Data Description and Data Access Mechanism in Distributed Workflow System

(Work-in-Progress)

Wu Hongli, Yin Baolin, Zhao Xia, Xiang Gang
School of Computer Science & Engineering, Beihang University, Beijing 100083, China
E-mail:honglwu@sohu.com

ABSTRACT

Aiming at the problem that the existing workflow management systems are not effective in describing and establishing data flow, conceptions of interior data flow and exterior data flow are introduced in this paper. Based on analyzing characteristics of data flows, a new architecture of workflow data management in distributed environments is proposed. In the proposed architecture the runtime nodes in runtime level are responsible for control flow, and the data nodes in data level are responsible for data flow. The execution of receiver activity on runtime node triggers the data replication mechanism of the data level. Then the required data flow between runtime nodes can be established. Compared with the existing methods of data flow implementation, the proposed method of data description and data access mechanism provide data transmission for the exterior data, and convenient the description and implementation for the interior data. The proposed method has value in designing data management in workflow management system.

Categories and Subject Descriptors

H.2.6 [**Database Management**]: Database Machines; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – Query Formulation

General Terms

Management, Design

Key words

data flow; control flow; workflow data; data description; data access mechanism

1. INTRODUCTION

In existing Workflow Management Systems (WfMSs),Many scholars have paid attention to workflow data managements from different points of view, such as special data management [1,2], data delivery [3], data distribution [4], data ETL process optimization [5,6], data consistency [7] and data flow management [8,9], and have solved corresponding problems. The common of these methods is that data flow direction is consistent with control flow. After a large number of business processes have being analyzed from the perspective of data flow, we find data flow isn't always consistent with control flow. These data flows inconsistent with control flow and the data transmitted by these data flows have not gotten enough attention. Manual method or assistant software has to be used to implement these data flows and data delivery. From the overall view of WfMS, neither manual method nor assistant software has taken these data flows management as part of WfMS, which against the idea of automatically cooperating resources, people and tools distributed over a wide geographic. Based on an extensible organized flexible workflow model, this paper will solve the problem of establishing the data flow which is inconsistent with control flow.

Conference name: Infoscale 2007, June 6-8, 2007, Suzhou, China
Copyright number (LaTeX \crdata{}): 978-1-59593-757-5

2. TYPES OF DATA FLOW

According to the direction of data flow and control flow and the position of provider and receiver activities, data flow can be classified into coincident data flow, striding data flow, converse data flow and inter-process data flow, as shown in Figure 1.

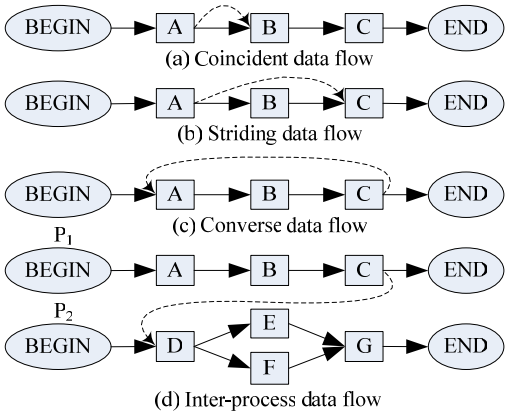

Figure 1: Four types of data flow in workflow

(1) Coincident Data Flow

Coincident data flow is consistent with control flow, and the provider and receiver activities are exactly the same as those of the control flow, as shown in Figure 1(a).

(2) Striding Data Flow

Striding data flow is also consistent with control flow, except that it spans several control flows, and the provider and receiver activities are different from those of control flows, as shown in Figure 1(b).

(3) Converse Data Flow

Converse data flow conflicts with control flow in the same business process, as shown in Figure 1(c).

(4) Inter-process Data Flow

Inter-process data flow emerges among different business processes, as shown in Figure 1(d).

When the data transmitted by a data flow can be transmitted by control flow, the data flow is called **interior data flow**. Coincident data flow and striding data flow belong to interior data flow. When the data transmitted by a data flow can not be transmitted by any control flow, the data flow is called **exterior data flow**. Converse data flow and inter-process data flow belong to exterior data flow. The data transmitted by interior data flow is call **interior data**. The data transmitted by exterior data flow is call **exterior data**.

3. WORKFLOW DATA MANAGEMENT

According to the realization of data flow, the existing distributed WfMSs can be classified into WfMS using control flow and WfMS using special database [2,8,9]. These systems are appropriate for interior data flow, but inappropriate for exterior data flow. The fundamental reason is that the existing systems don't separate the data flow management from the control flow management completely and don't offer appropriate methods for data description and data access mechanism. So a new architecture of workflow data management in distributed environment is proposed in this section. In this architecture the runtime nodes in runtime level are responsible for control flow, and the data nodes in data level are responsible for data flow. Execution of the receiver activity triggers the underlying data duplication mechanism in data level so that the data flow between runtime nodes is established.

3.1. Architecture of Workflow Data Management

The architecture of workflow data management consists of build-time level, runtime level and Data level. Build-time level is responsible for the definition of workflow; runtime level is responsible for control flow; and data level is responsible for data flow.

Build-time level is composed of a group of workflow build-time nodes. Every **build-time node** which contains a group of workflow build-time tools is responsible for defining business process within the scope of jurisdiction, and coordinates with other build-time nodes.

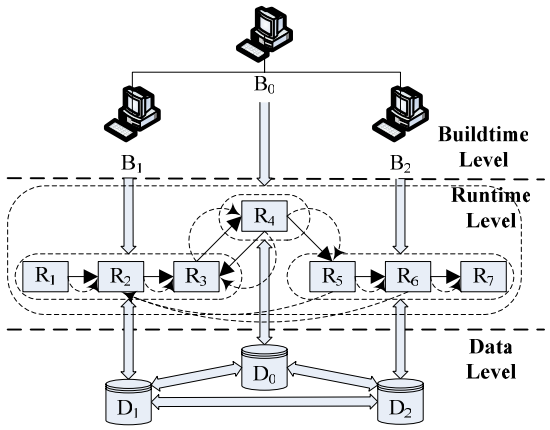

Figure 2: Architecture of workflow data management in distributed environment

Runtime level is composed of workflow runtime nodes. Every **runtime node** containing a workflow engine is responsible for the execution of activities assigned to the node, and coordinates with other runtime nodes. All runtime nodes and build-time nodes form a tree structure in which runtime node is the leaf node and build-time node is the branch node. For each runtime node has a build-time node as its parent node, the build-time node is called **local build-time node** of the runtime node. As shown in Figure 2, the local build-time node of runtime nodes R_1, R_2 and R_3 is B_1, the one of runtime node R_4 is B_0, and the one of R_5, R_6 and R_7 is B_2. The jurisdiction of a build-time node is all the nodes in its sub-tree. A runtime node can only receive workflow definition from its local build-time node and the ancestor build-time nodes. For example, the runtime node R_3 can only receive from build-time nodes B_1 and B_0.

Data level is composed of a group of workflow data nodes. When a build-time node has runtime nodes as child nodes, there is a data node related to the build-time node. Every **data node** maintains workflow data provided by runtime nodes within the scope of jurisdiction, establishes data communication with these runtime nodes, and exchanges required data with other data nodes. The data node is called **local data node** of the build-time node and those runtime nodes. As shown in Figure 2, data node D_0 is the local data node of build-time node B_0 and is responsible for the workflow data provided by runtime node R_4. D_1 is the one of B_1, responsible for the data provided by R_1, R_2 and R_3. And D_2 is the one of B_2, responsible for the data provided by R_5, R_6 and R_7.

In this architecture, runtime node is in charge of control navigation, and data node is in charge of data access. Every runtime node can only access data from its local data node. When providing workflow data, the provider activity commits data to its local data node. When requesting workflow data, the receiver activity receives data from its local data node. If the required data is not maintained at local data node, the data duplication mechanism of data level will be triggered.

3.2. Data Description

In the existing systems, workflow data description is mainly about the interior data [1,2,8,9], lack of the exterior data description, so these systems hardly support exterior data flow and exterior data management. In this section a new data description method will be put forward. Three basic conceptions of *DataStub*, *DataTable* and *DataFile* on build-time node and data node are given.

***DataStub*:** *DataStub* records describing information of providing data defined or maintained by build-time or data node. Its formal description is:

$$DataStub = \{s \mid s = (Did, e, a)\} \quad (1)$$

DataStub is a set of the stub *s*. *s* is a triple group of *Did*, *e* and *a*. *Did* denotes data identifier which can uniquely identify *s*. *e* denotes existing state. *a* denotes the data location. When e=0 which denoting that the data doesn't exist in this node, *a* denotes the address of the data node in which the data is maintained. When e=1 which denoting that the data exist in this node, *a* denotes the location of *DataTable* in this node.

***DataTable*:** *DataTable* records value of data items produced by provider activity. Its formal description is:

$$DataTable = (Did, R) = (Did, \{r \mid r = (Tid, v_1, v_2, \cdots, v_n)\} \quad (2)$$

DataTable is a binary group of *Did* and *R*. *R* is a set of *r*. *r* is a n+1 group of *Tid*, v_1, v_2, …, and v_n. *Tid* denotes a process instance identifier which can uniquely identify *r*. v_i ($1 \le i \le n$) denotes a data item of *r*, and is also the data item produced by provider activity. One *DataTable* relates to one *s* in *DataStub*.

***DataFile*:** *DataFile* is a file that is formed by data items, which are provided by provider activity, based on a data template. Its formal description is:

$$DataFile = (Did, Tid, RD)$$

$$=(Did, Tid, \{(v_i, d_i) \mid v_i \in (v_1, v_2, \cdots, v_n)(1 \le i \le n),$$

$$d_i \text{ is the display form of } v_i\}) \quad (3)$$

DataFile is triples group of *Did*, *Tid* and *RD*. *RD* is a set of data items and their display form. d_i is the display form of v_i. One *DataFile* related to one *r* in *DataTable*.

3.3. Data Access Mechanism

In Figure 2, for each data node only maintains the data provided by the runtime nodes within the scope of jurisdiction, how to get the location of the required data and how to access the data must be supported. Stub establishment mechanism and data

duplication mechanism discussed this section will solve the problem of data location and data access respectively.

3.3.1. Stub Establishment Mechanism

Stub mechanism is used to solve the problem of data location. When defining a business process, a build-time node also defines the data that is provided or received by an activity. When issuing the definition, the build-time node should do the followed work: (1) the providing information of interior data is issued to the local data node, and the data node inserts a new stub in its *DataStub*; (2) the providing information of exterior data is issued to the local data node, and the data node inserts a new stub; (3) the providing information of exterior data is issued to all build-time nodes within the sub-tree, and these build-time nodes insert a new stub in its *DataStub*; (4) the providing information of exterior data is issued to those local data nodes which are related to all build-time nodes within the sub-tree, and these data nodes insert a new stub in its *DataStub*.

3.3.2. Data Duplication Mechanism

Data duplication mechanism is used to solve the problem of data access. It provides file duplication and data item duplication.

(1) File Duplication

File duplication duplicate the *DataFiles* from the data node which contains the requested data to local date node. File duplication includes: **Single-file duplication** duplicates a DataFile whose primary key satisfies certain condition. It is applied in interior data flow and exterior data flow; **Same-type multi-files duplication** duplicates some *DataFiles* from the same *DataTable* whose general data items satisfy certain condition. It is applied in exterior data flow; **Multi-types multi-files duplication** duplicates some *DataFiles* from different *DataTables* whose general data items satisfy certain condition. It is applied in exterior data flow.

(2) Data Item Duplication

Data item duplication duplicates data items in *DataTable* from the data node which contains the requested data to local date node. Data item duplication includes: **Single-record whole data items duplication** duplicates whole data items of the record whose primary key satisfies certain condition. It is applied in interior data flow and exterior data flow; **Same-type multi-records whole data items duplication** duplicates whole data items of some records from the same *DataTable* whose general data items satisfy certain condition. It is applied in exterior data flow; **Multi-types multi-records whole data items duplication** duplicates whole data items of some records from different *DataTables* whose general data items satisfy certain condition. It is applied in exterior data flow; **Single-record part data items duplication** duplicates some data items of the record whose primary key satisfies certain condition. It is applied in interior data flow and exterior data flow; **Same-type multi-records part data items duplication** duplicates some data items of some records from the same *DataTable* whose general data items satisfy certain condition. It is applied in exterior data flow; **Multi-types multi-records part data items duplication** duplicates some data items of some records from different *DataTables* whose general data items satisfy certain condition. It is applied in exterior data flow.

4. CONCLUSION

Aiming at the problem that the existing workflow management systems are not effective in describing and establishing data flow, the architecture of workflow data management in distributed environment is proposed. Compared with the existing WfMSs, the proposed method gains much more benefits: (1) **Establishing the data channel for exterior data flow**. The runtime node using the data description and data access mechanism proposed in this paper triggers the data duplication mechanism of data level when it receives data. (2) **The description of interior data flow is more convenient**. The proposed method does not need describing data on every control flow that the data had to pass through, only needs describing data on provider activity and receiver activity. (3) **The receiving mode of data is diversiform and extensible**. Three modes of file duplication and six modes of data items duplication are put forward in this paper. The receiver activity can assemble many modes of data receiving. By adding a new data description of the receiving data and a new mode of data duplication in data level, the proposed receiving mode of data can be extended. (4) **The logic level of the system is more clearly**. Separating data flow from control flow reduces the affect on runtime level imposed by data level.

5. REFERENCES

[1] Zhu Li, Yin Jianwei, Chen Gang, Dong Jinxiang, Wang Bingbing. Research on multi-libraries based collaborative document management system. *Computer Integrated Manufacturing Systems*. 2006, 12(3), 440~445

[2] Hyerim Bae, Wonchang Hu, Woo Sik Yoo, Byeong Kwon Kwak, Yeongho Kim, Yong-Tae Park. Document configuration control processes captured in a workflow. *Computers in Industry*. 2004, 53, 117~131.

[3] Han Zongfen, He Kang, Zhang Qin, Shi Xuanhua. A strategy of data transferring of grid workflow based on weighted directed graph. *J. Huazhong Univ. of Sci. & Tech. (Nature Science Edition)*. 2005, 33(12), 112~114

[4] Schuster H., Heinl P.. A Workflow Data Distribution Strategy for Scalable Workflow Management Systems. *Proceedings of ACM Symposium on Applied Computing (SAC'97)*. 1997. 174 -176

[5] Simitsis A., Vassiliadis P., Sellis T. State-Space Optimization of ETL Workflows. *IEEE Transactions on Knowledge and Data Engineering*. 2005, 17(10). 1404 – 1419

[6] Tan Zhipeng, Feng Dan, Wu Yongying, Peng Feng. Workflow-based extraction and transformation-loading of data. *J. Huazhong Univ. of Sci. & Tech. (Nature Science Edition)*. 2006, 34(2). 61~69

[7] Luo Haibin, Fan Yushun, Wu Cheng. A Consistency Protecting Framework for Workflow Data. *Computer Integrated Manufacturing Systems.* 2002, 8(4). 320~325

[8] Sadiq S., Orlowska M., Sadiq W., Foulger C.. Data flow and validation in workflow modelling. *Proceedings of the Conferences in Research and Practice in Information Technology*, Dunedin, New Zealand. 2004. 207-214.

[9] Alonso G., Reinwald B., Mohan C.. Distributed data management in workflow environments. *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE'97)*, Birmingham, England, 1997. 82-90.