

基于位置服务的工作流任务分配方法^①

吉朝明, 汪 松

(四川交通职业技术学院 现代教育技术中心, 成都 611130)

摘 要: 在工作流管理系统中, 任务分配控制策略对系统的性能影响较大. 为了缩短任务执行时间, 提高流程运行效率, 提升工作的客户满意度, 本文在前人研究成果的基础之上, 综合考虑任务参与者的工作负载、任务类型等因素, 提出了一种基于位置服务的工作流任务分配方法, 较好的解决了流程中任务分配的问题, 保证任务被尽早、尽快、高质量地完成.

关键词: 工作流; 任务分配; 位置服务

Workflow Task Assignment Method Based on Location Based Service

JI Chao-Ming, WANG Song

(School of Modern Education Technology Center, Sichuan Vocational and Technical College of Communications, Chengdu 611130, China)

Abstract: Task assignment strategy has a great impact on the performance of the workflow management system. In order to reduce the execution time of task, improve the operation efficiency of process and promote the customer satisfaction, this paper proposes a workflow task allocation method based on the location service, which is based on the basis of predecessors' research results. This method can solve the task assignment problem of workflow and ensure the task be finished with high quality as soon as possible.

Key words: workflow; task assignment; location based service

随着网络技术的发展, 工作流技术已被广泛运用到各类信息管理系统当中. 工作流将业务过程从应用程序中分离出来, 进行独立管理, 使得软件更容易支持多变的业务流程, 或更高效地并行处理任务, 从而提高工作效率. 工作流管理系统的一个重要任务就是在适当的时间将需要人工参与的任务交付给合适的执行者. 而对于人工任务, 工作流管理系统必须采用一定的任务分配算法将任务分配给合适的执行者, 从而推动流程实例的继续运行. 因此, 设计一个合理的任务分配算法可以将任务分配给最合适的执行者, 能够缩短任务的执行时间, 提高流程的执行效率, 提升工作的满意度.

1 研究现状

工作流当中任务分配的研究领域主要分为两个方

面: 任务分配控制和任务分配规则描述. 本文关注的是任务分配控制的研究, 即任务在运行过程中根据何种策略选择任务的执行者. 目前, 针对工作流中任务分配控制这一问题, 国内外已经进行了较为深入的研究, 根据研究成果大致可以分为两类: 第一类是在流程定义文件中直接指定每个任务节点上对应的角色、组织或个人, 这样工作流引擎可以方便快捷地对任务参与者进行调度, 流程逻辑比较简单, 但是这种方式不能够根据用户的需要, 在流程执行过程中将任务根据具体情况动态分配给活动参与者, 这样就无法满足用户对任务分配的一些不确定性需求, 限制了系统的灵活性; 另一类就是在任务执行时, 综合考虑工作人员的工作能力、负载情况、任务类型, 以及相关的回避策略等因素由工作流引擎按照规则进行动态的分配, 这种方式实现了任务的动态分配^[1].

^① 基金项目: 四川省教育厅自然科学基金项目(16ZB0524)

收稿时间: 2016-05-06; 收到修改稿时间: 2016-06-12 [doi: 10.15888/j.cnki.csa.005564]

对于任务的动态分配情况,文献[2]在任务分派时不仅考虑当前待分配任务的候选资源的情况,还考虑了其前置执行者对候选资源的影响,其中用到了最短处理时间、最短完成时间、平均工作负载和最短工作列表这 4 种传统、简单的任务分配算法,但其未能充分考虑任务分配过程中的负载均衡问题.文献[3]在任务分配时不仅基于当前待分配任务选择执行的资源,也基于整个案例流程进行考虑,根据历史的执行记录优先选择合作次数多的资源共同执行流程,也未能充分考虑资源的负载情况.文献[4]在对任务参与者进行负载预测的基础上,综合考虑任务参与者的工作负载、对不同类型任务的完成质量和兴趣等因素,根据预测负载偏差,对任务参与者的负载进行等级划分,提出一种基于任务参与者负载平衡和经验值的分配策略.文献[5-7]也都提出对任务进行动态分配,能够综合考虑任务参与者的工作负载、对不同类型任务的完成质量和兴趣等因素,比较符合现实意义.

在实际应用当中可能存在业务所涉及地域范围较广的情况,企业需要在最短的时间内分配工作人员到达现场执行任务,提高客户的满意度,所以单从任务参与者的工作负载、对不同类型任务的完成质量和兴趣等因素可能无法完全满足实际需求.因此,在前人研究成果的基础之上,综合考虑任务参与者的工作负载、任务类型等因素,本文提出了一种基于位置服务(Location Based Service, LBS)的工作流任务分配方法,缩短任务参与者参与任务的时间,提高工作效率,以保证任务被尽早、尽快、高质量地完成,从而提升任务完成的满意度.

2 基于位置服务的工作流任务分配方法

伴随无线通信技术和智能移动终端的快速发展,基于位置的服务以其移动性、实用性、随时性和个性化的特点,在军事、交通、物流等诸多领域得到广泛应用^[7].基于位置的服务,是指通过移动终端和移动网络的配合,识别出移动用户的实际地理位置,从而提供用户所需要的与位置相关的信息的服务形式^[8].

本文所提出的基于位置服务的任务分配方法,就是在工作流的人工任务分配过程中,通过实时获取工作人员的位置信息,从而获知任务位置附近的工作人员,再结合其他诸多因素,将任务分配给最合适的工作人员,具体的方法流程如下图 1 所示.

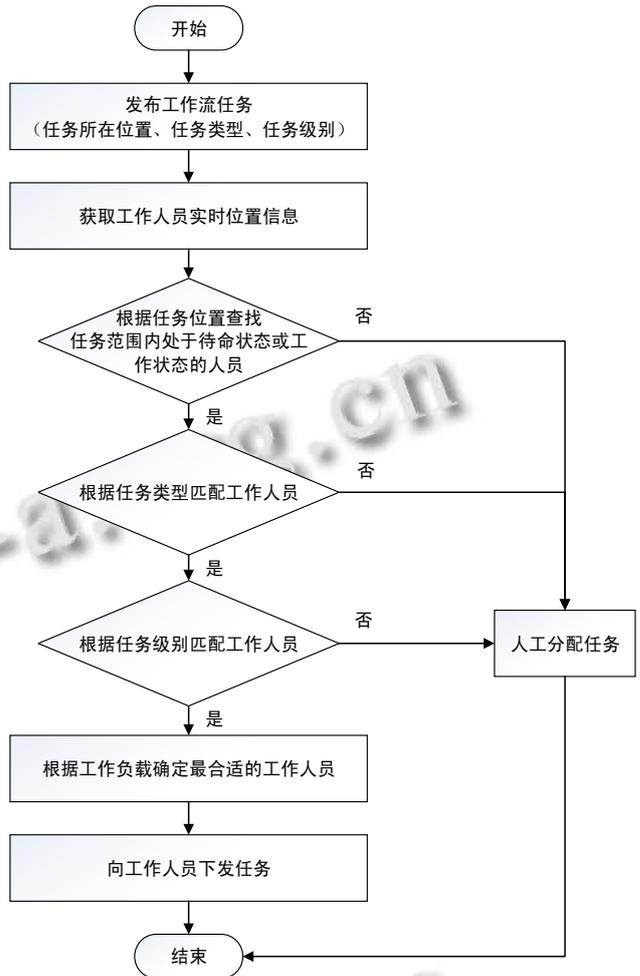


图 1 基于位置服务的工作流任务分配流程

(1) 任务创建人在创建任务时,需要填写任务所在的地址信息、任务类型以及任务级别信息.其中,任务类型用于表示执行该任务所需的技能类型;任务级别表示执行该项任务,工作人员所需的技能成熟度.为了方便描述,对新建的任务 J_{new} 进行如下形式化定义,将任务 J_{new} 定义为一个四元组:

$$J_{new} = \{ID, e, C_s, R_j\}$$

其中 ID 表示任务标识号; e 表示任务的重要程度,值越大表示重要程度越高; C_s 表示此任务所含任务类型的集合; R_j 表示任务级别的集合,指此任务中每种任务类型的难度级别.

(2) 通过移动终端获取工作人员的实时位置信息,并根据所发布的任务位置查找该范围内的工作人员.在此过程中,每个工作人员都需要配置移动终端,用于实时记录并上报自己所在的具体位置和状态信息.移动终端每隔一段时间通过 WiFi 或移动网络向工作

流管理系统发送位置信息、状态信息以及时间信息。若无网络连接,则将相应的信息记录在缓存中,待有网络时再统一发送。为了便于描述,将工作人员形式化定义为一个五元组:

$$U = \{ID, s, J_s, S_k, E_{xps}\}$$

其中 ID 表示工作人员标识号; s 表示工作人员的状态; J_s 表示工作人员已有的任务集合; S_k 表示工作人员具有的技能类型集合; E_{xps} 表示工作人员完成各类任务的经验集合。

(3) workflows 管理系统根据人员状态、任务类型、任务级别查找可执行任务的工作人员。

步骤1: 状态匹配。系统获取工作人员的状态信息,并查找任务范围内处于待命状态或工作状态的人员并添加到候选队列中。工作人员状态信息包括工作状态(working)、待命状态(waiting)、离线状态(offline)以及请假状态(leaving)。工作人员需要随身携带移动终端,并登录相应的应用程序。如果应用程序关闭后,系统自动识别并将该工作人员标记为离线状态。若工作人员通过移动终端上报请假信息,系统则标记为请假状态;如果工作人员在任务中,则标记为工作状态;如果工作人员无任务且无请假,则标记为待命状态。在任务创建后,系统自动检测工作人员的状态信息,在任务范围内查找处于待命状态或工作状态的人员,如果未找到则扩大查找范围,直至到设定的任务范围为止;若在任务范围内没有查到待命状态的人员,则给任务管理人员发送提醒信息,提示手工分配任务。

形式化描述: 从设定范围内获取到所有工作人员集合 $W(U)$, 从中查找出工作状态 s 为 working 或 waiting 的工作人员, 得到新的候选工作人员集合 $S_1(U)$ 。

步骤2: 任务类型匹配。判断候选队列中工作人员的技能类型是否匹配发布的任务类型。在本文中系统预先设置所有人员的信息,包括技能类型、技能成熟度。在筛选出待命状态或工作状态的人员后,查看其技能类型是否与任务类型相匹配。若不匹配,则继续扩大范围查找,直至在任务范围内查找到符合条件的人员为止。若在任务范围内没有匹配到相应任务类型的人员,则给任务管理人员发送提醒信息,提示手工分配任务。

形式化描述: 根据 J_{new} 任务中的任务类型集合 C_s , 从工作人员集合 $S_1(U)$ 中逐一筛选出其技能集合 S_k 包

含有 C_s 的工作人员, 得到工作人员集合 $S_2(U)$ 。

步骤3: 任务级别匹配。判断人员技能成熟度是否与任务级别所需的技能成熟度相符。本文中工作人员的技能成熟度通过其成功执行的任务数量进行判断。当工作人员成功执行完某个任务时,系统将该技能类型的任务数量加1。当成功执行任务的数量达到设定值时,则更改工作人员的技能成熟度。系统根据任务级别查找该技能成熟度范围内的工作人员,如果找到,则将其添加到候选队列中;如果未查找到则给任务管理人员发送提醒信息,提示手工分配任务。

形式化描述: 根据 J_{new} 任务中难度级别集合 R_j , 从工作人员集合 $S_2(U)$ 中筛选出 E_{xps} 中每种技能经验值都高于或等于任务 R_j 中的每种任务难度, 得到工作人员集合 $S_3(U)$ 。

(4) 根据候选队列中工作人员的负载程度确定合适的工作人员,并向工作人员下发任务指令。对于工作人员负载程度的衡量,在本文具体实现部分说明。

3 具体实现

3.1 位置服务

本文使用百度地图 API 来实现地理信息的可视化,其具体的实现流程如下:

(1) 系统初始化,对工作人员进行实时定位操作。百度地图 Android SDK 在使用之前需要申请 API key,因此在应用程序进行初始化操作时,需对百度地图 API Key 进行验证操作,同时添加位置改变监听器,如果输入的 API Key 不正确,则不能执行定位功能。同时,在程序中通过对监听器接口的 onLocationChanged() 方法进行重写,实现定位功能,得到经纬度值。利用百度地图 API 提供的异步函数 reverseGeocode(), 将经纬度对应的实际地址信息在 MKSearch 类的 onGetAddrResult() 方法中给出,并将定位信息返回给系统,更新到系统数据库中。

(2) 当任务创建人提交任务以后,系统通过调用百度地图的 Geocoding API 实现地理位置的编码,将所发布任务的位置信息转换为经度和纬度值。

(3) 以任务所在的经度和纬度为中心,检索周边一定范围内的工作人员。在具体实现过程中通过初始化 MKSearch 类,注册搜索结果的监听对象 MKSearchListener,实现异步搜索服务,而周边检索则需要调用 MKSearch 类的 poiSearchNearBy() 方法检索以指定坐标点(任务的位置信息)为圆心的一定半径范围内的全部兴趣点。检索服务使用完成之后,调用

MKSearch 的 destory()方法来释放资源。

当系统查找周边的工作人员时,需要工作人员登录客户端才能获取到其经纬度信息.本文以 10km 为初始搜索半径范围,当在该范围内没有找到处于待命或工作状态的人员时,系统将查找半径扩大为 20km 进行查找,以此类推,直到达到所设定的搜索范围为止.若达到最大范围还未找到处于待命或工作状态的工作人员,则给任务管理人员发送提醒信息,提示其进行人工分配任务;若检索到相应的工作人员,系统返回 JSON Object 数据类型,分别有 msg_detail 和 result,返回的数据如表 1 所示.

表 1 返回数据类型

返回值(msg_detail)	描述
MSG1000	未通过认证,不能调用
MSG2000	精读不允许为空
MSG3000	纬度不允许为空
MSG4000	获取周围工作人员信息失败,请稍后再试
MSG5000	周围没有工作人员信息
MSG6000	登录成功,此时才会返回 result
返回值(result)	
worker_no	工作人员工号
worker_name	工作人员姓名
carry_load	工作人员负载
gis_time	最近上传位置信息的时间

3.2 负载量化

在系统运行过程中,由于人力资源的有限性,任务的数量将远大于工作人员的数量,因此同一个员工在同一时刻,可能同时占有多个任务,为了协调任务之间的关系和调整任务的处理次序,避免任务调度倾斜,保证任务能够及时得到处理,在实施任务分配的过程中,系统设计需要考虑工作人员的负载,使任务分配更加合理可行.而衡量用户负载的因素有很多,如用户当前的任务总数、用户的能力和经历、任务的属性等.为了简单起见,本文中用户负载由用户当前的任务总数和用户执行每个任务的工作负载共同组成,任务的工作负载由用户历次完成此类任务的平均时间来衡量,并根据任务的重要程度给定一个影响因子,目的是使工作负载相同但重要性不一致的两个任务对用户负载产生不同的影响,认为重要性高的任务产生较高的用户负载^[9].

文献[9]探讨了如何对用户的负载进行量化,对于待分配的新任务 J_{new} ,其任务的重要程度为 e_{new} ,用户 U_i 当前任务集合 T_s 中已有的任务总数为 l ,用户 U_i 执行第 j 个任务的平均执行时间用 $t_i(J_j)$ 来表示,则用户 U_i 的当前绝对总负载 $Load(U_i)$ 可以表示为:

$$Load(U_i) = \sum_{j=1}^l \lambda(e_j) * t_i(J_j), J_j \in T_s$$

其中 $\lambda(e_j)$ 表示第 j 项任务的重要程度 e_j 对工作负载的影响因子,如果 e_j 不大于 e_{new} ,影响因子为 1;否则随着 e_j 的增大而缓慢增大.

计算出每个用户的绝对总负载后,进行归一化处理,得出每个用户的相对总负载

$$Load(U_i) = \frac{\max(Load(U)) - Load(U_i)}{\max(Load(U)) - \min(Load(U))}$$

式中, $\max(Load(U))$ 表示候选用户集合中用户负载最大值, $\min(Load(U))$ 表示候选用户集合中用户负载最小值. $Load(U_i)$ 越大,表示用户的负载越小,越容易获得新任务.

4 结语

本文所设计的基于位置服务的任务分配方法,可以根据工作人员的位置信息、任务负载程度、技能类型以及技能成熟度等信息自动分配工作任务,极大的降低了任务分配时间成本以及中间的沟通成本,提高了工作效率,提升了工作的满意度.

参考文献

- 1 钱鹰,王寸涛,韦庆杰.一种用于 workflow 引擎的任务预测与分配算法.计算机应用与软件,2014,31(8):75-78.
- 2 Xu JX, Huang ZG, Yu Y, Pan ML. A performance analysis on task allocation using social context. Proc. of the 2nd Int'l Conference on Cloud and Green Computing (CGC). IEEE. 2012. 637-644.
- 3 Yang HD, Wang CK, Liu YB, Wang JM. An optimal approach for workflow staff assignment based on hidden Markov models. Proc. of the Move to Meaningful Internet Systems: OTM 2008 Workshops. Berlin, Heidelberg. 2008. 24-26.
- 4 刘怡,张戡.基于负载平衡和经验值的工作流任务分配策略.计算机工程,2009,35(21):57-59.
- 5 Ho CJ, Vaughan JW. Online task assignment in crowdsourcing markets. Proc. of the AAAI. Toronto, 2012. http://www.cs.ucla.edu/~cjho/pub/AAAI12_TaskAssignment.pdf.
- 6 Ho CJ, Jabbari SH, Vaughan JW. Adaptive task assignment <http://jmlr.csail.mit.edu/proceedings/papers/v28/ho13.pdf>.
- 7 Björnson E, Jorswieck E. Optimal resource allocation in coordinated multi-cell systems. Foundations & Trends® in Communications & Information Theory, 2013, 9(2).
- 8 唐科萍,许方恒,沈才樑.基于位置服务的研究综述.计算机应用研究,2012,29(12):4432-4436.
- 9 姜劲松,杨波,缪志敏,朱宝山.基于任务和用户属性的 workflow 任务分配算法.计算机仿真,2015,32(12):222-225.