The Service (Processing) Network Paradigm & BPR

Quotes from Harrison M. and Loch, C.
"Operations Management and **ReEngineering**", Unpublished, December 1995.

- Business Process ReEngineering (**BPR**) is without doubt the **most influential development in management thinking** to occur in the 1990's. ReEngineering concepts, publicly praised by business leaders and management gurus, have been driving organizational change in many leading companies of Europe and North America.
- BPR has two distinct and separable elements:
  - process *design* (engineering) and
  - change *management*.
- Our focus here is on **principles of (service) process design**.
- On one extreme it might be argued that reengineering concepts represent a new *foundation* for operations management (**OM**), rendering obsolete most or all of the field's traditional thinking about process design. At the other extreme, some dismiss BPR as a mere *fad*, a collection of catch phrases appealing to gullible managers who crave quick and easy solution to productivity problems. From the perspective of an OM professional, the latter position is certainly too harsh, because whatever its shortcomings, the reengineering movement has directed attention to the operations side of business, increasing top management *awareness of systems and processes* ...
- In focusing attention on processes as the means of achieving effective operations, reengineering leaders have reinforced a central theme of the 1980's *quality movement* and in that regard *BPR complements (***TQM***)* the work of Deming, Juran and other quality gurus. To be effective, organizations must put creative energy into the design, documentation and maintenance of processes that satisfy customer needs on a routine basis,..."

Quotes from Loch, C. "Operations Management and Reengineering", European Management Journal, 16, 306-317 1998. (A descendent of the unpublished article above.)

- As corporations turn away from cost cutting to growth generation, **BPR is disappearing** from the headlines, but **process redesign and improvements can be turned towards performance improvement and revenue generation** just as effectively as they have been used for efficiency gains in the past.

The I.E. angle: "ReEngineering is I.E. in a non-factory environment.

Our **Process models** are **Service Networks** (s-nets). These are *dynamic* networks, where the dynamics is that of serving human customers, either directly or through phone-calls, documents etc. Since the focus is often on the service *process*, we shall also refer to service networks as **Processing Networks**. A third terminology, **Queueing Networks** (q-nets), was motivated by the real-world examples at the beginning of the course (Recall: queues in services are analogous to inventories in manufacturing, with additional human attributes such as complaints, abandonment, spreading around negative impressions, etc.).

Service (Processing) Network Primitives and Representations:

Graph (Flow Chart, Process Map): focusing on the **resources**, or **activities (tasks)**, or both if possible.

Service process = dynamic flow of customers (tasks) or information, or both.

Primitives: **Customers, Activities, Resources, Queues, Protocols**.
Specifically:

- *Customers (jobs): the units that "flow" through (arrive, leave) the system.* Customers attributes are: arrivals, services, routes, patience, ...
- *Activities (tasks, services, operations): these are what the jobs are made of*, and they are partially ordered via *precedence constraints*, summarized in terms of a precedence diagram.
- *Resources: process the tasks.* Resources could be machines/computers or human servers. Resource attributes are: processing capacity (as opposed to storage capacity), customers' constituency,.... A resource's processing rate is bounded by its processing **capacity: maximal sustainable processing rate.** (In discrete events: capacity = 1 / (average service time).)
- *Queues/Buffers are where customers/activities wait for their service process to continue.* Queues have (static) capacity (which could, nevertheless, vary with time; ideally, it could also be set to infinity). Operational queues are either *resource* queues, where the wait is for a resource to become available, or *synchronization* queues, where the wait if for a precedence constraint to be fulfilled.
- *Protocols*: for admission, routing, scheduling, *data* archiving and retrieving procedures, quality monitoring, specification of performance measures, ...

The Processing Network Paradigm (an attempt at a "definition"): The (service) system is envisioned (modelled) as a graph whose nodes represent either activities or resources; customers (jobs) flow through the system as their tasks are performed by the resources; tasks processing adheres to precedence constraints and each resource serves the tasks within its constituency.

Examples of Service (Processing) Networks:

- Project Management (PERT/CMP), in a stochastic dynamic (multi-project) environment: the Israeli Electric Company, Critical-Chain/Buffer-Management.
- Arrest-to-Arraignment (ATA) Process: in Larson's paper we have a fork-join q-net, where the fork is into people, documents, information and the join is at the judge.
- Research and Development (R&D): two articles by Adler et. al. one in Management Science (details) and one in the Harvard Business Review (managerial).
- Software Development: IE&M project, difficult to model, only partially successful.
- Bug Management at Intel: a recently completed IE&M project, which won the Gutvirth prize for 2003. A summary (in the format of your Homework 4) appears in our website (Homeworks).
- QC Labs (Quality Control) in the Pharmaceutical and Biotech Industry: such labs are testing samples from production batches, in order to approve batch-releases (within 12 to 30 days). Quoting a graduating student of the course (who consulted for QC labs): "QC labs are run with an R&D mentality, high labor costs, and within an environment that is significantly stochastic".
- Examples from your experience, so far and in the future, are and will be most welcome.

Service (Processing) Network Models:

Mathematical Models: Queueing or Fluid Networks. (Contrast with Simulation, Physical).

A **Queueing Network (q-nets)** is conveniently conceptualized in terms of a graph, in which particles traverse arcs and are delayed at nodes. While being prevalently stochastic, viewing q-nets through a "deterministic eye" is often appropriate and useful. (See our bottleneck analysis of National Cranberry Cooperative RP1.) This gives rise to the **Fluid View**, which has been a central subject of the first part of our course.

The nodes of the graph that represents a q-net are single-station queueing models, as taught in elementary courses. These nodes are integrated by the customers (jobs) that "flow" among them. The simplest theoretically tractable models of q-nets are *Jackson* networks, in which customers are homogenous, hence services are associated with servers; we shall study these towards the end of the course. The next step are *BCMP/Kelly* networks, where customers are divided to types; their theory is taught in more advanced courses. Further generalizations are precedence constraints (fork-join, or split-match networks) and models with one-to-many correspondence between customer types and resources (skills-based routing, agile workforce). We shall touch on the latter later on in the course. Fork-join nets will be described in today's lecture, via simulations.

(Re)*Engineering, or I.E.* perspective: adding to the fluid view the outcomes of Work (Time and Motion) Studies, captured in terms of say histograms, would give rise to service networks as described above. This is one way of understanding what is meant by the *Service (Processing) Network Paradigm*.

Three (sometimes Four) Main Steps in Analyzing a Service Network (demonstrated in class via DS-PERT/CMP networks (fork-join networks)):

1. **Can we do it?** Deterministic capacity analysis, via process-flow diagrams (spreadsheets, linear programming), which identifies resource-bottlenecks (or at least such candidates) and yields utilization profiles.

2. **How long will it take?** Typically stochastic *response-time analysis*, via analytical q-net models (exact, approximations) or simulations, which yields congestion curves. Note: When predictable variability prevails and dominates then the Fluid View is appropriate; the analysis is then deterministic, via inventory buildup diagrams. (e.g., The trucks of National Cranberries, in class and readings.)

3. **Can we do better?** sensitivity and parametric ("what-if") analysis, of MOP's or scenarios, which yields directions and magnitudes for improvements.

4. **How much better can we do?** or simply: **What is optimal to do?** Optimal control (exact, asymptotic) that is typically difficult but more and more feasible