

Message Transformation Designer for Messaging System Using Streaming Transformation for XML (STX)

Ms Rajani M V¹, Mr Annappa Swamy D R²

MTEch Student, Department of Computer Science & Engineering, MITE, Moodabidri, India¹

Associate Professor, Department of Computer Science & Engineering, MITE, Moodabidri, India²

Abstract: Fields like logistics, travel, financial services and insurance needs exchange of messages in wide variety of formats. Message contains elements which carry the data between applications or systems. Message exists in various formats like text, XML, CSV etc. Different application uses different message format, to reduce the difference in communication between such varying systems, the architecture and design of high performance message transformation designer is needed. Streaming Transformation for XML approach is efficient for transformation of message from the legacy/proprietary format to standard format.

Keywords: XML, STX, message transformation, encryption, Apache Camel.

I. INTRODUCTION

In today's world applications that are integrated with the messaging system mainly depends on the common data format. As the major source of information exchange between the organizations is done through the exchange of files consisting large volume of data. Application system use data in wide variety of formats some applications may use relational models, some may use flat files while other application may use XML documents. So that the development of the integration solution is required to solve the difference between such varying systems for communication.

STX (Streaming Transformations for XML) is used for the message transformation, the tree-driven transformation language for the XML. The conventional XML usually loads the whole XML document into the memory which consumes lot of memory space and more time to load the document [1]. STX allows performing the transformation with streaming character. STX does not allow building the tree structure representing the whole source document into the memory. The result events are generated as soon as the source events appear and they are processed. Since the STX uses streaming character for transformation the source document will be provided in the form of the streams of XML event, so the source document will be called as the source streams of XML event. Similarly the target document also generated as stream of XML event, which will be called as result stream of XML event. [2].

Major problem of transferring the details of the payment information which includes details of card number, CVC number, card type and information related to card holder which is in the form of plain text leads to the problem of hacking or misusing of the information. So the data are transferred from the plain text format into xml format and the xml file is encrypted and being transformed. Entire

xml document can be encrypted or only certain part of the document can be encrypted and shared between multiple users. Only the sensitive information related to payment present inside the XML document can be encrypted. [3].

File transfer protocol or Active MQ are used for transferring the files between the client and the server. The files received from client through FTP/Active MQ are stored in the database. Apache Camel is an integration framework which supports these components for transferring the files across the network. Apache Camel provides the users to define their own transferring rules and transferring large number of files.

II. LITERATURE SURVEY

A. Extensible Markup Language

XML files are used for representation of data over web. XML documents are simple and it is very easy to read the data present inside the document by just looking into the document. XML documents are stored into relational database either by model mapping or structure mapping. Model mapping uses two tables, node table and data table, node table keeps the node id and node name information whereas data table keeps information about the node values [4]. Structure mapping or schema based mapping uses XML schema or Document Type Definition to map XML documents into relational database [5].

B. Streaming Transformation of XML

STX is an XML based language used for transforming one XML document into other XML document. STX performs the transformation task based on the event specified in the XML documents; it consumes less memory space as it doesn't load complete data into memory. The software used for running STX transformation is referred to as STX processor. [2]

A. Integration framework

Apache Camel is open source integration framework. Camel allows multiple DSL's in various programming language like JAVA, groovy and it allows specifying the routing rules in the XML file. Camel supports more than 80 components for transferring of the data. Beans (or POJOs) are commonly used in the camel which allows us to use the camel's built in functions with our custom code. [6].

III. SYSTEM ARCHITECTURE

A. Before transformation

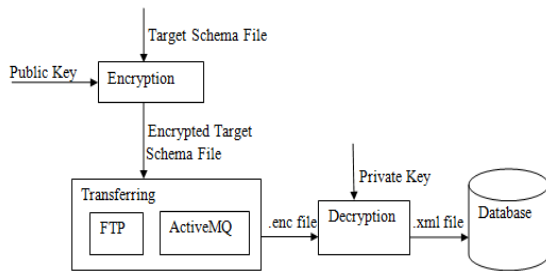


Fig 1: System Architecture before transformation

Fig 1 represents the architecture of the system before transformation of the target schema file. The target schema file is encrypted using the receiver public key, encrypted file is transferred across Active MQ, FTP to the receiver. The encrypted target schema file is decrypted using receiver private key and the file is saved in the database.

B. On and after transformation

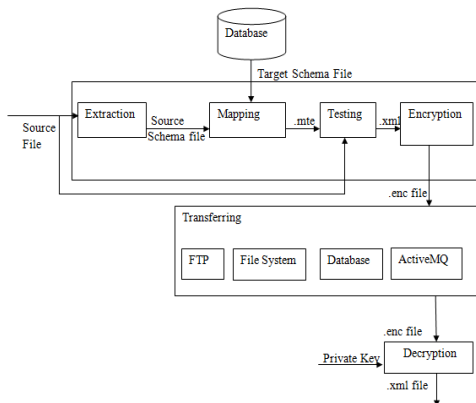


Fig 2: System Architecture on and after transformation

Fig 2 represents the architecture of the system on and after transformation of the message. The source file (actual input file) is read and the schema is generated for the source file, target schema file is retrieved from database. Both source schema file and target schema file is used for mapping the fields of the message. Source file and .mte file (file with all the details of mapping) are used transformation to obtain the desired output file. Output file is encrypted and sent back to the client through the component in which the target file was received. Later client decrypts the file with his private key and obtains the desired output.

IV. METHODOLOGY

A. Configuring the message

Before the transformation of the message is done generation of the schema file is required. Existing source XML schema file can be loaded into the designer tool or the schema file can be generated using the tool by extracting the fields from the source file [7]. If the file is text then the contents of the file is read and the fields are extracted based on the STX events. The extracted events are represented in the form of the XML structure. The chunk (group of words) read option is provided if the message format is text, chunk delimiters (eg “,;”) field value is set so that delimiter can be ignored while reading the message. Field attributes are also set like field name, field length, field data type and number of occurrence. Single schema file is sufficient for the transformation if files have similar group name and the field name but different field values.

B. Mapping the fields

The fields from the source schema file are mapped to the fields of the target schema file based on the user defined events [8]. User defined events are the rules on which the transformation is done. Template label is used to map the groups of the source and target. Templates have fragment and decider which is used to map the fields of the group. Fields of particular group should be mapped under particular template. Decider is used to exit the transformation based on condition defined by the user. Add Fragment label is used for mapping the fields which are directly under the root, which doesn't belong to any group. Variables can be declared either as local variables which is used for only one template where as global variables can be used by all the templates defined in the mapping section. Logical functions, mathematical functions can be applied to the field values, according to user requirements.

C. Transformation of the file

Once the fields are mapped, mapped file is saved with the extension .mte. This file contains all the details about the fields in source schema file and the target schema file which were used for mapping, also contains the details of the functions used on the fields. The details inside the file are represented as XML structure. The .mte file with the contents represented as XML structure contains details of all the STX functions used for the transformation of the file. Root element for the STX transformation is stx: transform, it contains various child elements like stx: group, stx: template, stx: variable etc, all this elements are used for the transformation.

D. Encryption of the file

In order to provide the security for the file, encryption of the file is done using RSA algorithm [9]. The target schema file is encrypted by client with server public key. Server receives the encrypted target schema file decrypts the file using his private key and transformation is done. After the transformation the transformed file is encrypted

by server using the client’s public key and it is transferred to client. Once the client receives the encrypted file he decrypts it using his private key and obtains the desired output file.

E. Transferring the file

Once the file is transformed as per as the defined events, the file will be transferred through FTP or ActiveMQ. The input file can also be read from the database and the transformed file can also be written into the database. When working with large number of files automatic pooling of the file is done which transforms the file and transfers it across various components.

V. RESULTS

A. Transferring the file

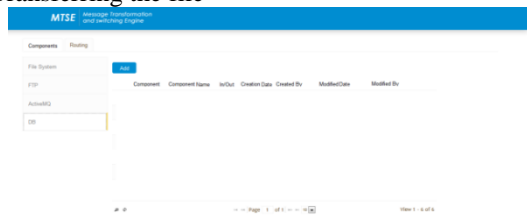


Fig 1: Transferring Screen

Fig 1 represents main screen for transferring the message before the transformation of the message. The file can be transferred through FTP or Active MQ, and also it can be taken from database or file system.

B. Configuring the message

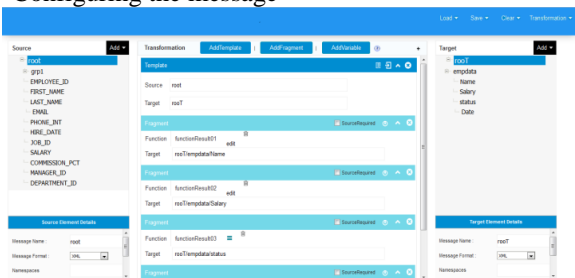


Fig 2: Configuring Screen

Fig 2 represents main screen for configuring the message both on the source and the target side. Along with that mapping the fields from the source and the target file.

C. Mapping the fields

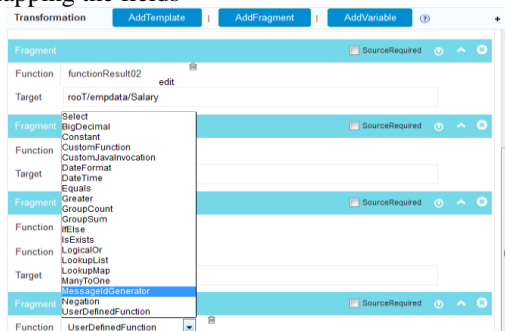


Fig 3: Mapping the fields

Fig 3 represents the mapping area. Various functions are applied after mapping the fields of source and target, functions can be user defined, logical, mathematical or Boolean functions applied on the field values.

D. Transformation of file

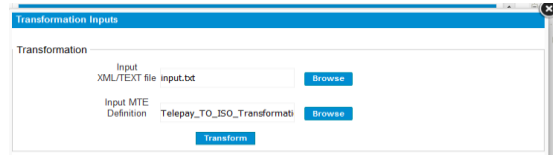


Fig 4: Transformation of the file

Fig 4 represents the screen for transformation of the file. Transformation of the file is done by providing the input file and .mte file. Result is obtained as XML file with desired values as per user requirement.

VI. CONCLUSION

In this paper we propose an efficient transformation method for converting message from text format to xml format which reduces communication difference between message producer and message consumer. Since the transformation is done using STX memory requirement for the transformation of the file is less and also the time taken for file conversion is less.

ACKNOWLEDGEMENT

I dedicate this work to my guide **Mr Annappa Swamy D R**, Associate Professor, whose interest and guidance helped me to complete this work successfully. I also extend gratitude to Prof. **Dr. Nagesh H R**, Professor & HOD, Dean (Academics) for his support and guidance.

REFERENCES

- [1]. Deborah AleyneLapeyre and B. Tommie Usdin“Introduction to XSLT Concepts” Mulberry Technologies, Inc.
- [2]. Streaming Transformations for XML (STX), stx.sourceforge.net/documents
- [3]. Azath.M, Sathish Kumar Konga“Secure Payment Information Using XML Technology”, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSE), Vol 2, Issue 5, May 2012.
- [4]. Dr Mrs Pushpa Suri, Divyesh Sharma “A Model Mapping Approach for storing XML documents in Relational database”, International Journal of Computer Science Issues, Vol 9, Issue 3, No 3,May 2012
- [5]. Dr.Mrs.Pushpa Suri, Divyesh Sharma“ Schema Based Storage Of XML Documents in Relational Database”, International Journal on Web Service Computing(IJWSC), Vol 4, No 2, June 2013.
- [6]. Claus Ibsen,JonathanAnstey“Camel in Action” (ISBN 978-1-935182-36-8), 2011.
- [7]. Ravi Varma, Dr. G. Venkat Rami Reddy“A Review of XML Parallel Parsing Techniques”, International Journal of Computer Science and Mobile Computing(IJCSMC), Vol. 3, Issue. 4, August 2014
- [8]. Bhavana V. Kumbhare, S. J. Karale“A Semantic Search over an XML data” International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSE),Volume 3, Issue 5, May 2013
- [9]. Saranya , Vinothini, Vasumathi “A Study on RSA Algorithm for Cryptography” International Journal of Computer Science and Information Technologies (IICSIT), Vol. 5 (4) , 2014, 5708-5709.