Using formal analysis techniques in business process redesign

Kees M. van Hee, Hajo A. Reijers

Faculty of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{wsinhee, hreijers}@win.tue.nl

Abstract. Formal analysis techniques can deliver important support during business process redesign efforts. This chapter points out the (potential) contribution of these formal analysis techniques by giving an outline on the subject first. Next, a specific, newly developed formal technique is discussed.

1 Formal techniques in business process redesign

A thorough analysis of several redesign alternatives can help to make the choice for the most effective solution. Also, analyzing the design of a new business process may point out whether once set redesign targets are still realistic. In this paragraph we will give a short background on business process redesign, so that the role of analysis techniques in general can be explained.

1.1 Business process redesign

'Panta rhei' is the famous adage of the ancient Greek philosopher Heraclitus: everything is always changing. Business processes are prime examples of this statement. Everywhere around the world, in almost every industry, processes are being fine-tuned, downsized, re-engineered, value-added and re-aligned.

Drivers behind this phenomenon are manifold. In the first place, companies feel the increasing pressure of a globalising market. Cost reduction has become prevalent to survive. Secondly, the historically strong position of suppliers in many markets is becoming less dominant compared to that of the customer. To keep customers coming back, companies have to please them by shortening their production time or increasing their product quality. The last major change driver is technology. Technology offers a wide variety of new possibilities to manage business process better. The widespread application of Enterprise Resource Planning Systems and Workflow Management Systems in industry is a strong example on this note.

A business process re-engineering effort (to use one of the many possible labels) may aim at stabilizing, reducing, or improving one or more, different and often dependent entities (e.g. cost, production time, service quality, efficiency). Many business processes changes are put in motion without an accurate picture of the expected earnings at forehand, but rather on a ‘gut feeling’. There may be a well-understood positive or negative effect of the process change on entities such as the throughput time or production cost, but a reliable quantitative estimate is often lacking.

1.2 Phases in redesign

The redesign of a business process is a complex activity. Specialists from different disciplines are involved, new technology is introduced, staff is confronted with drastic change, and money and time are in short supply. To handle this complexity, a redesign effort is often carried out in the form of a project. This means, among other things, that during the effort distinct *phases* are distinguished in which different activities are planned. We will describe a generally applicable model to distinguish such phases within a redesign project, based on the IPSO methodology as described in [3]. The six phases are:

1. *Create vision*:
   A vision is created on the desired and necessary changes within an organization to improve its performance. A proper recognition of the organization's *Critical Success Factors* (CSF's) is required: which organizational issues determine success or failure of the organization? For these (or some) CSF's redesign targets are set. Next, it has to be established which *business processes* contribute mostly to the development of these CSF's. This determines the scope of the redesign. Lastly, the new organizational principles and technology have to be identified that will be the backbone of the redesign. In essence, the result of phase 1 is a *conceptual design* of the change.
2. *Diagnosis*:
   A thorough analysis of the selected processes is carried out to obtain an understanding of their current performance. This performance is usually expressed in terms of rates on *key performance indicators* (KPI's) such as the throughput time, customer satisfaction, product quality, etc. Next, the causes of low scores on KPI's within the current processes are determined, if possible. This knowledge can be applied in phase 3.
3. *Process redesign*:
   For the selected scope of the redesign, targets of the most important KPI's are set, in line with the desired development of the CSF's. Then, alternative process redesigns are developed in the form of *models*. The alternative models are to be analyzed and compared. Eventually, one of the models is selected as the definite redesign, after which it is detailed. The definite redesign is to be analyzed to ensure that set goals can be satisfied. From the detailed process design the requirements follow on the necessary supporting technology and information systems within the processes. At this point, there is a *detailed design* of the change.
4. *System design and construction*:
   On basis of the requirements of the definite process design, an outline of the technology components must be made, as well as their interaction patterns. This is the architecture of the new process. Furthermore, (information) systems and applications have to be designed that enable the new process to function in correspondence with its detailed design. Finally, both the technology architecture and the applications have to be actually constructed and tested.
5. *Transfer and implementation*:
   The newly designed process in combination with the supporting technology is integrated and transferred to the organization. New procedures, system functionality, communication structures, and responsibilities have to be communicated and explained to the relevant parties (e.g. employees, management, and customers). Feedback is generated to fine-tune the new process, after which it can be taken into full operation.
6. *Evaluation*:
   After implementation, a continuous phase of monitoring starts. The KPI's that have been determined in the diagnosis phase will be measured on regular intervals. This will usually trigger gradual adjustment and improvements of the new process(es). On basis of this information it can be determined whether the redesign effort is a success.

1.3 Formal analysis techniques for business processes

In general, there are two different categories of formal analysis techniques that can be used in the context of redesigning business process: *qualitative* and *quantitative* techniques. Qualitative techniques focus on the question whether a process design meets a specific property. Quantitative techniques are used to calculate or approximate the size or level of a specific property. For example, whether a process design meets the demand that a bank employee never can validate a cash transfer that he has initiated himself is a qualitative question. To determine how long customers have to wait before their telephone call is responded to by the call-center typically a quantitative analysis is required.

Quantitative techniques can be categorized into *simulation* and *analytical techniques*. If one compares these two types of techniques, it can be said that simulation is an approximation technique, where analytical techniques deliver exact numbers. During a simulation of a business process, at specified intervals *cases* (e.g. new orders) are generated for the model in execution. In response, each of the components within the model will behave in accordance with its specification. For instance, on receipt of a new order the computer will simulate an employee inspecting the order on completeness. The actions performed by the model in execution are realistic, but are not necessarily exactly the same or take place at the same moment as in a real-life situation. During execution, information is gathered on items that result from the interaction of the modeled components. For example, the frequency of message exchanges between two specific components is measured or the accumulation of work in front of an overloaded resource.

An analytical technique, on the other hand, is based on an *algorithm* that yields an exact result on basis of both the formal model and some well-understood relationships between the specified components. For example, a business process can be modeled as a network of nodes connected to each other by arcs, expressing precedence relations. On basis of such a network model, the shortest path leading from a new order to fulfillment can be calculated. Popular formalisms and mathematical theories to model and analyze business processes in this analytical way are, for example, Markov chains, queuing theory, CPM, PERT and GERT (e.g. [6], [7], and [9]).

Often, an exact result is preferred over an approximated result. However, the complexity of a specific business process model can be such that a quantitative, simulation approach is the only feasible means of analysis. Given a specific process model, there are several aspects that determine whether a qualitative or quantitatively analytical approach is feasible at all and, if so, preferable over simulation. For example, if both the synchronization structures within a process (e.g. parallelism) and the behavior of resources is too complex, no known general analytical techniques are available to determine the throughput patterns of work packages. Although simulation is a very flexible technique suited to investigate almost any type of business process, a common disadvantage is that, in non-trivial situations, numerous and lengthy simulation runs have to be carried out to obtain reliable results.

For all types of analysis, qualitative or quantitative, holds that a formal model of the business process underlies the analysis. Depending on the set of properties that is taken into consideration in the redesign effort, elements of the real business process are incorporated in the model. If, for example, the redesign effort is primarily concerned with the optimization of the *logistics* of the process, elements typically found in a process model are buffers, resources, routings of jobs, service times, and order arrivals. If, for example, the accent is on *cost reduction*, elements such as labor time, material costs, and depreciation factors will be part of the model.

1.4 Relevance of formal techniques

On basis of the phases within a process redesign project we have distinguished in paragraph 1.2 and the outline of techniques in paragraph 1.3, we can now pinpoint the places in the redesign process where analysis techniques can be useful. More in specific, whether a qualitative or quantitative is useful. As it turns out, the diagnosis and the redesign phases are best served by applying formal analysis techniques.

Diagnosis phase

As the first phase in a redesign project, the vision phase, is concerned creating prescriptive targets and means, it is not very suitable to apply formal analysis methods. During the subsequent, diagnosis phase it is paramount to come to a clear conception of the current process operations in practice. It can be argued that this could be established for a great deal on basis of observation and data gathering alone. Nonetheless, there are at least two situations when applications of analysis techniques are useful. In the first situation, it turns out to be difficult to understand the relations between different entities within a process by observation alone. For example, although historic information on resource availability and service quality at a specific work center is known, it is unclear how these two are related. The use of, for instance, a simulation model may indicate which of the alternative interaction patterns is most suited to explain current operations. It is rather awkward to interfere in the day-to-day operations to establish such a relation without a simulation.

The second situation occurs when one is interested in the *future* operations of a process. Clearly, this cannot be established by observing current processes alone. A future analysis can be performed by constructing a formal model of the current business process, and experiment with changes in the context of the process by inserting (partly) imaginary data, and analyze it accordingly. For example, a foreseen rise in demand can be projected on the model of the current business process, while keeping resources stationary. Simulation runs, then, may indicate where and in which sequence overload of capacity will occur. Qualitative analysis may point out whether dangerous or unwanted situations may occur.

Redesign phase
The subsequent phase, the redesign phase, is primarily a creative process aimed at developing new designs. Once developed, the alternative models have to be compared and the optimal solution is modeled in detail. There are three areas, normally dealt with in consecutive order during this phase, in which formal techniques can be useful:

1. Design construction,
2. Performance analysis,
3. Verification of design.

Ad 1. The design of a new process is primarily a creative process, which is likely to incorporate some level of human ingenuity for some time to come. Nonetheless, formal analytical techniques can offer support during the creation of a new design. On basis of desired properties of the process model on the one hand and characteristics of the elementary design blocks on the other hand, a formal technique may suggest (near) optimal design constructions or indicate the boundaries that the process design should stay within. For example, suppose there is a number of machines, each with a known capability. On basis of the characteristics of the desired end product, an algorithm can be used to, first, determine the required machine capabilities and, next, to indicate the alternative sets of machines that should be minimally available within the process. The designer makes the ultimate decision about which machines are chosen and at what locations within the process the machines are placed. In paragraph 2 we will describe in more detail an algorithm that can be used within this area.

Ad 2. Having created several designs, performance analyses are useful to compare these designs on how well they perform in different fields. Simulation can be used in much the same way such as described in the diagnosis phase. However, the creation of detailed simulation models and the subsequent lengthy simulation runs themselves may take too much time when roughly considering many alternatives redesigns. Analytical approaches that aim at measuring specific aspects of the designs in efficient ways are much more useful at this time. Depending on the performance indicator that is of prime interest, specific algorithms are applied. An example of such an algorithm, the computation of the throughput time, is treated in more detail in paragraph 2. In the end, when the preferred process design is established it may become much more interesting to develop a full simulation model to achieve an accurate view of the expected outcomes of the redesign effort. This may lead to tuning the goals to realistic levels.

Ad 3. Before the detailed process design is actually transferred to subsequent phases, such as the system construction, it should be established that the design is *correct*. This is very important considering the cost involved in developing a new process only to change it directly when it has been established. Although simulations used to assess the performance may have indicated no errors, this cannot be taken as that the process incorporates no faults. After all, simulations cover only a finite number of situations. Analytic, qualitative techniques may be used to determine important properties of the design, such as the absence of dead-locks or the eventual termination of the process once work is taken on.

2. Throughput analysis

In this paragraph, we will describe a recently developed formal technique that can be used when redesigning business processes ([4]). It focuses on the computation of the throughput time of a specific business process model. We start by introducing the throughput concept. Next, we will describe the construction of business process models within this approach and the algorithm to compute the throughput time of the business process modeled in such a fashion.

2.1 Throughput time

One of the most important performance indicators in industry is the *throughput time*. Although authors from different disciplines use different terminology for this concept such as *passage*, *cycle* and *traversing time*, we stick to our term for reasons of popularity in the field of workflow management from which our approach originates. The throughput time of a *specific job* is the total amount of time spent from the moment that the handling of the job started until the moment it is completed. The throughput time of a job is the sum of its *service time* and its *waiting time*. Waiting time for a job is created when no task can be executed due to the unavailability of resources. When there is at least one task processing a job, this counts as service time.

The wide-spread use of the throughput performance indicator can be explained from the fact that it is concerned with the 'flowing' of work through the business process, rather than with the exact manipulations that take place. Very often, a low or stable throughput time is a desirable or even necessary characteristic of a business process. Imagine, for instance, a government agency that handles tax forms and decides whether they are valid. National regulations may be violated when the processing of a job takes over one year.

The throughput time *of a process* can be expressed in several ways. This is caused by the fact that jobs that undergo the same processing often do not share the same throughput time. In other words, there is throughput variance. An ordinary cause for this phenomenon is that resources do not deliver constant productivity. Another cause may be fluctuations in market demand, possibly flooding the system, leading to waiting time. A very common approach is to express the throughput time of a process as the average throughput time of the jobs it handles. Although this may be fine as an approximation, this average is not always a good reflector of the performance of the process. For example, if minimum and maximum throughput times of jobs are far apart, the average throughput time is hardly suitable to give customers guarantees about delivery times. An alternative sometimes used, is to declare the throughput time of a process by means of a fraction percentage and a cut-off value. For example, 90 % of the jobs going through a specific business process is finished within 6 weeks. If the throughput of jobs varies, the most detailed expression of the throughput time is as a histogram or a probability distribution of the job throughput times.

Regardless of the exact definition used, the computation of the throughput time for a business process already in action is straightforward. Actual throughput figures on job throughput times can be used to express the throughput time following either definition. A problem arises, when the throughput time is to be determined of a newly designed process. Depending on historic information only puts the designer in an awkward position. He cannot design a process with desirable throughput characteristics without putting the process to work first. Especially when design alternatives are to be compared, such as required in the redesign phase sketched in paragraph 1.2, this is not very practical.

A proper alternative, propagated in this chapter, is to apply formal quantitative techniques on a model of the designed business process. One of the possible approaches is to apply simulation. As argued before, simulation can be time consuming. An analytical approach that is time-efficient and yet powerful to yield reliable results (by being exact) would therefore be preferable in this situation. In the next paragraph we will explain the basics of such an approach.

2.2 Elements of the process model

When choosing an analytical approach, it is important to incorporate those aspects within the formal model that determine the throughput time of the process. Bringing back in mind the distinction between service time and waiting time, the following aspects are relevant:

1. the *structure* of the business processes: a business process is a set of tasks that have to be completed in some kind of order (possibly differing for each new job); the possible routes through the process, leading to the execution of individual tasks, is essential for determining the throughput time,
2. the *resource schedule*: the way how resources are distributed over the different tasks within the process; both the available type of resources and the number of resources may determine the flow of jobs through specific points in the process,
3. the *service characteristics* of the resources active within the process: differences in service productivity per resource influence throughput time,

4. the *arrival rate* of new jobs: the balance between new arrivals and available resources determine whether waiting time arises.

When starting to model a business process, it is generally up to the modeler how accurate each of these aspects is modeled. There are, however, practical limits to this accuracy. Usually it is not feasible to model the characteristics of each individual job or resource. Instead, *classes* of different jobs and resources are modeled. A certain pattern of behavior, then, holds for each member of a class.

A second practical limitation is that the exact chain of cause and effect *underlying* specific behavior is unknown or perhaps irrelevant. For example, the arrival of new orders may depend on the price level of a competitive product. That price strategy may be unknown.

Moreover, the exact cause for, for example, a resource to work slower at some times is less relevant than accurately modeling the phenomenon itself. A stochastic approach is then the answer. This means that, given a specific component of the model, relevant patterns of behavior are distinguished. Each pattern is assigned a probability weight, instead of modeling the specific cause. For example, fluctuations in the arrival of new orders are expressed by a probability distribution. A common characteristic to model an arrival pattern of new cases is a standard Poisson distribution.

Similarly, to model changes in the productivity of a resource, a distribution is used that asserts probabilities to each possible service time that resource may deliver. Realistic behavior is induced even more when stochastic processes are modeled to be dependent on each other. Specific resource behavior may be dependent on the resource behavior shown earlier. On the other hand, to simplify calculations, many times standard stochastic distributions are used, or even deterministic behavior. More realism can be achieved by using arbitrary or mixed distributions.

Even when (3) the service characteristics and (4) the arrival rate are modeled stochastically, analytical analysis is not straightforward. Both (1) a complex process structure or (2) a complex resource schedule may trouble the analysis. An example of a difficult resource schedule is the use of so-called "butter flies", resources that are not assigned to a fixed task within the process, but wander around. Difficult process structures are those in which other relations can be applied than mere sequential orderings of tasks, for example by allowing the parallel execution of tasks.

General applicable analytical techniques to compute the performance of a business process model with an arbitrary, independent stochastic arrival pattern, with arbitrary, independent stochastic service characteristics, with a complex resource schedule, and with a complex process structure are not available. Reduction of complexity of the model is therefore required. Many existing performance analysis techniques concentrate on omitting complex process structures from the model, such as the choice, parallel, and cycle constructions. In addition, stochastic resource behavior or stochastic arrival patterns are modeled using standard probabilistic distributions, such as normal distributions or negative-exponential distributions.

In the approach presented in this paragraph, we will apply reductions in another dimension, allowing for greater realism in two other dimensions. Our approach comes down at assuming a liberal resource schedule. No matter the number or types of resources required for the execution of a task, we will assume that sufficient resources are always available when there is a job to process by that task. In other words, the resource capacity is infinite; no waiting time can occur due to the lack of resources. The throughput time of a job under these conditions is equal to the total service time spent. In a real business process, of course, this situation will hardly ever occur. It means that the process is either in expensive excess of capacity or in a terrible lack of orders. On the other hand, this assumption allows for more accurate modeling in two other dimensions, typically neglected in other approaches. In the first place, there is a great set of possibilities to investigate the effect of complex process structures on the throughput time. It is possible to apply choice, parallel, *and* cycle constructions. Secondly, its is possible to achieve a much higher level of accuracy when modeling the service characteristics by using arbitrary (yet independent) probability distributions. Note that the exact arrival pattern has become irrelevant, because handling of a specific job cannot influence the throughput of another job.

In the next paragraph we will start the formal description of the approach. The basic framework of the model used are high-level stochastic Petri Nets.

2.3 Petri nets

To create the structure of a business process, we will use classical Petri nets. For a formal definition, the reader is referred to [2] or [5]. A Petri net is a triple (P, T, F) that consists of two node types called *places* and *transitions*, and a flow relation between them. We will use places to model milestones reached within a business process and transitions as the individual tasks within the business process to execute. Places are represented by circles; transitions are represented by rectangles. The process constructions that can be applied in our approach to build a business process are the so-called *blocks*. These are: sequence, choice, parallelism, and iteration. The blocks that express these constructs are depicted as Petri nets in Figure 1.

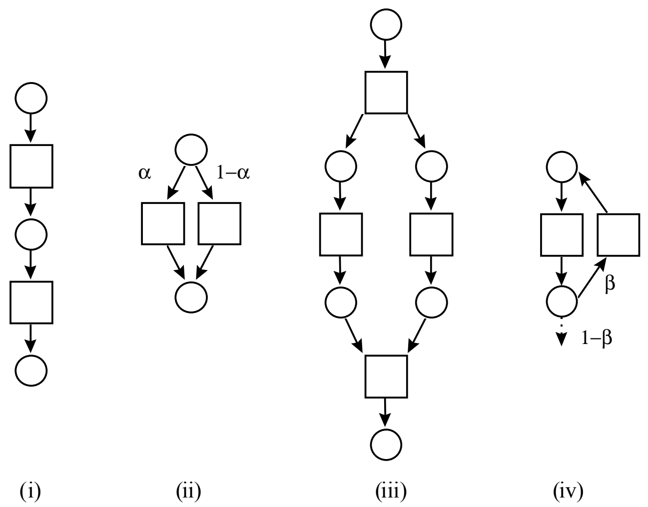

Fig. 1. Sequence (i), choice (ii), parallelism (iii), iteration (iv).

The first block, the sequence block, represents a process construction that puts two tasks in sequential order. The first task has to be completed before the second can be started. The second, choice block represents a construction in which exactly one of two alternatives is carried out. The third block shows how two tasks can be modeled such that they can be executed simultaneously. The last block, the iteration block, represents the process construction in which the execution of a task can be repeated.

Arc labels occur in the choice (ii) and iteration (iv) blocks. They represent the values of a Bernoulli-distributed random variable that is associated with these blocks. An independent draw from such a random variable determines the route of the flow. Each new application of such a block is accompanied by the introduction of a new, independent random variable.

As the starting point of each Petri net model construction we will take a simple start net. This net is depicted in Figure 2. We will refer to this specific Petri net as *SN*, for start net.

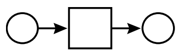

Fig. 2. The start net *SN*.

The next step is to extend the original net, by subsequently applying blocks on parts of the model that are similar to the start net. We shall clarify this replacement rule informally. In Figure 3 the creation of business process model is depicted as the result of successive applications of blocks. The initial start net is extended by applying the construction rule with the sequential block. Both within the upper and the lower half, constructions similar to the start can be distinguished. The upper half of the resulting net is transformed into a choice construction. On the lower half, the parallel composition is applied. Finally, the right path of the parallel construction is modified with the iteration block.

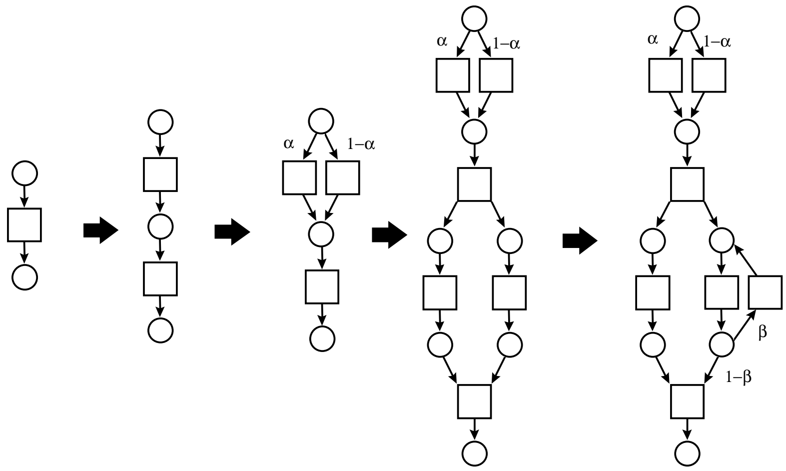

Fig. 3. Building a business process model.

The next step in building the business process model is to assign the service characteristics to the constructed model. In our approach, all service times for one specific task - a timed transition - are independently sampled on basis of the same probability distribution. We will call this distribution the *service distribution*. Its matching probability density is the *service density*. The time which is taken by a service of transition t is called the service time. The service time is a discrete random variable $\underline{t}$. Its matching probability density $f_t: \mathbb{N} \rightarrow \mathbb{R}$ is called the service density;

$$f_t(k) = \mathbb{P}(\underline{t} = k) \text{ , for } k \in \mathbb{N}.$$

Its matching probability distribution $F_t: \mathbb{N} \rightarrow \mathbb{R}$, is called the service distribution;

$$F_t(k) = \mathbb{P}(\underline{t} \leq k) \text{ , for } k \in \mathbb{N}.$$

The service time $\underline{t}$ is bounded: there is an upper bound $u_t \in \mathbb{N}$ which is the smallest value such that for all $j \in \mathbb{N}$ and $j \geq u_t$ holds that $f_t(j) = 0$.

Our assumption of the service time to be discrete is no real constraint: for practical purposes it is always possible to find an appropriate representation. As we will see, we do need the boundedness of the service time to perform some of the computations to come.

2.4 Blocks

In this paragraph it will be shown how the *throughput time* can be computed of a business process model that is constructed in the presented fashion. The throughput time of a business process is defined as the time that elapses between the arrival of a token at the source place and the corresponding arrival of a token in the sink place. Similar to the service time notions, it is possible to distinguish the throughput distribution and throughput density of a process. Assuming the service densities to be known of each of the transitions within a block, we will show how the throughput density of an entire block can be computed. Each of the blocks requires a specific algorithmic approach.

Sequence block
Consider the sequence block B in Figure 4 with two transitions *s* and *t*. Transition *t* can only be executed when *s* is completed.

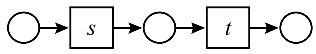

Fig. 4. Sequence block.

The throughput density f_B, given f_s and f_t, can be computed as follows:

Let $y \in \mathbb{N}$.

$$f_B(y)$$

= { all possible combinations over transitions s and t; y non-negative }

$$\sum_{i=0}^{y} \mathbb{P}(\underline{s} = i \wedge \underline{t} = y - i)$$

= { $\underline{s}$ and $\underline{t}$ probabilistically independent }

$$\sum_{i=0}^{y} \mathbb{P}(\underline{s} = i)\mathbb{P}(\underline{t} = y - i)$$

= { definition convolution; represent f_s and f_t as vectors, service density in i is the i^{th} coefficient of the vector }

$$f_s \otimes f_t(y)$$

A straightforward computation of the convolution $f_s \otimes f_t$ would require at least $u_s u_t$ multiplications (product of the upper bounds), a quadratic number. To make the computation more efficient, the Fast Fourier Transform (*FFT*) is applied. The FFT is an algorithm that computes the Discrete Fourier Transform (*DFT*) of a vector in $\theta(n \log n)$ steps. Computing a convolution of two vectors, then, comes down at multiplicating the Fourier Transforms of those vectors, after which this product has to be transformed back in a normal vector representation. Using the Fast Fourier transform, a vector representation of $f_s \otimes f_t$ can be computed in $\theta(n \log n)$ time, with n the smallest power of two that is at least twice as large as the maximum of the upper bounds of tasks s and t. For a thorough explanation of the Fourier Transform the reader is referred to [1].

Parallel block

The block we will consider next is the parallel block. Consider the parallel block B in Figure 5 with transitions *k*, *l*, *m,* and *n*.

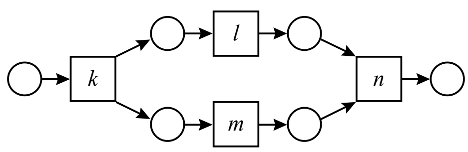

Fig. 5. Parallel block.

Due to the structure of B, transitions l and m can be executed in parallel. That is, there is no precedence constraint between transitions l and m. When both transitions l and m have ended, transition n can be executed. We want to compute throughput density f_B, given f_k, f_l, f_m, and f_n. Without loss of generality we assume that $f_k(0) = 1$ and $f_n(0) = 1$ (they are *logical* transitions).

Let $y \in \mathbb{N}$.

$$f_B(y)$$

= { structure block; k and n instantaneous }

$$\mathbb{P}(\underline{l} \max \underline{m} = y)$$

= { definition max }

$$\mathbb{P}((\underline{l} = y \wedge m \leq y) \vee (\underline{m} = y \wedge \underline{l} < y))$$

= { independent random variables }

$$f_l(y)\sum_{i=0}^{y} f_m(i) + f_m(y)\sum_{j=0}^{y-1} f_l(j)$$

= { definition service distribution; distinguish cases $y = 0$ and $y > 0$ }

$$\begin{cases} f_l(y)F_m(y) + f_m(y)F_l(y-1), & y > 0 \\ f_l(y)f_m(y), & y = 0 \end{cases}$$

The computation of the distribution function F_m can be done in u_m steps, just as the distribution function F_l can be computed in u_l steps. Therefore, the total computation of f_B can be done in $\theta(t)$ time, with t equal to the maximum of upper bounds u_l and u_m.

Choice block
The next block we will consider is the choice block. The choice block B is depicted in Figure 6.

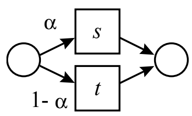

Fig. 6. Choice block.

The block B consists of two transitions s and t. Initiating block B results in either the execution of transition s or transition t, with respective chances α and 1 - α. When the selected transition is completed, the block itself is completed. We would like to compute throughput density f_B, given f_s and f_t.

Let $y \in \mathbb{N}$.

$$f_B(y)$$

= { structure block; introduce random variable $\underline{I}$ which determines whether transition s or transition t is executed }

$$\mathbb{P}((\underline{I} = s \wedge \underline{s} = y) \vee (\underline{I} = t \wedge \underline{t} = y))$$

= { $\mathbb{P}(\underline{I} = \text{s}) = \alpha$; $\mathbb{P}(\underline{I} = \text{t}) = 1 - \alpha$; $\underline{I}$, $\underline{s}$, and $\underline{t}$ are independent }

$$\alpha f_s(y) + (1-\alpha) f_t(y)$$

From the last expression follows that we can compute f_B in $\theta(n)$ time, with n equal to the maximum of u_s and u_t.

Iteration block
The final and most complex block is the iteration block, depicted in Figure 7. It consists of transitions t and u. The choice for either transition u or termination after completion of transition t is a matter of chance. After each firing of transition t transition u will fire with probability α.

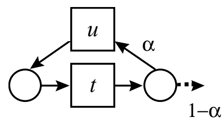

Fig. 7. Iteration block.

The throughput density f_B, given f_t and f_u, can be computed as follows:

Let y ∈ $\mathbb{N}$.

$$f_B(y)$$

= { t will always be executed one time more than u; define random variable $\underline{n}$ as the number of times that transition u is executed }

$$\sum_{n=0}^{\infty} \mathbb{P}\left(\sum_{j=1}^{n+1} \underline{t}_j + \sum_{j=1}^{n} \underline{u}_j = y \wedge \underline{n} = n \right)$$

= { $\underline{n}$ has geometrical distribution with $\mathbb{P}(\underline{n} = n) = (1 - \alpha)\, \alpha^n$; random variables independent }

$$\sum_{n=0}^{\infty} (1-\alpha)\alpha^n \mathbb{P}\left(\sum_{j=1}^{n+1} \underline{t}_j + \sum_{j=1}^{n} \underline{u}_j = y \right)$$

= { definition of service density f; service times are based on same service density; definition convolution; introduce notation $\bigotimes_{j=1}^{n} a_j = a_1 \otimes a_2 ... \otimes a_n$ }

$$\sum_{n=0}^{\infty} (1-\alpha)\alpha^n \mathbb{P}\left(\bigotimes_{j=1}^{n+1} f_t \otimes \bigotimes_{j=1}^{n} f_u(y) \right)$$

At this point we realize that a more suitable computational approach is to derive the Discrete Fourier Transform of the vector representation of f_B from the transforms of $\vec{t}$ and $\vec{u}$. Doing so, it is possible to use an inverse transformation to actually compute $\vec{B}$ - the vector representation of f_B. Leaving undecided yet what is the proper length of $\vec{B}$ we denote the index of the *DFT* with *l*.

$$DFT_l(\vec{B})$$

= { recall derivation of $f_B(y)$; use · for pointwise vector multiplication and + for pointwise addition }

$$DFT_l\left(\sum_{n=0}^{\infty} (1-\alpha)\alpha^n \left(\bigotimes_{j=1}^{n+1} \vec{t} \otimes \bigotimes_{j=1}^{n} \vec{u} \right) \right)$$

= { *DFT* distributes over multiplication and addition }

$$\sum_{n=0}^{\infty} (1-\alpha)\alpha^n DFT_l\left(\bigotimes_{j=1}^{n+1} \vec{t} \otimes \bigotimes_{j=1}^{n} \vec{u} \right)$$

= { convolution theorem; convolution is associative }

$$\sum_{n=0}^{\infty} (1-\alpha)\alpha^n DFT_l^{n+1}(\vec{t})\ DFT_l^{n}(\vec{u})$$

= { calculus }

$$\frac{(1-\alpha)DFT_l(\vec{t})}{1-\alpha DFT_l(\vec{t})DFT_l(\vec{u})}$$

Obviously, we can not expect f_B to have an upper bound. After all, t and u could be executed infinitely often if α is non-zero. So by choosing n as proposed we may end up with a vector representation of f_B that is too short. That is, there may be interesting values of f_B that will not be represented. We will show how a relevant, estimated length of $\vec{B}$ can be determined before actually computing $\vec{B}$. We would be most pleased to find a value v such that for some very small ε holds:

$$\mathbb{P}(\underline{B} \geq v) \leq \varepsilon \qquad \text{(i)}$$

We are looking for an estimation of v that takes the distribution of values within f_t and f_u into account. For this purpose we will use Chebyshev‘s inequality, that we present without proof (see [8]). This inequality is often used to find a rough upper bound in mostly theoretical applications.

Theorem (Chebyshev's inequality) For any random variable $\underline{x}$ for which $E\underline{x}^2$ exists:

$$\mathbb{P}(|\underline{x} - E\underline{x}| \geq c) \leq \frac{\text{var}\,\underline{x}}{c^2}$$

With this inequality in the one hand, and the mean and variance of the throughput density f_B in the other, it can be determined which probability part of the density falls before or after a hypothetical border. As the service time for any transition t is denoted by $\underline{t}$ we will denote its mean by $E\underline{t}$ and its variance by var $\underline{t}$.

$$E\underline{B}$$

= { structure block}

$$E\underline{t} + \alpha(E\underline{u} + E\underline{B})$$

= { calculus }

$$\frac{E\underline{t} + \alpha E\underline{u}}{1-\alpha}$$

The computation of the variance of the iteration block is as follows.

$$\text{var}\,\underline{B}$$

= { structure block; definition variance }

$$\alpha\,\text{var}(\underline{t} + \underline{u} + \underline{B}) + (1-\alpha)\,\text{var}\,\underline{t} + \alpha(1-\alpha)(E(\underline{t} + \underline{u} + \underline{B}) - E\underline{t})^2$$

= { calculus; previous result for $E\underline{B}$ }

$$\frac{\text{var}\,\underline{t} + \alpha\,\text{var}\,\underline{u}}{1-\alpha} + \alpha\left(\frac{E\underline{u} + E\underline{t}}{1-\alpha}\right)^2$$

With Chebyshev's inequality we can determine a relevant part of vector $\vec{B}$ by modifying our original requirement (i) with:

$$\mathbb{P}(\underline{B} \geq v \vee \underline{B} \leq -v) \leq \varepsilon \qquad \text{(ii)}$$

This is only apparently a stronger requirement than (i), as $\underline{B}$ can never be negative and the sum of positive values of f_B is 1. Immediate application of Chebyshev on equation (ii) is possible and yields the following:

$$v \geq c + E\underline{B}$$

and

$$\varepsilon = \frac{\text{var}\,\underline{B}}{c^2}.$$

From these equalities we can derive that:

$$v \geq E\underline{B} + \sqrt{\frac{\text{var}\,\underline{B}}{\varepsilon}}.$$

Concluding, given f_t and f_u, we can compute a vector representation of f_B for the iteration block by using the *DFT*:

$$DFT_v(\vec{B}) = \frac{(1-\alpha)DFT_v(\vec{t})}{1-\alpha DFT_v(\vec{t})DFT_v(\vec{u})}$$

with

v is the smallest power of two such that $v \geq E\underline{B} + \sqrt{\dfrac{\operatorname{var} \underline{B}}{\varepsilon}}$, and

$$\operatorname{var} \underline{B} = \frac{\operatorname{var} \underline{t} + \alpha \operatorname{var} \underline{u}}{1-\alpha} + \alpha \left(\frac{E\underline{u} + E\underline{t}}{1-\alpha} \right)^2 .$$

With the *FFT* we can compute a vector representation of f_B in (v log v) time, with v as specified. To appreciate its efficiency we have to establish the computing time of calculating f_B in a straightforward manner. The complexity of this calculation depends on the maximal number of successive times that transitions t and u can be executed. We know that if both $f_t(0)$ and $f_u(0)$ are equal to zero, at most v executions of these transitions are of interest. Any more executions of transitions u and t would result in throughput times that we do not take into consideration. As a result, a straightforward approach requires the convolution of v times the function f_t and f_u. This is an operation requiring $\theta(n^v)$ time, with n the maximum of upper bounds of transitions t and u. A comparison with the $\theta(v \log v)$ time required by our newly found computation method illustrates the efficiency of the latter.

2.5 Overall computation

Suppose we have constructed a business process model from the start model *SN* with n subsequent applications of the construction rule. Then, for $n > 1$ we can distinguish the intermediate models W_1, $W_2, \ldots, W_{n-1}$. W_1 is the result of the application of the construction rule using one of the blocks on the start model S; W_n results from the n^{th} application of the construction rule on intermediate model W_{n-1}. For each of the other intermediate models W_i holds that it is the result from the i^{th} application of the construction rule on intermediate model W_{i-1}. Note that we can represent all intermediate process models hierarchically in a derivation tree. It is, in fact, this derivation tree that we will step through during our computation.

To compute the total throughput time of the constructed process model W we assume that the service density of each of its transitions is known. In addition, all probability distributions that involve choices or iterations are known too. Now we take the opposite direction of the construction route. Starting at the end, we consider the last application of the construction rule that leads from W_{n-1} to W. We know which part of W_{n-1} is replaced by which specific block. We call this part S_{n-1} and the block B_{n-1}. Recall that S_{n-1} is isomorphic with the start model. B_{n-1} is the part of which we compute its throughput density t_{n-1}. This is feasible, as we know all relevant service densities in W.

The resulting throughput density t_{n-1} is a fair characterization of the time that it takes B_{n-1} to process a job. What is more, the block can be seen as a detailed specification of the behavior of S_{n-1} that it has replaced. Therefore, the resulting throughput density is a fair characterization of the service time that it takes S_{n-1} to process a job as well. When the only transition in S_{n-1} should have a service density that is equal to t_{n-1}, the total throughput time of W_{n-1} would exactly be the same as that of W. S_{n-1} can be seen as a "black box" for the applied block. The effort of computing the throughput time of W has now become the effort to compute the throughput time of W_{n-1}.

We can repeat this approach for each of the n transformations. When we ensure that we follow the construction route in opposite direction, we can be confident that all relevant data is available to compute the throughput time of each start model that has been replaced. Finally, we end up with the start block *SN*. This block will have only one transition of which its service density characterization is exactly the throughput characterization of the entire process model W. And this is exactly the throughput density we were looking for.

2.6 Numerical Experience

To give an indication of the efficiency of the presented algorithm to analyze the throughput time of a business process model in the above way, we have computed the throughput density of the rightmost process model depicted in Figure 3. As tool for these calculations we have used the mathematical software package Maple V® running on a Pentium 100 MHz computer. Each of the seven transitions in the process model has been modeled to behave in accordance with distinct service densities. These service densities, together with the process model under consideration, are depicted in Figure 8. The service densities in this example are bounded by 64 time units.

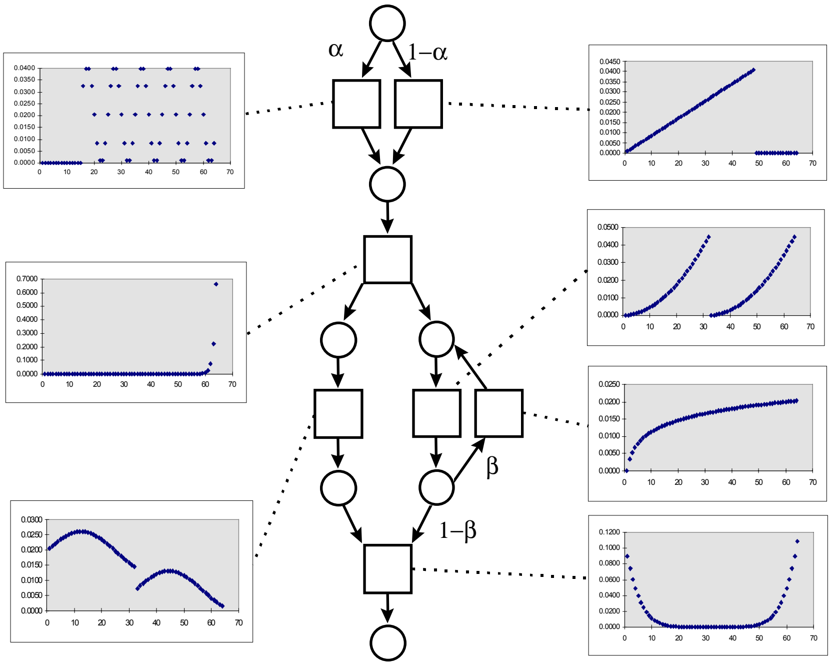

Fig. 8. Numerical example.

The probabilities α and β of the model have been set on 0.3 and 0.15 respectively. For the computation of the throughput density of the iteration block an inaccuracy (ε) of 1 percent has been allowed. The resulting throughput density for the entire net is depicted in Figure 9.

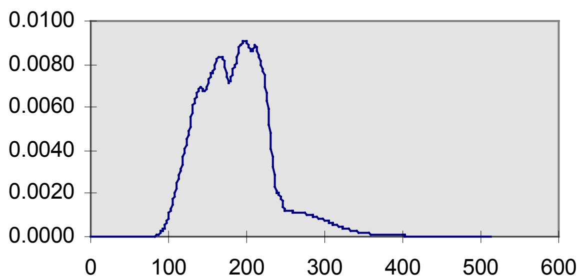

Fig. 9. Outcome throughput time.

The computation of the throughput density of the example net has been performed for several sizes of the throughput density domain. These results can be found in Table 1.

Table 1. Computation time for example

Bound (service time units)	64	128	256	512	1024	2048
Computation time (seconds)	13	28	64	152	377	990

As can be seen, the computation time increases somewhat faster than linearly in the bound dimension. It can be easily verified that the same relation exists between the number of transitions and the computation time.

3. Conclusion

In this chapter we have made a case for the application of formal analysis techniques during the redesign of business processes. We have identified the phases in a redesign project where formal techniques can be applied. On a more detailed level, we have shown how an analytical technique can be used to establish the throughput time of a business process model. This analysis technique is only one of the many possible quantitative analytical techniques that can offer support in this field. A redesign tool that incorporates these kinds of techniques, presenting relevant analytical characteristics of a business process can be a valuable asset to designers. In this way, redesign measures formerly only justified by “gut feeling” can be rationalised. Considering the money, time and stakes involved with BPR we would embrace an increased rationality of the redesign process that may be acquired with the use of a redesign tool.

References

1. Cormen, T. H., Leiseron, C.E., Rivest, R.L.: Introduction to Algorithms. The MIT Press, Cambridge (1990).
2. Hee, van, K.M.: Information Systems Engineering: a Formal Approach. Cambridge University Press (1994).
3. Hee, van, K.M., Aalst, van der, W.M.P.: Workflow Managament: Modellen, Methoden en Systemen. Academic Service, Schoonhoven (1997) [In Dutch]
4. Hee, van, K.M., Reijers, H.A.: An Analytical Method for Computing Throughput Times in Stochastic Workflow Nets. Proceedings of the 7th Analytical and Numerical Modelling Techniques Conference (1999) [Forthcoming]
5. Jensen, K: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Vol. 1, Basic Concepts. Springer-Verlag, Berlin (1992).
6. Levin, R., Kirkpatrick, C.: Planning and control with PERT/CPM. McGraw-Hill, New York (1966).
7. Neuman, K., Steinhardt, U.: GERT Networks, vol. 172 of Lecture Notes in Economic and Mathematical Systems. Springer-Verlag, Berlin (1979).
8. Thomasian, A.J.: The structure of Probability Theory with Applications. McGraw-Hill Book Company, New York (1969).
9. Tijms, H. C.: Stochastic Models: an Algorithmic Approach. John Wiley & Sons, New York (1994).