Dematerialized Deposits using XSI- An Application over XSI

V.Radha[a,1],Vasili Satyanarayana[a,2],Surabhi Upender[a,b,3],N.V.K.D.Ramesh Gorrila[a,b,4],B.Vijay Kumar[a,b,5]

[a]Institute for Development and Research in Banking Technology, Road #1, Castle Hills, Masab Tank, Hyderabad – 500 067 (A.P), INDIA
[b]Department of Computer and Information Sciences, University of Hyderabad, Hyderabad – 500 046 (A.P), INDIA
[1]vradha@idrbt.ac.in, [2]satyanarayanavasili@gmail.com, [3]surabhi.upender@gmail.com, 4gnvkdramesh@gmail.com, [5]vijaykumar.hcu@gmail.com

Abstract

Traditional banking was based on paper documents like 'deposit form', 'withdrawal form' etc. The comfort level for the customer on paper based business was high as they can retain the paper and show as proof in case of disputes. The present Internet and Core banking systems which are widely in use are transaction based. One drawback of the present situation is that customer is not able to keep a copy of the transaction due to various reasons like multiple channels and lack of suitable infrastructure at client end etc. As a via media, we tried to bridge the gap and see whether we can give the comfort and trust levels of the paper document coupled with the intelligence of the technology.

In this model we propose a method to describe financial product like "Fixed Deposits" using XML technology. At present, banks give the deposits in paper form with the deposit amount, maturity date and interest rate prescribed in the document. The banks system calculates the interest and pays the customer. The paper itself can't calculate interest. It just helps the customer as a proof alone. In our model, we are trying to see whether we can embed the interest calculation algorithm into XML based deposit and give to the customer, so that customer himself can find how much he gets using tools like XSI. This will act as a good verification mechanism, instead of solely depending on banks' systems. To show case this model as a POC, we used XSD, XML and XSI (an in house tool with IDRBT).

Keywords: *XSD, XML, Fixed Deposit, GUI , XSI.*

1. Introduction

In Today's world, every user wishes to carry out transactions instantly from present location with mobile devices. This fact drives the organizations not only to offer services online but also to ensure that the services reach end users in a flexible and convenient way on a wide variety of

[1] Corresponding author. Tel: 91-040-23534981-85; fax: 91-40-23535157

devices. In present scenario, developers are maintaining different versions of the same application, each version corresponding to a particular device. Our objective is to develop a XML schema definition (XSD) based user interface using which developers can maintain just a single version of their applications, instead of maintaining multiple versions. The whole idea in XSD based user interface is that a user input screen is painted based on the XSD and device properties; and once the user inputs the data, the data is exchanged as an XML document.

Our Contributions in this paper include

1. XSD based user interface as a service.
2. XSD based user interface on client device.

1.1 Introduction to XSI

The XSI tool is under development in IDRBT. The XSI dynamically generates GUI for the customer from composite XML schema document. Customer can enter the data into the GUI and then XSI generates an XML document conforming to the given XSD.

1.2 XSI – XSD based universal User Interface

The prominent steps involved in implementing XSD based user interface involves following four basic steps.

a. Generation of user interface (display) for requested schema so that user can enter data.
b. Extracting data entered through interface.
c. Creation of XML document with data entered by the user and elements from schema.
d. Validating this document against the schema.

These steps can be performed in two different approaches. Two methodologies are proposed in this paper based on whether these steps are to be carried out at client side or at server side.

1.3 XSI as Service

In this methodology, the majority of the steps are carried out at the server (as a service). It is aimed to offer services to clients equipped with less processing and memory resources. It makes use of the existing contemporary middleware technologies (such as JINI, UPnP and WAP) to offer the services to a wide variety of client devices.

1.3.1 XSI service:

It consists of different entities such as XSD and XSL repository, GUI generator and validation engine. Based on availability of requested schema document, GUI generator fetches that document from XSD repository and generates the interface based on the type of client device.

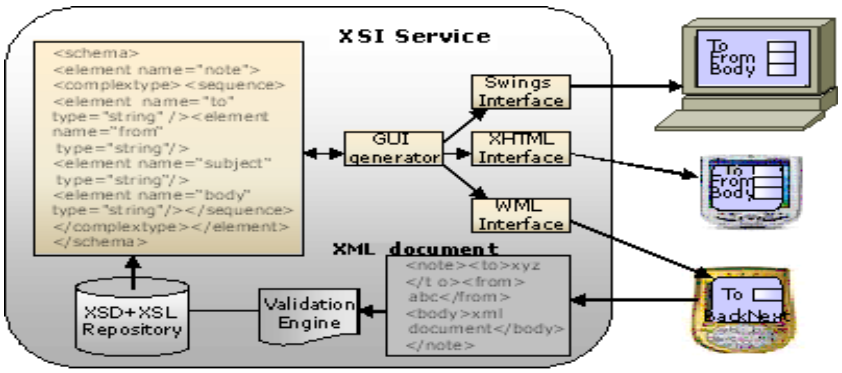


Figure 1. Architecture of XSI as service.

1.3.2 Gateway:

The gateway acts as a service provider because where the XSI service is deployed. It acts as an entry point for clients to access services (schemas) and to submit XML documents and as end point for organizations to deploy services for its clients.

1.3.3 XSD Repository:

The services are defined in the form of XSD and XSL documents. All XSD documents corresponding to different services are kept in XSD repository at the XSI service. When the end user requests for a particular service, then the corresponding schema document for that service is fetched from XSD repository and is passed to the GUI generator component to provide the interface.

1.3.4 GUI generator:

- For PCs, Java provides a rich interface capabilities through swings and the interfaces developed using swings are easily portable to other platforms. Even though Abstract window Toolkit (AWT) provides front-end capabilities, it suffers from portability problems due to its dependencies on native components during runtime. The PCs have sufficient processing capability and enough space to accommodate Java virtual machine. So, the whole interface can be displayed in a single screen using swings components. This swings interface represents XML document elements. Interface is generated at the service and then that code is passed on to the client through applets. Thereby, the end user will interactively transact with swings based interface.

- For mobile devices, WAP gateway connects mobile devices to the Internet to access XSI service. Mobile devices do not have enough resources for processing and storage. So, they do not support conventional web browser. Instead, mobile devices have a separate browser called micro browser that handles WML content. Some mobile devices such as PDAs can also handle XHTML content. The XHTML or WML interface based on the mobile device is generated for mobile end users from the requested schema document. Another challenge faced by mobile devices is the limitation of display screen. It is mandating that the interface be displayed in multiple screens. So the same content which can be displayed, as a single screen in PC would be displayed in multiple screens on mobile.

For any new device, appropriate conversion mechanism is handled at service without the knowledge of the application service provider and users.

1.3.5 Validation Engine:

The end user enters data through the input screen displayed on his device. That data is sent in XML document format to service. This XML document is validated against the corresponding schema and XSL document to satisfy certain constraints by using the validation engine at the server side. It is performed to ensure that data provided by user conforms to the schema definition.

1.3.6 Pros and Cons of this approach:

By this approach, the service can be extended to new kinds of devices without requiring any additional modifications by application developers. Scalability is the standard feature of this approach.

This approach relieves the burden on client devices in terms of processing because each of the above steps is carried out at the server side. Unfortunately, there is no way to ensure that the information provided by the end user conforms to schema at the client side itself. This slightly increases the network traffic and inconvenience to the end user by repeatedly entering the data in the interface. This can be overcome by providing some scripting type of mechanisms such as JavaScript at the client side. It solves the problem only to some extent.

1.4 XSI at Client

In this methodology, the service provider provides only a schema document corresponding to requested service. That document is passed on to client as input to the client's specific device software. The major entities involved in this approach are XSD and XSL repository, device specific client software, and validation engine.

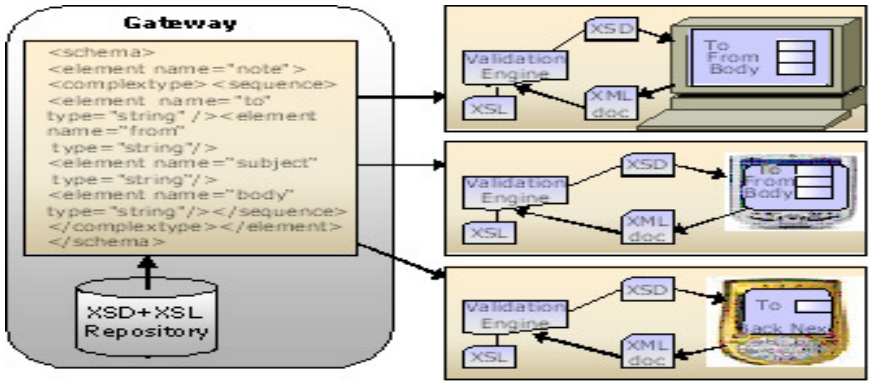


Figure 2. Architecture of XSI as service.

1.4.1 XSD repository:

For different types of services, the corresponding schema documents are added to the repository. The list of available services present in schema repository is displayed in Java applet. Through which, the clients can download the schema document and XSL document for the desired service.

1.4.2 Device specific software:

Once the client receives the XSD and XSL files, it is the job of client to generate the interface from the schema using the device specific software. To meet this purpose, the device specific client software is used to generate interface, create XML document and perform validation. The version of the software for a specific device can be usually obtained from the service provider website. It also handles the type of interface to be generated depending on the type of client device.

- For PCs, GUI interface is provided using Java's swing components and is displayed in a single screen.

- For mobile devices, the interface is to be generated in XHTML for PDAs and WML for mobile clients. This requirement is constrained by the limitation of display screen. It is the role of the specific software to display the interface in multiple screens in order to overcome this constraint and to provide flexible interface.

In this way, the device specific software generates interface through which the users can transact. The XML document is generated using data entered by the end user in the interface. It is validated using validation engine (also a component of the device specific software) at the client side. If XML document conforms to schema definition and XSL constraints, then it can be sent to the gateway or stored locally depending on user choice.

1.4.3 Pros and Cons of this approach:

By this approach, users have the flexibility to ensure that the information entered conforms to schema definition at the client itself. It reduces network traffic considerably and also relieves the burden on the server. But, this approach suffers from the scalability feature. If it is to be supported for a new type of device, it requires developing of device specific client software corresponding to that new device.

The information contained in XML document is in a structured format. The organizations incorporate application programs to extract the information from these documents and store it into native xml databases. Here, it is possible to deploy web services to do these activities and thereby providing platform independent services and automates the back end processing of requests. The collected information in databases is readily available for analysis purposes to get unknown information. This, in turn, provides opportunities for improving business efficiency, competitiveness and productivity.

1.4.4 Implementation Work Done

This section discusses the implementation work done to develop the universal user interface. The existing XML parsers are only supporting to parse XML documents but not for handling the schema documents. The development of this tool is divided into following phases.

- Parsing
- Structure analysis
- Rendering into interface using GUI components
- Customizing layout
- Creation of XML document

- Validation
- Serialization

The structure analysis is to be performed while parsing the schema document. During this analysis, the schema elements are rendered into GUI components by using different rendering languages such as swings, XHTML, WML based on client device. The interface represents a XML document form for entering the data. Once the data is entered, it provides various options such as save, send, close to allow the user to create XML document. Modern XML parsers are also providing support for validating the documents against given schema. After that, XSL is used to associate with XML document to stylize and specify dependencies among elements.

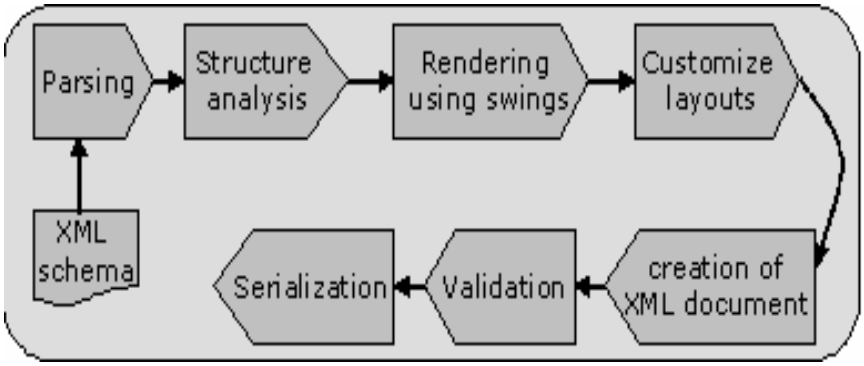


Figure 3. Design Steps.

1.5 Fixed Deposit

Fixed deposit is a money deposit at a banking institution that cannot be withdrawn for a certain "term" or period of time. When the term is over it can be withdrawn or it can be held for another term.

Interest rate is paid on deposit accounts by commercial banks and other depository financial institutions. Fixed interest rate is the rate which is fixed before the investment of money in the bank and cannot be changed until the maturity of the deposit is over.

1.6 Taking input

XSI can create new XML documents or edit existing XML documents. For new documents, XSI takes schema as input. For editing, XSI takes XML instance documents as input.

1.7 GUI generator

This module presents the user with GUI for the user to enter the data.

1.8 Validator

In this module user entered data is verified against its corresponding XSD document. Users would be alerted for appropriate corrections, in case the data is not in conformance with the XSD.

1.9 XML document generator

This module of XSI generates XML document by taking the XSD document as the input and user entered data.

1.10 Save/Send the document

The ultimate goal of XSI is to send the document to establish good communication between customer and the bank. The generated XML is either saved locally or sent through the network. The following diagram gives the complete idea about the work flow of XSI.

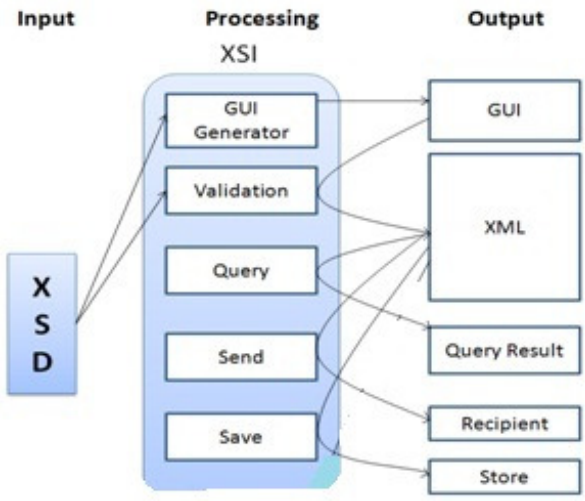


Figure 4. XSI – Conceptual View

2. Application Overview

We developed the prototype using XML, XSD and XSI. In this section, we explain the prototype components and work flow of the application.

2.1 Relationship among schema, advertisement and deposit document

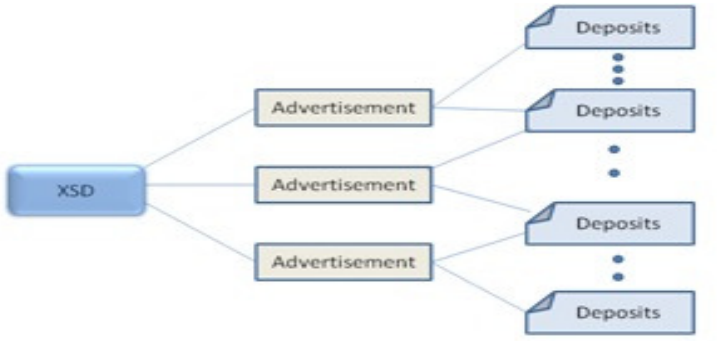


Figure 5. M-M Relationship among schema, advertisement and deposit document

- There can be one fixed deposit XSD structure to define the Deposit format.
- However, the interest rates keep changing and banks advertise different rates at different points of time, based on different criteria like Principal amount, term duration etc. So, each advertisement has to be represented as an XML document.
- There can be many customers depositing money against each advertisement and so there can be many deposit instances which are again in the form of XML.

2.2 The Application Components

The application crosses three boundaries in the entire work flow. They are 1. Bank's Server, 2. Bank's Teller and, 3.Bank's Customer

2.2.1 Bank's Server

The Bank's server hosts XML and XSD files which contain all the information about bank deposit forms as well as their structure. The banks display advertisements which show FD-Fixed Deposit interests from time to time. These advertisements are coded by using XML and XSD.

2.2.2 Bank's Teller

When the customer approaches the bank's teller machine, it issues the fixed deposit certificates to the customers. The bank teller contacts with the bank server to get the details in the form of XSD and XML and then fills the deposit with the customer details. Final deposit certificate is generated in the form of XML. Teller sends one copy of this XML to bank server and another to customer.

2.2.3 Bank's Customer

The customer receives the final deposit certificate in XML format. The customer can verify the deposit certificate at any point of time using XSI at his/her end.

2.3 Work flow of the model

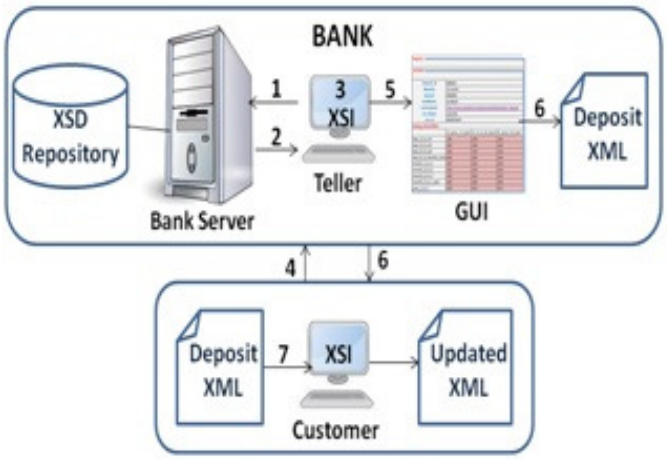


Figure 6. XSI Deposit Certificate model

1. The teller sends request to bank's server to get appropriate XSD/XML advertisement for the customer requirement.

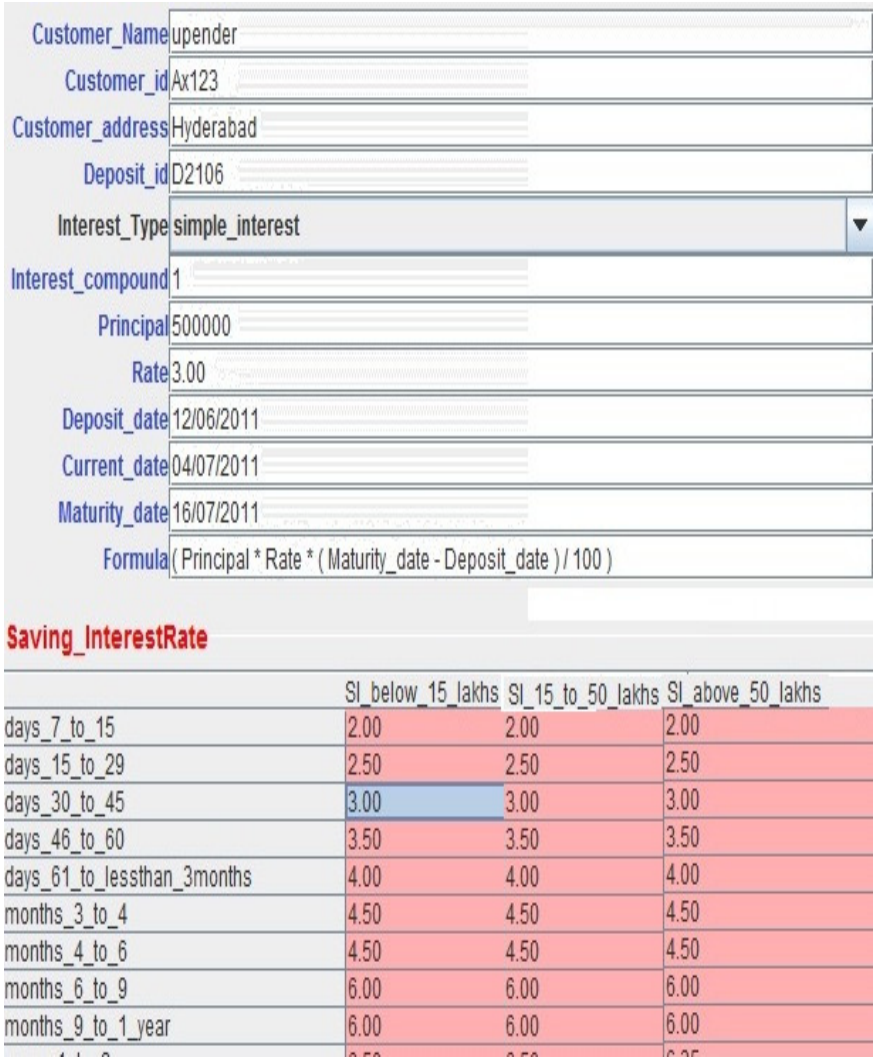
PERIOD	INTEREST RATES ON DOMESTIC DEPOSITS (%)				
DEPOSITS	Interest Rate			Interest Rates for Senior citizen	
	on Deposits Below Rs 15 lakhs	on Deposits of Rs 15 lakhs < Rs 50 Lakhs	on Deposits of Rs 50 lakhs < Rs 1 Crore	on Deposits below Rs. 50 lakhs	on Deposits of Rs 50 lakhs < Rs 5 Crore
9 months to less than 1 year	6.00	6.00	6.00	6.75	6.75
1 year to less than 2 Years	6.50	6.50	6.35	7.25	7.10
2 years to less than 3 years	7.10	7.10	7.10	7.85	7.85

Figure 7. Fixed deposit Advertisement

2. Bank server provides required XSD/XML advertisement to teller. These advertisements are coded by using XML and XSD.
3. The XSI generates GUI for the teller to enter customer specific details.
4. The customer approaches the bank and provides details for taking the fixed deposit at the teller. Deposit advertisement is created in GUI using deposit XSD.
5. The teller enters the information in the GUI and submits.

```
<Deposit schemaLocation="D:\Deposit\Deposit.xsd">
    <include schemaLocation="D:\Deposit\Deposit.xsd"/>
    <Axisbank>
         <Customer_Name></Customer_Name>
          <Customer_id></Customer_id>
          <Customer_address></Customer_address>
          <Deposit_id></Deposit_id>
          <Interest_Type></Interest_Type>
          <Interest_compound></Interest_compound>
          <Principal></Principal>
            <Rate></Rate>
            <Deposit_date></Deposit_date>
          <Current_date></Current_date>
          <Maturity_date></Maturity_date>
        <Formula>( Principal * Rate * ( Maturity_date -
Deposit_date ) / 100 )</Formula>
         <Saving_InterestRate MaturityTime="days_7_to_15"
AmountRange="SI_below_15_lakhs">2.00</Saving_InterestRate>
          <Saving_InterestRate MaturityTime="days_7_to_15"
AmountRange="SI_15_to_50_lakhs">2.00</Saving_InterestRate>
                        :
                        :
        <Saving_InterestRate MaturityTime="years_2_to_3"
AmountRange="SI_15_to_50_lakhs">7.10</Saving_InterestRate>
        <Saving_InterestRate MaturityTime="years_2_to_3"
AmountRange="SI_above_50_lakhs">7.10</Saving_InterestRate>
    </Axisbank>
</Deposit>
```

Fig. 8. XML based advertisement conforming to deposit XSD



Customer_Name	upender
Customer_id	Ax123
Customer_address	Hyderabad
Deposit_id	D2106
Interest_Type	simple_interest
Interest_compound	1
Principal	500000
Rate	3.00
Deposit_date	12/06/2011
Current_date	04/07/2011
Maturity_date	16/07/2011
Formula	(Principal * Rate * (Maturity_date - Deposit_date) / 100)

Saving_InterestRate

	SI_below_15_lakhs	SI_15_to_50_lakhs	SI_above_50_lakhs
days_7_to_15	2.00	2.00	2.00
days_15_to_29	2.50	2.50	2.50
days_30_to_45	3.00	3.00	3.00
days_46_to_60	3.50	3.50	3.50
days_61_to_lessthan_3months	4.00	4.00	4.00
months_3_to_4	4.50	4.50	4.50
months_4_to_6	4.50	4.50	4.50
months_6_to_9	6.00	6.00	6.00
months_9_to_1_year	6.00	6.00	6.00

Figure 9. GUI format for above XML

- XSI uses the Deposit XSD to validate the information entered by the customer.
- The Deposit XSD is used to validate the information entered by the customer in the GUI. If there are any violations, customers are alerted to correct the information.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Axisbank">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Customer_Name" type="xs:string"/>
                <xs:element name="Customer_id" type="xs:string"/>
                <xs:element name="Customer_address" type="xs:string"/>
                <xs:element name="Deposit_id" type="xs:string"/>
                <xs:element name="Interest_Type" type="xs:decimal">
                <xs:simpleType>
                            <xs:restriction base="xs:decimal">
                            <xs:enumeration value="simple_interest"/>
                            <xs:enumeration value="compound_interest"/>
                             </xs:restriction>
                </xs:simpleType>
                 </xs:element>
                  <xs:element name="Interest_compound" type="xs:decimal"/>
                  <xs:element name="Principal" type="xs:decimal"/>
                 <xs:element name="Rate" type="xs:decimal"/>
                  <xs:element name="Deposit_date" type="xs:date"/>
                  <xs:element name="Current_date" type="xs:date"/>
                  <xs:element name="Maturity_date" type="xs:date"/>
                   <xs:element name="Formula" type="xs:string"/>
                <xs:element name="Saving_InterestRate" type="InterestRate" minOccurs="30" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="InterestRate">
        <xs:attribute name="MaturityTime" type="xs:string" use="required"/>
        <xs:attribute name="AmountRange" type="xs:string" use="required"/>
    </xs:complexType>
</xs:schema>
```

Figure 10. XSD for Deposit form

6. If the validation is successful, the teller sends the XML document to the customer and one copy is sent to the bank's server for future reference. This XML file contains all the information about the deposit money, interest rate and valid or maturity date of the deposit.
7. The customer can use XSI tool to view the XML based deposit document.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Deposit schemaLocation="D:\Deposit\Deposit.xsd">
    <include schemaLocation="D:\Deposit\Deposit.xsd" />
    <advertisement schemaLocation="D:\date\Deposit.xml" />
  - <Axisbank>
      <Customer_Name>upender</Customer_Name>
      <Customer_id>Ax123</Customer_id>
      <Customer_address>Hyderabad</Customer_address>
      <Deposit_id>D2106</Deposit_id>
      <Interest_Type />
      <Interest_compound>1</Interest_compound>
      <Principal>500000</Principal>
      <Rate>3.00</Rate>
      <Deposit_date>12/06/2011</Deposit_date>
      <Current_date>04/07/2011</Current_date>
      <Maturity_date>16/07/2011</Maturity_date>
      <Formula>( Principal * Rate * ( ( Maturity_date - Deposit_date ) / 365 ) / 100 )</Formula>
      <Saving_InterestRate />
    </Axisbank>
    <Interest>1397.2602739726028</Interest>
    <Present_Value>501397.2602739726</Present_Value>
  </Deposit>
```

Figure 11. Final Deposit XML with customer deposit details

3. Advantages

3.1 Advantages to Bank

- Frequent changes of XML document for representing interest rates is allowed and so changes are reflected from time to time as documents just like in physical documents.
- Generates structured information contained in XML documents. The structured data is easier to handle, store, process, and analyze by application systems. Thereby, it helps organizations to make better decisions to improve efficiency, competitiveness and reduce fraud.
- Excessive processing on the server is reduced as customers need not approach the bank always. With tools like XSI, their own systems can participate in application processing.

3.2 Advantages to Customer

- We foresee the replacement of traditional browsers in all the financial transactions with browsers like XSI.
- The customers can calculate present deposits based on interest rates at any time.
- This tool is providing GUI, so customer can enter data in an easy way in offline mode as well, without connecting to Bank's server and he can keep all copies of his requests to bank(s) and the corresponding responses from bank(s) in a single view.
- Since the deposit is in standard XML format, it will help the customer to track his transactions in the case of banks' mergers.
- Since the Deposit Certificate is in standard XML form, new products like trading the deposits also can become a reality in future.

4. Conclusion

The XSI tool takes both XML documents and XSD as input and generates output in an XML format. It validates the information without human intervention. Through this tool we can provide an easy way to calculate the interest rates in the form of XML and these forms are sent to customers. So, the customer has valid proof for further verification. Bank servers will update the customer information in an easy way by changing the XML document.

5. References

[1] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen. Extensible Markup Language http://www.w3.org/TR/2008/REC-xml-20081126

[2] The World Wide Web Consortium Document Object Model (DOM) http://www.w3.org/DOM

[3] XSD tutorial from W3C http://www.w3schools.com/schema/default.asp

[4] V. Radha, S. Ramakrishna, " XSI- A XML schema based universal customer interface", Enabling Technologies for Smart Appliances (ETSA), IEEE Hyderabad, Jan 12-14, 2005.

[5] V. Radha, S. Ramakrishna, N. Pradeep Kumar: "Generic XML Schema Definition (XSD) to GUI Translator". ICDCIT 2005: 290-296.

[6] "SIM-Structured Information Messaging" CSI 2007 Annual Convention, Nov 28-Dec 1, 2007, by V Radha, M. Shiva Kumar and Kota Pavan Kumar.

[7] Patrick Garvey, "Generating User Interfaces from composite schemas", http://www.idealliance.org/xmlusa/03/call/xmlpapers/03-03-04.994/.03-03-04.html

[8] V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, A. Le Hors, G. Nicol, J. Robie, R. Sutor, C. Wilson and L. Wood. Document Object Model (DOM) Level 1 Specification. W3C Recommendation. October 1998. http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001

[9] Bergulund, S. Boag, D. Chamberlin, M.F. Fernandez, M. Kay, J. Robie, and J. Simon. XML Path Language (XPath) 2.0 Technical Report, W3C Working Draft. Available at http://www.w3.org/TR/XPath20/. 2001.

[10] Apache XML Project. Xerces parser, http://xml.apache.org/xerces-j/

[11] DevEdge Online – Gecko Developer Central http://developer.netscape.com/tech/gecko

[12] J. Einsenstein, J. Vanderdonckt, and A. Pureta, "Applying Model-Based Techniques to the Development of UIs for Mobile computers," Proc. Conf. Intelligent User Interfaces, pp. 69-76, 2001.

[13] Extensible Markup language (XML). The World Wide Web Consortium, http://www.w3.org/XML/.

[14] Microsoft's Universal Plug and Play Homepage. http://www.upnp.org/.

[15] F. Paterno and C. Santoro, "One Model, Many Interfaces," Proc. Fourth Int'l Conf. Computer-Aided Design of User Interfaces, pp. 143-154, 2002.

[16] P. Szekely, P. Sukaviria, O. Castells, J. Muthukumarasamy, and E. Salcher, "Declarative Interface Models for User Interface Construction Tools: The MASTERMIND Approach," Eng. For Human-Computer Interaction, L.J. Bass and C. Unger, eds., pp. 120-150, 1995.

[17] The Hypertext Markup Language (HTML) Home Page http://www.w3.org./MarkUp.

[18]The World Wide Web Consortium Document Object Model (DOM). http://www.w3.org/DOM.

[19] "Wireless Application Protocol", http://www.wapforum.org/.

[20] XForms–The Next Generation of Web Forms, http:// www.w3.org/MarkUp/Forms/ , 2003.

[21] XML Schema definition (XSD) Home Page. http://www.w3.org/XML/Schema.

[22] Steve Hashman and Steven Knudsen, "The Application of Jini Technology to enhance the delivery of mobile services", whitepaper (Dec 2001), http://wwws.sun.com/software/jini/whitepapers/index.html.

[23] Standard ECMA-262 ECMAScript Language Specification.http://www.ecma-international.org.

[24] Standard ECMA-262 ECMAScript Language Specification. http://www.ecma-international.org.