,4,5}
全组生产力	{1,2,3,4,5}
工作总有效性	{1,2,3,4,5}
工作回报性	{1,2,3,4,5}

表 3 角色表

角色	角色难度	在项目中总贡献性	有用性	技术支持性	完成水准
工程师	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}
小组长	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}
开发经理	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,345}	{1,2,3,4,5}	{1,2,3,4,5}
计划经理	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}
质量监督经理	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}
技术支持经理	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}
总计					

表 4 分值表

分值	含义
1	最差
2	较差
3	一般
4	较好
5	最好

2 概念格

形式概念分析理论是由德国的 Wille 教授首先提出的[2],以序论(Order Theory)、完全格理论(The Theory Of Complete Lattices)为理论基础,是一种优良的数据分析方法,在软件工程领域[11]已有成功的应用。形式概念分析为再工程、软件重用、面向对象程序设计等领域中某些问题的解决提供了理论支持,并已取得了应用成果。

概念格模型是形式概念分析的核心数据结构。概念源于哲学含义,由外延和内涵两部分组成,在形式概念分析中,概念外延定义为属于此概念的所有对象集合,概念内涵定义为属于此概念的所有对象共有的属性集合。所有概念及概念间的泛化例化关系构成概念格。概念格是一个层次结构,表现了概念间的泛化例化关系,对应的 HASSE 图实现了数据可视化。

2.1 基本定义

形式概念分析通常起始于形式背景这一基本概念。形式背景被定义为一个三元组 $K=(U,D,R)$,其中 U 是对象集,D 是属性集,R 是 U 和 D 间的二元关系,即 $R\subseteq U\times D$;U 和 D 的元素分别被称为对象和属性,而 oRd(即 $(o,d)\in R$)被读作:对象 o 具有属性 d。在 K 中,U 的幂集和 D 的幂集之间可以定义两个映射 f 和 g,如下:

$$\forall O_1\subseteq U:\ f(O_1)=\{\ d\mid \forall x\in O_1(xRd)\ \}$$

$$\forall D_1\subseteq D:\ g(D_1)=\{\ x\mid \forall d\in D_1(xRd)\ \}$$

称作 O_1 和 D_1 之间的 Galois 连接。二元组 (O_1,D_1) 如果满足两个条件:$O_1=g(D_1)$ 和 $D_1=f(O_1)$,则被称作形式背景 K 的一个形式概念,其中 D_1 和 O_1 分别被称为概念 (O_1,D_1) 的内涵和外延。分别用 $Intent(C)$ 和 $Extent(C)$ 表示。所有形式概念的集合标记为 $CS(K)$。$CS(K)$ 上最重要的结构是由亚概念-超概念关系(又称泛化-例化关系,或前驱-后继关系)产生的,其定义是:如果形式概念 (O_1,D_1) 是 (O_2,D_2) 形式概念的亚概念 $(O_1\subseteq O_2)$,则记为 $(O_1,D_1)\leqslant(O_2,D_2)$。通过这个关系,得到一个有序集 $\underline{CS(K)}=(CS(K),\leqslant)$,这是一个完全格,被称为形式背景 K 的概念格。

概念格可以图形化表示为有标号的线图(labelled line diagram)。在线图中,使用黑色的圆圈表示形式概念,向上的线段表示亚概念-超概念关系。如果概念 C 是包含某对象的最小概念,则此对象的名称附在 C 对应的圆圈上。对于一个属性,如果 C 是包含该属性的最大概念,则属性的名称被附在 C 所对应的圆圈上。概念格的有标号线图通常被用作通信模式,这使得给定数据背景的概念结构变得清晰和易于理解,从而实现了概念格的可视化显示。

知识的表示,并不能对知识进行系统的处理,要实现它还需要包括知识的推理和获取,并且应当建立工具用于知识的通信。为了对实现这些任务的概念知识系统建模,就需要对概念知识进行规范说明。概念知识处理使用对象、属性和概念(concept)作为它的基本成分,并且通过 4 个关系(对象具

有属性、对象属于概念，属性从概念中抽象出来，以及概念相互之间的父子概念关系）将这些基本成分链接在一起。概念知识处理使用形式概念（concept）作为它的核心概念（notion），而使用概念格作为知识表示方法。在形式概念分析中，可以很好地完成对这些结构化元素的数学化，因此形式背景及其概念格可作为对概念知识的基本要素进行形式化表示的一种适当的数学结构。

2.2 概念格的生成和显示

对给定的形式背景，国内外学者已经提出了许多算法来构建概念格[3,4]。Bordat 提出了非渐进式的批算法，采用自上而下的策略，由概念格的顶部概念节点开始，生成所有的子节点，并对每个新节点重复该过程。Godin 等人采用渐进式算法来构造概念格，其关键是标注更新节点和产生子节点。

2.3 模糊集合及模糊概念格

模糊集合是一种特殊定义的集合，其隶属度函数反映了模糊集合中的元素属于该集合的程度。模糊集合有很多表示方法，但都需要表示其包含的元素和相应的隶属度函数。根据模糊集合论，可将模糊引入形式背景，使背景从标准变为模糊、背景的每列定义为一个模糊语言表示的模糊集合，这样得到的就是一张模糊的形式背景[4]。

定义 1 已知模糊形式背景为 $K=(U,D,I)$，U 为对象集，$U=\{O_1,O_2,\cdots,O_n\}$，D 为模糊属性标识集，$D=\{D_1,D_2,\cdots,D_n\}$，I 是隶属度函数，$O\times D\to[0,1]$，或写成 $I(o,d)=m$，$0\leqslant m\leqslant 1$，隶属度函数可以是一般函数。

定义 2 模糊形式背景的截运算为：对于每个 $d_i\in D$，定义 ϕ_{d_i}，$0\leqslant\phi_{d_i}\leqslant 1$，定义模糊形式背景的截运算为 $\lambda K=(O,D,I_\lambda)$，$I_\lambda$ 为：

$$I_\lambda(o,d)=\begin{cases}I(o,d), & \text{如 } I(o,d)\geqslant\phi_{d_i}\\ 0, & \text{如 } I(o,d)<\phi_{d_i}\end{cases}$$

定义 3 在模糊形式背景 K 中，定义模糊概念 $C_i=(O_i,D_i)$，$O_i\subseteq O$，$D_i\subseteq D$，在 O_i 和 D_i 间可定义两个映射 f 和 g，如下式表示：

$$\forall O_i\subseteq U: f(O_i)=\{d_i\mid\forall o_i\in O_i, I(o_i,d_i)>\phi_{di}\}$$

$$\forall D_i\subseteq D: d(D_i)=\{o_i\mid\forall d_i\in D_i, I(o_i,d_i)>\phi_{di}\}$$

f 和 g 称 O 的幂集和 D 的幂集之间的 Galois 连接。

定义 4 如 (O_1,D_1)（$O_1\subseteq O, D_1\subseteq D$）满足：$O_1=g(D_1)$，$D_1=f(O_1)$，则称为模糊背景 K 的一个模糊概念，O_1，D_1 是模糊概念 C_1 的外延和内涵。K 的所有模糊概念集合记为 $CS(K)$。$CS(K)$ 上的结构是通过泛化例化关系产生的，其定义为：如 $O_1\subseteq O_2$，则 $(O_1,D_1)\leqslant(O_2,D_2)$。通过此关系得到的有序集 $\underline{CS(K)}=(CS(K),\leqslant)$ 称为模糊形式背景 K 的概念格。

在生成概念格的已有多种方法中，渐进法经证明是性能优良，可应用渐进的思想构造格。这里给出一种生成模糊概念格的算法描述[4,7]，首先给出几个主要参数的定义：

E 定义为 C_i 中所有对象在所有属性中的隶属度的均值，反映了模糊概念 C_i 的“平均模糊程度”。根据模糊集合理论，先计算 C_i 的对象集 O_i 的所有对象在模糊属性 d_i 上的隶属度均值 E_{d_i}，因 C_i 包含属性集 D_i，故再计算 E_{d_i} 关于 D_i 的均值 E，计算公式如下：

$$E_{d_i}=\frac{1}{|O_i|}\sum_{o_i\in O_i}I(o_i,d_i)\quad(|O_i|\text{是集合 }O_i\text{ 的元素个数})$$

$$E=\frac{1}{|D_i|}\sum_{d_i\in D_i}E_{d_i}\quad(|D_i|\text{是集合 }D_i\text{ 的元素个数})$$

δ 定义为所有 $\delta_{d_i}(C_i)$ 的均值，其中 $\delta_{d_i}(C_i)$ 是所有对象在属性 d_i 中相对于 E 的方差，δ 反映了 C_i 中隶属度值的离散程度。根据模糊集理论，先计算 O_i 中所有对象在属性 d_i 中相对于 E 的方差 $\delta_{d_i}(C_i)$。因 C_i 包含属性集 D_i，故再计算 $\delta_{d_i}(O_i,D_i)$ 关于 D_i 的均值 $\bar{\delta}$，计算公式如下：

$$\delta_{d_i}(O_i,D_i)=\delta(\{I(o_i,d_i)\mid o_i\in O_i\})=\sqrt{\frac{\sum_{o_i\in O_i}(I(o_i,d_i)-E)^2}{|O_i|}}$$

$$\bar{\delta}(O_i,D_i)=\frac{1}{|D_i|}\sum_{d_i\in D_i}\delta_{d_i}$$

在格的构造算法中，为了能够用渐进式方法计算参数 E，$\bar{\delta}$，引进中间结果 k_{d_j}，h_{d_j}。

$$k_{d_j}(c+x^*)=\sum_{o_i\in O}I(o_i,d_j)+I(x^*,d_j)$$

$$h_{d_j}(c+x^*)=\sum_{o_i\in O}(I(o_i,d_j))^2+I(x^*,d_j)^2。$$

由中间结果 k_{d_j}，h_{d_j}，可以计算 E，$\bar{\delta}$ 如下：

$$E=\frac{1}{|D|}*\sum_{d_j\in D}E_{d_j}=\frac{1}{|D|}*\sum_{d_j\in D}\frac{k_{d_j}}{|O|}$$

$$\bar{\delta}(O,D)=\frac{1}{|D|}*\sum_{d_j\in D}\delta_{d_j}$$

$$=\frac{1}{|D|}*\sum_{d_j\in D}\left(\frac{h_{d_j}-2*k_{d_j}*E+\sum_{o_j\in O}E^2}{|O|}\right)^{\frac{1}{2}}$$

具体生成算法[4]如下：

(1)做数据预处理，对应每个属性确定阈值 F_{d_j}。大于阈值的隶属度保持原值，小于阈值的隶属度取 0。

(2)概念格 L 初始化为空。

(3)从处理背景取出一个对象 x^*，形成结点 $(x^*,f(x^*))$，如果格中没有结点 c 使得 $f(x^*)\subseteq\text{int}(c)$，则将结点 $(x^*,f(\{x^*\}))$ 加入 L。

计算并保存 E 和 $\bar{\delta}$ 的中间结果，即 $k_{d_j}=I(x^*,d_j)$，$h_{d_j}=(I(x^*,d_j))^2$。

(4)扫描 L 中所有结点，找出所有内涵小于等于新增对象 x^* 内涵（$int(c)\subseteq f(\{x^*\})$）的格结点 c，则格结点 c 为更新点。将每个更新结点更新为 $(ext(c)\cup\{x^*\},int(c))$，边不更新。更新并保存 E 和 $\bar{\delta}$ 的中间结果，即 $k_{d_j}:=k_{d_j}+I(x^*,d_j)$，$h_{d_j}:=h_{d_j}+(I(x^*,d_j))^2$。

(5)如果格结点 c 与 x^* 的内涵交集不等于 L 中任意格结点的内涵，在这样的结点中取外延最大的一个，定义为产生子结点，将每个产生子点与此 x^* 一起生成新生结点 $(ext(c)\cup\{x^*\},int(c)\cap f(x^*))$。连接新生结点到它的子结点和父结点。更新并保存 E 和 $\bar{\delta}$ 的中间结果，即

$K_{d_j}:=K_{d_j}+I(x^*,d_j)$，$h_{d_j}=h_{d_j}+(I(x^*,d_j))^2$。

(6)直到所有的对象加入格中，否则转(3)继续。

(7)搜索所有没有子结点的结点，如果这样的结点多于一个，生成底结点 (Φ,D)，增加底结点到这些点的边。

(8)对每个结点已计算，得到 E 值和 $\bar{\delta}$，即

$$E=\frac{1}{|D|}*\sum_{d_j\in D}\frac{K_{d_j}}{|O|}$$

3 基于概念格的软件过程改进

3.1 基于模糊概念格的关联规则挖掘

在以往研究中，概念格多是基于标准二值背景构造的，提取的规则是确定规则[8]。但实际上，信息是模糊、不确定的，

本文在广义的模糊概念格基础上，给出了一种提取模糊关联规则的方法[4]：

对每个概念 $C_i=(O_i,D_i)(O_i\subseteq O,D_i\subseteq D)$，应用模糊背景的信息计算模糊参数 E 值和 δ 值。采用 E 和 δ，可在从格提取关联规则时，避免生成冗余频繁结点对，进而避免生成因高偏差导致的无意义规则。

模糊关联规则是形如 $D_1\Rightarrow D_2$ 的隐含式，D_1，D_2 是 D 上两个模糊集合，称为规则的前件和结论。

如果 c_1 和 c_2 是模糊概念，c_1 的属性集是 A，c_2 的属性集是 $A\cup B$，$\bar{\delta}(c_1)$，$\bar{\delta}(c_2)$ 是模糊概念 c_1 和 c_2 的参数，则关联规则 $A\Rightarrow B$ 的两个度量 Sup，Conf，可用下式计算，分别称为这条规则的支持度和置信度。

$\mathrm{Sup}(A\Rightarrow B)=\dfrac{|ext(c_2)|}{|U|}$，阈值 SUP，

$\mathrm{Conf}(A\Rightarrow B)=\dfrac{|ext(c_2)|}{|ext(c_1)|}$，阈值 ψ。

3.2 算法

基于模糊概念格的关联规则提取算法如下[4]：

(0)预处理：模糊化标准背景，生成模糊概念格。

(1)对模糊概念格中任意结点 c(外延集、属性集、参数 E、参数 $\bar{\delta}$)，如果 c 的 $\bar{\delta}$ 值小于等于某阈值 γ，则 c 是健壮结点；

(2)对模糊格中的健壮结点对(c_1,c_2)，如果 c_1 和 c_2 为父子结点，其中 c_1 是父结点，c_2 是子结点。且给定置信度阈值 ψ 和支持度阈值 SUP，如果满足：$\dfrac{|ext(c_2)|}{|ext(c_1)|}\geqslant\psi$，$\dfrac{|ext(c_2)|}{|U|}\geqslant$ SUP，则(c_1,c_2)称作满足阈值的候选二元组；

(3) 当且仅当结点对(c_1,c_2)是候选二元组，则提取关联规则：

$r_1\Rightarrow r_2$，$r_1=int(c_1)$，$r_2=int(c_2)-int(c_1)$。

4 实验及分析

考察 TSP 的度量[12,16]，发现各度量元间的关联关系，可以为后续的过程改进提供建议，形式概念分析理论可以对开发过程中的各影响因素进行分析，提高开发工作的效率。

在 TSP 整个度量模型中定义了很多度量，有些是基本度量，比如规模、时间、任务完成日期、缺陷数、阶段、产品部件，另外还有些是可推导度量，比如时间的阶段分布、计划价值、获得价值、缺陷数的阶段分布、缺陷数的部件分布等。各个基本度量可能存在着某些联系，尤其对于可推导度量，由于它的计算方法以及在计算它的过程中要用到其他度量，因此更有些千丝万缕的联系。挖掘出各种度量之间的联系不仅可以使开发人员了解各种度量之间的关系，在考察某一度量的同时重点考察与其相关的其他度量，同时可以在知道某一度量状态的同时推出其他度量的相应状态。重点考虑 TSP 度量方法中，几个基本度量之间的关系，主要涉及规模、配置申请、风险问题、错误缺陷、小组士气、任务进度。考察当某一度量处于一状态时，与其相关的其他度量可能的状态。例如当规模大并且提交的配置改变较多时，会不会导致任务进度的拖延和小组士气低。如果能得到这样的关联关系，那么就可以使开发人员对过程有一个更好的理解，并对后续的过程改进提供建议[17-19]。

在描述某一度量元的状态时，通常总是用自然语言。比如说规模大，那么究竟什么样的规模叫做“大”，这种大的概念又如何去表示？比如假设代码行大于 3000LOC 的产品叫做规模大，那么这样严格的区分就导致 2999LOC 代码的产品就是规模小的产品，而 3001 行代码的产品就是规模大的产品。仅相差两行代码就跨越了两个层次，这对于描述现实中的问题显然是不合理的。基于这个原因，本文试图用模糊的概念来描述规模大的概念，这样才能更好地反映实际中的问题。

本节首先介绍如何由 TSP 数据生成模糊形式背景表，然后在生成的模糊概念格的基础上提取模糊关联规则，其中还涉及规则的清洗和过滤。

步骤如下：

(1) 依据 LOGD，LOGT，ITL，SUMS，SUMQ，SUMP，CSR，CCR，SCHEDULE，WEEK 和 PEER 表，收集关于规模、小组士气、风险问题、配置改变、进度以及错误缺陷的数据，如表 5 所示。

表 5 过程基本数据表

项目编号	代码行数	批准的配置改变数	主要风险和问题数	引入的主要缺陷数	小组成员的综合评价分数	进度拖延的百分比
Q	3000LOC	23	11	167	4.2 分	15%
Q	4500LOC	12	14	123	3.6 分	4%
A	1000LOC	15	9	98	2 分	30%
A	2500LOC	11	7	74	3.4 分	27%
A	3000LOC	24	7	77	3.6 分	19%
……	……	……	……	……	……	……

(2) 划分各个度量元的状态，即是形式背景表中的各个属性。例如规模有三个状态，即“规模大”、“规模中”、“规模小”，具体的划分见表 6。

表 6 度量元状态划分表

基本度量元	状态	基本度量元	状态
规模	规模大 d_1	引入错误	缺陷多 d_9
	规模大 d_2		缺陷多 d_{10}
	规模大 d_3		士气低 d_{11}
配置改变	配置改变多 d_4	小组士气	士气低 d_{12}
	配置改变多 d_5		士气低 d_{13}
	风险大 d_6		未拖延 d_{14}
风险问题	风险大 d_7	项目进度	拖延 d_{15}
	风险大 d_8		

(3) 由开发人员选择度量元集合，根据实际情况设定各个度量元的状态，划分隶属度函数，生成一张由对实际数据进行计算得到的模糊的形式背景表，如表 7 所示。表中显示的是开发人员想要考察小组士气，引入错误和进度三个度量元之间的关系，经过计算得到的模糊形式背景表。

表 7 由小组士气、错误和进度组成的模糊形式背景

属性对象	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
1	0.9	0.1	0.7	0.3	0.0	0.2	0.8
2	0.8	0.2	0.3	0.6	0.1	0.3	0.7
3	0.2	0.8	0.1	0.9	0.0	0.7	0.3
4	0.7	0.3	0.8	0.2	0.0	0.4	0.6
5	0.4	0.6	0.1	0.1	0.8	0.8	0.2

(4) 在所生成的概念格基础上进行模糊关联规则挖掘。阈值 γ 设为 0.09，则健壮结点为$(4,d_9d_{11}d_{15})$ $(124,d_9d_{15})$ $(14,d_9d_{11}d_{15})$ $(3,d_{10}d_{12}d_{14})$ $(5,d_{10}d_{13}d_{14})$，置信度阈值 $\psi=0.6$ 和支持度阈值 SUP＝0.3，则候选结点对为{(124，d_9d_{15})，(14，$d_9d_{11}d_{15}$)}。

(5) 最后得到一组有价值的关联规则，可以对这些规则进行评估和解释，并可以依据这些规则对后续的过程改进提供建议。在例子中可以得到一条如下规则：

$d_9\wedge d_{15}\Rightarrow d_{11}$ 支持度：40% 置信度：66.7%

取逆规则，得到结果直观含义如下：

如果小组士气低，则引入缺陷多且项目拖延。

定位问题的原因，可能是由于任务量的加剧，而导致了成员的工作压力增大，造成了士气低落，并在开发过程中继续不断地引入错误，对完成开发任务没有信心，继而无法保质保量完成开发任务，致使整个项目开发进入恶性循环。挖掘出这样的规则，可以对后续的开发有效地进行过程跟踪和控制，当引入缺陷多，且造成进度拖延时，要时刻监控小组成员的士气，积极与他们沟通，发现困难并帮助解决。如果存在技术上的难题，技术支持经理则需对小组进行一次有效的技术培训，以提高成员的工作能力，继而提高整个开发小组的士气，逐渐减少引入的缺陷数。当考察的因素增多时，通过挖掘，还能够得到更多因素间潜在的关联关系。

结束语 本文改进了原有的 TSP 度量方法，将形式概念分析理论融入到 TSP 度量当中。提出了基于模糊概念格的关联规则挖掘的技术，用于挖掘过程数据库中的有价值信息，其可以对后续的过程改进提供有效的建议。

参 考 文 献

[1] Humphrey W S. 小组软件开发过程[M]. 韩丹，袁昱，译. 北京：人民邮电出版社，2002

[2] Wille R. Restructuring Lattice theory: an approach based on hierarchies of concepts// Rival I, ed. Ordered Sets. Dordrecht Reidel, 1982: 445-470

[3] 谢志鹏. 概念格及扩展模型研究[D]. 合肥工业大学计算机学院，2001

[4] 强宇. 概念格分布处理及其框架下的知识发现研究[D]. 上海：上海大学计算机工程与科学学院，2005

[5] Han Jiawei , Kamber M . Data Mining : Concept and Techniques[M] . Simon Fraser University , Canada , 2000

[6] 谢志鹏，刘宗田. 概念格的快速渐进式构造算法. 计算机学报，2002，25(5)：490-496

[7] 强宇，刘宗田，等. 一种模糊概念格构造算法研究. 计算机工程与应用，2004，4：13-19

[8] 谢志鹏，刘宗田. 概念格与关联规则发现. 计算机研究与发展，2000，37(12)：1415-1421

[9] 强宇，刘宗田，等. 模糊概念格在知识发现的应用及一种构造算法. 电子学报，2005，33(2)：350-353

[10] 强宇，刘宗田，李旭. 一种基于模糊概念格的关联规则构造方法. 计算机科学，录用

[11] Pressman R S. 软件工程：实践者的研究方法[M]. 梅宏，译. 北京：机械工业出版社，2002

[12] Function I, Point Users Group. IT 度量——专家实践[M]. 方德英，译. 北京：清华大学出版社，2003

[13] 卡耐基梅隆大学软件工程研究所. 能力成熟度模型(CMM)：软将过程改进指南[M]. 刘孟仁，等译. 北京：电子工业出版社，2001

[14] Galin D, Avrahami M. Do SQA programs work - CMM works: a meta analysis // Software - Science, Technology and Engineering, 2005 Proceedings. 2005: 95-100

[15] Dangle K C, Larsen P, Shaw M, et al. Software Process Improvement in Small Organizations: A Case Study. Software, IEEE , 2005, 22(6): 68-75

[16] von Konsky B R, Robey M. A Case Study: GQM and TSP in a Software Engineering Capstone Project. Software Engineering Education and Training // Proceedings. 18th Conference. April 2005: 215-222

[17] Sebern M J , Hilburn T B . Integrating Software Engineering Process in an Undergraduate Curriculum// Software Engineering Education and Training, CSEE&T 2005. Proceedings. 18th Conference. April 2005: 245-248

[18] Priya G V S. Walking the talk: building quality into the software quality management tool. Seshagiri, Quality Software // Proceedings. Third International Conference. Nov. 2003: 67 - 74

[19] 李建章，万江平. 中小型软件企业项目管理的思考. 计算机应用研究，2003，22(9)：14-17

(上接第 257 页)

结合区域理论，合理分配控制库所和资源库所中的资源来保证新的控制器系统不存在死锁状态。当原网模型资源分配改变时，只需改变控制器中控制库所的标识而不必改变其结构，即可获得结构简单的，许可行为趋于最优的控制器。

参 考 文 献

[1] Ghaffari A, Rezg N, Xie X L. Design of a live and maximally permissive Petri net controller using the theory of regions. IEEE Trans. Robot. Automat. , 2003, 19(1): 137-141

[2] Wysk R A, Yang N S, Joshi S. Detection of deadlocks in flexible manufacturing systems. IEEE Trans. Robot. and Automat. , 1991, 7(6): 853-859

[3] Banaszak Z, Krogh B H. Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows. IEEE Trans. Robot. And Automat. , 1990, 6(6): 724-734

[4] Hsien F S , Chang S C . Dispatching - driven deadlock avoidance controller synthesis for flexible manufacturing systems. IEEE Trans. Robot. And Automat. , 1994, 10(2): 196-209

[5] Xing K Y, Hu B S, Chen H X. Deadlock avoidance policy for Petri-net modelling of flexible manufacturing systems with shared resources. IEEE Trans. Automatic Control. , 1996, 41(2): 289-295

[6] Park J , Reveliotis S A. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. IEEE Trans. Robot. and Automat. , 2001, 46 (10): 1572- 1583

[7] Abdallah I B , ElMaraghy H A . Deadlock prevention and avoidance in FMS: a Petri net based approach. International Journal of Advanced Manufacturing Technology, 1998, 14(4): 704-715

[8] Jeng M D , Xie X L , Peng M Y . Process nets with resources for manufacturing modeling and their analysis. IEEE Trans. Robot. and Automat. , 2002, 18(6): 875-889

[9] Ezpeleta J , Colom J M , Martinez J . A Petri net based deadlock prevention policy for flexible manufacturing systems. IEEE Trans. Robot. and Automat. , 1995, 11(2): 173-184

[10] Li Z W, Zhou M C. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. IEEE Trans. Syst. , Man, Cybern. , 2004, 34(1): 38-51

[11] Giua A, DiCesare F, Silva M. Petri net supervisors for generalized mutual exclusion constraints// Proc. 12th IFACWorld Congr. Sydney, Australia, 1993: 267-270

[12] Chu F, Xie X L. Deadlock analysis of Petri nets using siphons and mathematical programming. IEEE Trans. Robot. and Automat. , 1997, 13(6): 793-804

[13] Li Z W, Wei N. Deadlock control of flexible manufacturing systems via invariant-controlled elementary siphons of petri nets. International Journal of Advanced Manufacturing Technology, 2007, 33(1-2): 24-35

[14] 赵咪，李志武. 一类离散事件系统的非阻塞监督控制器设计. 西安电子科技大学学报，2006，33(5)：735-738