VINCA ProData: A Data-Centric Application Integration Tool for Cloud Computing Environments (Postprint)

Authors: Wang Guiling, Liu Chen, Zhang Peng, Ji Guang, Xu Xuehui

Date: 2017-03-10T00:00:00+00:00

Abstract

Traditional service computing research often neglects data sharing and data flow. To better support data resource sharing and application collaboration during enterprise application integration system upgrades, this paper focuses on issues including data resource access in cloud computing environments, user-centric data resource integration, and data-driven business collaboration. The research findings can be applied to logical integration and information sharing of multi-information systems that span management domains and are based on cloud resource centers, as well as to the extraction and real-time presentation of characteristic data for comprehensive integration systems.

Full Text

Preamble

Vol. 9 No. 1
Information Technology Letter
VINCA ProData: A Data-Centric Application Integration Tool in Cloud Computing Environments
Guiling Wang, Chen Liu, Peng Zhang, Guang Ji, Xuehui Xu

Abstract

Traditional service computing research often neglects data sharing and data flow. To better support data resource sharing and application collaboration during the upgrade of enterprise application integration systems, this paper focuses on data resource access in cloud computing environments, user-centric data resource integration, and data-driven business collaboration. The research results can be applied to logical integration and information sharing of multi-information systems across management domains based on cloud resource centers, as well as to the extraction and real-time presentation of characteristic data from comprehensive integration systems.

Keywords: Enterprise Application Integration; Data Service Composition; Business Process Modeling

1 Introduction

The upgrade and evolution of traditional industry or enterprise application integration (EAI) systems in cloud computing environments presents a new challenge that requires leveraging private cloud and industry cloud concepts from the perspective of industry informatization. Application integration encompasses a broad range of technologies, which can generally be categorized into three levels based on integration objects: data integration, process integration, and interface integration [1]. Among these, data integration and process integration technologies are paramount. Since "business processes" and "data" constitute the core assets of enterprise information systems, business collaboration through process coordination—both intra- and inter-enterprise—and unified data views through data integration for querying and analytical functions play crucial roles in enterprise informatization. This is reflected in the product lines of major middleware vendors such as IBM and Oracle, where service-oriented business process middleware and service-oriented data integration middleware form core components.

However, current enterprise application integration systems face new challenges. Process integration tools and data integration tools remain disconnected, often confusing business users about when and how to use them, or which tool to choose [2]. Business personnel urgently need unified, organically integrated tools and programming models. In recent years, cloud computing has emerged as a typical distributed computing paradigm, with large-scale distributed applications exhibiting data-intensive characteristics. Examples include Google' s search engine based on Map-Reduce for large-scale data analysis, systems using Hadoop for enterprise data retrieval and analysis [3], and scientific workflow systems. Such applications process data dispersed across different departments and organizations in internet environments, with data volumes (or cumulative data from concurrent user requests) typically reaching terabyte (10^12 bytes) or even petabyte (10^15 bytes) scales. These applications serve massive user bases, with some requiring internet-based collaboration among users, making data sharing and data flow the primary concerns for integration. Although data and business processes hold equally important positions in enterprise application integration systems, traditional service computing research often neglects data sharing and data flow. This oversight leaves existing enterprise application integration systems ill-equipped to handle data-intensive applications in cloud computing environments that involve massive volumes (TB or PB scales), heterogeneous data formats (structured data, text, and semi-structured web data), and dynamically joining data sources.

Against this backdrop, this paper focuses on the following issues:

2 User-Centric Data Resource Integration

Data-intensive distributed applications in cloud computing environments present diverse user integration requirements. Providing methods and tools for data aggregation and analysis to meet these large-scale, diverse needs represents a significant challenge. Offering end-users simplified means to perform data resource integration themselves constitutes one solution path, termed "user-centric data resource integration." Furthermore, internet data changes dynamically, while traditional data integration methods—whether "materialization" or "virtualization" [4]—struggle to reflect data changes in real-time or support dynamic addition of new data sources at runtime. Therefore, this section emphasizes two related technologies for user-centric data resource integration: first, IT system data resource servitization technology, which addresses how heterogeneous data resources can be uniformly encapsulated and abstracted as services; and second, user-centric data service composition technology, which addresses how, after data resources are encapsulated as data services, users can dynamically integrate data resources by composing these services in a user-centric manner.

2.1 Data Resource Servitization Technology

IT system data resources exist in various types, including web pages, databases, and software modules. Among these, encapsulating software modules as services is most common, with numerous research achievements and software tools available. For instance, Apache' s open-source project Axis facilitates easy development of SOAP services based on Java objects. Whether servitizing web pages or databases, the essence lies in extracting relevant resource information according to client requests, converting this information into predefined message formats (such as SOAP messages used by SOAP services), and returning it as response messages to clients.

For HTML web pages and Web database resources on the internet, their poor structural characteristics make direct composition with other resources difficult. Therefore, mechanisms are needed to support end-users in encapsulating HTML pages and Web database resources—existing as services—into easily composable XML services, i.e., "web information resource servitization." Such services can be viewed as a specific form of wrapper [5] that extracts data from web pages and assembles it into XML. Extensive research on automatic and semi-automatic wrapper construction exists in the web information extraction domain, with detailed surveys available in [6,7]. Current research can be categorized into unsupervised learning methods and (semi-)supervised learning methods based on user involvement. Our work targets large numbers of ordinary users without programming knowledge, while existing wrapper construction methods inadequately consider end-user characteristics and personalized needs. Consequently, our challenge lies in establishing an efficient, robust, and highly adaptive web information resource servitization mechanism that allows users to fully express personalized requirements and complete resource servitization through simple "browse, select, configure" operations.

Internet resources also exist in other forms, such as Open APIs and Web services, which feature different interface description methods or lack explicit interface descriptions, with varying service invocation methods. Such heterogeneity creates significant obstacles for composing these differently-shaped services. Shielding differences between services and providing users with a consistent service model becomes necessary in certain contexts—a process typically termed "service unification." Industry has already produced multiple products for service unification, most notably Oracle' s AquaLogic Data Services Platform [8]. This platform' s primary function is wrapping various heterogeneous data resources as XML data services. Its implementation mechanism involves examining metadata from heterogeneous data sources (such as SQL metadata for relational databases or WSDL for Web services) to automatically generate one or more physical data services, enabling data source data to be expressed in XML format. For relational databases, each table or view generates a data service, with each row expressed as XML; for Web services, each operation' s return type generates a data service, with data operations implemented by the Web service operation; other data source types can form data services through similar mechanisms.

Besides AquaLogic Data Services Platform, WSO2' s Data Services Server supports similar functionality, publishing data services from relational databases, Excel spreadsheets, Google spreadsheets, and CSV files.

Despite these data resource servitization products, data service standards remain non-uniform—while vendors have established standards for their products, data services generated by different vendors' products remain difficult to interoperate.

3 User-Centric Data Service Composition Technology

User-centric data service composition can be broadly divided into three categories: visual languages, visual dataflow programming, and spreadsheet programming.

3.1 Visual Languages

Visual language approaches provide users with a visual operating environment that accepts various user operations (including dragging visual controls and keyboard input) to generate scripts for operating data resources, using these scripts to establish composite views of data resources. In visual language approaches, the design and generation of scripts for operating data resources constitute the core problem.

Typical examples include AquaLogic, Liquid Data [9], XQBE [10], and Xing

[11]. AquaLogic provides a workspace interface where users can select previously generated source data services, simultaneously construct a target data service, and establish data mapping by connecting source and target services, ultimately generating XQuery code to operate these data services. Similar to AquaLogic, Liquid Data differs primarily in allowing mixing of data services with XML documents (i.e., incorporating static data). However, if users manually edit the automatically generated XQuery code, the visual view cannot be reused, whereas AquaLogic supports bidirectional editing.

Visual language approaches suffer from two drawbacks: first, user operations are separated from their effects, making it difficult for users to see data changes resulting from their operations in real-time; second, various visual languages lack unified principles for representing data resources and operations, requiring users to learn each visual representation separately.

3.2 Visual Dataflow Programming

Visual dataflow programming approaches involve users constructing dataflows that originate from data services as sources, undergo a series of data processing and transformations, and ultimately converge in a data sink, thereby achieving data resource integration through dataflow construction. Both dataflows and data processing transformations are represented graphically for user comprehension.

Typical examples include Yahoo Pipes [12], IBM Damia [13], VINCA4Science [14], and Marmite [15]. Yahoo Pipes is an influential visual dataflow programming environment providing modules for data import, generic data operators, string operations, and numeric operations. Users can drag these modules in a workspace, connect them via pipes, and configure data processing details to implement complete data flows from RSS/ATOM/REST services through processing to final output. Compared with Yahoo Pipes, IBM Damia supports more data service forms and categorizes data operation modules into eight types: Augment, Merge, Filter, Sort, Group, Union, Transform, and Publish, centralizing diverse operations into these limited modules to enhance operational consistency.

Visual dataflow programming has two disadvantages: first, similar to visual languages, user operations remain separated from their effects, preventing real-time visibility of data changes; second, even for relatively simple data integration, involving many data resources or operations results in highly complex dataflows that impede normal usage.

3.3 Spreadsheet Programming

Spreadsheet programming approaches provide users with an Excel-like interface where data resources appear in tabular form. Users directly perform insert, delete, and update operations on tables, with complex operations decomposed into composable, reusable small operations that allow immediate observation of resulting data changes after each step.

Typical examples include SheetMusiq [16], AMICO [17], and SpreadATOR [18]. In SheetMusiq, researchers constructed a spreadsheet algebra on two-dimensional tables, designing a group algebraic operation to establish recursive grouping relationships on data for expressing complex data objects. Other algebraic operations support querying this grouped data, with expressive power equivalent to monolithic SQL statements. The approach also specifically supports query modification by changing previous operations to modify query results.

These approaches' advantage lies in direct table operations on data resources, enabling users to view operation results at any time—a significant improvement over visual languages and visual dataflow programming. Additionally, since spreadsheets represent familiar interfaces, users can operate without special learning, providing considerable convenience. However, limitations exist: two-dimensional tables struggle to represent complex nested data, and establishing complex expressive capabilities through 2D table operations can be difficult for users to understand. As shown in Figure 1: see original paper, SheetMusiq' s grouping operation results on a data table (with background grid added for clarity) demonstrate that 2D tables have inherent deficiencies in expressing complex data relationships without auxiliary mechanisms.

To address 2D table limitations, nested tables combined with spreadsheets have been proposed for data resource integration. This method adopts nested tables from nested relational algebra as the representation mechanism for data resources, establishing various data operations on nested tables to integrate data from different sources. Figure 1: see original paper shows the same data converted to nested table form, where grouping relationships become clearly visible. Compared with 2D tables, nested tables more easily express complex data operations (such as nested queries). As a data resource representation structure, nested tables combined with spreadsheets offer promising potential for creating user-friendly, highly expressive data resource integration methods.

```
ID  Model  Price Year  Mileage Condition

## 322  Civic  $16,000 2006  73,000  Good

(a) SheetMusiq  中的一个数据表例子 Model Condition Jetta Excellent Excellent Civic

## 2006 Good

Price  Mileage 872  $15,000  50,000 901  $16,000  40,000 304  $14,500  76,000 723  $17,500
```

Figure 1. Examples from SheetMusiq and nested tables

Based on this investigation and analysis, we find that spreadsheets provide intuitive operation modes, while nested tables effectively represent data objects that 2D tables struggle with. Therefore, spreadsheet programming, particularly the combination of nested tables and spreadsheets, offers a promising path for implementing user-centric data service composition.

4 Data-Driven Business Process Modeling Technology

Academic research on data-driven business process modeling has emerged in recent years. Traditional process collaboration modeling methods, such as Van der Aalst' s peer-to-peer (P2P) approach [19], can ensure that participating organizations' "private workflows" (workflows within individual autonomous agencies) satisfy "public workflow" (workflows involving multiple autonomous agencies) constraints. This method works well in environments with defined boundaries and relatively stable partners. However, since it predefines inter-process interaction relationships, modifications to the public workflow are required when requirements change or new organizations join, followed by re-constraining private workflows. This approach fails to eliminate dependencies between public and private workflows and cannot dynamically support partner joining and departure.

In contrast, works like CrossFlow [20] and DynaFlow [21] adopt bottom-up approaches. The process involves: first, publishing task nodes from private workflows that need to be externally provided to a public registry; second, matching partners in the registry and forming collaboration policies describing partners' roles and responsibilities; and finally, implementing inter-process collaboration through middleware support while monitoring the collaboration process. This approach offers advantages: (1) collaboration builds upon autonomous agencies' existing business processes, preserving stronger autonomy; (2) autonomous agencies can interact directly without relying on public workflows, providing greater flexibility to adapt to dynamic requirement changes.

Despite these advantages, this approach still suffers from dependencies on specific process modeling languages and software systems. With the emergence of Web services, the World Wide Web Consortium (W3C) proposed WS-CDL 1.0 [22] to support peer-to-peer collaboration. However, this method only considers activity-centric descriptions of inter-process dependencies and cannot support real-time monitoring of critical data during process execution.

In 2005, Van der Aalst proposed the Case Handling programming paradigm [23] supporting flexible business processes. This paradigm establishes data objects as basic workflow elements, providing data object modeling. A data object abstracts a set of composite information—entities with multiple properties or attributes, excluding single-value items like width. Data objects establish mappings to external data sources and present through forms. A "Case" represents a process instance, associating data objects with activities to allow viewing all data objects related to process instances during execution. Data objects can define Event-Condition-Action (ECA) rules to trigger Cases, supporting coordination between different Cases. Case Handling supports runtime monitoring of Case execution and all data object information, with Case status determined by all data object statuses. This work further supports data-centric process collaboration, but Case Handling cannot connect data object sources to Web information sources and lacks data analysis and complex event processing capabilities, limiting process collaboration effectiveness.

In cross-domain distributed environments, application collaboration often requires many sub-processes to cooperate, with process structures potentially changing at runtime. Manual modifications are error-prone and may cause deadlocks or delays due to incomplete dependency discovery by modifiers lacking deep process knowledge. To address this, D. Müller proposed the COREPRO method [24], supporting definition of complex data structures and providing data-driven automatic process structure transformation methods. This approach reduces manual process structure modification workload and ensures correct process collaboration. However, it only suits simple process collaboration, lacks procedural descriptions supporting complex processes like BPEL, and provides no business activity monitoring functionality.

Scholars represented by IBM' s R. Hull [25] further recognized the importance of combining data and processes in process collaboration, proposing Artifacts as basic building blocks for process collaboration. Artifacts are business-related conceptual entities, typical examples including purchase orders, invoices, shipping documents, insurance claims, and customer interaction histories. They presented an Artifact-centric business process modeling method that integrates heterogeneous business processes and supports cross-domain workflows. Artifacts require domain expert modeling, with information models capturing key data for business objectives and "macro-lifecycles" modeling public business process schemas. Two lifecycle description methods exist: state machine-based and declarative-based. Artifact-service association occurs either through procedural descriptions linking services to Artifact state transitions or through declarative business rules. At deployment, engines generate Business Operation Models (BOM) describing service composition results based on Artifact-service associations. At runtime, users can view real-time Key Performance Indicator (KPI) data changes and define trigger rules on KPIs to improve business processes or generate warnings. This method fundamentally changes process modeling approaches, solving cross-domain process heterogeneity issues, but requires high modeling expertise from domain experts to model Artifacts and establish Artifact-service associations.

These works provide respective methods or architectures for cross-domain process collaboration. The trend shows solutions evolving from centralized to decentralized orchestration, from activity-centric to data-centric, and from pre-orchestration to ad-hoc orchestration. Under this trend, researching a decentralized, data-driven, ad-hoc orchestration-supporting business process modeling approach is meaningful.

5 Our Work

The following illustrates our work and prototype system through an application scenario in emergency supplies management. Emergency supplies management encompasses the entire process from demand analysis, procurement, storage, transportation, distribution, and usage to consumption. Characteristics include: suddenness, uncertainty (duration, impact scope, intensity), strong time-effectiveness, and comprehensive participation (no single logistics center can achieve complete support alone; numerous logistics centers and enterprises must participate under government organization). We detail a scenario where an earthquake occurs in a remote mountainous area on a certain day in 2010.

Scenario: After the disaster, the local Red Cross is responsible for purchasing, collecting, managing, and distributing relief supplies such as tents, instant noodles, and water. Emergency supplies allocation involves a complex, multi-department collaborative business process. This design selects and simplifies a fragment to analyze business requirements and potential problems. The simplified scenario: After the disaster, the Red Cross Emergency Office commands the entire relief process. It first dispatches rescue departments to the disaster area to assess conditions, aid victims, and report required relief supplies based on disaster progress. The Emergency Office then notifies the disaster preparedness department to release relief supplies according to needs. Finally, the Emergency Office coordinates with the disaster area government’ s special agency to dispatch sufficient transport vehicles to deliver supplies.

The departments requiring collaboration and their roles are shown in :

Table 1. Disaster relief departments and their functions

Red Cross Emergency Office	Commands the entire relief process and coordinates with other departments
Red Cross Disaster Preparedness Department	Manages Red Cross warehouses and releases relief supplies according to needs
Red Cross Rescue Department	Dispatches rescue teams to aid victims and reports required supplies based on disaster progress
Disaster Area Government Special Agency	Dispatches transport vehicles to deliver relief supplies to the disaster area

As the commander, the Red Cross Emergency Office needs timely, comprehensive, and accurate disaster intelligence and supplies demand lists; must optimize limited resources for maximum value; and must monitor supplies flow status to ensure fastest, most reasonable distribution to victims most in need. To achieve these objectives, the Emergency Office must flexibly coordinate different departments’ business processes to accomplish timely, reasonable supplies allocation. However, due to the disaster’ s sudden, variable, and complex nature, the coordination process must address several challenges:

- To make correct supplies allocation decisions promptly, the Emergency Office must understand changes in key business data in real-time, much of which is hidden in other departments' internal business processes.

- The Emergency Office needs not only to obtain business data hidden behind different processes but also to rapidly aggregate and analyze this data for decision-making.
- The disaster information' s complexity and variability make it difficult to pre-build a complete business process that satisfies requirements. The Emergency Office often must determine next actions based on changing business data (e.g., latest disaster information, inventory status, vehicle allocation).

5.2 ProData: A Data-Centric Application Integration Tool in Cloud Computing

To address these challenges, we developed VINCA ProData, a data-centric application integration tool for cloud environments. It comprises two main components: a data integration component including data object modeling and KPI modeling tools plus data acquisition and processing modules; and a business process management component including business process modeling tools, execution engines, and monitoring tools. Using VINCA ProData to solve practical problems typically involves several key steps:

1. Data Object Modeling To support Red Cross commanders in viewing key data in real-time during process execution, we propose the data object concept, using nested tables as the basic structure. The scenario can establish three data objects:

`No Longitude Latitude No Type Required Quantity Available Quantity No Longitude Lat`

Figure 2. Data object models

Rescue teams belong to the Red Cross Rescue Department, warehouses to the Red Cross Disaster Preparedness Department, and transport vehicles to the local government special agency. Since these data objects' sources may come from databases or files distributed across different departments, data acquisition and processing are required. Data object modeling tools are primarily used by IT personnel from each department, employing a "two-sided definition + middle convergence" approach: the two sides refer to business-domain-oriented data objects and IT-domain-oriented distributed data sources, while middle convergence refers to data mapping and transformation.

2. Key Performance Indicator Modeling To enable Red Cross commanders to view their concerned KPIs in real-time, we propose KPI modeling, where KPIs are set or numeric calculation formulas using data objects as operands, computed in a spreadsheet-like programming environment and presented through graphs or charts. This scenario includes the following KPIs:

- Count(RescueTeam.Supplies(Tents, Instant Noodles, Water).RequiredQuantity)
- Count(RescueTeam.Supplies(Tents, Instant Noodles, Water).AvailableQuantity)

- Count(RescueTeam)

3. Business Rule Definition To capture KPI data changes in real-time and support dynamic collaboration across departmental processes, we define event-triggered process plans through business rules. These rules are derived through event algebra composition operators and can be automatically converted into Event Processing Language (EPL) to capture KPI changes in real-time. shows partial business rules for this scenario.

Table 2. Business rules and triggered process plans

Rule ID | Rule Content | Triggered Process Plan |
RescueTeam.Event.Longitude>0 && RescueTeam.Event.Latitude>0 → Level 1 Rescue Process Plan |
RescueTeam.Event.Longitude>0 && RescueTeam.Event.Latitude>0 → Level 2 Rescue Process Plan |
RescueTeam.Supplies(Tents, Instant Noodles, Water).RequiredQuantity > RescueTeam.Supplies(Tents, Instant Noodles, Water).AvailableQuantity → Warehouse A Release Process Plan |

4. Process Orchestration Business personnel from each department orchestrate business processes. In this scenario, processes from three departments are involved: Red Cross Rescue Department, Red Cross Disaster Preparedness Department, and Disaster Area Government Special Agency. Since the local government understands road conditions, the transportation process is handled by the special agency organized by the disaster area government. These three departments operate relatively independently, each handling its own business logic while performing decentralized, uncontrolled dynamic collaboration during emergency supplies management to jointly complete supplies allocation and distribution.

5. Ad-hoc Decision Making When KPI changes satisfy business rule conditions, corresponding events are captured by the Red Cross decision center. Commanders analyze these events combined with data object information to make ad-hoc decisions about which process plan to execute from the business rule 预案, as shown in [Figure 3: see original paper]. For example, after rescue teams arrive at the site requiring substantial supplies, Red Cross commanders may choose to execute the Warehouse A Release Process Plan based on inventory levels at Warehouses A and B. When dispatching transport teams, they can use each team' s location information to distribute supplies fastest and most reasonably to victims most in need.

Figure 3. Dynamic collaboration of business processes

6 Conclusion

In recent years, with cloud computing' s rise, traditional enterprise application integration systems face upgrade challenges. Since traditional service computing research often neglects data sharing and data flow, existing enterprise application integration systems encounter numerous challenges when handling data-intensive applications in cloud computing environments. This paper analyzes two major issues in data integration and business process modeling for enterprise application integration systems in cloud environments, focusing on technologies for data resource servitization, user-centric data service composition, and data-driven business process modeling. Finally, using an emergency supplies management scenario, it introduces VINCA ProData, a data-centric application integration tool for cloud computing environments. Future work will further investigate scientific issues in data service modeling, data service composition, process modeling, and optimization for data-centric application integration in cloud computing environments.

References

[1] Yanbo Han, Guiling Wang, Chen Liu, Jing Wang, Zhuofeng Zhao. *Principles and Practice of Internet Computing—Exploring the Essential Issues and Key Technologies Behind Grid, Cloud, and Web X.0.* Science Press, 2010.7-

[2] Halevy, A. Y., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., Rosenthal, A. and Sikka, V. (2005) Enterprise information integration: successes, challenges and controversies. In SIGMOD ' 05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 778-787. ACM.

[3] K.Beyer, V.Ercegovac,R.Krishnamurthy. Towards a Scalable Enterprise Content Analytics Platform. http://sites.computer.org/debull/A09mar/sandeep.pdf

[4] Richard Hull and Gang Zhou, A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches, pp. 481–492, 1996.

[5] J. Hammer, H. Garcia-Molina, S. Nestorov, etc. Template-Based Wrappers in the TSIMMIS System, In Proc. of the International Conference on Management of Data (SIGMOD), Tucson, Arizona, USA, 1997, 532-535

[6] A.H.F. Laender, B.A. Ribeiro-Neto, etc. A brief survey of Web data extraction tools, SIGMOD Record, 2002, 31(2): 84-93

[7] C. Chang, M. Kayed, M. R. Girgis, etc. A Survey of Web Information Extraction Systems, IEEE Trans. Knowl. Data Eng, 2006, 18(10): 1411-1428.

[8] Borkar, Vinayak, Michael Carey, Sebu Koleth, Alex Kotopoulis, Kautul Mehta, Joshua Spiegel, Sachin Thatte, and Till Westmann. “Graphical Xquery in the Aqualogic Data Services Platform.” Paper presented at the SIGMOD ' 10: Proceedings of the 2010 international conference on Management of data 2010.

[9] N. Mangtani and M. Carey. Liquid Data: XQuery-based enterprise information integration. In BEA WebLogic Developer' s Journal, April 2003.

[10] Daniele Braga, Alessandro Campi, Stefano Ceri, XQBE(XQuery By Example): A visual interface to the standard XML query language, ACM Transactions on Database Systems (TODS), v.30 n.2, p.398-443, June 2005

[11] Martin Erwig, A Visual Language for XML, Proceedings of the 2000 IEEE International Symposium on Visual Languages (VL' 00), p.47, September 10-13, 2000

[12] Yahoo Pipes, Inc. http://pipes.yahoo.com/, 2010.

[13] Simmen, D. E.; Altinel, M.; Markl, V.; Padmanabhan, S.; Singh, A. Damia: data mashups for intranet applications, SIGMOD ' 08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, 2008, pp. 1171-1182.

[14] Jing Wang; Yanbo Han; Shuying Yan; Wanghu Chen; Guang Ji; "VINCA4Science: A Personal Workflow System for e-Science," Internet Computing in Science and Engineering, 2008. ICICSE ' 08. International Conference on, vol., no., pp.444-451, 28-29 Jan. 2008

[15] J. Wong and J. I. Hong, Making mashups with marmite: towards end-user programming for the web, In Proc. of the 2007 conference on Human factors in computing systems (CHI), San Jose, California, USA, 2007, pp. 1435-1444.

[16] B. Liu and H. Jagadish. A spreadsheet algebra for a direct data manipulation query interface. In Proceedings of the 35th international conference on Very large data bases. 2009. pp. 417-428.

[17] Ž. Obrenović and D. Gašević. End-User Service Computing: Spreadsheets as a Service Composition Tool. IEEE Transactions on Services Computing. 2008, 1(4). pp. 229-242.

(Continued on page 7)

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.