A Practical Method for Converting Transaction Flow Diagrams into Data Flow Diagrams

Zhen-sheng HUANG\textsuperscript{1,a}, Mei-zhen LIANG\textsuperscript{1,b}, Chao-qun ZHANG\textsuperscript{1,c}

\textsuperscript{1}Guangxi University for Nationalities, 530006, Nanning, China
\textsuperscript{a}zhenshengh@163.com, \textsuperscript{b}9479880@qq.com, \textsuperscript{c}25713893@qq.com

Keywords: Transaction flow diagram; data flow diagram; conversion; method

Abstract. Internet+ era, application software (system) development has achieved unprecedented development. This paper attempts to discuss a problem often encountered in the process of software development or system development, that is, the problem of transforming the Transaction flow diagram into a data flow diagram, propose transformation ideas and conversion rules, and explore a more practical conversion method. The example proves that it has achieved good results in daily practical application and teaching, and expects a better conversion method to be born.

Introduction

In the development of software engineering or information system engineering, the management transaction is often investigated in detail, and then the transaction flow diagram is drawn, but the data flow diagram needs to be drawn in the system analysis stage. Is there any connection between the data flow diagram and the transaction flow diagram? How to export (or convert to) a data flow diagram from a transaction flow diagram? What principles should be followed during the conversion process? How to establish a design specification for transaction flow diagrams and data flow diagrams? These issues have not yet been seen in the literature or the literature has a detailed introduction and explanation in these aspects. Therefore, the conversion of transaction flow diagrams and data flow diagrams is a very worthwhile question. The author has been engaged in information system development and curriculum teaching for many years. I have been troubled by these problems for many years. I have made some discussions and attempts in these aspects, and summed up a more practical conversion method.

Transaction Flow Diagram and Data Flow Diagram Overview

Transaction flow diagrams and data flow diagrams are the main diagrams for information system development.

A.Transaction Flow Diagram

The Transaction Flow Diagram (TFD) uses a number of prescribed symbols and connections to represent a specific transaction process. It is a graph that describes the transaction relationships, job sequences, and management information flows between units and people in the system. It can help analysts identify unreasonable flows in transaction processes for reorganization and innovation. \cite{1}It is physics model. The drawing of the transaction flow diagram is carried out according to the actual processing steps and processes of the Transaction. \cite{4}Its basic symbols and meanings are shown in Fig.1.

![Fig.1 Four pixel symbols of the transaction flow diagram](image-url)
[Example 1] An enterprise purchasing department set up a supplementary ordering system in order to ensure a certain inventory level. The transaction process is described as follows: The warehouse staff inputs the data of the warehouse to the system through the warehouse terminal, and the system compares the data to be sent and received with the inventory data to determine the amount of supplementary order, and generates an order report to the purchasing department. Try to map out the corresponding transaction process according to the transaction description.

The transaction personnel and departments involved in the example 1 are: the purchasing department, the warehouse staff, and the management department of the warehouse management system (the warehouse management department); the documents or statements involved include: sending and receiving data lists, inventory data, supplementary order lists, and order reports. Obviously, the warehouse management department is the transaction processing center, and the corresponding transaction flow diagram is shown in Fig.2.

![Transaction Flow Diagram](image)

Fig.2 transaction flow diagram corresponding to instance 1.

The transaction flow diagram mainly describes the transaction direction, which is to express the "Transaction process description" obtained by detailed investigation in a graphical way, which is more intuitive and easier to understand. The transaction process diagram is centered on the transaction (hosting as a person or unit) process, and promotes documents, statements or bills, mainly describing the direction of the transaction, and generally has no concept of data.

B. Data Flow Diagram

The Data Flow Diagram (DFD) is a tool for describing the system data flow. It abstracts the data independently and graphically describes the ins and outs of the information and the actual process.[1]

It is a primary tool for comprehensively describing the logical model of an information system. It uses four symbols to comprehensively reflect the flow, processing, and storage of information in the system. Data flow diagrams are abstract and general.[4] There are four types of pixels in the data flow diagram, as shown in Fig.3.

![Data Flow Diagram Symbols](image)

Fig.3 Four graphs of the data flow diagram

1. External entities. An entity that represents a person outside the system, which can be a person, thing, or other software system.
Data flow. A data stream is a path in which data travels within a system and is therefore composed of a fixed set of data. For example, the order form consists of the passenger's name, age, unit, ID number, date, destination and other data items. Since the data stream is data in the flow, there must be a flow direction. The data stream should be named with a noun or a noun phrase, except that the data stream with the data store is not named.

Processing. Processing is the unit that processes data, it receives certain data inputs, processes it, and produces output.

Data storage. Represents the static storage of information, which can represent files, parts of files, elements of databases, and so on.

The idea and method of transforming TFD into DFD

A. The idea of converting TFD into DFD

1. Both the transaction flow diagram and the data flow diagram dynamically examine the analysis object from the perspective of the process, and all of them use the graphical symbols to abstractly represent the survey results.

2. The connection between data and Transaction is embodied in: the data flow is generated along with the transaction process, it is a derivative of the transaction process; the data is basically collected according to the organizational structure or transaction process; when the data is aggregated, We also concentrate the data in different processing steps of the same transaction in the transaction process unit; the data flow diagram is drawn in accordance with the whole process of transaction processing.

3. There is a certain correspondence between the data flow diagram and the transaction flow diagram. The corresponding data flow diagram can be derived from the transaction process diagram. There are two common ideas: one is to first sort out the transaction process according to the transaction flow diagram, and then separate the data and forms that are in the corresponding investigation process, and then examine the flow of data, processing and storage, and They are drawn together into a complete data flow diagram [2]; the other is to separate the processing from the transaction process, and then examine the input and output data of each process, and all the processes in the transaction process. The organic integration of the input and output data streams forms a complete data flow diagram [3]. These two ideas are discussed in some monographs. The author introduces a third idea, that is, starting from the transaction process description and transaction flow diagram, the transaction process flow diagram and the data flow diagram are converted according to certain rules and steps.

B. A practical method for converting TFD into DFD

Starting from the practical application, the data flow diagram can also be generated from the transaction flow diagram to form a rough data flow diagram, and then combined with the transaction process description to further decompose and refine, merge and integrate to form a relatively complete data flow diagram. The specific conversion rules are described as follows:

1. Directly convert "people" or "departments" with only one-way arrows (only input arrows or only output arrows) into "external entities" in the transaction flow diagram, there will be two-way arrows (both input arrows and output arrows) "person" or "department" is directly converted to "handling";

2. Convert the documents (that is, the documents or forms promoted by personnel or departments) in the transaction flow diagram directly into the “data stream”, and only the documents or forms with "out" or only “in” Convert to "data store";

3. The “processing” converted from the transaction flow diagram is further broken down according to actual needs until it meets the requirements;

4. The preliminary data flow diagram formed is further optimized according to the transaction description. Such as reasonable consideration of data storage, name and numbering standardization.

[Example 2] A corresponding data flow diagram was drawn according to the transaction description and the transaction flow diagram of [Example 1].
Step 1: Generate a preliminary data flow diagram according to rules (1) to (2), as shown in Fig. 4.
Step 2: According to rules (3) and (4), the "processing" is decomposed and optimized according to the transaction description. ① Check and send data to the warehouse, compare the sending and receiving data with the inventory data, determine the name and quantity of the missing goods, and form a list of supplementary data;

Fig.4 Preliminary data flow diagram generated by the conversion

② classify and summarize the supplementary order data list, and form a supplementary order report form to the purchasing department. Finally, the data flow diagram generated after optimization, proper consideration of data storage and normalization is shown in Fig.5.

Fig.5 DFD generated after optimization

Conclusions

This paper presents the idea and method of transforming a transaction flow diagram into a data flow diagram, but the method of conversion is not unique. Some skilled and experienced system analysts do not draw transaction flow diagrams after detailed investigation, but draw data flow diagrams directly. But for beginners, the detailed investigators directly get the transaction flow diagram. Mastering this conversion method will greatly help the understanding and mastery of the development technology, and also expect the birth of a better method.

Acknowledgement

This paper is funded by the “2017 Guangxi University for Nationalities Information Management and Information System Specialty Construction Project”. I would like to express my heartfelt thanks!

References