Volume 2 No.6, JUNE 2011 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

A Framework to Discover New Business Processes through Business Data Integration

[1] Yih-Feng Hwang, [2] David Rine
[1,2] Department of Computer Science, George Mason University, Fairfax, VA 22030-4444
[1] yhwang1@gmu.edu , [2] DavidCRine@aol.com (drine@gmu.edu)

ABSTRACT

It is common to see that many telecommunications businesses IT-ICT systems are hosted within large data-driven telecommunications business telecommunications companies such as Verizon, after merging with former competitive companies, such as MCI. To continue providing business services to all existing customers, after completing each business merger, most existing telecommunications business IT-ICT systems from each merged company remain the same. However, to win new and sustain current customer markets among such customer data-driven competitive companies, new and improved telecommunications business processes and services from the integrated company resulting from the merger are required. However, developing new telecommunications business processes and services on top of multiple existing telecommunications business IT-ICT systems is a challenge. In this paper, we introduce a new framework that can be used to develop customer data-driven telecommunications business IT-ICT systems on top of existing telecommunications business IT-ICT systems, supporting new business processes and services. Using this framework, the existing telecommunications business IT-ICT systems will be reused at a system level, instead of focusing on reusing processes or reusing source-codes, to support new business services. This framework also provides a simpler approach to integrate the existing telecommunications business IT-ICT systems.

Keywords— customer data driven telecommunications business, integration of telecommunications information systems, integration of information systems, integration of telecommunications business systems, rule-based information distribution systems, reusable information systems, Foundations of business process models

I. INTRODUCTION

Within the global business community new telecommunications business processes and services are needed to win new customers and sustain current customers in today's competitive global market. How to develop new telecommunications business services within a desired period of time and under controlled cost becomes one of the biggest challenges.

Cost is great for a company to develop new telecommunications business IT-ICT (MB IT-ICT) systems so as to provide new customer data driven telecommunications business processes and services because existing, possibly legacy, customer data driven IT-ICT systems might be required to be part of these new telecommunications business services. Furthermore, in most scenarios, lack of communications between the existing information systems becomes one of the biggest hurdles in integrating these existing customer data driven IT-ICT systems.

Within production environments consisting of many current customer data driven IT-ICT systems, as well as legacy customer data driven IT-ICT systems, how to reuse those existing customer data driven IT-ICT systems to build new telecommunications business services, while sustaining current services, is complicated with potential cost-effectiveness along with high risk [1]. High risk is due, at least in part, to these customer data driven IT systems having different data representations, business processes, business rules, and system implementations, as well as different platforms that support these customer data driven IT-ICT systems. Different research approaches—such as Broker approach [2], Build approach [3], CASE tool approach [4], and others [5, 6, 7, 8, 9, 10, 11, 12, 28, 30, 31, 32, 33] had been proposed to address different difficulties and issues when integrating such customer data driven IT-ICT systems. The general issues during the integration addressed in previous research are as follows:

- ***System Communications*** failed because the entire integrated scopes are not defined well at the beginning of the integration [13]. The *scopes* refer to the domains of customer data driven IT-ICT systems, business policies, business processes, business rules, interface functions, and data that are going to be integrated into a unique telecommunications business software system, as depicted in Figure 1.
- ***Business Processes*** failed because of misunderstandings or missing business rules or business knowledge [14] in different customer data driven IT-ICT systems that will be integrated.
- ***Business Rules*** failed because they contain requirement faults that are caused by inconsistent, conflict, redundant, or incomplete integrated processes.
- ***Business Functions*** failed due to inconsistent, conflicted, redundant, and incomplete requirements.
- ***Data Integrity*** failed due to incorrect/duplicated data, wrong data format, incomplete/missing data,

missing link data (wrong primary key), and chained-inference effect by misleading data.

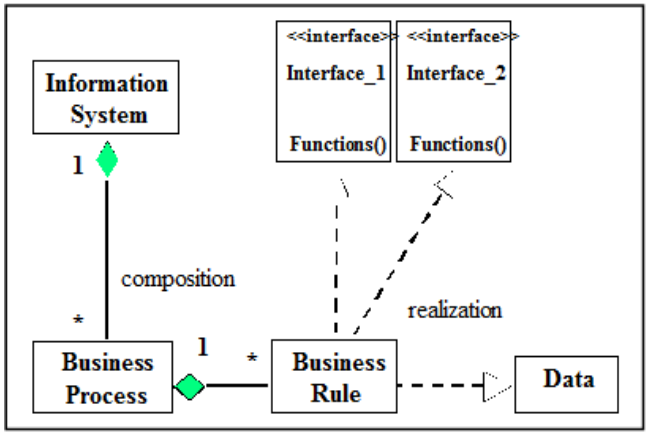

Figure 1: Relationships in a Telecommunications business Software System

Most research work in software reuse has focused on how to utilize reusable-source-code components [15], reusable objects in software components and earlier development processes [16, 17], software reuse processes [18, 19], and software reuse detection [20] in the existing software systems. None of these reuse approaches has focused on integrating two or more customer data driven IT-ICT systems.

II. RESEARCH PROBLEM

It is common to see that many customer data driven IT-ICT systems are hosted within a large Telecommunication company such as Verizon, after merging with former competitive companies, such as MCI. To continue providing business processes and services to all existing customers, after completing each merger, most existing customer data driven IT-ICT systems from each merged company remain the same.

Furthermore, it is not surprising that within such a telecommunication company, there are many customer data driven IT-ICT systems providing similar business functionalities, such as order provisioning and billing processes. These customer data driven IT-ICT systems were individually implemented in different business departments, such as long-distance service departments and local service departments. Due to the development of technologies and the impacts of using the Internet, business competition has become more complex.

To win new customer markets while sustaining existing markets among competitive companies, new telecommunications business services from such a large telecommunication company are required. However, developing new telecommunications business processes and services on top many existing customer data driven IT-ICT systems is a challenge.

In this paper, we introduce a new framework that can be used to develop customer data driven IT-ICT systems on top of many existing customer data driven IT-ICT systems to support new telecommunications business services. Using this framework, the existing customer data driven IT-ICT systems will be reused at a system level, instead of focusing on reusing processes [18,19] or reusing source-codes [20], to support new telecommunications business services. This framework also provides a simpler approach to integrate the existing customer data driven IT-ICT systems.

The rest of this paper is organized as follows. Section 2 introduces the related works in reuse and integrating of customer data driven IT-ICT systems. In Section 3, a new framework is presented providing a solution for how to reuse and integrate the existing customer data driven IT-ICT systems. Section 4 discusses a study case of using this framework. Finally, Section 5 presents the conclusions.

III. RELATED WORKS

Many kinds of approaches to integrate customer data driven IT-ICT systems have been proposed [28]. These past approaches include the following: *Conceptual model framework* [30] manages knowledge within organizational processes consisting of a set of engagements defined as sequences of interactions. Within an engagement, a broker interacts with telecommunications software agents from different telecommunications devices to perform collaborative business logic through telecommunications services [31]. Execution of telecommunications services result in one or more messages are exchanged between software agents. This conceptual model framework does not offer a solution for how to integrate with legacy systems using software agents.

Intelligent Agent-based framework [32] consists of personal agents, proxy agents, buying or selling agents, yellow page agents, and white-page agents, where buying or selling agents are telecommunications agents. A personal agent running on a user's telecommunications device allows a user to configure a mobile agent or a buying or selling agent. Other non-mobile agents are hosted in a mediator server. Mobile agents perform the tasks defined in Call for Proposal. This framework does not detail how to integrate with legacy systems through intelligent agents

Semantic Web-based Enterprise Information integration platform [33], using Ontology, semantic web [34], and rule based reasoning [35], consists of service level, semantics, mapping, and resources layers. Business services are provided to customers through the service layer. Integrating with other systems, including legacy customer data driven IT-ICT systems, through the resource layer were not discussed in detail.

Open system standards approach [21] which is based on the Application Portability Profiles (APP) and consists of 6 major functional standards—operation system, database management, data interchange, network services, user interfaces, and programming services.

Based on 6 APP [21] components, three models of *system* integration are proposed. They are organization Software Interface standards, organization Standard Services and Industry standards, and organization Standard Services and Formal Methods. However, most existing customer data driven IT-ICT systems to be integrated are not at all standardized.

Software Risk Management approach [1] measures software technical risks of system integration during the shift from hardware to software, where the role of software engineering is increasing and is becoming more central in system integration. This *Software Risk Management* approach advanced the premise that a process of risk assessment and management is also the sine qua non requirement ensuring against unwarranted time delay, cost overrun, and failure to meet performance criteria. The Hierarchical Holographic modeling (HHM) [22] consists of 7 perspectives—software development, temporal perspective, leadership, the environment, the acquisition process, quality, and technology, and is used to identify most resources of risks associated with system integration.

CASE Tool software Development methodology [4] uses two phases-- Reverse Engineering, and Forward Engineering & Data Conversion -- in software development methodology to translate system requirements into a universal form. Reverse Engineering is used to extract all relevant *business rules* information from the system to be integrated. Forward Engineering is used to change business rules to improve business processes. Data Conversion is used to utilize a CASE tool to reverse engineer legacy system records to Structure Query Language (SQL).

Modeling and simulation [23], based on *Computer-Aided Engineering* (CAE) tools, allows engineers to analysis, design, and test an object that can emulate the end of a product, such as consumer electronics and engineering workstations, at any stage of development. With a well-defined process, software components, generated by Auto-Code generator, have been employed to replace existing software modules to produce system level testing.

Structure model of integration [24] uses two different testing approaches—non-incremental testing and incremental testing. Within a non-incremental testing approach, such as Big-Bang integration testing [25], individual modules of a system are first tested separately, and then totally integrated for a complete system test. Within an incremental testing approach, such as top-down, bottom-up, critical module integration testing, and mixed integration testing, an un-tested module of a system will first be combined with a set of tested modules in one or more systems, and then the new set of modules will be tested.

Build approach [26] uses builds to facilitate a phased integration of a software system. A build is a group of related *functions* in software systems. Each new build must have regression testing with all previous builds tested. Each build could contain several constructors. Functional modules are first integrated into constructs, then from constructs into builds. Overlay functional requirements could be implemented between two different constructors.

Broker approach [2] focuses on the *data* communication between different existing systems. A broker takes care of forwarding relevant data to other systems. Each system within the Broker architecture only interfaces to the Broker with whatever formats, and does not exchange data directly with other systems, so that each existing system will be functioning as usual along with existing databases. Since there is only one communication between all systems at the data level, additional translations are required for all systems to work together due to different processes and business rules in other systems. Also, the Broker does not guarantee data consistency.

IV. A FRAMEWORK TO REUSE AND INTEGRATE CUSTOMER DATA DRIVEN IT-ICT SYSTEMS

A. The Framework

Within this paper, a new framework is introduced to reuse existing customer data driven IT-ICT systems, their business processes and business rules to support new telecommunications business services. All the existing systems will be working as they are. As shown in Figure 2, a telecommunications business process software system could be represented as 4 components—*I*nterfaces, *D*atabases, process-flow *E*ngine, and *A*gents (IDEA). The entire IDEA-based telecommunications business process software system is reusable in order to support new telecommunications business services. Furthermore, instead of building a new, large telecommunications business software system to support new telecommunications business services, the reusable IDEA framework can be used to integrate the existing customer data driven IT-ICT systems, as shown in Figure 3.

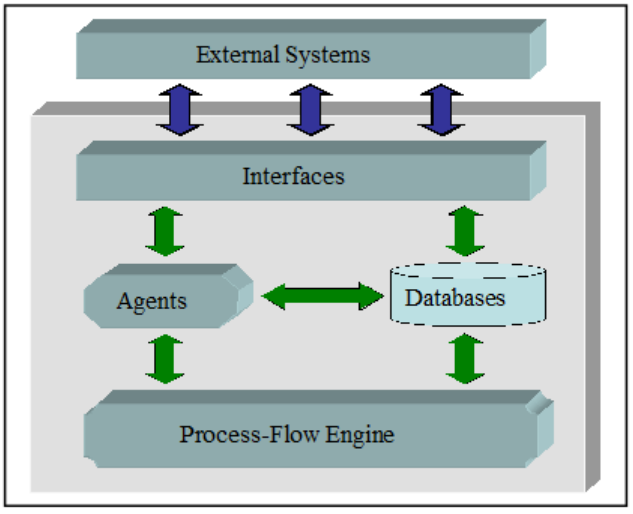

Figure 2: Reusable Framework: IDEA

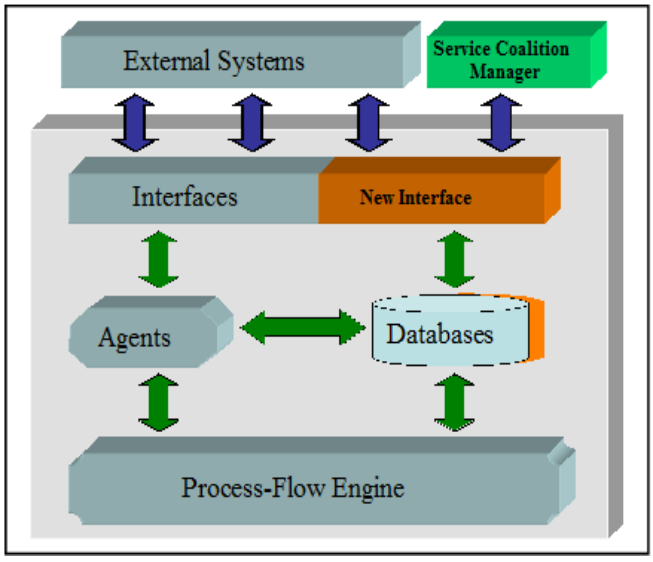

Figure 3: Integrating Framework: IDEAS

The integration approach first identifies the right customer data driven IT-ICT systems among the many the existing customer data driven IT-ICT systems, and then integrates these identified customer data driven IT-ICT systems through a Service Coalition Manager. The Service Coalition Manager functions like a global process-flow engine, which performs the providing of each service request of new telecommunications business services. In order to make all the required customer data driven IT systems work together, new telecommunications business processes as well as the corresponding business rules for new telecommunications business services will be added into the existing customer data driven IT-ICT systems. Any new telecommunications business function interacting with the external customer data driven software system will be implemented within the Interfaces component. The following section provides details for each component within the framework IDEAS.

B. Components of the Integrating Framework-IDEAS

This section details each component of the proposed framework—IDEAS. Five components are as follows:

1) Interfaces

The Interfaces layer provides a gateway from the existing customer data driven software system to one or more existing external customer data driven IT-ICT systems through different communication protocols, such as HTTP, SFTP/FTP, SOAP, TCP/UDP and/or Socket communications. Data is exchanged through this interfaces layer. Each interface within the Interfaces layer implements one or more business rules of that telecommunications business software system. If needed, an interface could use one or more agents within the Agent component to facilitate tasks computing and to improve the performance of the responses sent back to the external customer data driven IT-ICT systems.

An interface could be an ASP/HTTP/JSP page, a TCP/UDP socket program, a CORBA component, and/or SFTP/FTP UNIX script. The communication protocol has to be compatible between two or more communicating customer data driven IT-ICT systems.

Within two or more identified to-be-integrated customer data driven IT-ICT systems among many, new customer data driven processes and/or business rules will be implemented into one or more interfaces within the Interfaces layer to support new telecommunications business services. If there is no pre-required communication protocol between the existing customer data driven IT-ICT systems, then the HTTP protocol to exchange data in XML format between two customer data driven IT-ICT systems will be used.

2) Databases

Databases are used to store all necessary data records, which could be used by any interface in the *Interfaces* layer, any agent available in *Agents*, or the *Process-Flow Engine* within that telecommunications business software system. If needed, databases could be scaled to have master-slaves architecture. A *master* database is used for insert/update/delete transactions and is used to replicate all the data to all *slave* databases, which provide read-only transactions from any interface in the *Interfaces* layer, any agent available in *Agents*, or the *Process Flow Engine*.

3) Process-Flow Engine

The process-flow engine provides all the service requests based on the pre-defined business processes, consisting of a set of business rules.

There are two different processing modes—event-driven and proactive, for which a process-flow engine can be implemented. In an event-driven mode, a process-flow engine engages tasks when a pre-defined event occurs. An event could be, for instance, a change of a service status, which will trigger the process-flow engine to perform one or more business rules [27]. In the other mode—pro-active, the process-flow engine checks the current process status of a service, and then engages any internal and/or external communications, based on the pre-defined business rules, to advance the processing status of the service. A process-flow engine could be implemented in either one of two modes, or both of them, depending on the business need of a telecommunications business software system.

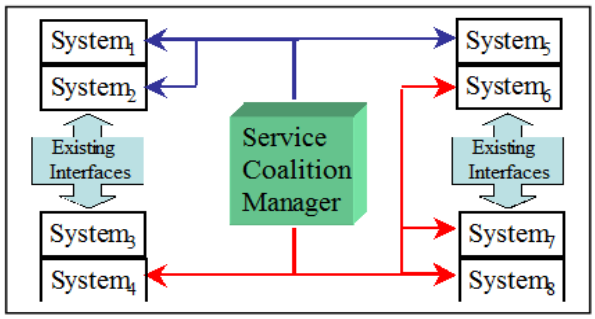

Figure 4: Service Coalition Manager with customer data driven IT-ICT systems

4) Agents

The Agents component consists of a set of shared libraries and common utilities. Each agent in the Agents component provides one or more business functions, such as calculating costs to a business service request, to other components in the IDEAS framework. An agent could also provide functions, such as formatting data, to help computing tasks. Any heavy computing task should be carried out in one or more agents, so that the process-flow engine could focus on controlling the process flow of all service requests. Each agent could be implemented as a Java bean, a set of web services, or SQL programs used to facilitate the computing tasks.

5) Service Coalition Manager

As shown in Figure 4, a Service Coalition Manager can be used to integrate the existing business services to create new customer data driven services. For instance, ordering service within an order-entrance customer data driven software system from different business departments, domestic call service and international call service could be integrated into one new order-entrance customer data driven software system, which provides customers both domestic and international telephone calls service. A Service Coalition Manager will perform the following tasks:

Coordinating the end-to-end process flow among different customer data driven IT-ICT systems for new customer data driven services.

Transforming in-and-out information to smooth communications between customer data driven IT-ICT systems involved.

Monitoring customer's service, and sending alarms, if needed, to the right supporting teams.

Providing real-time service status for each customer service request.

Providing daily statistic reports of customer services to improve business sales strategies.

Instead of creating a new customer data driven system to support new customer data driven services, a Service Coalition Manager provides a loosely coupled integration approach. The advantages of using Service Coalition Manager are as follows:

- Distributing development efforts to required customer data driven IT-ICT systems.
- Providing real-time information of new customer data driven services.
- Coordinating business process flows at the global level.
- Automating service processing to shorten processing time for customer's service requests.
- Providing a new framework with loosely coupling and less-dependency to the existing customer data driven IT-ICT systems.

V. A BUSINESS STUDY CASE OF AN IDEA PROJECT APPROACH

The business case study is based upon a prior investigation of discovering and deriving new business processes within a Verizon oriented telecommunications project. There were many customer data driven IT-ICT systems from more than 10 business departments involved in this USA based corporate communications business project, building the next generation of customer data driven software system to support new sets of business services. Within each attended business department, each Subject Matter Expert (SME) only knows the business processes and business rules within their own customer data driven IT-ICT systems.

About 80 to 90 people, including SME, technical leads, managers, and directors, joined the daily conference calls to figure out how to build the new next generation customer data driven systems from existing business processes. After spending more than 20 Millions dollars within the first 6 months, the first-phase project failed to deliver the product to production. The following are two observed fatal points causing the project failure:

1. Lack of reusing the existing customer data driven systems at a system level. Instead of developing new plug-in components on top of existing to be merged customer data driven IT-ICT systems to support new customer data driven services, code changes were embedded into the existing software systems, which added more complexity to the already complicated business processes and rules.
2. Lack of globally tracking functionality. It was no surprise that no one knows what caused an ordering provision failure, when an upper-flow process passed the wrong order information to the down stream process, where an exception is identified.

The above two fatal points often cause the wasting of much un-necessary human resources, spending countless hours to preventable issues.

A. An Example of Service Provisioning

A service request of a network access service will require several customer data driven IT-ICT systems in a telecommunication company to provide that service, before the network access service is available to a customer. First, a service contract will be issued and agreed by both the customer and the network service provider in a service contract customer data driven software system. Once the contract is approved, the network service provider will need to sequentially perform several system-provisioning steps before the network service available to the customer. These system-provisioning steps, for instance, could include as follows:

- Customer order entrance
- Network planning
- Equipment purchasing
- Logical network provisioning
- Physical provision
- Field service testing
- Service activating
- Billing process

Each system-provisioning step above is executed within a process-flow engine in a customer data driven software system. For instance, the system-provisioning step of logical network provisioning will have the following main process-flow steps:

- Received service-order information
- Received logical inventory information
- Completed circuit engineering

The process-flow engine in the logical network provisioning customer data driven software system will make sure the above three steps are completed for every single service request order.

B. Integrating New customer data driven Services into the Existing customer data driven IT-ICT Systems

Instead of building a new customer data driven software system, consisting of business processes from the existing customer data driven IT-ICT systems, to support new customer data driven services, business analysts will first identify the existing customer data driven IT-ICT systems, which will be required to support new customer data driven services.

Within each identified customer data driven IT-ICT systems, one or more new interfaces will be developed to communicate with existing external customer data driven IT-ICT systems and the Service Coalition Manager, which will perform as a global process-flow engine to the new customer data driven services. Each new process-flow step within different system-provisioning steps will be tracked in the Service Coalition Manager, which provides real-time end-to-end service status.

Within each impacted customer data driven software system, that customer data driven software system's database will be modified to store new data records to support new customer data driven services. One or more agents will be developed to support any new computing tasks. The process-flow engine will implement new customer data driven processes to support the new customer data driven rules. All the changes to the any one of four components (Interfaces, Databases, Agents, and process-flow Engine) supporting new customer data driven services, will be developed as new plug-in components to the existing customer data driven IT-ICT systems. Each plug-in component will be removed anytime, if needed, without any impacts to other existing customer data driven software system.

VI. CONCLUSIONS

To win new customer markets and sustain existing markets among competitive companies, new customer data driven business processes and services from a large integrated telecommunication company are required. However, developing new customer data driven services on top of many existing customer data driven IT-ICT systems is a challenge.

In this paper, a new reusable framework—IDEA, is introduced to represent an integrated customer data driven business process software system. Data communications from the IDEA-based customer data driven software system to any external system will go through the Interfaces component. The Agents component provides a set of shared libraries and common functionalities available to other components within the IDEA-based integrated customer data driven software system. The process-flow engine provides every service request based on either a set of pre-defined events or service statuses stored within the Databases component.

A new integration framework—IDEAS, utilizes a Service Coalition Manager to bring the existing systems together to provide new customer data driven services without impacting the existing working business processes. Changes used to support the new customer data driven services will be implemented as plug-in components, which could be removed anytime, if needed, without impacting the existing functionalities of the existing customer data driven IT-ICT systems.

REFERENCES

[1] Chittister, Y. Haimes, "System Integration via Software Risk Management", *IEEE Trans on Systems, Man, and Cybernetics*, vol. 26, no. 5. pp. 521-532, 1996.

[2] D. Clark, J. Krumm, "Broker: A System Integration Approach", *in Proc. of the 2nd International*

Conference on Systems Integration, 1992, pp. 162-170.

[3] D. Schultz, "A Case Study in System Integration Using The Build Approach", *in Proc. ACM annual conference*, 1979, pp. 143-151.

[4] R. Brown, R. Stan, "Systems Integration Using CASE Tool Development Software", in *Proc. IEEE National Aerospace and Electronics Conference*, vol. 2, 1996, pp. 818-824.

[5] A. Berztiss, "Domain Analysis for Business Software Systems", *Information software systems*, Elsevier Science, vol. 24, no. 7, pp. 555-568, 1999.

[6] W. Fan, H. Lu, S. Madnick, D. Cheung, "Discovering and Reconciling Value Conflicts for Numerical Data Integration", *Information software systems*, Elsevier Science, vol. 26, no. 8, pp. 635-656, 2001.

[7] J. Ganesh; S. Padmabhun; D. Moitra, "Web services and multi-channel integration: a proposed framework", in *Proc. IEEE International Conference on Web Services*, 2004, pp. 70-77.

[8] P. Green, M. Rosemann, "Integrated Process Modeling: An Ontological Evaluation", *Information software systems*, Elsevier Science, vol. 25, no. 2, pp. 73-87, 2000.

[9] P. Huang, P. Kar, C. Fandrich, "Using Modeling and Simulation for Rapid Prototyping and System Integration", in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, 1997, pp. 2812-2817.

[10] M. Pavlic, A. Kaic, Z. Dobrovic, S. Candrlic, J. Ostojic, "Design phase of information software systems integration process", in *Proc. IEEE International Conference on Information Technology Interfaces*, 2003, pp. 143-148.

[11] V. Uraikul, C.W. Chan, Tontiwachwuthikul, P., "An architectural framework for a distributed process control information software system", in *Proc. IEEE Conference on Electrical and Computer Engineering*, vol. 3, 2004, pp. 1375-1378.

[12] R. Wieringa, E. Dubois, "Integrating Semi-Formal and Formal Software Specification Techniques", *Information software systems*, Elsevier Science, vol. 23, no. 3-4, pp. 159-178, 1998.

[13] Y. Hwang and D. Rine, "Verification framework and algorithms for integrating information distribution systems", *Information and Software Technology*, vol. 48, no. 9, pp. 876-889, Sep. 2006.

[14] G. PATRIOTTA, "On studying organizational knowledge", *Journal of Knowledge Management Research and Practice*, vol. 2, no. 1, pp. 3-12, 2004.

[15] R. Selby, "Enabling reuse-based software development of large-scale systems", *IEEE Trans on Software Engineering*, vol. 31, no.6, pp. 495-510, June 2005.

[16] Y. Liu, A. Yang, "Research and Application of Software-reuse", *in Proc. ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, vol. 3, Jul. 30-Aug. 1, 2007, pp. 588-593.

[17] M. Hull, P. Nicholl, Y. Bi, "Approaches to component technologies for software reuse of legacy systems", *Computing & Control Engineering Journal*, vol. 12, no. 6, pp. 281-287, Dec. 2001.

[18] E. de Almeida, A. Alvaro, D. Lucredio, V. Garcia, S. de Lemos Meira, "A survey on software reuse processes", in *Proc. IEEE International Conference on Information Reuse and Integration*, Aug. 15-17, 2005, pp. 66-71.

[19] M. Laguna, B. Gonzalez-Baixauli, O. Lopez, F. Garcia, "Introducing systematic reuse in mainstream software process", in *Proc. of Euromicro Conference*, Sept. 1-6, 2003, pp. 351-358.

[20] W. Qu, M. Jiang, Y. Jia, "Software Reuse Detection Using an Integrated Space-Logic Domain Model", in *Proc. IEEE International Conference on Information Reuse and Integration*, Aug. 13-15, 2007, pp. 638-643.

[21] D. Kuhn, "On the Effective Use of Software Standards in Systems Integration", in *Proc. International Conference on Systems Integration*, 1990, pp. 455-461.

[22] Y. Haime, "Hierarchical Holographic modeling", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 606-618, 1981.

[23] P. Huang, P. Kar, C. Fandrich, "Using Modeling and Simulation for Rapid Prototyping and System Integration", in Proc. *IEEE International Conference on Systems, Man, and Cybernetics*, 1997, pp. 2812-2817.

[24] C. Ramamoorthy, "Distributed Techniques in Software Systems Integration", in *Proc. IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems*, 1995, pp. 252-256.

[25] G. Myers, "Software Reliability", John Wiley, 1976.

[26] D. Schultz, "A Case Study in System Integration Using The Build Approach", in *Proc. ACM annual conference*, 1979, pp. 143-151.

[27] Y. Hwang, D. Rine, "Algorithms to detect chained-inference faults in Information Distribution Systems", in *Proc. ACM Symposium on Applied Computing*, Las Vegas, NV, Mar. 11-14, 2001, pp. 679-683.

[28] Y. Hwang, D. Rine, "Verification framework and algorithms for integrating information distribution systems", *Information and Software Technology* Volume 48, Issue 9, September 2006, Pages 876-889

[29] Y. Hwang, "Integration of Information Distribution Systems", USA Patent Number 7707548, April 27, 2010.

[30] I. Hawryszkiewycz, R. Steele, "A Framework for Integrating Mobility into Collaborative Business Processes", in *Proc. IEEE International Conference on Mobile Business*, 2005, pp. 89-93.

[31] I. Hawryszkiewycz, R. Steele, "Service Oriented Design Methodologies for Mobile Applications", in *Proc. IEEE International Conference on Mobile Business*, 2006.

[32] Z. Weng, T. Tran, "An Intelligent Agent-Based Framework for Mobile Business", in *Proc. IEEE International Conference on Management of Mobile Business*, 2007.

[33] Z. Wei, M. Kang, W. Zhou, "A Semantic Web-Based Enterprise Information Integration Platform for Mobile Commerce", in *Proc. IEEE International Conference on Management of e-Commerce and e-Government*, 2008, pp. 57-60.

[34] J. Davies, D. Fensel, H. F. Van, "Towards the semantic web: ontology-driven knowledge management", Chichester, England; Hoboken, NJ, John Wiley, 2003.

[35] M. Negnevitsky, "Artificial Intelligence: A Guide to Intelligent Systems", 2nd ed., Addison-Wesley, 2005, pp. 25-55.