DOI：10.7652/xjtuxb201302020

# 一种用户主导的跨组织数据按需集成方法

温彦[1,2]，刘晨[2]，韩燕波[2]
（1.中国科学院计算技术研究所，100190，北京；2.北方工业大学云计算研究中心，100144，北京）

**摘要**：为了在互联网上实现跨组织数据共享，以及及时响应多变的临机业务的需求，在数据服务模型的基础上，提供了一种以用户为主导的数据服务组合方法。该方法可帮助用户按需集成跨域数据，以基于概率的模式匹配方法作为算子实现的基础，将该方法用于一个交互式的组合环境中，通过向用户不断推荐合适操作来完成组合过程。利用对用户反馈的学习来优化推荐内容和组合过程，通过案例、原型系统和相关工作的比较分析，验证了该方法具有较好的实用性。

**关键词**：数据服务；按需集成；服务组合
**中图分类号**：TP393 **文献标志码**：A **文章编号**：0253-987X(2013)02-0116-08

## On-Demand Integration of Cross-Organizational Data in a User-Guided Way

WEN Yan[1,2], LIU Chen[2], HAN Yanbo[2]
(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
2. Cloud Computing Research Center, North China University of Technology, Beijing 100144, China)

**Abstract**: To enable cross-organizational data sharing on the Internet, and to achieve quick adaptation to situational and changing requirements, a user-guided data service composition approach is proposed based on the data service model. The proposed approach can help users integrate the cross-organizational data in an on-demand manner. A set of data service composition operators is proposed, and uses the probability-based schema matching techniques as its implementation basis. An interactive composing environment is developed, in which the schema matching techniques are used to recommend appropriate operations and users' feedbacks are learned to optimize the composition process. The usability of this approach is validated via a use case study, a preliminary implementation and the related work comparisons.

**Keywords**: data service; on-demand data integration; service composition

近年来，越来越多的企业和组织都选择依托开放互联网实现动态业务协作，即时共享业务数据，实现跨域数据的动态集成。集成需求具有临机性、即时性的特点，往往需要用户的参与。由于数据源的自治性、异构性，使集成需求无法直接提供给用户，传统的基于中介模式的方法[1-2]则存在着集成代价大、灵活性差的问题，无法满足即时集成需求，因此一般业务用户无法快速、正确地构建集成逻辑。

为应对上述挑战，基于前期的工作[3-4]，本文提出了一种支持用户即时汇聚多源数据、构造个性化数据视图的方法，其主要贡献包括：提出了一组数据服务组合算子，并且在一个交互式的组合环境中，以基于概率的模式匹配方法作为这些算子的实现基础，通过向用户不断推荐合适的操作来完成组合过程，同时利用用户的反馈来优化推荐内容和组合过程。

收稿日期：2012-07-05。 作者简介：温彦(1984—)，女，博士生；韩燕波(通信作者)，男，教授，博士生导师。 基金项目：国家自然科学基金资助项目(60903137，61033006)。
网络出版时间：2012-12-10 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20121210.1008.002.html

# 1　案　例

某公司试图从 2 个候选的研究所中选择一个作为学术合作对象。该公司的经理尝试整合并分析这 2 个研究所的成员信息，而这些信息来源于不同的数据源。例如，该经理希望得到研究所成员所发表的论文列表，以便用论文的被引用次数来评估论文质量。成员的论文列表信息从研究所的网页上获取，论文的引用信息从 Google Scholar 上获取。该经理需要上述数据源形成的综合视图，以便于立即获得其中的关键信息。上述过程涉及的数据源模式如图 1 所示，其中 $S_1$、$S_2$ 为 2 个研究所的成员及其发表论文信息的数据模式，$S_3$ 为从 Google Scholar 获得的论文引用情况的模式。该过程的实现存在若干挑战和需求：①数据源分布异构，使该经理无法直接集成它们；②用户有即时的、按需集成的需求，例如，该经理可能需要添加成员的教育背景或论文发表刊物，以获得更加全面的信息，且该经理应当能够决定以何种方式集成这些数据，如图 1 所示，$S_1$、$S_2$ 的集成结果可能是 $V_1$ 或 $V_2$，这取决于实际需求，其中，$V_1$ 表示对各研究所成员信息进行比较后形成的视图，$V_2$ 表示将各研究所成员信息进行合并形成的视图；③数据源具有多种匹配的可能性，如在图 1 中，$S_1$、$S_2$ 具有相同的名字和结构，但由于它们来自于不同的研究所，因此它们的数据实例是不相交的，经理应当了解这些信息并据此按需集成数据。

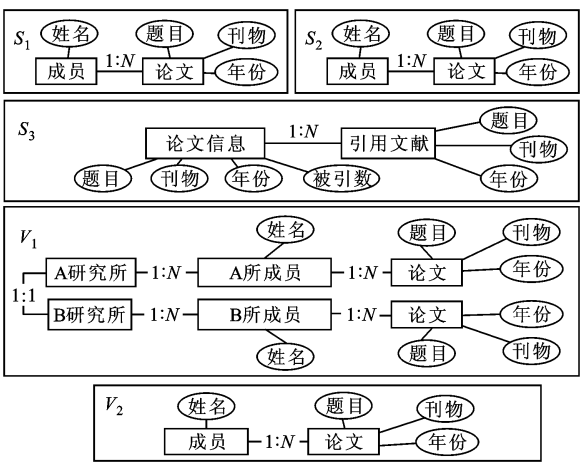


图 1　案例中的数据源模式和视图模式

# 2　iViewer 方法的原理

## 2.1　数据服务模型

传统的 Web 服务模型及服务组合技术主要关注业务流程的构建，不强调数据方面的特征，因此不是很适用于数据集成。本文基于之前的工作[3-4]，提出了数据服务模型，实现了互联网环境下对跨域分布数据的透明访问。嵌套关系模型简单、能够表达复杂数据对象，可直接可视化为嵌套表格，是互联网上常见的结构化和半结构化数据良好的承载模型[3]。数据服务利用嵌套关系作为基本数据模型，两者的定义如下。

**定义 1**　嵌套关系包含模式和实例。令 $A$ 是属性名称全集，嵌套关系模式 $R(S)=(A_1,A_2,\cdots,A_m)$，其中 $R\in A$，$R$ 是关系模式的名称，$S$ 是属性名称列表，$A_i$ 是原子属性，或者形如 $R_i(S_i)$ 的子关系属性，$A_i\in A$。$R(S)$ 的一个实例是形如 $(a_1,a_2,\cdots,a_m)$ 的有序元组集合，若 $A_i$ 是原子属性，则 $a_i$ 是基础数据值；若 $A_i$ 是关系模式，则 $a_i$ 是它的实例。

对于嵌套关系模式 $R$，令 $R_{AA}$ 为 $R$ 的原子属性集合，$R_{RA}$ 为 $R$ 的子关系属性集合，$R_{AT}$ 为 $R$ 递归包含的所有原子属性集合，$R_{RT}$ 为 $R$ 递归包含的所有子关系属性集合。假设 $t$ 是 $R$ 实例中的一个元组，且 $A$ 为 $R$ 的属性，则 $t[A]$ 表示 $r$ 在属性 $R$ 上的取值。

**定义 2**　数据服务为五元组 $\langle U,N,P,R,I\rangle$，其中 $U$ 为 Web 可访问位置及其全局标识，$N$ 为名称，$P$ 为输入参数，$R$ 为服务所暴露数据的模式（嵌套关系模式），$I$ 为该服务运行时的数据实例（嵌套关系实例）。

一个数据服务可直接在嵌套表格中呈现，其模式表现为表头，其实例如表 1 所示。嵌套表格是类似于 Spreadsheet 的最终用户编程环境，它简单、直观，便于用户进行业务数据分析和操作[5]，并具有嵌套关系代数这一强大的理论根基作支撑。

表 1　研究所成员发表的论文信息

| 成员姓名 | 论文 | | |
|---|---|---|---|
| | 题目 | 年份 | 刊物 |
| Hua Wei | A Dimensionality Reduction Framework for Detection of Multiscale Structure in Heterogeneous Networks | 2012 | J. Comput. Sci. Technol. |
| Shen | Exploring the Structural Regularities in Net Works | 2011 | Physical Review E |

## 2.2　iViewer 方法

基于数据服务的概念，本文所述方法的基本原理如图 2 所示。各种类型的数据源利用文献[4]中

的方法被封装为数据服务，将一组数据服务上的组合算子作为对文献[3]中的单表数据和模式处理算子的扩展，该单表算子如表 2 所示。本文的组合算子为数据服务间基本的组合模式，在此基础上，提出了基于概率的匹配方法，用以发现数据服务间的映射关系，支持数据服务组合。基于嵌套表格的可视化服务组合环境，组合过程采用交互式方法实现(见图 2)。系统通过匹配关系向用户推荐合适的操作，用户接受或者拒绝系统的推荐，系统接收并分析用户的反馈，调整后续的推荐操作，优化组合过程，直至完成所需数据视图的构建。

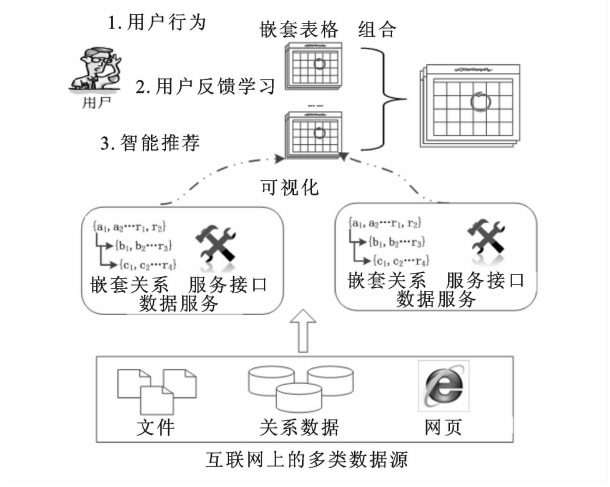


图 2 iViewer 方法的原理

## 3 数据服务组合算子

由于数据服务为具有一定存取约束的嵌套关系数据，兼具服务和数据特性，因此可通过输入输出的关联实现组合，而服务的调用会返回嵌套关系的元组集合，可在其上进行基于嵌套关系的数据组合操作。具体的服务组合算子如表 3 所示。

考虑到服务所封装数据源的异构性，将组合算子 2～5 建立在服务间的映射关系上。在基于嵌套关系的服务之间，存在着原子属性层面和关系层面的映射关系，此处假设映射关系是 1-1 形式的，它们的定义如下。

**定义 3** 数据服务 $s_1$、$s_2$ 数据模式上的原子属性的映射关系 $M_A$，子关系属性的映射关系 $M_R$，全部映射关系 $M$ 为满足下列条件的属性对集合：

$$(1)\ (M_A \subset s_1.R_{AT} \times s_2.R_{AT}) \wedge (\forall <a_1,b_1>, <a_2,b_2> \in M_A, a_1 \neq a_2 \wedge b_1 \neq b_2)$$

$$(2)\ (M_R \subset s_1.R_{RT} \times s_2.R_{RT}) \wedge (\forall <a_1,b_1>, <a_2,b_2> \in M_R, a_1 \neq a_2 \wedge b_1 \neq b_2) \wedge (\forall <a,b> \in M_R, (\forall u \in a.A, \exists v \in b.A, <u,v> \in M_A) \wedge (\forall v \in b.A, \exists u \in a.A, <u,v> \in M_A) \wedge (\forall u \in a.R, \exists v \in b.R, <u,v> \in M_R) \wedge (\forall v \in b.R, \exists u \in a.R, <u,v> \in M_R))$$

表 2 单表数据和模式处理操作算子

| 类型 | 操作 | 描述 | 相应嵌套关系代数算子 |
|---|---|---|---|
| 数据处理 | 过滤：Filter(col,condition) | 根据条件表达式 condition 对嵌套表格中某一列 col 的数据进行过滤，该列可以为原子或子关系属性 | $\sigma$ |
| | 排序：Sort(col,order) | 对嵌套表格中的数据以某列 col 的值为条件进行排序，order 为升序或降序，该列仅为原子属性 | $s$ |
| | 头部：Header(count) | 截取嵌套表格开头的部分数据(数量为 count) | |
| | 尾部：Tail(count) | 截取嵌套表格结尾的部分数据(数量为 count) | |
| | 增加函数：AddFunction(col,function) | 对嵌套表格中的一个或多个列进行算术或字符运算，并将其结果作为新增列，col 为原子属性 | |
| | 合并实例：MergeInstance(col) | 合并某列 col 上数值重复的数据 | |
| 模式处理 | 增加列：AddColumn(col,pos) | 在特定的位置上插入一列 | |
| | 删除列：DeleteColumn(col) | 将嵌套表格中的某列 col 删除 | $\pi$ |
| | 重命名列：RenameColumn(col,name) | 将嵌套表格中的某列 col 重命名为 name | $\rho$ |
| | 嵌套：Nest(cols,newCol) | 将普通的表格转换为嵌套表格 | $\nu$ |
| | 解嵌套：Unnest(col) | 将嵌套表格转换为普通表格 | $\mu$ |

(3) $M=M_A \bigcup M_R$

将 $M$ 中存在于 $s_1$ 模式中的属性集合记为 $M_L$，将 $M$ 中存在于 $s_2$ 模式中的属性集合记为 $M_R$。对于属性 $u$，令 $P_M(u)$ 为它在 $M$ 中映射的属性，则交集操作的定义如下。

**定义 4**　对于数据服务 $s_1$、$s_2$，假设他们的映射关系为 $M$，则 $s_1$、$s_2$ 的交集(记为 $\bigcap M$)仍为数据服务，记为 $s$，则 $s.U$ 和 $s.N$ 分别为新服务的访问地址和名称。$s.P=s_1.P \bigcup s_2.P, s.R=s_1.R, s.I=s_1.I \bigcap_M s_2.I, \bigcap_M$ 为嵌套关系上的相交操作，其定义如下：

$$(1)\ r_1 \bigcap_M r_2=\{t \mid \exists t_1 \in r_1, \exists t_2 \in r_2 \mid \forall u \in (M_L \bigcap R_{1AA}) \mid t[u]=t_1[u]=t_2[P_M(u)] \wedge t[R_{1AA}-M_L]=t_1[R_{1AA}-M_L]\}$$

其中 $r_1$、$r_2$ 为 2 个平面关系，$R_1$ 为 $r_1$ 的模式；

$$(2)\ r_1 \bigcap_M n_2=\{t \mid \exists t_1 \in r_1, \exists t_2 \in n_2 \mid t[R_{1AA} \bigcap R_{2AA}]=\bigcap_M(t_1, t_2[R_{2AA}]) \wedge (t[A_1]=\bigcap_M(t_1, t_2[R_{21}]) \wedge (\forall a_1 \in A_1, \exists b_1 \in R_{21AT}, <a_1, b_1> \in M)) \wedge (t[A_2]=\bigcap_M(t_1, t_2[R_{22}]) \wedge (\forall a_2 \in A_2, \exists b_2 \in R_{22AT}, <a_2, b_2> \in M)) \wedge \cdots (t[A_l]=\bigcap_M(t_1, t_2[R_{2l}]) \wedge (\forall a_l \in A_1, \exists b_l \in R_{2l}.A_T, <a_l, b_l> \in M)), A_l \in R_1.A, R_{2j} \in n_2.R, (1 \leqslant i, j \leqslant l)\}$$

其中 $r_1$ 为平面关系，$n_2$ 为嵌套关系，$R_1$、$R_2$ 分别为 $r_1$、$n_2$ 的模式；

$$(3)\ n_1 \bigcap_M n_2=\{t \mid \exists t_1 \in n_1 \mid t[R_{1AA}]=\bigcap_M(t_1[R_{1AA}], n_2) \wedge t[R_{1l}]=\bigcap_M(t_1[R_{1l}], n_2) \wedge \cdots \wedge t[R_{1l}]=\bigcap_M(t_1[R_{1l}], n_2), R_{1i} \in R_{1RA}, (1 \leqslant i \leqslant l)\}$$

其中 $n_1$、$n_2$ 为 2 个嵌套关系，$R_1$、$R_2$ 分别为 $n_1$、$n_2$ 的模式。

在执行操作时，由于用户完全正确地指明所有的匹配关系比较困难且易出错，因此需要在部分匹配指定时能够正确地向用户展现中间结果，并且保证在逐渐完善匹配关系时，能基于新的匹配关系计算在当前中间结果上的增量。表 3 的组合算子具有如下的匹配分解性质，可在递增的匹配集合上正确执行。

**定理**　数据服务 $s_1$、$s_2$ 上的组合算子 $\bigcap_M$、$\bigcup_M$、$-_M$ 和 $\bowtie_M$ 满足以下匹配分解性质

$$s_1\ o_M\ s_2=s_1\ o_{M_1}\ s_2\ o_{M_1 \cup M_2}\ s_2 \cdots\ o_M\ s_2,\quad M_i \subset M, o \text{ 为} \bigcup, \bowtie, \bigcap \qquad (1)$$

$$s_1 -_M s_2=(s_1 -_{M_1} s_2) \bigcup (s_1 -_{M_1 \cup M_2} s_2) \bigcup \cdots \bigcup (s_1 -_M s_2),\quad M_i \subset M \qquad (2)$$

对于定理的证明过程是将嵌套关系扁平化为关系数据，则 $s_1$、$s_2$ 上的组合算子可以被看作是相应的关系代数操作，而 $M$ 可看作是关系间的连接条件。式(1)、式(2)的两边具有等价的选择语句，因此能够生成相同的关系数据，从而可通过多个嵌套操作最终生成相同的嵌套关系数据。

表 3　数据服务组合算子

| 类型 | 操作 | 缩写 | 含义 |
|---|---|---|---|
| 顺序组合 | connect($s_1$, $s_2$, {< $s_1.o_1$, $s_2.i_1$>, …, <$s_1.o_n$, $s_2.i_n$>}) | $\tau_M$ | 将服务 $s_1$ 的输出的属性 $o_1, \cdots, o_n$ 与服务 $s_2$ 的输入参数 $i_1, \cdots, i_n$ 连接 |
| 结果组合 | union($s_1.r_1$, $s_2.r_2$) | $\bigcup_M$ | 将服务 $s_1$ 和 $s_2$ 中的某个子关系 $r_1$ 与 $r_2$ 合并，基于嵌套关系的 union 操作 |
| | intersect($r_1$, $s_2.r_2$) | $\bigcap_M$ | 将服务 $s_1$ 和 $s_2$ 中的某个子关系 $r_1$ 与 $r_2$ 相交，基于嵌套关系的 intersect 操作 |
| | difference($s_1.r_1$, $s_2.r_2$) | $-_M$ | 将服务 $s_1$ 和 $s_2$ 中的某个子关系 $r_1$ 与 $r_2$ 求差，基于嵌套关系的 difference 操作 |
| | join($s_1.r_1$, $s_2.r_2$, $m$) | $\bowtie_M$ | 将服务 $s_1$ 和 $s_2$ 中的某个子关系 $r_1$ 与 $r_2$ 基于映射关系 $m$ 连接，基于嵌套关系的 join 操作 |

## 4　模式的不确定性匹配

数据服务组合算子的实现在很大程度上依赖于数据服务模式间的映射关系，但在互联网环境下，数据的模式很难精确获取和描述，尤其对于半结构化和无结构化数据，因此映射是不确定的[6]。针对数据和模式在不同方面的特征(例如属性名、关系、取值域等)开发了多种匹配方法，同时将它们综合使用以提高准确率[1-2]。传统的方法通常采用平均等方法将多个匹配结果聚合，可能会导致匹配证据的模糊化，例如图 1 中的 $S_1$ 和 $S_2$，基于名称和结构匹配器，他们是匹配的，而根据实例覆盖匹配器，他们则是不匹配的，而 2 个匹配结果的聚合会模糊具体结果，因此不利于用户理解。

为了解决映射关系不确定性的问题，分离了属性对之间由不同匹配器产生的不同匹配结果，并采

用概率值的方式来描述它们。基于多个匹配器[1]计算原子属性间可能的关系，为了实现数据源间灵活的集成方法，本文中属性间包含属性匹配($M$)和属性不匹配($\overline{M}$)两种关系。对各个匹配器设定不同的权值 $w_i$，以反映它们对匹配结果的影响。对于任意的属性对$<t_1,t_2>$，为匹配和不匹配结果分别设定分数，该分数是产生该结果匹配器的权值和。假设匹配器的集合为 $W$，对于匹配关系 $M$ 来说，其分数为

$$s(<t_1,t_2>.r=M)=\sum_{m_i.m_a(t_1,t_2)=M\wedge m_i\in W}w_i \quad (3)$$

式中：$m_a$ 为匹配器的匹配函数。基于式(3)中的分数，为属性对的各个匹配度设定一个可能性取值 $p$，并以此作为它的不确定性度量，即

$$p(<t_1,t_2>.r=M)=s(<t_1,t_2>.r=M)/\sum_{m_i\in W}w_i \quad (4)$$

由于灵活的嵌套和解嵌套操作，嵌套关系中的原子属性比子关系属性更稳定，因此利用子关系属性所递归包含的原子属性的匹配结果，来计算子关系属性的匹配结果，对于任意的子关系属性对$<T_1,T_2>$，则

$$p(<T_1,T_2>r=M)=\Big(\sum_{t_1\in T_1.A_T}\max_{t_2\in T_2.A_T}(p(<t_1,t_2>.r=M))+\sum_{t_2\in T_2.A_T}\max_{t_1\in T_1.A_T}(p(<t_1,t_2>.r=M))\Big)/(|T_1.A_T|+|T_2.A_T|) \quad (5)$$

式中：$T_1$、$T_2$ 为原子属性，与 $M$的计算方法类似。

## 5 交互式的数据服务组合过程

基于数据服务组合算子和不确定的模式匹配方法，交互式的服务组合过程是一个智能推荐→用户反馈→组合过程优化的循环，如图 2 所示。

当用户采用第 3 节的算子组合 2 个数据服务时，系统会计算属性间的映射关系，并以此作为智能推荐的基础。在映射匹配概率的基础上计算属性对的推荐优先度，并以此作为排序依据。假设服务 $s_1$ 与服务 $s_2$ 输出模式的原子属性集合分别为 $A$ 和 $B$，并且在 $A$ 中为 $B$ 中的每一个元素 $b_i$ 构造一个空属性 $b_i'$形成 $A'$，同理构造 $B'$，对任意属性 $a\in A$，它与 $B$ 中任意属性映射的排序优先度(设为 $d$)的计算方法为：① 若 $a_i\in A,b_j\in B$，则 $d(<a_i,b_j>)=p(<a_i,b_j>=M)$；② 若 $a_i\in A,b_j\in B'-B$，则 $d(<a_i,b_j>)=1-\max_k(p(<a_i,b_k>=M))$，其中 $b_k\in B$；③若 $a_i\in A'-A,b_j\in B$，则 $d(<a_i,b_j>)=1-\max_l(p(<a_l,b_j>=M))$，其中 $a_l\in A$。

对上述的推荐优先度进行排序，排序结果若来自①，则说明 $a_i$ 和 $b_j$ 形成较大匹配；若来自②，则说明 $a_i$ 需形成新列；若来自③，则说明 $b_j$ 需形成新列。例如，若$<a,b>$的匹配概率为 0.9，则在匹配的推荐排序中较为靠前，若 $a$ 与 $B$ 任意列实现匹配的概率均低于 0.1，则 $a$ 在添加新列的推荐排序中靠前。对子关系的匹配结果也进行如上排序过程，并以批量映射的形式推荐给用户以提高效率，推荐内容以示例数据呈现。

除了原子属性的映射关系，嵌套关系的结构特征也很重要，嵌套关系间存在原子属性匹配但嵌套关系层次不一致的情况，如 $A<B$、$C<D$、$E\gg$和 $A<D$、$C<B$、$E\gg$中 $B$-$B$、$D$-$D$ 和 $E$-$E$ 为实现映射的原子属性，但两者存在嵌套结构上的差异性。设$<a_1,b_1>$、$<a_2,b_2>$分别实现了匹配，但是服务 $s_1$ 中 $a_1$、$a_2$ 存在上层、下层、平级、不相关 4 种嵌套层次的关系，而 $s_2$ 中 $b_1$ 和 $b_2$ 也存在这 4 种关系，当 $a_1$、$a_2$ 的嵌套层次关系和 $b_1$、$b_2$ 的关系不同时，则需要对 $s_1$ 或 $s_2$ 进行结构调整，而嵌套关系的嵌套和解嵌套操作提供了灵活的调整方法，可选其中的一个服务为基准，对模式结构进行调整。

在数据内容上，可能存在组合的服务间匹配属性数据的处理操作不一致，而在很多情况下则要求合并的属性间应保持一致性，因此需要推荐数据操作的内容。

根据计算出的映射关系，针对内容合并、结构异构、内容异构等，提出了若干推荐规则，假设服务 $s_1$ 与服务 $s_2$ 的原子属性集合分别为 $A$ 和 $B$，子关系属性集合分别为 $P$ 和 $Q$(见表 4)。用户对推荐内容的反馈，表明了用户实际期望的数据处理方式，同时也会影响包含其祖先、孩子以及兄弟属性(同树结构中的概念)。反馈是系统优化后续组合过程的基础，并通过实时地调整匹配结果来实现。调整将会根据嵌套层次向上传播至其上级关系，因此会相应地调整推荐的顺序。

映射调整策略如表 5 所示，假设 2 个数据服务 $s_1$ 和 $s_2$ 组合，$a$、$c$ 是 $s_1$ 的原子属性，$b$、$d$ 是 $s_2$ 的原子属性。表 4、表 5 所示的推荐和调整规则及策略，覆盖了包含一元算子中的数据处理和模式处理部分，也覆盖了二元算子中的数据融合部分。其中，二元算子中映射关系的计算和推荐是核心。对于 2 个模式中所有的原子属性，只存在合并和不合并2种

表 4　推荐规则

| 类型 | 条件 | 推荐内容 |
|---|---|---|
| 内容合并 | $<a,b>$实现匹配，其中 $a\in A,b\in B$ | $a$ 和 $b$ 合并 |
| | $<a,b>$实现匹配，其中 $a\in A,b\in B'$-$B$ 或 $a\in A'$-$A,b\in B$ | 添加 $a$ 或 $b$ 属性为新列 |
| | 2 个子关系属性$<p,q>$匹配，$p\in P,q\in Q$ | 子关系 $p$、$q$ 的递归批量合并 |
| | $<p,q>$实现匹配，其中 $p\in P$，$q\in Q'$-$Q$ 或 $p\in P'$-$P,q\in Q$ | 添加 $p$ 或 $q$ 为新列 |
| 结构异构 | $<a_1,b_1>$、$<a_2,b_2>$实现匹配，但 $a_1$、$a_2$ 的嵌套层次关系和 $b_1$、$b_2$ 的嵌套层次关系不同 | 推荐嵌套和解嵌套操作以适配嵌套结构 |
| 内容异构 | 若$<a_i,b_j>$实现匹配，且 $a_i$ 上有过滤、排序、函数运算、合并重复列的数值操作 | $b_j$ 也进行同样操作，反之亦然 |

表 5　映射调整策略

| 推荐内容 | 用户反馈 | 调整规则 |
|---|---|---|
| 映射 $<a,b>$ | 接受 | 将其匹配概率调整为 1，将 $<a,d>$和$<c,b>$的匹配概率设为 0，其中 $c\neq b$，同时 $d\neq a$ |
| 映射 $<a,b>$ | 拒绝 | 将其匹配概率调整为 0 |
| 添加一列 $c$ | 接受 | 将任意$<x,c>$的匹配概率设为 0，其中 $x$ 是 $s_1$ 的原子属性 |
| 添加一列 $c$ | 拒绝 | 增加任意属性对$<x,c>$的匹配概率，其中 $x$ 是 $s_1$ 的原子属性，同时 $x$ 尚未形成匹配，且未被推荐与 $c$ 匹配 |

情况，因此表 4 中内容合并和结构异物的推荐规则能够覆盖模式合并的需求，并在此基础上进行结构和数据上的调整。表 5 中的策略主要是基于用户反馈进行映射关系的调整，调整内容包括了对匹配和不匹配两种关系的用户接受和拒绝反馈的调整，实现了对内容合并的 2 种情况的覆盖。对于结构异构性，用户可在不接受系统推荐的内容时通过拖拽的方式实现，最终转化到数据服务的嵌套和解嵌套操作上。

# 6　系统实现和案例分析

## 6.1　系统实现

用户界面采用 Flex 实现，如图 3 所示，工作区分为 5 个主要的区域，区域①是服务目录，被选服务的示例数据以嵌套表格的形式呈现在区域②中。区域③是当前的工作表单，用于呈现临时组合结果，用户将②中的数据拖拽至③，触发系统对数据匹配关系的计算，并由此推荐操作。区域④是推荐区，呈现推荐列表。区域⑤为历史操作记录，系统通过 MVC 架构实现，由控制器捕获用户行为，触发相应的算子执行过程或模式匹配过程。

原型系统中引入了多个匹配器用于计算数据服务的匹配结果，并生成有序的推荐操作列表，在用户反馈时更新列表内容和序列。当用户逐步地增加服务间的匹配关系时，利用定理来计算中间结果。

## 6.2　案例分析

采用本文开始提出的案例：经理获取研究所成员所发表的论文列表，并通过这些论文的被引用次数来评估其质量，最终以比较研究所的研究水平来说明本文方法的效果。

6.2.1　论文发表信息　为了获得 2 个研究所成员的论文信息，需要将图 1 中的 $S_1$ 和 $S_3$ 进行连接操作，采用本文所述的不确定性方法，计算 2 个服务各个属性间的匹配关系，形成较高匹配概率的原子和子关系属性如下：

$M_1$：$<S_1$：论文：题目，$S_3$：论文信息：题目$>$

$M_2$：$<S_1$：论文：刊物，$S_3$：论文信息：刊物$>$

$M_3$：$<S_1$：论文：年份，$S_3$：论文信息：年份$>$

$M_4$：$<S_1$：论文：题目，$S_3$：论文信息：引用文献：题目$>$

$M_5$：$<S_1$：论文：刊物，$S_3$：论文信息：引用文献：刊物$>$

$M_6$：$<S_1$：论文：年份，$S_3$：论文信息：引用文献：年份$>$

$M_7$：$<S_1$：论文，$S_3$：论文信息：引用文献$>$

$M_8$：$<S_1$：论文，$S_3$：论文信息$>$

未形成较高匹配概率的属性为：$S_3$. 论文信息. 被引数。系统首先推荐基于 $M_7$（包含 $M_4\sim M_6$）的批量映射作为连接条件，然而这并非用户需求，在用户拒绝后，$M_4\sim M_7$ 的匹配概率均调整为 0，系统继续推荐 $M_8$（包含 $M_1\sim M_3$），用户接受。之后，系统推荐添加 CitedNum 作为新列，用户接受，完成了 $S_1$、$S_3$ 的集成，称为 $S_4$。$S_2$ 和 $S_3$ 也用类似方法集成，称为 $S_5$。


图 3 原型系统的用户界面

6.2.2 比较 2 个研究所的情况 根据匹配计算的结果，$S_4$ 和 $S_5$ 的模式基本相同，但是由于 $S_4$ 和 $S_5$ 不存在任何相交数据，因此系统推荐两者 union，但这并非用户需求，因此被用户拒绝，系统推荐将 $S_5$ 整体作为一个新列加入 $S_4$，用户接受，从而完成了集成。

### 6.3 相关工作分析比较

本文的目标是实现在互联网环境下异源数据的按需集成，以应对临机和即时的业务需求，同时降低用户的集成工作量。目前，已有一些工作与本文的追求相同，例如各类 mashup 系统提供了面向用户的可视化编程方式的数据集成方法，其中数据流[7]和 Spreadsheet 是常见的 mashup 模式[5,8]。

Spreadsheet 是一个典型的最终用户编程环境[1,3,8]，Spreadsheet 将单元格作为第一位的元素，而缺少一个具有良好根基的数据模型作为基础，因此难以建立语义映射，从而增加了大量的人力工作[1]。因此，有些工作[3,8]在使用 Spreadsheet 作为可视化数据呈现和操作工具的同时，采用某些具有良好理论基础的数据模型（如关系模型）来支撑更强的集成能力。

为了实现对多变业务需求的快速响应，同时降低集成代价，也有一部分工作尝试包括即用即付（pay-as-you-go）的数据集成[9-10]，尽力而为（best effort）的数据抽取和集成[11]，不确定性的模式匹配和数据集成[6,12]，等等。

通过案例对多个相关工作进行三方面的比较：①系统实现；②用户参与集成的方式和代价；③组合能力。比较的工作涵盖了传统的[2]（Prompt）和近期的（SCP）交互式组合技术，流行的用户主导的数据 Mashup 方法（Yahoo! Pipes[7]），以及基于表格的集成模式（SpreadATOR[1]，SCP）。

6.3.1 系统实现 Prompt 关注本体和集成，其他的工作均关注包含网页、在线 feeds、数据文件等数据源的集成。SCP 基于扁平表呈现数据，Yahoo! Pipies 是流程图式的 Mashup 环境，两者均未定义一致的数据模型。Prompt 基于树形结构呈现本体，iViewer 同样面向多类数据源，利用嵌套关系作为统一的数据模型，易于重用和建立语义映射，而集成过程在嵌套表格上进行，也易于用户理解和操作。

6.3.2 用户参与和用户操作代价 在 Prompt 中，用户接受或拒绝系统推荐的 merge 或 copy 操作，Prompt 据此生成推荐列表，并发现潜在的不一致性等错误。在 SCP 中，用户使用简单的复制粘贴操作来导入和集成数据，系统尝试发现数据的结构以及过滤和连接的条件等。用户提供对自动完成内容的接受和反馈操作，代价较低。在 Yahoo! Pipes 中，用户操作定制的算子，形成数据处理流，用户完全主导 Mashup 过程，系统不吸收用户反馈，缺乏一些通用集成逻辑的定义，如连接操作，数据源的连接需通过抽取数据项，并在多个循环中逐个比较来实现，因此代价较高。在 SpreadATOR 中，用户需手动定义数据呈现和查询的脚本，工作量较大，虽然系统定义了一些可重用的模式以简化用户的工作，但系统不吸收用户反馈。在 iViewer 中，用户可执行简单的数据处理算子，多个服务的组合是通过交互式的系统推荐和用户反馈学习过程实现的。

6.3.3 组合能力 仿照 XML2OWL 的方法将嵌套模型转化为 OWL 文件并在 Prompt 上运行时，所得结果不能满足需求，原因在于它不考虑元素间的多种匹配关系，推荐能力有限。SCP 从扁平表中发现相同的列，并以此为连接条件附加新增的列，因此 SCP 无法处理具有名称和结构异构性的层次结构数据。SpreadATOR 基于强结构化的实体概念模型及其查询语言[13]构建 Mashup，因此灵活性较差。由于 Yahoo! Pipes 中的算子设计是高度定制的，缺乏对某些常见操作的抽象，唯一融合多数据源的操作为union，因此无法实现数据的直接连接。iView-

er 尝试寻找数据服务组合能力和用户操作代价之间的平衡，前者通过基本的组合操作算子实现，表达能力能够覆盖嵌套关系等语言，匹配和推荐方法以及用户反馈的学习能够有效地降低集成的代价。

## 7 结　论

本文在数据服务模型的基础上，提供了一种用户主导的数据服务组合方法，该方法可帮助用户按需集成互联网数据。本文以基于概率的模式匹配方法作为算子实现的基础，将该方法用于一个交互式的组合环境中，通过向用户不断推荐合适操作来完成组合过程。通过对用户反馈的学习、优化推荐内容和组合过程，以及案例、原型系统相关工作的比较分析，验证了该方法具有较好的方便实用性。

**参考文献：**


[1] DO H H, RAHM E. COMA. a system for flexible combination of schema matching approaches [C]// Proceedings of the Very Large Database Conference (VLDB). New York, USA: ACM, 2001: 610-621.

[2] NOY N F, MUSEN M A. The PROMPT suite: interactive tools for ontology merging and mapping [J]. International Journal of Human-Computer Studies, 2003, 59(6): 983-1024.

[3] WANG G, YANG S, HAN Y. Mashroom: end-user mashup programming using nested tables [C]// Proceedings of the 18th International Conference on World Wide Web. New York, USA: ACM, 2009: 861-870.

[4] 杨少华，林海略，韩燕波. 针对模板生成网页的一种数据自动抽取方法 [J]. 软件学报，2008，19(2)：209-223.
YANG Shaohua, LIN Hailue, HAN Yanbo. Automatic data extraction from template-generated web pages [J]. Journal of Software, 2008, 19(2): 209-223.

[5] KONGDENFHA W, BENATALLAH B. Rapid development of spreadsheet-based web mashups [C]// Proceedings of the 18th International Conference on World Wide Web. New York, USA: ACM, 2009: 851-860.

[6] MAGNANI M, RIZOPOULOS N, MCBRIEN P. et al. Schema integration based on uncertain semantic mappings [C]// Proceedings of the 24th International Conference on Conceptual Modeling. Berlin, Germany: Springer-Verlag, 2005: 31-46.

[7] JONES M C, CHURCHILL E F. Conversations in developer communities: a preliminary analysis of the Yahoo! pipes community [C]// Proceedings of the 4th International Conference on Communities and Technologies. New York, USA: ACM, 2009: 195-204

[8] LIU B, JAGADISH H. A spreadsheet algebra for a direct data manipulation query interface [C]// Proceedings of the Very Large Database Conference. New York, USA: ACM, 2009: 417-428.

[9] SARMA A D, DONG X, HALEVY A. Bootstrapping pay-as-you-go data integration systems [C]// Proceedings of the 28th International Conference on Management of Data. New York, USA: ACM, 2008: 861-874.

[10] IVES Z G, KNOBLOCK C A. Interactive data integration through smart copy & paste [C/OL]// The Biennial Conference on Innovative Data Systems Research (CIDR) 2009, Online Proceedings. (2009-03-12)[2012-06-12]. http://www.cidrdb.org/cidr2009/cidr2009zip.

[11] SHEN W, DEROSE P, MCCANN R, et al. Toward best-effort information extraction [C]// Proceedings of the 28th International Conference on Management of Database. New York, USA: ACM, 2008: 1031-1042.

[12] DONG X L, HALEVY A Y, YU C. Data integration with uncertainty [C]// Proceedings of the Very Large Database Conference. New York, USA: ACM, 2007: 687-698.

[13] ADYA A, BLAKELEY J A, MELNIK S, et al. Anatomy of the ado. net entity framework [C]// Proceedings of the 27th International Conference on Management of Data. New York, USA: ACM, 2007: 877-888.



**[本刊相关文献链接]**

伍文，孟相如，马志强，等. 模块化动态博弈的网络可生存性态势跟踪方法. 2012，46(12)：18-23.

吴俊峰，牟轩沁，张砚博. 一种快速迭代软阈值稀疏角 CT 重建算法. 2012，46(12)：24-29.

翟治年，奚建清，卢亚辉，等. 任务状态敏感的访问控制模型及其有色网仿真. 2012，46(12)：85-91.

李健，黄庆佳，刘一阳. 云计算环境下的大规模图状数据处理任务调度算法. 2012，46(12)：116-122.

丁维龙，韩燕波，王菁，等. 时间滑动窗口上数据流极值聚集的空间优化. 2012，46(11)：106-111.

杜立新，孟祥旭，刘士军. 基于服务提供者获益的服务响应选择方法. 2012，46(11)：112-119.

（编辑　管咏梅　赵大良）