SERVICE IDENTIFICATION FOR BUSINESS PROCESS MANAGEMENT

Bournemouth University

Olaolu Sofela

Faculty of Science and Technology

Bournemouth University

This dissertation is submitted for the degree of

Doctor of Philosophy

December 2016

I would like to dedicate this thesis to my loving parents and nephew (*Emmanuel*).

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Olaolu Sofela

December 2016

Acknowledgements

First and foremost, I wish to start expressing my deepest gratitude to God Almighty for guidance and quest for knowledge discovery.

I would like to express my profound gratitude to my first supervisor, Dr Lai Xu for the guidance and support. Lai gave me the chance to enter the world of research. I also truly appreciate her insightful comments, timely, professional and personal advice during my entire time in Bournemouth University. I thank Lai for sharing endless supply of her ideas. I cannot be proud of my work without thinking of her entralling support. Thank you Lai.

The guidance of my second supervisor, Dr Paul de Vrieze is highly appreciated. Paul has given his indispensable and ursurpassed knowledge during my PhD. Thank you Paul

Moreover, I would like to say thanks to Dr Edward Apeh, Alimohammad Shahri and other colleagues in Bournemouth Unniversity for their professional assistance when needed. Last, but no mean least, I want to show appreciation to my sponsor, Mr and Mrs Sofela. Without this financial support, this PhD study would have been impossible. I love you.

I am forever grateful to my parents Mr and Mrs Sofela, for the moral support they have given to me, their understanding and love has always got me through difficult times over the last four years. I owe a huge debt of gratitude to my mother Elizabeth for the continuous prayers, encouragement and understanding. To my father Abiodun, his faithful and unconditional support aspire me to continue to work harder.

Abstract

Over the years Service Oriented Architecture (SOA) has gained momentum and is becoming the standard for providing systematic business solutions. Likewise, the requirements for identifying business services are fast changing and a solution to the service identification problem needs a robust approach. It is known that this task of identifying candidate services is the first and the most important step in developing service-oriented business systems. The recent approaches of identifying candidate services have some shortcomings (*defined data type size, unrepeatable approach, inapplicable to all enterprise information system and unadaptable to business factor change*). Some approaches focus on fixed cases or certain types of organizations (*single or collaborating organizations*) neglecting the enterprise systems which are either (*open or closed*) single or collaborating enterprise information system, which makes some past approaches not applicable to some real-life business cases. This thesis focuses on solving the headline issues and introduces a new approach for service identification applicable to different organization's business processes. The thesis also proposes a new step-by-step algorithm and methodology that identify business services derived from data-set from any given business case.

Acronyms

AND parallel Gate

A2A Application-to-Application

BPM Business Process Management

P2P Person-to-Person

P2A Person-to-Application

SOA Service Oriented Architecture

SOC Service Oriented Computing

OR Inclusive Gate

XOR Exclusive Gate

(SI) Service Identification

SOBPM Service-Oriented Business Process Management

WFM Workflow Management

IT Information Technology

B2B Business-to-Business

B2C Business-to-Customer

BPMS Business Process Management System

BIA Business IT alignment

BIT Business Information Technology

SIM Service Identification Methods

BPMN Business Process Management Notation

ERP Enterprise Resource Planning

HRM Human Resource Management

CRM Customer Relation Management

EIS enterprise information system

VO Virtual Organization

DbSIF Data-based Service Identification Framework

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Enterprise architecture (EA[1]) was proposed in 2005 to allow organisations that adopt it to reduce IT systems complexity and to improve the alignment of their business with (Information Technology (IT)), resulting into reduction in organization's cost (Alves et al., 2013; Lankhorst, 2005).

Lampe et al. (2013); Schulte et al. (2014) highlight "....15% to 20% of banks' overall administrative expenses are attributed to the IT cost. This arguably much more than 15% to 20% as banks are reducing staff (therefore, increase spending on IT even if the IT is efficient). Furthermore, Mai (2012) presents the banks overall IT spending as followings, 1% on overhead, 24% on infrastructure , 45% on application operations and maintenance and 29% on application development, corresponding to last twelve years of operation, all these figures sum up to 70% cost. DB research made a study of the IT cost on various continent, it states that IT costs differ substantially, ranging from USD 270 billion to USD 460 billion for their 2013 budgets, resulting to 7.3% of their revenues, as found by Forrester Research Inc (Mai, 2012). Therefore, it can be said that majority banking administrative expenses are partially classified as IT cost.

[1]Lankhorst (2005, p.3) defined EA as" a coherent whole of principles, methods and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure"

Aside from cost, financial institutions have to fulfil the banking regulatory requirements which contributes to increased IT costs. In other sectors like manufacturing and e-retail, the overall administrative expenses will be higher than financial sector because of lack of developing and supporting tools for extensive support for simulation, business intelligence, business re-use, case management and many more.

The aim of this research is to investigate potential approaches to reduce the overall cost on IT through reduction on process model wastage, i.e. (*specific process model only last for a purpose*). Organizations need to address the wastage of process model by identifying possible reusable services from the old (single or collaborative) process models. Therefore, banks‘ or any organizations‘ IT cost can be highly reduced when services are identified with the right[2] size of granularity. Also, IT cost can be reduced if their processes are service-oriented and they can outsource for other people to maintain, because they system is easy to integrate and they can pay-per-use.
This thesis explores various weaknesses of current methods of service identification and studies how these can be addressed to improve business services and reduce cost.

1.1 Problem Background

In modern time, organizations are still exploiting every means of improving their services and reduce cost, even though there are so many works and frameworks with exhaustive approaches that has been introduced into improving business services. Several numbers of used and wasted process models cannot be accounted for every time requirement and organization goal change.
In this regard, there is an evident necessity for new approaches and guidelines in achieving re-usability of business services which are adaptable to “change factor." As an attempt to

[2]The use of the word “right” is the best terminology for this domain which satisfies the technical metric (low coupling and high cohesion) and managerial metrics (low reuse cost principles)

address the identified problem above, the main question that is needed to be answered is: How can business services with large collections of process models be (defined, analysed, evaluated) to suite the future facilitation improvement and development of new collaborative process models, providing new challenges, enhancement and opportunities? This question has been sub-divided into several sub-questions:

-What are the requirements and guidelines for service identification in the context where service reuse in case of change is important?

-How does right identification of services improve business process variability?

1.1.1 Motivation

This work is based on the current knowledge of service identification which are less flexible and applicable to any business instance. The identification of business services with right sizes of granularity are important because (i.) too small size of services requires too much time to compose and achieve any interesting functions, and (ii.) too big size of services required might not be reusable which is against the principle ofSOA. Therefore, the "right-size" of business service has to be identified which fits for purpose in every business instance. Also, industries are having fewer requirements to migrate their legacy systems into service oriented systems, because there is no formal (semi)-automatic approach to identify services for supporting and building business process management systems (BPMS).

1.2 Research Aims and Objectives

Our work will explore the need to improve business process flexibility and variability, in business service. The result will motivate organizations to achieve great business goals and interest. Specifically, the research will provide the following:

- create an improved dynamic and adaptive business process system, a system with high granularity that will not just create higher complexity in any business governance but also improve process exchange, calls between services and collaboration between inter or intra-organization.
- To develop a design method that is able to identify large amount of service using (semi)-automatic approach. There is no unified approach or method in identifying services, which the industries need to implement SOA systems. This research provides a clear understanding and recommendations in achieving the right service identification method for supporting and building BPMS.
- a guideline, approach and recommendation for the design of context-adaptive business process, in both imperative and declarative way.

1.3 Research Methodology

In this thesis, we are focused on using both qualitative and design science knowledge, as addressed in each sub-sections, as it is obvious from the research questions that broad knowledge of the background problems which aims to improve and give a framework to generate candidate business services from any organization's requirement and business process model.

We follow a new methodology in achieving a new framework to generate candidate business services and also measuring the alignment of services with business goals. The goal of new approach in identifying business process is to improve business agility which reduces complexity and redundancy of services.

Several researchers have focused on high re-usability, capability, entity-centric approaches recently. The drawback of entity-centric approach is that it does not show the internal view of business processes. In our approach, we will focus on the internal representation and view of business process called data-centric approach. This approach can early adapt to requirement change and further improve on re-usability, and capability. The rest of the paper is organized as follows. Section 3.2 presents and overview of the adopted methodology. Section 3.3 presents the goal-, model- and data-based principles and their characteristics. This will guide into choosing the requirement for the best principle for building new approach in detail in an attempt to answer the research questions. Section 3.4 summarises the chapter. The methodology used in this research is called *design-oriented or -science research* Verschuren and Hartog (2005), Hevner et al. (2004)

Design science creates and evaluates IT artefacts intended to solve identified organizational problems. Such artefacts are represented in a structured form that may vary from software, formal logic, and rigorous mathematics to informal natural language descriptions. Design science addresses research through the building and evaluation of artefacts designed to meet the identified business need.

Also, the motives of the research methodology give deep understanding of the research questions (*see Section 1.1*) and how to solve them (*see Chapter 4*). In this thesis, we present the step-by-step directions for developing a new approach in achieving the research objectives (see Section 1.2). Moreover, we use, analyse and evaluate a given case study to demonstrate the practicality of the new approach (*see Chapter 4*)

In achieving the objectives, the following steps are taken. Firstly, we reviewed several numbers of publications (*see Section 2.5 and 2.6*), which gives qualitative knowledge needed in understanding and exploring new method. Secondly, in creating a new method, we use an exploratory study of different concepts namely data derivation and dependencies in Business Process Management (BPM) and documents (files), proto-service clustering algorithm, and service identification algorithm. Figure 1.1 carefully illustrates the phases in creating a new method which includes the foundation, initial design, proposition, design and prototype, validation, and evaluation and reflection phases, as shown in the next subsections, each of these phases are elaborated on.

1. Phase I -The foundation

   This phase focuses on the initial definition of the background issues and find the related research questions that will be suitable for creating the “right" services. Section 2.6 largely demonstrates the initial work that has been done in service identification in business process management and software development. Our aim is to create a dynamic and adaptive business services with less complexity in any business governance and improved process/data exchange, calls between services and collaboration between inter or intra-organization. In devising the aim, some research questions are defined to assist in creating the methodological approach in achieving the research aim.

   In Figure 1.1, each of the phases illustrates the objective of the research. The completion of the foundation phase leads to the proposition phase which we explain in the following sub-section.

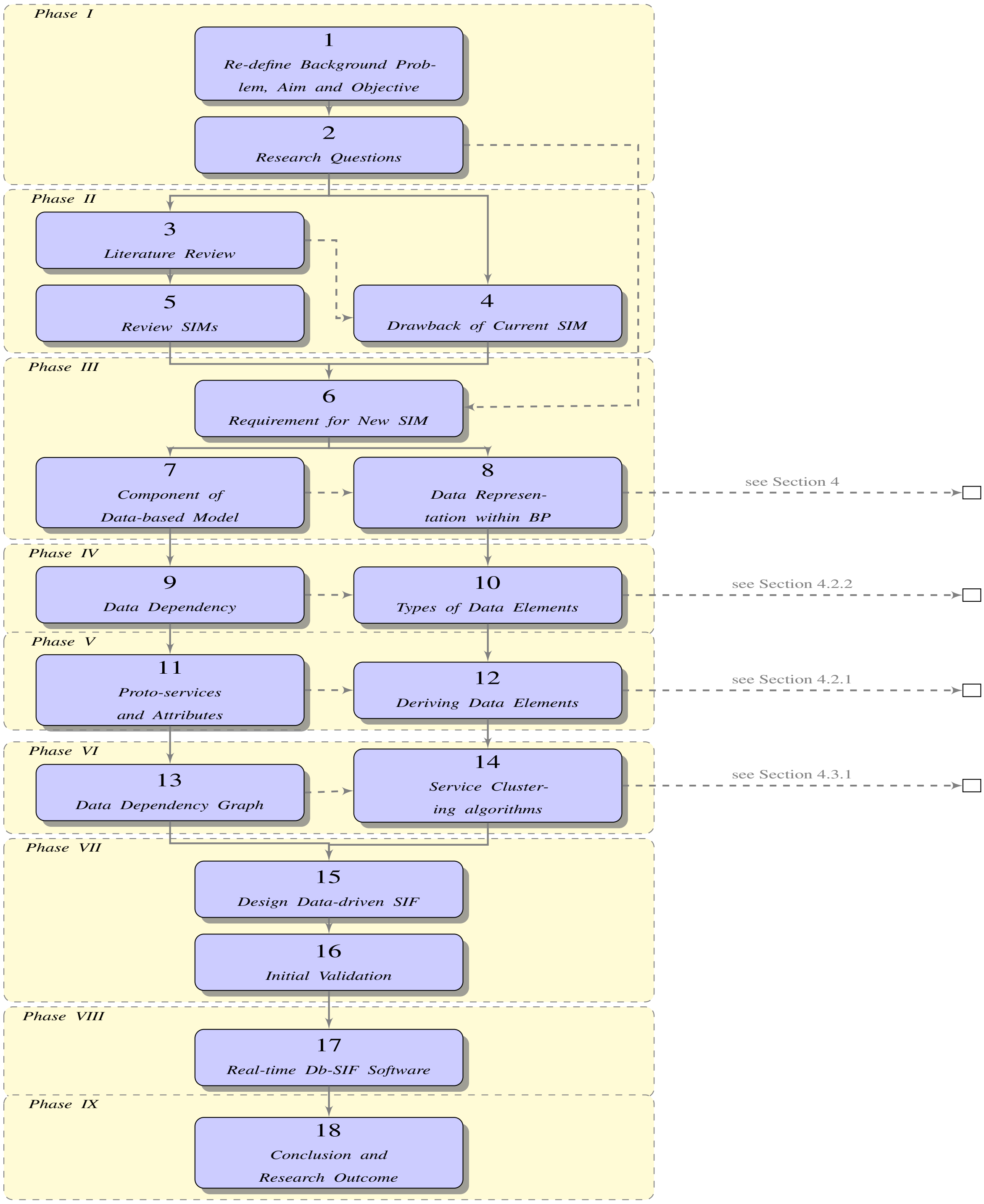


Fig. 1.1 Research Methodology

2. Phase II -The proposal

After defining the research questions and investigating the related works as stated in foundation phase, we propose the possible steps in giving solutions to the research questions which are classified into four activities as shown in Figure 1.1. The solution to this phase give us the requirement for designing business service identification methods for any enterprise information systems and Service-Oriented Business Process Management (SOBPM). These activities are the exploratory study, the past and recent researches on service identification, and the drawbacks of the current Service Identification Methods (SIM).

A full exploratory study on related works is shown in Chapter 2 where we reviewed with description and understanding of the following concepts namely: BPM, SOA and SOBPM, demonstrating each concept's challenges.

The initial finding phase comprises of reviewed literature comprising of their techniques, approaches and methods of identifying both software and business services. Furthermore, we compared all the techniques, approaches and methods using certain criteria (*see Section 2.6*).

The finding and comparisons deduce the drawbacks which gives us the understanding of the current techniques and new development. Based on these drawbacks of the existing methods and the comparison generated, the weaknesses of the current service identification approach (*value-based service identification, goal service modelling, domain decomposition, asset analysis, Use case analysis, and goal service modelling approaches etc.*).

Therefore, in developing new service identification methods, there are series of requirements that are considered (*see Section 2.6.3*). The completion of the initial finding phase leads to the initial design phase which we explain in the following chapters. The rest of the phase are shown in the diagram and it is self-explanatory.

1.4 Major Contribution of the Thesis

There are several approaches introduced into industry that the are static i.e. they are built for a tailored case. Our work is driven by both the academic and industry and will contribute in

1. Understanding the notion, importance of applicability of service identification in enterprise information system.

2. Creating new requirement for identifying business service in enterprise information system

3. Creating a data-oriented method with repeatable approach and undefined data type size

1.5 List of Publications

1. Sofela O, Xu, L and De Vrieze, P Context-Aware Process Modelling through Imperative and Declarative Approach. 114th IFIP WG 5.5 Working Conference on Virtual Enterprise. pp 191-200, 2013

2. Sofela, O., Xu, L. and De Vrieze, P., 2016. Needs of Service Identification for Service-Oriented Business Process Management. In: SQM 2016: 24th International Software Quality Management Conference, 21-22 March 2016 Bournemouth, UK.

3. Sofela, O., Xu, L. and De Vrieze, P., 2016. Service Identification Requirements for Enteprise Information Systems. In: SKIMA: 10th International Conference on Software, Knowledge, Information Management and Applications Conference. China.

4. Sofela, O., Xu, L. and De Vrieze, P., 2017, Service Identification Framework for Enterprise Information System. (Awaiting Journal Publication)

1.6 Thesis Structure

This thesis is structured as follows,

- Chapter 1 describes the research aims and objectives, the research questions and the methodology used to achieve the objectives.
- Chapter 2 provides a state-of-art review of the key notions of BPM research and also an overview of the most relevant existing methods and approaches used in service identification.
- Chapter 3 investigates requirements of service identification from different types of information systems, from single systems to collaborative systems, from closed systems to open systems. The research is important for providing a solid foundation for further identifying services for developing different service-oriented systems. The chapter clearly state the importance of distinguishing the methodology to be used for every involvement (i.e. single system or collaborative). Same methodology cannot be used to identify services in all systems or organization. Therefore, organizations using different enterprise information systems can have different involvement and operations. In solving the issue, we introduce two-dimensional enterprise information system, one dimension shows the representation of organization system, i.e. "involvement" dimension and the other dimension represents operation of organizations which can be seen as open or closed for other partners, i.e."operation" dimension. For any the enterprise information systems in any axis or dimension, certain service orientation are defined which further assist in business service identification with the right granularity. In doing so, we state the requirement for service identification for each dimension.
- Chapter 4 introduces the data derivation within business processes model or document in single and collaborative systems and presents the data-driven service identification framework.

- Chapter 5 presents the evaluation procedure for data-based service identification framework using the quality metrics.

- Chapter 6 provides the conclusions that arise from the work done in each chapters and formulates the future direction for the integration of the framework into making organizations realize their potentials.

Chapter 2

Literature Review

The rise and inclusion of information technology into organization has played significant roles from the initial design stage, day-to-day running of the business processes and operating in a common business domain which align business needs with their systems' needs to burst profit and compete and survive in their domain.

In any organization, the *business needs* have to properly defined which involves stating business goals, desired quality requirements[1] and undergo “metabolic change" by studying and managing customers and potential prospect.

After stating the business needs, the *system needs* also need to defined. This involves the analyzing the non-functional and functional requirements. From defined the business and systems needs, the step-by-step representation and alignment can expressed by using UML models , business process models and many more models.

The rapid movement towards service orientation has further aligned business-system needs and technology to offer unprecedented opportunities. In this chapter, the review of strategic alignment of service orientation with business process management to create desired, efficient and re-usable candidate business services which are cost efficient are provided

Section 2.1 discusses business process management in general. Section 2.2 presents business

[1]What makes a successful and profitable organization stand out from average one is the quality of service

architecture and modelling approaches. The service-oriented computing and its principles are presented in Section 2.3, the service oriented BPM are presented with the current approaches and methods for identifying business services in Section 2.4, and Section 2.5, several researches related to service identification are reviewed and compared.

2.1 Business Process Management

Business process management is a 3-dimensional concept which involves "*business*", "*process*" and "*management*". Each of these dimensions can be combined, presenting another term. The combination can results to business process, business management(Zimmerer et al., 2008) and process management(Benner and Tushman, 2003) (*cf.figure 2.5*). In this thesis, we focus on business process, management of business processes, and related topics. First, the management of business processes involves conversion of business requirements to unprecedented ideas which produces predictable output. The input and output are modelled to demonstrate the step-by-step understanding of the system. This resulted to the evolvement of the concept of BPM. BPM has gained momentum, recognition and contributed to the world of computer science, and management of corporate assets.

Secondly, there have been several definitions of BPM based on performance, management practice, structural and IT roles. Dumas et al. (2013) defines BPM as the art and science of orchestrating the organizational duties to produce and manage the consistent outcomes and take advantage of improvement opportunities.

According to Weske (2009), business process management is said to be concepts, methods and techniques that support the design, administration, configuration, enactment and analysis of business process. Whereas, Aburub and Almahamid (2010) defined BPM as a concept that provides a generalized knowledge and view of what business process, combining a set of activities or task within an enterprise with a structure showing their logical order and dependence. Aburub and Almahamid (2010); Weske (2009) definitions highlight the

management and structural practice without stating the user(s).
However, Ko et al. (2009) pinpointed the involvement of humans, organizations, applications, documents and information in the designing, enacting, controlling and analysing of operational processes. Antunes and Mourão (2011) definition shows more attention on IT role. Defining BPM as a concept that integrate a collection of technologies which are capable to translate business process models into computer-supported activities, ceasing the sequence of managerial action and control tasks. Verner (2004) buttress Antunes and Mourão (2011) points by presenting the connectivity of BPM with IT. Verner (2004) further state that BPM technology provides not only the tools and infrastructure to define, simulate, and analyse business process models, but also the tools to implement business processes in such a way that the execution of the resulting software artefacts are managed from a business process perspective.

In the same wavelength, we can define BPM as closed loop concept or methodology that uses every organizational resources (human interactions, information, applications, IT systems and many more) to iteratively identify, design, develop, deploy, update and manage their day-to-day business process.

Thus, the above definitions emphasize all the characteristics of BPM, business processes at its core.

2.1.1 BPM History

The foundation of business process management lies on the combination of the industrial labour and optimization of information systems. Gillot (2008); Stoilova et al. (2006); van der Aalst (2013), all first referred and mentioned Frederick Winslow Taylor's book "Shop Management" in 1903 for the pioneering management thinking. Afterwards, modern industrial engineering and process improvement was developed by Taylor, which was first used by

Henry Ford in 20th century for the introduction of mass production of “T-Fords”(van der Aalst, 2013).

In the field of information systems, the involvement of computer and its technologies[2] like database or enterprise applications enabled new capabilities to coordinate business activities within organization, the inter-related activities work together to collect, retrieve, store and distribute information for the certain business purpose to produce a product or service, which established joint relationship between the business administrations and information systems. The advancement on information systems and technology came up with the need to store data which led to the creation of database management system (DBMS) like Oracle, resulted to the creation of office automation. Furthermore, the transition of information systems from data-oriented to process-oriented which brought about process-aware information system (PAIS)[3]

Afterwards, many organizations focus on the process-oriented way of managing their business needs, which gave rise to industrial usage of WFM.(Dumas et al., 2013; Harmon, 2010; Mendling et al., 2010; van der Aalst, 2013). WFM[4] came into existence as a result of industrial revolution and the need for controlling, monitoring and automating of business process using information systems like Enterprise Resource planning (ERP) systems, Work flow Management System (WFMS), and Customer Relationship Management (CRM) systems(Stoilova et al., 2006). The adoption of WFM helped to focus on improving the effectiveness and efficiency of business processes within one organization, and its automated and human processes, information document or product created and shared between steps known as *work-flow automation* (Van Der Aalst and Weske, 2013).

[2]The adoption and rapid advancement of Internet technology, business competition and capabilities and the wish for a more evolutionary approach has always triggered the progressive business knowledge, power and quality which changes the way business executes iteratively

[3]PAIS manages and executes operational processes involving users, applications, and/or data sources on the basis of process models (Dumas et al., 2005).

[4] Work flow management (WFM) is a technology supporting the re-engineering and capturing the controlling and coordinating the execution of tasks and providing a (re)design and (re)implementation of the processes as the business needs and information systems changes (Georgakopoulos et al., 1995).

Aalst et al. (2003) review the history of work-flow technology, highlighting the fundamental aim and approach of WFM. The importance of WFM has been historically demonstrated in office automation, email, document management, software management (Zur Muehlen, 2004).

WFM system shortcomings surfaced when organizations lack the ability to support their process flexibility, analysis, reuse and case management (van der Aalst, 2013). The existed WFM systems could not cope with unpredictable and fast changing business system conditions and resources. With organization adopting some key areas of process innovation through the use of internet technologies, e.g. in the area of supply chain and Enterprise Resource Planning (ERP) application, it became transparent that workflow systems could not enable the development of existing ERP applications.

The ideas into BPM was brought about because of the lack of developing and supporting tools for extensive support, for supporting simulation, business intelligence, business-re-use and case management and so forth (van der Aalst, 2013).

In the *1990s*, new innovations and technologies pave the way for BPM and its automation, likewise, business process change was introduced, which was derived from Business Process Re-engineering (BPR) (Smith and Fingar, 2006). The combination of IT and business process redesign transformed organization and improve business process (Verma, 2009).

The incompatibility of different systems during this period created problems. This led to the introduction of communication standards and distributed systems for example eXtended Makeup Language (XML) and competing standards (WSFL, BPEL, BPML, BPDM, etc.). Web services and SOA also contributed into the established research and optimization of business process automation and re-engineering which acted as interchange format specification and design for business process modelling for the re-designing of often ignored human factors proprietary solutions, which mainly focused on operations productivity (Mendling et al., 2005).

Today, organizations and governments all over the world pay attention and contribute to business processes in such a way to improve, automate and manage their processes.

2.1.2 BPM Life-cycle

Business process management can be clearly visualized through the phase of the life-cycle. Clearly, it is not a new phenomenon to have life-cycle in describing the development process of products (business area) and data or processes (information systems). BPM life-cycle will give a bed-eye view of the vision of the phases in processing of the models and also show what directly comes after the other.

Weske (2009) and Ko (2009) highlighted the BPM lifestyle in four phases (Design and Analysis, Configuration, Enactment and Evaluation) to understand the features and terminologies of BPM, while van der Aalst (2013) emphasized that "the scope of BPM extends far beyond the implementation of business processes", describing the life-cycle of BPM in three phases using the two role of model based analysis[5] and data-based analysis[6] namely: (re)design, implementation/configuration, run and adjust.

The BPM life cycle (*cf.fig 2.2*) is adapted from van der Aalst (2013) as this complements other life-cycles Nowak et al. (2011), Ko (2009), Weske (2009) with artefacts which represents how business processes can be managed for better services. The relevance of van der Aalst (2013) life-cycle (van der Aalst, 2013) is that it demonstrate and emphasize on the analysis phase(model-based and data-based analysis) of each phase which are important for verification and validation of models. It involves three core components ((re)design, run and adjust, and implement and configure) where each phase has its own obligation and follows the arrow direction indicating the sequential flow of the phases. Below is the detailed explanation of each phase of the business process management life-cycle.

[5] Consideration of model state, informal model are hard-coded in conventional software, else the model will be in executable form. At design phase, the analysis is used to check for correctness of a new design

[6] Event data are collected while the system is running to discover bottlenecks, waste, and deviations

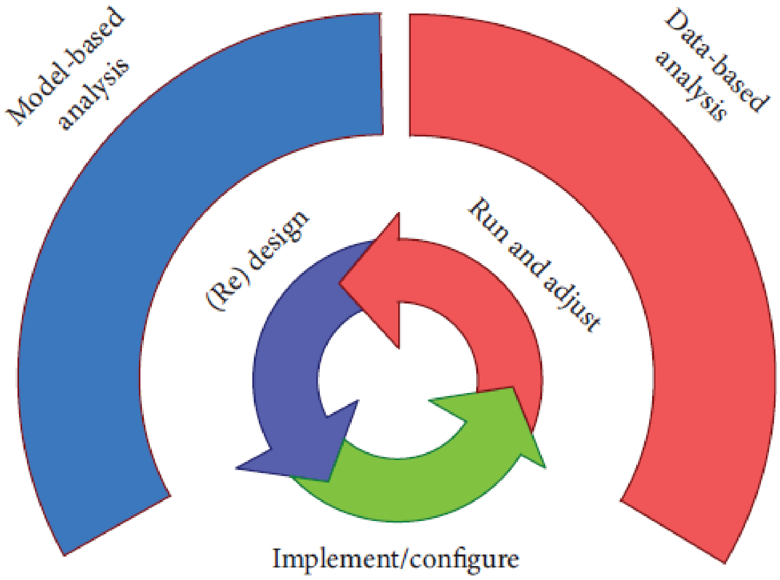


Fig. 2.1 BPM life-cycle (van der Aalst, 2013)

1. (Re)design

   It can be seen from the (*Fig.2.1*) that it has no fixed starting point for the process, it is often common it begins from the design phase. it is a graphical representation of as-in business process phase. Within this phase, first steps is gathering the organization's business goals, polices, sub-policies and rule, then the requirement which will fit for purpose can be identified and served as the input to be modelled. The analysis step comprises of an automated verification (Koehler et al., 2014) of process models which checks for deadlocks and also validation (Combi et al., 2011) which checks for correctness with the business requirement (van der Aalst, 2013) and also check for for soundness (Wynn et al., 2009) known as model-based analysis (Van Der Aalst and Weske, 2013). In phase, business process identification and modelling is also done. Sometimes, this phase can be ignored if there is another way of sourcing its process models, which is faster and easier rather than modelling its requirement from the scratch. Ways of sourcing design model are from event data, merging models, compose models from the repository (van der Aalst, 2013).

2. Implement and configure

   This phase involves the implementation and configuration of process models. The model appear in either executable form or can be informal form i.e., the model is hard-coded in conventional software to initial set of satisfaction. Configuration of process model can either be done by designing the configurable model, merge models into configurable models or configure the configurable model(van der Aalst, 2013).

3. Run and adjust

   The design model is executed on the technical infrastructure to show the resulting modelled business process in which the performance can be analysed at run-time known as data-model analysis, to check the conformance and performance of the process and also identify and improve on bottlenecks loopholes in the process model. The major steps in this phase are: run the model design, refine (adjust) model, log event data, monitor the performance of the resulting process and adapt while running (van der Aalst, 2013). This phase is sub-divided by some authors Li et al. (2011); Saylam and Sahingoz (2013); Van Der Aalst et al. (2011) into monitoring and process mining.

   One of the major work is done at the design stage which involves conversion of business requirement into a process model. This process model is prone to errors and unpredicted changes at any time. In the case of change in the large business process, there are several questions that can be asked. For instance, are we remodelling or redesigning? How long is it going to take? Which all these questions depends on time and cost. This leads to the reason for service identification, which gives solution the questions, time and cost convenience.

2.1.3 Benefits of BPM

BPM allows organizations to create, execute and reuse its business processes from technology innovations and enables them to change their own business quickly according to their business requirements and needs. Weske (2009) presents the benefits of BPM namely: increase in organization flexibility, quick response to changes, improved process asset.

Other benefits are argued by (Workpoint LLC, 2011) to include time reduction, numbers of steps of a process, error cycles and resources.

Also, BPM has increased business process efficiency in maintaining effective regulation compliance in intra or inter-organization. Organization are motivated to use BPM to increase responsiveness between system-to-system, Business-to-Customer (B2C) and Business-to-Business (B2B). Likewise, it beneficiary for organization to see beyond competitive threat and the need for improved quality which create instantaneous improvement in organisation working culture.

These are the convincing reasons organisations move their work flow to the BPM solution. Due to hypothetical unstable business circumstance, organizations move for stability. In these day and age where globalization and expansion of organization is paramount in the quick business transactions and dealings, BPM can uniquely adapt and fit all the business resources from far and near into one organization which plays key role in business profitability and fast product-ability Not to forget the competition, it is important for organization to stand ready to gear the processes to fulfil cost, quality and fast delivery time.

As shown in Figure 2.2, BPM application success can be measured in terms of financial results (Garimella et al., 2008)

Despite all the highlighted benefits of BPM, there are past failures of adopting BPM initiatives. Majority of the failures are human errors which has to be carefully addressed. Some publications have addressed these failures which can be some up to be misunderstanding of the BPM concepts and principles. Technology vendors or management

$$V_{net} = V_{new} - [Cost + Time + Waste]$$

Where:

Vnet : the net value produced by the project or process
Vnew: the total new value produced
Cost: the total cost of the new process and system
Time: the operational cycle time or development time
Waste: unused or discarded systems or capabilities

Fig. 2.2 : BPM Financial results. (Garimella et al., 2008)

consultant has puts BPM as a "lipstick" on their products which makes failing organization not to have desired results while these consultants or vendor make profit from selling their products without giving the full needed information in achieving their individual business goal (Trkman, 2010).

Another reason for BPM failure is the full alignment of business, IT and SOA. In a case when requirement changes or other change factors (economical, information system etc.) are play, to adapt to the change will involve re-designing the business process every time. However, section 2.4.1 assesses the relationship between BPM and SOA.

Other technical benefit of using BPM initiative are addressed in business process management flexibility and re-usability, discussed in section 2.2.

BPM Flexibility

The accommodation of change[7] should be a top priority of every organization. In clinical system, where PAIS, and human effort exists, deviations from pre-specified process or normal standard procedure can sometimes occur, depending on the case at hand Reichert and Weber, 2012, pg. 16. Dadam and Reichert (2009) also explains the flexibility and adaptivity of clinical process.

[7] "It is a bad plan that admits of no modification."-Publilius Syrus

No enterprise processes should be in steady state i.e., to be unable to quickly and flexibly react on momentarily or evolutionary changes on market, disaster, legal conditions over time. For example, there is a need to understand how process changes, and how to make use to the changes over time. These can be valuable for organizations to be change-aware, else these can lead to big disaster especially when the flow time are long. Groefsema et al. (2012); Rosemann and Recker (2006) further explain the flexibility support in BPM..

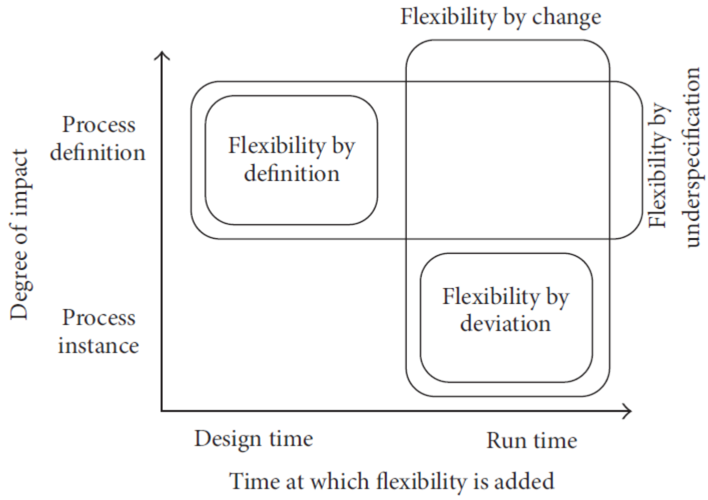


Fig. 2.3 Taxonomy of process flexibility identifying four main flexibility types. adapted from (van der Aalst, 2013)

Types of flexibility

As discussed above, processes changes over time and the degree of impact can be introduced both at process instance and process definition (cf.Fig 2.4). In the Figure 2.4 above show the types of flexibility: *flexibility by definition, flexibility by deviation, flexibility by underspecification, flexibility by change*.

(van der Aalst, 2013), (Ayora et al., 2012) and (Reichert and Weber, 2012) discussed extensively on types of flexibility. In this thesis, managing flexibility by deviation and flexibility by change aree related to our research.

(van der Aalst, 2013) defined Flexibility by change as "*the ability to modify a process defini-*

tion at run time such that one or all of the currently executing process instances are migrated to a new process definition.". Several BPM researcher have investigated the flexibility by change. As seen in Figure 2.4, it only flexibility by change that affects both the process definition and process instance. In flexibility by change, business constraints have to be set, which refers to the requirement imposed in the separation of complaint behaviour and the non-complaint behaviour (Maggi et al., 2011). Currently, there are few constraint languages supports flexibility by change like *DECLARE*(Maggi et al., 2011), and *ADEPT*(Dadam and Reichert, 2009). These languages models declarative processes which explicitly specift the process flow and describes the a set of constraints which must be satisfied (Maggi et al., 2011). To understand how flexibility by change works, declarative process modelling must be understood. (Groefsema et al., 2012), (Reichert and Weber, 2012), (Weske, 2009) further explain the usage of declarative process modelling, with different case studies.

As shown in Figure 2.4, flexibility by deviation only requires process instances at run-time. (van der Aalst, 2013) defined flexibility by deviation as "*is the ability for a process instance to deviate at run-time from the execution path prescribed by the original process without altering the process definition itself*". Typical deviations are skip, ignore, reset(van der Aalst, 2013). (Aur et al., 2011) discussed the deviation detection and handling during process enactment, approaches for flexible deviation handling (risk assessment, guidance and late handling) and formalization of the approach using temporal logic.

BPM Re-usability

Variability has been discussed in several fields ranging from software engineering to software product line, to BPM systems. (Groefsema et al., 2012) highlighted that introducing variability to BPM has given greater support for flexibility and re-usability, enhancing the readability, maintainability and redundancy issues . The research of Variability Management has increased, with the emergence and adoption of SOA(*see section 2.5.1*) in BPM.

(Sofela et al., 2013) "*defined variability in BPM as the is abstraction in which organization preserve its standard business process but also allows other templates to be built, customized and adapted on the exciting processes*". The current two variability approaches are namely: imperative variability and declarative variability (Schonenberg et al., 2008),which are offer at two stages of BPM lifecycle: design-and run-time respectively (Aiello et al., 2010).

Design-time variability

Variability at design-time offers specific changes at prescribed points in the process model known as variation points (Groefsema et al., 2012) which enable re-usability of business models. Variables may or may not be added to an existing template at design time (Sofela et al., 2013). The imperative view on variability within the any business process aids the different views of *design templates*, instead of designing the *design process* from scratch

Run-time variability

Brand new way of designing business processes where the basic principles or requirements of a process in a template are defined; variants are created and validated from the template(Sofela et al., 2013).

(Groefsema et al., 2012) (2012) proposed a new approach to making a business process more flexible, it combines the properties of imperative, declarative variability and newly developed process modelling environment with graphical elements. Their work is summarized as follows:

- when end users want to build a process model for their situational needs, they provide context information which a mate-level process model can be identified.

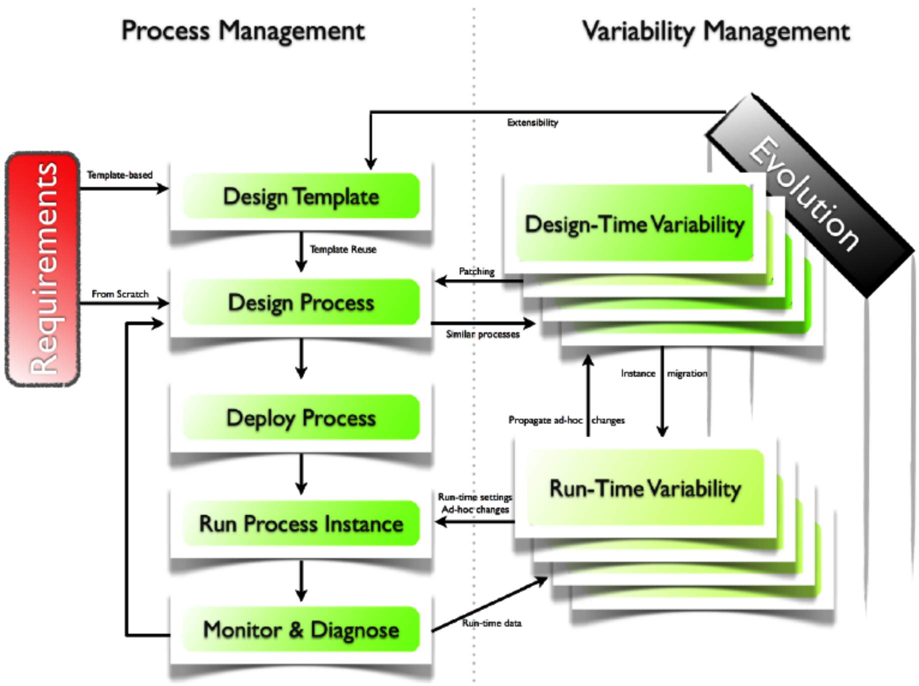


Fig. 2.4 Process Lifecycle and variability management [(Aiello et al., 2010), p. 2]

- they can select the main parts or frozen parts of business activities or sub-processes and identify the flexible parts or close area. Logic and temporal dependencies can be created according to guides.
- based on the frozen group, close areas, and logic temporal dependencies, the activity floating will be generated according to logic and temporal dependencies provided.
- until all constraints (frozen group, close areas, and logic and temporal dependencies) are satisfied, the process model will not be executed.

2.1.4 Technical challenges of BPM

Despite the life-cycle and benefit of BPM implementation and utilization in industrial application, BPM still appear to be facing some technical issues in direct direction of use, from intra and inter-organization collaboration, and inclusion of the cloud infrastructure into BPM. Issues in BPM is a large topic which can be seen from development, technical and business[8] angles.

In this research, we restrict ourselves to the technical issues that pose as a threat in full specification, realization and implementation of BPM.

van der Aalst (2013) raised some key concerns regarding BPM which exist in the process modelling, enactment and analysis stages. Likewise,Liu et al. (2008) highlighted the challenges in collaborative BPM, whereas Kirkham et al. (2011) explores the challenges of BPM in different domains in future SOA, and Schulte et al. (2015) highlighted the infrastructure challenges of elastic BPM[9] as shown in Table 2.1. From all these challenges highlighted in different publications, there are uniformity and also there appear some contrasts in the challenges.

[8]General and managerial aspects of enterprise has been widely discussed in several journals and articles based on the front-view challenges.

[9]Elastic BPM provides on-demand platforms and software over the Internet using cloud computing providing the same service at different price points through different cloud-based market mechanisms(Schulte et al., 2015).

Table 2.1 Comparison of the authors' challenges of BPM

Liu et al (2009)(Liu et al., 2008)	Kirkham and Winfield (2011)(Kirkham et al., 2011)	**Schulte et al (2015)**(Schulte et al., 2015)
1. Modelling collaborative business processes, 2. tracking collaborative business processes, 3. privacy and confidentiality, 4. process evolvement management. Another potential concerns are highlighted as(Liu et al., 2008): 1. Process integration(Jeston and Nelis, 2014), 2. process patterns (Dijkman et al., 2012), 3. process collaboration(Norta and Grefen, 2007).	1. Factory downtime, 2. Standardized work-flows, 3. User centric security, 4. Trusted federated identity management, 5. Seamless integration and controlling live production lines, 6. Safety and reliability of live production lines	1. Process scheduling, 2. Resource allocation, 3. Monitoring and collection of audit data or process, 4. Storage and state management of the application 5. The cost-efficient provisioning and de-provisioning of virtualized infrastructure, 6. Decentralized coordination for process enactment.
Uniformity: 1. Liu et al. (2008) highlights the challenges of collaborative BPM as shown above, which the similarity is close to elastic BPM by Schulte et al. (2015), i.e. if the challenges in elastic BPM are solved, it will definitely improve collaborative BPM.	1. the tracking of collaborative business process and monitoring and collection of audit data or process are discussed to achieve same motive.	
Contrast: 1. The modelling collaborative business process was only mentioned by Lui et al (2009) but right model alignment and coordination is important and core challenge in BPM.	1. In context to the cost-efficiency, Schulte et al. (2015) made cost consideration for elastic BPM, where as, in Lui et al (2009) highlighted the challenges without considering the cost of developing collaborative BPM and possibly the virtualized infrastructure.	1. The challenges of BPM by Kirkham et al. (2011) discusses issues process designer face in using dynamic service in different domains. This is different from other authors' challenges.

Narrowing down all the challenges shown in Table 2.1, it can be seen that there is still a big nut to crack to fulfil the full capability and benefit of BPM. Challenges of BPM is larger than what the authors have presented, reason being that there are growing technologies, service change and moving of BPM into Cloud while BPM functionality has not been fully plumbed. Therefore, it is important to tackle the issue from the start (design phase) to the end (run-time phase) before thinking of the moving BPM into Cloud(*cf.fig 2.5*)namely:

- correct and aligned modelling technique,
- correct validation and verification standard,
- correct tool support,
- business process re-use

Likewise, at the run-time phase, the challenges shown in BP enactment infrastructure and the evaluation (*cf.fig 2.3*) will also affect the validity of our research.
From these concerns we derive the key technical issues which are related to our research, which affect realization and implementation of BPM and SOA in any organization.

1. Aligned Modelling Technique
   As much as modelling and analysis of business process are key in BPM, it is essential to choose the right language that is best for the modelling a particular case (van der Aalst, 2013). van der Aalst (2003) highlighted the three classes of languages namely "(a)Formal Language, (b)Conceptual Languages, and (c)Execution Languages"(van der Aalst, 2013).
   Much of these languages are basic standard which can affect collaboration of two different organizations running different languages. Standards[10] are important in BPM, which act as the "trading language" for any BPM user.

[10] Standards should be discovered, not invented (Vincent Cerf, 1998)

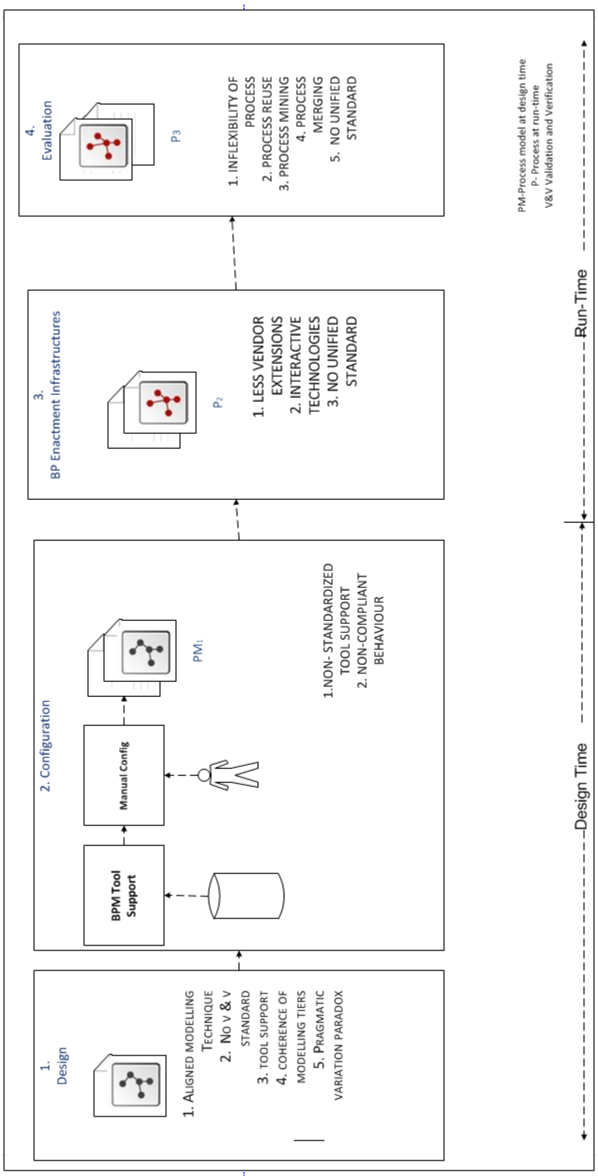


Fig. 2.5 Challenges at every phases of BPM

2. Correct verification is the checking for the logical-error in resulting systems. The result of not checking for this logical errors can result into deadlock and live locks (Aalst, 2003) and validation is said to be the process of checking if the resulting process behaves as the expected systems (Dumas et al., 2005). These two have also been a challenge at design-time because there have a less attention on performance analysis of process models to check for the correctness or soundness of models (Sadiq and Orlowska, 2000). Verification and validation is often neglected which results to systems failure.

3. Tool support: In business modelling, there exists a unified modelling tools, which seems to be resolved, now process model can be remodelled on any platform, likewise it can collaborate with another process.

4. Business process re-use: Koschmider et al (2014) (Koschmider et al., 2014) defined process model reuse as *" building up new business process models by assembling already designed ones"*, the effort of process modelling and modeller might be reduced if option of process reuse is further focused on instead of building process models from the scratch (Koschmider et al., 2014).
   When processes are very large and complex, BPM systems cannot provide the facilities to re-use this processes(Fantinato et al., 2012). In paper [(Dijkman et al., 2012), (Ekanayake et al., 2010)], describe the number of process models of Suncorp, it has over 6000 process models for insurance only.

2.1.5 Summary

A full description of BPM has been addressed in general, from the evolvement to BPM, to life-cycle, benefit and technical challenges of BPM. From all these, it is evident that a new concept is need that will manage large collections of process models regarding to their best services for client‘s urgent needs, adapt to unforeseen circumstance and increase process flexibility and reuse.

The next chapter will give elaborate on the current business process architecture and modelling approaches.

2.2 Business Process Architecture and Approaches

Process is vast word used in various context in the past, recent works and different lines of discipline for different purposes, and by different people of different backgrounds. Process is been used in different line of studies to the industries, for instance, in software engineering, process is defined as the relationship between related activities, while in industrial line, process is said to be a dynamic linking of business activities (Jr and Sommer, 2002).

Process has been used in many contexts, from production process, business process, mediation process (Moore, 2014) and in production, office, coordination, machine and mechanical engineering, manufacturing, material, information to business and many more.

In respect to Business process, Zairi (1997) defines "a process is an approach for converting inputs to outputs. It is the way in which all the resources of an organization are used in a reliable, repeatable and consistent way to achieve its defined goal."

Chinosi and Trombetta (2012b) defined business Process as a set of one or more linked procedures or structured set of activities executed following a predefined order by (potential several) actors (humans, computers and/or machines) in an organizational and technical environment which collectively defines the organizational objectives or business goal.

These authors definitions shows that business process may have predictable and definable input and gives a clear orchestrated picture of output of definable results.

This business process can exist as single business process or collaborative business process which have continuously developing, inter-dependent and interact with various systems, processes (Collaborative processes) and people (Westerman, 2009).

The effectiveness and efficiency of the business process is as result of automating business and information process.

The management of business processes(*conversion of business inputs, ideas to produce predictable output*) resulted to the evolvement of the concept of Business Process Management BPM. Likewise, the fact that business process needs to be modelled, also emerged BPM studies and systems, which gained momentum, received recognition and contributed to the world of computer science and management of corporate assets.

Weske (2009) defined Business Process Management as “concepts, methods and techniques to support the design, administration, configuration, enactment and analysis of business process." BPM provides a generalized knowledge and view of what business process is, as it combines a set of activities or task within an enterprise with a structure showing their logical order and dependence whose objective is to generate certain result(s) (Aburub and Almahamid, 2010). Thus, using the above definitions of BPM, they all emphasize that the characteristics of BPM, business processes at its core.

2.2.1 Business Process Modelling Languages- An Overview

In this section, we describe the chosen BPMLs which are well established in research or industry, for evaluation. Furthermore, an insurance claim will be used as an example to describe each languages. In the area of BPM (section 2.3), several standards have emerged to design, implement, enact and monitor business process models. And many more has been designed in the recent years (Heidari et al., 2013). Weske (2009) gives an overview of process modelling languages with detailed explanation of the languages

1. Business Process Modeling Notation (BPMN) [(Weske, 2009), pp. 205–225],
2. event-driven process chains (EPCs) [(Weske, 2009), pp. 158–169],
3. Petri nets [(Weske, 2009), pp. 149–158],
4. Yet Another Workflow Language (YAWL) [(Weske, 2009), pp. 182–200].

In the following subsections, the first three listed process modelling languages is presented in more details for the purpose of this report. The languages have had numerous numbers of extensive research work on these five languages.

Business Process Modelling Notation

Business Process Management Notation (BPMN) was developed with the goal of providing a notation that is easily readable, accessible and understandable for the all business users whom design and analysis, implement, monitor or diagnosis(OMG, 2014). It still remains the primary standard since 2004. M.Bridgeland and R.Zahavi (2008) observes that this is one of the most popular process modelling languages across the BPM industry . BPMN is a rich graphical modelling or representation language for business analysts to specify business process (OMG, 2014). The BPMN meta-models consists of four different categories: *Flow Objects* has three set of elements namely: events, activities, gateways which are used to model a basic business process, *Connectors Object* is either a sequence flow, message flow or an association flow, *Swimlanes*, and *Artifacts* consist of the text annotation, data object and group(Weske, 2009).

An example of BPMN representation is depicted in Figure 2.7. The diagram presents the collaboration processes of both the patient and the doctor's office, and the choreography

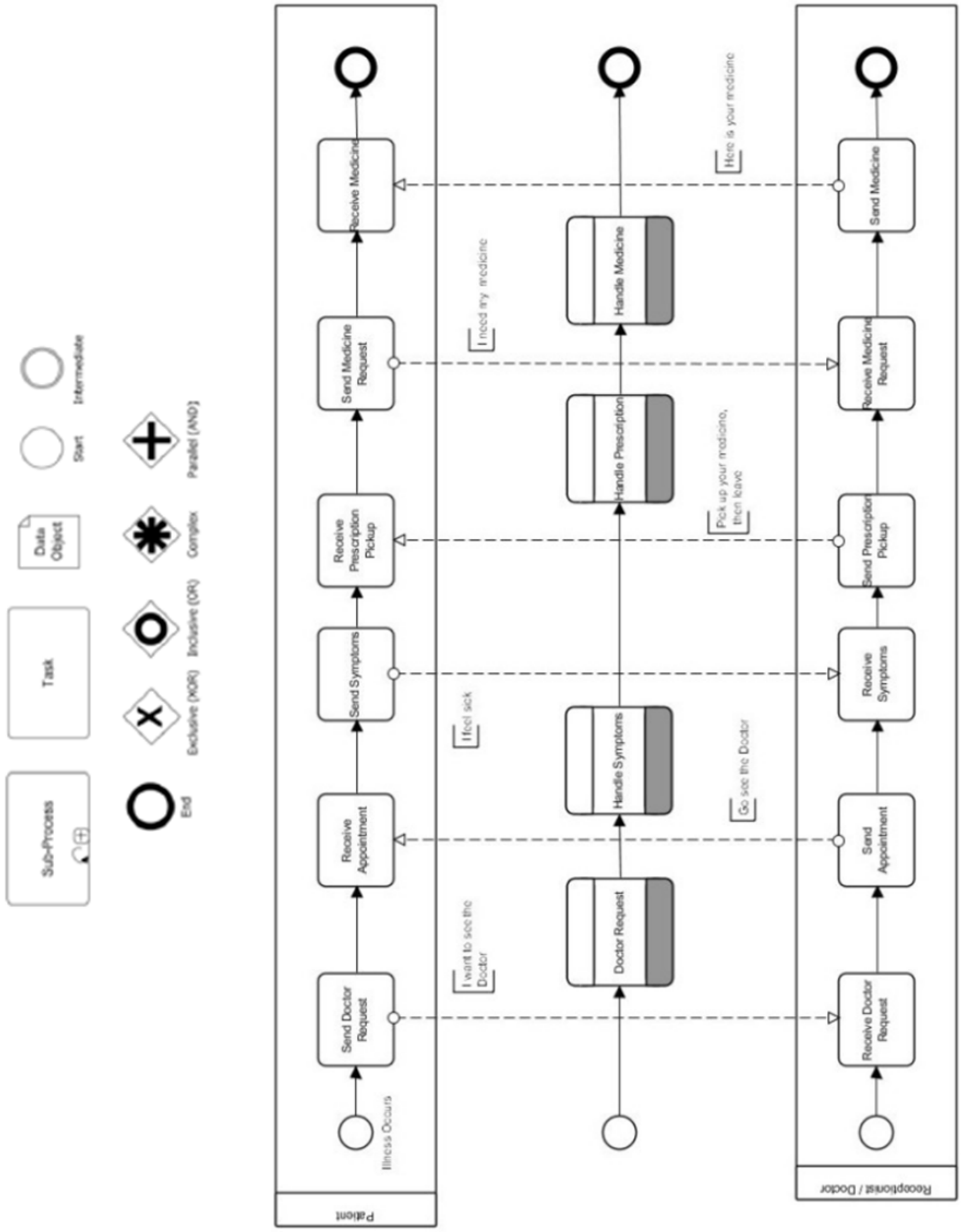


Fig. 2.6 BPMN Example of a Modelled Patient Treatment Process

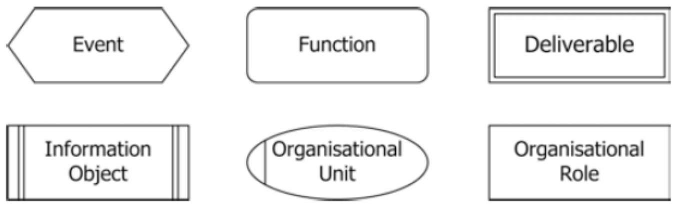
Event Driven Process Chain

was developed at the Institute for Information Systems of University of Saarland, in the 1992, in collaboration with SAP AG to create a suitable business process modelling language to document the processes of the SAP R/3 enterprise resource planning system. This project produced two major results: the definition of EPCs and the documentation of the SAP system in the SAP Reference Model [(Mendling, 2008), p. 17]. The EPC is based on the concepts of stochastic networks and Petri nets. The following formal definition is based on [(Weske, 2009), p. 162] combined with some other terminologies are used from [(Mendling, 2008), pp. 22–23]. A basic EPC consists of *functions*, *events* and *logical operators*. Definition 2.1 (Event-driven process chain) An event-driven process chain is a 5-tuple (E, F,C,m,A) for which holds:

1. E is a non-empty set of events. Events are created by process functions and acts as a pre-condition of one function.
2. F is a non-empty set of functions. Functions are active elements and model the activities within the company
3. C is a set of connectors.
4. m : C (*And,Or,Xor*) is a mapping which assigns to each connector a connector type, representing *And, Or* or *Xor* (exclusive or) semantics. events, functions and connectors such that the following conditions hold: – G := (N,A) is a connected graph. – Each function has exactly one incoming and exactly one outgoing arc. – There is at least one start event and at least one end event. Each start event has exactly one outgoing and no incoming arc. Each end event has exactly one incoming and no outgoing arc. All other events have exactly one incoming and one outgoing arc (intermediate event). – Each event can only be connected by functions, and each function can only be followed via connector, by events. – There is no cycle in an EPC which consists of connectors only.

– No event is followed by a decision node, i. e., an OR split connector or an XOR split connector

The extended EPC consists on the following elements namely: organization unit, information objects,and deliverables, as show in the figure 2.7 below.



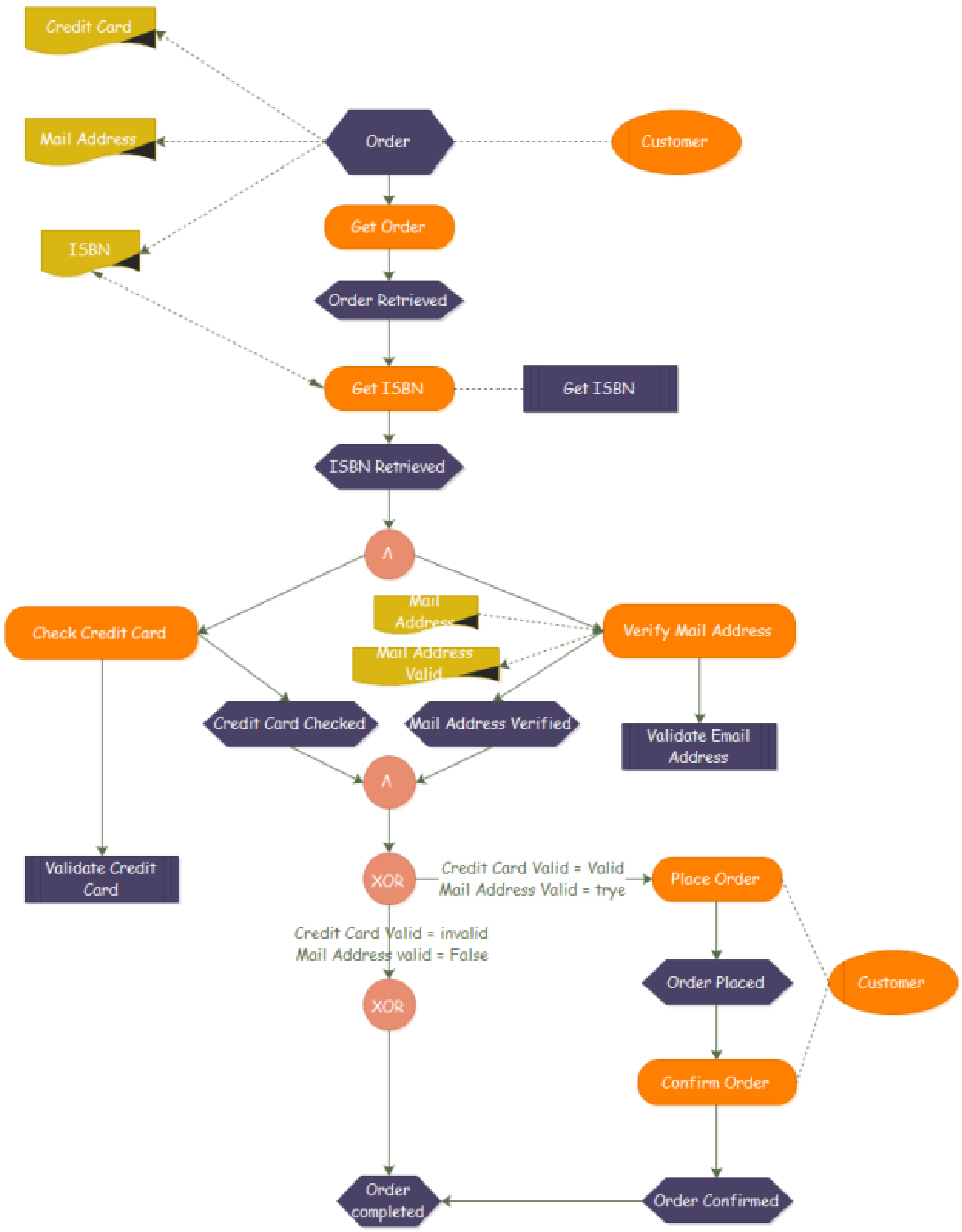


Fig. 2.7 EPC Example of a Modelled Book Ordering Process

Petri nets

Petri Net was designed for modelling, analysing and simulation of of dynamic systems with concurrent and non-deterministic procedures. A Petric Net is directed graph which consists of four elements, namely *places*, *transition*, *tokens* and *directed arcs* (*cf. fig 2.9*).

Figure 2.9 shows an example of petri net. In the example, it can be seen that the Petri Nets is not able to show OR-nodes, the business process is a bit different from the BPMN or EPC languages. When the first transition fires, the token move from the start or inception to the resulting position. For instance, if we are using the insurance claim process. the token moves from the *Record the Claim* to *Calculate the Insurance Sum*, then moves to the next transition *T2*, till it get to the last transition *T6*, and finally closes the case. As shown in figure 2.8, the activity diagram can be modelled in Petri Net as shown in Figure 2.9.

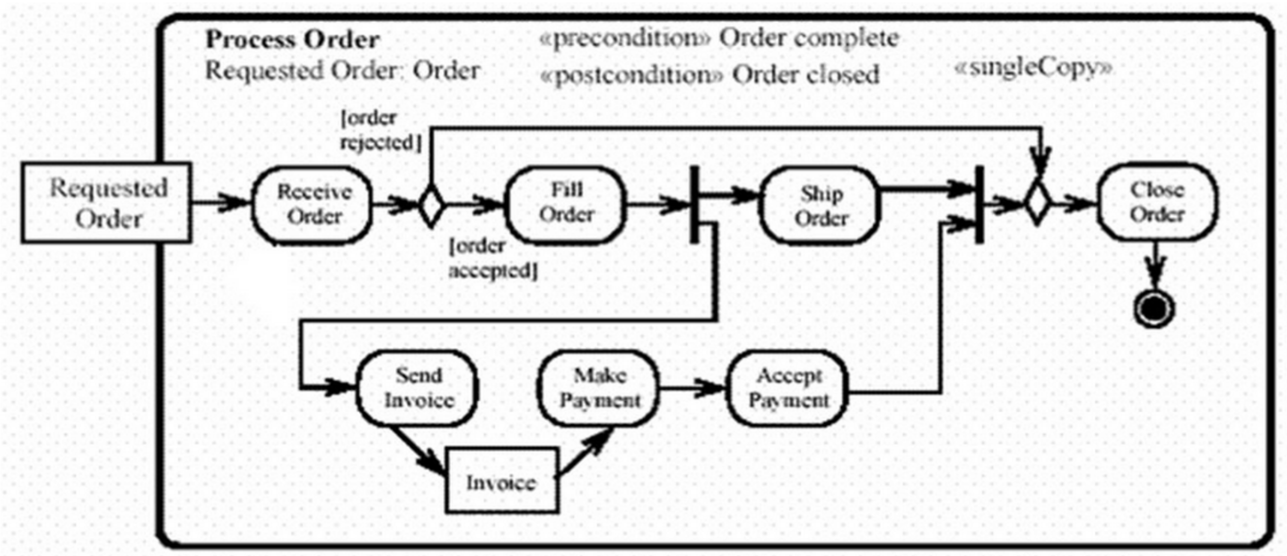


Fig. 2.8 Activity diagram to Petri Net example (Uottawa Institute, 2013)

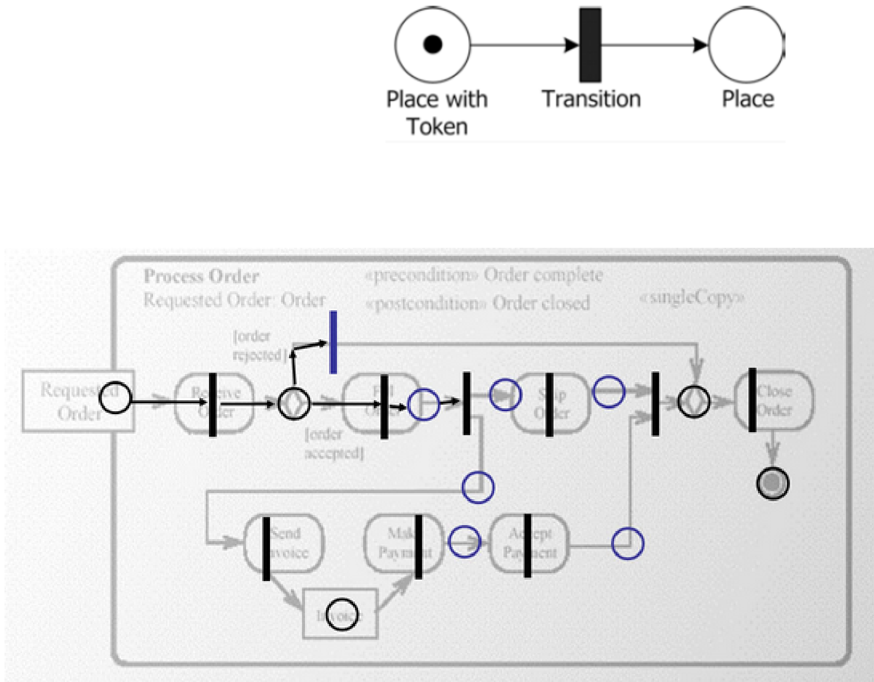


Fig. 2.9 Petri Net example (Uottawa Institute, 2013)

2.2.2 Business Process Modelling Approaches

Business Process Modelling represents a key part of the whole BPM discipline, although these two concepts uses the same abbreviation (BPM), and this can be mixed up or confused for each other.

From the inception of business process, in 1980s, there has been increased interest in the different techniques, methodologies and tools to facilitate a common understanding, analysis and modelling of business processes(BPs).

Many researchers utilised different BPM techniques in order to improve and facilitate BPs. Cull and Eldabi (2010) indicates that literature revealed that common modelling techniques are designed to satisfy one particular purpose and thus is not able of modelling all process aspects. As the number of techniques and references on business modelling are increasing, it is important to choose the right technique and definition for the right purpose and business needs.

Business process modelling has been defined in different context, attempted by different researchers. Business process modelling can be described as the analytical representation or illustration of organization with the aim to define and re-engineer organization's business process (Aguilar-Savén, 2004).

Modelling Standards

Few researchers recognized the absence of modelling standard in the techniques or methods in the business process management. In BP, there should be a standardized technique for modelling a similar business need. Since 2005, when van der Aalst highlight the needs for "well-established process modelling technique for business solution, there is lack of standardization which still creates "horses for courses" approach. This lack of specified standard has caused the absence of an adequate definition of a business process leading to different language and terms. van der Aalst (2003) highlighted the three classes of process modelling languages namely "(a)Formal Language, (b)Conceptual Languages, and (c)Execution Languages"(van der Aalst, 2013). Much of this languages are have their basic standard which can affect collaboration of two different organizations. Standards[11] are important of BPM, which act as the "trading language" for any BPM user. Currently there are several standards that aids the smooth running from the design stage to run-time stage like BPMN to BPEL namely:

- *Graphical standards*: expresses business processes and their possible flows and transitions in a diagrammatic way(Liu et al., 2008). Unified Modelling Language activity diagrams-UML AD (Object Management Group (OMG, 2004b), Business Process Modelling Notation BPMN, event-driven process chains, Role Interaction Diagram-

[11] Standards should be discovered, not invented(Vincent Cerf, 1998)

RID and role-activity diagram (RAD) are examples of graphical standards(Ko et al., 2009).

- *Execution standards*: the modelled process can computerized and aids the automation and deployment of the business process like BPEL(Ko et al., 2009; van der Aalst, 2013). Business Process Execution Language-BPEL, Yet Another Work-flow Language (YAWL) are examples of execution standards.

- *Interoperability standards*: This standard is the translation of the graphical standards to execution standards and exchanging the business process models between different modelling platforms(Ko et al., 2009)(Mendling et al., 2005). Business Process Definition Metamodel -BPDM, XML Process Definition Language -XPDL are examples of interchange standards.

- *Diagnosis standards*: this standard audits, queries and monitors the bottlenecks of the post model(Ko et al., 2009). Business process run-time interface (BPQL), Business Activity Monitoring Language (BAML), Business Process Audit Schema (BPAS) and Business process Query Project (BPQP) are all examples of diagnosis standard.

After choosing the right language, another challenges that can affect modelling is verification and validation of process models (Aalst, 2003). Validation is said to be the process of checking if the resulting process behaves as the expected systems. Verification is the checking for the logical-error in resulting systems. The result of not checking for this logical errors can result into deadlock and livelocks(Aalst, 2003). Using the stages of Process Modelling, from the design-time and run-time to analyse the each of the challenges of BPM, as shown in figure 2.3. This diagram can be used to visualize the possible challenges which is not intended to be completed. With time, new challenges may/will arise or existing ones may vanish due to advancement and improvement in business services and BPM. At the design stage, requirement, modelling, validation and verification are the known challenges of designing a business process. Aside from the standardization, Indulska et al. (2009) highlights the business process modelling issues and challenges within five years time. The authors identified and prioritized the significant challenges namely: business-IT-alignment, service orientation, collaborative modelling. Another issue is context awareness, Sofela et al. (2013) highlighted the context-aware process modelling ,demonstrating the currrent process modelling approaches and tools, allowing end users to identify a business process model from a process repository according to different context information.

Typically, the user does not need to be involved in the modelling but Aguilar-Savén (2004) highlighted the need to choose the right technique, putting the modelling purpose into consideration. Aguilar-Savén (2004) also distinguishes between the uses of business process models: one for traditional software development and the other is the restructuring of business processes. From the inception of business process in 1980s, there have been several tools to model business process. Process orchestration and choreographies are the necessary tools for modelling a single process model or collaborative process model.

Process Orchestration

Business process models specifies the activities and their relationships, likewise it specifies the process orchestration (Weske, 2009). Process orchestration is notion in which services (processes and execution constraints) are invoked with a detailed view in explicit order. To express process orchestration, control flow pattern has to be known (Weske, 2009) namely:

- And Split and Join ((Weske, 2009), pp 128)
- Exclusive Split and Join (Xor) ((Weske, 2009), pp 130)
- Or split and Or Join ((Weske, 2009), pp 131)

An *and-split* is a point in a process model where a single activity A is split into multiples of other activities B and C.

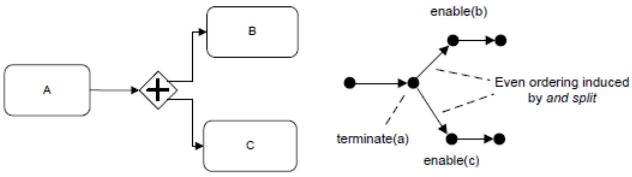


Fig. 2.10 And-Split pattern Adapted from (Weske, 2009)

Using the Insurance claim example, it can be be explained that the activity *A* terminate at the (a), and it will then enable (b) and (c) concurrently. This process can be said to be incomplete or non-working process because it is an unending process. Unlike Figure 2.10, it starts with a single activity (record a insurance claim, A), then other activities (Calculate the insurance sum B) and (Check customer history, B), and few other activities. In order for Figure 2.11, to be complete, and additional activities are need, likewise an *And Join*.

An *and-join* is a converging point in a process model where multiple concurrent activities meet and combine to become one single activity, as shown in figure 2.11. In this figure, it can be seen that the (b) and (c) terminates and converges at the and-join gateway, which enables (d). The combination of the And split and And-Join forms a complete process. Finally, it ends with a single activity or service (pay or do not pay insurance sum, D).

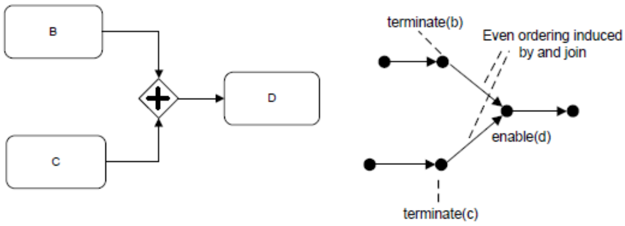


Fig. 2.11 And-Join pattern Adapted from (Weske, 2009)

An *xor-split*is a point of divergence where incoming single activity (A) splits into resulting activities (B and C) and only one path or branch can be chosen, but not both. Option 1 is enabled (b) while Option 2 is null, c= 0 (figure 2.10). If B is running or executed, then C is terminated. This is also an incomplete process.

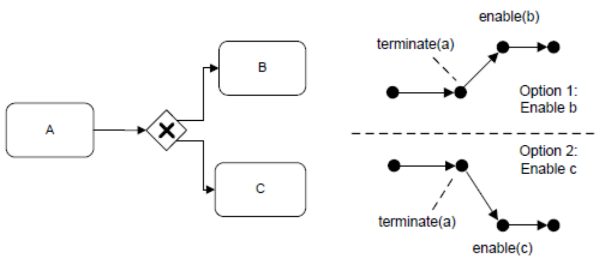


Fig. 2.12 Xor-Split pattern Adapted from (Weske, 2009)

Weske (2009) defined *xor-join* as “a point in a process model where two or more alternative threads come together without synchronization". This shows that only one of the option (1 or 2) will be come true or enabled to become (d).

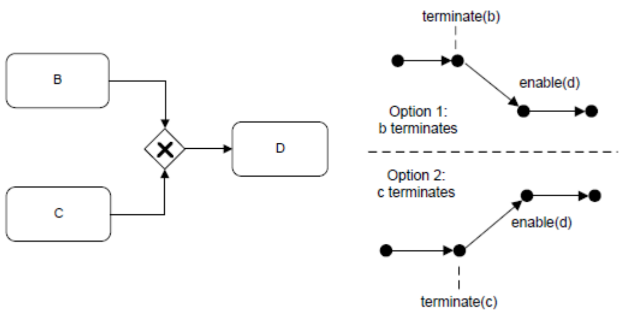


Fig. 2.13 Xor-Join Adapted from (Weske, 2009)

Or-split is point in a process model where an activity split or diverge into a number of branches. Weske highlighted that "An or split restricts the events of related activity instances as follows: for each termination event of *(a)* there is a subset of enable events of (*b*) and (*c*)." (Weske, 2009), as shown in figure 2.14

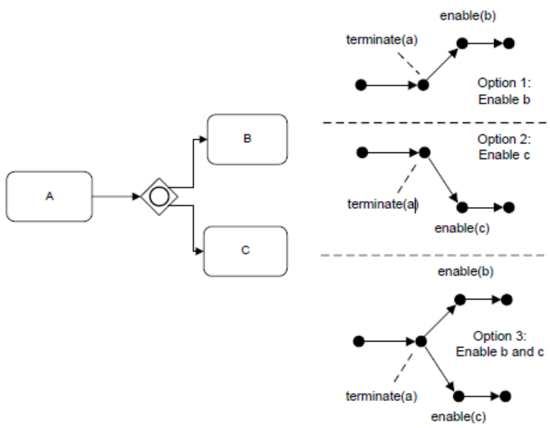


Fig. 2.14 Or-Split Adapted from (Weske, 2009)

Or-Join is point in a process model where multiple activities or tasks combine to be a single activity. In figure 2.15, it shows a process model with activity models, B, C and D and a gateways G, there are three behavioural options for the join. The first option (option 1) is for the upper activity B, i.e only activity instance *(b)* is taken, whereas the rest of the instance (c) is null. The second option (option 2) is for the activity instance *(c)* is taken, whereas the other instance *(b)* is null and the last option (option 3) is for the two activity instances are both enabled. Weske (2009) highlight that or-join is a problematic control flow pattern, because if incoming branch is triggered, the or-join either "*Wait before the activity instance d is triggered because the other incoming path—which completes in activity instance (c) can still be executed.*" or "*Trigger instantly after the termination of b*" (Weske, 2009), this can be a problematic in modelling process instance because there might be unnecessary wait or trigger which may affect the resulting activity or might cause deadlock or live-lock.

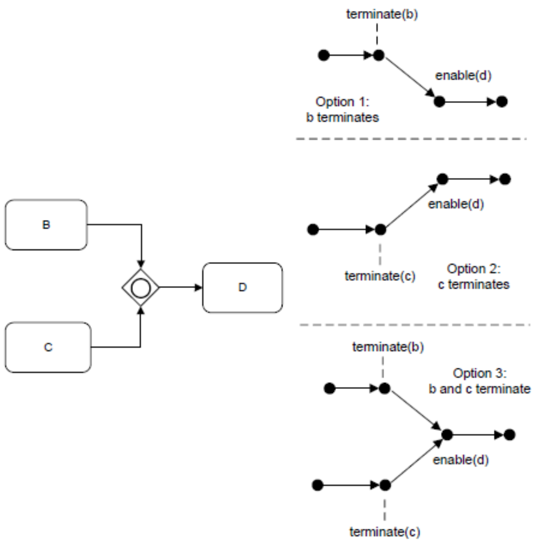


Fig. 2.15 Or-Join Adapted from (Weske, 2009)

Process Choreographies

Process orchestration has shown the relationship between activity, gateways (*Xor, And* and *Or*) and the execution. However, it shows a single direct transaction or business. What happens when it comes to business-to-business collaboration?

To realise these collaborations, process orchestrations interoperate by sending and receiving messages, whereas choreographies ensures the interoperability between these orchestrations, each of which is performed by different participant in business-to-business collaboration (B2B) (Weske, 2009). For example, consider a insurance claim scenario between a Company *X*, and insurance company *Y*. The first activity is to be performed by Company X is send its application for insurance claim (*A1*) and check its compatibility and order constraints like interoperability and may more.

Relations between Process Choreographies and Process Orchestration

The execution of activities and the relationship between these activities is captured in *process orchestration*, while the collaboration and interoperability of different process orchestrations is known as the *process choreographies*. The relationship between these two are namely:

- they share and interlock in the same platform,
- in the world of today, where companies jointly combine their services to provide a more efficient brand, they need will need the requirement of process choreography and orchestration,
- they both follow the same BPMN rules, they both need each other for properly collaborate, i.e. choreographies cannot execute without the use of orchestration, likewise two or more orchestration cannot interoperate with the use of process choreographies e.g. pool and swim lane.

2.3 Service-Oriented Computing

Service-Oriented computing (SOC) is the design paradigm for distributed computing platforms that utilizes services as compositional and fundamental elements used to build software and application system solutions (Papazoglou, 2003; Papazoglou and Heuvel, 2004).

Huhns, Michael and Singh (2005); Papazoglou (2003) highlights the concepts, principles and the direction of SOC. They further discuss software as a service, its concepts, the basic and extended SOA.

Currently, the technical solution mostly adopted for the development of services-oriented computing is Web Services (Papazoglou, 2003), and its technologies.

SOA with its enabling Web Services is currently contributing to the best technological solutions to the distributed and loosely-coupled collaborative business application (Papazoglou, 2003).

Realizing the SOC promise involves the development of SOA, likewise the development of SOA will improve the platform-base of BPM, which will further increase the flexibility and productivity of business processes. The next sub-subsections will explain what SOA is, the layers of SOA, the SOA and web services, benefit and challenges of SOA.

SOA has been a topic of interest in the computer science as a whole. Hirschheim et al. (2010) defines SOA as "is an IT architecture where data and logic functionality are "black boxed" or encapsulated with only their input and output exposed for others to use". Web Services can be understood as on-line self-describing distributed components that expose services and functionalities through on-line interfaces, and can be universally described, published, located, and dynamically invoked by means of Internet communication protocols and standardized XML-based programming ranging over (SOAP, WSDL, UDDL, BPEL4WS to WSCI) [12] [13] (Liu et al., 2011).

Implementing distributed computing via the Web Services technology will increase portability and interoperability of cross-organizational business application; organizations' flexibility, re-usability and scalability can be further improved; increase competitive and service quality and simplifies the cross-organizational business application systems.

2.3.1 Service Oriented Application

SOA is a multi-layer computing paradigm (Papazoglou, 2003), distributed information system that emphasize on dynamic service discovery, composition, and interoperability (Legner and Heutschi, 2007), interplaying between different related ITs and industrial application including BPM, to either improving or achieving high business process quality, efficiency and agility. SOA is a multi-layer integration architecture with different layers namely; desktop

[12] SOAP is known as Simple Object access Protocol, WSPL is Web service description language(Fisher et al., 2013)

[13] BPEL4WS is business process execution language for web services

integration layer, work-flow integration layer, services, the application system layer as shown in figure 2.16 below, the four layers are shown with the flow logic connecting the components (Legner and Heutschi, 2007). The architectural components of SOA are distinguished by application-related view highlighting the components that implements business logic (Legner and Heutschi, 2007). application-neutral view highlighting the component that implement integration mechanisms and infrastructure components, which provides services and protocols for system integration and implementation (Legner and Heutschi, 2007).

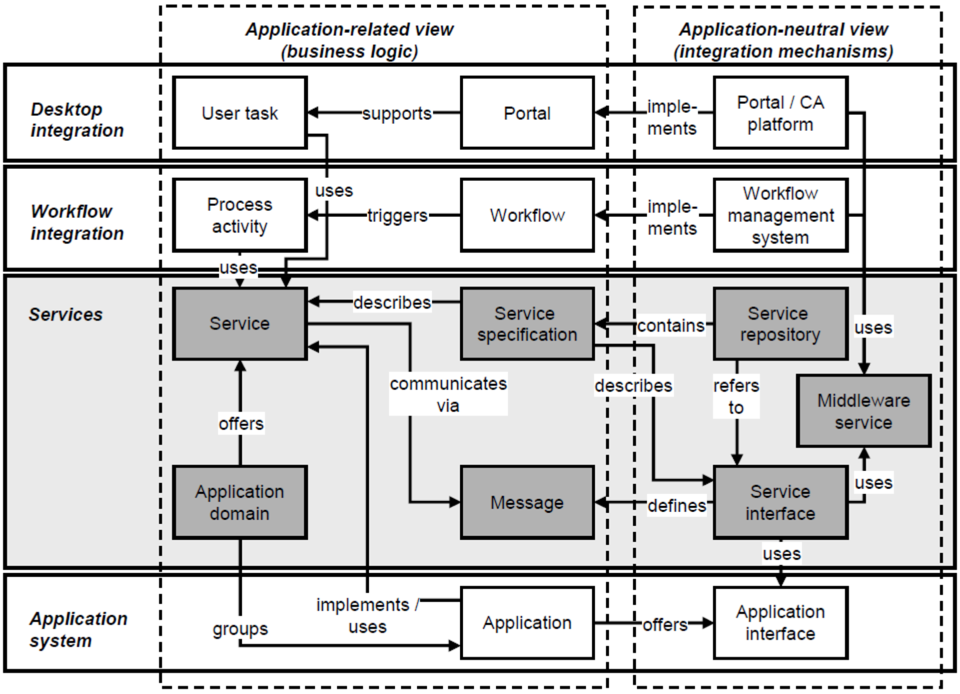


Fig. 2.16 SOA Layers and Architecture Components(Legner and Heutschi, 2007)

2.4 Service-Oriented Business Process Management

Modern BPM expands to cover the inter and intra-organizations' business services, thereby the need to support organizations' information flow and maintain a competitive level, service oriented solutions have been a useful paradigm in the support, development, improvement, automation and integration of business process-centric information systems and which has been is linked into service-oriented BPM. For any organizations to stay dynamic and to adapt to any change, their use of BPM systems must also facilitate technologies like cloud computing Brown et al. (2011); Purcell et al. (2013), big data[14] and concepts like SOA. Integrating these two paradigms has been a subject of considerable research interest in recent years, observed through the different authors. Recent contributions to this community show how sharing knowledge documented in process models and how they are co-connected with other models in another intra/inter organization. Among these, Juric and Pant (2008) highlighted the SOA approach to business processes, which they explained that "SOA introduces technologies and languages that reduce the semantic gap between the business processes (picture or paper-based) and the actual applications (code), which acts as a guiding principle and technical architecture to develop business processes for modelling and execution tools like BPMN and BPEL. Grefen et al. (2009) analyse the requirements for the support of dynamic business processes with service oriented computing across the boundaries of organizations. He concludes that "the current state-of-the-art does not yet provide an integrated solution, but that many capabilities are available or under development".

The emergence of these concept has accelerated the flexibility and variability of BPM system, likewise opened a wide range of automation and integration (Aiello et al., 2010). Currently, the technical solution mostly adopted for the development of services-oriented computing is Web Services (Lins et al., 2012), and its technologies. SOA with its enabling Web Services is currently contributing to the best technological solutions to the distributed and loosely-

[14]There will be a special need to carve out a place for the human: to reserve space for intuition, common sense, and serendipity (Cukier and Mayer-Schoenberger, 2013)

coupled collaborative business application (Lins et al., 2012). The strong adoption of SOA in BPM is as a result of its characteristics, among which we can highlight: Increasing the flexibility of the organisations' change, continuous process improvement, and narrowing the gap between the business and its software systems. Integrating these two paradigms (BPM and SOA) has been a subject of considerable research interest in recent year, observed through the different authors Jabbar et al. (2015); Li et al. (2009); Menzel et al. (2009); Zimmermann et al. (2005).

2.4.1 SOA and BPM together

(Hirschheim et al., 2010) highlighted that the whole purpose of linking BPM and SOA is the need to focus on specific business outcomes, which gets us nearer the (Holy Grail of IT in business)[15].Before getting to the Holy Grail of IT, there are some issues that has to be noted. (Behara, 2006) reports some issues when these two "*concepts*" are used as stand-alone concept. One of the issues addressed by (Behara, 2006) is that the use of BPM concept without the inclusion of SOA is only useful for building applications, but difficult to extend to the enterprise and SOA without BPM is useful for creating re-usable and consistent services, but lacks the ability to turn those services into an agile, competitive enterprise.

Realizing the SOA promise involves the development and accommodation of BPM platform, which will enable building applications that are extendable and re-usable by enterprise which further increase the flexibility and productivity of business processes and services.

With the augmentation of SOA into BPM, it has had impact on the implementation and performance of BPM.Noel (2005) discuss the goal of SOA in the implementation of BPM namely:

1. SOA provides readily available platform or framework for the communication and integration of organization's computing assets.

[15]The ability to translate business policy statements automatically into outcomes which are delivered by IT systems

2. SOA provides a loosely coupled integration platform which condone changes to application instances without affecting the core integration technology.

3. SOA helps in the creation of service independence which helps in the alignment between the business process models and the actual enterprise implementation as shown in Figure 2.19.

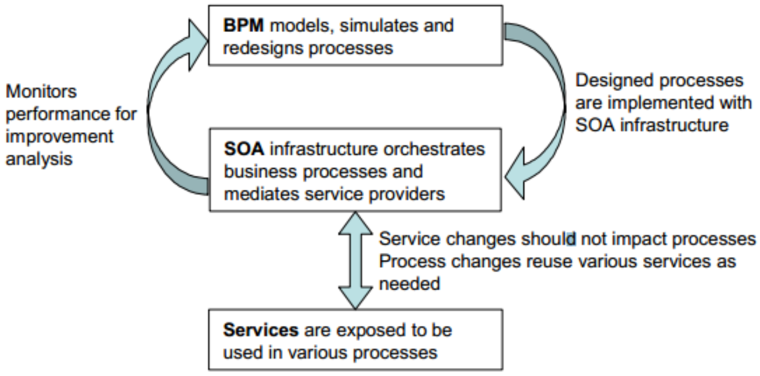


Fig. 2.17 Creation of service independence (Noel, 2005)

Beimborn and Joachim (2011) investigated and proved by empirical evaluating of several firms for the impact of SOA and BPM on business process quality. As much as organizations are deriving resourceful benefits in BPM, the inclusion or the further adoption of SOA into BPM will improve the implementation and performance monitoring and delivering outstanding quality. (Behara, 2006) demonstrated the process implementation and performance monitoring in SOA and BPM enabled joint-concept *cf.Fig 2.20*.

In the joint-concept, the BPM model, re-design, simulate business processes and SOA can invoke business process which increase business flexibility, better business practices, easier integration, re-use of assets and reduction of risk (van der Aalst, 2013). Therefore, the gap between the process and application is narrowed down (Behara, 2006).

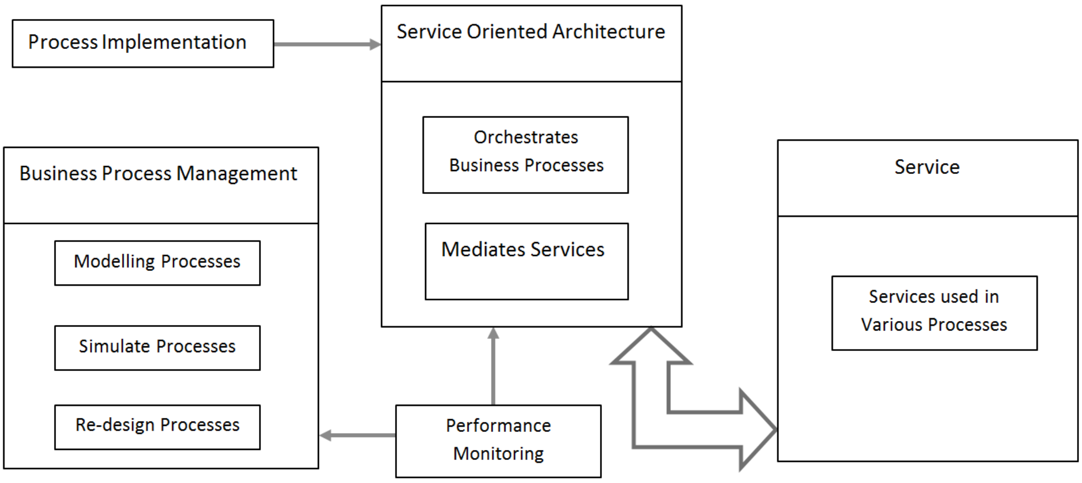


Fig. 2.18 Performance monitor in SOA and BPM relation (Behara, 2006)

Furthermore, the implementation of distributed computing via the Web Services technology will increase portability and interoperability of cross-organizational business application; organizations' flexibility (re-usability) and scalability can be further improved; increase competitive and service quality and simplifies the cross-organizational business application systems.

The combination of BPM (*BP reference models, documentation, analysis*),Service-oriented (*Architecture and Enterprise*) and defined technology arise to the SO-BPM which has emerged as the semantic gap between the paper-modelled business process and actual application systems, and changes in the processes. The benefits of SOBPM are:

- promise of business process re-use
- proper standardization and consolidation of business process and resources.
- aligned business process re-design
- business process agility, resilience, and flexibility
- Autonomic systems and quality of service
- improved collaboration between inter-organization and innovative business models

Organizations all over the world are subject to changes due to deregulation of market, global competitions, and increase of business or customer needs (Kohlborn et al., 2009a), as organizations are craving for improved and vigorous system which suits their needs ,likewise is emerging service technologies has forced non-service based software systems to become legacy (Adjoyan et al., 2014), which requires service identification methods matching the challenge in closed information environment, defence department of any country is known closed information environment which act as barriers to knowledge and information sharing. It is known that current challenges exist within a closed information environment (Goh and Hooper, 2009) and no consideration for open information environment i.e. identification of services within a one organization is not yet finite, but a dive-in into open information environment will be appreciated, thereby, organization‘s interoperability[16] and re-usability will increase.

Goh and Hooper (2009) highlighted the differences between the closed and open information environment. One of the key differences is that closed information environment acts as an good example of what most organizations experience which is the prompted trust and protective information exchange within its organization, collaboratively or among staff members, so not to give in to information bridge. Therefore service identification within closed information environment takes information security more critical within its proximities. In that case knowledge management and information sharing has to put into consideration.

[16]Interoperability is known key challenge in service identification because there is no clearly unified, specified terms and standardized approach

2.4.2 Benefits of SOBPM

The combination of BPM (BP reference models, documentation, analysis),Service-oriented (Architecture and Enterprise) and understand a defined technology arise to the SO-BPM which has emerged as the semantic gap between the paper-modelled business process and actual application systems, and changes in the processes. The benefits of SOBPM are:

- promise of business process re-use
- proper standardization of process and resources.
- aligned business process re-design
- business process agility, resilience, and flexibility
- Autonomic systems and quality of service
- improved collaboration between inter and intra organization and new innovative business models

2.4.3 Inter-relationship between Service-Oriented Architecture, BPM systems and other paradigms

Modern BPM expands to cover the inter and intra-organizations' business services, thereby the need to support organizations' information flow is linked into other related technologies. In order to organizations to stay dynamic and to synergically adapt to any change, their use of BPM systems must also be facilitating technologies and concepts like SOA, big data, cloud computing and smart asset. The emergence of these technologies has accelerated the flexibility and variability of BPM system, likewise the adoption of SOA and its standards like Web Services has opened a wide range of automation and integration (Aiello et al., 2010). Cloud Computing has increase the number of researches towards BPM outsourcing.

In order to maintain a competitive level, service oriented applications have been a useful paradigm in the support, development, improvement, automation and integration of business and its data and process-centric information systems.
Juric and Pant (2008) mentioned the relationship of SOA with business process management "SOA introduces technologies and languages that reduce the semantic gap between the business processes (picture or paper-based) and the actual applications (code), which acts as a guiding principle and technical architecture to develop business processes. SOA has been a topic of interest in the computer science as a whole. Hirschheim et al. (2010) defined SOA as"is an IT architecture where data and logic functionality are "black boxed" or encapsulated with only their input and output exposed for others to use". Also, SOA has been interplaying between different related ITs and industrial application including BPM, to either improving or achieving high business process quality, efficiency and agility.
Bajwa et al. (2009) has noted that the "partnership of SOA[17] has been fruitful by merging the benefits of both sides" Bajwa et al., 2009, pg. 677. Talking about business process quality, Beimborn and Joachim (2011) investigated and evaluated the impact of SOA and BPM on business quality . The ease with which business streamline or upgrade processes bringing flexibility as the business changes is what SOA does. Gary Gomersall highlighted that "The whole basis of the link between BPM and SOA is the need to focus on specific business outcomes, which gets us nearer the (Holy Grail of IT in business)[18]" (James, 2014).
Within these trends, there are challenges related to the adoption of BPM with this technological trends. Liu et al 2008 highlighted the challenges that arises during the adoption of SOA in collaborative BPM, (Liu et al., 2008) namely: "service composition (Hirschheim et al., 2010), service discovery,(Ese, 2012), service monitoring, (Norta and Grefen, 2007), service orchestration (Aalst et al., 2010; Yung and Wong, 2011).

[17]"SOA is the driving concept towards that business outcome and "BPM is the 'what', controlling whatever the business process is designed to do," explains James (2014)

[18]The ability to translate business policy statements automatically into outcomes which are delivered by IT systems

Scientific and industry domains have given considerable attention to SOA, BPM and Service Oriented Business Process Management (SOBPM) which is explained further in Section 2.4. Also, most computing objects (e.g., mobile phone, Ipad etc.) are equipped with wireless connections to a global server serves as one of the official tool for pervasive computing in realization of daily task which also help and create an ideal platform at real time BPM. Liu et al (2008) also identified the challenges in managing business processes related to ubiquitous computing namely (Liu et al., 2008): formalisation of business processes and Radio frequency identification (RFID)data, swarming behaviours, RFID network dynamics. This formalization of business process, radio frequency and network dynamics are being upgraded to business standard as computing objects, the reseason why most organization (banks) can use Apple ipad within or outside their business organization without restrictions.

Cloud computing

Purcell et al. (2013)invented a dynamically optimized distributed cloud computing-based (BPM) system shows how they allocated different service providers to each task and work flow server can receive information for ones of the tasks.

Also, Brown et al. (2011) highlights that ‘the way information technologies are deployed are changing too, as new developments such as virtualisation and cloud computing reallocate technology costs and usage patterns while creating new ways for individuals to consume goods and services and for entrepreneurs and enterprises to dream up viable business models."

These authors show that the adoption of cloud computing into BPM will morph in unexpectedly improve business services.

Big Data

Big Data[19] (Cukier and Mayer-Schoenberger, 2013; Mayer-Schönberger et al., 2013) is another trend that has impact not only BPM but also all approaches, data, and technologies which is portrayed as a big-ticket IT endeavour (Research, 2013). The concept of Big Data have been around for few years, but it is said by different authors Bennett et al. (2013); Brown et al. (2011); Research (2013)), that big data is having difficulty making its way out of the IT department into the business stream. Adopting big data in organization will improve customer intelligence, identify new trends and opportunities, improve overall business insight and customer service (Research, 2013), and likewise it will aid improved data and process mining. So far in 2014, the computer research group (2013) reported in a survey that "the interest in and roll-out of big data strategies have moved on markedly in the last year" (Research, 2013). Big data will not only enable us to experiment faster and explore more leads but also support the improvement and full materialization of BPM benefits and applications.

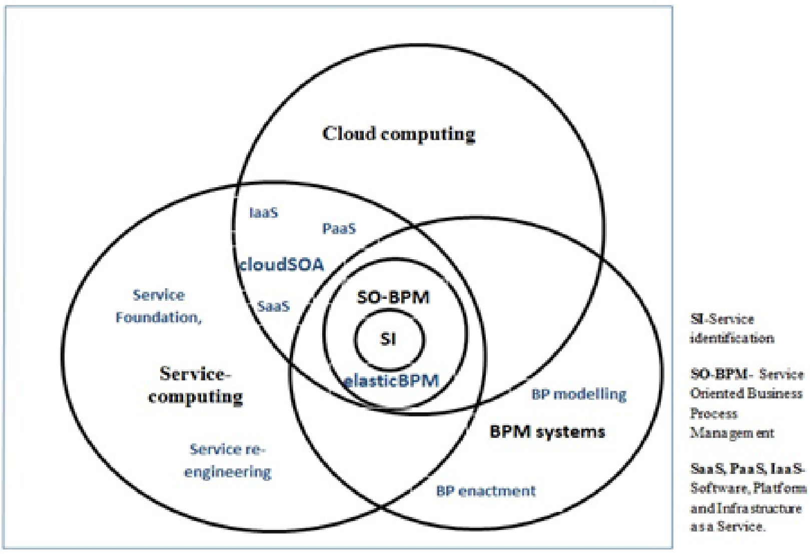


Fig. 2.19 The inter-relationship between BPM systems, Service Computing and Cloud computing using Venn Diagram

[19]There will be a special need to carve out a place for the human: to reserve space for intuition, common sense, and serendipity (Cukier and Mayer-Schoenberger, 2013)

2.5 Service Identification

Service identification is one of the most essential and challenging phases of SOA development for several reasons, one major reason is the flexibility and reuse of services can be properly be utilized with the right level of granularity of the services, which has influence the effectiveness of SOA architecture (Bianchini et al., 2011) and service modelling.

Since the adoption of SOA into enterprise applications, a growing number of researches have been done on the identification of different types of services. However, there is no unified understanding, approach and technique in achieving this goal, which constituent different service identification methods (SIMs). In building a SOA, it is necessary to understand the key architectural decisions about the SOA layers, which must demonstrate well aligned business services and decisions before they can be composed into applications taking into account the service providers and consumers. As shown in Figure 2.20, Arsanjani (2004) describe the three steps for service-oriented modelling and architecture, components and flows: identification, specification and realization of services.

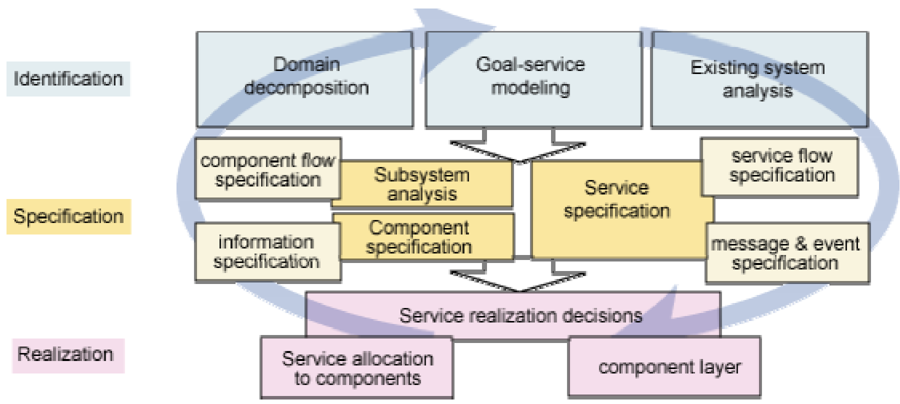


Fig. 2.20 Service-oriented modelling and architecture method (Arsanjani, 2004)

Service identification is not a new paradigm, it has been on over the last decade, it has the capability of improving business standard and goals, that is if the right business service is identified. In this paper, business service is not ordinarily used on its own but right business service is. (*Software/Business*) service is defined as the rightly identified service with acceptable performance metrics like right level of granularity, low coupling and high cohesion while satisfying low reuse cost principle at application-oriented, business-related level of granularity using widely applied standards (Boerner and Goeken, 2009; Jamshidi et al., 2008; Legner and Heutschi, 2007), derived from high-level business requirements and business process models. Software services are clearly different from business services, In organizations, any service being used or has been used must have been a satisfactory service which satisfies the need of any business need. How do we really know if a service can be addressed as a right business service? In several publications, the term service/ business service has been used based on different input, which generate different types of services, as shown in *Table 2.2*.

During the last decades, different input resources have been introduced, but there is no consensus on the best input method for different inputs types, likewise the approaches, techniques have all been different which make up the components of SIMs. These component differs from input (*Table 2.3, 2.4 and 2.5*), predominant approach, technique (*Table 2.8*) used in different publications.

2.5.1 Input types and Approaches for Identifying Services

There have been diverse use of different input resources in service identification in existing service-oriented systems. Two major reasons for diversity in the usage of different input resources are:

- the organization's current asset, business values and condition, which determines which type or availability of desired input that is implemented in the organization.

- the approach (Top down, meet in the middle, bottom-up and hybrid) determines choice of input to use.

Alahmari et al. (2010) identified seven services types generated by the SIMs used namely: process services, infrastructure services, business services, master-data service, utility services, transactional-data services and composite services. Gu and Lago (2010) identified six various service output types namely: business process services,, data services, composite services, information technology service, web services, partner services. All these types of services can be either business-related or software related, therefore, in this report, we use the term business services and software services as shown in *Table 2.2*.

Table 2.2 Classification of service types

Service type	**Functional Scope**
Software service	encapsulates application logic functionalities
Business service	aggregating several business logic or entity representation

1. Software Services

   It is the re-usable functions of encapsulated application which based on business requirements (Cai et al., 2011; Mirarab et al., 2014), consumed separately by different entities (Mirarab et al., 2014). In the identification of software services, the input types varies from the use of source code to traditional legacy systems.

Table 2.3 Types of Input used in software service identification methods (sSIMs)

Input type	**Description**	**Approaches**	**Total**
Source code analysis	is an extraction of static or dynamic information for automated testing of source code or artifact for debugging program or application(Binkley, 2007)	Adjoyan et al. (2014), Ricca et al. (2009), Alahmari et al. (2010),	3
Legacy system	is existing non service-oriented software assets or mainframe-based applicationsZhang et al. (2005)	Zhang et al. (2005), Alahmari et al. (2010) Rim Samia et al. (2004)	3
Use Case	a sequence of graphical actions or variants which produces an observable result of value(Si et al., 2009), beneficial by human actors	)Fareghzadeh (2008), Vemulapalli and Subramanian (2009), Kim and Doh(2009)Kim and Doh (2009)	3
Activity diagram	is graphical representations that capture the flow of actions and control of organizational process using Unified Modelling Language (UML)	Vemulapalli and Subramanian (2009)	1
Database	is the transformation of collections of relational information into sound web services (Baghdadi, 2006)	Baghdadi (2006)	1
User interface	is the look and feel design, visible to and processed by different user to define service requirement(Mani et al., 2008)	(Mani et al., 2008), (Quartel et al., 2004)	2

2. Business Services

   A business service is a comparatively aggregated business logic or action used in describing business function or goal in an organization (Cai et al., 2011; Mirarab et al., 2014; Zadeh, 2011). In the identification of business services, there have been various usage of input types as described in (*Table 2.4*).

Table 2.4 Types of Input used in business service identification methods (bSIMs)

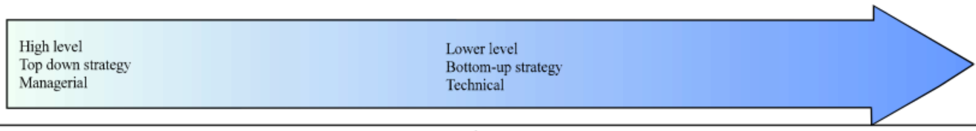
Input type	Description	Approaches	Total
Business process	is defined as a set of one or more linked procedures or structured set of activities executed following a predefined order by (potential several) actors (humans, computers and/or machines) in an organizational and technical environment which collectively defines the organizational objectives or business goal Chinosi and Trombetta (2012b).	Arsanjani et al. (2008), Fareghzadeh (2008), Jamshidi et al. (2008), Azevedo et al. (2009),Bianchini et al. (2009, 2011),Bianchini et al. (2014), Ma et al. (2009), **?**Kohlborn2009, Guan et al. (2012),Mohamed et al. (2014), Kang et al. (2008),Klose et al. (2007),Chen et al. (2005), Wu et al. (2013), Amsden (2010), Mani et al. (2008), Ricca et al. (2009), Inaganti and Behara (2007)	19
Master data and logical data models analysis	is the extraction of business rules from top-level abstraction business concepts derived from used data from business processes, organizational units and information systems (Huergo et al., 2014; Nigam et al., 2003)	Huergo et al. (2014)	1
Enterprise architecture goal	is the alignment of business and IT application architectures, highlighting a unified business goal	Suntae et al. (2008), Rim Samia et al. (2004)	2

Table 2.5 Types of Input used in combined (software and business service identification methods (sbSIMs))

Input type	Description	Approaches	Total
Mix	the combination of traditional legacy system and other types Mirarab et al. (2014) or the collection of different input types	Ricca et al. (2009), Jamshidi et al. (2008), Fareghzadeh (2008), Mani et al. (2008), Zhang, Zhang and Yang (2004), Cho et al. (2008),Flaxer and Niga (2004)	7

Also from the comparison done by (Zadeh et al., 2012) regarding the various inputs types, the criteria used are namely: Machine readability, interaction details, goals coverage, possibility to decomposition, clarity, choreography, easy to achieve by SMEs. As shown in figure 5, it shows the comparison of the input type, labelling the degree of fulfilment regarding to each criteria.

Our research will solely consider the use of business process model and documents because it exposes the major criteria like clarity, choreography, readability, interaction details and the decomposition of business process model easily expose business functions. (Gu and Lago, 2010) stated that by studying various number of existing input types for service identification, they deduce that "*it is quite understandable that a large number of SIMs widely used business process as their input type because of the nature of services.*"



Criteria / Input	Enterprise goals	Domain knowledge	Business process: BPMN	Business process: C3	Use case	Activity diagram	Source code	Data base
Machine readability	○	○	■	○	◆	◆	■	■
Interaction details	○	◆	■	◆	◆	■	■	■
Goals coverage	■	■	◆	◆	◆	○	○	○
Possibility to decomposition	■	■	■	◆	◆	◆	◆	○
Clarity	■	◆	◆	◆	◆	◆	○	○
Choreography	■	○	■	◆	◆	◆	◆	◆
Easy to achieve by SMEs	■	○	○	◆	◆	◆	■	■

■: The input type completely fulfils the criteria , ◆: Inputs that only fulfil some part of the criteria, ○: Signifies that the related input cannot fulfill the criteria

Fig. 2.21 Comparison of the service identification input types [(Zadeh et al., 2012)]

The comparison results considered the use BPMN, enterprise goals and existing system assets(source code and databases) as input type for any small and medium enterprise SMEs(Zadeh et al., 2012). In this regard, the use of business process models, enterprise goals,and existing system assets are justified in fulfillment to the criteria which are need for.

2.5.2 Service Identification Methods

In creating SIMs, the type of service to be identified is either software services or business services, which the components (the input type, strategy, approach and technique) used are different from service type to its usage. In Table 2.6, it shows the software service identification methods by different publications, likewise in Table 2.7, it shows the business service identification methods by different publications.

Table 2.6 Software Service Identification Methods (sSIMs)

	Adjoyan et al.(2014)	Jamshidi et al. (2008)	Ricca and Marchetto(2009)	Alahmari et al.(2010)	Zhang et al(2005)
Input type	Legacy source code	enterprise business model	source code	Legacy source code	Legacy systems
Strategies	bottom-up	top-down	meet-in-the-middle	bottom-up	bottom-up
Approach	Migration approach 1. Object-to-service mapping model, 2. Quality measurement model of services 3. Clustering process	clustering technique, named Elementary business process and business Entity Affinity analysis Technique (EEAT) is used	The quick and dirty approach	Service Identification framework- 1. Analysis and Re-engineering stage, 2. service identification elements 3. service evaluation	Clustering approach
Technique	analysis	guideline	guideline	guideline	analysis

Table 2.7 Business Service Identification Methods (bSIMs)

	Arsanjani(2008)	Fareghzadeh(2008)	Azevedo et al.(2009)	(Bianchini et al., 2009)	Ma et al.(2009)	Kohlborn et al(2009)
Input type	business process model	business process, requirement, visions and ontology	business process model, system requirement, business requirements	business process model	business process model	business model and context
Strategies	meet-at-the-middle	meet-at-the-middle	top-down	top-down	top-down	top-down
Approach	1. goal-service modelling (GSM)., 2. domain decomposition, 3. existing asset analysis	1. Input analysis, 2. business use case modelling, 3. service taxonomy	1. Selection of activities, 2. identification and classification of candidate services, 3. consolidation of candidate services.	Process-to-Service methodology 1. business process analysis, 2. candidate service identification • value-based service identification, • candidate service refinement 3. candidate service reconciliation	1. Measurement for service identification • business activities partitioning, • measure the features of services in the portfolio based on design metrics, • conduct overall evaluation of the identified services portfolio by normalizing the metrics' values 2. service portfolio modelling	Consolidated approach
Technique	Guideline	Guideline	Guideline	Algorithm	Guideline	Guideline

2.5.3 Service Identification Techniques

In several publications(Bianchini et al., 2014), (Adjoyan et al., 2014),(Kohlborn et al., 2009b), different techniques have been introduced and used for service identification. Such as systematic mapping study and analysis, pattern, information consolidation, algorithm, guidelines and ontology which produces formal, informal or abstract results. Formal techniques are codify formulas or rules to specify how services are identified (Gu and Lago, 2010), such as algorithm, ontology, pattern and information manipulation (*Table 2.8*). Informal techniques are gives a guided routine or advice on how to identify services(Gu and Lago, 2010), such as guidelines and analysis (*Table 2.9*)

Table 2.8 Formal techniques used in SIMs (Mirarab et al., 2014),(Gu and Lago, 2010)

Type	**Description**	SIMs	Total
Algorithm	A set of formal rule to be followed to solve a problem	Azevedo et al. (2009), Kim and Doh (2009), Dwivedi and Kulkarni (2008), Zhang et al. (2005), Wang and Wang (2006), Mani et al. (2008), Jamshidi et al. (2008)	7
Ontology	a scientific conceptualization of an idea or domain knowledge	Bianchini et al. (2014), Bianchini et al. (2009), Chen et al. (2005)	3
Pattern	is the formalization of recursive solution to a recursive problem	Baghdadi (2006)	1
Information manipulation	is a way of looking into useful information as a text process technique, such as information retrieval or textural similarity analysis	Kim and Doh (2007)	1

Table 2.9 Informal techniques used in SIMs (Mirarab et al., 2014),(Gu and Lago, 2010)

Type	**Description**	SIMs	Total
Guidelines	a general predefined regulations, principle used to get a result.	Klose et al. (2007), (Kohlmann and Alt, 2007), Inaganti and Behara (2007), Ricca et al. (2009)	4
Analysis	detailed examination, interpretation and reasoning of specific data	Chen et al. (2005), Amsden (2010), Fareghzadeh (2008), Kohlborn et al. (2009b), Cho et al. (2008), Aversano et al. (2008)	6

2.5.4 Service identification strategies

In the process of identification of services, several publications have has discussed different strategies of achieving their proposed approach. Some researches propose the top-down strategy which uses the business process model or business requirement as the starting point, whereas some other publication utilize the information systems as the starting point in a bottom-up approach. Other approaches combine both points of views producing an integrated approach called meet-at-the-middle approach. It can be said several publications utilize top-down strategy more that any other strategy, as shown in *Table 2.10*.

Table 2.10 Strategies used in SIMs

Strategy	**Description**	**Approaches**	**Total**
Top-down	The starting point of top-down approach is the business process model, by decomposing the business process into sets of tasks that produces a finer grained functionality (Mani et al., 2008)	Bianchini et al. (2014), Ma et al. (2009), Papazoglou and Van Den Heuvel (2007), Rim Samia et al. (2004), Kohlborn et al. (2009b), Kim and Doh (2009)	6
Bottom up	this approach utilizes the existing application or information systems as the starting point by analyzing it to identify re-usable functionality which are further identified as services **?**	Hepp et al. (2005)	1
Meet in the middle	is the integration of both perspectives (top-down and bottom-up) into a hybrid strategy Kim and Doh (2009)	Arsanjani et al. (2008)	1

2.6 Current Research and Method in Identifying Business Services

In our research, we studied over 90 research works related to service identification. We analysed the service identification method in each work, comparing the basic characteristics, technical context, design principle and their method of engineering of each service identification methods. To derive a concrete comparison results, we short-listed the service identification method based on certain criteria for choosing the 10 publications which are as follows:

1. Formality of the technique: One major criterion is the formality used, whatever the method or approach proposed by authors, it is important to see the background terms of usage or implementation process. The majority of the service identification methods chosen have documented their methods by using case studies, formal or informal techniques. Formal description allows for better understanding and analysis, and the subset of formally described methods covers a broad range of approaches. Therefore, the chosen of selection is based on formal technologies which is (semi)-automatic approach to identify a good amount of services.

2. Popularity of the papers, reuse or improvement of the method by other service identification methods or same author: In choosing the service identification methods, the methods that have been referenced i.e. well-cited more than two times or improved upon are considered, for instance, Bianchini et al. (2009) method was improved upon in 2014.

These criteria give us the understanding of the differences, similarities and the short-coming of several numbers of publications in service identification methods. Selected corresponding evaluation criteria are used to give an overview of the review and comparison of the related systematic literature and scientific journals are selected to pave way towards the understanding and foundation for current service identification methods. From the comparison of the service identification methods, most of the approaches lack rigorousness, constructiveness and re-usable features in (re)-creating service(s) in any business case with the right level of granularity for maintainability and re-usability purposes across the enterprise. Therefore, it is important to note that our method will consider and develop upon the shortcomings of the current service identification methods. In achieving these, we adopt the top-down. With the growing numbers of service identification methods (SIMs) in the past decade, there has not been an intense in-depth comparison of the methods. To create a new SIM, basic knowledge about the type of service to be identified has to be known i.e. understanding of services. There have been several publications which are business process-oriented, software process-oriented and also consolidated approach which is both software and business process oriented. To further understand the characteristics of SIMs, the following are looked into namely:

1. understanding of services- SOA paradigm of each SIM proposed, business-oriented granularity, development direction,

2. Technical context of SIMs- this criterion describes the technical knowledge and quality of SIM namely: orchestration vs choreography, criteria of information technology,

interaction with user, call frequency, strong cohesion and loose coupling, technical standardization, functional standardization, use of open standards, service performance,

3. Method of engineering in SIMs - activities, results, sequences of activities.

2.6.1 Comparison of SIMs

The comparison of SIMs‘ methods is based on the criteria that illustrated in the SIMs-based criteria in previous sub-section presented in (*Table 2.11, 2.12 and 2.13*). In the table, we use a trivalent scale (-, +, ++) with the following semantics: - stands for very lightweight, while '+' represents moderate degree of prescription, and ++ marks highly prescriptive to analysis the interaction of users, call frequency, activities, results and evaluation method.

From table 2.11, 2.12 and 2.13, it can be seen that publications used more of business process model as their input type showing that more work is been done into realizing the right identification of business services, but the downside of table 13 is that same publications has illustrated informal techniques in the identification of business services, only Bianchini et al. (2009, 2014, 2011) demonstrated their approach using algorithm.

Klose et al. (2007); M. (2006) presented the principles of identifying services using the quality metrics (e.g. low coupling, high cohesion and high level of granularity). Jamshidi et al. (2008) presented their approach by considering the enterprise business process model and enterprise entity model and proposes an method for enterprise software service identification. Arsanjani et al. (2008), as well as Zimmermann et al. (2009), describe what an overall approach could look like, but they fail to give more information, as their approaches are proprietary.

In this report, criteria concept used in Kohlborn et al. (2009b) and Klose et al. (2007) are used for comparison of current methods in the last decade.

Table 2.11 Comparison of service identification methods

	Klose et al(2007)	*Kohlmann and Alt (2007)*	*Arsanjani et al (2008)*	*Jamshidi et al (2008)*	*Fareghzadeh(2008)*	*Kohlborn et al (2009)*	*Yun et al (2009)*	*Azevedo et al (2010)*	*Guan et al (2012)*	*Bianchini et al (2011) (2014)*
Basic characteristics										
Understanding of services	Business process-oriented	business process-oriented	Business process-oriented	Software process-oriented	Business process-oriented	Business and Software process-oriented	Business process-oriented	Business process-oriented	Business-oriented	Business process-oriented
SOA paradigm	Architectural concept	Architectural concept	Architectural concept	Architectural concept	Architectural concept	Comparison and architectural concept	Architectural (DFD) concept	Architectural concept	Graphical concept	Analysis using algorithms
Business-oriented granularity	Fine	From coarse to fine	Coarse	Coarse	fine	fine-grained	Controlled granularity	Depends on usage	Coarse	Fine
Development direction	Hybrid	Hybrid	Hybrid with focus on top-down	Top-down	Hybrid	Consolidated approach	Bottom-up	Top down	Top down	Top down

Table 2.12 Comparison of service identification methods

	Klose et al(2007)	*Kohlmann and Alt (2007)*	*Arsanjani et al (2008)*	*Jamshidi et al (2008)*	*Fareghzadeh(2008)*	*Kohlborn et al (2009)*	*Yun et al (2009)*	*Azevedo et al (2010)*	*Guan et al (2012)*	*Bianchini et al (2011) (2014)*
Technical Context										
Orchestration vs Choreography	Orchestration	Orchestration	Not clear	Choreography	Orchestration	Choreography	Choreography	Orchestration	Orchestration	Orchestration
Criteria of information technology	Design principles for SOA	Reusability	Reusability and flexibility	Reusability	Autonomy and reusability	Reusability	Not considered	Reusability	Not considered	Reusability and flexibility
Interaction with users	++	–	–	–	++	++	++	++	–	++
Call Frequency	–	–	–	–	–	–	–	–	–	–
Strong cohesion and loose coupling	Not considered	Not considered	Not considered	Not fully considered	Considered, not used	Considered, not used	Considered	Not considered	Not considered	Considered and used
Technical standardization	None	None	None	None	None	None	DFD Modeling	None	Graph Theory	None
Functional standardization	None	None	None	None	None	None	service-oriented analysis and design (SOAD)	Semantic analysis using Heuristics	Topology clustering	Quality metric
Use of open standards	SOA standards	SOA standards	SOA standards	SOA standards, case modelling standards	SOA standards	SOA standards	SOA standards	SOA standards	Not considered	SOA standards
Approach	Decomposition of business process (with BPM) and SOA principles	BPM, asset analysis	Goal service modelling, domain decomposition, asset analysis	Use case analysis, enterprise service modelling	in-depth business process analysis, asset analysis and goal service modelling	Comparison, capability decomposition and mapping of domains and capabilities	DFD Based SOAD Approach Analysis	Systematic analysis from business process models	clustering algorithm	value analysis, structural analysis, quality metric analysis
Service performance	Not considered	Value Performance		Metric performance	Not considered	Not considered	Not considered	Not considered	Not considered	Considered

Table 2.13 Comparison of service identification methods

	Klose et al(2007)	*Kohlmann and Alt (2007)*	*Arsanjani et al (2008)*	*Jamshidi et al (2008)*	*Fareghzadeh(2008)*	*Kohlborn et al (2009)*	*Yun et al (2009)*	*Azevedo et al (2010)*	*Guan et al (2012)*	*Bianchini et al (2011) (2014)*
Method Engineering										
Activities	++	+	++	++	++	++	++	++	+	++
Results	+	+	++	++	++	++	+	++	+	++
Sequences of Activities	Sequential	Sequential, iterative where applicable	iterative and fractal	Sequential	Sequential	Sequential	iterative	Sequential	graph network	Sequential
Design principle										
Type of Input	Application domain	Application domain	++	Business process	Mix	Mix	Data	business process	Business process	Business process
Type of Output	Process Service	Data service	Process service	Software service	Process services	Process services	Software service	Business service	Business service	Process services
Output format	informal service specification	informal service specification	informal service specification	informal service specification	informal service specification	informal service specification	informal service specification	informal service specification	formal specification	formal specification
Strategy	Business Process Decomposition	Business Functions Decomposition	++	Business Process Decomposition	Existing Systems Functions	Business Functions Decomposition	Business Entity Objects	Business Process Decomposition	Describe business process simply by employing Graph Theory and network topology Clustering algorithm	Business Process Decomposition,Value Analysis
Techniques	Guidelines	Guidelines	++	Algorithm	Analysis	Analysis	Analysis	Guidelines	Algorithm	Ontology
Documentation	process model, formal criteria, case study	process model, case study	No case study	process model,criteria	No case study	criteria, case studies	Case study	Case study	Case study	process model, formal criteria, case study and quality evaluation
Evaluation Method	++	+	+	+	-	+	-	-	-	++

2.6.2 Drawbacks of the existing methods

With a considerable number of publications on service identification, there have been a few drawbacks in identification of right services. The comparison in *Table 12,13,and 14* show the drawbacks in identification of right services as shown below:

1. Design principle of services

   From *Table 12* it can be seen that most of the publications are business-oriented but they still exhibit the following:

   - Different input, different output

     From *Table 13*, the design principle showed the input type, output type, output format. Four of the 10 papers in the Table 14 start with business process model and context (Azevedo et al., 2009; Bianchini et al., 2014; Guan et al., 2012; Jamshidi et al., 2008), whereas (Kohlborn et al., 2009b) considered business process, business and software context, and few others considered the use of application domain as their input type like (Kohlmann and Alt, 2007).

     As a result of different input types, the output type are likewise different, some methods focused on outputting business services whereas others focus on software services.

     The output format of the identified services follows either informal or formal specifications. Guan et al. (2012) and Bianchini et al. (2014) provided a formal specification for their output either with the use of algorithm or ontology.

     These differences in input type can create a set-back in creating formal service specification as seen in the output format in Table 2.13.Bianchini et al. (2009, 2014, 2011); Guan et al. (2012) produce a formal output for service identification, while other paper were informal.

- Different techniques and different strategies

  From the various publications, it is seen there is no systematic methods for service identification using business analysis. As seen in *Table 2.11*, development direction differs significantly, ranging from bottom-up to top-down strategy or the hybrid strategy, from source code extraction to business domain analysis, from ontology-based process to guideline-driven process. This will also be a set back in creating a formal service specification.

2. Formal representation of services identification approach

   Some papers made mention of the quality metrics which was adopted from software engineering which was applied into improving work-flow design. The use of use case and existing asset analysis is also used for identification of services. Adaptation of quality metric in (Bianchini et al., 2014) used to measure the functionality, reliability and usability within service or related services has shown the formal representation and specification of services.

3. Re-usability of Service

   Adjoyan et al. (2014); Bianchini et al. (2014); Kohlborn et al. (2009b); Ma et al. (2009) consider the modularity and autonomy based on their design requirements which conforms to the principle of cohesion and loose coupling. The metric for getting these principles are heterogeneous, reason why resulting services do not have a guaranteed fit into inter-organizational functional needs. The re-usability is not yet fully dealt into.

4. Unified methodical approach for service identification

   One of the most important aspect in constructing service-oriented solution is the identification of services with the right level of abstraction (Guan et al., 2012), which has become a trend of computing paradigm (Chen et al., 2005; Zhang et al., 2005) to describe business logic, self-containment and functionalities (Adjoyan et al., 2014). The need for definition and identification of services has been recognized by several authors Fareghzadeh (2008); Josuttis (2007); Paper (2007), who also recommend the use of business process models as a starting point, but there is no aligned method of designing process model. Currently, several techniques have immensely improved the design quality of service-oriented applications but there exist no unified methodical approach for identifying services, most current approach largely lack systematic efficiency. There have been several techniques and algorithm for identifying services,(Adjoyan et al., 2014; Bianchini et al., 2014; Guan et al., 2012; Kohlborn et al., 2009b; Ma et al., 2009; Wu et al., 2013; Zhang et al., 2005). Service identification is very challenging and is error-prone, any mistake made can result into incorrect Service Selection(Bianchini et al., 2011; Ma et al., 2014) which will affect the overall SOA-based systems.

5. Documentation

   Major researches have mainly focused on the aspect of the organizations‘ business process and how to impact and improve it. The past research has outlined and enlighten the background knowledge, history, discipline and business service identification approach which Bianchini et al. (2014); Guan et al. (2012) considered the use formal criteria for identification of services.

It can be seen that publications used more of business process model as their input type showing that more work is been done into realizing the right identification of business services, but the downside of comparison is that same publications has illustrated informal techniques in the identification of business services, which means manually case by case analysis involved, Bianchini et al. (2014) demonstrates their approach using algorithm, which could deal with requirements of large and complex systems. Adjoyan et al. (2014); Bianchini et al. (2014) presented the principles of identifying services using the quality metrics (e.g. low coupling, high cohesion and high level of granularity). Also publication (Jamshidi et al.,

2008) presented their approach by considering the enterprise business process model and enterprise entity model and proposes a method for enterprise software service identification. Kohlborn et al. (2009b); Kohlmann and Alt (2007) describes what an overall approach could look like, but they fail to give more information, as their approaches are proprietary.

The whole point of comparison is to improve on the weaknesses or short-comings of past research work, thereby creating a tweaking and fine tuned service identification methods. This is one of the motive of this research work which give an intense formal definition of each approach, technological terms and techniques that will be used in realizing the aim of the research, adding a layer of universal solidity into the service identification, creating workable solution for different organizations.

Formal techniques are codify formulas or rules to specify how services are identified (Gu and Lago, 2010), such as algorithm (Azevedo et al., 2009; Dwivedi and Kulkarni, 2008; Jamshidi et al., 2008; Kim and Doh, 2009; Mani et al., 2008; Wang and Wang, 2006; Zhang et al., 2005), ontology (Bianchini et al., 2009, 2014; Chen et al., 2005), pattern (Baghdadi, 2006) and information manipulation(Kim and Doh, 2007). Informal techniques are gives a guided routine or advice on how to identify services(Gu and Lago, 2010), such as guidelines (Inaganti and Behara, 2007; Klose et al., 2007; Kohlmann and Alt, 2007; Ricca et al., 2009) and analysis (Amsden, 2010; Aversano et al., 2008; Chen et al., 2005; Cho et al., 2008; Fareghzadeh, 2008; Kohlborn et al., 2009b) .

In creating a new service identification methods, the development direction intend to follow the following steps: manipulate the data elements (top-down strategy), the analysing and determination of the values for respective data elements using cluster algorithm, the structural analysis and formalization of the alternatives operations based on ontology and patterns, service quality analysis using the quality metrics like cohesion and coupling, and service performance like processing time, service response time and other considerable and applicable time.

2.6.3 Requirements For Creating A New Method

Informed by the properties of existing approaches, two types of requirements, industrial and academic, can be distinguished for a new method. Industrial requirement involves a large quantity of processes, where a large majority of processes involves at least 100 activities, and several numbers of services can be derived from the activities.

A survey shows that there are increasing research interest in the management of industrial collection of process since 2005.

Examples of such collections, often described in the literature include: the BIT process library (735 process models), the SAP reference model (604 process models), a reference model for Dutch municipalities (around 600 process models) Suncorp's process repository contains more than 3,000 process models for their insurance sector, with models ranging from 25 to 500 activities (Dijkman et al., 2012), while China Mobile's process model repository contains more than 40,000 models (Gao, 2013). In total, a legacy system could easily have over 302,000 activities and over 600,000 data elements. For migrating the legacy system to service-oriented BPM systems, it is impossible to identify all the services manually which satisfy all constraints (such as reuse principle).

Based on the drawbacks of the existing methods and the comparison tables (*Table 2.11,2.12 and 2.13*),the weaknesses of the current service identification approach (value-based service identification (Bianchini et al., 2014), Goal service modelling, domain decomposition, asset analysis, Use case analysis, enterprise service modelling, in-depth business process analysis, asset analysis and goal service modelling approach).Therefore in developing new service identification methods, there are series of requirements have to be considered:

1. **Repeatable approach and undefined data type size**

   It is important that the workable approach should go through cycle of (re)checks for any error i.e. variability in the result for a given case can be detected early. Most of the current approach and method never considered this factor. Also, it important that our approach will not have a specified number of data type to use. It is required to take more or fewer data types for any case given.

Other important requirements which will be presented in our future research publications and thesis are service compatibility, Service composability (M., 2006), call frequency, interaction with users, re-usability, service functionality and self-containment (Adjoyan et al., 2014), service interoperability (Zimmermann et al., 2005): It shown that some of the researches on service identification are omitting the important part (Fareghzadeh, 2008), which is the interoperability of identified services which describes the technicality (e.g. transfer protocols, data formats)conceptual details(clearly unified and specified terms and standardized data models) which are intended to be open, platform independent and widely diffusible.

2. **The type of information system**

   All organizations achieves their business goals, interest and needs through service sharing, usage and exchange, which contains information. These information could contain free flow or secretive data which can be accessed by the right agencies or individuals of a company. In an organization where services are created, reuse and exchange like the universities and colleges, then thus seem to be an open information system. In a situation, where there is collaboration between two or more organization, then it is called closed information system, i.e. tight security is balanced with free flow of information. Therefore, in developing a new service identification methods, the following has to be put into consideration namely, current practices and policies (Cai et al., 2011), information communication technologies in place (Fareghzadeh, 2008), current level of information sharing and organization culture.

3. **Service Interoperability**

   None of the past publications considered the service inter-operability. In supporting ever changing business demands, the degree of service inter-operability should relate to the service change which is resistant to changing information system. This will describe the technicality (e.g. transfer protocols, data formats), conceptual details (clearly unified and specified terms and standardized data models) which are intended to be in (open, closed) information system, platform independent and widely diffusive.

4. **Modularity and Autonomy**
   Some researchers consider the modularity and autonomy based on their design requirements which conforms to the principle of cohesion and loose coupling. Byrd et al. (2001)(Espinosa et al., 2011) highlighted that "..*the benefit of modularity is that it provides organizations with the ability to add, modify and remove business processes with little or no widespread effects*". Current methods does not consider the feasibility of add, modifying and removing of services especially when business requirement changes, this increases the agility and versatility of organizations. The metric for getting these principles are heterogeneous, reason why the resulting services do not have a guaranteed fit into inter-organizational functional needs. Abuhussein et al. (2014) highlighted that "*the autonomous nature of services implies that services communicate to maintain control over the resources and to coordinate with other components of the SoA*". Majority of the methods by other authors are independent, therefore the method proposed can be fully implemented on any platform including the cloud.

2.6.4 Discussion and Conclusion

This chapter has given broad background knowledge that guide this thesis from the inception (*problem*) to the Evaluation (*application of the solution*). This Literature focus on the improvement of Business IT alignment (BIA) which help to improve and create adaptable software systems from changing business needs, realising speedy business growth and profit. The literature presents glance of BPM, its history and life-cycle, and the technical challenges.

From the concept of BPM, several authors were able to deliver the benefits of adopting BPM. Organizations all over the world are subject to changes[20] due to deregulation of market, global competitions, and increase of business or customer needs (Kohlborn et al., 2009a). Technologies and concepts in the Business Information Technology (BIT) world are gearing to these changing factors adaptation. The emergence of these technologies has accelerated the flexibility and variability of BPM system, likewise the adoption of SOA and its standards like Web Services has opened a wide range of automation and integration (Aiello et al., 2010). Cloud Computing has increase the number of researches towards BPM outsourcing. To understand the challenges of BPM, service-oriented computing (SOC) has to be included in the literature. SOA and service-oriented modelling are reviewed. Likewise, service-oriented business process management (SOBPM) was also reviewed.

With all these paradigms and reviewed topic, it gives an overall knowledge of the research work to be undertaken in the next chapter.

[20] Change is constant

Chapter 3

Service Identification Requirement for Enterprise Information System

3.1 Introduction

The growing complexity of fast-changing business environments pose different challenges with intra and inter-organization interaction, among them are the integration of systems between autonomous and heterogeneous EISs[1]. As cooperative environment progressively evolve, organizations integrate a number of different applications, protocols and service format, linking Business-to-Business (B2B) such as Covisint[2] (*In general, EISs are applicable to organizations and among organizations*), Customer-to-Business (C2B) such as priceline.com[3], (C2B not a big concern to EIS) to fulfil a business need and objective.
Likewise, organization are also known to be internet-enabled business environment (cf. fig 3.1), improving and attaining a more flexible, adaptable, less rigid organizational philosophy which reduces operation and maintenance cost and enhance faster total system and business

[1]Enterprise information system is channel across intra or inter–organization to integrate, enhance, support and disseminate business processes on a robust platform
[2]https://www.covisint.com/
[3]https://www.priceline.com/

quality than that provided by the inflexible and mechanistic organization (Komoda, 2006).

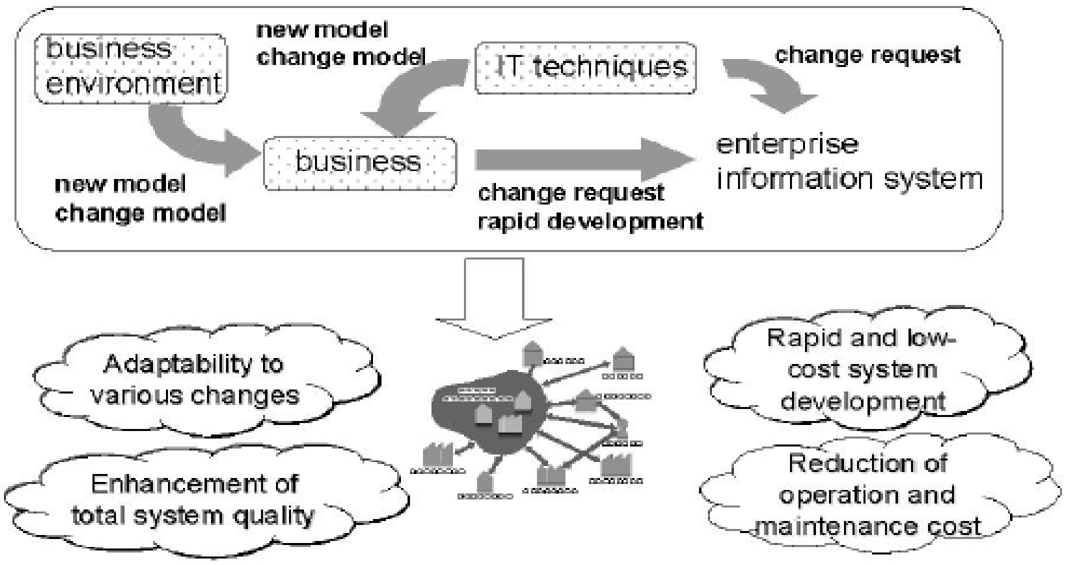


Fig. 3.1 Change of Business Environment (Wetherly and Otter, 2014)

With the introduction of SOA concepts into business system, it re-defines and structurally change the system architecture, transforming the closed monolithic platform-type to multi-level applications-type, developing industry-specific solutions and deepens lateral communication.

Nowadays, organization's internal processes are accessible by other organization(s) for openness and transparency by using the Web- technologies like Web services and markup languages like extensible markup language (XML) and extensible HTML.

At this point, identification of services within intra and inter-organizations can be tedious. Also, of particular interest is that organizations are finding it difficult to keep up with the speed of business change. As such, business operations and environments are unpredictable, leading to increased costs as changes arise. Therefore, it is more beneficial to have service-oriented organization which are more considerate, accurate, cost-efficient, productive and agile.

In this chapter, we present the level of capability of services in an organization type, and the requirement for services in each EIS and its fundamental properties. The benefit of defining the requirement for each system is that it presents an extensible-platform functionality for any organization to identify their services at real time. Also, we give general description and properties of single systems, collaboration, running an open and closed operation, and analyse the service solution for each involvement and operation.

3.1.1 Single systems

Single systems can be defined from different perspectives. The word "single" has been defined in Oxford dictionary as an attribute to quantify a noun[4]. In general context, single system can be said to be one system. In science, an organization with one system can contain bundle of sub-systems from different organizational units, e,g. organization system operating a software solution e.g. SAP[5] series of products running several sub-systems like Human Resource Management (HRM) and ERP systems is referred as a platform-based single system. Likewise, an organization system using one system containing several software applications is referred as a application-based single system e.g. Insurance company with several software applications for claim and risk management, fraud detection, product accelerator, risk management, reinsurance management and finance management.
Therefore, we can re-define single system as the combination one or more systematic structures containing several software applications and/or platforms (*containing bundle of sub-systems*) which are distinguishable for demonstrating the competencies, resources, and sharing responsibilities and benefits, in order to overcome market turbulence and achieve their business goals accuracies. For instance, a university has several departmental units' i.e.

[4]a word used to identify any class of people, places, or things
[5]SAP SE is a German multinational software corporation that produces enterprise software to manage business operations and customer relations

faculties, finance, marketing and house-letting departments etc., and those constituents and describes the potentials of the university. The university can be said to have a combination of several sub-systems (*using series of SAP products*) and platforms (*several software applications*). Organization with single system can either be open, closed single system.

Open single system

To define "open" single system, we use the "*usage by others*" perspective. A single system is "open" when several functions or applications can be accessed, shared and reused by other systems or users. Open single EISs are normally built on platforms of big organization. For instance, Amazon shop and Amazon platform, Amazon is not a typical EIS, it can be EIS when it support Amazon shops. Several things are built on Amazon platform which the HR is not open to the shops. Technically, an interface and governing standards are defined for the systems to interact, i.e. exchange of messages with systems (entities of different departmental units) is possible by using the standard, technology and purpose. The systems are interconnected with required resources using specified policies to achieve highly centralized system and service. The technical limitations in the openness feature of a system is the accessibility of process by different partners which can lead to hacking or bridge of data security.

Closed single system

Single system can also be defined using the "*usage by others*" perspective as system with limited or no access by people or application outside of the system, for certain reasons like security, bridge of trust and competition. Technically, closed single system is a secured interaction between the service provider and requestor in one or more systems or organizations. The degree of interoperability and standard are defined for secure networking with low centrality to the system(s) and users. The technical limitations of closed single systems are: less service composition and certification, no reuse of system applications and services by other systems, and the system are not available for extension leading to no degree of reusability. Closed single system can become open when, an enterprise decides to outsource its services from another business partner. Open single system can become closed systems as a result of a merger involving two organizations, in which one of their concerns is data security.

3.1.2 Collaboration

Collaboration is a broad and encompassing term (Jin and Ahn, 2006), and it can be defined in different context and needs clarification. According to Niehaves and Plattfaut (2010), the word "collaboration" is said to emerge from the study of organization boundary in management and organization research, driven by the growing concerns in the area of supply chain, production networks of interconnected organizations, collaboration dynamics, outsourcing, Virtual Organization (VO) (Afsarmanesh, 2007), and of development in information system. In this thesis, the concept "collaboration" and "network" are both treated from its related technical viewpoint and context. However, the concept "network" or "networking" is key term in many business and marketing researches (Fujimoto, 2008). In recent researches in business networking field, networking distinguishes the different forms of business by value chain orientation, life span, and degree of virtualization or hierarchical structure (Camarinha-Matos et al., 2010).

Camarinha-Matos et al. (2010) defined networking as the need to communicate with specified duration, decision making power features and relationship. Whereas collaboration is the mutual involvement of participants for joint goal/responsibilities with specific process(es) in which they share information, resources and capabilities (Afsarmanesh et al., 2009) . Therefore, collaboration can be said to have common goals and participants, but can be defined in different technical viewpoints from networking.

Collaboration and networking has been used as one term to define networks of different entities (people, enterprise, computer systems etc.). Afsarmanesh et al. (2009); Camarinha-Matos et al. (2010) described collaborative network as a geographically distributed network of variety of entities (organization, people, machines) sharing resources and core competencies to add competitive advantages to their business, which are largely autonomous and heterogeneous in terms of their operating environment, business culture, and goals.

Collaborative network can involve the interaction of organizations, virtual organizations (VO) or virtual enterprises (VE) for certain advantages; the most relevant one is summarized in (Camarinha-Matos et al., 2010) namely: agility, complementary roles, competitiveness, resource optimization and innovation which are said to fit for purpose when it is specified by the collaborating partners. Therefore, collaboration network has to be explicitly designed, structured and managed to increase business potentials and efficiency.

In past action research studies (Afsarmanesh et al., 2009; Camarinha-Matos et al., 2010; Rajsiri et al., 2007), collaborative network have shown its wide diversity with the use of different architectural support, interaction and behavior patterns for different business purposes. From these studies, there exist a wide range of structures which differ from the process-oriented chain structure (e.g. in supply chain), to the project-oriented federated networks (e.g. in virtual organization).

We concentrate more on the term "collaboration" but not network because of relationships and interactions of multi-partner's business processes, from the peer-to-peer and hierarchical technological structure Papazoglou et al. (2007b).

These relationships and interactions demonstrate how multi-partners integrate with themselves by selecting the part of their business processes to make public depending on the agreed-upon standard. Schmidt et al. (2005) further propose a business process management (BPM) infrastructure which provides the run-time monitoring the execution of individual partner's open and closed business processes using the federated dashboard. Furthermore, the introduction of enterprise service bus (ESB) provided the multi-partners with SOA enabled and event-driven infrastructure that acts as an intermediary layer to supports a flexible connectivity between services, bridging heterogeneous environments (Chappell, 2004).

In this chapter, we classify collaboration into internal function, external function, and internal-external function collaborations using the classification in (Li, 2007).

Collaboration can be addressed in an internal function when an organization owns different systems that are in charge of the sourcing, supplies, and distributions using the supply chain network. Also, collaboration can be addressed in an external function network organization when an organizations that provides services, either directly or indirectly (second-tier supplier), using the production networks of interconnected organizations.

Collaboration can be addressed in internal-external function when an organization is responsible for the service distribution between locations internally and externally, using VO.

From all these classifications, collaboration has to be addressed by the relationship of partners at business, and/or system, and/or service levels, depending on the business situation and purpose.

For instance, UK Barclays bank is responsible for the local (internal) and global (external) distribution of their services for example, mortgage and loan service, credit and debit card service. For the calculation of account charges and money exchange rate on international transactions involving different clients in different countries, the collaboration of the bank's virtual organization can interoperate on an standardized interface to calculate and give charges to the clients and partners at real-time.

On a more fundamental level, Jin and Ahn (2006); Rajsiri et al. (2007) suggest that in internal collaboration; very few partners have fully achieved internal integration of their systems because the cost of collaboration Minson and Mueller (2012).

The full integration of systems towards collaboration are more efficient to achieve singular goal, but in reality, the complete integration is hard to achieve and very costly because most old systems are built on flexi-system. Therefore, it is not always possible to achieve full integration, regardless of the cost of integration and other factors like (*integration of old flexi-systems, time variants*). Achieving full integration i.e. collaboration will maximize business productivity, profit, efficiency and competition.

Technically, there are fewer issues with system interoperability and adaptability internally, for internal collaboration defines its clarity on what to achieve, and how to achieve it.

Collaboration at service level

At service level, collaboration arises from the interaction with each other's exposed services by providing a set of capabilities which are visible to recognizable business functions of partners. Organization in collaboration can communicate with what it does known as business capabilities, without revealing how it does it Luo et al. (2005), which are viewed as "black boxes". The business capability to some degree needs to be externally apparent for collaboration without revealing how it is realized internally (*depending on agreement on policies*).

The degree of collaboration differs in different environments at system and service level. Therefore, to define open or closed collaboration, we use the "*usage by others*" perspective. From the "*usage by others*" perspective, collaboration is said to be "open" when several partners can access, share and reuse their systems. Technically, collaboration is can be said to "open" e.g. supply chain, production network when:

a.) there is defined standards for unified communication accessible by the partners,

b.) the partners have fully re-configured their services internally and externally, and scale up for fully integration by defining common vocabulary and processes, and specifying which partners performs what or use part of the actual business process.

On the other hand, collaboration is said to be closed when:

a.) the organizations emerging in different heterogeneous application domain have restricted joint policies and contract to which it results in personal interest e.g collaboration of bank's VO,

b.) the network in same application domain have secured joint integration for both business growth and interest e.g. collaboration of NASA with LEGO to create an intelligent education tool.

Technically, collaborative organizations is can be said to "closed" (e.g. virtual organization) when:

a.) there is agreed upon dynamic standards that help in mapping the functional interface guidelines, and functional interface, remotely accessible by the partners according to specified functional needs,

b.) there is secured service information (metadata) that are embed the required parameters to support the service partner's adaptation.

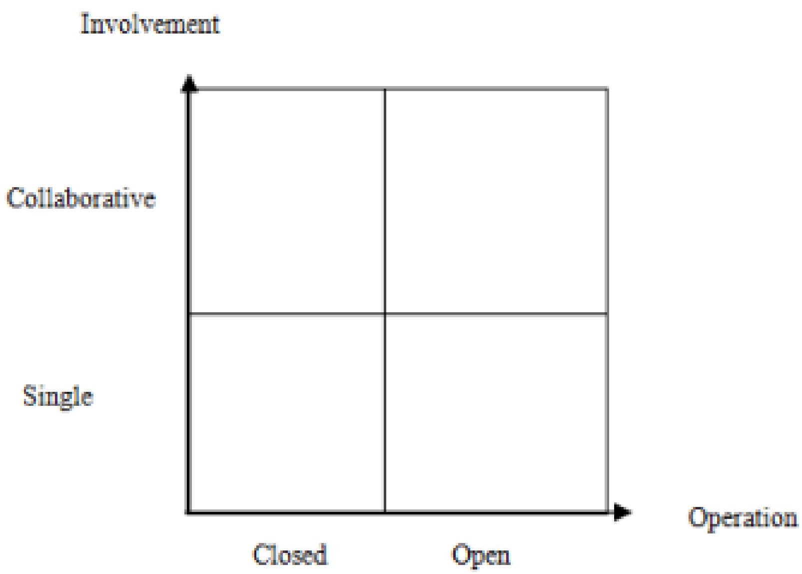


Fig. 3.2 Two-dimensional EIS

As mentioned above, there are different domains that EISs support, which one or many EISs can be used in one organization. Therefore, one conventional methodology cannot be used to identify service for organization with several different classifications and operation. In doing so, we state that an organization with one or many EIS can be classified into two dimensions, which will support identification of right services for enterprise system.

One dimension shows the representation of organization system, i.e. “organization system” dimension (Fig. 3.2). The organization system dimension denotes that an enterprise system can change from a single system to a collaborative network or in contrast. Another dimension represents operation of organizations which can be seen as open or closed for other partners, i.e. “EIS” dimension.

An EIS can be “open”, “closed” operation, and organization(s) and implement a “single", “collaborative" process. These terms are relatively and dynamically changing its interaction between business partners from open to closed, or in contrast. These results to the four axes as shown in figure 3.2, describing the four possibilities of an organization i.e. an organization can exist either as open single system, closed single system, open collaborative process, and closed collaborative process. For instance, an insurance company (*appears in open single system axis*) creates several policies which require several business processes for execution. A new policy “closed-garage car holders” is introduced into the competitive market which requires outsourcing data from the government “houses with garage” into their database. In other words, this policy requires collaboration with the government (*closed single system*), also the government needs to know the amount of cars in each county to evaluate road tax payment. This leads to joint venture between their business processes (*both partners now appear on open collaborative process*). Therefore, for faster development of new type of insurance policy and tax detail information, there must be fast approach to identifying service from the open collaborative process. Organization can be classified into an axis based on its requirements at system and service levels which may change dues to business needs.

Our highly distributed approach involves the monitoring of the changes as a result of variation in partner’s businesses, growing services demands, and scope of the organizations, and the identification of services within the intra or inter-organizations at real-time. Furthermore, the approach defines the degree of service orientation of each axis as the extent to which services are designed to easily composed, coupled and adapt in order to cope with the changing factors. With this approach, new requirement for service identification is proposed which can adapt to organization changes using the degree of service orientation of each axis.

3.2 Service Orientation Degree for Enterprise Information Systems

In recent business world, organizations changes as a result of variation in partner's businesses, growing services demands, and scope of the organizations. An organization is said to effective only if they are on-demand and adaptable (Cherbakov et al., 2005), resulting to the transformation of tradition enterprise to SOA supported EIS. A recent survey by Forrester shows that the rate of SOA adoption among enterprises is strong and increasing (Heffner et al., 2011), by using architectural style that increases emphasis on flexibility and efficiency. One of the key promises and benefits of SOA is the seamless integration of business services Cherbakov et al. (2005), by describing the service orientation.

Service orientation is the internal representation (or metaphorically its "DNA") toward services Oliveira and Roth (2012), which enables enterprise to react quickly to a frequent business demand Cherbakov et al. (2005).

Defining the degree of service orientation in each axis presents the overall tendency to deliver a service excellence. Johnston (2004) argues that service excellence is about "being easy to do business with". Following Johnston (2004) argument, we further argue that organizations need to have the requisite service orientation measurement that makes it reliably easy for intra or inter-organization integration. Therefore, there are growing needs for valid measurement scales that describe business overall internal service-based competencies. The methodology employed in this chapter builds upon prior research (Aldris et al., 2013). Aldris et al. (2013) describe the measurement of service orientation by degree as “the extent to which services are designed in such a way that allows them to be easily coupled, adapted and combined in order to cope with changing environment”, and provide service excellence in EIS.

For achieving a given goal, EIS interacts between each other depending on certain principle (Aldris et al., 2013).

Elvesæter et al. (2006) states that interoperability solutions should be driven by first, the business needs, and then the software solutions as the second. Furthermore, quality principles of a system (interoperability, adaptability and re-usability) should be driven by business and software needs at system level, and which has to conform to the existing SOA best practice at service level. Below are the selected quality principles at system level:

1. System Interoperability.

   This principle states the extent or level at which two or more systems can exchange information in a meaningful way (Aldris et al., 2013). Carney et al. (2005) lengthened the definition, in (Elvesæter et al., 2006) by adding the notion of purpose related to goal of interoperation and the notion of relation in the environment in which the entities exist. Interoperability is defined as the “ability of collections of communicating entities to (i.) share specified information and (ii.) operate on the that information according to a shared operational semantics (iii.) in order to achieve a specified purpose in given context”. (Panetto, 2007), in (Elvesæter et al., 2006), complements Carney et al. (2005) definition, stating that “interoperable systems are by necessity compatible, but the converse is not necessarily true”. A service is said to be interoperable when the level of message exchange between different services interconnected semantically or by agreed upon syntax. Service interoperability depends on level of information exchange between the services through distinct interfaces that specify the usage and behaviour of the systems. Interoperability problems may arise due to: organization type, different levels (department) in organization, different kinds of enterprise application as well as due to the varying levels of abstractions of the services. Therefore, the degree for interoperability depends on the type of enterprise system and the level of service contract standardization (e.g. the use of communication protocol), service abstraction and service loose-coupling. The higher the system level or cost of exchange of information, the higher the interoperability.

2. System Adaptability.

   This principle states how a system can accommodate changes within or outside of its environment (Aldris et al., 2013). Adaptability includes the scalability of evolving software, hardware and operational environment. Service adaptability is the level of service control over its environment and displays efficient request processing (Aldris et al., 2013). Adaptability also depends on the enterprise system and the degree of statelessness and autonomy. The higher the scalability of the service design, the higher the service adaptability, achievable by the level of statelessness and autonomy.

3. System Re-usability.

   This principle advocates that a segment of a system's asset can be used again in different context to add new functions with slight or no modification (Aldris et al., 2013). Re-usability of software systems is possible when a system successfully adapt to changing needs with the environment. Service re-usability is the level at which a service can participate in multiple composition in different context to add new functions with slight or no modification (Aldris et al., 2013). Service re-usability depends on SOA design principles namely: discoverability, genericity and composability.

Based on SOA design principles which can be found in different works (Aldris et al., 2013; Erl, 2008; Papazoglou et al., 2007a), we refine the quality principles which play key role in service orientation at service level:

1. Service Contract Standardization.

   Service contracts define the capabilities and its relationships of services. Standardization of service contract enables efficient communication amongst services (Aldris et al., 2013), which has to be defined before the implementation of the services. This principle ensures consistent communication, transparency and helps in service contract reuse.

2. Service Loose Coupling.

   This principle ensures that services can condone changes to application instances without affecting other services. This principle must be studied from different negative types of coupling as Contract-to-Functional Coupling, Contract-to-Implementation Coupling, Contract-to-Logic Coupling and Contract-to-Technology Coupling, see (Aldris et al., 2013).

3. Service Abstraction.

   This principle turns services into "black box", publishing only the required information need about the services to the consumer. This information can be changed as service design changes e.g., when a service is composed of other services.

4. Service Statelessness.

   This principle requires that services in SOA-based system are to avoid the management of state tasks (e.g. keeping trace of interaction-specific) (Aldris et al., 2013). Service Autonomy. This principle advocates that services have maximum control over underlying runtime execution environment (Aldris et al., 2013). For instance, service A and service B are not designed to have direct access to the same object on a database.

5. Service Discoverability.

   This principle ensures that services have the ability to be effectively discovered and interpreted by supplementing services with communicative metadata (Aldris et al., 2013).

6. Service Composability.

   This principles represents the design approach to which services are effective in service compositions to create new services (Aldris et al., 2013).

7. System Modularity

   This is defined as the level at which a service can be changed without have huge impact on the whole or other services. The degree of modularity depends on indirect service dependence or coupling, decomposability, composability, continuity.

8. System Replace-ability

   This principle ensures that a system can substituted with another in the same environment for the same purpose (Aldris et al., 2013). A system might be replaced by other system creating disruption in the services as a result from new version, model etc., to upgrade or prevention of failure or maintenance. For example, unavailability of certain service in an enterprise system application will be transparent to the consumers, instead of creating new service from the scratch for availability; the gap can be filled by replacing it the available one. The degree of replace-ability depends on direct precedence, causal footprints and protocol, and causal behavioural profiles.

3.3 Service Orientation Degree of Open Single Enterprise Information Systems

Technically, it is important to distinguish between interoperability and adaptability since failure to do so, sometimes confuse the debate on how to achieve them at system and service level, which result to the classification of the open single EIS into two perspectives. Open single systems have been addressed in two perspectives namely, i.) platform-based open single system ii.) application-based open single system.

One of the key features of the open single system is convenient accessibility to the system and the service. The service orientation of each of the perspectives is addressed based on its properties, functionalities and operations of B2B organization at service level.

As shown in Table 3.1, there are some required SOA principles for every service orientation, which differs in systems or services.

Table 3.1 Service Orientation Requirements at System and Service Levels

Service Orientation	**SOA Principle**	
	System Level	Service Level
Interoperability	Level of system abstraction	Level of service loose coupling, abstraction
Adaptability	Level of system autonomy	Level of service loose coupling and autonomy
Re-usability	Level of system discover-ability and statelessness	Level of service loose coupling, genericity, discover-ability, composability and statelessness

3.3.1 Service orientation for platform-based open single system

For an organization using a platform (e.g. SAP), running several software systems (e.g. Customer Relation Management (CRM)) that generate service A and service B, the two services can be interoperated iff the following quality and SOA principles is conformed with: At service level, the degree of interoperability of two services A and B depends on the following SOA principles.

1. Level of service contract standardization.

   Within an organization using a platform (e.g. SAP), the applications are explicitly required to adopt the specified SAP's standard for application accessibility, system integration, service interconnection, and service management meta-data.

2. Level of abstraction.

   The service contract between the services aid the effective utilization of the service needed giving out the required information to consumer. Service A and service B can expose its logic and functionality from multiple different underlying applications by using another service C. The level of abstraction of Service C is determined to extent by the collective levels of abstraction attained by service A and service B operations.

3. Level of loose coupling.

   A given service A is directly coupled via the information exchange to another service B iff there exists information used in A that is defined in B, change in service A is likely to affect B. Thus, to achieve loose coupling between the services, there must be a level of isolation.

The degree of adaptability and re-usability of two services is depended on the level of the standardized service contract which is easy to determine because they belong in the same platform. With the standard, the level of SOA principles can be specified. For improved interoperable services, well-established standards must be defined for easy communication and ensuring less/no negotiation power between the applications in the platform, creating a standardized degree of abstraction and loosely-coupled, autonomous and generic services.

3.3.2 Service orientation for application-based open single system

In an organization (e.g. NASA), with several bundles of sub-systems that generate different services for unified business goal. For the services to interoperate, the systems have to be adaptable. The level of system adaptability depends on the service autonomy and statelessness of each sub-system in the open single system, before the interoperability of the services can be considered. The degree of adaptability of the services depends on the following SOA principles namely:

1. The level of system adaptability. This is possible by agreed-upon system standard; the specified standard will aid the interoperability of the services of the systems. With established standard, systems are compatible (i.e. increase service interoperability), connected seamlessly, providing an efficient and simplified business process, regulated for the amount of information exposed (i.e. increase service abstraction), clearly documented and accessible (service discoverability) and exposed for services functionality for reuse (service genericity) which can be composed to deliver a new service (service composability).

2. The degree of service interoperability.
   After achieving system adaptability, then the service interoperability can be defined following the specified steps.
   Loose coupling of service in open single system can be expressed, for instance when a department depends less on another department in an organization to execute its business process; it might not be coupling on technical level, but bit more operational level. An organization running open single system defines the level of standardization that they incorporate into their business process, which ensures low impact of individual failure between the systems.
   Interoperable services are by necessity adaptable in open single system, but the con-

verse is not necessarily true. To realize the power of service interoperability through robust data exchange, one must look beyond adaptability.

3.4 Service Orientation Degree of Open Collaborative Enterprise Information System

In fast growing business world, collaboration does not only exist with similar businesses, for instance, software companies (e.g. *Microsoft and Apple*), Universities (e.g. *MIT and Harvard*), but more of unrelated businesses, for instance, the collaboration of *Microsoft and Toyota* for intelligent energy consumption (Turiera and Cros, 2013), *Mercedes-Benz and Facebook* for new frontier social driving (Turiera and Cros, 2013), *Puma, Adidas and Innovalley* for intelligent sportswear and accessories (Turiera and Cros, 2013), *NHS and Facebook* for enormous potential of health's socializing, *Evernote and Moleskine* for information overload management and many more.

Therefore, the two different collaborative perspectives described above (i.e. the related and unrelated open collaborations) cannot be treated the same way.

Each of the perspective is described and the collaborations are achieved based on the level of service orientation which conforms to SOA principles.

Technically, the level of service orientation for open collaboration is different for every case. For instance, organization A and B have to specify the level of collaboration for their systems to interoperate, adapt and reusable.

3.4.1 Service Orientation of related open collaboration

At system level, organization A and B in similar business drive can easily define the level of interoperability and adaptability of their systems based on the agreed upon standard. In doing so, the following quality principles have to conform the following SOA principles with namely:

1. The level of adaptability of the system
   For an organization A and B to adapt, agreed-upon standards have approved by both organizations giving rise to the development of an interface which will be the collaborating environment for the organizations. In defining adaptability, organization A and B specifies the portion of its business processes to collaborate with, which is accessible by both organizations using defined interface. Inside of the interface, the related open collaboration defines the level of service autonomy when its services exercises control logic over their underlying and execution environment or interface which has to be loosely coupled.

2. The level of adaptability of services.
   After the adaptability of the systems, service adaptability is dealt with. Service adaptability in related collaboration or network is far more achievable as long as the levels of standardized service contract, loose coupling, statelessness and autonomy are all defined, creating interoperable services. For instance, MIT and Harvard universities are collaborating to provide free online courses, known as Massively Open Online Courses (MOOCs) (Pappano, 2012). With established standards and interface, the two universities can collaborate using the standardized bridge between the specified services (data and resources). The next step is the interoperability of the services of the two universities. To safe cost and time, instead of creating new suitable services, the adaptability of their existing services is considered. Each organization defines its level of abstraction, coupling and service contract duration with each other as shown in Figure 3.3.

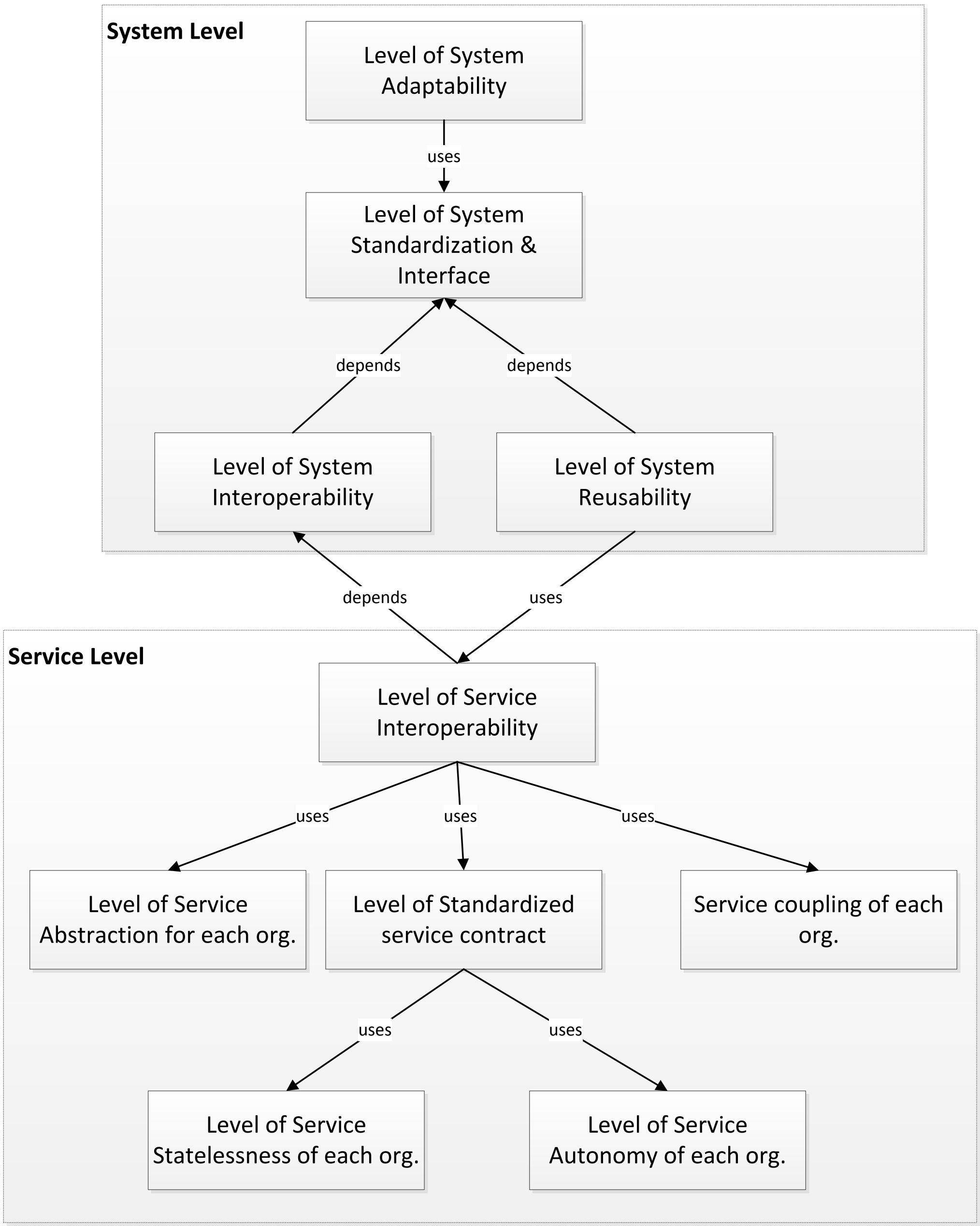


Fig. 3.3 Service Orientation of related open collaboration

3.4.2 Service Orientation of unrelated open collaboration

In recent collaborations, more of unrelated organizations are partnering to improve or establish new services, e.g. health sector and social media (i.e. NHS and Facebook) collaborate to create sociable health care to raise awareness about the need for donations (Turiera and Cros, 2013). This collaboration entails more detailed work which involves the looking at the service orientation of each collaboration processes which conforms to SOA principles at system and service levels. The following quality and SOA principles have to be conformed to namely:

1. The level of system adaptability.
   depends on what they are working on and the d level of relation of organization A and B. An interface is created for collaborating organization A and B by agreed-upon standard which uses on each organization's service autonomy and statelessness.

2. The level of system interoperability.
   With the interface created, the next step is establishing standardized service contract which depends on each organization's level of service abstraction and coupling. The next section discusses the requirements for open single system and open collaboration because of page limitation.

3.5 Service Identification Requirements of Enterprise Information Systems

As the current methods exist, there are over twenty methods for identifying services which are not adaptable to changes and less applicable to every business case. Also, the approaches are too simple to satisfy the common principles that are supported by SOA platforms. In order to realize or create a new method, the following requirements are to be satisfied for

each EISs.
Identification of services in EISs has some requirements leading to modification of the service design. Likewise, identification of service orientation for a specific EIS will improve the service design, and services can be identified as business requirements change.

3.5.1 Service Identification requirement for open single system

Services in open enterprise systems operate in a highly dynamic manner; independently subject to have less boundaries (*depending to the level of standard*) running different business processes or applications. There might be variations due to changes in business goal and environment in one department, making the services to slightly change or loose-coupled. The effect of the slight change in the business goal or environment plays a risk on the right identification of services.
In recent business world, organizations are subject to changes in requirements and goals, therefore, it mean their business services are subject to change as it switches dimension. For example, an organization 'A' running an open systematic structure has different resulting values of the service design description depending on the task parameters. The organization 'A' can become closed systematic structure as a result of a merge with low centrality system. Therefore, to make the services in open enterprise system to have the right service design description for every changing business case, the following requirements have to be satisfied: The service identification requirements are as a result of the reviewing and classification of the open single system. The requirements for identifying services in open single systems are as follows:

1. Analyze the type of services they provide, what means (platform or application-based)

2. ***If*** *it is platform-based*:

(a) The dynamic relationship of the services is defined by analyzing the service orientation of the services identified (level of loose coupling and abstraction) in the enterprise system as it changes round the dimensions. In platform-based single system, standardization is not a major concern for the organization; it conforms to the platforms standard. The only concern is the level of coupling, which involves less abstraction.

Else if *it is application-based*:

(b) The requirement for application-based is more difficult because it is tailor-made to the business specification. Creating a specific level of interoperable services within an organization using different applications, the following condition has to be met namely: Agreed-up standard, the level of service abstraction, autonomy, statelessness, discoverability, composability and genericity, and loose-coupling of the services.

3. Then, dynamic candidate services from the integrated business processes (top-down approach) can be identified depending on the type of system and service orientation.

3.5.2 Service Identification requirement for open collaboration

As discussed in Section 4.4, collaboration is a broad topic to dive into. In this thesis, collaboration has been dealt with in the area of interaction of partnering organizations, people and machines. Therefore, defining the requirements to identify services in open collaborating organizations, The service identification requirements are as a result of the reviewing and classification of the open collaboration in Section 4.4.

1. Analyse the type of services they provide, what means (related or unrelated)

2. ***If*** *it is related collaboration*:

(a) Standardization has to be agreed upon for services to interoperate or adapt on a defined interface.

(b) Each organization defines its level of abstraction, coupling and service contract duration), as it may changes.

Else if *it is unrelated collaboration:*

(c) Standardization for unrelated collaboration is more difficult as it needs more consideration on the service orientation. The standard has to be highly agreed-upon, creating a connecting medium for collaboration.

(d) Each organization have to well-define the standard for collaboration, high level of abstraction, coupling and service contract duration), as it may changes.

(e) For instance, 'Toyota' and 'Microsoft' can collaborate in one business aspect of their business process or they jointly create new business processes in the standardized interface.

3. It is required that separate service can keep track of collaborative services' transactions or sessions in the collaborating organization for monitoring accuracy, appropriateness, time behaviour, co-existence, user error, authenticity (service statelessness).

4. Therefore, collaborating tasks and entities are defined from the interface. Analyse the level of service orientation of collaborating organizations and its entities..

5. Then, the dynamic services can be identified from the integrated business processes model using the quality and service orientation principles as shown in figure 3.

It is important for organizations to define their level of service orientation for inter, intra collaboration. With Figure 3.4, service depends on the operational requirement enforce by the given standard body (e.g. ISO 9001, 25010). Furthermore, the services are bound by dimension which is a set of service orientation.

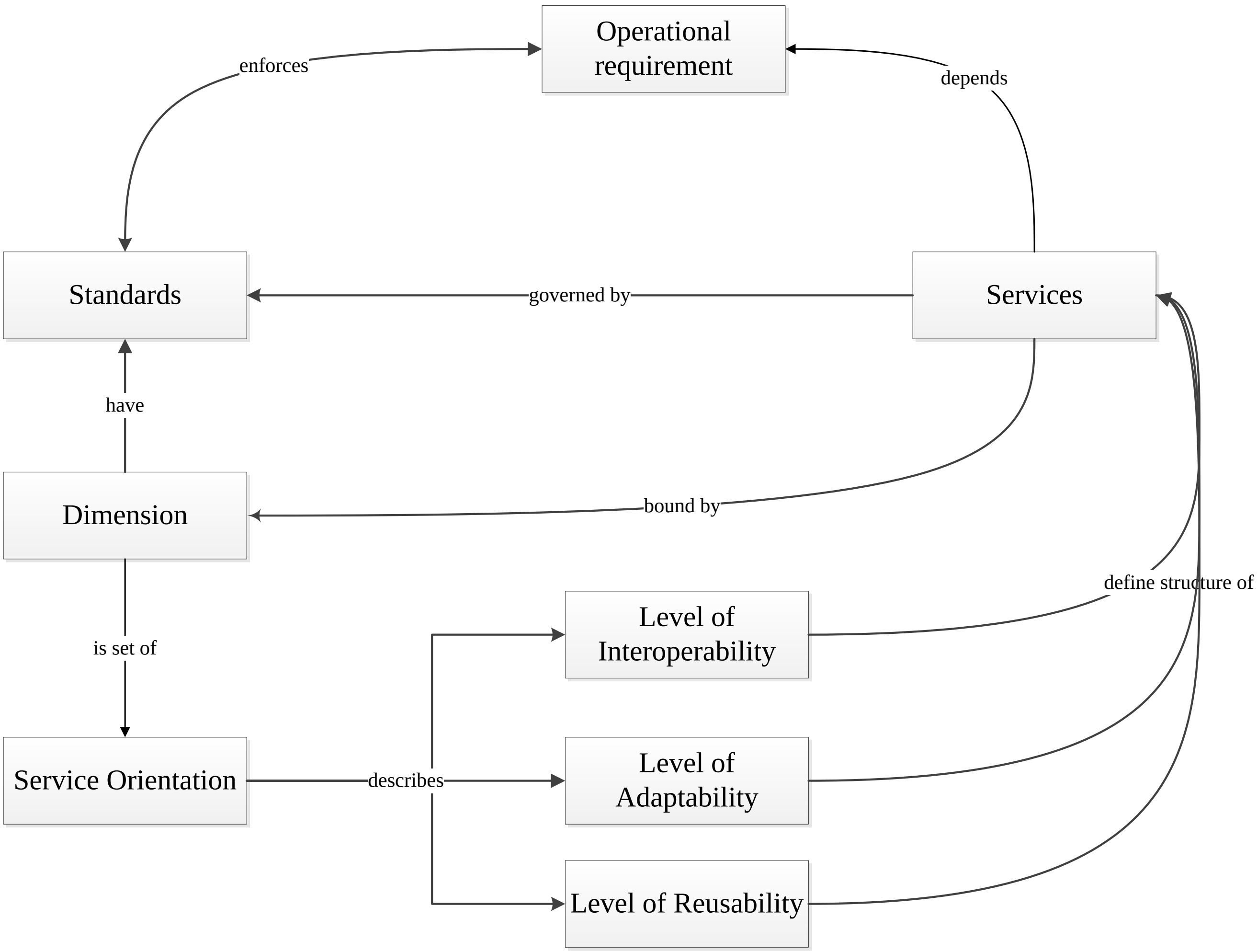


Fig. 3.4 Service Identification Requirement Map

3.6 Summary and Conclusion

In this chapter, we progressively described the requirements for service identification for EIS by looking into different organization systems. With the continuous changing factors like economy, market competition,security and location, enterprises are subject to changes or improvement in services or systems. Therefore, organization can remotely understand their domain (i.e. dimension) and adapt to change regardless the situation or cause.

This chapter also address organizational agility which is achievable when loosely coupled services are identified by knowing the level of service orientation for the enterprise system. The next chapter involves the role of organizational data in modelling, representation and business service identification.

Chapter 4

Data-driven Service Identification Framework

4.1 Introduction

This chapter presents the step-by-step approach of identifying business services (see Figure 4.1), which guide developers in creating functional service design. This chapter presents on the pre-analysis and modelling the data elements while the post-analysis of the approach presented in chapter 5.

One of the most important aspects in constructing service-oriented architectures (SOA) is the identification of services at the right level of abstraction (*see Chapter 3*) provide the optimum re-use of services for today's enterprise information systems.

Since the inception of service identification in both software and business management, there has not been was primarily focused data. As shown in Section 2.8, majority of methods are model or goal-driven.

The use of model or goal driven methods are is less adaptable to changes and reusable Sofela

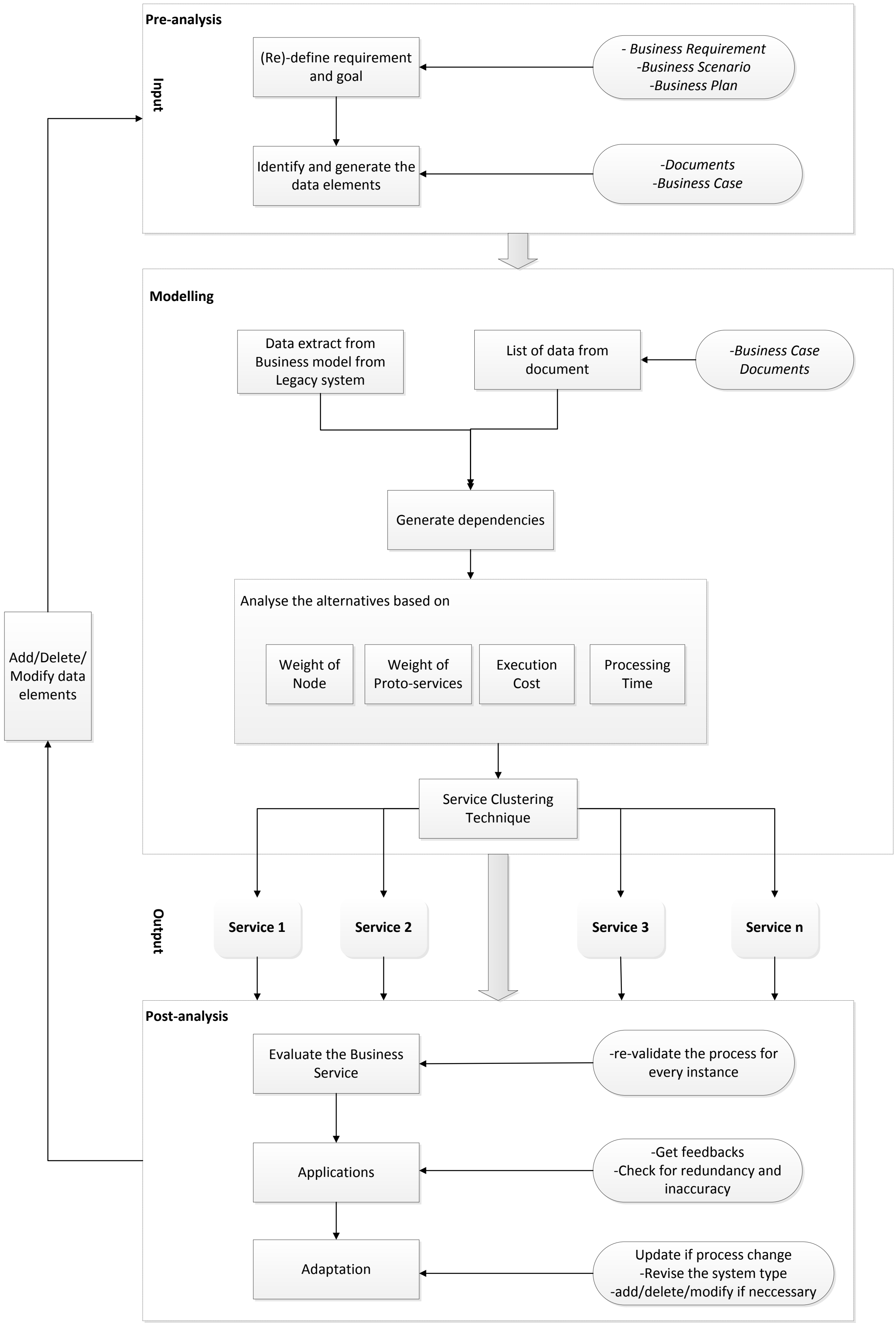


Fig. 4.1 Process Flowchat for Business Service Identification

et al. (2016) than data focused methods i.e whenever organization requirements change[1], the data also change. Adapting to change in model and goal-driven methods involves a lot of task and time i.e re-modelling. Moreover these model and goal driven methods have difficulties for use in migrating large non service-oriented systems to service-oriented systems because a goal-driven approach always need human interpretations Sofela et al. (2016).

This chapter introduces a data focused service identification method. Inspired by the work of Vanderfeesten et.al.(2008), this proposed service identification method provides a method for deciding service granularity that is repeatable, objective and effective. Based on flowing data within a system, we first identify proto-services (i.e. initial service elements) from the dependencies among data. Using proto-services as input and other constraints, a family of service clustering algorithms is designed to identify the final services based on different criteria. It is worth to note that proto-services do not need to be implemented, and in the real world would be too small to be feasible (any system would be overwhelmed by communication costs). The skeleton of our method is as follow:

1. List of flowing data within a system (e.g. Table 4.1)
2. Analyse dependences of data (Section 4.2) and identify proto-services (Section 4.3)
3. Proto-services and other constraints (such as service invocation costs, service invocation fluency, service locations, etc.) (Table 4.2) are used as input of the service clustering algorithms (Section 4.3),
    - Target amount of services, T.
    - Adjust constraints for different situations and service identification results
4. Calculating coupling and cohesion of different service aggregations (Chapter 5)
    - Re-adjust constraints

[1] For example, the foreign trade policies change frequently over the years due to political, economy interest and difference, creating new policies and organizational requirement which result to new model.

- Re-invoke service clustering algorithms
- Re-calculate related coupling and cohesion of service aggregations

The first 3 steps can already independently identify services. Step 4 provides further insights to evaluate the service identification design using cohesion and computing metrics. Step 3 is repeatable for different purposes e.g. certain services may need to be in different locations with change of invocation costs. Step 4 could return to Step 3 for certain justification of the design and as such improve the overall quality of the service portfolio.
To demonstrate this approach, we use a case (see Appendix I) modified from the sofa production model case presented in Bianchini et al. (2014) as illustrating example. Table 4.1 shows the relevant data in the case.

$$\boxed{dataElements = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}}$$

The structure of this chapter is as follows. Section 4.2 presents a detailed approach for deriving and modelling data dependency using the Sofa production and ordering process as the running example. Section 4.3 presents the analyses and identifies the data dependence and proto-services, and creates the algorithm for clustering proto-services and other constraints (such as service invocation costs, service invocation fluency, service locations, etc.). Section 4.4 describes how the algorithm works. Section 4.5 presents aggregation for the clustered services.

4.2 Data Dependency

A data element can have combination of data types such that it is not subdivided within the model. If it is sub-divided, the components are elements and the combination is a tuple.

Table 4.1 Data elements of the sofa manufacturing and ordering process.

	Data Element
1	Order
2	OrderEval
3	LoC
4	Bill
5	BackCompSpec
6	RawMaterials
7	BackComp
8	DeliveryNote
9	AccessCompSpec
10	Material
11	Accessory
12	DeliveryAccessory
13	Sofa
14	Sofa Package
15	Invoice

It crucial to know that there are several types of data which interact based on specified attributes between the activities. Each of these data types plays key role in modelling of the data dependency graph. Each business instance describes the types of data in a process distinguished by distinct activity.

Definition 4.1: R_i: Root data element.

It is the single resulting data element that is not used as input for any other operation. It requires conditional data input, reference, operational and decision data to create the resulting data. The reference, decision and operational data elements are subset of root data.

$$\{r^{n}{}_{i} \wedge D^{n}{}_{i}\} \subset R^{n}{}_{i}$$

The root data elements are also known as the trigger data elements that initiate another process in an intra or inter-organization.

For example, when order is received from a client, if the order is complete (OrderEval "2" $\stackrel{f}{\Rightarrow}$ order "1"), it triggers the generation of list of component (LoC) in the purchasing Office, leading to the price estimation. The Bill triggers the internal manufacturing department to analyse the backbone component and also triggers the ordering of the accessory from another accessories providers. The assembling of sofa is triggered when BackComponent and Accessory are both delivered to the purchasing department. The completion of the assembling of the sofa triggers the packaging, deliver of sofa and sending of invoice by the sales office. Therefore (*"1, 2, 3")* are the root data element as shown in Figure 4.2.

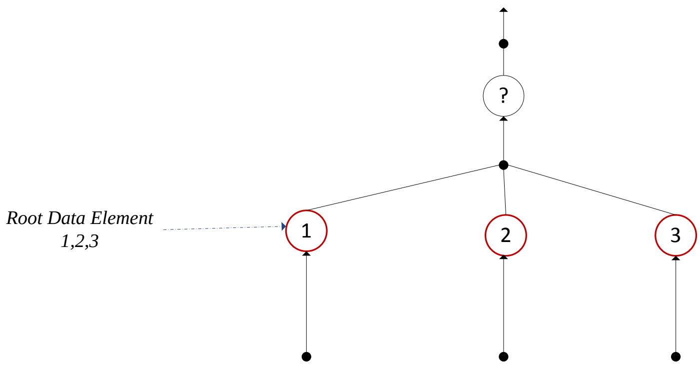


Fig. 4.2 Root data element from Sofa manufacturing example.

Definition 4.2: D_i: Decision data element.

It is the data element that can be produced by multiple alternative operational data. The decision data is a subset of operational data.

$$\{r^{n}{}_{i}\} \subset D^{n}{}_{i}$$

For example:: In figure 5.2, in the manufacturing department, the BackComponent and DeliveryNote are created either when the BackComponent is readily available for collection or the company produces the BackComponent when it is not available. Likewise, in the production and delivery of the accessories by the external provider. Therefore Proto-service Ps03 derives from two optional decision data elements (6 and 10) or (7, 9, 11) as shown figure 4.3.

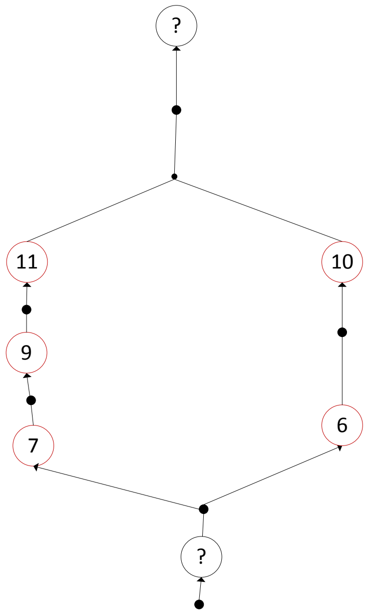


Fig. 4.3 Decision data element from Sofa manufacturing example.

Definition 4.3: r_i: Reference data element.

The data that initiate the whole process is known as the reference data. It is the input data element to an operation without input elements.

$$r^n{}_i \in O^n{}_i$$

For example: The data element “1" order activates orderEval, and other data elements in other departments. Therefore, order “1" as shown in Figure 4.4 of the case).

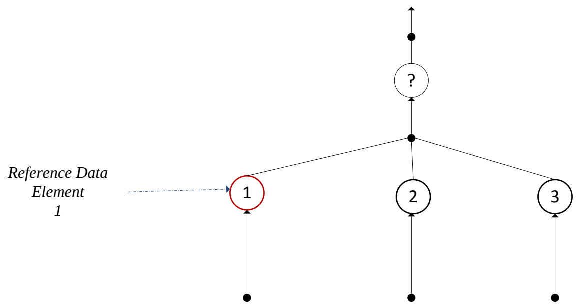


Fig. 4.4 Reference data element from Sofa manufacturing example.

4.2.1 Data Dependency graph

First, we deriving the data elements and its dependencies which is graphically represented in the data dependency graph as shown in Figure 4.5.

The data dependency graph consists of the data elements depicted as circles, the connecting arrows and the proto-services depicted as black thick dot. The data elements are the abstract representation of the data that is processed in the business process.

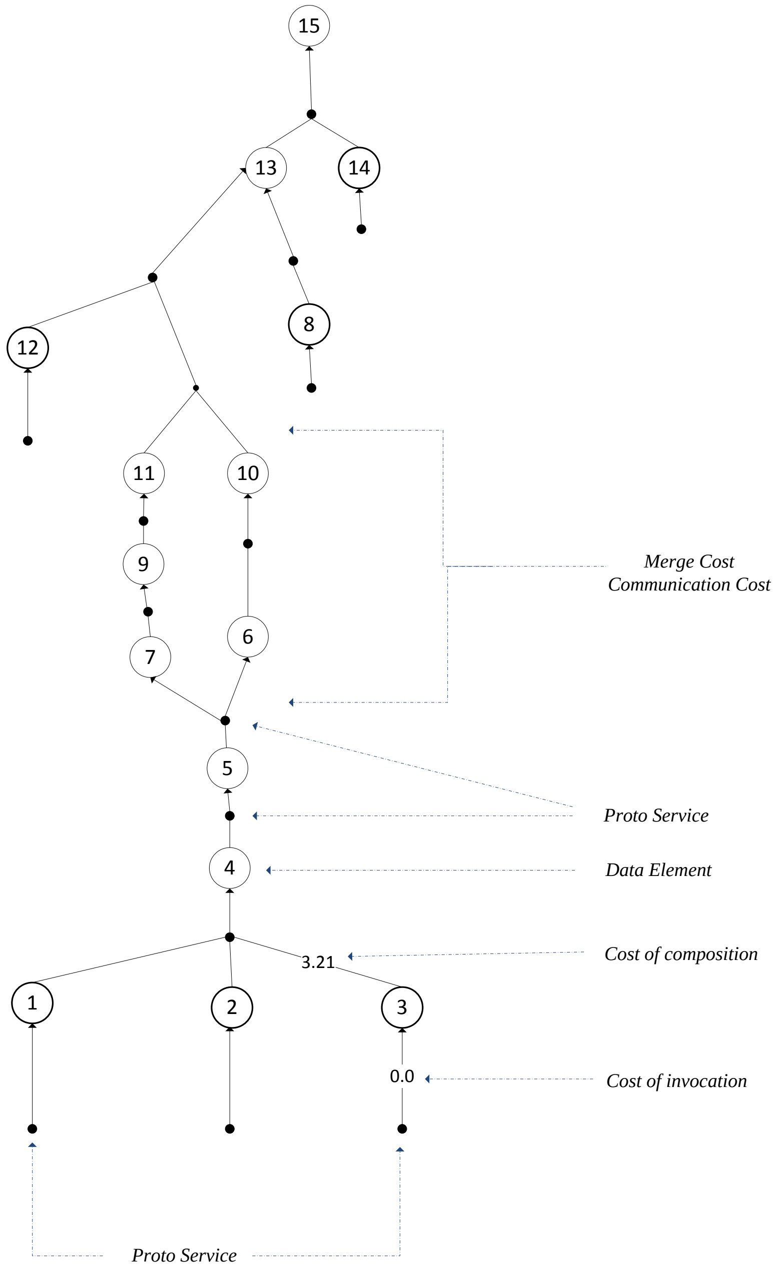


Fig. 4.5 Data dependency graph.

Proto-Service

The proto-services are the action(s) that are taken on the data elements values. The connecting arrows displays the properties of interaction between data elements.

In general, a proto-services can have a number of attributes associated to it that describe its characteristics in more details.

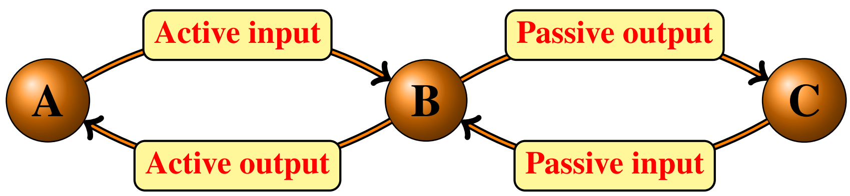
From the perspective of proto-service B, in proto-service parts A, B, C as shown in Figure below, involving different data elements namely:

1. Input (active) data element – result from calling another service/function (A) from service/function (B)

2. Input (passive)data element – result from calling another service/function (B) from service/function (A)

3. Output (request) data element– sending data from service/function (B) to other service/function (C)

4. Output (response) data element – The result of the service/function to the caller



4.3 Proto-Service Clustering Algorithm

This section gives a formal description of the approach for identification of right size of service at right level of abstraction which offers optimum functionalities without interfering with the pre-defined service design concepts. To ensure proper identification and generation of right candidate services, we propose to cluster and estimate

- the corresponding non-functional attributes of the proto-services during initial and execution stages, alternatively distributed round five targets (t= 0,1,2,3,4).
- the coherence of the whole process

A proto-service part can be invoked by multiple services. All parts of the calling service are combined in this sense. Given m services there are n=m+1 possibilities, either the part is an element of one of the callers or it is a service itself. The choice is that choice with minimal cost. This however requires a cost of coupling (at least if distance is not a factor), otherwise it can never be independent unless there is a locality restriction.
For every flow of data within services, there are cost of data transfer which can be measured. An invocation can involve any number of data elements (parameters) -> tuple. A service has a “probability” to invoke another service/function with a certain tuple (this can be more than 1 if it invokes the service multiple times). Every function invocation can result in multiple different active inputs (call results) each with an occurrence frequency (the total of the tuples should be the same as the frequency of the call tuple - this is per call, not per service).
Optionally a service can have an internal cost [2] (*that is the time spent inside the service*). For a given service cost of calling a service with a given tuple as input is: the frequency of that operation $*$ cost of sending that tuple + internal cost + sum(for each output)[frequency $*$ cost of sending the tuple] Cost of sending are the additional costs over regular data transfer and is normally a function of size and distance Some data elements have fixed locality (therefore

[2]It makes no difference in the calculation, but it’s probably good to show that it does not

cannot be inclined, but must be transferred) – eg. Google search index. The algorithm involves the use of three attributes namely the cost of invocation, composition and merging the proto-services.

4.3.1 The cost of invocation proto-services

The cost that associated with the invocating the proto-service from one state to another (given by its parameters). For instance, there is a given cost of invoking proto-service which leads to the data creation (Order received (1), order evaluation (2) and list of component (3) in the running example of Table 4.1).

4.3.2 The cost of composition of proto-services

The cost associated with the composing a proto-service resulting from the combination of data. As shown in Figure 4.5, there is a cost of composing the pro-service that used the order received (1), order evaluation (2) and list of component (3), resulting to the creating of the bill (4).

Definition 4.4: (Composition Cost.)

Let a data model be given with set of fixed nodes (N) and set of proto-service (A). The cost of composing the services (given by probability and parameters distribution).

Definition 4.5: Weight of the Node.

Let a data-based model, D_m consisting of set of nodes $\{n_1, n_2, ... \in N\}$ and set of proto-service $\{a_1, a_2..... \in A\}$. A node can be of one of three types namely, root R_i, decision D_i and reference r_i data elements. The set of nodes ($n_1, n_2, ... \in D_m$) are inter-connected by the proto-service $\{a_1, a_2..... \in A\}$.

Thus,

$$\boxed{w(n) = \{N \cup D_m\}}$$

The weight of a node N, w(n) is defined in (Vanderfeesten et al., 2011) as follows:

$$w(n) = \begin{cases} 1, if n \in \{C\} - typeAND \\ \frac{1}{d}, if n \in \{C\} - typeXOR \\ \frac{1}{2^{\mathrm{d}}-1} + \frac{2^{\mathrm{d}}-2}{2^{\mathrm{d}}-1} \cdot \quad \frac{1}{d}, if n \in \{C\} - typeOR \\ 1, if n \in \{D_{\mathrm{m}}\} \end{cases}$$

where d: the number of all input and output in a node or proto-service

4.3.3 The cost of merging proto-services

The cost of merging Node 4 and 1 is: CompletionCost(1+4- completionCost(1) + completionCost(4,6), (4, 5), (4, 3) (4,2)

For every merge m-1, skipped nodes

$$MaxSkipping = \frac{(N)(N-1)}{2} \tag{4.1}$$

$$EC(\vec{N}) = \sum_{n_1, n_2 \in \vec{N}, \forall a \in \vec{N}} \frac{CompletionCost(n_1, n_2)}{\frac{N \cdot (N-1)}{2}} \tag{4.2}$$

Algorithm 1: Create data list,elements and proto-services

```
Input  : A set J of business process fragments (f_1, f_2, f_3, ... f_j, ... f_n ∈ F_m)
Output : A set w of business services (s_1, s_2, s_3, ... s_j ∈ S_m)

foreach f_i ∈ J do
    // Data Element types
    T_1 ← T_2 ← T_3 ← T_4 ← False;
    if f_i ∈ R_d then
        a ← protoService({f_i});
        if a is compatible with f_i then                        // O(left, f_i) = 1
            if a < f_i then
                s_i ← Union(a, f_i)
            else
                s_i ← Union(f_i, a)
    else if f_i ∈ L_d then
        a ← protoService({f_x | i ≤ x < j});
        if a is compatible with f_i then                        // O(a, f_i) = 1
            if a = f_i then
                s_i ← Union(a, f_i);
                /* this is put with up to keep the tree as flat as
                   possible                                                  */
            else
                s_i ← Union(f_i, a);
                /* this is linked to up                                      */
    else if f_i ∈ C_d then
        a ← protoService({f_x | j ≤ x ≤ n});
        if a is compatible with f_i then                        // O(a, f_i) = 1
            if a < f_i then
                s_i ← Union(a, f_i);
                /* this is put under up to keep the tree as flat as
                   possible                                                  */
            else
                s_i ← Union(f_i, a);
                /* this is linked to up                                      */
    else if f_i ∈ Q_d then
        a ← protoService({f_x | i ≤ x < j});
        if a is compatible with f_i then                        // O(a, f_i) = 1
            if a < f_i then
                s_i ← Union(a, f_i);
                /* this is put to the right of up to keep the tree as
                   flat as possible                                          */
            else
                s_i ← Union(f_i, a);
                /* this is linked to up                                      */
return w
```

1. *Set the value exchange $\tilde{w}$ between b_o and Proto-service $A_{1..n}$*
2. **For** each value exchange $\tilde{w}$
3. *set the boundary of s as a cluster*
4. *create a new service s_i and add s to candidate service S*
5. **foreach** s ∈ S
6. check for the lowest composition cost
7. Input the new service S into service portfolio
8. **End**

4.4 Service Identification Algorithm

A graph of all “*parts*” where each part starts an individual service is generated as shown in the figure 4.6. Some parts have locality restrictions,and there will be external services that cannot be influenced: for example, representing users calling a web service, in some cases they need to be represented in aggregate. Between each service there can be an arbitrary amount of connecting arcs. There are some “unassigned” localities representing servers that may host services.

There is a function with given two localities and a tuple which calculates the cost of transferring that tuple. It is important to note that locality is a flexible notion and could be a separate container on the same physical server with a low cost.

Each external source has its frequency which is a relative amount compared to the other sources. Starting with these frequencies, we determine how frequent each “service” is invoked (i.e *#caller* $*$ *sum(invoke frequencies for service)*.

All services are allocated to a random initial “host”, starting from the most call-costly service. we also determine the host on which the overall system costs are minimal (*this only depends on the cost of the direct callers*). Services with the lowest accumulated call costs with internal services as callers (*the total of costs of calling and returning from the service for all callers*) are candidates to incorporate a service.

Based on the total call frequencies, we calculate the service with the highest overall call cost (*the service whose cumulative invocation is most expensive*) and calculate for all callers the new costs when incorporating that service as well as the cost for retaining the service at each of the available hosts. Select the minimum service (if it was not consumed) that are eligible for the next round. An evaluation that does not result in a change does not count as a round. When a round does not finish because no changes are made, the system is put on halt. One of the costs will be a cost of complexity, a polynomial of size `(x^2)`.

The step-by-step execution of the operations that may occur during the execution of the sofa production process is shown in Figure 4.5, the introduction of the initial state with the leaf data elements which are executed depending on the flowing attributes (execution cost, processing time, execution conditions, failure probability and resource class).

The values for some of the root data elements (1 *;2; 3; 8; 12; 14*) are available
(indicated by bold red circles). We refer to this situation as the *initial state*.
Executable operations by steps: Ps08; Ps07; Ps*03; Ps12 and P*s01;.
a. The value for data element 4; 5 is produced by Proto service Ps08 as shown in bold blue circles.
b. The value for data element 7; 6 is produced by Proto-service *P*s07 as shown in bold yellow circles. The data value for 9; 11 is dependent on the proto-service Ps13. The value of data elements 10 is dependent on the proto-service Ps13.
c. The value for data element 13 is produced by proto-service Ps03 and Ps12 shown in bold purple circle.
d. The final data element 15 is produced by proto-service Ps01 shown in bold black circle.

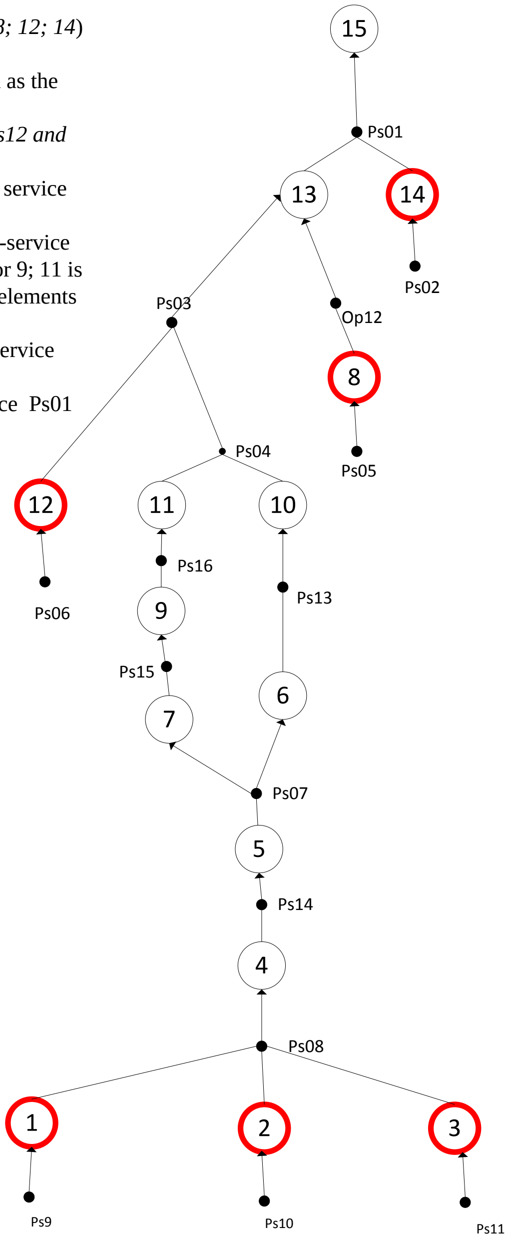


Fig. 4.6 The data dependency graph for the sofa production case.

Step 1: Assign frequencies to all nodes

Step 2: Calculate $c_{x,y}$ for all edges

Step 3: Determine the $max_{c_{x,y}}$ for every incoming node y

Step 4: Create a sorted list of nodes bsed on $max_{c_{?,y}}$ for node y

Step 5: Pick the last node Y from the sorted list with the highest cost

Step 6: forall called x of node y, calculate the difference in costs if merging y into x

Step 7: Choose the node x_{min} with the highest savings

Step 8: if x_{min} i worse than not merging, pick the previous element from the sorted list and go to Step 6. If no more nodes then finish

Step 9: Merge node y into node x to become X‘

-recalculate all c for all nodes

-calculate $max_{c_{?,x}}$ for X‘

- Remove x and y from the sorted list

- add the new merged node X‘ to the list

f_x = frequency of activity invocation for activity x

$r_{x,y}$ = ratio of single invocation of x leading to an invocation of activity Y

$c_{x,y}$ = the cost of invoking x,y (input + output)

$d_{x,y}$ = the distance between x,y. Note:d =1. When merged, d=0.

$s_{x,y}$ = size of data between x,y i_y = cost of all incoming calls of X

Target T = (0, 1, 2,...n)

$$c_{x,y} = f_x * r_{x,y} * s_{x,y} * d_{x,y}$$

$$i_{max} = \max_{x,y \in N} \frac{CompletionCost(c_{x,y})}{\frac{f_x \cdot (f_x - 1)}{2}} \tag{4.3}$$

It is important that more than one proto-service may be executable at a certain point in time or alternative operation can be executed instead of waiting. For instance, Ps16 can be chosen over Ps13 depending on the execution time and cost. From Table 4.2, the execution target for Ps16 and Ps13 is 1.0, whereas the cost of executing Ps13 is lower than than Ps16 (4.785: 7.564) using equation 4.3. Therefore, it is better of to choose Ps13 over Ps16.

Table 4.2 Operations and their attributes for the sofa production running example.

	Output	Input	Cost	T
Ps01	15	13, 14	6.6275	5.0
Ps02	14	-	1.0	5.0
Ps03	13	12, Ps04	10.6435	1.0
Ps04	Op3	11,10	6.1275	2
Ps05	8	-	1.0	0.0
Ps06	12	-	1.0	0.0
Ps07	7, 6	5	6.23	1.0
Ps08	4	1, 2, 3	3.21	5.0
Ps09	1	-	0.0	0.0
Ps10	2	-	0.0	0.0
Ps11	3	-	0.0	3.0
Ps12	13	8	2.215	1.0
Ps13	10	6	4.785	1.0
Ps14	5	4	0.0	1.0
Ps15	9	7	5.303	1.0
Ps16	11	9	7.564	1.0

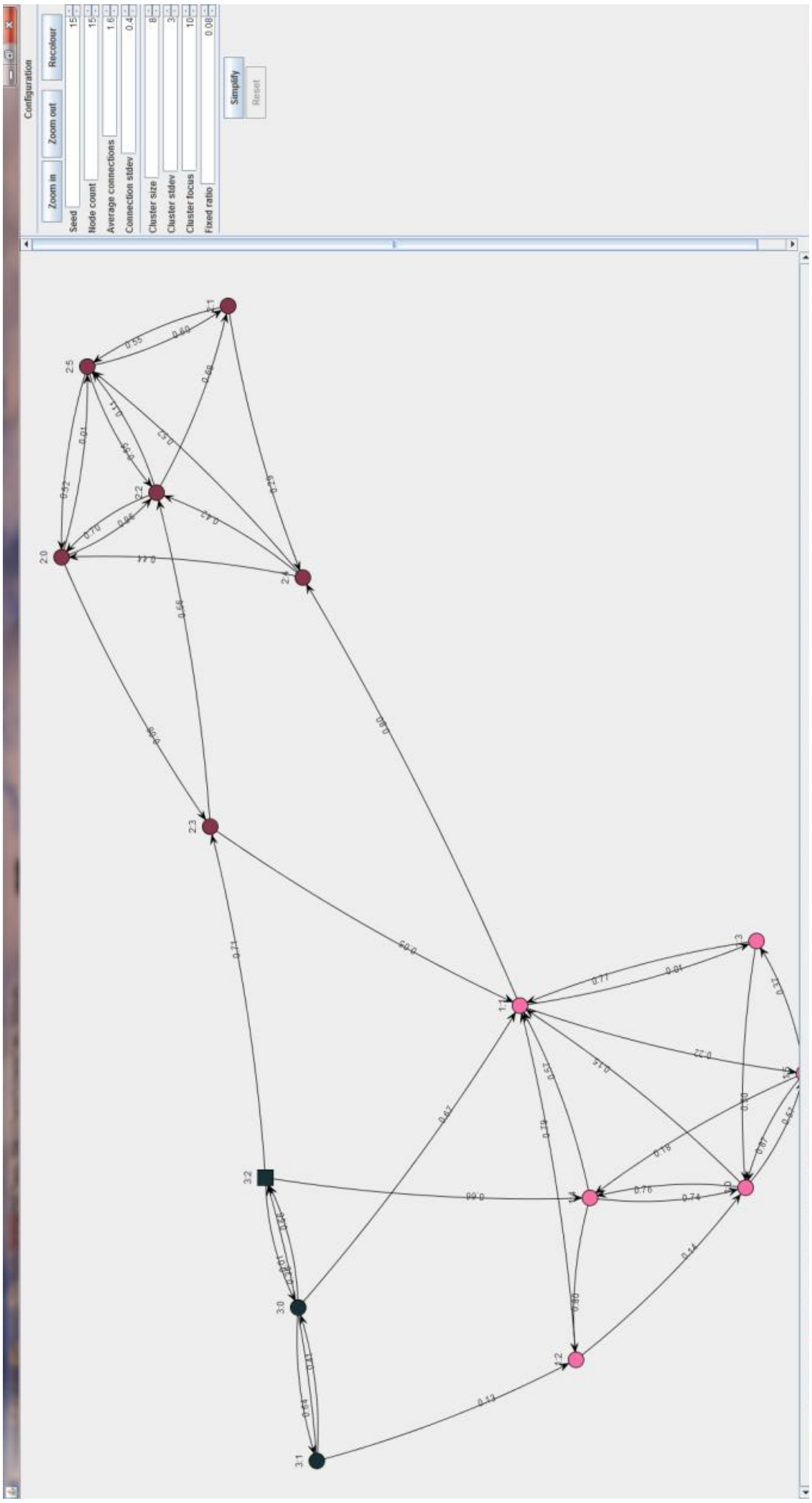


Fig. 4.7 Running example of Service Identification

4.5 Service Aggregation

Services are known not to be generic, services in a specific process model can be different in different instance,therefore additional properties are raised for right identification of business services.

1. a service is triggered and derived by a specific reason which can be said to be a data.

2. in collaborative systems, high service interoperability can help in service identification.

For any alternative (*1, 2, 3....n*), we calculate the cost of connection, the results are compared to aggregate the services based on pre-defined conditions namely:

1. the cohesion value of each split in each alternatives.

2. if cohesion coefficient of alternative 1 is higher than the other, then the partition of alternative 1 is preferable.

3. if the cohesion coefficient of partition of alternative 1 is higher than other, then the larger activities of alternative 1 is preferable

4. in all cases, the heuristic is indecisive.

For specific instance, the service size is defined from the alternatives, which in another instance can be of undefined-sized and logically decoupled which can be invoked by an action within its internal or external process. Therefore, it cannot be measured in terms of absolute digits or formulae but can be analysed in terms of cohesion and coupling criteria to improved service structure, and lead to high interoperability.

Chapter 5

Evaluation of Data-driven Service Identification Framework. Using the Sofa Ordering Process Case Study

5.1 Introduction

The previous three chapters have extensively dealt with each body of the research methodology (Section 1.3). It is neccessary to instantiate, validate and verify the entire framework using the ordering process as Case study. The reason for using the ordering process is because it is subject to more drastic changes in trade policy due to political factor (e.g. Brexit[1])(Sapir et al., 2016), legal and economic factors. Likewise, in ordering process, several type of enterprise information systems are being used e.g. ERP, CRM and many more. The ordering process demonstrates the link between internal single systems, collaborating with other organization systems as shown in Appendix I. Thus, it is necessary to re-use the same case to verify our requirement and approaches in Chapter 3 and 4 respectively.

This chapter is structured as follows: Section 6.2 introduces the general description of the

[1]Exit of United Kingdom from European Union

case study used in (Bianchini et al., 2014) which is used to analyse the framework. Section 6.3 presents the service identification for the sofa ordering process with the decomposition process and the formalization of the alternatives. Section 6.5 compares our approach with the Bianchini et al. (2014) work. Section 6.6 presents the impacts of coupling and cohesion in service-oriented business process.

5.2 Case Study: Description in General

As a motivating example, we present a case study "*sofa production and ordering process*" reproduced from the recent publication (Bianchini et al., 2014). The sofa production process depicts the processes in single and collaboration system(*as discussed in chapter 4*).

The sofa production and ordering process (Figure 4.1) describes the production of the sofa components and outsourcing accessory from another suppliers. The sofa production process involves the assembling the backbone component and the accessories. The backbone component is produced within the single system of the sales and purchasing department known as the administrative unit whenever an order is received from a client.

The accessories are outsourced from a supplier which involves receiving and analysing order, and delivery of the accessories to the sofa manufacturer. A complete representation of the data input and output of each activity in Figure 4.1 is summarized in the Appendix II.

In any defining business process model BP_m (e.g. "*sofa production and ordering process*"), there are three components that are relevant in business process modelling namely: the business activity *BA*, interaction of activity called action, *A* and the gate, *R*. Aside from the basic definitions and representations of business activity in Chapter 2, business activity is also formally defined.

Definition 5.1: (Business Activity, BA.)

It is a set of triple, the name of the business activity, *n*, the set of its defined attributes *A*,and its relations between the business activities *R*.

$$BA = \{n, A, R\}$$

Furthermore, there are other abstract components in the business process model namely the data objects, rules and messages. The interaction of set of activities $BA_{i \geq 1}$ connected by gates *G* are dependent on set of attributes, the input IN_d and output OUT_d data and the relation between the set of business activity $BA_{i \geq 1}$, distinguished by action *A* and the gate *R*.

Therefore, Business activity BA can be said to be 3-tuple, the relations between the business activities *R*, data objects interacting based on its properties defined, (IN_d and output OUT_d).

$$BA_{i \geq 1} = \{n, R_{i \geq 1} \Cup IN_{d \geq 1}, OUT_{d \geq 1}\}$$

Each data abstracts the behaviour that an actor displays in the context of an activity. For example, the first activity in sofa ordering process (*see Figure 4.1*), the sofa manufacturer receives an “*order*" which lead to the “orderEval" which can be said to be data input and data output respectively. The set of data in an activity “*Check Order*" defines its abstract behaviour which triggers more activities and actors.

5.2.1 Decomposition of Sofa Manufacturing Process

Table 5.1 shows the description of the sofa situation which is formalized to determine the steps of operations (Op) in each package of activities (P_{A+1}). This helps to monitor and address any issue within the process and thus this increases re-usability.

For instance P_1, P_2, P_6 partitions are easily executed while as P_4, P_{13} and P_{17} are clustered within its activity, to output k (accessory), it requires the input of j (material) and e (accessory component specification) but e is a prerequisite in producing j, which means for every sofa production, all the materials are used and no material is left for re-use. In a situation where there is high demand for a specific sofa, there should be ready-made material which will reduce process time. P_4, a (order) has been a major requirement in getting d,c,e and f which is

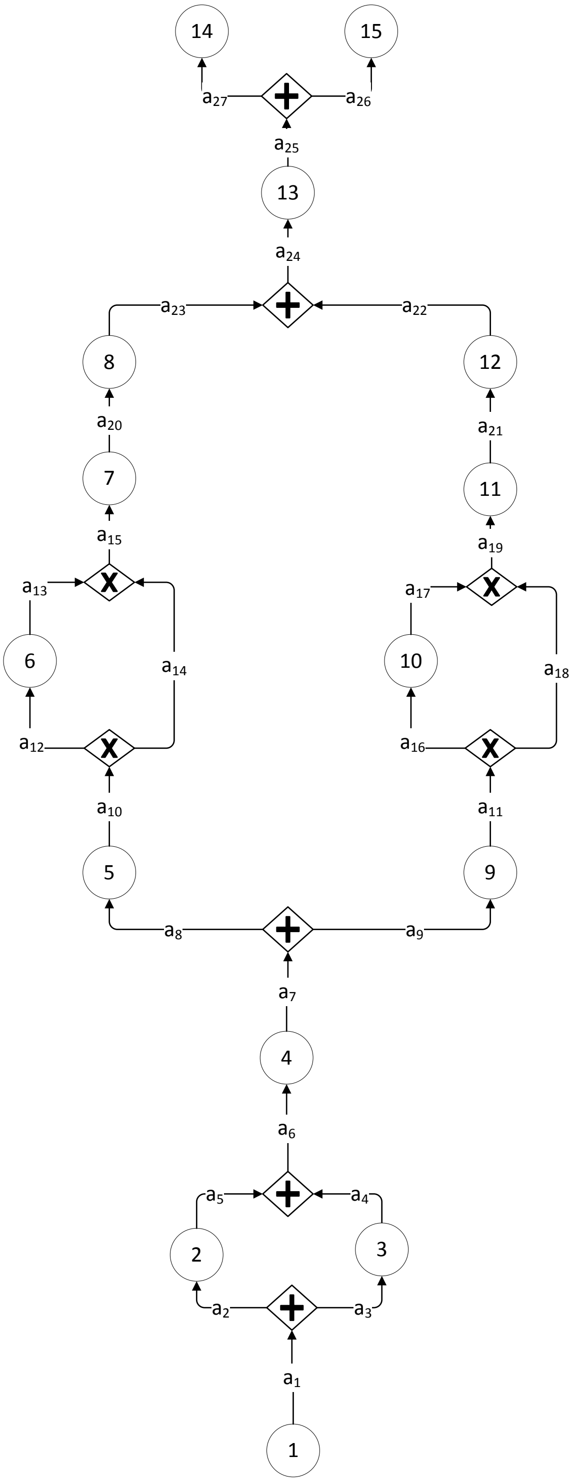


Fig. 5.1 Data element of the sofa production process

clustered in its activity. The production of h (Backbone component) in P_6 is the combination of g (rawMaterial) and d (backbone component specification) but the g can only produced if there exist d, which should makes the process incomplete. The situation of P_6 and P_4, P_{13} will hold the complete of (P_{17}, P_{20}). The solution to easing up the need for d and e is splitting activity 4 to become (4a, 4b, 4c), where 4a is equal to 2.

Table 5.1 Formalization of the partitioned data elements

P_A	P_B	P_C	P_D	P_E
(“Op1”, 1[2], (0,0))				
(“Op2”, 3[2], (0,0))				
(“Op3”, 6[4,3,5], (0,0))				
	(“Op4”, 8[7,4], (0,0)),			
	(“Op5”, 9[8], (0,0))			
		(“Op6”,11[10,5], (0,0))		
			(“Op7”, 11[10,5], (0,0))	
			(“Op8”, 12[9,11], (0,0))	
				(“Op9”, 13[2], (0,0))
				(“Op10”, 9[13], (0,0))
				(“Op11”, 14[6,9,13,2], (0,0))

Alternative designs 1: Using the criteria above to generate the key performing activities in the sofa production. The first design has the following activities namely: Evaluate the order "A", estimate bill "B", delivery backbone component "C", deliver accessories component"D", assemble the sofa "E", deliver sofa "F", send invoice "G" as shown in Figure 5.2.

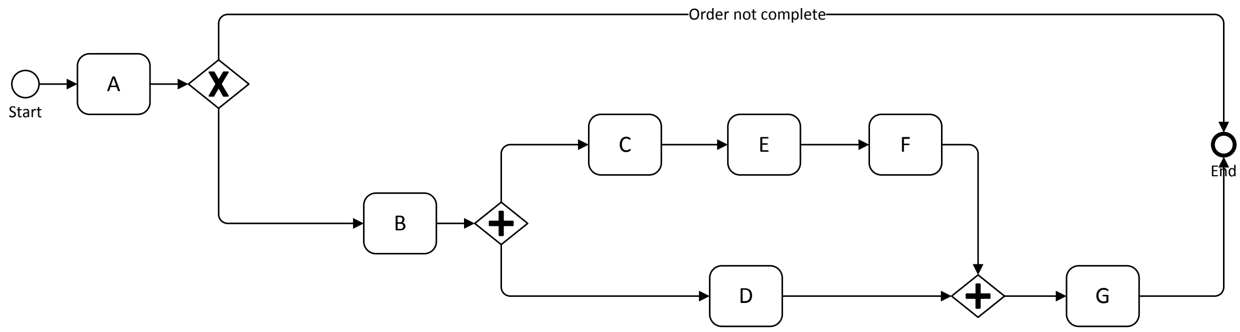


Fig. 5.2 Process model for the first alternative

From figure 5.2, it can be seen that values of activity A is clustered in the operation of data element "1 and 2", the value of activity B is clustered in the operation of data element "1, 2, 3 and 4", the value of activity C is clustered in operation of data element . The description of the preliminary design in figure 5.3 can be formalized in a data model as follows:

Table 5.2 Data elements types in Figure 4.4.

Data Element	Fig 5.4
R_i	(14 and 15), (11 and 12),(7 and 8),4, 2
O_i	$\{(2,3) \in 4\}$, $\{(5,6) \in (7 \text{ and } 8)\}$
r_i	1, 4

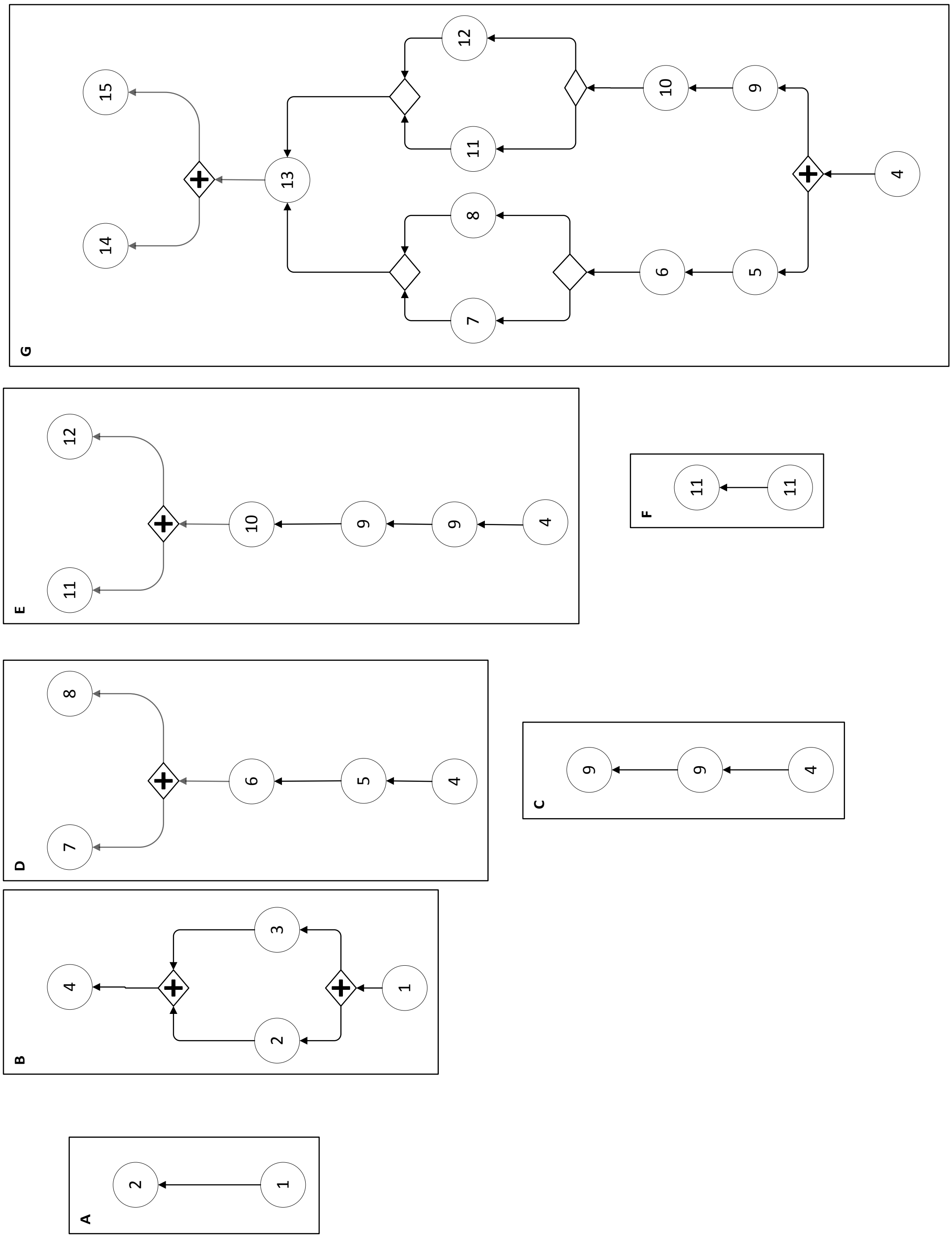


Fig. 5.3 First Alternative of the Sofa ordering Process

Alternative design 2: Using the criteria above, generate the key performing activities in the sofa production which produces same process model as Alternative 1 (cf. Fig 5.3). This alternative is generate when both backbone component and accessory are readily available.

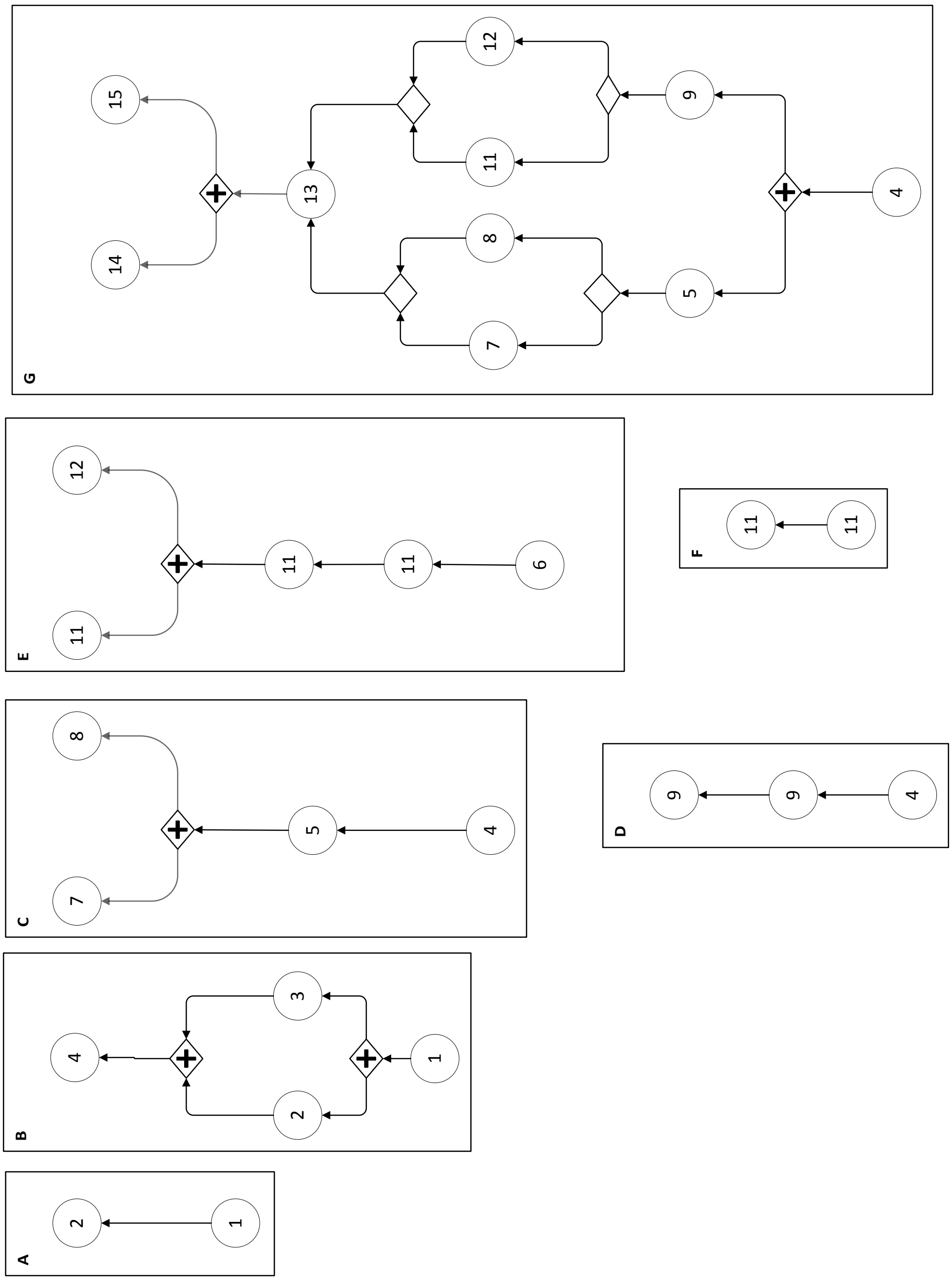


Fig. 5.4 Second Alternative of the Sofa Ordering Process

Alternative design 3: This alternative involves combining the processes of (i.) sale office and purchasing offices (ii.) purchasing and manufacturing office. Using the same criteria above, four activities is in this alternative namely: Activity A, "Check order", Activity B" sale + purchasing processes", Activity C "purchasing + manufacturing processes ", Activity D "sales + purchasing + manufacturing processes" as shown in Figure 4.7.

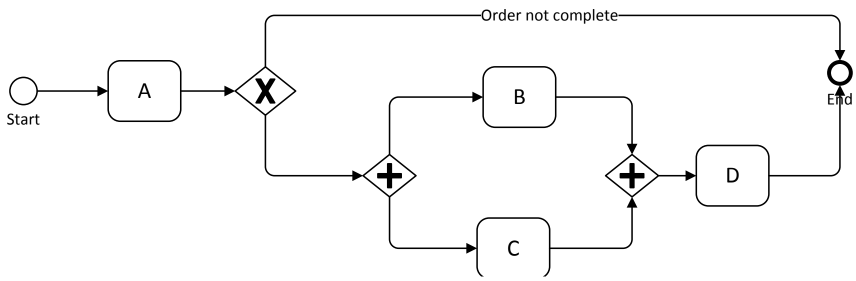


Fig. 5.5 Process model for the first alternative

Table 5.3 Data elements types in Figure 4.7.

Data Element	Fig 5.7
R_i	(14 and 15), 13 ,4, 2
O_i	$\{(2,3) \in 4\}, \{(5, 9, 6, 10, 7, 8, 11,12) \in 13\}, \{(5, 9, 13) \in (14, \text{and } 15)\}$
r_i	1, 4

The three alternatives are preferable, the short coming of third alternative is deadlock that might happen between task B and C. The first and second alternative show the level of granularity which tend to increase process flexibility. In section 5.4.2, we will formalize the alternatives to check if this verdict is valid.

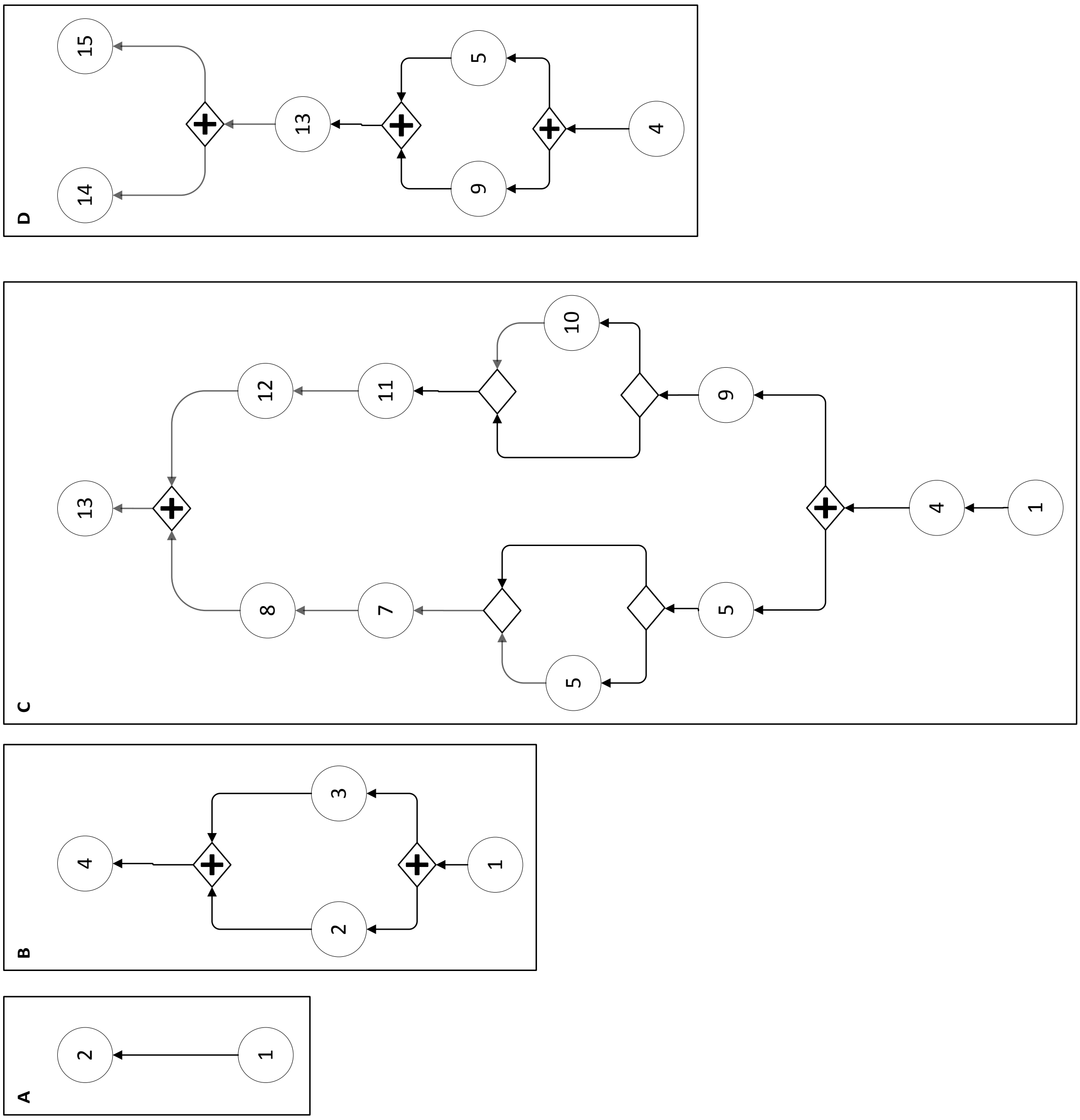


Fig. 5.6 Third alternative of the sofa ordering process

5.2.2 Formalization of the alternatives

The original process design Figure 4.2 is extracted of the alternatives to produces three data element structures resulting two process model (cf.fig 5.3 and 5.6). The set of data elements that are being processed denoted as **D** and the set of operations on the data elements denoted as **O** which are applied to formalize and generate the operation structure of our motivating example.

Definition 5.12 (Operations structure)

An operations structure is a tuple (D,W, O) with:

1. D: the set of data elements that are being processed.

   - D = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}

2. W = {w}, set of single instance types that are being processed.

   - O = {(2,w{1}), (3,w{1}), (4,w{2,3}), (7,w{5,6}), (11,w{10,9}), (13,w{8,12}), (14,15,w{13})}
   - $\bar{O}$ = {(2,{1}), (3,{1}), (4,{2,3}), (7,{5,6}), (11,{10,9}), (13,{8,12}), (14,15,{13})}
   - $\hat{O}$ = {(2, 1), (3, 1), (4,(2, 3)), (4,(2, 3)), (7,(5, 6)), (11,(10, 9)), (13,(8, 12)), ((14,15),13) }

In figure 9, merging activities G and H resulted into the second alternative: Combining O_{1_B} and O_{1_C}

$$O2_{\text{A}} = \{(2, w, \{1\})\}$$
$$O2_{\text{BC}} = \{(4, w, \{3,2\}), (5, w, \{3,2\}), (3, w, \{2\}), (6, w, \{3,2,4,5\})\}$$
$$O2_{\text{D}} = \{(7, w, \{4\})(8, w, \{7,4\}), (8, w, \{4\}))\}$$
$$O2_{\text{E}} = \{(5, w, \{5\})\}$$
$$O2_{\text{F}} = \{(9, w, \{8\})\}$$
$$O2_{\text{G}} = \{(5, w, \{5\})\}$$
$$O2_{\text{H}} = \{(10, w, \{5\}), (11, w, \{10,5\})), (11, w, \{5\}))\}$$
$$O2_{\text{I}} = \{(11, w, \{11\})\}$$
$$O2_{\text{J}} = \{(12, w, \{8,11\}), (13, w, \{12\})\}$$
$$O2_{\text{K}} = \{(9, w, \{13,9\})\}$$
$$O2_{\text{L}} = \{(14, w, \{6,13\}), (13, w, \{6,13\})\}$$

Combining $O2_{\text{H}}$ and $O2_{\text{I}}$

$$O3_{\text{A}} = \{(2, w, \{1\})\}$$
$$O3_{\text{BC}} = \{(4, w, \{3,2\}), (5, w, \{3,2\}), (3, w, \{2\}), (6, w, \{3,2,4,5\})\}$$
$$O3_{\text{D}} = \{(7, w, \{4\})(8, w, \{7,4\}), (8, w, \{4\}))\}$$
$$O3_{\text{E}} = \{(5, w, \{5\})\}$$
$$O3_{\text{F}} = \{(9, w, \{8\})\}$$
$$O3_{\text{G}} = \{(5, w, \{5\})\}$$
$$O3_{\text{HI}} = \{(10, w, \{5\}), (11, w, \{10,5\})), (11, w, \{5\}), (11, w, \{11\}))\}$$
$$O3_{\text{J}} = \{(12, w, \{8,11\}), (13, w, \{12\})\}$$
$$O3_{\text{K}} = \{(9, w, \{13,9\})\}$$
$$O3_{\text{L}} = \{(14, w, \{6,13\}), (13, w, \{6,13\})\}$$

Combining $O3_J$ and $O3_K$

$$O4_A = \{(2,w,\{1\})\}$$
$$O4_{BC} = \{(4,w,\{3,2\}),(5,w,\{3,2\}),(3,w,\{2\}),(6,w,\{3,2,4,5\})\}$$
$$O4_D = \{(7,w,\{4\})(8,w,\{7,4\}),(8,w,\{4\}))\}$$
$$O4_E = \{(5,w,\{5\})\}$$
$$O4_F = \{(9,w,\{8\})\}$$
$$O4_G = \{(5,w,\{5\})\}$$
$$O4_{HI} = \{(10,w,\{5\}),(11,w,\{10,5\})),(11,w,\{5\}),(11,w,\{11\}))\}$$
$$O4_{JK} = \{(12,w,\{8,11\}),(13,w,\{12\}),(9,w,\{13,9\})\}$$
$$O4_L = \{(14,w,\{6,13\}),(13,w,\{6,13\})\}$$

Combining $O4_{JK}$ and $O4_L$

$$O4_A = \{(2,w,\{1\})\}$$
$$O4_{BC} = \{(4,w,\{3,2\}),(5,w,\{3,2\}),(3,w,\{2\}),(6,w,\{3,2,4,5\})\}$$
$$O4_D = \{(7,w,\{4\})(8,w,\{7,4\}),(8,w,\{4\}))\}$$
$$O4_E = \{(5,w,\{5\})\}$$
$$O4_F = \{(9,w,\{8\})\}$$
$$O4_G = \{(5,w,\{5\})\}$$
$$O4_{HI} = \{(10,w,\{5\}),(11,w,\{10,5\})),(11,w,\{5\}),(11,w,\{11\}))\}$$
$$O4_{JKL} = \{(12,w,\{8,11\}),(13,w,\{12\}),(9,w,\{13,9\}),(14,w,\{6,13\}),(13,w,\{6,13\})\}$$

For each alternative (*1, 2, 3*), we calculate the metrics, the results are compared to aggregate the services based on defined conditions namely:

1. the cohesion value of each split in each alternatives.
2. if cohesion coefficient of alternative A is higher than the other, then the partition of alternative A is preferable.
3. if the cohesion coefficient of partition of alternative A is higher than other, then the larger activities of alternative A is preferable
4. in all cases, the heuristic is indecisive.

Definition 5.13 (Data cohesion)

The measure of activity responsibilities and tightness within a service is known as *cohesion*, the higher the cohesion, the higher the activity dependencies Vanderfeesten et al. (2008). For an activity T = (t,e) on an operations structure (D,O), the data cohesion c(t) is defined as follows:

$$c(t) = \lambda \cdot \mu(t)$$

where λ is the data relation cohesion and $\mu(t)$ is the data cohesion

Combining $O4_{JK}$ and $O4_{L}$ resulted into the outputting of data elements 14 and 13 which represents invoice and sofaPackage respectively in t_{20}. Therefore to calculate the cohesion of this JKL, separate relation cohesion and data cohesion is calculated between the two outputs (13 or 14) and the activity t_{17},t_{18}, and t_{19}.

$$\lambda_4(O_{4_{\text{JKL}}}) = \frac{|(12,12,13,13,13,13|}{4\times 3} = \frac{6}{12} = 0.5*2 = 1$$

$$\mu_4(JKL) = \frac{|12,13|}{6} = \frac{2}{6} = 0.333*2 = 0.667$$

$$c_4(JKL) = 1\times 0.667 = 0.667$$

Definition 5.14 (Process cohesion)

$$\boxed{ch = \frac{c(t)}{|n|}}$$

Therefore, the data cohesion of each service, is divided by the number of activities (n) in the service. i.e.

$$ch(1) = \frac{0+0.667+0+0.222+0+0+0+0.3335+0+0.25+0+0}{12} = 0.123$$

$$ch(2) = \frac{0+0.667+0.222+0+0+0+0.3335+0+0.25+0+0}{11} = 0.134$$

$$ch(3) = \frac{0+0.667+0.222+0+0+0+0.25+0.25+0+0}{10} = 0.139$$

$$ch(4) = \frac{0+0.667+0.222+0+0+0+0.25+0.267+0}{9} = 0.156$$

$$ch(5) = \frac{0+0.667+0.222+0+0+0+0.25+0.667}{8} = 0.226$$

Table 5.4 The total process coupling of the first alternative

	$O1_A$	$O1_B$	$O1_C$	$O1_D$	$O1_E$	$O1_F$	$O1_G$	$O1_H$	$O1_I$	$O1_J$	$O1_K$	$O1_L$	Total
$O1_A$	0	1	1	0	0	0	0	0	0	0	0	0	2
$O1_B$	1	0	1	1	1	0	1	1	0	0	0	0	6
$O1_C$	1	1	0	1	1	0	1	1	0	0	0	1	7
$O1_D$	0	1	1	0	0	1	0	0	0	1	0	0	4
$O1_E$	0	1	1	0	0	0	1	1	0	0	0	0	4
$O1_F$	0	0	0	1	0	0	0	0	0	1	1	0	3
$O1_G$	0	1	1	0	1	0	0	1	0	0	0	0	4
$O1_H$	0	1	1	0	1	0	1	0	1	1	0	0	6
$O1_I$	0	0	0	0	0	0	0	1	0	1	0	0	2
$O1_J$	0	0	0	1	0	1	0	1	1	0	1	1	6
$O1_K$	0	0	0	0	0	1	0	0	0	1	0	1	3
$O1_L$	0	0	0	0	0	0	0	0	0	1	1	0	3
Total	2	6	7	4	4	3	4	6	2	6	3	3	50

$$cp_1 = \frac{50}{12*11} = \frac{50}{132} = 0.379$$

$$cp_2 = \frac{36}{11*10} = \frac{36}{132} = 0.327$$

$$ch_3 = \frac{32}{10*9} = 0.356$$

$$ch_4 = \frac{27}{9*8} = 0.375$$

$$ch_5 = \frac{26}{8*7} = 0.464$$

$$ch_5 = \frac{26}{8*7} = 0.464$$

Definition 6 (Process coupling/cohesion ratio)

For the two alternative designs for the sofa production case the coupling/cohesion ratio is:

$$\Gamma_1 = \frac{cp_1}{ch_1} = \frac{0.379}{0.123} = 3.08$$

$$\Gamma_2 = \frac{0.327}{0.134} = 2.44$$

$$\Gamma_3 = \frac{0.356}{0.139} = 2.56$$

$$\Gamma_4 = \frac{0.375}{0.156} = 2.40$$

$$\Gamma_5 = \frac{0.464}{0.226} = 2.05$$

Overall evaluation

Table 5.5 C&C value for three alternative designs to the sofa production case

	Process cohesion	Process coupling	Coupling/cohesion ratio
Alternative 1	0.123	0.379	3.08
Alternative 2	0.134	0.327	2.44
Alternative 3	0.139	0.356	2.56

The C&C is used to find the preferable and best design between the alternatives. The design with lowest coupling/cohesion ratio is to be considered for usage. Therefore, the first design is preferable, it shows that alternative 5 contains activities that are not over granulated i.e. not too small or not too large as shown table 5.5.

5.3 Comparison between value-based and C&C-based service identification

it is important to state the comparison between the two service identifications based on their representation and formal merits. Firstly, it should be known that value-based service identification does not only contain core services, but also represent quality features relating to these services, such as “convenience” and “economical” but these services might be loosely coupled.

In the current age of global networks, the focus cannot limit itself to a quality features like economical or convenience but should be on re-usability of loosely-coupled services. The value model is also used in identifying more business services and specification of business rules and policies governing the services which enables semantic agreement among services. As stated in Nayak et al. (2007), “*the current trend toward a service-oriented enterprise necessitates a formal characterization of business architecture that reflects service-oriented business thinking.*” Therefore, we will make a comparison between these two service identifications using formal representation:

Main Contribution

Considering the above, evaluation of sofa production using C&C and the presented comparison could be summarized:

- the approach transforms the model
- the transformation differs from the result shown in Bianchini et al. (2014)
- the impact of coupling and cohesion has on the results based on the formalization, as discussed in the next sub-section

5.4 Impact of coupling and cohesion on service-oriented business process

The coupling and cohesion has most impact after the formalization and design of new service identification methods. The essence of using these coupling and cohesion metrics is to measure the adaptability and performance of the method for every business case that is tested on. It is important to note that, the majority of service identification methods rely on ad-hoc and experience-based business cases, which has limiting support for any varied case in a business domain. In ad-hoc-based service identification methods, sometimes the applicability or adaptability on similar ad-hoc is assumed to be a “blue sky" situation focusing on just a particular case.

Bianchini/Vanderfeesten approaches : It explains why Bianchini et.al paper has a different result differ from our result.

Table 5.6 C&C Metrics results of Bianchini and our calculation using Vanderfeesten approach

Author		Cohesion	Coupling	C&C ratio
Bianchini	Alternative	0.877	0.084	0.09578
Our Method	Alternative 1	0.873	0.8	1.39
	Alternative 2	0.828	0.917	1.107

The generated results from the two alternatives is compared with the Bianchini’s result (Bianchini et al., 2014), it shows that the results are different and better. Our approach has lower granularity in single system and also in collaboration.

Chapter 6

Conclusion

This research has investigated and demonstrated the possibility of identification of candidate services from business process model, business requirement and corresponding pre-defined goal, where data elements can be semantically represented by data modelling. This was in an attempt to simplify and reduce the maintenance cost and time wastage in process design. This approach will enhance and automate the service identification process, and whenever the change factors arise, organizations can modify there SOA-based system.

6.1 Research Outcome

The following summarries the main finding of this research in each chapter as follows:

- Chapter 1 discusses the key issues that affect organizations (e.g banks and manufacturing sectors) are wastage of process models and rigid IT spending. It is important for organization to focus on cost reduction and reuse of their process models. This research will pave way for a generalised, extensible and reusable framework ro generate their business services within their single and collaborative systems.

- Chapter 2 gives a full knowledge of BPM, its benefits and application of SOA into BPM, as addressed in Section 2.4. The important work in the chapter is the review of the major methods and approaches of service identification which pave way into understanding the shortcomings of the methods. With our framework, services can be identified regardless the size of the business or conceptual data, it is applicable to any organization type (*see Section 3*), adaptable to changes and fast error detection i.e. our approach can go through cycle of re-checks for any error i.e variability in the result for any given instance. Also, our approach supports information communication and organization culture. With the ever changing business demands, services identified from the Data-based Service Identification Framework (DbSIF) are clearly and technically described (e.g. transfer protocols, data formats), conceptual details (clearly unified and specified terms and standardized data models) which are intended to be in (open and closed) information environment, platform independent and widely diffusible.

- Chapter 3 investigates requirements of service identification from different types of information systems, from single systems to collaborative systems, from closed systems to open systems. The research is important for providing a solid foundation for further identifying services for developing different service-oriented systems.
  The chapter clearly state the importance of distinguishing the methodology to be used for every involvement (*i.e. single system or collaborative*). Same methodology cannot be used to identify services in all systems or organization. Therefore, organizations using different enterprises information systems can have different involvement and operations. In solving the issue, we introduce two-dimensional enterprise information system, one dimension shows the representation of organization system, i.e. "involvement" dimension and the other dimension represents operation of organizations which can be seen as open or closed for other partners, i.e. "operation" dimension. For any the enterprise information systems in any axis or dimension, certain service orientation are

defined which further assist in business service identification with the right granularity. In doing so, we state the requirement for service identification for each dimension.

- Chapter 4 introduces the data representation within business processes in single and collaborative systems which introduce the semi-automatic approach into data extraction from business process and requirement to produce data-based model. Chapter 4 also present the use of data-based model (data dependency graph) to capture data and proto-services in generating candidate service based on semantic relationship and attributes.
  Finally, we introduced the data-driven service identification framework.

- Chapter 5 presents the evaluating procedure for the method and framework, by using a case study to validate the process.

In conclusion, this approach is applicable to any organization, regardless the operation size, and the type of system they run.

References

Aalst, V. D., Wil, M., Hofstede, T., and Arthur, H. (2003). Workflow Patterns. *Distributed and Parallel Databases*, 14:5–51.

Aalst, W. M. P., Hee, K. M., Hofstede, a. H. M., Sidorova, N., Verbeek, H. M. W., Voorhoeve, M., and Wynn, M. T. (2010). Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects of Computing*, 23(3):333–363.

Aalst, W. M. P. V. D. (2003). Challenges in business process management: Verification of business processes using Petri nets. *Bulletin of the EATCS 80*, pages 174–199.

Abuhussein, A., Bedi, H., and Shiva, S. (2014). Exploring Security and Privacy Risks of SoA Solutions Deployed on the Cloud. (JULY).

Aburub, F. and Almahamid, M. (2010). A method for deriving system models based on business process models. *Journal of Information Technology Management*, 21(2):35–43.

Adjoyan, S., Seriai, A.-D., and Shatnawi, A. (2014). Service Identification Based on Quality Metrics. *Proceedings of the 26 International Conference on Software Engineering & Knowledge Engineering (SEKE2014)*, pages 1–6.

Afsarmanesh, H., Ermilova, E., Msanjila, S. S., Torre, Q., and Capatica, M. (2009). Modeling and Management of Information Supporting Functional Dimension of Collaborative Networks. In *Transactions on Large-scale Data-and Knowledge-centered Systems I*, pages 1–37.

Afsarmanesh, L. M. C.-m. H. (2007). A comprehensive modeling framework for collaborative networked organizations. *ournal of Intelligent Manufacturing*, 18(5):529–542.

Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2):129–149.

Aiello, M., Bulanov, P., and Groefsema, H. (2010). Requirements and Tools for Variability Management. *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, pages 245–250.

Alahmari, S., Zaluska, E., and De Roure, D. (2010). A service identification framework for legacy system migration into SOA. *Proceedings - 2010 IEEE 7th International Conference on Services Computing, SCC 2010*, pages 614–617.

Aldris, A., Nugroho, A., Lago, P., and Visser, J. (2013). Measuring the degree of service orientation in proprietary soa systems. In *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, pages 233–244. IEEE.

Alves, C., Alves, V., and Niu, N. (2013). A systematic mapping study on business. 5(1):1–21.

Amsden, J. (2010). Modeling with SoaML, the Service-Oriented Architecture Modeling Language. pages 1–16.

Antunes, P. and Mourão, H. (2011). Resilient Business Process Management: Framework and services. *Expert Systems with Applications*, 38(2):1241–1254.

Arsanjani, a. (2004). Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., and Holley, K. (2008). SOMA: A method for developing service-oriented solutions. *IBM Systems Journal*, 47(3):377–396.

Aur, M., Bendraou, R., Robin, J., and Blanc, X. (2011). Flexible Deviation Handling during Software Process Enactment. *In Enterprise Distributed Object Computing Conference Workshops (EDOCW).*, pages 34–41.

Aversano, L., Cerulo, L., and Palumbo, C. (2008). Mining candidate web services from legacy code. *Proceedings - 10th IEEE International Symposium on Web Site Evolution, WSE 2008*, pages 37–40.

Ayora, C., Torres, V., Reichert, M., and Weber, B. (2012). Towards Run-time Flexibility for Process Families : Open Issues and Research Challenges. pages 1–12.

Azevedo, L. G., Santoro, F., Baião, F., Souza, J., Revoredo, K., Pereira, V., and Herlain, I. (2009). A method for service identification from business process models in a SOA approach. *Lecture Notes in Business Information Processing*, 29 LNBIP:99–112.

Baghdadi, Y. (2006). Reverse engineering relational databases to identify and specify basic Web services with respect to service oriented computing. *Information Systems Frontiers*, 8(5):395–410.

Bajwa, I. S., Samad, A., Mumtaz, S., Kazmi, R., and Choudhary, A. (2009). BPM Meeting with SOA: A Customized Solution for Small Business Enterprises. *2009 International Conference on Information Management and Engineering*, pages 677–682.

Behara, G. K. (2006). BPM and SOA : A Strategic Alliance. (5):1–7.

Beimborn, D. and Joachim, N. (2011). The joint impact of service-oriented architectures and business process management on business process quality: An empirical evaluation and comparison. *Information Systems and e-Business Management*, 9(3):333–362.

Benner, M. J. and Tushman, M. L. (2003). Exploitation, exploration, and process management: The productivity dilemma revisited. *Academy of management review*, 28(2):238–256.

Bennett, P., Giles, L., and Halevy, A. (2013). Channeling the Deluge : Research Challenges for Big Data and Information Systems. In *22nd ACM international conference on Conference on information & knowledge management*, pages 2537–2538.

Bianchini, D., Cappiello, C., De Antonellis, V., and Pernici, B. (2009). P2S: A methodology to enable inter-organizational process design through web services. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5565 LNCS:334–348.

Bianchini, D., Cappiello, C., De Antonellis, V., and Pernici, B. (2014). Service Identification in Interorganizational Process Design. *IEEE Transactions on Services Computing*, 7(2):265–278.

Bianchini, D., Pagliarecci, F., and Spalazzi, L. (2011). From service identification to service selection: An interleaved perspective. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7000 LNCS:223–240.

Binkley, D. (2007). Source Code Analysis: A Road Map. *Proc. FOSE*, pages 104–119.

Boerner, R. and Goeken, M. (2009). Service identification in SOA governance literature review and implications for a new method. *2009 3rd IEEE International Conference on Digital Ecosystems and Technologies, DEST '09*, pages 588–593.

Brown, B., Chui, M., and Manyika, J. (2011). Are you ready for the era of Big Data?

Cai, S., Liu, Y., and Wang, X. (2011). A survey of service identification strategies. *Proceedings - 2011 IEEE Asia-Pacific Services Computing Conference, APSCC 2011*, pages 464–470.

Camarinha-Matos, L. M., Boucher, X., and Afsarmanesh, H. (2010). *Collaborative networks for a sustainable world*. Springer.

Carney, D. J., Smith, I., Place, P. R., et al. (2005). Topics in interoperability: Infrastructure replacement in a system of systems.

Chappell, D. (2004). *Enterprise service bus*. " O'Reilly Media, Inc.".

Chen, F., Li, S., Yang, H., Wang, C.-h., and Chu, W. C.-c. (2005). Feature Analysis for Service-Oriented Reengineering. *Software Engineering Conference, 2005. APSEC'05.*, (12th Asia-Pacific):8.

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., and Rackham, G. (2005). Impact of service orientation at the business level. *IBM Systems Journal*, 44(4):653–668.

Chinosi, M. and Trombetta, A. (2012a). BPMN : An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.

Chinosi, M. and Trombetta, A. (2012b). BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.

Cho, M. J., Choi, H. R., Kim, H. S., Hong, S. G., Keceli, Y., and Park, J. Y. (2008). Service Identification and Modeling for Service Oriented Architecture Applications. *Sepads 08: Proceedings of the 7th Wseas International Conference on Software Engineering, Parallel and Distributed Systems*, pages 193–199.

Combi, C., Gambini, M., and Migliorini, S. (2011). Towards Structured Business Process Modeling Languages. *In ADBIS*, 2:1–10.

Cukier, K. and Mayer-Schoenberger, V. (2013). Rise of Big Data: How it's Changing the Way We Think about the World. 1:2013.

Cull, R. and Eldabi, T. (2010). HYBRID PROCESS MODELLING WITHIN BUSINESS PROCESS. *European and Mediterranean Conference on Information Systems*, page 2010.

Dadam, P. and Reichert, M. (2009). The ADEPT project: a decade of research and development for robust and flexible process support. *Computer Science - Research and Development*, 23(2):81–97.

Deng, Z., Liu, J. W. S., Zhang, L., Mouna, S., and Frei, A. (1999). An open environment for real-time applications. In Operating Systems and Services.

Dijkman, R., Rosa, M. L., and Reijers, H. a. (2012). Managing large collections of business process models—Current techniques and challenges. *Computers in Industry*, 63(2):91–97.

Dumas, M., Rosa, M. L., Mendling, J., and Hajo, A, R. (2013). *Fundamentals of Business Process Management*.

Dumas, M., van der Aalst, W. M. P., and A. H.M. ter Hofstede (2005). *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, New York.

Dwivedi, V. and Kulkarni, N. (2008). A model driven service identification approach for process centric systems. *Proceedings - 2008 IEEE Congress on Services, SERVICES 2008*, pages 65–72.

Ekanayake, C. C., Chathura, C., Rosa, L., Arthur, H. M., Christine, M., Rosa, M. L., Hofstede, A. H. M., and Fauvet, M. (2010). Fragment-based version Management for repositories of business process model. Technical report.

Ellis, C. A. (2005). Workflow Technology. page 14.

Elvesæter, B., Hahn, A., Berre, A.-J., and Neple, T. (2006). Towards an interoperability framework for model-driven development of software systems. In *Interoperability of enterprise software and applications*, pages 409–420. Springer.

Erl, T. (2008). *SOA design patterns*. Pearson Education.

Ese, T. H. (2012). Towards Creating Context-Aware Dynamically-Adaptable Business Processes Using Complex Event Processing.

Espinosa, J. A., Boh, W. F., and DeLone, W. (2011). The organizational impact of enterprise architecture: A research framework. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1–10.

Fantinato, M., Felgar de Toledo, M. B., Thom, L. H., Gimenes , Itana, M. d. S., and Garcia, D. Z. G. (2012). A survey on reuse in the business process management domain. *Business Process Integration and Management*, 6(1):52–76.

Fareghzadeh, N. (2008). Service identification approach to SOA development. *Proceedings of World Academy of Science Engineering and Technology*, 35(November):258–266.

Fisher, M., Elbaum, S., and Rothermel, G. (2013). An automated analysis methodology to detect inconsistencies in web services with WSDL interfaces. (February 2011):27–51.

Flaxer, D. and Niga, A. (2004). Realizing business components, business operations and business services. *Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business, CEC-East 2004*, pages 328–332.

Fujimoto, H. (2008). Collaborative networking in a multi-stage industrial channel. *International Journal of Physical Distribution & Logistics Management*, 33(3):229–235.

Gao, X. (2013). Towards the Next Generation Intelligent BPM –. *11th International Conference, BPM 2013*, pages 4–9.

Garimella, K., Lees, M., and Williams, B. (2008). *BPM basics for dummies*. Wiley.

Georgakopoulos, Diimitrios, Hornick, M., and Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and parallel Databases*, 3(2):119–153.

Gillot, J.-N. (2008). *The Complete Guide to Business Process Management: Business process transformation or a way of aligning the strategic objectives of the company and the information system through the processes*.

Goh, C. H. T. and Hooper, V. (2009). Knowledge and information sharing in a closed information environment. *Journal of Knowledge Management*, 13(2):21–34.

Grefen, P., Eshuis, R., Mehandjiev, N., Kouvas, G., and Weichhart, G. (2009). Internet-based support for process-oriented instant virtual enterprises. *IEEE Internet Computing*, 13(6):65–73.

Groefsema, H., Bulanov, P., and Aiello, M. (2012). Imperative versus Declarative Process Variability : Why Choose ? pages 1–52.

Gu, Q. and Lago, P. (2010). Service Identification Methods: A Systematic Literature Review. In *Towards a Service-Based Internet*, volume 35, pages 37–50. Elsevier.

Guan, Q., Feng, S., and Ma, Y. (2012). A Network Topology Clustering Algorithm for Service Identification. *2012 International Conference on Computer Science and Service System*, pages 1583–1586.

Harmon, P. (2010). The scope and evolution of business process management. In *Handbook on Business Process Management 1*, pages 37–81. Springer Berlin Heidelberg.

Heffner, R., Leganza, G., and Blackburn, L. (2011). Soa adoption 2010: Still important, still strong. *Forrester Research*.

Heidari, F., Loucopoulos, P., Brazier, F., and Barjis, J. (2013). A Meta-Meta-Model for Seven Business Process Modeling Languages. *2013 IEEE 15th Conference on Business Informatics*, pages 216–221.

Hepp, M., Leymann, F., Domingue, J., Wahler, a., and Fensel, D. (2005). Semantic business process management: a vision towards using semantic Web services for business process management. *IEEE International Conference on e-Business Engineering (ICEBE'05).*

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105.

Hirschheim, R., Welke, R., and Schwarz, A. (2010). Service-oriented architecture:myths, realities, and a maturity model. *MIS Quarterly Executive*, 9(1):37–48.

Huergo, R. S., Pires, P. F., and Delicato, F. C. (2014). A method to identify services using master data and artifact-centric modeling approach. *Proceedings of the 29th Annual ACM Symposium on Applied Computing - SAC '14*, pages 1225–1230.

Huhns, Michael, N. and Singh, M. P. (2005). Service-oriented computing: Key concepts and principles. *Internet Computing, IEEE*, 9(1):75–81.

Inaganti, S. and Behara, G. K. (2007). Service Identification : BPM and SOA Handshake. *BPTrends*, 3(March):1–12.

Indulska, M., Recker, J., Rosemann, M., and Green, P. (2009). Business process modeling : current issues and future challenges. *21st International Conference on Advanced Information Systems*, (June):8–12.

Jabbar, Z. A., Kumar, M., and Samreen, A. (2015). Designing Conceptual Framework for Aligning Service Oriented Architecture with Business Process. (April):11–22.

James, M. (2014). In-depth: BPM and SOA for business agility (Next-generation enterprise IT).

Jamshidi, P., Sharifi, M., and Mansour, S. (2008). To establish enterprise service model from enterprise business model. *Proceedings - 2008 IEEE International Conference on Services Computing, SCC 2008*, 1:93–100.

Jeston, J. and Nelis, J. (2014). *Business process management*. Routledge, Jeston2014.

Jin, J. and Ahn, G.-J. (2006). Role-based access management for ad-hoc collaborative sharing. In *Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 200–209. ACM.

Johnston, R. (2004). Towards a better understanding of service excellence. *Managing Service Quality: An International Journal*, 14(2/3):129–133.

Jones, O. and Crompton, H. (2009). Enterprise logic and small firms: a model of authentic entrepreneurial leadership. *Journal of Strategy and Management*, 2(4):329–351.

Josuttis, N. M. (2007). *SOA in Practice . The Art of Distributed System Design*. Paris.

Jr, T. R. G. and Sommer, R. a. (2002). Business process management: public sector implications. *Business Process Management Journal*, 8(4):364–376.

Juric, M. B. and Pant, K. (2008). *Business Process Driven SOA Using BPMN and BPEL : From Business Process Modeling to Orchestration and Service Oriented Architecture*.

Kang, D., Song, C. Y., and Baik, D. K. (2008). A method of service identification for product line. *Proceedings - 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008*, 2:1040–1045.

Kim, Y. and Doh, K.-G. (2007). The service modeling process based on use case refactoring. *Business information systems*, pages 108–120.

Kim, Y. and Doh, K. G. (2009). Formal identification of right-grained services for service-oriented modeling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5802 LNCS:261–273.

Kirkham, T., Winfield, S., Haberecht, T., Müller, J., Alessandro, I., and Consiglio, F. (2011). The Challenge of Dynamic Services in Business Process Management. Technical Report 3.

Klose, K., Knackstedt, R., and Beverungen, D. (2007). Identification of services. A stakeholder based approach to SOA development and its application in the area of production planning. *Proceedings of the 15th European Conference on Information Systems*, pages 1802–1814.

Ko, R. K., Lee, S. G., and Lee, E. (2009). Business Process Management (BPM) standards: a survey. *Business Process Management Journal*, 15(5):744 – 791.

Ko, R. K. L. (2009). A Computer Scientist's Introductory Guide to Business Process Management (BPM). *Crossroads XRDS*, 15(4):11–18.

Koehler, J., Tirenni, G., and Kumaran, S. (2014). From business process model to consistent implementation: a case for formal verification methods. *Proceedings. Sixth International Enterprise Distributed Object Computing*, pages 96–106.

Kohlborn, T., Korthaus, A., Chan, T., and Rosemann., M. (2009a). Identification and analysis of business and software services-a consolidated approach. *IEEE Transactions on Services Computing*, 2(1):50–64.

Kohlborn, T., Korthaus, A., Chan, T., and Rosemann, M. (2009b). Identification and Analysis of Business and Software Services ;A Consolidated Approach. *IEEE Transactions on Services Computing*, 2(1):50–64.

Kohlmann, F. and Alt, R. (2007). Business-Driven Service Modeling - A Methodological Approach from the Finance Industry. *Sabre 2007*, pages 1–14.

Komoda, N. (2006). Service oriented architecture (soa) in industrial systems. In *2006 4th IEEE International Conference on Industrial Informatics*, pages 1–5. IEEE.

Koschmider, A., Fellmann, M., Schoknecht, A., and Oberweis, A. (2014). Analysis of Process Model Reuse: Where Are We Now, Where Should We Go From Here? *Decision Support Systems*, page 37.

Kumaran, S., Liu, R., and Wu, F. Y. (2008). On the duality of information-centric and activity-centric models of business processes. *Advanced Information Systems Engineering.*, pages 32–47.

Lampe, U., Müller, A., Wenge, O., and Schaarschmidt, R. (2013). On the relevance of security risks for cloud adoption in the financial industry. *Proceedings of the 19th Americas Conference on Information Systems (AMCIS 2013)*, pages 2537–2544.

Lankhorst, M. (2005). *Enterprise architecture at work: Modelling, communication and analysis*. Berlin Heidelberg Springer-Verlag.

Legner, C. and Heutschi, R. (2007). SOA Adoption in Practice - Findings from early SOA Implementations. *Proceedings of the Fifteenth European Conference on Information Systems*, (February):1643–1654.

Li, C., Reichert, M., and Wombacher, A. (2011). Mining business process variants: Challenges, scenarios, algorithms. *Data & Knowledge Engineering*, 70(5):409–434.

Li, G., Muthusamy, V., and Jacobsen, H.-A. (2009). A distributed service-oriented architecture for business process execution. *ACM Transactions on the Web*, 4(1):1–33.

Li, L. (2007). *Supply chain management: concepts, techniques and practices enhancing the value through collaboration*. World Scientific.

Lins, F., Damasceno, J., Souzal, A., Bruno, S., Aragão1Medeiros, R., Sousa, E., and Rosa, N. (2012). Towards automation of SOA-based business processes. *International Journal of Computer Science, Engineering and Applications*, 2(2):1–17.

Liu, C., Li, Q., and Zhao, X. (2008). Challenges and opportunities in collaborative business process management: Overview of recent advances and introduction to the special issue. *Information Systems Frontiers*, 11(3):201–209.

Liu, S., Xiong, G., Fan, D., and Yao, J. (2011). An optimized modeling method based on BPEL4WS. *Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics*, pages 57–61.

Luo, M., Goldshlager, B., and Zhang, L.-J. (2005). Designing and implementing enterprise service bus (esb) and soa solutions. In *2005 IEEE International Conference on Services Computing (SCC'05) Vol-1*, volume 2, pages xiv–vol. IEEE.

M., W. J. A. (2006). Service-Oriented Design and Development Methodology. *International Journal of Web Engineering and Technology*, 4(2):412–442.

Ma, Q., Zhou, N., Zhu, Y., and Wang, H. (2009). Evaluating service identification with design metrics on business process decomposition. *SCC 2009 - 2009 IEEE International Conference on Services Computing*, pages 160–167.

Ma, S.-p., Fanjiang, Y.-y., and Kuo, J.-Y. (2014). Dynamic service Composition Using Core Service identification. *Journal of information science and engineering*, 30(4):957–972.

Maggi, F. M., Montali, M., Westergaard, M., and Aalst, W. M. P. V. D. (2011). Monitoring Business Constraints with Linear Temporal Logic : An Approach Based on Colored Automata. *Business Process Management*, pages 132–147.

Mai, H. (2012). IT in banks : What does it cost ? *Deutsche Bank DB Research*.

Mani, S., Sinha, V. S., Sukaviriya, N., and Ramachandra, T. (2008). Using user interface design to enhance service identification. *Proceedings of the IEEE International Conference on Web Services, ICWS 2008*, pages 78–87.

Mayer-Schönberger, Viktor, and Kenneth, C. (2013). *Big data: A revolution that will transform how we live, work, and think*.

M.Bridgeland and R.Zahavi (2008). Business Modeling: A Practical Guide to Realizing Business Value. *MK/OMG Press*.

Mendling, J. (2008). Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. *Business Information Processing*, 6(193).

Mendling, J., Neumann, G., and Nuttgens, M. (2005). A comparison of XML interchange formats for business process modelling. In *Workflow handbook*, pages 185–198.

Mendling, J., Reijers, H., and van der Aalst, W. (2010). Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52(2):127–136.

Menzel, M., Thomas, I., and Meinel, C. (2009). Security Requirements Specification in Service-Oriented Business Process Management. *2009 International Conference on Availability, Reliability and Security*, pages 41–48.

Minson, J. A. and Mueller, J. S. (2012). The cost of collaboration why joint decision making exacerbates rejection of outside information. *Psychological Science*, 23(3):219–224.

Mirarab, A., Fard, N. G., Reza, A., and Kenari, R. (2014). A New Framework for Service Identification in SOA. 14(5).

Mohamed, M., Mohamed, B. S., El, M., and Chergui, A. (2014). A Hybrid Particle Swarm Optimization For Service Identification From Business Process.

Moore, C. W. (2014). *The mediation process: Practical strategies for resolving conflict*. John Wiley & Sons.

Nayak, N., Linehan, M., Nigam, A., Marston, D., Jeng, J.-J., Wu, F. Y., Boullery, D., White, L. F., Nandi, P., and Sanz, J. L. C. (2007). Core business architecture for a service-oriented enterprise. *IBM Systems Journal*, pages 723–742.

Niehaves, B. and Plattfaut, R. (2010). Collaborative Business Process Management : Exploring Themes , Achievements , and Perspectives. *ECIS*.

Nigam, Anil, and Caswell., N. S. (2003). Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3):428–445.

Noel, J. (2005). BPM and SOA: Better Together. pages 1–12.

Norta, A. and Grefen, P. (2007). Discovering patterns for inter-organizational business process collaboration. *International Journal of Cooperative Information Systems*, page 37.

Nowak, A., Leymann, F., and Schumm, D. (2011). The Differences and Commonalities between Green and Conventional Business Process Management. *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, pages 569–576.

Oliveira, P. and Roth, A. V. (2012). Service orientation: the derivation of underlying constructs and measures. *International Journal of Operations & Production Management*, 32(2):156–190.

OMG (2014). Business Process Modelling Notation.

Panetto, H. (2007). Towards a classification framework for interoperability of enterprise applications. *International Journal of Computer Integrated Manufacturing*, 20(8):727–740.

Papazoglou, M. (2003). Service-oriented computing: concepts, characteristics and directions. *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, pages 3–12.

Papazoglou, M. and Heuvel, W. V.-D. (2004). Service Oriented Computing: State-of-the-Art and Open Research Issues. *Security Technical Report*.

Papazoglou, M., Traverso, P., Dustdar, S., and Leymann, F. (2007a). Service-oriented computing: State of the art and research directions. *IEEE Computer Society*, 40(11):64–71.

Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. (2007b). Service-oriented computing: State of the art and research challenges. *IEEE Computer*, 40 (November 2007(11):38–45.

Papazoglou, M. P. and Van Den Heuvel, W.-J. (2007). Life Cycle Methodology. *Communications of the ACM*, 50(10):79–85.

Paper, E. (2007). 10 Steps to Business-Driven SOA Visionary architecture always requires good building plans ! (March).

Pappano, L. (2012). The Year of the MOOC. Technical report.

Purcell, J. L., Supakkul, T. K., Thomas, M., and Wong, J. (2013). Dynamically optimized distributed cloud computing-based business process management (BPM) system.

Quartel, D., Dijkman, R., and Van Sinderen, M. (2004). Methodological support for service-oriented design with ISDL. *Proc. of the Second Int. Conf. on Service Oriented Computing (Icsoc 2004*, pages 1 – 10.

Rajsiri, V., Lorré, J.-P., Fréderick, B., and Pingaud, H. (2007). Collaborative process definition using an ontology-based approach. *In Working Conference on Virtual Enterprises*, pages 205–212.

Reichert, M. and Weber, B. (2012). Enabling Flexibility in Process-Aware Information Systems. Technical Report c, Berlin, Heidelberg.

Reijers, H. a. (2003). *Design and Control of Workflow Processes: Business Process Management for the Service Industry*.

Reijers, H. a., Limam, S., and van der Aalst, W. M. P. (2003). Product-Based Workflow Design. *Journal of Management Information Systems*, 20(1):229–262.

Research, C. (2013). Big data : what ' s holding you back. Technical Report April.

Ricca, F., Marchetto, A., Bruno, F., and Irst, K. (2009). A_quick and dirty meet-in-the-middle approach for migrating to SOA. *Proceedings of the joint international and annual ...*, pages 73–77.

Rim Samia, Souveyet, C., and Rolland, C. (2004). Eliciting service composition in a goal driven manner. *Proceedings of the 2nd international conference on Service oriented computing - ICSOC '04*, page 308.

Rosemann, M. and Recker, J. (2006). Context-aware process design exploring the extrinsic drivers for process flexibility. *BPMDS*.

Sadiq, W. and Orlowska, M. E. (2000). Analyzing process models using graph reduction techniques. *Information Systems*, 25(2):117–134.

Sanz, J. L. C., Nayak, N., and Becker, V. (2006). Business services as a modeling approach for smart business networks.

Sapir, A., Wolff, G. B., et al. (2016). One market, two monies: the european union and the united kingdom. Technical report.

Saylam, R. and Sahingoz, O. K. (2013). Process Mining in BPM:Concepts and challenges. *Electronics, Computer and Computation (ICECCO), International Conference on Digital Object Identifier*, pages 131–134.

Schmidt, M.-T., Hutchison, B., Lambros, P., and Phippen, R. (2005). The enterprise service bus: making service-oriented architecture real. *IBM Systems Journal*, 44(4):781–797.

Schonenberg, M. H., Mans, R. S., Russell, N. C., and Mulyar, N. A. (2008). Process Flexibility : a Survey of Contemporary Approaches. *In Advances in Enterprise Engineering I*, pages 16–30.

Schulte, S., Janiesch, C., Venugopal, S., Weber, I., and Hoenisch, P. (2014). Elastic Business Process Management: State of the art and open challenges for BPM in the cloud. *Future Generation Computer Systems*, 46:36–50.

Schulte, S., Janiesch, C., Venugopal, S., Weber, I., and Hoenisch, P. (2015). Elastic Business Process Management: State of the art and open challenges for BPM in the cloud. *Future Generation Computer Systems*, 46:36–50.

Si, H., Ni, Y., Yu, L., and Chen, Z. (2009). A service-oriented analysis and modeling using use case approach. *Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, (60773163).

Smith, H. and Fingar, P. (2006). *Business process management: the third wave*, volume 1.

Sofela, O., Xu, L., and Vrieze, d. P. (2013). Context-Aware Process Modelling through Imperative and Declarative Approach. In And, L. M. C.-M. and Scherer, R. J., editors, *Collaborative Systems for Reindustrialization - 14th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2013*, pages 191–200, Dresden, Germany. Springer.

Sofela, O., Xu, L., and Vrieze, d. P. (2016). The Needs of Service Identification for Service-Oriented Business Process Management. Conference: BCS Quality SG's SQM 2016 conference, Dorset, England.

Stoilova, K., Stoilov, T., Stoilova, K. P., and Stoilov, T. a. (2006). Evolution of the workflow management systems. In *International Scientific Conference on Information, Communication and Energy Systems and Technologies*, pages 225–228.

Suntae, K., Minseong, K., and Sooyong, P. (2008). Service identification using goal and scenario in service oriented architecture. *Neonatal, Paediatric and Child Health Nursing*, pages 419–426.

Trkman, P. (2010). The critical success factors of business process management. *International Journal of Information Management*, 30:125–134.

Turiera, T. and Cros, S. (2013). *50 examples of business collaboration*. Zero Factory S.L., 2nd edition.

Uottawa Institute (2013). Introduction to Petri Nets.

Van Der Aalst, W., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., van den Brand, P., Brandtjen, R., Buijs, J., et al. (2011). Process mining manifesto. pages 169–194.

Van Der Aalst, W. M. and Weske, M. . (2013). Interorganizational Workflows An approach based on Message Sequence Charts and Petri Nets. *Seminal Contributions to Information Systems Engineering: 25 Years of CAiSE*, page 289.

van der Aalst, W. M. P. (2013). Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, 2013:1–37.

Vanderfeesten, I., Reijers, H. a., and van der Aalst, W. M. (2008). Evaluating workflow process designs using cohesion and coupling metrics. *Computers in Industry*, 59(5):420–437.

Vanderfeesten, I., Reijers, H. a., and Van Der Aalst, W. M. P. (2011). Product-based workflow support. *Information Systems*, 36(2):517–535.

Vemulapalli, A. and Subramanian, N. (2009). Transforming functional requirements from UML into BPEL to efficiently develop SOA-based systems. *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, pages 337–349.

Verma, N. (2009). *Business process management: profiting from process*. Global India Publications.

Verner, L. (2004). BPM: The Promise and the Challenge. (March).

Verschuren, P. and Hartog, R. O. B. (2005). Evaluation in Design-Oriented Research. pages 733–762.

Vincent Cerf (1998). Where the wizards stay up late.

Wang, M. and Wang, H. (2006). From process logic to business logic - A cognitive approach to business process management. *Information and Management*, 43(2):179–193.

Weske, M. (2009). *Business process management- Concepts, Languages*. Springer Berlin Heidelberg New York.

Westerman, J. (2009). The Case for Business Process Management. Technical report.

Wetherly, P. and Otter, D. (2014). *The business environment: themes and issues in a globalizing world*. Oxford University Press.

White, S. (2006). BPMN 1.0 business process modeling notation — OMG final adopted specification.

Workpoint LLC (2011). WOrkpoint: BPM with Torque.

Wu, B., Lin, R., Liu, Z., Chen, J., and Peng, Y. (2013). Rapid Service-Oriented business process generation method based on ontology. *China Communications*, 10(9):13–32.

Wynn, M., Verbeek, H., Aalst, W. V. D., Hofstede, a. T., and Edmond, D. (2009). Business process verification – finally a reality! *Business Process Management Journal*, 15(1):74–92.

Yung, P. and Wong, H. (2011). *Formalisations and Applications of Business Process Modelling Notation*. PhD thesis.

Zadeh, A. T. (2011). Service Identification in SMEs : Appropriate Elements and Methods. 3(Icmlc):50–54.

Zadeh, A. T., Mukhtar, M., Sahran, S., and Khabbazi, M. R. (2012). A systematic input selection for service identification in SMEs.

Zairi, M. (1997). Business process management: a boundaryless approach to modern competitiveness. *Business Process Management Journal*, 3(1):64–80.

Zhang, Z., Liu, R., and Yang, H. (2005). Service Identification and Packaging in Service Oriented Reengineering. *Proc of the 17th International Conference*, pages 241–249.

Zhang, Z. and Yang, H. (2004). Incubating services in legacy systems for architectural migration. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, pages 196–203.

Zimmerer, T. W., Scarborough, N. M., and Wilson, D. (2008). *Essentials of Enterpreneurship and Small Business management*. Pearson Prentice Hall.

Zimmermann, O., Doubrovski, V., Grundler, J., and Hogg, K. (2005). Service-Oriented Architecture and Business Process Choreography in an Order Management Scenario : Rationale , Concepts , Lessons Learned. *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications - OOPSLA '05*, pages 301–312.

Zimmermann, O., Krogdahl, P., and Gee, C. (2009). Elements of Service-Oriented Analysis and Design. pages 1–18.

Zur Muehlen, M. (2004). *Workflow-based process controlling: foundation, design, and application of workflow-driven process information systems*.

Appendix A

Appendix I

A.1 Data extraction algorithm

The algorithm displays the steps in achieving the generating data elements and proto-services. Line 1-3 of algorithm extracts the data from each activities in the business process. Each input/output are associated to a complex I/O name and its attributes. Each activities displays its input data at the beginning and the output at the end of the process's activities A = [Order, OrderEval] as shown in Table 4.1. These data elements of other activities are partitioned in relation to each activity $A_{1...n}$

In any running example in Table 4.1 that contains set of t_i with information element d_i, set of task input IN_d, and task output OUT_d

Line 4-5 of algorithm 1 extracts the data from each process model and interconnect using data dependency graph .

Line 5-6 demonstrates the use of dependency action and data type in the data representation. The *If* statement in line 5 and 7 means the if more than one activity creates specific data, this may be data quality problem. For example, Table 4.1 shows that OrderEval *C(OrderEval)* is created whenever order input is read *R(Order)*, in such case, if there is another activity with *C(OrderEval)* as output, then this shows the right order evaluation is not correctly done

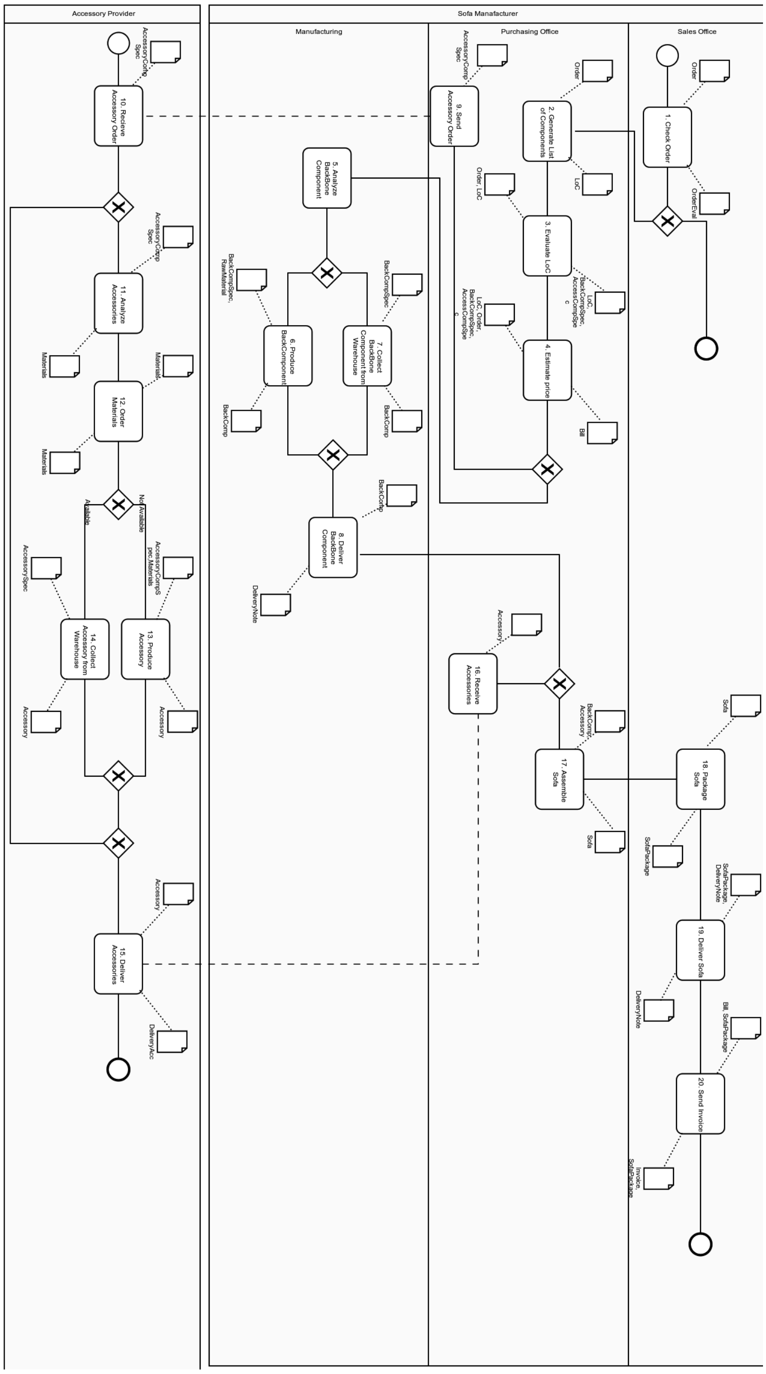


Fig. A.1 Sofa Production and Ordering Process which can lead to less data quality, consistency and efficiency. It is necessary to check the data output for repetitions.

INPUT: the set of Business process, $\mathscr{BP}_m$ = [bp_1,bp_2,.. bp_i,...bp_n], 0≥i ≥n, each containing set of activities A = [A_1, A_2..A_x,..A_y], 0≥x ≥y

OUTPUT: The set ∑ of data objects ∈ { set $\check{o}$ of R_i, O_i, r_i}

Begin

1. **For** each Business Process Model bp_1 in $\mathscr{BP}_m$ **do**
2. *add each data functionality (IN_d, OUT_d) as A = [A_1, A_2..A_x,..A_y]*, 0≥x ≥y
3. ***End for***
4. ***foreach*** *Activity* $A_y \in \mathscr{BP}_m$ ***do***
5. create a partition $\mathscr{P}_A$
6. add every data element (R_i, O_i, r_i) that is read, updated and deleted from the in each activity $A_{i_{1...n}}$ into $\mathscr{P}_{A_{1....n}}$
7. **end foreach**

   close $\mathscr{P}_A$ and add $\mathscr{P}_A$ to ∑
8. then begin step 2; until last created business object is added into $\mathscr{P}_B$ and increment the counter
9. **end foreach**
10. **return**

Appendix B

Appendix II

Table B.1 Activities Descriptors and Data Dependencies for the Sofa Production Process.

t_y	Name	Input	Output
t_1	{Check, order}	{Order}	{OrderEval}
t_2	{generate, list, component}	{Order }	LoC
t_3	{evaluate, list, component}	{LoC, order}	{LoC, BankboneComp.Spec., AccessoryComp.Spec.}
t_4	{estimate, price}	{LoC, Order, BackboneComp.Spec., AccessoryComp.Spec.}	{Bill}
t_5	{analyze, backbone, component}	{BackboneComp.Spec}	{RawMaterial}
t_6	{produce, backbone, component}	{BackboneComp.Spec., RawMaterials}	{Backb.Comp.}
t_7	{collect, backbone, component, warehouse}	{backboneComp.Spec}	{Backb.Comp.}
t_8	{deliver,backbone, component}	{Backb.Comp.}	{DeliveryNote}
t_9	{send, accessory, order}	{AccessoryComp.Spec}	
t_{10}	{receive, accessory, order}	{AccessoryComp.Spec}	
t_{11}	{analyze, accessory}	{AccessoryComp.Spec}	{Materials}
t_{12}	{order, material}	{Materials}	{Materials}
t_{13}	{produce, accessory}	{AccessoryComp.Spec., Materials}	{Accessory}
t_{14}	{collect, accessory, warehouse}	{AccessoryComp.Spec., Materials}	{Accessory}
t_{15}	{deliver, accessory}	{Accessory}	DeliveryAccessory
t_{16}	{receive, accessory}	{Accessory}	
t_{17}	{assemble, sofa}	{Backb.Comp., Accessory}	{Sofa}
t_{18}	{package, sofa}	{Sofa}	{Sofa}
t_{19}	{deliver, sofa}	{SofaPackage, DeliveryNote}	{DeliveryNote}
t_{20}	{send, invoice }	{Bill, SofaPackage}	{Invoice, SofaPackage}

Appendix A

Appendix I

Table A.1 The degrees and weights for the nodes in the data model of figure 3.6

Node (n)	Degree	Weight (w(n))
1	3	1
2	2	1
3	2	1
5	2	1
7	2	1
m	1	1
C_1	1	$\frac{1}{2^3-1} + \frac{2^3-2}{2^3-1} \cdot \frac{1}{3} = \frac{3}{7}$
C_2	3	1
C_3	4	1
C_4	4	1

The weight for each of the arc is calculated as:

Note: In this example, C_1 is a split which we call it OR-join connector, C_2, C_3 and C_4 are all AND-join connector. Hence, the weight of the arcs are:

$w(a_1) = w(1) \cdot w(OR) = 1 \cdot \frac{3}{7} = \frac{3}{7}$

$w(a_2) = w(OR) \cdot w(2) = \frac{3}{7} \cdot 1 = \frac{3}{7}$

$w(a_3) = w(OR) \cdot w(3) = \frac{3}{7} \cdot 1 = \frac{3}{7}$

$w(a_4) = w(1) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_5) = w(1) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_6) = w(3) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_7) = w(AND) \cdot w(5) = 1 \cdot 1 = 1$

$w(a_8) = w(5) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_9) = w(3) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_{10}) = w(2) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_{11}) = w(AND) \cdot w(7) = 1 \cdot 1 = 1$

$w(a_{12}) = w(5) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_{13}) = w(7) \cdot w(AND) = 1 \cdot 1 = 1$

$w(a_{14}) = w(AND) \cdot w(m) = 1 \cdot 1 = 1$

In figure 3.6, It can be seen that there are several path from node “1" to node “m", which leads to the several alternative in achieving “m". In this figure, node 1 and node m are connected through the following paths p (p_1, p_2, p_3,...p_n).

$p_1 = \{(a_{14}, a_{13}, a_{11}, a_8, a_7, a_6, a_3, a_1) = (W_{14} \cdot W_{13} \cdot W_{11} \cdot W_8 \cdot W_7 \cdot W_6, W_3 \cdot W_1) = \frac{6}{49}\}$

$p_2 = \{(a_{14}, a_{13}, a_{11}, a_8, a_7, a_4) = (W_{14} \cdot W_{13} \cdot W_{11} \cdot W_8, W_7 \cdot W_4) = 1\}$

$p_3 = \{(a_{14}, a_{13}, a_{11}, a_5 = (W_{14} \cdot W_{13} \cdot W_{11} \cdot W_5) = 1\}$

$p_4 = \{ (a_{14}, a_{12}, a_7, a_6), a_3, a_1) = (W_{14} \cdot W_{12} \cdot W_7 \cdot W_6 \cdot W_3 \cdot W_1) = \frac{1}{9}\}$

$p_5 = \{(a_{14}, a_{12}, a_7, a_4) = (W_{14} \cdot W_{12} \cdot W_7 \cdot W_4) = 1\}$ $p_6 = \{(a_{14}, a_{10}, a_2, a_1) = (W_{14} \cdot W_{10} \cdot W_2 \cdot W_1) = \frac{1}{9}\}$

Appendix B

Glossaries

- **Activity** is an atomic described task or sub-process triggered by some certain event within an organization to complete a certain mission(Kumaran et al., 2008).

- **Task** depicts the unit of work undertaken with an organization, which combines sets of activities using (XOR,OR, AND) gates and step by step executed either by human or machine logically within the process and sub-processes (Cull and Eldabi, 2010), (Kim and Doh, 2009) .

- **Existing Business Functions** are collection of inter-related tasks or processes or operations that support certain enterprise system operation (Klose et al., 2007).

- **Non-existing business functions** are collections of non existing task operations which are applicable to other business functions in any organization(Klose et al., 2007), (Dwivedi and Kulkarni, 2008).

- **Business process** is defined as a set of one or more linked procedures or structured set of activities executed following a predefined order by (potential several) actors (humans, computers and/or machines) in an organizational and technical environment which collectively defines the organizational objectives or business goal (Chinosi and Trombetta, 2012b).

- **Service** is business process or function driven that represents a capability within an enterprise, which provides pieces of self-contained business functionalities(Ese, 2012) for other services and architectural enterprise to implement.(Alahmari et al., 2010)

  Process service is a service that reactively triggers various activities by certain business events being used by business and information service. (Dwivedi and Kulkarni, 2008).

  Business service is service with executable business process, logic, or actions that is built upon the business context of any organization over IT and data service (Sanz et al., 2006).

  Composite Service is service with conjunctional or aggregation of multiple services being internally invoked by customer's unified view.(Dwivedi and Kulkarni, 2008).

  Software/IT Service is a service with functions of software applications derived from a business requirements which can be reused separately by several entities. (Kohlborn et al., 2009b).

  Data Service is a service that is generated from critical data entities such as Order, Claim etc.(Alahmari et al., 2010)

  Utility Service is a service which shares its operational details among various other services as it is commonly set to do, such as payment details, police database system, etc.(Dwivedi and Kulkarni, 2008)

  Infrastructure service is a service that is automatically distributed over various services which provides infrastructure capabilities.(Dwivedi and Kulkarni, 2008).

  Web Service is a service that convert desktop application to web application using web service technology.(Dwivedi and Kulkarni, 2008)

Collaborative Service is a service that is exchangeable between inter-organizations based on company agreed terms. (Dwivedi and Kulkarni, 2008)

- **Service coupling** is the degree of complexity of information element within a service, which maintains certain relationship that minimizes dependencies between the information elements.(Ese, 2012)

- **Service cohesion** is the degree of inter-relationship between different services which depends on the numbers of operations (Huergo et al., 2014).

- **Service granularity** is the defined as the scope of functionality or size of operations that individual services implement to provide the right granular level (Ese, 2012).

- **Right business service** is right identified service with low granularity, low coupling and high cohesion at application-oriented, business-related level of granularity using widely applied standards (Boerner and Goeken, 2009), (Legner and Heutschi, 2007).

- **Process orchestration** is notion in which services (processes and execution constraints) are invoked with a detailed view in explicit order.(Weske, 2009)
- **Closed information environment** is any organization operating in an escalated security or non-collaborative posture, using and running its own process service within, which reduces the time spent on series of fragments of information, increasing efficiency and effectiveness in slving problems.(Goh and Hooper, 2009)

- **Open information environment** is any organization with existing process models,real-time application scheduled to interrelate its tasks according to a service calls or specific algorithm within a collaborative organizations and sub-organizations.(Deng et al., 1999)

- **Workflow** is the flow of business cases that specific, execute, monitor and coordinate the flow of processesor tasks within or outside an organization, for example is the mortgage application, handling loan application, and registration of new clients.(Ellis, 2005)

- **WFM system** is system enactment of workflow models or process which give capabilities for the end-users to ulter the business process during its execution run-time(Reijers, 2003) .

- **BPM system** is system that defines, manage, evaluate and enact business processes through the execution of software which execute the process model to represent the business process logic(Weske, 2009).

- **Verification** is the determining the basic characteristic features and behaviour which a certain process model exhibit(Wynn et al., 2009), workflow systems such as (Yet Another Workflow Language)YAWL provide verification capabilities(van der Aalst, 2013)

- **Validation** is checking for the correctness of a process model or service model.

- **SOA** is a modern information technology (IT) approach where data, logic functionality and software systems are (re)implemented as a set of robust and interoperable services.(Alahmari et al., 2010)

- **Process configuration** is the customization of configurable models to a specific solution, applied over various variation point (Alves et al., 2013)

- **Service-Oriented computing** (SOC) is the design paradigm for distributed computing platforms that utilizes services as compositional and fundamental elements used to build software and application system solutions (Papazoglou, 2003), (Papazoglou and Heuvel, 2004).

- **Application Logic** is coordination of business logic and infrastructure components according to the business requirement (Wang and Wang, 2006).

- **Enterprise logic** refer to ideological underpinnings on organizational activities which are built on the need for shared business values and creativities(Jones and Crompton, 2009).

- **Business Logic** is the coordination of process routing, operational constraints, exception handling and business strategy for appropriate business situation (Wang and Wang, 2006).

  Source code is an executable description of computer information or program compiled to perform certain executable duty or form (Binkley, 2007).

- **Process model** is a foundational representation of a business process(Weske, 2009) which captures the different in which a process instance is handled (van der Aalst, 2013).