第 33 卷 第 9 期 计 算 机 工 程 2007 年 5 月
Vol.33 No.9 Computer Engineering May 2007
·人工智能及识别技术· 文章编号：1000—3428(2007)09—0186—03 文献标识码：A 中图分类号：TP18

基于流程知识的 BPM 系统异常处理研究

陶亚雄[1,2]，王 坚[1]
（1. 上海同济大学 CIMS 研究中心，上海 200092；2. 重庆职业技术学院，重庆 400712）

摘 要：随着信息技术的发展和市场全球化程度加深，企业业务流程的动态、敏捷、多变等特点日趋明显，对业务过程管理系统的柔性提出了更高的要求。该文在研究流程知识与 BPM 系统异常关系的基础上，借鉴知识管理的相关理论和 BPM 异常处理研究成果，提出了利用流程知识进行 BPM 系统异常处理，赋予 ECA 法智能化处理异常的能力，扩展了 ECA 法的使用范围，提高了 BPM 系统的异常处理能力，控制异常发生，减小异常的负面影响，增加了 BPM 系统柔性。

关键词：BPM 异常；流程知识；ECA 规则；相似度

Research on BPM Exception Handling Based on Process Knowledge

TAO Yaxiong[1,2], WANG Jian[1]
(1. CIMS Center, Tongji University, Shanghai 200092; 2. Chongqing Vocational & Technical Institute, Chongqing 400712)

【Abstract】 Accompanied with the development of IT and market globalization, business process shows more and more dynamic, nimble and variable, that leads to more strict requirement on business process management system. Based on the relevant theory about KM and the conclusion of BPM exception, this paper analyzes the relation between process knowledge and BPM exception, and brings forth the meaning of dealing with BPM exception by making use of process knowledge, which intelligentizes the ECA rule and spreads its applying scope. Furthermore, it can improve the BPM system's capability for exception through controlling and descending exception's negative affection, and then make the BPM system more flexible.

【Key words】 BPM exception; Process knowledge; ECA rules; Similar degree

信息技术的飞速发展和市场全球化趋势的日益加深，使企业的组织形式、生产流程以及与客户、供应商和合作者的交互方式发生巨大变化，竞争范围已经扩展到企业的供应链之间。企业流程跨越了组织边界延伸到组织外部，敏捷、临时和动态的时变特点对相应的业务流程管理BPM提出了敏捷、实时和动态需求。这些需求归结起来就是对BPM系统的柔性要求，也就是系统对于变更和异常的处理能力[1]，其有关策略和方法的研究已逐渐成为业界的研究热点。

1 BPM 的异常

1.1 异常定义

任何业务过程在运行过程中，都可能会受到来自系统内、外各种因素的影响，从而与原来定义的过程发生偏差，这就是 BPM 系统的变更和异常。

变更一般指在业务流程建模或流程运行过程中，以流程顺利演进、优化为目的的主动的、可自动执行的流程更改，它一般都是可预知的。目前，BPM系统处理变更的策略、方法较多，主要有从流程建模角度着手的提前建模＋推后建模、规则后绑定、定义调整等，以及在流程运行中实施的实例调整(如向前/向后恢复、继续、迁移等)等[2]。

异常主要是指流程运行中偏离正常情况、可能导致障碍的流程变化，异常往往是不可预知或不可完全预知、被动进行、需要人工活动进行干预的[3]。由于异常发生的可预知性较小，BPM系统在处理异常的过程中也相对被动，只能由系统的异常检测控件在目标测试点的采样结果分析后得出异常判断，进而挂起运行中的实例、分析异常原因及系统的处理条件后，作出相应的处理决定。

1.2 异常处理的特点和意义

基于不同的角度，异常有多种分类方式。如按异常的产生源，可将异常分为系统异常和任务异常两类，前者指由BPM系统本身产生的异常，如操作系统故障，任务异常又可再细分为时间异常、数据异常和资源异常 3 种；按异常是否可预见可将异常分为可预见异常与不可预见异常等[3,4]。无论哪种分类，异常均被公认是流程运行中与定义的偏移，具有如下特点：(1)仅仅发生在过程的运行期间；(2)异常导致过程不能正常运行，但也有一些异常的负影响可忽略不计；(3)异常在过程定义阶段是不可预见的，不能在模型中对其进行说明；(4)所有的异常处理都不同程度地需要人工干预，尤其对非预见异常，人工干预是其主要处理手段。

通常把 BPM 的柔性归结为系统的灵活性、动态性和自适应性，究其实质，前二者是系统对变更的处理，而自适应性指 BPM 系统对于所有运行中偏移模型定义的各种状况的处理，这就是 BPM 系统的异常处理能力。因此，BPM 系统的异常处理能力对系统的柔性影响至关重要。

由于异常发生大都不可预知或起码不可完全预知，因此，提高系统对于异常的预测能力，以及在异常发生时能够尽早采取有效的解决措施，就成为提高 BPM 系统柔性的有力措施。本文提出了基于流程知识进行 BPM 异常处理，在系统的异常处理策略、方法选择和处理过程中，借助以往流程的经

作者简介：陶亚雄(1967－)，女，博士生，主研方向：智能生产系统，业务流程管理系统；王 坚，教授、博导

收稿日期：2006-09-13 **E-mail：**taoyx@vip.sina.com

验知识，提高系统对于异常处理的有效性，减小异常对流程结果的负面影响，增加系统柔性。

2 BPM 的异常处理

2.1 BPM 的异常处理步骤和策略

BPM 系统从发现异常到消除异常，通常包括异常检测、异常传播和异常处理 3 个环节。系统监控模块中的异常检测控件不断通过目标模块的输入输出与期望值进行比较，判断是否产生异常。一旦发现并确认异常现象后，生成基于确定消息机制的异常事件报告，在系统内部进行传递，直到找到可以处理该异常的责任人，再由他采用相应的措施进行处理。

由于异常可能发生在BPM系统的任何层次(网络层、操作系统层、数据层、应用层、用户层)，对异常的处理首先应由发生异常的那一层负责。除非该层不能处理，再将其传递到其它层或直接传递到顶层，由人工参与异常处理。异常传播既可在系统不同层次之间传递，也可在相同层次之间传递，如在用户层将异常从一个责任人传递到另一个责任人[3,5]。

确定异常发生后，BPM 用于修正或弥补其影响的异常处理策略主要包括：

(1)忽略：当出现异常的活动执行结果不影响整个流程目标时，忽略该异常，直接回到正常运行状态，继续流程运行。

(2)重试：停止异常活动，将该活动重新执行，直到任务正确完成或重试次数超过最大设定值。这个过程中可实施一些外部操作以提高运行可靠度，如恢复网络连接、重启系统等；

(3)修改实例：修改实例的某些属性或数据，以解决异常。

(4)回滚：通过相应补偿行为，使实例状态恢复到可执行状态，再重新启动实例运行。

(5)修改模型：对由于模型与实情不符造成的异常，通过修改过程模型以解决异常。这一修改会影响当前运行的实例，需将其按旧模型继续运行或迁移到新模型；而后续启动的新实例则按新模型运行。

(6)组合：对于复杂的异常，结合上述两种或两种以上策略进行处理[6]。

2.2 BPM 异常处理方法

对于异常处理方法，目前主要有失败补偿法、ECA 法、案例推理法等，都不同程度地需要人工干预才能完成。

(1)失败补偿法：对BPM流程的每个活动都给出一个补偿活动，用于消除已执行活动产生的影响。当一个活动发生异常时，就启动相应的补偿活动消除其影响。补偿法是目前商业软件中处理异常的主流方式，但实施中难以做到准确界定有效恢复范围[7]。

(2)ECA(Event-Condition-Action)规则法：将异常处理条件和行为分别用E、C、A进行描述，其中E指异常事件、C指异常对应条件、A是异常修正或补偿活动。ECA规则一般用于处理可预见异常，即事先将异常处理程序写成ECA规则，当异常E发生并满足条件C时，调用规则处理程序A[4]。

(3)案例推理法(Case Based Reasoning)：将异常处理作为历史案例保存在案例库中；当新异常发生时，将新异常与库中案例进行相似度匹配，找到相似度最大的案例；若二者情况相同，则按案例的解决方案处理新异常，若不完全一样，则以案例解决方案为模板进行调整和修改，获得新异常的解决方案，并把该新异常及解决方案添加到案例库中。案例推理法具有较强的推理功能，适合处理不可预见的异常[5,8]。

3 基于流程知识的业务过程管理系统 PK-BPMS

本文提出基于流程知识的业务过程管理系统 PK-BPMS(Business Process Management based on Process Knowledge)模式，借助知识管理的理念和方法，从业务流程中提取流程知识，经过加工整理后再将其用于流程管理，以更有针对性的方式进行 BPM 异常处理。其实施系统元模型框架如图 1 所示。

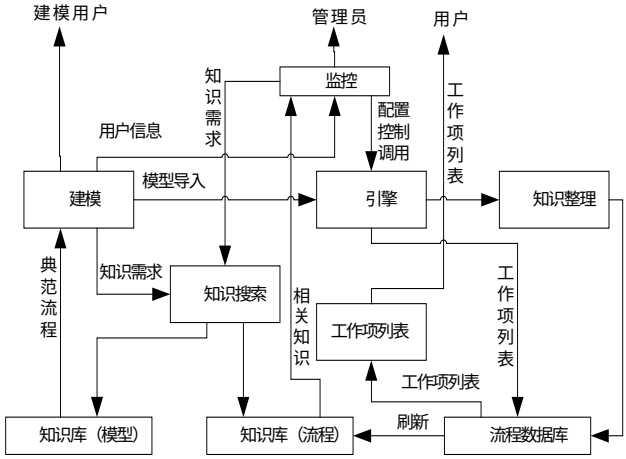

图 1 基于流程知识的 PK-BPM 系统

PK-BPMS 将流程活动与其所需知识在活动的属性中以知识需求的形式关联起来，通过引擎解析，使监控模块可以获取知识库中相关知识，根据监控模块反馈的实时数据，对流程运行异常做出及时、准确的判断，并借助库中相关异常处理经验，选择恰当的异常处理方案，及时调整和修改流程，保障流程运行。与此同时，上述流程操作过程及其结果又再送入相关数据库，经过筛选、整理，形成更新的流程知识，用于后续流程运行指导[6,9,10]。

4 利用流程知识处理 BPM 异常

2.2 节中的 ECA 规则法，对那些规则库中规定好的异常给出了明确的处理方案，具有较高的处理效率。但由于该方法本身没有学习能力，对那些不符合 ECA 库中规则特征描述的异常，则只能将其抛出。

本文的 PK-BPM 系统在异常处理方面基于 ECA 规则法，但对其进行了扩展。它利用以往流程的异常处理经验，将那些在 ECA 库中没有处理规则的异常，按照相似度最小的定则，找到与之最相似的案例，结合本次异常具体情况，调整其处理方式而形成本次处理方案。并将该次异常处理过程作为新规则添加到 ECA 库中，使 ECA 法具有了“智能化”处理异常的能力，提高 BPM 系统对异常处理的有效性。下面具体介绍其异常处理过程。

4.1 基于 ECA 规则的异常处理相关定义

定义过程为三元组 $P=\langle A,C,R\rangle$，其中 $A=\{a_1,a_2,...,a_m\}$ 为过程活动集合。则每个活动 $a_i(1\le i\le m)$ 的属性由

$\langle Performer, Application, TimeConstra\,\text{int}, Inputdata, Outputdata\rangle$

构成。它们依次分别表示活动执行者、活动调用的应用程序、时间约束、输入和输出数据。$C\subset A\times A=\{c_1,c_2,...,c_n\}$ 是活动间连线，表示活动控制逻辑，每一连线 $c_j(1\le j\le n)$ 均对应一个条件表达式 $Cond_j$，用以判断该连线的激活条件。$R\subset A\times C=\{r_1,r_2,...,r_k\}$ 是过程逻辑规则，描述活动汇入/汇出逻辑关系，有与/Joint、或/Joint、与/Split、或/Split类型。

对ECA规则库 $ECASET=\{ECA_1,ECA_2,\cdots,ECA_l\}$ 中的每个ECA规则，可定义成一个三元组

$ECA_i=\langle Ev,Co,Ac\rangle,1\le i\le l$

(1) Ev为异常事件，是一个三元组

$Ev=\langle Type,Time,Locate\rangle$

其中：

1) *Type* 为枚举型数据，说明异常事件类型以对应不同的处理方法，一般为组合类型。本文的 PK-BPM 系统中，首先将异常分为系统异常和任务异常，其中任务异常又再分时间异常、数据异常和资源异常 3 种。本系统仅针对可预见异常处理，其相应的类型结构为

$$Type = \{\Pr ed.System, \Pr ed.Task.t, \Pr ed.Task.d, \Pr ed.Task.r\}$$

2) *Time* 指异常发生时间；

3)*Locate* 表异常发生的地点，可以是活动、子过程或一个系统模块，是一个枚举型数据，可表示成

$$Locate = \{Activity\} \vee \{Subprocess\} \vee \{Module\}$$

(2)*Co* 是异常处理的前提条件，是一个谓词语句。

(3)*Ac* 是规则包含的处理动作，它是一个<*Oper*,*Proced*>结构的二元组。其中：

1)*Oper* 是异常操作，有

$$\langle Ignore, \mathrm{Re}try, ModifyInt, Modify\,\Pr oc, Back\,\mathrm{Re}set, Comp\rangle$$

共 6 种类型，分别代表忽略、重试、修改实例、修改模型、后向恢复和组合操作。

2)*Proced* 是具体的处理过程，可以是一个活动、子过程、应用程序等。

4.2 基于流程知识的 ECA 扩展定义

以 *Case*=<*PInsStatus*,*AInsStatus*,*Activity*,*ExcType*,*Time*, *Participant*,*Characteristic*>来描述 PK-BPMS 的异常及其相关信息。其中，*PInsStatus*、*AInsStatus* 分别指异常发生时过程实例和活动实例的状态(初始化、激活、运行、暂停、中断、完成)；*Activity* 指引发异常的活动；*ExcType* 为异常类型(系统异常/任务时间异常/任务数据异常/任务资源异常)；*Time* 是异常发生时间；*Participant* 是引发异常的活动执行者；*Characteristic* 是特征数据，它随不同流程的应用领域而变。

PK-BPMS 中，若系统捕捉到的异常事件 e_1 不能从案例库中找到与之完全匹配的案例 $e_{\equiv}$，则转而在库中根据案例相似度搜索与其最相似的案例。PK-BPMS 选择若干主要属性 $a_j\,(1 \le j \le m)$ (如异常类型、引起异常的活动、异常影响范围、异常发生时间等)，将其依次按照在流程中的重要性进行排列，确定出每个属性的概念层次后，分别求出两个案例在各个属性上的相似度值，再将其加权求和，所得即为两案例之间的相似度 $dist(e_1,e_2)$。

其中概念层次(Concept Hierarchy)指概念域中从一般到特殊的层次结构。如在制造企业组织中，可以将其组织的层次从大到小、从一般到特殊依次划分为如图 2 所示情况。再据此层次结构给每个概念层次赋予一个层次值(level value)，通常将末级节点的层次值(图 2 中的叶节点)为 0，由下向上层次值逐渐加大，如图中各概念旁所标注的数值。

令 $dist(a_{j1},a_{j2})$ 为选择属性 $a_j\,(1 \le j \le m)$ 对应的两个案例 e_1、e_2 的属性 a_{j1}、a_{j2} 间的相似度，当 a_{j1}、a_{j2} 是数值型时，其相似度值就等于两者相减的绝对值；否则就以 a_{j1}、a_{j2} 对应概念层次中的最小相同节点层次水平为相似度。

求得案例之间各属性的相似度后，将其按规定因子加权求和，即得到异常事件 e_1 与 ECA 库中当前案例 e_2 之间的相似度值 $dist(e_1,e_2)$，即

$$dist(e_1,e_2) = \sum_{j=1}^{m} w_j \bullet dist(a_{j1},a_{j2})$$

其中，w_j 为选择属性 a_j 的权重。显然，相似度越小，两个案例之间越相似，其处理方案对当前异常事件处理的参考借鉴价值也越大。

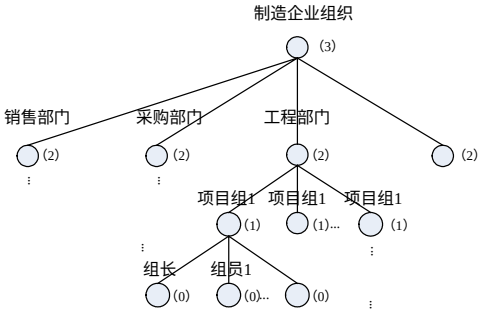

图 2 制造企业组织概念层次结构

4.3 基于流程知识的 PK—BPM 系统异常处理过程

根据上述基本定义，基于 ECA 规则的异常处理流程如图 3 所示。

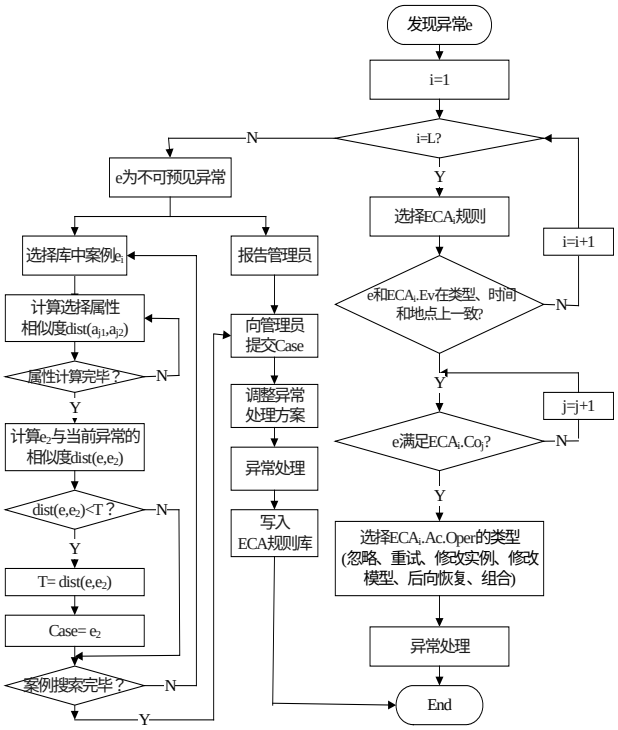

图 3 PK-BPM系统的异常处理流程

其执行步骤描述如下。

步骤 1 当捕捉到异常事件e_1时，从规则库*ECASET*中选择一条*ECA*规则ECA_i；

步骤 2 判断e_1和$ECA_i.Ev$在类型*Type*、时间*Time*和地点*Locate*上是否一致。

(1)当不一致时

1)继续在规则库中选择另一条规则，重复步骤 2；

2)当遍历完所有规则仍未找到匹配项时，则认为异常事件e_1不能按现有ECA规则处理，需报告管理员，同时启动相似度搜索，查找相似的异常；

3)从ECA库中选择一条案例e_2，依次分别计算e_2与当前异常e_1之间在各选择属性的相似度值$dist(a_{j1}, a_{j2})$；

4)完成所有属性相似度计算后，将其加权求和，得到两案例之间的相似度$dist(e_1,e_2)$并赋给T，用Case记录所选案例。

5)判断案例库是否搜索完毕，如果没有，继续选择另一条案例，重复步骤 3)和步骤 4)，计算相似度值。如果新求得的值小于T，将该值赋给T，并用当前案例覆盖Case中的案例。如果已经完毕，则Case中的案例即为所求与当前异常e_1最相似的案例。

（下转第 209 页）