

一种工作流中的服务选择算法

余波^{1,4} 周龙骧² 钟锡昌³ 张倪³

(中科院计算技术研究所 北京 100080)¹ (中科院数学与系统科学研究院 北京 100080)²

(中科院软件中心 北京 100080)³ (中科院研究生院 北京 100080)⁴

摘要 与其他服务发现相比,工作流中的服务选择有着自身的特点。在目前情况下,大多数的服务选择只是从对服务本身需求的角度来考虑,并未涉及到多个服务之间的关联约束和协调性问题。而在工作流中,这些问题同样是影响服务选择的主要因素。本文提出一种服务选择算法,在该算法中,利用规则从多个方面来反映服务之间的约束关系并指导服务选择的方式,进一步对每个候选服务与相关联服务之间的兼容性进行了比较,最后得到一个同时满足自身和流程需求的服务。

关键词 服务发现,工作流,兼容比较,规则

An Approach for Selection of Service in Workflow

YU Bo^{1,4} ZHOU Long-Xiang² ZHONG Xi-Chang³ ZHANG Ni³

(Institute of Computing Technology, CAS, Beijing 100080)¹ (Institute of Mathematics, AMSS, CAS, Beijing 100080)²

(Software Center, CAS, Beijing 100080)³ (Graduate School of the CAS, Beijing 100080)⁴

Abstract Compared with others in service discovery, the selection of service in workflow has its own characteristic. At the present, the selection of the service is always according to the requirement of the single service in most researches, which didn't consider the constraint relationships and the coordination problems among the services. But in workflow, those problems are also the important factors that impact the selection of the service. The paper put forward a new algorithm used in service selection, which based on rule and compatibility comparison. The purpose of the rules that guide the selection process is to reflect the relationships among the services. After removing some services provided by the advertiser which don't obey the rule, the algorithm compares the compatibility between the candidate service and other related services in the following step. In the end, a service which satisfied the requirements both of its own and the entire workflow is obtained.

Keywords Service discovery, Workflow, Compatibility, Rule

1 引言

随着 Internet 及相关技术的深入发展,各个企业和组织机构之间的交流和合作更加频繁。电子商务、电子政务等借助 Web 技术的新型办公模式已逐渐走进人们的日常生活当中,商务流程或普通的办公流程也不再仅仅局限于单个组织内部,社会分工形成的专业化使得更多的机构将借助于市场或其他组织提供的服务来构建自己的业务流程。从实际应用效果的角度来看,这种跨组织的工作流在一方面能够通过借助他人提供的服务来弥补自身功能上的不足,另一方面也可以通过采用更廉价的服务来达到提高效益的目的。但与传统的工作流相比,新的应用模式在实现上也带来了许多技术上的挑战^[1,2]。

在跨组织工作流的工作模式下,流程制定者可以通过自身组织提供的功能来完成流程中的部分关键活动(Activity),其他的活动可通过调用第三方提供的功能来完成。在有些流程中,甚至所有活动都是通过第三方来完成。SOA(Service-oriented Architecture)的提出和 Web 服务及相关标准的大力推广极大地减少了这种工作模式在技术上实现的难度,在

SOA 中,服务是一个自包含、自描述、不依赖于应用上下文或其他服务状态并能提供某项功能的实体。通常,采用 SOA 构建的系统具备松耦合的特点。在典型的 SOA 中,服务提供商将服务的相关信息注册到一个注册机构中,服务的使用者通过查找注册机构,找到合适的服务及其调用方式,然后直接调用该服务^[3]。

目前,流程中的活动基本上都是采取静态绑定的方式,即在流程执行前,流程制定者通过手工的方式为每个活动查找并绑定具体的服务。但是随着 Web 服务和网格等技术的进一步应用,越来越多的商业机构会成为服务提供商。即使是提供同一功能的服务,流程制定者都有可能面临着成千上万个选择,继续要求采取手工的方式对服务进行绑定,不仅会导致效率低下,而且极易导致错误的绑定。另外,由于动态变化的业务需求,往往要求在很短的时间里或者不需要人工干预的情况下,对绑定的服务做出动态的调整。因此,能够对服务进行自动绑定是非常重要的。

在服务绑定中,服务的选择是其中的关键,目前对于服务选择有着大量的研究工作^[4~6]。在这些研究中,大多是从语法或语义的角度在服务需求和服务描述之间进行各种方式的

余波 博士生,主要研究领域:工作流技术、服务网格;周龙骧 教授,博导,主要研究领域:数据库系统实现技术、分布式数据库技术和电子商务技术;钟锡昌 研究员,博导,主要研究领域:软件工程、应用软件技术;张倪 研究员,主要研究领域:软件工程、嵌入式软件开发。

匹配查找。但是,对于 workflows 中的服务选择,直接应用这些查找算法主要存在着下面几点不足:1)动态绑定的需求。与普通应用程序相比,workflow 通常具有长运行周期的特性,传统事先静态选择服务的方式并不能很好地适应业务需求和运行环境的动态变化。2)在服务发现中,对服务选择大多是基于对单个服务的需求来进行的。而在 workflow 中,多个活动之间通常有着一定的依赖关系,某个活动的一些属性通常由其他活动来决定。同时,用户也往往对某几个相邻的服务做出了一些整体上的约束,如果仅从单个服务的角度来进行服务的选择,有时并不能很好地反映用户的需求。3)在 SOA 中,从服务的选择到服务的执行事实上是一个用户和服务提供者共同参与的过程,用户根据自己的需求选择合适的服务,而要想得到使用该服务的效果,就必须按照服务提供商的要求,提供合适的前提条件并遵从正确的方式来调用该服务。但目前的研究更多的只是从用户需求的角度来考虑服务选择问题。在 workflow 中,由于非常强调服务之间的协调,因此这个问题显得更加突出。

针对上面提到的问题,本文在利用服务发现领域现有的一些研究成果的基础上,提出一种新型查找的方法来解决跨组织 workflow 的服务选择问题。本文接下来的内容组织如下:在第 2 节中,先从不同的角度对服务的特性进行分析,接着给出该方法的总体思路以及相关的说明。在第 3 节中,对规则和兼容性比较作出了较为详细的说明,依此进一步提出了一个服务选择算法,并对该算法做简要的分析。在第 4 节中,介绍了相关研究并与该方法做了简要的比较。最后给出了结论以及对未来工作的展望。

2 设计思路

对于 SOA 中的服务,业界有着许多不同的看法,至今没有形成一个统一的定义。与构件等技术相比,服务是一个更贴近业务用户的概念。人们对服务选择的考虑也不再仅仅要求功能上的实现,一些如成本、响应时间等服务质量方面的因素也起着至关重要的作用。另外,为了提高可重用性和松耦合的目的,服务往往被设计成一个大颗粒度软件实体,而不仅仅只是提供某个简单的操作。为了完成该服务提供的某项功能,通常会要求用户在提供必要的前提条件的基础上再进行一系列有序的操作。

针对服务的特点和前述提到的问题,本文拟在现有的研究基础上,采取一种新型的方法对流程中的服务进行选择。在这种方法中,将利用规则(Rule)来反映多个服务之间关联关系的需求,通过兼容性(Compatibility)比较来衡量流程中多个服务能否协调运行。

规则的目的是为了反映流程中服务之间的关联关系。在流程中,相邻服务之间通常存在着一定的约束关系,后继服务的某些属性可能直接由前向服务的状态来决定,因此后继服务的选择应该受到前向服务运行和选择的影响。虽然对于单个服务而言,各自的选择具有很强的动态性,但它们之间的约束关系则是相对稳定的,本文将采用规则来反映这种关系。在服务选择过程中考虑规则对服务属性的约束,从而保证流程中多个服务在整体性上的要求。另外,在流程的整个生命周期中,服务提供商的状态也有可能发生变化,服务候选集合会不断出现新的服务加入、旧的退出或更新等情况。但通常,客户对服务选择的标准相对变化较小。本文也将通过将这种标准表示为规则的形式,让服务选择过程能够适应环境的变

化。

服务的选择与使用是用户与提供者共同参与的过程,目前的研究大多是从用户的角度来考虑服务的选择过程。与之不同的是,兼容性比较则是从服务提供商的角度来考虑。首先,在客户选择服务的同时,提供者也需要衡量客户是否具备使用该服务的必要条件。再者,在使用服务过程中,还需进一步衡量客户能否遵循服务的使用规范。在流程中,服务间的兼容性比较显得尤为重要,流程任务的完成是通过多个服务共同协调完成的。在目前情况下,服务之间的协调性问题都是通过手工来解决的。而当在系统自动绑定服务的情况下,如果选择的服务不能与流程中其他服务协调执行,将会直接导致流程运行的失败。

3 基于规则和兼容性比较的选择算法

本文提出的服务选择算法,主要目的不是代替目前研究中提出的一些算法,如文[4]、[5],而是在该类算法的基础上,针对 workflow 的特点,对该类算法进行改进,使之能解决前面提到的问题。因此,在算法中,流程中服务选择的候选集合设定为满足个体功能需求的服务。针对该候选集合,算法利用规则和兼容性进行筛选,最后得到合适的服务。

3.1 规则

在本文中,将采用规则来反映用户对流程中多个服务之间关联关系的需求。就服务而言,在通常意义下,规则大致可分为反映操作间的和反映属性间的约束关系,如:对于服务 A, B, if A 的操作 1 then B 的操作 2, A 属性 1 = B 属性 2。而在 workflow 中,服务间的操作次序关系一般在过程模型中能得到较好的体现。因此,本文中的规则主要是用来反映服务在属性方面的关联关系。这里,服务的属性主要是指服务在非功能方面的特性,如具体提供商、使用费用、响应时间等方面的性质。

对于规则,本文规定所有的规则都必须具备原子性,即该规则不能进一步拆分成多个子规则。如果进行拆分,则得到的子规则就不能独立,也不能正确反映用户所希望描述的服务间约束关系。对于服务间一些复杂的、不具备原子性的约束关系,则必须将之转换成多个具备原子性的规则来表示。为了叙述的方便,本文采取了一个四元组来对规则进行建模。

定义 1 对于规则 R,可表示为 $R=(O,C,P,T)$ 。其中, O (Ontology)是规则所采用的本体库,通过该本体库指定了描述规则用到词汇的语义; C (Collection)是规则适用的流程集合,领域专家、系统开发人员、普通用户都可以是规则的制定者。不同角色制定的规则对应着不同的适用范围,例如领域专家制定的规则通常具有普遍性,适用于多个不同流程,而普通用户制定的规则只适用某单个流程。集合可通过列举、描述等多种方式来表示。 P (Policy)是具体的规则,如前所述,规则主要是用来反映服务之间非功能属性之间的约束关系。 T (Type)是规则的类型,根据规则涉及到的服务数量 and 是否可协商,本文将规则分为四类:

a) R_s 单服务、强制性的

每条规则反映的约束关系只限于流程中的单个服务,但可涉及该服务中的单个或多个属性。该约束关系是强制性的,即候选服务必须满足该规则,如要求酒店服务 S1:标准=三星 $\wedge$ 标准间费用 < 500 元/天。

b) R_m 多服务、强制性的

与 R_s 不同的是,这类规则反映流程中多个服务属性之间

的约束关系,该约束关系也是强制性的。如要求酒店服务 S1,快餐服务 S2;S1.城市=S2.城市。

c) R_s 单服务、非强制性的

该规则也是反映对单个服务属性方面的要求。与 R_s 不同的是,该规则对候选服务的属性没有一个强制的规定,而是一种选择建议。如对于酒店服务 S1;min(S1.标准间费用)。

d) R_m 多服务、非强制性的

与 R_s 类似,不同的是该规则是反映用户对流程中多个服务属性之间的一种要求。如要求酒店服务 S1,快餐服务 S2;min(S1.标准间费用+S2.费用)。

根据种类的不同,规则在服务选择过程中起着不同的作用。对于非强制性的规则,即 R_m 和 R_s ,目的是在众多候选服务都满足客户要求的情况下,能有一个优选的标准,选择一个最合适的服务,同时也能起到让选择过程适应服务提供商状态变化的作用。对于单服务的规则,即 R_s 和 R_m ,目的是能够反映行业领域中的一些规范、制度、惯例等。在服务选择时,作为隐性的要求对客户显式需求的一种补充。同时也是其他规则的基础,能够最直接有效来缩小服务的候选集合。对于利用多服务规则,即 R_m 和 R_m 来对服务进行选择,则相对较为复杂。在对某服务进行选择时,若该规则中的其他服务及相关属性此时均已确定,则可将该规则转化为单服务规则。若该规则还存在有其他未确定的服务,可利用人工智能中的规划技术该规则涉及到的多个服务同时进行选取^[8]。

由于在上述分类中,存在着非强制性规则,因此有可能会出规则之间相互冲突的现象。为了避免此问题的发生,本文规定了这四类规则在服务选择中的优先次序为: $R_s = R_m > R_m > R_s$,在服务选择时,对于 R_s 和 R_m ,候选服务必须满足。当 R_m 和 R_s 之间存在着冲突时,则采取整体优先于局部的策略进行选择。而当多个 R_m 或 R_s 同类型规则之间存在着冲突时,则由用户指定或由系统给定默认优先级别的方式进行选择。

3.2 兼容性比较

与规则不同,兼容性比较则是从服务提供商的角度来考虑服务选择问题。在流程中,服务的用户通常是其他服务,使用某个服务不仅仅意味着实现期望的功能,往往还需要提供必要的前提条件和遵从适当的调用方式,因此服务之间的兼容对流程的实际可执行性起着至关重要的作用。

兼容性比较就是从这一思想出发,从服务提供商的角度来依次衡量流程中其他与之交互的服务能否提供必要的前提条件且在操作上兼容,从而确定该候选服务是否真正能够在流程中得以执行。

服务间的兼容从大的方面可分为条件兼容和操作兼容。条件兼容指的是在调用某服务时,用户必须提供该服务所要求的必要前提条件,如用户身份信息。而协议兼容则指的是在调用服务的过程中用户必须遵循的操作顺序。在流程中,候选服务的用户是其他第三方服务,同样第三方服务的运行也要遵循自身规定的操作顺序,因此服务间操作顺序的兼容是操作兼容的主要内容。对于条件兼容的判断,本文以用户能否提供候选服务所要求的参数来进行衡量;而对于操作兼容的比较,则包括操作顺序和对应参数的比较。

a) 条件兼容。目的是确认用户在调用候选服务前具备该服务要求的所有前提条件。判断是通过比较用户提供的和候选服务要求的参数的方式来完成,参数的具体内容随候选服务要求不同而相应发生变化,一般包括身份认证等信息。

为了叙述的方便,本文首先定义参数之间的兼容关系。

定义2 对于参数 $p1, p2$,定义一个偏序关系 $p2$ 兼容 $p1$,表示为 $p1 \leq p2$,若满足条件:

1) 当参数 $p1$ 可表示为 $p1 = (t1', t2', \dots, tk')$ 时,参数 $p2$ 均可表示为 $p2 = (t1, t2, \dots, tk, o1, o2, \dots, on)$ 。其中,对于 $i=1$ 到 k , ti', ti 分别为参数 $p1, p2$ 的子项, $o1, o2, \dots, on$ 为 $p2$ 的可选项。

2) 在条件1中,对 $\forall ti'$,都有 ti 语义兼容 ti' 。

说明:对于参数间语义兼容,本体和语义匹配标准起着至关重要的作用。其中,本体给比较的双方提供了描述事物的共同词汇,而匹配标准通常可分为精确相等、子类、超类三种关系,具体的选择要随应用而定。目前,语义匹配有着大量的研究,这里直接引用相关技术。

定义3 在流程 W 中,对于服务 S1, S2,定义 S2 条件兼容 S1,表示为 $S1 \triangleleft WS2$ 。当满足条件:对于参数 $p1, p2$,有 $\forall p1 \in \text{Input}(S1, S2, W)$,都存在着一个参数 $p2 \in \text{Output}(S2)$,使得 $p1 \leq p2$,其中, $\text{input}(S1, W)$ 为在流程 W 调用 S1 相关功能必须提供的参数,即前提条件。 $\text{Output}(S2)$ 为 S2 的输出。

说明:在流程中,一个服务可能会提供多项功能并同时与多个服务进行交互,针对不同的服务调用有着不同的前提条件要求。因此在判断服务间条件兼容时,必须根据流程模型和候选服务自身的描述找出相关前提条件。

b) 操作兼容。在调用服务的过程中,用户必须严格按照候选服务规定的次序依次调用服务提供的各项操作。而作为用户的服务,对自身的操作顺序也有着一定的限制。操作兼容的目的就是衡量这两者之间操作能否相互协调。服务,作为大颗粒度的软件单元,通常能够提供多个不同的功能调用。而对于不同的功能,对应着不同的操作调用次序,这里将操作次序称之为协议(Protocol)。在流程中,对于多个服务之间的一个交互过程,本文称之为一次会话(Session)。会话成功与否,则取决于参与其中的服务能否在操作顺序和对应消息上兼容。

定义4 在流程 W 中,对于服务 S1, S2,定义 S2 操作兼容 S1,表示为 $S1 \trianglelefteq WS2$ 。当满足条件:

1) 对于协议 $\text{prt1}, \text{prt2} \in \text{Sen}$, $\text{prt1} \in S1, \text{prt2} \in S2, \text{prt1} \in \text{prt2}$ 。

2) 对于 prt1 中任意输入参数 $p1$ 和对应 prt2 中的输出参数 $p2$,都有 $p1 \leq p2$ 。

说明:其中, Sen 是在流程中两服务交互产生的会话, $\text{prt1} \in \text{prt2}$ 表示协议 prt1 与 prt2 兼容。对于 $\text{prt1} \in \text{prt2}$ 的判断,目前有着许多相对较为成熟的研究,采取的方式主要是先利用有限状态自动机、Petri 网等对操作的顺序进行建模,然后比较两模型之间的可交互性等方式来确定协议之间的兼容性^[13~15]。

另外,对一些简单、小颗粒度的服务,其交互过程相对比较简单,通常是直接对某个接口的调用。这时,操作兼容的比较则直接可转化为相关参数兼容的比较。

3.3 服务选择算法

综合上述两个方面的考虑,本文提出一个服务选择算法,其中的关键步骤见算法1。算法采取的大致思路是:在得到满足服务个体要求的候选集合基础上,在规则库中找到所有与该服务和流程相关的规则,首先通过强制性规则来衡量服务之间的约束关系是否得到满足,以便对候选集合进行缩减。

进一步,对筛选后得到的每一个候选服务,验证流程中与之交互的其他服务能否与之兼容。最后,根据推荐规则找到一个最优的候选服务。

ProcessModel 过程模型, Candidate 服务候选集合, RuleDb 规则库。

算法 1 服务选择算法

```
0 Service SelectService(ProcessModel, Candidate, RuleDb) {
1 Rule=FindRule(ProcessModel, RuleDb); // Rule=Rule1 ∪ Rule2
  ∪ Rule3 ∪ Rule4;
2 for all r in Rule1 do {
3 Candidate=CheckRule(r, Candidate); //筛选出符合 Rsc 类型规则的
  候选服务
4 if Candidate==Null then return(Fail);
5 }
6 Candidate=Plan(Candidate, Rule2, ProcessModel);
7 if Candidate==Null then return(Fail);
8 For all service in Candidate do {
9 IsCompat=CompareService( service, ProcessModel)
10 if (IsCompat == false) candidate = Cancel (service, Candi-
  date); //从候选集合中删除该服务
11 }
12 service=Recommand(Candidate, R3, R4)
13 }
```

其中, FindRule() 为在规则库中找到所有与该服务和流程相关的规则;在 Plan() 中,利用 R_m 类型规则对候选集合进行处理,对于其中的每条规则,如果该规则中相关的其他服务都已确定,则可直接转化为 R_x 类型的规则对该候选集合进行筛选。若还存在着其他未确定的服务,则利用规则形成的约束关系对相关服务做规划处理;在 Recommend(), 对于 R_x 和 R_m 类型规则,采用与 CheckRule() 和 Plan() 类似的方式对候选服务进行处理;在 CompareService() 中,对每个候选服务与流程中其他与之交互的服务做兼容性比较。如前所述,兼容性比较包括前提兼容和操作兼容两个部分的比较。

算法 2 服务兼容性比较算法

```
0 Boolean ComareService(service, ProcessModel) {
1 Compat=True;
2 For all session in Process {
3 p=FindProctol(Process, session); //找出该候选服务的相应协议
4 p1=FindProctol(Process, service, session); // 找到流程中其他
  与之交互的服务的协议
5 compat=comparecondition (service, process); //前提条件兼容
  比较
6 compat= CompareProctol(p, p1)&&compat; //操作兼容比较
7 if compat==false break;
8 }
9 return compat;
10 }
```

算法分析:在该算法中,由于对候选服务的分析比较深入,计算复杂度相对较大。其中,计算量主要集中在 Plan() 和 CompareService() 子程序上。在 Plan(), 计算量随着规则中涉及的未确定服务个数以及相关候选服务集合的大小呈指数级数增长。目前,改进该方法采取的主要途径是尽可能利用 R_x 类型规则来减小服务候选集合。而在 ComareService() 和 Recommend() 中,虽然同样面临着该问题,但经过 R_x 和 R_m 类型规则的筛选后,通常服务候选集合相对较小,其计算复杂度问题不是影响整个算法性能的主要因素。

4 相关研究与比较

目前,服务发现是 e-service 研究的热点问题。事实上,在服务发现中,服务选择与服务描述采取的方式是紧密相关的。目前,对服务的描述基本上可分为两大类:一类是以 WSDL(Web Service Descript Language)为代表的方式。在这类方式中,将服务简单地定义为一组实现某种功能的接口,对服务的调用即为对接口的简单调用。在服务选择时,只是从语法的角度采取关键字匹配的方式进行查找,存在着查全率

低、语义冲突等诸多弊端。而另一类则是以 DAML-S 及其后继 OWL-S 为代表的方式^[16]。在这类方式中,利用本体从语义的角度来描述服务的功能,并对服务的实现过程和调用方式有着更为详细的说明。目前,以此为基础的各种服务查找算法不断出现。

在流程中,多个服务之间形成的关联与协调关系对服务的选择提出了进一步的要求,但目前从这一角度出发来考虑服务选择的研究相对较少。其中,比较有代表性的工作包括[8,10,11]。在文[8]中,作者从服务质量的角度出发,给出了一个服务质量模型以及流程整体服务质量的计算方法,并根据流程整体要求而对多个候选服务形成的约束关系列出了相关的线性不等式,采取线性规划的方式计算出一个最优的服务选择方案。但在该方法中,由于流程中所有服务参与规划算法,随着服务候选集合的增大,计算复杂度成倍增长。本文则是对规则中涉及到的服务做规划,将一个大的规划问题分成多个小的规划问题,能有效降低计算量。在文[10]中,也利用了规则来指导服务的选择,但该规则主要是反映服务提供商的服务使用策略上的限定关系,而本文的规则主要是从用户的角度来反映需求。对于服务的调用协议模型,主要可分为两大类:一类是采取 Mealy 机为代表的,主要强调的是消息的交换次序;另一类以 Roman 模型为代表的,强调各项操作的顺序。在服务操作兼容比较的过程,本文采用了文[14]中的方法,用有限状态自动机的方式对服务操作过程进行描述并进行兼容比较,同时本文也强调前提条件兼容的必要性。

结论与进一步的工作 在跨组织的工作流中,服务的选择有着不同于其他服务发现的一些特点,在考虑单个服务自身的要求外,还应该考虑到服务之间的约束关系和相互协调对服务选择的影响。本文就是从这个角度出发,一方面利用规则来反映用户对服务之间约束关系的需求,通过对规则的分类,明确了各个约束关系对服务选择过程中的作用。另一方面针对每个候选服务,衡量流程中与之交互的服务能否提供必要的调用条件和遵循相应的调用协议。除了跨组织的工作流,本文所作的工作对于服务的自动或者半自动合成也有着一定的借鉴意义。

由于本文提出的选择算法还处于原型系统设计阶段,对于多服务类型规则中涉及到多个未确定服务、在服务交互中多于两个服务共同参与等复杂情况并未做深入的研究,这也是本文进一步工作的方向。

参 考 文 献

- 1 Alonso G, Casati F, Kuno H, et al. Web Services Concepts, Architectures and Applications. Springer Verlag, 2004
- 2 van der Aalst W M P. Don't go with the flow; Web services composition standards exposed. IEEE Intelligent Systems, 2003
- 3 Orlowska M, Weerawarana S, Papazoglou M P, et al. Service-Oriented Computing. In: ICSOC 2003, First International Conference, Trento, Italy, Proceedings Springer, 2003
- 4 li Le, Horrocks I. A Software Framework for Matchmaking based on Semantic Web Technology. In: Proceeding of the WWW 2003 Conference, Budapest, 2003. 331~339
- 5 Paolucci M, Kawamura T, Payne T R, et al. Semantic Matching of web services capabilities. In: Proceedings of the First International Semantic Web Conference on The Semantic Web Sardinia, Italy June 2002. 318~332

(下转第 136 页)

则流程转至位置 c5。如果无需部门预先处理则可直接将文件送交院级领导批示(流程转至位置 c13)。文件在部门处理过程中,首先由部门办公室秘书转办,然后由部门领导返回处理意见。系统在部门领导返回处理意见后检查是否还有待处理部门,如有则流程回到位置 c9,再由系统自动送往其他的待处理部门;如果没有则流程转至 c13,文件将送交院级领导批示。

文件经院级领导批示时,由领导解决是否需要再送某些

部门提出处理意见最后综合考虑。如果需要则流程转至位置 c8 由联系秘书选择需提出处理意见的部门;否则文件被送往在拟办(或批示)过程中指定的主办部门,同时由系统自动将文件送交需协办的部门(位置 c16)。最后文件由主办部门处理并给出办理结果(位置 c18)。最后,所有经系统处理过的文件在预定义的时间后自动存档。整个收文业务流程最后终止于节点 o。

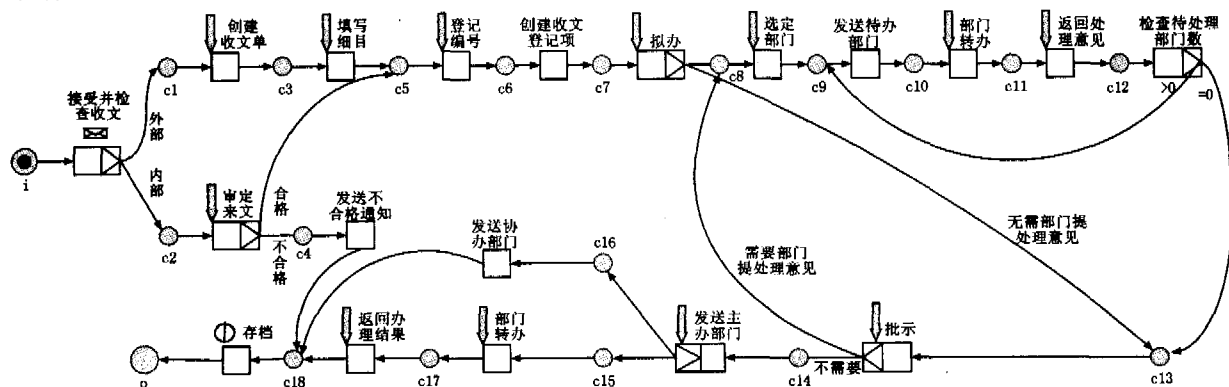


图4 用WF-net建立的收文过程模型

结束语 本文介绍了 petri 网的起源及定义,讨论了由 petri 网发展起来的 WF-net 的定义、工作流执行结构及任务触发方法。以高职院校收文为背景,介绍 WF-net 建立工作流过程模型的应用。通过此项技术,我们已成功开发出一套职业技术学院办公自动化系统并投入使用。通过学习和实践,我们认识到,使用 WF-net 对于复杂工作流程的设计和分析是比较先进和科学的,其特点主要体现在^[2]:(1)提供了统一的表示方法来描述系统的各个特性,对于非专业人员而言,在直觉上容易理解和应用,而对于专业人员来说,又提供了强大而又形式化的描述能力;(2)WF-net 可以方便地进行层次化的建模,使得复杂的问题在层次化变得比较简单,这适合于

自顶向下的建模及各个阶段的独立验证和确认;(3)WF-net 有坚实的理论基础和比较成熟的分析方法,确保了流程的有效性、正确性和可复用性,缩短了软件开发周期,提高了工作效率。熟练运用此项技术,在保证系统安全性、合理化的同时,可达到缩短开发周期、提高开发效率的目的。

参考文献

- 1 van der Aalst W, van Hee K 著. 工作流管理—模型、方法和系统 [M]. 王建民, 闻立杰等译. 北京: 清华大学出版社, 2004. 28~30
- 2 张卫兵, 宋顺林, 等. 基于 petri 网公文流转工作流模型的设计 [J]. 计算机工程, 2004, 30(2): 193~195

(上接第 133 页)

- 6 Gao Xiang, Yang Jian, Papazoglou M P. The Capability Matching of Web Services. In: Proceedings of the 4th international symposium on Multimedia software engineering Newport Beach California, USA, 2002
- 7 Mecella M, Pernici B, Craca P. Compatibility of e-Services in a Cooperative Multi-platform Environment. In: Technologies for E-Services; Second International Workshop TES 2001, Rome, Italy, 2001
- 8 Zeng Liangzhao, Benatallah B, Dumas M. Quality Driven Web Services Composition. In: Proc. of the 12th International Conference on the World Wide Web (WWW). Budapest, Hungary. ACM Press, 2003
- 9 Oaks P, ter Hofstede A, Edmond D. Capabilities: describing what services do. In: Proc. of ICSOC-2003 (1st Intl. Conf. on Service-oriented Computing), Trento, Italy, 2003
- 10 Chun Soon A, Atluri V, Adam N R. Policy-based Web Service Composition. In: Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications, Boston, USA, 2004. 85~92
- 11 Hacid M S, Benbernou S. Resolution and Constraint Propagation for Semantic Web Services Discovery. Journal of Distributed and Parallel Databases (DAPD) / Special Issue on Web Services, 2005
- 12 Liu Chuang, Foster I. A Constraint Language Approach to Matchmaking. In: 14th International Workshop on Research Issues on Data Engineering (RIDE 2004), Boston
- 13 Benatallah B, Castai F, Toumani F. Analysis and Management of Web Service. In: Protocols Proceedings of 23rd International Conference on Conceptual Modeling, Shanghai, China, 2004. 524~541
- 14 Wombacher A, Fankhauser P, Mahleko B, et al. Matchmaking for Business Processes. IEEE International Conference on E-Commerce, 2003
- 15 Bultan T, Fu X, Hull R, et al. Conversation specification: a new approach to design and analysis of e-service composition. WWW'03. ACM, 2003
- 16 The Business Rules Group. Defining Business Rules: What Are They Really? July 2000. <http://www.businessrulesgroup.org/>
- 17 OWL-S: Semantic Markup for Web Services. <http://www.daml.org/services/owl-s/1.0/owl-s.html>, Nov2003
- 18 RuleML Website. <http://www.ruleml.org/>