(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102170360 A
(43)申请公布日 2011.08.31

(21)申请号 201110098097.0
(22)申请日 2011.04.19
(71)申请人 北京神州数码思特奇信息技术股份有限公司
地址 100085 北京市海淀区上地九街 9 号数码科技广场二层
(72)发明人 李晓静
(74)专利代理机构 北京轻创知识产权代理有限公司 11212
代理人 杨立
(51)Int.Cl.
H04L 12/14(2006.01)
H04L 12/24(2006.01)

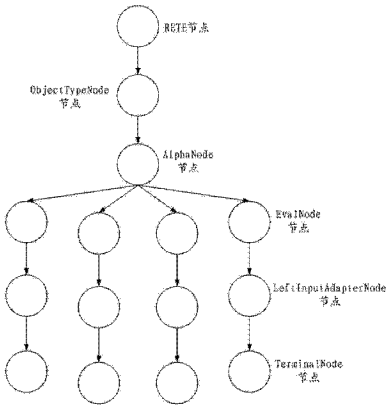
权利要求书 1 页 说明书 5 页 附图 2 页

(54)发明名称

一种规则引擎的模式匹配方法和 RETE 网络

(57)摘要

本发明涉及一种规则引擎模式匹配方法和 RETE 网络，所述方法包括：构建 RETE 网络，其中所述 RETE 网络中包含一 RETE 节点，所述 RETE 节点下只有一个对象类型节点(ObjectTypeNode)，所述对象类型节点下有 AlphaNode 节点、EvalNode 节点、LeftInputAdapterNode 节点和 TerminalNode 节点；插入事实，并直接在所述 RETE 网络中进行条件匹配，不生成所述事实的事实句柄，不保存所述事实的信息；将所匹配的规则放入一优先级队列，并顺序执行。本发明一方面能够提高系统的灵活性，将原来固定在代码中的计费规则分离出来，用规则引擎的前台规则管理系统管理起来，增删查改都很方便，可以灵活配置；另一方面，应用规则引擎之后的内容计费系统性能，在话单处理效率和内存使用量方面优于原内容计费系统的话单处理效率。



CN 102170360 A

1. 一种规则引擎的模式匹配方法，其特征在于，所述方法包括以下步骤：

步骤 1：构建 RETE 网络，所述 RETE 网络中包含一 RETE 节点，所述 RETE 节点下仅有一个对象类型节点，所述对象类型节点下有 AlphaNode 节点、EvalNode 节点、LeftInputAdapterNode 节点和 TerminalNode 节点；

步骤 2：插入事实，并直接在所述 RETE 网络中进行条件匹配，不生成所述事实的事实句柄，不保存所述事实的信息；

步骤 3：将所匹配的规则放入一优先级队列，并顺序执行。

2. 一种 RETE 网络，其特征在于：所述网络包含一 RETE 节点，所述 RETE 节点下仅包括一个对象类型节点，所述对象类型节点下有 AlphaNode 节点、EvalNode 节点、LeftInputAdapterNode 节点和 TerminalNode 节点。

一种规则引擎的模式匹配方法和 RETE 网络

技术领域

[0001] 本发明涉及电信行业中的内容计费过程，具体来说是一种用于内容计费的规则引擎模式匹配方法和 RETE 网络。

背景技术

[0002] 复杂企业级项目的开发和维护面临着软件必须“随需而变”的问题。由于随外部条件和需求经常变动的商业决策和业务逻辑被固化在代码中，这使得软件的开发和维护变得异常困难，耗费大量的时间和成本，因此迫切需要分离商业决策者的商业决策逻辑和应用开发者的技术决策，并把这些商业决策放在中心数据库或其他统一的地方，让它们能在运行时可以动态地管理和修改，从而提高软件系统的柔性和适应性。规则引擎正是应用于上述动态环境中的一种解决方法，它增加了软件的可维护性，为软件提供可供用户直接修改业务逻辑的功能，使得软件项目不会因为维护成本太高而最后亏本。

[0003] 规则引擎 (Rule Engine) 由推理引擎发展而来，是一种嵌入在应用程序中的组件，实现了将业务决策从应用程序代码中分离出来，并使用预定义的语义模块编写业务决策。规则引擎接受数据输入，解释业务规则，并根据规则做出业务决策。

[0004] RETE 算法是 Charles Forgy 博士于 1979 年提出的一种效率很高的模式匹配算法，广泛应用于各种规则引擎中，其核心思想是将分离的匹配项根据内容动态构造匹配树，以达到显著降低计算量的效果。

[0005] 对于内容计费来说，实际上是针对“内容产品”展开的计费。内容计费业务种类多，而且计费规则经常变动，系统维护困难。因此需要将规则引擎应用在内容计费上，把计费规则从代码中分离，用规则引擎进行管理。

[0006] 现有的 RETE 算法是这样的 :规则引擎的 RETE 算法子系统根据编译子系统生成的规则包，构建 RETE 网络。当有事实插入（Insert）时，生成事实句柄 FactHandle，然后在 RETE 网络中与各节点匹配，同时会在匹配成功的每个节点对应的内存中保存该事实信息 ；在匹配之后要撤销（Retract）每个事实句柄，来删除在 RETE 网络中进行匹配时记录的所有事实信息。这样，会占用很多内存来记录事实信息，匹配过程也的时间开销也很大，在匹配完执行规则之后，还要花费时间 Retract 事实来删除匹配过程中在 RETE 网络中留下的事实记录。这样必然导致使用规则引擎的运算过程要比把规则直接固化到代码中的系统运算速度慢得多。

[0007] 因此，使用现有的规则引擎存在着性能上的问题。规则引擎的核心算法是 RETE 算法，由于该算法中，事实匹配规则的过程不一定独立，多个事实同时满足某个条件时才执行该条规则所对应的动作，原 RETE 算法在插入事实的过程中在 RETE 网络中保存了所有事实的信息，这样就使得该算法建网匹配过程复杂，并且占用大量的内存空间，使得应用规则引擎的系统性能大大低于将业务逻辑固定在代码中的系统，从而使得规则引擎无法在计费系统中进行广泛的应用。

发明内容

[0008] 本发明所要解决的技术问题是提供一种改进的规则引擎模式匹配方法，使其在兼容现有 RETE 算法的同时，使应用改进后的规则引擎的内容计费系统的处理效率接近原内容计费系统。

[0009] 本发明解决上述技术问题的技术方案如下：

一种用于规则引擎的模式匹配方法，包括以下步骤：

步骤 1：构建 RETE 网络，所述 RETE 网络中包含一 RETE 节点，所述 RETE 节点下仅有一个对象类型节点（ObjectTypeNode 节点），所述对象类型节点下有 AlphaNode 节点、EvalNode 节点、LeftInputAdapterNode 节点（左输入适配节点）和 TerminalNode 节点（终止节点）；

步骤 2：插入事实，并直接在所述 RETE 网络中进行条件匹配，不生成所述事实的事实句柄，不保存所述事实的信息；

步骤 3：将所匹配的规则放入一优先级队列，并顺序执行。

[0010] 其中，事实即为数据，是指用户插入的、符合规则引擎中用户定义的事实类的对象。比如，本发明中提到的应用规则引擎后的内容计费系统中，用户订购关系数据即是事实。

[0011] RETE 节点为 RETE 网络的根节点，为整个 RETE 网络的入口。

[0012] 对象类型节点的作用是提供对对象所属类进行筛选，根据内容计费的具体需要，本发明中只定义了一种对象类型节点。

[0013] AlphaNode 节点用来提供对象属性值的筛选，即评估对象的字面条件（literal conditions），当一个对象有多个字面条件的话，逐条评估顺序进入相应 AlphaNode 节点。

[0014] LeftInputAdapterNode 节点用于提供事实到元组的转换。

[0015] EvalNode 节点为判断节点，会返回表达式的真值。其中可以使用任何有效的语言表达式，只要它最终能被求值为 boolean 数据类型（Boolean 数据类型：变量存储为 16 位（2 个字节）的数值形式，但其值只能是 True（真）或者 False（假））。

[0016] TerminalNode 节点表示一条规则的完全匹配，即表明一条规则已经匹配了它的所有条件。

[0017] 由于本发明提供的规则引擎模式匹配方法，其所构建的 RETE 网络只有一个对象类型节点，其下有 AlphaNode 节点、EvalNode 节点、LeftInputAdapterNode 节点和 TerminalNode 节点，从而不会出现传统 RETE 网络中的 JoinNode，ExistNode，NotNode 等其他节点，因此每个订购关系事实匹配规则的过程是独立的，不会出现两个或两个以上的事实同时满足某个条件时才执行该条规则所对应的动作，所以在匹配过程中在 RETE 网络中保存事实信息没有必要。在匹配过程中不记录事实信息，从而也就不用撤销事实，因此大大减少了内存的开销和时间开销。这种匹配模式适用于内容计费规则这种情况：没有 JoinNode、ExistNode、NotNode 等节点（JoinNode、ExistNode 和 NotNode 节点用于比较两个或两个以上的事实对象，这些事实可能属于相同或不同的对象类型。本发明所提供的规则引擎每次只进行单个事实的匹配，不需要 JoinNode、ExistNode 和 NotNode 对两个对象进行比较），每个事实匹配规则的过程是独立的。

[0018] 本发明改进后的规则引擎应用于内容计费系统，一方面能够提高系统的灵活性，将原来固定在代码中的计费规则分离出来，用规则引擎的前台规则管理系统管理起来，增

删查改都很方便，可以灵活配置；另一方面，应用规则引擎之后的内容计费系统性能，在话单处理效率和内存使用量方面优于原内容计费系统的话单处理效率。

[0019] 根据上述规则引擎模式匹配方法，本发明还提供了一种 RETE 网络，该 RETE 网络包含一 RETE 节点，所述 RETE 节点下仅包括一个对象类型节点，所述对象类型节点下有 AlphaNode 节点、EvalNode 节点、LeftInputAdapterNode 节点和 TerminalNode 节点。

附图说明

[0020] 图 1 为本发明的方法中用于内容计费系统的 RETE 网络结构示意图；

图 2 为利用本发明方法的内容计费系统示意图；

图 3 为本发明中嵌入改进的规则引擎之后的内容计费系统的执行流程图。

具体实施方式

[0021] 以下结合附图对本发明的原理和特征进行描述，所举实例只用于解释本发明，并非用于限定本发明的范围。

[0022] 如图 1 所示，本发明中的内容计费的规则有如下特点：内容计费的规则构建的新的 RETE 网络中包含一 RETE 节点，该 RETE 节点下只有一个对象类型节点（ObjectTypeNode 节点），该对象类节点下有 AlphaNode 节点、EvalNode 节点、LeftInputAdapterNode 节点和 TerminalNode 节点。不会出现现有的 RETE 网络中的 JoinNode，ExistNode，NotNode 等节点，因此每个订购关系事实匹配规则的过程均是独立的，不会出现两个或者两个以上的事实同时满足某个条件时才执行该条规则所对应的动作。所以在匹配过程中在 RETE 网络中保存事实信息没有必要。

[0023] 根据上述内容计费规则的这个特点，本发明提出了如下 RETE 算法的改进方案：

在规则引擎的推理引擎部分设计了一种新的规则引擎匹配模式 (simple_match)，对于插入的事实不再生成 handle（事实句柄 FactHandle），而是事实直接在 RETE 网络中进行条件匹配，也不保存事实的任何信息，最后匹配的规则全部放到一个优先级队列，顺序执行。因为在匹配过程中没有记录事实信息，也就不用 Retract 事实了，大大减少了内存开销和时间开销。这种匹配模式适用于像内容计费规则这种情况：没有 JoinNode，ExistNode，NotNode 等节点，每个事实匹配规则的过程是独立的。具体的方法步骤包括：

步骤 1：构建 RETE 网络，其中所述 RETE 网络中只有一个对象类节点（ObjectTypeNode），该对象类节点下有 AlphaNode 节点、EvalNode 节点和 TerminalNode 节点；

步骤 2：插入事实（fact），并直接在所述 RETE 网络中进行条件匹配，不生成所述事实的事实句柄，不保存所述事实的信息；

步骤 3：将所匹配的规则放入一优先级队列，并顺序执行。

[0024] 本发明中所提到的事实即为数据，是指用户插入的、符合规则引擎中用户定义的事实类的对象。比如，在本发明中提到的应用规则引擎后的内容计费系统中，用户订购关系数据即是事实。

[0025] 如图 2 所示，把改进后的规则引擎，作为中间件嵌入内容计费系统中，用规则引擎的规则管理子系统来管理内容计费规则，用后台的推理引擎来完成订购关系事实和计费规

则的匹配，从而得出计费结果。运行时，规则引擎选择 simple_match 匹配模式。

[0026] 嵌入上述规则引擎之后的内容计费系统的执行流程，如图 3 所示，包括：

1）编译内容计费的计费规则，生成规则包；

2）加载规则包，构建 RETE 网络；

3）创建 workingMemory；

4）读数据库增量表的一条数据，构造订购关系事实；

5）插入事实到 workingMemory，在 RETE 网络中匹配规则，返回计费结果；

6）根据计费结果写话单，写数据库；

7）循环执行步骤 5)、6)，直至收到退出信号。

[0027] 其中，内容计费的计费规则是由规则引擎的规则管理子系统提供。数据库增量表是订购关系同步程序从 CRM 系统（Customer Relationship Management System）同步过来的内容计费订购关系的增量表。最后输出的话单文件将会传给综合账务系统。

[0028] 本发明将改进后的规则引擎应用于内容计费系统，一方面提高了系统的灵活性，将原来固定在代码中的计费规则分离出来，用规则引擎的前台规则管理系统管理起来，增删查改都很方便，可以灵活配置；另一方面，应用规则引擎之后的内容计费系统性能，在话单处理效率和内存使用量方面优于原内容计费系统的话单处理效率。

[0029] 本发明的方法应用于吉林移动内容计费系统，将原来固定在代码中的计费规则分离出来，用规则引擎的前台规则管理系统管理起来，增删查改都很方便，可以灵活配置。内容计费系统的后台应用了规则引擎的推理引擎来处理用户订购关系事实，匹配计费规则，得出正确的计费结果。同时，新内容计费系统的性能优于原内容计费系统：使用了规则引擎的内容计费程序在处理速度和 CPU 使用量方面略高于原内容计费系统，但内存使用量是原内容计费系统的 1/90，能够满足生产系统对内容计费系统的需求。

[0030]

内容计费规则样例 1

```
rule "1_INC_1"
salience 20
no-loop true
when
    f:MonthBillingData(billing_type==3)
 eval( f->order_stat.compare("05")==0&&
f->first_order_time_cur.compare("00000000000000")==0&&
f->first_order_time_his.compare(f->month_begin)<0&&
f->change_time.compare(f->month_begin)>=0)
then
    //cout<<"1_INC_1"<<endl;
    f->state=11;
    f->deduct_type=0;
end
```

内容计费规则样例 2

```
rule "1_MONTH_1"
salience 20
no-loop true
when
    f:MonthBillingData( )
eval( f->billing_type==1&&
f->first_order_time_his.compare(f->month_begin)<0&&
f->first_order_time_cur.compare("00000000000000")==0&&
(f->order_stat.compare("05")==0||f->order_stat.compare("06")==0))
then
    //cout<<"1_MONTH_1"<<endl;
     f->state=11;
     f->deduct_type=0;
end
```

以上所述仅为本发明的较佳实施例，并不用以限制本发明，凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

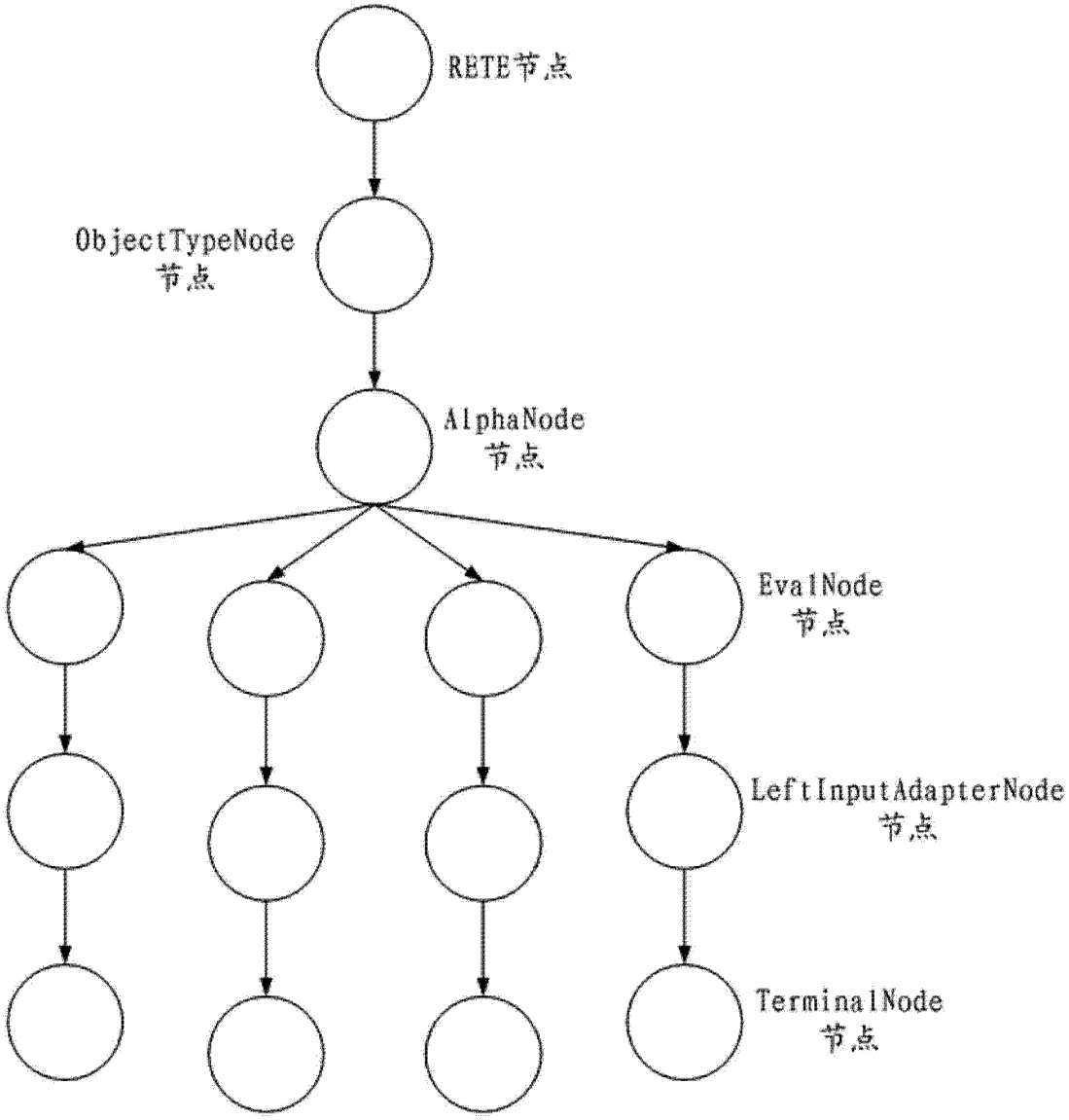

RETE节点
ObjectTypeNode
节点
AlphaNode
节点
EvalNode
节点
LeftInputAdapterNode
节点
TerminalNode
节点

图 1

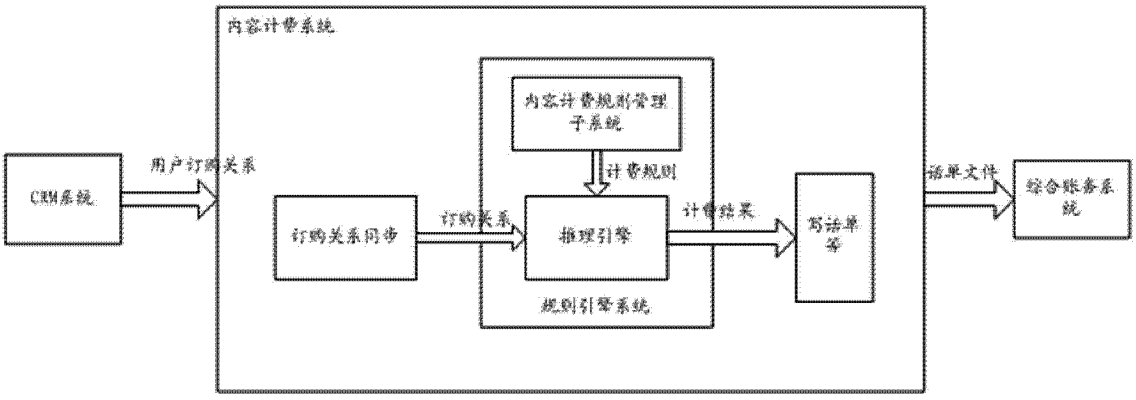

内容计费系统
内容计费规则管理
子系统
计费规则
CRM系统
用户订购关系
订购关系同步
订购关系
推理引擎
计费结果
写话单
等
规则引擎系统
话单文件
综合账务系
统

图 2

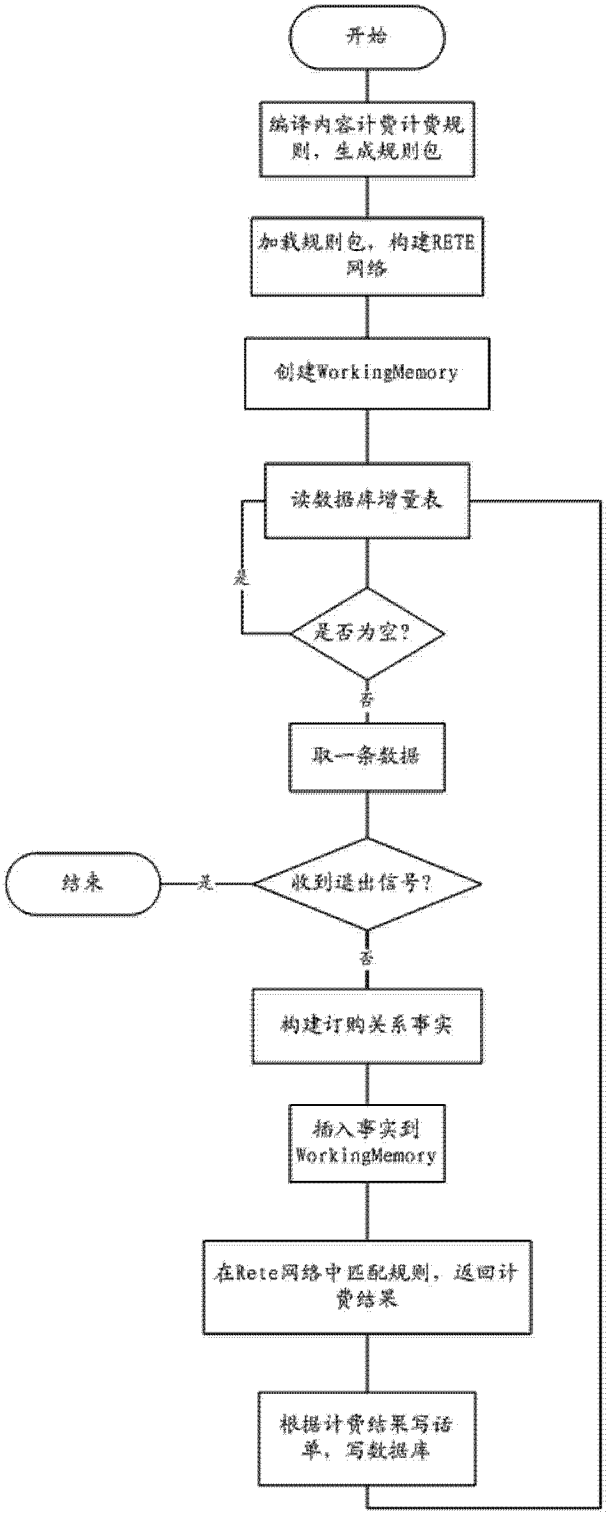

开始
编译内容计费计费规则，生成规则包
加载规则包，构建RETE网络
创建WorkingMemory
读数据库增量表
是
是否为空?
否
取一条数据
结束
是
收到退出信号?
否
构建订购关系事实
插入事实到WorkingMemory
在Rete网络中匹配规则，返回计费结果
根据计费结果写话单，写数据库

图 3