Automatic Building of ER and Data Flow Graph: A Business Process-based Approach

Yanhong Zhao, Hongqi Li, Liping Zhu, Jiandong Wang

Beijing Key Lab of Data Mining for Petroleum Data, China University of Petroleum, Beijing, China

Abstract

Nowadays, many tools are available for drawing entity relationship(ER) diagram and data flow graph, but the graph layout is manually designed and the drawing process is tedious and time-consuming. Moreover, the management of business model is independent of the data model, the inconsistency is frequently occurred. This paper proposes an entity-based process model called Entity-Process Model (EP-Model) to build up the linkage between the business model and data model. Based on the EP-Model, this paper gives the construction algorithms of ER diagrams and data flow graph. Finally, a modeling tool is developed for business process modeling and data modeling. Using this platform, the consistency between business model and data model is guaranteed, and the ER diagram and data flow graph can be built automatically. It plays an important role in the process modeling, data modeling and the integration of information systems in the future.

Key words: Business Model, Entity Relationship Diagram, Data Flow Graph, Modeling Tool.

1. INTRODUCTION

Traditional application system is built through data-driving mode and mainly concerning about the role of business data. As the start in building application system, data model consists of three parts during its establishment phase, Concept Modeling, Logic Modeling and Physical Modeling. Concept Modeling which is the basis of data modeling needs to clear the relationship between entities and the processing procedures of entities while entity relationship diagram and data flow graph are more popular owing to the intuition. There are many modeling tools supporting the drawing of ER diagram and data flow graph but weak in graphical layout function, also most of them are mainly rely on manual layout modelers with a rather complex procedure. Sybase's case toolset power designer provides the reverse engineering capabilities which can automatically generated the ER diagram from the data model under the condition of existed data model. The literatures (Zhao.Hou andWen,2012;Zeng andHe, 1995) proposed some graph layout algorithms for automatic drawing of ER diagram and data flow graph. However the relationship between entities are sorted by modelers under the foundation of understanding about the business context. While in a complex system, the carding of the relationship between entities will be a time-consuming and laborious work.

With the development of process management methodology, process-driven approach to build application systems have been widely used and business model has become the starting point for applications built. Business model is the prerequisites of the data model building while the data model is the refining and derivation of business model and provide data support for business model, the two complement each other. But in the practical building process of application system, business model and data model is independent and managed separately so that it is difficult to ensure the consistency of data models and business models. In order to ensure the consistency, we need to establish the relationship between data model and business model. Literature(Russell, Hofstede, Edmond and Aalst, 2005) describes the data patterns in workflow, such as data visibility patterns, data interaction patterns, data transfer patterns, data-based routing, intending to obtain the various ways of data representation and data using in workflow. Although there are many differences between workflow and business process, they still follow the main data patterns(Xin and Rong, 2014). Literature(Liu, Li,Wang and Qu, 2005) proposed DP-Graph model to describe the relationship between business models and data models, examining exceptions in data models according to the business models. However, DP-Graph only can be used under the condition of the existence of data models and business models, so it belongs to the later verification of this model and is unable to ensure the consistency of business models and data models in the modeling process.

With the support of modeling tools, users can realize the drawing of ER diagram and data flow diagram, also most of them are mainly rely on manual layout with a rather complex procedure. There are many universal layout algorithms and sophisticated graphics software supported automatic layout. Common methods of graph layout include Orthogonal layout algorithm (Battista, Didimo, Patrignani and Pizzonia, 1999), force-directed layout (Holten and Wijk, 2009;Fruchterman and Reingold, 1991), a hierarchical layout algorithm (Bastert and Matuszewski,2001), the circular layout (Baur and Brandes, 2007) and so on. The selection of graph layout
algorithms depends on users’ requests. Design graph layout algorithm needs to understand the graph theory, graph algorithms, geometry, topology, information visualization, interactive graphics, etc. otherwise it would be difficult to achieve the breakthrough and innovation. The point of this article is not the design and innovation of graph layout algorithms, so we realize automatic graphic layout by means of a third-party’s algorithm libraries, such as jGraph(Bagga and Heintz, 2001), Graphviz (Ellson, Gansner and Koutsofios,2004) and so on. Graphviz is a open source tools package from AT&T Lab, use the DOT as the script language and then parse the script by the layout engine to complete the automatic layout. The graphics can be exported in GIF, PDF, SVG, PNG and other forms. As a graphic description language in text (North, 1993), DOT can be used to describe undirected and directed graphs, also it can be understood by humans and computer programs. Graphviz contains numerous placer, such as dot, fdp, circo, neato (Riesco, Fondón and Alvarez, 2009), twopi. With the help of powerful graphics descriptive capabilities of DOT and Perfect layout management abilities of Graphviz, users can draw finite state machine, ER diagram, Petri nets, data structures, flow charts and so on. Our contributions are three-fold as follows. Firstly, we gives out the formal definition of business model named EP-Model. The business model and data model are integrated in the same framework to ensure the consistence, providing business related data to customers. Secondly, we offer the automatic construction algorithms of ER diagram and data flow graph. Thirdly, we develop a platform for business process modeling and data modeling.

2. ENTITY-BASED PROCESS MODEL

Business model and data model is an important achievement in requirements modeling phase of application system, also it can help designers understand the specific business processes and processing objects, serving as an important role in guiding software development and related technology research. In the data modeling process, ER diagrams and data flow diagrams which belong to conceptual modeling stage, are the basis for logical modeling and physical modeling. ER diagram depicts the links between the entities, data flow diagrams graphically show the transfer process of data flow from input to output. Entities and data do not exist in isolation, they generate links with the operation of the business processes, constantly changing. The creation of data flow diagram and ER diagram depend on the business model. However, in the actual building process of application system, business model, ER diagram and data flow diagram are independent and managed separately, this will lead to inconsistency between business models and data models. Therefore, we must put forward a unified description model to describe the relationship between business model and data model. Business model and data model should be studied in the same framework to ensure the consistency of the data model and business model throughout the modeling process. Meanwhile, according to the model, users can automatically extract the contact and data flows between the entities, laying a good foundation for the automatic generation of ER diagrams and the data flow diagrams.

Business model describes the flow processes of business within the enterprise, also it divides the business into a series of activity set to achieve the conversion from input to output through mutual cooperation between the activities. This paper proposed Entity-based Process Model (EP-Model) to describe the business model so as to achieve the association between business models and data models. EP-Model consists of a set of activities set, transfer set and route set according to definition 1.

Definition 1: Business Model(EP-Model) = {A, Rt; T}. It is a directed network, A and Rt are the sets of nodes, T is the set of arcs. A denote the set of business process activities and Rt is a collection of routing nodes while T describes the logic timing relationship between the two activities.

A = {A₁, A₂, …, Aᵢ} (m>=1), Aᵢ is a business activity.

Rt= [Rt₁, Rt₂, …, Rtᵢ], Rtᵢ represent one of the routing node in business process, route represent the business process in control mode. There are four kinds of structure: serial routing, parallel routing, selection routing and loop routing (Aalst, Hee, 2004). In this article, we take the implicit realization in case of the order route, ignoring routing node. Routing nodes have four types. Andsplit represents the concurrent split, andjoin represents the concurrent aggregation, orsplit represents the radio division, orjoin represent radio convergence.

T = {T₁, T₂, …, Tᵢ} ⊆ A × A U A × Rt U Rt × A U Rt × Rt (n >= 1), which denotes predecessor or successor relationships between two nodes. < A₁, A₂ >= T shows that the successor activity of A₁ is A₂ and belongs to sequential relationships in most cases. < Aᵢ, Rtᵢ, >= T represents the successor node of activity Aᵢ, is a routing node Rtᵢ, corresponding to the convergence mode of andjoin or orjoin. < Rtᵢ, Aᵢ >= T denotes the successor node of routing node Rtᵢ, is activity Aᵢ generally expressing the split mode of andsplit or orsplit. < Rtᵢ, Rtᵢ, >= T represents the successor node of routing node Rtᵢ, is also a routing node, this situation occurs mainly in radio division mode.

Definition 2: Let Aᵢ is a business activity, Aᵢ = {Attrs, Roles, Entities, Relations}, in which "Attrs" is the set of attributes, "Roles" is a set of roles and "Entities" is the entity set while "Relations" is the relation set of between activities and entities or activities and roles. This paper suspects that the business activity is the
smallest execution unit in business process, so that you can generate sub-processes through the combination of activities.

Attrs=\{activityId, domainId, baseActivityId, activity Name, description \}, "domainId" represents the business process id which the business activity belongs to. Because the business process is configurable, a business activity can exist in different business processes, marking with the property of "domainId". If the activity is the reuse or derivation of another activity, we save the id of another activity in the property of "baseActivityId".

Roles=\{ R_1, R_2, \ldots, R_n \} (n=1). R_i is one of the role in business activity A_n there could be many roles in one business activity. Role R_i=\{roleId, roleName, domainId, description \}. Roles are managed according to the domain, while in this paper with the restriction that the role name should be unique in the same domain. After the process instanced, the business activities are assigned to the person designated with roles through the processing engine.

Entities=\{ E_1,E_2,\ldots,E_n \} (n=0). Entity set defines the conceptual object associated with the business activity, in this paper mainly refer to the data entities, that is the data information model related to business activities. Entity E_i=\{entityName, domainId, attributeSet, description, etc.\}. In this paper, the business entity name is unique under the same domain; Attribute sets = \{attributeName, attributeType, isEmpty, etc.\}. During the modeling phase of data concept, we only focus on the entity. During the logical modeling phase of data, entities are elaborated through property collection. During the physical modeling phase of data, well defined entity object can be converted into SQL code to create the database objects. Business model and data model related with each other through the definition of business implementation.

\[ \begin{align*}
\text{Relations} &= \text{RA} \cup \text{EA} \cup \text{AE}, \\
\text{RA} &= \{ R \times A \mid R \subseteq \text{Roles} \} \times A,
\end{align*} \]

*Figure 1.* The relationship between business activities and roles or entities

Relations=\text{RA} \cup \text{EA} \cup \text{AE}, consists of three subsets, as shown in Figure 1. \text{RA} \subseteq \text{Roles} \times A = \{ R_i, R_2, \ldots, R_n \} \times A_i represents the relation between roles and business activity A_i, while \langle R_i, A_i \rangle \in \text{RA} represents role R_i take part in the activity A_i. \text{EA} \subseteq \text{Entities} \times A = \{ E_1, E_2, \ldots, E_n \} \times A denotes the inputting relationship between entities and business activities, \langle E_i, A_i \rangle \in \text{EA} denotes the business activity A_i use the entity E_i. This paper use "read" to express the relation between ordered pairs \langle E_i, A_i \rangle . \text{AE} \subseteq A \times \text{Entities} = A \times \{ E_1, E_2, \ldots, E_n \} denotes the outputting relationship between entities and business activities, \langle A_i, E_i \rangle \in \text{AE} represents the business activity A_i create or update the entity E_i, so this paper use "create" and "update" to represent the relation between ordered pairs \langle A_i, E_i \rangle.

For \forall A_i \in A, \forall E_i \in E, \text{EA}_i denotes the input set of A_i, \text{AE}_i denotes the output set of A_i, \text{Op}<A_i,E_i> denotes the operation relations between business activities and entities, we define the following rules to identify the operation type.

\begin{align*}
\text{if} \ E_i \in \text{EA}_i \text{ and } E_j \notin \text{AE}_i, \text{ then } \text{Op}<A_i,E_j> &= \text{read} \\
\text{if} \ E_j \notin \text{EA}_i \text{ and } E_i \in \text{AE}_i, \text{ then } \text{Op}<A_i,E_j> &= \text{create} \\
\text{if} \ E_i \cap \text{AE}_i \neq \emptyset \text{ and } E_j \in \langle \text{EA}_i \cap \text{AE}_i \rangle, \text{ then } \text{Op}<A_i,E_j> &= \text{update}
\end{align*}

3. CONSTRUCTION ALGORITHMS OF ER DIAGRAMS AND DATA FLOW GRAPH

Section 2 gives the formal definition of EP-Model, the relationship between business activities and the business activities are represented by the ordered pairs abstractly. Through the analysis of ordered pairs, it is possible to obtain the connection between entities and business activities, along with the logic flow and transformation procedure in the business process. Entity relationship model (ER diagram) is a conceptual model that can help users with database design. ER model mainly consists of entity set, attribute set and link set. In this
paper, relation set are equal with business activity set. An entity may participate in multiple business processes, also a business process can have a number of different entities. Therefore, the study of the entity relationship include two cases: one is the ER model under the background of single business that is choosing the specific business processes to build the related ER model in the business model. The specific construction algorithm are shown in algorithm 1. The other is the ER model of multi-business background, which means choosing an entity field to study all ER models in this field. One entity may associate with multiple business activities, so it is called ER model of multi-business background, as is shown in algorithm 2.

**Algorithm 1:** The ER construction of single business background

**Input:** business model identification processID

**Output:** the ER diagram of single business background

1. According to processID, obtaining the ActivitySet of this business model;
2. Defining the variable ERSet, which is used to store the entities relation set;
3. for each activity in ActivitySet{
   Obtaining the relation set EA and AE that include Ai;
   For each entity Ei in EAU AE{
      If (Ei, Ai) not in ERSet
      put (Ei, Ai) in ERSet;
   }
}
4. using dot script to represent the entity relation set ERSet;
   defining the string ERDotStr to store the information of entity relation shown in dot script:
   (1) using dot to represent the information of entity attribute, entities are shown in matrix"box", while attributes are shown in "ellipse";
      for each entity Ei in ERSet{
         EntityDotStr+=" node (shape=box, style=filled, color = yellow );"+EntityNme+"n";
         Obtaining the attribute set AttrSet of entity Ei
         for each attri in AttrSet{
            EntityDotStr+=" node (shape=ellipse, style=filled, color=gray);"+AttributeName+"n";
            The edges of entity nodes and attribute nodes;
            EntityDotStr+= EntityName +"--" + AttributeName +"(color=gray)"+"n";
            }
         }
   (2) using dot to represent the node of activity which is expressed in diamond;
         for each activity Ai in ERSet{
            EntityDotStr+= ActivityName + "(style=filled, shape = diamond , color=green ) ";
         }
   (3) using dot to represent the relationship between entities and activities;
         for each relation (Ei, Ai) in ERSet{
            EntityDotStr += EntityName +"--" + ActivityName +"(color=blue)"+"n";
         }
5. Calling the layout engine graphviz to parse the dot script EntityDotStr and generate the ER diagram.

**Algorithm 2:** The ER construction of multi-business background

**Input:** entity domain domainID

**Output:** ER diagram of multi-business background

1. Obtaining all the entity set (EntitySet) in this field according to the domainID;
2. Defining the variable ERSet to store the entity-relation set of multi-business background;
3. For each entity Ei in EntitySet{
   Obtaining the relation set EA and AE that contains Ei;
   For each activity Ai in EAU AE{
      If (Ei, Ai) not in ERSet
      put (Ei, Ai) in ERSet;
   }
}
4. using dot script to represent the entity-relation set ERSet.
5. Calling the layout engine graphviz to parse the dot script and generate the ER diagram.
Data flow diagram graphically depicts the data conversion process from input to output in the perspective of data transmission and processing. Data flow diagram which is the main tool for structured systems analysis, represents the flow direction of information in the system and the logic processing capabilities of the system. EP-Model defines the input-output relationship of business activities and entities in business processes. This paper use "read" to describe the relationship between business activity and input entity, while "create" or "update" represent the relationship between business activity and output entity. In the EP-Model, transition set and route set defines the execution order of business processes. During the construction of a data flow diagram, an ordered set of business activities are firstly generated according to the conversion set and route set based on the business model; Then orderly analyzing the relationship between business activity and entity, such as "read", "update" and "create" and numbering each pair of relationship; Finally showing the data relationship order in dot script and calling layout engine of graphviz to parse it. After these procedures, data flow diagrams are constructed successfully. The specific construction process is shown in algorithm 3.

Algorithm 3: the construction of data flow diagram
Input: the identification of business model processID
Output: the data flow diagram under the business background

1. Obtaining the ActivitySet, RouteSet and TransSet of the business model according to the ProcessID;
2. Sorting the activity set according to the executing order of process and Obtaining the ordered set of activities(OrderActivitySet).
3. Defining the variable DFSet to store the data-flow set of this business background
4. Defining the variable int dataSeq to record the serial number of the data stream and is initialized to be zero;
5. Defining the variable EntityList to record the related entity set in activities and is initialized to be empty;
6. For each activity Ai in OrderActivitySet{
    (1) Obtaining the inputting entity set EA of Ai;
        Using "read" to identify the relationship between activity and inputting entities; The activities were triggered only if all the inputting entities of the activity are satisfied. Therefore all the inputting data of this activity share the same serial number;
        For each entity Ei in inputting set EA {
            put(dataSeq,"read",< Ei, Ai>) in DFSet;
            if(Ei not in EntityList)
                put Ei in EntityList;
        } dataSeq++;
    }
    (2) Obtaining the outputting entities set AE of Ai;
        There are two kinds of relation between the activity and outputting entities, these are "update" and "create"; Supposing after the finishing of the activities’ execution, the system output all the entities at the same time. Therefore all the outputting data flow share the same serial number;
        For each entity Ei in outputting set AE {
            if(Ei not in EntityList){
                put Ei in EntityList;
                put(dataSeq,"create",< Ai, Ei>) in DFSet;
            }
            else{
                put(dataSeq,"update",< Ai, Ei>) in DFSet;
            }
        } dataSeq++;
    }
7. Using dot script to represent the data-flow set DFSet;
8. Calling the layout engine graphviz to parse the dot script and generate the data flow diagram.
4. THE REALIZATION OF THE PLATFORM

4.1. Platform Overview

We have developed the platform which help to the modeling of data and business on the basis of EP-Model and achieve the integrated management of business entities and business models. The design ideas of the platform is shown in Figure 2.

The platform mainly consists of two parts: business entity management and business model management. Business entity is divided into different entity domains so that entities can be managed by domain, also the inputting and outputting entities which are required by business model are defined in each domain. During the definition phase of each single entity, attributes can be added to complete the entity. The property field of entities include name, code, data type, length, non-null and so on. Through standard definition of entity and some additional constraints, the conceptual model of these entities can be converted into a logical data model.

![Figure 2](image-url)

Figure 2. The research ideas of auxiliary platform for data and business modeling

The management of business model includes batch Import, design, storage and visualization. During the definition of inputting and outputting entities of the business model. The relation between business models and entity models are built to ensure the consistency of business model and data model by the reference to the business entities. Algorithm 1, algorithm 2 and algorithm 3 constructed the entity-relation set and the ordered data-relation set through the business model and achieved the automatic drawing of ER diagrams and data flow diagrams with the help of layout engine Graph Viz. Business model diagram are generated and persisted to the database through visual process designer or process batch entering template. Business model that are persistent stored can be parsed reversely to the relationship between multiple objects, as shown in Figure 3. The core classes include Process class (Process), activity class (Activity), Transition class (Transition), the entity domain class (Entity Domain), entity class Entity, role class (Role), the relation class between activities and entities (Activity Entity), the relation class between activities and roles (Activity Role).
4.2. Function Introduction

Data and Business Auxiliary modeling platform mainly includes two modules of business entity management and business model management. Business entity management is shown in Figure 4. Business entities are divided into different levels in accordance with the field, each entity's name in the domain is unique. The core functions of the module are batching import of entities and attributes and checking entity-relation diagram under the background of multiple business.

Figure 3. The core class diagram of auxiliary platform

Figure 4. The management of business entities
Figure 5. An instance of ER diagram under the background of multi-business

Figure 6. An instance of data flow graph
Business entities in the same domain can participate in a different business process, they generate links through different business activities. With the data already existed in the system, for example, there are more than one hundred entities in the domain of “well measuring” at present. Once the "Relation Diagram" button in entity management interface is clicked, the relationship diagrams between these entities(ER model under the background of multiple businesses) are shown clearly, as shown in Figure 5.

The core functions of management module of business model are batching import of business models, process designer, the visualization of business model, the preview of entity-relation diagram under the background of single business and data flow diagram. The batching-entering templates for the activity set and transition set of business model is established to achieve the quick storage and visualization refer to the definition of EP-Model. After the import of business models, the system set up the relationship between business activities and business entities. Once clicked the "Data Flow Diagram" tab in the management module of business model, system will automatically generates the data flow diagram under special business background, as shown in Figure 6. The left side of the toolbar provides some setting parameters of data flow diagram.

Once the tab of "entity-relation diagram" in management module of business model is clicked, system will automatically generate the entity-relation diagram under the background of such business, as is shown in Figure 7. The left side of the toolbar provides some setting parameters of ER diagram.

The system provides the restructuring functions of business activities, a new business model can be generated through the combination of existed activities. Because EP-Model defines the link between business activities and entities, the system will automatically generate the data flow diagram and ER diagram of new business models for the combination of any activities. In the business model design interface, if user click on the business activities that need to be reused with the right button, these activities will be added to the custom area, as shown in Figure 8. In custom interface, users need to set the execution order of these activities, the system will automatically generate a new business model diagrams, data flow diagrams and ER diagrams after setting.
5. CONCLUSIONS

This paper presents a formal definition of the business model called EP-Model and proposes the algorithms for automatic construction of ER diagram and data flow graph, along with the platform for business process modeling and data modeling. These works have achieved the integrated management of business entities and business models. Using this platform, users can understand the business model and data model from different perspectives. First, users are able to browse the data flow graph and ER diagram of any levels or any nodes according to the structure of business model. Second, users can have a clear view of business activities related to single or multiple entities and entities-relation diagram under the background of the multi-service activities. Third, users can choose to combine the business model related to the theme according to specific requests, in this way, the system will construct the corresponding ER diagram and data flow graph automatically.

Acknowledgements

This research is supported by National Major Scientific and Technological Special Project of China during the Twelfth Five-year Plan Period under grant 2011ZX05020-009 and the Step Plan of Beijing Municipal Science and Technology Commission under grant Z121109009212008.

REFERENCES