EPiC Series in Computing
Volume 63, 2019, Pages 53–62
Proceedings of 32nd International Conference on
Computer Applications in Industry and Engineering
EPiC Computing

AI Based Data Architecture Impact Analysis

Dr Debasis Chanda
Dean-Academic & Professor-Operations Management
MDI Murshidabad
West Bengal, India, Pin 742 235
cdebasis04@yahoo.co.in, dc@mdim.ac.in

Abstract

Enterprises today are technology driven and comprise of plethora of applications that may be categorized based on the technology that they are developed and deployed on. For enterprises that have existed across years and across multiple business cycles, the technologies may be classified as legacy, mature or emerging. The challenge lies in interoperability within and without the organization, especially with respect to the business objects that are required across business functions, to realize the capabilities of the organization. This is also true for scenarios of M&As (Mergers & Acquisitions) and also during creation of JVs (Joint Ventures).

Enterprise Architecture (EA) defines the Business-Technology alignment in organizations, and is an established methodology for business transformation and establishing enterprise maturity in the keenly competitive business world. Business objects are defined as Data Architecture artifacts within the ambit of EA.

The challenges to business object interoperability arises due to the incompatibility of technologies used by the applications. This leads to the well explored n*(n-1) scenario, where n is the number of application interfaces. This has serious implications towards business health of the organization, and risk to the BAU (Business As Usual) of the organization. This is because in a complex mesh like n*(n-1) scenario, it becomes practically impossible to identify the impact of changes to business capabilities in an inconspicuous attribute of a business object in an application domain.

Thus the impact analysis of business objects / data as defined by traditional description is a challenge to business sustainability of organizations. These challenges in data architecture impact analysis may be mitigated by the AI (Artificial Intelligence) paradigm, by taking recourse to the very powerful features of AI, by defining predicate calculus based knowledge bases.

In our paper we consider the Banking domain for carrying out our discussions.

Keywords: AI, Knowledge Base, Predicate Calculus, Business Object, Interoperability

Q. Yuan, Y. Shi, L. Miller, G. Lee, G. Hu and T. Goto (eds.), CAINE 2019 (EPiC Series in Computing, vol. 63), pp. 53–62

1 Introduction

Today, enterprises are facing rapidly changing business environment and can no longer make predictable long term provisions, because of the turbulent market conditions, regulation of the working conditions, fast technological mutations and so on.

As such, enterprises focus only their core capabilities on their value chain while they collaborate with other enterprises that have other complementary capabilities. This strategy requires the management of collaborative networks across business partners, and even business competitors. [5].

The virtual enterprise (VE), which is in general the collaborative partnership between business partners in value chains, has become a key factor for survival in the competitive business environment.

The core requirements for efficient collaboration among business partners are agility and interoperability of the VE models to cope with the changing business environment. The VE is designed to increase competitiveness, to optimize resource utilization, to increase scale of the business, and to take advantages of the complementary capabilities of the business partners [5].

In the world of business the manifestation of VEs is through M&As (Mergers & Acquisitions) / JVs (Joint Ventures).

Service-oriented computing is becoming the prominent paradigm for leveraging inter-enterprise information systems to complete higher-order business transactions at the heart of the modus operandi of the virtual enterprise (VE). [4].

The goal of this paper is to present a new data / information modeling framework for the VE, which is focused on data interoperability. This AI (Artificial Intelligence) based framework uses Predicate Calculus Knowledge Bases. The rest of this paper is organized as follows: Section 2 reviews the previous approaches which are related to our modeling framework. In Section 3, we introduce our modeling framework and an example scenario to facilitate understanding. In Section 4 we carry out discussions on our proposed Framework. Finally, Section 5 provides some conclusions.

2 Related Work

The related work focuses on interoperability of processes and inter process communication.

Originally coming from the business world, service-oriented architecture (SOA) paradigm is expanding its range of application into several different environments. [1].

Web services have acquired enormous popularity among software developers. This popularity has motivated developers to publish a large number of Web service descriptions in UDDI registries. [2].

The research ‘An XML-based schema definition for model sharing and reuse in a distributed environment’ [3] leverages the inherent synergy between structured modeling and the eXtensible Markup Language (XML) to facilitate model sharing and reuse in a distributed environment.

The paper [5] suggests a framework for designing the agile and interoperable VEs.

Marcos Lopez-Sanz et al. [6] propose the specification of an architecture centric model driven development method. To achieve this, they study the architectural properties of the SOA paradigm and follow a development approach based on the MDA proposal.

Claus Pahl’s paper [7] presents ontology-based transformation and reasoning techniques for layered semantic service architecture modeling.

Choreography is an essential element of business integration that allows the modeling of the external behavior of services for a specific interchange or collaboration. Arroyo et al. [8] describe the practical application of a semantic web service-based choreography framework.

David Chen et al. [9] define and clarify basic concepts of enterprise architectures. An overview on architectures for enterprise integration developed since the middle of the 1980s is presented.

The advances in information and communication technologies and economic factors impelled organizations to engage in new forms of collaboration, such as collaborative networks (CNs) [10].

Karim Baına et al. [11] aim, through their paper, to present a model supporting dynamic heterogeneous workflow process interconnection.

Daniela Grigori et al. [12] present a set of integrated tools that supports business and IT users in managing process execution.

The paper by Ricardo Jardim-Goncalves et al. [13], after illustrating the general motivations the industrial SMEs have to adopt open architectures to achieve interoperability for extended and collaborative enterprise practices, presents the emerging model-driven and service-oriented architectures.

In their paper [14], Jagdev et al. show how emerging semantic web services technologies such as the web services execution environment (WSMX) facilitate the creation of.

Jung et al [15] in their paper proposes an architecture for integrating knowledge management systems (KMSs) and business process management systems (BPMSs) to combine the advantages of the two paradigms.

The paper by Rezgui [16] describes research aimed at supporting the formation and operations of virtual enterprises through the setting-up of service-oriented workspace environments.

3 AI Driven Data Modeling Framework

The global economic scenario (past, current & present) has been witness to continuing M&As (Mergers & Acquisitions), especially in the Banking & Financial Institutions sphere. In order to avoid legal, financial and other complex problems related to M&As of two banks/financial institutions, we have presented a proposal of virtual consolidation involving interoperability of data. For the purpose of Virtual Consolidation, there would be a Memorandum of Agreement between 2 banks for implementation of services to the public, so that each bank can use the other's resources at the required time and place, as if the two banks are one bank.

3.1 Proposed Approach

Our Paper proposes an AI driven Service Oriented Knowledge Modeling Framework.

For our purpose, the representative domain of discourse considered in the example scenario is the deposit function of two banks viz. Bank1 and Bank2. The realization of the business process 'display balance' (i.e. the display of balance in an individual's account/s) for the consolidated bank (following the merger of the two representative banks) requires the orchestration / composition of the two business processes for each of the two banks into one consolidated business process, and thus interoperability of relevant data between the two banks. The orchestration / composition of the business process 'display balance' is achieved by means of our proposed modeling framework based on Predicate Calculus and knowledge (as opposed to data) search techniques based on pattern search [17].

3.2 Knowledge Representation Model Using Predicate Calculus: / Realization of Processes

For the domains of discourse (Deposit Functions of Bank1 & Bank2), expressing the knowledge represented in the UML diagrams Fig 1, Fig 2, Fig 3 and Fig 4, we arrive at the following representative set of Predicate Calculus expressions:

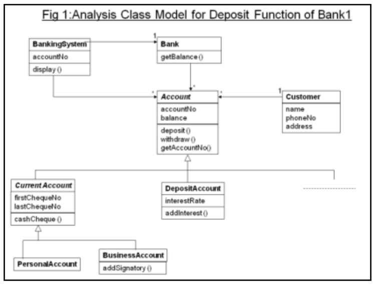

Figure 1: Analysis Class Model for Deposit Function of Bank1

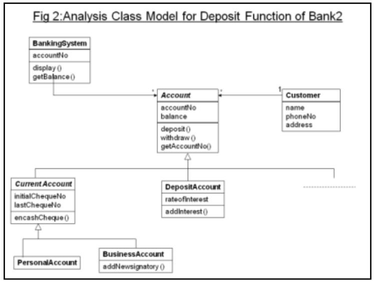

Figure 2: Analysis Class Model for Deposit Function of Bank2

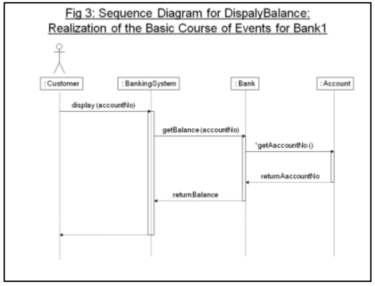

Figure 3: Sequence Diagram for DisplayBalance: Realization of the Basis Course of Events for Bank1

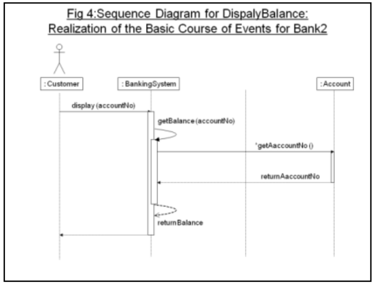

Figure 4: Sequence Diagram for DisplayBalance: Realization of the Basis Course of Events for Bank2

1. class (bankingsystem)
2. bankingsystem (banksystem1)
3. banksystem1 (private (int (AccountNo)), display ())
4. class (bank)
5. bank (int(getBalance(AccountNo)))
6. multiplicity (bankingsystem , bank, one)
7. abstractclass (account)

8. account (protected (int (AccountNo)), protected (int (Balance)), deposit ( ), withdraw(),int(getAccountNo()))
9. abstractclass (currentaccount)
10. subclass(account,currentaccount)
11. depositaccount (protected (int(InterestRate)), addInterest ( ))
12. currentaccount (protected (int (FirstChequeNo, protected (int (LastChequeNo), cashCheque())
13. subclass(account,depositaccount)
14. subclass (currentaccount,personalaccount)
15. subclass (currentaccount,businessaccount)
16. businessaccount (addSignatory())
17. has (customer,account)
18. multiplicity (bankingsystem,account, many)
19. multiplicity (customer, account, many)
20. navigation (banksystem1, bank, unidirectional)
21. navigation (bankingsystem, account, unidirectional)
22. class (customer)
23. customer (private (string (Name)), private (int (PhoneNo)), private (string (Address)))
24. navigation (account, customer, bi-directional)
25. bank (bank1)
26. bankingsystem (banksystem1)
27. bankingsystem (banksystem2)
28. customer (p_kumar)
29.p_kumar(prakash_kumar,12345678,po_box_19_AB_Avenue_Kolkata)
30. has (p_kumar, da_1001)
31. depositaccount (da_1001)
32. da_1001 (12345, 2500)
33.banksystem2 (private(int(AccountNo)),display(),getBalance())
34.equivalent(InterestRate,RateofInterest) 35.equivalent(addSignatory(),addNewsignatory())
36.equivalent(InitialChequeNo,FirstChequeNo)
37.bank(bank2)
38.equivalent(encashCheque(),cashCheque)
39. customer(p_dev)
40. depositaccount (dep_1005)
41. has (p_dev, dep_1005)
42. dep_1005 (2468,5000)
43.p_dev(pratik_dev,98765432,street_10_number_18_CD_Avenue_Kolkata)

All static structure features of UML including inheritance, composition, aggregation, etc. can be properly represented in Predicate Calculus terminology. In Predicate Calculus, predicates, which represent relationships between zero or more objects, are expressed in lower case, variables start with upper case, constants in lower case, functions start with lower case. ***The above Knowledge Base itself can be distributed or centralized in nature depending on the requirement scenario.***

3.2.1 Process / Service Representation Using Predicate Calculus knowledge based Data Model

The Service ‘display balance’ for Bank1 is a goal in AI. The goal for Bank1 (Fig 3) is the expression: displaybalance (bank1). This is represented by the implication:

{customer (X) Λ message (customer (X), bankingsystem (P), display (AccountNo)) Λ message (bankingsystem (P), bank (R), getBalance(AccountNo)) Λ message (bank (R), account(N), getAccountNo ( )) → displaybalance (bank1)

The services (goal) for Bank2 (Fig 4) is the expression: displaybalance (bank2). This is represented by the implication:

{customer (Y) Λ message (customer (Y), bankingsystem (Q), display (AccountNo)) Λ message (bankingsystem (Q), getBalance(AccountNo)) Λ message (bankingsystemm (Q), account (Z), getAccountNo ( )) → displaybalance (bank2)

The above services (goals) are WFF (Well Formed Formulae) in AI, having proper syntax & semantics. Also, the inferences (services/goals) feed into the Knowledge Base & become a part of it (Inferences can be goals or sub goals).

3.2.2 Proposed Data Model: Solution Graphs for Realization & Implementation of the Services

The graphical representation uses predicate calculus expressions in list syntax.

3.2.2.1. Solution Sub graph for Bank 1

The Solution Sub graph for Bank 1 has been furnished in Fig 5. Here the set of unifications that satisfies each sub goal is returned by the pattern_search algorithm [17]. The following unifications/substitutions take place, using the unification algorithm [17].

Impact of any change in an entity / attribute is evident through the unification / substitution process. That is, any change in an entity / attribute will impact all sub-goals, and finally the goal. For example, any change in the entity (predicate) customer will impact the following sub-goals:

- *customerX*
- *message ((bankingsystemP), (bank R), (getBalance (AccountNo))*

Therefore, in turn, the following goal will be impacted

- *displaybalance (bank1)*

Similarly, any change in the entity (predicate) bank will impact the following sub-goals:

- *message ((bank (R) (account(N) (getAccountNo))*

Security through authentication is inherent in the unification/substitution process:

{p_kumar/X}, {banksystem1/P}, {bank1/R}, {da_1001/N}, {12345/AccountNo}

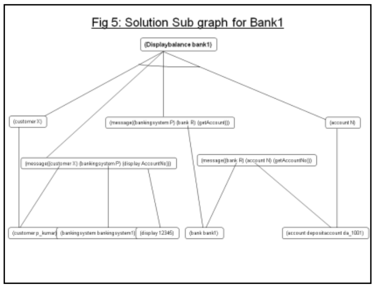
Fig 5: Solution Sub graph for Bank1
(Displaybalance bank1)
(customer X)
(message((bankingsystem P) (bank R) (getAccount()))
(account N)
(message((customer X) (bankingsystem P) (display AccountNo)))
(message((bank R) (account N) (getAccountNo()))
(customer p_kumar)
(bankingsystem bankingsystem1)
(display 12345)
(bank bank1)
(account depositaccount da_1001)

Figure 5: Solution Sub graph for Bank1

3.2.2.2. Solution Sub graph for Bank 2

The Solution Sub graph for Bank 2 has been furnished in Fig 6. The set of unifications that satisfies each sub goal is returned by the pattern_search algorithm [17]. The following unifications/substitutions take place using the unification algorithm [17].

Impact of any change in an entity / attribute is evident through the unification / substitution process. That is, any change in an entity / attribute will impact all sub-goals, and finally the goal. For example, any change in the entity (predicate) customer will impact the following sub-goals:

- *customerY*
- *message ((bankingsystemP)(bank R) (getAccount))*

Therefore, in turn, the following goal will be impacted

- *Displaybalancebank1*

Similarly, any change in the entity (predicate) bank will impact the following sub-goals:

- *message((bankR)(account)(getAccountNo))*
- *message((bankingsystemQ)(bankingsystemQ)(getAccount))*

Security through authentication is inherent in the unification/substitution process:
{p_dev/Y}, {banksystem2/Q}, {bank2/S}, {dep_1005/Z}, {2468/AccountNo}

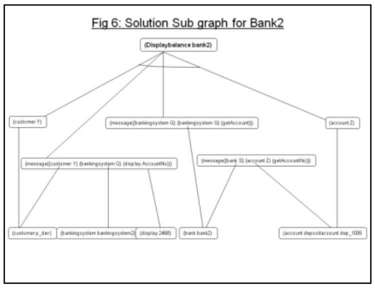
Fig 6: Solution Sub graph for Bank2
(Displaybalance bank2)
(customer Y)
(message((bankingsystem Q) (bankingsystem Q) (getAccount)))
(account Z)
(message((customer Y) (bankingsystem Q) (display AccountNo)))
(message((bank S) (account Z) (getAccountNo)))
(customer p_dev)
(bankingsystem bankingsystem2)
(display 2468)
(bank bank2)
(account depositaccount dep_1005

Figure 6: Solution Sub graph for Bank2

3.2.3 Consolidated Scenario: Process Composition

Let us look at the scenario when there is consolidation of the services, e.g. let us consider that the Consolidated Service (Goal) is the expression: displaybalance (bank_ newbank). Let us assume that that the services are realized based on the input provided by the customer.

The goal for the consolidated scenario is represented by: displaybalance (bank1) $\vee$ displaybalance (bank2) $\rightarrow$ displaybalance (bank_ newbank). From the point of view of dynamic behavior of systems as represented by Petri Nets, this is synonymous to OR-Split/Or-Join. The goal gives a valid result if the customer of the new bank is either a customer of Bank 1 OR if he is a customer of bank 2, but he is simultaneously not a customer of both the banks.

Now if the customer is a having an account in both the banks (Bank1 & Bank2), the above goal representation to evaluate to a business fault condition. However, if we have the following goal representation for the 'display balance' service, the fault condition would not occur. From the point of view of dynamic behavior of systems as represented by Petri Nets, this is synonymous to AND-Split/AND-Join.

(displaybalance (bank1) Λ displaybalance (bank2) $\rightarrow$ displaybalance (bank_ newbank)

The new service displaybalance (bank_ newbank) is an orchestration of the two services displaybalance (bank1) & displaybalance (bank2). Atomic services as well as composite services (which have been arrived at by orchestrating atomic/composite services) can be well realized by means of Goals & Sub goals.

The above services (goals) are WFF (Well Formed Formulae) in AI, having proper syntax & semantics, thus assuring quality of the composed services. Also, the inferences (services/goals) feed into the Knowledge Base (KB) & become a part of it.

The challenge lies in interoperability within and without the organization, especially with respect to the business objects that are required across business functions, to realize the capabilities of the organization.

Thus the impact analysis of business objects / data as defined by traditional description is a challenge to business sustainability of organizations. These challenges in data architecture impact analysis may be mitigated by the AI (Artificial Intelligence) paradigm, by taking recourse to the very powerful features of AI, by defining predicate calculus based knowledge bases, as outlined in Section 3.2.2.

3.2.4 Layered Architecture

The Layered Architecture diagram for our proposed AI / Knowledge Based Architecture has been furnished in Fig 7.

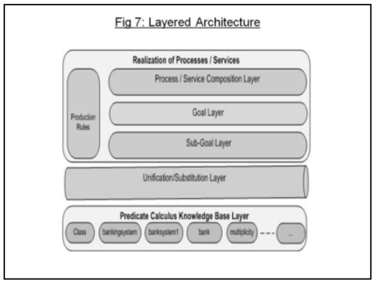
Fig 7: Layered Architecture
Realization of Processes / Services
Production Rules
Process / Service Composition Layer
Goal Layer
Sub-Goal Layer
Unification/Substitution Layer
Predicate Calculus Knowledge Base Layer
Class
bankingsystem
banksystem1
bank
multiplicity
...

Figure 7: Layered Architecture

4 Improvements Achieved Through Proposed AI Based Architecture With Respect to Business Process Composition

We discuss features that are realizable through our proposed AI based architecture, but not achievable through UML / Petri Net based modeling.

Thus the impact analysis of business objects / data as defined by traditional description is a challenge to business sustainability of organizations. These challenges in data architecture impact analysis may be mitigated by the AI (Artificial Intelligence) paradigm, by taking recourse to the very powerful features of AI, by defining predicate calculus based knowledge bases.

In our paper we consider the Banking domain for carrying out our discussions.

4.1 Knowledge Based Inferencing & Implementation Framework

The proposed architecture provides a framework for implementation of the individual & composite processes / services as goals. These goals are WFF (Well Formed Formulae) in AI, having proper syntax & semantics, thus assuring quality of the composed services. Also, the inferences (services/goals) feed into the Knowledge Base (KB) & become a part of it.

4.2 Process / Service Registry Discovery

The Knowledge Base (KB) serves the purpose of Process / Service Registry since service information would be maintained in the KB itself on the form of predicates. Service Discovery is through pattern search & substitution/unification. Service binding is implemented through substitution/unification.

4.3 Pattern Based KB

Integration patterns, Business Patterns, Workflow Patterns, other relevant patterns can be modeled / defined and maintained as predicates in the KB. The patterns can be searched and retrieved by means of Pattern Search & Unification / Substitution. This is a key strength of our proposed AI driven Architecture.

4.4 Business Intelligence

The AI driven Architecture provides Business Intelligence by way of discovery of hidden knowledge through inferences.

4.5 Quality

The proposed Architecture uses syntactically & semantically correct Well Formed Formulae (WFF) for realizing & orchestrating services thus assuring quality of the orchestrated / composed processes.

4.6 Extensibility & Performance

The Predicate Calculus based Knowledge Base can be appropriately extended as per business requirement. For example, if another Bank is merged with our New Bank, the Knowledge Base can be appropriately augmented. It would not entail any change to the Pattern Search & Unification algorithms. Here we can observe again that the proposed Architecture is highly extensible as well as flexible.

4.7 Reusability

Our proposed Architecture scores high on the reusability front as the realized services (Goals & Sub-goals) become part of the Predicate Calculus Knowledge Base; these Goals & Sub-goals can be reused from the Knowledge Base. Also, the Inferences derived from the Production Rule based Expert System can likewise be incorporated into the Knowledge Base, and reused.

5 Conclusion

Our proposed AI (Knowledge Based) based approach leads to a full-fledged Architecture. The challenges in data architecture impact analysis are mitigated by the AI (Artificial Intelligence) paradigm proposed in this paper, by taking recourse to the very powerful features of AI, by defining predicate calculus based knowledge bases.

References

[1] Goncalo Candido, Jose´ Barata, Armando Walter Colombo, Francois Jammes, SOA in reconfigurable supply chains: A research roadmap, Engineering Applications of Artificial Intelligence 22 (2009) 939–949

[2] Marco Crasso, Alejandro Zunino, Marcelo Campo, Easy web service discovery: A query-by-example approach, Science of Computer Programming 71 (2008) 144–164

[3] Omar El-Gayar, Kanchana Tandekar, An XML-based schema definition for model sharing and reuse in a distributed environment, Decision Support Systems 43 (2007) 791–808

[4] Chua Fang Fang, Lee Chien Sing, Collaborative learning using service-oriented architecture: A framework design, Knowledge-Based Systems 22 (2009) 271–274

[5] Tae-Young Kim, Sunjae Lee, Kwangsoo Kim, Cheol-Han Kim, A modeling framework for agile and interoperable virtual enterprises, Computers in Industry 57 (2006) 204–217

[6] Marcos L´opez-Sanz, Cesar J. Acuna, Carlos E. Cuesta, Esperanza Marcos, Modelling of Service-Oriented Architectures with UML, Electronic Notes in Theoretical Computer Science 194 (2008) 23–37

[7] Claus Pahl, Semantic model-driven architecting of service-based software systems, Information and Software Technology 49 (2007) 838–850

[8] Sinuhe Arroyo, Miguel-Angel Sicilia, Juan-Manuel Dodero, Choreography frameworks for business integration: Addressing heterogeneous semantics, Computers in Industry 58 (2007) 487–503

[9] David Chen, Guy Doumeingts, Francois Vernadat, Architectures for enterprise integration and interoperability: Past, present and future, Computers in Industry 59 (2008) 647–659

[10] Claudia-Melania Chituc, Americo Azevedo, Cesar Toscano, A framework proposal for seamless interoperability in a collaborative networked environment, Computers in Industry 60 (2009) 317–338

[11] Karim Baına, Khalid Benali, Claude Godart, DISCOBOLE: A service architecture for interconnecting workflow processes, Computers in Industry 57 (2006) 768–777

[12] Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal, Ming-Chien Shan, Business Process Intelligence, Computers in Industry 53 (2004) 321–343

[13] Ricardo Jardim-Goncalves, Antonio Grilo, Adolfo Steiger-Garcao, Challenging the interoperability between computers in industry with MDA and SOA, Computers in Industry 57 (2006) 679–689

[14] Harinder Jagdev, Laurentiu Vasiliu, Jim Browne, Michal Zaremba, A semantic web service environment for B2B and B2C auction applications within extended and virtual enterprises, Computers in Industry 59 (2008) 786–797

[15] Jisoo Jung, Injun Choi, Minseok Song, An integration architecture for knowledge management systems and business process management systems, Computers in Industry 58 (2007) 21–34

[16] Y. Rezgui, Role-based service-oriented implementation of a virtual enterprise: A case study in the construction sector, Computers in Industry 58 (2007) 74–86

[17] George F Luger, AI Structures and Strategies for Complex Problem Solving, Pearson Education, Fourth Edition, 2006

[18] Mark Priestley, Practical Object Oriented Design with UML, Tata McGraw-Hill Publishing Company Ltd, 2nd Edition, 2005

[19] Debasis Chanda, D Dutta Majumder, Swapan Bhattacharya, "Virtual Consolidation: A New Paradigm Of Service Oriented Distributed Architecture For Indian Banking System", Proceedings of International Conference on Emerging Applications of Information Technology, Elsevier, pp. 57-62,Kolkata, 2006

[20] Debasis Chanda, D Dutta Majumder, Swapan Bhattacharya, "Knowledge Based Service Oriented Architecture for M&A", Accepted for Publication in SEKE 2010 (The 22nd Conference on Software Engineering and Knowledge Engineering), San Francisco Bay, USA, July 1-3, 2010