REENGINEERING SYSTEM

GENERAL OBJECTIVES OF THE SUBJECT

At the end of the course, Individuals will analyze the characteristics of the materials in the industries, taking into account its advantages and functionality, for their proper application according to the same.

1. REENGINEERING SYSTEM

1.1 Introduction of Reengineering System

At the beginning of the 21st century, process thinking and Business process reengineering (BPR) (Hammer and Champy, 1993) have become mainstream thinking for business people and systems people alike. At the same time, methodologies for application development have matured and substantial standardization is taken place in the field of modeling techniques, e.g. UML (Kruchten, 2000). However, little attention has been paid so far how application development should be aligned with the new design of a business process. It is not difficult to imagine the consequences of such a lack of alignment. It is shown in this paper that these consequences are indeed encountered in practice.

The contribution of this paper is that it presents a methodology which links the redesign of a business process to the development of information systems that support such a design. The tightness of this link is accomplished by showing how the different proposed models used during the phases of a BPR project are related to each other. Models are placed within an architectural framework, which is a variant on that of Zachman (1999).

The heart of the methodology is the Product Based Design (PBD) approach, which can be used to create efficient and effective business process designs (Reijers and Voorhoeve, 2000; Aalst, Reijers and Limam, 2001). The part of the methodology which is concerned with the actual application development can be classified as component based (CB) (Booch, 1994; Garlan and Shaw, 1993; Szyperski, 1998). A CB approach in systems development has several advantages over more traditional ones, such as a clearer separation of concerns and an increased control of the development project. Considerable attention within the presented methodology is paid to formal correctness of the derived models and their validation with end users. This is to ensure as much as possible that a

new process – once supported by its aligned information systems – will indeed render the performance as envisioned at the start of the BPR project.

The structure of the paper is as follows. First, we will give a frame of reference for the presented methodology. Then, our practical experience with BPR and application development in a recent project for a Dutch bank is described. The methodology is given in some detail, distinguishing several steps and, for each step, its intent and its deliverables. We have included a non-trivial example of developing a new business process in a financial environment and an overview of related work. Finally, we present a summary and the intended extensions of the presented methodology.

1.2 Product Based Design

Product Based Design (PBD) is an approach to derive business process designs in administrative settings. PBD is a prescriptive design method which basically translates a manufacturing concept to the world of administrative processes, such as found in banking, insurance, government, etc. Material Requirements Planning, often referred to as MRP-I, determines the production schedule based on the ordered quantifies, current stock, and the composition of a product as specified in a so-called *Bill-of-Material* (Orlicky, 1972).

In other words, production is driven by the structure of the product. With PBD the structure of an informational product, such as a mortgage loan or a social insurance permit, is decomposed into a structure of informational elements which are used to derive a process design. Actual information elements of an administrative product may be related to each other in several ways. Aalst et al. (2001) describe strategies for the derivation of an optimal process design on basis of the information element structure. The problem of deriving such a design is to select a proper set of production rules and subsequently order them in such a way that an optimal performance with respect to the optimization goals may be expected.

Actual application of PBD has rendered process designs that are radically different from the existing processes they replaced and render flow time reductions of up to 35 % and cost reductions of up to 75 % (Reijers and Voorhoeve, 2000; Crom and Reijers, 2001). Note that an important difference between PBD and traditional approaches is that PBD does not take the existing process as the starting point of the BPR initiative. Rather, it focuses on the very legitimization of the process: the products it should deliver.

1.3 An Architectural Framework

REENGINEERING SYSTEM

To discuss the use of the PBD deliverables in a system development effort it is useful to distinguish the architecture of an information system first. System architectures focus on the structure of a system, which comprises smaller *components*, the externally visible properties of those components, and the relationships among these components (Shaw and Garlan, 1996).

An ***architecture*** can be defined as the fundamental organization of a system embodied in its components, their inter-relationships, the relations to their environment, and the principles guiding its design and evolution (IEEE, 2000). System architectures are important because they provide descriptions of the system at various levels of abstraction. Hence, system architectures enable various stakeholders to deal effectively with the complexity of a system and to reason about it at various levels of abstraction. Moreover, architectures are a means to integrate the various views of a system (Kruchten, 1995).

We will use an architectural framework that is a variant of the popular information systems architecture of Zachman (1999). Zachman defines an information systems architecture as a set of architectural representations (or models) placed within two dimensions: the ***perspective*** and the ***description type*****.** The distinguished perspectives agree with the interests of different stakeholders in a system development effort. A manager's view on the system differs from that of an designer, a programmer's view may be completely different from both. Each of the participant's views is, however, relevant to develop the system successfully. We distinguish the following perspectives:

- ✓ the ***business perspective*** distinguishes the purpose of the system in terms of the objectives of the company
- ✓ the ***logical perspective*** focuses on a logical description of the information system (its functionality) to support the business goals,
- ✓ the ***technical perspective*** is concerned with the software that realizes the desired functionality of the information system,
- ✓ the ***infrastructural perspective*** involves the hardware and general software required for the business-specific software to be executed.

Note that in distinguishing these perspectives we condensed Zachman's original five layers into merely four. In our consultancy practice we experienced that these layers are better recognized by all stakeholders than the complete Zachman framework, which is often too complex for the problem class.
The second dimension of the framework involves the different ***types*** of descriptions one can make of an information system. These types are applicable for each of the distinguished perspectives. Zachman proposes the universal 'what', 'how' and 'where'

questions as the important types. At the same time, he admits that the 'who', 'when' and 'why' question may be just as important, but dismisses them from his framework. We believe that distinguishing the data, the functions, and the process type of descriptions on the one hand and components on the other hand can capture a comprehensive treatment of the most important issues:

- a ***data*** model involves a description of the relevant entities or objects,
- a ***function*** model focuses on a description of the involved functions or services,
- the ***process*** model expresses how data and functions are integrated into an ordered network,
- the ***component*** model introduces hierarchy and encapsulation and is therefore a mean to reduce and divide the complexity of a system.

Note that unlike the data, functional, and process model that focus on a ***single*** aspect of the ***entire*** **system**, a component integrates *all* aspects of a particular *part* of the system in its environment. Also note the similarities with the ARIS framework (IDS Scheer, 2001).

1.4 Practical Experience

The development methodology as presented and illustrated in the following sections is based on our experiences in a large BPR project. During the years 2000 and 2001, we participated in a project to redesign a major bank's process for handling credit applications of commercial parties. The group to which the bank belongs is a global financial institution of Dutch origin, active in the field of banking, insurance, and asset management in 65 countries with more than 100,000 employees.

The process that was redesigned is executed at all its Dutch offices and handles on a yearly basis some 25,000 applications for loans and credit facilities. The project also involved the development of new applications, systems integration with existing applications, and the introduction of a Workflow Management System to support the process execution.

At the start of the project in the beginning of 2000, the PBD methodology was selected for the technical redesign of the process. Simultaneously, the choice was made for a particular software development method. Directly from the start, two project teams were formed that respectively concerned themselves with the process design (the ***process team***) and application development/systems integration (the *IT team*). Both teams started off simultaneously. The IT team started to analyze the existing situation with respect to

the existing systems and architecture, while the process team tried to understand the process that was to be redesigned.

Subsequent activities of both teams focused on analyzing information requirements within the process, modeling found data dependencies, defining required services of applications, etc. These activities clearly overlapped, although the relations between the delivered models of the separate teams was unclear. After four months, the process team finished its design and handed it over to the IT team. Some inconsistencies became directly obvious between the needs of the proposed business process on the one hand and the proposed services to be delivered by the new applications on the other.

The IT team was given the responsibility of exactly finding and sorting out all the differences. However, the IT team found it difficult to break away from the form and the content of their initial models. As suspicion continued about the correctness and consistency of the various models, members of both teams proposed to build a prototype of the new process, including the rough functionality of the new applications.

The general management could, however, not be convinced of the cost-effectiveness of such a prototype. Application development then took off without a clear point of understanding between the involved parties. After half a year of development, applications were build that could not support the needs of the new business process from the perspective of the process team. In response, a different development methodology was selected but this did not improve the quality of the delivered applications. We identified the following two major issues that caused the troublesome course of the project:

- ***the lack of alignment between the process redesign and application development*** From the start of the project, it was unclear how the activities and the deliverables of the project teams were linked to each other. Although their perspectives are inevitably different, it was not clear how the various models were related during the entire course of the project.
- ***the lack of validation and verification activities early in the project.*** No points were introduced early in the project to test and assess the models rendered so far against the expectations of the various stakeholders.

1.5 A Development Methodology on the basis of PBD

REENGINEERING SYSTEM

A. Determine Existing Architecture

- ***Intent:*** The challenge of this first step is to analyze the current situation and to describe it in a number of consistent architectural models.
- ***Deliverables:*** The deliverable of this step is a complete architectural description of the information systems that needs to be changed, as well as its environment.

B. Redesign Process with PBD

- ***Intent:*** In the second step, a process redesign project is carried out. The goal is a new or an improved business process design. In this methodology, we propose the use of PBD.
- ***Deliverables:*** Applying PBD requires results in two detailed deliverables:
    1. An information element structure including production rules and
    2. A process design. Both models relate directly to the ***business*** needs, but are also of interest from information system designers.

C. Analyze Performance

- ***Intent:*** The performance of the design is determined with respect to performance indicators such as throughput time, service time, waiting time, occupancy rate, etc. Simulation or analytical approaches may be used for this purpose (see e.g. Desel and Erwin, 2000).
- ***Deliverables:*** If the design is satisfactory, then there are no new deliverables. In case the design is unsatisfactory, the design is altered. Another possibility is that a choice between various alternative designs is made after this step.

D. Design Logical User Interface

- ***Intent:*** In the third step of the methodology, the logical user interfaces of the systems that will support the process execution are defined.
- ***Deliverables:*** The design of logical user interfaces may result in an adapted process model, with tasks that are either fused or split up in sub-tasks. Moreover, for each task a logical user-interface is defined that can be used by GUI designers as starting point for further development.

E. Validate by Prototyping or Gaming

- ***Intent:*** The purpose of this control step is the user validation of the *content of the tasks* within the process design. This validation step is done by prototyping the new system or by gaming (Guha, Kettinger and Teng. 1993). A methodical

way of doing this on basis of PBD is presented by Crom and Reijers (2001), as well as a case description.

- ***Deliverables:*** The validation step renders an improved process model with respect to the content of the tasks. Also, different grouping of information elements on the user interface may be determined.

F. Componentize' Business and Create Class Model

- ***Intent:*** This main step aims at creating component and class models for the system development effort. They cover the data and function aspects in detail on the logical level.
- ***Deliverables:*** The specific deliverables of this step are a component model, a class model, a component interaction model, and a cross-reference table. In the first two models, the business data and services are structured into logical entities.

A component and class model at the business level define the relation between business components and classes on the one side and all the business processes on the other. The component models contain the following parts: a component structure, a class model, and a life cycle of the component itself (or of its principal class). In the third model – the cross reference table – the interaction between these structures is described. Also, the relation between the redesigned process and the components is specified. In a cross-reference table, the tasks of the process are listed on the horizontal axis and the methods of the components are listed on the vertical axis. An intersection point of two elements from the list, indicates the usage of the method provided by the component in the respective task.