

Revisiting Bpr: A New Framework & Model For Future

Prasenjit Kundu

Research Scholar

Department of Computer Science

Utkal University

Bikram Kesari Ratha

Faculty Member

Department of Computer Science

Utkal University

Debabrata Das

Lecturer

Bengal School Of

Technology & Management

IJERT

Abstract

Business Process Reengineering is all about fundamental reconsideration and rethinking of different processes associated with a business & redesigning these processes to obtain dramatic, drastic and sustainable change and improvement in quality and service with reduction in cost. The concept of BPR was advocated by Michael Hammer in 1990 and since then BPR has become an important area for conceptual and empirical research. Over the years with the rapid growth in information and communication technology (ICT), the application of ICT in BPR has taken the centre stage of contemporary BPR research. Researchers developed numerous models for analyzing, interpreting and implementing BPR and among those models the Object Oriented Models (OOM) and the Knowledge Based Models (KBM) rely heavily on the applications of ICT in BPR and that is why these two models are the most interesting and important from the researchers' point of view. Service oriented architecture (SOA) is a framework to integrate business processes and supporting IT infrastructures into secure, standardized components services that can be reused and combined to address changing business activities and priorities. Dynamic program analysis is the analysis of computer software which is performed by executing programs built from that software system to predict the behaviour of the system as well as to fine tune performance. This paper proposes a mixed model of BPR by combining the OOM and KBM and calls it the Object Oriented Knowledge Based Model or OKB Model. The OKB model first identifies business processes at the top or strategic level, middle or supervisory level and bottom or operational level and then breaks down these processes into repetitive activities. Next, the OKB model converts these repetitive activities into services as per the SOA framework. Once an enterprise wide SOA implementation blueprint is ready, dynamic program instrumentation techniques are used to fine tune, optimize and reverse engineer the existing legacy systems across the organization. The OKB model is unique in the sense that it uses the concept of SOA and dynamic program instrumentation, i.e.; it combines the principles of organizational reengineering with the tools of ICT and thus builds an effective, easy to understand and easy to implement framework for business process reengineering.

1. Introduction

Research on BPR has a long history and researchers across the disciplines of business management and software engineering contributed heavily in this field and thus the concepts and

principles of BPR has become an independent field of study now a days. Despite the maturity of the discipline, scope of developing models and framework in the field of BPR is still growing with the growing use of the tools and techniques of information and communication technology as well as the increasing competitiveness in the global business arena. The object oriented models check the process flow analysis (PFA) of the worn out, error prone or obsolete manufacturing systems and derive the minimum essential information (MEI) requirements to identify the data sources, information destinations and timing requirements to improve the communication between all human actors, machines and computers and ultimately modify the process flow analysis (PFA). This modified process flow analysis (PFA) is called Information Process Flow Analysis (IPFA). Knowledge Based Models (KBM) support next generation dynamic business environments by using the concepts of Artificial Intelligence (AI), Expert Systems (ES) etc. and help in system migration, organizational reengineering and heterogeneous system interoperation. Future belongs to the proper integration of ideas from both the business management and information and communication technology areas and this paper is one such attempt where a perfect integration has been done by combining two different existing models of BPR with the help of the concept of Service Oriented Architecture (SOA) and dynamic program analysis and plug in instrumentation to create a new model of BPR which is versatile, easily conceivable and future ready.

2. Literature Review

Research on BPR and enterprise modeling started with the emergence of global competitiveness in developing high quality low cost products and this led to the development of various approaches like CIM, JIT, lean manufacturing, concurrent engineering etc ([1], [2], [3], [4], [5], [6], [7], [8]). The first thought on reengineering manufacturing systems and connecting BPR with enterprise modeling was proposed by J.H. Manley. Manley describes ways through which industrial engineers can assist in reengineering worn out, error prone or obsolescent real-time manufacturing systems (embedded systems) by helping computer systems and also how the communication engineers ensure that critical information control loops, both feed forward and feedback, are complete and efficient. Manley used two conceptual models, the Embedded Computer System (ECS) physical model and the Object Transformation Process Model (OTPM) to guide a modified process flow analysis (PFA) of existing large-scale, complex embedded systems that takes into account the

process-supporting information. According to Manley this modified PFA is called an Information Process Flow Analysis (IPFA) [1]. Kenneth C Hoffman presents a general model of the management structure to implement a systems integration program, using an enterprise-wide Information Systems Architecture (ISA) as a roadmap, which is supported by a defined set of measures and metrics. This Information System Architecture (ISA) approach is comprehensive in covering application software and data architecture and also the computing and communications hardware infrastructure and other automation technologies that support the overall business process [2]. Based on this two early approaches of enterprise modeling, J. H. Manley developed his marvelous idea of a three-phase information system analysis and design methodology. This approach can be used to continuously improve enterprise information systems as part of a six-step annual business improvement process. According to this approach, after the senior management's strategic decisions on next year's product and/or service portfolio content, the related financial, management, engineering, and quality improvement processes are analyzed to determine their output product and/or service quality and timeliness. Subsequently, facilities, equipment, and personnel resources required for individual processes are inspected for possible immediate or future improvement. Then the minimum essential information (MEI) requirements are analyzed using the Object Transformation Process Model (OTPM). Then, individual OTPM models are linked to help identify all pertinent data sources, information destinations, and timing requirements and lastly the linked OTPM models are mapped onto an Embedded Computer System (ECS) model that defines a physical architecture for improving telecommunication paths between all humans, machines and embedded computers that are component parts of the integrated processes. According to Manley, this approach yields comprehensive information system logical and physical architectural models that can recursively guide high-leverage enterprise-wide improvement projects over succeeding fiscal years [3].

3. The Object Oriented Knowledge Based (OKB) Model

The proposed Object Oriented Knowledge Based (OKB) model has eleven steps and the steps are as follows :(ref. Fig. 1)

Step I: Departmentation of the Organizations.

In this step the hybrid approach of Departmentation should be used. Every organization structure has two dimensions i.e. one horizontal and the other

vertical. The horizontal aspect refers to the grouping of activities while vertical dimension is the hierarchy of superiors and subordinates. Grouping of activities is an essential step in designing an organizational structure.

The following pattern may be used for grouping of activities into departments i.e.

- Departmentation by functions
- Departmentation by product
- Departmentation by territory
- Departmentation by customers
- Departmentation by process or equipments
- Departmentation by process or equipments

In practice, no single pattern is ideal to suit all situations. Therefore, no single basis is followed for grouping of activities. Rather, most of the big companies follow a composite or combination of several structures ([9], [10]).

Step II: Identification of the process at three levels of management i.e. at the top or strategic level, at the middle or tactical level and at the bottom or operational level.

At the strategic level, the primary importance is on planning and goal setting or defining objectives or determining ways to fix the vision of the company.

At the middle or tactical level, the primary objective is to ensure the proper execution of the work as per the objective set by the top management as well as to provide advisory or tactical support to the operational level.

At the bottom or operational level, the primary objective is proper execution of work or proper implementation of the instructions given by the upper levels ([11], [12]).

Step III: Classifications of the processes into a) Core Process b) Associated Process and c) Auxiliary Process.

Researchers have attempted to classify and integrate business processes from different perspectives ([13], [14], [15], [16], [17]). Researchers of Content Based Process Modeling

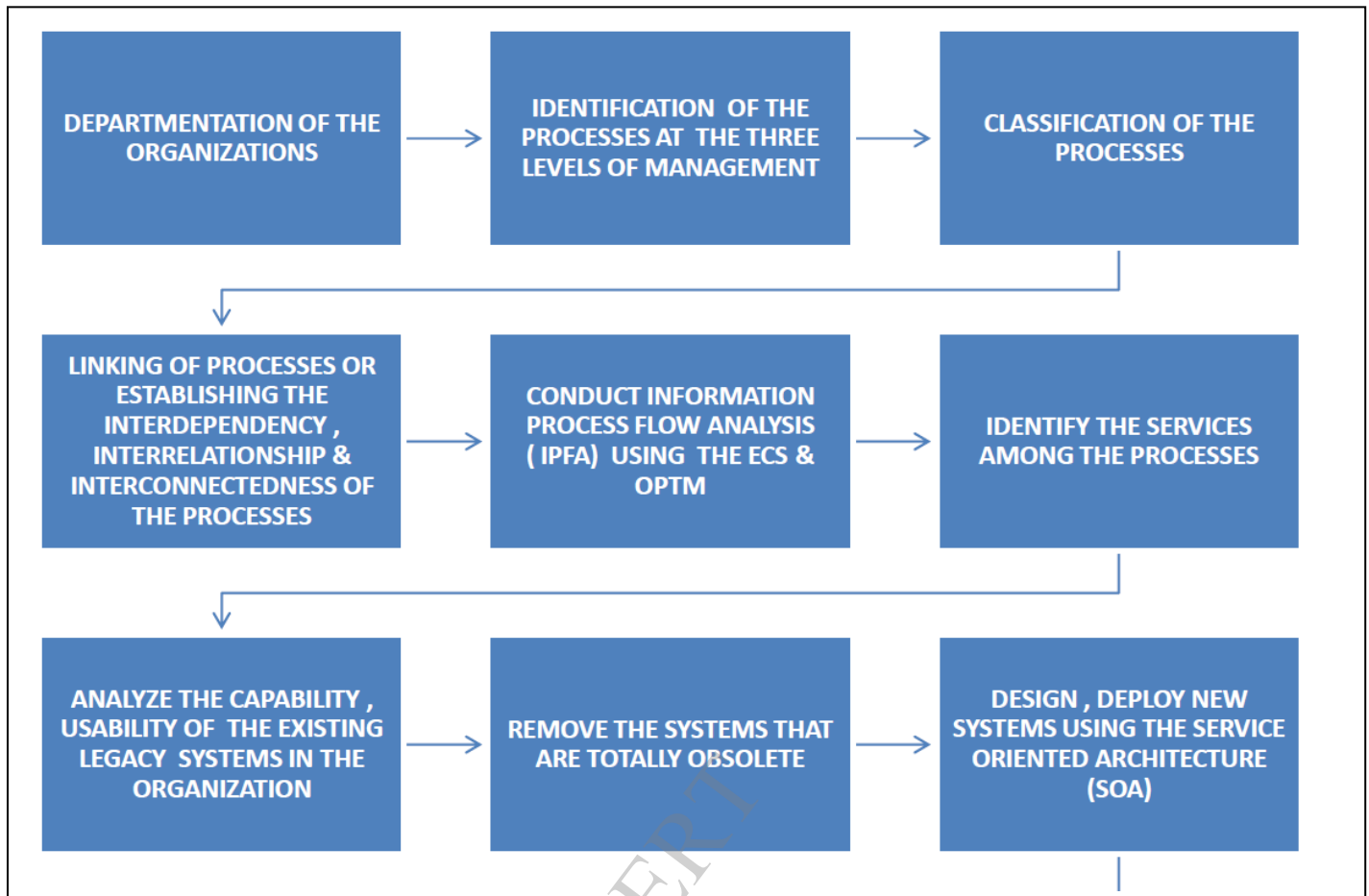


Fig.1. The Object Oriented Knowledge Based Model

(CBPM) propose that processes should be classified in to three categories namely major processes, main processes and business processes. Business processes denote processes at the elementary level and these are the sub set of main processes. In the same way, the main processes denote processes at the middle level and these are the sub set of major processes. Lastly the major processes are the processes at the top level. This way of classifying processes is based on (1) the principle of separability i.e. a particular business process is classified under one main process only and a specific main process is classified under one major process only; and (2) the principle of additivity i.e. as functionalities and processes are separable, a model can be formed from the conjunction of major processes, thus main processes and thus business processes and corresponding activity flows [13]. Mili et al propose a business process metamodel in which the activities of a business process are performed by actors playing particular roles. Activities may be initiated by events and may in turn generate events

of their own. The activities of a process can be linked through resource dependencies or control dependencies. The actors work within the context of organizational boundaries. Organizations perform specific business functions and roles can support functions. According to this meta model, process can be represented through four distinct views in the following manner: -

- (1) The functional view: It denotes the functional dependencies between the process elements.
- (2) The Dynamic view: It gives sequencing and control information about the process i.e. when certain activities are needed to be performed and how.
- (3) The informational view: It contains the description of the entities that are produced, consumed or otherwise manipulated by the process.
- (4) The organizational view: It defines who performs each task or function and where in the organization. [14].

The proposed OKB model takes a hybrid approach in classifying processes across the organization. This hybrid approach classifies processes from the functionality point of view as well as from the inter dependency among the processes point of view. It also takes into account the sequence of the processes in the organization and the importance of the processes in producing goods or rendering services. According to this approach the processes should classify into three categories viz (1) The Core Processes (2) The Associated Processes and (3) The Auxiliary Processes.

The Core processes are the processes which take part directly in the production of goods or rendering of services. The Associated processes are the processes which support the core processes but do not take part directly in production or rendering of services. The Auxiliary processes are the processes which are redundant in nature i.e. these processes are performed or executed occasionally when situation arises i.e. these processes are not directly related to production or rendering of services but these processes help the core processes when situation arises.

Step IV: Linking of processes or establishing the interdependency, Inter relationship and Inter connectedness among the processes.

Researchers of business process modeling strongly believe that only structural transformation of business processes can make the processes future ready i.e. more efficient, effective and flexible [10]. Structural transformation requires seamless integration of processes through the use of information and communication technology [21]. According to Neiger & Churilov, goal oriented business process modeling aims to extend traditional business process modeling methodologies and tools that address the how of the business process concerned with the efficient execution of the business processes to also include the way to ensure the effectiveness of business processes. This is achieved by developing goal models that describe business goals and relationships between them and by using business goals to drive process decomposition.

However existing business process modeling tools fails to address the effectiveness and efficiency concerns in an integrated manner. [20]. Rosa et al proposes a novel concept of linking domain models and process models for reference model configuration. This approach is based on the fact that repetitive business processes like purchasing, recruitment, customer services are basically similar in nature across the firms and so to promote the reuse of these processes, enterprise system vendors have developed generic reference process models which uses notations like Configurable Event Driven process Chains and sometimes this adds to

the complexity of the models. The approach adopted by Rosa et al is unique in nature because it allows users to configure reference process models independently of the process modeling notation employed[19]. According to Back et al, business process transformation requires conversion of business processes into knowledge processes. Once the enterprise wide knowledge processes can be developed, then it would be easier to link the processes through the knowledge networks [22].

Step V: Conduct Information Process Flow Analysis (IPFA) using the ECS and OPTM

Author names and affiliations are to be centred beneath the title and printed in Times 12-point. This process is concerned with the removal of bottlenecks and optimization. An embedded computer system (ECS) is distinguished from an automatic data processing system by how it is developed, acquired and operated. An ECS has three main features:

An ECS is physically incorporated into a larger system with a primary function that is not data processing.

An ECS is an integral part of a larger system from a design, procurement and operations viewpoint.

An output of an ECS usually includes system performance information, control signals and computer data [23]. The Object Transformation Process Model (OTPM) is a field-tested process modeling tool for re-engineering enterprise-wide information system architectures in support of the "new manufacturing paradigm." The OTPM assists conflict reduction between cost accounting and engineering nonfinancial measurement systems. This tool also improves "smart" product manufacturing through similar treatment of hardware and software component information. The OTPM offers a novel approach to complementing traditional industrial engineering process flow analysis, and supports the so-called "vanilla" approach to software modeling [24].

Step VI: Identify the services among the process.

Step VII: Analyze the capability, usability of the existing legacy systems in the organization.

In this step we need to classify the systems in terms of capability and usability and we must determine whether (1) the systems are totally obsolete, (2) partly obsolete but can be reengineered or (3) new, usable and future ready.

Step VIII: Remove the systems that are totally obsolete.

Step IX: Design, Deploy New Systems using Service Oriented Architecture (SOA).

Step X: Reengineer the partly obsolete systems using dynamic program instrumentation.

Step XI: Integrate the new systems with the reengineered one.

The last six steps of the OKB model is slightly different from their predecessors because these six steps together in an integrated manner denote a particular process having four sequential phases ANALYZE , DESIGN , REENGINEER AND INTEGRATE and as per the name of the phases the last six steps of the OKB model can be termed as ADRI approach.

4. The ADRI Approach

The ADRI approach has been conceptualized with the single objective of combining the concept of services oriented architecture (SOA) and dynamic program analysis to produce a hybrid approach of system analysis, system design, system reengineering and system integration. ADRI is unique just because of its capability of performing all the four tasks of system analysis , system design , system reengineering and system integration in order to fulfill the ultimate objective of OKB model i.e. business process reengineering. ADRI approach has six steps and four phases i.e. Analysis, Design, Reengineer and Integrate. All the six steps are sequential in nature and these steps are needed to be performed in the four phases(Ref. Fig. 2).The first phase has three steps and the rest of the three phases have one step each. Fig. 3 denotes the relationship between the phases and steps of ADRI approach. As the ADRI approach proposes to combine the concept of SOA and dynamic program instrumentation it is better to highlight the elementary concept of SOA and dynamic program analysis before actually explaining the steps and phases and working principle of ADRI approach. Recent research on reengineering CRM software by researchers shows the practical applicability of the ADRI approach. In this paper the ADRI approach has been explained with the help of the result of this research. Service oriented architecture can be defined as a framework to integrate business processes and supporting IT infrastructures into secure, standardized components services that can be reused and combined to address changing business activities and priorities[25].For the last couple of years different organizations across different sectors are adopting SOA just because of its capability to ensure business process improvement and thus in this way to gain competitive advantage[26].The reason behind strong adaptation of SOA is its link between IT and

business and through SOA, services which require human interaction can be configured easily and systematically so that the whole system operates as per the operational needs. For this reason SOA reference architecture has become a clear framework for any enterprise operation [27].According to Newcomer and Lomow , SOA should be seen as a style of design that provides guidance of creating and using business services throughout their lifecycle as well as defines and makes provisions for the IT infrastructure that allows different applications to exchange data and participate in business processes seamlessly irrespective of the operating systems or programming languages underlying those applications. The elementary SOA ingredients are (1) Infrastructure,(2) Architecture (3) Process and (4) Governance[28]. According to Kraf-zig, Banke and Slama, SOA is based on four key abstractions: (1) Application Front Ends (2) Service (3) Service Repository and (4) Service Bus or Enterprise Service Bus (ESB).Actually the Application Frontend is the owner of the business process while the services provide business functionality which the application frontends and other services can use. A service consists of an implementation that contains business logic and data, a service contract and a service interface. The service contract specifies the functionality, usage and constraints for a client of the service and a service interface physically exposes the functionality. The service repository stores the service contracts of the individual services of an SOA and the service bus (ESB) provide the linkage between the application front ends and services. A client can either be an application front end or service. It is always an application front end that initiates a business process and receives the result. Service is a software component of unique functional meaning that typically encapsulates a high level business concept. The application frontend basically interacts with the human user and that's why it is called sometimes the client. In some cases the client can be the service itself. The client uses the interfaces to invoke the service through the ESB

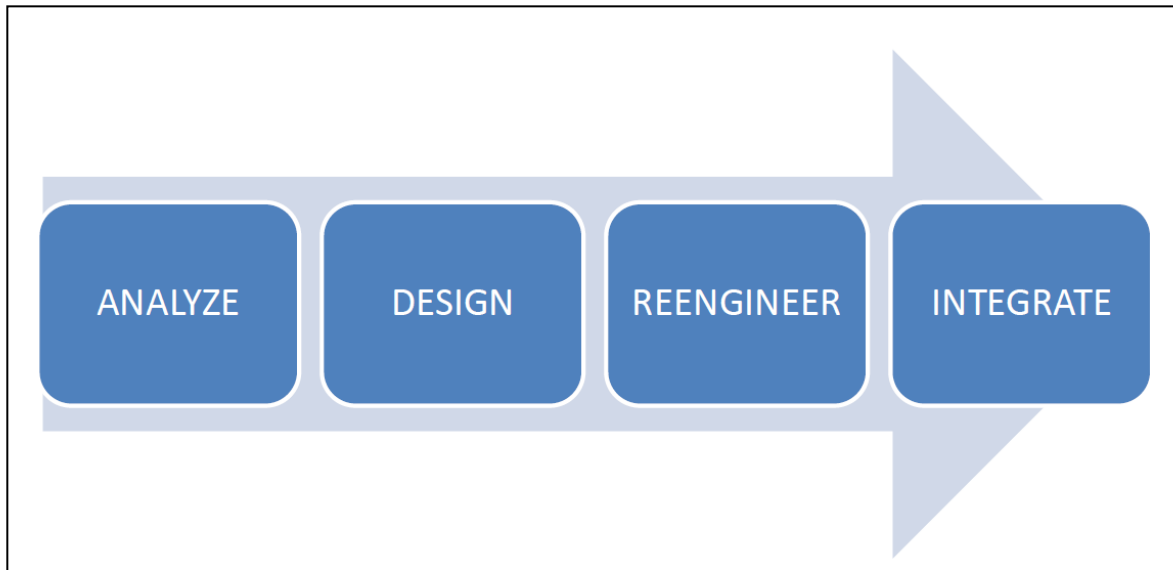


Fig.2. The ADRI Approach

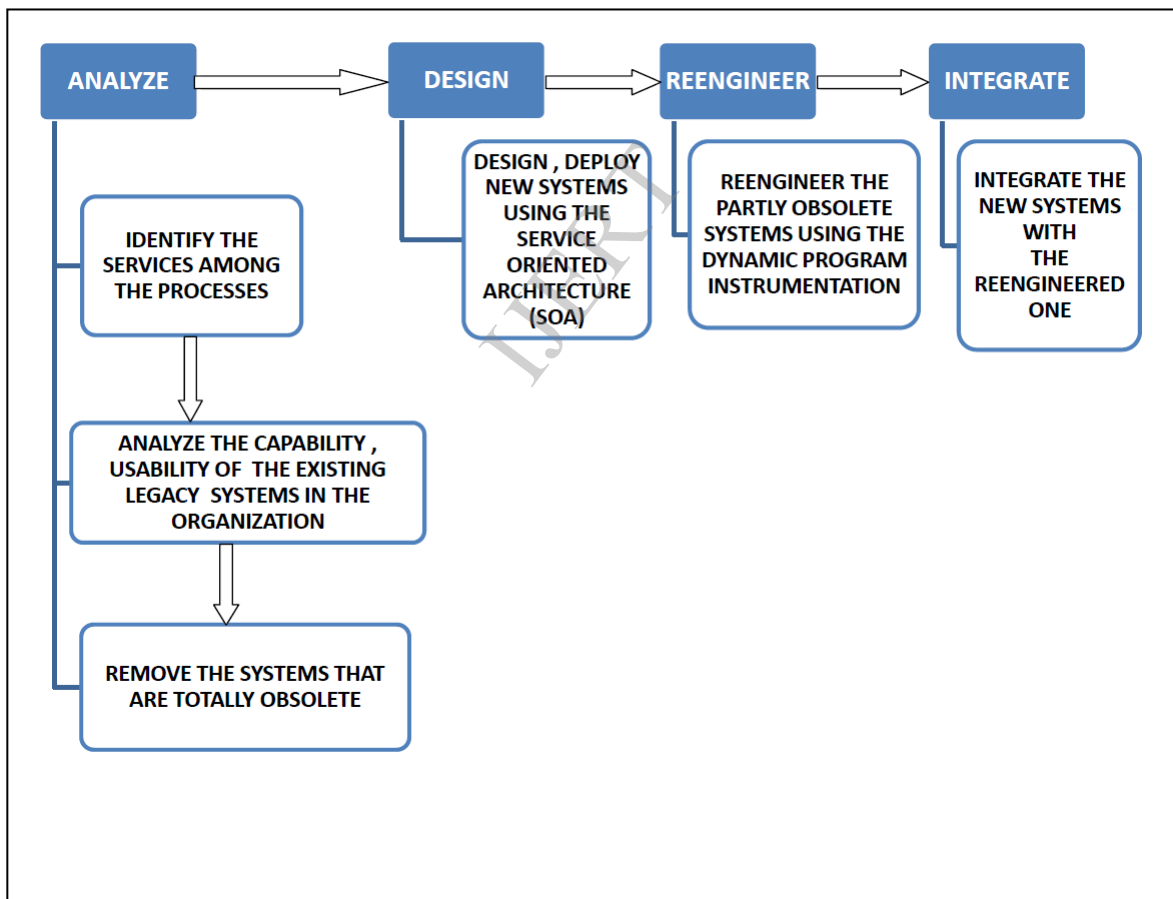


Fig.3. The relationship between the phases and steps of ADRI approach.

(enterprise Service Bus).The interface is based on the service contract and the service contract describes the service and the service fulfills the service contracts and the service contract is stored in service repository [29].This framework can

finally provide a basis for end user build front ends & applications developed according to Web 2.0 service definitions, such as mashups and the quick deployment of user oriented applications created in AJAX or similar development tool [27]. Dynamic program analysis is the analysis of computer software that is performed by executing programs built from that software system on a real time basis. For dynamic program analysis to be effective, the target program must be executed with sufficient test inputs to extract and produce interesting behaviour. Dynamic program analysis helps to make a computational system reason automatically (or at least with little human assistance) about the behaviour of a program and draws conclusions that are useful to help the software developers to determine exploitability of vulnerabilities or to rapidly develop an exploit code [30]. Dynamic analysis produces output, or feeds into a subsequent analysis, that enables human understanding of the code and makes the design and testing task easy for the developers. Dynamic program analysis approach attempts to tune the application software during execution without stopping, recompiling or even rerunning the application. To achieve this objective it is necessary to use dynamic instrumentation techniques that allow the modification of the application code on the fly [31]. Program instrumentation is a general way to understand what an executing program is doing [32]. The principle of dynamic program instrumentation involves deferring program instrumentation until it is in execution and then inserts, alter and delete this instrumentation dynamically during the actual program execution. The Paradyn group at the University of Wisconsin and University of Maryland first used this approach to develop a special API that supports dynamic instrumentation and the result of their work was called DynInst API. DynInst is an API for runtime code patching that provides a C++ class library for machine independent program instrumentation during application execution. It allows attaching to an already running process or starting a new process, creating a new piece of code and finally inserting created code into the running process. The next time the instrumented program executes the modified block of code i.e. the new code is executed and the program being modified is able to continue its execution and does not require to be recompiled, re linked or restarted [32]. Researchers have already proposed to develop CRM processes as services using the concept of Service Oriented Architecture (SOA) [36].

4.1 The Analysis Phase

As integration of SOA with legacy system is always a challenging job, it is better to make a preliminary analysis of identifying the applications

which are needed to be integrated with SOA. The process of SOA-enabling all applications in a single phase would be a mistake because it would make the integration process more complex, considerably reducing the likelihood of success [46].

4.2 The Design Phase

Das and Kundu developed a conceptual model for implementing a CRM model named PREMASA Model through SOA approach [37]. This conceptual model is based on the approach adopted by Kraf-zig, Banke and Slama [29]. Das and Kundu tried to implement PREMASA Model from system level point of view and that's why they recommend that the first three phases of the PREMASA Model are the target phases which are needed to be streamlined and implemented as part of the CRM software. According to them the activities of these three phases can be transformed and grouped into five different services [37]. Researchers faced difficulty in using Web Services effectively because there was no mechanism to describe the web services. In response to the urgent need of describing various characteristics of a web service unambiguously WSDL was created [38]. WSDL or Web Services Definition Language can be defined as a specification for describing the methods or data types of SOAP interface [39]. Taking cue from the Das and Kundu's approach in the paper "Application of Service Oriented Architecture (SOA) in Implementing a CRM Process: A Conceptual Approach", Kundu, Das and Ratha further described and detailed the five services using WSDL specification [40]. The way of specifying services using WSDL is based on the methods described by Ethan Cerami in the book "Web Services Essentials: Distributed Applications with XML-RPC, SOAP, and UDDI & WSDL" [41]. Kundu, Das and Ratha's approach is seminal because they proposed the roadmap and the actual WSDL codes to design services as per the SOA specifications and their work is vital for the design phase of the ADRI approach.

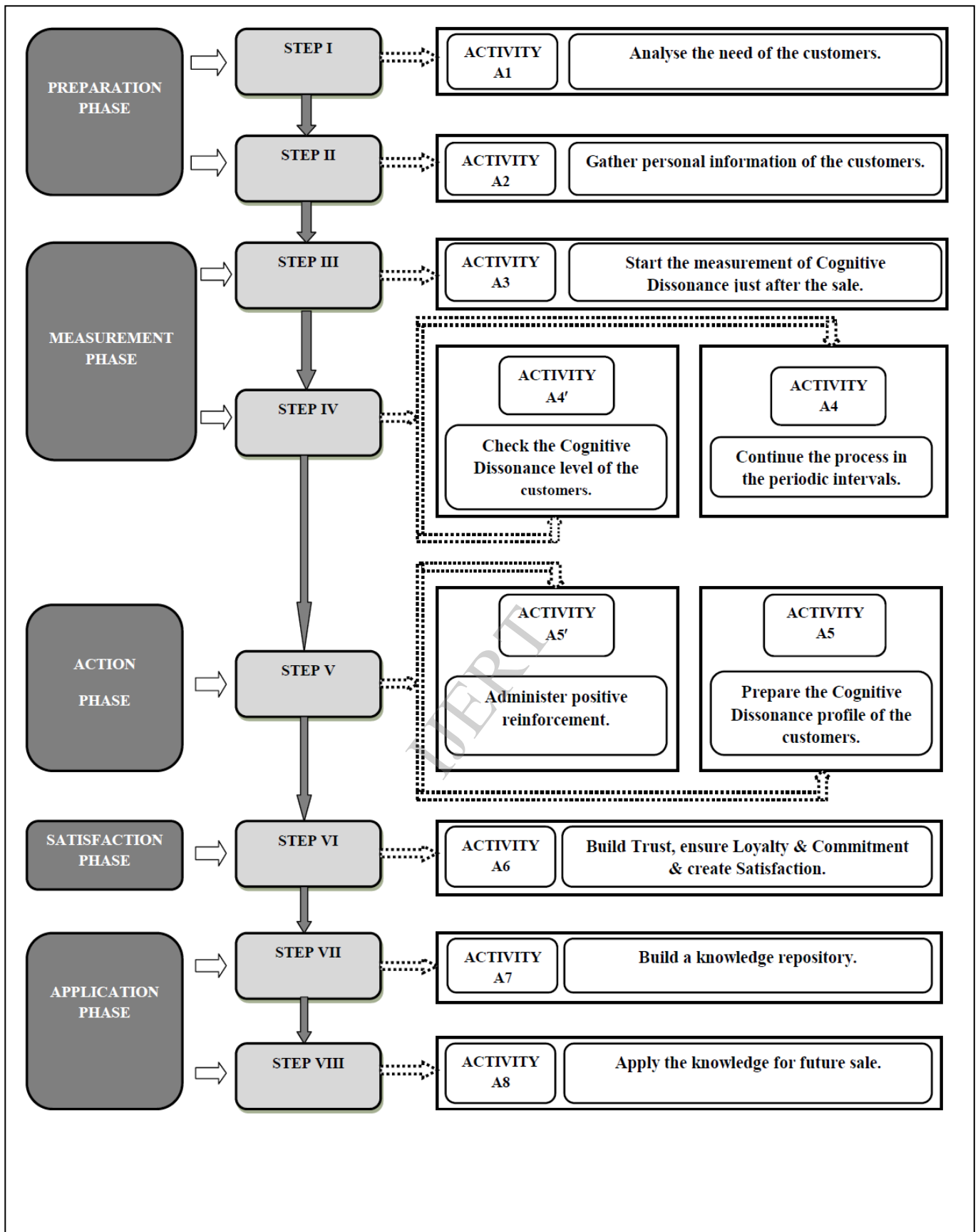


Fig.4. The interrelationship between the activities & steps of the PREMASA MODEL.

4.3 The Reengineer Phase

Recent research on reengineering CRM software by researchers shows the practical applicability of the

ADRI approach. In this paper the ADRI approach has been explained with the help of the result of this research. Das's seminal work on developing a customer relationship management model has resulted the creation of PREMASA model which can bring paradigm shift in the area of CRM research. PREMASA model proposes total eight steps. Some of these steps have only one activity & some steps have several activities. Out of these ten activities some are needed to be performed simultaneously & some are needed to be performed sequentially. The model is multi dimensional in nature because some of the activities are simultaneous & some of these activities are sequential. The model integrates these sequential & simultaneous activities into eight steps. The steps 4 & 5 have two simultaneous activities each and activities A4 & A4' as well as the activities A5 and A5' are needed to be performed simultaneously. All these ten activities have been arranged into five categories of phases & these are as follows:-

Activity A1 & A2 fall into Preparation Phase.

Activity A3, A4, & A4' fall into Measurement Phase.

Activity A5 & A5' fall into Action Phase.

Activity A6 falls into Satisfaction Phase.

Activity A7 & A8 fall into Application Phase.

The categorization of the activities in five phases has been done on the basis of the tasks performed by the activities. Another important basis of categorization of activities into five unique phases is that activities are grouped together as per their time of execution. Each phase denotes a unique period of time when the activity /activities within that phase are needed to be performed [42]. Fig.4 shows the interrelationship of the activities, steps and phases of the PREMASA model. In traditional CRM software there is no option to measure the cognitive dissonance and satisfaction levels of the customers and that is why the activities mentioned in the measurement and action phase of the PREMASA model need to be integrated and arrangements must be made so that these activities can be done through a CRM software. Das and Kundu developed the Measurement Module which can perform the activities mentioned in the measurement and action phase of the PREMASA model and they also proposed a methodology which is based on the approach adopted by Bach et al to insert the Measurement Module within the existing CRM software by using dynamic plug-in instrumentation on the binary at runtime which will eliminate the need to modify or recompile the application's source and it will also support the

instrumentation of programs that dynamically generate code([33],[43]). Kundu's research on software visualization shows that it is also possible to change instrumentation at any time during execution by modifying the application's binary image [34]. Since dynamic program analysis is being used to modify an existing software system (CRM package) to make it more powerful and updated, program optimization is the desirable option to achieve faster execution, less memory storage and to draw less power. Dynamic analysis is preferred rather than static analysis for this proposed framework because it has the following advantages [35]:-

- It identifies vulnerabilities in a runtime environment.
- Automated tools provide flexibility on what to scan for.
- It allows for analysis of applications in which the actual code is inaccessible.
- It identifies vulnerabilities that might have been false negatives in the static code analysis.
- It permits the developer to validate static code analysis findings.

It can be conducted in virtually any application.

The dynamic program instrumentation shall insert the measurement module within the legacy CRM software source code without hampering the current execution of the software package. The alteration and updating will be performed on real time basis and during the actual runtime of the CRM software. When finished, the updated package will contain the measurement module and it can perform the predefined task. In this regard Kundu and Das's work is worth mentioning because they attempted to develop a conceptual framework to identify the runtime technical flaws of CRM software and secondly to take measures to solve the technical flaws using software visualization which will ultimately increase the level of program understanding and the secondary purpose of the proposed framework is to use some optimization mechanisms based on death code elimination and code re-ordering approach to reduce the processing time of a CRM software package[44]. Later Kundu, Das and Banerjee designed a plug in instrumentation tool to perform compiler optimization and their approach has been used in this paper to reengineer existing legacy CRM software [45].

Fig 5 focuses on a detailed analysis of different phases of compiler for inserting the said measurement module into the existing legacy CRM software using program instrumentation followed

by a set of optimization methods to reduce the instrumentation overhead.

In Fig. 5 a legacy CRM software source code is fed inside the compiler which has several sequential phases. It is better to take a look at each of the sequential phases of compiler for understanding how and where instrumentation can be applied for inserting the measurement module within the legacy CRM software to make it more acceptable and realistic in dynamic business world. The compiler phases are as follows:-

1. The Lexical Analysis phase_takes the source program as an input and produces a string of tokens.
2. Syntax Analysis phase__imposes a hierarchical structure on the token streams which is called syntax tree.
3. Semantic Routine phase checks the source program for semantic errors and gathers type information for the subsequent code generation phase and also performs the type checking.
4. Intermediate code generation phase works on syntax and semantic analysis to generate an explicit intermediate representation of the source program which is easy to produce and converts it into target program.
5. Code optimization phase__attempts to improve the intermediate code, so that faster run-time machine code can be obtained.
6. The final phase of Compiler is code generator which first generates the target code consisting of machine code /assembly code which can be relocated, then translates each instruction into sequence of machine instructions.

Just after the lexical analysis and the syntax analysis phase we are using an instrumentation procedure to insert the measurement module to the CRM software package and then the combined code is getting passed through semantic routines and other phases before getting converted into an executable CRM software package/target machine code. Now as the insertion of the 'Measurement Module' through program instrumentation in the existing CRM software may have instrumentation overhead, we are also providing a sequences of optimization methods for proper treatment before the last phase of the compiler called code generation phase. This code generation phase will generate the target machine code containing the 'Measurement module' and thus the existing legacy CRM software will become more appropriate and acceptable as per the recent demands of changing business environment.

4.4 The Integration Phase

Researchers have developed methods on how to integrate legacy systems into SOA([46], [48],[49],[50],[51],[52],[53],[55],[56]).Researchers believe that the process of integrating legacy systems into SOA starts with the identification of the main points of integration into the legacy mainframe and there are three main points which are as follows :

Presentation Layer:

This layer provides some level of service to the consumer to improve the user experience and enable access to other applications through the front end.

Business Layer:

In most of the legacy applications, business processes are often not discrete, standalone services but is a procedure call wrapped in a SOA service. It is often difficult to determine where a process starts and where it ends.

Data Layer:

The data layer allows users to use native data calls, and enable quick access to relational or non-relational data stores [46].

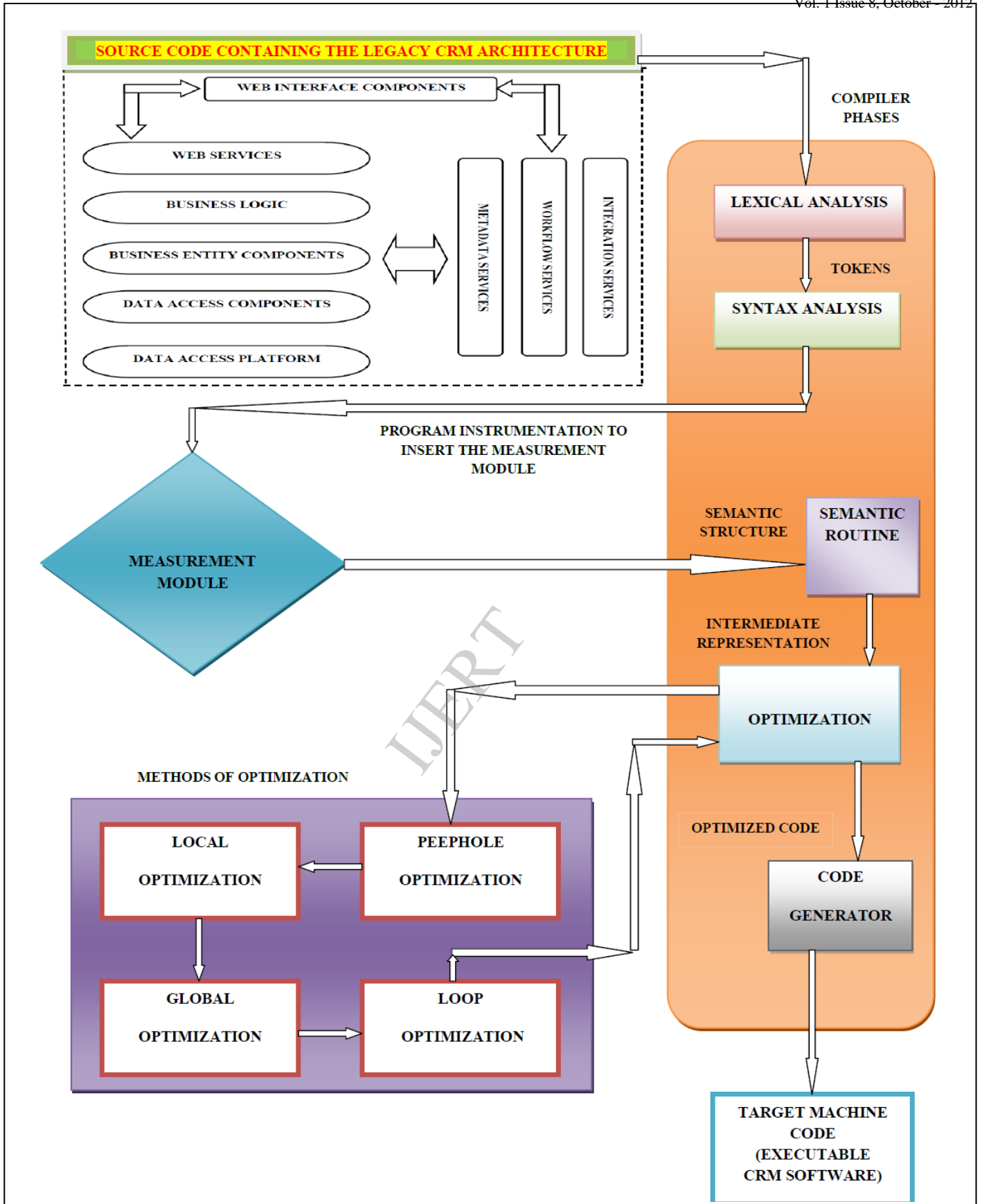


Fig.5. Analysis of different phases of compiler

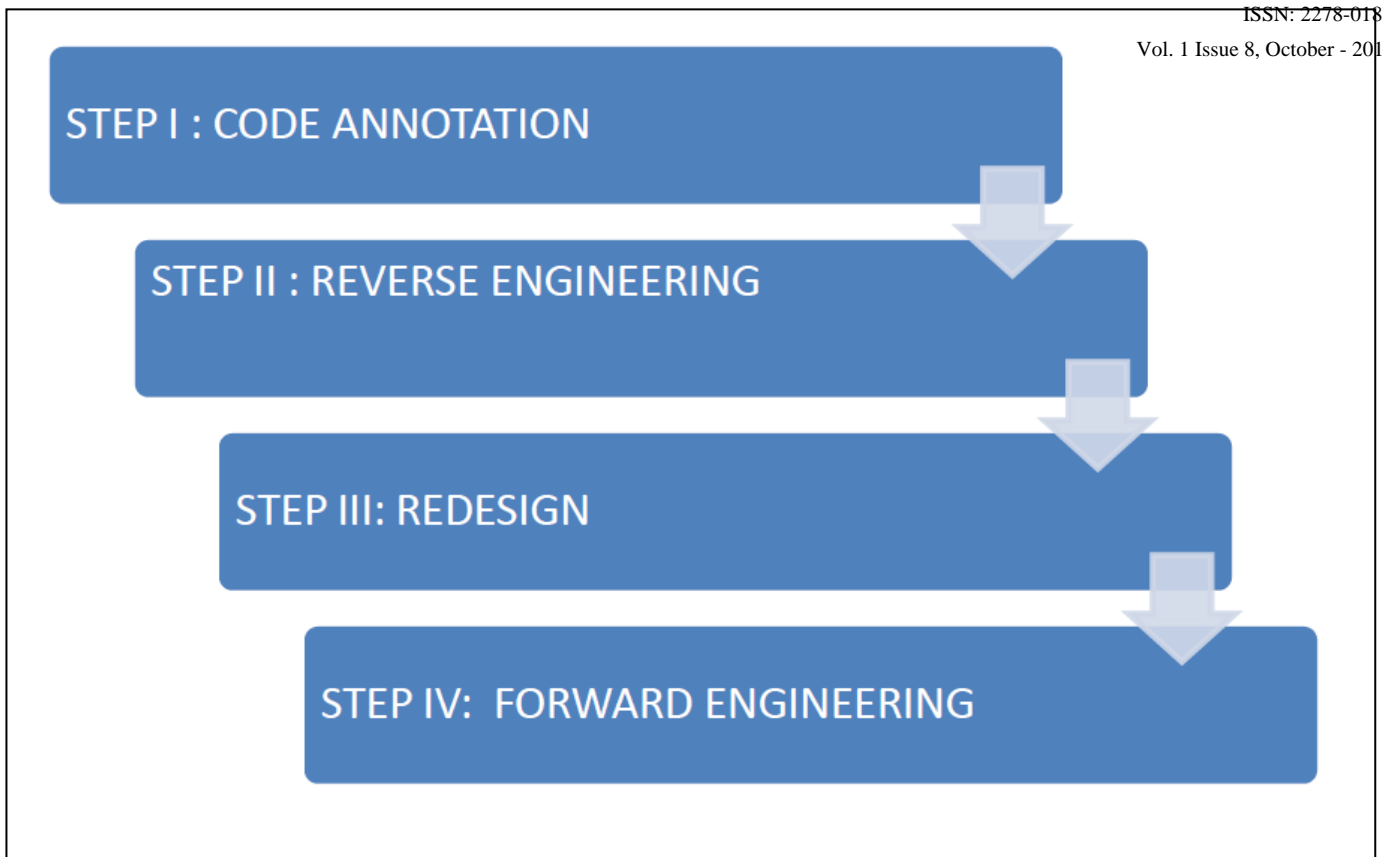


Fig.6. Two dimensional transformations.

According to Matos & Heckel, migration from legacy systems to Service Oriented Architecture (SOA) requires necessary changes along the technological and functional dimensions. This two dimensional change methodology follows technological restructuring in terms of the layering of software systems leading to the development of a 3-tiered architecture. This three tier architecture contains business logic, business data or user data and the user interface (UI). The next level of change must occur along the functional dimensions leading to separation and grouping of components as services. This methodology of two dimensional transformations involves four sequential steps which are as follows (Ref. Fig. 6):

Step I: Code annotation

At this step the source code is first annotated and then blocks of source code are labeled to the various elements of the target architecture where they will be mapped to.

Step II: Reverse engineering

At this step, a graphical representation of the code is being generated on the basis of the gathered information from the annotation process. Though the graph does not have a one to one mapping with the source code, it is based on a metamodel that consists of a type graph containing code structure information, its categorization and its association to architectural elements. The biggest benefit of this graphical representation is that it provides the visualization of the transformation process.

Step III: Redesign

At this step, using the graph transformation rules the target architecture is being produced. In the technological dimension the rules attempt to reorganize the model into a 3-tier architecture to comply with the primary condition of SOA i.e. separability of business logic from presentation logic.

In the functional dimension the rules restructure the model to comply with another two vital condition of SOA i.e. loose coupling of services and the coarse-grained nature of services

Step IV: Forward engineering

At the last step a log of the transformations are kept and applied at model level and later this is used to drive the code level transformations and in this way the target code is obtained [47].

5. Limitations & Future Directions

The proposed OKB Model and the ADRI approach is based on the tools and techniques of SOA and plug in instrumentation and that is why the inherent limitations of SOA and plug in instrumentation techniques will be present in this new model of business process engineering. Though the applications of SOA have a promising future, many integration related issues are needed to be addressed while the transformation of legacy systems to SOA or migration to SOA platform. As legacy systems generally have proprietary data definitions, it creates the semantic discrepancy between legacy systems and other applications. Researchers have developed ways and means to resolve this issue in order to integrate legacy systems within the SOA [49].

This research paper does not highlight anything about the architecture of the target CPU where the optimized CRM software is needed to be executed because compiler optimization also depends on the target CPU architecture that means the no of registers usage, type of instruction set, pipelines and number of functional units present etc. Hopefully future research will try to work on the above limitation. Recent research has also highlighted the importance of Enterprise Service Bus in the transformation of legacy systems to SOA or integration of legacy systems with SOA. According to this new thinking Enterprise Service Bus should work as mediation and virtualization layer helping to separate business and logical view of the process from its technological implementation and reduce dependencies. This new approach first analyses the legacy software in order to identify the business and presentation logic components and next redesign the legacy code by isolating the business logic and performing code stripping. At the end, coarse-grained and loosely coupled SOA services in the medium and upper layer are created to achieve the benefits of SOA architecture [48].

6. Conclusion

This paper revisits the early concepts of business process reengineering and tries to combine two separate models of BPR to conceptualize a new future ready model by integrating the concepts of Service Oriented Architecture (SOA) and plug in instrumentation and thus opens up new avenues of thinking which will lead the researchers to

conceptualize and theorize new models and framework in the area of business process modeling and business process reengineering.

7. References

- 1]. J.H. Manley, "Information process flow analysis (IPFA) for reengineering manufacturing systems", *Computers & Industrial Engineering*, vol. 25, issue 1-4 ,September, 1993, pp. 275-278.
- 2]. K.C. Hoffman, "Management of enterprise-wide systems integration programs", *Journal of Systems Integration*, vol. 3, issue 3-4, September 1993,p p. 201 – 224.
- 3]. J.H. Manley, "Enterprise information system modeling for continuous improvement", *Computers & Industrial Engineering* , vol. 31, issue 1-2, October, 1996,pp. 273-276.
- 4]. H. Rozenfeld, A.F. Rentes, W. Konig, "Workflow Modeling for Advanced Manufacturing Concepts" , *CIRP Annals - Manufacturing Technology*, vol. 43, issue 1, 1994,pp. 385-388.
- 5]. R. P. Anjard, "Re-engineering basics: one of three, new, special tools for management, quality and all other professionals", *Microelectronics Reliability*, vol. 36, issue 2, February, 1996, pp. 213-222.
- 6]. S. Jarzabek, W. L. Tok, "Model-based support for business re-engineering" , *Information and Software Technology* ,vol. 38, issue 5, May, 1996, pp. 355-374.
- 7]. S. Kim, K. Jang, "Designing performance analysis and IDEF0 for enterprise modeling in BPR", *International Journal of Production Economics*, vol. 76, issue 2, March 21, 2002,p p. 121-133.
- 8]. K. Mertins, R. Jochem, "Architectures, methods and tools for enterprise engineering", *International Journal of Production Economics*, vol. 98, issue 2, November 18, 2005,pp. 179-188.
- 9]. Koontz, H., H. Weihrich, *Essentials of Management: An International Perspective*, TATA McGraw – Hill, New Delhi, 2008, pp. 160 – 172.
- 10]. Weihrich, H., M. V. Cannice, *Management*, Tata McGraw-Hill Education, New Delhi, 2010,pp. 190 – 205.
- 11]. Nieuwenhuizen, C., D. Rossouw, *Business Management: A Contemporary Approach*, Juta & Co., Cape Town, South Affrica, 2008,pp. 43 – 45.
- 12]. DuBrin, A., *Essentials of Management*, South Western Cengage Learning, Mason, Ohio, 2008, pp. 2-4.
- 13]. Wasser, A., M. Lincoln and R. Karni, "Accelerated Enterprise Process Modeling through a Formalized Functional Typology", *Business Process Management*, Van der Aalst. W.M.P., B. Benatallah, F. Casati,

F.Curbera (Eds.), Springer-Verlag , Berlin , Heidelberg 2005 , pp. 446 – 451.

14]. Mili, H., A. Leshob, E. Lefebvre, G. Levesque, G. El – Boussaidi, “Towards a Methodology of Representing & Classifying Business Processes” , *E-Technologies: Innovation in an Open World: 4th International Conference, MCETECH 2009, Ottawa, Canada, May 4-6, 2009, Proceedings*, Babin. G., P. Kropf, M. Weiss (Eds.), Springer-Verlag, Berlin, Heidelberg 2009, pp. 196 – 211.

15]. Aitken, C., C. Stephenson and R. Brinkworth, “Process Classification Framework”, *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture*, Vom Brocke. Jan., M. Rosemann (Eds.), Springer-Verlag, Berlin, Heidelberg 2010, pp. 73-92.

16]. Mancioffi. M., O. Danylevych, D. Karastoyanova and F. Leymann, “Towards Classification Criteria for Process Fragmentation Techniques”, *Business Process Management Workshops 2011 Proceedings*, Daniel. F., K. Barkaoui and S.Dustdar (Eds.), Springer-Verlag, Berlin, Heidelberg 2012, pp. 1- 12.

17]. Grossmann, G., M. Schrefl, M. Stumptner, “Classification of Business Process Correspondences and Associated Integration Operators”, *Conceptual Modeling for Advanced Application Domains*, S. Wang et al. (Eds.), Springer-Verlag, Berlin, Heidelberg , 2004, pp. 653- 666.

18]. Sharp, A., P. McDermott, *Workflow Modeling: Tools For Process Improvement & Application Development*, Artech House, Inc, Noorwood, MA, 2009.

19]. Rosa, M.L., F. Gottschalk, M. Dumas, “Linking Domain Models and Process Models for Reference Model Configuration”, *BPM 2007 Workshops, LNCS 4928 Proceedings*, A. ter Hofstede , B. Benatallah , H.Y.Paik (Eds.),Springer-Verlag, Berlin, Heidelberg, 2008, pp. 417 – 430.

20]. Neiger. D., L. Churilov, “Goal Oriented Business Process Modeling with EPCs and Value –Focused Thinking”, *BPM 2004 LNCS 3080 Proceedings*, J.Desel, B. Pernici, M. Weske (Eds.), Springer-Verlag, Berlin, Heidelberg, 2004, pp. 98 – 115.

21]. Kalakot.R., M.Robinson, *E-Business 2.0: Roadmap For Success*, Pearson Education India, 2004.

22]. Back. A., G.V.Krogh, A. Seufert, E. Enkel, *Putting Knowledge Networks into Action: Methodology, Development, Maintenance*, Springer-Verlag, Berlin, Heidelberg 2005.

23]. Peters. J.F., W. Pedrycz, *Software Engineering: An Engineering Approach*, John Wiley and Sons, Inc., New Delhi, 2008.

24]. Manley, J.H, “OTPM and the new manufacturing paradigm”, *Computers & Industrial Engineering*, vol. 23, issue 1-4, November, 1992, pp. 427-430.

[25].Bieberstein.N., S.Bose, M. Fiammante, K.Jones and R.Shah, *Service Oriented Architecture Compass: Business Value, Planning and Enterprise Roadmap*, Pearson Education, Inc, Upper Saddle River, 2006.

[26]. Lawler. J.P., H.Howell- Barber, *Service Oriented Architecture: SOA Strategy, Methodology and Technology*, Auerbach Publications, 2007.

[27].Bieberstein.N., R.G.Laird, K.Jones and T.Mitra, *Executing SOA: A practical Guide For The Service Oriented Architect*, Pearson Education, Boston, 2008.

[28].Josuttis. N.M., *SOA In Practice: The Art of Distributed System Design*, O’Reilly Media Inc, Sebastopol, 2007.

[29]. Krafzig. D., K.Banke, D. Slama, *Enterprise SOA: Service Oriented Architecture: Best Practices*, Pearson Education, Upper Saddle River, 2005.

[30] Branco. R.R., “Dynamic Program Analysis and Software Exploitation: From the crash to the exploit code”, 2011. Retrieved January 15, 2012, from http://www.troopers.de/wp-content/uploads/2011/04/TR11_Branco_Dynamic_program_analysis.pdf

[31] Cesar.E., A. Morajko, T. Margalef, J. Sorribes, A. Espinosa, E. Luque, “Dynamic performance tuning supported by program specification”, *Performance oriented application development for distributed architectures*, M.Gerndt, (Ed.), IOS Press ,Amsterdam:, 2002, pp. 35-44.

[32]Ladner.R.E., R.Fortna, B.H. Nguyen, “A Comparison of Cache Aware and Cache Oblivious Static Search Trees Using Program Instrumentation,” *Experimental algorithmics: from algorithm design to robust and efficient software*, R.Fleischer, B. M. E. Moret, & E.M.Schmidt, (Eds.), Springer –Verlag ,Berlin, 2002, pp.78-92.

[33] Bach.M., M.Charney, R.Cohn, E.Demikhovsky, T.Devor, K.Hazelwood, A.Jaleel, C.K.Luk, G.Lyons, H.Patil, A.Tal, “Analyzing Parallel Programs with Pin,” 2010.Retrieved January 15, 2012, from <http://www.cs.virginia.edu/kim/docs/ieeeComputer10.pdf>

[34] Kundu.P., “Visualization of Program for Extracting Object Oriented Behaviour & Optimization: A Futuristic Approach,” *Research and Higher Education in Computer Science and Information Technology*, S. S. Sau and A. D. Gupta, (Eds.), SM ,Kolkata, 2012, pp.108-112.

- [35] Jackson. W., "Static vs. dynamic code analysis: advantages and disadvantages," 2009. Retrieved January 15, 2012, from <http://gcen.com/articles/2009/02/09/static-vs-dynamic-code-analysis.aspx>
- [36]. Bhattacharya.S., "Integrate legacy systems into your SOA initiative: Convert legacy applications into SOA-based applications", 2007. Retrieved on 2nd April 13, 2012 from <http://www.ibm.com/developerworks/webservices/library/ws-soa-legacyapps/>
- [37] Das.D., P.Kundu, "Application of Service Oriented Architecture (SOA) in Implementing a CRM Process: A Conceptual Approach", *Research and Higher Education in Computer Science and Information Technology*, S. S. Sau and A. D. Gupta,(Eds.), SM,Kolkata,2012, pp.148-152. Retrieved on 10th April 2012 from <http://ssrn.com/abstract=2016277>
- [38] Nghiem.A.D., *IT Web Services: A Roadmap for the Enterprise*, Pearson Education, Inc., Upper Saddle River, 2003.
- [39] Iverson.W., *Real World Web Services*, O'Reilly Media, Inc., Sebastopol, 2004.
- [40]. Kundu.P., D.Das, B.K.Ratha, "WSDL Specification of Services for Service Oriented Architecture (SOA) Based Implementation of A CRM Process" *International Journal of Scientific & Engineering Research*, vol. 3, issue 10, October2012.
- [41] Cerami.E., *Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL*, O'Reilly & Associates Inc., Sebastopol, 2012.
- [42].Das.D., "PREMASA Model: An Integrated Approach To Customer Relationship Management," *IEM International Journal of Management & Technology*, vol.2, no.1, Jan. 2012, pp.111-118.
- [43].Das.D., P.Kundu, "Applications of Dynamic Program Analysis in a CRM Process: A Futuristic Concept," *International Journal of Scientific and Research Publications*, vol.2, issue. 4, Apr.2012, pp. 306-318.
- [44]. Kundu.P., D.Das, "An Attempt to Analyze & Resolve the Pitfalls in CRM Software through Plug-In Instrumentation" *International Journal of Scientific and Research Publications*, vol. 2, Issue 5, May 2012, pp. 313 – 320.
- [45].Kundu.P., D.Das, A.Banerjee, "Plug-In Instrumentation: A futuristic ICT approach for teaching the Runtime Behaviour of Software", *U.G.C Sponsored Seminar on I.C.T in Higher Education: Opportunities & Challenges in the 21st Century Proceedings*, SPS Education India Pvt. Ltd., Kolkata, pp. 24-27.
- [46].Papkov. A., "Develop a migration strategy from a legacy enterprise IT infrastructure to SOA-based enterprise architecture", 2005. Retrieved on 2nd April 13, 2012 from <http://www.ibm.com/developerworks/webservices/library/ws-migrate2soa/>
- [47]. Robinson.R. , "Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture", 2004. Retrieved on 2nd April 13, 2012 from <http://www.ibm.com/developerworks/webservices/library/ws-esbscen/>
- [48]. Channabasavaiah. K., E. Tuggle, Jr., K. Holley "Migrating to a service-oriented architecture", 2003. Retrieved on 2nd April 13, 2012 from <http://www.ibm.com/developerworks/webservices/library/ws-migratesoa/>
- [49]. Caine.J., J. Hardman, "Design strategies for legacy system involvement in SOA solutions: Understand the challenges of business transformation using SOA in legacy environments", 2007. Retrieved on 3rd April 13, 2012 from <http://www.ibm.com/developerworks/webservices/library/ws-soa-legacy/>
- [50]. Lawrence. C., "Adapting legacy systems for SOA: Finding new business value in older applications", 2007. Retrieved on 3rd April 13, 2012 from <http://www.ibm.com/developerworks/webservices/library/ws-soa-adaptleg/>
- [51]. Nott. C., "Patterns: Using Business Service Choreography in Conjunction with an Enterprise Service Bus", 2004. Retrieved on 2nd April 13, 2012 from <http://www.redbooks.ibm.com/redpapers/pdfs/redp3908.pdf>
- [52]. Keen. M. et al., "Patterns: Implementing an SOA Using an Enterprise Service Bus", 2004. Retrieved on 5th April 2012 From <http://www.redbooks.ibm.com/redbooks/pdfs/sg246346.pdf>
- [53]. Laszewski.T., J. Williamson, "SOA Integration in the Legacy Environment", 2008. Retrieved on 2nd April 13, 2012 from <http://www.oracle.com/technetwork/articles/laszewski-williamson-soa-legacy-082027.html>
- [54]. Matos.C., R.Heckel, "Migrating Legacy Systems to Service-Oriented Architectures", Proceedings of the Doctoral Symposium at the International Conference on Graph Transformation (ICGT 2008).
- [55]. Sheikh, M.A.A., H.A. Aboalsamh, A. Albarrak, "Migration of legacy applications and services to Service-Oriented Architecture (SOA)", 2011 *International Conference and Workshop on Current*

Trends in Information Technology (CTIT) Proceedings, 2011, pp. 137 – 142.

[56]. Wang. Xiaofeng., S.X.K.Hu, E. Haq, H. Garton, “Integrating Legacy Systems within The Service-oriented Architecture”, *IEEE Power Engineering Society General Meeting 2007 Proceedings*, 2007, pp.1 – 7.

IJERT