

XML Based Representation of DFD

Removal of Diagramming Ambiguity

Swapna Salil Kolhatkar

Assistant Professor, Department of MCA, Marathwada Mitramandal's College Of Engineering,
Karvenagar,
Pune, India

Abstract—In the world of Information Technology, the working of a information system is well explained with the use of Data Flow Diagrams (DFD). DFDs are one of the three essential perspectives of the Structured Systems Analysis and Design Method (SSADM).^[3] The sponsor of a project and the end users are briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. But, various practical problems exist with the representation of DFDs. Different tools are available in the market for representing the DFDs. These tools are user friendly and based on the object oriented features. The diagrams drawn using these tools can be sent over the network for communicating with others.

On the other hand, the XML is platform independent, textual information and is totally extensible. XML is structured and is an excellent way to transfer the information along with the metadata over the network. XML can be used to transfer the information related to DFDs, there by removing the problems related to understanding the diagrammatic notations and concentrating more on the information flow in the system.

This paper is aimed at understanding the problems related to DFDs and representing it in XML^[4] format for their storage and communication over the network. The discussion is divided into four main topics – introduction to XML and DFD, problems related to DFD, an XML representation for DFDs and finally the conclusion.

Keywords-data flow diagrams; XML; metadata; diagramming tools.

I. INTRODUCTION

The development and implementation of any successful IT system, requires communication and coordination amongst team members spread over different geographical locations across the world. SSADM is a well defined standard means of understanding the process and data flow in any system. The data flow diagrams are one of the three essential perspectives of SSADM. The DFDs are useful in visualizing the various processes in a system along with the data that flows within the system. The DFDs either follow the Gane and Sarson or DeMarco

and Yourdan method of representation which causes differences in their pictorial representation. In such a demanding scenario, many tools related to DFD that have been developed, may also differ. Users of such tools are used to

working with pictorial representation that may differ depending upon the notations. Such diagrams have to be communicated over the network to other locations where the tools may differ.

Communication over the network has evolved over a time and in the current situation, XML has become a language of data communication over the network and also acts like a data store. Applications and systems communicate with each other using the XML format. Amongst these applications, those that are developed for diagramming purpose are already developed using the object oriented features. Such applications may store the data in XML format which is useful for network communication as well as for storage purpose. Let us consider a few important factors related to XML and DFD for our discussion related to the above.

- Extensible Markup Language (XML)

XML^[1] is structured; XML documents are easily committed to a persistence layer, are platform independent, contain textual information, is an open standard, is language independent, totally extensible, supports share-able structure (using DTDs) and enables interoperability. The XML file (.xml) was designed to transport and store data. The Document Type Definition (DTD) helps to define the legal building blocks of an XML document along with the document structure with a list of legal elements and attributes. The schema document (XSD) is used to define the schema and data types for elements and attributes. Together, these three files provide the complete information on data and metadata. This paper focuses on these three main files for further discussion.

- Data Flow Diagrams (DFD)

The data flow diagram is a simple graphical technique which is easy to understand and helps in defining the boundaries of the system. It is useful for communicating current system knowledge to the users along with being a part of system documentation file so as to explain the logic behind the data flow within the system. There are different notations to draw data flow diagrams (Yourdon & Coad and Gane & Sarson – shown in fig 1.1f), defining different visual representations for processes, data stores, data flow, and external entities. These differences in notations have given rise to the problems^[2] discussed in following section.

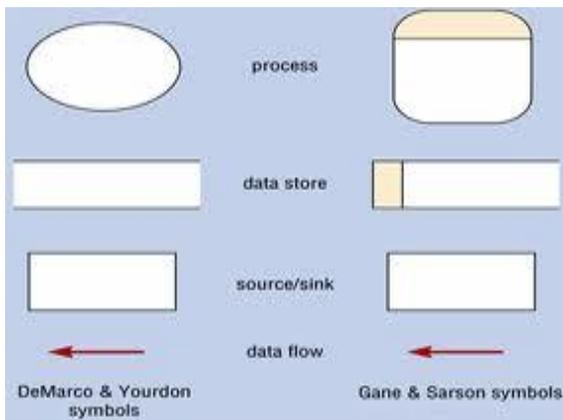


Fig 1.1f Differences in notations

II. PROBLEMS WITH DFD

- Time Consuming^[2] - The Data flow diagrams undergo a lot of alteration before going to users, so as to make the process little slow. A faster and efficient communication over the network will surely help in reducing the delay.
- Ambiguity in understanding^[2] - Different DFD models have different symbols like in Gane and Sarson process is represented as rectangle where as in DeMarco and Yourdan symbol it is represented as eclipse. These differences in pictorial representation cause ambiguity in system understanding at times.

III. XML REPRESENTATION OF DFDS.

Let us consider three XML based sample files : file1.xml, file2.dtd and file3.xsd

file1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- New document created with EditiX at Thu
Aug 19 03:07:32 PDT 2010 -->
<!DOCTYPE mySystem SYSTEM
"PressAutoSystem.dtd">
<mySystem>
  <context>
    <process id="0">Press Automation
      System</process>
    <entity id="S">Supplier</entity>
    .....
    <dataflow id="df1">
      <process id="0"></process>
      <entity id="S"></entity>
      <direction>S</direction>
      <content>Request</content>
    </dataflow>
    .....
  </context>
  <firstlevel>
    <process id="1">Registration</process>
    .....
    <entity id="S">Supplier</entity>
    .....
```

```
<dataflow></dataflow>
</firstlevel>
</mySystem>
```

The xml file ie file1.xml is actually used for sending the DFD related data or diagrams on the network. The file contains the metadata also in order to make the data self explanatory. The file can contain data related to context diagram (consisting of a single process, external entities and the data flows) and first level DFD (consisting of more detailed processes, external and internal entities, data stores). The same file can be used for the detailed logical and physical processes as well. As the level of detail goes on adding, the file consequently consists of other information.

file2.dtd

```
<!-- DTD created at Thu Aug 19 03:19:14 PDT 2010
with EditiX. Please insert an encoding attribute header
for converting any DTD -->
<!ELEMENT mySystem (context,firstlevel) >
<!ELEMENT context (process,entity+,dataflow+) >
<!ELEMENT process (#PCDATA) >
<!ATTLIST process id CDATA "0" >
<!ELEMENT entity (#PCDATA) >
<!ATTLIST entity id CDATA "E1" >
<!ELEMENT dataflow
(process,entity,direction,content) >
<!ELEMENT direction (#PCDATA) >
<!ELEMENT content (#PCDATA) >
<!ATTLIST dataflow id CDATA "DF1" >
<!ELEMENT firstlevel
(process+,entity+,dataflow+,datastore+) >
<!ELEMENT datastore (#PCDATA) >
```

The second file ie file2.dtd is used for providing the rules for correctness of file1.xml. It specifies details related to the root element and its sub-elements along with details related to the number of occurrences of elements. This file is used in order to check the correctness of xml file at the client and server side.

The third file ie file3.xsd can be used additionally in order to specify the data types of the contents of elements. It can be also used to specify the optional values in the said diagrams.

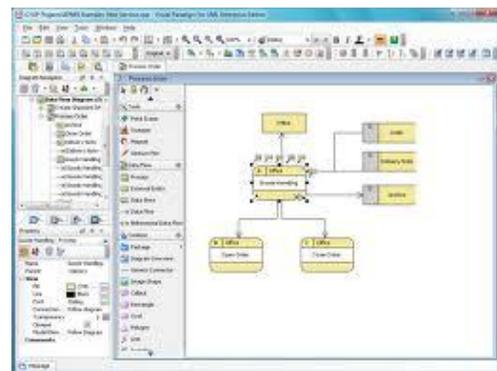


Fig 3.1 A Sample Tool for DFD

[5] Tools by which such diagrams can be drawn are any, for example Visio, Edraw, SmartDraw etc. A sample tool can be seen in Fig 3.1. Most of these tools are object oriented and are therefore easy to implement. Fig 3.2 shows a tool that saves data to an XML based file format. Accordingly a parser / translator that converts the data flow diagrams in XML format or reads an XML file for converting it into a diagram by identifying the diagramming notation that the user is acquainted with, can be provided additionally so as to take care of the problem of differences in diagramming notations.



Fig 3.2 Edraw Flowchart saves to an XML-based file format.

The communication over the network also becomes efficient due to the comparatively small size of xml files as compared to the size of files related to diagramming tools. XML files are easier to transport and improve readability of data due to the metadata.

IV. CONCLUSION

The current tools used for drawing of DFDs are quite user

friendly and an addition of an interface to interpret the xml file for drawing of the diagrams would prove beneficial for speedy transportation as well as customizing the diagrammatic representation. The XML and its supporting formats together provide a correct data and metadata for parsing to be done at the client application. This removes any prerequisites of software tool understanding or implementation, thereby customizing the data flow diagrams as per the user's understanding.

XML technology is a very popular, efficient and effective technology for communication of the data over the network. The implementation of any diagrams as xml format will be very beneficial for handling such pictorial well defined data representation as data flow diagrams. It facilitates better system understanding by removing any ambiguities in the notations.

REFERENCES

- [1] w3schools – XML, DTD, XSD.
- [2] Wiki answers.com : The advantages and disadvantages of data flow diagrams
- [3] Chris Gane and Trish Sarson. Structured Systems Analysis: Tools and Techniques. McDonnell Douglas Systems Integration Company, 1977
- [4] The Relevance of the XML Data Format for Access to Historical Datasets and a Strategy for Digital Preservation : D-Lib Magazine, February 2005, Volume 11 Number 2, ISSN 1082-9873, The eXtensible Past
- [5] Thinking XML: A glimpse into XML in the financial services industry Interesting lessons from a no-nonsense territory for XML Uche Ogbuji (uche@ogbuji.net), Principal Consultant, Fourthought, Inc.
- [6] An introduction to Model Driven Architecture Part I: MDA and today's systems Alan Brown (dwinfo@us.ibm.com), Staff, IBM <http://www.ibm.com/developerworks/rational/library/3100.html>
- [7] The Importance of Data Modeling as a Foundation for Business Insight by Larissa Moss, Method Focus, Inc. and Steve Hoberman, Steve Hoberman & Associates, LLC