

面向产业链协同 SaaS 平台的多源异构信息动态集成技术研究

吕 瑞 王淑营 孙林夫 潘 华
(西南交通大学 CAD 工程中心 成都 610031)

摘 要 为降低协作企业内部系统与产业链协同 SaaS 平台交换大量数据所消耗的时间和空间代价,提高实时数据查询的准确性,对多源异构信息集成服务技术进行了研究。设计了面向产业链协同 SaaS 平台的多源异构信息集成模型;构建了多源异构信息动态发布和动态发现机制;提出了以 SaaS 平台为核心的多源异构信息转换与注册算法以及事件驱动的数据源动态调用算法;实现了面向产业链协同 SaaS 平台的多源异构信息实时准确查询服务。该模型和算法在汽车零部件产业链协同平台服务商配件库存管理中的应用证明了该技术的可行性和有效性。

关键词 产业链,软件即服务,协同平台,多源异构信息,动态集成

中图分类号 TP393.0 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.2.004

Research on Multi-source Heterogeneous Information Dynamic Integration Technology for Industrial-chain Coordination SaaS Platform

LV Rui WANG Shu-ying SUN Lin-fu PAN Hua
(CAD Engineering Center, Southwest Jiaotong University, Chengdu 610031, China)

Abstract In order to reduce the costs of time and space for exchanging large amounts of data for cooperative enterprise's internal systems and industry chain collaboration SaaS platform, and improve the accuracy of real-time data query, the multi-source heterogeneous information integration service technology was researched. Multi-source heterogeneous information integration model for industrial chain collaboration SaaS platform was established. Multi-source heterogeneous information conversion and registration algorithms were established. User's identity and business driven dynamic invocation algorithm was also proposed. Multi-source heterogeneous real-time accurate information inquiry service of industry chain collaboration SaaS platform was realized. The applications of the model and algorithms in service provider's parts inventory management of automotive parts industry chain collaboration platform prove the technology is feasible and effective.

Keywords Industrial-chain, SaaS, Coordinate platform, Multi-source heterogeneous information, Dynamic integration

1 引言

随着网络和信息技术的不断发展,跨企业的业务集成和协同^[1]已成为企业信息化发展的主旋律,得到了国内外学者的广泛关注,特别是随着云服务和大数据时代的到来^[2],跨企业的实时数据集成已成为以 SaaS 为代表的新一代 Web 应用所面临的首要问题。

关于企业间的数据集成方法通常有两类,一类是对数据进行物理上的整合,常用的方法主要有数据复制(DBR)及数据抽取转换技术(ETL)。数据复制方法是各个数据源的数据复制到与其相关的其它数据源上,并维护数据源整体的数据一致性。例如文献[3]通过在数据源安装数据交换客户端适配器的方式进行数据复制与交换,实现由用户身份和交换

业务关键字触发的数据集成方式。当数据量不断膨胀或数据源不断增加时,这种数据复制方式将增加平台的数据存储压力,同时安全性难以保障。数据复制适用于数据源相对稳定的情况,当数据分布性较广、网络延迟较大而系统对数据访问的实时性要求又较高时,数据复制的集成方法就不能满足企业的需求。数据抽取转换技术通过编制专门的数据抽取转换程序实现,这种方法简单成熟。例如文献[4]提出基于 ETL 技术的异构数据集成优化方案,但其需要在各个数据源编制接口软件,操作不便。另外一类数据集成技术是采用联邦数据库、中间件或 Web Service 的模式集成方法。联邦数据库由半自制数据库系统构成,通过数据源之间相互提供的访问接口分享数据。文献[5]对面向中小企业的网络化制造协同管理平台和系统集成体系结构进行了研究,基于联邦模式解

到稿日期:2014-12-16 返修日期:2015-05-28 本文受国家科技支撑计划课题(2013AA040606),四川省科技计划项目(2015GZ007)资助。

吕 瑞(1988—),女,博士生,主要研究方向为产业链协同平台技术、商务智能;王淑营(1974—),女,博士,研究员,主要研究方向为网络化制造、电子商务公共服务平台;孙林夫(1964—),男,教授,博士生导师,主要研究方向为网络化制造、电子商务、商务智能;潘 华(1982—),男,博士生,主要研究方向为网络化制造、电子商务公共服务平台。

决企业间协同规划、协同伙伴选择、协同信息交换等。但是基于联邦数据库的数据集成方法一般无法处理非结构化的数据源,而且在集成时需要为每个数据源单独编写大量的通讯接口。基于 DCOM/COM/CORBA 等技术的中间件集成方案通过在数据层与应用层提供统一逻辑视图的方式来隐藏底层的数据细节,使得用户可以把集成数据源看为一个统一的整体。例如文献[6]使用 D2RQ 引擎实现了遗留交通数据向本体形式数据片段的映射功能,通过本体中间件将多源本体数据片段集成为一个统一的数据语义视图。这类基于中间件的集成解决方案一般都使用专有协议进行通讯,系统较为复杂且成本高。基于 Web 服务的数据集成方案,开发标准一致,并且可以满足高实时性的要求。文献[7]设计了一种企业服务总线平台下的数据集成模型,该模型采用 WSDL 和 XML 描述,能够结合 ESB 系统的集成场景进行数据集成。用于应用集成的服务由第三方负责统一封装,当动态增加异构数据源时,需要开发人员对新增数据源进行处理,不能很好地支持数据源的动态变化。文献[8]及文献[9]提出了分布式环境下基于 SOA 技术的跨系统信息集成方案,其仅支持关系数据库数据集成,那些未保存在企业数据库中、需要与贸易合作伙伴共享的非结构化数据往往被忽略。

针对以上问题,本文围绕多源异构信息动态集成技术展开研究。针对产业链跨企业协同的信息化需求,提出了异构数据源动态发布与动态发现机制,它支持在数据源动态变化环境下的异构数据加工、整合;设计了与用户角色及权限相关、由事件驱动的数据动态集成解决方案。

2 产业链协同 SaaS 平台多源异构信息动态集成需求

2.1 基于产业链协同 SaaS 平台的产业链协同与资源共享解决方案

产业链协同 SaaS 平台是一个支持产业链上下游企业以动态联盟方式进行业务协同的第三方服务平台^[10],支持各个联盟围绕产品原材料及零配件采购、销售、售后服务等进行业务协作^[11,12]。平台以一套软硬件支持多联盟多企业群的业务协同^[13],将各联盟协作企业业务信息进行整合,通过产业链整体协同进行资源共享。

以面向车辆全国联保的维修服务配件库存优化为例,配件库存优化模型需要通过集成车辆信息、车辆配置 BOM 信息、客户信息、客户所属片区信息、车辆零部件历史更换信息,建立支持快速响应的产业链分布式配件库存安全模型,以降低产业链配件库存成本。分布式配件库存的分布性主要体现在整车制造厂配件库存、各片区中转库库存、各服务商本地库存,是一个树状的三级分布式库存结构。在产业链分布式配件库存安全模型的基础上,为提升服务的响应速度,需要建立基于平台的分布式配件库存协同调拨机制(见图 1),盟主企业根据来自各片区中转库及其下游服务商的车辆零部件历史更换信息、车辆配置 BOM 信息等,掌握整个产业链的实时库存状况,在产业链上各节点保持一定缓冲库存,并逐级或直接进行库存补给。当市场需求波动较大时,有可能存在库存补货不及时而出现急缺件的情况,这就需要盟主企业或片区管理员查看临近的中转库、服务商的实时库存,选择周转时间最短的调拨策略。

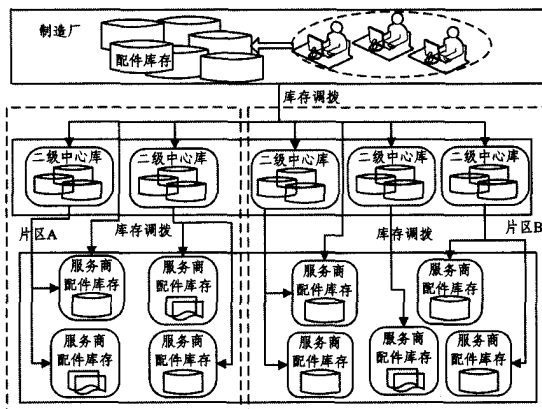


图 1 汽车产业链配件协同库存调拨关系

2.2 产业链配件库存协同对多源异构信息动态集成的需求

分布式配件库存整体协同涉及到的协作企业众多、数据量大且变化频繁,并且存在数据异构的问题,具体体现在以下方面。

(1)数据源多、数据量大:参与产业链协作的企业众多,每个协作企业的业务数据是一个独立的数据源。例如:面向车辆全国联保的维修服务体系涉及的服务商可达上千家之多,每种型号汽车的配件数以万计。

(2)数据异构:各服务商及中转库的业务数据由于其应用需要而选择了不同的数据管理方式,如 Mysql、Sybase、SQL Server、Oracle 等 DBMS 方式,也可能只是采用 Excel 来存储业务数据;并且可能存在命名冲突、数据类型差异等问题^[14],这种差异性也给数据集成带来了屏障。

(3)数据变化频率高:随着协作业务范围的变化,数据源会不断地增加或减少;同时,伴随着三级结构的分布式库存上下级节点及同级节点间频繁的库存调拨,各个库存的数据会经常发生变化。

(4)实时性高:分布式配件库存数据是否准确将影响配件协同调拨的及时性,因此对数据实时性要求较高。

针对上述配件库存数据源的特点及企业对分散于各地的异构数据源动态访问的需求,对面向产业链协同 SaaS 平台的多源异构信息动态集成技术进行研究。

3 SaaS 平台的多源异构信息模型

为实现面向 SaaS 平台多源异构信息集成服务,根据上述分析的服务集成需求,建立如图 2 所示的 SaaS 平台协作关系管理及多源异构信息集成模型。

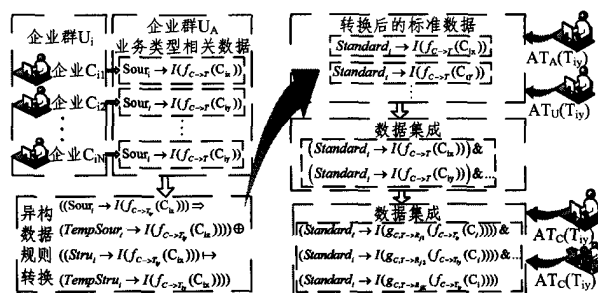


图 2 多源异构信息集成模型

3.1 SaaS 平台协作关系管理模型

为了便于对模型的描述,先对 SaaS 平台协作关系涉及的对象进行定义。

定义 1 定义 $U_i = \{U_{id}, U_{na}, C_i, T_i, R_i(T_i)\}$ 表示产业链某联盟协作企业群的协作关系模型。

- U_{id} 代表联盟 U_i 的联盟标识符;
- U_{na} 代表联盟 U_i 的联盟名称;
- C_i 代表某产业链节点的企业集, $C_i = \{C_{i1}, C_{i2}, C_{i3}, \dots, C_{ix}, \dots, C_{iN}\}$ 为一有限集, 其中 $\forall C_{ix} \in C_i, N$ 为协作企业的数量, $C_{ix} \in C_i$ 代表该产业链的盟主企业;
- T_i 代表协作企业类型空间, $T_i = \{T_{is}, T_{id}, T_{if}, \dots\} = \{\text{制造商, 中转库, 服务商, } \dots\}, \forall T_{iy} \in T_i$;
- $R_j(T_i) = \{R_{j1}(T_{iy}), R_{j2}(T_{iy}), R_{j3}(T_{iy}), \dots, R_{jk}(T_{iy}), \dots, R_{jK}(T_{iy})\}$ 代表联盟内协作类别 T_{iy} 的协作企业片区划分集, 其中 $\forall R_{jk}(T_{iy}) \in R_j(T_i), K$ 为协作片区数量。

定义 2 $AT(T_{iy})$ 代表协作类别 T_{iy} 的企业用户分类集, $AT = \{AT_A(T_{iy}), AT_U(T_{iy}), AT_C(T_{iy}), AT_T(T_{iy})\}$, 其中 $AT_A(T_{iy}), AT_U(T_{iy}), AT_C(T_{iy}), AT_T(T_{iy})$ 分别表示片区管理员、盟主企业管理员、服务商、中转库管理员。

定义 3 定义 $Ag_i(T_{iy})$ 代表联盟 U_i 协作类别为 T_{iy} 的用户集合, $Ag_i(T_{iy}) = \{Ag_{i1}(T_{iy}), Ag_{i2}(T_{iy}), Ag_{i3}(T_{iy}), \dots, Ag_{iM}(T_{iy})\}$, 其中 $Ag_{iM}(T_{iy}) = \{Ag_{iMN}(T_{iy}), Ag_{iMid}(T_{iy}), Ag_{iMC}(T_{iy})\}$, $Ag_{iMN}(T_{iy}), Ag_{iMid}(T_{iy}), Ag_{iMC}(T_{iy})$ 分别表示用户名、用户标识、用户所属企业名称。

规则 1: 企业与产业链协作企业建立协作关系的过程可以看成是协作企业集 C 中的元素向协作类型空间 T 和协作片区 R 的映射, 用 $f_{C \rightarrow T_{iy}}(C_{ix})$ 和 $g_{C, T \rightarrow R}(f_{C \rightarrow T_{iy}}(C_{ix}))$ 来表示任一企业 C_{ix} 与协作企业建立一次协作关系的过程。

规则 2: 若用 $I(C_{ix})_{RL}$ 表示 SaaS 平台对某联盟节点企业 C_{ix} 的信息管理模型, 则有:

$$I(C_{ix})_{RL} = I(C_{ix}) \cup I(f_{C \rightarrow T_{iy}}(C_{ix})) \cup I(g_{C, T \rightarrow R}(f_{C \rightarrow T_{iy}}(C_{ix})))$$

- $I(C_{ix})$ 代表企业 C_{ix} 的基本属性信息, 包括: 企业 ID、企业名称、企业简介、所属省市等;
- $I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 表示企业 C_{ix} 相关的协作类型信息, 包括协作类型的基本信息: 协作类型 ID、协作类型名称等;
- $I(g_{C, T \rightarrow R}(f_{C \rightarrow T_{iy}}(C_{ix})))$ 表示企业 C_{ix} 相关的协作片区信息, 包括: 片区名称、协作类别 ID 等。

规则 3: 用 $S_{AT \rightarrow Ag}(Ag_{iM}(T_{iy}))$ 表示用户 $Ag_{iM}(T_{iy})$ 的用户类型属性, 若 $Ag_{iM}(T_{iy}) \in C_{ix}$, 用 $I(Ag_{iXM}(T_{iy}))_C$ 表示协作企业 C_{ix} 的任一用户 $Ag_{iM}(T_{iy})$ 的信息模型, 则有:

$$I(Ag_{iXM}(T_{iy}))_C = I(Ag_{iM}(T_{iy})) \cup I(S_{AT \rightarrow Ag}(Ag_{iM}(T_{iy})))$$

3.2 SaaS 平台多源异构信息动态集成模型

各协作企业用户的数据管理方式存在差异性。为消除这种差异性, 需要对异构数据源进行数据结构标准化映射并完成数据类型转换。之后, 按照平台用户类型的不同, 动态选择不同的数据源进行数据装载以及呈现。多源异构信息动态集成过程可以用如下的形式化语言描述。

定义 4 定义 $D_i(T_{iy}) = \{DT_i(T_{iy}), Stru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})), Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})), TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})), TempSour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})), Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))\}$ 为协作企业群 U_i 业务类型为 T_{iy} 的多源异构信息数据模型。其中:

- $DT_i(T_{iy})$ 表示业务数据类型标识;
- $Stru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 表示协作企业业务类型相关的数据结构;
- $Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 表示协作企业业务类型相关的数据内容;
- $TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 表示业务类型为 $I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 的数据在平台的标准数据结构信息;
- $TempSour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 表示业务类型为 $I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 的数据在平台的标准数据类型信息;
- $Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 表示转换后业务类型为 $I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 的标准数据。

提取异构数据源数据结构、数据内容, 把按照业务类型 T_{iy} 的标准数据结构 $TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 进行映射的过程记为 \mapsto , 将参照平台标准数据类型 $TempSour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 进行转换的过程记为 \Rightarrow , 把数据源按照转换后的数据结构、数据类型进行装载的过程记为 \oplus , 则由定义 4 可将多源异构信息的转换过程描述如下:

$$Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})) = ((Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))) \Rightarrow (TempSour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})))) \oplus ((Stru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))) \mapsto (TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))))$$

用户对库存信息的实时查询驱动平台对服务商、中转库数据的动态集成, 因此结合定义 3、规则 3, 有以下定义。

定义 5 一次查询事件可以用一个四元组 $INQ_i = \{INQ_{id}, INQ_{time}, Ag_{iMid}(T_{iy}), S_{AT \rightarrow Ag}(Ag_{iM}(T_{iy}))\}$ 表示, 其中:

- INQ_{id} 表示查询事件的标识;
- INQ_{time} 表示查询事件的属性;
- $Ag_{iMid}(T_{iy})$ 表示用户标识;
- $S_{AT \rightarrow Ag}(Ag_{iM}(T_{iy}))$ 表示用户类型。

不同的用户类型在平台的操作权限不同。盟主企业管理员、片区主管可按片区对各服务商、中转库的配件库存信息进行集成查询。服务商、中转库管理员限定查询范围为本服务站、本中转库库存信息, 即单点查询, 因此有以下数学描述。

(1) 对于某一协作企业 C_{ix} 业务类型为 T_{iy} 的用户 $Ag_i(T_{iy})$, 若 $S_{AT \rightarrow Ag}(Ag_{iM}(T_{iy})) = AT_C(T_{iy})$ 或 $S_{AT \rightarrow Ag}(Ag_{iM}(T_{iy})) = AT_T(T_{iy})$, 则通过平台对本企业业务类型 $I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 的异构数据的查询为单点查询, 查询到的业务数据即为平台转换后的标准数据:

$$Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$$

(2) 对于盟主企业用户 $S_{AT \rightarrow Ag}(Ag_{iM}) = AT_U$ 以及片区用户 $S_{AT \rightarrow Ag}(Ag_{iM}(T_{iy})) = AT_A(T_{iy})$, 对联盟内各协作企业业务数据的查询, 需要按照片区对各企业数据进行集成。如果用 $\&$ 表示企业 C_{ip} 与企业 C_{iq} 业务数据的集成, 则某一片区 R_{jk} 的业务数据可表示为:

$$Standard_i \rightarrow I(g_{C, T \rightarrow R_{jk}}(f_{C \rightarrow T_{iy}}(C_i))) = (Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ip}))) \& (Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{iq}))) \& \dots$$

该联盟内所有企业的业务数据可表示为:

$$(Standard_i \rightarrow I(g_{C, T \rightarrow R_{j1}}(f_{C \rightarrow T_{iy}}(C_i)))) \& (Standard_i \rightarrow I(g_{C, T \rightarrow R_{j2}}(f_{C \rightarrow T_{iy}}(C_i)))) \& \dots (Standard_i \rightarrow I(g_{C, T \rightarrow R_{jk}}(f_{C \rightarrow T_{iy}}(C_i))))$$

定义6 用四元组 $WS_i = (WSID_i, WSN_i, WST_i, ASSP(I(fc \rightarrow T_{iy}(C_{ix})))$ 表示企业 C_{ix} 与业务类型相关的 Web Services 服务信息。其中:

- $WSID_i$ 表示服务 ID;
- WSN_i 表示服务名称;
- WST_i 是对 Web Services 服务的描述信息;

• $ASSP(I(fc \rightarrow T_{iy}(C_{ix})))$ 表示协作企业 C_{ix} 协作类型为 $I(fc \rightarrow T_{iy}(C_{ix}))$ 的 Web Services 服务地址。

4 多源异构信息动态发布及发现机制

多源异构信息集成技术分为信息的动态发布与发现两个过程,其框架如图3所示。

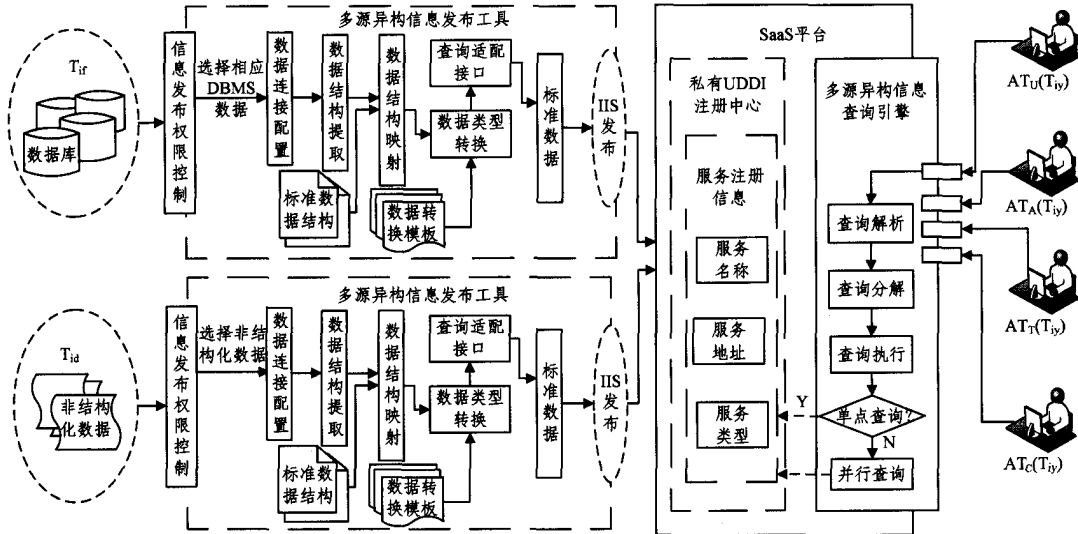


图3 多源异构信息集成框架

多源异构信息的动态发布通过将异构数据结构映射功能、数据类型转换功能、查询适配接口、数据查询服务、动态注册功能封装成多源异构信息发布工具(Multi-source Heterogeneous Information Dynamic Release Tool, MHIDRT)来实现。企业执行MHIDRT来完成异构数据的动态发布。

多源异构信息的动态发现通过查询引擎(Query Engine, QE)实现。由事件驱动QE对数据查询命令的解析及分解,通过检索平台私有的UDDI注册中心来获得企业通过MHIDRT发布的查询服务入口,并根据企业类型及用户角色对返回的数据进行动态重组。

面对复杂的云环境,多源异构信息动态发布与发现机制不依赖集中控制来完成异构信息的归一化操作以及信息的注册与发布。对分散的资源进行管理,使其形成一个统一的资源池,并支持动态可扩展。对信息的发现机制,能适应资源的动态变化。在面向SaaS云环境的应用下,解析企业群用户权限自动完成信息的重组与分配。

4.1 多源异构信息动态发布

XML是一种具备良好的自描述性和平台独立性的可扩展标记语言^[15],允许用户对自己的标记语言进行定义,并且数据存储格式不受显示格式的制约,因此采用XML作为中间件实现数据透明访问。而WSDL、UDDI等协议都基于XML,因此,协作企业通过MHIDRT将业务信息封装成Web服务并发布后,联盟内的其他企业就可以通过私有UDDI注册中心对服务进行实时查询、调用而不用考虑系统数据管理方式的差异性,从而实现了跨平台调用。所有的异构数据映射及发布工作全部封装在客户端MHIDRT内,平台管理员不需要对协作企业的业务数据进行分别转换及发布,方便对不断变化的协作企业进行业务管理,构建动态业务单元。

提取平台的标准数据结构文档,在MHIDRT中抽取出一个公共的接口,创建各类数据源的连接模块。根据用户选

择的数据源类别,调用相应的数据连接模块。由用户在对应的连接模块界面输入数据源连接的关键字。将连接关键字生成相应的数据源连接字符串,并写入配置文件(见图4)。用户根据读取的数据源与数据转换模板中的数据结构,配置数据源与标准数据结构的映射关系,消除数据源的语义异构问题。在MHIDRT中,集成服务组件按照数据源类型及配置文件中的数据源连接字符串调用查询适配接口的数据访问方法。为了实现对异构数据的转换,通过数据源连接字符串提取该用户的业务数据结构 $Stru_i \rightarrow I(fc \rightarrow T_{iy}(C_{ix}))$,与保存了业务数据表的字段名、字段的数据类型等信息的平台标准数据结构 $Sour_i \rightarrow I(fc \rightarrow T_{iy}(C_{ix}))$ 进行匹配(见图5)。

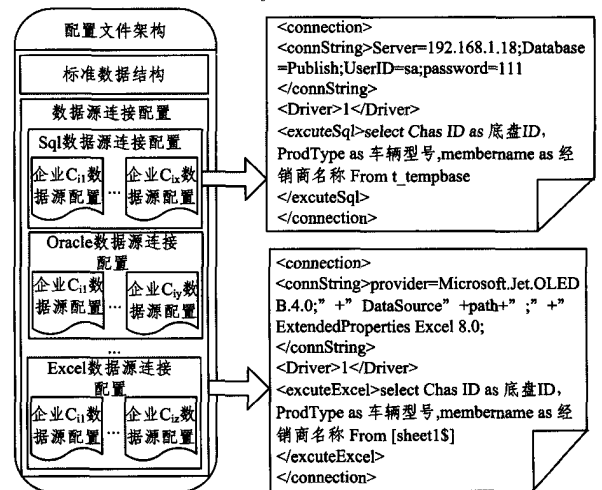


图4 配置文件架构

完成异构数据转换过程后,将MHIDRT中生成的数据访问接口地址通过IIS发布,同时将由用户填写的企业信息注册至UDDI。协作企业就可以通过服务访问接口查询转换后的多源数据,从外部看所有数据源的数据命名方式和数据模

式都是统一的,从而实现对各协作企业结构数据、非结构的查看与集成,无需与多种系统交互。对于协作企业来说,通过这

种方式将企业内部数据与平台集成,部署和集成的费用大大降低。

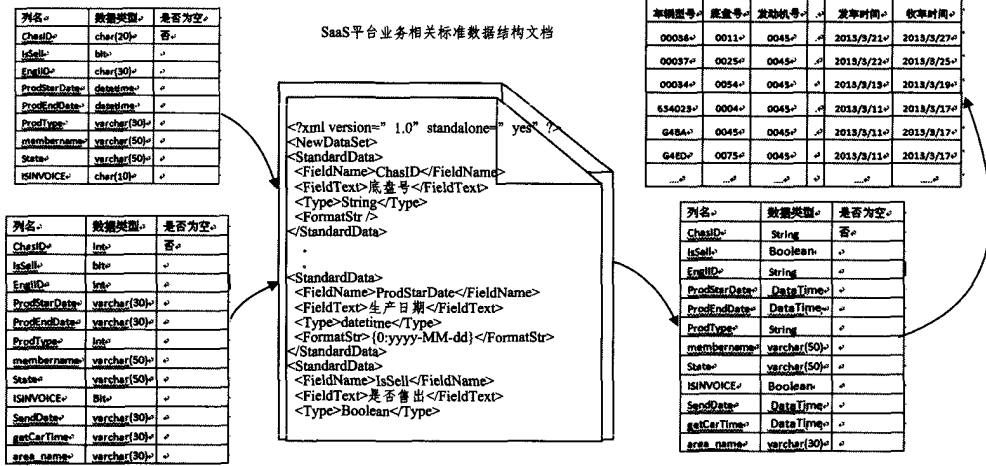


图5 异构数据转换

多源异构信息转换与注册算法如下。

输入:用户标识 $Ag_{Mid}(T_{iy})$ 及用户所在企业业务类型相关的数据内容 $Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$

输出:服务的描述信息 WS_i

BEGIN

Step1: /* 根据 $Ag_{Mid}(T_{iy})$ 解析协作企业的 $I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 及片区信息 $I(gc_{T \rightarrow R}(f_{C \rightarrow T_{iy}}(C_{ix})))$ */

Step1-1: $I(f_{C \rightarrow T_{iy}}(C_{ix})) = GetCorpName(Ag_{Mid}(T_{iy}))$;

Step1-2: $I(gc_{T \rightarrow R}(f_{C \rightarrow T_{iy}}(C_{ix}))) = GetAreaName(Ag_{Mid}(T_{iy}))$;

Step2: if $I(f_{C \rightarrow T_{iy}}(C_{ix})) \neq Null \ \& \ I(gc_{T \rightarrow R}(f_{C \rightarrow T_{iy}}(C_{ix}))) \neq Null$

Authority($Ag_{Mid}(T_{iy})$) /* 属于该联盟,授予用户向平台注册的权限 */

Step3:

Step3-1: $DataSourceType = SelectDSType(I(f_{C \rightarrow T_{iy}}(C_{ix})))$ /* 选择本地数据源类型 */

Step3-2: $LoadDSCControl(DataSourceType)$ /* 加载对应的数据连接控制模块 */

Step4: $SQLDB = CreateDBSql()$ /* 根据用户输入的数据源连接关键字,生成数据源连接字符串 */

Step5: $SaveSQLDBToWeb(SQLDB)$ /* 将数据源连接字符串写入 Web Services 中 */

Step6: $TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})) = GetDSstruct(I(f_{C \rightarrow T_{iy}}(C_{ix})))$ /* 根据企业协作类型,获取 MHIDRT 中的平台标准数据结构 $TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ */

Step7: $Match(TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})), Stru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})))$ /* 提取数据源的数据结构与平台标准数据集结构进行映射,控制 Web Services 服务中所展示的数据列的个数及列名 */

Step8: /* 对源数据 $Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 进行数据类型标准化转换 */

foreach (DataRow dr in 表中的行)

{ if (dr["Type"]. ToString() == "Boolean")

{ DataColumn dc = new DataColumn(dr["列名"]. ToString(), typeof(Boolean)); dt. Columns. Add(dc); }

else if (dr["Type"]. ToString() == "datetime")

{ DataColumn dc = new DataColumn(dr["列名"]. ToString(), typeof(DateTime));

dt. Columns. Add(dc); }

else if (dr["Type"]. ToString() == "String") { ... } }

Step9: $Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})) = GetDS(SQLDB)$ /* 根据数据源连接字符串,提取业务数据 */

Step10: /* 循环访问业务数据内容,按照 Step8 的数据集结构进行数据类型转换 */

Foreach(DataRow dr in $Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$)

Convert. ChangeType ($Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})), TempSour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})). Columns[h]. DataType$) 转换后构成标准数据:

(($Sour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))) \rightarrow (TempSour_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))) \oplus ((Stru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))) \rightarrow (TempStru_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})))$)

Step11: /* 将 MHIDRT 的安装地址通过 IIS 发布 */

Step12:

Step12-1: $ASSP(I(f_{C \rightarrow T_{iy}}(C_{ix}))) = CreateWebAddress(IISAddress)$ /* 根据 IIS 发布的地址生成企业 C_{ix} 的服务地址 */

Step12-2: $RegistToUddi(WSID_i, WSN_i, WST_i, ASSP(I(f_{C \rightarrow T_{iy}}(C_{ix}))), IISAddress)$ /* 将 WS_i 注册至 UDDI 中心 */

END

4.2 多源异构信息动态发现

为实现多源异构信息动态发现与动态集成(见图3),满足产业链协作平台多企业用户对多源异构数据的不同查询需求,将QE分为查询解析、查询分解、查询执行3个模块。根据用户身份将调用结果按照业务类型限定的数据格式传递给用户。查询解析:解析用户身份,控制界面查询控件的生成,限定控件查询范围,将用户提交的查询进行编译,生成查询计划。查询分解:利用查询解析的结果,把对片区数据的查询、联盟所有片区数据的查询分解为针对各个异构数据源的查询。查询执行:执行查询解析及查询分解动作完成后的查询任务。为提高查询处理效率,基于Hadoop框架,将查询分解后得到的多数据源查询任务转换成并行MapReduce作业组成的作业调度提交给查询执行器执行。

按照用户类型、需求的不同,对数据源的动态发现分为单点数据源调用和多点数据源动态集成,具体如下:

(1)若 $S_{AT \rightarrow Ag}(Ag_{AM}(T_{iy})) = AT_C(T_{iy})$ 或 $S_{AT \rightarrow Ag}(Ag_{AM}(T_{iy})) = AT_T(T_{iy})$ 。直接对注册的服务进行查询,是对单点数据源的动态调用结果,查询到的业务数据为: $Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 。

(2)若 $S_{AT \rightarrow Ag}(AG_{GM}(T_{iy})) = AT_A(T_{iy})$ 或 $S_{AT \rightarrow Ag}(AG_{GM}(T_{iy})) = AT_U(T_{iy})$ 。针对此类多数据源的动态查询,为提高查询执行效率,构建一个 Hadoop 集群,将集群分成两大类的角色:Master 和 Slave,Master 主要配置 NameNode 和 JobTracker 的角色,负责总管分布式数据和分解任务的执行^[16]; Slave 主要配置 DataNode 和 TaskTracker 的角色^[17],负责分布式数据存储以及任务的执行。JobTracker 把搜索任务分发到若干台 DataNode,TaskTracker 根据分配的任务对数据进行访问。

Map 函数的主要任务是根据查询范围对数据进行筛选,它以企业名称及企业数据作为输入,即 $(C_{ix}, Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})))$; 输出以企业名称为 key,以筛选后的数据为 value 值。Reduce 主要完成数据的重组操作,Reduce 函数以 Map 函数的输出作为输入值,根据 key 值对数据进行重组,返回符合用户查询需求的数据集合。下面将利用 QE 实现由查询事件驱动的数据源动态调用算法。

输入:用户标识 $Ag_{Mid}(T_{iy})$

输出: $Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 筛选后的数据

BEGIN

Step1: $I(f_{C \rightarrow T_{iy}}(C_{ix})) = GetCorpName(Ag_{Mid}(T_{iy}))$; /* 平台根据用户标识 $Ag_{Mid}(T_{iy})$,解析用户所在协作企业的 $I(f_{C \rightarrow T_{iy}}(C_{ix}))$; */

Step2: if ($S_{AT \rightarrow Ag}(AG_{GM}(T_{iy})) = AT_A(T_{iy})$)

Get($I(g_{C.T \rightarrow R}(f_{C \rightarrow T_{iy}}(C_{ix})))$);

Else if ($S_{AT \rightarrow Ag}(AG_{GM}(T_{iy})) = AT_U(T_{iy})$)

加载该联盟的所有片区信息;

Step3: $DataSet.ReadXML();$ /* 读取 $Temp_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ 的 xml 文件,按标准数据结构生成空 dataset */

Step4: $ForEach(WSID_i \text{ in } Uddi)$ /* 查找注册中心所有企业服务信息 */

Step5: for each C_{ix} in $R_{ix}(T_{iy})$:

$Wsdynamicpory ws = new Wsdynamicpory(C_{ix} + "?WSDL")$

/* 生成动态代理类动态调用服务 */

$object obj = ws.CreateInstance("servicename");$ /* 创建服务代理类实例 */

$DataSet ds[x] = (DataSet)(ws.Invoke(obj, servicename));$

/* 动态调用 Web Services 的方法,将结果存入数据集 */

$result = result + \{ds[x]\};$

Step6: /* Step5 的结果为 map 函数的输入,基于 MapReduce 对数据进行查询优化 */

$map(key, value)$

for each $Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix}))$ in value;

$valueinfo.set(C_{ix} + ";" + get(Query + Standard_i \rightarrow I(f_{C \rightarrow T_{iy}}(C_{ix})))$);

$keyinfo.set(R_{ix}(T_{iy}))$;

$context.write(keyinfo, valueinfo)$;

$reduce(String key, Collection value)$ /* 同片区数据进行合并输出 */

END

5 多源异构信息动态集成技术在配件库存协同调拨的应用

多源异构信息动态集成技术已经在汽车产业链协同 SaaS 平台的售后服务管理模块进行了应用验证。汽车产业链协同 SaaS 平台为汽车及零部件多条产业链的上下游企业之间的业务协作提供信息化支持,业务范围涵盖整车协同销

售、整车及零部件协同售后服务及零部件的协同采购等,目前已应用该平台的企业达 5000 多家^[12]。

通过多源异构信息动态集成技术的应用,支持在协同售后服务过程中盟主企业以及片区主管对分布式库存各节点的实时数据查询,从而实现分散在全国各地的服务库存进行及时的配件调拨。

(1) 多源异构配件库存数据转换与动态发布验证

各个服务商在平台下载 MHIDRT 包,选择数据源类型及链接地址,根据给出的数据源信息读出配件库存数据,进行数据表列名的映射(见图 6)。自动将映射值、数据源类型及链接地址写入 MHIDRT 中的数据接口文件,在 MHIDRT 中包含的服务页面中调用该数据文件,提供数据访问服务。将 MHIDRT 中生成的服务地址通过 IIS 发布,同时自动将用户填写的企业信息注册至 UDDI 注册中心(见图 7)。

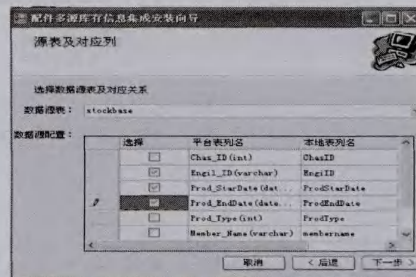


图 6 配件多源异构库存信息发布工具界面

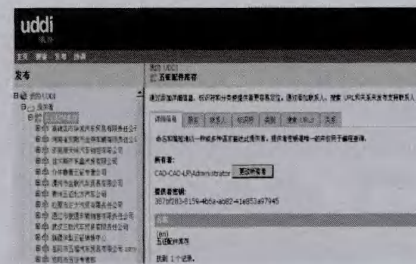


图 7 UDDI 配件库存服务注册中心界面

(2) 配件库存数据动态发现与集成验证

注册成功后,各类型用户可通过平台调用多源异构信息。图 8、图 9 分别为盟主企业用户登录平台后对全国所有片区各中转库、服务商配件库存数据动态查询的界面。

转库名称	中转库片区	配件型号	查询	备注用户					
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	019	杭州-威震汽车零部件有限公司	False	已上线	3895	四川片区

图 8 盟主企业、片区管理员对中转库配件库存信息的整合查看界面

转库名称	服务商片区	配件型号	查询	备注用户					
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区
1.12024799172426	0	2012-8-10:00:00	PC1040721K	19024	成都新瑞汽车零部件有限公司	False	已上线	3895	四川片区

图 9 盟主企业、片区管理员对服务商配件库存信息的查看界面

图 10 为中转库管理员对其配件库存数据动态查询界面。
图 11 为服务站业务员配件库存信息查看界面。

配件ID	销售状态	生产日期	配件型号	中转库编号	中转库名称	交接状态	备注ID	片匠名
L22048991704126	0	2012-8-5 0:00:00	WE4103P-10	019	川渝成都东康汽车服务有限公司	已上车	5893	四川片区
L22048991339100780	0	2012-7-28 0:00:00	F21304P21K	019	川渝成都东康汽车服务有限公司	已上车	5893	四川片区
L22048991339100700	0	2012-8-5 0:00:00	F21304P21K	019	川渝成都东康汽车服务有限公司	已上车	5893	四川片区

图 10 中转库管理员配件库存查看界面

配件ID	销售状态	生产日期	配件型号	服务站编号	服务站名称	服务站交接状态	服务站平台ID	发布日期	片匠名
L22048991339101990	0	2012-8-5 0:00:00	WE4103P-10	19034	成都和顺汽车服务有限公司	已上车	5893	2013-3-7	四川片区
L22048991409126102	0	2012-8-5 0:00:00	F21304P21K	19034	成都和顺汽车服务有限公司	已上车	5893	2013-3-12	四川片区
L22048991339100700	0	2012-8-5 0:00:00	F21304P21K	19034	成都和顺汽车服务有限公司	已上车	5893	2013-3-18	四川片区
L22048991339100700	0	2012-8-5 0:00:00	F21304P21K	19034	成都和顺汽车服务有限公司	已上车	5893	2013-3-18	四川片区

图 11 服务站业务员配件库存查看界面

结束语 本文针对产业链协作企业及其数据动态变化以及数据分散在各地难于集成的现状进行分析,对多源异构信息集成模型、多源异构信息转换与注册技术、事件驱动的数据源动态调用技术等进行了研究。通过对多源异构信息动态发布工具 MHDRT 的设计,实现了对动态变化的异构数据源的自动转换及动态发布。通过对事件驱动的查询引擎 QE 的研究,完成了用户权限及角色相关的数据源动态集成方法。将多源异构信息服务集成技术应用于产业链协同 SaaS 平台,有助于产业链中小型协作企业以低成本快速部署实现信息的共享,满足了数据源动态变换环境下的跨企业数据集成和查询,大幅度提高了业务协作紧密配合所需数据的准确性和实时性。

参考文献

[1] Schuberta P, Legnerb C. B2B integration in global supply chains: An identification of technical integration scenarios[J]. Strategic Information Systems, 2011, 20(3): 250-267

[2] Marozzo F, Talia D, Trunfio P. P2P-MapReduce: Parallel data processing in dynamic Cloud environments[J]. Journal of Computer and System Sciences, 2012, 78(5): 1382-1402

[3] Wang Shu-ying. Dynamic data exchange technology for collaborative commerce platform of industrial chain[J]. Computer Integrated Manufacturing Systems, 2010, 16(6): 1336-1345(in Chinese)

王淑莹. 面向产业链协同商务平台的动态数据交换技术研究[J]. 计算机集成制造系统, 2010, 16(6): 1336-1345

[4] Wang Shi-shui, Wang Yuan-yuan, Gao Ying-bo. Research and realizations of data integration ETL-based optimization [J]. Journal of Tianjin Polytechnic University, 2013, 32(3): 79-81(in Chinese)

王世水, 王元元, 高应波. 基于 ETL 的数据集成优化研究与实现[J]. 天津工业大学学报, 2013, 32(3): 79-81

[5] Fan Wen-hui, Xiao Tian-yuan. Integrated architecture of cloud manufacturing based on federation mode[J]. Computer Integrated Manufacturing Systems, 2011, 17(3): 469-476(in Chinese)

范文慧, 肖田元. 基于联邦模式的云制造集成体系架构[J]. 计算机集成制造系统, 2011, 17(3): 469-476

[6] Li Wen-xiong, Yan Mao-de, Wang Jian-wei. Ontology-based data

integration for intelligent transport systems[J]. Journal of Central South University, 2013, 44(7): 3038-3045(in Chinese)

李文雄, 闫茂德, 王建伟. 智能交通系统本体数据集成[J]. 中南大学学报, 2013, 44(7): 3038-3045

[7] Fan Jing, Xiong Li-rong, Xu Cong. Research and Application of Data Integration in Distributed Enterprise Service Bus Platform [J]. Computer Science, 2014, 41(2): 206-214(in Chinese)

范菁, 熊丽荣, 徐聪. 分布式企业服务总线平台数据集成研究及应用[J]. 计算机科学, 2014, 41(2): 206-214

[8] Wang Yi, Yi Shu-ping, Long Yue, et al. Study of Car Corporate Technology Data Integration Based On SOA[J]. Journal of Hunan University, 2010, 37(5): 36-39(in Chinese)

王毅, 易树平, 龙跃, 等. 基于 SOA 的整车企业技术数据集成研究[J]. 湖南大学学报, 2010, 37(5): 36-39

[9] Yang Jian-jun, Li Ju-bo, Deng Xiao-zhong, et al. A Web services-based approach to develop a networked information integration service platform for gear enterprise[J]. Journal of Intelligent Manufacturing, 2012, 23(5): 1-12

[10] Shi Yu-liang, Luan Shuai, Li Qing-zhong, et al. TLA Based Customization and Verification Mechanism of Business Process for SaaS[J]. Chinese Journal of Computers, 2010, 33(11): 2055-2067(in Chinese)

史玉良, 栾帅, 李庆忠, 等. 基于 TLA 的 SaaS 业务流程定制及验证机制研究[J]. 计算机学报, 2010, 33(11): 2054-2067

[11] Li Xiao-na, Li Qing-zhong, Kong Lan-ju, et al. Research on multi-tenant data partition echanism for SaaS application based on shared schema [J]. Journal on Communications, 2012, 33(Z1): 111-120(in Chinese)

李晓娜, 李庆忠, 孔兰菊, 等. 基于共享模式的 SaaS 多租户数据划分机制研究[J]. 通信学报, 2012, 33(Z1): 111-120

[12] Wang Shu-ying. Adaptive Evolution Technology for Collaborative SaaS Platform of Industrial Chains[J]. Journal of Southwest Jiaotong University, 2012, 47(1): 39-45(in Chinese)

王淑莹. 支撑产业链协同的 SaaS 平台自适应演化技术[J]. 西南交通大学学报, 2012, 47(1): 39-45

[13] Li Hou-fu, Han Yan-bo, Hu Song-lin, et al. An Approach to Constructing Service-oriented and Event-driven Application Dynamic Alliances[J]. Chinese Journal of Computer, 2005, 28(4): 739-749(in Chinese)

李厚福, 韩艳波, 虎嵩林, 等. 一种面向服务、事件驱动的企业应用动态联盟构造方法[J]. 计算机学报, 2005, 28(4): 739-749

[14] Zheng Li-yu, Terpenney J. A hybrid ontology approach for integration of obsolescence information[J]. Computers & Industrial Engineering, 2013(65): 485-499

[15] Chang F, Dean J, Ghemawat S, et al. Bigtable: a distributed storage system for structured data[J]. ACM Transactions on Computer Systems, 2008, 6(2): 4-9

[16] Belter A, Martinez J R, Aznar J I, et al. The GEYSERS optical testbed: A platform for the integration, validation and demonstration of cloud-based infrastructure services [J]. Computer Networks, 2014, 61(1): 197-216

[17] Lu Lu, Shi Xuan-hua, Jin Hai, et al. Morpho: A decoupled Map-Reduce framework for elastic cloud computing[J]. Future Generation Computer Systems, 2013, 36(7): 80-90