

**AN ONTOLOGY BASED FRAMEWORK FOR ELECTRONIC PAYMENT SERVICES IN
ELECTRONIC COMMERCE**

N.I. Job-Ujah, I.O. Omotosho, J.O. Emuoyibofarhe

(cdtulah@yahoo.com, mysoulbless@yahoo.com, eojustice@gmail.com)

*Department of Computer Science and Engineering , Ladoko Akintola University of
Technology, Ogbomoso, Oyo State, Nigeria.*

ABSTRACT

Payment authorization and settlement services carried out by payment processing services have, to a large extent, been based on a tight coupling between merchant and payment processing service with little flexibility for end users in their choice of preferred payment means while using merchant websites. Presently, merchant websites limit their end users to the payment processor of their (merchant website) choice therefore depriving users, whose credit/debit card providers are not accessible by the payment processor, of their services. This can be likened to a shopping mall using Point of Sale (POS) machines that can process ATM card purchase from few banks while located in a community with many banks offering ATM card services.

In keeping up with the World Wide Web as a distributed source of functionality, payment ontology provides a flexible means with the potential to effect online payments from different sources by defining an industry standard explicit specification of the payment domain and using this specification to instantiate various payment processing services to be advertised in a service registry. This paper presents the payment domain ontology in a framework which achieves flexibility in the interaction between merchant websites and payment processors to give end users a personalized user experience with the ability of a merchant approaching each user's payment service requests based on the user's preferences.

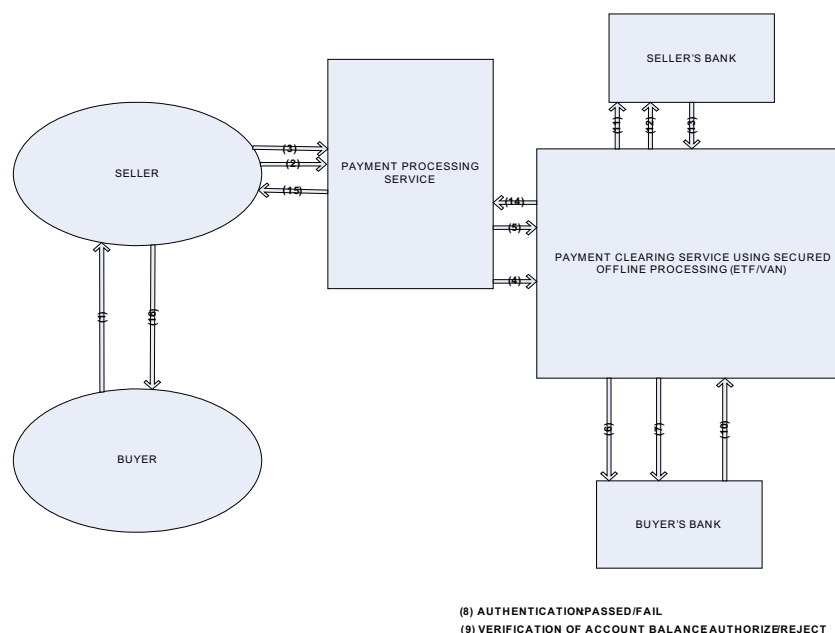
KEY WORDS: Payment Processor, Ontology, Semantic web Services

1.0 INTRODUCTION

Payment processors are middleware applications hosted online as web services used for the purpose of payment authorization and settlement transactions on behalf of online merchants (e.g. amazon.com). A payment processor via appropriate interfaces receives relevant information about a transaction from or on behalf of a merchant website and passes such information to an Automated Clearing House (ACH) responsible for authenticating the transaction details, authorizing the transfer of funds from the giving banks or credit card company involved and settling transactions with the institution receiving funds while at the same time settling transactions charges on behalf of the payment processor over a private, controlled network.

The Payment Processor provides encapsulated functionality on behalf of Merchant Websites thereby relieving Merchant Website designers of the complexity of functions involved in payment processing as well as relieving merchants of the exuberant cost of building a payment processor for its website. Merchants websites need a payment processor only when an end user is making payment for a product online so most medium and small scale merchant websites, on a limited budget, experiencing few hits per minute on its server would find it a cost saving venture settling for a third party payment processor.

Figure 1.0 shows a detailed architecture and Information flow involved in a typical payment processing scheme based on electronic funds transfer (EFT).



STANDARD PAYMENT PROCESSING SYSTEM ARCHITECTURE

Figure 1.0 Payment Processing Communication Model Architecture

2.0 MOTIVATION

It is a known fact that every commercial activity is incomplete until value changes hands, where value could be goods, services or financial instruments. In the physical economy, an effective and unhindered flow of cash cannot be over emphasized, it is important that a local economy be able to maintain a continuous cash flow as well as a balance in the amount of cash in circulation in relation to living standards and population.

Today, we live in a global economy where no single local economy can survive without interaction with foreign economies by way of providing services/goods internationally as well as using products of foreign economies. Since its evolution, the internet and World Wide Web has proved to be a valuable tool for promoting global trade with volumes of transactions taking place online. This has given rise to a new virtual economy (electronic economy) online and with every economy, there is the need for a mechanism, effective and secure, to handle financial transactions. What makes such mechanism robust is its ability to be flexible in terms of identifying and accepting value. Payment processing services, which provide such capacity to authorize and settle debit and credit transactions online, have locked out a lot of potential payment means as a result of the technology used in payment processing, thereby negatively affecting cash flow for online merchants and hampering potential

growth of the electronic economy. With flexibility in choice and use of payment processors affiliated to different financial institutions, merchants can deal differently with each user, provide a more robust customer friendly service and maximize its profits. These are the solutions this work seeks to provide.

3.0 LITERATURE REVIEW

Mans quest has been the unification of knowledge rather than its fragmentation, this, basically, has been the goal of an ontology from the philosophical perspective down to its application in computer science. Some questions ontology (the philosophy of being) has tried to answer for thousands of years include:

1. What are things?
2. What is the essence that remains inside things when they change physically?
3. Do concepts (objects) exist outside our minds?
4. How can entities of the world be classified?

In ontology we can distinguish between essence and existence. The essence of a thing is what the thing is (Gambra, 1999), what it can be used for or its purpose however, existence is to be physically present among things in the real world. For instance, money has essence as a medium of exchange of value and it also exists physically as the Nigerian currency (The Naira). When the Naira

is exchanged in the money market for the Dollar, it changes physically to the consumer however; the essence of that dollar still remains a medium of exchange of value as it was with the Naira. Ancient Greeks tried to find the essence of things through the changes they undergo and Aristotle's answer to the question "what remains in the essence of a seed when it becomes a tree?" was that the seed is a tree not completely fulfilled thus it is not the case of something which is not a tree becoming a tree, but that the tree changes its mode of being from not completely fulfilled (tree in potency) into completely fulfilled (Tree in act). So we can say the money changed from a weaker means of exchange (the Naira) to a stronger one (the dollar).

According to Nino Cocchiarella (1991), formal ontology is the systematic, formal, axiomatic development of the logic of all forms and modes of being. It studies the formal properties and the classification of the entities of the world (physical objects, events etc.) and the categories that model the world (concepts, properties etc.). In the words of Ortega Gasset (1939, page 91), *"una realidad que vista desde cualquier punto resultase siempre identical es un concepto absurdo"*. Therefore, ontologies must make unity compatible with variety. By deduction, the essence of an object (that exists) remains unchanged even after the object has changed thus we can say that Ontology is aimed at capturing the essence of concepts (Objects).

The word ontology was taken from philosophy where it means a systematic explanation of being; in computer science, it is defined as an explicit specification of a conceptualization (Gruber, 1993). Since ontologies are widely used for different purposes in different communities in computer science, it may be seen to necessarily include a vocabulary of terms and some specification of their meaning. This includes a definition and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretation of terms (Ushold et. al., 1999). Other communities (the database community, object oriented design community) have attempted to give an encompassing definition of ontology however some of the most widely accepted definitions are "Ontology is a formal and explicit specification of a shared conceptualisation" (Studer et. al., 1998), Fensel (2001) gave a technical definition of ontology as a model which looks very much like an ordinary object model in object-oriented programming which also consists of classes, inheritance and properties. Sheath (2003) defined ontology to consist of definitional aspects such as high-level schemas and assertion aspects such as entities, attributes, interrelationships between entities, domain vocabulary and factual knowledge, all connected in a semantic manner. Ontologies provide a common understanding of a particular

domain. They allow the domain to be communicated between people, organizations and application systems. Ontologies provide the specific tools to organize and provide a useful description of heterogeneous content.

4.0 DESIGN ANALYSIS

The vision for the web is a distributed source of information and with the emergence of semantic web and web services, the vision has been extended to include a distributed source of functionality. In using the web to search out information three critical activities are involved.

- (1) **Search:** A user, using a search engine is able to type in keywords of a search and receive a set of matching results.
 - (2) **Processing:** The user analyzes each member of the set of search results to select the most appropriate resource that meets his needs.
 - (3) **Connection:** The user types in the URL (Uniform Resource Location) of the selected resource in a browser window to establish a connection to the server hosting the resource and then receives the complete resource on screen. In searching for functionality, the web is used in a different way and mostly by experienced users/programmers
- (1) **Search:** Here the user, using a search engine types in keywords of a required

service and receives a set of matching results

- (2) **Processing:** The user analyzes, selects and queries the selected service providing required functionality by typing “?WSDL” at the end of the URL of the service. This action returns a Web service Description language (WSDL) document in Xml (Extensible markup language) format, specifying all accessible web methods accessible from the service and the required parameters and parameter types required to invoke the service delivering the required functionality.
- (3) **Connection:** The user identifies the web method, parameters and parameter types required to invoke the web method and create a web service reference from his web site which links to the required service using its URL (Uniform Resource Locator) using standard protocols such as HTTP (Hyper Text Transfer Protocol), SOAP (Simple Object Access Protocol) and UDDI (Universal Description Discovery and Integration). With the web reference, the user is able to instantiate a service reference object in order to make remote method calls to the web service's server while passing the required parameters in specified parameter types from his web site's application code.

4.1 ANALYSIS OF INFORMATION FLOW

Based on the conventional payment

processing architecture described in figure 1.0, a user at a workstation, on deciding to buy a product from an online merchant, first of all ensures that his available payment means is acceptable by the merchant and then goes ahead to submit relevant information needed to effect a transaction; information such as credit card details represented by activity (1). Activity (2) depicts the transfer of the user's information to the payment processor along with the merchants account information as in activity (3). The payment processor, in activities (4) and (5), pass information to the payment clearing service which uses an offline interface to communicate with the banks involved in the transaction for authorization and settlement of funds.

Depending on the outcome of the transaction, a success or failure notice is sent back to the merchant websites interface for final presentation.

It is important to note that security, being a major issue in information processing, is taken very seriously as there is an authentication activity between the payment clearing service and the bank to confirm that information submitted by the end user is valid and to verify that to a large extent, the individual supplying the information at the workstation is the owner of the account. Since all bank account holders are expected to keep their PIN (Personal Identification Number) codes secret, it serves as a good means of verifying the user.

Once the account is authenticated and user validated, the transaction is also validated to confirm that the available balance in the account is greater than the total value of the transaction to guard against having negative balances. Since this architecture is a Service Oriented Architecture (SOA), these activities of authentication and validation are carried out implicitly by the bank involved just in the same way a cheque passed cross the counter in the banking hall is processed without the user of the cheque knowing the details of authentication and verification processes the cheque goes through.

All entities involved in this model operate based on the request/response pattern in a client/server architecture. If the transaction succeeds, an appropriate response is sent back the same track to the merchant's website where details of the successful transaction is logged. Logged details enable the merchant confirm successful transaction and begin shipment procedures to deliver the product to the user.

This seemingly perfect model has some shortcomings which include:

1. In the event of a failure of a payment processor, the merchant's activities are abruptly halted.
2. Scheduled and unscheduled down times of the payment process would imply an unscheduled down time of the merchant's service.

3. There is an inflexible coupling between the merchant website and payment processor implying a manual way of connecting to the service in code therefore additional cost is incurred by the merchant in making a switch from one payment processor to another.
4. The problem in (3) above implies that the merchant has indirect communication links only to those banks connected to the payment processors services and as such, a client base limited to users of such banks.

These problems all point in one direction with respect to the merchant, LOSSES! To the end user, it is perceived as 'RIGIDITY' on the part of the merchant's website because only few payment means are acceptable. With more financial institutions involved, there will be increase in monies available for e-commerce as well as the number of online traders hence a boost in the

global virtual economy. The presented system lacks the level of flexibility needed to achieve this.

5.0 PROPOSED FRAMEWORK DESIGN

The analysis revealed that the major bottleneck is the existence of a tightly coupled M: 1 relationship (many to one) between merchant websites and payment processing services.

The proposed framework consolidates the standard payment processing system Architecture and the centralized Discovery Architecture II (Vassileois Tsetos *et. al*, 2007) in a flexible framework using the Payment Domain Ontology with instances advertised in service registries such as Universal Description Discovery and Integration Registry to achieve a flexibly coupled M: M (many to many) relationship between merchant websites and payment processing services. The framework enables software agents automatically select, process/compose and connect to a desired Payment Processing service without having to manually code a new connection to the new payment processing service.

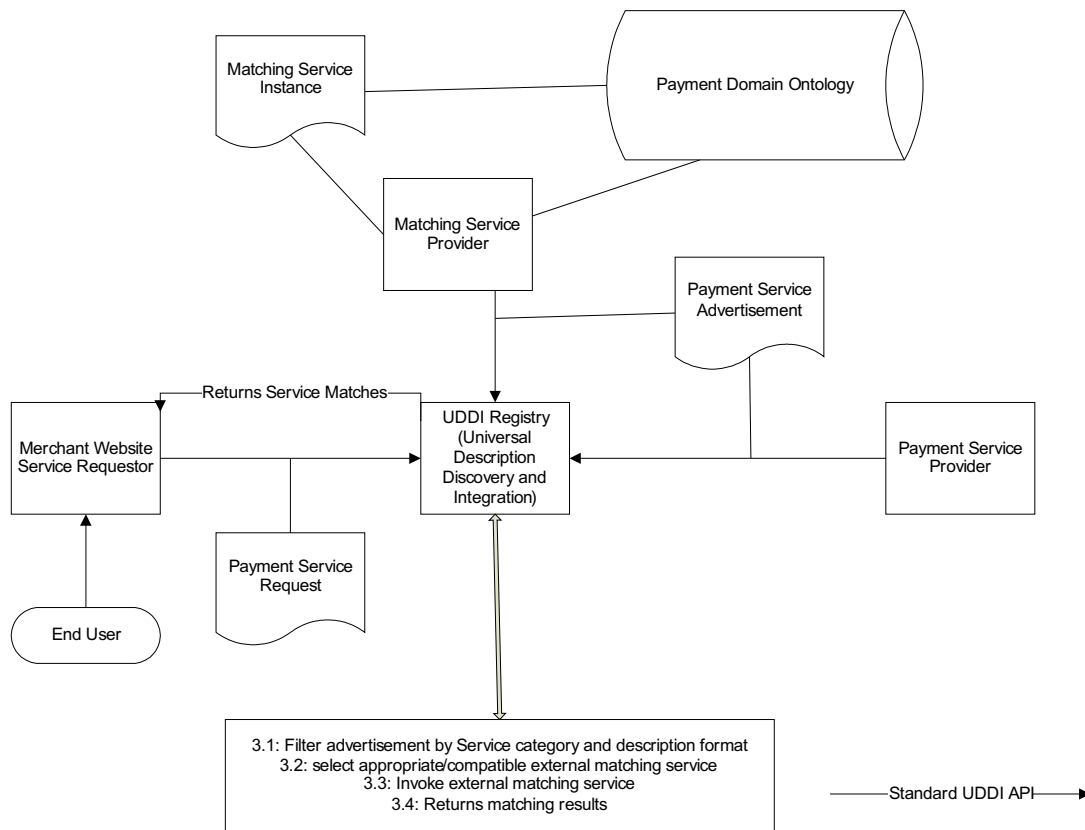


Figure 2: Centralized Discovery Architecture II.

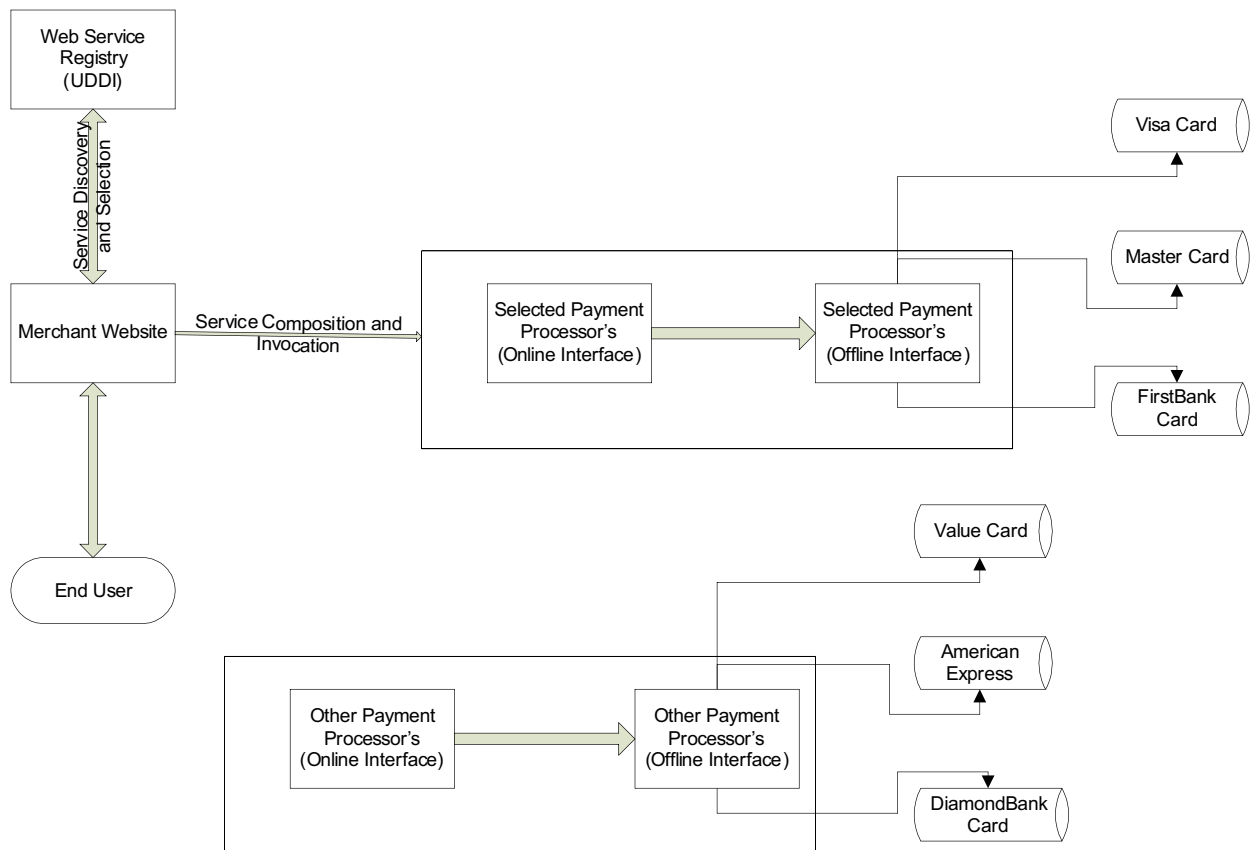


Figure 3: Architectural Framework for payment processing systems using Standardized Payment ontology.

6.0 PAYMENT ONTOLOGY

In the ontology design process, a seven step methodology was used, as proposed by Natalya and McGuinness (2001), which include:

1. Determine the domain and scope of the ontology
 2. Consider reusing existing ontologies
 3. Enumerate important terms in the ontology
 4. Define the classes and the class hierarchy
 5. Define the properties of classes/slots
 6. Define the facets of the slots
 7. Create instances
- **Determine the domain and scope of the ontology**

The ontology covers the online payment aspect of the payment domain and is used to enable software agents search for useful information contained in the ontology to make far reaching decisions without any human intervention. Information contained in the ontology includes such information for selection consideration (selection criteria) such as (1) Information for negotiation, where several ontology instances meet the selection criteria (2) Information for invocation of payment processor service.

- **Competency questions.**

Competency questions (Gruninger and Fox 1995) which serve as the test to determine the scope of the ontology and their answers include:

- **Q:** What criteria should be considered when selecting ontology instances that meet a user's need?
- **A:** The acceptable payment options, reliability of service
- **Q:** What payment characteristics should be considered when choosing a payment processor?
- **A:** Service charges, reliability of service
- **Q:** What information in an ontology instances are relevant in invoking a payment processor?
- **A:** URI, method name, method parameters
- **Q:** What characteristics of a payment processor affect its appropriateness for a merchants use?
- **A:** The available payment options connected to the payment processor
- **Q:** What payment options are available?
- **A:** credit card, wire transfer etc.
- **Q:** What is the base currency of the payment processor? It is important to note that the currency is relevant for converting from currency of the buyer to that of the seller. Since at online transaction time, it is the Information and not the physical cash that moves across boundaries therefore it is important to note the exchange rate between both parties and ensure that the buyer's funds in terms of the seller's and payment processor's currency are

sufficient to complete the transaction.

From the list of questions, several categories of information are required:

▪ **Information for Selection:**

- Names of payment options (banks and credit card companies) connected to the payment processor.
 - Base currency used by payment processor for each payment option.
 - Reliability of service
 - Payment processor's service charge

▪ **Information for Invocation**

- Uniform Resource Identifier (URI) of each payment option.
 - Method Name
 - Method Parameters
- **Consider reusing existing ontologies**

This ontology is an ad-hoc ontology built from scratch to suit online payment systems.

▪ **Enumerate important terms in the ontology**

The terms used in the payment ontology includes:

1. Payment Instrument: Cash, Gold, Stock, Bonds. The payment instrument determines the set of information to be supplied by an end user
2. Payment Scheme: wire transfer, credit card, e-payment
3. Charges

4. Currency

5. Acceptable Payment Options: master card, visa card, first-bank card

(Instances of the payment scheme)

6. Acceptable Currency (An instance of class currency)

7. Payment Details Format (depending on the payment scheme last_name/first_name/CVV/billing_addresses/amount)

▪ **Define Classes and Class Hierarchy**

The classes used in the payment domain ontology include:

- Payment Class
- Payment Scheme Class
- Payment Instrument Class
- Currency Class
- Payment Profile Class

Sub-classes of each class are shown as in the figure below. Relationships between classes show how instances of each class are related to instances of other classes.

▪ **Create instances**

The steps involved include:

- Defining classes in the ontology.
- Arranging the classes in a taxonomic (subclasssuper class) hierarchy.
- Defining slots and describing allowed values for these slots.
- Filling in the value of slots for instances.

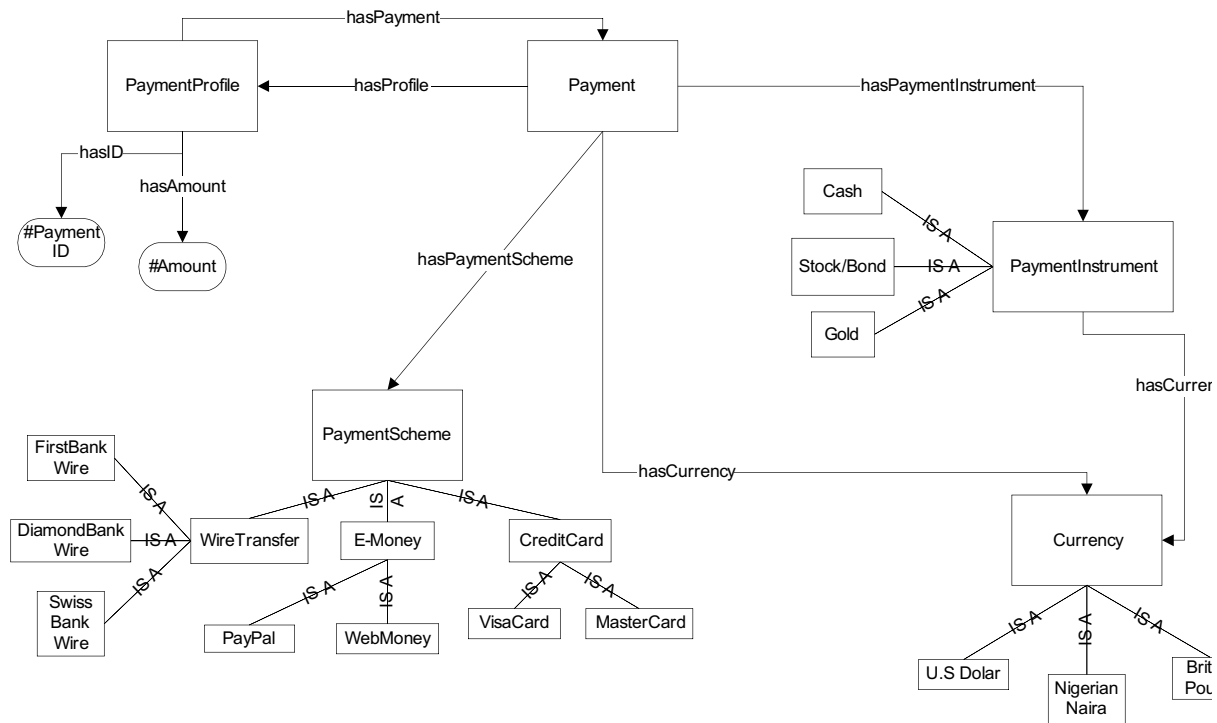


Figure 4.0: The Payment Domain Ontology

The Payment Domain Ontology defines a standard specification of concepts within the online payment sub-domain of the payment domain. For each Payment Service Provider, multiple instances of Payment Schemes can exist. Also Payment Schemes can have a single instance of Payment Profile, multiple currencies and multiple Payment Instrument class instances. Payment Profiles define the inputs acceptable by a payment Scheme as well as its outputs; this will aid the search process because the end user is looking for a scheme that accepts the input information he/she has. A Degree of Match (DoM) algorithm based on matching user requests against the input is used for match making and subsequent selection of Payment Processor.

▪ Architectural Framework

Operation

The Ontology Based Payment Service Framework (OBPSF) is based on the foundation principles as laid out by the authors of the Semantic Web Service Discovery Architecture, where information about web services in a given domain are advertised in a service repository accessible to web applications. These repositories are also accessible to services intending to advertise their services

- Payment service provider (payment processor) advertises list of service with the ontology repository
- Buyer enters payment preferences including the card company and maximum charge payable on a merchants web site user interface

- Merchant web site searches the ontology repository to locate payment processors offering services to the selected payment option. If none is found then an appropriate response is sent to the user interface
- Merchant web site processes relevant information about payment services found from the repository to determine which service provider is most efficient in meeting the needs of the buyer and merchant, economically.
- Once the best option is found, the merchant receives the parameters to be supplied on behalf of the buyer and displays them on the interface

Once the buyer enters necessary payment details, the merchant web site begins communicating directly with the payment processing service (master card, visa card etc.) directly.

▪ **Architectural Framework Design**

The proposed payment processing framework comprises of the Centralized Discovery Architecture II (CDA-II) (Vassileios Tsetsos *et. al*, 2007) embedded within the payment processing architecture as shown in figure. This architecture is shown in the figure 3.0.

7.0 CONCLUSION

This paper has proposed an architectural framework for payment processing services and payment domain ontology to ensure flexibility in

the mode of interaction between merchant websites and payment processing services, and ensure a broad base for effective cash flow in our global, virtual economy. The model, at its infant stage of development, forms part of ongoing research efforts to contribute towards the further realization of the World Wide Web as a Central Digital Nervous System (CDNS) able to “think”. Further work is in progress towards the implementation of this framework.

REFERENCES

- Jorge Cardoso: The Syntactic and the Semantic Web, Yurchak Printing Inc., 2007.
- Rama Akkiraju: Semantic Web Services, Yurchak Printing Inc., 2007.
- Sheng-Uei Guan, Sin Lip Tan and Feng Hua: A Modularized Electronic Payment System for Agent-based E-commerce, 2004
- Credit card Payment Processing-How it works. Retrieved May 04, 2009 From <http://www.yahoo.com/Credit Card Payment Processing--How It Works - Yahoo! Small Business Help.htm>
- Natalya F. Noy and Deborah L. McGuinness: Ontology Development 101: A Guide to Creating Your First Ontology, 2001
- Andr e C. M et al.,: CORES: Context-aware, Ontology-based Recommender system for Service recommendation
- Dr. P.E.A. Vandenbossche: A Pattern for Global Payment Optimization via REA

Ontology

- David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srin Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, Katia Sycara: OWL-S: Semantic Markup for Web Services, 2004
- J.Breuker and W.V.de Velde: The Common KADS Library for Expertise Modelling, IOS Press, Amsterdam, 1994.
- Chandrasekaran, B.: Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design, IEEE Expert, 1, No.3, pp.23-30, 1986.
- B. Chandrasekaran, J. R. Josephson and Richard Benjamins, Ontology of tasks and methods, ECAI98 Workshop Notes on Applications of Ontologies and Problem-Solving Methods, 1998.
- FEIGENBAUM, E.A. (1977) "The art of artificial intelligence Themes and case studies of knowledge engineering" Proc. of 5th IJCAI, pp.1014-1029, 1977.
- Genesereth, M. and Nilsson, N. (1987) Foundation of Artificial Intelligence, 1987.
- Gruber, T. <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
- Guarino N., Some Organizing Principles for a Unified Top-Level Ontology. Revised version of a paper appeared at AAAI 1997 Spring Symposium on Ontological Engineering (LADSEB-CNR Int. Rep.02/97)
- Guarino, Nicola: Some Ontological Principles for Designing Upper Level Lexical Resources. Proc. of the First International Conference on Lexical Resources and Evaluation, Granada, Spain, 28-30 May 1998.
- Kahn, G. and J. McDermott: MORE: An intelligent knowledge acquisition tool, Proc. Of IJCAI-85, 1, pp.581-584, 1985.
- <http://www.opencyc.org/>
- Mizoguchi, R. et al.: Proc. of JKAW92.
- Mizoguchi, R. et al. (1995) Task ontology for reuse of problem solving knowledge, Proc. KB&KS95, Enschede, The Netherlands.
- Smith, B. & Welty, C. (2001) Proc. of the Second International Conference on Formal ontology and Information Systems: FOIS2001, ACM ISBN 1-58113-377-4 ACM Press
- Sowa, J. (1995) Distinction, combination, and constraints, Proc. of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing.
- <http://ontology.teknowledge.com/>
- Standard Upper Ontology, <http://suo.ieee.org/>
- <http://www.semanticweb.org>