A Framework for Business Process Redesign Using Unified Process Concepts

Mohammed Alsaigh
King Abdulaziz University
Jeddah, Saudi Arabia.

Abstract

People involved in business process redesign include both managers who own the processes and technicians who are responsible for implementing the redesigned processes. Information systems play an important role in linking these two parties together. The literature shows that many problems in business process redesign are related to using a traditional, linear life cycle approach and to communication gaps between stakeholders. To address these two issues, this paper proposes a framework inherited from a systems development framework which is based on unified process concepts. The purpose of this framework is to create a common language to be used by all parties in business process redesign projects in all phases in order to save time and effort and increase project efficiency.

1. Introduction

Business process redesign (BPR) can be defined as a managerial approach to improving efficiency and effectiveness of business processes that exist within and across organizations (Boar 1993; Hammer 1990; Davenport 1993). In fact, information systems (IS) have historically played an important role in BPR, and are considered by some as major enablers for new forms of working and collaborating within an organization and across organizational borders. This role can be represented as the link between technology and business (Figure 1). Attaran (2004) has classified the role of IS in BPR projects into three stages: *before design* process, *during design* process, and *during implementation.* Table 1 shows some of the activities that take place in each stage.

On the other hand, the success rate of BPR projects is not satisfactory. Research show that approximately 70% of BPR projects fail and some believe that figure may be even higher (Grant 2002; Bashein et al. 1994; Davenport and Beers 1995). A review of the literature shows that there are several factors that are commonly linked to failure in BPR projects. These can be summarized as follows:

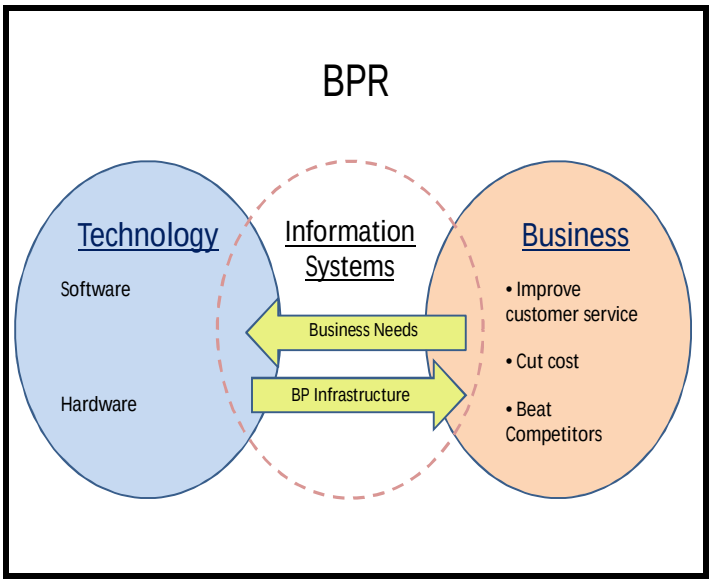

Figure 1: The relation between BPR and IS

- BPR projects commonly follow the traditional waterfall life cycle structure. The problem is that this approach is linear while business processes in reality are non-linear (Dewalt 1999).
- Typical BPR projects involve a combination of managerial staff, who are the owners of the processes, and technical staff, who are the implementers of the project. However, often there is a communication gap between these two groups. There is a need for a common language, which allows the two groups to communicate and exchange information with each other (Alan et al. 2008). This common language should be able to represent business processes understandably for both IS and managerial staff, and should be flexible in being applicable through different stages in BPR projects (Frazaneh 2003).
- There exist transition problems between the various stages of the BPR project life cycle: identification of current processes, analysis, design, and implementation. Thus there is a need for bridging these communication gaps (Ganesh 2000).
- Other obstacles to BPR have been documented, such as the lack of sustained management commitment and leadership, unclear definition of BPR projects, unrealistic scope and expectations, resistance to change, and inadequate resources (Bashein et al. 1994; Davenport and Beers 1995; Kim 1998; Klein 1994).

The goal of this paper is to improve the BPR environment by proposing a framework for the BPR life cycle using the unified process (UP) as a model. The proposed framework will address the first four issues listed above, while the last issue will be out of the scope of this paper, as it is related to external factors.

Before Design	During Design	During Implementation
Create infrastructures and manage information that support evolving organization	Bring vast amounts of information into the process	Create a digital feedback loop
Foster process thinking in organizations	t Bring complex analytical methods to bear on the process	Establish resources for critical evaluation of the reengineered process
Identify and select process for redesign	Identify enablers for process design	Institute a program of ‘‘cleanup’’ and damage control in case of failure
Participate in predicting the nature of change and anticipate the information needs to support that change	Enhance employees’ ability to make more informed decisions with less reliance on formal vertical information flows	Improve IT processes to meet increasing needs of those divisions that have gone under reengineering processes
Participate in designing measures of success/failures of reengineering	Capture the nature of proposed change and match IT strategy to that change	Communicate ongoing results of the BPR effort

Table 1: IS role in BPR

2. Systems development environment

There is an obvious similarity between the traditional systems development (SD) life cycle and BPR life cycle. Both follow similar stages, and their structure is almost identical. Figure 2 shows that both, BPR and SD have analysis, design, implementation, and testing phases; the main difference is that SD starts with requirements gathering, while BPR starts with identifying the current processes. Moreover, SD is faced with some similar problems to BPR; these issues can be summarized as follows:

- Traditional methods of SD are linear, while the process of development is non-linear in its nature (Satzinger 2003).
- There exist transition gaps between analysis and design stages (Satzinger 2003).

These similarities in environments and problem issues are the motivation for investigating how these issues have been addressed in the SD environment. Satzinger (2003) shows how to use the unified process (UP) to deal with the problem issues of the waterfall approach. Using the UP as a framework for the whole SD process solves bridging the gap between the analysis stage and the design stage in the project life cycle. Also, he shows that the unified modeling language (UML) diagrams, such as use case diagrams, activity diagrams, and system sequence diagrams, are useful for all stages, and have the capability to serve as a container for all development requirements and for modifications through all stages.

3. The Unified Process and the Unified Modeling Language

The unified process is a comprehensive, object oriented (OO) systems development methodology originally developed by Jacobson, Booch, and Rumbaugh in 1999. The UP draws on accepted best practices such as risk mitigation, iteration, and model-driven development and is now widely accepted as a leading (if not standard) OO development methodology. The focus on risk and iteration is grounded in the spiral model developed by Barry Boehm in 1988. The spiral model changes the emphasis on the development project from a linear, waterfall process to a non-linear spiral process. Project requirements posing the greatest risk are addressed in the first iterations (Satzinger 2003).

The unified modeling language (UML) is a standard modeling notation that premiered in 1997. Three prominent object-oriented programming professionals, Gray Booch, Ivar Jacobsen, and James Rumbaugh are the principle authors of UML. UML establishes a collection of graphical symbols as well as semantics to support and define these symbols. This collection can be broken down into three kinds of building blocks: objects, relationships, and diagrams. Objects are the abstractions that are first-class citizens in a model; relationships tie these objects together; diagrams group interesting collections of objects. There are nine different kinds of diagrams in UML: class, object, use case, sequence, collaboration, state-chart, activity, component, and deployment (Rumbaugh et al. 1999; Dewalt 1999).

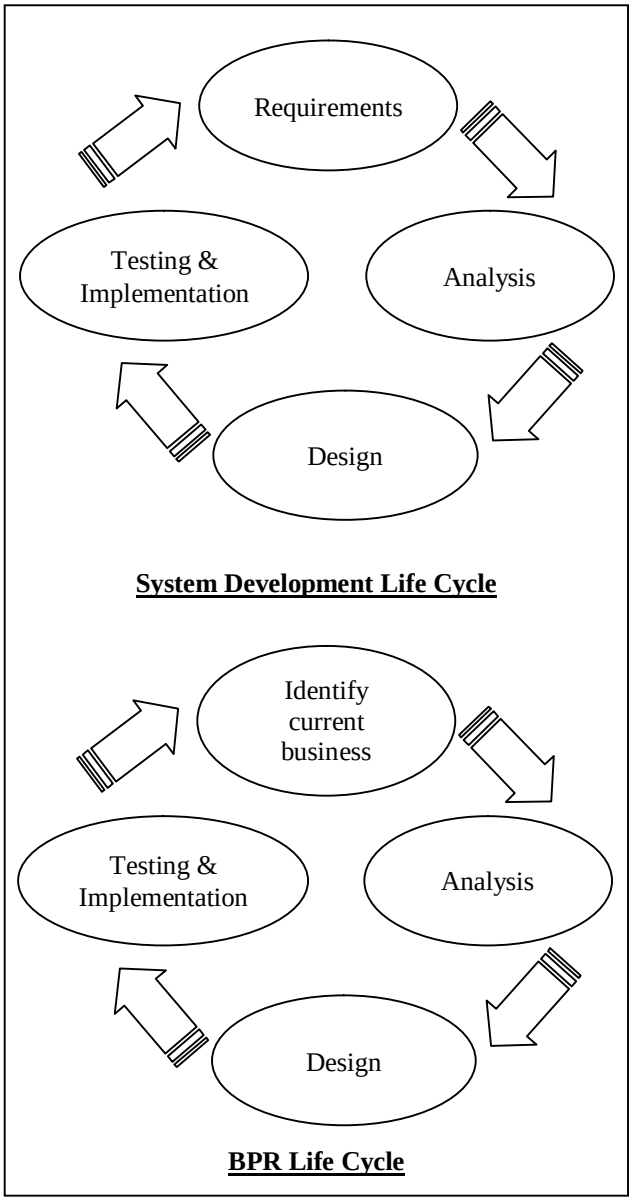

Figure 2: SDM and BPR Life Cycles

Dewalt (1999) stated that although UML is built upon many of the principles of object-oriented programming and object-oriented analysis, it is likely that UML is also effective at representing real-world entities. One limitation to building UML diagrams that are not going to be directly used to build an information system is that many definitions in UML include references to information systems. Some degree of interpretation is therefore necessary in order to use UML outside its intended purpose. Dewalt also stated that a business process is a set of activities that takes an object as an input and adds value to it in order to meet specified requirements. A business process model is therefore a set of components that shows a set of activities.

Usage of UML in BPR and suggestions from Dewalt (1999) can be summarized as follows:

- Use case diagrams are an excellent way to display business processes in a visual model because the powerful abstraction allowed by the simple notation makes them both, easy to build, and easy to understand.
- Placing several use cases in a use case diagram is a good strategy in order to demonstrate the logical connection between the use cases.
- There are two ways to capture detailed business process information that provide more information in order to supplement use cases: sequence diagrams and activity diagrams.
- In UML there is no built-in support for critical performance measures in business process redesign. However, UML provides extensibility mechanisms that can be used to capture information such as performance measures.
- Appropriate performance measures are cost, quality, service, and speed.

4. BPR Framework

The proposed framework, which will address the mentioned issues in the BPR environment, is inherited from the UP framework, which is used by Satzinger (2003) to address similar issues in the software development environment. In fact, the following points are the motivating factors in choosing this direction:

1. BPR and systems development life cycles appear to be very similar.
2. There are common problems between BPR and systems development in their traditional approaches.
3. The UP and UML are used to structure and model reality, and business processes organize and describe reality.
4. UML has the capabilities to represent business processes effectively.
5. BPR projects involve a combination of technical and management people. Having a common language for all parties increases the efficiency of the model.

Therefore, the first step in creating the framework is to use the same model which is used by Satzinger (2003) in order to create an iterative life-cycle, and the second step is to use some UML diagrams with minor changes to be able to transfer BPR data between project phases in order to fill the representation and communication gaps which may emerge between different parties.

4.1 Step I

One of the innovations of the UP is its approach to an iterative lifecycle model. Like the traditional lifecycle, the UP has sequential phases, but with different names and implications. Each phase includes one or more iterations (Figure 3), and each iteration includes activities from multiple of the traditional phases of the traditional life cycle. To adaptthis model tothe BPR environment, the objectives of the project team in each of the four phases are described briefly as follows:

Phase1) **Inception**

Actions:

1. Define the scope of the project.
2. Develop an approximate vision of the project.
3. Identify the current business processes.
4. Nominate team's members.

Phase2) **Elaboration**

Actions:

1. Refine the vision.
2. Identify and describe all processes including main processes.

3. Finalize the scope.
4. Implement the core architecture.
5. Resolve high risks.
6. Produce realistic estimates for cost and schedule.

Phase3)**Construction**
Actions:

1. Iteratively implement the remaining lower risk, predictable, and easier elements for each part separately.
2. Prepare for implementation.

Phase 4)**Transition**
Actions:

1. Complete the new process.
2. Create testing procedures.
3. Create expectations and requirements for the new changes, and implementation.

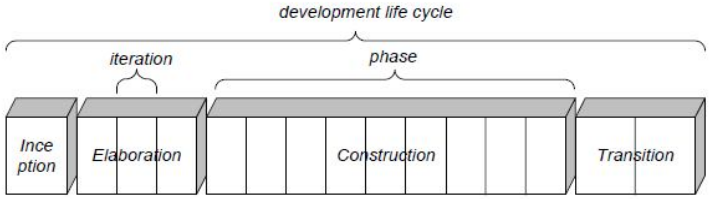

Figure 3: The Unified Process Framework

All iterations in all phases involve some mix of activities, which are called disciplines or workflows. Disciplines include business modeling, identifying current business processes, identifying the goals, design, implementation, testing, deployment, configuration and change management, project management, and managing the development environment. Note that all these disciplines are included to varying degrees in all iterations and in all of the phases, i.e., there may be more emphasis on some disciplines over others in the various iterations.

4.2 Step II

This step shows how to use UML diagrams in the phases defined in the first step. The main contribution of this step is the capability of transferring the required information through project steps, and the understanding and communication of the information by or to all parties. Four kinds of UML diagrams will be used in this framework, as well as an additional component, which can be used as a standard extension to the original UML. The first one is the use-case diagram, which will be used as a description of a set of sequences of actions, including variants, that a system performs that yields an observable result of value to an actor. Figure 4 shows an example for four use cases in a hiring process, which will be explained in details in section 5. The second kind is the package diagram, which is used to separate a diagram into multiple smaller diagrams (Figure 5), if there too many use cases on the same diagram. In fact, UML provides this type of diagram to reduce confusion or complexity through organizing various UML elements, including use cases, into smaller groups. The third and fourth kind of diagrams are sequence diagrams and activity diagrams, which will be used to capture detailed business process information that supplement use cases. The difference between these two types of diagrams is that sequence diagrams are a kind of an interaction diagrams that emphasize the time ordering of messages between objects (see figure 6); while activity diagrams show the flow from activity to activity (see figure 7). Finally, the four performance measures, cost, quality, service, and speed, in addition to any information that does not seem to belong anywhere in particular, will be presented using an additional component called “tagged values”. “Tagged values” is a mechanism that permits extending the UML in controlled ways (see figure 8).

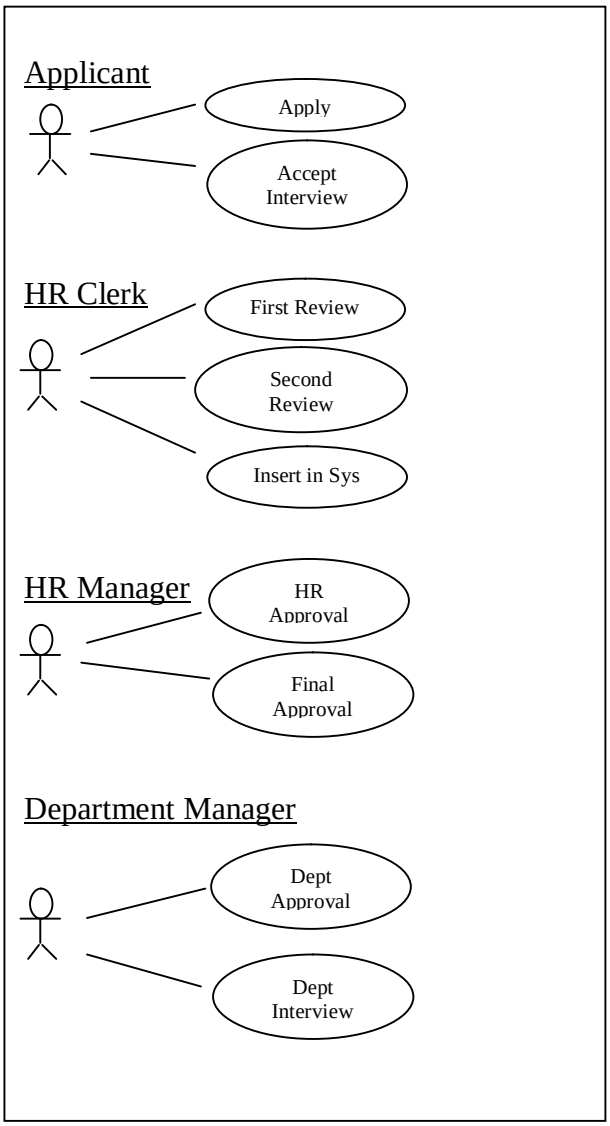

Figure 4: Current Use-case Diagram

In the inception stage, the four diagrams will be used to represent current business processes of the organization, and they will be called inception-use-cases, inception-packages, inception-sequence diagrams, and inception-activity diagrams. In each iteration of analysis, analysts will use the inception diagrams and the previous analysis diagrams to create new analysis diagrams, which have the same structure asthe inception diagram. Then, designers will use analysis diagrams to create design diagrams, and thereby make it easy for the technical people to implements these changes because they all use the same kind of diagrams in all stages, with no loss of valuable information during transition from team to team.

5. Illustrative Example

The goal of this example is to show the mechanism of the proposed framework, and to demonstrate its usefulness. The example is about improving the processes in a Human Resources (HR) Department in an organization. To simplify this example, we will focus only on the “Hiring Process”, and we will go through the four phases of the framework as follows:

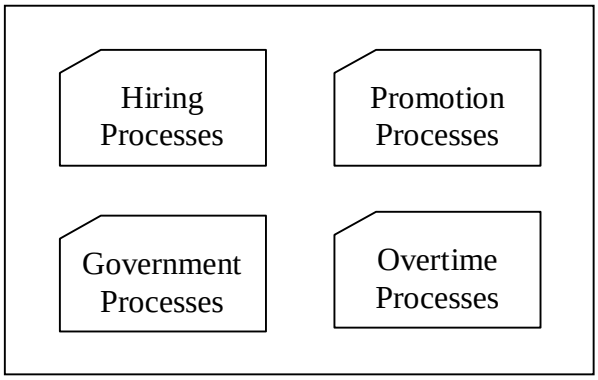

Figure 5: Package Diagram

5.1 Inception Phase

In this phase, project owners should specify the goal and scope of the project. For the hiring process in this example, the goal may be improving the process by decreasing process time from three days to one day. Also, team leaders need to be selected by the project owners, and teams assembled, to work on the project, following the organization's rules. Finally, the technical department will be assigned to extract the selected processes and specify them in UML format.

5.2 Elaboration Stage

The processes need to be grouped into several packages according to their functions, such as hiring and promotion. Then, the hiring package is divided into three sub packages: hiring, firing, and rehiring (Figure 9). Finally, each sub package will be scheduled and assigned to the teams to do the analysis, design, and implementation for each package.

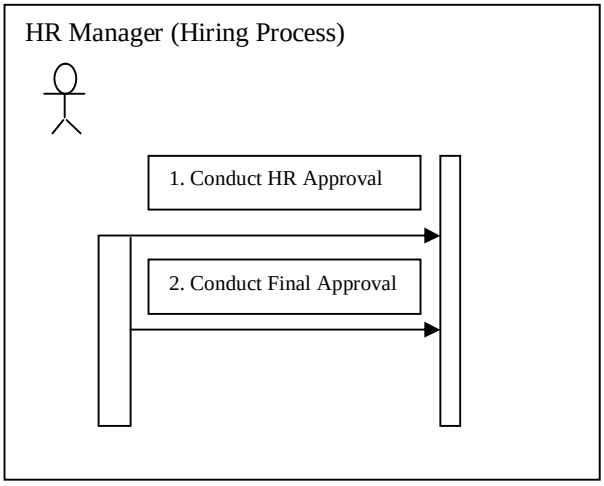

Figure 6: Current Sequence Diagram

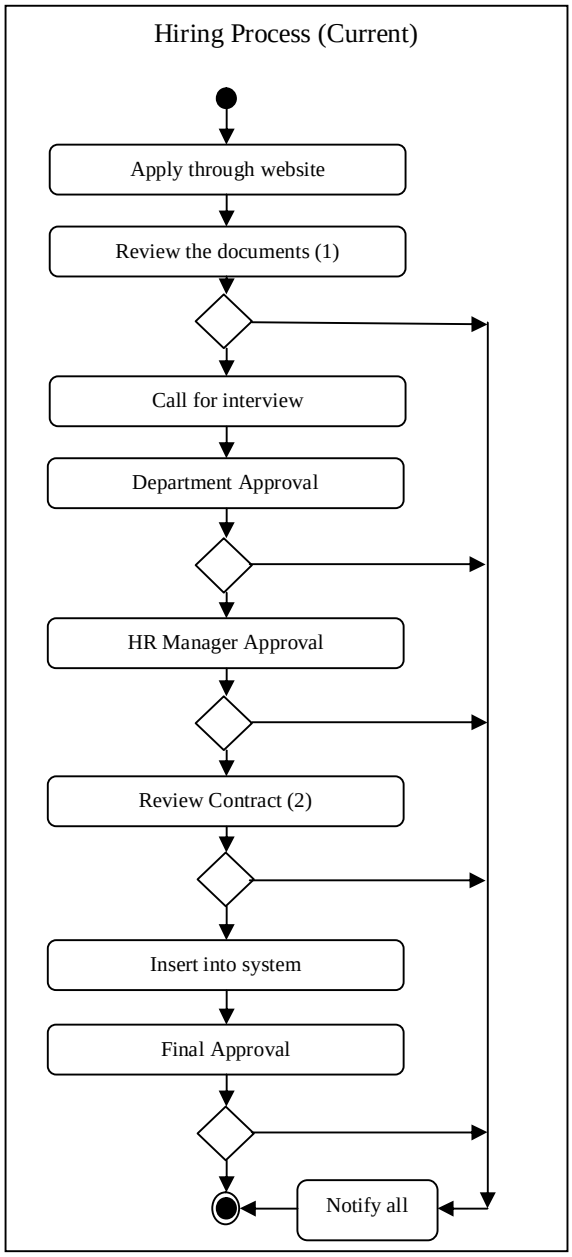

Figure 7: Current Activity Diagram

5.3 Construction Stage

In this stage, the assigned analysis team will work on the four diagrams for the hiring process to determine what makes the process takes so much time. Hypothetically, let's assume that currently the HR manager must approve the process twice, and the suggestion for improving the process is to reduce that to only once. Therefore, they will add their analysis as a note in the same diagrams (Figure 8). Then, the designing team will work on the given notes to generate the new process, and their output will be in the same three diagrams format. Finally, the implementation team will work on the new diagrams to modify the existing application.

5.4Transition Phases

By looking at the new and current diagrams, it will be easy to figure out what kind of training will be required for the employees to start working on the new processes. Also, it will be easy to identify which part of the application should be tested.

From this example, we can see that the same diagrams have been used throughout all phases and used by all parties. This is expected to save time and effort in BPR projects.

6. Contribution

The contribution of the proposed framework can be summarized as follow:

- As we have seen, both managerial and technical employees are involved in BPR projects, and the proposed framework offers a common language, which can be used by all stakeholders in the project and through all stages.
- Although UML is a pure technical tool, its simplicity and easiness allows nontechnical people to understand it and get familiar with it in a very short time. This makes the proposed framework easy to apply.
- The first step in BPR projects is extracting the current processes, which will be redesigned. Using the proposed framework by organizations that use UML in their technical documentation will save the time and effort required for extracting the current processes.
- Similar to the previous point, using the framework will shorten the implementation time, especially for organizations that use applications based on object oriented languages, because the output diagrams of the framework will be used directly in the implementation.

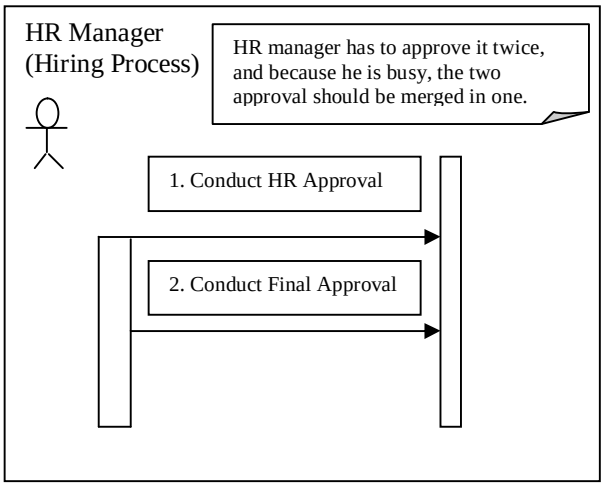

Figure 8: Analyzed Sequence Diagram

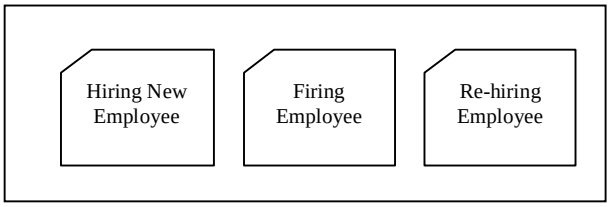

Figure 9: Sub-package for hiring processes

7. Conclusion

Literature shows that there are common issues or problems in Business Process Redesign (BPR) and Systems Development (SD). In addition, the literature also shows that both areas have very similar traditional life cycles. In fact, these similarities are the primary motivation of proposing a complete framework for BPR, which is inherited from a well-known framework in SD environment called the Unified Process (UP).

The proposed framework produces an iterative life cycle, which simulates the reality of BPR projects. In addition, the framework uses a common language, UML, among all stakeholders, such as managerial and technical people, and this helps in protecting the flow of information through the life cycle phases.

Finally, this paper creates opportunities for future research for improving the proposed framework and conducting different case studies in different organizations to examine test it. Also, studying the behavior of each phasein the proposed life cycle maybe interesting fand useful infuture research.

8. References

Boar, B.H. (1993), "The art of strategic planning for information technology: crafting strategy for the 1990s", John Wiley & Sons, New York, NY.

Hammer, M. (1990), "Reengineering work: don't automate, obliterate", Harvard Business Review, Vol. 68 No. 4, pp. 104-2.

Attaran, M. (2004), Exploring the relationship between information technology and business process reengineering. Information & Management, 41(5), 585.

Grant, D. (2002), A wider view of business process reengineering. Commun. ACM, 45(2), 85-90.

Bashein, B.J., Markus, L., and Riley, P (1994), Preconditions for BPR success: And how to prevent failures. Information Systems Management, 7–13.

Davenport, T. and Beers, M (1995), Managing information about processes. Journal of Management Information Systems 12, 1 57–80.

Kim, C. (1998), A model for the planning of business process reengineering. Journal of Computer Information Systems 39, 1, 84–90.

Klein, M. (1994), The most fatal reengineering mistakes. Information Strategy: The Executive Journal, 21–26.

James Rumbaugh, Ivar Jacobsen, Grady Booch (1999), The Unified Modeling Language Reference Manual. Addison-Wesley. Reading, Massachusettes.

Alan, E., Hanifa, S., & Andrea, R. (2008). A model for improving the role of IT in BPR.Business Process Management Journal, 14(5), 629.

Farzaneh, F. (2003). TQM vs. BPR.Quality Progress, 36(10), 59.

Ganesh, D. B. (2000). Exploring the relationship between information technology, infrastructure and business process re-engineering. Business Process Management Journal, 6(2), 139.

Dewalt, C. (1999). Business Process Modeling with UML. Johns Hopkins University.