ISSN 2047-3338

Toward Structured Method for Business Process Re-engineering

Salah Mohamed Ali Hassan[1] and Abdelgaffar Hamid Ahmed[2]

[1]College of Computer Science and Information Technology, Sudan University of Science and Technology (SUST), Khartoum, Sudan

[2]College of Computer Science and Information Technology, Department of Computer Science and Information Technology, Sudan University of Science and Technology, Khartoum, Sudan

[1]salahmali@yahoo.com, [2]elgaffari@gmail.com

***Abstract*– Improving business processes in organizations such as purchasing process by adding some quality (i.e., speed) is an ultimate goal of re-engineering methods. The central activity in this process is modelling business processes using formal language such as Design and Engineering Methodology of Organizations (DEMO). DEMO has two qualities: it considers the organizations from ontological level that abstracted from any kind of implementation and follows the white box approach. Because generally this research area is not mature, this paper examines the knowledge gap in DEMO for the purpose of supporting a structured method. This paper adds two values to DEMO, a package concept and business process variability from a well-established domain in software engineering. Both provide a structure approach for studding and detecting the waste in business processes. A real case study is taken to evaluate these concepts. Because Meta-Object Facility (MOF) is a standard and widely recognized language for supporting engineering tools, a DEMO MOF metamodel is developed and extended to add these concepts to contribute to business process re-engineering domain.**

***Index Terms*– Business Process, DEMO, Concepts and Re-engineering Domain**

I. INTRODUCTION

BUSINESS processes have been studied for more than 25 years. Key paper is Hammer's "Reengineering work, Don't Automate, Obliterate" [1]. Hammer advocates radically change business processes to achieve the same result but vastly faster, more accurately, and with lower cost [1]. The concept of result is intended to accomplish something more fundamental in business process. The concept of speech act is a key concept which offers a way to see fundamental things in business process. Robert Colomb and Ahmed defined speech act as "*A speech act is something that is said which changes how the world is*" [13]. Dietz has developed trend in business process modelling is DEMO (Design and Engineering Methodology of Organizations) [4], is based on the theory of speech acts [5]. Dietz states that DEMO is used to re-design and re-engineering business process. Business process re-engineering lacking of tools and structured methodologies is motivation behind this paper. On other hand, Meta Object Facility (MOF) is an Object Management Group (OMG) widely used great product used as standard language [2] for designing and modelling systems. In this paper we provide a DEMO metamodel based on MOF language since there is no standard DEMO metamodel based on MOF is available. Improving DEMO to enable re-engineering is discussed in this paper and accordingly an extension to DEMO metamodel is provided. The equipment of DEMO metamodel with business process package extension and business process variability extension are main concepts that support re-engineering process. The paper organization is as follows. Section I about introduction. In section II DEMO is explained. Section III is about the case study. Section IV DEMO metamodel is demonstrated. Section V is about improving DEMO. In Section VI paper conclusion is drawn.

II. DEMO

A business process has been studied since 1990. One of the common definitions of business process is a chain of organizational or inter-organizational activities that are necessary of accomplishing a product or service [6], think of electronic banking system, another definition is "*A set of activities that, taken together, produce a result of value to a customer*" [7]. DEMO is a methodology for designing and engineering organization which has a theoretical basis. DEMO consider organizations from a new perspective called the Language Action Perspective (LAP) which assumes that a communication is kind of action [5], [8]. DEMO relies on a sound theory which identifies the principals and definitions of the system and entities within that system. This theory defines the world, the existing entities in it, the behaviour of these entities and their interdependencies. DEMO applies a white-box model to understand organizations, applying this technique is a good approach for business process re-engineering. DEMO considers organizations as a category of social system consisting of actors with responsibility and authority. An actor is an individual or collective subject that performs essential actions. An essential action causes changes in the object of business. Fig. 1 is an essential model of DEMO [4].

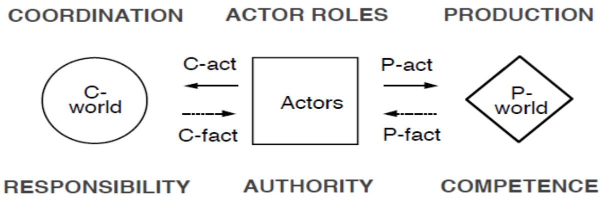

Fig. 1. DEMO essential model [4]

Actors coordinate their activities by seeking commitments. To reach a commitment they have to engage in conversation, which is a commutative action. By a conversation, actors commit themselves to an essential action. The combination of order conversation (O-phase for short), essential action (E-phase for short), and result conversation (R-phase for short) is called a transaction. A transaction is the basic building block of a business process in the organization, Fig. 2 exhibits the notion of business process [4].

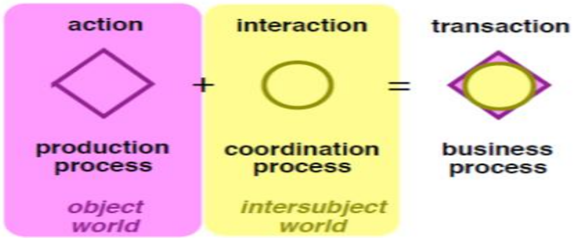

Fig. 2. DEMO Business Process [4]

In the O-phase, the initiator and the executor negotiate for achieving consensus about the production act (P-act for short) that the executor is going to bring about [4]. The main coordination act (C-act for short) in the O-phase is the REQUEST and the PROMISE. The result of successfully performing a C-act is a coordination fact (C-fact for short). In the execution phase, the P-act is brought about by the executor (the result of successfully performing a P-act is a production fact (P-fact for short). In the R-phase, the initiator and the executor negotiate for achieving consensus about the P-fact that is actually produced (which may differ from the requested one). The main C-acts in the R-phase are the STATE and the corresponding ACCEPT. There are also some other C-acts (in case of failure) such as DECLINE, QUIT, REJECT and STOP [4]. Nested transaction notion provide a way to link transactions to each other's [4]. An example of a p-act is a customer buy a book from Amazon. Collections of c-acts like place order by the customer, promise order and request order payment by Amazon sales, promise and payment by a customer, these c-acts are needed to perform the p-act a customer buy a book from Amazon; for Amazon sales to perform the p-act may need to contact a publisher to supply a book and or a distributor to deliver the ordered book to the customer. Supplying a book by a publisher and delivering a book by a distributor ate two inner p-acts needed to perform the main p-act a customer buy a book from Amazon. DEMO offers a number of models for business process's engineer: Construction Model (CM) which is divided into two models Interaction Model (IAM) and Interstriction Model(ISM), the Process Model(PM), Action Model(AM), and State Model(SM) [4]. These models are expressed in diagrams except the action model which descripe the business rules in a semi algorithmic way. Fig. 3 shows the diagrams.

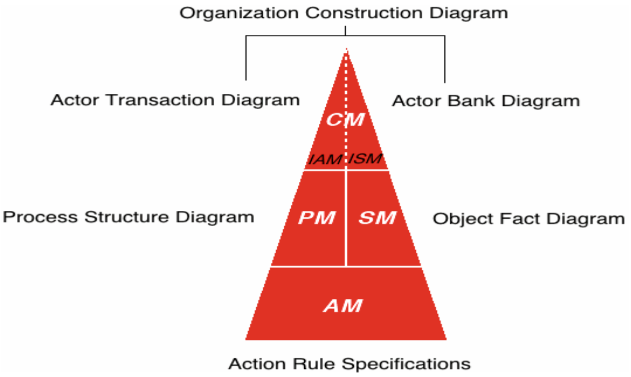
Organization Construction Diagram
Actor Transaction Diagram
Actor Bank Diagram
CM
IAM ISM
Process Structure Diagram
PM
SM
Object Fact Diagram
AM
Action Rule Specifications

Fig. 3. The business rules diagram [4]

Table I shows DEMO models with its diagrams and associated tables.

Table I. DEMO models and diagram

DEMO model		Related diagram	Associated table
CM	IAM	Actor transaction diagram (ATD)	Transaction result table (TRT) (BCT)
	ISM	Actor bank diagram (ABD)	
PM		Process structure diagram (PSD)	Information use table (IUT)
AM		Action rules specification (ARS) not diagram	
SM		Object fact diagram (OFD)	Information use table (IUT), bank content table (BCT), (TRT), object propriety list(OPL)

III. THE CASE STUDY

Customer management system case study: This is a real case produced by direct observation and contact with key users who represent domain experts in Sudanese Electricity Distribution Company (SEDC).

A part of SEDC business process is to manage new customers connection (install a meter to customers), below is the description of new connection business process. SEDC has new connection system that serves the prospective customers to avail electricity supply from the company. The applicant applies for connection service from sales engineer and after paying application fee register his connection requesting to sales engineer in connection sub-division in his regional office. The sales engineer will decide on the connection type (normal or VIP) depend on customer demand. In case of a normal connection type, the staff in charge of connection process at regional office visits applicant house as the first phase of new connection workflow for feasibility, then issue work order after having prepare the technical study. Before client pay for work order he or she may ask for discount, sales engineer secretary applies for discount approval, the SEDC general manager is responsible of discount approval. Sales engineer secretary take care of sending discount application to the general manger and keeps a copy of it in discount application file. In case of discount approved, sales engineer secretary informs the client about it, if the discount rejected client have to pay full invoice (work order) in cash, through bank or by other means. After work order have been paid, field engineer execute the work order; in doing so field engineer prepare material for metering from store after a manager confirmation. A stock control manages and monitors stock periodically and orders material from supplier to keep stock controlled. The metering consists of cabling, meter erection, circuit breaker and user interface unit (UIU). The prepaid client purchase electricity token directly, through ATM, or bank. When a meter has failure, a client phone the call center for maintenance, an agent then passes the request to clerk in regional office to tackle maintenance process. To do so clerk passes again maintenance request to maintenance engineer to fix the problem and after finish he informs clerk about the work has been done, then he informs the agent who is going to report to the CRM system what is done, and the client informed to close the ticket. While if it is VIP connection type, sales engineer passes applicant request to planning engineer in another directorate to study the request, then planning engineer in planning department will tackle the request. The same case like normal connection type client may ask for discount, it is the general manger responsibility to approve or decline the discount, again planning directorate secretary applies to general manager for discount and inform the client about the decision. after the work order having paid for, the planning engineer may request construction engineer in his department to execute the work order, then construction engineer request material from store after manager confirmation, connecting (metering) VIP consists of a transformer erection, cabling, metering, circuit breaker, UIU, cabinet. In some cases according to the output from technical inspection has been done when visiting applicant has external extension needed, sales engineer in regional office and planning engineer in planning directorate are responsible to perform work order. To execute the work order internally, field engineer is responsible to perform the work for normal connection, while construction engineer will do for VIP or outsource the work to some contractor after state body approval. To declare work completion, the client has to sign a document.

A) *The Case Study with DEMO*

First we are going to identify DEMO concepts in ontological (abstract) level, by identifying actors and related transactions; the result is a Transaction Result Table (TRT). The TRT lists the transaction kind with its participants; also we classified the related transactions and associated them to their business processes. Table 2 depicts actors and related transaction and business processes in customer management system case study. Due to the limited space the TRT is fragmented to many parts, each table shows business processes with its related transactions.

Table II. Transaction Result Table and Actors for BP1 to BP4

Business process BP1 application			
Transaction kind	Result kind	initiator	executor
T01 apply	T01 is applied	applicant	Sales engineer
T02 pay application fee	T02 application fee is paid	Sales engineer	applicant
Business process BP2 Normal new connection (N.C)			
T14 N.C C	T14 N.C C is completed	applicant	Sales engineer
T03 N.C C visit	T03 N.C C is visited	Sales engineer	Field engineer
T04 prepare study	T04 N.C C study is prepared	Sales engineer	Planning engineer
T05 discount approval	T05 discount N.C C is decided	Sales secretary	General manager
T06 pay work order	T06 work order N.C C is paid	Sales engineer	Client(customer)
T07 execute work order	T07 work order N.C C is executed	Sales engineer	Field engineer
T08 prepare material	T08 material N.C C is prepared	Field engineer	stock
To9 metering	T09 meter N.C C is erected	Field engineer	Field technicians
T10 cabling	T10 cable N.C C is installed	Field engineer	Field technicians
T11 circuit breaker	T11 breaker N.C C is erected	Field engineer	Field technicians
T12 UIU	T12 UIU N.C C is erected	Field engineer	Field technicians
T13 sign off	T13 N.C C is signed	Sales engineer	Client(customer)
Business process BP3 monitor stock			
T15 monitor stock	T15stock monitored	stock	stock
Business process BP4 order material			
T16 order	T16 order O is	stock	supplier

material	ordered		
T17 receive material	T17 order O is received	supplier	stock
T18 pay order-cheque, TT, L.C	T18 order O is paid	supplier	payable

At this stage of the study the transaction actors table (TAT) with the Process Structure Diagram (PSD) is enough and relevant to our problem. Using PSD enable us to identify causally related transactions and group them into one business process. Using Table II and the case study description the PSDs for business processes BP1 and BP4 is drawn respectively. The same way we can draw PSDs for our all business processes.

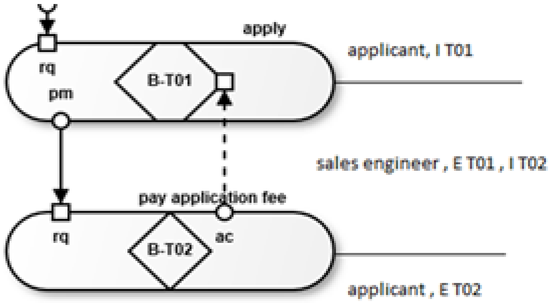

Fig. 4. BP1 PSD

Fig. 4 shows the BP1 with constituent from two related transactions T01 and T02, the notation of B-T01 is used in DEMO3[9] to denote that a transaction is on the business level, all our transactions Shawn in this paper are business level transactions, so we use the traditional notation Txx in all provided tables. From the case study specification, T01 is started externally by applicant actor as initiator. T01 execution by sales engineer as executor will wait until T02 is accepted which means the fee is paid. Fig. 4 is PSD for BP1.

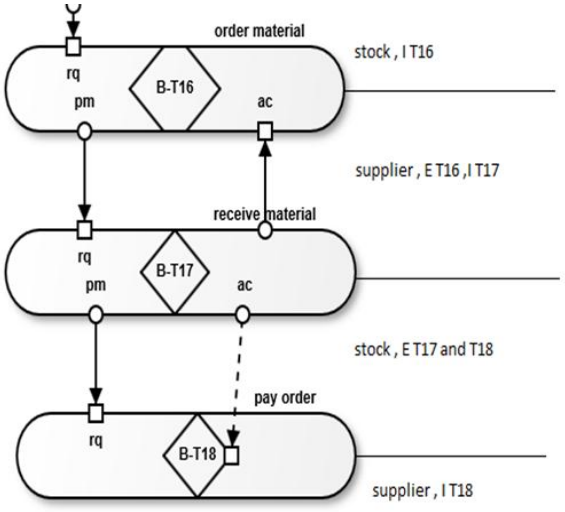

Fig. 5. PSD for BP4

The PSD diagram for BP4 shows three transactions, order material (T16) for order O, receive material (T17) for order O, and pay order O (T18) all of them constitute one business process. This diagram also shows that stock actor is the initiator of T16 and executor of T17 and T18. The supplier actor is the executor of T16 and initiator of T17 and T18. The relationship between these transactions as specified in the case study, that is: T16 (order material) for order O transaction is considered finished after T17 and T18 are accepted. The execution of T18 (pay order) transaction for order O payment is waited until transaction T17 is accepted which means that the company will not pay for the order unless stock receive the goods. It is clear that from this sketching DEMO is rich enough to express business process in more details (classification) and can insure consistency (i.e., such as waiting notation) between transactions not like other BP modeling tools such as BPMN. In the same way we can model the rest of PSDs for other business processes; to mention it here again is kind of repetition.

The second fragment for transactions kind and actors is given in Table III. Table III lists the transaction kind and result kind for business process BP5 and the initiator and executor for each transaction.

Table III. Transaction kind and result with actors for BP5

Business process BP5 VIP connection (VIP.C)			
Transaction kind	Result kind	initiator	executor
T19 VIP connection	T19 VIP.C C is completed	Sales engineer	Planning engineer
T20 VIP visit	T20 VIP.C C is visited	Planning engineer	Study engineer
T21 prepare study	T21 work order VIP.C C is prepared	Study engineer	Planning engineer
T22 discount approval	T22 discount VIP.C Cis decided	Secretary planning	General manager
T23 pay work order w.o	T23 VIP.C C work order is paid	Planning engineer	Client(customer)
T24 execute w.o	T24 VIP.C C w.o is executed	Planning engineer	construction (cons) engineer
T25 prepare material	T25 material for VIP.C C is prepared	Cons engineer	stock
T26 metering	T26 meter for VIP.C C is erected	Cons engineer	Field technician
T27 cable	T27 cable for VIP.C Cis installed	Cons engineer	Field technician
T28 breaker	T28 breaker VIP.C C is erected	Cons engineer	Field technician
T29 cabinet	T29 cabinet VIP.C C is erected	Cons engineer	Field technician
T30 UIU	T30 UIU VIP.C C is erected	Cons engineer	Field technician
T31 transformer (trans)	T30 trans VIP.C C is erected	Cons engineer	Field technician
T32 sign off	T32 VIP.C C is signed off	Cons engineer	Client(customer)

The last fragment table according to our case study specification is for transactions kind and actors for business process BP6 to BP9 as shown in Table IV.

Table IV. Transaction kind and actors for BP6 to BP9

Business process BP6 external connection (EX.C)			
Transaction kind	Result kind	initiator	executor
T33 EX.C normal	T33 EX.C E is executed	Sales engineer	Field engineer
T34 EX.C approval	T34 EX.C E approval is decided	Sales engineer	State body
T35 EX.C normal	T35 EX.C E is executed	Sales engineer	contractor
Business process BP7 external connection (EX.C) for VIP			
T36 EX.C VIP	T36 EX.C E for VIP is executed	Planning engineer	Construction engineer
T37 EX.C VIP approval	T37 EX.C E for VIP is decided	Planning engineer	State body
T38 EX.C VIP	T38 EX.C E for VIP is executed	Planning engineer	Construction engineer
Business process BP8 electricity purchase			
T39 purchase	T39 purchased direct	client	Direct sales
T40 purchase	T40 purchased POS	client	agent
T41 purchase	T41 purchased ATM	client	agent
T42 purchase	T42 purchase bank	client	agent
Business process BP9 maintenance			
T43 start claim	T43 claim cl is started	client	SEDC agent
T44 maintenance	T44 claim cl is maintained	SEDC agent	Maintenance team
T45 end claim	T45 claim cl is ended	SEDC agent	client

IV. DEMO METAMODEL

The meta object facility (MOF) is an object management group [OMG] standard language used to model systems. The MOF provides a language to describe design languages. A metamodel is a conceptual model for system model syntax [2].

There is no standard DEMO metamodel based on MOF language is available. A DEMO metamodel based on MOF language is designed for the purpose to study DEMO business process. Fig. 3 shows DEMO metamodel that capture concepts related to our problem. The core concept in DEMO is a transaction. Each Transaction metaclass in the DEMO metamodel has name and identifier attributes that inherited from Identification superclass. Initiator and Executor metaclasses are kind of Stakeholder that participate in each Transaction; there are only two Stakeholders play a role in the Transaction. The Stakeholder can play more than one role according to the metamodel description, the Stakeholder also has name and identifier attributes inherited from Identification superclass. Each Transaction is composed of three phases according to the DEMO theory: Order, Execution, and Result phase [4]. The relationship among these phases and Transaction is very strong, so we use full aggregation to associate them. Order and Result classes are kind of C-act, each of them has specific C-act determined in two enumeration classes OKind and RKind. Each C-act produce C-fact in contributing to the bringing about result. A collection of C-act makes up a P-act. The main P-act has zero or more Inner P-act that is kind of P-act. Each P-act produces P-fact. A P-act can have more than one Transaction, which has zero or more SubTransaction. The BusinessProcess metaclass has one or more related causally Transactions, each of them has many ProcessStep. The ProcessStep either a combination of C-act with its resulted C-fact or zero or more Execution step. A BusinessProcess may have one or more ProcessStep. The Flow and Condition are kind of Link, which links PorcessStep in specific sequence.

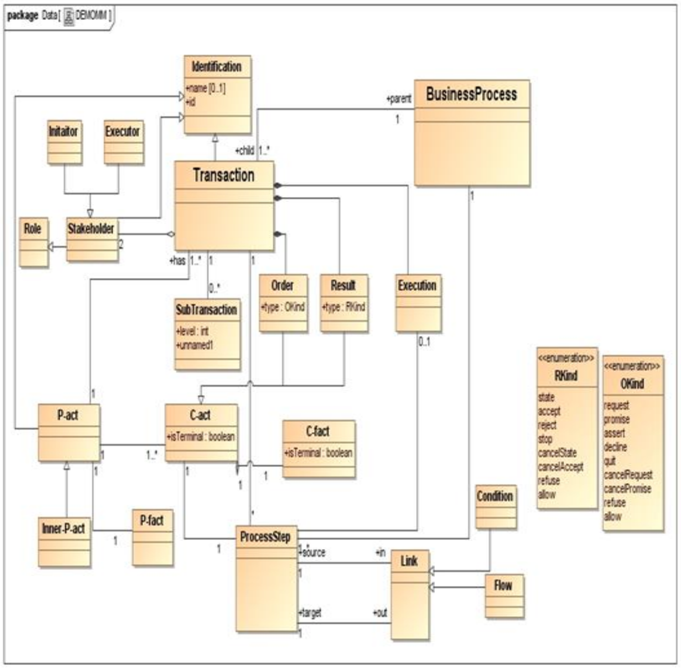

Fig. 6. DEMO MOF-Metamodel

V. IMPROVING DEMO

A) Introduction

Two concepts are introduced: the package concept is utilized from UML and variability from software product line to improve DEMO methodology. The MOF language is used to design a metamodel for business process package and variability. The MOF provides two ways to design a metamodel: using MOF itself or using profile mechanism offered by UML [2]. The choice of one alternative depends on the domain we want to construct a metamodel for it. Using MOF to design a metamodel provide flexibility but it is costed, need to develop tools (code) for full support. Using a profile provides the possibility to reuse the repository of tools supporting UML, but in profile API to connect to other application is not an easy task. We are going to use a metamodel since there is no standard DEMO metamodel based on MOF is available; nevertheless MOF based DEMO metamodel is enabling to free introduce these DEMO concepts.

B) Package

Package business process concept is a mechanism to organize and manage complexity. The value of package is that it provides understanding for human beings by allowing to present large models in small enough ones.OMG uses this to handle the complexity of 800 pages of standard specification for UML. From DEMO and in our case study for *new connection system* this will add value if we group BP1, BP2, BP4 and BP6 business processes to make up *normal connection* (N.C) package; also by grouping BP1, BP4, BP5 and BP7 in VIP connection package.

A package is a kind of name space, such that names of classes, associations, properties etc. are unique within the packages that own them. To do that we use package name as a prefix for names of classes, associations and properties [2], for example VIP and N.C are package names, we find planning engineer (P.E) is a class name in both packages so the name space will be VIP::P.E and N.C::P.E which differentiate between P.E classes that belong to different packages. This problem was not considered in DEMO so far. A package can have subpackage (nestingPackage, nestedPackage) [2]; VIP and N.C packages can be grouped together in one package new connection package (N.CN) VIP and N.C are subpackages from new connection (N.CN) package. Abstract classes serve to factor out attributes and associations that are common to a number of business processes. Attributes and associations reflect resources consumption that are shared between business processes. One of the benefits of package is that it helps us recognize highly consuming resources that show us the commonality between business processes.

Then if we have enough knowledge about this commonality we will double the benefits of reducing it, the result has great effect in business process re-engineering. As examples of consuming resources are actors who are domain experts in Sudanese electricity distribution company maintenance business process, like (managers, engineers, technician etc). We use the concept of package in business process to allow us to study specific domain, in some cases we may have more than one business process and we want to study them for optimizing purpose. Package business process help us to scope our study of them.

This the reason behind borrowed UML package concept to extend our DEMO metamodel which makes a business process package explicit in DEMO models. First we provide a UML package model that is related to our study. Figure 6 shows the package concept from UML.

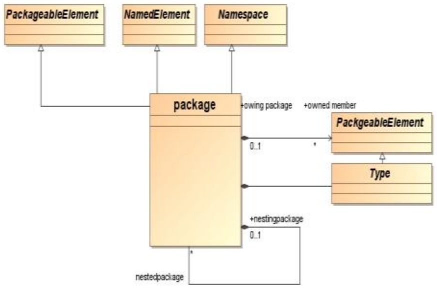

Fig. 7. Package model [2]

The package is a kind of abstract metaclass PackageableElement (abstract class has no direct instances, but through there concrete subclasses). A concrete class is a class with instances [2]. A package also is a name space, which means the names of classes, properties and associations are unique within the package that owns them, its about using package name as a prefix for names of classes, associations and properties [2], for example VIP and N.C are package names, planning engineer (P.E) is a class name in both packages so the name space is VIP::P.E and N.C::P.E is to differentiate between P.E classes that belong to different packages. A package can own other subpackages, the association NestedPackage and NestingPackage shows the relationship between packages [2].

The business process re-engineering lack of structured methodology that can be used; the ultimate goal of our study is to provide tool based on systematic way to support business process re-engineering. The MOF metmodel is a flexible framework for tackling this sort of problem. MOF will help introduce the Package concept to DEMO. The DEMO metamodel and it's business process package and business process variability extension metamodels are considered starting point to develop tools that support business process re-engineering; this goal drive the researcher to use a metamodel mechanism to extend DEMO instead of using profile mechanism. Fig. 8 shows the business process package metamodel. The business process is a kind of package.

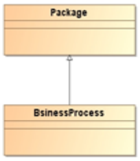

Fig. 8. Business process package extension

The shape below is the proposed business process package notation.

The VIP and N.C business process packages and their owned N.CN package model is shown in Fig. 9. It is clear from the case study that BP1, BP2, BP6 can be packaged together in N.C package; also BP1, BP5, BP7 can be packaged together in VIP package.

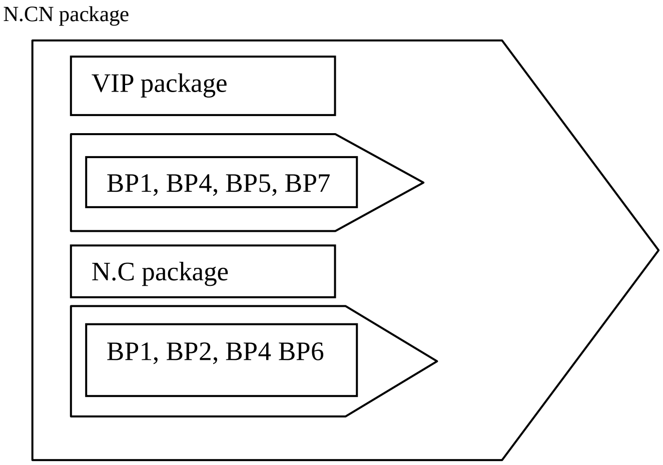

Fig. 9. Business process package notation

C) Variability

Since the general assumption of the re-engineering approach is that there is one business process has more quality (i.e., speed, cost, etc.) over the working process, this suggests that there are at least two alternative designs for this business process in this situation. This observation is an idea behind calling a variability concept from well-established field in software engineering. The concept of variability is borrowed from software product line (SPL) and used to model design alternatives; so it will help us to study business process variants. This is consistent with fact that usually engineers look at alternatives when they think about re-engineering problem. So variability and variants are formalizing the problem to be as a selection mechanism where variants could be evaluated if model is became informative enough.

One of the variability definitions that is suitable for our problem is that *"variability is a variable item of the real world or a variable property of such an item."*[10]. In customer management systems in our case study the VIP connection and Normal connection share many things and differ in the way of billing. Also from the variability knowledge, each variability subject has instance which is variability object, for example, in electricity *purchase methods*, electricity purchase method is a variability subject with direct purchase, Point Of Sale (POS) and Automatic Teller Machine (ATM) are examples of variability objects. Two terms are used to model variability in business process domain: variability point which is a representation of a variability subject within domain artefacts, and variant term is a representation of a variability object within domain artefacts. There are kinds of variability identified by [11]: variability in time and variability in space which is relevant to our problem. Variability in space refers to an artefact in different shapes at same time [12]. These two concepts provide us with some causes of variability in origin.

Variability is classified also as external variability [12], the one that is visible to customer so he has ability to select variants, and internal variability [12] is one that is hidden from customer. Both are of our interest because the approach taken by DEMO is white-box regarding the modeling of business process. This means an external variability requires internal variability. In electricity purchase system customer can choose direct purchase, POS purchase or ATM purchase; example of internal variability is new connection system has VIP and Normal connection.

Variability Modeling

DEMO models can be seen as domain requirement artefacts. To model DEMO business process variability a metamodel is provided with variability notations. Figure 10 is variability metamodel. We use MOF language to extend our DEMO metamodel to model Transactions. A transaction in DEMO is apparent element that can show variants. These variants are intrinsic or demonstrate one sort of variability. In order to make the difference between these two situations transaction class is extended.

A Transaction is a kind of variant metaclass that associated with variant point abstract metaclass; the association constraints are: each variant point is associated with zero or more variant, and each variant has to be associated with one variant point. Note that in this case one flow can take the role of variation point which is main one and another might take the related variation point. In this case the identity will differentiate between them. Both are supposed to be supplied by business processes engineer. Variability in this way provides a systematic way to represent business process's alternatives and/or optional. This facility or feature adds value to DEMO where one can study waste such as by adding the cost or ROI for each variant.

The study of waste is essential to business process re-engineering definitions phase. This also makes a study of waste explicit and systematic where variability in space is shown using DEMO. Fig. 9 is the extension of DEMO metamodel. The transaction is a kind of variant metaclass associated with variant point abstract metaclass. The association has attribute variability dependency; this association needs to be formalized and modeled. From the case study regarding electricity purchase it is optional to choose ATM, POS etc., while in new connection business process in case of external connection is needed, state body approval is mandatory.

The Table V depicts business process variability notation.

Table V. The variability description

Variability string	description	Graphical notation
V11	Mandatory(man)	–
V12	Alternative(alt)	↔
V13	Optional(opt)	◊---

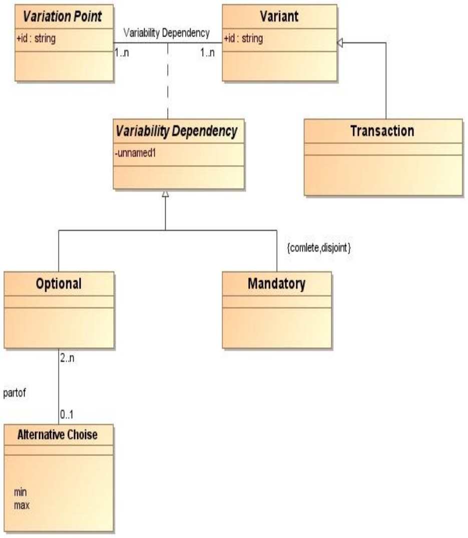

Fig. 10. Business process variability

VI. CONCOLUSION

The context of this paper is re-engineering business processes. The lack of structured methods for re-engineering is the main motivation behind this work. Because formalizing business process using languages such as DEMO has an advantage of more elaborated models in addition of taking the white-box approach to this problem, it has been the focus of this paper.

Since the general assumption of the re-engineering approach is that there is one business process will improve the work process and so it adds quality (i.e., speed, cost, etc.), this suggests that there are at least two alternative designs for this business process in this situation. This observation is an idea behind calling a variability concept from the well-established field in software engineering: software product line. Also DEMO lacks the concept of package .A package concept is introduced to add two values to DEMO, ambiguity which is realized through Package and disambiguating concepts, and expressing commonality and focus on a domain.

Both features are contributing to re-engineering process. In order to enable these two concepts: variability and package, a metamodel approach is taken for extension. MOF metamodel is developed for DEMO and the two concepts are introduced.

REFERENCES

[1]. M, Hammer, Reengineering Work: Don’t Automate, Oblitrate, Harvard business review, 1990.

[2]. OMG Unified Modeling Language (OMG UML) Version 2.5 www.omg.org 2015.

[3]. Booch, G, Rambaugh, J, Jacobson, I., The Unified Modeling Language Reference Manual, Addison-Wesley, Reading, 1999

[4]. Jan L. G. Dietz. Enterprise Ontology: Theory and Methodology. Springer, May 2006. ISBN 3540291695.

[5]. Van Reijswoud, V.E, J.B.F. Mulder, J.L.G Dietz, Communicative action-based business process and information systems modelling with DEMO, Information Systems Journal, 1999.

[6]. Davenport, T.H, Process Innovation: Reengineering Work Through Information Technology, Harvard Business School Press, Boston, 1993.

[7]. Hammer M., Champy, C., Re-engineering the Corporation: A Manifesto for Business Revolution, Brealey, London, 1993

[8]. Dietz, J.L.G., J.B.F. Mulder, Realizing Strategic Reengineering Objectives with DEMO in Proc. International Symposium on Business Process Modeling, Spriger-Verlag 1996.

[9]. Jan L.G Dietz. Demo-3 Models and Representations. Available at www.DEMO.nl, September 2009.

[10]. Svahnberg, M., Gurp, J. Van, Bosch, J. A Taxonomy of Variability Realization Techniques. Proc.ACM:Software—Practice &Experience. July 2005. 35(8), pp. 705–754.

[11]. Bosch, J., Florijn, G., Greefhorst, D., Kuusela, J. Obbink, H. Pohl, K. Variability Issues in Software Product Lines. In: Proceedings of the 4th International Workshop on Product Family Engineering (PFE-4). Software Product-Family Engineering .Bilbao, Spain: Springer, Berlin Heidelberg New York, pp. 13–21; 2002.

[12]. Pohl, K. and Böckle, G. and J. van derLinden, F. Software Product Line Engineering: foundations, principles and techniques. Springer, Berlin 2005.

[13]. M Ahmad, C Robert, S Shazia - PACIS 2008 Proceedings, 2008 - aisel.aisnet.org.