



车联网环境下基于 MAS 的行车主动服务模型

梁军^{①②*}, 赵振超^①, 陈龙^①, 徐永龙^①, 朱宁^②

① 江苏大学汽车工程研究院, 镇江 212013;

② Department of Mechanical, Shizuoka Institute of Science and Technology, Fukuroi, Shizuoka, 437-0032, Japan

* E-mail: liangjun@ujs.edu.cn

收稿日期: 2016-02-20; 接受日期: 2016-11-04; 网络出版日期: 2016-11-28

国家自然科学基金项目(批准号: U1564201, 6157371, 51108209)、交通运输部信息化项目(编号: 2013-364-836-900)、中国博士后科学基金(编号: 2016M600375)、江苏省高校优势学科建设工程项目(PAPD)、江苏省六大人才高峰项目(编号: DZXX-048)、全国统计科学研究项目(编号: 2014596)、江苏省政府留学基金项目(编号: JS-2012-253)和江苏省自然科学基金项目(编号: BK20140570)资助

摘要 针对当前“请求-响应”被动式行车服务存在的个性化、人性化、实时性等不足, 提出了基于多智能体(multi-agent system, MAS)分层集中控制的“自动识别-主动推送”形式的行车主动服务系统(driving active service system, DASS). DASS 将行车环境实体与服务作为建模对象, 提出基于规划图的行车服务需求模型及用户需求与服务能力匹配算法. 以交通安全预警和实时动态路径规划为例, 通过自主搭建的驾驶员在环半实物联合仿真平台对 DASS 典型应用展开实验研究, 充分验证了 DASS 的实时性、主动性与高效性. 这对于降低交通拥堵、提升通行效率, 加强车辆安全以及改善驾驶体验具有重要价值.

关键词 行车主动服务, MAS 分层控制, 车联网

1 引言

传统行车信息服务系统(traditional traffic service systems, TTSS)是指通过传感器、RFID、车辆定位等技术, 实时感知车况及控制系统、道路环境、车辆与车辆、车辆与人、车辆与道路基础设施、车辆当前位置等信息, 为用户提供终端信息服务^[1,2]. 目前常见的 TTSS 主要有日本丰田公司的 G-BOOK 系统(开创了 TTSS 的先河)、通用的 Onstar、苏州金龙 G-BOS、陕汽重卡天行健、宇通客车安节通等. TTSS 在极大满足人们出行需求的同时, 也带来了一些问题, 比如, 误报、延报等对正常驾驶的干扰, 导致驾驶员关闭

TTSS 功能.

最近两年来, 能够对行车用户的服务需求进行主动的、清晰的、有效的感知与辨识, 再通过车联网中各个网络节点的信息交换, 自主生成或搜寻服务资源, 最终实现向用户主动推送相应的服务并对服务质量提供有效保障的行车主动服务系统(driving active service system, DASS)正成为车联网远景规划中重要的组成部分和研究热点^[3]. 与 TTSS 相比, DASS 有着不同的工作模式. TTSS 采用“请求-响应”被动服务模式(passive service model, PSM), DASS 采用“自动识别-主动推送”的行车主动服务模式(active service model, ASM). PSM 需要驾驶员主动进行服务

引用格式: 梁军, 赵振超, 陈龙, 等. 车联网环境下基于 MAS 的行车主动服务模型. 中国科学: 技术科学, 2016, 46: 1303-1314
Liang J, Zhao Z C, Chen L, et al. The driving active service system model based on MAS for internet of vehicle environment (in Chinese). Sci Sin Tech, 2016, 46: 1303-1314, doi: 10.1360/N092016-00066

的查询和请求, 极大增加了服务到达的时间和产生事故的风险. ASM 则由系统自动实现从信息感知到服务推送的全过程, 无需驾驶员的参与, 使得驾驶员可以更专注于驾驶行为, 提高了服务效率、减少了驾驶员的等待时间, 同时降低了潜在的交通事故风险.

ASM 是目前车联网研究的一个主要方面^[4~6]. 2014年, Birrell 等人^[7]在实际道路上对一款车内智能驾驶辅助系统进行了测试, 车联网环境下采集周围车辆的行驶信息、车内用 OBD 读取自车行驶数据, 驾驶辅助系统根据这些数据能主动地推荐驾驶员加速、换挡等. 测试结果表明, 装有该辅助驾驶系统车辆能有效降低油耗. 为推动和加速 ASM 的研究与发展, 2014年8月, 美国国家科学基金委员会(NSF)一期资助 6000 万美元在 UCLA 成立了车联网信息服务研究中心(CASN), 联合周边大学展开了“车联网主动式服务模型”的研究项目, 重点研究基于专用短程无线通信(DSRC)的行车服务¹⁾. 2015年2月, 麻省理工大学主办的非营利性技术评论杂志将 DASS 总结为改变未来人类生活方式的 10 种新兴技术之一²⁾. 2015年3月, 美国《商业周刊》也将 DASS 列为有可能掀起车联网服务产业新的效益增长点和未来车辆四大关键技术之一³⁾. 2015年10月, Sakthi 和 Kumar^[8]研究了云计算和车联网环境下的车载云服务系统, 设计了一个 3 层云服务模型, 在仿真环境下创造了一个无事故交通系统. 国内对行车主动服务实现关键技术展开研究的主要是清华大学、吉林大学、北京交通大学、同济大学、长安大学、江苏大学等高校和其他研究机构. 张尧学和方存好^[9], 韦理等人^[10]提出了支持主动服务的描述模型, 并给出了具体应用实例. 在此基础上, 文献[11]提出了一种服务挖掘模型, 有针对性地进行服务资源的动态匹配. 文献[12]提出云计算平台异构服务整合环境下基于失效规律的节点资源动态提供策略, 并设计实现了一个整合异构负载的云计算模拟器平台和系统资源的多维度失效模型框架, 但有效性还有待试验和工程实践的进一步验证. 北京大学的莫同博士等人^[13]提出了一种事件驱动的主动服务发现方法, 并以机场候机提醒为例验证了

这种情境感知方法的有效性, 但文章没有对交通领域的行车主动服务进行研究.

本文以 Agent 的形式设计了 DASS 模型框架, 对行车环境实体和服务进行建模, 深入研究了车联网环境下 DASS 实现主动服务的关键算法, 并以交通安全预警和实时动态路径规划为例对 DASS 进行了仿真验证. 仿真结果表明 DASS 具有实时性、主动性、高效性的特点, 对提高交通安全、宏观交通环境调控有重要的意义.

2 DASS 框架

2.1 行车信息流模型

行车过程中, 交通信息的传输与处理对于行车主动服务的实现至关重要. DASS 接收的信息主要包括: 车辆传感器信息, 如速度、加速度、位置、环境参数等; 路侧设备提供的信息, 如道路交通量、车流密度、信号灯位置及状态, 车辆、障碍物、行人的状态等; 网络传送的紧急事件, 如汽车抛锚、交通事故等; 为汽车服务的社会资源信息, 如停车位、加油站、消费场所位置等. 图 1 为建立的行车信息流模型. 信息流在行车用户、车辆和环境之间有效的流通, 各个部分的协调控制作用能够实现对车辆进行实时的调控, 实时、高效、个性化地为用户提供所需要的行车服务.

2.2 车辆 Agent 模型

车辆 Agent 兼具车辆和驾驶员二者的特性, 是一种根据行车环境中的驾驶行为而抽象出来的智能实体, 图 2 为其结构图. 它可实时获取交通状况信息, 并拥有自己的行为知识库和决策判断能力, 可根据周围的交通情况自主地调整驾驶行为, 从而在确保行车安全的基础上提供个性化的行车服务.

车辆 Agent 包括静态的物理属性和动态的物理指标. 静态属性包括车辆型号、外形尺寸、轮胎半径、整备质量等. 车辆的动态特征是由于驾驶员对车辆的操纵, 从而使车辆表现出随时间变化的特征, 其中

1) Estrin D, et al. "CASN annual progress report 2015". <http://www.casn.ucla.edu/>

2) Van der Werff T J. Ten emerging technologies that will change the world. MIT's Technology Review. <http://www.globalfuture.com/mit-trends2015.html>

3) Green H. Telematics service industry revolution. Business week online. http://www.businessweek.com/magazine/content/2015_3_4/b3846622.html

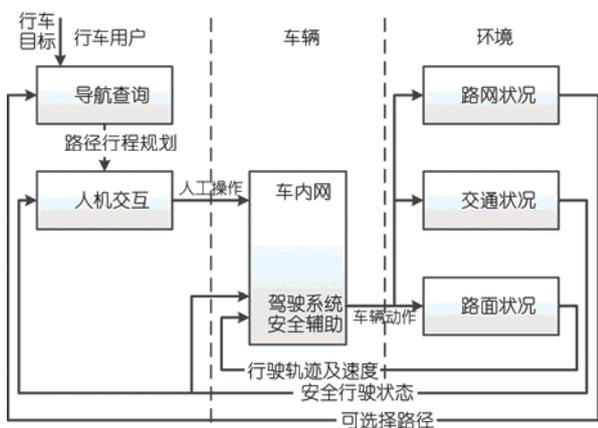


图1 行车信息流模型

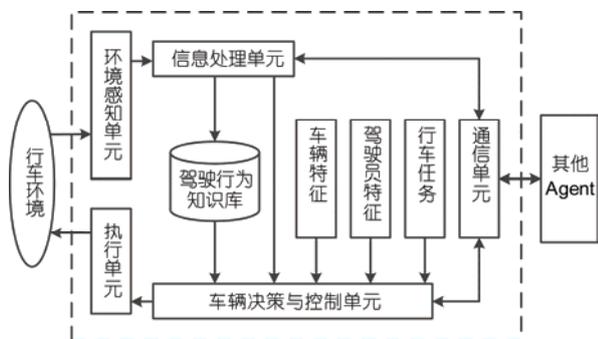


图2 车辆 Agent 结构图

加速度、速度、位置坐标是最重要的动态属性. 此外, 车辆 Agent 还包含了车辆的间接动态属性, 如: 跟驰行驶、车道变换、制动停车等状态. 这些静态和动态的属性, 决定了车辆 Agent 对环境的感知、对感知的决策和对决策的执行, 其最终结果是适应环境、作用于环境甚至改变环境.

2.3 基于 MAS 的 DASS 模型

在建立了行车多源信息模型与车辆 Agent 模型后, 提出了基于 MAS 分层控制的 DASS 模型框架. 如图3所示将整个 DASS 模型分为平台感知层、协调执行层、事件代理层、交互界面层等. 各处理模块以 Agent 的形式进行设计, 并明确各 Agent 功能, 同时为 Agent 之间的通信增设代理, 集中处理层与层之间消息的存储与转发.

为了更好地满足用户行车的个性化需求, 系统获取并记录用户常用的服务、使用习惯、交互行为等信息, 为每个用户建立一个用户 Profile 配置文件, 并

存储到事件代理层的知识库中, 作为识别用户身份和提供个性化服务的基础. 通过对 Profile 配置文件信息进行数据挖掘, 建立用户知识库, 可以为用户行车需求的输入、服务的选取、访问和实施提供参考; 当用户提出新的服务请求时, 系统也可根据用户 Profile 中记录的用户使用习惯和兴趣, 在与用户交互过程中提供建议, 更好地满足用户个性化服务的需求.

2.4 车辆换道控制模型

车辆换道是一种最基本的交通状态变换行为. 图4为车辆换道控制模型, 在微观交通流下的 DASS 服务: 单个车辆的行车需求发生在车辆驶入或驶出

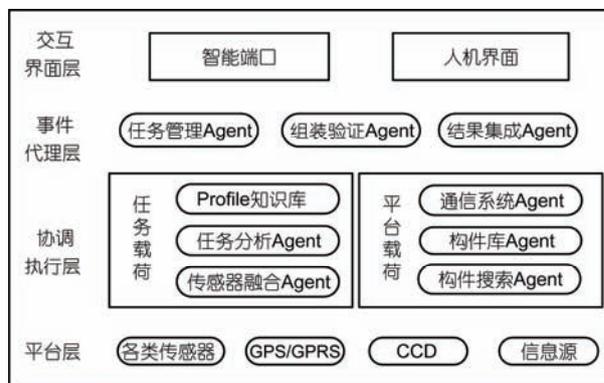


图3 DASS 模型框架

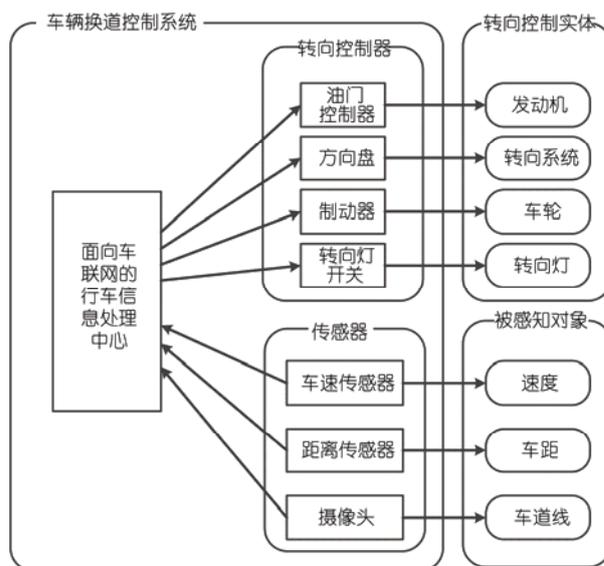


图4 车辆换道控制模型

路口、匝道, 走错车道. 在宏观交通流下 DASS 服务: 同时为多个车辆提供服务, 车辆行驶车道上前车发生故障停止行驶, 但由于视线遮挡等原因看不到时, 尽早提示驾驶员采取相应的措施.

通过车载传感器的实时检测及车联网通信, 对相关车辆(前车、目标车道的车辆、后车等)的车距、车速以及车道线信息进行打包, 将行车信息发送给行车信息处理中心, 通过数据挖掘预测出下一时刻可能的交通状态, 将换道指令发送给转向控制器, 由控制器做出一系列动作从而安全地完成整个换道的动作, 同时也形成了一个换道的交通信息流传递.

在这个车辆换道应用场景中, 由以下两大部分组成.

(1) 外部环境. 即整个 DASS 所处的现实物理环境. 将与 DASS 发生交互的物理世界中的所有对象称为环境实体, 每个环境实体都拥有其静态参数(标识、属性)和动态行为部分. 在车辆换道应用场景中涉及到的环境实体其他相关车辆、道路、障碍物等.

(2) 行车信息控制系统. 以行车过程中的交通信息流为基础, 具体由以下 3 部分组成: ① 由传感器和检测装置所构成的交通数据传感网络, 用于采集行车环境实体的各项信息; ② 车辆控制器及产生对应动作的控制实体等, 用于控制行车中的动态行为; ③ 基于车联网的各类后台信息处理器, 根据所采集到的环境信息, 各处理器之间协同工作, 按特定服务规则向设备控制器发出合适的指令信息.

上述对车辆换道应用场景的分析, 可以推广到其他基于车联网环境下 DASS 的应用场景中.

3 关键算法

3.1 基本概念

行车环境中将具有相同属性、动作的环境实体可以被抽象为类, 称为环境实体类, 其形式化定义如下.

定义 1: 一个环境实体类表示为 $T=\{\text{Tag}, \text{Cha}, \text{Act}\}$, 其中 Tag 为环境实体类的标识符; Cha 为环境实体属性的集合; Act 为环境实体允许的操作集合.

定义 2: 一个行车服务形式化表示为一个三元组 $RA=\{S, E, Q\}$, 其中 S 表示行车服务的标识符; E 表示一个环境实体类型的集合; Q 为一个时间自动机, 用

于描述服务的动态行为. 其中, 时间自动机是一种用于描述、分析实时系统行为的形式化模型. 它在有限状态自动机的基础之上扩充了取实数值的时钟变量, 用以刻画连续变化的时间^[14].

对于一个行车主动服务系统, 所关注的环境实体是各个车辆、道路、行人等, 由各种信息传感设备提供的基本服务称为设备服务, 设备服务分感知型服务、控制型服务和处理型服务.

(1) 感知型服务. 这类服务通过与行车环境的交互来采集信息, 感知外部环境的状态变化, 并将获取的行车信息传输给其他服务.

(2) 控制型服务. 这类服务接收从其他服务发送的对某个行车设备的控制信息, 然后将这些信息转换为行车设备可接受的控制命令, 并将其发送给对应的交通实体.

(3) 处理型服务. 这类服务并不直接与环境实体进行交互, 而是负责提供业务逻辑, 把从感知型服务中获取的信息进行综合处理, 然后根据特定规则将控制信息发送给控制型服务.

3.2 行车主动服务需求组建算法

对用户服务需求及时准确全面地获取, 是保证高质量行车服务的关键基础. 行车主动服务的实现是建立在个性化的信息感知的基础上, 用户意图发现与辨识是实现行车服务的关键, 保证服务以合适的方式、约定的服务质量和最小的代价向用户提供所需服务.

服务需求的建模方法已有很多研究并且有很多显著的成果. 例如: 文献[15]利用协议组合公理将服务需求映射成涵盖协议、角色、协议组合公理的服务实例的集合, 但是这是一种从全局来单向刻画需求的方式. 而 DASS 在车联网环境下需根据个体车辆的相关状态确定车辆的行车需求(即局部会影响服务需求), 因此这种方法并不适用于 DASS. 再如, 文献[16]提出一种基于环境的物联网三元问题域建模框架, 给出了物联网服务建模本体及相应的原则, 但该算法对具体的需求建模描述较少. 基于以上算法的问题, 考虑到环境实体类 T 在行车服务和需求之间的桥梁作用及规划图算法能使并发执行任务最大化的优点, 我们采用了相对稳定的规划图算法对行车环境实体与服务需求进行建模.

以规划图算法作为理论基础^[17], 用一个类规划

图结构来表示组合服务需求,使得系统能够动态地获取行车的服务需求,利用车联网技术将人-车-路-环境实时的情况进行反馈,在对数据整合及处理的基础上清晰地描述出服务需求模型,从而提供主动、实时、保真、个性化的行车主动服务构建方案.据此提出行车主动服务需求的规划图结构组算法.

定义3: 原子任务(atomic task, AT)表示满足一定的条件时,一组环境资源 T 的状态发生变化并产生一定的影响时,这个状态变迁的过程称为一个原子任务.原子任务是构成行车服务需求的基本单元, $AT = \{Prec, Tras, Effect\}$.

其中, $Prec$ 表示状态变迁过程中所必须的前提条件集合,每个前提条件都对应一个环境资源状态; $Tras$ 表示一组状态变迁的集合,若该集合包含的状态变迁可以划分为可选的小事务时,称为条件任务; $Effect$ 表示状态变迁带来的影响,该影响也作为资源的状态集.特别地,当一个原子任务满足:

- 1) $\forall tras \in AT.Tras$ 时, $tras.p_0 = tras.p_t$;
- 2) $\forall tras_1, tras_2 \in AT.Tras$ 时,
 $tras_1.p_0 \cap tras_2.p_0 = tras_1.p_t \cap tras_2.p_t = \emptyset$;
- 3) $AT.Prec = AT.Effect = \bigcup_{1 \leq i < k} tras.p_0$.

则该原子任务为一个保持任务.

定义4: 行车服务需求(driving service requirement, DSR)可用一个八元组来表示: $DSR = \{T, P_0, P_t, P_s, T2InitMappingfunc, T2GoalMappingfunc, M, C\}$.

其中, T 是行车环境实体类的资源集合,表示行车过程中所涉及到的所有环境信息集合; P_0 是给定 T 的初始状态集,是行车服务需求组建的初始条件; P_t 是目标状态集,表示行车服务需求期望的资源状态; P_s 是中间状态集,随着类规划图的扩展不断变化. $T2InitMappingfunc: T \rightarrow 2P_0$ 表示由环境资源 T 到初始状态集 P_0 的幂集的映射函数,用于由环境资源 T 给出初始状态集; $T2GoalMappingfunc: T \rightarrow 2P_t$ 是由环境资源 T 得到目标状态集 P_t 的映射函数,用于给出目标状态集; M 是原子任务 AT 的任务集合; C 是状态集与任务集之间、状态内部、任务内部的映射关系集合.

行车主动服务需求组建算法描述:

输入: 初始状态集 P_0 , 目标状态集 P_t 和任务集 M .

输出: 服务需求的类规划图结构 DSR .

Step 1: 将初始状态集 P_0 作为类规划图的第 0 层

命题,且令 $P_s = P_0$;

Step 2: 判定任务集 M 是否为空,若有 $M = \emptyset$,则结束扩展返回,否则继续;

Step 3: 扩建新的动作层,取 $M_1 \subseteq M$,对于所有 $m \subseteq M$,若满足条件 $m.Prec \subseteq P_s$,则有 $m \subseteq M_1$;

Step 4: 为动作层添加保持任务, $\forall t.p \in P_s - M_1.Prec$ (其中 $M_1.Prec = \bigcup m.Prec$),添加一个保持任务 m_t ,使 $m_t.Prec = m_t.Effect = m_t.tras.p_0 = m_t.tras.p_t = t.p$ 从而构成一个含有 $|P_s - M_1.Prec|$ (集合的势,表示集合元素的个数)个保持任务的集合 M_2 ;

Step 5: 为任务集 $M' = M_1 \cup M_2$ 中的各个任务添加前提条件边、效果边和互斥关联边,具体的添加过程可参考文献[18],且重置 $M = M - M_1$;

Step 6: 扩建新的命题层,重置中间状态集 $P_s = M'.Effect$,并将其中的各个状态作为新命题层中的命题元素,转 Step 2.

3.3 行车主动服务类型匹配算法

基于语法的匹配算法与基于语义的匹配算法相比,在查全率与查准率方面都逊色不少.本文在前人基于语义的匹配算法研究基础上^[19],考虑了行车主动服务概念之间的联系和概念所处的深度,通过构造的相似度函数来确定服务的类型,提高行车服务匹配的效率.行车主动服务进行匹配时,首先要确定服务的类型,过滤掉很多类型的服务,有利于提高匹配效率.类型匹配时,有相等、包含、无关 3 种关系.

定义5: 概念 c_1 与 c_2 的语义距离为 $Dis(c_1, c_2)$; 概念 c_1 是 c_2 的子集用 $Sub(c_1, c_2)$ 表示; c_1 具有 c_2 属性用 $Pro(c_1, c_2)$ 表示.

定义6: 关系 R 包含 Sub 和 Pro ; 两个节点的路径为 $Pat(c_1, c_2, \dots, c_n)$, c_n 为根节点.

定义7: 若 c_1 和 c_2 在路径表中显示相邻称为直接路径关系; 若 c_1 和 c_2 路径相连但不相邻称为间接路径关系.

服务类型匹配算法步骤如下:

Step 1: 设定权重.继承关系权重和属性关系权重分别为

$$\begin{aligned} W[Sub(c_1, c_2)] &= 1 + 1/f(\text{dep}(c_2)), \\ W[Pro(c_1, c_2)] &= m \times [1/f(\text{dep}(c_2))], \quad m > 1, \end{aligned} \quad (1)$$

其中, $\text{dep}(c)$ 为概念的深度, $\text{dep}(c_n) = 0$; $f(\text{dep}(c_2))$ 是 $\text{dep}(c)$ 的增函数, m 是为了控制概念属性关系的权重.

Step 2: 记录节点路径表. 记录每个节点到达根节点的路径.

Step 3: 计算语义距离. 计算 $\text{Dis}(c_1, c_2)$ 的步骤如下:

1) 判断 c_1 和 c_2 是否为同一概念, 若是 $\text{Dis}(c_1, c_2) = 0$, 否则转入步骤 2);

2) 判断 c_1 和 c_2 是否存在直接路径关系, 若是 $\text{Dis}(c_1, c_2) = \min\{W[R(c_1, c_2)], W[R(c_2, c_1)]\}$, 若否转入步骤 3);

3) 判断 c_1 和 c_2 是否存在间接路径关系, 若是 $\text{Dis}(c_1, c_2) = \sum\{W[R(c_1, c_m)] + \dots + W[R(c_m + i, c_2)]\}$, 当 c_1 和 c_2 间有多条相对路径, 则

$$\text{Dis}(c_1, c_2) = \min\{Dis(c_1, c_2), \dots, Dis(c_1, c_2)\},$$

若不存在相对路径, 则转入步骤 4);

4) 更新

$$\text{Dis}(c_1, c_2) = \min\{\text{Dis}(c_1, c_n)\} + \min\{\text{Dis}(c_2, c_n)\}.$$

Step 4: 计算概念间的相似度. 构造相似度函数:

$$SF = \frac{1}{\text{Dis}(c_1, c_2) + 1}. \quad (2)$$

自变量为 Dis 的减函数, 相似度函数满足 $\text{Dis}(c_1, c_2) = 0$ 时, $SF = 1$, 即需求的服务与匹配的服务关系为相等; $\text{Dis}(c_1, c_2) = \infty$ 时, $SF = 0$, 即需求的服务与匹配的服务无关; SF 的值在 $[0, 1]$ 之间时, 需求的服务与匹配的服务之间为包含关系, 且值越大, 关系越相近.

3.4 行车主动服务需求与服务能力匹配算法

用户需求与主动服务能力要达到匹配才能保证行车服务的顺利进行, 因此必须满足在应用领域相同的情况下, 用户需求必须在主动服务 Agent 所规划的服务需求范围内. 当用户需求领域与主动 Agent 的服务领域相同时, 从相同的初始状态经过一些中间状态集和输入输出集, 达到一个相同的目标状态, 则称为一个主动服务 Agent 的服务能力能够满足一个用户需求. 服务需求与能力匹配算法的研究多采用相似度函数、语义距离等方法^[19, 20], 本文采用更适合主动服务 Agent 的服务需求与能力匹配算法, 并作出

如下定义.

定义 8: 子功能模块的需求 $RS_i = \{S_{io}, S_{im}, S_{it}, Out_i\}$, 其中, S_{io}, S_{im}, S_{it} 分别表示 RS_i 的初始状态, 中间状态和目标状态, Out_i 表示 RS_i 的输出集.

定义 9: 主动服务 Agent 的服务能力 $SA_i = \{S_{io}, S_{im}, S_{it}, Out_i, Rel_i\}$, 其中, $S_{io}, S_{im}, S_{it}, Out_i$ 分别表示 SA_i 的初始状态, 中间状态, 目标状态和输出集. Rel_i 表示 SA_i 的依赖关系集.

用户需求与服务能力匹配算法描述如下. $RSMATCHSA(rs, sa)$:

InPut: $RS_i \in RS = \{RS_1, RS_2, \dots, RS_n\}$, $SA = \{SA_1, SA_2, \dots, SA_n\}$,

OutPut: TRUE(匹配), FALSE(不匹配).

Step 1: 将个子模块的需求 RS_i 在需求注册表 DF 中注册, 即 $RS_i \subseteq DF$.

Step 2: 设置不匹配判断条件, 定义一个变量来确定当前的 RS_i 是否匹配, If $(O_{RS} = O_{SA})$, IsMatch = FALSE.

Step 3: Complete(SA_i, Rel_i) 表示 SA_i 依赖的其他服务是否全实现, 并设置一个匹配判断条件: 对每个 SA 中的 SA_i , 若 Complete(SA_i, Rel_i) = 1, IsMatch = TRUE. 即: 如果一个服务能力能满足子模块的需求返回真 (Complete(SA_i, Rel_i) = 1), 代表 $RS_i.S_{io} = SA_i.S_{io}$ and $RS_i.S_{im} = SA_i.S_{im}$ and $RS_i.S_{it} = SA_i.S_{it}$ and $RS_i.Out_i = SA_i.Out_i$

Step 4: 返回 RS_i IsMatch//结束.

该算法描述了如何判断主动服务 Agent 的服务能力与子模块功能需求是否匹配, 只有主动服务 Agent 的服务能力中的所有条件都满足子模块的需求时, 算法返回 TRUE, 主动服务 Agent 的服务能力与子模块功能需求才匹配; 否则主动服务 Agent 的服务能力与子模块功能需求不匹配.

4 仿真实验

4.1 交通安全预警仿真

实验利用交通仿真软件 PreScan 搭建交通仿真场景, 与 Matlab/Simulink 进行联合仿真 (Simulink 中编写控制模块), 进而生成 PreScan 软件中的三维模型. 场景中两辆车在经过 Y 形交叉路口时发生了碰撞事故, 并对整条路的交通状况带来了影响, 以期望

DASS 发挥调控作用, 使在同一条路上行驶的其他 3 辆车不致产生二次交通事故或者交通延误.

临界安全车距是保证事故预警效果的重要安全指标. 交通安全预警采用安全距离法, 由于不考虑前车减速情况, 故采用传统的临界安全车距模型:

$$S_v = S_0 + V_0 t + \frac{V_0^2}{2j_{a2}}, \quad (3)$$

式中, S_0 为制动停车时两车的安全间距, V_0 为预警车辆制动的初速度, t 是制动系统响应时间, j_{a2} 为预警车辆制动减速度. 取 S_0 为 2 m, 即制动结束时预警车辆与前方事故车辆的距离为 2 m; V_0 取 20 m/s(72 km/h), 这个速度在城市道路工况下已达到一般城市道路的限制速度; 液压制动系统响应时间一般在 0.015~0.03 s 之间, 考虑制动力增长需要一定的时间, 制动系统响应时间 t 取 0.05 s; 从汽车应有的制动能力来说, 车辆制动减速度 j_{a2} 可分为 3 种: (1) 紧急制动, 车辆最大制动减速度在 5.5~8 m/s^2 , 为紧急避险而采取的极不舒适的制动行为; (2) 中级制动, 车辆的平均制动减速度在 2.5~5.5 m/s^2 , 是一种提前避险行为; (3) 普通制动, 车辆平均制动减速度在 0~2.5 m/s^2 , 在正常跟车过程中的减速行为. 这里 j_{a2} 取 3.5 m/s^2 (为避险而采取的中级制动强度).

在路侧及路中央有传感器及专用短程无线通信 (DSRC) 实时对行车数据进行采集和传输, 交通信息

中心对数据进行综合分析并依据不同的行驶环境提供相应的 DASS 服务. 为了更清楚地显示出各个车辆所在区域的控制策略, 用不同的颜色块将整个行驶场景分为 3 个区域. 图 5 为仿真实验的二维场景, 其中 Y 型路口为发生碰撞事故路口, DASS 对区域 1 及区域 2 内的车辆推送不同的指令; 图 6 交通信息中心发生交通事故时的触发模型, 碰撞检测模块检测到事故后, 将事故信号发送到交通信息中心, 经 DASS 处理后向目标车辆发送改变路径的服务信息; 图 7 和 8 为仿真实验中车辆转向和速度的控制模型, 实验

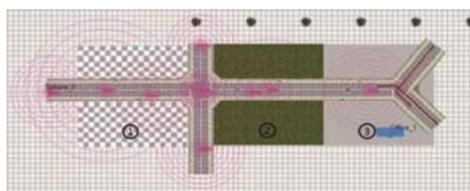


图 5 DASS 二维场景示意图

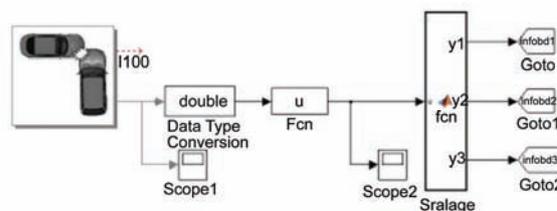


图 6 交通信息中心事故触发模型

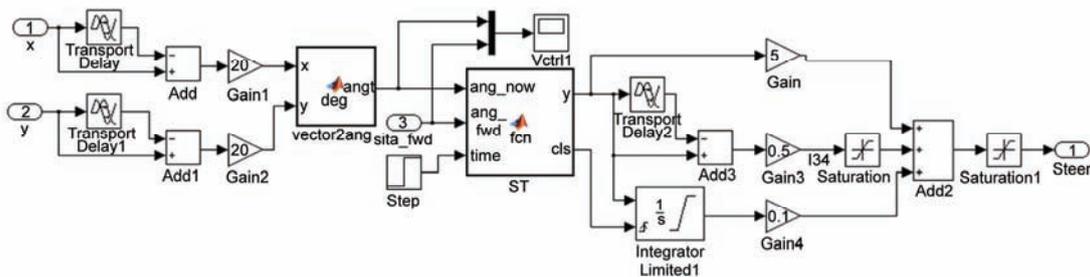


图 7 车辆转角控制模型

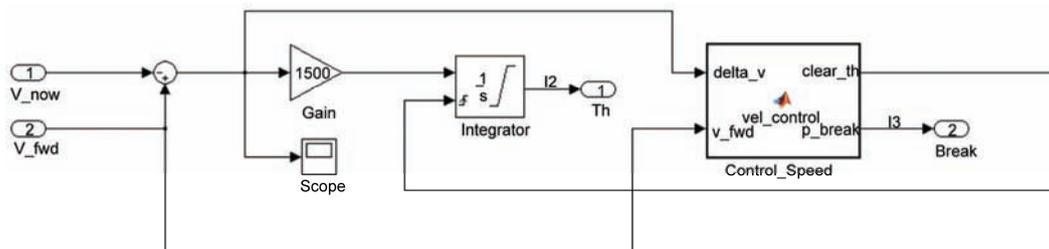


图 8 车辆速度控制模型

中仿真车辆接收到服务信息后直接决策改变路径.

DASS 对车辆的控制策略程序如图 9.

实验中, 在前方道路交叉口发生一起两车碰撞事故, 通过路边的专用短程无线通信(DSRC), 将传感器采集到的区域范围内车辆 ID、速度、位置坐标、碰撞状态等信息通过 DSRC 转发给信息中心处理. 交通信息中心的 DASS 实时接收并分析数据、调用控制策略, 将最优控制方案反馈给影响路段的车辆, 这些车辆对原来的决策控制信息做出适当的调整. DASS 根据传感设备提供的信息, 在感知到路口发生交通事故后, 计算临界安全车距为(以预警车辆以 20 m/s

速度行驶计算) $S_v = S_0 + V_0 t + \frac{V_0^2}{2j_{a2}} = 2 + 20 \times 0.05 + \frac{20^2}{2 \times 3.5} \approx 60 \text{ m}$. 安全车速为 $\min(15 \text{ m s}^{-1}, (\text{dist}-5)/3) \text{ m s}^{-1}$, 结合路侧传感设备提供的车辆位置信息, 主动对行驶车辆做出的推送服务如表 1 所示: 特别地, 表中距离事故地点小于 60 m 时还应大于 28 m(以紧急制动强度 $j_{a2}=8 \text{ m/s}^2$ 计算得到的临界安全距离, 以保证车辆的行驶安全).

实验表明, DASS 可以有效地将预警信息及时有效地传递到受交通事故影响路段的车辆. 从宏观上大大降低了交通拥堵, 对事故救援也大有益处; 从微

表 1 DASS 主动推送服务策略类别及评价

策略触发交通条件	主动推送策略	宏观交通流评价
驶近十字路口	减速, 调整路径并转向	降低拥堵, 缓解交通压力
驶过十字路口, 距离事故地点大于 60 m	降低车速, 以安全车速行驶	减少行车延误和冲突, 预知路况保障安全
驶过十字路口, 距离事故地点小于 60 m	以中级制动强度制动避险	提前预警, 防止交通事故

观上可以做到对每辆车的主动安全预警, 根据车辆不同的位置、速度信息, 提供最安全、合理、高效的建议. 图 10 和 11 为仿真实验前后在 DASS 服务区域内车辆的位置, 实验前可知各个车辆均采用了适当的行驶策略, 保证了最优的服务效果.

一方面, 在车联网环境下, 对于受交通事故影响的路段, 上述的行车策略都适用, 而不需要重新定义环境及行车控制策略, 从而为解决特定交通场景问题提供了新的思路和方法; 另一方面, 对行车环境中的多源信息进行融合判断, 通过交通信息中心分析得到对应的控制信息, 进而通过 DASS 实时推送, 保证区域内服务车辆的行车安全, 体现出车联网环境下信息传递和车辆控制的有效性, 在此基础上 DASS 还可以提供更高形式的主动服务, 如停车位预定、餐厅订位等人性化功能.

```
function strategy(vel,crash_car1,crash_car2,turn_cross)
    accident.x=(vel(crash_car1).x+vel(crash_car2).x)/2;
    accident.y=(vel(crash_car1).y+vel(crash_car2).y)/2;
    for i=1:3
        dist=sqrt((vel(i).x-accident.x)^2+(vel(i).y-accident.y)^2);
        if dist < 60
            control(i,0,0);
        else
            if vel(i).x <= (turn_cross+5)
                if (vel(i).x >= (turn_cross))
                    control(i,-200,90);
                else
                    control(i,5,-200);
                end
            else
                control(i,min(15,((dist-5)/3)), -200);
            end
        end
    end
end
```

图 9 DASS 对车辆的控制策略程序

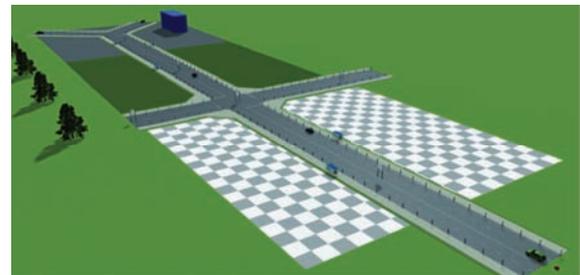


图 10 实验前的场景效果



图 11 DASS 控制的场景效果

4.2 实时动态路径规划仿真

在上一仿真实验的基础上, 利用车联网环境的信息综合能力和大范围车辆服务的优势, 针对实时动态的 DASS 路径规划问题用模拟退火蚁群算法进行了仿真实验. 实验假设在车辆行驶的可能路段的某一路口发生交通事故, 通过交通信息中心的决策, 将重新规划的路径行车服务向车辆主动推送, 使用户可以规避交通事故地点, 既不继续造成堵车, 又能按照最优路径迅速到达目的地.

实验选取某市区一段真实道路在 PreScan 中搭建如图 12 所示的微观道路交通场景. 该场景中一共有 20 个路口, 其中无叉路口有 5 个, 三岔路口共有 12 个, 十字路口共 3 个, 共计 29 段路, 每段路都是双向四车道. 路侧选用了 DSRC 作为车辆通信装置 (PreScan 中使用的 DSRC 模块支持发送 19 个双浮点型数据, 根据实际情况对各个信道做了分配, 使其在进行数据交互的时候互不干扰、规范统一, 便于数据处理), 智能车辆上安装有 DSRC 的双向通信装置及用于感知行车环境的传感器用于行车信息交互. 图 13 为对应的城市交通场景三维效果.



图 12 交通场景的平面效果图



图 13 交通场景的 3D 效果图

在利用算法求解路径选择问题时, 通常需要将模型搭建的路网用抽象的图直观表示出来, 用节点代表其中的交叉口, 线段代表其中的路段, 并且根据路网的特点表示出交通的流向, 将路网规划转化到图中进行求解, 更加清晰明了. 如图 14 所示, 本文根据搭建的交通场景模型描绘了所对应的有向拓扑结构图, 其中道路的连接都是双向的, 并对路网中的交叉口进行了编号.

建立微观交通场景后, 根据道路联通状态, 在 PreScan 的对应场景中提取出各个路口的节点坐标如表 2 所示.

设定实验车辆起始路口节点为 1, 目的路口节点为 14. 将仿真实验分为两组进行对比, 第 1 组是事故

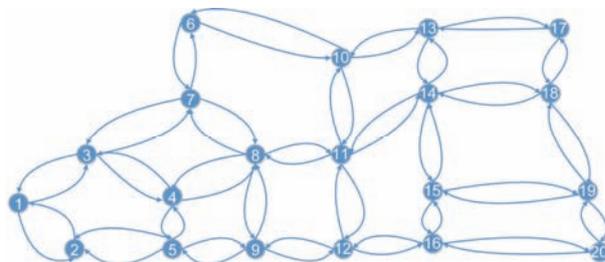


图 14 道路联通状态图

表 2 道路交叉口坐标

路口节点	X 坐标	Y 坐标
V ₁	-275.7	182.2
V ₂	-246.524	110.19
V ₃	-214.53	206.99
V ₄	-171.445	156.99
V ₅	-182.3	101.75
V ₆	-74.2	322.3
V ₇	-90.23	253.5
V ₈	-65.22	192.42
V ₉	-81.5	78.15
V ₁₀	76.96	258.95
V ₁₁	32	151.67
V ₁₂	-9.54	52.53
V ₁₃	157.6	248.9
V ₁₄	154.67	182.15
V ₁₅	151.74	116.22
V ₁₆	148.41	44.26
V ₁₇	276.96	242.78
V ₁₈	274.02	176.84
V ₁₉	299.19	70.69
V ₂₀	306.6	38.9

发生前分别利用传统蚁群算法和模拟退火蚁群算法对最优路径进行规划; 第 2 组是模拟事故发生后, 分别利用传统行车路径选择与 DASS 服务路径优化优化方式作了对比。

通过第 1 组的仿真实验得到两种方法最优路径均为: 1→3→4→8→11→14, 对应的最优路径长度为 475.80 m, 而两者的差异体现在求解的速率上: 对两个算法运行时间的平均值进行了统计, 结果表明本文的模拟退火蚁群算法比传统算法平均快 21.78%, 保证了运行的速率远高于传统算法. 由于实验中的路网相对简单和理想, 因此将该算法应用到实际的复杂路况时, 本文提出的采用模拟退火蚁群算法的 DASS 优势将会更加明显. 第 2 组实验中, 利用 DASS 对路径优化的过程如下: 假设当 14 号路口发生交通事故的时候, 车辆事故信号检测器将事故信息发送给行车服务中心, 系统自动辨识事故情况发生的位置, 通过路旁的 DSRC 将所在位置的编号发送给 DASS 处理, 并且对路径中含有第 14 号路口的车辆的路径利用模拟退火蚁群算法重新规划, 主动提示原路径异常并推送最优路径. 与此同时将禁止该路口的所有的车辆的进入, 而离开该路口的车辆则不进行限制. DASS 服务路径优化是通过行车服务中心实时的解算, 将最优路径信息通过车联网推送给车载终端设备, 此时重新规划拓扑结构后则形成如下的联通状态如图 15 所示。

将 PreScan 对应路口的地理坐标提取出来, 映射到算法的数据集中, 图 16 为车辆在突发情况前后规划出的路径轨迹. 图 17 为路径轨迹在仿真交通场景中的示意图, 其中红线代表期望最优路径, 黄线代表实时调整的期望路径. 图 18 为普通行车路径轨迹图。

作为对比实验, 传统行车路径选择在未得知路口出现交通事故时, 车辆行驶到事故路口, 由于后续车辆的不断涌入极易造成交通拥堵, 如果此时重新

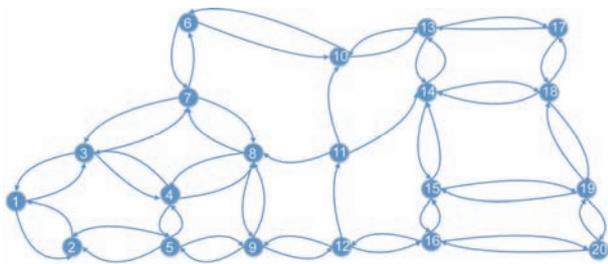


图 15 重新规划路网的联通状态

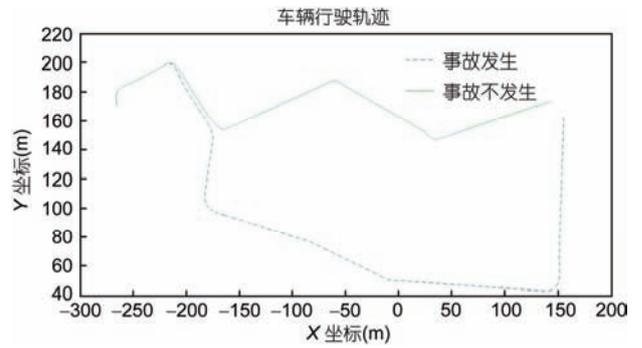


图 16 车辆行驶轨迹对比

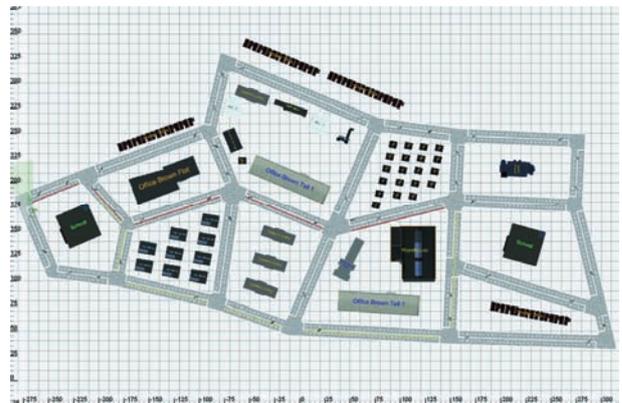


图 17 路径轨迹在仿真交通场景中的示意图



图 18 普通行车路径轨迹

规划行驶路径, 则得到图 18 轨迹示意图(红线标注), 与 DASS 服务对比, 既加剧了交通拥堵, 又耽误了自己的行程时间, 体现出 DASS 极大的优越性. 通过第 2 组的两个实验结果的仿真对比, 得到两次实验的最优路径及其长度如表 3 所示。

实验分别设计了在不同路段接收到路口事故信息, 进而及时调整路径. 由表 3 可以看出, DASS 在路径规划上面与传统的路径规划相比, 在时空上体现出了巨大的优势, 具体表现在: 接收到路口交通事故

表3 路径优化实验结果对比

路径规划类型	信息接收节点	规划路径	距离(m)	时间(s)	相对节省时间比率
TTSS 路径规划	无	1→3→4→8→11→8→7→6→10→13→14	903.43	82.13	0
	即将行驶到节点 1	1→3→7→6→10→13→14	581.33	52.85	0.36
	节点 1-3 之间	1→3→7→6→10→13→14	581.33	52.85	0.36
DASS 路径规划	节点 3-4 之间	1→3→4→5→9→12→16→15→14	664.41	60.40	0.26
	节点 4-8 之间	1→3→4→8→7→6→10→13→14	692.60	62.96	0.23
	节点 8-11 之间	1→3→4→8→11→8→7→6→10→13→14	903.43	82.13	0

的信息越早, 路径规划的距离和时间就越短, 特别地, 当进入到事故路段后才感知到有事故时, 只能通过上述算法的分类, 优先保证车辆和驾驶员安全, 在进行路径规划. 也正因如此, 此时 DASS 的路径规划与 TTSS 的路径规划是相同的路径. DASS 系统可以为驾驶员提供高效、及时的路径规划, 在没有进入到事故路段时, 总能提供比 TTSS 路程、时间更短的路径, 节省 23% 以上的时间(以平均速度 11 m/s 计算), 避免了进入事故路段.

无论传统的行车规划采用的算法如何高效, 在没有感知到交通信息变化的条件下, 路径规划的节点极可能出现重复, 加剧事故路段的交通拥堵情况, 影响自身的行车安排. 而 DASS 的路径规划的优越性就在于信息的自动获取, 能及时改变原有的规划策略, 从而避免车辆行驶到进退两难的事事故路段.

综上所述, DASS 在综合车联网的交通环境信息、感知到事故后即时做出响应, 向车辆推送更改的行车路线, 在规避交通事故的前提下以最短路径到达终点. 并且, 利用模拟蚁群算法可以实时的规划最

优路径, 将主动服务的思想加入到路径规划的行车当中, 以保证在行车途中发生意外事故时, DASS 能及时推送信息, 使车主能够及时重新规划行车路线, 体现出 DASS 的高效性、主动性和实时性.

5 结论

(1) 建立了包含行车信息流模型、车辆 Agent 模型、基于 MAS 分层控制的抽象的具有推理功能和信息处理能力的 DASS 模型, 自动获取交通信息并分析处理, 主动为用户提供个性化的服务.

(2) 研究了基于规划图的行车服务需求模型、行车服务类型匹配算法及用户需求与服务能力匹配算法, 为主动服务 Agent 提供技术支持.

(3) 建立了车辆转向与速度控制模型, 在特定交通场景下(交通预警与实时路径规划)验证了 DASS 系统的安全性、个性化、主动性和实时性.

(4) 对于 DASS 功能的具体实现技术有待进一步具体的研究, DASS 功能有待进一步的开拓.

参考文献

- 1 杨兆升. 基于动态信息的智能导航与位置服务系统关键技术及其应用. 北京: 中国铁道出版社, 2012. 5
- 2 王笑京. 新一代智能交通与物联网. 物联网技术, 2012, 2: 15-17
- 3 李海舰. 道路交通信息获取多参量感知与传感器网络优化. 博士学位论文. 北京: 北京交通大学, 2014
- 4 Gebremeskel G B, Chai Y, Yang Z. The paradigm of big data for augmenting internet of vehicle into the intelligent cloud computing systems. In: Internet of Vehicles—Technologies and Services. Berlin: Springer International Publishing, 2014. 247-261
- 5 Liu T, Wang Y, Wenjuan E, et al. Vehicle collision warning system and algorithm at intersection under internet-connected vehicles environment. In: The Twelfth COTA International Conference of Transportation Professionals. Beijing: CICTP, 2012. 1177-1185
- 6 Cheng C, Wu Z. Design of a system for safe driving based on the internet of vehicles and the fusion of multi-aspects information. In: International Conference on Computational Intelligence & Security. IEEE, 2013. 692-696
- 7 Birrell S A, Fowkes M, Jennings P A. Effect of using an in-vehicle smart driving aid on real-world driver performance. IEEE Trans Intell Transp Syst, 2014, 15: 1801-1810
- 8 Sakthi U, Kumar S S. Vehicular cloud service model for secure transportation system. Int J Appl Eng Res, 2015, 10: 19659-19666
- 9 张尧学, 方存好. 主动服务——概念、结构与实现. 北京: 科学出版社, 2005
- 10 Wei L, Fang C H, Zhang Y X. Semantic description model for active services. Comput Eng Appl, 2015, 41: 1-4

- 11 Wang H, Yu L. Service mining model in the direction of active service. *J Wuhan Univ Technol*, 2014, 32: 284–288
- 12 Tian G H, Meng D, Zhan J F. Reliable resource provision policy for cloud computing. *Chin J Comput*, 2014, 33: 1859–1872
- 13 莫同, 褚伟杰, 李伟平, 等. 基于情境感知事件的主动服务发现方法. *软件学报*, 2011, 22: 41–51
- 14 李力行, 金芝. 基于时间自动机的物联网服务建模和验证. *计算机学报*, 2011, 34: 1365–1377
- 15 李扬, 怀进鹏, 郭慧鹏, 等. 一个基于服务层叠网的分层服务组合框架. *软件学报*, 2007, 18: 2967–2979
- 16 李戈, 魏强, 李力行, 等. 物联网服务建模: 一种基于环境建模的方法. *中国科学: 信息科学*, 2013, 43: 1198–1218
- 17 叶荣华, 冯伟. 基于规划图的主动服务组合需求模型. *计算机科学与探索*, 2013, 7: 169–179
- 18 Blum A L, Furst M L. Fast planning through planning graph analysis. *Artif Intell*, 1997, 90: 281–300
- 19 李文清, 孙新, 张常有, 等. 一种本体概念的语义相似度计算方法. *自动化学报*, 2012, 38: 229–235
- 20 彭京, 杨冬青, 唐世渭, 等. 基于概念相似度的文本相似计算. *中国科学: 信息科学*, 2009, 39: 534–544

The driving active service system model based on MAS for internet of vehicle environment

LIANG Jun^{1,2}, ZHAO ZhenChao¹, CHEN Long¹, XU YongLong¹ & ZHU Ning²

¹Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China;

²Department of Mechanical, Shizuoka Institute of Science and Technology, Fukuroi, Shizuoka 437-0032, Japan

According to the lack of personalized, humanized and real-time problems of inadequate “request-response” current passive traffic services, the driving active service system (DASS) model of the hierarchical control “automatic recognition-active push” based on MAS (Multi-Agent system) is proposed. On the basis of modeling the driving environment entity and service, proposed the traffic service demand model based on planning graph matching problem, and researched on the algorithm of the key issues about service capabilities to match the users’ needs. Through the independent construction of the driver in the loop PreScan-Logitech G27 semi physical joint simulation platform to collect data, using the traffic safety warning and real-time dynamic path planning experiment verified the real-time, active and efficient DASS. Research on DASS is conducive to reduce traffic congestion and improve traffic efficiency. At the same time, DASS can strengthen vehicle safety and improve the driving experience.

driving active service, MAS hierarchical control, internet of vehicle

doi: 10.1360/N092016-00066