Computing the Cost of Business Processes

Dissertation

an der Fakultät für Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universität München

zur Erlangung des Grades

Doctor rerum naturalium (Dr. rer. nat.)

vorgelegt von

Partha B. Sampathkumaran

München, January 9, 2013

Erstberichterstatter: Prof. Dr. Martin Wirsing
Zweitberichterstatter: Prof. Dr. Hubert Baumeister
Tag des Rigorosums: 25.01.2013

To my parents,
without whom this would not have been possible.

Abstract

Designing and optimizing a business process based on its financial parameters is a challenging task. It requires well defined approaches, actions and recommendations which when implemented lead to tangible and quantifiable results. The resulting business process has to minimize the expenses for the business actors and maintain the estimated profitability.

In recent years, after service oriented architecture and business processes have taken center stage, a lot of research has been done to establish methodologies which evaluate and optimize business process implementation. New fields of research have come up forefront which makes implementation of business processes feasible and profitable for organizations. These establish a management perspective to the implementation of a service, stressing upon financial and economic factors such as returns, the cost of implementing a service or making sure that there is reliability in the service being offered. Existing frameworks recommend best practices which optimize business process by considering them with surrounding soft factors such as behavioral aspects of involved human resources and accordingly evaluate the success (or failure) of the process. Nevertheless computing the cost of a business process such that it is a tangible and measurable value continues to be a complicated and cumbersome process. Estimating the profitability of an idea before it is implemented is a difficult task and the need for techniques which allow us to do this as early as possible play a very important role in decision making. This requires business process design approaches and practices that incorporate techniques for tangible evaluation of expenses and benefits for each implementation.

In this work we propose a new methodology by which the cost of a business process is calculated by considering the cost and reliability of each action or task in the process. The methodology breaks the business process, represented using 'Business process model and notation (BPMN)', into repetitive patterns and a cost and reliability factor for each of these patterns is calculated. As a result the overall cost, reliability and the cost incurred to achieve one successful execution of the business process, the business cost of the process, is achieved. Based on this concept we propose an extension of financial and economic parameters to theoretical foundations representing service invocation and execution of long running transactions. We use Sagas as basis for this extension. Long running transactions are Sagas when they can be broken down as sequence of transactions which interleave with other transactions. We have implemented the new methodology to examine individual business processes with the help of sensitivity analysis so as to find problem areas where an optimization can be implemented. Using this methodology the study evaluates different existing frameworks and well-known best practices and their financial impact on the processes. The study is backed by an application (Java based) which automates the evaluation process.

Zusammenfassung

Die Gestaltung und Optimierung eines Geschäftsprozesses auf der Grundlage seiner finanziellen Parameter ist eine herausfordernde Aufgabe. Diese setzt wohldefinierte Ansätze, Handlungen und Empfehlungen voraus, die in ihrer Umsetzung zu konkreten und quantifizierbaren Ergebnissen führen sollen. Der daraus resultierende optimierte Geschäftsprozess soll die Aufwendungen für die Entscheidungsträger minimieren und die erwartete Rentabilität erhöhen.

In den vergangenen Jahren stand die Service oriented architecture im Mittelpunkt und die Forschung wurde vor allem mit dem Ziel betrieben, Methoden zu entwickeln, welche die Implementierung von Geschäftsprozessen evaluieren und optimieren. Diese Ansätze nehmen eine Managementperspektive bei der Umsetzung von Dienstleistungen ein und betonen finanzielle und ökonomische Faktoren wie Rentabilität, die Kosten der Leistungserstellung oder die Absicherung der Zuverlässigkeit der angebotenen Dienste. Die bereits bestehenden Systeme empfehlen bewährte Verfahren (Best practice) zur Optimierung der Geschäftsprozesse, die deren vollständige Abläufe berücksichtigen und dabei auch verbundene weiche Faktoren wie das Verhalten der Humanressourcen einbeziehen, um so den Erfolg oder Misserfolg der verschiedenen Prozesse zu bewerten. Dennoch bleibt die Berechnung der Geschäftsprozesskosten ein komplizierter und umständlicher Vorgang. Die Abschätzung der Rentabilität einer Prozessinnovation vor ihrer Implementierung stellt eine schwierige Aufgabe dar und Techniken, die dies so früh wie möglich leisten können, spielen eine wichtige Rolle bei der Entscheidungsfindung. Dies erfordert Ansätze zur Gestaltung von Geschäftsprozessen und Praktiken, die Methoden für die quantifizierbare Bewertung von Aufwendungen und Erträgen für die verschiedenen denkbaren Umsetzungen der Prozesse beinhalten.

In dieser Arbeit wird ein neuer Ansatz vorgeschlagen, welcher die Kosten eines Geschäftsprozesses auf der Grundlage der Kosten und Zuverlässigkeit jeder einzelnen Handlung oder Aufgabe des Prozesses berechnet. Diese Methode unterteilt den Geschäftsprozess, dargestellt durch ein Business process model and notation (BPMN), in sich wiederholende Muster und berechnet einen Kosten- und Zuverlässigkeitsfaktor für jedes dieser Muster. Als Ergebnis werden die gesamten Kosten, die Zuverlässigkeit und die durch eine erfolgreiche Umsetzung des Geschäftsprozesses verursachten Kosten, die Business cost, ermittelt. Die Arbeit nutzt diesen Ansatz und schlägt eine Erweiterung von finanziellen und wirtschaftlichen Parametern als theoretische Grundlage für die Darstellung von Serviceaufrufen und die Ausführung von langfristig laufenden Transaktionen vor. Als Basis für diese Erweiterung wird Sagas benutzt. Langfristig laufende Transaktionen sind Sagas, wenn sie unterteilt werden können in eine Abfolge von Transaktionen, die mit anderen Transaktionen verschachtelt sind. Mit der Umsetzung dieses Ansatzes kann man individuelle Geschäftsprozesse mit der Hilfe einer Sensitivitätsanalyse bewerten und Problembereiche eingrenzen, die optimiert werden können. An-

hand dieser Methodologie analysiert die Arbeit verschiedene existierende Systeme und bekannte bewährte Verfahren in Bezug auf ihre finanziellen Auswirkungen auf Geschäftsprozesse. Die Arbeit wird von einer Java-basierten Applikation unterstützt, welche diesen Evaluierungsprozess automatisiert.

Acknowledgement

I thank my supervisor Prof. Dr. Martin Wirsing for his support and guidance without which this thesis would not have been possible. Under his supervision I have developed and evolved professionally. The friendliness and warmth with which Prof. Dr. Martin Wirsing supervised me is unforgettable. Even with his very busy and completely packed schedule he has always had time for my discussions and has been easily approachable. I treasure this time and work with him. I also thank Prof. Dr. Hubert Baumeister for being the second referee. I thank him for his comments and evaluation of my thesis.

I also thank the colleagues at the PST (Programming and Software Technique) chair for the constructive discussions. They have always been very welcoming and have helped me with this work. Some of my discussions with Mirco Tribastone, Moritz Hammer and others have had a deep impact on my work. I take this opportunity to thank them.

This thesis would not have been possible without the support of my parents, my sister Padmashree, her husband Dirk, and my wife Arundhathi. I thank them for their patience and understanding during this phase.

Contents

1

Introduction

Profitability is almost always (if not always) the primary goal of any business enterprise. The success of a business is dependent on high income and low and controlled expenses. Processes are implemented so as to either directly contribute or support this goal of the organization. And every organizations interest is to make these processes successful.

As information technology becomes industry oriented the factors that make a process successful have taken center stage. This is especially visible in the service industry. The aim of achieving higher quality and at the same time keeping the costs controlled or reduced are of high importance to the business. Due to these reasons the returns from and optimization of processes assumes utmost importance to an organization deciding to implement a business process. A calculation such as this decides if to have the process at all or not. Fields of research which have come forefront provides for structured approaches to designing and implementation of business processes. Business process management (BPM) or business service management (BSM) belongs to these fields to name a few. These establish a management perspective to (re)organize business process management within an enterprise. These are implemented top-down i.e. at the level of the organization and broken down to level of the processes.

This is not the same when it comes to financial management at an operative level. Business processes at an operative level are defined with different perspectives and objectives, be it internal or external, customer or industry oriented, product or process oriented. Even though the intention and pressure to reduce and control costs at the operational level is of great significance, methodologies and frameworks which have foundational reasoning to achieve this do not exist. Such an approach at the opera-

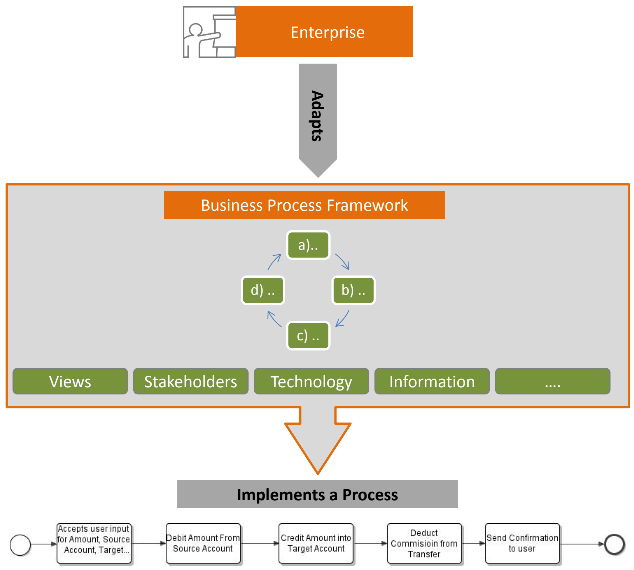


Figure 1.1: Enterprise adapting a business process framework

tional level is not elaborated enough and has not achieved a maturity that the industry can use.

The aim when undertaking financial optimization is to find out how we could implement a process which gives the maximum returns. The concern is how to calculate and reason out the cost of a process (either implemented or still in design).

1.1 Financial evaluation and optimization

Business process management within an enterprise starts with adapting a business process framework. Some of the well-known frameworks include Cimosa[78] or Work centered analysis framework [76]. The frameworks divide the organization into views such as technology, stakeholders etc. These are taken as basis to define the workflows and in turn the processes which are implemented at the operational level.

Financial evaluation and optimization is the process to find out how we could implement a service such that it gives maximum returns. This process aims to calculate and optimize the performance of the financial parameters of a process. The concern here is not if there should be a process implementation in the first place. In other words the reason or the objective of the business to (re)design a process already exists.

The concern of an organization from a business perspective is to make sure that every process implemented leads to a certain financial profitability. This profitability is either a revenue generation or an optimization of an existing expenditure. Immaterial of how a business process is defined or why it is defined, it comes with two distinct characteristics; one is cost and the other is reliability. Every business process when executed incurs a certain amount of cost and performs at certain reliability. The aim of the business is to keep the costs at the minimum and the reliability at the maximum. This is with the aim to make positive returns from a business process.

Below are two examples to elaborate this topic:

1.1.1 Case study 1 - A money transfer process:

Consider a global bank which has presence in most parts of the world, including the European union (EU) and the Asian continent. Due to the difference in the banking system (for example the local currency or local policies of the respective government) between the EU and the countries in Asia, the banking software varies between these countries and with the setup in EU. Nevertheless the core banking processes to serve the customers of the bank are all in place and functioning according to expectations.

The bank wants to provide a new service to increase its profitability. The customers can now transfer money between any two countries. The bank wants to achieve two objectives here:

1. **Service objective:** make the service reliable, robust and easy to use.
2. **Business objective:** to charge a commission for the service. The commission is a fixed rate and has to be above the cost of executing this process so as to be profitable.

The high level process is as shown in Fig. 1.2.

So as to step into the implementation of this process, a formulation on an activity basis for the service is as shown in Fig. 1.3. The process accepts the inputs for the

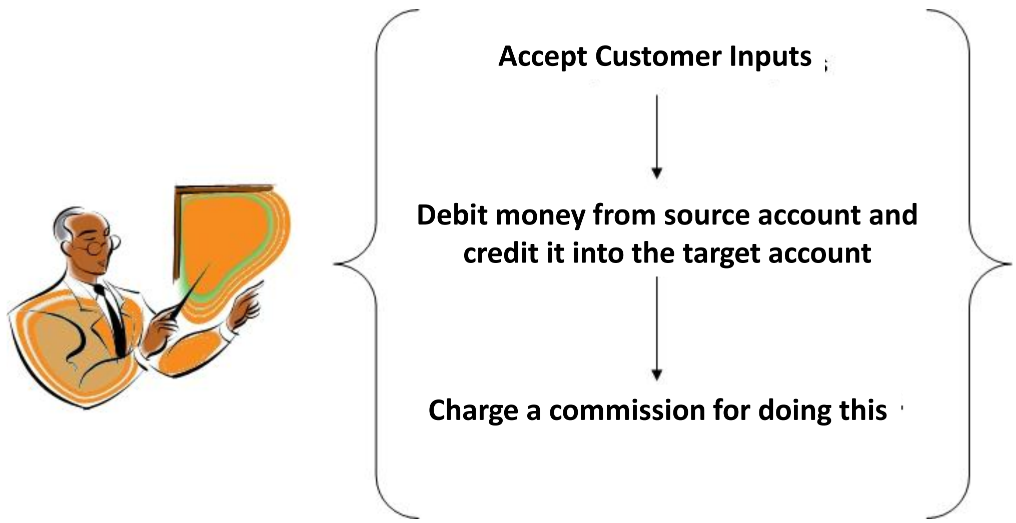


Figure 1.2: A management view of a money transfer process; simplified in three steps

accounts from the user, deducts from the source account and credits the amount to the target account. The user is informed through a confirmation. This is a process with dependencies; in this case the banking system which is in place and has to interact with each other etc. The process diagram covers these aspects as well.

As shown in the Fig. 1.3, there are two paths in this process. The successful path is when the money is successfully transferred. The failure path is when the transfer of money fails. In all cases the process works completely without any abrupt exits. In other words, from an implementation perspective, the process 'Works'.

Every time the process is executed the bank incurs a certain cost. The bank wants to charge a commission on every transaction so as to make this a profitable service. Hence the bank needs to know this cost and in turn calculate the commission. The commission can be charged only when the money is really transferred i.e. the successful case. The bank loses money in case the process flows through the other path i.e. the failure case.

The bank wants to answer the following questions:

1. over a fixed period of time, assuming that the process is executed "n" number

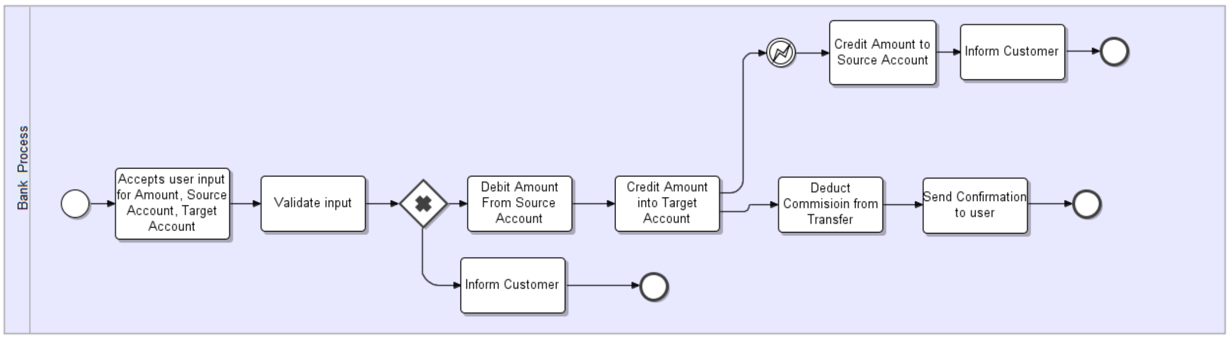


Figure 1.3: A detailed operative representation of the money transfer process with error handling

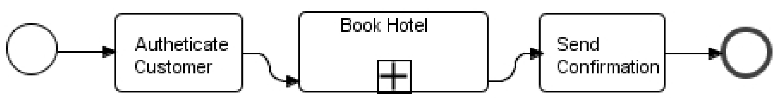


Figure 1.4: A business process diagram to book a hotel

of times, has a known reliability with which it will follow the successful path, what would be the cost that the bank is paying so that service is offered to the customers?

2. reliability of which of the tasks in the process has the maximum impact in controlling the costs?

1.1.2 Case study 2 - Hotel booking process:

We consider a hotel booking agency which books a room in a hotel according to the customer's request. This is an example which will consider again in other chapters as well. The process diagram in Fig. 1.4 shows the business process of a hotel booking agency. The customer is first authenticated, a hotel is booked for the dates entered, and then a confirmation is sent back to the customer.

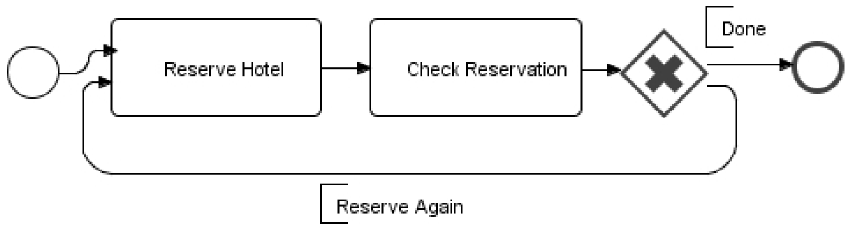


Figure 1.5: Detailed business process diagram for booking a hotel with a decision condition to check for reservation in many hotels

The process diagram in Fig. 1.5 is a blow up of the "Book Hotel" task from Fig. 1.4. We see that the task tries to book a room in "n" hotels until it has successfully booked one of the hotels or there are no free rooms for the entered dates in all the n hotels. If we assume that there are 10 hotels and each time we try to book we pay 2 units of cost, the minimum cost is 2 and the maximum cost is 20 making the average 11. It could be that we incur the maximum cost i.e. 20, and still the task of booking a hotel is not achieved.

The agency wants to answer the following questions:

1. How many hotels can the agency call such that it is still profitable?
2. Does its profitability vary on the order of the hotels it selects?
3. Is it financially worth calling a hotel or not, taking the probability of availability of a room in that hotel into consideration?

1.2 Aim

The aim of this research is to propose a methodology by which the cost of a business process is calculated by considering the cost and reliability of each action or task in the process. This methodology breaks the business process, represented using business process model and notation, into repetitive patterns and a cost and reliability factor for each of these patterns is calculated. The patterns are interleaved with each other through relationships, for example through a gateway. The patterns are then put together according to these interleaving relationships and the cost and reliability of the combination is calculated. Continuing this approach the overall cost, reliability and the cost incurred to achieve one successful execution of the business process; the business cost of the process is achieved. This approach is used as the basis to propose an extension to theoretical foundations representing process invocation and execution such that it can incorporate the financial and economic parameters. The methodology is programmed as an application based on Eclipse for business process extensions using Java programming language. The application takes a process diagram as input to generate the cost, reliability and business cost parameters for the process. By implementing the concept individual business processes are evaluated with the help of sensitivity analysis so as to find problem areas where an optimization can be achieved. The methodology is used on different case studies as examples.

The research introduces the different business process frameworks, the performance evaluation methodologies and the recommended best practices for business process design which are available in literature. The recommended best practices are evaluated through the proposed methodology so as to show as to where and when a recommended practice is best usable. The impact on the performance of a process due to the implementation of the best practice is evaluated through the proposed methodology.

1.2.1 Structure of this document

In the next chapters, this document captures the importance and need of the business value of a process, the methodology proposed to calculate the costs and the results of implementing this methodology.

The document starts with presenting the well know business process engineering techniques, performance evaluation techniques and the business process model and notation. Business process design and engineering / reengineering is a topic of research form the early 1980's. The literature is vast on this topic and there are number of frameworks which have been proposed for reengineering. The chapter **Business process frameworks** 2 introduces some of the established and well known frameworks available in literature.

The aim of this research is to define an approach for the calculation of costs for a business process implementation at an operative level. The research classifies cost and reliability as performance factors of a process. The chapter **Cost, reliability and**

business cost: Performance management and modeling 3 introduces the parameters cost, reliability and business cost and the different interpretations available. The chapter introduces the performance modeling and management techniques available for cost and reliability. Along with these the frameworks and techniques for evaluation of performance business processes are discussed. The chapter introduces the evaluation technique called 'Devils Quadrangle' which is later used for the evaluation of recommended best practices.

Business process model and notation (BPMN) is a graphical notation to bring the business and technical requirements at an operative level together. The chapter **Business process model and notation** 4 introduces the notation and the artifacts.

The chapter **Cost calculation with patterns** 5 proposes the methodology for calculation of cost, reliability and business cost of the process. It takes business process diagrams as input for calculation and divides them into patterns. It covers the different patterns that are commonly seen in the business processes and achieves a formula based calculation mechanism for the same. The chapter also introduces the sensitivity analysis and how it can be used to find areas of optimization within a process.

An important step when bringing together the business process together with well-known theoretical foundations is to take care of the transactional properties in the execution. The process definitions should include mechanisms for compensation and boundaries so as to limit the scope of a process flow. We take the semantics defined for flow composition languages and extend the same to show development of cost, reliability and business cost when using patterns in a business process. This is covered in the chapter **Theoretical basis for cost calculation** 6.

The proposed methodology is programmed as an application 'Business Process Diagram Cost Analyser' on the eclipse platform. The application takes a business process diagram as input to calculate the parameters and displays the different patterns generated. The Chapter **Application: 'Business Process Diagram Cost Analyser'** 7 covers the details of the application.

The methodology proposed in the chapter **Cost Calculation with Patterns** 5 is implemented on two examples in the chapter **Examples: Business cost calculations from a business process diagram** 8. The first example is a simple example which deals with payment processing which we come across commonly in many instances. The second example is a complicated example from a well-established automobile manufacturing enterprise in Germany. The example covers the order management process at the dealer of the enterprise.

The literature on business processes recommends a host of best practices for business process design. Nevertheless there is no real guidelines as to which best practice needs to be implemented when and how. In the chapter **Best practice evaluation** 9, we evaluate the most commonly recommended best practices in literature for their effectiveness. We implement the best practices on examples and calculate the costs before and after implementing the recommendation.

The document is appended by a list of references, list of figures and list of tables.

1.3 Related Work

At the core of this research are business processes at the operative level, their financial performance evaluation and calculation, and their positioning in the overall concept of business process frameworks and Performance monitoring tools. The research brings together different areas such as business process modeling within an enterprise, monitoring cost and reliability as performance factors within a process , notation for business process model, patterns within workflows etc. and evaluates them with the aim of calculating the cost of an operative process. This research bases itself upon numerous contributions and development in these fields. Some of these are recent and have been brought forward with the prominence of business process management whereas some are older but have gone through many revisions (or generations) over the years.

The aim of calculating costs from a business process diagram came forefront after business process model and notation (BPMN) specification was released. The literature in this field for the different approaches for the calculation of costs can be categorized into groups. Research has been done and proposals have been made to calculate the costs directly from a process diagram. The flexibility offered by the notation makes it difficult to calculate costs directly from the process diagrams. Magnani and Montesi [3]. have said 'BPMN diagrams can be very complex' and 'the free-form nature of BPMN can create modeling situations that cannot be executed or will behave in a manner that is not expected'. The concept of cost calculation using a BPMN diagram was proposed by Magnani and Montesi [3]. The methodology proposes an extension to each BPMN artifact with a textual parameter for cost. The parameter ranges between a minimum and maximum value to accommodate looping and conditions. The proposed calculation methodology will then calculate the costs of the whole BPMN diagram by putting the relationships together. The costs thus calculated ranges again within a minimum and a maximum value. Another contribution also from Magnani is [4]. In their proposal they have identified classes of diagrams capturing the most typical processes and for which efficient analytical solutions exist. The proposal addresses processes with single token, multiple token and nested processes. The cost calculation is still based on the minimum and maximum range concept.

Another approach for cost calculation is from a service perspective i.e. how much does a service cost. Approach[46] which concentrates on services and their costs have shown that the accurate costs of a service are impossible to calculate because the service doesn't know with which other service it will have to interact. Each of the services with which an interaction took place will have an impact on the cost and behavior of the original service. As the interactions are not predefined, there can be many possibilities for the flow of process. This approach projects the costs within a range of values.

Other contributions in this area also use the methodology of extending business process Model and Notation for capturing and evaluation of parameters as required. Saeedi et al. [5] extends the business process notation to capture parameters for quality. The quality parameters are defined as time, cost and reliability. Another approach is to categorize the information needed from a process into functional and non-functional properties and extend the modeling notation to represent these. Bocciarelli et al. [6] propose an extension to the BPMN which is based on model driven architecture and allows for specification of performance and reliability properties. Apart from these there are also other proposals which rely on simulation techniques to calculate the cost of processes.

Relating or backing these approaches with theoretical foundations which model the interaction between processes plays an important role in establishing the approach. Long running transactions block resources for a long time until they complete. SAGA [8] is a workflow model which breaks them into transactions which can be interleaved. In case a transaction fails, the rest of the successful transactions are compensated for. A transactional calculi which takes this concept and addresses the boundaries and compensation mechanism for programming languages is defined by Bruni et al. [9]. The behavior of long running transaction is modeled in this approach with respect to the result of each of the broken-down transactions.

An enterprise or an organization is made up many processes/workflows. The preferred way to model these processes is together with the enterprise it belongs to. Over the years many frameworks or approaches to model business processes as part of an enterprise have been proposed. The concern each of these approaches or frameworks addresses is which aspects of the enterprise the modeling of a process should consider.

A term used for any system which brings machines and/or humans who work with together with information for a product or process is a Work System. Steven Alter [30] defined a method to understand such work systems at the level required. Alters approach to a work system is based on the logic that it is a combination of business process, the infrastructure, humans and the service or product that the customer is being offered. His approach is presented as a framework called the Work centered analysis framework.

A model proposed by Jablonski is called MOBILE: a modular workflow model and architecture [35], this provides for separating the multiple aspects of a workflow. It is a workflow model which separates a workflow into the functional aspect (what to execute); the behavioral aspect (when to execute);the organizational aspect (agents to perform) and the informational aspect (data flow between workflows).

Another approach to modeling an enterprise is based on CIMOSA [78] and is proposed by Berio et al. [31]. This is a business process driven modeling approach and features a 'clear separation between the concept of processes (modeled as workflows) and the concept of agents or resources (modeled by state machines). The link between the two concepts is materialized by functional operations (elementary actions) as well as resource capabilities and competencies'.

A recent approach to modeling business processes is called the Extended framework for business process reengineering [73] . This has been developed by H.A. Reijers et al. and brings together the components of all the other proposed frameworks. This framework is made up of six elements and connects the customers, products with the information, technology and environment with the help of business processes.

Performance management came into a lot of attention as the interest in business process management took center stage. The question that needed to be answered was how the performance of processes and in turn the that of the enterprise be monitored, measured and optimized. A simple and well known tool was proposed by Kaplan et al. and was called the Balanced scorecard [79]. The aim of the Balanced scorecard is to define measures which are both financial and non-financial and assigning targets to the same. Once done the aim is to measure against these targets and hence be in a position to evaluate progress or the lack of it.

Another performance management tool is the Performance Prism [54] and is proposed by Adams et al. The aim of the Performance Prism is to bring the views and expectations of the stakeholders as the central component of an organization. Based on these demands the strategy of the enterprise is defined. The performance management tool concentrates on measuring the strategy in such a way that the stakeholder's expectations are met.

Each of the performance management tools discussed till now were either driven by enterprise level objectives such as stakeholders or were empirical as in the case of the Balanced scorecard. Brand.N et al.[13] concentrated more on performance monitoring at operational level of a workflow or a process. They proposed that each process has four major parameters which are Quality, cost, Time and Flexibility. The proposal was that the redesign of a business process which leads to a betterment of any of the parameters also has an impact on the rest. This impact did not have to be a positive development. A reduction in cost could also mean a reduction in Quality. This tool was called the Devils quadrangle [13].

One of the basis for business process optimization are the set of best practices which have evolved over the years. Almost all business process frameworks recommend a list of best practices when designing and implementing business processes. A comprehensive analysis of these has been done by H.A. Reijers et al. The analysis [73] compares the different frameworks and monitoring tools available in literature with the best practices that. It then categorizes these according to the situations best suited for their implementation.

1.4 Conclusion

A study through the available literature shows that business processes have been of interest for researchers for a long time now. Frameworks for business process modeling have gone through many iterations/updates and are relevant even today. In addition to these many contributions in recent years have explored possibilities to standardize business process design principles and its impact on the organizations.

A considerable amount of research done in this field has a top down approach towards business process engineering. The objective of the underlying framework or principle is to optimize business processes such that they are efficient and allow for the smooth functioning of the organization. The aim is to achieve a high process maturity such that the process executes with no errors and all stakeholders are aware of their responsibilities and play their part. In other words the objective is to make sure that the business is running.

The work till now has put together a strong foundation for business process design and engineering. However fundamentals for analysis and design of processes at an operative level such that its profitability can be evaluated are yet to be defined. Defining and controlling objectives for a process such that it achieves a business value which can be measured by financial parameters has not yet been investigated i.e. the question "Has the business process achieved its business objective?" remains unanswered. Techniques and methodologies by which the cost and profitability of an operative process can be evaluated needs to be developed. Ways and means by which the quality of a process can be interpreted as the probability that the process will achieve a business objective needs to be defined. This in turn will impact the cost or profitability of the process.

2

Business process frameworks

Business process engineering and reengineering of a process varies on the primary aspect if this is the first time a process needs to be designed, developed and implemented or if there already exists a process that needs to modified, optimized etc. Literature shows that business process design and engineering / reengineering are a topic of research form the early 1980's. First available literature is from IBM [52] and CIMOSA [78]. Some researchers claim that the idea of design principles and engineering / reengineering started when it was formulated by after Michael Hammer [53] who first raised the visibility of business processes under the topic of business process reengineering. The literature is vast on this topic and there are number of frameworks which have been proposed for reengineering.

Business process Reengineering is dealt almost always as an organizational initiative and is handled through an organizational vision and motivation. Frameworks in business process reengineering are not business process development models. They have been developed to handle the different influencing factors which come into play either independently or in relation with others. These include softer aspects such as people management etc. The frameworks come with a recommendation on the different perspectives and views that need to be considered when a business process needs to be reengineered.

Over the years these frameworks and best practices have been analyzed and developed further. The best practices are formulated as heuristic rules. These lead to evaluation of business process design, thus allowing for redefining and formulation and analysis of alternatives.

In this chapter we introduce the different frameworks which are available in literature for business process design. As the literature is vast we cover the mainly well-established frameworks. We cover the best practices (Resequencing of tasks [70, 25], Knock out order [23], Task elimination, Order type and triage [23], Parallelism) in the chapter **Best Practice Evaluation** 9. In the same chapter we also evaluate these best practices for their impact on implementation in different scenarios.

2.1 CIMOSA

'Computer integrated manufacturing open system architecture' or CIMOSA [78] is an enterprise modeling framework which was developed as part of the European strategic program for research and development in Information technology (ESPRIT) by a group of major European vendors. The CIMOSA model is represented as a three dimensional cube representing the view, lifecycle and generic dimensions. This is shown in the Fig. 2.1.

This framework offers the guidelines and constructs so as to put business requirements together. This can be then translated into CIM system design and implementation. CIMOSA defines three modeling levels; for the definition of the business requirements, for the evaluation of a solution, and for the implementation of the solution. Apart from the modeling levels, CIMOSA classifies all functions in an organization as generic, partial and particular. The generic classification is an existing catalog of CIMOSA architectural constructs or building blocks in CIMOSA, hence making modeling much easier. Partial level contains models which can be implemented to only a subset of the manufacturing industries and the particular level has models which are very particular only to one manufacturing enterprise.

As shown in the figure, CIMOSA takes three dimensions into consideration: life cycle dimensions, enterprise view dimensions, generic dimensions. CIMOSA classifies the functions within a manufacturing organization. The functions are classified as generic and specific. These are put together to form a model which is used for process simulation and analysis, especially in the manufacturing organizations which includes dispatching, scheduling etc. CIMOSA separates these functions using two interrelated concepts.

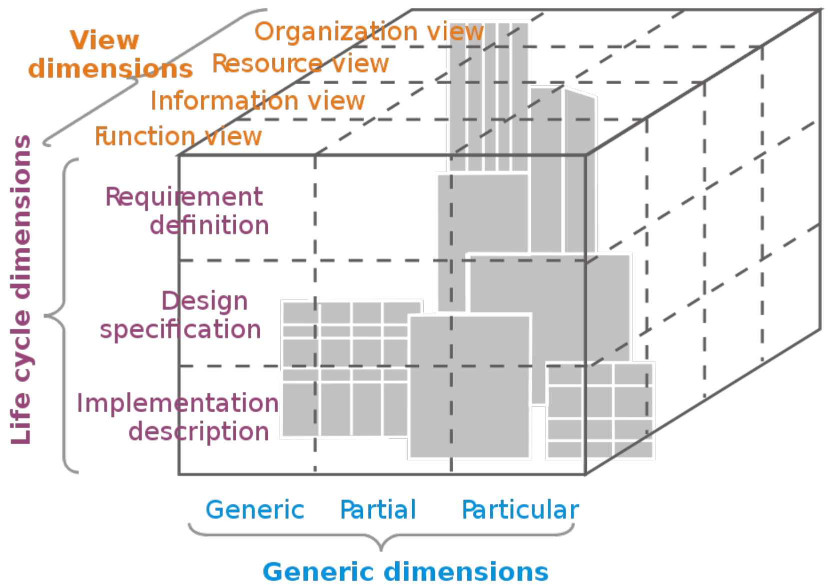


Figure 2.1: The CIMOSA cube from [78] with the view, lifecycle and generic dimensions

- **CIMOSA modeling framework:** Using this framework the specific and generic functions are clearly separated.
- **CIMOSA integrating infrastructure:** This is the infrastructure which supports the execution of generic functions and linking of specific functions. It is effectively the communication system which interconnects all of the functions in the CIM system.

The views within the framework are as below:

Generic dimensions

This dimension deals with the operations within a manufacturing organization. The functions are classified as generic and specific. It defines the following two layers.

- **Generic layer:** Manufacturing organizations have functions which are common in all the organizations independent of what they are manufacturing. These functions fall into the generic functions. Some examples listed out are control of work flow, administration of information, integration of resources and management of communications.

- **Specific (partial and particular):** The functions which are not generic across organizations, are dealing with particular scenarios or specialties are dealt in this layer. These include functions in producing a product, processing of orders etc. These functions are performed by machines, humans and computers.

Life cycle dimensions

This dimension starts with capturing the requirements, evaluating a design and finally implementing the same. The dimension is broken down in the following layers:

- **Requirements definitions:** This layer deals with capturing the requirements at the level of the users. The requirements and the needs of the users are captured and documented in a clear language which can be understood by all.

- **Design specification:** The requirements from the users which have been documented in the requirements definition level are taken as the basis to evaluate solutions to take care of the problem. A design is generated which will take care of the problems.

- **Implementation specification:** In this layer is the logical next step to the design specification layer. A detailed solutions according to the design evaluated in the previous layer is put together here. It takes all the constraints into consideration.

Enterprise View dimensions

The functions in an organization come with many perspectives and stakeholders. The view dimensions cover these with four modeling views of the enterprise functions. These are as below:

- **Function view:** This view deals with the work flow as a function. This describes the work flow in the process.

- **Information view:** For the function defined in the function view, this view adds the inputs that go into the function and the outputs that this function delivers.

- **Resource view:** This view deals with all the resources which are needed to perform this function. This includes all internal resources, external resources, humans, machines, and control and information systems.

- **Organization view:** The people who are in authoritative and responsible position for this function are covered in this view.

Integrating infrastructure

The CIMOSA Integrating infrastructure is made up of services which are as below:

- **Business services:** This service controls the work flow which has been defined by the function model. This service interprets the function view.

- **Information services:** The information view defines the information input and output. The information services has the generic functions for handling the information which has been defined in the information view.

- **Dialogue services:** The functions defined in the function model and the resources defined in the resources view are linked together through the services offered in the Dialogue services.

- **System management services:** The functions in these services are provided to be used by people who are in responsible positions as defined in the organization view. Some examples are: to change, release, activate, start, and stop etc. models, both off-line and on-line.

- **Common services:** All services which are common for the Integrated Infrastructure services, such as the ones used in communication handling are defined a part of the Common Services.

2.2 Work centered analysis framework

The Work centered analysis framework (WCA) [76], developed by Steven Alter, organizes a work system such that the business process is a central component with links to all the elements involved with it. The framework represents work as six linked elements. The element customer is placed as the peak of the pyramid; this is followed by products or the services which are delivered to the customer. The base of the pyramid is the element for business processes which in turn has information, participants and technology as elements below it. WCA has a top down approach towards work flow systems but allows for classification of complex work flow systems. It allows for linking the business process with the environment with which it works.

The figure shows the links between these elements.

Customers

The customers, internal or external, are the entities who are receiving the service or product from the work flow system. People within the organization are classified as internal customers and the rest are classified as external customers.

Products

The final service or product which is delivered to the customer by the work system is classified under the element products. Products do not take into account the raw materials or other inputs needed by the work flow to achieve the output.

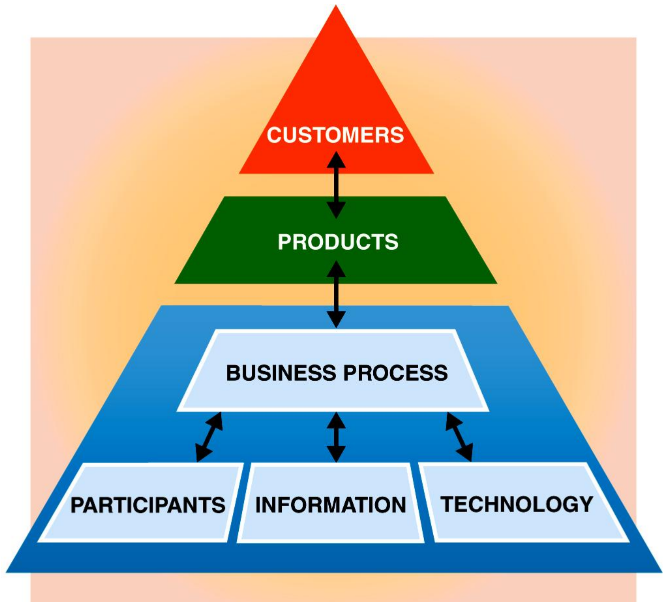


Figure 2.2: The WCA framework[76]

Business process

Business process is the set of tasks or activities that are executed in a pre-defined order so as to achieve the end product in such a way that the goals of the organization are met.

Participants

The people who accomplish the tasks as specified in the business process are classified as participants.

Information

Business processes, when executed, use and generate data which are needed for the successful accomplishment of the tasks. These data are classified under the element information. The final product is not considered here.

Technology

All kind of hardware, software, devices etc. which are not human and are needed to execute the business process to completion is classified under technology.

2.3 Mobile model

MOBILE [32] is a workflow management model and stands for a modular workflow model and architecture. Workflows within the mobile model are represented in four different perspectives depending on certain aspects of the workflows.

These perspectives are as below:

Functional perspective:

This perspective divides the process such that it can be represented as a repeatable set of workflows. In other words the workflows are part of different processes achieving functional objectives of the organization. The workflows are classified under:

- **Prescriptive workflows:** Perspective workflows are well defined workflows such that sub-workflows and application instances which are part of this workflow are declared.
- **Descriptive workflows:** Workflows which define activities to be done but allow for ambiguity as to when and with what refinement are classified as descriptive workflows.

Behavioral perspective:

This perspective considers the behavioral aspects of workflows. The control flows within work flows defines the flow of execution of the workflow and its collaboration or interaction to other workflows in the system.

Organizational perspective:

This perspective deals with the organizational issues or questions for a work flow. This perspective addresses the issue as to 'Who' is responsible or executes a work flow. Organizational concepts which are covered in this perspective are:

- **Organization:** This forms the base of the concept and it is independent of the workflow system. Organization aspect describes all the entities within the organization.

- **Notification and synchronization:** The notifications [32] describes what to do, why to do it, and how to do it, in order to make the execution context clear to the resource and is made up of lists defining what needs to be done. All the agents in the organization have one or more notification lists and the agent is responsible for performing the task which could be to execute an application or to authorize the same.

- **Organizational policies:** Policies are the links between the agents, notifications and synchronizations. The organization policy defines which agent is responsible for executing which task / application and in which part of the workflow.

Informational perspective:

Workflows use data as input and they produce data as output. These are covered under the Informational perspective. The data is classified as control and production data. Control data is data exchanged between workflows and contains control information such as against what data set should the workflow execute etc. Production data on the other hand comprise all data that are essential for an application area.

Architecture of MOBILE

The architecture[32] [50] implementation distinguishes between a build-time and a run-time architecture.

The build-time architecture consists of a work area repository. The work area contains the definitions of workflows, the organization and the data and functions in the system.

The repository is a huge class library which stores all workflow relevant information. This assists as well when new workflows are being defined as the new workflows can be put together by existing workflows from the repository.

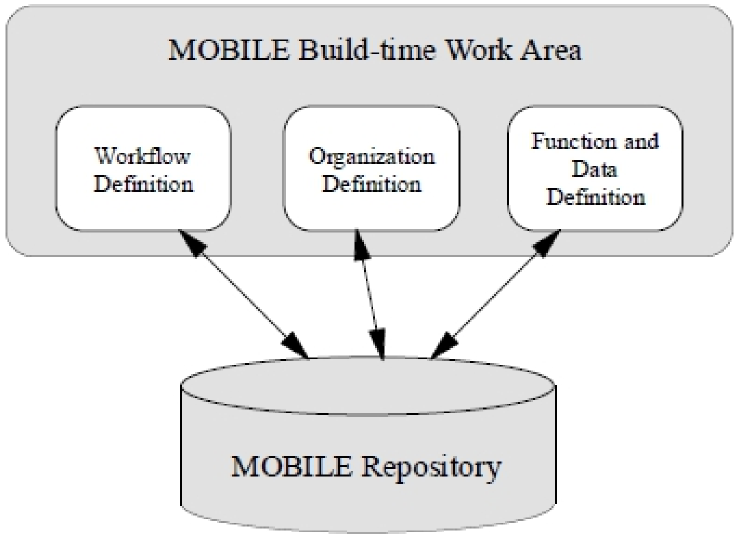


Figure 2.3: MOBILE build time architecture from [32]

The run-time architecture consists of two major blocks, the kernel and the shell. These two together define the execution engine. The kernel interprets the specification of the workflows, puts them together in the right sequence and executes them. The workflows are in the repository. The shell is multiple servers and it surrounds the Kernel.

2.4 Extended framework for business process reengineering

The extended framework for business process reengineering is a rather new framework when in comparison with the other frameworks available in the literature. It has been proposed by Reijers and Mansar [73]. The framework is put together as a combination of WCA framework [76], the mobile workflow model[32] and the CIMOSA enterprise modeling views [78]. The concept is based on taking in techniques and best practices from other practiced frameworks to develop a methodology for business process reengineering implementation. The framework is as shown in the fig. 2.5.

The framework has six elements which are linked to each other as below:

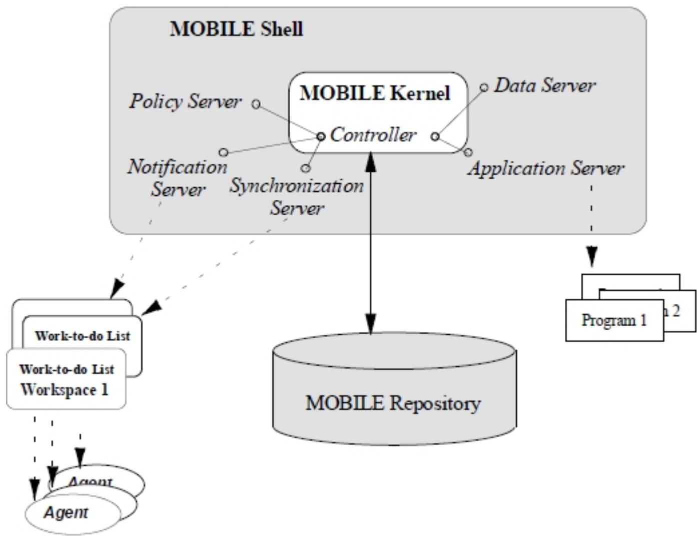


Figure 2.4: MOBILE run time architecture from [32]

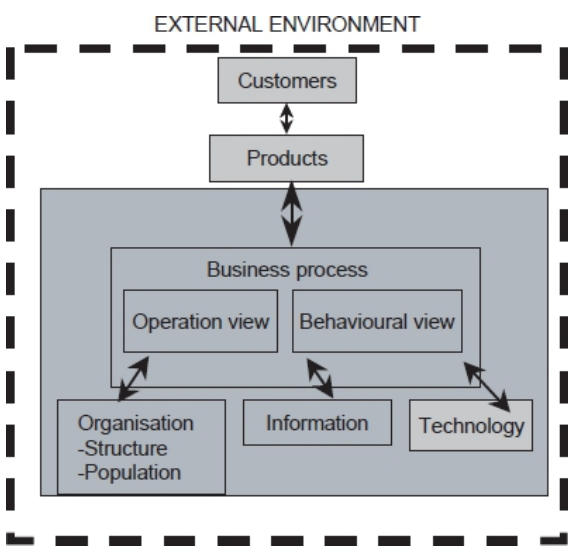


Figure 2.5: Extended framework for business process reengineering as proposed by Reijers and Mansar [73]

- **Customers:** This element covers all the internal and external customers of the business process.

- **Products:** This element covers all the products and services which are generated by the business process.

- **Business process:** This element covers the core of the framework, the business process. It is organized into two views under this element.
  - Operational view: This view takes care of the operational level of the workflow. All operational aspects such as tasks in a job, kind of job, size etc. are covered in this topic.
  - Behavioral view: This view covers all the behavioral aspects of a workflow such as sequence of tasks, scheduling etc.

- **Organization:** This covers the organizational aspect involved in the workflow. These include
  - Organization structure: includes structural inputs such as roles, users, groups etc.
  - Organization population: covers the individuals (agents) to who tasks are assigned and are responsible for execution and the relationships between them.

- **Information:** The input and output of information in the workflows are covered in this element.

- **Technology:** This element covers all the technology that the business process uses.

- **External Environment:** Everything outside the internal environment falls under this element.

2.5 Conclusion

In this chapter we looked into four major and established frameworks that are available in literature. CIMOSA, as one of the very first frameworks available looked into a manufacturing enterprise with different dimensions covering all the perspectives of the process within the organization. The models MOBILE and Work centered analysis framework both are workflow centric and map the same to the rest of the organization. The last framework 'Extended Framework for business process Reengineering' is a combination of the other frameworks and is driven by the techniques, methodologies and best practices recommended by those frameworks.

Even though all these frameworks are very well detailed and can be used as basis for the design and engineering of business processes, none of these take into consideration the cost parameter of business processes (only the CIMOSA framework was planned to add a cost module). Also the best practices from these frameworks, which we will discuss in the chapter **Best Practice Evaluation** 9, recommend implementing certain design principles which could possibly lead to a cost optimization. Nevertheless the frameworks themselves do not specifically mention in which situations these practices should be implemented and how much the total savings from such an implementation would be.

3

Cost, reliability and business cost: performance management and modeling

The business directory [56] defines performance as:

> *The accomplishment of a given task measured against preset known standards of accuracy, completeness, cost, and speed. In a contract, performance is deemed to be the fulfillment of an obligation, in a manner that releases the performer from all liabilities under the contract.*

Performance management is the process of setting goals or objectives, measuring against them and taking actions when they are not met. It is a process where the current situation is continuously evaluated against the set objectives with the aim of achieving them in a cyclic manner. Performance management can be done for a vast range of entities such as an organization, a process or software or even an individual.

Every organization defines performance parameters and these are measured and managed with the help of standardized frameworks. The parameters that come into consideration are many and depend on the organizational goals i.e. for some organizations it is the quality of a service whereas for some it is that the cost is at a minimum. It is necessary that the processes implemented in these organizations improve these factors substantially. Cost and reliability are two prominent performance parameters of a process. Every business process when executed incurs a certain amount of cost and

performs at certain reliability. The best case scenario is to have a high reliability and a low cost. performance management and measurement revolves around three major activities, namely: setting goals that need to be achieved (this is done as part of performance modeling), collecting information and data which show the current status of the organization with respect to the goals specified, and finally take actions which will make achieving these goals possible.
The first phase of performance management is the modeling for performance. The modeling phase evaluates a current system with preset conditions against expected target values or benchmarks for specific performance parameters. From a business perspective, performance parameters could be financial parameters such as cost or revenue or parameters such as productivity of the employees or customer satisfaction etc. From an IT perspective these could be parameters such as CPU utilization, turnaround time or number of bugs in the software etc. In modeling, scenarios are evaluated to check against these benchmarks.

This chapter

1. introduces cost, reliability and business cost as parameters of performance.
2. describes the different classifications and modeling techniques available in literature.
3. details the different evaluation techniques which deal with identifying and evaluation of performance measures in an enterprise.
4. introduces the devils quadrangle, a measurement framework dealing with performance at the operational level of a business processes.

3.1 Cost

Every business process when executed costs a certain amount of money. In fact cost is the most prominent factors to define the benefit of a business process. Cost is everything that is spent. We define cost as everything that is spent and is measurable in executing the process. We define costs at the level of a task in a business process; in other words every atomic activity or an event in a business process has a cost. From our perspective cost is a quantifiable factor, a measurable quantity. It is immaterial how many individual components make up the cost of an activity or in which classification they fall. They could be the costs paid to use an infrastructure, time invested by an employee in executing a task etc. The definition of cost is different in every organization, which might be again redefined for a particular business process.

A search on the definition of cost throws up many interpretations; some of them are as below:

- Amount of money, time, etc. that is required or used; A negative consequence or loss that occurs or is required to occur; to incur a charge, a price [57].
- The total spent for goods or services including money and time and labor.[58]
- Price: value measured by what must be given or done or undergone to obtain something.[59]
- In business, retail, and accounting, a cost is the value of money that has been used up to produce something, and hence is not available for use anymore. [60]
- In business, the cost may be one of acquisition, in which case the amount of money expended to acquire it is counted as cost. [60]

Depending upon the different factors which are taken into consideration as the definition for cost, different way of cost categorization exists.

Fixed and Variable costs

- **Fixed costs:** The costs which will be incurred immaterial of the situation are categorized as fixed costs. These are incurred on a fixed basis. These costs do not change. For example these are costs such a labor costs which is the salary to be paid to the employees, or costs such as the money spent in establishing the infrastructure.

- **Variable costs:** Costs which will vary according to the situation are categorized as variable costs. These are variable in nature like for example the costs invested to achieve higher quality etc.

Direct and Indirect costs

- **Direct costs:** The costs that can be traced directly to a product which was manufactured or a service which has been performed. For example, the cost of manufacturing a product.

- **Indirect costs:** costs that cannot be traced back to a particular activity or task and is indirect to it falls under the category as an Indirect cost. These are for example all costs incurred for keeping the infrastructure running etc.

Short term and long term costs

- **Short term costs:** Costs that are incurred in the immediate or near future and are planned and foreseeable fall under the category of short term costs. These are usually not repetitive i.e. once incurred they do not come up again.

- **Long term costs:** Costs which do not come up immediately but will come up in the future are called Long term costs.

3.1.1 Cost models

Literature offers a host of cost models and estimation techniques that an organization can use to calculate costs. Different researchers [55] [11] [12] have analyzed these models and also recommend categorizations. Evans et al. [89] have put together a comprehensive collection of the cost models available in literature. They are parametric cost estimates, neural networks, expert, function costing, feature costing, group estimation, case based estimating, knowledge based, generative and activity based. The cost models primarily vary in their approach to model and estimate costs. Some are based on logic of parameters (parametric) and attributes (neural) whereas there are models which depend on the existing knowledge (expert and knowledge). Models such as case or group check for existing cases or groups of entities being estimated and take them as basis for estimation.

Two of these cost models take a bottom approach for costing by attaching a cost to each task in the process; namely generative and activity based [62]. The difference between the two is that in generative the overhead costs are proportionally divided over the tasks whereas in activity based costing the overhead costs are done for each task. This makes it difficult to implement the activity based model. Nevertheless newer techniques allow for easier implementation through automation. Activity based costing is widely accepted and implemented.

3.2 Reliability

Every activity has a certain reliability at which it performs. reliability has different interpretations. reliability almost always is interpreted as a stability of the technology or the infrastructure. The probability that a server is running all the time is a question of technical reliability. This definitely contributes to the reliability of the activity. Nevertheless the reliability of an activity is much more than its technical robustness. It should take the business objective into consideration as well.

We interpret reliability in a broader sense. As with cost, reliability is a combination of all factors that make a certain activity successful. This includes the reliability of the underlying infrastructure. We would like to stretch this interpretation to include other factors that lead to an activity being successful.

We take the example of booking a room in a hotel as an activity as discussed in the case study. We want to consider reliability of this task from a business perspective. Reliability of this activity includes infrastructure reliability such as the telephone is working. The question that we want to answer is "what is the reliability that there is a room available in the hotel where we call for a room". This will decide if the business goal has been achieved, which is to book a hotel room. We see reliability as a combination of the technical reliability (which includes the infrastructure etc.) and the business reliability which is the reliability at which the business goal will be met or the service will lead to its successful completion. We define reliability as the rate at which the process will reach its business goal. This is the goal to which the management of the organization will relate to.

3.2.1 Reliability Models

Reliability modeling methods available in literature can be broadly categorized into two groups. A detailed analysis has been done by Slater et al. [61].

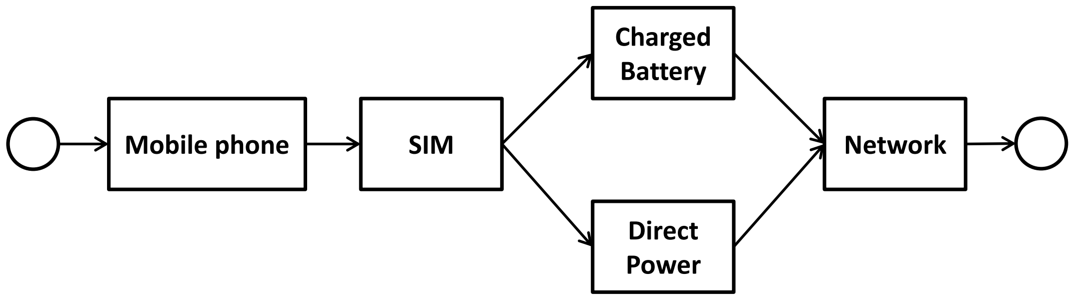


Figure 3.1: Reliability block diagram for a mobile telephone

Combinatorial model types

The combinatorial reliability models consider the different components in the systems and the different conditions when failure cases can be produced. Reliability block diagrams, event trees and fault trees are the most widely used combinatorial reliability models.

The reliability block diagram represents all components of a system as blocks which are connected to each other through directed connections. The blocks are either sequential or parallel to other components. Sequential components show that their failure will lead to a system failure. Components in parallel are components which act as back up for each other i.e. when one component fails; the other component plays as a back-up and hence does not lead to a complete failure. Fig. 3.1 is a reliability block diagram for a mobile telephone. It has components in serial order such as mobile phone or the SIM whose failure is immediately a failure of the mobile telephone. It also shows a parallel relationship between charged Battery and direct power showing that at least one of them will be needed for the system to work.

Event tree analysis works similar to the block diagram but depends on the events in the components. It is a binary representation such that an event either succeeded or failed at runtime; in turn the component succeeded or failed. As each event triggers the next (or fails) the event tree is used to find the path from the trigger to each of the final situation and the probability that this situation will be achieved. Figure Fig. 3.2 from [61] shows an event tree analysis for a brake system. The flow of events is boolean; marked with 'Yes' and 'No' and each possibility is attached with a probability. The

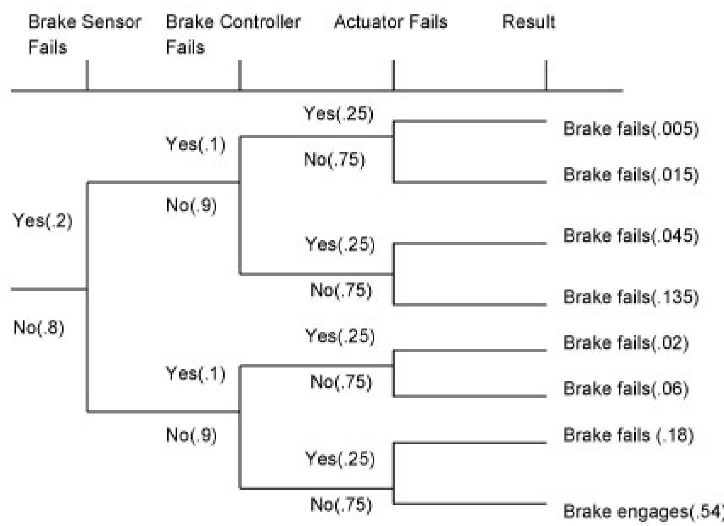


Figure 3.2: An event tree for a brake system from [61]

end result is either that the system is working i.e. brake engages or that the system fails which is brake fails. Again each result is shown with the probability that it happens.

A fault tree analysis is another reliability model which bases itself on particular events or happenings. In comparison to event trees, a fault tree allows for combination of particular conditions which allow for a failure situation. These events can happen at one level i.e. in parallel to each other or one after another which is then in a sequential order. At each level the events are put together through Boolean logic gates 'AND' and 'OR'. 'AND' gates put the events together whereas 'OR' gates show that even one of the events can produce the situation. Figure Fig. 3.3, from [61], is a fault tree representation of the brake system.

State space models

State space models come into use for situations where reliability evaluations cannot be done by binary combinations. This model allows for representation of complicated scenarios of interactions between components. Well know models such as Markov capture such interactions.

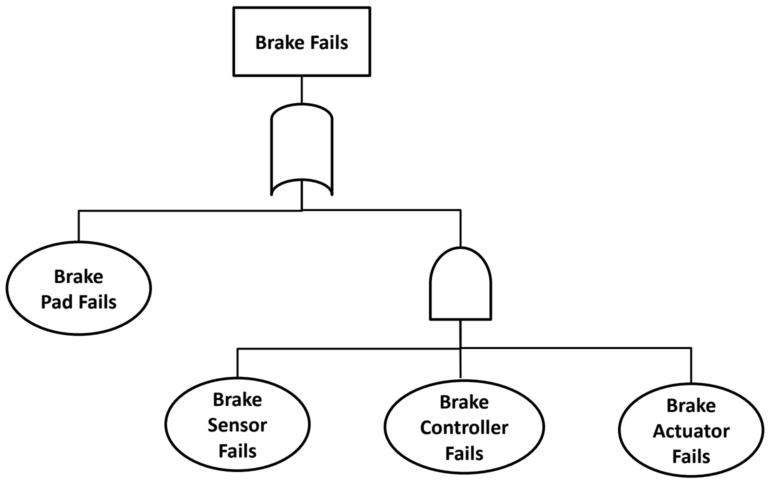


Figure 3.3: A fault tree for a brake system from [61]

A Markov model allows for state transitions probabilities showing at what rates a system can move from one state to another. These transitions can be grouped, as a transition group, which all takes place at a point in time. Using such a model the systems states can be grouped together as a state group; for example the failure state group. For such a group, statistical analysis can be done on spent time, rates of transition etc.

3.3 Business cost

The amount of money spent so as to execute a business process once is the cost of the process. Nevertheless it is very important to take the business perspective into consideration here. Every business process has been put in place so as to achieve a business value. This value could be a profit statement, for example: the business process should earn so much money, or it could be from a savings perspective, for example: the business process should cut costs such as etc.

We define the business cost as the cost incurred to execute the process such that it leads to a successful achievement of the business goal. business cost, like cost and reliability previously, comes into play both at the level of every single activity in a business process and at the level of the business process itself.

We defined reliability as a factor which includes the business perspective. Hence, the business cost of a process P whose cost is C and reliability is R is given by:

$$Business\ cost\ = C\ /\ R,\ where\ C\ >=\ 0\ and\ 0\ <\ R\ <=\ 1 \tag{3.1}$$

As the business cost is dependent on the reliability and the reliability is either equal to or below one, the business cost is either equal to the cost of the process or is more than cost of the process.

3.4 Performance evaluation techniques

The aim of an organization to achieve processes whose costs can be calculated, managed and optimized exists as long as the organization itself. An implementation of a performance management technique or an evaluation framework allows for the classification, tracking and management of such factors. The techniques available are spread across the enterprise i.e. there are techniques which can be implemented at the very operational level such as a process, and there are other techniques which can be implemented at the level of the organization or the enterprise. Some of the major techniques are as below:

3.4.1 Performance pyramid

The performance pyramid is defined by Lynch and Cross [71],is as shown in the Fig. 3.4. At the very bottom of the pyramid is the operational view on the business processes. These evaluate the quality, delivery, cycle time and waste at the process level. The middle and upper part of the pyramid brings in the business requirements on performance view. It also makes explicit the differences between measures that are of interest to external parties in this case the customer satisfaction.

3.4.2 Performance measurement matrix

Keegan et al.[72], in 1989, presented the performance measurement matrix shown in Fig. 3.5. The performance measurement matrix divides the performance measures un-

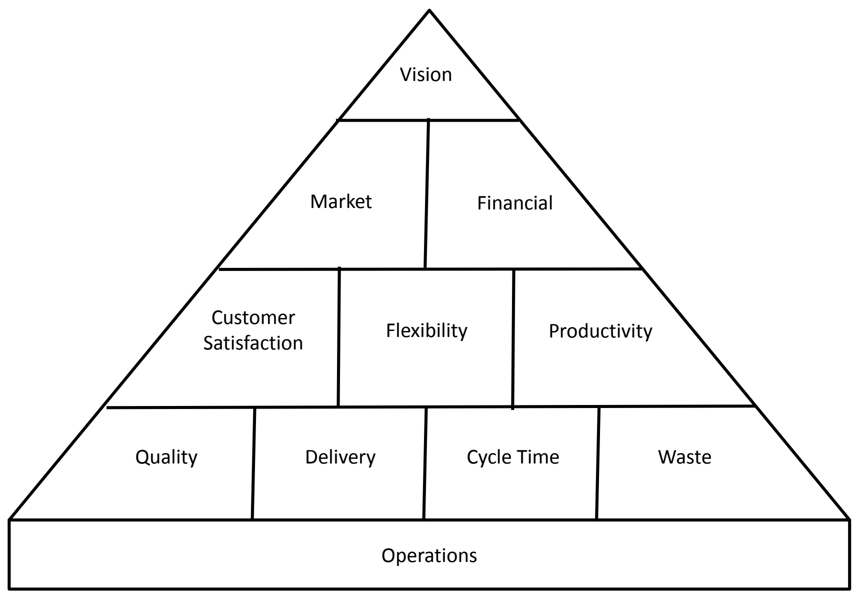


Figure 3.4: The performance pyramid as proposed by Lynch and Cross [71]

der 2 dimensions. The first dimension is if the measure is an internal measure or an external measure. The second dimension classifies the measure as a financial or a non-financial one. The drawback of this concept is that it does not link and show the interdependencies of the factors.

3.4.3 Results/Determinants Matrix

The results/determinants matrix is a framework proposed by Fitzgerald et al. [81] and was developed for the service industry. This is shown in Fig. 3.6. The concept behind the framework was to address the determinants which in turn would control the results. The measures for result are competitiveness, financial performance and that for the determinants are quality, flexibility, resource utilization and innovation. This framework, with this logic, took care of the criticism in the performance measurement matrix that the measures were not linked to each other.

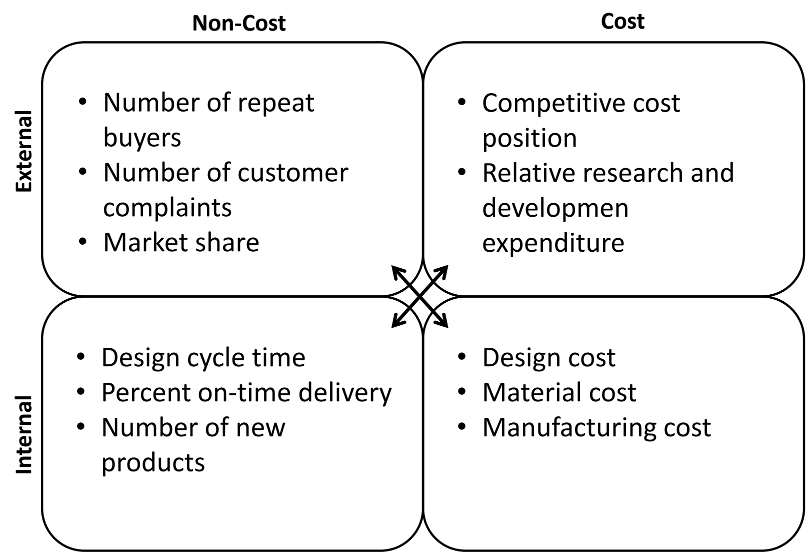


Figure 3.5: The performance measurement matrix as proposed by Keegan et al.[72]

Results	**Financial Performance**
	Competitiveness
Determinants	**Quality**
	Flexibility
	Resource utilisation
	Innovation

Figure 3.6: The results/determinants matrix framework from Fitzgerald et al. [81]

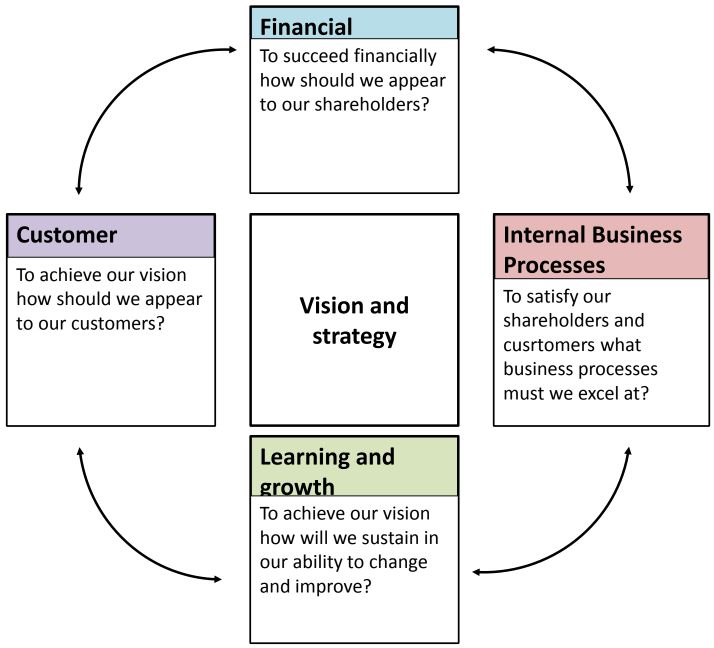


Figure 3.7: The balanced scorecard from Kaplan and Norton [79]

3.4.4 Balanced scorecard

The balanced scorecard [79] is a performance measurement framework developed in the 1990's by Kaplan and Norton and is presumably one of the most widely used. As the name suggests the framework is a scorecard of measures, both financial and non-financial, with base measures or target values attached to each one of them. The measures categorize themselves under four dimensions, i.e. the customer, internal business process, innovation and financial perspective. This is as shown in the Fig. 3.7.

The dimensions cover areas as below:

- **Learning and growth perspective:** This perspective covers human resources part. It addresses the growth of the employee by trainings, seminars etc. which are related to individual.

- **Business process perspective:** All the internal business processes are covered within this perspective. Metrics from this perspective gives a good indication if the business is running or not.

- **Customer perspective:** This perspective covers all aspects with the customers, their requirements and how satisfied they are.

- **Financial perspective:** All aspects covering the financial data of the organization are covered under this perspective.

3.4.5 Devils quadrangle

One of the only frameworks which both the industry and research believe as the best suited for performance evaluation of a workflow is the devils quadrangle. The devils quadrangle was proposed by Brand and Van der Kolk [13]. The devils quadrangle is represented as a quadrangle as shown in the Fig. 3.8.

It defines four dimensions i.e. time, cost, quality, and flexibility. Every business process needs to create a balance on these dimensions. When used, these dimensions are interpreted differently or at different maturity levels as the situation demands. Any change that is done to a business process leads to an impact on these dimensions. It is not necessary that the betterment on the value of one of the dimensions leads to an automatic betterment of the other one as well. One example which is seen almost always is the effort to decrease the cost dimension. This dimension usually shows that the quality dimension starts coming down or in other words the quality of the business process starts decreasing.

3.5 Conclusion

Cost as a financial performance factor has long been a core interest for the organizations for continuous improvement. In this chapter we have introduced cost, reliability and business cost and their role in performance evaluation. The chapter brings together the performance modeling, especially in the case of cost and reliability and prominent performance measure evaluation frameworks available in literature. These frameworks

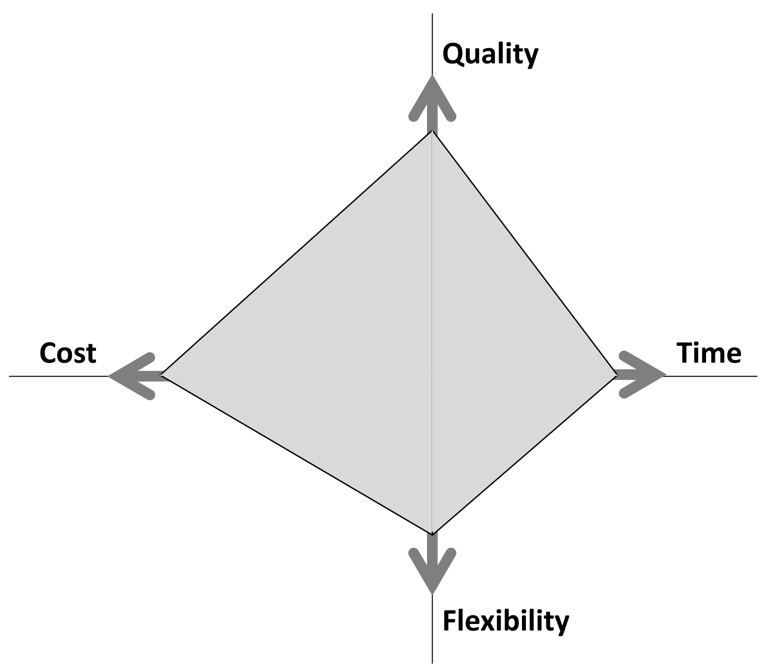


Figure 3.8: The devils quadrangle as proposed by Brand and Van der Kolk [13]

lay a basis for dimensions on which an organization should define and track performance measures. The measures are dependent on the organization which puts these frameworks in use. Once the measures have been defined, organizing these measures and tracking them are well aided by these frameworks.

Though almost all the frameworks are targeted as an enterprise level solution, only the devils quadrangle is really dealing at an operational level. The devils quadrangle deals with the implementation of a business process. Our interest is in the parameters for cost and reliability of a process. As already mentioned, these frameworks put together guidelines but do not specify in detail as to how somebody can actually measure these performance factors. The framework expects that the organization collects data and information which is relevant and do the calculation themselves.

In the chapter **Best practice evaluation** 9, we will be evaluating the different best practices recommended by the frameworks and evaluating their performance and impact on a business process. So as to have a representation of their impact, we will use the devils quadrangle to represent the difference in the measurement before and after implementing the best practice in these particular situations. We will be adapting the devils quadrangle with dimensions that we track and measure. This adaption is also covered in the chapter **Best practice evaluation** 9.

4

Business process model and notation

The business process model and notation (BPMN) [7] is a standardized graphical notation for drawing business processes. BPMN was developed by business process management initiative [7] and is being maintained under the ownership of object management group (OMG) [7]. BPMN offers a notation which can be understood by both the business and the technical side. Using the elements of BPMN, business process diagrams (BPD) can be drawn. The contents of this chapter have been referred from the BPMN specification[7] from OMG[7].

In this chapter we

1. Introduce business process model and notation, its aim and a brief overview of the developments till now.

2. Introduce the graphical notations available in BPMN for building a business process diagram

4.1 Aim of BPMN

BPMNs main aim to provide for a notation which can be used by business as well as IT to easily represent and understand processes. BPMN achieves this with graphical elements which can be used for representing the process steps, immaterial of its complexity. These are then mapped to constructs of execution languages, in particular business process execution language (BPEL).

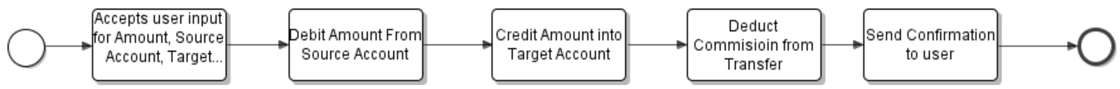


Figure 4.1: Sample business process diagram

In other words, the aim was to build a business process through the graphical notation which could be converted directly from its graphical representation into executable code. BPMN grew very fast in its popularity and this was especially because of the business users. Nevertheless it had, and still has, challenges in representing all the process requirements which are needed to completely generate an executable piece of code. The business process management Initiative and object management group have till now released four revisions of BPMN specifications. The recent one is BPMN 2.0. BPMN 2.0 has many enhancements and new features, the most prominent of them is the XML-based serialization which allows for interoperability between the different tools. The industry has nevertheless been slow in picking up the new release.

4.2 Elements of BPMN

The specification of BPMN 2.0 sticks to the aim of BPMN which is to create simple and understandable business process diagrams but at the same time represent and handle all the complexity that lies beneath the process. The Elements of BPMN are divided into five basic categories.

1. Flow objects
2. Data
3. Connecting objects
4. Swimlanes
5. Artifacts

4.2.1 Event

Something that happens within a process is called an event. Events have an impact on the flow of the process. Events are of three types, namely; start, intermediate, and end.

vents.pdf E vents.pdf

Figure 4.2: Events

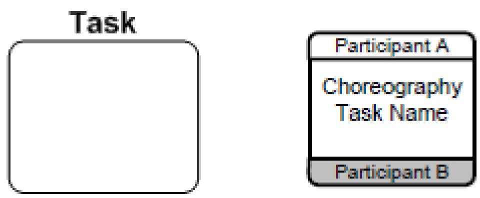


Figure 4.3: Activity

Element	Description	Notation
Flow Dimension (e.g. Start, Intermediate, End), Start, Intermediate, End	The three events are named according to when they are used in the process. Start events are used to start the process, Intermediate events are used in between the process which will affect the flow of the process, end events are used to end the process.	See Fig. 7.16
Type Dimension (e.g. None, Message, Timer, Error, Cancel, Compensation, Conditional, Link, Signal, Multiple, Terminate.)	Events in BPMN can have a trigger which are the cause of the event. These are for start and intermediate events. End events can have a result as well. Start events can "catch" a trigger, end events can "throw" a result. Intermediate events can "catch" and "throw".	See Fig. 7.16

4.2.2 Activity

An activity is a term used to represent and kind of performed work which is either atomic or non-atomic. sub-process and task are the types of activities in a process model.

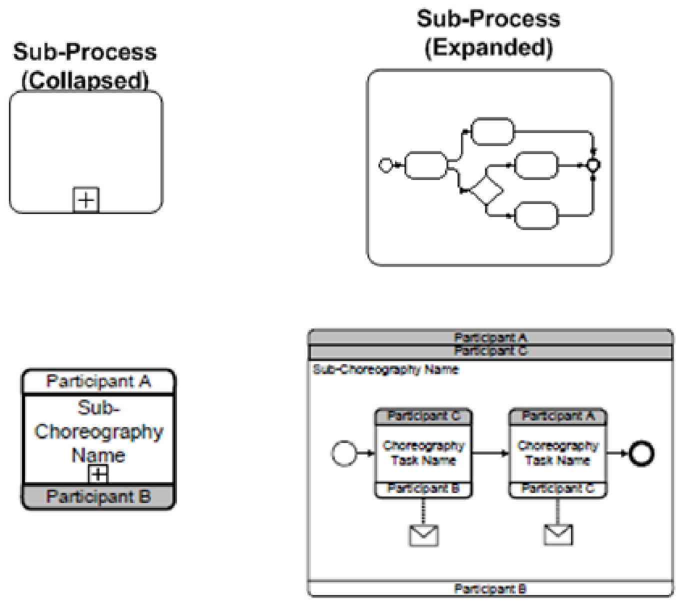


Figure 4.4: Process

Element	Description	Notation
Task (Atomic)	A task is an atomic Activity when it cannot be broken into a finer level.	See Fig. 6.2
Choreography task	An atomic activity within a choreography is called a choreography task.	See Fig. 6.2

4.2.3 Process/Sub-process (non-atomic)

A compound activity which can be broken down into finer levels and is part of a process or a choreography is called a sub-process.

Element	Description	Notation
Collapsed Sub-process	The BPMN symbol to show that there exists a sub-process but is not expanded at this moment.	See Fig. 6.3

Expanded sub-process	The sub-process is expanded and all the details (a process) are visible within its boundary.	See Fig. 6.3
Collapsed sub-choreography	The BPMN symbol to show that there exists a sub-choreography but is not expanded at this moment.	See Fig. 6.3
Expanded sub-choreography	The sub-choreography is expanded such that the details are visible.	See Fig. 6.3

4.2.4 Gateway

Process flows in BPMN are controlled with gateways. A gateway determines branching, forking, merging, and joining of paths.

Element	Description	Notation
Gateway control types	Gateways are diamond shaped. The symbols or icons which are inside the gateway define the type of the gateway and hence the behavior of the flow. The types of control include: Exclusive decision and merging: Both exclusive and event-based perform exclusive decisions and merging. These are shown either with or without the X marker. Event-based and parallel event-based gateways: both trigger a new instance of the process. Inclusive gateway decision and merging. Complex gateway: used in complex conditions. Parallel gateway: forking and joining.	See Fig. 4.5

4.2.5 Sequence Flow

The order in which the activities will be performed are shown with a sequence flow.

Element	Description	Notation

Normal flow	Normal flow are paths of sequence flow that starts from all events but the intermediate event attached to the boundary of an activity.	See Fig. 4.6
Uncontrolled flow	Uncontrolled flow are flows that are nor controlled by any conditions or gateway.	See Fig. 4.6
Conditional flow	This is exactly the opposite of uncontrolled flows. Conditional flows which are sequence flow which have a condition expression that are evaluated at runtime.	See Fig. 4.6
Default flow	In case of certain conditional gateways such as data-based exclusive gateways or inclusive gateways, one of the outgoing flows needs to be marked as default which will be used only when the other flows are not used.	See Fig. 4.6
Exception flow	Exception flow triggers a flow of the process outside the normal flow. This is started from an intermediate event attached to the boundary of an activity and throws an exception during the process.	See Fig. 4.6
Message flow	Flows which show the exchange or flow of messages between 2 participants are called Message flows.	See Fig. 4.6
Compensation association	A compensation intermediate event which is triggered because a transaction failed or a throw compensation event triggers a compensation association.	See Fig. 4.6

4.2.6 Data object

Element	Description	Notation
Data object	These provide information on the activities that need to be performed with their input and output.	See Fig. 4.7

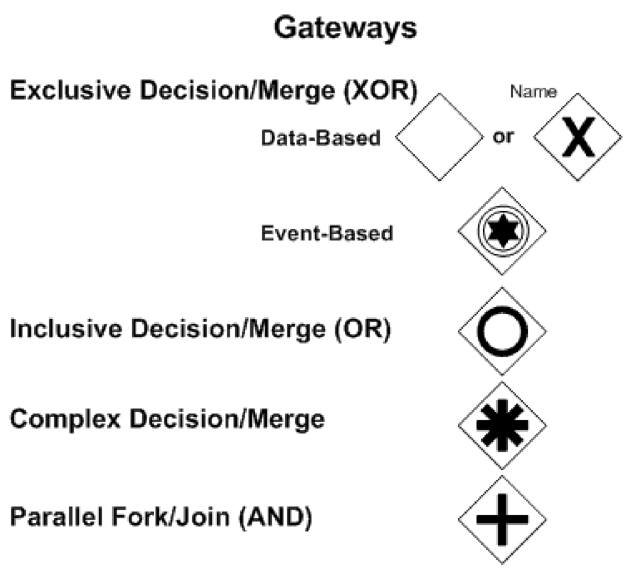


Figure 4.5: Gateways

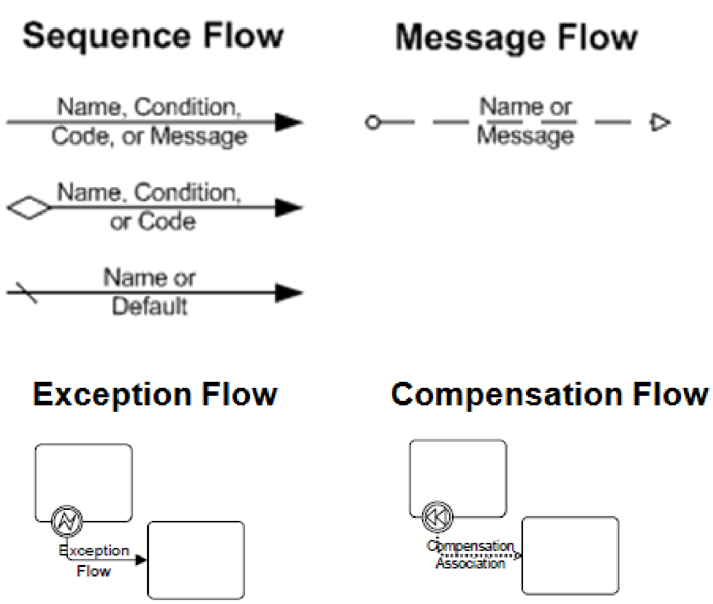


Figure 4.6: Sequence flow

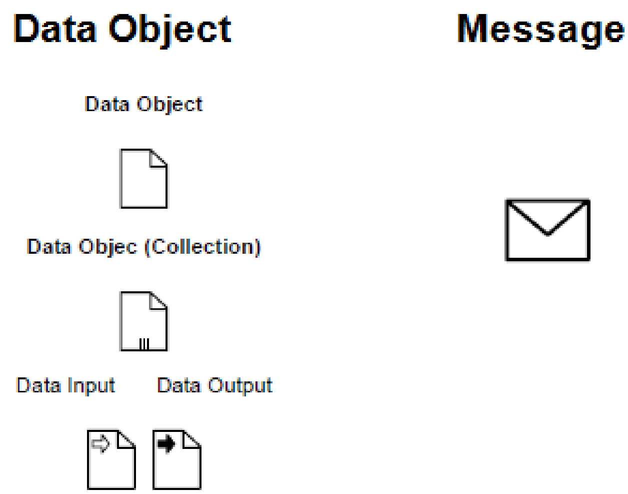


Figure 4.7: Data object and message

4.2.7 Message

Element	Description	Notation
Message	the contents of the communication between any two participants is packed into a message.	See Fig. 4.7

4.2.8 Fork

Element	Description	Notation
Fork	A path is divided into two with the help of a fork to enable parallelism in BPMN (also known as an AND-Split). There are two options: multiple outgoing sequence flows can be used to represent uncontrolled flow or a parallel gateway can be used which is usually in combination with other gateways.	See Fig. 4.8

4.2.9 Join

Element	Description	Notation
Join	Combining of two or more parallel paths into one path is done with Join.	See Fig. 4.8

Exclusive	The alternatives from such a decision are based on conditional expressions contained within the outgoing sequence flows . Only one alternative is chosen.	See Fig. 4.9
Event-based	The alternatives from such a decision are based on an event that occurs at that point. The event decides the alternative. Only one alternatives is chosen.	See Fig. 4.9
Inclusive	The alternatives from such a decision are based on conditional expressions contained within the outgoing sequence flows. It is a grouping of related independent Binary (Yes/No) decisions. Each of the paths are independent of the other and hence more than one alternatives can be chosen.	See Fig. 4.9
Merging	This combines two or more paths into one path.	See Fig. 4.9

4.2.11 Looping

Looping can be implemented in two different ways.

Element	Description	Notation
Activity looping	The attributes within the tasks and sub-processes will determine if the task is executed once or if they are repeated. There are two types of loops: standard and multi-Instance. A small looping indicator will be displayed at the bottom-center of the activity	See Fig. 4.10
Sequence flow looping	This is a well known way of producing a looping condition. These are created by connecting a sequence flow to an upstream object.	See Fig. 4.10

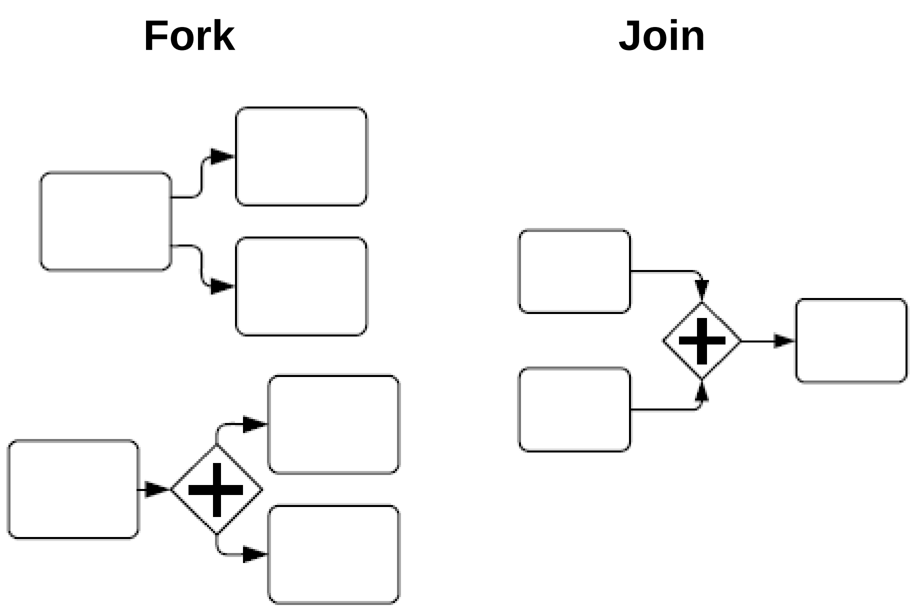


Figure 4.8: Fork and join

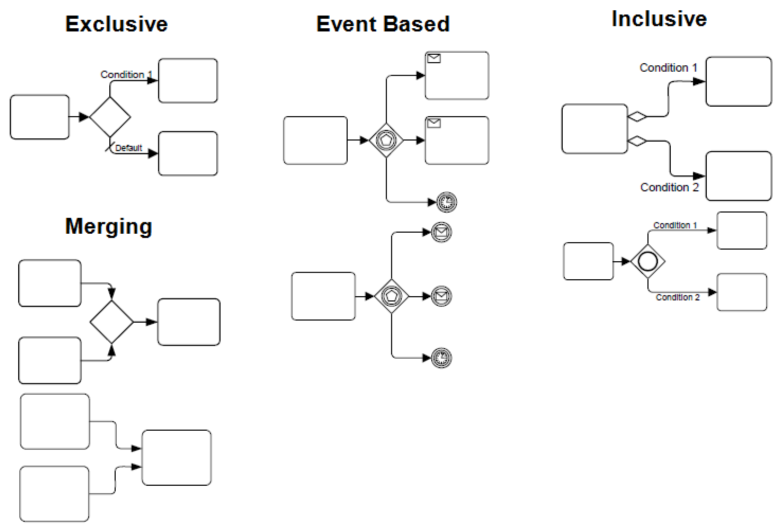


Figure 4.9: Decision and branching point

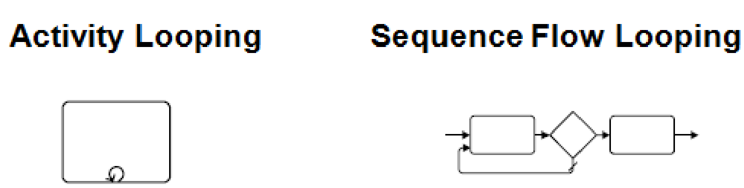


Figure 4.10: Looping

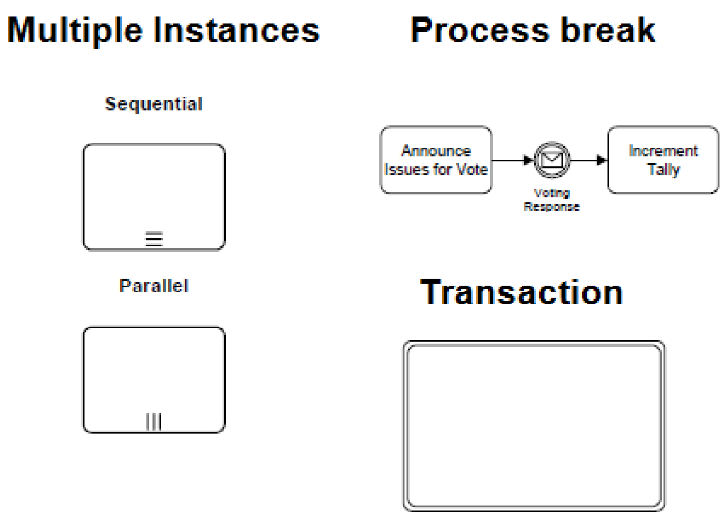


Figure 4.11: Other symbols

4.2.12 Other elements

Element	Description	Notation
Multiple instances	Tasks and sub-processes contain attributed which will determine if they are single instance or multiple.	See Fig. 4.11 and Fig. 4.12
Process break (pausing the process)	An expected delay in the process is represented with the help of a process break. An intermediate event is used to show the actual behavior.	See Fig. 4.11 and Fig. 4.12
Transaction	A sub-process that it is clearly defined when the activity is completed or canceled is called a transaction. The attributes of the activity will determine if the activity is a transaction.	See Fig. 4.11 and Fig. 4.12
Nested/Embedded sub-process	An activity that shares the same set of data as its parent process is called a nested/embedded sub-process.	See Fig. 4.11 and Fig. 4.12

Group (a box around a group of objects within the same category)	A group is a graphical grouping elements which belong to the same category. It does not affect the sequence flows.	See Fig. 4.11 and Fig. 4.12
Off-page connector	An indicator to show that a sequence flow has left one page and restarted in another page.	See Fig. 4.11 and Fig. 4.12
Association	Anformation and artifacts are linked to the graphical elements in BPMN through associations.	See Fig. 4.11 and Fig. 4.12
Text annotation (attached with an association)	An annotation where the modeler can provide textual information.	See Fig. 4.11 and Fig. 4.12
Pool	A participant is graphically represented through a pool.	See Fig. 4.11 and Fig. 4.12
Lanes	Lanes are partitions within a pool which are used to organize and categorize activities.	See Fig. 4.11 and Fig. 4.12

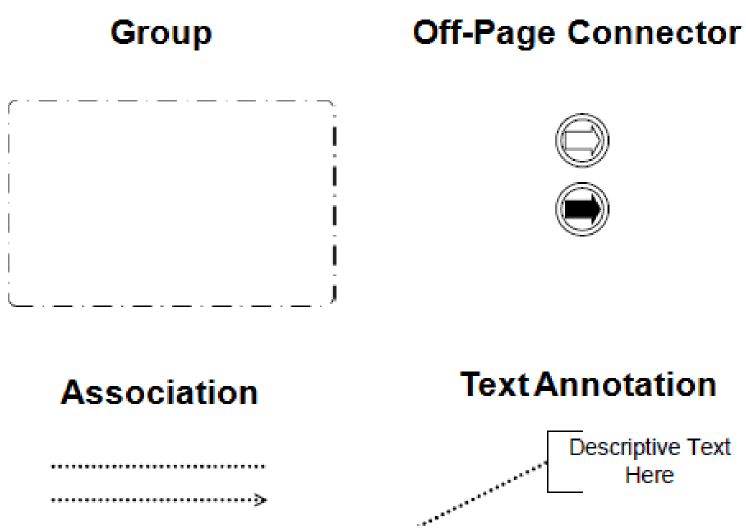


Figure 4.12: Other symbols

5

Cost calculation with patterns

BPMN is a standardized graphical notation for drawing business processes in a workflow [7]. It offers a standard notation with enough flexibility to represent complex situations in a manner which is understandable by all stakeholders. BPMN concentrates mainly on representing business processes. It does not deal with the quantitative aspects of a process such as cost and reliability.

In this chapter we propose a methodology for cost calculation by dividing a business process into patterns. Pattern is a word defined over and over depending on the context in which it is used. There are Patterns in the field of architecture, pure sciences, mathematics etc. In software a pattern is usually referred to the Design patterns. Design patterns are general solutions to regular problems in design of software.

Patterns in workflows are a widely implemented concept as they provide for standardized solutions to known and recurrent situations and problems. The patterns are identified or defined by considering the different perspectives of information systems. A research project [36, 43] in early 2000 defined a basis for the wide usage of workflow patterns. This research differentiates the perspectives in the system into four blocks: Control flow, Resource, Data, Exception handling. In total there are more than a hundred patterns documented as part of this research. The library of patterns continues to grow as more and more patterns are recognized.

The methodology we propose considers artifacts within a business process which is represented in business process diagram and attaches a parameter for cost and reliability to the same. It then breaks the business process into repetitive patterns and the

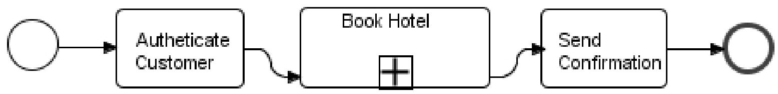


Figure 5.1: Business process diagram to book a hotel

cost, reliability and business cost of each pattern is calculated. In turn the overall cost, reliability and business cost of the complete business process are generated.

In this chapter we

1. Extend the elements of a business process diagram with parameters of cost and reliability.
2. Define a methodology by which the cost of a business process is calculated by considering the cost and reliability of each action or task in the process.
3. Break the business process, represented using BPMN, into repetitive patterns and a cost and reliability factor for each of these patterns is calculated.
4. Calculate the overall cost, reliability and the cost incurred to achieve one successful execution of the business process; the business cost of the process.
5. Implement a sensitivity analysis for a task and a pattern to identify areas which have the maximum impact on the business cost.

Example: Hotel booking process with costs

We consider an example of the hotel booking process as the basis here. The business process diagram (BPD) in Fig. 5.1 shows the business process of a hotel booking agency. The customer is first authenticated, a hotel is booked for the dates entered, and then a confirmation is sent back to the customer.

The business process diagram (BPD) in Fig. 5.2 is a blow up of the "Book Hotel" task from Fig. 5.1. We see that the task tries to book a room in "n" hotels until it has successfully booked one of the hotels or there are no free rooms for the entered dates in all the n hotels. If we assume that there are 10 hotels and each time we try to book

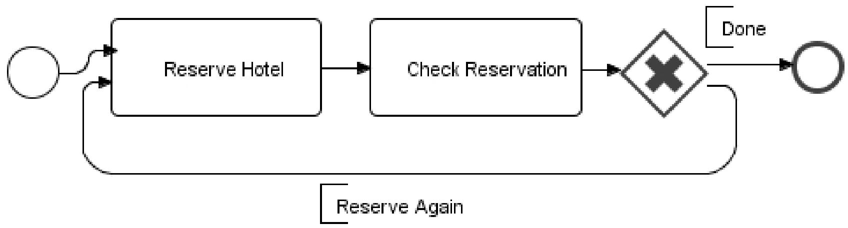


Figure 5.2: Detailed business process diagram for booking a hotel

we pay 2 units of cost, the minimum cost is 2 and the maximum cost is 20 making the average 11. It could be that we incur the maximum cost i.e. 20, and still the task of booking a hotel is not achieved.

5.1 Representation of cost and reliability in BPD's

A business process diagram (BPD) represents the business process and is made up of the different artifacts from BPMN. To calculate the cost of the process, we consider primarily the elements from BPMN which are atomic in nature and present a task or job that is done. Such an artifact is the most elementary level of a cost and reliability representation and cannot be broken down anymore. In BPMN, the artifacts which represent these characteristics fall under flow objects category. An atomic activity or events are artifacts which are atomic. The rest of the artifacts such as a sub-process etc. are not. The other flow object artifacts such as gateways decide on the flow of the process and do not represent a task to be done.

We assign two properties to each of the artifacts, one for the cost and the other for the reliability. We do this by the use of a simple extension in the form of an attached textual property. We define the property cost and reliability of an artifact as

$$Cost = C, where\ C >= 0 \tag{5.1}$$

$$Reliability = R, where\ 0 < R <= 1. \tag{5.2}$$

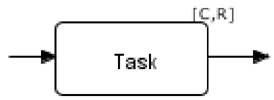


Figure 5.3: BPD with one task

Cost	Reliability	Business cost
5	1.0	5.00
5	0.9	5.55
5	0.8	6.25
5	0.7	7.14

Table 5.1: Variation of business cost according to reliability

Artifacts such as gateways do not need this property. In such cases the cost would be zero and the reliability would be one. In the Fig. 5.3 we consider one single task having a cost C and reliability R.

The rate of success of the task is given by the reliability of the task. We defined business cost as the cost to achieve a successful result. Hence the business cost in this case is the result of dividing the cost by the reliability.

$$BusinessCost = C/R \tag{5.3}$$

Table 5.1 shows a sample variation of the business cost as the reliability of the task changes.

The calculation of the business cost in this way is by the assumption that we always pay for a service to use it. That the service itself will be successful is dependent on its reliability. In other words, the business cost is always going to be higher than the cost when the reliability of the task is less than 1.

$$BusinessCost = Cost, \ Reliability = 1 \tag{5.4}$$

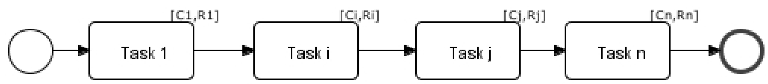


Figure 5.4: n tasks in a sequential order

$$BusinessCost > Cost,\ 0 < Reliability < 1 \tag{5.5}$$

5.2 Patterns for cost calculation

When we have BPD's with a number of tasks and events executed either in sequential, parallel or over certain pre-conditions, the cost, reliability and the business cost can be calculated by recognizing patterns which are repetitive. The values generated for each of these patterns put together gives the business cost of the overall business process.

We define four common patterns which we come across in business processes to evaluate these factors.

5.2.1 Pattern 1: n tasks in a sequential order

We are considering a pattern/process with

- n tasks or events in a sequential order
- Each has a cost and reliability associated with it
- No compensation
- No alternative tasks/flow to increase business reliability

A BPD representing n single tasks in a sequential order is as shown in Fig. 5.4

The cost of the BPD is the summation of the costs and the reliability of the BPD is the product of the reliabilities of all the events. These can be represented as:

$$Cost = \sum Ci,\ where\ Ci\ is\ the\ cost\ of\ task\ i \tag{5.6}$$

No. of tasks	Cost	Reliability
1	C1	R1
2	C2	R2
3	C3	R3
n	Cn	Rn

Table 5.2: Cost and reliability for n tasks

$$Reliability = \prod Ri,\ where\ Ri\ is\ the\ reliability\ of\ task\ i \tag{5.7}$$

Equation 5.3 gives us the business cost for a single task. Business processes contains tasks which are mutually inclusive i.e. the execution of a task is dependent on the successful completion of the tasks before it. Equation 5.3 can be used for patterns containing tasks which are mutually exclusive in nature. In case of a pattern as the one shown in Fig. 5.4 where we have n tasks, the formula above fails. The business cost is calculated by a recursive way, depending upon the number of tasks in the pattern. We consider, as shown in Table 5.2, n tasks where each task comes with a cost and reliability.

$$BusinessCost(1,1) = C1/R1 \tag{5.8}$$

In case of two tasks

$$BusinessCost(1,2) = C2/R2 + (C1/R1)/R2 \tag{5.9}$$

$$BusinessCost(1,2) = (C2 + (C1/R1))/R2 \tag{5.10}$$

$$BusinessCost(1,2) = (C2 + BusinessCost(1,1))/R2 \tag{5.11}$$

Going by equation 5.11, the business cost of a sequential/serial pattern with n tasks can be defined by equation 5.12:

$$BusinessCost(1,n) = (Cn + BusinessCost(1,n-1))/Rn \tag{5.12}$$

As the tasks are mutually inclusive, each task is executed only when the previous completes successfully. In the case that reliability of every task in the order is "1" i.e. the task never fails, the cost and business cost of such a pattern would always be the same. This is shown in the 5.13.

$$BusinessCost(1, n) = Cost(1, n)\ , when\ Reliability\ =\ 1 \tag{5.13}$$

In case the reliability is less than 1 the business cost is always higher than the cost of the n tasks in the pattern. This is shown in the 5.14.

$$BusinessCost(1, n) > Cost(1, n)\ , when\ Reliability\ >\ 1 \tag{5.14}$$

As the business cost considers the impact of successful execution of the previous tasks it is also true that the business cost of the pattern is always lesser than the cost of the pattern directly divided by the reliability of the pattern when it is less than 1. Reason for this is that the cost of a pattern sums up the costs of each task without considering the results of the tasks executed before it. This is shown in 5.15.

$$BusinessCost(1, n) < Cost(1, n)/Rel(1, n), when\ Reliability\ <\ 1 \tag{5.15}$$

Example Table 5.3 represents a sample of four tasks in a sequential order with a reliability variation.

From Table 5.3

$$Cost = 5 + 5 + 5 + 5 = 20 \tag{5.16}$$

$$Reliability = 1 * 0.9 * 0.8 * 0,7 = 0.504 \tag{5.17}$$

The resulting business cost is as shown in Table 5.4.

Task	Cost	Reliability
1	5	1
2	5	0.9
3	5	0.8
4	5	0.7

Table 5.3: Tasks with cost and reliability

Task	Cost	Reliability	Business cost
1	5	1	5
2	5	0.9	11.11
3	5	0.8	20.14
4	5	0.7	35.91

Table 5.4: Business cost

5.2.2 Pattern 2: n tasks in a parallel order

We are considering a pattern/process with

- n tasks in a parallel order
- Each task has a cost and reliability associated with it
- No compensation for any task
- No alternative tasks/flow to increase business reliability

A BPD representing n tasks in parallel order is as shown in Fig. 5.5.

When we have a BPD which has tasks arranged in a parallel manner, each flow in this pattern is a sequential flow with one or more tasks. A break up is as shown in Fig. 5.5. For each of the sequential patterns, having one or more tasks, the cost, reliability and the business cost is calculated as shown in Pattern 1.

The resulting cost and reliability of this parallel pattern then would be:

$$Cost = \Sigma Ci, \ (Ci \ is \ the \ cost \ of \ each \ flow \ in \ the \ parallel \ flow) \tag{5.18}$$

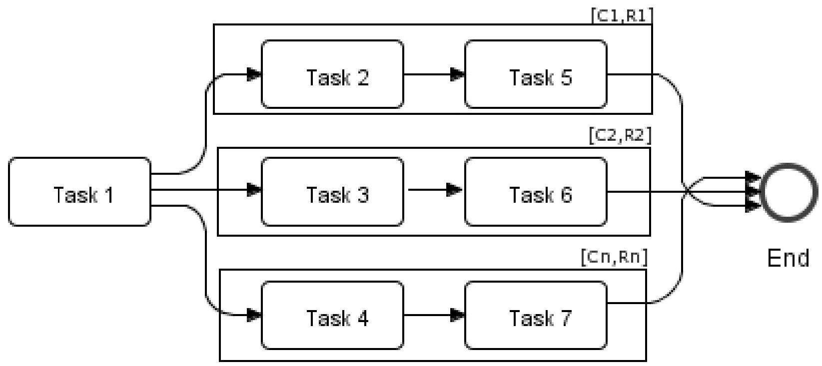


Figure 5.5: n tasks in parallel order

$$Reliability = \prod(R) \tag{5.19}$$

$$BusinessCost = \Sigma BusinessCost(i)\ (i\ is\ the\ pattern) \tag{5.20}$$

5.2.3 Pattern 3: Conditional branching

We are considering a pattern/process with

- A conditional branching
- Leading to different execution paths
- Each task has a cost and reliability associated with it

The situation here is the same as mentioned in the case of sequential tasks in Pattern 1. Nevertheless a probability has to be attached to each flow out of the gateway. The corresponding cost of the path is then multiplied by the probability which will lead to the cost of the whole branching.

$$Cost = \sum Pi * Cost(i),\ (Pi\ is\ the\ probability\ of\ taking\ path\ i) \tag{5.21}$$

$$Reliability = \sum PiRi,\ (Ri\ is\ the\ reliability\ of\ path\ i) \tag{5.22}$$

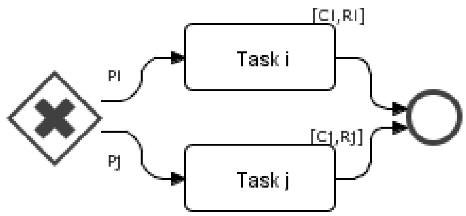


Figure 5.6: BPD with conditional branching

Branch	Probability	Reliability	Cost	Business cost
1	0.5	0.9	1	1.11
2	0.5	0.9	2	2.22

Table 5.5: Branches with probability

$$BusinessCost = \sum Pi * BusinessCost(i) \tag{5.23}$$

Example Table 5.5 represents a sample of conditional branching with tasks in a sequential order with a reliability variation.

Considering values from Table 5.5:

$$Cost = (0.5)1 + (0.5)2 = 1.5 \tag{5.24}$$

$$Reliability = 0.5 * 0.9 + 0.5 * 0.9 = 0.9 \tag{5.25}$$

$$BusinessCost = 0.5 * 1.1 + 0.5 * 2.2 = 1.66 \tag{5.26}$$

5.2.4 Pattern 4: "n" successive possibilities

We are considering a pattern/process with

- n different services each performing the same function
- Each task has a cost and reliability associated with it
- There is no compensation involved, either the service achieves the business value or fails

This is the pattern represented in the Fig. 5.2 which tries to book a room in n hotels. At an execution level, we see that the token first talks to the first hotel and then to the second and so on. Let's assume that there are "n" hotels such that

$$Hotel = Hotel1, Hotel2, ., ..Hoteln \tag{5.27}$$

$$Cost(Hoteli) = Ci \tag{5.28}$$

$$Reliability(Hoteli) = Ri \tag{5.29}$$

The cost is the sums of the costs of all the possibilities i.e. in this case the hotels. However in a pattern such as this the probability that hotel "n" will be contacted is dependent on the failure of the previous hotels i.e. till hotel "n-1" to provide the required service. This means that the cost increase when hotel "n" is contacted the cost of hotel "n" multiplied by the failure rate of the previous hotels. We define this cost as the "Actual cost" in such a pattern. In case of n possibilities each having a cost C, and reliability R, the actual cost is calculated as:

$$ActualCost = C \text{ , } \textit{with one possibility} \tag{5.30}$$

$$ActualCost(1,2) = ActualCost(C_1) + C_2 * (1 - R_1) \text{ , } \textit{with two possibilities} \tag{5.31}$$

$$ActualCost(1,n) = ActualCost(C_{n-1}) + C_n * (1 - R_{n-1}) \text{ , } \textit{with n possibilities} \tag{5.32}$$

No. Hotels	Reliability	ActualCost (AC)	BusinessCost
Hotel(1,1)	1-(1-R_1)	C_1	C_1/R_1
Hotel(1,2)	1-(1-R_1) $*(1-R_2)$	AC(Hotel(1,1))+$C_2(1-R_1)$	AC(1,2)/R_2
Hotel(1,n)	(1-((1-R_1) $*(1-R_2)..*(1-R_n)$)	AC(Hotel(1,n-1))+$C_n(1-R_{n-1})$	AC(1,n)/Rel_n

Table 5.6: Actual and Business cost

Option	Cost	Reliability	Cumulative-Rel	Actual cost	Business cost
Hotel 1	6	0.9	0.9000	6.00	6.67
Hotel 2	6	0.8	0.9800	6.60	6.73
Hotel 3	6	0.7	0.9940	6.72	6.76
Hotel 4	6	0.6	0.9976	6.76	6.77
Hotel 5	6	0.5	0.9988	6.77	6.78
Hotel 6	6	0.4	0.9993	6.78	6.78

Table 5.7: Variation of cost and reliability

The reliability factor goes up as the number of hotel goes up. In Table 5.6 we consider three cases; one hotel, two hotels, and n hotels. For each case we calculate the costs and the reliability.

Results from Table 5.6 can be represented with the following equations:

$$ActualCost(N_i, N_n) = ActualCost(Hotel(1, n-1)) + C_n(1 - R_{n-1}) \tag{5.33}$$

$$BusinessCost(N_i, N_n) = ActualCost/Reliability \tag{5.34}$$

$$Reliability(R_1, R_n) = 1 - ((1 - R_1)(1 - R_2)..(1 - R_n)) \tag{5.35}$$

Table 5.7 gives the variation of the costs and reliability on some sample values.

From Table 5.7 we see that the actual cost and the business cost increases as the reliability increases. But after attaining a certain level of reliability these costs equate

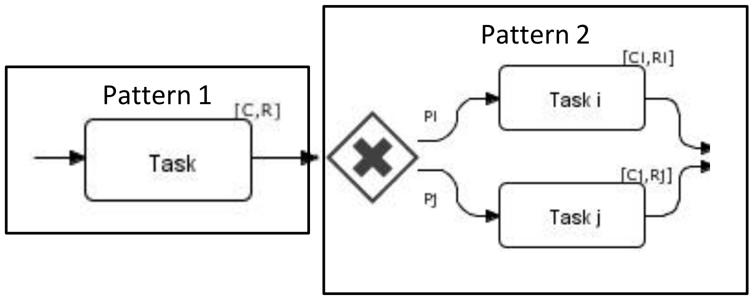


Figure 5.7: Sequential patterns

to each other.

5.3 Putting Patterns together

A business process is broken into patterns and for each pattern the cost, reliability and business cost is calculated as already discussed. In the previous section patterns which are commonly recognizable in a business process were defined. To calculate these factors for a business process the combination of the patterns and their interaction with each other needs to be taken into consideration. We define the interleaving combinations here.

5.3.1 Sequential Patterns

We are considering two patterns here such that each has a cost, reliability and business cost attached to it. The patterns internally need not be a sequential execution of tasks. An example is as shown in Fig. 5.7.

The figure shows a combination of two patterns which are sequential. Note that pattern 2 in the figure is in itself a conditional branching pattern. Cost of a business process is the summation of the costs of all the patterns in the process. The reliability is the product of the reliabilities of all the patterns.

$$Cost(Process) = \sum Cost(Pattern_i) \tag{5.36}$$

$$Reliability(Process) = \prod Reliability(Pattern_i) \tag{5.37}$$

The business cost calculation is dependent on both the business cost of the patterns and their reliabilities. For a process which has only one pattern, say P1, the business cost of the process would be the business cost of the pattern itself i.e.

$$BusinessCost(Process(P1)) = BusinessCost(P1) \tag{5.38}$$

In case the process has two patterns P1 and P2, the business cost of the process is calculated as

$$\begin{aligned} BusinessCost(Process(P1, P2)) = BusinessCost(P2) + \\ BusinessCost(Process(P1))/Reliability(R2) \end{aligned} \tag{5.39}$$

The logic here is the same as in the case of tasks in a sequential order. The patterns are mutually inclusive of each other and hence the need to consider the reliability while summing up the costs. Hence the business cost of a process with n patterns in a sequential order is calculated as:

$$\begin{aligned} BusinessCost(Process(P1, .., Pn)) = BusinessCost(Pn) + \\ BusinessCost(Process(P1, .., Pn-1))/Reliability(Rn) \end{aligned} \tag{5.40}$$

5.3.2 Patterns in parallel combinations

The situation where more than one pattern is executed in a parallel order is as in the pattern 'n tasks in a parallel order'. The calculation for cost, reliability and business cost is as already defined.

5.3.3 Patterns with error flow

Error flow in BPMN is addressed with the error event. An error event can either be a start event for a sub process, or be an intermediary event which shows an error flow

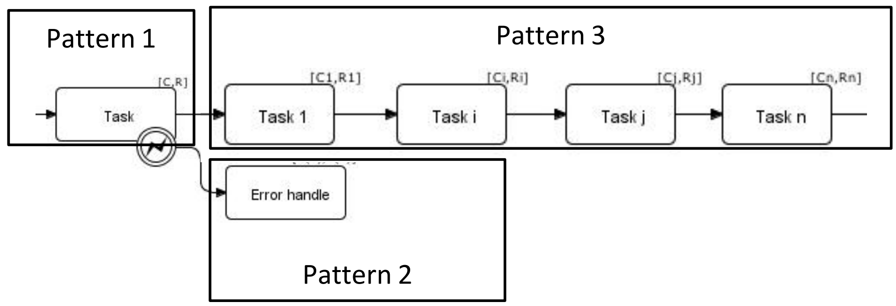


Figure 5.8: Error flow

or it is a final event which ends the process or triggers a new flow all together. An error event splits the process flow into two; one for the normal execution flow which is followed in case no error is encountered, the second flow is when an error occurs. An example is as shown in Fig. 5.8. Pattern 1 breaks into a normal flow pattern, pattern 3, and an error flow pattern, pattern 2.

The error flow and the normal flow are mutually exclusive i.e. at any time only one of the flows is followed. The cost calculation for the error flow depends on the probability of it being executed which is 1− the probability of the normal flow being executed. This is shown in equation 5.41.

$$Probability(ErrorFlow) = 1 - Probability(NormalFlow) \tag{5.41}$$

From the example as in Fig.5.8, the calculation of cost, business cost and reliability of both the normal (pattern P3) and error flow (pattern P2) patterns are as in the equations 5.42, 5.43, 5.44.

$$\begin{aligned} Cost(P2, P3) = Probability(P2) * Cost(P2) + \\ Probability(P3) * Cost(P3) \end{aligned} \tag{5.42}$$

$$\begin{aligned} Reliability(P2, P3) = ProbabilityP2 * Reliability(P2) + \\ ProbabilityP3 * Reliability(P3) \end{aligned} \tag{5.43}$$

$$\begin{aligned} BusinessCost(P2, P3) = &ProbabilityP2 * BusinessCost(P2) + \\ &ProbabilityP3 * BusinessCost(P3) \end{aligned} \tag{5.44}$$

5.4 Patterns triggering Compensation

Compensation is a set of steps or process flow which is outside the normal process flow. Compensation as a concept in BPMN is implemented with an intermediate and an end event. An intermediate event is attached to the activity and flows to the compensating activity which compensates the original activity. The end compensating event is defined further in the flow and triggers the compensating activity. A compensation end event can trigger any compensating event across the business process. The link or reference is in the handler. The triggering of one compensating event does not trigger the next compensating event in a sequential order as in a logical compensation scenario.

BPMN offers for a detailed Compensation procedure with the help of an event sub-process. An event sub-processes in BPMN is started due to a triggering end event and is a process which executes outside the normal flow. This allows for modeling a situation where all the compensating activities are modeled together in a sub-process which is executed when the situation arises, see Fig. 5.9.

A detailed compensation procedure as in the Fig. 5.9 is a combination of patterns on its own. The cost, reliability and business cost can be calculated with the pattern definitions made till now. The impact of such a pattern on the overall business cost of the process is dependent on the compensation triggering event. i.e. the event which triggers this compensation. When the compensating event is triggered the corresponding compensation procedure is executed. Fig. 5.9, the handle compensation is a pattern which triggers the activities 'Cancel Flight' and 'Cancel Hotel' in a sequential order. It also adds another activity 'Update Customer Record' in sequential order. This execution pattern is the same as 'Tasks in a sequential order' or 'Sequential Patterns'. The costs in such a case add up and the reliability is again the product of the reliabilities. The business cost is calculated in the same form as with sequential patterns.

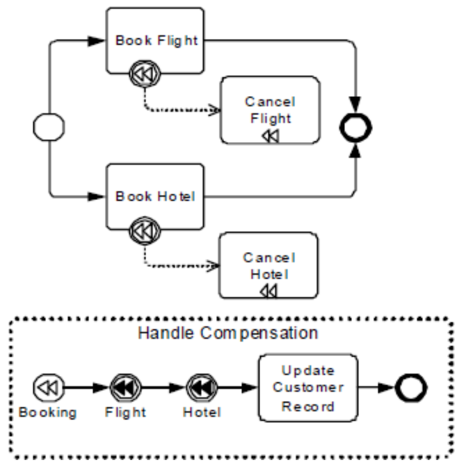


Figure 5.9: Compensation event sub process (source: BPMN specification)

5.5 Cost Analysis through Sensitivity Analysis

Service Level Agreements (SLAs) for service related process flows define at a contractual level the different obligations and agreements between the service provider and the customer. The service related process flows need to adhere to these agreements over a period of time. The metrics generated out of such a process flow lead to the evaluation of the quality of the Service. So as to achieve the right quality, it is important to first estimate and then analyze the different parameters which lead to this quality. Allan Clark and Stephen Gilmore [2] have analyzed the behavior of an automotive rescue situation with estimates for the time-duration involved. This allows the evaluation of a service on the amount of time which is needed to fulfill the same.

Calculation of the business cost plays a very vital role to evaluate if the conditions set in the service level agreements can be fulfilled or not. The next step, once the costs are calculated, will be to analyze and optimize the business costs of the process. The optimization of a process for the business cost can be done by varying the parameters attached to the elements of the business process. However it is impractical to vary the parameters of all the elements in the process or the pattern. There will be elements whose parameters when varied have absolutely no or very less impact on the business cost. Whereas there will be other elements whose parameters when varied will have a large impact on the business cost. A systematic and structured approach for parameter variation is done with the sensitivity analysis [20].

Cost	Reliability	Business cost
5	1.0	5.00
5	0.9	5.55
5	0.8	6.25
5	0.7	7.14
5	0.6	8.33
5	0.5	10.00
5	0.4	12.50
5	0.3	16.67
5	0.2	25.00
5	0.1	50.00

Table 5.8: Sensitivity analysis of business cost on reliability variation

Sensitivity analysis is the study of the changes in the output of a statistical model when the input parameters of the model undergo a systematic variation. By a sensitivity analysis we can identify the elements whose parameters have the maximum impact on the business cost of the process. The parameters are the cost and reliability which are attached to each of the elements. With the proposed approach for cost calculation, a sensitivity analysis can be implemented at the level of a single task, or at the level of a pattern and finally at the level of the process itself. This will help identify the most critical parts of the process which when optimized will help in fulfilling the agreements according to the Service Level Agreements.

The business cost of a task is as in the equation 5.3. An optimal situation is achieved when the reliability is at 1 i.e. no failures in this task. In such a case the business cost and cost of the task are the same. The sensitivity of the business cost with respect to the reliability can be analyzed by varying the reliability of the task between its minimum and maximum ranges. The table 5.1 varies the reliability of a single task between 1.0 and 0.7. So as to verify the complete sensitivity of the business cost with respect to reliability of the task, the table 5.8 varies the reliability from the minimum of 0.1 to the maximum of 1.0. The figure 5.10 is a graphical representation of the variation.

This shows that the business cost varies between 5, which is the actual cost, and 50 which is ten times the actual cost. Such an analysis shows the impact of the reliability,

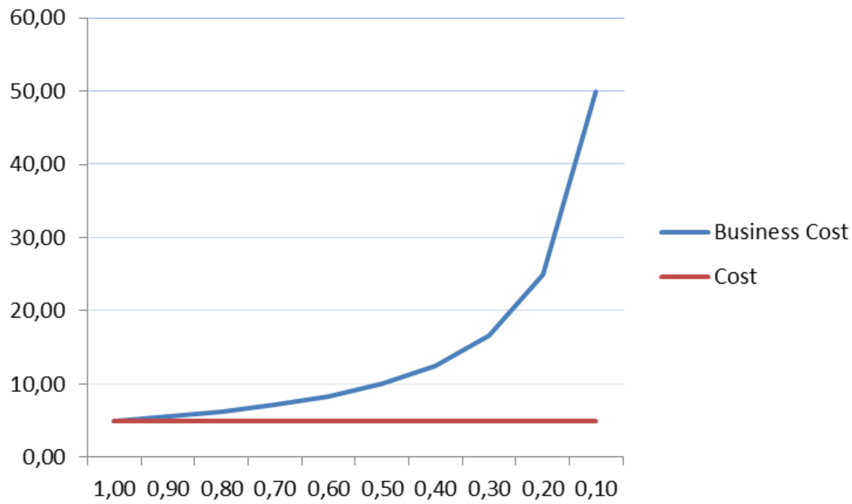


Figure 5.10: Sensitivity analysis of business cost

or in this case a low reliability, towards the business goals of the task or the process. An impact of such a non-functioning task or a task with low reliability on the financial parameters becomes measurable.

A sensitivity analysis on a pattern allows for identification of the task(s) which has(have) maximum impact on the business cost of the process. As the reliability is varied it allows for the analysis of the business cost of the complete process. Below is a case study which is used to show the impact of reliability on the business cost of a pattern through sensitivity analysis.

5.5.1 Case study: bank burglary scenario

We take as case study an example of a process that deals with a bank. Let's assume there is a service level agreement (SLA) between a bank and a private security office. The SLA deals with providing security such that in case of an alarm a security team reaches the bank in some pre-specified time. The process flow is as below:

1. A burglary at the bank begins.

2. An alarm rings at the local office of the private security.
3. The staff of the security evaluates the situation to decide if this is a real burglary or a false alarm.
4. In case the decision is that it is a real burglary, the security office dispatches a team to stop the burglary.
5. In case the decision is that this is a false alarm, then no team is dispatched.

The alarm might be a false alarm for a lot of reasons, maybe some employee pressed the button by mistake or there is a connection problem or this was a drill exercise within the bank etc. The private security office, every time the alarm goes off, believes there was a burglary. Its aim is to make sure that a security team reaches the bank in some pre-specified time.

So as assess the role of the parameter cost we induce a couple of conditions in performing this service at the level of the SLA. In the SLA we add conditions that are cost based as following:

- The security office is paid on an annual basis for this service.
- The payment is fixed and is independent of the number of the times the alarm goes off.
- Every time the security office fails in turning up at the bank in time, it pays an amount back to the bank as failure of service responsibility.

Due to the conditions above the security office always wants to keep the cost of executing the process as low as possible and at the same time make sure every time the security team reached the bank when a genuine alarm is raised.

We do a sensitivity analysis of the bank burglary scenario on the reliability factor. We assume costs for the different tasks in the process as shown in the table 5.9.

In the process steps of the business process, we consider the situation where the security team reaches the bank on time as the most important step to achieve the SLA. For this reason, in the table showing the costs, we have on purpose set the cost of the

Task	Cost
Alarm button is pressed at the bank	10
Alarm bell rings at the security office	5
Security office finds the bank and location	15
Security office decides on authenticity of alarm	5
Security team reaches the bank on time	80

Table 5.9: Bank burglary scenario

Task	Cost	Reliability	Business cost
Alarm button is pressed at the bank	10	0.95	10,53
Alarm bell rings at the security office	5	0,95	16.3
Security office finds the bank and number	15	0.95	33.0
Security office decides on authenticity of alarm	5	0.95	40.0
Security team reaches the bank on time	80	1.0	120.0

Table 5.10: Bank burglary scenario

task “ Security team reaches the bank on time” to a high amount. This is done so because the final action is the culmination of the agreed service and at the same time it also logically has the maximum cost. We now add reliability on each of these tasks and calculate the business cost of each task. We set the reliability of the task “ Security team reaches the bank on time” at 1.0 i.e. 100 percent. This is because we believe that once decided that this was a real alarm, the aim is to make sure the team reaches the bank.

The development of the business cost over the steps according to the reliability is as shown in the Fig. 5.11.

We implement sensitivity analysis to find out which task in the process has the maximum impact on the business cost of the process. We do this on a task by task basis. We vary the parameter ’Reliability’ of each of the tasks to see its impact on the development of the costs. We vary the reliability from 0.1 to 0.9 for each of the tasks. The figures show the impact on the business cost of the process when we vary the tasks reliability.

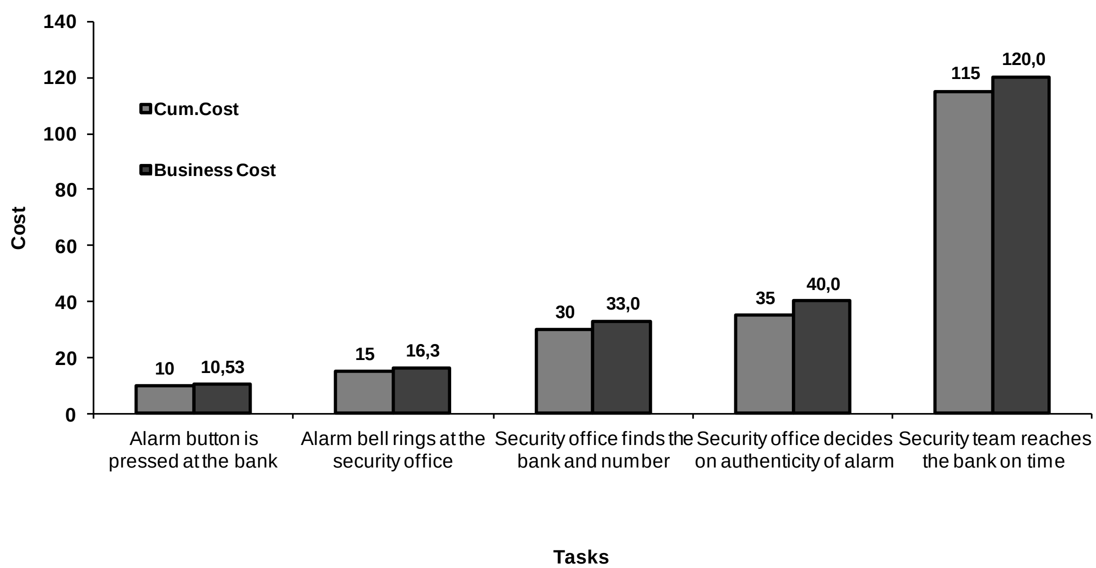


Figure 5.11: Cumulative cost and business cost development for bank burglary scenario

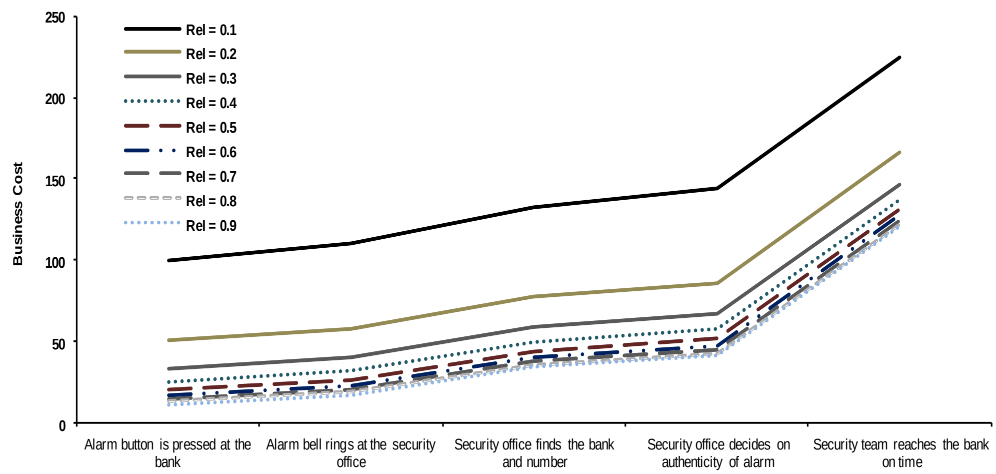


Figure 5.12: Reliability variation for “Alarm button is pressed at the bank”

We see from these figures that as the reliability goes down the business cost also increases, this is logical. What we also see is that as we move down the process chain, the reliability goes lower; it leads to rapid increase in the business cost. With variation of reliability of each task down the process, the resulting business cost increases i.e. when we compare the result of the variation of reliability on each of the task on the business cost, we see that the initial tasks have an impact which is much lower on the business cost then the tasks which are in the latter part of the process. Fig. 5.12 shows the variation of the reliability of the task “Alarm button is pressed at the bank”. We see that this has the minimum impact on the business cost in comparison to other tasks. A low reliability of 10 percent here pushes the business cost to 249.3.

We take the Fig. 5.15 as the variation in in reliability in “Security office decides on authenticity of alarm” pushes the business cost to 459.9. We see from the Fig. 5.16 that when the action “Security team reaches the bank on time” performs with a reliability of 0.1 i.e. 10 percent, this leads to a business cost of 1199.9. Whereas when this action performs with a reliability of 90 percent, this leads to a business cost of 133.3.

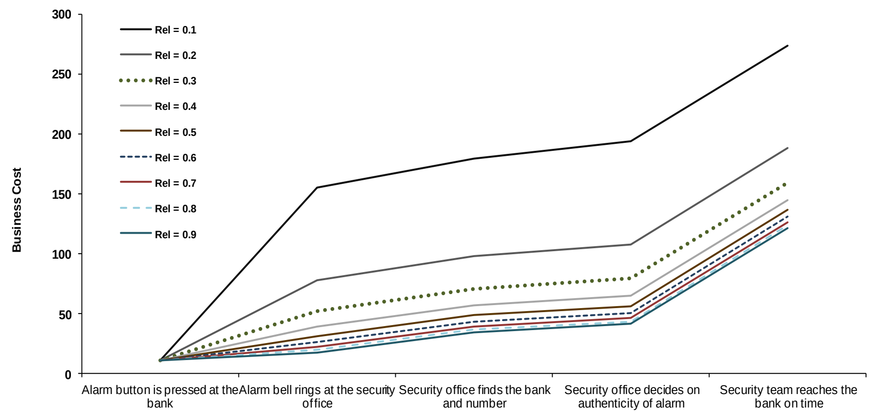


Figure 5.13: Reliability variation for “Alarm bell rings at the security office”

Hence it makes more sense to invest energy in trying to optimize and control the reliability of the tasks “Security office decides on authenticity of alarm” and “Security team reaches the bank on time” rather than on the other tasks in the process.

5.6 Conclusion

Design and execution of a business process is dependent on the underlying complex business requirements at an operative level. A modeling notation such as BPMN, with the different artifacts, allows for representation of simple and complex process requirements in a flexible manner.

In this chapter, we have presented a methodology by which the cost, business cost and reliability of a business process can be calculated. This methodology breaks a BPD into repetitive patterns and calculates these patterns for each of them. It considers the reliability as a varying factor which has an impact on achieving a business case i.e. to achieve one successful execution cycle of the business process. A sensitivity analysis is

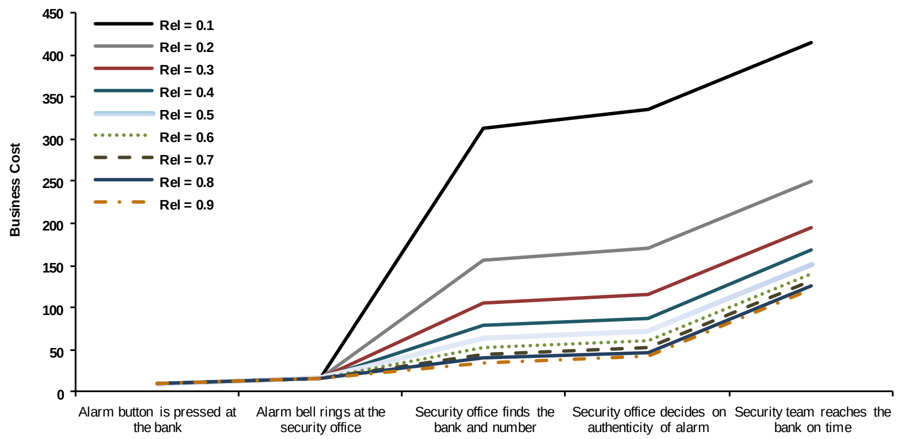


Figure 5.14: Reliability variation for "Security office finds the bank and number"

implemented to show the impact of parameter variation on the overall cost of the process. Through this approach the optimization activities on cost based parameters can be localized to particular activities of the process. The approach and results discussed in this chapter have been published[86].

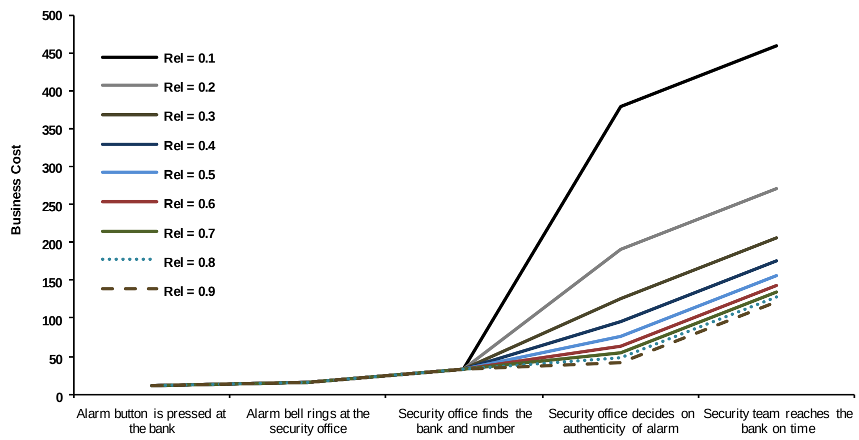


Figure 5.15: Reliability variation for "Security office decides on authenticity of alarm"

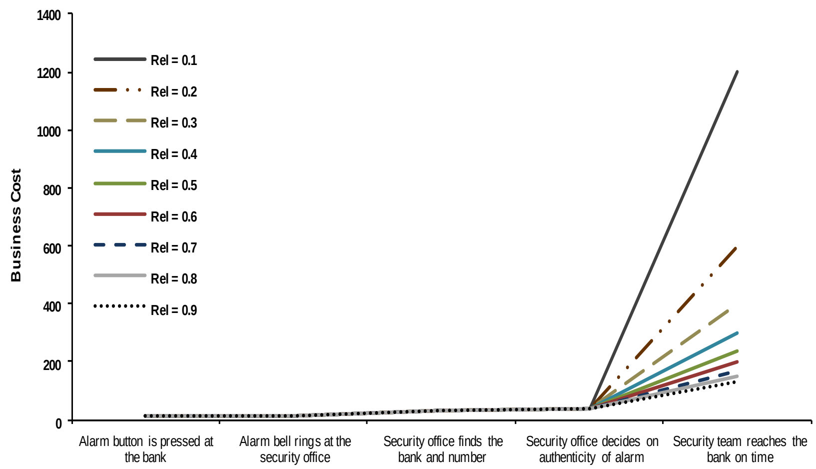


Figure 5.16: Reliability variation for “Security team reaches the bank on time”

6

Theoretical basis for cost calculation

BPMN is a graphical notation for business process modeling with the objective to support business process management for both technical users and business users. This flexibility which can bring out such representations allows the modeler to design workflows with very little restrictions. A lack of standardized guidelines which needs to be used by the whole industry leads to a situation where formalism or relating BPMN to an already well known and established data structure is challenging.

Due to BPMN's focus on a graphical notation, a representation in a way that can be related to Process algebra or calculus is not a priority. Formalism on the behavioral aspects of processes built with BPMN does not exist. We have, in the previous section, considered a BPD at its represented graphical level, broken them into patterns, and defined a mathematical approach for calculation of cost, reliability and business cost. We defined cost, reliability and business cost as:

- **Cost:** Every business process when executed costs a certain amount of money. Cost is everything that is spent. We define cost as everything that is spent and is measurable in executing the process. We define cost at the level of an activity in a business process; in other words every activity in a business process has a cost. Cost is a quantifiable factor, a measurable quantity.

- **Reliability:** Reliability is a combination of the technical reliability (which includes the infrastructure etc.) and the business reliability which is the reliability at which the business goal will be met or the service will lead to its successful completion. Reliability is the rate at which the process will reach its business goal.

- **Business cost:** Business cost is the cost incurred to execute the process such that it leads to a successful achievement of the business goal. When a task having a cost C and reliability R is executed n times, it incurs a cost C every time it is executed but it achieves the business objective only n*R times. The business cost i.e. the cost of achieving the business value, is the cost divided by the reliability of the task.

In this section we aim to model a formal approach for the behavioral aspects of a process defined through BPMN. We define a formalism through which the concept of patterns can be represented and fits the expressiveness of BPMN. We use Saga[8], a workflow model for long running transactions and the extended work done by Bruni [85] to establish the behavior for tasks and patterns in business Processes. For the rules proposed we calculate the cost, reliability and business cost.

In this chapter we:

1. Introduce the workflow model Saga and the formalism presented [85].
2. Define tasks and patterns, propose semantics for sequential, parallel, decision, error flow and compensation compositions.
3. Extend these semantics with cost, reliability and business cost.

6.1 Sagas

Sagas[8] is a workflow model for Long running transactions. Transactions which are long running and extend into hours if not days block precious resources until they complete. Saga presents a workflow model where Long running transaction can be broken down into smaller parts, which execute in a sequence and can be interleaved with the other transactions. The aim of the complete process is achieved when all the interleaved transactions complete successfully. In case one (or more) of them fail, the concept of compensation is introduced. In case of a failed transaction Saga proposes two ways to recover. Backward recovery is to trigger a compensating transaction which will undo the effects of the already completed transactions. Forward recovery is to execute the rest of the transactions. Forward recovery is not always suitable and hence Backward recovery through compensation is the preferred choice. By implementing this workflow model, exclusive access and lock down of resources to particular transactions can be reduced.

With Sagas as the basis, Bruni[85] has proposed a transactional calculi for programming languages. This calculi takes into consideration two key aspects in transactional flow: boundaries and compensation; and models the behavior of transactions in sequential, parallel, nesting, exception handling, and programmable compensations. A Saga is a sequence of atomic activities. The atomic Activities are called actions, steps or sub transactions. The sub transactions either complete successfully or abort. A partial execution of Saga is not desirable and hence when it occurs it should be compensated for. For some activity Ai, the compensation Bi is executed when Ai was successful but some activity Aj failed such that $j > i$ in the execution order. In such a case the order of execution of the compensating activities will be $Bj_1 \rightarrow Bj_2 \rightarrow Bj_3$ ......

According to the semantics proposed, the set of all Sagas is represented by Step (an activity and a compensating activity), process (combination of activities) and the set of processes itself which form the Saga. The possible result of the execution of a Saga is represented by the set which contains commit, compensated or an abnormal termination as results. The result of the execution is represented a context Γ . This context is a partial function which maps any activity to the result obtained with its execution i.e. either commits, is compensated or abnormally aborts.

6.2 Definitions

We base ourselves on the mathematical approach to break BPD into patterns for the calculation of cost, reliability and business cost. Our aim is to define a calculi which will model the behavioral aspects from BPMN through patterns for cost and reliability calculation. The rules define the interleaving situations and possible outcomes. We base our approach on transactional calculi proposed by Bruni[85]. We believe that a business process which has been modeled using BPMN is a long running transaction and all the activities defined perform as part of a Saga S. In our case such a Saga is made up of patterns which are in turn made up of activities or tasks which cannot be broken down anymore.

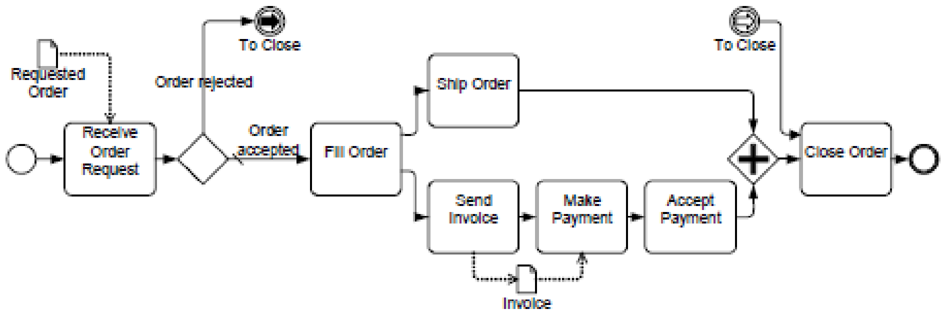


Figure 6.1: A sample business process (source: BPMN specification)

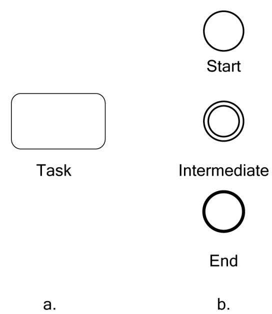


Figure 6.2: Tasks and events in BPMN

We define task and patterns as:

Definition 1 - Task: An activity or event which represents an act of doing something and is atomic in nature. In simplicity it can be represented by the following grammar.

$$(Task)\ T ::=\ 0 \mid T1 \tag{6.1}$$

BPMN events and tasks are as shown in Fig. 6.2

We define a task as the most elementary representation of doing an activity and cannot be broken down any further. By this definition a task in BPMN is an atomic activity and belongs to the flow object category. It also includes events (start, intermediate and end) as they are used to show that something happens. Artifacts which do not belong to this category are gateways as these are used to decide on the process flow through decisions or merging. Gateways do not represent a work that needs to be

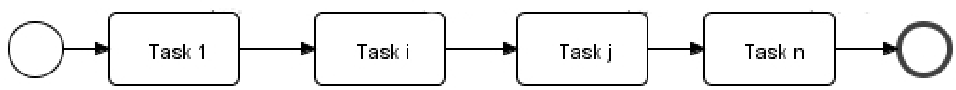


Figure 6.3: A pattern

done. Also sub-processes which are of type activity are not part of this as they are not atomic in nature.

Definition 2 - Sequential pattern: A set of tasks T which execute in a predefined sequential order such that the first task has either none or more than one incoming connector and no task has more than one outgoing connector forms a pattern.

In more detail: *A set of tasks T1,T2,..,Tn forms a pattern P when*
Sequential Execution: *The tasks execute in a sequential order i.e. T1 ; T2 ; .... ; Tn*
First task: *T1, has either none or more than one incoming connector*
Last task: *Tn, has either no outgoing connector or has more than one outgoing connector*
Intersection: *For all other patterns Pi, the intersection P ∩Pi = null*

It is represented with the following grammar:

$$(Pattern)\ P\ ::=\ T \mid T\ ;\ P \tag{6.2}$$

A pattern is as shown in the figure
We define a Saga as the combination of patterns.

$$(Saga)\ S ::= [P] \tag{6.3}$$

The result of the tasks when executed lead to a direct impact on the result of the patterns which in turn impacts the result of the business Process itself. We define the result as commits successfully, fail with(out) an error flow or abnormally terminates. The possible results of the execution of a task, pattern and business process is represented by the set R, where R is represented by

$$R = \{\odot, \oplus, \otimes\} \tag{6.4}$$

where
$\odot \rightarrow$ stands for successful commit case
$\oplus \rightarrow$ stands for failure with(out) an error flow case
$\otimes \rightarrow$ stands for abnormal termination

This information on the result of the execution is represented by a context Γ which is a partial function over the atomic activity T and the pattern P and maps them to the result obtained with its execution.

$$\Gamma : P \rightarrow \{R\} \tag{6.5}$$

In the coming sections we define the rules for the different execution patterns. The rules cover the three result cases already mentioned. Apart from these, we also define a business case scenario for the 'Successful commit case' which calculates the business cost for the situation.

6.3 Basic task semantics

The basic rules for behavior of tasks are as shown in Rule.6.6 and Rule.6.7. These rules cover two situations i.e. a task either commits successfully or terminates abnormally. Note that the situation where a task runs through an error flow situation is not taken into consideration here. Reason is the way we divide the business process into patterns. The error flow out of a task is outside the normal flow and hence is another pattern. We define rules for this outcome in the section for patterns.

$$T \rightarrow \odot \tag{6.6}$$

$$T \rightarrow \otimes \tag{6.7}$$

Our aim is to create a representation of the behavior of cost and reliability when a task is executed. i.e. each task in Saga has a certain cost C and has a reliability of R at which it succeeds. We also assume that when a task is performed, immaterial of the result of the performed task, the cost C is incurred. Hence, we extend the rules with the addition of two new parameters, one for reliability and the other for cost;

$$C = Cost\ Incurred \tag{6.8}$$

$$R = Reliability \tag{6.9}$$

In the successful completion case, Rule.6.10, the task costs C and achieves its intention with a reliability R. In case of a failure, Rule.6.11, the task still costs C but will achieve this result with a reliability of $1 - R$.

In the business case i.e. the cost of achieving a successful completion in 'n' trials would be the result of dividing of the cost by reliability. This rule is shown in Rule.6.12.

$$T \xrightarrow{C,R} \odot \tag{6.10}$$

$$T \xrightarrow{C,1-R} \otimes \tag{6.11}$$

$$T \xrightarrow{C/R} \odot \tag{6.12}$$

6.4 Tasks in a sequential order

The result of the execution of tasks in a sequential order is dependent on the result of each of the tasks. The Rule. 6.21 shows the case where two tasks execute in sequential order and both commit. The costs are the summation of the costs of the tasks and the reliability is the product of the tasks. The rules, Rule. 6.14 and Rule. 6.14, shows the case of a failure where either the first or the second task fails. A generalized representation is shown in Fig. 4.9 where a series of tasks which execute in sequential order fails.

The reliability that this happens is 1 minus the reliability of a successful completion.

$$\frac{\Gamma \mapsto T_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto T_2 \xrightarrow{C_2,R_2} \odot}{\Gamma \mapsto (T_1;T_2) \xrightarrow{\sum_{i=1}^{2} C_i, \prod_{i=1}^{2} R_i} \odot} \tag{6.13}$$

$$\frac{\Gamma \mapsto T_1 \xrightarrow{C_1,1-R_1} \otimes}{\Gamma \mapsto (T_1;T_2) \xrightarrow{C_1,1-R_1} \otimes} \tag{6.14}$$

$$\frac{\Gamma \mapsto T_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto T_2 \xrightarrow{C_2,1-R_2} \otimes}{\Gamma \mapsto (T_1;T_2) \xrightarrow{\sum_{i=1}^{2} C_i, R_1\times(1-R_2)} \otimes} \tag{6.15}$$

$$\frac{\Gamma \mapsto T_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto T_2 \xrightarrow{C_2,R_2} \odot \quad ....T_{j-1} \xrightarrow{C_{j-1},R_{j-1}} \odot \; T_j \xrightarrow{C_j,R_j} \otimes}{\Gamma \mapsto (T_1;T_2;....;T_j) \xrightarrow{\sum_{i=1}^{j} C_i, 1-\prod_{i=1}^{j} R_i} \otimes} \tag{6.16}$$

Business case For a business case situation where the cost of achieving of one successful commit in 'n' trials is as shown in Fig. 4.10. Business processes contains tasks which are mutually inclusive i.e. the execution of a task is dependent on the successful completion of the tasks before it. Hence the business cost of executing T1 and T2 successfully in a sequential manner is dependent on the business cost of T1. In case of n tasks which are executing in a sequential order the business cost is calculated by a recursive way, depending upon the number of tasks.

The business cost is calculated as shown in Rule. 6.17.

$$\frac{\Gamma \mapsto T_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto T_2 \xrightarrow{C_2,R_2} \odot}{\Gamma \mapsto (T_1;T_2) \xrightarrow{(C_2+(C_1/R_1))/R_2,\prod_{i=1}^{2} R_i} \odot} \tag{6.17}$$

The rule can be represented as in Rule.6.18.

$$\frac{\Gamma \mapsto T_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto T_2 \xrightarrow{C_2,R_2} \odot}{\Gamma \mapsto (T_1;T_2) \xrightarrow{(C_2+BusinessCost(T_1))/R_2,\prod_{i=1}^{2} R_i} \odot} \tag{6.18}$$

We extend this for three tasks in sequential order. In such a case the business cost would be as shown in Rule. 6.19.

$$\frac{\Gamma \mapsto T_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto T_2 \xrightarrow{C_2,R_2} \odot \quad \Gamma \mapsto T_3 \xrightarrow{C_3,R_3} \odot}{\Gamma \mapsto (T_1;T_2;T_3) \xrightarrow{(C_3+BusinessCost(T_1,T2))/R_3,\prod_{i=1}^{3} R_i} \odot} \tag{6.19}$$

6.5 Patterns

A pattern is a collection of tasks which are executed sequentially. The result of a pattern is dependent on the result of each task in the pattern.

In more detail: *If P is a pattern with tasks T1,T2,..,Tn then*
Successful Completion: $\forall$ *Ti such that Ti* $\in$ *P, Ti* $\rightarrow \odot$
Abnormal termination *there is at least one Ti such that Ti* $\in$ *P, Ti* $\rightarrow \otimes$
Failure with error flow *There exists a pattern Pi, which is triggered when Tn reaches a failed state.*

If P is a pattern such that P is the sequential execution of n tasks then the cost of P is the summation of the cost of all tasks and the reliability is the product of the reliabilities of the tasks. The rules here are as in the case of tasks. The sequential execution of tasks means that the execution of each task is dependent on the successful completion of the tasks before it i.e. they are not mutually exclusive in nature. The business cost of the process is calculated recursively as it is dependent on the completion of each task. In case of one task, two and three tasks the business cost are as defined in the rules for tasks.

6.6 Error flow representation

As already defined a pattern is a set of tasks which execute in a sequential order. The last task in a pattern has either none or more than one connector to other BPMN artifacts. The artifacts to which the connectors flow into, form patterns again. The concept of error flow in BPMN is addressed with the error event. An error event can either be a start event for a sub-process, or be an intermediary event which shows an error flow or it is a final event which ends the process or triggers a new flow all together.

When we separate a process into tasks and patterns, the error flow combination as represented in BPMN separates into another pattern. This error handle pattern is executed when the original pattern runs into an error flow situation. We use this concept as we define the rules for pattern combinations in the sections below.

6.7 Sequential pattern combinations

We extend the semantics for sequential patterns as below:

$$(Pattern)\ P\ ::=\ P\ ;\ P \tag{6.20}$$

Patterns are interleaved with other patterns either sequentially, in parallel or through particular decision conditions (gateways). We consider two patterns P_1 and P_2 which execute sequentially. Rule. 6.21 is the case when the patterns commit successfully. The costs sum up and the reliability is the product.

$$\frac{\Gamma \mapsto P_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto P_2 \xrightarrow{C_2,R_2} \odot}{\Gamma \mapsto (P_1;P_2) \xrightarrow{\sum_{i=1}^{2} C_i, \prod_{i=1}^{2} R_i} \odot} \tag{6.21}$$

Rules in Rule. 6.22 and Rule. 6.23 is the case where an error flow is triggered. An unsuccessful result is achieved when any one of the two patterns does not commit successfully. Either of the patterns, when unsuccessful, triggers an error flow pattern. The Rule. 6.22 is the case when P_1 fails and triggers the error flow pattern $\mathrm{P}_{1ErrorFlow}$. The Rule. 6.23 is the case when P_1 commits but P_2 fails and in turn triggers the error flow pattern $\mathrm{P}_{2ErrorFlow}$. The result is dependent on the result of the error flow pattern. If the error flow commits successfully then the process reaches a failure case but with an error flow, else the process reaches an abnormal termination.

$$\frac{\Gamma \mapsto P_1 \rightarrow \oplus \quad \Gamma \mapsto P_{1ErrorFlow} \rightarrow \lambda}{\Gamma \mapsto (P_1;P_2) \rightarrow \gamma} \tag{6.22}$$

Where
$\gamma \rightarrow \oplus \; if \; \lambda \rightarrow \odot$
$\gamma \rightarrow \otimes \; if \; \lambda \rightarrow \otimes$

$$\frac{\Gamma \mapsto P_1 \rightarrow \odot \quad \Gamma \mapsto P_2 \rightarrow \oplus \quad \Gamma \mapsto P_{2ErrorFlow} \rightarrow \lambda}{\Gamma \mapsto (P_1;P_2) \rightarrow \gamma} \tag{6.23}$$

Where
$\gamma \rightarrow \oplus \; if \; \lambda \rightarrow \odot$
$\gamma \rightarrow \otimes \; if \; \lambda \rightarrow \otimes$

An abnormal termination is reached when any one of these patterns reaches an abnormal termination. Rule. 6.24 shows the generalized representation of an abnormal situation. In this case there is no error flow defined.

$$\frac{\Gamma \mapsto P_1 \xrightarrow{C_1} \gamma \quad \Gamma \mapsto P_2 \xrightarrow{C_2, 1-R_2} \lambda}{\Gamma \mapsto (P_1; P_2) \xrightarrow{\sum_{i=1}^{2} C_i, 1-\prod_{i=1}^{2} R_i} \otimes} \tag{6.24}$$

Where
$\gamma \rightarrow \odot \; if \; \lambda \rightarrow \otimes$
$\gamma \rightarrow \otimes$ *and* P_2 *is not executed*

Business case The business case calculation is dependent on both the business cost of the patterns and their reliabilities. In case there was only one pattern, the business cost of the process would be the business cost of the pattern itself. In case of two patterns P_1 and P_2 which execute sequentially, the business cost of the pattern is calculated as in Rule. 6.25.

$$\frac{\Gamma \mapsto P_1 \xrightarrow{C_1, R_1} \odot \quad \Gamma \mapsto P_2 \xrightarrow{C_2, R_2} \odot}{\Gamma \mapsto (P_1; P_2) \xrightarrow{BusinessCost(P_2) + BusinessCost(P_1)/R_2, \prod_{i=1}^{2} R_i} \odot} \tag{6.25}$$

6.8 Parallel pattern combinations

We extend the semantics of patterns as below:

$$(Pattern)\ P \ ::= \ P \ || \ P \tag{6.26}$$

Parallel patterns can be represented in BPMN as shown in Fig. 6.4.

The Rule. 6.27 is the case when two patterns execute in parallel to each other and commit successfully. The cost and reliability calculations follow the same principles as in case of sequential combinations. The costs sum up whereas the reliability is the product.

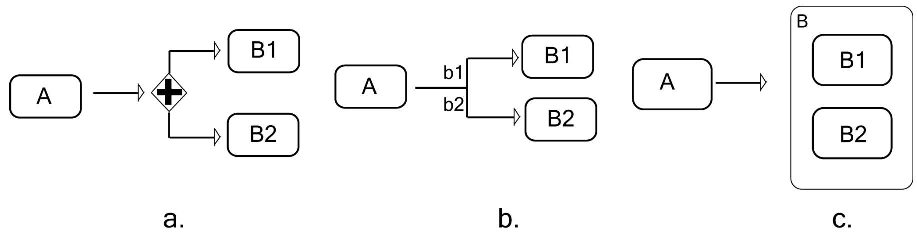


Figure 6.4: Parallel patterns in BPMN (source: BPMN specification)

$$\frac{\Gamma \mapsto P_1 \xrightarrow{C_1,R_1} \odot \quad \Gamma \mapsto P_2 \xrightarrow{C_2,R_2} \odot}{\Gamma \mapsto (P_1||P_2) \xrightarrow{\sum_{i=1}^{2} C_i, \prod_{i=1}^{2} R_i} \odot} \tag{6.27}$$

Rule. 6.28 is the case when the pattern(s) reach a failure situation and trigger an error flow pattern.

$$\frac{\Gamma \mapsto P_1 \rightarrow \oplus \quad \Gamma \mapsto P_2 \rightarrow \oplus}{\Gamma \mapsto (P_1||P_2) \rightarrow \otimes} \tag{6.28}$$

This result is achieved when any one or both the patterns does not commit successfully. Note that even though there is an error flow designed, the parallel execution reaches an abnormal termination. This is because a deadlock situation is reached. When the patterns are executing together and when one of them completes, it waits for the other pattern to also complete before triggering the next pattern. This waiting phase does not terminate when the other pattern reaches a failure state. BPMN does not allow for a situation where the unsuccessful or abnormal termination of one of the patterns stops the other pattern which is running in parallel. In such a case the process cannot continue anymore and hence reaches a deadlock situation.

The behavior in case of an abnormal termination is the same as in the failure case. This is shown in the Rule. 6.29.

$$\frac{\Gamma \mapsto P_1 \rightarrow \lambda \quad \Gamma \mapsto P_2 \rightarrow \gamma}{\Gamma \mapsto (P_1||P_2) \rightarrow \otimes} \tag{6.29}$$

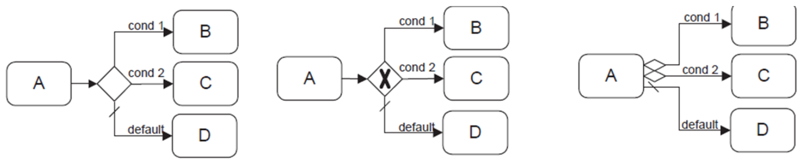


Figure 6.5: BPMN decision flows (source:BPMN Specification)

Where either $\gamma \; or \; \lambda \; is \; \otimes$

Business case The Rule. 6.30 shows the business case situation. The business cost is the sum of the business cost of the two patterns. The reliability is the product of the reliabilities of the patterns.

$$\frac{\Gamma \mapsto P_1 \xrightarrow{C_1, R_1} \odot \qquad \Gamma \mapsto P_2 \xrightarrow{C_2, R_2} \odot}{\Gamma \mapsto (P_1 || P_2) \xrightarrow{\sum_{i=1}^{2} BusinessCost(P_i), \prod_{i=1}^{2} R_i} \odot} \tag{6.30}$$

6.9 Decision/Gateway patterns

We extend the semantics for gateway conditions with the symbol Δ .

$$(Pattern) \; P \;\; ::= \;\; P \; \Delta \; P \tag{6.31}$$

Decision situations are captured in BPMN with the help of gateways. Possible combinations are as shown in the Fig. 6.5.

Gateways allow for the selection of one of the flows from a list of possible flows depending upon a condition. So as to express the behavior of the gateways we consider a pattern P_1 which completes and encounters a gateway which in turn selects one of the patterns ranged between P_{21} to P_{2n}. Rule. 6.32 is the case of a successful commit. This case comes up when the selected pattern from the gateway commits successfully.

$$\frac{\Gamma \mapsto P_{2i} \xrightarrow{C_{2i}, R_{2i}} \odot}{\Gamma \mapsto P_1 \to \odot, \Gamma \mapsto (\Delta P_{21} \Delta P_{22} ... \Delta P_{2n}) \xrightarrow{\sum_{i=1}^{n} C_i \times Probability_i, \sum_{i=1}^{n} R_i \times Probability_i} \odot} \tag{6.32}$$

In an unsuccessful case the pattern P_1 commits after which through the decision gateway the pattern P_{21} is executed. Pattern P_{21} fails and triggers an error flow situation. If the error flow commits successfully then the process terminates with an error flow else the process terminates abnormally. This rule is shown Rule. 6.33.

$$\frac{\Gamma \mapsto P_{2i} \to \oplus \quad \Gamma \mapsto P_{2iErrorFlow} \to \gamma}{\Gamma \mapsto P_1 \to \odot, \Gamma \mapsto (\Delta P_{21} \Delta P_{22} ... \Delta P_{2n}) \to \lambda} \tag{6.33}$$

Where
$\lambda\ is\ \oplus\ if\ \gamma\ \odot$
$\lambda\ is\ \otimes\ otherwise$

Business case The business case is captured in the Rule. 6.34. 'Probabilityi' is the probability that pattern i will get selected as a decision at the gateway. The overall business cost is the summation of the weighted business cost of each pattern according to the probability of its selection. The reliability is the product of the weighted reliability of each pattern according to the probability of its selection.

$$\frac{\Gamma \mapsto P_{2i} \xrightarrow{C_{2i}, R_{2i}} \odot}{\Gamma \mapsto P_1 \to \odot, \Gamma \mapsto (\Delta P_{21} \Delta P_{22} ... \Delta P_{2n}) \xrightarrow{\sum_{i=1}^{n} BusinessCost(P_i) \times Probability_i, \sum_{i=1}^{n} R_i \times Probability_i} \odot} \tag{6.34}$$

6.10 Compensation

The Saga workflow model relies on the concept of breaking long running transactions into smaller pieces which interleave. So as to achieve consistency when something goes wrong with one or more of the interleaving transaction a compensating transaction is executed. When the need for compensation arises, the compensating activities are triggered in a sequential manner.

So as to achieve the same effect as in Saga, BPMN allows for a compensation event sub-process. The compensation event sub-process allows for a process definition which is much more than a compensation procedure. It is a sub-process and hence can have many more tasks and activities embedded within the sub-process. The handling of compensation is the same as in the case of a sequential pattern situation . Hence the rules defined for the sequential situation also fits the compensation situation. The compensating pattern is triggered sequentially when a condition is true, or on a logical decision. We extend the semantics for pattern with ÷ for the compensating pattern.

$$(Pattern)\ P\ ::=\ P_1\ \div\ P_2 \tag{6.35}$$

Note that in the equation 6.35, the pattern P_2 is not the compensating pattern for P_1. Pattern P_2 is a compensation pattern which runs outside the normal flow of the business process. Pattern P_1 triggers the execution of pattern P_2 in a sequential manner. Hence the rules defined for the Sequential execution of patterns can be used here as well.

6.11 Conclusion

BPMN allows for flexibility in designing business process such that the technical and functional requirements can be represented at an operative level. The semantics defined in the chapter take as basis the mathematical approach to break a business process into patterns. The semantics allow for a generalized representation of behavior of the tasks, patterns and the interleaving combination. With these as basis the development of the financial parameters cost, reliability and business cost is calculated.

7

Application: 'Business Process Diagram Cost Analyser'

The aim of the application is to consider a business process as input and calculate the cost, reliability and business cost of the same. The scope of the application is:

1. Accept a business process diagram as input.
2. Parse through the diagram to
   - Evaluate each task for its cost and reliability
   - Generate the elementary patterns in the diagram.
   - Generate the composite patterns in the diagram.
3. Represent the parsed information in the form or tables and graphs.
4. Paint a sensitivity analysis at the level of a pattern.
5. Finally, calculates the cost, business cost and reliability of the diagram and represents it with the help of a devils quadrangle.

The implementation of the proposed methodology is done with the java language and uses the Eclipse platform. The application has two parts:

1. A business process modeling notation (BPMN) modeler.
2. A custom made java application.

7.1 BPMN modeler

The BPMN modeler is needed so as to put business process diagrams (BPD) together such that the elements in the BPD can be extended to have cost and reliability as properties. The application uses the Eclipse STP BPMN modeler. The modeler is an industry contribution, 'Intalio Inc' and is based on the Graphical modeling framework (GMF). GMF is based on the Eclipse modeling framework (EMF) which allows for code generation from data models. The STP BPMN modeler allows for creation of BPMN diagrams and generates corresponding BPEL or object models.

The user interface of the modeler is as shown in the Fig. 7.1.

7.1.1 Adaptation of the eclipse STP BPMN modeler

The Eclipse STP BPMN modeler allows for developing business process diagrams. The modeler offers the different shapes from the BPMN which can be used to put BPDs together (Fig. 7.2). For the BPD and for each element in the BPD, the modeler offers a vast set of properties which can be configured. These include properties for the display such as font, color etc. and properties which belong to the modeling framework. The properties are shown in the Fig. 7.3.

Creation a BPD in the modeler

A new BPD can be created in the modeler as a BPMN diagram under the context menu. For every diagram that is created through the modeler, the modeler generates two files:

1. The diagram File: The diagram file has the extension "bpmn$_{d}iagram$".$Thiscontainsallthevisualan$

The corresponding XML file looks like:

```
<children xmi:type="notation:Node" xmi:id="_iK4VgBdEEeCzXdzzLITmhQ" type="4001"/>

<styles xmi:type="notation:FontStyle"
```

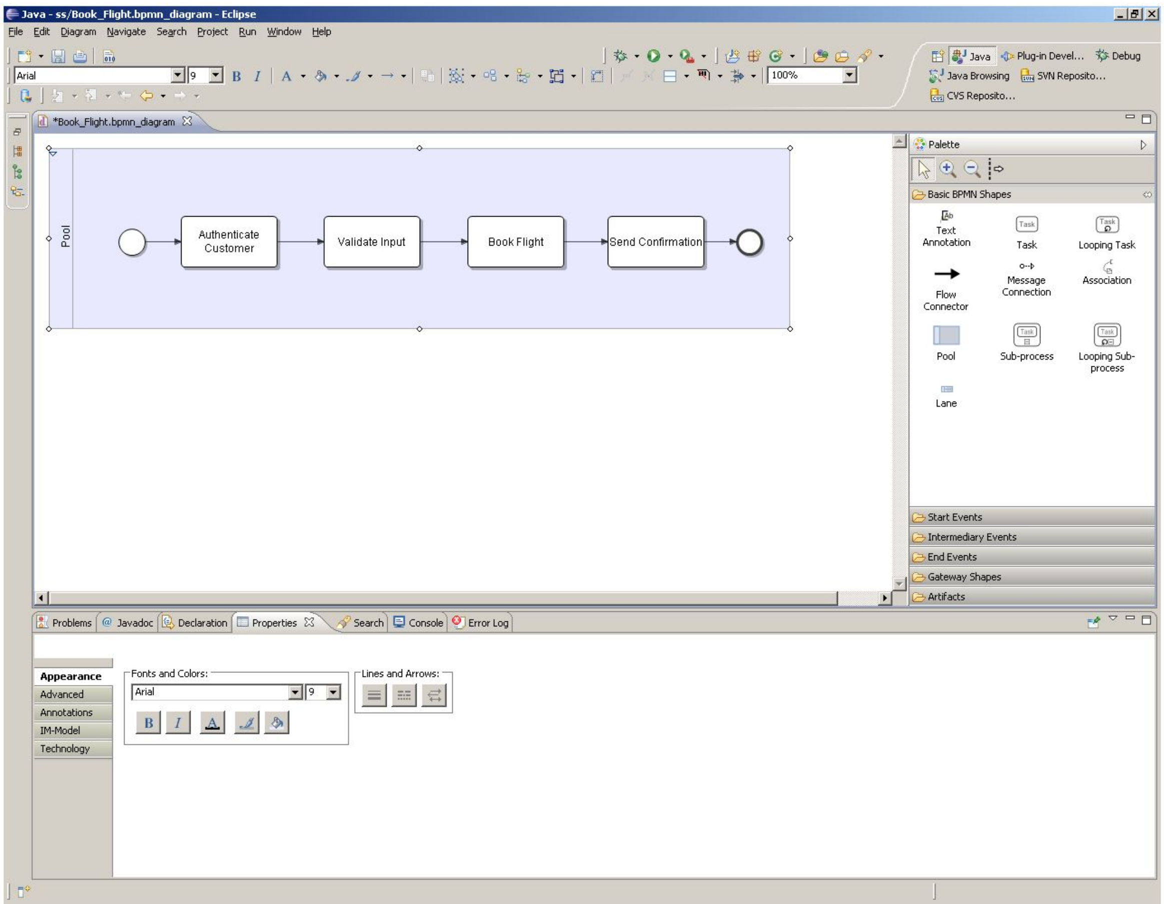


Figure 7.1: STP BPMN modeler user interface

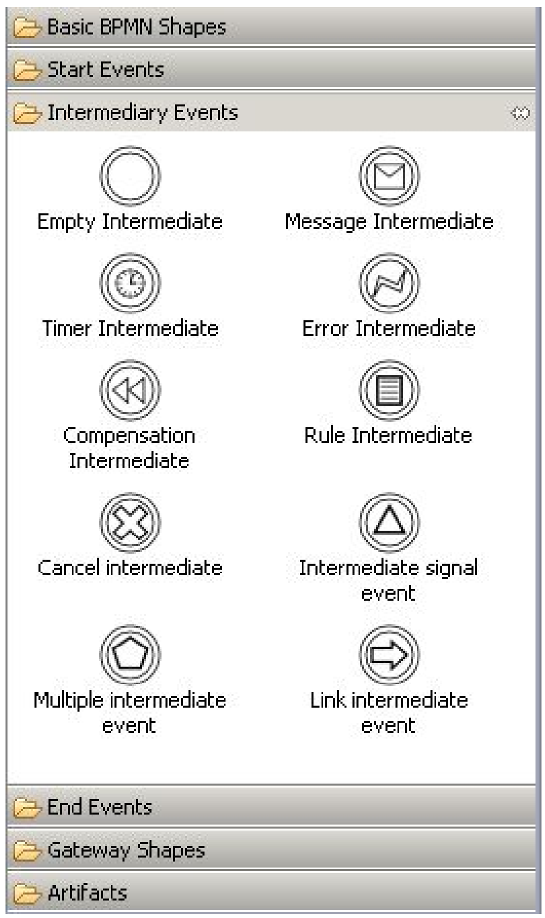


Figure 7.2: Shapes in the eclipse modeler

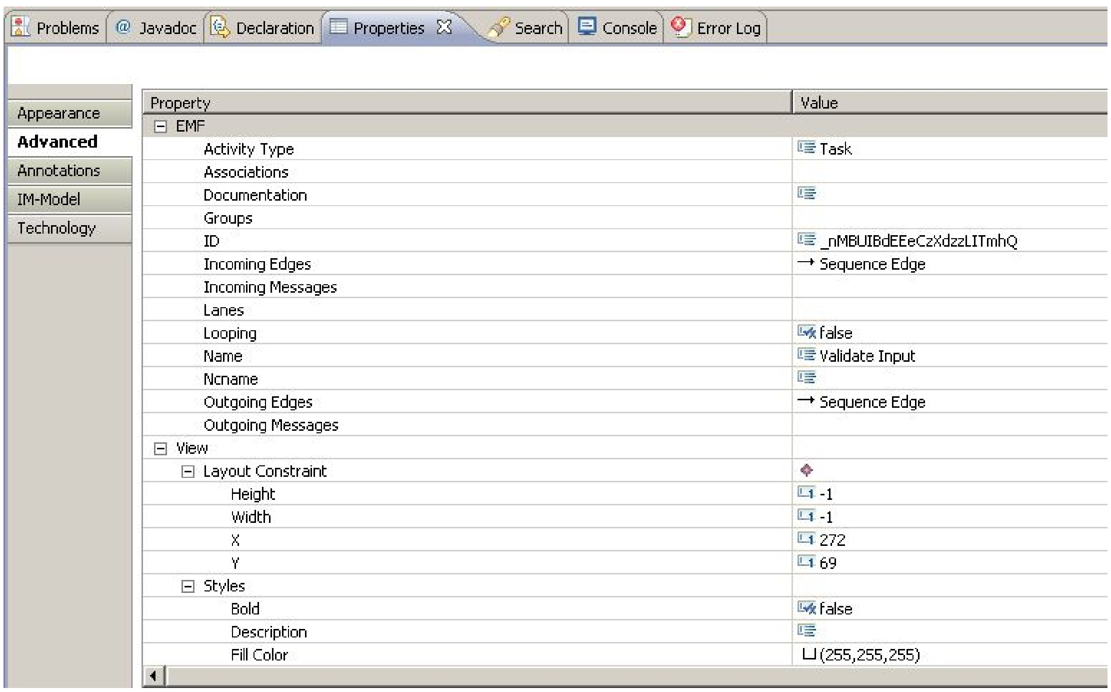


Figure 7.3: Properties in the eclipse modeler

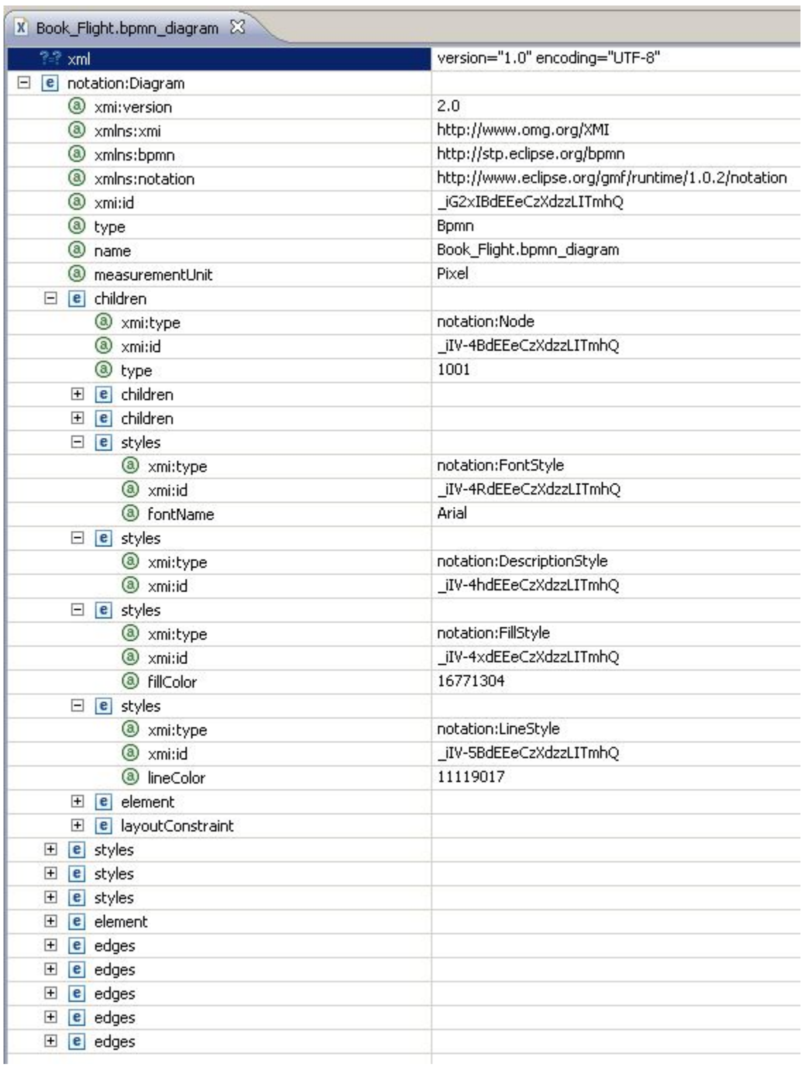


Figure 7.4: BPMN diagram file

```
xmi:id="_iKukgRdEEeCzXdzzLITmhQ"
fontName="Arial"/>

<styles xmi:type="notation:DescriptionStyle"
xmi:id="_iKukghdEEeCzXdzzLITmhQ"/>

<styles xmi:type="notation:FillStyle"
xmi:id="_iKukgxdEEeCzXdzzLITmhQ"/>

<styles xmi:type="notation:LineStyle"
xmi:id="_iKukhBdEEeCzXdzzLITmhQ"
lineColor="0"/>

<element xmi:type="bpmn:Activity"
href="Book_Flight.bpmn#_iHwJARdEEeCzXdzzLITmhQ"/>

<layoutConstraint xmi:type="notation:Bounds"
xmi:id="_iK4VgRdEEeCzXdzzLITmhQ"
x="114" y="69" width="111" height="61"/>

</children>
```

2. The BPMN file: the bpmn file is an xml file with the listing of the properties of all the elements of the diagram and the relationships. Each element is recognized by a unique id. The bpmn file shows the type of the element, its name, and the list of incoming and outgoing edges. The file also contains the list of edges and their respective ids. A sample file is shown in Fig. 7.5.

7.1.2 Adaptation of the modeler

The modeler needs to be adapted so that each element in the BPD can have properties for representing cost and reliability. This is done by adding these properties to the Eclipse modeling framework (EMF). The regeneration of code can then directly generate the properties required.

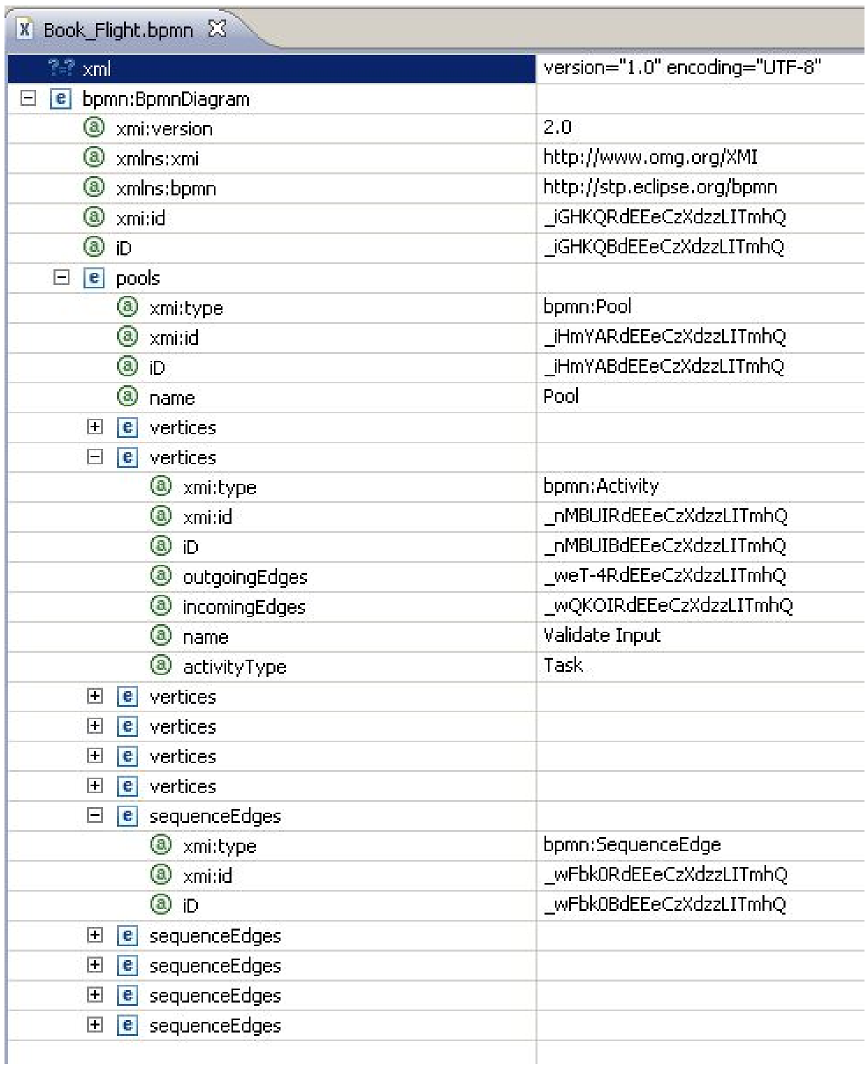


Figure 7.5: BPMN file

```
<eStructuralFeatures
        xsi:type="ecore:EAttribute"
        name="cost"
        eType="ecore:EDataType
        http://www.eclipse.org/emf/2003/XMLType#//Integer"/>

<eStructuralFeatures
        xsi:type="ecore:EAttribute"
        name="Reliability"
        eType="ecore:EDataType
        http://www.eclipse.org/emf/2003/XMLType#//Decimal"/>

<eStructuralFeatures
        xsi:type="ecore:EAttribute"
        name="Probability"
        eType="ecore:EDataType
        http://www.eclipse.org/emf/2003/XMLType#//Decimal"/>
```

Figure 7.6: Attribute extensions

This is done by adding attribute types to the date model of the STP BPMN modeler. In this case, the model details are captured in the ecore file. cost, reliability and probability are added as attributes to the DataTypes. The code generation will then add these attributes as part of the current objects. This is shown in the Fig. 7.6. Note that this is the standard method for adding or modifying attributes in a plugin development process on eclipse (this method is used by many involved in this field).

The generated code for the BPMN modeler attaches the three properties to each element in the BPD. The modeler shows these as properties in the advanced EMF properties. This is shown in the Fig. 7.7.

The generated BPMN file will add the properties to the elements as well.

```
<vertices xmi:type="bpmn:Activity"
xmi:id="_bFd046hdEd-_l4NHyePM0w"
```

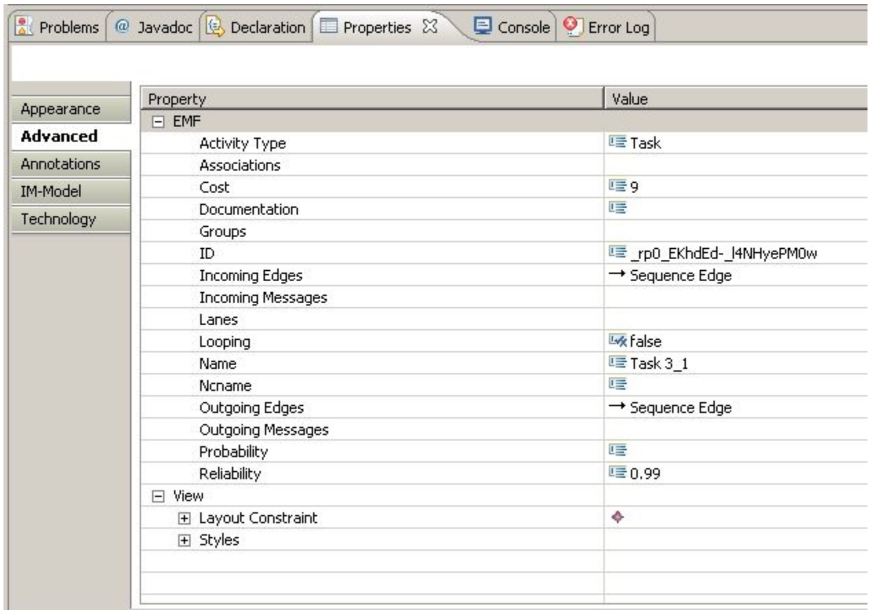


Figure 7.7: Eclipse modeler with cost and reliability as properties

```
iD="_bFd04qhdEd-_l4NHyePM0w"
outgoingEdges="_2gXq0ahdEd-_l4NHyePM0w _2_q04ahdEd-_l4NHyePM0w"
incomingEdges="_i7CC8ahdEd-_l4NHyePM0w"
name="Task 1"
activityType="Task"
cost="5"
Reliability="0.8"/>

<vertices xmi:type="bpmn:Activity"
xmi:id="_eo2cEahdEd-_l4NHyePM0w"
iD="_eo2cEKhdEd-_l4NHyePM0w"
outgoingEdges="_i7CC8ahdEd-_l4NHyePM0w"
name="Start"
activityType="EventStartEmpty"
cost="1"
Reliability="1"/>
```

```
<vertices xmi:type="bpmn:Activity"
xmi:id="_kPhwsahdEd-_l4NHyePM0w"
iD="_kPhwsKhdEd-_l4NHyePM0w"
outgoingEdges="_3-_k8ahdEd-_l4NHyePM0w"
incomingEdges="_2gXq0ahdEd-_l4NHyePM0w"
name="Task 2_1"
cost="8"
Reliability="0.9"/>
```

7.2 Java application

7.2.1 Overview of the user interface

The application is built on the Java platform and takes the bpmn file from the BPMN modeler as input. The application is stand alone and starts with an initial dialog for file selection and is made up of four tabs:

7.2.2 Initial dialog

The initial dialog lets the user select a bpmn file as input. It contains the Analyse button which when pressed starts the analysis of the process diagram. A user interface is shown in Fig. 7.8

7.2.3 Summary tab

The summary tab displays the results of parsing the input bpmn file. It shows the calculated cost, business cost and the reliability of the complete business process diagram. The tab also shows the results in the form of a devils quadrangle thus graphically representing the variation between the cost and the business cost of the process due to reliability. The summary tab is as shown in the Fig. 7.9.

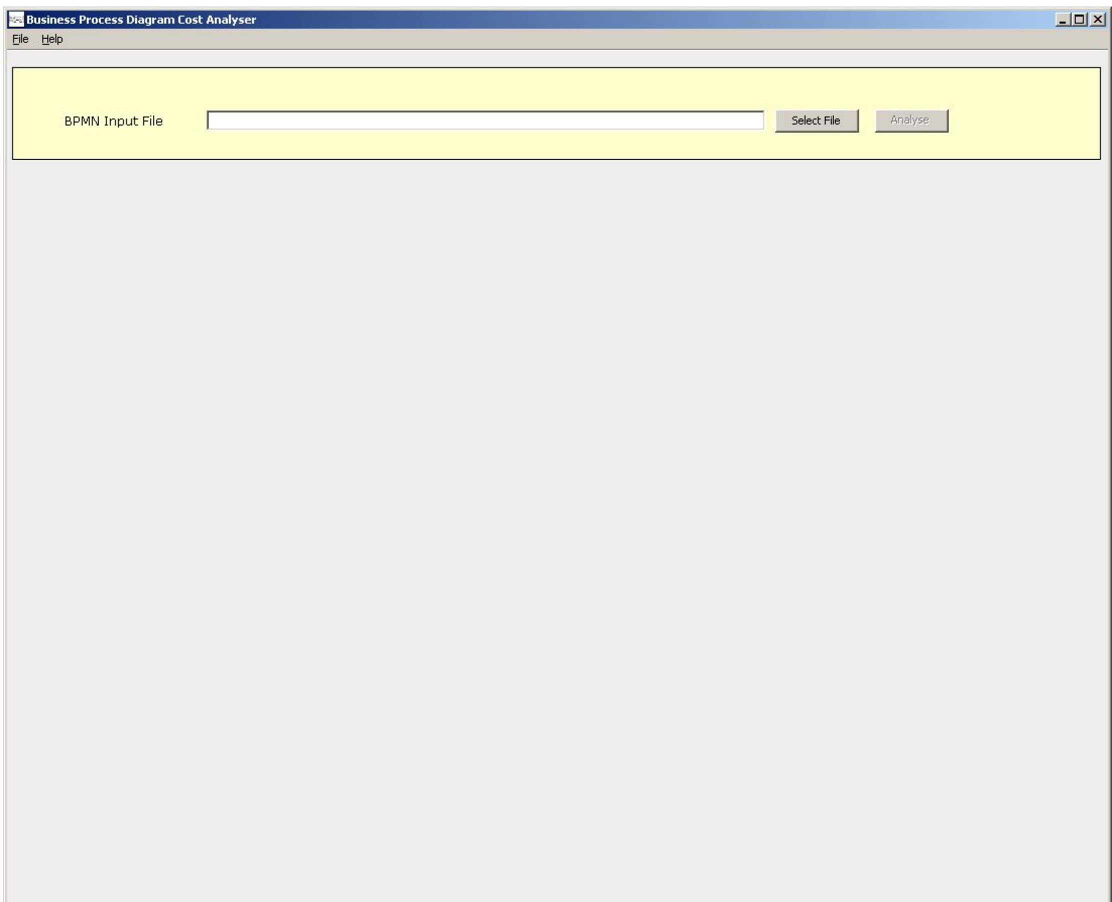


Figure 7.8: Initial dialog

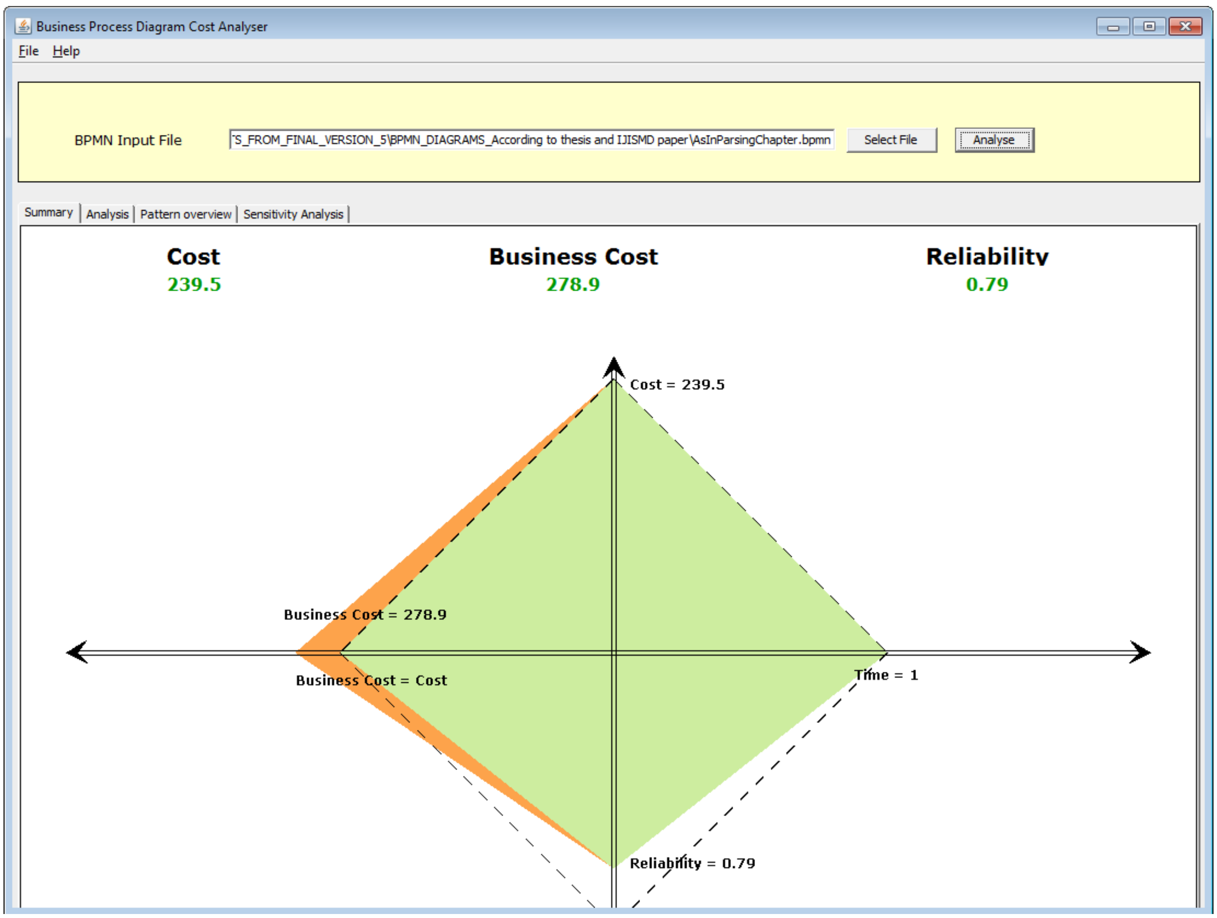


Figure 7.9: Summary tab

7.2.4 Analysis tab

The Analysis tab represents the result of the analysis done on the file. As the file is parsed, the important parameters such as number of elements, patterns and their relationships etc. are put together here. The analysis tab is as shown in the Fig. 7.10.

The tab is divided into four parts. The header of the tab shows the summary of the number and type of shapes found in the input BPMN file.

This is followed by a table with a list of all the elements in the BPD. The table shows the name of the element followed by its type, cost, reliability and resultant business cost of the element. This is as shown in the Fig. 7.11.

The list of patterns found in the BPD is shown right below the list of elements. The table lists each of the patterns with the starting and ending elements, the number of shapes in the pattern, and is followed by the parameters cost, reliability and business cost. The patterns are named in an ascending order starting from 0. This is as shown in the Fig. 7.12.

The central parsing logic is to find the patterns and the relationships between them. Patterns come together with other patterns to form composite patterns. They are either merging together or splitting or are sequential to each other etc. The table "Composite Patterns" lists the composite patterns found in the BPD file. Each of the patterns is listed with its incoming patterns and relationship and its outgoing patterns and relationships. This is as shown in the Fig. 7.13.

7.2.5 Pattern overview tab

The Pattern overview provides the details of each of the elementary patterns in the diagram in detail. For each elementary pattern found, the details of the pattern including its starting and ending tasks, the number of elements in the pattern, and the cost, business cost and reliability are displayed. A graphical representation showing the growth of cost, business cost and reliability at the level of a task and that of the complete pattern are displayed as well. The pattern overview tab is as shown in the Fig. 7.14.

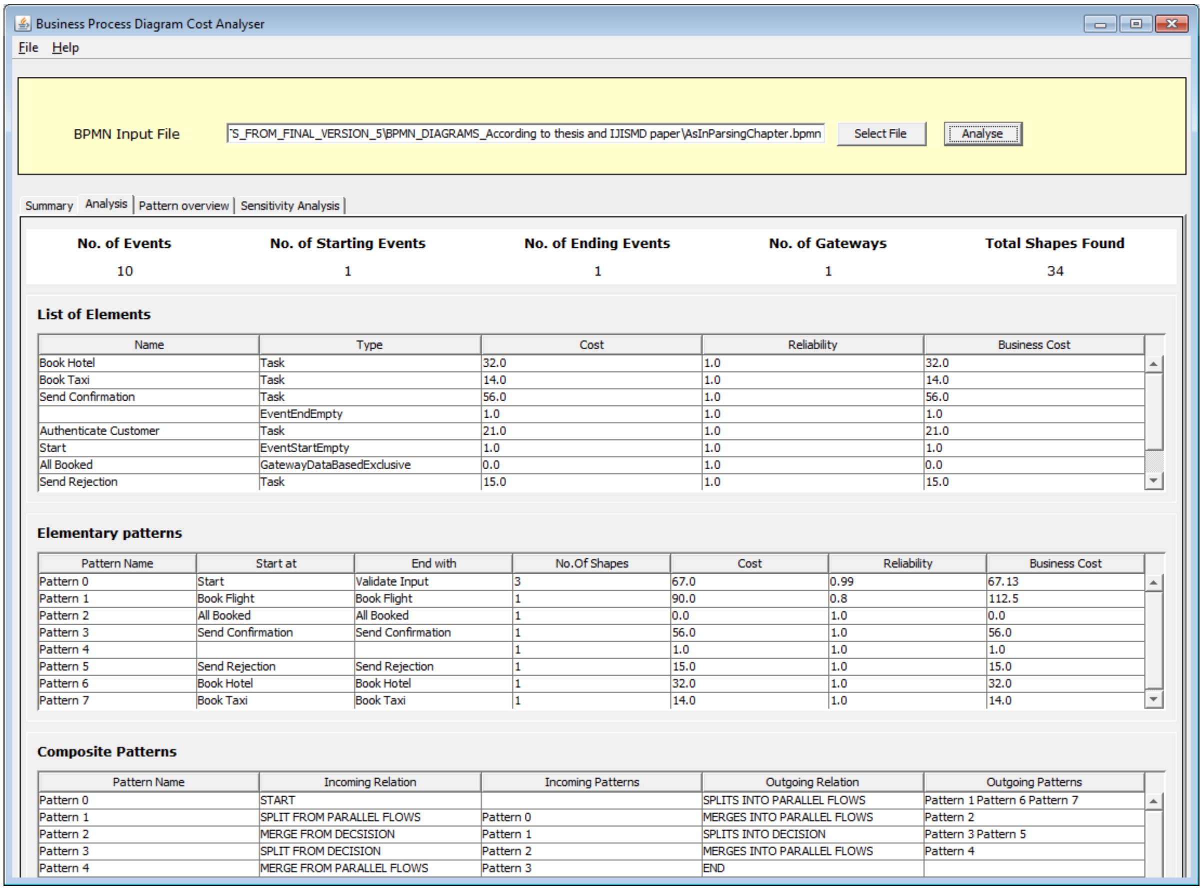
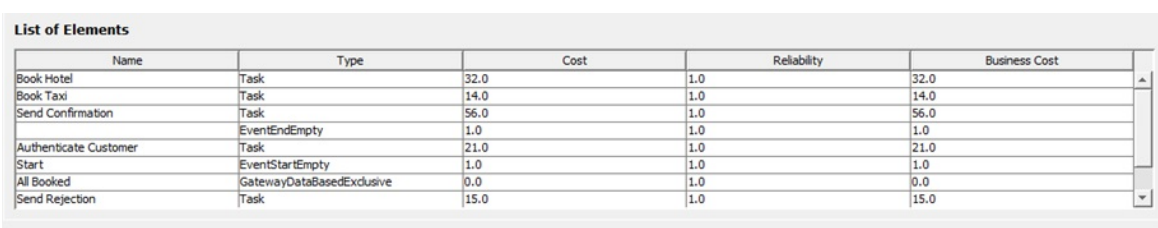


Figure 7.10: Analysis tab


List of Elements

Name	Type	Cost	Reliability	Business Cost
Book Hotel	Task	32.0	1.0	32.0
Book Taxi	Task	14.0	1.0	14.0
Send Confirmation	Task	56.0	1.0	56.0
	EventEndEmpty	1.0	1.0	1.0
Authenticate Customer	Task	21.0	1.0	21.0
Start	EventStartEmpty	1.0	1.0	1.0
All Booked	GatewayDataBasedExclusive	0.0	1.0	0.0
Send Rejection	Task	15.0	1.0	15.0

Figure 7.11: List of elements

Elementary patterns

Pattern Name	Start at	End with	No.Of Shapes	Cost	Reliability	Business Cost
Pattern 0	Start	Validate Input	3	67.0	0.99	67.13
Pattern 1	Book Flight	Book Flight	1	90.0	0.8	112.5
Pattern 2	All Booked	All Booked	1	0.0	1.0	0.0
Pattern 3	Send Confirmation	Send Confirmation	1	56.0	1.0	56.0
Pattern 4			1	1.0	1.0	1.0
Pattern 5	Send Rejection	Send Rejection	1	15.0	1.0	15.0
Pattern 6	Book Hotel	Book Hotel	1	32.0	1.0	32.0
Pattern 7	Book Taxi	Book Taxi	1	14.0	1.0	14.0

Figure 7.12: Elementary patterns in the diagram

Composite Patterns

Pattern Name	Incoming Relation	Incoming Patterns	Outgoing Relation	Outgoing Patterns
Pattern 0	START		SPLITS INTO PARALLEL FLOWS	Pattern 1 Pattern 6 Pattern 7
Pattern 1	SPLIT FROM PARALLEL FLOWS	Pattern 0	MERGES INTO PARALLEL FLOWS	Pattern 2
Pattern 2	MERGE FROM DECSISION	Pattern 1	SPLITS INTO DECISION	Pattern 3 Pattern 5
Pattern 3	SPLIT FROM DECISION	Pattern 2	MERGES INTO PARALLEL FLOWS	Pattern 4
Pattern 4	MERGE FROM PARALLEL FLOWS	Pattern 3	END	

Figure 7.13: Composite patterns in the diagram

7.2.6 Sensitivity analysis tab

The sensitivity analysis tab considers each of the elementary patterns found in the diagram for a sensitivity analysis. For each elementary pattern found, the task whose cost to business cost has the maximum variation is selected for a sensitivity analysis. The reliability of this task is varied from 0.1 to 1.0 and in each case the development of the business cost for the whole pattern is calculated. The result is displayed in a graph. The sensitivity analysis tab is as shown in the Fig. 7.15.

7.3 BPMN-file analysis and pattern generation

The core of the application parses the input file and generates the required patterns and calculation of the same. These are done as below:

7.3.1 Algorithm

A BPD is a result of putting the artifacts from BPMN together to achieve a business process. The mathematical approach through which a BPD can be broken into repet-

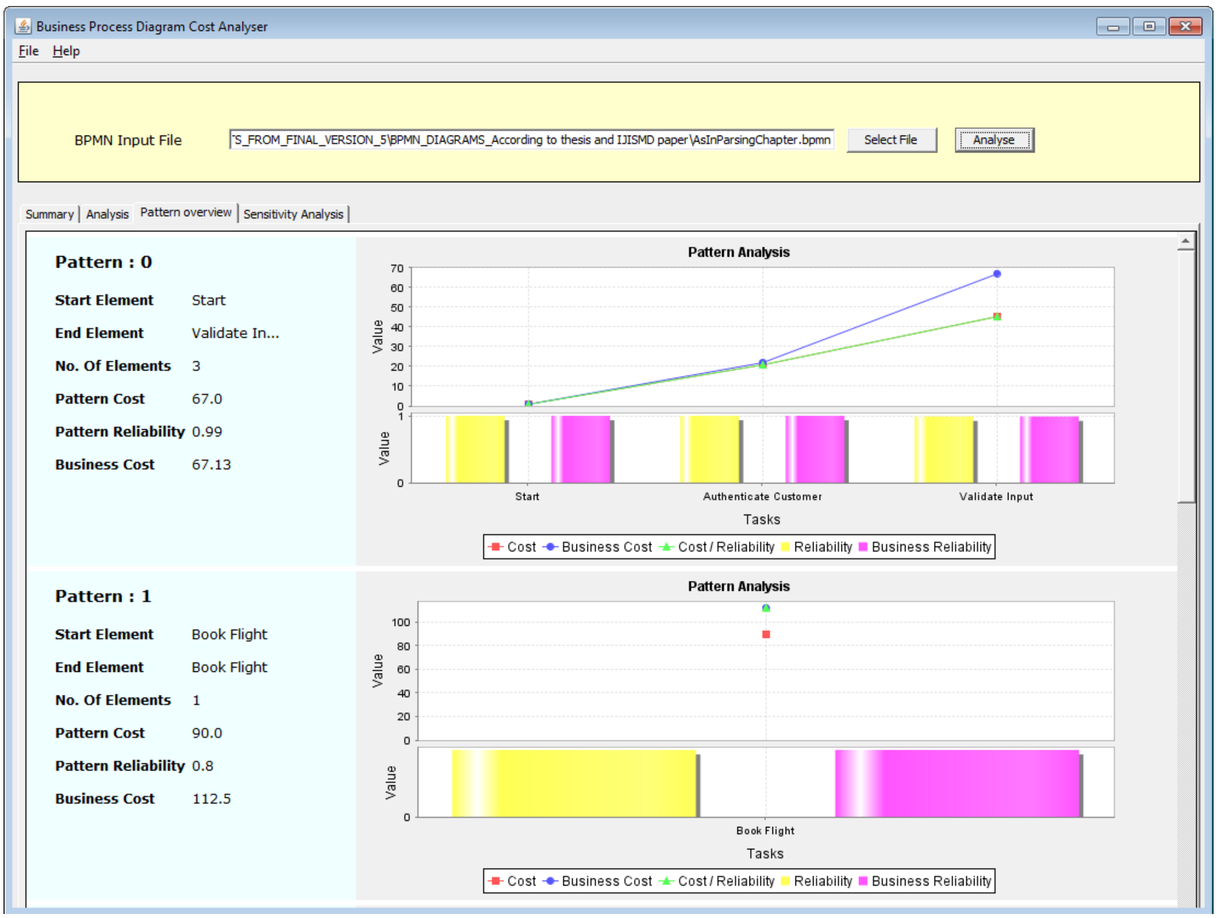


Figure 7.14: Pattern overview tab

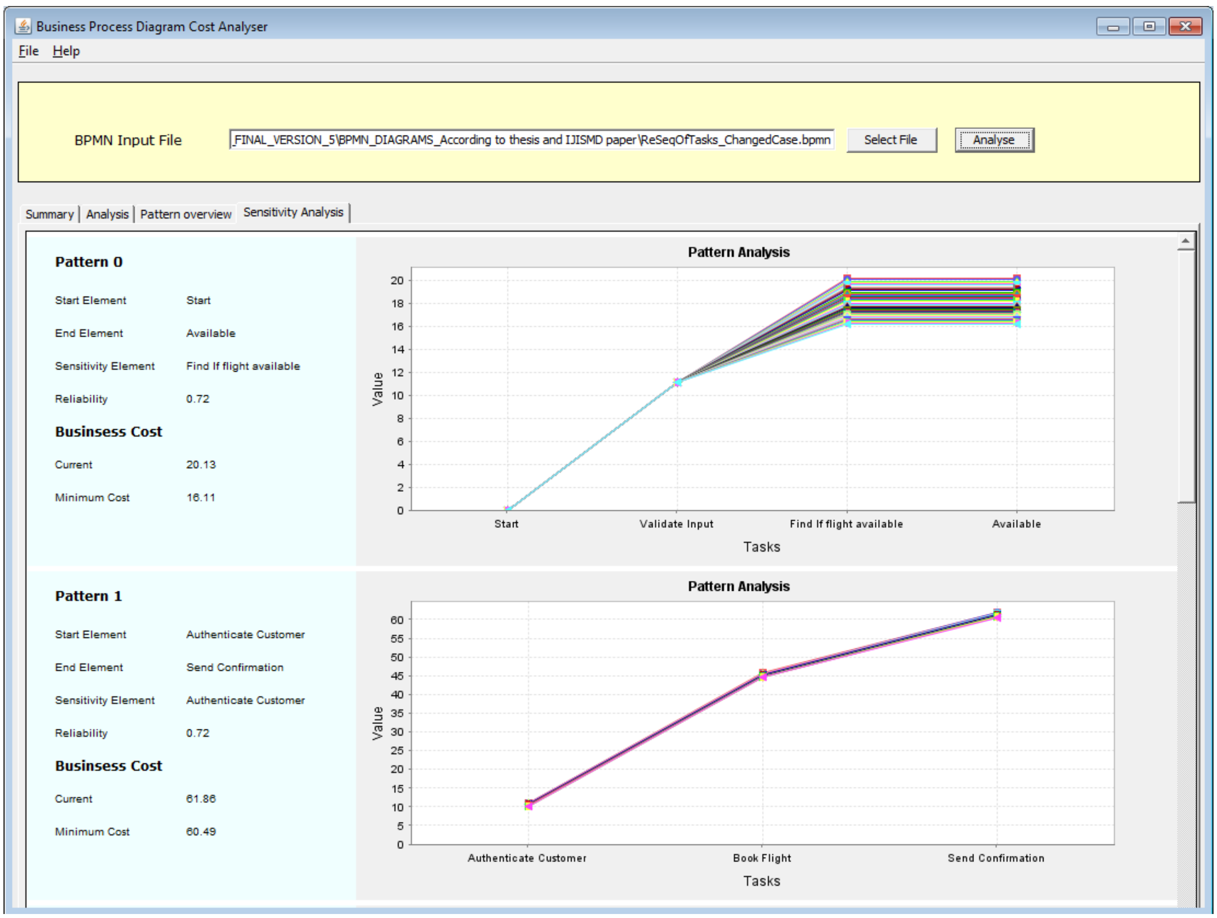


Figure 7.15: Sensitivity analysis tab

Figure 7.16: Breaking a BPD into Patterns

itive patterns has already been defined. Each pattern is made up of artifacts from BPMN which is atomic in nature and executes in a sequential order. The patterns interleave with one another to represent the business process as a whole.

Fig. 7.16 shows a sample BPD. The BPD shows the patterns within it (grouped by boxes).

A BPD is a graphical structure. It allows for conditions such as having more than one start event i.e. more than one root and looping conditions etc. So as to convert a BPD into a representation which represents the patterns and its interleaving relationships between them, the BPD needs to be parsed for every artifact in the process flow. The artifacts parsed needs to be grouped together as patterns and the relationship between the patterns need to be established.

The patterns are then merged with other patterns depending on the relationship between them. With each merge a new pattern is formed which takes the properties of its merged patterns. This is done until there is only one pattern left. This financial parameters of this pattern are the parameters for the BPD.

The methods in the algorithm is as shown in 7.17 7.18 7.19 7.20

Parsing the file

The bpmn file is an xml file and arranges the information according to "Vertices". For each element in the BPD there exists a Vertices entry in the xml file with all the required information. These include the type of the element, its cost, reliability, Incoming and outgoing edges etc. The xml file is parsed by the ReadXMLFile class. The class reads the file and populates the GraphElement objects.

```
<vertices xmi:type="bpmn:Activity"
xmi:id="_vXzFoahdEd-_l4NHyePM0w"
iD="_vXzFoKhdEd-_l4NHyePM0w"
outgoingEdges="_7pPpIahdEd-_l4NHyePM0w"
```

```
parseInputBPD(BPDfile input)
{

        Global graphElementList
        Global patternElementList

        for all input.Node do
                AddGraphElement(input.Node)

        Addpattern(sourceNode)

        CalculateFinancialParametersForBPD

}
```

Figure 7.17: Algorithm: method parseInputBPD

```
incomingEdges="_48-3MahdEd-_l4NHyePM0w _7D12AahdEd-_l4NHyePM0w"
name="Task 4"
activityType="Task"
cost="10"
Reliability="0.96"/>
```

7.3.2 Components

The class diagram of the application is as shown in the Fig. 7.21

```
AddGraphElement(sourceNode)
{
        Local graphElement
        sourceNode.visited ← true

        graphElement ← sourceNode
        graphElement.nodeProperties ← sourceNode.nodeProperties
        graphElement.incomingEdges ← sourceNode.incomingEdges
        graphElement.outgoingEdges ← sourceNode.outgoingEdges
        graphElement.calculateBusinessCost

        if graphElement.outgoingEdges.count > 1
                graphElement.nextRelation ← parallel
        else
        if graphElement.nextNode.type = gateway
                graphElement.nextRelation ← decision
        else
        if graphElement.incomingEdges.count > 1
                graphElement.prevRelation ← merge
        else
        if graphElement.type = intermediateEventCompensate
                graphElement.nextRelation ← compensate
        else
        if graphElement.type = intermediateEventErrorFlow
                graphElement.nextRelation ← errorFlow
         End if

        graphElementList.addnode(graphElement.id,graphElement)
}
```

Figure 7.18: Algorithm: method AddGraphElement

```
AddPattern(sourceNode)
{
        Local patternElement
        Local counter ← 0
        Local nextNodeCount ←1


        patternElement.addNode(counter) ← sourceNode
        counter ← counter + 1
        patternElement.previousRelation ← sourceNode

        while sourceNode.nextNodeCount = 1
                patternElement.addNode(counter) ← sourceNode
                counter ← counter + 1
                sourceNode ← sourceNode.nextNode

        patternElement.nextRelation ←sourceNode.nextRelation

        patternElement.calculateCost
        patternElement.calculateReliability
        patternElement.calculateBusinessCost


        patternElementList.add(patternElement)

}
```

Figure 7.19: Algorithm: method AddPattern

```
CalculateFinancialParametersForBPD()
{

Local counter ← 0
Local newPattern

for all patternElementList.size do
{

        currentPattern ← patternElementList.getpattern(counter)
        if currentPattern.nextRelationship = sequential
                newPattern ←mergePatternsSequentially(currentPattern, currentPattern.nextPattern)
                patternElementList.remove(currentPattern, currentPattern.nextPattern)

        if currentPattern.nextRelationship = parallel
                newPattern ←mergePatternsParallel(currentPattern.allNextPatterns)
                patternElementList.remove(currentPattern.allNextPatterns)
        end if

        if currentPattern.nextRelationship = error
                newPattern ←mergePatternsError(currentPattern.normalFlow,
        currentPattern.errorFlow)
                patternElementList.remove(currentPattern.allNextPatterns)
        end if

        if currentPattern.nextRelationship = decision
                newPattern ←mergePatternsDecision(currentPattern.allNextPatterns)
                patternElementList.remove(currentPattern.allNextPatterns)
        end if

        if currentPattern.nextRelationship =
                newPattern ←mergePatternsDecision(currentPattern.allNextPatterns)
                patternElementList.remove(currentPattern.allNextPatterns)
        end if

        patternElementList.add(newPattern)
}

If patternElementList.size > 1
CalculateFinancialParametersForBPD()

}
```

Figure 7.20: Algorithm: method CalculateFinancialParametersForBPD

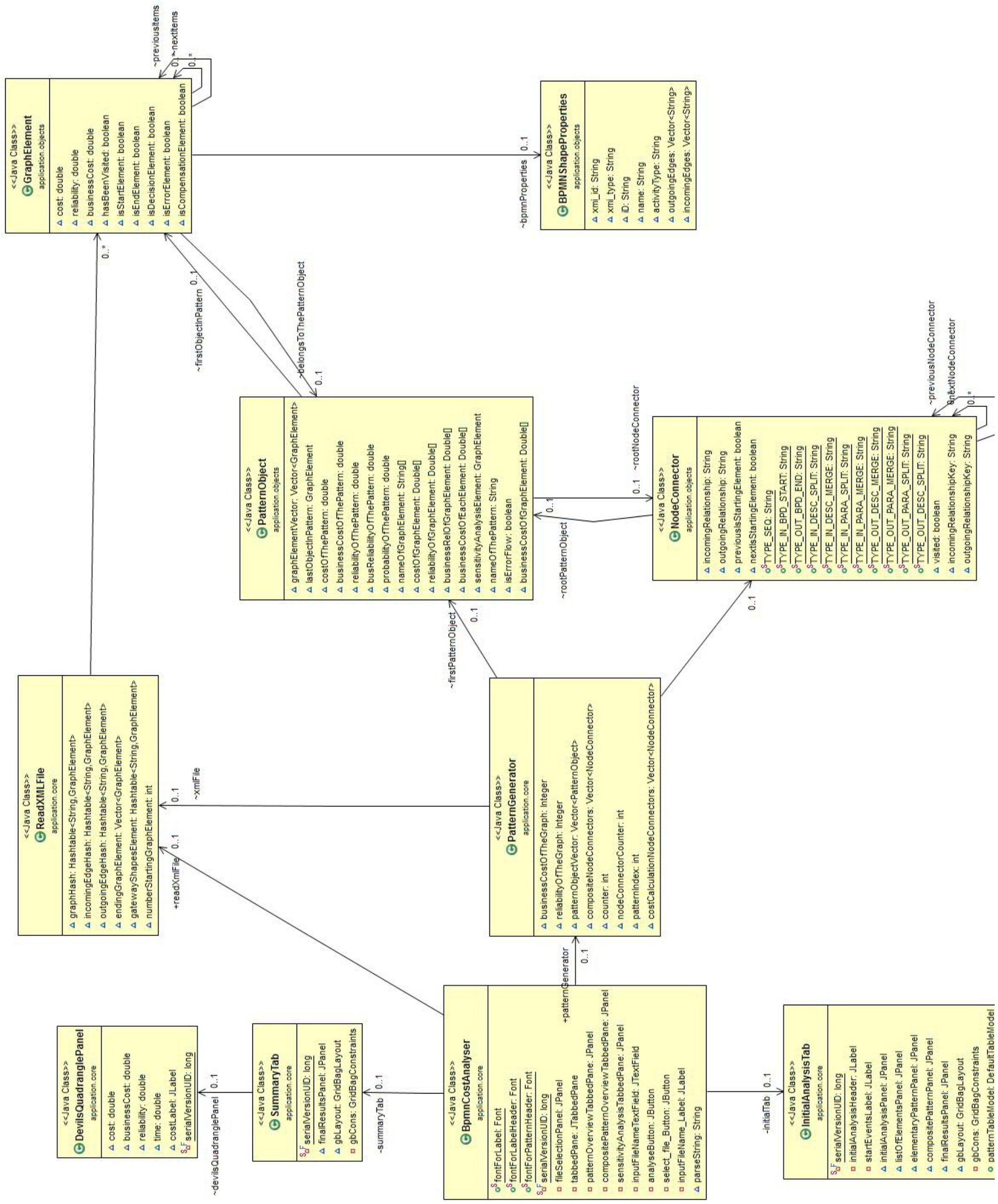


Figure 7.21: Class diagram

(a) BPMNCostAnalyser: The BPMNCostAnalyser is the parent class for the graphical interface of the application. The class combines all the child tabs and panels required for representing the results to the user.

(b) ReadXMLFile: This class is responsible for parsing the input file which is in XML format and grouping the properties in the form of objects which are related to each other. These relationships between the objects are then evaluated to form patterns.

(c) PatternGenerator: The PatternGenerator class is the central class of the application. It implements the algorithm to evaluate the relationship between the objects, form patterns from the objects, and calculate the cost, reliability and the business cost of the business process.

(d) GraphElement: Each instance of GraphElement represents one element of the business process diagram. These objects are generated as part of the xml parsing. The object holds all the details of each element in the diagram. The object contains details of the element for the calculation such as cost, reliability and business cost of the element. It also saves references to the element(s) which are previous to the current element and which come immediately next to the current element.

(e) BPMNShapeProperties: Each instance of GraphElement references one instance of the BPMNShapeProperties object. For each GraphElement, the BPMNShapeProperties saves the properties of the element which are related to the BPMN element. This contains details such as id, type of the element, name of the element etc.

(f) PatternObject: Each instance of the PatternObject represents an elementary pattern in the diagram. It refers to one or more GraphElements which fall together to form an elementary pattern. The PatternObject also has the cost, reliability and business cost of the pattern which is needed for the pattern. The PatternObject also has references to the NodeConnector object.

(g) NodeConnector: The NodeConnector object represents the relationship between the different patterns with each other. The object has a list of one or more input patterns, and lists for all outgoing patterns, with the relationship between them.

7.4 Conclusion

Applications for BPMN Modeling through which users can design business processes are many in the market. A few of these are freeware as well. However a modeler which can capture financial parameters and evaluate the profitability of the business process does not exist. So as to make this possible a modeler will have to first allow the capture of financial information for each artifact and this will then have to be parsed to generate the required results.

The application 'Business Process Diagram Cost Analyser' uses the Eclipse BPMN modeler to model business processes. The BPMN modeler is configured such that the cost and reliability for each artifact can be added as properties. The application uses business process diagrams and parses them with the pattern based methodology to calculate cost, reliability and business cost of the process.

The application evaluates the BPD by breaking them into patterns. The cost calculations are carried out for each pattern; the same is displayed in the user interface. The overall cost, business cost and reliability of the process are shown graphically as a devils quadrangle. The application also allows for an evaluation of the impact of each task on the cost of the patterns by implementing a Sensitivity Analysis.

The aim of the application is to allow the designer of the business process to check the financial impact of the process. This it does by working directly with the business process diagram that the designer constructed. The results generated by the application will also make the designer aware of the points where a decision to change the process design could lead to a benefit for the organization. It makes the impact of such decisions measureable by generating the financial parameters at the level of the artifact, the patterns and the overall process.

8

Examples: Business cost calculations from a business process diagram

In this chapter we consider two examples where we calculate the cost, reliability and business cost of the business process. We will implement the pattern based concept so as to do this.

8.1 Example: A payment process

We consider a business process whose objective is to accept a payment from a customer for a product that the customer has bought. This is a standard example from the BPMN organization. The business process is shown in the Fig. 8.1.

The business process is simple and has just four steps. The first step is to identify the payment method, then it either accepts a cash/check or processes the payment through a credit card. Once this is done it prepares the package for the customer.

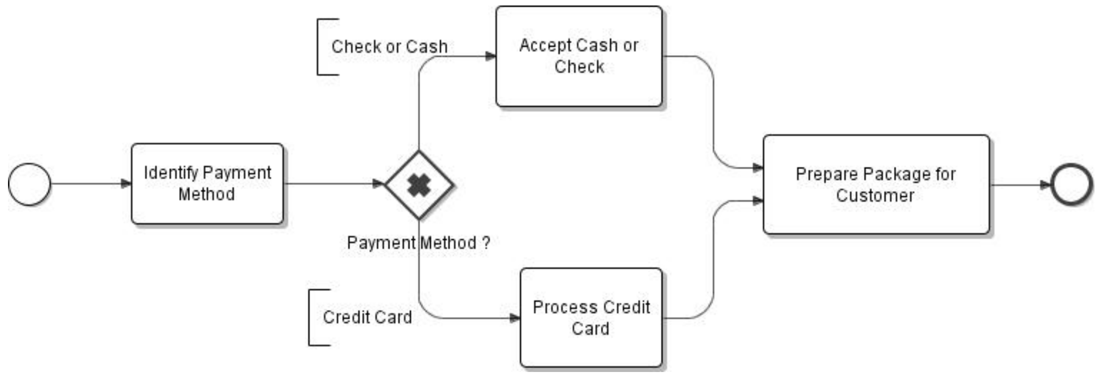


Figure 8.1: Business process for payment processing (source: BPMN specification)

Task	Cost	Reliability
Identify Payment Mode	10	1.0
Payment Method	1	1.0
Accept Cask or Check	12	0.9
Process Credit Card	25	0.8
Prepare Package for Customer	20	0.9

Table 8.1: Cost and reliability for all the tasks

8.1.1 Step 1: Identify cost and reliability:

We make assumptions on the cost and reliability of each task in this process. This is as shown in the table 8.1.

We have a decision box in "Payment Method" which identifies the payment mode, hence we make the assumption that the probability of the payment mode being cash or check is equal to the payment mode being a credit card. In other words the probability is 0.5 for each way out of the decision box.

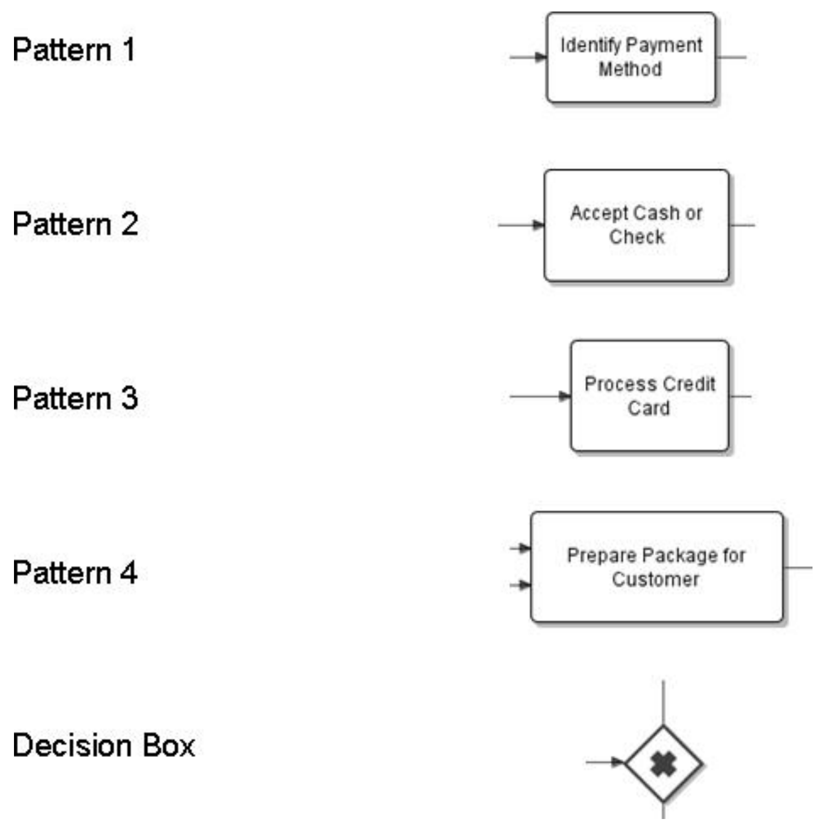


Figure 8.2: Patterns in the business process

8.1.2 Step 2: Divide the business process into patterns:

We now divide the business process into elementary patterns. Each pattern will contain one or more tasks in it. We start identifying the pattern from the start of the business process. The patterns identified are as shown in the Fig. 8.2.

Pattern name	Starting task	Ending task	Cost	Reliability	Business cost
Pattern 1	Identify Payment	Identify Payment	10	1.0	10
Pattern 2	Accept Cask or Check	Accept Cask or Check	12	0.9	13.33
Pattern 3	Process Credit Card	Process Credit Card	25	0.8	31.25
Pattern 4	Prepare Package	Prepare Package	20	0.9	22.22

Table 8.2: Cost, reliability, business cost of the patterns

Pattern name	Cost	Reliability	Business cost
Pattern 2 and Pattern 3	18.5	0.85	22.29

Table 8.3: Composite pattern

8.1.3 Step 3: Finding the cost, reliability and business cost of the patterns:

We calculate now for each elementary pattern the cost, reliability and business cost. The same is as shown in the table 8.2.

8.1.4 Step 4: Putting the composite patterns together:

We now put the simple patterns together into composite patterns. The patterns 2 and 3 are combined through a decision gateway. We assumed that each has a probability of 0.5. Hence the value of the composite pattern is as shown in the table 8.3.

We are now left with three patterns which are in a sequential order i.e. pattern 1; pattern 2 and pattern 3; pattern 4. This is the same as a sequential pattern with three patterns involved. The cost, reliability and business cost for this business process is as shown in the table 8.4.

Factor	Value
Cost	48.5
Reliability	0.765
Business cost	60.06

Table 8.4: Cost, reliability, business cost of the process

8.2 Example: Automobile order management process

The automobile industry in Germany belongs to the best in the world. They are known for their high quality and well recognized products worldwide. The industry has a very high maturity and has standardized processes for all domains. The generalized automobile domain model is as shown in the Fig. 8.3.

in this section we consider the order management process implemented by a well-known automobile manufacturer from Germany. The business process is executed at the sales end of the manufacturer i.e. at the showrooms by the dealer. The process begins when a customer is interested in a vehicle. The ordering process leads to the transaction, books the order at the plant for manufacture, takes care of invoicing and delivery communication to the customer. We first look at the background of the business before stepping into the business process implemented.

The organization in involved in manufacture of light and heavy vehicles internationally. In Germany, the organization manufactures cars in four plants. Each plant can produce the same car. These plants are called G1, G2, G3 and G4. The central distribution is nevertheless done from the plant G4. The organization, due to the high demand for its car and stiff deadlines that the customer demands, does not wait till an order is placed so as to manufacture a car. The organization always has a standard stock of cars in particular models already manufactured. When a customer is interested in a car, the dealer is first expected to look into the stock for cars which have already been manufactured. The dealer, in case a car is found in the stock which meets the customer demands to a large extent, proposes the stock car to the customer first. The customer is also informed of any deviations that the stock car might have to the initial demands of the customer.

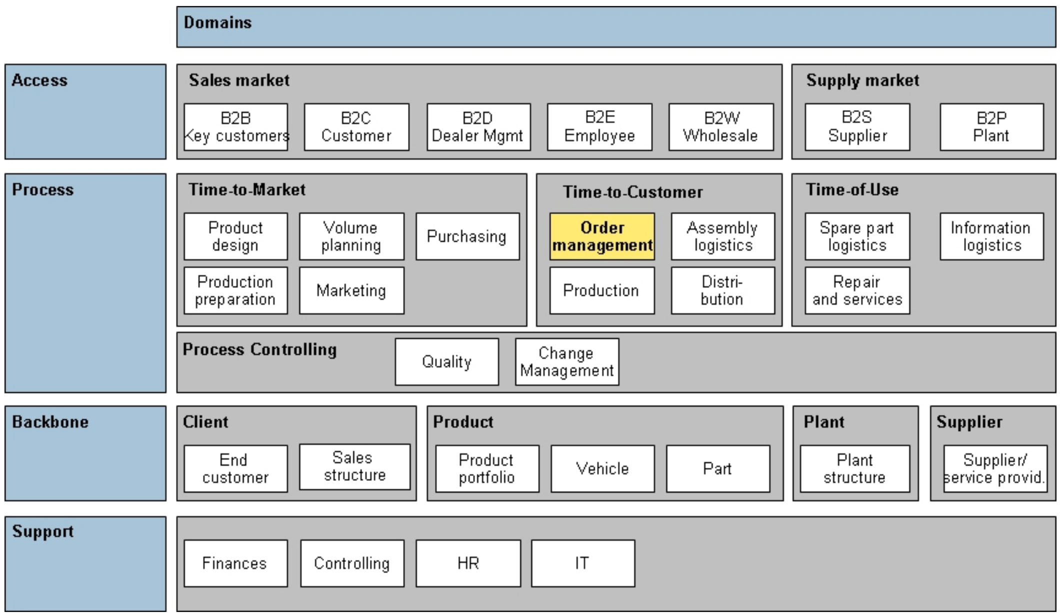


Figure 8.3: Generalized automobile domain model

Only in case there is no car in stock that meets the preferences of the customer or if the customer is not interested in the stock car because of the deviations that it might have, will the dealer go into configuring a car according to the customers wishes. This car then needs to be "Built To Order". The processes distinguish itself independently of the kind of vehicle. Reasons to propose the stock car first are many, mainly:

- The stock cars are readily available and can be delivered immediately.
- The longer they stay with the organization, the older they get. Hence value depreciates.
- It also costs money to stack them in the organization.
- A new car which needs to be built according to the customers wishes will take time to build. Hence customer has to wait.
- Etc.

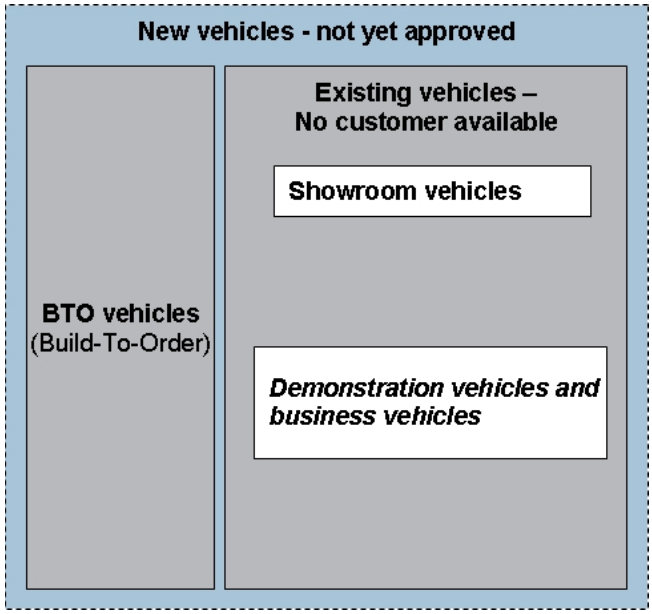


Figure 8.4: New vehicle classification

Planning for the production of cars is done by the headquarter of the organization on a monthly basis. At the same time certain codes (features inside the car which a customer wishes) can be produced only for up to a certain quantity each month in a plant. If an allocation is used up, the respective model or the code can no longer be produced in the month (and the production as well as the delivery date must be postponed to a later date). The volume planning influences the origins of the order and/or the date of the order completion. This volume planning is done for all the 4 plants in Germany.

When a customer wants a configured car which is built to order, the dealer needs to configure the car according to the customers wishes, look for the earliest delivery dates from the 4 plants, and if the customer agrees to these dates the dealer has to block the production capacity for the same.

The complete business process to order a car and book it through the ordering system is as shown in Fig. 8.5. The process covers all the peculiarities as already discussed above. In this section we calculate the cost, reliability and business cost of this process.

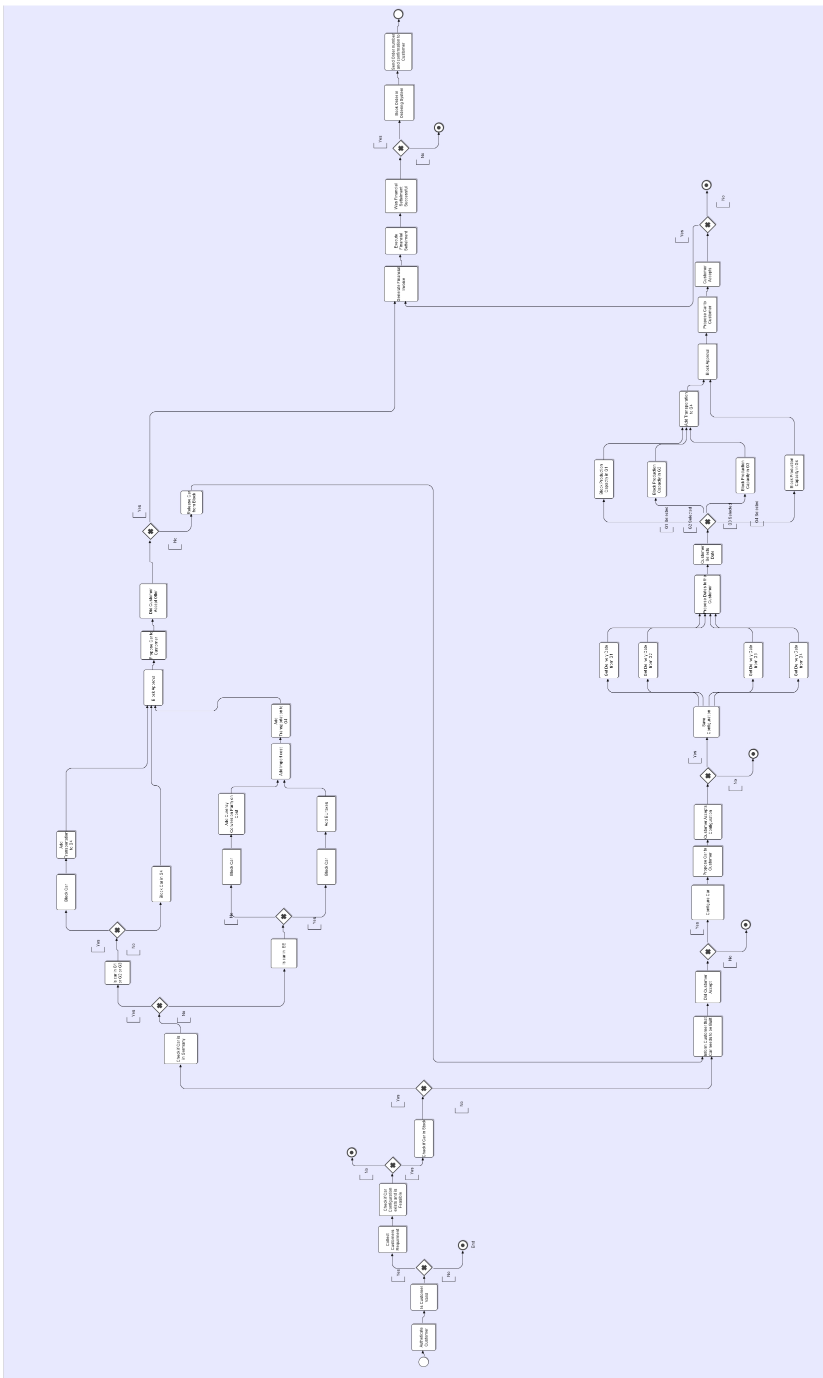


Figure 8.5: Order processing process

8.2.1 Elementary patterns

The Fig. 8.5 shows the complete business process of capturing an order for a car. We now break this into elementary patterns. We name the patterns as pattern - 1, pattern - 2, ... , pattern - n.

When broken into patterns we have found in total 35 elementary patterns as shown in Fig. 8.6. Each of the tasks in the elementary patterns comes with cost and reliability attached. In the table we now calculate the cost, business cost and reliability for all the patterns.

Task	**Cost**	**Reliability**
Pattern 1		
Authenticate Customer	10	1
Is Customer Valid	5	0.995
Business cost	**20.101**	
Reliability	**0.995**	
Pattern 2		
Collect Customer Requirements	10	1
Check if Car Configuration is Valid and Feasible	25	0.98
Business cost	**35,714**	
Reliability	**0,980**	
Pattern 3		
Inform Customer	5	1
Business cost	**5**	
Reliability	**1**	
Pattern 4		
Inform Customer	5	1
Business cost	**5**	
Reliability	**1**	
Pattern 5		
Check if Car in Stock	20	1
Business cost	**20**	
Reliability	**1**	
Continued on next page		

Table 8.5 – continued from previous page

Task	**Cost**	**Reliability**
	Pattern 6	
Check if Car is in Germany	20	1
Business cost	**20**	
Reliability	**1**	
	Pattern 7	
Is Car in G1 or G2 or G3	10	1
Business cost	**10**	
Reliability	**1**	
	Pattern 8	
Block Car	20	0.995
Add Transportation to G4	25	0.995
Business cost	**45.327**	
Reliability	**0.990**	
	Pattern 9	
Block Car in G4	20	0.995
Business cost	**20.101**	
Reliability	**0.995**	
	Pattern 10	
Block Approval	20	0.995
Propose Car to Customer	20	1
Did Customer Accept Offer	20	0.900
Business cost	**66.778**	
Reliability	**0.896**	
	Pattern 11	
Is Car in EE	10	1
Business cost	**10**	
Reliability	**1**	
	Pattern 12	
lock Car	20	0.995
Add Currency Conversion Parity on cost	25	0.995
Business cost	**45.327**	
Reliability	**0.990**	
	Pattern 13	

Continued on next page

Table 8.5 – continued from previous page

Task	**Cost**	**Reliability**
Block Car	20	0.995
Add EU taxes	10	1
Business cost	**30.101**	
Reliability	**0.995**	
Pattern 14		
Add Import cost	10	0.900
Add Transportation to G4	15	1
Business cost	**26.111**	
Reliability	**0.900**	
Pattern 15		
Release Car from Block	10	1
Business cost	**10**	
Reliability	**1**	
Pattern 16		
Inform Customer that Car Needs to be Built	10	0.995
Did Customer Accept	15	0.900
Business cost	**27.834**	
Reliability	**0.896**	
Pattern 17		
Configure Car	20	0.995
Propose Car to Customer	20	1
Customer Accepts Configuration	20	0.9
Business cost	**66.778**	
Reliability	**0.896**	
Pattern 18		
Save Customer Data as Potential Customer	10	1
Business cost	**10**	
Reliability	**1**	
Pattern 19		
Save Configuration	10	1
Business cost	**10**	
Reliability	**1**	
Pattern 20		

Continued on next page

Table 8.5 – continued from previous page

Task	**Cost**	**Reliability**
Save Customer Data as Potential Customer	10	1
Business cost **Reliability**	**10** **1**	
Pattern 20		
Get Delivery Date from G1	15	0.995
Business cost **Reliability**	**15.075** **0.995**	
Pattern 22		
Get Delivery Date from G2	15	0.995
Business cost **Reliability**	**15.075** **0.995**	
Pattern 23		
Get Delivery Date from G3	15	0.995
Business cost **Reliability**	**15.075** **0.995**	
Pattern 24		
Get Delivery Date from G4	15	0.995
Business cost **Reliability**	**15.075** **0.995**	
Pattern 25		
Propose Dates to the Customer Customer Selects Date	10 15	1 0.900
Business cost **Reliability**	**27.778** **0.900**	
Pattern 26		
Block Production Capacity in G1	15	0.995
Business cost **Reliability**	**15.075** **0.995**	
Pattern 27		
Block Production Capacity in G2	15	0.995
Business cost **Reliability**	**15.075** **0.995**	
Pattern 28		

Continued on next page

Table 8.5 – continued from previous page

Task	Cost	Reliability
Block Production Capacity in G3	15	0.995
Business cost	**15.075**	
Reliability	**0.995**	
Pattern 29		
Block Production Capacity in G4	15	0.995
Business cost	**15.075**	
Reliability	**0.995**	
Pattern 30		
Add Transportation to G4	10	1
Business cost	**10**	
Reliability	**1**	
Pattern 31		
Block Approval	10	1
Propose Car to Customer	15	1
Customer Accepts	15	0.900
Business cost	**44.444**	
Reliability	**0.900**	
Pattern 32		
Save Customer Data as Potential Customer	10	1
Business cost	**10**	
Reliability	**1**	
Pattern 33		
Generate Financial Invoice	20	1
Execute Financial Settlement	10	1
Was Financial Settlement Successful	5	0.900
Business cost	**38.889**	
Reliability	**0.900**	
Pattern 34		
Book Order in Ordering System	10	0.995
Send Order No. and Confirmation to Customer	10	1
Business cost	**20.050**	
Reliability	**0.995**	
Pattern 34		

Continued on next page

Table 8.5 – continued from previous page

Task	Cost	Reliability
Inform Customer, Save Order	10	0.995
Business cost **Reliability**	**10.050** **0.995**	

Table 8.5: Elementary patterns found in the business process

8.2.2 Composite patterns

We recursively evaluate the relationships between these 35 elementary patterns. With the elementary patterns as basis we find the composite patterns in this business process. Composite patterns have more than one elementary patterns coming together. We make and assumption that each flow out of every decision gate has an equal probability of execution. The composite patterns are as shown from Fig. 8.7 to Fig. 8.14. The corresponding calculations for cost, reliability and business cost are shown in the table.

Pattern	Cost	Reliability	Business cost
Composite pattern 1			
Pattern 8 Pattern 9	45.000 20.000	0.990 0.995	45.327 20.101
Relationship - Decision flows	**32.500**	**0.993**	**32.714**
Composite pattern 2			
Pattern 12 Pattern 13	45.000 30.000	0.990 0.995	45.327 30.101
Relationship - Decision flows	**37.500**	**0.993**	**37.714**
Composite pattern 3			
Composite pattern 1 Pattern 7	32.500 10.000	0.993 1.000	32.714 10.000

Continued on next page

Table 8.6 – continued from previous page

Task	Cost	Reliability	Business cost
Relationship - Sequential patterns	**42.500**	**0.993**	**42.714**
Composite pattern 4			
Composite pattern 2	37.500	0.993	37.714
Pattern 11	10.000	1.000	10.000
Pattern 14	25.000	0.900	26.111
Relationship - Sequential patterns	**72.500**	**0.893**	**79.126**
Composite pattern 5			
Composite pattern 3	42.500	0.993	42.714
Composite pattern 4	72.500	0.893	79.126
Relationship - Decision flows	**57.500**	**0.943**	**60.920**
Composite pattern 6			
Pattern 6	20.000	1.000	20.000
Composite pattern 5	57.500	0.943	60.920
Pattern 10	60.000	0.896	66.778
Relationship - Sequential patterns	**137.500**	**0.844**	**134.808**
Composite pattern 7			
Pattern 21	15.000	0.995	15.075
Pattern 22	15.000	0.995	15.075
Pattern 23	15.000	0.995	15.075
Pattern 24	15.000	0.995	15.075
Relationship - Parallel patterns	**60**	**0.980**	**60.302**
Composite pattern 8			
Pattern 19	10.000	1.000	10.000
Composite pattern 7	60.000	0.980	60.302
Pattern 25	25.000	0.900	27.778
Relationship - Sequential patterns	**95**	**0.882**	**94.779**
Composite pattern 9			

Continued on next page

Table 8.6 – continued from previous page

Task	Cost	Reliability	Business cost
Pattern 26 + Pattern 30	25.000	0.995	25.075
Pattern 27 + Pattern 30	25.000	0.995	25.075
Pattern 28 + Pattern 30	25.000	0.995	25.075
Pattern 29	15.000	0.995	15.075
Relationship - Decision flows	**22.500**	**0.995**	**22.575**
Composite pattern 10			
Composite pattern 9	22.500	0.995	22.575
Pattern 31	40.000	0.900	44.444
Relationship - Sequential patterns	**62.500**	**0.896**	**69.528**
Composite pattern 11			
Composite pattern 8	95.000	0.882	94.779
Composite pattern 10	62.500	0.896	69.528
Relationship - Sequential patterns	**157.500**	**0.790**	**175.368**
Composite pattern 12			
Composite pattern 11	157.500	0.790	175.368
Pattern 18	10.000	1.000	10.000
Relationship - Decision flows	**83.750**	**0.895**	**92.684**
Composite pattern 13			
Composite pattern 17	60.000	0.896	66.778
Composite pattern 12	83.750	0.895	92.684
Relationship - Sequential patterns	**143.75**	**0.801**	**167.229**
Composite pattern 14			
Composite pattern 13	143.750	0.801	167.229
Pattern 20	10.000	1.000	10.000
Relationship - Decision flows	**76.875**	**0.901**	**88.649**
Composite pattern 15			
Pattern 16	25.000	0.896	27.834

Continued on next page

Table 8.6 – continued from previous page

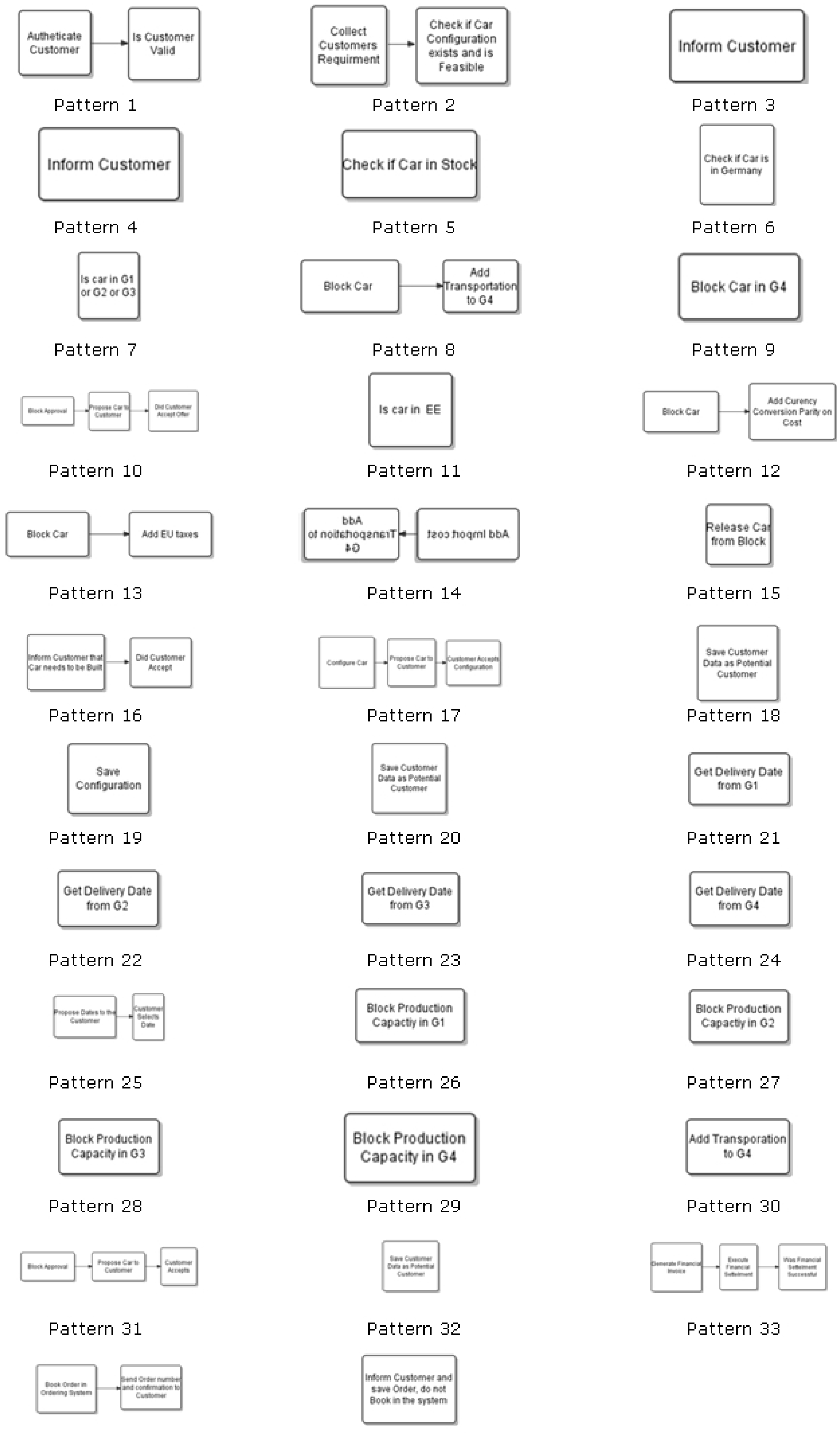
Task	**Cost**	**Reliability**	**Business cost**
Composite pattern 14	76.875	0.901	88.649
Relationship - Sequential patterns	**101.875**	**0.807**	**119.551**
Composite pattern 16			
Pattern 34	20.000	0.995	20.050
Pattern 35	10	0.995	10.050
Relationship - Decision flow	**15.000**	**0.995**	**15.050**
Composite pattern 17			
Pattern 33	35.000	0.900	38.889
Composite pattern 16	15.000	0.995	15.050
Relationship - Sequential patterns	**50.000**	**0.896**	**54.135**
Composite pattern 18			
Pattern 15	10.000	1.000	10.000
Composite pattern 15	101.875	0.807	119.551
Composite pattern 17	50.000	0.896	54.135
Relationship - Sequential patterns	**161.875**	**0.722**	**201.481**
Composite pattern 19			
Composite pattern 17	50.000	0.896	54.135
Composite pattern 18	161.875	0.722	201.481
Relationship - Decision flows	**105.938**	**0.809**	**127.808**
Composite pattern 20			
Composite pattern 6	137.500	0.844	134.808
Composite pattern 19	105.938	0.809	127.808
Relationship - Sequential patterns	**243.438**	**0.683**	**294.462**
Composite pattern 21			
Composite pattern 20	243.438	0.683	294.462
Composite pattern 15	101.875	0.809	119.551

Continued on next page

Table 8.6 – continued from previous page

Task	Cost	Reliability	Business cost
Relationship - Decision flows	**172.656**	**0.745**	**207.006**
Composite pattern 22			
Pattern 5	20.000	1.000	20.000
Composite pattern 21	172.656	0.745	207.006
Relationship - Sequential patterns	**192.656**	**0.745**	**233.859**
Composite pattern 23			
Pattern 4	5.000	1.000	5.000
Composite pattern 22	192.656	0.745	233.859
Relationship - Decision flows	**98.828**	**0.872**	**119.430**
Composite pattern 24			
Pattern 2	35.000	0.980	35.714
Composite pattern 23	98.828	0.872	119.430
Relationship - Sequential patterns	**133.828**	**0.855**	**160.367**
Composite pattern 25			
Composite pattern 24	133.828	0.855	160.367
Pattern 3	5.000	1.000	5.000
Relationship - Decision flows	**69.414**	**0.927**	**82.684**
Composite pattern 26			
Pattern 1	20.000	0.995	20.101
Composite pattern 25	69.414	0.927	82.684
Relationship - Sequential patterns	**89.414**	**0.923**	**104.356**

Table 8.6: Composite patterns found in the business process



Pattern 1 Pattern 2 Pattern 3
Pattern 4 Pattern 5 Pattern 6
Pattern 7 Pattern 8 Pattern 9
Pattern 10 Pattern 11 Pattern 12
Pattern 13 Pattern 14 Pattern 15
Pattern 16 Pattern 17 Pattern 18
Pattern 19 Pattern 20 Pattern 21
Pattern 22 Pattern 23 Pattern 24
Pattern 25 Pattern 26 Pattern 27
Pattern 28 Pattern 29 Pattern 30
Pattern 31 Pattern 32 Pattern 33
Pattern 34 Pattern 35

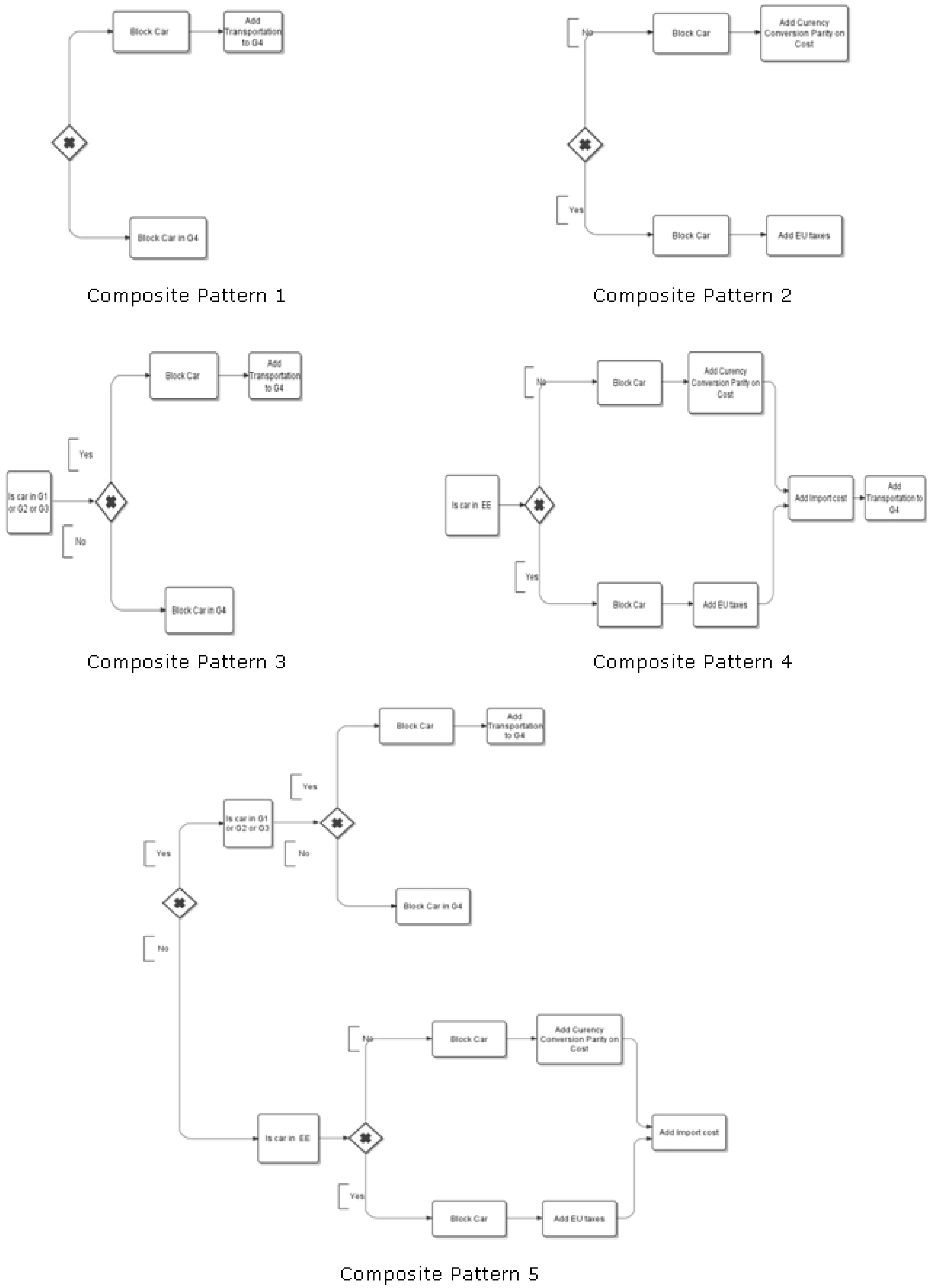
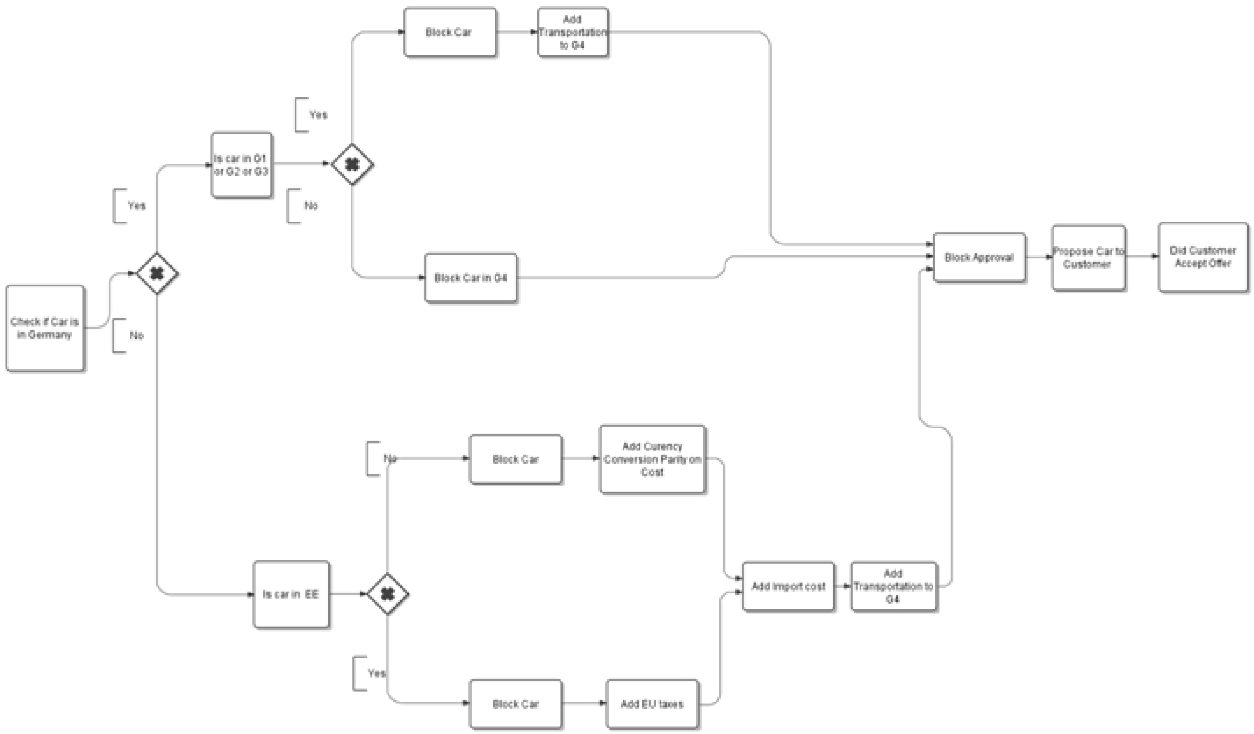


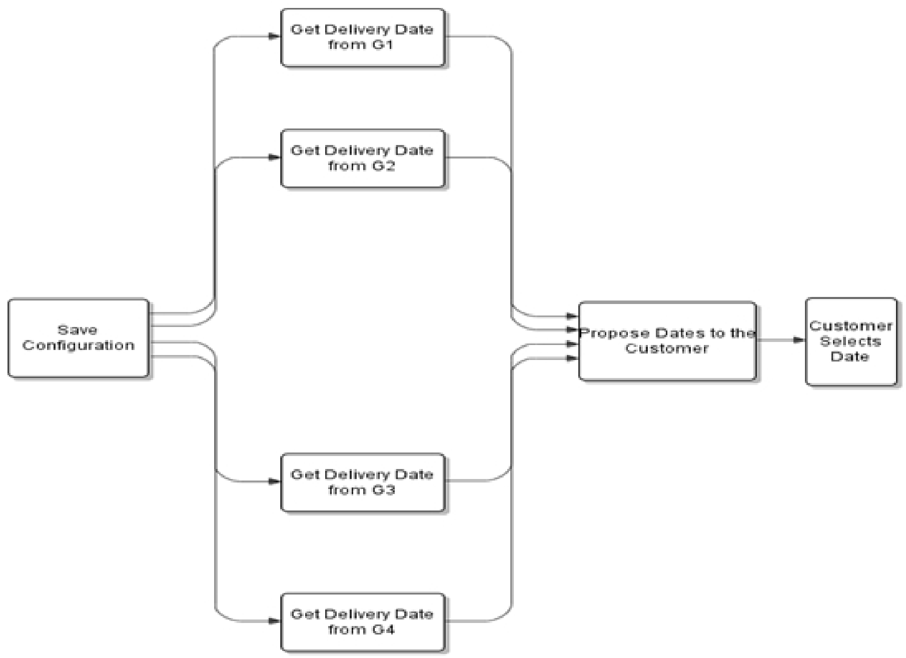
Figure 8.7: Composite patterns



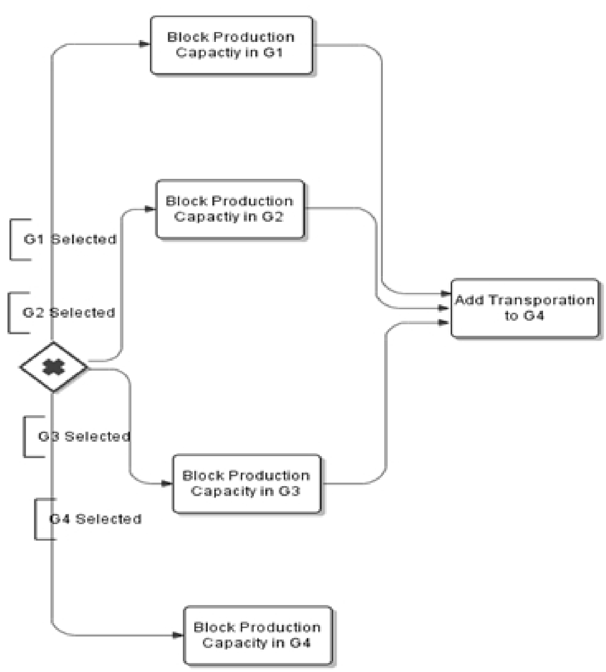
Composite Pattern 6

Composite Pattern 7

Figure 8.8: Composite patterns

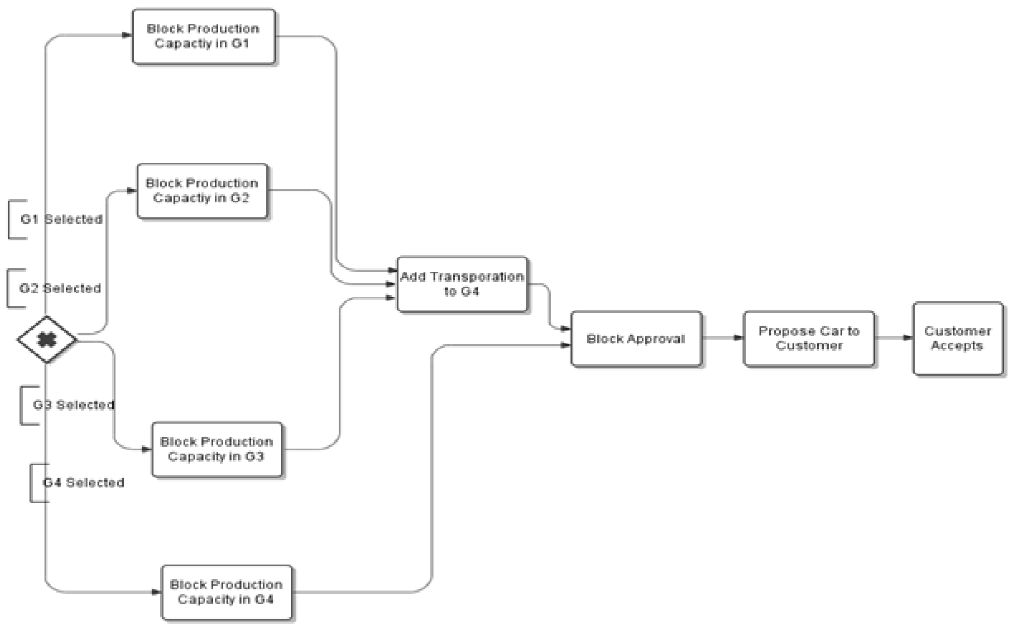


Composite Pattern 8

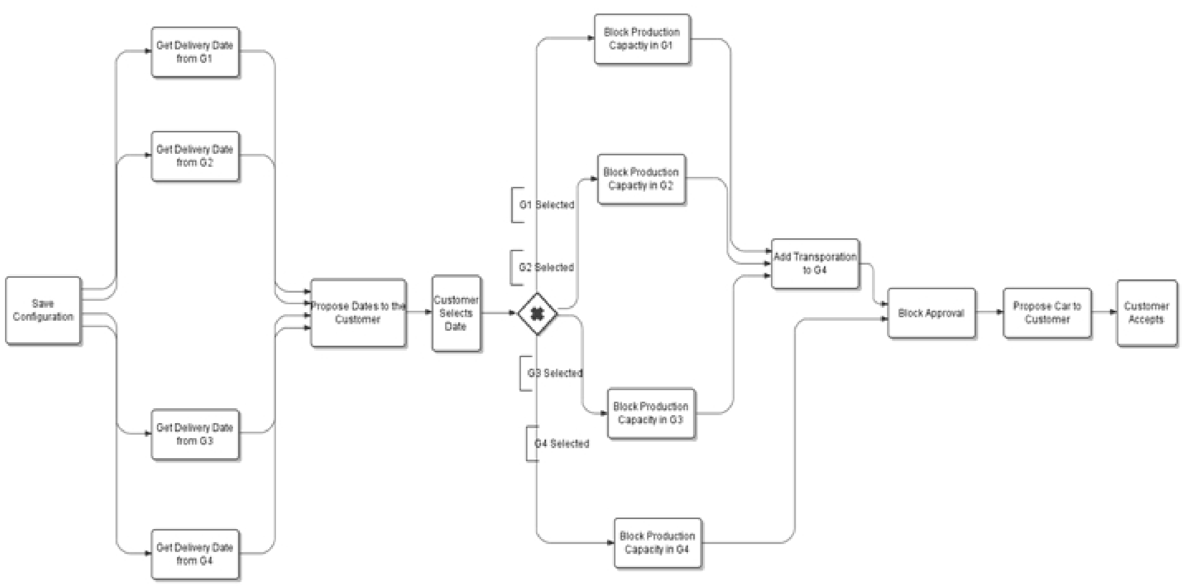


Composite Pattern 9

Figure 8.9: Composite patterns



Composite Pattern 10



Composite Pattern 11

Figure 8.10: Composite patterns

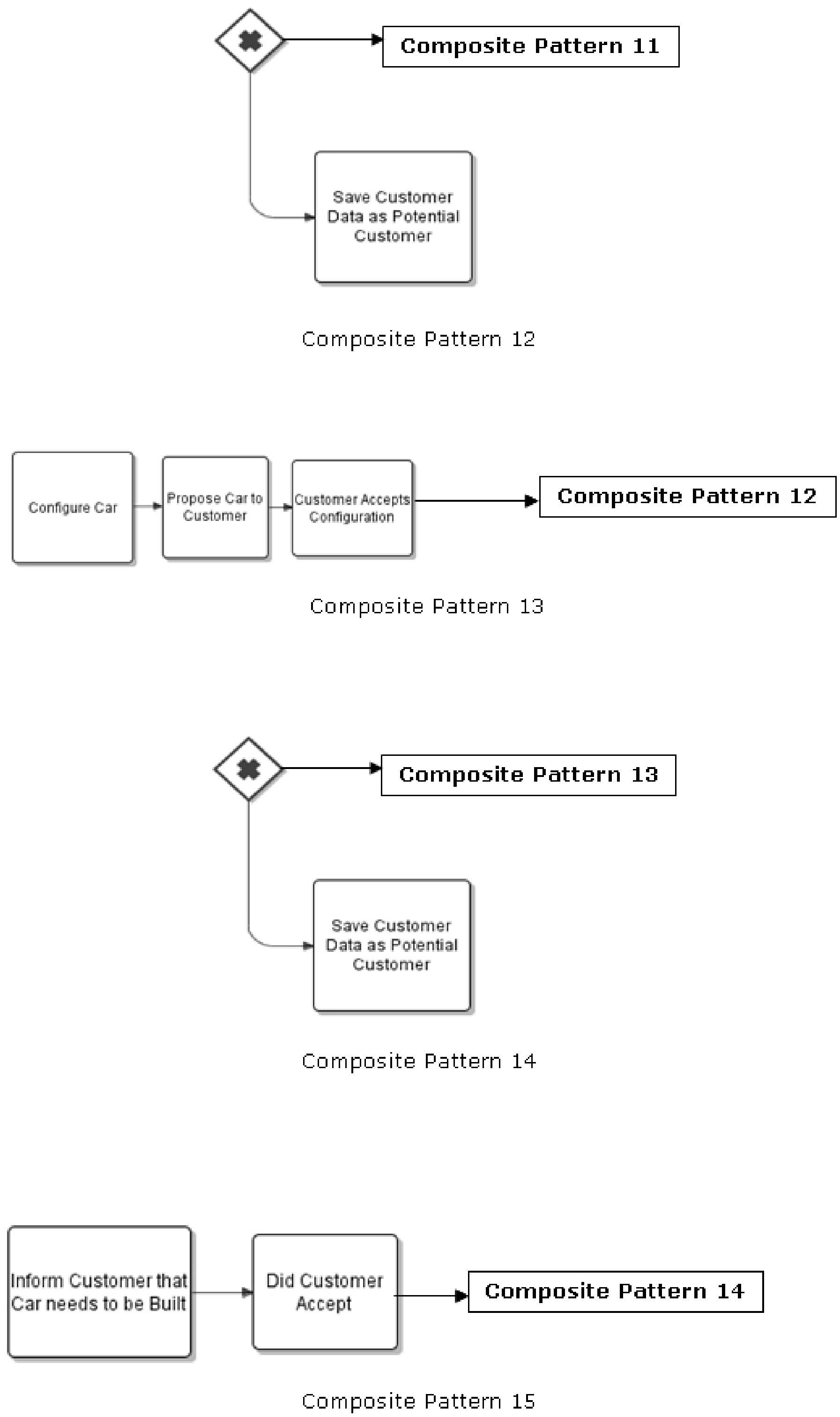
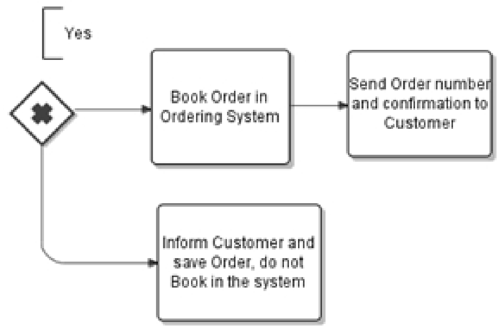
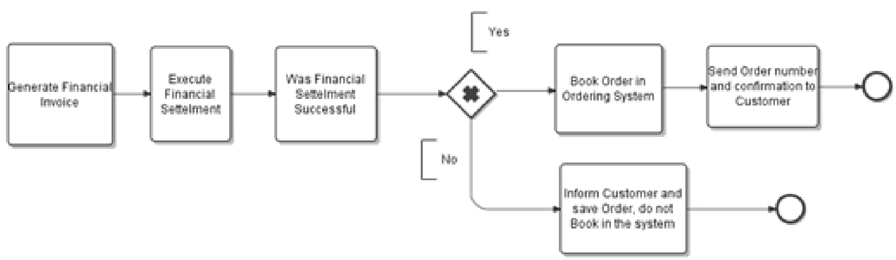


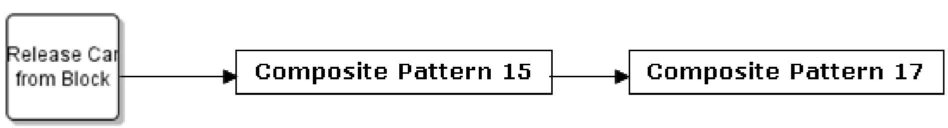
Figure 8.11: Composite patterns



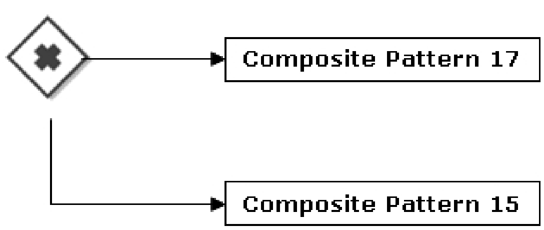
Composite Pattern 16



Composite Pattern 17



Composite Pattern 18



Composite Pattern 19

Figure 8.12: Composite patterns

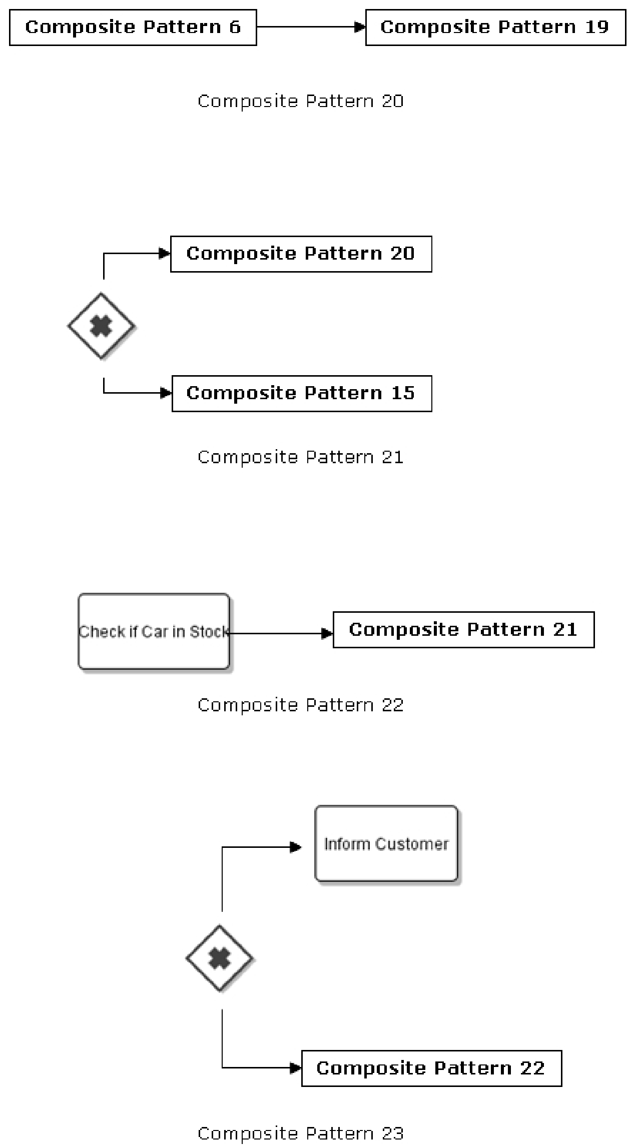
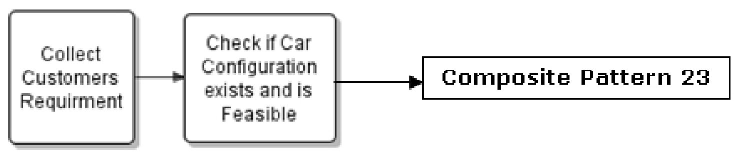
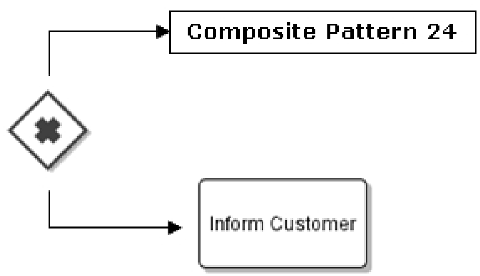


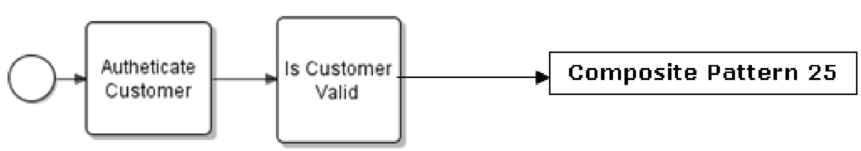
Figure 8.13: Composite patterns



Composite Pattern 24



Composite Pattern 25



Composite Pattern 26

Figure 8.14: Composite patterns

8.2.3 Result

From our calculations we see that the process has a cost of 89.414, reliability of 0.923 and a business cost of 104.356.

$$Cost = 89.414 \tag{8.1}$$

$$Business\ cost = 104.356 \tag{8.2}$$

$$Reliability = 0.923 \tag{8.3}$$

8.3 Conclusion

In this chapter we have considered two processes to calculate the financial parameters using the pattern based methodology. The first example, the payment process, is simple and is made up of 4 patterns. A practical implementation would probably have many more steps and details to process the payment.

The second example, the automobile order management process, is close to a real life implementation in the industry. The steps and in turn the patterns are many and their interactions with each other have a considerable impact on the resulting financial parameters. In both case the methodology to calculate business cost by identifying patterns and their interactions has been successful.

The results of these examples have also a business interpretation. We consider the order management process as this is more industry relevant. In practice an order management process such as this is expected to execute with 100% reliability i.e. a reliability of 1.0. This is especially true for the automobile industry. A reliability of 0.923 can also be interpreted as a failure rate of 0.077. This means that when this process is used 1000 times an order is placed only 923 times leading to a loss of 77 orders. Hence the reliability of this process needs to be increased enormously so as to achieve a higher order rate. An increase in the reliability will also reduce the business cost of this process.

9

Best practice evaluation

Over the past years, with the evolution of many frameworks, there are many practices which are recommended as best practices. These are practices which have shown by implementation that they lead to a positive effect on or more parameters of the business process. The practices are many and deal with all aspects of an organization. For example, there are best practices which also deal with the softer aspects such as motivating the employees etc. A vast collection of best practices and their qualitative analysis has been done by Reijers et al.[23]; this work [23] forms a very reliable source for different other studies, including this chapter. This chapter covers the most commonly recommended best practices for reducing costs in a business process.

In this chapter we

(a) Adapt the devils quadrangle to produce a graphical representation of the impact of the best practice on a process.

(b) Look into the commonly recommended best practices which deal with cost parameter of the business processes.

(c) Analyze these practices by evaluating it as patterns to see its impact on the business cost of the process.

(d) Analyze business processes on different parameters in the "as is" condition and after implementing the recommended best practice.

This contents and results of this chapter are published in [87] and [88].

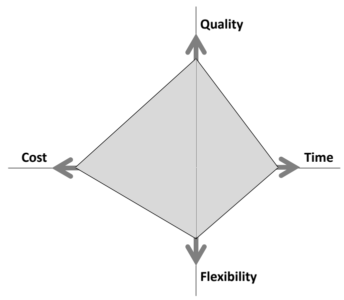


Figure 9.1: Devils quadrangle

9.1 Devils quadrangle

Devils quadrangle is a framework to evaluate the performance of a business process and was proposed by Brand and Van der Kolk [13]. Industry practitioners and researchers believe that the devils quadrangle is best suited to evaluate the performance of a workflow as it has all the performance measures needed. The devils quadrangle is represented as a quadrangle as shown in the Figure 9.1. It defines four dimensions i.e. time, cost, quality, and flexibility. Every business process needs to create a balance on these dimensions. When used, these dimensions are interpreted differently or at different maturity levels as the situation demands. Any change that is done to a business process leads to an impact on these dimensions. It is not necessary that the betterment on the value of one of the dimensions leads to an automatic betterment of the other one as well. One example which is seen almost always is the effort to decrease the cost dimension. This dimension usually shows that the Quality dimension starts coming down or in other words the quality of the business process starts decreasing.

Our study revolves around three major parameters of a business process i.e. cost, reliability and business cost. The devils quadrangle provides for a strong foundation to evaluate our study. The impact of reliability on the cost and in turn on the business cost of a business process can be well represented through this framework. So as to make this possible we will have to adapt the parameters from the devils quadrangle.

We define the devils quadrangle with the following four parameters:

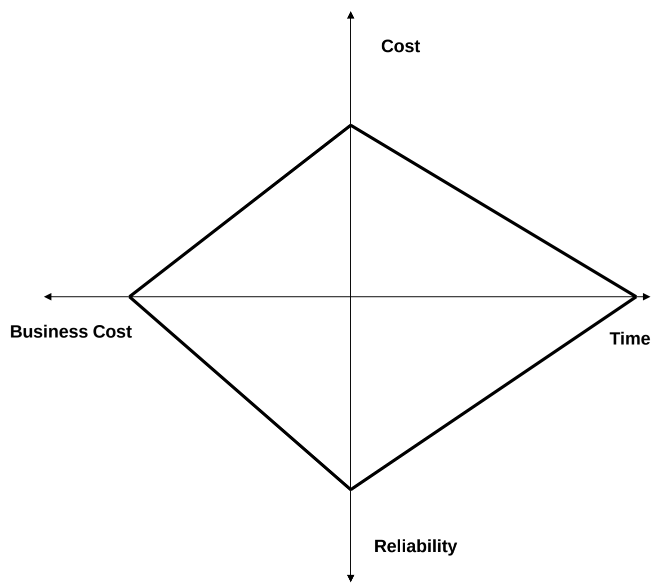


Figure 9.2: Adapted devils quadrangle

(a) **Cost:** Here we do not differ from the devils quadrangle. We take cost as the sum of all expenses so as to keep a business process running.

(b) **Reliability:** We replace the original parameter quality with the parameter reliability.

(c) **Business cost:** We replace the original parameter flexibility with the parameter business cost.

(d) **Time:** We have not considered time as a parameter in our study. We use this parameter so as to complete the quadrangle. As we do not take this into consideration we always make the assumption that it is constant.

9.1.1 Ideal case for a devils quadrangle

We use the adapted version of the devils quadrangle to evaluate the impact of implementation of the best practices on business processes. The parameters of the devils

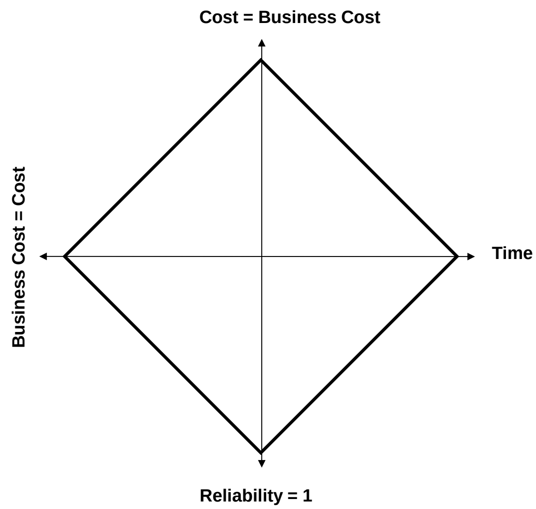


Figure 9.3: Ideal case - Adapted devils quadrangle

quadrangle are inter-related to each other. A betterment of one of the parameters either leads to an improvement or decline of one or more of the other parameters. The cost of a business process can be calculated as shown in the chapter “Patterns”. The reliability of a business process ranges between 0 and 1. As the reliability decreases it increases the business cost and vice-a-versa. The parameter time is constant. The aim of the business process is to increase the reliability and reduce the business cost. The aim is to achieve the reliability as close as possible to 1 and in turn bring the business cost as close as possible to the cost of the business process. In other words the ideal case for the devils quadrangle is when the Reliability = 1 and the Cost = Business cost. The graphical representation is shown in the Fig. 9.3.

9.2 Resequencing of Tasks

This best practice is also called "Process Order Optimization" and is mentioned by Klein [70, 25]. In a business process, the ordering of tasks does not reveal the logic behind the process and hence it could be that tasks in a process are executed even though it is not required at that moment. This best practice recommends that tasks such as these when re-sequenced in a business process help in cost reductions. The logic is to execute a task only when the task is really needed to be executed. In the flow of a business process which contains conditional executions, tasks which need to be done only when a condition is fulfilled needs to be moved to the point when the condition is really true.

9.2.1 Original Order

So as to evaluate this best practice, we consider a business process to book a flight. The corresponding BPD is as shown in the Fig. 9.4. The business process authenticates the customer, validates the inputs and finds the flights that meet the criterion. If a flight is available then the flight is booked and confirmation is sent to the customer. If the flight is not available then the customer is informed of the unavailability. We assume that the flight is available in 50 percent of the cases. Also, we make assumptions on the cost and reliability of the tasks in the business process.

Cost calculation: So as to calculate the business cost of this process, we divide the BPD in patterns. The break up is as shown in the Fig. 9.5. For each of the patterns, we make assumptions on the cost and reliability of all the tasks. In turn we calculate the business cost for each of the patterns.

The calculation at the level of the patterns is as shown in the tables 9.1 9.2 9.3.

From the calculations in the table we get the cost, reliability and business cost of the patterns as:

$$Cost(Pattern\ 1) = 25 \tag{9.1}$$

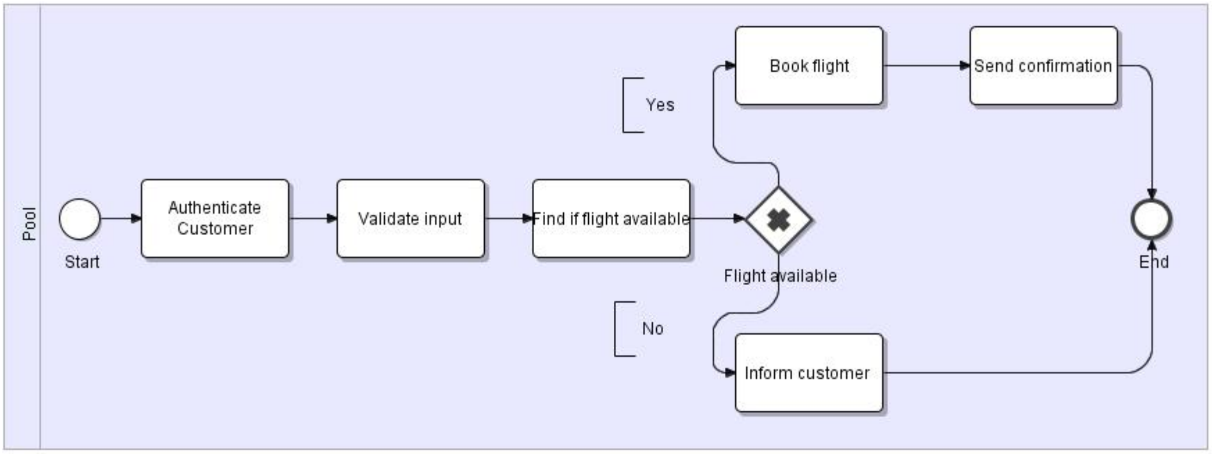


Figure 9.4: Business process diagram to book a flight

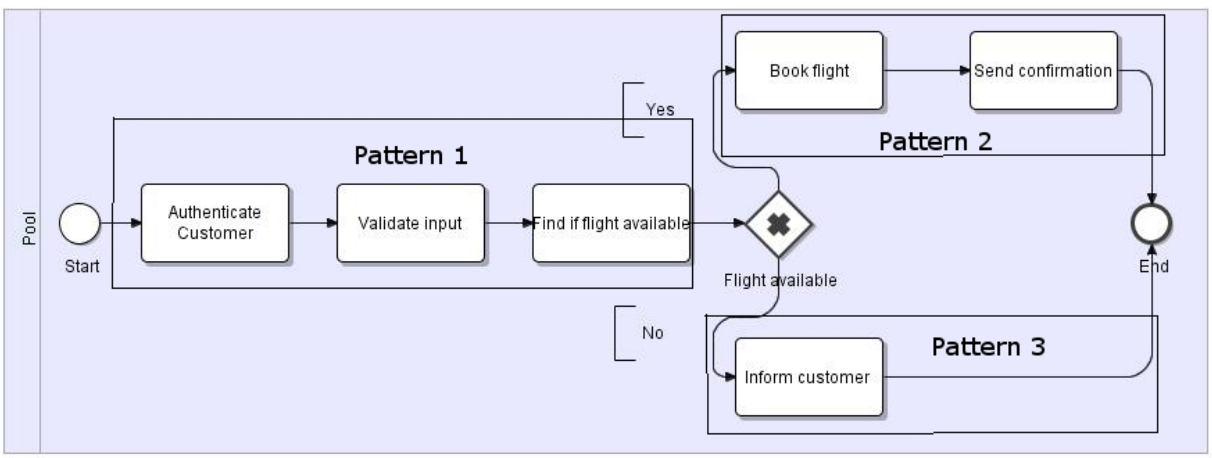


Figure 9.5: Patterns in the business process diagram

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate	10	0.9	11.11	11.11
Validate input	10	0.9	11.11	23.5
Find if flight available	5	0.8	6.25	35.6

Table 9.1: Flight Booking - Pattern 1

Task	Cost	Reliability	Business cost of the task	Business cost
Book Flight	30	0.9	33.33	33.33
Send confirmation	10	0.9	11.11	48.1

Table 9.2: Flight Booking - Pattern 2

Task	Cost	Reliability	Business cost of the task	Business cost
Inform Customer	5	0.9	5.56	5.56

Table 9.3: Flight Booking - Pattern 3

$$Cost(Pattern\ 2) = 40 \tag{9.2}$$

$$Cost(Pattern\ 3) = 5 \tag{9.3}$$

$$Reliability(Pattern\ 1) = 0.648 \tag{9.4}$$

$$Reliability(Pattern\ 2) = 0.81 \tag{9.5}$$

$$Reliability(Pattern\ 3) = 0.9 \tag{9.6}$$

$$Business\ cost(Pattern\ 1) = 35.6 \tag{9.7}$$

$$Business\ cost(Pattern\ 2) = 48.1 \tag{9.8}$$

$$Business\ cost(Pattern\ 3) = 5.56 \tag{9.9}$$

We assume that there is always a 50 percent chance of finding the flight according to the customer inputs. Hence the cost, business cost and reliability of pattern 2 and pattern 3 would then be:

$$Cost(Pattern\ 2\ with\ Pattern\ 3) = 40 * 0.5 + 5 * 0.5 = 22.5 \tag{9.10}$$

Task	Cost	Reliability	Business cost of the task	Business cost
Validate input	10	0.9	11.11	11.11
Find if flight available	5	0.8	6.25	20.1

Table 9.4: Flight Booking - Pattern 1

$$Business\ cost(Pattern\ 2\ with\ Pattern\ 3) = 48.1 * 0.5 + 5.56 * 0.5 = 26.85 \quad (9.11)$$

$$Reliability(Pattern\ 2\ with\ Pattern\ 3) = 0.81 * 0.5 + 0.9 * 0.5 = 0.85 \quad (9.12)$$

Hence the total cost, business cost and reliability of the BPD would then be:

$$Cost(Pattern\ 1\ (Pattern\ 2\ with\ Pattern\ 3)) = 22.5 + 25 = 47.5 \quad (9.13)$$

$$Business\ cost(Pattern\ 1\ (Pattern\ 2\ with\ Pattern\ 3)) = 26.85 + 35.6/0.85 = 68.5 \quad (9.14)$$

$$Reliability(Pattern\ 1\ (Pattern\ 2\ with\ Pattern\ 3)) = 0.648 * 0.85 = 0.55 \quad (9.15)$$

9.2.2 Changed Order with Tasks Resequencing

In the BPD in the Fig. 9.4 we re-sequence the tasks in such a way that a task is executed only when it is required. We move the task “Authenticate Customer” to the part when the flight has already been booked. This change is shown in Fig. 9.5. Fig. 9.5 also shows the patterns in the BPD.

Tables 9.4 9.5 9.6 show the cost and reliability of the tasks and the patterns together.

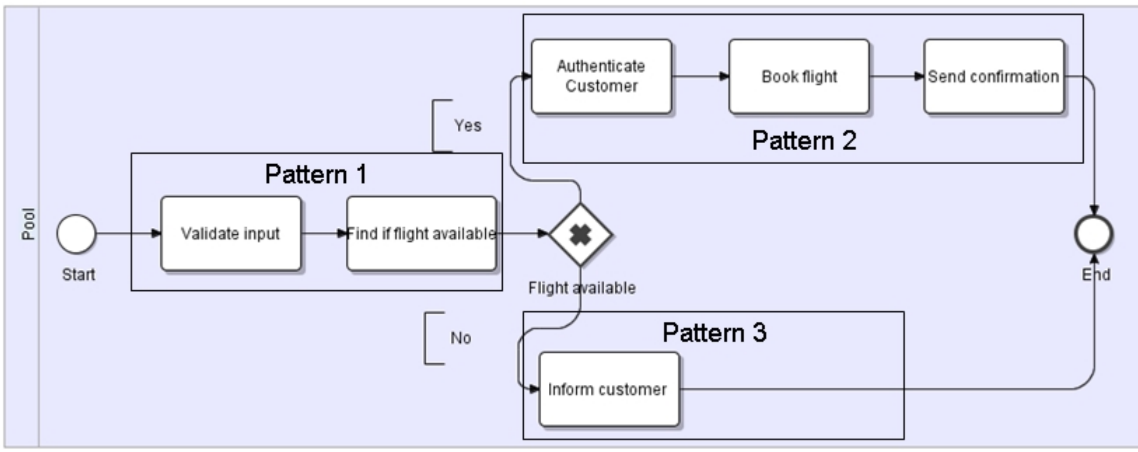


Figure 9.6: Business process diagram to book a flight

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate Customer	10	0.9	11.11	11.11
Book Flight	30	0.9	33.33	45.7
Send confirmation	10	0.9	11.11	61.9

Table 9.5: Flight Booking - Pattern 2

Task	Cost	Reliability	Business cost of the task	Business cost
Inform Customer	5	0.9	5.56	5.56

Table 9.6: Flight Booking - Pattern 3

From the calculations in the tables we calculate the cost, business cost and reliability as:

$$Cost(Pattern\ 1) = 15 \tag{9.16}$$

$$Business\ cost(Pattern\ 1) = 20.1 \tag{9.17}$$

$$Reliability(Pattern\ 1) = 0.72 \tag{9.18}$$

$$Cost(Pattern\ 2) = 50 \tag{9.19}$$

$$Business\ cost(Pattern\ 2) = 61.9 \tag{9.20}$$

$$Reliability(Pattern\ 2) = 0.73 \tag{9.21}$$

$$Business\ cost(Pattern\ 3) = 5 \tag{9.22}$$

$$Business\ cost(Pattern\ 3) = 5.56 \tag{9.23}$$

$$Reliability(Pattern\ 3) = 0.9 \tag{9.24}$$

We assume that there is always a 50 percent chance of finding the flight according to the customer inputs. Hence the cost, business cost and reliability of pattern 2 and pattern 3 would then be:

$$Cost(Pattern\ 2\ with\ Pattern\ 3) = 50 * 0.5 + 5 * 0.5 = 27.5 \tag{9.25}$$

$$Business\ cost(Pattern\ 2\ with\ Pattern\ 3) = 61.9 * 0.5 + 5.56 * 0.5 = 33.71 \tag{9.26}$$

$$Reliability(Pattern\ 2\ with\ Pattern\ 3) = 0.73 * 0.5 + 0.9 * 0.5 = 0.814 \tag{9.27}$$

Hence the total cost, business cost and reliability of the BPD would then be:

$$Cost(Pattern\ 1\ (Pattern\ 2\ with\ Pattern\ 3)) = 27.5 + 15 = 42.5 \tag{9.28}$$

$$Business\ Cost(Pattern\ 1\ (Pattern\ 2\ with\ Pattern\ 3)) = 33.71 + 20.1/0.814 = 58.4 \tag{9.29}$$

$$Reliability(Pattern\ 1\ (Pattern\ 2\ with\ Pattern\ 3)) = 0.72 * 0.814 = 0.59 \tag{9.30}$$

9.2.3 Impact of resequencing of tasks

We see from the calculations that the change in the sequence leads to a change in the cost, business cost and the reliability. The change is very much dependent on the task which is re-sequenced. The devils quadrangle is as shown in the Fig. 9.7. The original patterns are shown in orange whereas the implemented pattern is shown in green. We see that the change in the sequence leads to a change in all the parameters i.e. cost, business cost and reliability of the process.

9.3 Knock out order

Every business process has conditions that need to be checked for. If the conditions are not fulfilled then the execution of the process is terminated. This best practice recommends that conditions which have the highest probability to terminate the process should be executed right at the start, followed by the condition which has the next highest probability to terminate the process and continue so on. As a result of the implementation of this best practice, the tasks within a business process will be moved or re-sequenced. Hence this implies a logic through which the resequencing best

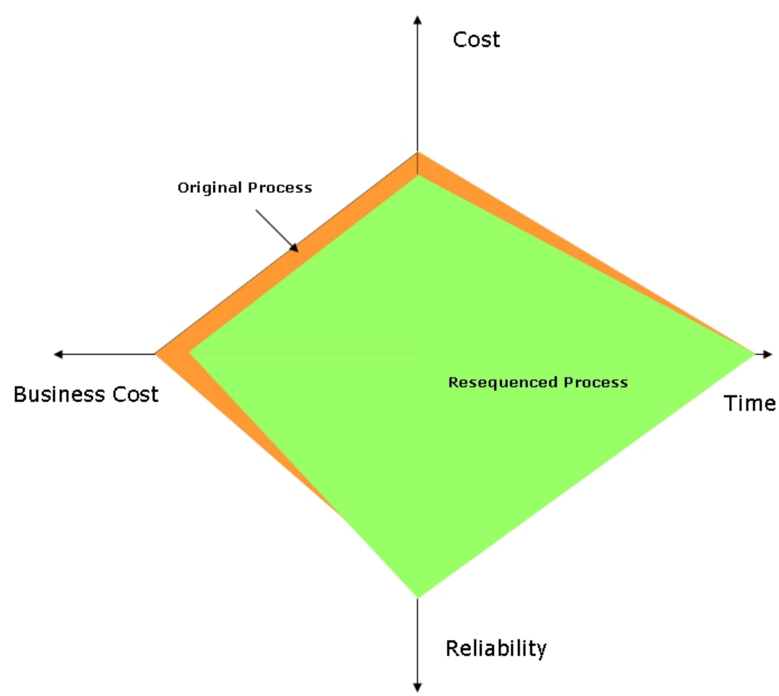


Figure 9.7: Devils quadrangle before and after implementation of “Resequencing of tasks”

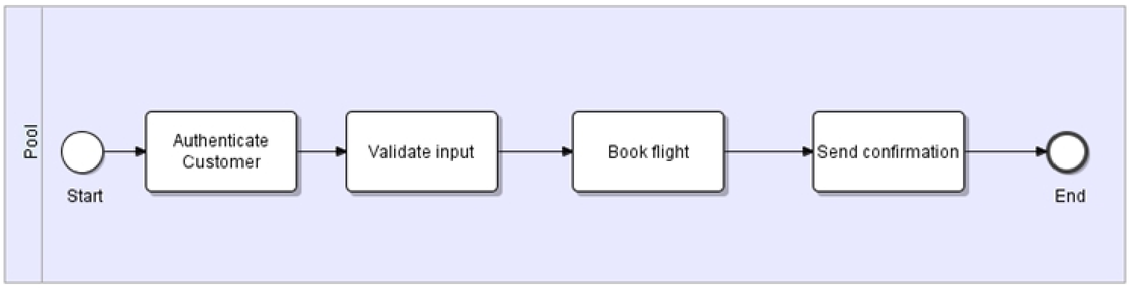


Figure 9.8: Business process Diagram to book a flight

practice[23] can be implemented.

In such a case the reasons why a business process needs to be terminated is accomplished at the very start. This will make sure that the rest of the business process is executed with a high probability of achieving the business value. Van der Aalst [33, 36, 37, 68, 43, 23] mentions this best practice and also gives quantitative support for its optimality.

This best practice, in its representation, believes in evaluating the conditional tasks right in the front. For us, every task in the business process has a certain reliability with which it performs. Hence the interpretation of this best practice in our case would mean that the business cost of the process will be lower in case the initial part of the process has a lower reliability than the latter. So as to evaluate this we consider the business process to book a flight, nevertheless we will assume that all the flights are available. Hence there is no condition involved to check for the availability of the flight. This is shown in the Fig. 9.8.

9.3.1 Original case - Task order with descending reliability

We make assumptions on the cost and reliability such that the reliability of the tasks has a descending order. We assume the costs are the same on each of the tasks. The calculation of the business cost is as shown in the table 9.7.

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate Customer	10	0.9	11.11	11 .11
Validate Input	10	0.8	12.50	26.4
Book Flight	10	0.7	14.29	52.0
Send Confirmation	10	0.6	16.67	103.3

Table 9.7: Flight booking - Task order with descending reliability

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate Customer	10	0.6	16.67	16.67
Validate Input	10	0.7	14.29	38.1
Book Flight	10	0.8	12.50	60.1
Send Confirmation	10	0.9	11.11	77.9

Table 9.8: Flight booking - Task order with ascending reliability

Hence the cost, business cost and reliability with descending order of reliability would be:

$$Cost(Descending\ reliability) = 40 \tag{9.31}$$

$$Business\ Cost(Descending\ reliability) = 103.3 \tag{9.32}$$

$$Reliability(Descending\ reliability) = 0.30 \tag{9.33}$$

9.3.2 Changed case - Task order with ascending reliability

We make assumptions on the cost and reliability such that the reliability of the tasks now has an ascending order. The calculation of the business cost is as shown in the table 9.8:

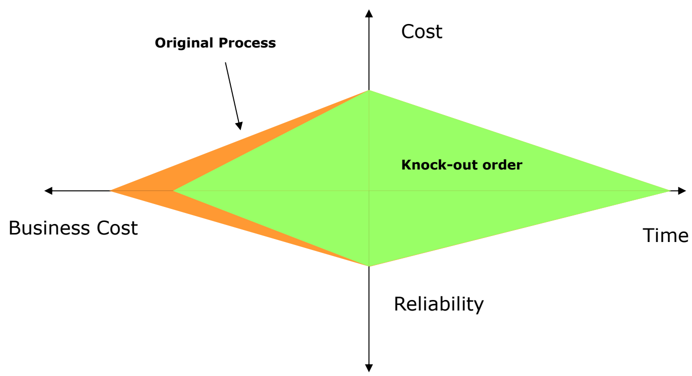


Figure 9.9: Devils quadrangle before and after implementation

Hence the business cost and reliability with ascending order of reliability would be:

$$Cost(Ascending\ reliability) = 40 \tag{9.34}$$

$$Business\ Cost(Ascending\ reliability) = 77.9 \tag{9.35}$$

$$Reliability(Ascending\ reliability) = 0.30 \tag{9.36}$$

9.3.3 Impact of knock-out order

The example shows that the knock-out order brings the business cost of the business process down. The knock-out sequence pushes the tasks which have the highest probability of terminating the process to the front and in turn makes sure that the rest of the process is executed only when all the conditions are met. The best practice does not change the overall reliability and cost of the process. This is shown in devils quadrangle as in the Fig. 9.9.

9.4 Task elimination: Eliminate unnecessary tasks from a business process

This best practice recommends that the tasks which are having no value or tasks which are redundant should be taken out or eliminated. A task in a business process when eliminated reduces the business cost of the process. Nevertheless eliminating a task in the business process will have to make sure that the task is completely unnecessary or the task is now done as part of another task. In case tasks in a business process are eliminated from the optimization perspective, this will then lead to a compromise on the quality of the process. In which case it is a case by case decision if the task should be eliminated or not. There are different ways in which an evaluation can be done so as to find if tasks are unnecessary or redundant, one of these is to take the customer perspective. Another common way is to look into tasks which consider iterations. Iterations indicate that a certain task is done "n" number of times because it has not achieved the business value at once.

We consider an example to book a hotel to evaluate this best practice. Consider a business process as shown in Fig. 9.10 where an agency tries to find a room in a hotel according to the inputs given by the customer. Finding a room in a hotel is an iterative process. The travel agency nevertheless would like to try in every hotel possible to find a room before until a room is found. Hence this task would be executed iteratively until the business objective is met. In case every loop in this iteration costs some money, the travel agency will need to decide on the number of hotels that they are willing to contact to find a room.

The cost of a task in a business process which has a looping to provide for business reliability varies according to the order in which each of the providers is called for. We use the BPD from the hotel booking process (see Fig. 9.11) in this case. Let's assume that there are six hotels with which the process interacts in a sequential order.

To check for the variations it could bring in the cost we make assumptions here such as: the hotel which provides the highest reliability also has the highest service charges or costs.

We execute this iteration in two scenarios:

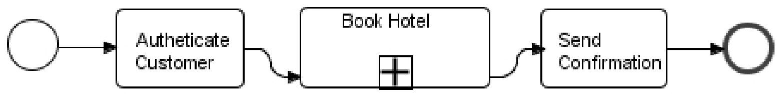


Figure 9.10: Business process diagram to book a hotel

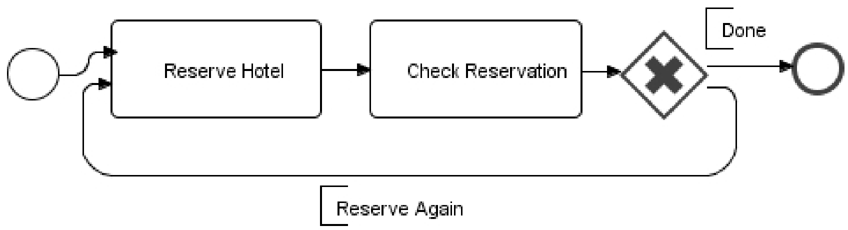


Figure 9.11: Detailed BPD for booking a hotel

9.4.1 Scenario 1: Highest reliability - Highest cost

In this scenario we lay highest priority on the reliability of the called service. Table 9.9 shows the hotels, there reliability, costs etc.

Fig. 9.12 shows the variation of costs with respect to reliability that a combination of

Option	Cost	Reliability	Cumulative-Rel	Actual cost	Business cost
Hotel 1	6	0.9	0.9000	6.00	6.67
Hotel 2	5	0.8	0.98	6.500	6.633
Hotel 3	4	0.7	0.994	6.580	6.620
Hotel 4	3	0.6	0.9976	6.598	6.614
Hotel 5	2	0.5	0.9988	6.603	6.611
Hotel 6	1	0.4	0.99928	6.604	6.609

Table 9.9: Sample values

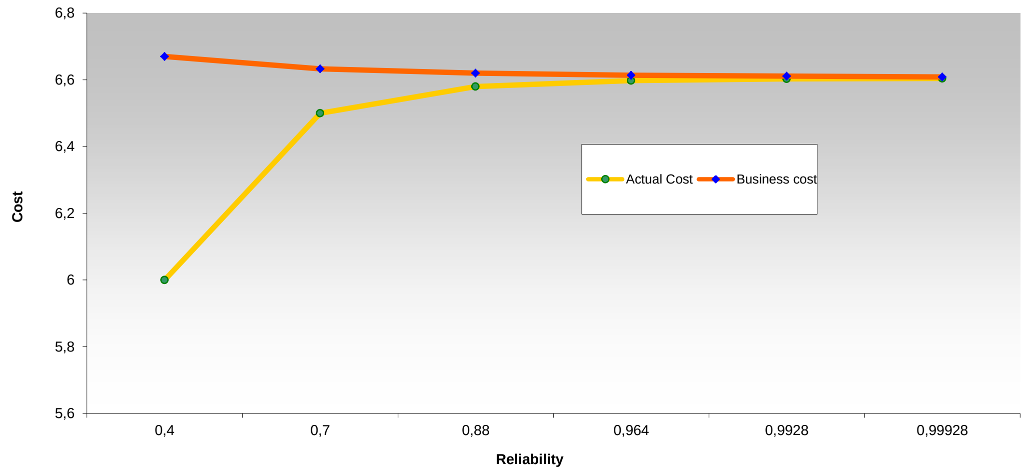


Figure 9.12: Variation of cost according to reliability

alternative services offer.

9.4.2 Scenario 2: Achieve minimum increase in actual cost with increase in reliability

In this scenario we start with the least reliable hotel and then select hotels with ascending order of reliability. Table 9.10 shows the hotels with the development of cost, reliability, etc.

Fig. 9.13 shows the variation of costs with respect to the reliability that combination of alternative services offer.

We see in the Scenario 2 that the development of Actual cost is much lower in comparison to Scenario 1. This is because the need to search for a new hotel arises only when the contact with the previous hotel(s) fails. Hence the cost of the new hotel is

Option	Cost	Reliability	Cumulative-Rel	Actual cost	Business cost
Hotel 1	1	0.4	0.4	1.000	2.500
Hotel 2	2	0.5	0.7	2.200	3.143
Hotel 3	3	0.6	0.88	3.100	3,523
Hotel 4	4	0.7	0.964	3580	3,714
Hotel 5	5	0.8	0.9928	3.760	3,787
Hotel 6	6	0.9	0.99928	3.803	3,806

Table 9.10: Sample values

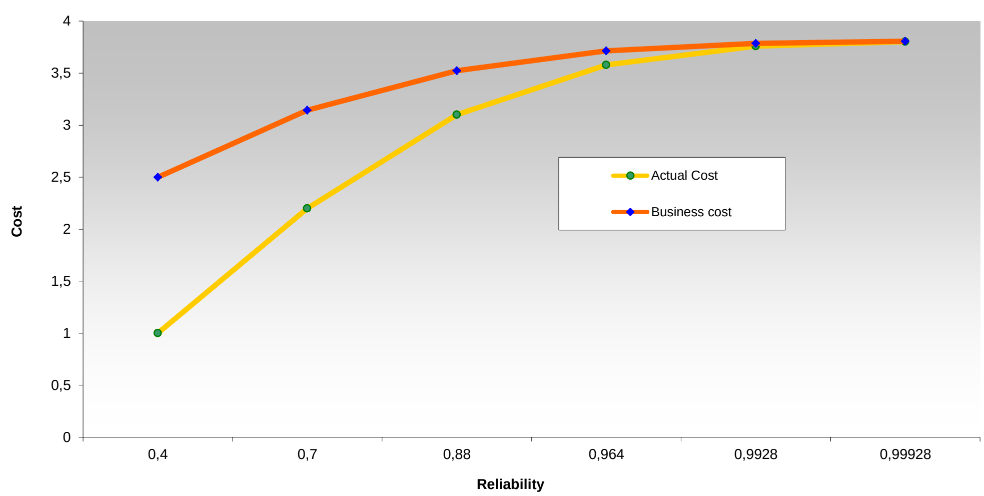


Figure 9.13: Variation of cost according to reliability.

multiplied with the failure i.e. 1 - reliability, of the hotels before it and added to the Actual cost.

9.4.3 Impact of Task Elimination

We see from the calculation as shown in the Table 9.9 for scenario 1 that by adding the sixth hotel on the list, the reliability that the business objective to find a room in the hotel increases from 0.998 to 0.999. The same is the case in scenario 2 where the reliability increases by very small percentage between the 5th and 6th hotel. Also, in scenario 1 we see that we check only the first hotel to reach a reliability of 0.9 whereas in the scenario 2 four hotels need to be checked before a reliability of 0.9 is achieved. Nevertheless, in both the cases the business cost does not increase by a huge margin and hence this could be an option to keep the iteration. But this situation could also be because the cost for each of the iteration is coming down in comparison to the previous iteration. This leads to the situation where it might be that the last iteration need not be executed at all. In other words this redundancy can be eliminated and in turn there will be no or very less impact on the business process or the costs that are involved.

9.5 Order type and triage

Reijer. et al. have analyzed the order type and triage as two different best practices. Order type [23] is 'determine whether tasks are related to the same type of order and, if necessary, distinguish new business processes'. Triage [23] is 'the division of a general task into two or more alternative tasks or consider the integration of two or more alternative tasks into one general task'. Both these best practices are similar to each other, at least in their intentions. Both of them are recommended so as to improve quality and in turn reduce costs by either breaking tasks into many or by grouping certain tasks together.

Both these best practices are mentioned by a host of researchers which includes [18], [38], [69], [70] and [77], etc.

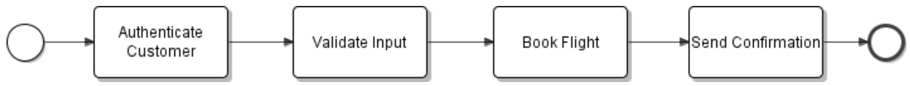


Figure 9.14: Process to book a flight

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate Customer	10	0.6	16.67	16.67
Validate Input	10	0.7	14.29	38.1
Book Flight	10	0.8	12.50	60.1
Send Confirmation	10	0.9	11.11	77.9

Table 9.11: Flight Booking

9.5.1 Original order

So as to evaluate this business process we use the process for booking a flight as shown in Fig. 9.14. The business cost of the process on a sample set of cost and reliability values is as shown in the table 9.11.

The cost, business cost and reliability in the original order is:

$$Cost(Original\ order) = 40 \tag{9.37}$$

$$Business\ Cost(Original\ order) = 77.9 \tag{9.38}$$

$$Reliability(Original\ order) = 0.30 \tag{9.39}$$

9.5.2 Changed Order

For the evaluation process we will put the tasks "Book Flight" and "Send Confirmation" together as one task. We will also assume that the cost of the new task is the summation of the costs of both the tasks and the reliability is the product of the reliabilities of the two tasks. In such a case the corresponding business cost is as shown in

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate Customer	10	0.6	16.67	16.67
Validate Input	10	0.7	14.29	38.1
Book Flight and Send Confirmation	20	0.72	27.78	80.7

Table 9.12: Flight Booking

the table 9.12.

The cost, business cost and reliability in the changed order is:

$$Cost(Changed\ order) = 40 \tag{9.40}$$

$$Business\ cost(Changed\ order) = 80.7 \tag{9.41}$$

$$Reliability(Changed\ order) = 0.30 \tag{9.42}$$

9.5.3 Impact of order type and triage

We see that the business cost has increased when we put the tasks together. This is shown in devils quadrangle as in the Fig.9.15. This is because the combined reliability of the task is less than the two individual tasks. Combining two tasks might increase the quality and increase optimization, nevertheless this doesn't necessarily mean that the cost of the process decreases. The combined task will produce a reduction in the business cost when:

$$Cost(New\ task)\ <=\ Cost(A)\ +\ Cost(B),\ Rel(New\ task)\ >=\ Rel(A)\ *\ Rel(B) \tag{9.43}$$

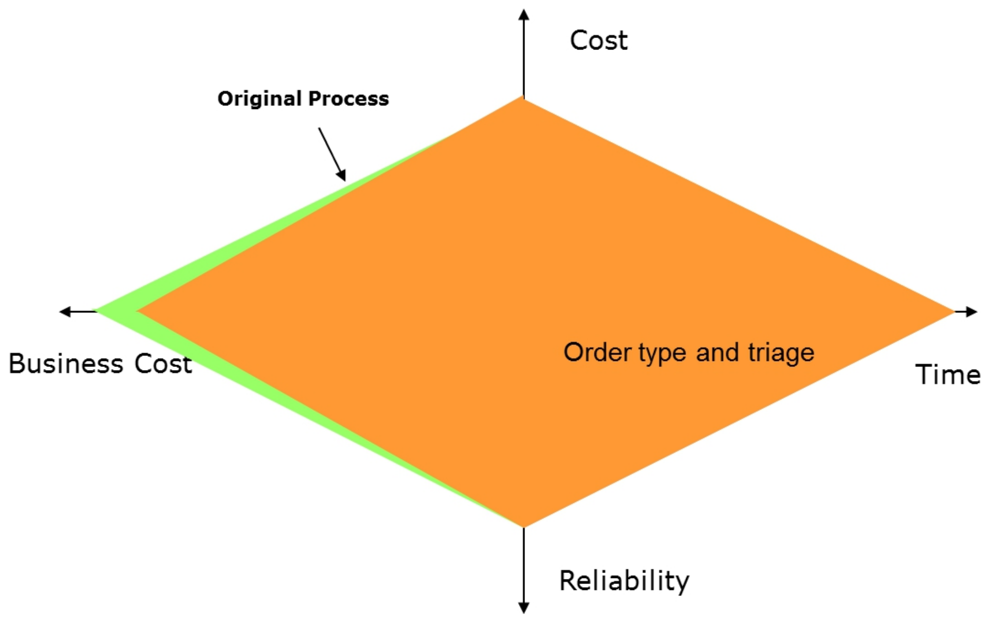


Figure 9.15: Devils quadrangle before and after implementation

9.6 Parallelism: consider whether tasks may be executed in parallel

This best practice recommends that wherever possible tasks should be executed in parallel instead of the sequential order. By doing this the business cost has an effect, probably bringing it down. At the same time the quality and co-ordination efforts increase due to the parallel execution of the tasks.

9.6.1 Original order

So as to evaluate this best practice we take the business process to book a hotel and a flight depending upon customer inputs. We execute this process in sequential order to evaluate the impact on the business costs. The process is as shown in Fig. 9.16.

We make assumptions on cost and reliability. The table 9.13 shows the calculation of the business cost on these assumptions.

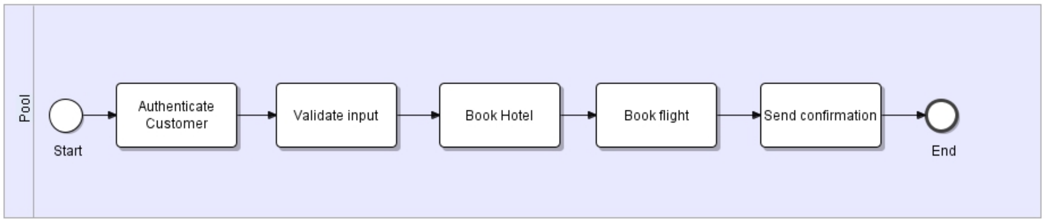


Figure 9.16: Hotel and flight booking in sequential order

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate	10	0.9	11.11	11.11
Validate input	15	0.9	16.67	29
Book Hotel	20	0.7	28.57	70
Book Flight	30	0.7	42.86	142.9
Send confirmation	10	0.9	11.11	169.9

Table 9.13: Hotel and flight booking in sequential order

From the calculations, the cost, business cost and reliability is:

$$Cost(Original\ order) = 85 \tag{9.44}$$

$$Business\ cost(Original\ order) = 169.9 \tag{9.45}$$

$$Reliability(Original\ order) = 0.357 \tag{9.46}$$

9.6.2 Changed order

Now we consider the same process in parallel order, we do the tasks “Book Hotel” and “Book Flight” in parallel to each other. This is shown in the Fig. 9.17. The pattern division is also as shown in the figure.

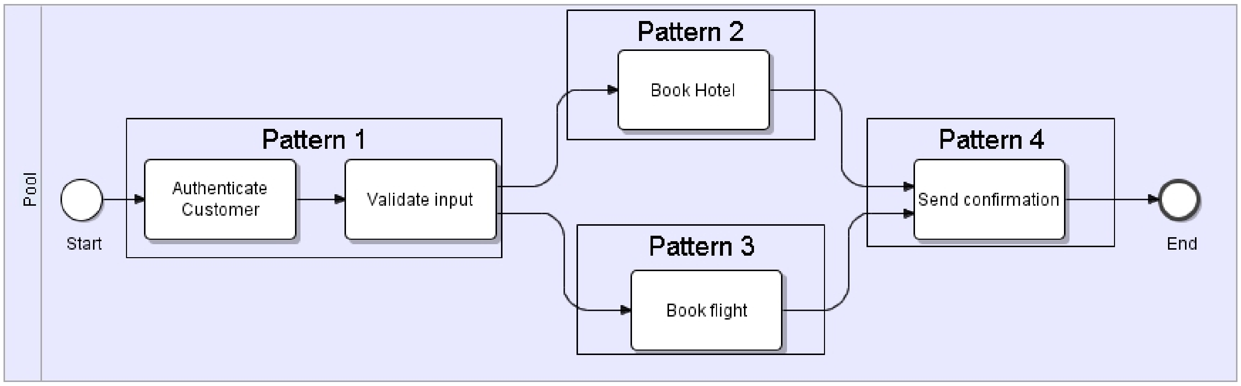


Figure 9.17: Hotel and flight booking in sequential order

Task	Cost	Reliability	Business cost of the task	Business cost
Authenticate	10	0.9	11.11	11.11
Validate input	15	0.9	16.67	29

Table 9.14: Hotel and flight booking in parallel order - Pattern 1

The tables 9.14 9.15 9.16 9.17 shows the calculation of the business cost according to the patterns.

As pattern 2 and pattern 3 are in parallel, the business cost from them together is as shown in table 9.18:

The total business cost by breaking the process in parallel is as shown in table 9.19:

The cost, business cost and reliability by breaking the process in parallel is:

Task	Cost	Reliability	Business cost of the task	Business cost
Book Hotel	20	0.7	28.57	28.57

Table 9.15: Hotel and flight booking in parallel order - Pattern 2

Task	Cost	Reliability	Business cost of the task	Business cost
Book Flight	30	0.7	42.86	42.86

Table 9.16: Hotel and flight booking in parallel order - Pattern 3

Task	Cost	Reliability	Business cost of the task	Business cost
Send confirmation	10	0.9	11.11	11.11

Table 9.17: Hotel and flight booking in parallel order - Pattern 4

Task	Cost	Reliability	Business cost of the task	Business cost
Pattern 2 || 3	50	0.49	71.43	71.43

Table 9.18: Hotel and flight booking in parallel order - Pattern 2 —— Pattern 3

Task	Cost	Reliability	Business cost of the task	Business cost
Pattern 1	25	0.81	29	29
Pattern 2 || 3	50	0.49	71.43	130.61
Pattern 4	10	0.9	11.11	156.23

Table 9.19: Hotel and flight booking in parallel order

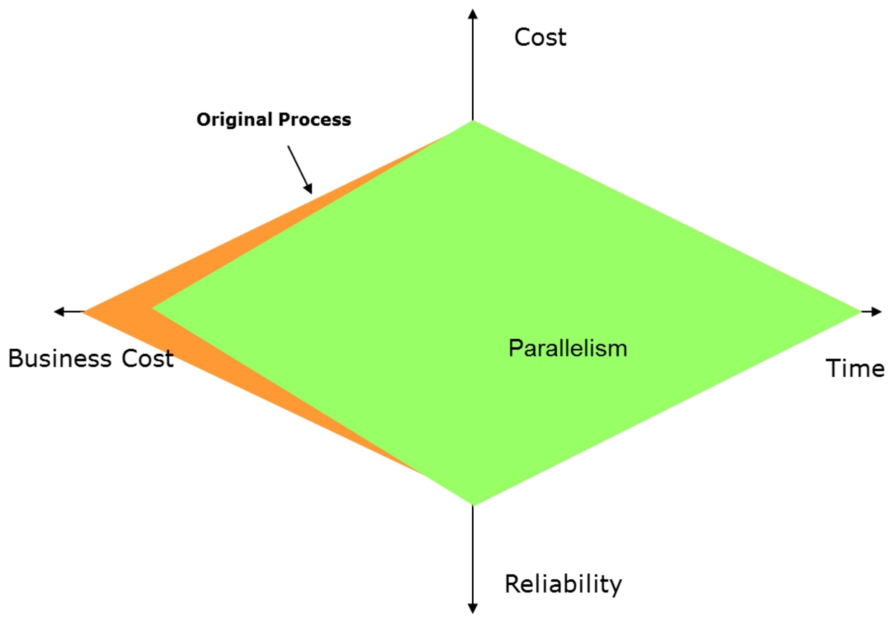


Figure 9.18: Devils quadrangle before and after implementation

$$Cost(Changed\ order) = 85 \tag{9.47}$$

$$Business\ cost(Changed\ order) = 156.23 \tag{9.48}$$

$$Reliability(Changed\ order) = 0.357 \tag{9.49}$$

9.6.3 Impact of Parallelism

We see from the calculations that the business cost of the process decreased by executing a process in parallel when compared to executing it in a sequential order. The cost and reliability remains the same. This is shown in the devils quadrangle as shown in Fig.9.18. However when a process runs in parallel, due to the complexities, more tasks will be needed to make the process reach a logical end. These include tasks such as compensation. For example, in the business process that we considered, we will need a compensation task in case only the flight or only the hotel is booked. The compensation will have to then cancel the other booking. These situations do not arise when tasks are done in a sequential order.

9.7 Conclusion

In this chapter we have implemented the most commonly recommended best practices on business processes. We have evaluated these processes on their cost, business cost and reliability as parameters before and after implementing the best practices. We see through this evaluation that these best practices achieves optimization on a case by case basis. With our methodology it is possible to evaluate the implementation of these best practices in a measurable way.

10

Conclusion

The intention to control cost and achieve profitability is an extremely prominent driving factor within any organization. The aim of every business is to design and implement processes which are financially profitable and reliable.

The field of interest to manage business processes efficiently is vast. Even though it has achieved prominence in the recent years under the topic 'Business process management', research and contributions in this field exist from much before. A large part of the existing research in this field concentrates on management and optimization in a top-down manner i.e. a process at an operational level is seen as a part of a bigger enterprise and the intent is to bring in a performance betterment in the whole enterprise and in turn in the process as well. Financial evaluation through a bottom-up approach i.e. of tasks within a process at an operational level and their impact on the overall goals of the organization has not been much of a priority.

This research aims to fill this gap. This research proposes a methodology to financially evaluate and optimize business processes, put together as business process diagrams with business process model and notations. This is done by calculating the three parameters cost, business cost and reliability of the process. The methodology is based on recognition of patterns and pattern based cost calculation to achieve this. The repetitive patterns are recognized and a cost calculation for each pattern is achieved. The patterns in the process are interleaved with each other and thus the cost of the process is calculated from the costs of the interleaving patterns. The approach considers the reliability and cost of the artifacts in the business process diagram as the basis for calculating the business cost; the cost to achieve one successful execution cycle of the business process.

The basic control flow patterns which are considered in the research show that the cost of a process can be calculated by simple formulas. The research derives these formulas mathematically and implements these on case study. A theoretical concept to model process behavior according to patterns and the impact on the financial parameters has also been proposed. A sensitivity analysis is implemented to show the impact of parameter variation of each task on the overall cost of the process. Through this approach the optimization activities on cost based parameters can be localized to particular activities of the process. The research is backed by an implementation.

With this calculation methodology as basis, the research has evaluated the most commonly recommended best practices in business processes. The evaluation determines the impact of the best practice on cost, business cost and reliability before and after implementing the best practice. We see through this evaluation that these best practices achieve financial optimization, nevertheless the variation and the impact on the parameters cannot be generalized. These are dependent on the process and the complexity that the process is handling. We also see that in certain cases the implementation of the best practice does not lead to any betterment of the reliability of the process, instead it could become more complex to control them.

The approach proposed in this research provides a for a strong foundation for future tasks in the field of defining and developing business processes by taking the reliability of the process into consideration. This would play an important decision making role in designing and developing business processes.

10.1 Future Work

Business process implementation in the real world is still not completely standardized and allows lots of room for individual interpretation. Even though new specifications and standards are expected very frequently, a rule book for designing business processes is not in use. This leads to a situation where the industry is not always up to date with the latest know-how on the best of the design principles which meet their financial targets. This research has evaluated some of the well-known best practices in process design for their impact on costs. To elaborate on the effects, a wider range of best practices needs to be considered. These need to be evaluated with real world business cases and processes which bring in more complicated logics. Examples here could be business cases which include cases of multiple failures and complex compensation mechanisms. Also the wide base of workflow patterns that have already been documented can be used to further develop the calculation methodology proposed in this research.

Also, the application developed as part of this research uses the BPMN editor from Eclipse. A standardized package / plug-in which can be adapted by the different process editors in market will make this the approach much more viable to use.

Bibliography

[1] Bravetti, M., S. Gilmore, et al. (2008). Replicating web services for scalability. Proceedings of the 3rd conference on Trustworthy global computing. Sophia-Antipolis, France, Springer-Verlag: 204-221.

[2] Clark, A. and S. Gilmore (2007). Evaluating quality of service for service level agreements. Proceedings of the 11th international workshop, FMICS 2006 and 5th international workshop, PDMC conference on Formal methods: Applications and technology. Bonn, Germany, Springer-Verlag: 181-194.

[3] Magnani, M. and D. Montesi (2007). "Computing the Cost of BPMN diagrams." Technical Report UBLCS.

[4] Magnani, M. and D. Montesi (2007). BPMN: how much does it cost? an incremental approach. Proceedings of the 5th international conference on Business process management. Brisbane, Australia, Springer-Verlag: 80-87.

[5] Saeedi, K., L. Zhao, et al. (2010). Extending BPMN for Supporting Customer-Facing Service Quality Requirements. Proceedings of the 2010 IEEE International Conference on Web Services, IEEE Computer Society: 616-623.

[6] Bocciarelli, P. and A. D'Ambrogio (2011). A BPMN extension for modeling non functional properties of business processes. Proceedings of the 2011 Symposium on Theory of Modeling Simulation: DEVS Integrative MS Symposium. Boston, Massachusetts, Society for Computer Simulation International: 160-168.

[7] OMG (2012). "Business Process Modeling Notation." Retrieved October 30th, 2012, from http://www.omg.org/spec/BPMN/2.0/.

[8] Garcia-Molina, H. and K. Salem (1987). Sagas. Proceedings of the 1987 ACM SIGMOD international conference on Management of data. San Francisco, California, United States, ACM: 249-259.

[9] Bruni, R., Hern et al. (2005). Theoretical foundations for compensations in flow composition languages. Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages. Long Beach, California, USA, ACM: 209-220.

[10] Clark, A. and S. Gilmore (2007). Evaluating quality of service for service level agreements. Proceedings of the 11th international workshop, FMICS 2006 and 5th international workshop, PDMC conference on Formal methods: Applications and technology. Bonn, Germany, Springer-Verlag: 181-194.

[11] Cavalieri, S., P. Maccarrone, et al. (2004). "Parametric vs. neural network models for the estimation of production costs: A case study in the automotive industry." International Journal of Production Economics 91(2): 165-177.

[12] Boehm, B. W. (1981). Software Engineering Economics, Prentice Hall PTR.

[13] Brand, N. and H. v. d. Kolk (1995). "Workflow Analysis and Design." Deventer: Kluwer Bedrijfswetenschappen.

[14] Davenport, T. H. (1993). Process innovation: reengineering work through information technology, Harvard Business School Press.

[15] Davenport, T. (1990). "The New Industrial Engineering: Information Technology and Business Process Redesign." Sloan Management Review 31(4): 11-27.

[16] Gamma, E., R. Helm, et al. (1995). Design patterns: elements of reusable object-oriented software, Addison-Wesley Longman Publishing Co., Inc.

[17] Gerrits, H. (1994). Business modeling based on logistics to support business process re-engineering. Proceedings of the IFIP TC8 Open Conference on Business Process Re-engineering: Information Systems Opportunities and Challenges, Elsevier Science Inc.: 279-288.

[18] Hammer, M. and J. Champy (2003). Reengineering the corporation: a manifesto for business revolution, HarperBusiness Essentials.

[19] Harrington, H. J. (1998). "Performance improvement: the rise and fall of reengineering." The TQM Magazine 10(2): 69-71.

[20] Triantaphyllou, E. (2000). Multi-criteria decision making methods: a comparative study, Kluwer Academic Publishers.

[21] Daniels, A. C. (2004). Performance Management: Changing Behavior That Drives Organizational Effectiveness, Performance Management Publications.

[22] Reijers, H. (2003). Design and Control of Workflow Processes: Business Process Management for the Service Industry, Springer.

[23] H.A. Reijers, S. L. M. (2005). "Best Practices in Business Process Redesign: an Overview and Qualitative Evaluation of Successful Redesign Heuristics." The International Journal of Management Science.

[24] Zachman, J. A. (1987). "A framework for information systems architecture." IBM Syst. J. 26(3): 276-292.

[25] Manganelli, R. L. and M. M. Klein (1994). The reengineering handbook: a step-by-step guide to business transformation, AMACOM.

[26] Carr, D. K. and H. J. Johansson (1995). Best practices in reengineering: what works and what doesn't in the reengineering process, McGraw-Hill.

[27] Grover, V., S. R. Jeong, et al. (1995). "The implementation of business process reengineering." J. Manage. Inf. Syst. 12(1): 109-144.

[28] Jaideep Motwani, A. K., James Jiang, Mohamed Youssef (1998). "Business process reengineering: A theoretical framework and an integrated model." International Journal of Operations Production Management 18(9/10): 964-977.

[29] Glykas;, G. V. M. (1999). "Critical review of existing BPR methodologies: The need for a holistic approach." Business Process Management Journal 5(1): 65-86.

[30] Alter, S. L. (1995). Information Systems: A Management Perspective, Benjamin-Cummings Publishing Co., Inc.

[31] Berio, G. and F. Vernadat (2001). "Enterprise modelling with CIMOSA: Functional and organizational aspects." Production Planning and Control 12(2 SPEC): 128-136.

[32] Jablonski, S. and C. Bussler (1996). Workflow Management: Modeling Concepts, Architecture and Implementation, International Thomson Computer Press.

[33] W. M. P. van der Aalst, P. J. S. B.(2001). Beyond workflow management: product-driven case handling. Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work. Boulder, Colorado, USA, ACM: 42-51.

[34] Seidmann, A. and A. Sundararajan (1997). "The effects of task and information asymmetry on business process redesign." International Journal of Production Economics 50(2-3): 117-128.

[35] Jablonski, S. (1994). "MOBILE: A Modular Workflow Model and Architecture." 4th International Working Conference on Dynamic Modelling and Information Systems.

[36] Aalst, W. M. P. v. d (2000). Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques. Business Process Management, Models, Techniques, and Empirical Studies, Springer-Verlag: 161-183.

[37] Reijers, H. A., S. Limam, et al. (2003). "Product-Based Workflow Design." J. Manage. Inf. Syst. 20(1): 229-262.

[38] Rupp, R. O. and J. R. Russel (1994). "The Golden Rules of Process Redesign." Quality Progress(12): 85-90.

[39] Castano, S., V. D. Antonellis, et al. (1999). "A methodology and tool environment for process analysis and reengineering." Data Knowl. Eng. 31(3): 253-278.

[40] van der Aalst, W. and K. van Hee (2002). Workflow Management: Models, Methods, and Systems (Cooperative Information Systems), The MIT Press.

[41] Zapf, M. and A. Heinzl (2000). Evaluation of Generic Process Design Patterns: An Experimental Study. Business Process Management, Models, Techniques, and Empirical Studies, Springer-Verlag: 83-98.

[42] Goverde, R. a. R., H; (1998). "Resource Management: a Clear-Headed Approach to Ensure Efficiency." Workflow Magazine 4:6: 26-28.

[43] Aalst, W. M. P. v. d. (2001). "Re-engineering knock-out processes." Decis. Support Syst. 30(4): 451-468.

[44] Poyssick G, Hannaford S. Workflow Reengineering. Mountain View:Adobe Press Editions 1996

[45] Kees Van Hee, H. R., Eric Verbeek, Loucif Zerguini (2001). "On the Optimal Allocation of Resources in Stochastic Workflow Nets." Proceedings of the Seventeenth UK Performance Engineering Workshop 23-34.

[46] Suermeli;, C. G. J. (2010). Estimating Costs of a Service. Proceedings of the 2nd Central-European Workshop on Services and their Composition, ZEUS 2010, Berlin, Germany, February 25–26, 2010. C. G. J. Suermeli;, CEUR-WS.org.

[47] Gunasekaran, A., H. B. Marri, et al. (2002). "E-commerce and its impact on operations management." International Journal of Production Economics 75(12): 185-197.

[48] Seidmann, A. and A. Sundararajan (1997). "Competing in information-intensive services: analyzing the impact of task consolidation and employee empowerment." J. Manage. Inf. Syst. 14(2): 33-56.

[49] Fan, M., J. Stallaert, et al. (2000). "The adoption and design methodologies of component-based enterprise systems." Eur. J. Inf. Syst. 9(1): 25-35.

[50] Hieleber R, Kelly T, Ketterman C (1998). Best Practices. Simon and Schuster

[51] Pree, W. and H. Sikora (1997). Design patterns for object-oriented software development (tutorial). Proceedings of the 19th international conference on Software engineering. Boston, Massachusetts, United States, ACM: 663-664.

[52] Engelke, H., J. Grotrian, et al. (1985). "Integrated manufacturing modeling system." IBM J. Res. Dev. 29(4): 343-355.

[53] Hammer, M. (1990). "Reengineering Work: Don't Automate, Obliterate." Harvard Business Review: 104-112.

[54] C. Adams and A. Neely. Prism Reform. Financial Management 2002

[55] Roy, R. (2003). Cost engineering: why, what and how? Decision Engineering Report (DEG) Series.

[56] BusinessDictionary.com (2013). "performance" Retrieved 5th March, 2013, from http://www.businessdictionary.com/definition/performance.html.

[57] Wiktionary (2012). "Cost." Retrieved 30th October, 2012, from http://en.wiktionary.org/wiki/cost.

[58] Dictionary, T. F. (2012). "Cost." Retrieved 30th October, 2012, from http://www.thefreedictionary.com/cost.

[59] Interglot (2012). "Cost." Retrieved 30th October, 2012, from http://www.interglot.com/dictionary/en/en/translate/cost.

[60] Wikipedia (2012). "Cost." Retrieved 30th October, 2012, from http://en.wikipedia.org/wiki/Cost.

[61] R, S. (1998). Safety Critical Systems Analysis. Dependable Embedded Systems, Carnegie Mellon University.

[62] Cooper R, Kaplan R (1988). Measure Costs Right: Make the Right Decisions-ABC. Harvard Business Review September-October. 96 - 103

[63] Fitzgerald, L. and C. I. o. M. Accountants (1991). Performance measurement in service businesses, Chartered Institute of Management Accountants.

[64] Longo, A. and G. Motta (2006). Design processes for sustainable performances: a model and a method. Proceedings of the Third international conference on Business Process Management. Nancy, France, Springer-Verlag: 399-407.

[65] Andy Neely, M. G., Ken Platts (2005). "Performance measurement system design: A literature review and research agenda." International Journal of Operations and Production Management 25(12): 1228 - 1263.

[66] Mariska Netjes, H. A. R. (2005). Analysis of Resource-Constrained Processes With colored Petri Nets. Department of Computer Science, University of Aarhus: 251 - 266

[67] Reijers, H. (2003). Design and Control of Workflow Processes: Business Process Management for the Service Industry, Springer.

[68] Zapf, M. and A. Heinzl (2000). Evaluation of Generic Process Design Patterns: An Experimental Study. Business Process Management, Models, Techniques, and Empirical Studies, Springer-Verlag: 83-98.

[69] Peppard, J. and P. Rowland (1995). The essence of business process reengineering, Prentice Hall.

[70] Klein M (1995). 10 Principles of Reengineering. Executive Excellence

[71] Cross, K. F. and R. L. Lynch (1988). "The SMART way to define and sustain success." National Productivity Review 8(1): 23-33.

[72] D.P. Keegan, R. G. E., C.R. Jones (1989). "Are your Performance Measures obsolete." Management Accounting 70(12):45(50).

[73] H.A. Reijers, S. L. M. (2005). "Best Practices in Business Process Redesign: an Overview and Qualitative Evaluation of Successful Redesign Heuristics." The International Journal of Management Science.

[74] Slack, N. (2005). "The flexibility of manufacturing systems." International Journal of Operations Production Management 25(12): 1190 - 1200.

[75] A. De Toni, S. T. (2001). "Performance measurement systems - Models, characteristics and measures." International Journal of Operations Production Management 21(1/2): 46 - 71.

[76] Alter, S. L. (1995). Information Systems: A Management Perspective, Benjamin-Cummings Publishing Co., Inc.

[77] Berg A, Pottjewijd P (1997). Workflow:Continuous Improvement by Integral Process Management. Schoonhoven Academic Service

[78] AMICE., E. C. (1993). CIMOSA: open system architecture for CIM, Springer-Verlag.

[79] Kaplan, R. S. and D. P. Norton (1996). The Balanced Scorecard: Translating Strategy into Action, Harvard Business Press.

[80] (1988). Measure Costs Right: Make the, Harvard Business School Reprint.

[81] Fitzgerald, L., Johnston, R., Brignall, S., Silvestro, R. and Voss, C (1991). Performance Measurement in Service Business. CIMA, London

[82] Kaplan, R. S. and D. P. Norton (1993). Putting the Balanced Scorecard to Work, Harvard Business Review OnPoint.

[83] Kaplan, R. S. and D. P. Norton (1996). The Balanced Scorecard: Translating Strategy into Action, Harvard Business Press.

[84] Lynch, R. L. and K. F. Cross (1993). Measure Up!: The Essential Guide to Measuring Business Performance, Mandarin.

[85] Bruni, R., Hern et al. (2005). Theoretical foundations for compensations in flow composition languages. Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages. Long Beach, California, USA, ACM: 209-220.

[86] Sampath, P. and M. Wirsing (2009). Computing the Cost of Business Processes. In: Jianhua Yang, Athula Ginige, Heinrich C. Mayr, Ralf-Detlef Kutsche (Eds.): Information Systems: Modeling, Development, and Integration, Third International United Information Systems Conference, UNISCON 2009, Sydney, Australia, April 21-24, 2009. Proceedings. Lecture Notes in Business Information Processing 20, Springer 2009, 178-183.

[87] Sampath, P. and M. Wirsing (2011). Evaluation of Cost Based Best Practices in Business Processes In: Terry A. Halpin, Selmin Nurcan, John Krogstie, Pnina Soffer, Erik Proper, Rainer Schmidt, Ilia Bider (Eds.): Enterprise, Business-Process and Information Systems Modeling - 12th International Conference, BPMDS 2011, and 16th International Conference, EMMSAD 2011, London, UK, June 20-21, 2011. Proceedings. Lecture Notes in Business Information Processing 81, Springer 2011, 61-74

[88] Partha Sampath, Martin Wirsing: Financial Evaluation and Optimization of Business Processes. To appear in Special Issue of BPMDS/EMMSAD 2011, International Journal of Information System Modeling and Design (IJISMD).

[89] Evans, Lanham, Marsh. Cost Estimation Method Selection: Matching user Requirements and Knowledge Availability to Methods. Systems Engineering and Estimation for Decision Support (SEEDS) Group, University of West of England, Bristol, UK

[90] Fourneau, J.-M., Le, et al. (2006). A precedence PEPA model for performance and reliability analysis. Proceedings of the Third European conference on Formal Methods and Stochastic Models for Performance Evaluation. Budapest, Hungary, Springer-Verlag: 1-15.

[91] Jeremy T. Bradley, N. J. D., Stephen T. Gilmore, William J. Knottenbelt (2003). Extracting passage times from PEPA models with the HYDRA tool: A case study. UNIVERSITY OF WARWICK.

[92] Gilmore, S., J. Hillston, et al. (2002). PEPA Nets: A Structured Performance Modelling Formalism. Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools, Springer-Verlag: 111-130.

[93] Bajwa, I. (2010). "Markov Logics Based Automated Business Requirements Analysis." International Journal of Computer and Electrical Engineering (IJCEE) 2(3): 481-485.

[94] Kok, S. and P. Domingos (2005). Learning the structure of Markov logic networks. Proceedings of the 22nd international conference on Machine learning. Bonn, Germany, ACM: 441-448.

List of Figures

List of Tables

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Sampathkumaran, Partha B.

Name, Vorname

München, 08.01.2013

Ort, Datum

Unterschrift Doktorand/in