A Process Modelling Framework Based on Point Interval Temporal Logic with an Application to Modelling Patient Flows

Chishti, I.

UNIVERSITY OF WESTMINSTER

School of Computer Science & Engineering
(Health and Social Care Modelling Group)

A PROCESS MODELLING FRAMEWORK BASED ON POINT INTERVAL TEMPORAL LOGIC WITH AN APPLICATION TO MODELLING PATIENT FLOWS

By Irfan Chishti

A thesis submitted in fulfilment of the requirements for the degree of Doctoral of Philosophy

March 2019

Dedication

My sincere thanks to my beloved parents for their continued support and patience during this PhD journey. Also, am grateful to my wife for standing by me to encourage and motivate to complete this thesis. Their pride and joy is seeable and cannot wait till my graduation. Lastly but surely, a big thanks to my twin boys, Hassan and Hussainwho needed me to spend time with them so now I would have ample for them to fo things that were put on hold.

Acknowledgement

I am extremely grateful to my director of studies Dr Artie Basukoski for his valuable support, guidance and constructive feedback. I am also thankful to Professor Thierry Chaussalet for his expert advice throughout the duration of my research journey, as and when required.

I sincerely acknowledge the support of Dr Alexandra Psarrou, Dr Andrzej Tarczynski and graduate school in providing the opportunity to disseminate the research work by attending conferences during this project. Also, the peers at the University have been very supportive and friendly and a big thanks to all of them.

Declaration

I, IRFAN CHISHTI, declare that the work is original and have not submitted to any other degree other than that studied at this University.

Signature: ______________________

Publications During the Study

1. Chishti I., Basukoski A., Chaussalet T., Beeknoo N. (2019) Transformation of UML Activity Diagram for Enhanced Reasoning. In: Arai K., Bhatia R., Kapoor S. (eds) Proceedings of the Future Technologies Conference (FTC) 2018. FTC 2018. Advances in Intelligent Systems and Computing, vol 881. Springer, Cham.
2. Chishti, I., Basukoski, A. and Chaussalet, T.J. (2018). Modeling Patient Flows: A Temporal Logic Approach. Journal On Computing, 6 (2). ISSN 2251-3043.
3. Chishti, I., Basukoski, A. and Chaussalet, T.J. (2017). Modeling and Optimizing Patient Flows. In: 8th Annual International Conference on ICT: Big Data, Cloud & Security, 21-22 August 2017, Singapore *(Best Paper award)*.
4. Chishti, I., Chausselat, T. J., Basukoski, A., (2017). Patient Flow Modelling and Scheduling using Point Interval Logic (poster) at Informatics for Health 24th April 2017, Manchester, UK.
5. Chishti, I., Basukoski, A., & Chaussalet, T. J. (2016). Business Process Modelling based on formal temporal theory with an application to hospital patient flows. In *8th IMA International Conference on Quantitative Modelling in the Management of Health and Social Care*. Institute of Mathematics and its Applications.

Abstract

This thesis considers an application of a temporal theory to describe and model the patient journey in the hospital accident and emergency (A&E) department. The aim is to introduce a generic but dynamic method applied to any setting, including healthcare. Constructing a consistent process model can be instrumental in streamlining healthcare issues. Current process modelling techniques used in healthcare such as flowcharts, unified modelling language activity diagram (UML AD), and business process modelling notation (BPMN) are intuitive and imprecise. They cannot fully capture the complexities of the types of activities and the full extent of temporal constraints to an extent where one could reason about the flows. Formal approaches such as Petri have also been reviewed to investigate their applicability to the healthcare domain to model processes.

Additionally, to schedule patient flows, current modelling standards do not offer any formal mechanism, so healthcare relies on critical path method (CPM) and program evaluation review technique (PERT), that also have limitations, i.e. finish-start barrier. It is imperative to specify the temporal constraints between the start and/or end of a process, e.g., the beginning of a process A precedes the start (or end) of a process B. However, these approaches failed to provide us with a mechanism for handling these temporal situations. If provided, a formal representation can assist in effective knowledge representation and quality enhancement concerning a process. Also, it would help in uncovering complexities of a system and assist in modelling it in a consistent way which is not possible with the existing modelling techniques.

The above issues are addressed in this thesis by proposing a framework that would provide a knowledge base to model patient flows for accurate representation based on point interval temporal logic (PITL) that treats point and interval as primitives. These objects would constitute the knowledge base for the formal description of a system. With the aid of the inference mechanism of the temporal theory presented here, exhaustive temporal constraints derived from the proposed axiomatic system’ components serves as a knowledge base.

The proposed methodological framework would adopt a model-theoretic approach in which a theory is developed and considered as a model while the corresponding instance is considered as its application. Using this approach would assist in identifying core components of the system and their precise operation representing a real-life domain

deemed suitable to the process modelling issues specified in this thesis. Thus, I have evaluated the modelling standards for their most-used terminologies and constructs to identify their key components. It will also assist in the generalisation of the critical terms (of process modelling standards) based on their ontology. A set of generalised terms proposed would serve as an enumeration of the theory and subsume the core modelling elements of the process modelling standards. The catalogue presents a knowledge base for the business and healthcare domains, and its components are formally defined (semantics). Furthermore, a resolution theorem-proof is used to show the structural features of the theory (model) to establish it is sound and complete.

After establishing that the theory is sound and complete, the next step is to provide the instantiation of the theory. This is achieved by mapping the core components of the theory to their corresponding instances. Additionally, a formal graphical tool termed as point graph (PG) is used to visualise the cases of the proposed axiomatic system. PG facilitates in modelling, and scheduling patient flows and enables analysing existing models for possible inaccuracies and inconsistencies supported by a reasoning mechanism based on PITL. Following that, a transformation is developed to map the core modelling components of the standards into the extended PG (PG*) based on the semantics presented by the axiomatic system.

A real-life case (from the King's College hospital accident and emergency (A&E) department's trauma patient pathway) is considered to validate the framework. It is divided into three patient flows to depict the journey of a patient with significant trauma, arriving at A&E, undergoing a procedure and subsequently discharged. Their staff relied upon the UML-AD and BPMN to model the patient flows. An evaluation of their representation is presented to show the shortfalls of the modelling standards to model patient flows. The last step is to model these patient flows using the developed approach, which is supported by enhanced reasoning and scheduling.

List of Figures

List of Tables

List of Appendices

Contents

CHAPTER 1 INTRODUCTION

Organisations employ a collection of activities that follow described procedures to achieve their vision and objectives. The set milestones accomplished by organising and structuring several tasks and corresponding flow representing a well-defined process. A well-defined concept of the business process (BP) incorporates all the involved activities representing a process model exhibiting the temporal flow between individual work elements [Scholz-Reiter and Stickel, 2012] to facilitate organisational design and analysis. In addition, the flow between the tasks determines their relationships with other linked processes (internal or external).

There are authors who provided a strong emphasis on the designing organisational BPs with a logical basis to facilitate correct modelling, analysis and transformation for better decision making [Blyth, 1995], [Tsalgatidou and Junginger, 1995], [Hansen, 1994] and [Curtis et al., 1992]. In addition, a process model ought to express high and low-level detail that may represent the desired features of an organisation [Jablonski and Bussler, 1996]. Out of many, two aspects considered the success of a business process model, which are consistent representation of the required operations and its decision-making capabilities.

These aspects are highly desirable for any industry in general but healthcare especially. Because healthcare is facing an unprecedented level of change, affecting the service delivered to diverse patient needs. For example, modelling a patient flow of highly sensitive nature such as an intensive care unit is not only tedious but also tremendously challenging to plan and schedule [Adlassnig, 2009]. The service delivery to patients follows specified paths known as patient flows or patient journeys.

Due to the complex nature of healthcare sector, face hardships in representing patient flows utilising available communication platforms. Primarily, these platforms are graph based simulating the effect of interaction and interrelationship of patient flow as a whole including its sub-parts. These tools may also use to report variations in the structure of the patient journey.

Mainly, the healthcare sector reliant upon intuitive flowchart-based graphical representations to show convoluted hospital activities. The model constructed utilising such techniques attempting to relay the communication between departments and personnel cannot comprehend the overwhelming burden of the sub-activities to correctly reason and represent [Gunal and Pidd, 2010].

Industry whether its health sector or any other commercial organisation, heavily reliant on modelling tools such as business process modelling notation (BPMN) and unified modelling language activity diagram (UML-AD) to represent patient flows for care service delivery. These intuitive business process modelling techniques require adaptation by healthcare professionals due to their complexity in representing clinical care processes as human lives could be at risk. However, these techniques belong to two different domains for one specific reason that is process modelling.

Their intuitiveness lies in their standard documentation using a wide variety of terminologies and graphical constructs. These artefacts' represent informal meaning (ontology) bound to be interpreted differently in the workplace. As the ontology used is not formal which comes in several forms such as lexicons, thesaurus or logic etc. And, these forms can provide the standardisation of the terminologies utilised in the business process modelling domain. Thus, if formal definitions of the concepts used in a domain provided then one can express their concise description, better understanding and unambiguous representation. Additionally, it is essential to have a communication mechanism that could reason and represent the knowledge consistently about BPs at all abstraction levels (High and low).

The problem above addressed by comprehending the terminologies used in both the techniques and associate the most commonly used modelling terms with some lexicons based on logic. This method can facilitate in providing the formal semantics of selected general terminology that would ease the path of representing processes in a consistent way. The advantage is that logics are very expressive for modelling complex behaviours such as healthcare patient pathways. Without such formal definitions, rigorous and mechanical verification of systems will be impossible.

Furthermore, these techniques attempt to present low-level information (high-level abstraction) through their models. Modellers tried to draw high-level details with very low or negligible success rate. However, it is possible to breakdown the system activities with associated temporal information into smaller parts. It would help in determining the structure and flow of the sub-parts of a business process and patient flow. Optimum arrangements of the coordinated tasks of a process can have a significant impact on determining the efficiency of the new structures [Orman, 1996] that is only possible if supported by the inference mechanism.

With the assistance of an inference mechanism, (provided by the temporal logic) one can derive new knowledge (from exhaustive temporal relationships between the individual piece of work elements of the linked processes) and establish the correct structure for better planning and scheduling. Temporal relationships express the different behaviour (flow) between the linked activities, hence a class of temporal logic considered integral in providing an exhaustive set of temporal relations to achieve not only the optimum flow but also enables the construction of a consistent model. In this way, models constructed are more detailed and easier to understand.

Thus, a comprehensive methodological approach would be beneficial that provide a set of general terminologies subsuming the most often used terminologies of commercial modelling techniques and subsequently formally describing them to support functional, structural and behavioural levels of abstraction. Knowledge domains such as business and healthcare modelling may benefit from this approach for their knowledge design, its representation and management.

1.1 Motivation

Business processes (BPs) are critical for organisations to execute activities and tasks that create value. Business values considered as the product of profitability, performance, and tightly coupled with the process (re)design and its execution. In general, organisations have made a great effort to lower the cost of improved products and services. They have also taken initiatives in time reduction of marketing efforts and customising the products and services with the time limitations to strengthen their relationships with customers and increase the satisfaction of its customer with maximising its profits. However, the healthcare sector like any other domain has similar goals to achieve, but their most important goal is its patient satisfaction for services provided in a time and resource bound environment.

These objectives push them into continuously improving their processes to provide better services. It shows the importance of this topic that could give an aid to design, structure, and control the BPs to achieve desired goals efficiently and flexibly. To describe and structure a BP, one needs to examine the ontology of the terms used in modelling a system comprised of several components. Therefore, a distinct description of the concepts involved plays a pivotal role in constructing a well-defined business process model.

The above discussion leads us to the need for understanding the concept of a business process and its utilisation in real-life. There are several varying BP definitions reported in the literature related to business process design and its modelling. Still, there is a vacuum for a profound business process definition as most of these definitions are isomorphic. Some researchers such as [Hammer & Champy, 1993] and [Davenport, 1993] defined BP by showing activities to achieve a goal. But failed to identify tits structural and configuration feature to distinguish between breakable and unbreakable actions and also the importance of occurrence of events neglected to show the flow in accomplishing a purpose or delivering a milestone.

To support the viewpoint, I will discuss the pin factory process example of [Smith, 1776], where he used the term ‘task’ to identify breakable activities. Various tasks performed for pin manufacturing including drawing out a wire, straightening, cutting, pointing and so on. He emphasised on the importance of the temporal flow associated with these tasks during their execution, which influenced by the occurrences of certain events such as the strike on a specific day or machine malfunctioning. Hence, a clear description of the business process and its flow embodied with its temporal association would enable modellers and modelling tools for their precise representation

Management and computer science researchers describe the structure and order of the components of BP to suit their needs based on the available definitions. Their interest in the modelling business processes multiplied over the last few decades to analyse, manage, represent and reason knowledge about an organisation. Industry developed different modelling techniques and tools to meet the varied needs of the different domains. For example, unified modelling language activity diagram (UML-AD) developed for the technical domain users and business process modelling notation (BPMN) developed for business modellers. These both standards document comprised of a wide variety of terminologies and constructs to represent the behaviour of either a system or a model.

Additionally, it is reported that both the standards borrowed concepts from Petri net [Wohed, 2004] and [Wohed et al., 2006], but failed to provide the formal semantics relevant to business process modelling domain. Though their claim for the semantics for the notation is provided in their corresponding metamodels, again their claim is not justified (providing only diagrammatic constructs). Because, both the standards have no formal semantics and no validation in real life [Van der Aalst, 2004a, b]. Moreover, the Petri net offers formal semantics but not designed for representing business processes. Also, modellers try to avoid using it due to its complexity and consider it unsuitable for business process modelling. A variant of Petri net, i.e. Time Petri net used to model systems by modelling temporal aspects without providing appropriate enumeration [Berthomieu & Diaz 1991].

A suitable enumeration relevant to business process associated with distinct temporal objects can be useful for the business process shaping. Therefore, process orientation based on the temporal description at all organisation levels to model their BPs or patient flows assist in effective knowledge representation, reasoning and quality enhancement of the services offered by the healthcare sector. For example, in the clinical process modelling, time perspective has been widely investigated [Combi and Gambini, 2009] and [Combi et al., 2012].

There is also an increase in demand to have effectual and improved quality of models, it is required to have a knowledge base that can assist in constructing a correct model to represent a patient flow [Edward 2005], [Newell 1982]. However, in the healthcare domain, such a knowledge base is missing to model a sound system [Clarkson et al. 2004]. A system refers to a model, which depicts a correct, i.e. consistent, representation of the processes involved. In addition, if these activities wrapped up with the extended qualitative and quantitative (if available) temporal information then I can address the challenges faced by healthcare at present and more importantly in the future.

Unfortunately, the wide variety of the notational support provided by both the informal modelling standards, i.e., UML-AD and BPMN, is not enough to address the practitioner's issues faced in real life in regards to utilisation of a general temporal theory for business process design and execution. Furthermore, these modelling tools' intuitiveness produces inconsistent models and failures occur noticed at the execution level that may result in financial problems to the organisations. The metamodel provided by the current modelling techniques are poorly defined accompanied by the graphical constructs. Thus, these standards present a considerable effect on standardising the business process design and its modelling.

To see the shortcomings of these techniques, I considered a healthcare sector case study based on hospital patient flows of King's College Hospital Trust, a national health service (NHS) foundation trust. The reason behind this case study to understand the representation of patient flows, involved sub-components by the domain experts at the hospital discussed in detail in the following subsection.

1.1.1 Patient Flows at King's College Hospital: Key Findings

The patient flow at King's College Hospital accident and emergency (A&E) department like other NHS Foundation Trusts considered as the journey of the patients through the hospital requiring quality care services bounded by time and resources to move them around (admission to discharge). The activities and corresponding flow diagrammatically represent the whole or part of the operation of a department. A series of meetings with the concerned staff at King’s College hospital revealed that their activities are modelled graphically utilizing the current business process modelling standards.

They adopted the available concepts of UML-AD and BPMN such as activity, action, task, process, sub-process, and flow, etc. for the modelling of patient flows. Their knowledge about these concepts based on the intuitive description, vaguely aligned with the corresponding concepts of healthcare in general and especially in hospital settings. The available information represented graphically using either of the modelling tools assist the modelling staff to label the available concepts with the respective names and description. But cannot address the issues of extended qualitative representation, quantitative representation to identify and manage the variability (interchangeability) within process activities that are resource-bounded to help with decision making. For example, a patient can move from one pathway to another associated with either quantitative time, i.e., a specific start and end time (if available), or with the help of inference made based on extended qualitative occurrences indicating a possible change in the original path. The variation occurs due to either human error or patient health condition changed.

1.1.1.1 Challenges

Patients move through various sections, i.e., registration, triage, consultants, Diagnostics and Ward utilising several resources at the King’s College hospital accident and emergency (A&E) department based on the type of the care required. The discussion I had with the hospital staff revealed the issue of uncertainty in resource allocation (staff) and the time required to deliver a quality care service to the patients attending the accident and emergency department. The amount of patients seen every day at different times fluctuates that make the patient flow modelling difficult to depict the correct scenario. For example, patient influx over the weekend and during the shift change at different times, i.e., 5 pm or 11 pm etc. makes the overcrowding to reach a high level. In addition, the patient’s number increases during Christmas and other seasonal events, but my focus is not to address such an issue. The scope of this research requires equipping domain experts who are involved in modelling of patient flows to express the correct behaviour of the system utilising efficiently the available resources (staff) in a time-restrained environment.

To determine the complexity of the healthcare activities and sub-activities, I have discussed a complex example to show the variability in the patient flow with respect to time and emphasised upon the need to use the extended qualitative and quantitative temporal information to plan effectively in resource (staff, equipment, time etc.) utilisation.

Example: A complex pathway (process) considered here presenting patients who brought in by someone with trauma. The flow starts with the registration of the trauma patient. After completing the registration process, the patient requested to wait in the waiting area. Waiting times varies due to the staff availability at the time of arrival of the patient. A triage nurse assesses the severity of the problem. The severity of trauma classified into three levels, minor, minor-major and major, the example noted here will discuss only patients with minor-major severity.

There are cases where patients leave without being seen due to the excessive wait times therefore, to be within the scope of this research, I am considering only patients flowing through the process. Immediately after triage, consultants take over the patient and examine the patient condition to provide the best suitable care service needed. Consultant examining a patient is a complex step which requires decision making that may involve admitting the patient to the hospital surgical ward and further divided into sub-activities (to assess the patient history and diagnostics).

Consultant requests diagnostics that are further sub-divided into sub-activities (blood tests and X-ray, MRI, CT-Scan) to assist consultant in providing patients with a prognosis. These activities become strenuous due to restraints of time attached and resource (man and machine) available. Upon evaluating the results received from the diagnostics team, either consultant makes a decision to prescribe the required medication (if required) with a discharge note or due to the change in severity and requested a move to the high dependency unit (HDU) at the hospital.

Each ward managed independently to make necessary decisions with respect to their capacity (number of beds) and resources (including staff and time). The hospital’s policy ensures even distribution of resources between the patients based on their needs, i.e., major to minor. Thus, effective time and resource utilisation assists in managing overcrowding at A&E and surgical wards.

Discussion & Critique: During discussions, some of the issues raised by the modelling staff at the King’s College Hospital are:

- Identifying activities (atomic and composite) and their relationships for better understanding and consistent representation
- The utilisation of their timely occurrences for effective patient flow representation

- Require assistance in better planning and decision making using the graphical representation of the patient flow
- Optimise the coordinated activities and sub-activities to meet the NHS set targets, i.e., 4-hour waiting time at the accident and emergency department
- The current modelling standards incapable of modelling activities and sub-activities ignoring the vital temporal information, i.e., enhanced qualitative/quantitative.

The above points require attention that might be achieved with the help of a rigorous examination of the data associated with patient flow design to achieve the desired goals. King's College Hospital utilising graphical representation to depict patient flows but failed to display the process activities with associated information of time and resources. A patient flow constitutes a set of coordinated activities and sub-activities that needs managing effectively to achieve the desired goal of satisfying both the patients and NHS in delivering quality service within specified time targets.

The modelling tools used to express the current behaviour of the system is inefficient and has no capability to support delivering a quality service. Because the tools used have no facility to incorporates a comprehensive temporal theory, verification and validation mechanism to determine the constructed models are consistent or not. In addition, planning and scheduling activities and sub-activities will ensure better decision making by the concerned staff at the hospital. An approach is required to describe activities and sub-activities to exhibit enhanced qualitative and quantitative temporal information for better resource management. Furthermore, it would facilitate correct modelling and easing the pressure on the staff by monitoring and controlling the operations efficiently (avoiding overcrowding).

However, the current modelling tools have no facility to model the desired activities associated with both quantitative and enhanced qualitative time. It is the crucial information and possible to implement using a point interval temporal theory that could assist in better decision-making and improved scheduling. Hence, the following points are required for modelling activities:

i. To have an effective and efficient model (precisely describing the activities and sub-activities) using value-added processes within the shortest possible time and

ii. To make improved decisions. Consequently, process description and modelling have become a critical strategic resource for any enterprise including healthcare.

The research motivation in this section set the research objectives that are discussed in the following sub-section.

1.2 Research Objectives

The industry relies on business process modelling standards (UML-AD and BPMN) to describe the intuitive structure of their constructs. These standards document a broad set of modelling terms and constructs to shape the respective business processes to capture different features but their representation is ambiguous and vague. Although, fewer terminologies specifically for business process modelling (BPM) with formal semantics make the system conceptually easier for the users to understand and utilise. It also builds trust and reduces the amount of effort needed to verify the model.

To address such issues including the ones noted in previous sections, it has become a requirement to provide distinct ontology (formal semantics) for the business process (design) and modelling (execution). And achieved by either revising or extending the current modelling standards by examining their terminologies and corresponding semantics for possible formalisation [Thomas and Fellmann, 2009]. It is the cumbersome procedure to provide a formal description for all the notational elements, therefore a selection of notational elements would be a step forward towards logical foundation (missing) based on their frequent utilisation by the industry. Following this procedure would assist in not only providing standardisation of the chosen process modelling languages but also facilitating the verification and validation of the constructed models to determine their correctness.

Moreover, the formal semantics used for modelling business processes can be of deductive and normative type [Boley et al., 2007]. Normative type mainly facilities the structure of the defined components. Deductive type facilitates in inferring new facts from the existent knowledge. For example, two processes X and Y constitute a process model, where X serves as a sub-process of Y to achieve an output Z. It implies that both X and Y serves towards in fulfilment of output Z. This inference would assist in answering specific queries such as patient or customer satisfaction relevant to a particular part of the process model. Thus a method which combines both types would be beneficial for the precise model design and execution.

In this thesis, I would develop a systematic approach to provide a logical base to remove ambiguity from the representation of business processes and their associated sub-components (constructed in UML-AD and BPMN) to express temporal aspects. This investigation will fill the existing gap requiring a grounding for BPM discipline. Thus, the aim of this research presented in the following sub-section.

1.2.1 Research Questions

The discussion so far in the research objectives identifies the need to construct a generic framework to model business process representing its temporal aspects that can also be utilised by the healthcare sector to meet their needs of precise modelling. The following questions stem from the discussion are as follows:

Question 1: Industry relies on different modelling approaches for modelling business processes. For this research, I would be considering the frequently used business process processing modelling techniques accepted as standards such as unified modelling language activity diagram (UML AD) and business process modelling notation (BPMN). They both use a variety of modelling terminologies aligned with graph-based constructs to represent a business process, but lack to build a precise model due to their intuitiveness. Other shortcomings include having too many modelling constructs in their standard documentation for modelling a process. It makes modeller confused in making a specific choice while modelling a business process that leaves many unused. Also, these constructs have no precise semantics (structure) provided in their standard documentation (metamodel). Due to these failings, tools considered burdened (increasing the redundancy) and semantical errors. Additionally, the results produced by them are ambiguous and not correctly exhibiting the temporal aspect.

Hence, a formal ontology describing the precise semantics of the (most used) terminologies play a vital role in representing a wide variety of operations consistently within an organisation. It would be a tiresome job to provide semantics to the extensive set of modelling terms used by the modelling standards. Thus, it is required to review these paradigms' the most often used modelling terms for better understanding. The desired solution of the problem would comprise of an enumeration of the core business process modelling terms based on a well-established logic in the literature, i.e. temporal logic. It will aid in providing precise semantics by formally defining the chosen terminologies associated with temporal objects to establish their sequence, order and the attached duration.

Question 2: Stemming from question 1, a review of different temporal theories considered vital because each I theory constitutes either point or interval or both. Therefore, I would examine the different temporal theories. The process will include analysing the worldly objects represented by a particular time theory for their suitability to this research. For example, a class of temporal theory presented by [McDermott 1982] focus on time point in describing a process. Where, [Allen 1983] use the interval to specify a business process, action and event with duration and neglected the existence of a time point. Due to their isolated use of the temporal objects in defining the process, action and event cause problems for instantaneous and non-instantaneous activities modelling.

Eventually, this investigation would result in selecting a suitable temporal theory to identify the temporal objects serve as lexicons for the domain of modelling a business process (BP). Subsequently, the lexicons associated with the most often terms used by the modelling standards formally defined a business process and its sub-parts. In addition, the intended framework would provide a mechanism to verify the business process for its correctness.

Question 3a: A simple, easy to use and understand graphical notation required, thus, an investigation needed to provide a formal but simple tool. That would use the defined terminologies here in this thesis aligned with the commercial terminologies and constructs modelling the business process and its sub-parts showing the authenticity of the framework proposed here ranging from novice to experts. In addition, the tool would assist in analysing the existing models constructed using UML-AD and BPMN for their correctness. Besides all this, a transformation performed of UML-AD and BPMN most often used terminologies and constructs to the developed formal approach.

Question 3b: A case study conducted to analyse existing patient flows, i.e. processes, of King's College Hospital NHS Foundation Trust accident and emergency (A&E) department modelled in UML-AD or BPMN. I would discuss the framework capabilities in addressing the issues faced by the accident and emergency department to model patient flows to establish the developed approach suitability and its novelty.

1.3 Contributions

Scientific knowledge design considered as contributions towards the knowledge and assessed based on its novelty, generality, and significance [Von Alan et al., 2004]. For this research, I would consider designing scientific knowledge that constitutes the artefacts required for devising a knowledge base. Henceforth, my contributions aligned with all three-assessment criterion achieving a)novelty through a distinct solution to the problems faced by the industry and b) the scientific knowledge design is general enough applied to any real-life domain with similar problems. Moreover, the third criterion furnished by considering a case from the King's College Hospital accident and emergency department to model their patient flows utilising the knowledge base developed. Which shows the significance of the approach that may result in the reduction of patient waiting times, and improving their care service delivery time. A systematic approach devised following the steps given below.

- Conducting an empirical study on applied semantics of the business process modelling (BPM) standards and Petri net. A comprehensive review of the (informal) modelling standards, i.e., UML-Ad and BPMN would determine the problems faced by the industry while using them. Moreover, I have reviewed Petri net which has a long presence (because of its formal underpinning) in the literature and used for various reasons including system modelling. Similarly, it has been adopted for the transformation of informal modelling techniques to provide a unique ontology for their modelling elements. However, its structure found to be irrelevant and cumbersome to business process modelling domain.
- Review of UML-AD and BPMN would facilitate in achieving the milestones by developing a framework comprised of two phases. First one would provide an enumeration of temporal objects (based on a general temporal theory) representing lexicons with logical meanings, i.e. ontology, defining them to make provision for formal semantics.
- The analysis of both the business process modelling standards would help me to identify the core modelling artefacts based on their utilisation. It would assist in the development of phase II of the framework devising the axiomatic system based on model-theoretic approach. The axiomatic system would introduce enumeration based on general terminology set. Subsequently, they are formally defined subsuming both the modelling standards most often used terminologies. In addition, a mechanism is provided for their verification and validation.
- A precise but easy to use graphical approach provided to model the axiomatic system. That would facilitate the transformation of most UML-AD and BPMN most often used terms and constructs to a formal approach authenticating the method developed here.

- After that, a transformation of the BPMN and UML AD key constructs to the axiomatic system provided for accurate process modelling. Moreover, the transformation would unify the industry standards due to their tenuous nature in their representation, as evident from the literature [White, 2004].

Besides, this approach will avert the burden of redundant terms used by the current business process modelling standards and assists in answering the research questions highlighting the contributions towards the knowledge. As part of the contribution, I have extended the point interval logic (PITL) of [Zaidi, 1999] by providing interval-point formalism and an added set of temporal relations (apart from interval-interval, point-point and point-interval temporal relations_ used as constraints for providing a consistent flow within a process. Furthermore, I have also extended the formal graphical tool point graph (PG) of [Zaifi, 1999] by adding binary operands for displaying concurrency within a process. This contribution to the knowledge has also laid a path towards the following additional contributions to the knowledge

1.3.1 Enumeration

Both UML-AD and BPMN utilise different terminologies to display intuitively a process, its sub-parts and their flow. The is known to the knowledge will include identification of the terms used frequently and similar in their functionality. For example, UML-AD uses the term ‘action’ representing an activity (atomic), and BPMN utilises a term called ‘task’ to represent the same. Other most used nomenclatures are ‘activity’ used in UML-AD and ‘process’ and sub-process’ by BPMN representing composite activities. Additionally, composite activities expressed their boundaries by ‘initial node’ and ‘final node’ in UML-AD, and ‘start event’ and ‘end event’. Importantly, these most used terminologies and constructs’ precise structure is not available in both techniques’ standard documentation.

In addition, their flow determines intuitive process design that failed to express precise occurrences of atomic activities along with other involved activities (whether atomic or composite). Because, without precisely defining atomic activities boundaries, modellers lack in expressing either they occur at the boundary of interacting activities, or during other occurring activities within a process or occurring simultaneously along with other atomic activities. The precise design and its temporal information (both qualitative and quantitative) have an enormous effect on the overall process design and expressiveness.

Hence, without an explicit catalogue (enumeration), a modelling technique or method unable to specify the core elements of a business process required for its modelling. Therefore, it is vital to provide general but distinct terms to construct a business process model (explicitly), considering as a contribution to the knowledge. That is possible with the support of point interval temporal logic of [Zaidi, 1999], which is extended (contribution to the knowledge) in providing a general set of artefacts with precise structure. Moreover, temporal inference mechanism could facilitate in representing a consistent flow of a business process.

The enumeration presented here in this thesis comprised of general terms based on temporal objects and associated with the core modelling artefacts of both modelling standards. Thereafter, these artefacts are formally defined (semantics) using first-order logic based on the model-theoretic approach to representing the precise structure of these artefacts. This step would establish the generality of the framework for modelling business processes offering a knowledge base.

Besides, these modelling standards do not provide any verification and validation mechanism to authenticate the constructed models to report any inconsistency. Thus, this research would contribute to the knowledge by developing a method that not only facilitates the precise design and structure of the business process for its correct representation but also verifiable. In this way, I can improve understanding, functionality and can help to design correct process models/systems. Moreover, the systematics approach developed here provides a solution for a better plan and improved scheduling.

1.3.2 Transformation

To perform the transformation, it is important that the informal modelling standards facilitated with formal semantics. The formalised semantics provided for the set of generalised artefacts in this thesis requires further a precise and straightforward graphical representation (answer to question 3a) for the transformation purpose. Therefore, a formal graphical tool is known as point graph (PG) presented by [Zaidi 1999] chosen and extended (contribution to the knowledge), notated as PG*. Reasons to utilise PG* are threefold:

i. It is graphical, precise and easy to use.
ii. It has a foundation in point interval temporal logic that treats both point and interval as primitives representing the precise structure of generalised terms.
iii. It offers an abundance of analysis techniques to check the correctness of the constructed process models.

Additionally, I would be able to perform a transformation (contribution to the knowledge) of both the business process modelling standards most often used artefacts to PG*. It is important to note that the proposed framework has not only the capability to analyse the models constructed using business process modelling standards but also can serve as a platform-independent representational tool. In addition, it would unify both the modelling standards. Question 3b is answered by applying the above contribution to the knowledge to King's College Hospital National Health Service (NHS) Foundation Trust' accident and emergency (A&E) department trauma patient flow modelling. The patient flow modelled in UML-AD and BPMN would be transformed into PG* to identify the issues concerning their consistency.

Nevertheless, many modelling techniques available whether they apply or not to business process modelling in general or patient flow modelling in specific are beyond the scope of this project. Hence, I can state that as the time of writing this thesis, no framework is available to unify the business process modelling standards.

1.4 Research Approach and Thesis Overview

To achieve the research objectives, I have made a choice of using 'constructive research method shown in figure 1.1 widely adopted by the researchers of computer science and healthcare sectors [Kasanen and Lukka, 1993], and [Shaw, 2001]. Because, this method attempts to seek solutions associated with theory and its subsequent implementation in real-life [Lassenius et al., 2001].

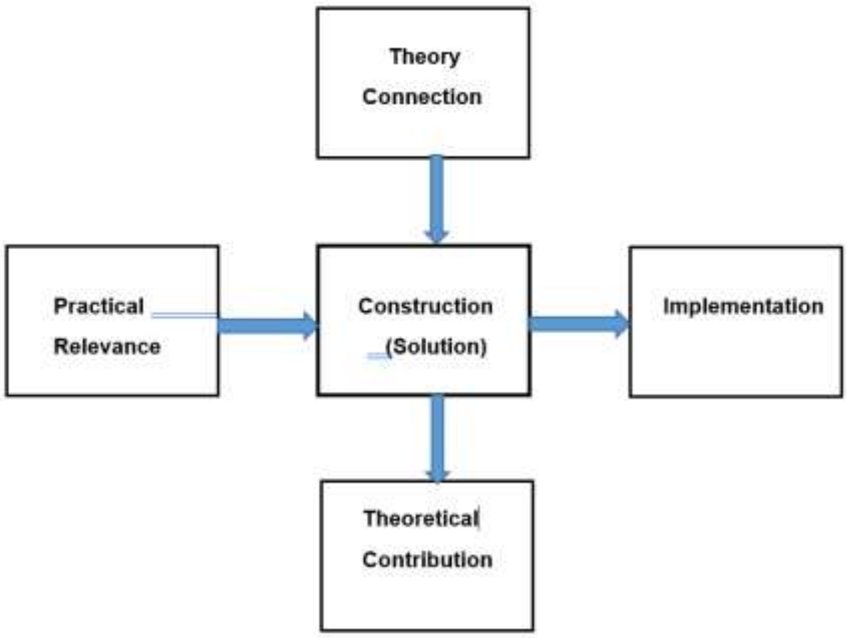


Figure 1.1 Components of Constructive Method

However, in constructive research method ‘theory’ refers to either the development of an innovative method to identify and understand the actual problem of the industry or develop a solution that works both theoretically and practically[Lukka, 2003]. In this thesis, the theory development helps in the scientific knowledge design deemed necessary to build a consistent business process. Furthermore, it provides practical value for its real-life usage. Therefore, to achieve this, I will conduct a comparative (empirical) analysis of the literature to discuss the need for a general framework to fill the gap.

Thesis structure organised as chapter 2 will provide a discussion on the business process modelling topic emphasising its conceptual and temporal aspects in representing healthcare processes (hospital patient flows). Chapter 3 present a comprehensive review of the modelling techniques (formal and informal), but the focus will be on the informal business process modelling standards (UML-AD and BPMN) and formal method Petri net. Chapter 4 will provide a discussion on the identification of core modelling terms used by UML-AD and BPMN. Chapter 5 will provides phase I of the framework development by analysing different classes of temporal logic to model processes and choose a suitable category to meet the research objectives of this thesis.

Chapter 6 provides the phase II of the framework development presenting the axiomatic system based on modelling theoretic approach. Chapter 7 will describe process enactment to simulate the axiomatic system developed. Chapter 8 includes a transformation of business process modelling standards to the axiomatic system based extended graph tool PG*. Chapter 9 presents a case study of the King’s College Hospital accident and emergency department patient flows presented. Trauma patient flow scenario considered for this thesis, constructed in UML-AD and BPMN and transformed them into the approach developed here in this research to remove any correctness issue, and schedule and optimise the patient flows.

1.5 Summary

This chapter discusses the need for enterprises to model their business processes correctly. Organisations differ in their structure, needs and requirements but they have the common goal of representing their operations in a non-technical way to meet all the stakeholders’ needs. Because most of them are not of aware technical jargon, and thus increase pressure on the organisations for a simple and easy to modelling method.

In addition, some organisations require a high level of details (low-level abstraction) displayed using a graphical approach to meet their modelling needs. And others require modelling of low-level details (high-level abstraction) enabling stakeholders to make a better decision. The issues described are the deciding factor for organisations to choose an appropriate modelling approach to suit their requirements. Most of the business process modelling methods (informal) based on graph-based approaches to represent the concepts vaguely to communicate. Hence, most of today's organisations require a communication mechanism to represent the artefacts, their relationships and interaction in an understandable way so a wide variety of stakeholders can interpret them explicitly.

The discussion in this chapter also establishes the need for a business process design in meeting the change effectively within the organisation. Organisations are overcoming such issues by continuously improving the design of the processes involved supported by introducing new concepts integrated with the existing concepts to accommodate the change. However, industry-leading modelling tools such as UML-AD and BPMN are based on conceptual schema but the composition of concepts within their standards only provide an intuitive description of the concepts. It creates a need to learn more about the conceptual modelling schemas presented in chapter 2 and 3.

A thorough discussion is provided to understand the problems faced by the King's College Hospital in representing their patient flows. Patient flows establish the complex nature of the healthcare sector requiring a clear understanding of the structure of the artefacts used and their qualitative and quantitative representation to help stakeholders in better decision making. An example is provided from the King's College Hospital for the readers' sake to determine the need for representing the precise structure of the artefacts that could assist in their optimum display.

Optimal representation of processes including patient flows provide a great value for the service/product end users. However, describing a clear structure of the involved concepts to represent a patient flow (process) pave the path of consistent execution of all artefacts. Additionally, the flow of the occurring activities (with precise structure) ensures efficient execution using scheduling techniques. Moreover, the coordinated activities express temporal flow that needs some exploration supported by a general temporal theory because it would be a contributing factor towards optimisation of the intended business process model.

A constructive research method is chosen to carry out the research and development of the solution to the problems stated in the research questions. The reason for choosing the constructive method is its relevance to the problem expressed here in this research. That would take us systematically to understand the problem faced by the industry and build up the foundation (theory) to apply in the real-life. The solution constructed and implemented in real life to show the importance of the topic and establish the contribution to the knowledge.

In the end, I would like to state that this chapter has demonstrated the need for a general framework providing systematically the formal semantics of frequently used artefacts of the business process modelling standards and verifying the models constructed to report errors (if any). Additionally, it has been emphasised in this chapter the challenges faced by the healthcare especially considering a case study from the King's College Hospital accident and emergency, and how this research could address its' issues such as long wait times and better resource utilisation etc. Besides an overview and structure of the thesis laid down for the ease of the reader. However, the next chapter will present a literature review to provide an empirical evaluation of the business process management concerning the business process as a concept and its corresponding representation.

Chapter 2 Background

This chapter forms the literature review to establish the backbone of this research relevant to the issues faced by the industry. Primarily, the business process (BP) constitutes an integral part of the business process management. It also makes provision for a bridge between information and communication technologies (ICT) and management fields. Therefore, business process management is a widely researched topic f for the development of method focusing on business process design and execution. For example, [Van der Aalst et al., 2003] states that business process management includes a collection of methods, techniques and tools to perform business process analysis for its design and execution. However, [Lindsay et al., 2003] emphasise that business process management enables businesses to identify the opportunities to improve the vital business components and their understanding to transform the performance supported by technology radically.

Thus, an analysis of the business process management lifecycle would assist in understanding the importance of its core component, i.e., process. Business process management lifecycle consists of phases organised in a cyclical structure presenting their related dependencies as shown in figure 2.1.

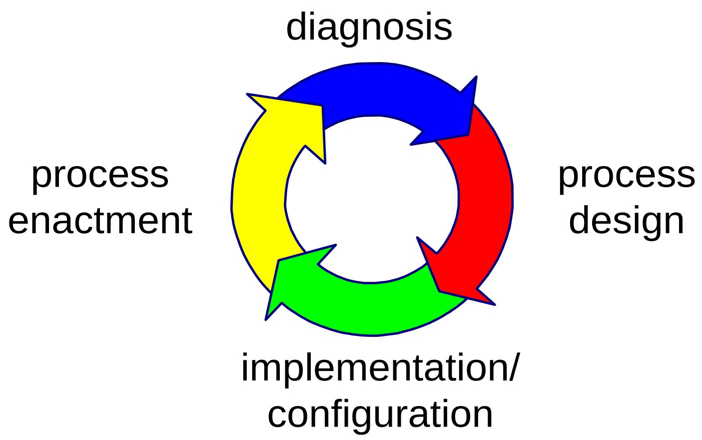


Figure 2 1 Business Process Management Lifecycle

Figure 2.1 shows the phases' occurrences establishing their reliance on each other focusing on the process design. During each of these phases, the process is revisited for continuous improvement. Because business process management has integrated the concept of continuous process improvement of the business process (re)engineering (BPR).

The conceived process subsequently configured and modelled based on the conceptual schema for its possible enactment. The model (diagrammatic representation of a process) constructed is further analysed for its structural properties to report any errors (process correctness) with its design or execution [Weske, 2007]. Hence, the model constructed (free of any bottlenecks) serves as a walkthrough for its stakeholders. Which is only possible at the diagnostic phase, that determines whether the design or execution have any undesirable representation to establish its (in) adequacy.

On the contrary, some organisations ignore the importance of the business process design and focus on its execution that present difficulties to them in the longer run. The reason behind their choice is the transaction value provided by the resources and planning tools by accessing to the crucial information such as some patients, staff levels, pharmacy, clinical/non-clinical materials, financial and administrative schedule. These tools add value to the organisations by transforming the planning and resource tools into practical solutions and have the capability to automate the procedures [Jarrar et al. 2000]. However, such tools are focused on execution without clear process (re)design (that requires continuous process improvement based on BPR later adopted by BPM) are not helpful rather create confusing models.

In both business process management and BPR, business process considered as a vital component and the authors [Smith and Fingar, 2003], [Ludwig et al., 1999] and [Luftman et al., 1999] agreed that it should be clearly defined (structure and boundaries) to enable one to meet the user requirements by achieving customer satisfaction. The discussion provided here emphasise the need for a method that could provide consistent process design and enactment. In addition, a precise description of the process would facilitate its effective enactment to improve the existing planning and resource tools efficiency and enhance organisation performance. Thus, an insight into process concept to establish its structure and boundaries require empirical evaluation for its possible standardisation.

2.1 Insight

Even though the concept of business process is cited since the 1990s but still the majority of the literature presented the definition(s) that only targeted the need of the researcher or practitioner with limited applications [Ferstl and Sinz, 1994], [Kueng and Kawalek, 1997]. Besides, the majority of the authors focused on specifying a business process to express varied aspects of the organisation. That makes the standardisation of the term BP cumbersome [Lindsay et. al., 2003] due to the constraints applied to it by the different domains [Melao and Pidd, 2000] and [Lindsay et al., 2003]. Besides, business process definitions provided lack in-depth to restrict its scope to make provision for a distinct meaning for its standardisation [Lindsay et al., 2003]. Because the business process as a concept conceived, configured and utilised differently by various domain experts depending upon their needs.

Although, the available definitions representing a specific domain having limitations to express a general view of the organisation's operations, therefore the corresponding models lack the correct representation of the system. This issue can be categorised either the descriptions provided by a domain are quite simple or too specific to express required features during its comprehensive implementation. In addition, the aforementioned authors noted various terminologies to describe a business process such as activity, task, process, function, output, input, information, human beings, machine, agent, resource, data, goal, object, product and service. It is important to note the variety of the terminologies present different ontology that used for the sake of describing a unique definition.

To discuss the term business process, I consider the primary definition is given in [Hammer and Champy, 1993] and [Davenport, 1993] that represent a business process as a collection of activities (partially ordered) providing value to its users. Both information and management domain adapted the understanding to define business process intuitively relying on varied terminologies (bearing different ontology) for subsequent modelling. Thus, it led to the development of several business process modelling tools furnishing the communication needs of the different domains to construct process models.

Analysts utilise the modelling tools to model the defined business processes to communicate the system behaviour to its stakeholders. However, the fundamental aim of the model to display a factual and consistent representation of the resources required to achieve the desired organisational objectives. That can be achieved with a correct model, i.e., free of bottlenecks, specifying the system capabilities. For the convenience of the readers, a generic business process model is depicted in figure 2.2.

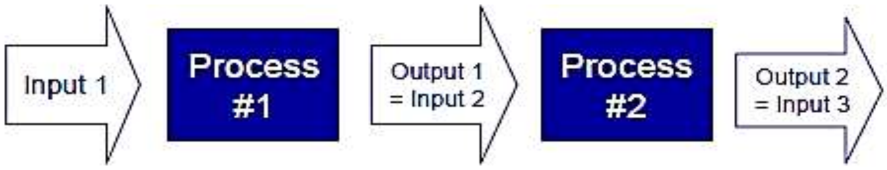


Figure 2 2 A generic business process model

There is some already compiled literature that unveils different features for further utilisation of process models:

- Focused on an understanding of the models to improve the design of the business process. And use corresponding models as communication channels [Curtis et al., 1992].
- Process manageability [Curtis et al., 1992] and [Neubauer et al., 2006].
- Models integrated with technical implementation to deal with organisational issues [Kueng and Kawalek, 1997].

Besides the importance of models in different domains, the analyst considered modelling as a wholesome approach. For example, modellers from the business domain require business process modelling tools to communicate the true meaning of the business process as a concept. Because the clear and concise purpose of a business process would enable a modelling technique to manage the knowledge better for its precise representation. That could further facilitate its enactment and possible automation. The industry has seen a development of numerous techniques, tools and methodologies focusing only on a specific problem and sought a solution that serves the enterprise best with attached primacy and pitfalls.

- helps achieve a full understanding of process representing organisation's rules and procedures [Curtis et al., 1992] and
- facilitates the gathering of knowledge; supports the testing of hypotheses and a learning process [Kueng and Kawalek, 1997]

In addition, [Kettinger et al., 1997] emphasised on the development of business process modelling techniques and methods making provision for continuous revision of the process for its suitability to the real-life and implementation.

The above discussion highlights the fact that effective communication within an enterprise achieved via a method that models a business process with precise description and subsequent verified. Because the modelling methods supply description of the concepts (business process and its sub-parts) to construct a model requiring procedures for its authentication. One can achieve this by analysing the modelling tools artefacts for the ontology used to specify business process and its components [Shanks et al., 2004] and [Gehlert and Esswein, 2007].

[Melao and Pidd, 2000] and [Aguilar-Saven, 2004] reviewed the modelling techniques featuring business process to address its different characteristics. [Aguilar-Saven, 2004] has provided a categorisation of several modelling techniques that are based upon two areas;

a) four-utilisation purposes by labelling them as either descriptive, decision support for design/development, execution, or provide support for enactment. Where b) distinguished between active and passive models, i.e. dynamic and static. However, the emphasis was on the (re)engineering of business processes and modelling as a domain to manage the knowledge effectively was ignored.

Besides, [Melao and Pidd, 2000] considered terminologies used to define business process and its sub-components for reviewing the modelling techniques based on four different viewpoints, i.e. algorithmic, intricate, vigorous for knowledge management, to determine their suitability in expressing the real world. Although, the fundamental concept of the business process includes all four viewpoints and could provide a baseline for the comparison between modelling techniques. However, none of the existing methods makes provision for a distinct business process description to facilitate such features together in a modelling technique (research gap). In addition, when a business process is instantiated generating a large volume of data used for different other purposes, but it is not what I will be considering (out of the scope of this study).

To fill this gap attempts made to consolidate the existing approaches by streamlining business process management, starting with several proposals for standardising business process modelling techniques [White, 2004]. To achieve this, business process description requires normalisation to accommodate change and transform modelling. In addition, the models constructed by the current modelling techniques (intuitive) depict the flow of activities to accomplish the desired goals. Therefore, [Vergidis et al., 2008] reviewed the available modelling approaches and proposed a categorisation that is demonstrated in figure 2.3.

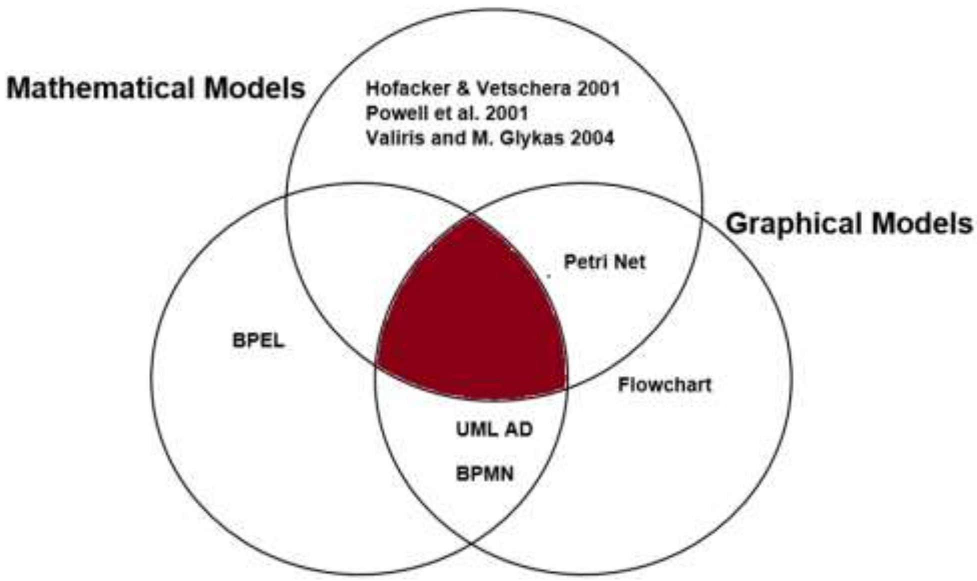


Figure 2 3 Categorisation of modelling techniques

Figure 2.3 represents the categorisation of existing modelling approaches and the area which I have highlighted with colour red represents the research gap needs to be filled. Graph-based approaches shown in the diagram, i.e., flowchart, unified modelling language activity diagram (UML-AD) and business process modelling notation (BPMN), have the ability to be more expressive than the rest of the approaches shown in the diagram. But, these techniques are considered informal [Zakarian, 2001] and present a vague description of the business process and its components. Moreover, these techniques do not support consistency for complex processes due to no formal underpinning [Valiris and Glykas, 1999]. Given the variety of modelling approaches displayed in the diagram, i.e., algorithmic, formal, graph based or execution, modellers prefer an approach with ease to model that presents a consistent representation of the operations.

Furthermore, all of these techniques are insufficiently equipped with relative and absolute temporal information that deter them in analysing the constructed models for their verification [van der Aalst, 1996] and [Phalp and Shepperd 2000]. Although, [Valiris and Glykas, 2004]. [Zakarian, 2001] and [Aguilar-Saven, 2004] reviewed the graphical modelling paradigms and considered them descriptive and lacked formal semantics. Moreover, they insisted upon developing modelling technique equipped with analytic capabilities for consistency and improved business process models.

The above discussion shows the research gap in the development of a methodology that could provide a verifiable conceptual schema of a process (re)design. If developed, such method can facilitate explicit and measurable targets to achieve strategic goals [Lewis, 1993]. Although, transformation is a desirable feature of a methodology that could provide a mechanism to map the intuitive model to the formal method for its verification using explicit temporal specification [Cheikhrouhou, 2015].

The essence of the analysis provided here concludes that formal modelling techniques are not the first choice of the designers due to their intricate structure. However, a method utilising the real-life knowledge formally presented then one can express the precise and clear understanding of the business process. Moreover, if the knowledge supported by a well-suited temporal structure [Juliane and Van der Aalst, 2004] then it can improve the overall business process modelling that is lacking in the current modelling techniques. Due to the issues identified above, many efforts have been made to bring about an approach that could address both aspects (formal semantics with a diagrammatic representation) [Chishti, 2014] providing a general knowledge base used for communication facilitating reasoning and representation of univocal business processes. Thus, it is important to have an insight into the conceptual modelling which is provided in the next subsection.

2.2 Conceptualisation

Concept based modelling provides the insight to the stakeholders for understanding the business structure, features and critical operations. The modelling techniques build upon conceptualisation approach assist in new system development for improved performances [Weske, 2007]. Furthermore, conceptual modelling describes the different views of physical objects used in human life for their comprehension and simplification (so that they could be represented and reasoned about). The terms belonging to a specific domain may be defined formally or informally to provide semantics. In addition, jointly they describe the characteristics of the world phenomenon for its better understanding, interpretation, dissemination and prognosis.

Commercial modelling techniques choose conceptual (informal) modelling to model business process and ignore the importance of accurate display of the related concepts. But, if the concepts are formally defined then the problem can be resolved. Similarly, It may be of great assistance to conceive the concept of the business process (coupled with its components) embodied with formal semantics to improve understanding and communication.

The intuitive knowledge base (conceptual schema) provided by the business process modelling standards resulted in an inaccurate depiction of a system. Although, formally defined modelling artefacts would assist in providing their precise semantics to present a model's correct behaviour but complex in nature. As a result, this section emphasises the need for formalising terminologies used to describe business process and its sub-components within a business process modelling method. Thus, to proceed with the discussion the identification of the knowledge base components (terminologies) required that consider them primitive for representing the system and its behaviour accurately. A formal description of such artefacts expresses the accurate and comprehensive representation of the proposed system.

Hence, it has increased the importance of describing the knowledge base for business process with clear semantics to model with better traceability of footprints. Development of an approach with comprehension and success in the boundaries of the business process management only achieved if the concept of the business process clearly defined to meet the requirements of continuous process improvement [Chishti et al., 2014]. Although, the meaning (ontology) of the terms establishing their semantics supported by ontological engineering. As it enables the desired concept description, understanding, interpretation and organisation [Guizzardi, 2005]. It also expresses substantiation of the facts by distinguishing the sub-parts of a concept and corresponding relationship shown in figure 2.4.

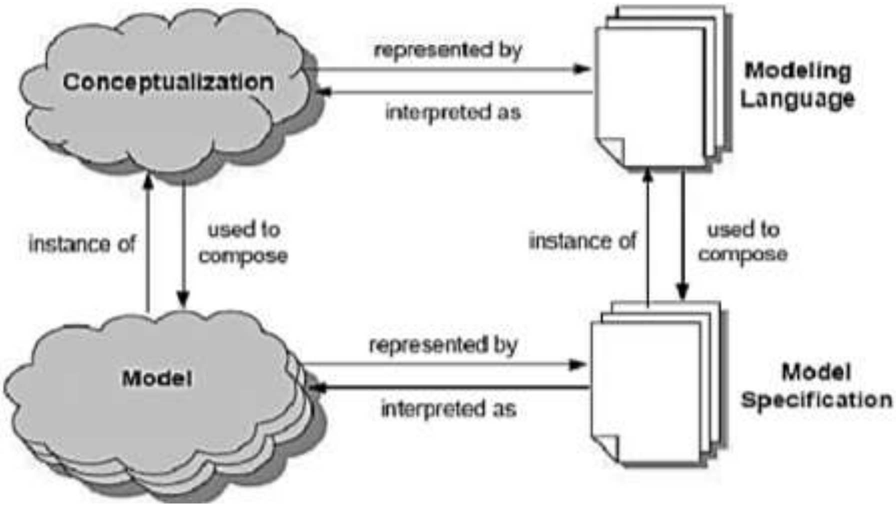


Figure 2 4 Relationship between conceptualisation, model, its specifications and language

Figure 2.4 emphasises that using conceptual modelling for constructing the business process models facilitates the easy to understand representation and supported by some verification and validation mechanism. More importantly, a knowledge base should be an exact fit for its real-life implementation. Because [Gehlert and Esswein, 2007] discussed the issue of development of a method for modelling (processes) with a certain number of modelling artefacts. They further provided mapping to support the argument of having a specific ontology for a certain construct as shown in figure 2.5.

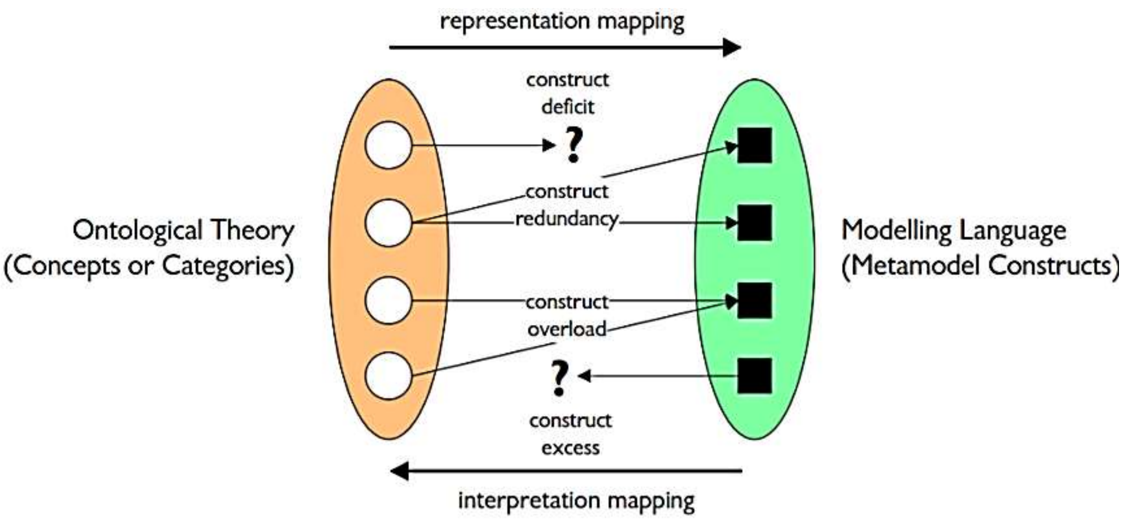


Figure 2 5 Concepts and corresponding notation 1:1 mapping

Figure 2.5 represents four categories for the modelling methods relying on concepts base on their ontology. However, out of four three categories mentioned in the diagram (apart from construct deficit) makes the modelling standards unclear.

Besides, process modelling techniques share a common composition regardless of their domain. Because they comprised of extensive concepts to structure a process and its outcomes to display a low-level abstraction for determine process enactment [Rolland, 1993]. The enactment can be improved by (re)design of the process and its flow for the optimisation purposes that require formal approach to analyse and evaluate the process(es).

Ultimately, the industry is interested in improving the understanding of organisations and their processes, facilitating process analysis and design and supporting process management in general and especially its modelling (for execution). Hence, to expand on the topic, I will briefly address the business and technical domains' viewpoints that use the different modelling tools (based on the conceptual schema) comprised of wide variety of artefacts (knowledge base) in their remit to express the behaviour of corresponding systems.

2.2.1 Business Domain

Enterprises considered as a collection of individual processes represented in a model to display their functioning that could help them in attaining the desired goals [Márquez 2007]. Therefore, industry concentrated on providing understandable models that are easy to conceive and represent the system behaviour in a simplified manner on a broader spectrum. That resulted in the development of variety of process modelling tools specifying the appropriate knowledge base to accomplish the overall business goals.

To be within the scope of this research, I have only considered concept based modelling technique, i.e. Business process modelling notation (BPMN) adopted as a standard for the business domain. It provides intuitive knowledge base facilitating understandable modelling of the enterprise' behaviour with no execution semantics. [Havey, 2005] identified the issue pertaining to limitations associated with BPMN such as model verification and validation procedures for business process execution. Although, it is equipped with a mechanism to map a developed model to be executed using the business process execution language (BPEL). The object management group (OMG) is continuously working to improve the standard. However, the direction of its efforts needs changing to meet the demands of the industry of a comprehensive modelling standard having no redundancy.

2.2.2 Technical Domain

Workflow management consortium (WfMC) described the business process as a collection of related activities worked together towards a mutual goal to represent an enterprise' structure by establishing operational roles and their relationships [WFMC 1999]. Thus, the need for a knowledge base required by the technical domain experts relies on a business process description consisting of procedural rules that could address and resolve the specific problems associated with system development. Technical experts quantify the value of the modelling approach by its organisation and enactment. Even in all domains including technical modelling approaches have their differences to model a business process. But, IT industry mainly reliant on concept based modelling technique known as unified modelling language (UML) activity diagram (AD) to model business processes specifying system-level behaviour.

Besides, the focus of business domain experts mainly providing easy to understand modelling tool though technical field experts concentrate on procedure-based system design. Because technical developers do not consider readability a major issue including interpreting manual tasks. However, there exists commonality between both domains about the tools utilised for modelling business processes having similar concepts. The difference between them only appears in their documentation relying upon different terminologies to serve the same purpose (business process modelling).

Both standards have documented their respective knowledge base comprised of massive terminologies supported by graphical constructs. But, modellers of the related domain are confused due to the overload of intuitive descriptions of the concepts (informally specified) and also leave the question for the industry to consider them as standards. Nonetheless, if these concepts supported by algorithmic-based accuracy then the respective knowledge base representation improved facilitating further analysis of the model constructed.

Per the paradigm shown in figure 2.5 suggests that both modelling techniques' artefacts redundant tools providing unclear semantics and inconsistent modelling. Thus, the use of fewer concepts (most often used artefacts) with a precise description for both domains facilitates expressing those concepts in a unified way to subsume those concepts. In addition, it will help in laying down a foundation for business process modelling.

Recently healthcare domain becomes more reliant on both tools (BPMN and UML-AD) for the modelling of their patient flows because of their concept based modelling schema. Therefore, to understand the viability of these tools in healthcare, I would provide a review of their suitability in the following subsection.

2.3 Modelling of Patient Flows in Healthcare

Patients in the United Kingdom under the spectrum of National Health Service (NHS) or around the world requiring quality services to improve their safety and time taken to deliver them. Modelling becomes more crucial when considering a high-dependency environment such as accident and emergency department at a hospital. The reason is to meet the high levels of resource required to deliver effective care to the patients. Thus, utilisation of the process viewpoint in the healthcare domain may support quality services delivery for better decision making. A quality service required by healthcare sector achieved by optimisation and scheduling of activities involved.

Furthermore, the healthcare sector deals with the human lives making its modelling efforts more cumbersome because failure occurrence noticed late in delivering care [Antonacci et al., 2016]. The reason behind such issues is timely resource allocation in delivering care to patients. Another challenge faced by the healthcare sector is the choice of modelling methods that are not flexible in modelling patient flows to accommodate variability. A method providing adaptability when modelling patient flows and inferring performance considered as an option [Bocciarelli et al., 2014].

Thus, healthcare modelling needs are dependent on process design and its precise modelling. Process design considered an integral part of business process management that has attracted the attention of the healthcare experts for the patient flow modelling [Stefanelli, 2004] because process orientation is not restricted to a specific domain. Primarily, process conceived by both the business and technical domain experts without considering the needs of healthcare domain, rely on modelling tools such as flow chart, unified modelling language activity diagram (UML-AD) and business process modelling notation (BPMN) asserting varied features of the organisations. Healthcare professionals adapted the suitable tool to represent multiple activities and their consolidation represents a particular process or patient flow in the attempt to deliver the care services. However, these methods are limited in their inception to (re)design the concept of process precisely that is required by the healthcare sector for the improvement of their services provided to patients [Berwick, 1996] and [Wilson and Harrison, 2002].

Moreover, patient flow modelling not only depicts the flow of the activities involved but also facilitate stakeholders with improved planning [Camann, 2001]. It relies on the methods and techniques of business and IT fields for a solution in an attempt to support better decision making [Perreault et al., 2001]. Thus, modelling of patient flows recognised as a medium to improve the overall quality of the services delivered to patients.

Besides, a consistent activity flow in the patient flow model ensures the patient safety and quality of service provided. That is achieved by the analysis of the (re)design of the activities and removing redundant activity flow for improved communication between the stakeholders [Curry and McGregor, 2005]. Therefore, it is vital to conceive the concepts that are logically consistent at all abstraction levels for an optimised patient flow [Horn, 2001], [Haraden and Resar, 2004], [Szwarcbord, 2005] and [Jensen et al., 2006]. Still, patient flow modelling not considered a key part of any healthcare initiatives neither within UK nor abroad. Due to this, existing process modelling methods not specifically designed for patient flow modelling and therefore, failed to capture the full complexities [Mans et al., 2008] of patient pathways. In addition, healthcare workers lack in the understanding of these techniques and the concepts used within for their adaptation to patient flow modelling [Jun et al., 2009].

However, the adaptation of the modelling standards to healthcare indicated that both the modelling standards lack in providing adequate support to facilitate communication and improvement in constructing patient flow models. The breadth of patient flow modelling is quite intense and therefore researchers tried to address the problems related to healthcare by providing rules for modelling clinical pathway [Seila, 2005]. Because healthcare has additional requirements to be expressed such as patient needs, safety and high levels of specialist knowledge required appropriate concepts for their consistent graphical representation.

The constructs provided by the modelling standards have no logical foundation to express the complex patient flows (pathways) that resulted in inconsistent models and poor support for decision making [Curry et al., 2005]. Although, standardisation of the patient flow modelling as a primary concept discussed by [Mills and Tanik, 1995], to date the healthcare sector lacks a modelling method that specifically defines its related concepts to express different perspectives (including temporal) for effective knowledge representation [Jensen et al., 2006]. The possible solution to such problems avoided due to the variability of the healthcare environment. But, if artefacts precisely defined accommodating their corresponding qualitative and quantitative temporal information then construction of a correct patient flow model is possible making provision for further analyse and identifying related performance issues.

Similarly, [Bhattacharjee et al., 2014] insisted on improved healthcare system that is possible with an improved methodology for the patient flow modelling. They further reviewed analytical and simulations methods for their appropriateness to modelling patient flows in hospitals. Analytical methods such as queuing and Markov chains considered not suitable due to their inability to model (not graphical) complex situations. On the contrary, simulation methods (not graphical) selected for performance analysis of the hospital operations. Again, the issue pertaining to the development of a method based on logically grounded concepts suitable for the healthcare domain not addressed because the focus of their research served only statistical modelling.

Yet, healthcare domain experts only rely on existent methods, which are not fully equipped to provide them with a reasonable solution to their problem. In addition, another important point which was missing in their study that no knowledge of 'what if scenario' was considered which provide a fundamental step towards alternatives flow labelled with the earliest and latest times (if available) to achieve optimal results. The focus of the current research is modelling techniques' enhancements and in the eyes of the author of this thesis, a methodology supported by the knowledge relevant to healthcare for an optimal solution.

This research will adopt a systematic approach to introduce an inclusive framework that provides the artefacts supported by their distinct ontology. Subsequently, these artefacts would support different levels of abstraction via consistent graphical representation. Hence, with the help of a specific enumeration consisting of fundamental lexicons or taxonomy could provide a formal semantics for general adaptation to model process correctly. Additionally, the specified concepts follow some time sequence to structure a model. Therefore temporal dimension needs to be explored to show the importance of quantitative and qualitative temporal information in the next subsection.

2.4 Temporal Perspective

A process model usually describes processes involved, their structure, how the related sub-components are coordinated and the corresponding enactment. Modelling techniques show the flow of the processes primarily associated with interval temporal logic such as 'process A occurs before process B and process B occurs during process C'. Business process modelling techniques such as UML-AD and BPMN represents the time vaguely. Because interval temporal logic has its limitations in representing time points. Therefore, both modelling standards deficient in expressing enhanced qualitative and quantitative information.

The reason to consider an explicit temporal class would facilitate in representing temporal aspects adding value to the precise display of a model to meet the stakeholders' requirements such as minimising the budgetary costs and reducing the waiting time patients spend at the hospitals. Healthcare sector could improve its services to meet customer satisfaction (which is time bound) with the incorporation of temporal aspects. In addition, introducing optimality of time can benefit hospitals to represent improved patient flows that impacts in reducing the costs involved to deliver resource bound services.

[Jablonski and Bussler 1996] reviewed the field of modelling presenting business processes with its different views. But, the concepts used for constructing models to address the issues such as recurrence and lack the conformity of the process models. Also, the question of process modelling addressed using various solutions including workflow patterns framework. The framework also expresses a wide range of viewpoints to direct the flow control, resource, data, time and anomalies. However, the temporal perspective provided does not explicitly encompass all the angles, i.e. enhanced qualitative and quantitative temporal information representation.

Moreover, existing Workflow Management Systems (WfMSs) utilised by the organisations to model processes has limitation in providing support for representing temporal conditions [Bettini et al., 2002] and [Pozewaunig et al., 1997]. Besides, it has attracted substantial attention in the workflow research community [Marjanovic and Orlowska, 1999]. These authors have dealt with time management based on different classes of temporal logic that present their strengths and drawbacks. Similarly, the explicit time constraints with reliability are missing and therefore not addressed.

Mainly the existing standards and frameworks only rely on interval temporal logic to represent duration of process or sub-components. However, authors failed to address the issue of breakable and unbreakable interval duration that is of great importance for modelling real-life scenarios. Additionally, casual use of temporal constraints related to the system' operations and unpredicted waiting times could interrupt the flow of the activities hampering overall consistency. This interruption could increase the costs of process modelling and enactment [Panagos and Rabinovich, 1997]. Therefore, it is crucial to specify the artefacts (associated with temporal objects) and corresponding temporal constraints while designing and managing business processes explicitly.

Albeit, it has been noted that choosing the right class of temporal logic could address the issues noted above. That would be vital in deriving constraints from a complicated process. Therefore, a temporal perspective considered pivotal in (re)designing the business process and its orderliness understandable and consistent by eliminating any similarities of occurring worldly objects' footprints. Furthermore, the temporal information incorporation(both qualitative and quantitative) achieved by the logical representation of the concepts specifying the enhanced temporal constraints and corresponding dependencies explicitly [Eder et al., 2000]. That can further supported by a mechanism to precisely schedule the process flow and required resources to achieve process optimisation.

The discussion addresses the problems related to business process representation starting from its conception, structure and design leading to its implementation, control and monitoring with respect to the time. These problems are present in the literature, but solutions provided not adequately address the foundational issue that is no logical basis for business process modelling. Therefore, it is of great importance that any recent or upcoming modelling methods should describe the necessary knowledge base (concepts) of a business process to provide precise details of a system supported by enhanced temporal constraints specifying boundaries between activities. In this way, the modeller would be able to analyse process design to improve overall process description and its understanding which may result in increased profitability/satisfaction by providing improved services.

2.5 Summary

This chapter has discussed the background of the related literature to identify the need for specifying the business process and its sub-components to be modelled graphically with regards to their timely occurrences (both qualitative and quantitative). The integral part of business process management lifecycle with regards to business process design and continuous improvement discussed to highlight its importance as the core concept of the knowledge base.

The literature reviewed provided insight into the existing modelling approaches covering different features of the enterprise. The problems indicated associated with the existing research work and the solutions provided. Existent modelling approaches noted in the literature are either algorithm based or graph based (intuitive). On one hand the algorithm based methods are formal but developed not considering the requirements (ease and simplicity) of the modellers for the process modelling. On the other hand, the graph based approaches have the ability to represent business process with ease but their informal structure cause inconsistency. However, there is an approach, i.e., business process execution language (BPEL), developed only to meet the execution needs of the graph based BPMN.

The main issue to consider is to provide a knowledge base that has the capability to specify a business process (with a certain number of the modelling artefacts) because current modelling languages considered redundant due to the availability of a large set of modelling artefacts and not all used within a specific business process model. Furthermore, existing modelling paradigms are based on conceptual schema but still lack in providing an exact enumeration for the modelling of a consistent business process. Besides, their ability to incorporate the temporal specification is limited. Hence, having too many or too little terminologies can make a modelling method not suitable for any domain. Therefore, it is of huge importance that a method with a certain number of modelling artefacts required to specify the enhanced qualitative and quantitative temporal information for consistent modelling.

Moreover, the existing modelling standards knowledge base comprised of intuitive artefacts that cause vagueness displaying a complex business process model. Albeit a knowledge base with precisely described concepts based on some well established logic could achieve the aims of this research and fill the gap identified in chapter 1. Because logic has capability to capture the concept and its boundaries with regards to their temporal occurrence. However, the current business process modelling standards rely on the interval temporal logic symbolically that does not suffice the industry requirements of constructing a correct business process model. Therefore, to address such issue I have reviewed the literature relevant to business and technical domain utilising the conceptual modelling approaches to incorporate temporal information.

Similarly, relevant literature analysed for the utilisation of business process modelling standards in the healthcare revealing their limitations in modelling the patient flows. Thus, in the eyes of the author, an approach based on conceptual schema comprised of a certain number of precisely defined artefacts (knowledge base) incorporating a more expressive temporal theory would assist in representing a well defined structure and organisation of the business process and it's sub-components to express the coherent and consistent business operations. Therefore with the assistance of a more general temporal theory would facilitate the modeller in constructing a correct business process model integrating the well defined temporal constraints associated with the individual artefacts. Ultimately it would help the healthcare industry in tackling with the time and resource bound activities to deliver improved services by reducing the waiting times at the hospitals.

Now, to understand better concepts of processes and its sub-elements, a review and critical analysis of the leading business process modelling languages, i.e. BPMN, UML-AD and Petri net, is required and presented in the next chapter.

Chapter 3 Review of Modelling Techniques

There are different modelling techniques used for business process modelling including informal and formal. To meet industry needs object management group (OMG) has released several versions of a unified modelling language (UML) especially for activity diagrams (ADs) to meet IT industry requirements. OMG considered the need of the business analysts to model business processes and released a standard, i.e. business process modelling notation (BPMN).

Mainly informal modelling techniques rely upon conceptual modelling because it has the flexibility in extending the modelling artefacts in a given structure. But as emphasised in the earlier chapter that the informal techniques loosely describes the concepts. The conceptual schema of the modelling standards comprised of artefacts vaguely describes concepts (to construct a business process model) represented graphically (known as constructs). For example, a term ‘Action’ adopted by UML-AD and ‘Task’ considered by BPMN to build a process model (expressing the same ontology in their functionality) associated with respective graphical constructs (metamodel). But their structure and organisation is informally defined that leaves room for their different interpretation by involved personnel such as an ‘action’ or ‘task’ occurs during another ‘action’ or ‘task’ and their boundaries information is missing, therefore, the modellers and analyst failed to specify an accurate depiction of a complex business process. That is evident when the constructs instantiated to represent a business process and its corresponding flow.

However, formal methods such as Petri Net adopted for business process modelling but has the intricate structure to model a business process and has no provision for ease to model a business process expressing wide variety of associated features. For example, the focus of this research is having precise enumeration of artefacts representing enhanced qualitative and quantitative temporal information. Thus, this chapter would comprehensively review the (informal) modelling standards, UML-AD and BPMN, and Petri Net (formal technique) for their suitability to the commercial world to model correct business processes with ease and simplicity. More importantly, considering their utilisation concerning the temporal perspective as discussed in chapter 2.

3.1 Unified Modelling Language

Over the last few decades, object orientation has evolved and adopted it since the 1990s for the system development. Because it has the flexibility of reusing the objects and provides the facility for developing system supported by tools that are platform-independent. However, different approaches use a diagrammatic representation of the objects to assist in the design of software but differ in their notation and specification. A variety of diagrammatic representations embodied together known as the unified modelling language [Cornwell, 1999]. It is used to express the functioning of systems' objects and their communication during the enactment phase [OMG 2015].

UML is categorised into 13 different diagrammatic representation consisting of various rules to suit the needs of the system development. These diagrams further divided into three types to represent the system's structure, its behaviour and corresponding management of the model constructed. Where, structural diagrams represent organisations of the objects and their idle relationship, e.g., data and function. Behaviour diagrams display the IT system operation such as the behaviour of system objects while executing. Model management diagrams represent the IT system modules around system objects. Furthermore, it provides cost-effective solutions by improving the system's overall (re)design and its subsequent development for possible execution. Besides, in the eyes of IT system developers, object-oriented techniques may assist further in the process automation.

UML applications found in various fields due to its extensive tool support. Therefore, the success of the object-orientation in the IT industry has led to UML utilisation in the business process modelling domain to improve the description of artefacts for an efficient model. Due to its technical adaptability, IBM and OMG had worked on a project such as UML-to-BPEL transformation [Koskela and Haajanen, 2007] for process model execution. Even though, UML is widely accepted and used in organisations and endorsed by heavyweights of the IT industry but considered imprecise to model a complex business process.

Albeit, modellers with a lack of technical knowledge of the object-oriented approaches have avoided it to use for business process (re)design and enactment [Eriksson and Penker, M., 2000]. Without the support of clear business process description, modellers restricted to represent models' different features. In addition, further analysis of the constructed models

Thus, the model constructed using UML requires further analysis for its correct objects description, relationship and flow. The modelling techniques with clear definitions of its components can be beneficial in building understandable business process models that provide insight into their structure and temporal features [Aalast et al., 2003]. In this thesis, UML-AD considered for modelling business process and patient flows so I will focus only on it

3.1.1 UML Activity Diagram (AD)

Unified Modelling Language (UML) Activity Diagram (AD) considered similar to the simple flow chart and data flow diagrams to represent the structure and behaviour of an enterprise. UML-ADs represent the different behaviour of the involved activities using control flow that illustrates the changing characteristics of a system [OMG 2015]. UML- AD belongs to the behavioural diagrams using tokens that resembled with Petri Nets [Wohed, 2004] that is a formal modelling paradigm

UML-AD adoption as a business process modelling standard makes it relevant to this study to be reviewed and analysed further. The recent revision of UML 2.5 compared with the UML 2.4 indicates that the meta-model almost remains the same for its concrete syntax. However, abstract syntax defined the notation, and its semantics (describing the ontology of the concepts) intuitively based on Petri Net to represent the activities' sequence flow with tokens. In general, UML-AD is comprised of different terminologies coupled with graphical constructs to express the behaviour of the system.

Besides, UML standard leaves the onus on modellers to opt for the best-fit constructs to model business processes, leads to different interpretation by different stakeholders. Therefore, identifying the most commonly used artefacts considered vital that will be discussed in chapter 4. Technical modellers use UML-AD to model the process objects and the variety of activity flow such as how to diagnose a patient in a hospital' accident and emergency (A&E) department but they failed to answer questions such as how to improve the patient flow concerning their waiting time at the hospital.

3.1.2 Critique of UML-AD

As stated in the previous sub-section, the intuitive semantics provided by the OMG as part of the standard documentation leave room for the inconsistent development of a typical business process. Where a typical business process model may contain several actions to represent procedural computation that invoke other activities to express the flow of control within the hierarchy [OMG 2015]. Therefore, further investigation and analysis of UML-AD required.

UML-AD is limited in expressing the precise ontology of its terminologies documented in the OMG standard to represent a business process using the constructs. The reason is its intuitiveness and no formal description availability to support them [Wohed et al., 2006] and [Russell et al., 2006]. The two main and widely used terminologies of UML-AD are activity and action. The action considered a core part of the activity diagram intuitively defined to represent the behaviour of the atomic operations invoking other actions/activities. However, the business may comprise of many actions coordinated together to show the sequence. Furthermore, the activities are segmented using swimlanes to represent different roles and organisational units. In addition, it includes no definition for deferred events and dynamic invocation and lacks in describing the "well-formedness" procedures to combine a fork and join.

[Eshuis, 2002] has attempted to provide semantics but the descriptions provided for the concepts are intricate and inaccurate. Furthermore, UML-AD has a limitation in expressing ideas of case and interaction of a business process model. However, modellers without technical knowledge are unable to use UML-AD to model a process with details (at all abstraction levels), i.e. high to a low level [Bell, 2004]. Due to these issues, UML-AD failed to attract practitioners.

Moreover, UML-AD restricted in representing data resources preventing it to model the organisation' archive and distribution (capability) resources. Albeit modellers rely on partitions to specify the organisational units and their respective roles involved in the collaboration but no provision for resource allocation. Because it may cause problems when one individual needed to assign to a single resource with a specified time restriction. However, there are no constructs to represent the time with an upper bound of specific actions in managing the activity deadline [Korherr, 2008]. And, the resolution to such problem is only possible at the time of execution of the concerted actions.

3.1.2.1 Limitations

Process models used to represent different aspects of an organisation so they should be analysed considering three main points that include their logic, time, and performance [Li et al., 2004]. However, further analysis of UML-AD constructs highlights the issues of missing these aspects to express the correct behaviour of a system. Conceptualisation can lay down a foundation to express the correct behaviour of the artefacts involved in a process model, i.e., verification. Moreover, UML-AD not specifically designed to model patient flows. Therefore, it does not have the capability to identify the time gaps of consigning patients on a constructed model. Besides it is deficient of defining any role types and assigning of physical resource to specific staff for a given time period to model a patient flow.

With time analysis, the modeller can express temporal constraints between process model artefacts, i.e., validation, during instantiation. The two points above can assist in functional consistency but lack in providing improved performance. With the help of performance analysis, the modeller can evaluate the requirements of the model to meet the strategic goals of an organisation. The aforementioned three aspects are missing that are missing in UML-AD.

Although business process performance analysis acknowledged by the industry that can provide quantitative analysis but so far no efforts are made in addressing the issue [Salimifard and Wright, 2001], however, the optimal process design is of great importance [Hofacker and Vetschera, 2001]. But, no mechanism is available to achieve optimisation at design and enactment stage of a process model [Völkner & Werners, 2000].

Many modelling techniques including UML-AD used to organise and structure the business processes but to achieve optimisation remained with modellers' intuition to choose a tool [Hofacker and Vetschera, 2001]. No method provided to meet business process optimisation [Zhou and Chen, 2003]. Besides, enterprises can be more competitive to retain the market share [Zhou and Chen, 2003] by satisfying their customers.

There are authors who have reviewed UML-AD for its use in healthcare settings such as [Goossen et al., 2004], [Saboor et. Al., 2005] and [Chishti et al., 2017] for modelling patient flows. [Goossen et al., 2004] focused on modelling generic nurse care processes without evaluating the developed model. Furthermore, the review was missing the vital points such as the viability of existing concepts and their use along with features necessary to model the nurse care process.[Saboor et al., 2005] review provided a method to enhance UML-AD making provisions for adding details to the clinical processes for quality assessment. They provided additional notation for evaluating the clinical processes (radiological process). But the problems of having a general knowledge base to accommodate the timely occurrences of each part of the clinical process for an improved model persisted.

An extension of UML-AD provided for the healthcare to model patient flow with clinical documentation [Spyrou et al., 2005] but no satisfactory evaluation provided. In addition, the concepts used for data representation (clinical documentation) are not enough to represent the particular features such as patient's safety with regards to the time associated with the flow. The importance of time whether qualitative or quantitative must be accompanied while modelling the patient flow for improving not only the patient journey but also it time stamps the associated medical records and resources utilised during the overall flow.

In another attempt [Lyalin and Williams, 2005] provided an additional notation to UML-AD to a single diagram for improving cancer registration process and suggested the additional concepts have the power to be used in other domains. The additional concepts used for descriptive purposes specifying vaguely the associated time information (qualitative and quantitative). However, the clarification provided for the use of relevant process timeline (vague) with regards to other resources utilisation does not precisely depict the behaviour of the cancer registration process. In addition, a what-if analysis not provided concerning time and resources based on additional concepts. Overall, the UML-AD enhancement failed to provide a knowledge base meeting the healthcare requirements regarding time and resource restraints implied in a hospital setting.

The above review shows that UML-AD lacks in addressing the aforementioned issues to specify the business process and its structure correctly. In addition, the focal point of this research to represent the exhaustive relative and absolute temporal information between the modelling artefacts is missing too. That could further assist in analysing the business process performance for optimisation, i.e. time and cost. For instance, reducing wait time at the hospitals' accident and emergency department, quality of service provided that may result in patient's satisfaction.

3.2 Business Process Modelling Notation (BPMN)

OMG considered the needs of the business analysts concerning process modelling and released a standard known as business process modelling notation (BPMN). BPMN is not limited to a simple modelling language but attempts made to provide a comprehensive solution for system design and development. The foundation of BPMN does not inhibit modeller in choosing an expression that is limited to a specific predecessor. Therefore, BPMN is a useful addition in high-level modelling processes aided by some free text annotation [Dumas et al., 2007]. BPMN as a standard has also attempted to provide a set of conception levels to combine business and system development [Lano, 2009].

BPMN serves as a communication channel for inter and intra-organisations' purposes. In addition, it has combined the earlier approaches such as UML-AD and Petri Net aiming to address the needs of the business domain. It has relied on the concepts used in UML-AD utilising different terminologies but bearing the same ontology. For example, BPMN introduces 'task' bearing the intuitive semantics of atomic activity that is exactly the same as used in UML-AD labelled with term called 'action'. In addition, BPMN uses a graphical collection notation known as business process diagrams to represent detailed meta-model. It shows different tasks a participant must fulfil and lets them communicate in a standardised and straightforward way [Kretschmer, 2014].

A classification of these diagrams provided in the standard documentation [OMG 2013] which modellers can use with no considerable training. The classification of these diagrams given below

a) ***Flow objects*** comprised of events, activities, and gateways. These concepts represent the systems' state, operation and flow respectively.

b) ***Data*** is used to represent data objects to show data addendum, its outcome and for the store.

c) ***Connecting objects*** used arrows to specify the order, i.e. sequence flows including communication flows, e.g., message flows, between collaborators. Associations and data associations used to link artefacts to elements.

d) **Swimlanes** are comprised of the pool and lane concepts to represent organisational aspects. To express roles within an organisation 'pool' used to represent a partition between activities. Swimlane is used for describing the organisation viewpoint.

e) ***Artifacts*** used to provide enhanced information via annotation such as group and text. They do not affect the behaviour of the process.

There are additions made to the existing graphical constructs to accommodate the changes within the industry to represent the behaviour of an organisational enhancement. Real-life business processes are complex and change regularly, but still, there are minimal efforts made by OMG to address the need using BPMN standard [Rogge, 2011]. However, it is best to have produced an accessible technique to concentrate on the business process main attributes to avoid any complicated addition to the method [Allweyer, 2016]. , I will review BPMN in the sub-section to provide critique and its limitation.

3.2.1 Critique of BPMN

BPMN representing a typical business process isomorphic to UML-AD representation. BPMN leave the onus on the modeller to use a wide variety of terminologies and constructs to design and specify the process models. But, the stakeholders within an organisation having different skills and expertise may interpret the used artefacts variedly and creates confusion. Furthermore, the BPMN semantics for process execution is platform-dependent that makes BPMN un-interoperable and not portable [Recker and Mendling, 2006], [Gao, 2006], [Ouyang et al., 2006], and [Weidlich et al., 2008]. Due to this, modellers have to go through the different sections of the standard documentation regularly to model a complex business process. Not only it makes the job of a modeller intricate but its other stakeholders too who require a model interpretation.

In addition, [Recker, 2010] considered the usage of the available BPMN graphical constructs and divided them into four different categories, the common core, the extended core, the specialist set, and the overhead. Common Core is comprised of a set of most used constructs. Whereas extender core and specialist set use a large variety of constructs that may use every now and then. However, the overhead collection is comprised of a large group of constructs that mainly not used. The reasons for this categorisation is for the different practical use of the constructs, and its benefits could be twofold. The first reason is to identify the lack of use of the extended and specialist constructs due to their complexity and the small additional value of these constructs. The second reason is the modellers' insufficient knowledge about the constructs and in particular the extended constructs.

[Muehlen and Recker, 2013] looked at the BPMN models of business analysts with different backgrounds and experiences in the process-modelling domain. When looking at the different models, they investigated the separation between core and extended constructs holds in practice concerning the user acceptance of BPMN. The constructs frequency distribution shown in figure 3.1. The findings shows (figure 3.1) that only 20% of almost 50 constructs are used. Besides, more than 50% of the models evaluated for the same reason and found only five constructs are utilised, e.g., a process initiates with an event (start) and completed with an event (end) used representing the corresponding flow.

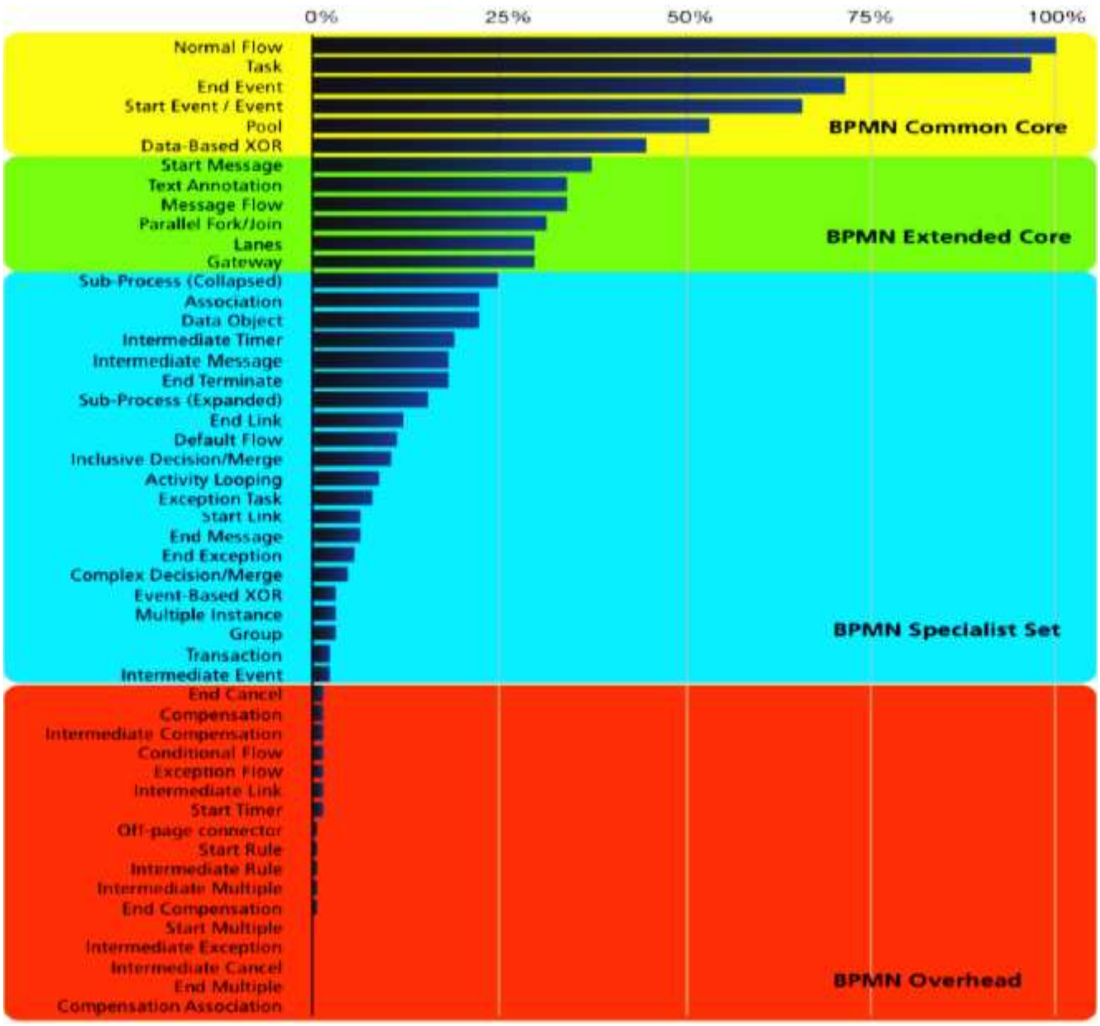


Figure 3 1 Frequency distribution of usage of constructs in BPMN

BPMN models use ‘pool’ to represent communications between business-to-business collaborators. For example, an instance of a specific role directed to communicate with an instance of another role already used in the pool [Dijkman et al., 2008], a very similar approach to UML-AD ‘partition’.

Furthermore, the construct used for swimlane has no impact on the functioning of a process model. Their existence is merely to show the roles of the personnel in the different collaborating organisational units. And they do not add value to their performance with no effect whatsoever on any information that they may use in the resource utilisation concerning their completion time. Similarly, it didn’t offer any information concerning the objects structure, its value and represented hypothetically [Lodhi et al., 2011]. In conclusion, BPMN terminologies used in constructing a (complex) business process model have semantic incorrectness due to its intuitiveness [Frappier and H. Habrias, 2012].

3.2.1.1 Limitations

[Recker et al., 2005], [Großkopf, 2007], [Dumas et al., 2007], [Wei, 2010] and [Völzer, 2010] and the list of authors noted here is not exhaustive who have reviewed the BPMN and reported the problems attached with the standard in modelling a typical business process. One of them is BPMN's large set of inter-definable terminologies represented graphically as part of the standards' metamodel, used for implementation that is imprecise due to inherent vagueness [Börger, 2012]. In addition, [Wohed, 2004] and [Wohed et al., 2006] found BPMN constructs difficult in their outlines compared with UML-AD and flowchart. However, in the eyes of the author, if one only considers the frequently used constructs of both BPMN and UML-AD then they found isomorphic to each other.

Moreover, BPMN standard documentation does not provide any support for the unclear physical process modelling [Recker and Mendling, 2007]. Because the concepts used in BPMN lack clear semantics in representing certain features including the time that uniquely related to business process design. It has been suggested to define only core constructs that may be of more use ensuring their utilisation to model a typical business process (complex) without burdening the standard [Börger and Thalheim, 2008].

Similarly, [Müller and Rogge, 2011] discusses the use of BPMN in healthcare process modelling focusing on role and task assignments. They added coloured tasks to attribute the role information using lanes. But the problems with BPMN to build a consistent model is not considered which is the actual demand of healthcare for patient flow modelling to provide safe and timely services to patients. However, [Barbagallo et al., 2015] used BPMN for the optimisation and scheduling of operation theatre' resource allocation. On one hand, they had to define the concept 'pathway' completely for it to be accommodated within the tool for its utilisation. On the other hand, for the scheduling and optimisation purposes, only 'duration' is utilised for the expected resource allocation.

The reason behind the aforementioned issues lie in the ontology of the different constructs provided, that is somewhat vague, making their conception needlessly complicated. The vague description of different overlapping constructs can result in the inconsistent process model, and the standard has not provided any solution to fix such issues. Because standard conformance is missing that is evident from representing and understanding a concept differently by the different stakeholders hampering the communication between them when deciding upon a concepts' interpretation.

However, [Van Gorp and Dijkman, 2012] pointed out that for a long time proper formalisation of BPMN constructs was lacking. That has an impact on the consistency of the model constructed due to vague semantics of the extended constructs. What can be seen from these studies is the consensus that the constructs, and in particular the extended constructs attached with their timely occurrences, are for some reason hard to use for business analysts. In entirety, the clear semantics considered a way forward in the author's opinion to provide consistent process modelling concerning their temporal perspective.

A survey conducted by [Cheikhrouhou, 2015] revealed that 'TIME BPMN' tried to provide a classification of flexible and inflexible use of qualitative and quantitative temporal conditions such as "As Soon as Possible" and "As Late as Possible" including other constraints. However, 'TIME BPMN' does not allow to model business processes and its sub-components to determine their relationships to represent the flow concerning their corresponding length such as 'an activity lasts 'x' time units and 'x' may be bounded by interval duration. Therefore, it has hampered the 'TIME BPMN' efforts to provide appropriate scheduling of activities for process optimisation.

A vital question needs answering is that does BPMN provide in its standard a clear description of the constructs used so that a practitioner can easily understand all levels of refinement and construct a consistent model? That also applies to the healthcare sector due to its adoption to model patient flows. To answer this question and as concluding remarks for the BPMN review, I can say that its standard documentation lacks the precise description of the concepts present in the metamodel and bringing more constructs to the standard is adding to the problems. These problems seen in the constructed process models exhibiting ambiguity, conceptual underspecification because of their unclear semantics. It emphasised on the fact that BPMN should revise their standard to meet the standard. More importantly, BPMN fails to display the extended qualitative and appropriate quantitative temporal information as part of the process representation. Which is a piece of crucial information that can heavily improve the graphical representation but also facilitate depiction of business process enactment to analyse the model performance for possible optimisation, i.e. time and cost

3.3 Petri Net

Both (informal) UML-AD and BPMN have adopted concepts from Petri Net (formal). Therefore it is necessary to look at Petri Net closely to find out more about the roots of their constructs conception. Petri Net is mainly a system modelling technique that has received the most attention [Reising et al., 1992]. It is a formal technique focusing on examining the constructed system to depict its operational changes mainly concurrency of workflows [Peterson, 1977] and [Peterson, 1981]. Petri Net not specifically designed for describing and modelling business processes but utilised later in an attempt to meet the industry requirements presenting no reasonable impact. Due to its algorithmic foundation supported by four main subtle graphical components, i.e., place, transition, token and arc shown in figure 3.2.

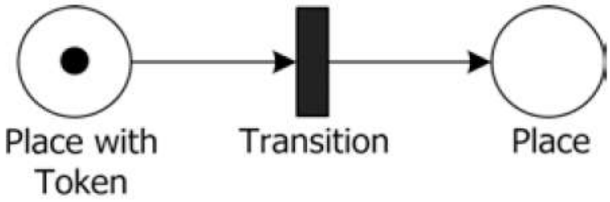


Figure 3 2 Petri Net essential elements

Figure 3.2 shows an input place carrying a token and connected to a transition, using an arc. The branching out arc from a transition linked to the output place. Keeping in mind that a place capacity determined by its weight using tokens. In the absence of any weight attached, it is assumed that place has one or infinite weight. The arc specifies the usage of the tokens assigned. Additionally, when a condition appears in a Petri Net, then the dedicated place and an arc would weight one. Therefore, if two states have met the requirements, then a transition is ready to fire [Peterson, 1977].

Besides, the input place would have the least value of the tokens assigned that is required for a transition. It is only possible when all the inflow tokens are accepted carried by the arc having enough capacity to sustain them. It enables a transition to fire consuming tokens received from input place that represent the performing of the tasks. The firing will result in placing the outgoing tokens in the specified output places that subsequently enables several transitions. Petri Nets are also considered for modelling in deterministic distributed systems to express their parallel behaviour. A Petri Net example is shown in figure 3.3.

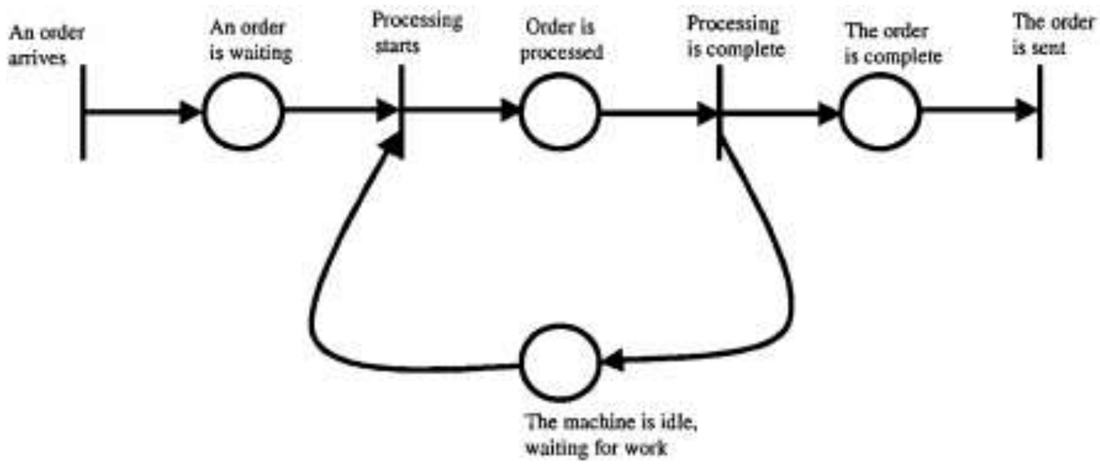


Figure 3 3 Order example using Petri Net

Petri Net provides a mechanism to avoid confusion while shaping the systems, and furthermore, the modeller has access to its analytic capabilities to display the operation. because of its formal nature, Petri Net extended by providing different variants such as colour, time, and hierarchy to address the need at a time to suit the researchers' interest in system modelling [Jensen, 1997]. Apart from the main three extensions, there are few other versions introduced to meet a specific need of the modelling [David and Alla, 1994]. These extensions provide a separate and particular set of rules for each respective extension to meet the required functional needs [Van der Aalst and van Hee, 1996].

As stated above Petri Net is adapted to model business processes, so to represent activities, actions, tasks and events graphically, transitions are used. Place used to show the state of a system occupying tokens and express marking of activity, action, task or event. However, the arc used to display the connection between transitions and places.

3.3.1 Critique of Petri Net

Strictly speaking, Petri Net has accepted widely as a formal technique for system' modelling, but its structure is not relevant (and precise) enough to administer and model complex business processes [Leymann and Altenhuber, 1994]. Additionally, Petri Net concepts and corresponding complex relationships between them are relative making it cumbersome to specify a complex business process [Hofacker and Vetschera, 2001] and [Tiwari, 2001].

[Eshuis and Wieringa, 2002] compared activity diagram and Petri Net for workflow patterns and found UML-AD more expressive than Petri Net due to its relevant structure to the business process modelling domain. In addition, a list of researchers [Valiris and Glykas, 2004], [Powell et al., 2001] and [Hofacker and Vetschera, 2001] noted difficulties in representing different modelling viewpoints such as temporal etc. using Time Petri Net. They have also claimed Petri Net lacks in its structure to capture the complexity of the business processes whether its' related to expressing decision or concurrency. Furthermore, Petri Net plays a deterrent role for business process modellers in perceiving, describing, understanding, managing a business function to verify and validate while providing absolute information of occurring activities concerning a process [Koubarakis and Plexousakis, 2002]. In the eyes of the author, Petri Net also incapable of accommodating extended temporal constraints [Chishti et al, 2014].

Real-life business processes are reactive where Petri Net has issues with modelling a responsive system. [Eshuis and Dehnert, 2003] has noted a few problems that are listed below:

- In Petri Net, events are expressed either as token, place or transition. The issue is when streamlining the event tokens takes place that consumed by the transitions can cause problems in some cases where events are cancelled. However, when events are modelled as transitions, then the issue of synchronisation raised causing problems to identify the transition triggered by the incoming event.
- Activities are modelled using transitions in Petri Net, and their enactment presents the change in a state of the system. However, activity is considered a non-instantaneous concept while modelling business processes and workflows. On the contrary, transitions considered instantaneous. Petri Net also limited in shaping specific actions to represent the transitions for routing and decision of the workflow management system. It cannot distinguish between the actions enacted either by the environment or by the management system and modifying them to behave differently can hamper the system functioning altogether. There are other issues in the use of tokens while transitions are ready to fire but have the opportunity to opt-out or in some cases can postpone indefinitely, whereas, in modelling business processes, a response is required for every event.

3.3.1.1 Limitations

Petri Net lack in the ability to deal with the process to expressiveness in describing its precise meaning [Hofstede et al., 2009]. Because Carl Adam Petri focus was not on modelling business processes. Therefore researchers and industry have continuously worked towards adapting it with some modifications. But, Petri Net and its variants including time Petri Net lack in providing the appropriate concepts' enumeration to deal with real-life business processes [Ter Hofstede et al., 2009]. For example, it is assumed that in Petri Net an instance ends excluding enactment of viable paths. Therefore, the relevant tools are not capable of capturing the full related expressions. Given the algorithmic basis of Petri Net and its variants accompanied by the reasons provided by [Ter Hofstede et al., 2009] that the business process modellers inability to use Petri Net due to their complex structure and relevance.

Although Petri Net formalism has faced problems in modelling data to inform the involved participants, therefore a variant of Petri Net known as coloured Petri Net (CPN) used to address the data representation issues. In CPN, different colours used to represent the activities modelled as transitions having the case data. Yet no procedure in place to ensure reliable data access without any limitations. However, there are few attempts being made to map BPMN [Djikman et al., 2007], [Djikman et al., 2008], [Großkopf, 2007] and UML-AD [Storrle, 2005] to Petri Nets but failed in providing an appropriate solution.

Furthermore, Petri Net and its variants equipped with computational power to model workflows but lack in its suitability and expressivity to model business processes. Because the concepts used in Petri Net have no relevance in terms of their structure and application. In addition the attempts to model complex business processes with OR split and OR Join operations in Petri Net using object-oriented programming languages failed [Ter Hofstede et al., 2009]. Besides, Petri Net lacks expressiveness to the showcase the thoughts of organisational viewpoint [Korherr, 2008] and model concurrent and recursive business processes [Mayr, 2000]. For that reason, the issue of Petri Net providing suitable and related concepts with clear semantics to be used in business process modelling domain still pending.

There are researchers who provided the semantics for the Petri Net but not fit for the business process modelling specifically [Mukhrjee, et al., 2004]. Because real-life business processes are complex and concurrent and to model them using Petri Net is not possible as per the rule of no two transitions can be fired simultaneously. Besides, transitions used for both activities and events and it is crucial to distinctly represent the activities and events using a transition with specific semantics.

Moreover, Petri Net limited in expressing the exhaustive qualitative (temporal) and suitable quantitative time information when attempted to construct a business process. The scope of the research mandated the aforementioned requirements to model the correct business process. Because, both the qualitative and quantitative temporal information can ease the modellers' life not only to graphically model the business process but also to make sure their enactment is not flawed. Which subsequently assist in model analysis and boosting the performance of the process model, i.e. time and cost.

Petri Net rarely used for the modelling of patient flows but when it is used focus remained only systems development [Mahulea et al., 2018]. But no concepts added towards creation of a general knowledge base to express the different properties of the healthcare sector. [Hughes et al., 1998] used coloured Petri Net to model flow of patients from high dependency unit in progressive care to support decision making and scheduling. Again the complex structure of Petri Net has limited its ability to communicate effectively with the stakeholders and unable to express the timely bound resources in such a critical environment.

Similarly, [Criswell et al., 2007] modelled patient flows in an emergency department relying on Petri Net supported by discrete event theory to predict hospital state. As the Petri Net structure allows mainly performing statistical analysis of the available data and diagrammatically representing such a complex process and their outcome. That makes the whole model intricate for communication to the stakeholders from the different backgrounds and experiences. Therefore, keeping the healthcare sector main issues in mind with respect to the easy to understand concepts associated with their timely bound resource utilisation could communicate effectively and can enhance comprehensively the representations and performance of the hospitals in general and especially the accident and emergency department.

3.4 Summary

More than two decades of substantial work carried out in the business process modelling domain to bring about standardisation such as UML-AD and BPMN but failed to agree by the practitioners on the distinct business process description to ensure corresponding model correctness [Hofstede et al., 2009]. UML and BPMN as industry standards use a wide variety of terms/constructs bearing no formal semantics that leads to an ambiguous representation of the processes. Besides, these techniques differ in their usage to model business processes such as BPMN used only by business modellers, and UML-AD is used by technical designers to shape the respective business processes. The results produced by both orientations have ambiguities in their representation of complex business processes because of a large number of intuitive modelling terms with relevant constructs and the one used lack logical foundation [Chishti et al., 2017].

UML-AD and BPMN constructs failed to provide specific absolute and extended relative temporal information. And, the constructs used for modelling different aspects lack formal semantics. Due to the intuitiveness of these techniques resulted in unclear meanings of the concepts used and stakeholders are responsible for their interpretation. However, a specific ontology of the key concepts used in these techniques provided with a precise time order based on a class of temporal theory may address the issues pertaining to their explicit representation. Furthermore, non-availability of the logical foundation resulted in the ambiguous representation of business processes. [Chishti, 2014].

One may suggest that the technical viewpoint of the business process permits a comparatively cost-effective transition between its analysis and (re)design to a computerised solution to support its automation. On the contrary, in such situations, organisations tend to focus on development of the tool i.e. automation, rather than business process (re)design and analysis. That is evident from the development of different tools based on BPMN for process automation. Therefore, tool complexity makes the life of the process modeller difficult which results in abandoning it completely (evident from the industry response towards Petri Net).

It is important for the reader of this thesis to understand that the scope of this research requires a platform-independent method making it cost-effective and resolves the issues stated above. The research objective is to establish the core modelling artefacts used by the aforementioned business process modelling standards and their corresponding ontology. So that their precise description (process (re)design) accompanied by the required extended qualitative and quantitative temporal information for its execution (enactment).

Moreover, due to the unavailability of modelling methods specifically designed for healthcare, UML-AD and BPMN adopted to address their modelling needs. Again, the business process modelling standards intuitive structure and lack of enhanced temporal information utilisation within the construction of a complex business process make them insufficiently equipped to display the correct model. Therefore, the healthcare sector still in search of a method that could be used to model patient flows accommodating the variability within the pathways. However, Petri Net with its statistical approach used to address the issues related to systems' performance and scheduling. Though, due to its inherent complexity, stakeholders failed to understand disseminate the result produced.

In the eyes of the author, a knowledge base comprised of general but suitable concept enumeration would facilitate not only building a complex business process but also considered helpful in modelling patient flows. The knowledge base could be extended with the introduction of more notation associating with the class of temporal logic considered here for its power to accommodate instantaneous and non-instantaneous activities. The healthcare sector specifically hospitals may be benefitted from such method in an attempt to reduce waiting times at the accident and emergency department. Therefore, the next chapter will address the issue by identifying the core modelling concepts along with the example of their usage in constructing business process model.

Chapter 4 Modelling Artefacts

The conceptual modelling schema-based Petri Net considered not suitable for business or technical process modellers due to its complex structure to describe and model the real world concepts used in business and healthcare domains. Due to its complexity, for this research, I would not be considering Petri Net. However, the industry relies upon informal modelling techniques considered as business process modelling standards (as discussed in chapter 3). These standards adapt the terminology intuitively (borrowed from Petri Net) and massively overload the standard documentation with mostly unused terminologies and constructs. Thus, stakeholders find it confusing and difficult to align with the understanding of the modeller who utilises the terms carrying varied meaning of the same concept that interpreted differently by different individuals to make the model construction and its understanding complex and vague.

Industry standards burdened their documentation with too many (vague and intuitive) unused terminologies for modelling business processes (whether business or technical domain) make them redundant tools. Therefore, the models created using either modelling standard can result in an imprecise representation of the system. To address such problems, the discussion presented in chapter 3 has outlined the solution for precise modelling of business processes by introducing only a certain number of (formalised) terminologies and constructs.

Several researchers [Wohed et al., 2006], [Russell et al., 2006], [White, 2004], [Wohed, 2004], and [Van der Aalst et al., 2003] reviewed the unified modelling language activity diagram (UML-AD) and business process modelling notation (BPMN) and found similarities between the most often used terminologies and constructs bearing the same ontology that makes them isomorphic in their utilisation. Therefore, it is necessary for the industry to have a unique framework that would represent the processes precisely. The subsections of this chapter would identify the most often used artefacts of the UML-AD and BPMN in an attempt to unify them.

4.1 UML-AD Most Often Used Artefacts

Unified Modelling Language activity diagram (UML-AD) metamodel comprised of a wide variety of constructs (providing intuitive semantics) to represent business processes graphically. UML-AD notation mainly based on the abstract syntax of 'Activity' specifying the systems' behaviour. However, another term such as 'Action' serves as a fundamental unit (graphically represented as an executable node) in an 'activity' consists of coordinated executable nodes (actions). It may alter the system behaviour based on input and output values. In the activity diagram, executable nodes along with the edges used to structure and organise the execution of an Activity.

As discussed in chapter 3, UML-AD standard documentation includes a large set of terminologies and corresponding constructs which modellers find it cumbersome while making a choice. Furthermore, the vocabularies present in the standard documentation are not formalised, therefore the stakeholders interpret them as per their choice making their representation more confusing. Thus, it is considered important to select a certain number of artefacts from UML-AD used for the construction of a typical business process based on the discussion provided in chapter 3. For example to represent patient flows in the hospital settings while delivering care to the patients, several coordinated sub-activities of activity triggered to complete the operation (patient flow) (diagrammatically representing all involved actions). By considering such an example, I have chosen the constructs from the UML-AD shown in figure 4.1 and described individually in the subsections below.

Figure 4 1 Key UML-AD artefact

4.1.1 Action

There are two main terms 'Action' and 'Activity' used in the UML-AD. The abstract syntax used the ontology of the term action intuitively to represent an atomic activity and represented graphically as an executable node in the corresponding metamodel. In the UML standard documentation, the term action intuitively describes main computing operations, manipulation and communication in the activities. Initiation conditions need to be fulfilled for a work to be carried out, and the ending provides initiation conditions for the proceeding operations. Thet may also invoke other collaborating activities using activity edges. In the case of an occurrence of an anomaly, the concerned work would be abandoned without an outcome [OMG 2015 pp372, 441].

Similarly, an activity may represent different actions (executable nodes) invoked either directly, i.e. call behaviour, or indirectly, i.e. call operation. There are input conditions attached with the start of the executable node that needs to be met. To complete an end of action may trigger proceeding executable nodes. Action (executable node) graphically represented as oblong shown in figure 4.2.

Figure 4 2 Executable Node

To show a piece of time-related information in activity diagram, ‘AcceptEvent’ construct used to represent a date. Including ‘AcceptEvent’ and other graphical constructs used to represent action are CallBehaviourAction’ and ‘SendSignal’ but are not considered here considering modellers ease to construct an understandable business process model and leave no room for misinterpretation by the stakeholders.

Importantly, the standard documentation does not specify clearly the atomic structure of the ‘action’ (executable node) when some decisions or conditional branching involved concerning other actions. The actions involved in flow only represents their intuitive structure and not specifically describing their precise formation. For instance, in UML-AD, the action doesn’t specifically provide a structure presenting its start and endpoints concerning other coordinated actions such as an action A endpoint occurs prior to the parallel action B in a decision or conditional branching to start another involved action C.

4.1.2 Activity Edge

Edge is used between the actions and activities to show the direction of the flow and maybe labelled with guards to describe its weight and name (if any). An edge is graphically represented as a line having an arrowhead [OMG 2015, pp378] as shown in figure 4.3.

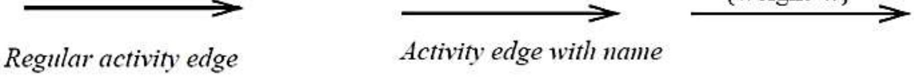


Figure 4 3 Activity Edges

There are other types of edges documented to specify an interruption of the operation because its utilisation and influence on the comprehension of a systems’ behaviour are minimal and therefore not considered as part of the necessary set of the modelling artefacts. Tokens are passed between the different executable nodes of activity with the help of edges to exhibit the operation (not considered here as part of the enumeration required for a typical business process construction). Because the token semantics does not support the enhanced qualitative and quantitative occurrences of the real-life actions concerning collaborative work units within an activity. A simple flow between two executable nodes shown in figure 4.4.

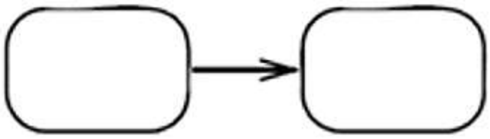


Figure 4 4 A simple example of edge usage between two actions

An activity edge utilised in UML-AD is labelled with information that is beneficial for the modellers. But using a different shape of edges to specify different behaviour confuses the stakeholder in its interpretation. Therefore, I would only be using a regular edge to describe the flow within activities.

4.1.3 Control Nodes

The scope of this research requires a necessary set of artefacts to build a business process model, therefore, I would consider control nodes (but only initial and activity final nodes), branching nodes, i.e. decision and merge, and concurrent nodes, i.e. fork and join nodes.

4.1.3.1 Initial Node

An initial node initiates an activity. The outgoing activity edge is carrying tokens that may be offered to connected executables nodes or collaborating activity. In the standard documentation, it is noted that If an exception occurs in operation to stop its movement downwards, then the initial node cannot hold a token expressed by the use of a guard. UML-AD standard permits use of more than one initial node within an activity that may have several outgoing flows [OMG 2015, pp385]. It is shown in figure 4.5.

Figure 4 5 Initial Node

4.1.3.2 Activity Final Node

UML-AD standard documentation includes two control nodes to express the completion, i.e. flow final (terminates a flow) and activity final node (terminates an activity).

An activity completes by accepting the available tokens on its inflow edges with the use of a final node construct having no outflow. Also, it stops all the live actions when it receives the first inflow edge with a token and accepts it out of several inflow edges (tokens) that are blocked/cancelled to complete the activity flow [OMG 2015, pp386]. The activity final node graphically represented in figure 4.6

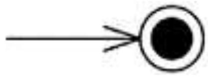


Figure 4 6 Final Node

The tokens reach the flow final node destroyed without affecting other paths of a model [OMG 2015] shown in figure 4.7.

Figure 4 7 Flow Final Node

The above discussion highlights the fact that flow final node has minimal to none effect on the behaviour of the overall system representing a typical business process so I would only be considering the activity final node as a most often used construct to constitute the necessary set of modelling artefacts (enumeration).

4.1.3.3 Decision Node

Branching behaviour of a system represented by using a decision node. The decision node is in operation when some of the actions have conditional flow in an activity. In such situations, only one outflow (after evaluating guards) is selected out of many discharges, i.e. ‘xor’ split. But, there is no mechanism available on the sequence of guards evaluation. There are specific rules, which makes decision node functionality limited such as all inflow and outflow edges are required to be either part of a set of object flows or control flows [OMG 2015, pp388]. It is noted that its modellers’ choice to choose the token for outflow to progress. A decision node is shown in figure 4.8.

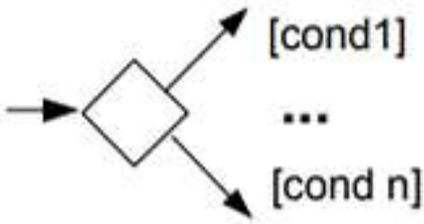


Figure 4 8 Decision node with guards

4.1.3.4 Merge Node

To represent a behaviour that expresses the situations where the flow of the system requires joining of inflows but no synchronisation (no tokens joining), a merge node is utilised to represent one outflow. Here, the same rule (as used in decision node) applies for inflows and outflows of the merge node [OMG 2015, pp387]. A merge node represented graphically isomorphic shape as of decision node, shown in figure 4.9.

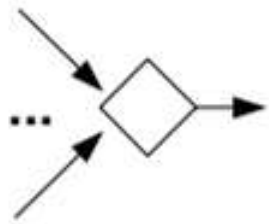


Figure 4 9 Merge Node

4.1.3.5 Fork Node

The concurrent flow of a system represented in UML-AD using the fork node. It is used to represent the split behaviour of a system, where several outflows bearing replicated tokens from a single inflow. In the case, at least one flow with a copy of the token is accepted then rest outflows can keep their tokens(duplicated) till their target consumes it based on first in first out queue [OMG 2015, pp 386]. The notation for fork construct is represented in figure 4.10.

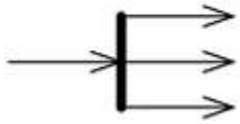


Figure 4 10 Fork Node

4.1.3.6 Join Node

A join node expresses a systems' behaviour where several inflows would result in one outflow. It is used to represent the synchronised behaviour of the concurrent activities within a system [OMG 2015, pp387]. Fork and Join nodes are parallel flows and expressed using the same construct as shown in 4.11.

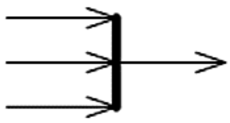


Figure 4 11 Join Node

With the help of both concurrent flow nodes, business processes initiate several instances at the same time to manage the flow. However, there are other constructs (in addition to the tokens) within the Activity diagram standard documentation such as pins, object node and object flow to represent the flow of control between the object nodes of activity. An object node is shown in figure 4.12.

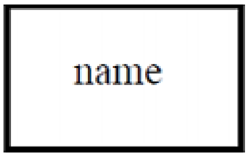


Figure 4 12 Object Node

There is no necessity to use object nodes because they have no significant impact on the overall operation to complete flow in representing a process correct structure. Furthermore, their graphical representation merely the same as an executable node, and the only difference appears when two objects are rendered to describe their flow of control using pins as shown in figure 4.13.

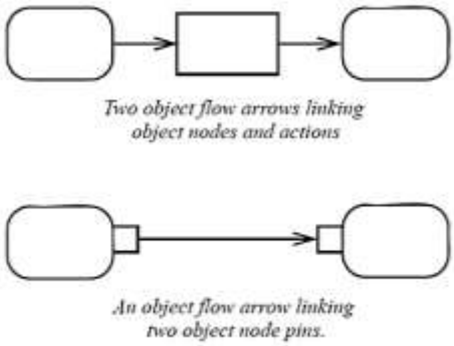


Figure 4 13 Object Flow

4.2 Discussion

The scope of this research requires preparation of a most often used construct set. Therefore, a discussion required to provide the relevance and suitability of the object node and partition constructs of the UML-AD standard. An example from the OMG standard considered to express its functioning concerning the aforementioned artefacts relevance and suitability.

Example 4.1 (Object Node): An order example [OMG 2015, pp 394] considered here shown in figure 4.14. In this example, Receive Order, Fill Order serve as Executable Nodes. UML-AD standard documentation describes Receive Order as an Initial Node that leads to a decision where Receive Order either rejected or accepted. In case of an order is being denied that leads to a merge node to Close Order. But, in case of order is being taken it needs to be prepared and filled using Fill Order action.

The Fill Order action led to a fork node showing the concurrency. In case the received order paid then it would be shipped and directed to a join node. An object node used to show the invoice (object) received. In this example, an executable node of Send Invoice used led to the creation of an object Invoice that is sent to the customer, and upon receiving the invoice, a Make Payment executable node has invoked the Accept Payment executable node. Furthermore, that meets to a Join node to synchronise the parallel behaviour. A Merge Node before Close Order used as per the standard rules leading to the activity final node.

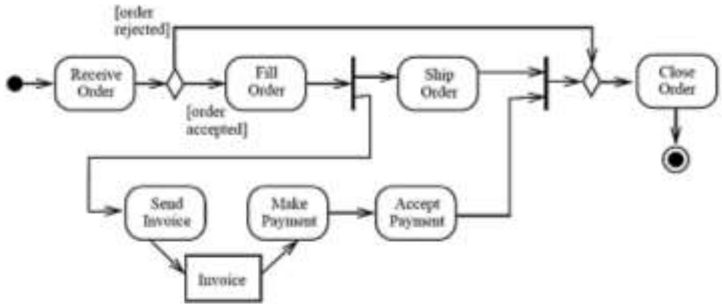


Figure 4 14 Order (process) example with object node

4.2.1 Analysis

An invoice considered as information communicated without an additional artefact. The invoice considered as the information required for the initiation of making payment (executable node) action may be expressed as annotation to the diagram instead adding another construct. That complicates the graphical representation with no added value to the overall structure and completion of the Order process. If I remove the node (object) of the Invoice then the above process modelled again without object node shown in figure 4.15.

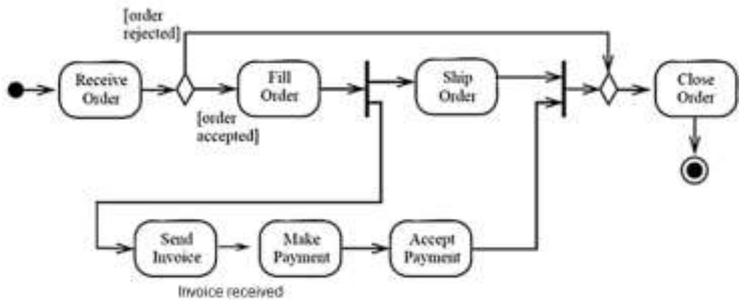


Figure 4 15 Order (process) example without object node

Figure 4.15 depicts a complete structure to represent the order process without the object node showing the lack of power and complicating the overall representation, therefore, not considered as a core construct in this research.

Activity partition of the UML-AD standard documentation used to represent the roles within the different units of an enterprise to display the operational aspects. However, it does not influence the behaviour of the activity diagram. Because it only showcases a specific part of the enterprise functioning applying global conditions to an executable node presenting just a limited view [OMG 2015, pp 406]. Furthermore, 'partition' bear no semantics shown in figure 4.16.

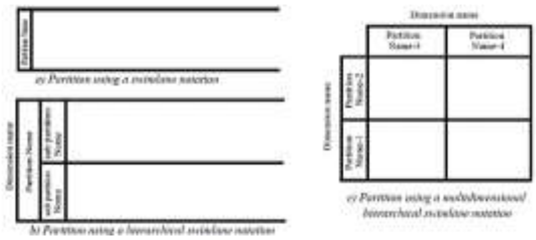


Figure 4 16 Swimlane

Example 4.2 (Activity Partitioning): I consider the Order example mentioned in[OMG 2015, pp 408] to establish the necessity of the activity partition concerning representing a typical business process. The three collaboration units in this order process are Order, Account and the customer shown in figure 4.17. The Order and Accounting units of the enterprise representing two internal entities whereas the customer represents an external participant.

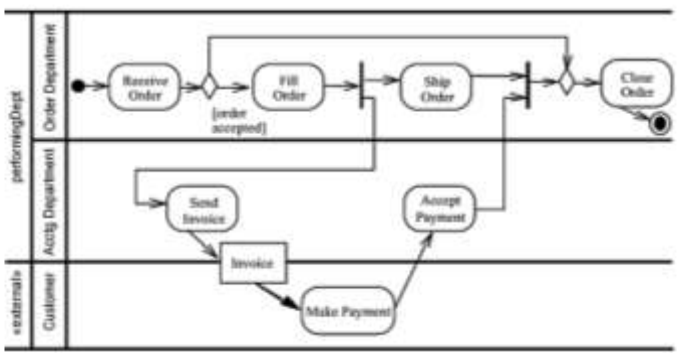


Figure 4 17 Order process using a swimlane

4.2.2 Evaluation

The order process depicted above via activity partitioning also known as swimlane indicating that the process has three participants. The only added information provided in this diagram is that naming the departments but the overall behaviour remains the same as mentioned in example 4.1 shown in figure 4.16. In addition, no quantification of roles provided within the specific unit facilitating distinct operations. Therefore, this construct adds no additional value to the functioning of the order process. UML-AD documentation uses this construct for only one reason that is stated above to specify the different units and roles, but the analysis and evaluation of the UML-AD as a standard provided in chapter 3 indicates that this construct has no impact on the overall achievement of the consistent and concise representation. Hence, I am not going to consider swimlane as a part of most often used artefacts.

4.3 BPMN Most Often Used Artefacts

Business Process Modelling Notation (BPMN) designed to serve as a communication channel for stakeholders (within or external to the enterprise). It is used to model the organisational processes depicting system behaviour. However, BPMN standard comprehensively documents syntactic rules for its different constructs but the corresponding semantics rendered based on inconsistent terminology [Dijkman et al., 2007]. Therefore, I will be identifying, discussing and analysing BPMN's most often used graphical constructs. These constructs will constitute the necessary set of modelling artefacts. The selection made here would facilitate in determining the terminologies used to model a typical business process. On one hand, this step would make the process modellers' life lot easier by choosing the specific construct in constructing a complex business process model.

In addition, if these constructs supported by the clear semantics then they would represent a concise ontology for the modellers and interpret by the stakeholders in a precise manner. The most often used constructs considered for this research are flow objects consists of events, activities including process, sub-process, tasks, gateways and connecting objects (sequence flow). The constructs such as data objects, pools, artifacts are completely ignored due to their lack of utilisation and irrelevance to modelling a typical business process making them beyond the scope of this research.

4.3.1 Events

The purpose of the term 'event' and its subsequent graphical representation (construct) considered in the BPMN documentation to influence the behaviour of a system (either by changing or stopping a flow within the process or sub-process in which it appears). BPMN standard documentation divided 'event' into three major types (start, intermediate and end) [OMG 2013, pp 238-276].

The naming convention used by the BPMN standard reflects on the functioning of these events. For instance, 'start event used to start a task and/or process known as the cause. ‘Intermediate event’ appears during a process to represent exceptions, e.g., a delay in the execution due to the wait for a message to arrive. ‘End event’ considered for the completion of a process known as result or impact. These events are utilised in various situations to describe a happening. For instance, a ‘catch’ serves as a trigger to initiate a process or task and the ‘end event’ throws” an outcome of the corresponding task or process. Modellers may use intermediate events to either create or react to the change in the behaviour of the system. These event types are shown in figure 4.18.

Figure 4 18 Start, intermediate and end events

Due to their significance, they are considered as a binding force to make the operation seamless for the achievement of the strategic goals defined in the enterprise objectives.

These three events are further appended to include a variety of extended ‘event’ constructs for modellers to report changes within the enterprise. These types are none, user, message, timer, rule, link, multiple, error, terminate etc., representing a difference of occurrences of the corresponding data artefacts. The various types of three events are shown in figure 4.19.

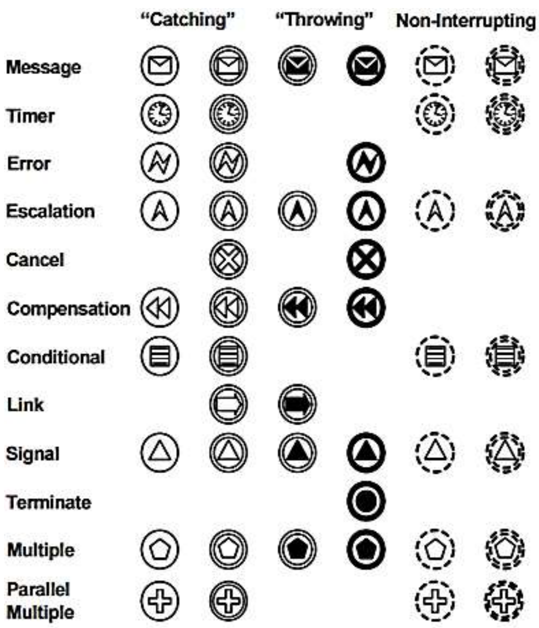


Figure 4 19 Event types (BPMN)

However, not all of them fully utilised making the standard documentation burdened and confusing modellers to choose which specific event type is suitable for a specific instance. Overall BPMN 2.0 review provided in chapter 3 highlighted the fact that most of these events are unused and only the main events are used to specify a typical business process.

A typical business process diagram depicts the expected and required behaviour of a process concerning the time and resources utilised. Therefore, a flow of such a business process would consist of the start event, a combination of tasks within a process and/or sub-process with a terminating event (end event). Furthermore, an exception may occur during the normal process operation that either delays or alter the expected outcome anticipated by the intermediate events e.g. timer event [Dijkman et al., 2007] graphically represented by attaching the intermediate event to its extremes.

BPMN does not provide process execution semantics that differentiates the usual and exceptional situations. Although, with its determining context used to highlight the unexpected behaviour within a process. Thus, I would be considering the basic event types, i.e. start and end, to be included in the necessary set (enumeration) of modelling artefacts.

4.3.2 Activities

BPMN standard includes the generic term “Activity” to represent the work performed. The term “Activity” further split into two types such as atomic and compound, to represent the composition of the process illustrating all the abstraction levels [OMG 2013, PP 29].

4.3.2.1 Task

Atomic activity termed as ‘task’ considered when the system is required to depict the unbreakable behaviour of activity having no internal structure [OMG 2013, pp 156-167]. The task is the fundamental element of an activity to provide low-level details of a model graphically represented as a round rectangle shown in figure 4.20.

Figure 4 20 Task

There are other types of tasks available in the BPMN documentation such as service tasks, manual tasks, user tasks, script task etc. but only differs in their graphical representation and no change in the semantics compared with the essential task with some variation, that has no impact on the overall representation. Therefore, I will only be using basic 'task' as part of the necessary set of the modelling artefacts for the development of the framework. I have not considered rest of the task types due to their lack of utilisation and irrelevance in constructing a complex business process.

4.3.2.2 Process/Sub-Process

Compound activity is termed as a 'process' and 'sub-process' to represent the activity comprised of sub-components. It represents a high level of detail consisting of a network of tasks within a process or sub-process. Furthermore, it facilitates the communication between the activities occurring external (but related) to the business environment [OMG 2013, pp 173-181]. Another type of compound activity known as a sub-process that further divided into different types such as collapsed and expanded sub-processes. These different types of sub-processes differ in their respective graphical representation e.g., collapsed sub-process does not express its details within its construct shown in figure 4.21.

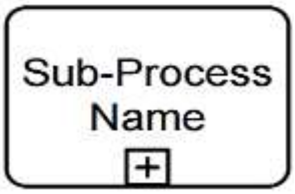


Figure 4 21 Collapsed Sub-Process

Expanded sub-process used to display the granular details of a sub-process as shown in figure 4.22.

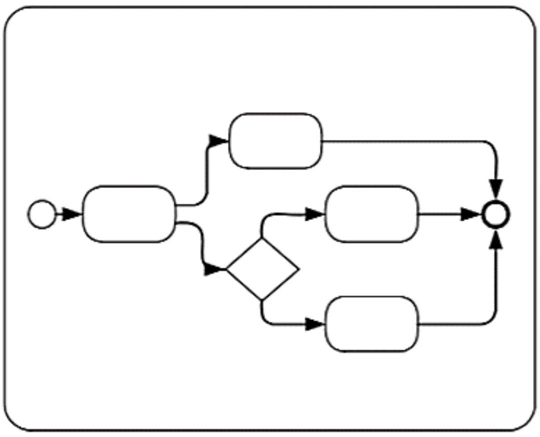


Figure 4 22 Expanded Sub-Process

Examining both the types of the constructs visually combined with their intuitive semantics, an expanded sub-process graphically considered more suitable in representing the coordination of tasks within a sub-process and more explicit in its structure and description. Within a process model, stakeholders may confuse themselves by considering the collapse sub-process structure as the task (due to their similarity in their graphical representation). Hence, the collapse sub-process not considered a part of the enumeration serving as the core modelling artefacts for this research. An example from [Weske, 2007] below strengthens the general understanding (discussed above) and representation of a sub-process concept.

Example 4.3 (Sub-Process)*:* A sub-process representing the credit risk evaluation composed of tasks acquiring credit data (Get Credit data), examining risk attached (Assess risk) and sending the results (Send evaluation) expressed as two types of sub-processes of BPMN (collapsed and expanded) shown in figure 4.23.

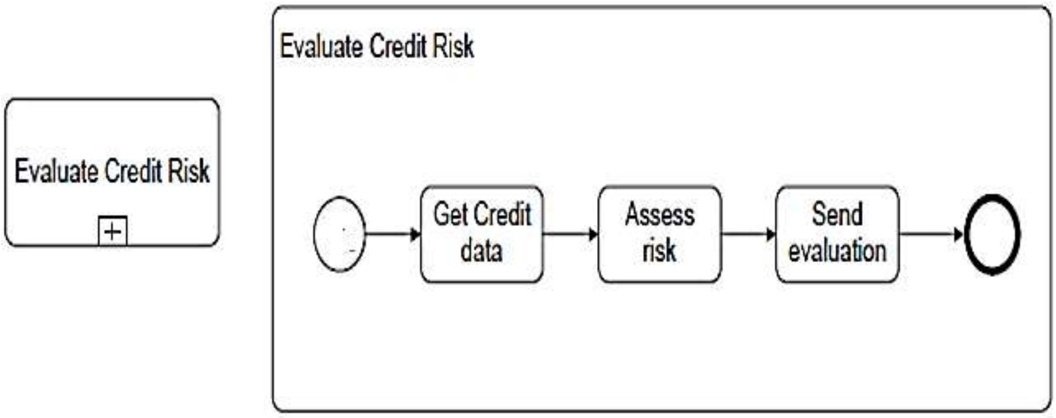


Figure 4 23 Collapsed and expanded sub-process example (BPMN)

The plus marker used within the standard task graphical construct representing the collapsed sub-process. It may confuse the modellers in distinguishing the constructs and stakeholders may interpret it differently. Hence, a graphical representation of a collapsed sub-process may confuse the stakeholders in understanding it as a task that uses a name in the middle of the diagram. Hence, not considered as a necessary artefact of the set for this study. The expanded sub-process distinctly and uniquely expresses the functionality of a sub-process, which is comprised of several tasks. The composition of the expanded process is more suitable to represent a part of the complex business process (sub-process) and considered as a part of the enumeration.

4.3.3 Sequence Flow

As the name implies, it establishes the order of the events, tasks and sub-processes performed within a compound activity. Additionally, BPMN documentation splits the sequence flow into a variety of other flow types such as normal, conditional, default, exceptional, un-controlled and message [OMG 2013, pp 34-35]. However, the investigation presented in chapter 3 found that the modellers resort to normal flow to model a usual business process to express the complex system behaviour. Hence, for this research, I would be only considering a normal flow to construct a typical business process.

4.3.3.1 Normal flow:

It mainly used to show the flow between different events, tasks and sub-processes emitting from the boundary of a stream of other events, tasks and processes/sub-processes etc. except for intermediate event. A normal flow graphically represented in figure 4.24.

Figure 4 24 Sequence Flow

BPMN has extended the standard documentation and included a few other types of sequence flows that have minimal to none effect on the processing functionality. Hence, I will only be considering the normal flow as part of the enumeration comprising of the core constructs and the rest of the flow types not considered in this thesis due to their lack of utilisation and irrelevance in constructing a complex business process.

4.3.4 Gateways

BPMN use a general term gateway to represent the control (normal or complex) behaviour within a process to precisely represent either several inflows or outflows. A generic diamond shape used to express the control mechanism. However, to show different control mechanism, markers used within the basic graphical construct to depict different control behaviours such as merging, joining, branching and forking [OMG 2013, pp 287-300]. These gateways provide a mechanism to avoid any conflicting occurrences that may change the behaviour producing a deadlock or livelock. Moreover, they assist the concerned stakeholders to understand and intervene at the right time. For example, 'Exclusive' gateway assist in establishing an input flow that is traversed into at most one output flow. Gateways would also result in reducing the redundancy [Börger and Sörensen, 2011].

A variety of control constructs representing exclusive. branching, inclusive, sophisticated parallel behaviour and other event-based gateways are shown in figure 4.25.

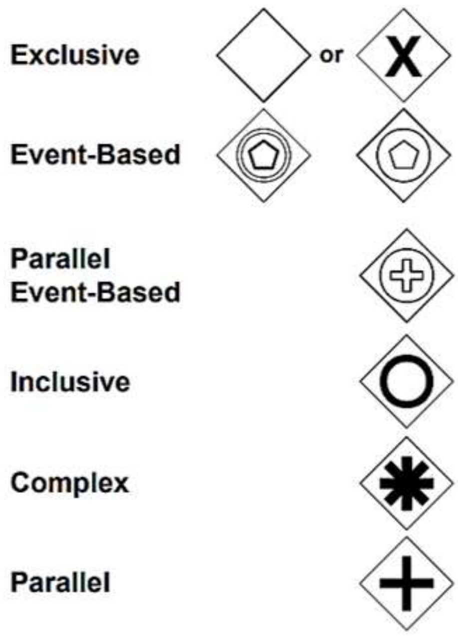


Figure 4 25 Gateway types (BPMN)

The list of gateways shown in figure 4.25 includes the extended control constructs to assist modellers with decision making. For example, a process with a new instance represented using event-based and parallel event-based gateways etc depending on the conditions attached to the new instance. The gateways affect the sequence flow of input and output flows of a process/subprocess.

In chapter 3, I have established through empirical evaluation (of related literature) that not all the gateways utilised. Therefore, I will only be considering the gateways that are widely used such as exclusive, inclusive and parallel gateways. These gateways best fit with the scope of this research in providing an enumeration suitable to construct a complex business process representing time and resources associated with the achievement of a set or desired milestones.

4.3.4.1 Exclusive Gateway

Exclusive gateway assist modeller in representing decision behaviour using branching structure to select an outgoing flow depending on the conditions attached. An internal marker "X" in a diamond may or may not be used to graphically represent it. An example of a branching structure shown in figure 4.26 without an internal marker.

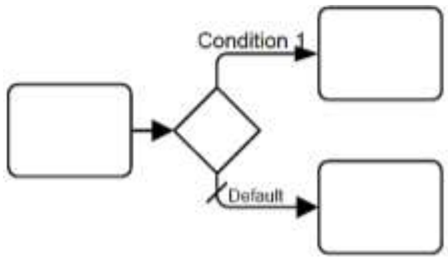


Figure 4 26 Exclusive gateway

In addition, the exclusive gateway used for converging the input flows by routing the token to output flow with no synchronisation. To show the importance of this gateway, an example is provided below.

Example 4.4: A process of credit risk evaluation [Weske, 2007] considered here and shown in figure 4.27. A task of evaluating the credit risk has triggered the process. Upon its completion, the exclusive gateway used to converge the output flows associated with conditions. The task of granting credit is activated upon the evaluation of the associated condition, i.e. low credit risk. If condition evaluated with an outcome of "medium credit risk" associated with the customers' credibility, then a sub-process (represented as collapsed) is initiated to carry out advance credit checks. If the evaluation of both the conditions is set to be false, then the third choice which is defined as default condition is undertaken, i.e. high credit risk, so reject the application.

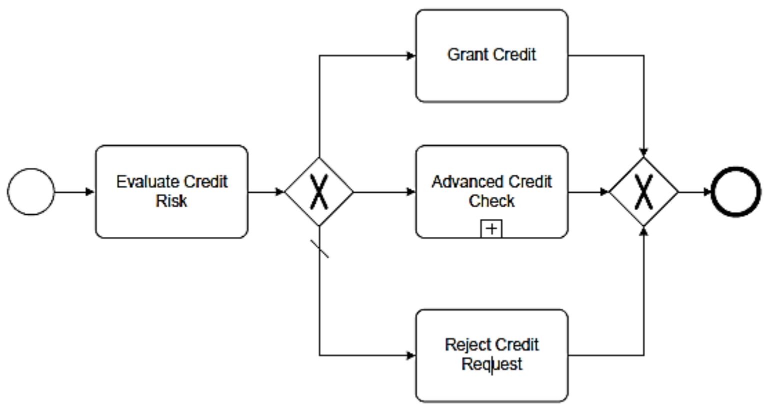


Figure 4 27 Sample business process with the exclusive gateway

Figure 4.27 shows the exclusive gateways using internal marker within the diamond construct.

Moreover, the BPMN standard supports the gateways and other graphical constructs with attributes. For most of the attributes are not graphically represented instead some rules are provided. For example, it is modellers' responsibility to specify the gateway conditions in the design ensuring every outflow is initiated. However, the exclusive gateway describes 'XOR' functionality similar to the decision control flow of the UML-AD. Like UML-AD merge, BPMN exclusive gateway takes all the incoming tokens without synchronisation. Therefore due to their similarity in functionality utilisation, I will be considering the exclusive gateway to be part of the enumeration to specify core modelling artefacts.

4.3.4.2 Inclusive gateway

The functioning of a diverging inclusive gateway is like an exclusive gateway; the only difference is that all conditions are evaluated. In the result of a correct evaluation of the requirements, a token is released to the output sequence flow that is ready to accept. However, all the paths are independent of each other, and a combination of zero to all tracks considered using the inclusive gateway. The default condition ensures that at least a path is chosen for the sequence flow. A circle is used as a marker within the general graphical construct of a gateway as shown in figure 4.28.

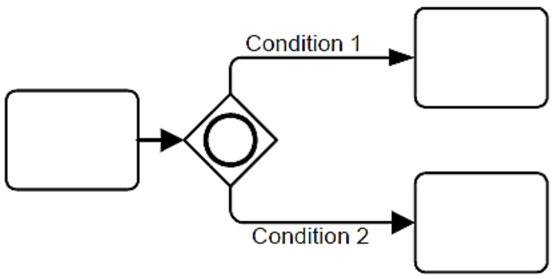


Figure 4 28 Inclusive gateway

Inclusive gateway evaluates the attached conditions in a sequence. Soon one of the attached conditions is assessed to be ‘true’ then the respective sequence flow would lead to a specific path by discarding others. The presence of the default condition ensures that in case all other conditions evaluated not true then it is traversed. The functionality of an inclusive gateway is like ‘OR split and in some cases provides synchronisation with the availability of token arriving late at the gateway, which is an exception though. Therefore, with the assumption of only using the inclusive gateway to describe the ‘OR split’ behaviour that is also possible with UML-AD using the fork (parallel split) where guards determine the branch coming out of the ‘fork’ requiring initialisation. Thus, on the basis of the functional similarity inclusive OR is selected to be part of the enumeration of core modelling artefacts.

4.3.4.3 Parallel Gateway

This gateway is used to represent concurrent flows with no condition evaluation provision resulting in a separate token passed to each outputs edge at the time of execution. Furthermore, it is used to combine parallel sequence flows for synchronisation. Graphically, a plus sign used in the diamond construct to represent the parallel gateway. The parallel gateway may be aligned to serve “AND split”, and “AND join” [Allweyer, 2016] representing concurrent behaviour of a process. The parallel gateway is shown in figure 4.29.

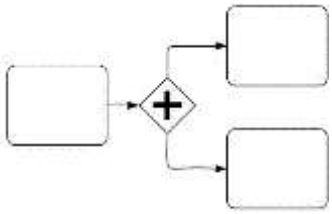


Figure 4 29 Parallel (Fork) gateway

In addition, synchronisation of two or more parallel paths represented by joining together the incoming flows expressing “AND Join” to produce an outflow shown in figure 4.30.

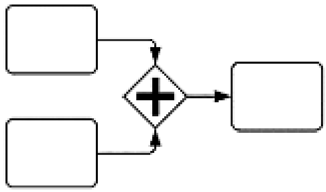


Figure 4 30 Parallel (Join) gateway

Example 4.5: An example of an order process [Dijkman et al., 2008] considered here which starts with the get order task. A parallel gateway is used triggering three parallel tasks, i.e. ‘AND split’ such as update inventory, ship goods and send an invoice. Upon receiving an update on the inventory a shipment of goods is made and subsequently an invoice is sent to the customer. Upon completion of each task, the ‘AND join’ synchronises all three to terminate the process as shown in figure 4.31.

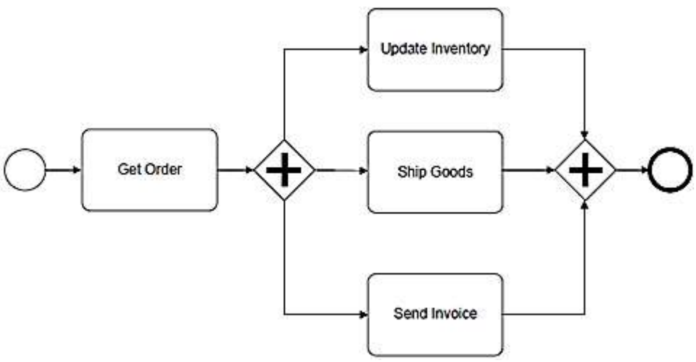


Figure 4 31 Example of a parallel gateway

The parallel gateway functionality is similar to the fork and joins control flows of UML-AD in specifying the concurrent behaviour. Therefore, I will be considering a parallel gateway for the enumeration of key modelling constructs.

4.3.5 Discussion on Other Modelling Artefacts

Unlike the parallel gateway, a complex gateway provides complex synchronisation having associated condition to describe the peculiar behaviour. However, like inclusive gateway divergence, the requirements related to the output flow decides upon the tokens to proceed with the chosen path, i.e. split. It also follows the synchronisation rule of the inclusive gateway when converging, i.e. join. An asterisk marker is used to represent the complex gateway as shown in figure 4.32.

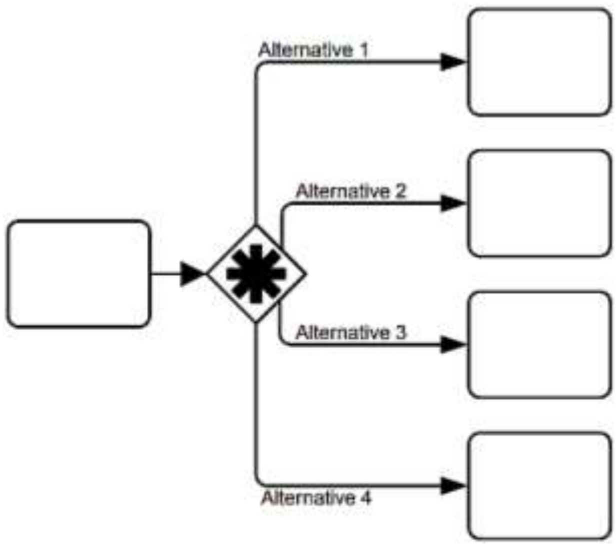


Figure 4 32 Complex gateway

A complex gateway depicts split with several outflows with conditions associated with each flow. It provides different choices to proceed with this gateway required to join. In addition, it may illustrate the validity of any pair of the sequence flows using an associated condition.

Moreover, BPMN offers controlling behaviour mechanism using events known as "event-based" gateways. The only difference between the "event-based" and other gateways (exclusive or inclusive) is that it triggers only when an event occurs to be evaluated rate than the attached condition. For example, an event can be a message received that triggers an outflow [Lano, 2009]. Though in the eyes of the author, their functioning is not different than each other (whether a condition or an event) because in either case a change occurs dues to their presence and the difference is in their description. Similarly, an event-based exclusive divergence (XOR) gateway triggers several tasks. If a task confirms the receipt of the message, the rest is ignored. These gateways not considered as necessary modelling artefacts (enumeration) due to their presence in other gateways that can depict the system behaviour with some descriptive modification and makes them irrelevant to the scope of this research.

BPMN relies on Pool to highlight collaborators within a business-to-business environment undertaking specific tasks [OMG 2013, pp305-308], which is similar to swimlane of UML-AD) as shown in figure 4.33.

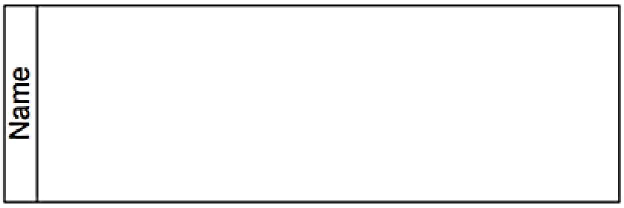


Figure 4 33 Pool

The only purpose 'pool' fulfils is the requirement of specifying the departmental units describing their names as a label to the construct. The 'pool' construct has no semantics provided in the standard documentation, therefore, it lacks expressivity when parallel activities occur. Because it fails to identify the specific time (qualitative and quantitative) order that confuses the stakeholders while interpreting which activity precedes than the other? In addition, it becomes more cumbersome when standard fails to meet the distinct needs of the industry to make provision for its semantics incorporating the boundaries of an organisation to satisfy the timely occurrence of the enterprise-wide activities communicating with each other. Instead of meeting the needs mentioned in this section, BPMN standard additionally offers another graphical construct called Lane used for the sub-partitioning of a pool by extending it as shown in figure 4.34.

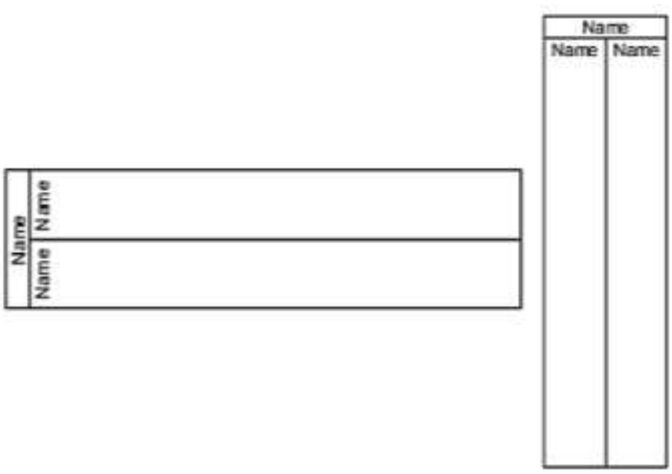


Figure 4 34 Lane

But, both Pool and Lane constructs only represent the departmental units with related roles (no quantification) responsible for carrying out assigned tasks within the organisation.

Overall, their functionality provides segmentation between the roles and units within the organisation but more importantly they do not change the comprehensive behaviour of the organisation. Therefore, keeping in mind the scope of this research requiring necessary modelling artefacts that are most often used to contribute towards the construction of a complex business process model. The above discussion emphasises the usage and suitability of the aforementioned constructs showing neither they are used more often nor their fitness in constructing a typical business process is of any benefit to the modellers. Therefore, I would not be considering them to assemble the enumeration of core modelling constructs.

4.4 Summary

The work in this thesis deals with the wider concept of business process, its sub-concepts and the flow. That requires identification of the fundamental terminologies to construct a complex business process. The current standards (presented in chapter 3 and 4) claimed that the concepts and their respective graphical representation present a comprehensive communication mechanism to express the behaviour for the sake of the stakeholders' ease. But, their redundant constructs make it cumbersome for the modellers to select an appropriate construct in constructing a consistent business process concerning time and resources.

Thus, there is a need for the communication channel which is precise with regards to its temporal reference in expressing the correct behaviour. Because the execution of a business process may lead to the creation of manifold processes and other relevant artefacts associated with the temporal reference. Therefore, a method is required that defines the constructs precisely and subsequently specify their distinct flow to represent a consistent business process.

Further to the review of the business process modelling standards (UML-AD and BPMN) conducted in chapter 3, I have examined various graphical constructs of both modelling standards for their suitability (concerning their temporal association). The Investigation suggests that modellers are at ease if provided with certain constructs accompanied with the precise specification which both modelling standards lack. However, the outcome of the investigation suggests (chapter 3) that both modelling standards are overwhelmed by unused constructs. Therefore, the important step required for the framework development is to enumerate modelling artefacts that would be necessary for the modelling of a typical process. This chapter facilitated in identifying the most often used modelling artefacts from both the process modelling standards comprising the enumeration (contribution to the knowledge). This step would also assist further (chapter 5) for their association with the temporal objects.

However, modellers of business processes from both the business and technical domain resorts to other modelling artefacts (of BPMN and UML-AD) but find they are not frequently used and not relevant to construct a typical business process. These constructs are mentioned in this chapter and provided a discussion to justify the decision made by the author of this thesis. Furthermore, the argument supported by a few examples and empirical comparison of the modelling paradigms to provide a certain number of modelling artefacts sufficient enough to specify a complex business process generally suit all the domains including healthcare.

With the identification of the most often used artefacts of UML-AD and BPMN, I have managed to lay down the premises for the formation of enumeration (a specific set of terminologies and constructs). It has been noted that the core constructs identified here from both the standards describe the similar ontology and functionality having no formal semantics. Therefore, it is considered vital for both industries (business and technical) serving the same purpose of an enterprise (which is business process modelling), to have a consensus on the terminologies, their unified ontology (formal semantics) and precise graphical representation. These requirements combining together would provide a unique platform in constructing precise business process models and increase the profitability for the organisations of both the domains.

After the identification of the key concepts, I need to establish their associated temporal reference. It would make provision for their association with the temporal objects that bear a concise definition and would assist in representing the correct behaviour. The efforts made in this research focused on the industry needs requiring a framework providing precise semantics of modelling artefacts fulfilling the requirements of modellers from both business and IT fields. With the approach described here would facilitate modellers to have no confusion in making a distinct choice of the construct that have precise meaning improving their understanding of the artefacts used. Furthermore, models constructed as result would enhance their communication supported with associated temporal reference. In entirety, it could enhance the enterprise’ ability to compete better in the competitive business world.

Chapter 5 Framework – Phase I

Distinctively, the second generation of the business process modelling has emphasised upon the necessity for logical foundation [Van der Aalst et al., 2003]. Where a logical foundation can lay down a path in developing a formal method that can provide precise definitions, i.e. semantics, of the terms used in the business process (general) and healthcare (specifically) modelling. Such a method would serve as a framework by answering all the research questions specified in chapter 1 supported by the discussion and evaluation provided in chapters 2, 3 and 4. So far the investigation has established that the enumeration comprising of modelling artefacts require some sort of temporal reference while constructing a business process making provision for effective communication between the stakeholders. Therefore, the next step is to explore the different approaches presented in the field of temporal logic to select a suitable class to assist with the development of the framework.

To choose a specific class of temporal logic, one needs to understand the concept of Time first. Because 'Time' as a concept has a pervasive role in referencing for frequent and general usage. Especially, organisations require to represent the timely order of activities while modelling that could further assist in controlling the business processes involved in the operation for enhanced reasoning. Due to its importance, domains such as technical (information systems), management sciences, cognitive science, linguistics, philosophy and history have deeply integrated the idea of time.

Therefore, 'time' as a concept should be defined clearly for its use in real-life. Keeping this in mind, [Findlay, 1941] and [Prior, 1955] cited in the literature making initial efforts to formalise time. After that, many researchers have made several attempts on problems such as regional anatomy, and partial correlation concerning time, and temporal representation of the activities in a model. However, since the early 1970s, several approaches have been suggested in the literature to deal with the issue of modelling real-life scenarios based on temporal concepts. These approaches are mainly isomorphic because they provide slight changes in the concepts used and their corresponding operation.

The domains such as artificial intelligence and many others have cited the application of temporal knowledge, reasoning and representation in [Bruce, 1972], [McDermott, 1982], [Shoham, 1987], [Peter, 1992], [Freksa, 1992] and databases in [Maiocchi et. al., 1992], [Ma and Knight, 1994] and [Bassiouni et. al., 1994], program verification on [Manna and Pnueli, 1995], software requirement specification languages in [Mylopoulos et.al., 1990], [Tuzhilin, 1995] and [Jungclaus et. al., 1996] and system modelling and scheduling in [Zaidi, 1999]. The domains that require continuous conformance and examination of their systems and respective functionalities then the related temporal information can be very significant in determining their correct behaviour.

In addition, the evaluation of the temporal facts within business process modelling domain concerning process design and development and its verification is of great importance. Such temporal approach may facilitate operational scrutiny including world wide web where system overseeing (business functions such as scanning) false activities detection and stoppage etc. Furthermore, an investigation concerning temporal facts is crucial to provide comprehension of conceptual processes and corresponding communication between the stakeholders [Zaidi and Wagenhals 2006]. However, temporal facts can also play a pivotal role in displaying the business process and incorporating activities in a timely fashion to achieve a consistent model.

Usually, both modelling standards rely on temporal reference associated with temporal intervals to establish the footprints of the activities to build a business process model. But the modelling standards are intuitive so do not utilise these timed footprints for examination and analysis of the operations. In addition, the temporal reference concerning only the temporal object 'interval' would be discussed later in detail for its suitability and relevance to the scope of this research.

The above discussion indicates that there are more temporal objects that can be utilised for temporal referencing. Thus an appropriate class of temporal logic is required to determine the temporal objects that can precisely represent the associated worldly domains such as business process modelling and healthcare modelling. A variety of classes belonging to temporal logic present different temporal objects such as interval, moment and point. But, not all temporal theories use all of these objects together, therefore, a comparison is required for their significance. In addition, such approach can further assist in associating the temporal objects with the real-life activities and finding the relationships between them for determining a precise process model. Hence, it is paramount to use temporal objects that can sufficiently describe the business process and its sub-components best. For example, in real-life, initiation of a business process can trigger an order of multiple sub-processes. And a sub-process further comprised of coordinated sub-elements.

To describe the objects of a real-life domain including business process modelling, their ontology considered vital since it provides their semantics and syntax. Hence, it is required to define the clear ontology of real-life concepts used in the business process modelling domain. Another critical point to consider while describing them is their interrelationships and the associated temporal reference. By doing so, one can address the following

- To establish the structure and boundaries of the real-life concepts such as process and
- The timely occurrences of the events/processes and their temporal relationship, for example, two processes A and B, occurs either before, after or equal etc.
- To determine the occurrences of related events and/or processes during the flow.
- Whether their duration is known or not.

The above discussion has laid the ground to explore further the different classes of the temporal logic before deciding on their suitability for utilisation in this research.

5.1 Point Temporal Logic

The point temporal logic, as its name implies, considered describing the real-life objects in the form of temporal points expressed as (*P,* ≤). Whereas “P” comprised of a collection of points and the less than equal to sign (≤) describes the partial or total order relations between them. The systems developed based on this class have derived the temporal intervals to represent the relations and corresponding order of the points [McDermott, 1982], [Van Benthem, 1983], [Bruce, 1972], [Shoham, 1987] and [Ladkin, 1992]. In addition, the algebraic representation used for the computation of the points by [Gerevini and Schubert, 1995] and [Drakengren and Jonsson, 1997].

5.1.1 Issues

Point temporal logic considers only time point as a primitive object and not the interval. To develop systems based on this class of temporal logic require distinct quantitative time-related information. Furthermore, the qualitative representation representing constraints between the points are quite limited to provide a coherent business process model. Thus, its application is limited in real-life scenarios such as the healthcare sector where not all the required information is sufficiently known or available. To express the limitation of point temporal theory, if I consider two events X and Y with no information about their starting and completing times and no knowledge of other occurrences between them. The only information that is available is about a specific incident which establishes that X occurs before Y. This insufficient knowledge would make the construction of a system based on point-based temporal theory very difficult though inferences can be made of qualitative temporal information but it is limited and therefore present an incomplete depiction. Furthermore, the interval is not considered as primitive part of the temporal objects set making it insufficiently equipped to describe real-life objects. In real life systems, the availability of complete absolute temporal information and comprehensive qualitative constraints' set often unavailable. Such temporal information is vital for consistent representation of the system that further utilised for reasoning purposes, makes the point temporal theory not suitable for this research.

Moreover, "dividing instant problem" is a significant issue with this class of temporal logic that is reported in [Van Benthem, 1983], [Allen and Hayes, 1989], [Vila, 1994] and [Ma and Knight, 2003]. This problem arises while deriving the extremes of two consecutive intervals in which some instantaneous activity occurs. Similarly, point temporal theory lacks to support real-life system development due to uncertainty and missing information relating to temporal order such as start or end event times. For example, it is failed to represent a scenario where patient one was assessed before patient two arrived or during the diagnosis of patient one patient two waited for two hours etc.

5.2 Interval Temporal Logic

[Allen, 1983] presented temporal interval theory based thirteen temporal relations (before, starts, equal, meets, during, overlaps, finishes, after, started by, met-by, contains, overlapped by and finished by). This class of the temporal logic only considers time interval as a primitive object and time point discarded for the representation of a system. Time interval as a natural object can facilitate temporal point representation as a common intersection of interval extremes [Ma, 2007]. Hence, the interval temporal theory considered to have specific characteristics that could present it as a substitute for temporal logic based on time point.

The only common attribute between both theories (point temporal theory and interval temporal theory) is that they both use their chosen temporal objects individually and separately to specify more delicate details of a system. Comparatively, temporal object interval show increased expressiveness than the temporal point in representing temporal order of real-life scenarios. For instance, a consultant was attending a patient for the first half of the day.

5.2.1 Issues

Interval-based theory handled the matter of “Dividing Instant Problem” with the thirteen temporal relations and overthrown the problems in modelling temporal object [Allen, 1984], [Van Benthem, 1983] and [Ma and Knight, 2003]. However, the temporal object “point” relegated while modelling due to its insignificance in the interval based temporal structure. Hence, the interval based time structure mainly based on the ontology of time interval and widely used for its increased expressiveness feature.

With some advantages, the interval-based theory, there are some limitations in representing un-interrupted variations within a system. The interval-based approach also lacks in providing the precise meaning of all the temporal objects available that can be used in designing and describing an accurate system [Galton, 1990]. Because it does not accommodate a precise time point attached to the interval as a primitive. In real-world, precise representation of a point (temporal) can be of the utmost value for describing the business process correctly. Hence, this class can cause serious issues while modelling complex processes. Therefore, the interval-based theory would not be a choice for this study.

5.3 Point Interval Temporal Logic

The point and interval-based temporal theories have limitations in presenting only limited temporal facts associated with the chosen temporal objects. Therefore, researchers such as in [Villain and Kautz, 1986] considered both point and interval as a specific class of point temporal logic. That was comprised of all the necessary and manageable point (temporal) relationships but discarding the temporal interval as the fundamental component of the temporal structure. And if both temporal objects considered as primitive then a wide range of temporal facts can be derived from them providing a comprehensive constraints network to build a consistent system.

Therefore, a need to combine these theories seemed eminent to the researchers working in the field of temporal logic and its applications. That resulted in the development of another class of temporal logic in the 1990s known as point interval temporal logic (PITL) presented by [Ma and Knight, 1994] and [Zaidi, 1999]. They bridged the research gap and considered point, interval and both point and interval primitive objects of the temporal theory to represent consistently the real-life systems.

PITL may be more beneficial compared to other two temporal theories for analysts in facilitating them to describe, design, develop and analyse the real-life concurrent systems. With the incorporation of temporal point and interval having precise structures to specify their properties such that a temporal point only occurs associated with a specific temporal interval. For example, a process can comprise of several sub-processes that may span over some duration inclusive of unique temporal point occurrences to specify events bearing no physical time, i.e. zero length.

5.3.1 Reasons to choose PITL

Computer science literature present well-established discussions relating to temporal objects considered as primitives particularly in the domain of the system's knowledge management and its representation. That has led to several temporal theories (explained in the previous sub-sections) considered by the industry and researchers to describe the temporal objects associated with natural phenomenon including processes and events.

The business process modelling standards vaguely use the interval temporal theory representing the activities involved in constructing a model making them insufficient to express details of a system. Because they do not facilitate their modelling artefacts with the comprehensive set of temporal objects precisely for modelling a business process comprised of sub-parts. To address such issue, a variety of formalisms developed establishing the temporal facts to provide reasoning about the processes and events.

However, the chosen PITL has enough temporal objects providing a necessary and sufficient set of constraints to determine temporal facts associated with the model. By doing so, I could set a foundation to develop a novel framework comprised of a methodical approach to answer the research questions stated in chapter 1 by identifying and defining the primitive objects (enumeration). Moreover, these temporal objects can be associated with the real-life business process modelling artefacts depicting a range of temporal relations between them to construct a correct model.

In addition, the chosen PITL for this research would be presenting the characteristics of the semi-open interval (a class of PITL) in which the time interval "A" constrained by all the feasible temporal relations between the primitive start point "sA" and an endpoint "eA". Furthermore, considering two intervals "A" and "B" within PITL having temporal relations "R" used as constraints between them, i.e. $AR_{ab}B$, where $R_{ab} = \{R_1, R_2, \ldots, R_n\}$; and in general, R_i considered to be the disjunction of relations written as $[(A\, R_1\, B) \vee (A\, R_2\, B) \vee \ldots \vee (A\, R_n\, B)]$.

Hence, a temporal relation $R_i \in R_{ab}$ is viable if and only if there exists a resolution which is comprised of logical temporal relationship between them. Similarly, it illustrates the precise semantics of the concerned objects. Another reason for making PITL to be used in this research is its capability in describing general concepts that may be useful to provide an ontology for the real-life modelling techniques.

Moreover, the enumeration provided in chapter 4 presents a firm bond with the PITL where every modelling artefact can be associated with either of the temporal objects to be represented with previse structure. For example, a modelling artefact either can have some duration with breakable structure (representing interval) such as process and sub-process, or occurrence of a subtle modelling artefact representing no internal structure (representing point) such as event. The distinct organisation of the modelling artefacts (if formally defined) is possible to build a consistent process model. In addition, the constructed model has the capability to be utilised for further analysis providing a mechanism for the modellers to reason about the developed system including finding errors and removing them.

The discussion above has established the need for primitive elements to represent a viable resolution. Those objects, if appropriately enumerated then they either be utilised to describe semantics or carry out semantic checks on a model. It may well suit the needs of the business process modelling domain because the standards available are ontological redundant and require a necessary set of the artefacts that have formal semantics. Additionally, analysts can ascertain the required behaviour required of a system enabling its correct representation and providing enhanced reasoning. So far, no such mechanism is provided for the solution to these problems. Therefore, I have chosen the PITL for this research for the benefits it provides in constructing a comprehensive model making provisions for its further analysis that existing modelling standards lack. In the next sub-section, a list of temporal objects provided that would be associated with generic terminology introduced in the axiomatic system representing an abstract business process.

5.3.2 Temporal objects

Discussion in the previous sub-section clarifies that treating point, interval, and point-interval both as primitives (PITL) considered vital in resolving the real-life business process modelling problems. Because, on the one hand, a temporal object such as point considered as primitive by [McDermott, 1982] required for both theoretical and practical modelling. And due to its innate nature that fits well with instantaneous activities termed as events by the modelling paradigm, i.e., BPMN, e.g. patient discharged at 11:30 am. On the other hand, the only interval considered as primitive by [Allen 1983] required for representing non-instantaneous activities, e.g., “patient is being assessed” etc. Due to their varied alignment with the real-world phenomenon to satisfy the limited needs of the logician and practitioners concerning a domain to model and reason about the desired system. Thus, both temporal point and temporal interval as primitive are significantly important in representing the different elements and their corresponding relations of a business process and/or healthcare patient flows. More importantly, these temporal objects with precise description would assist modellers to capture time perspectives of the constructed model (a prime requirement of this study).

In addition to point and interval as temporal objects, Allen considered a time ‘moment’ [Allen and Hayes, 1989] as an alternative to the time point having some positive duration. They considered it indivisible and attached limitations such as it could not meet other moments. However, [Galton, 1990] and [Ma and knight, 1994] reviewed their proposition and suggested their views of handling moment within a model development. However, [Zaidi, 1999] used a relation instead of considering ‘moment’ as a temporal object which is primitive, making his approach insufficient to this study. For this research, I would consider ‘moment’ as a primitive temporal object as part of the enumeration. The features attached with moment are that it is indivisible and can meet other ‘moments’, intervals or points. By doing so, I have extended the temporal theory presented by [Zaidi, 1999] and (contribution to the knowledge). Thus, the overall temporal objects selected for this research characterised as primitive and described below:

- Temporal point considered primitive to represent instantaneous activities distinctly with zero duration.
- Temporal interval (considered primitive) uniquely representing activities with a duration that can be further divisible, and
- Temporal moment treated a primitive and considered a subclass of interval representing activities with positive duration but unbreakable.
- The temporal point, interval and moment as primitive

These objects present a wide variety of relationships describing constraints to specify the flow between them. Although uncertainty exists within the real-life domains and modelling standards only rely on the temporal object interval including its (limited) thirteen relations to model such situations insufficiently displaying the required characteristics such as the structure of a process A is not fully defined with its start and endpoints concerning its parallel processes B and C to proceed with. In addition, the utilisation of full interval theory is not present within both standards' documentation. However, with the assistance of PITL temporal objects (presented above) and its inference mechanism suggests several possibilities that can be drawn to construct a consistent model depicting the correct behaviour. Therefore, these constituents provide the necessary set of elements required for the development of the axiomatic system (framework phase II).

Moreover, these objects supported by the logical meaning and if associated with the business process modelling most often used artefacts (chapter 4) can further provide their formal semantics and unify them under one umbrella. Now, I would define the extended PITL temporal objects below.

Definition 5.1 - Time Point: The temporal point "p" expressing zero duration that can be written as [p].

Definition 5.2 - Moment: A temporal object 'moment' expressed as unbreakable but semi-open interval bearing some positive duration bounded by start and endpoints. A moment can be expressed as a symbol "A" with its extreme points ([sA, eA]), i.e. sA < eA, where prefixes' and 'e' denotes start and endpoints of the moment 'A' with no other points or interval occurring in between.

Definition 5.3 - Interval: An interval is defined as decomposable and bounded by time points at both extremes (non-instantaneous). It is expressed in the same manner as moment [sA, eA]. To describe the decomposition an interval "A" with n numbers of moments written as the disjunction of moments [$a_l \vee a_2 \vee ...a_n$].

Definition 5.4 - Duration: In representing the absolute and relative information relating to a point, moment and interval, I have defined a "Dur" function, given below.

a) The duration "Dur" function calculates the duration of the point, moment and interval. In case of temporal point, "Dur" allocates positive real number including zero represented as $\boldsymbol{R^+}$ to establish the stamp. This function would assist in determining the lower and upper boundaries associated with the temporal point. To express the situations such as no later than and no earlier than written as pX ≥ Dur [pX] and pX ≤ Dur [pX], where Dur [pX] ∈ $\boldsymbol{R^+}$, where 'p' represents a time point.
b) To represent the interval and/or moment duration, "Dur" function allocates a positive real number excluding zero. For example, duration of interval 'A' represented as Dur [A], where Dur [A] = [sA, eA] and Dur [A] ∈ $\boldsymbol{R}$, and Dur [A] = eA – sA. Additionally, to express the lower and upper limits associated with the

interval 'A' denoted as "at least" written as "≥" and "at most" written "≤" respectively.

c) To represent the duration of sub-intervals of an interval 'A' can be written as $A_1, A_2, \ldots A_n$, whereas Dur $[A_1] = [t_1 – sA]$, Dur $[A_2] = [t_2 - t_1]$,..Dur $[A_n] = [eA-t_{n-1}]$. Therefore, the total duration would greater than zero but less than the duration of the parent interval.

Where $\boldsymbol{R^+}$ describe a subset of the real numbers $\boldsymbol{R}$ representing interval/moment (that is $\boldsymbol{R^+} \subset \boldsymbol{R}$). That may assist in incorporating absolute and relative temporal information within a system to determine the process behaviour precisely. In the absence of quantitative temporal information, their qualitative occurrence used to express their comprehensive relationships. The relations between these objects need to be specified that serve as constraints between them for their consistent execution. PITL inference mechanism facilitates in deriving a set of relations that assist in representing a variety of relationships between two temporal objects.

Definition 5.5 – Point - Point Relations: To represent point - point relations within PITL expressing the set of temporal relationships between two points that are [before, equal, after]. For example, point A and point B can be represented as [pA] and [pB] respectively and expressed on a single timeline as pA = pB, that can be written as [pA, pB].

Definition 5.6 - Interval - Interval Relations (R): A set of temporal relations represented as R = [before, after, equal, meet, met-by, starts, started-by, during, contains, overlaps, overlapped-by, finishes, finished-by]. These relations exist between two intervals and may also be sub-intervals and/or moments.

Definition 5.7 - Point–interval relations: A set of point-interval temporal relations comprised of [before, after, meets, starts, met by, during, finish] temporal relations between point and an interval.

Definition 5.8 – Interval - Point relations: A set of interval-point temporal relations [before, started-by, meets, after, met-by, contains, finished-by] represent relations between an interval and a point. Figure 5.1 below represents comprehensively mutually exclusive qualitative relations to illustrate the structure and semantics of extended PITL, which means if

a) a temporal relation R_i exists (belongs to the set R) then R does not contain any other temporal relation (R_j);

b) Also, a relations R_i exists between A and B (intervals) such that either A R_i B or B R_i A is true only if (A ⇔ B).

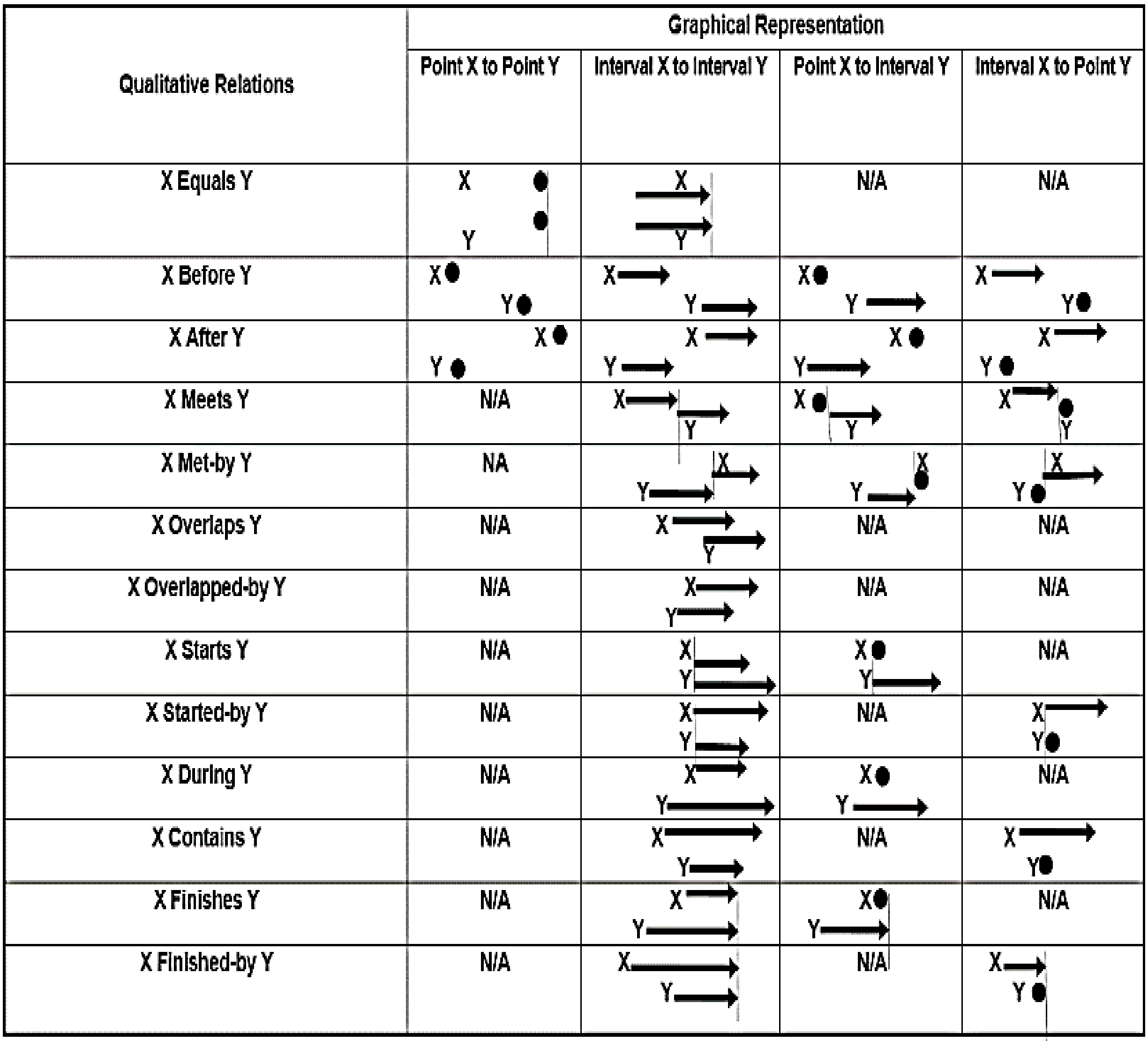


Figure 5 1 Extended PITL relationships

PITL proposed by [Zaidi, 1999] extended here by including an extra set of temporal relations, i.e., interval-point as defined above and therefore require to be supported by additional formalism presented below.

5.3.3 Interval-Point Formalism

[Zaidi, 1999] has only provided the formalism for a) interval-interval, b) point-point and c) point-interval. As he did not consider the interval-point relationship and therefore no formalism provided to express possibilities of the interval-point relationships. Therefore, as part of the extended PITL (contribution to the knowledge), I am equipping the knowledge base with extended relationships providing aid to the existing inference mechanism for the increased expressiveness. The formalism for the fourth set of temporal relations between interval-point provided in table 5.1.

X is an interval [sX, eX] and Y is a point [pY]	
D 1: sX < pY → eX > pY	**D 6:** eX < pY → sX < pY
D 2: sX = pY → eX > pY	**D 7:** eX = pY → sX < pY
D 3: sX ≤ pY → eX > pY	**D 8:** eX ≤ pY → sX< pY
D 4: sX > pY → eX > pY	**D 9:** eX > pY → sX < pY
D 5: sX ≥ pY → eX > Py	**D 10:** eX = pY → sX < pY
D 11: eX ≥ pY → sX < pY	

Table 5 1 Interval-Point formalism

The prefix ‘D’ of all the possibilities (formalism) represents its fourth place in the PITL presented by [Zaidi, 1999]. Additionally, the above formalism will assist in deriving extended relations between interval and point (contribution to the knowledge) given below in table 5.2.

sX Vs pY	**eX Vs pY**	**X R Y**
<	**>**	**d^{-1}**
<	**<**	**<**
<	**=**	**m, f^{-1}**
<	**≤**	**<,m, f^{-1}**
=	**>**	**s^{-1},m^{-1}**
≤	**>**	**$s^{-1}$$m^{-1}$, d^{-1}**
>	**>**	**>**
>	≥	**>, d^{-1}, m, f^{-1}**
≥	**>**	**>, s^{-1},m^{-1}**
?	**<**	**<**
?	**≤**	**<,m,f^{-1}**
?	**>**	**>, d^{-1}**
?	**≥**	**>, d^{-1}, m^{-1}, f^{-1}**
<	**?**	**<, m, d^{-1}, f^{-1}**
≤	**?**	**$s^{-1}$$m^{-1}$, d^{-1}**
>	**?**	**>, m, $d^{-1}$$f^{-1}$**
≥	**?**	**>, s^{-1},m^{-1}**
?	**?**	**<$d^{-1}$$f^{-1}$m,$m^{-1}$ s^{-1}, >**

Table 5 2 Analytical representation of interval point relationships

5.3.4 Properties of Relations

The qualitative temporal relations between any two intervals/moments do not depend on their absolute duration instead rely on their relative occurrence. The relative presence of the interval/moment may assist in deriving other relationships from systems' temporal statements. However, quantitative information (if available) may help in computation, e.g., an interval A duration can be computed from the specific relations between A and the other related intervals. Additionally, absolute and relative temporal information can assist in finding inconsistent ties between the occurring objects and other problems within a system description. Other properties of the interval-interval relations are given in table 5.3:

Relation	Reflexive	Symmetric	Transitive
Before	-	-	+
Meets	-	-	-
Overlaps	-	-	+
Starts	-	-	+
During	-	-	+
Finishes	-	-	+
Equal	+	+	+

Table 5 3 Properties of set R (Interval - Interval temporal relations)

"-" represents the non-presence of the property while "+" represents the existence of the property.

5.4 Inference mechanism based on extended PITL

To show the impact of the extended PITL on the existing inference mechanism provided by the original PITL, I consider an example. Suppose two intervals A and B having a relation between them and upon discovering a new temporal connection between intervals B and C, then with the help of the two temporal relationships of three intervals, the inference engine would assist in deriving a new connection between them, i.e. transitivity. Also, the inference mechanism would establish uncertain relations from the systems' intervals A and C combinatorically. The following example illustrates this issue:

Example 5.1: Consider a model described using natural language:

- An interval A_1 Meets an interval A_2 and also an interval A_3
- An interval A_2 Meets an interval A_4, and interval A_4 Finishes an interval A_3

The inference mechanism derives temporal relations between A_1 and A_4. There are two possible choices available to find the unknown temporal ties;

a) Known associations such as A_1 Meets A_2 and A_2 Meets A_4 can assist in a possible derivation of temporal relations between A_1 and A_4;
b) Also, relations such as A_1 Before A_3 and A_4 Finishes A_3 can help in deriving the relationships between A_1 and A_4. These choices would infer the non-conflicting relation which is A_1 Before A_4.

The inference mechanism helps in establishing the possible combination of relations (uncertain) by searching all the known combinations that produce an outcome following more than one-step as exhibited in the above example. To express this ($Y\ R_1\ I_1$), ($I_1\ R_2\ I_2$),……...($I_i\ R_i\ I_j$),…..($I_n\ R_n\ Z$). It would constitute several derivative relations from Y and Z to infer the uncertain relation. Also, the inference process requires the exhaustive calculation of all possibilities by searching all the combination of known relations (viable incorporation) that yields the result. However, some may suggest that a thorough search is not required; instead, it should be stopped soon the possible relation is derived, but with this approach only applied to the system with consistent priori.

5.5 Summary

The findings of the chapter 1-4 emphasised upon the necessary enumerated artefacts (that are logically grounded). Furthermore, the most often used modelling artefacts (enumerated objects) of UML-AD and BPMN required formal semantics due to imprecise structure described in the respective standard documentations. That is possible with the use of a suitable logic making provisions for clear and concise descriptions of identified modelling artefacts aiding the modelling paradigms towards their formalisation.

A systematic approach adopted to fill the research gap by developing a state of the art framework in two phases where phase I establishes the need of logic that is comprised of lexicons serving as the necessary set of formally defined artefacts, i.e., precise meaning. Furthermore, these logic-based objects have the power to be used for association with the real-life modelling artefacts in an attempt to distinctly model the business processes.

To choose a relevant logic, I have examined the different classes of temporal logic adopted by the researchers and practitioners for their suitability and relevance to this study so that an innovative and state of the art framework could be developed to fill the research gap presented in chapter 1. The choice of temporal logic made by the author of this study by keeping in mind its usability within the real-life domain. The investigation of the varied classes of temporal logic unveils the precise ontology of different temporal objects used by the different temporal theories. In addition, these theories (point temporal logic, interval temporal logic and point and interval temporal logic) concerning their relevant temporal objects and applications compared for their suitability to the overall development of a sound system and also examined their relevance to this study.

In the development of phase I of this research required a solid foundation to identify and describe the core lexicons (necessary set). Therefore, the comparison of the temporal theories conducted to explore the importance of the temporal objects present in the different temporal theories. On hand point temporal logic focuses only on ‘temporal point and on the other hand interval temporal logic provides interval as its core element to build systems. That leaves a room for the researchers’ to come up with a class of temporal logic that considers both temporal objects (point and interval) at equal footing. The class of temporal logic treating both point and interval as primitives presented by [Ma and Knight, 1994] and [Zaidi, 1999]. I have chosen the PITL presented by [Zaidi, 1999] for its added graphical support provided to diagrammatically represent the lexicons. However, it needed to be extended to be used in this study.

if the chosen PITL extended then it has all the necessary ammunition to be used for the purpose identified to develop the phase I of the framework. Therefore, I have extended the PITL by defining lexicons such as point and interval with corresponding temporal relations. In this way, I have contributed to the existing knowledge in the shape of extending the existing PITL of [Zaidi, 1999].

Another reason to extend PITL of [Zaidi, 1999] is that he has considered only point and interval but the moment is ignored. However, moment has the practical importance in the real-life modelling domain to define the semantics (formal) of certain terminologies such as action of UML-AD and task of BPMN representing unbreakable activities. The lexicons of the PITL (temporal theory) adopted here would determine the ontology of the most often used terminologies of both UML-AD and BPMN unifying both the modelling standards.

In addition, as a contribution to the knowledge only three relations between point and interval considered, i.e. point-point, interval-interval and point-interval, by [Zaidi, 1999]. There is a possibility of fourth set of temporal relation between interval-point. Which was not presented by the Zaidi’s PITL, however, I have extended the PITL by providing such set of temporal relations. Furthermore, the specified set supported by the formalism to identify the possibility of varied relationships between the primitive temporal objects. By extending, the temporal relationships between the temporal objects would increase the current constraints network (inference mechanism) to derive more possibilities between temporal objects. Thus, I have added interval-point relation and provided formalism to give a comprehensive knowledge base for phase I of the framework.

The knowledge base presented here in this chapter has the power to associate with the real-life modelling artefacts chosen for this study from the current modelling techniques to provide their formal semantics for their correct representation. Moreover, the inference mechanism of the temporal theory adopted here would assist in deriving relationships (unknown) could be vital for solving real-life issues. The current modelling standards do not consider such an approach that makes them less attractive for the industry unless they make drastic changes to their standard documentation.

This chapter provided the logical foundation that would help in defining the axiomatic system (phase II of the framework development) presented in the next chapter. These lexicons have the power to define unique but general terminologies that can be used in the real-life modelling replacing the existing most often used terminologies (of UML-AD and BPMN) laying down a path towards their unification.

With the provision of clear semantics, a precise structure can be defined that eventually assist in the model verification. However, these approaches are lacking in providing a mechanism to verify the models developed using their notation. Verification is pivotal to determine the correct structure of the business process model. In phase II of the framework discussed in next chapter, I have developed an axiomatic system that would benefit from the logical foundation provided in this chapter to further develop the method equipping framework with analytical support to evaluate business process making it attractive to the industry.

.

Chapter 6 Framework – Phase II

One of the reasons that attracted industry to utilise the current business process modelling standards is their conceptual modelling schema. That is easy to use and flexible to be blended within a wide range of domains. UML-AD serving technical domain and BPMN used by business domain experts supported by their respective standard documentations. That is comprised of a variety of modelling terms, however there are most often used terminologies (identified in chapter 4) of both the standards bear similar intuitive descriptions used to construct the business process model.

In addition, the modellers of both domains utilise the most often used terminologies but vaguely express the order (based on time) of business process model artefacts to represent a finished product or service. However, both domain experts share the same objective of modelling the business functionality relying on the (different) terminologies bearing the similar onology (intuitive). For example, UML-AD uses the term 'Action', and BPMN uses the term 'Task' to represent the fundamental unit of work expressing no internal structure and their boundaries are not specified. In addition, term 'activity' used by UML-AD and term 'process' used by BPMN to characterise a business process, again lacks in expressing corresponding structure of individual actions/tasks involved with respect to their boundaries to establish the orderly flow with respect to the associated qualitative and quantitative time. Furthermore, control flow and gateways are used by UML-AD and BPMN respectively to represent and maintain the concurrent flow within a model. However, they failed to display a consistent flow of a complex process presenting variations within the system specifications.

The reason behind their ambiguous representation in both standards' constructed models is their intuitive descriptions of their huge number of vocabularies. That makes it cumbersome even for OMG to formalise such a massive number of constructs. In addition, due to lack of use of all the available constructs of both standards make them ontological redundant and no adequate enumeration of modelling artefacts provided for unambiguous process representation.

Moreover, to provide clear semantics for the concepts used in system development, researchers adopted different approaches to defining ontology of a range of terminologies in a wide variety of ways such as lexicons and vocabulary etc. Even some researchers' have considered first-order logic to provide semantics for the chosen objects to meet the needs of a specific domain. All these approaches have something in common and that is the development of a knowledge base comprised of the terms with the standardised ontologies belonging to a domain. Hence, industry needs can be met by providing a standardised method composed of formalised artefacts. Based on the premises provided in this section, I would present the axiomatic system development in the next sub-section.

6.1 Axiomatic System

As we know from chapter 2 and 3 that the vocabularies (too many and intuitive) documented in the business process modelling standards. That further scrutinised (in chapter 4) to narrow down the number of terminologies and constructs based on their utilisation (frequency of use) and relevance in constructing a general business process. Therefore, it is considered pivotal to provide a list of generic terminologies that can combine the most often used terminologies of UML-AD and BPMN (chapter 4). Furthermore, first-order logic can be used to make provision for the formal semantics of the generic terminologies to develop the axiomatic system.

In addition, first-order logic facilitated by its inference mechanism providing an edge on other methods to describe the precise ontology of the artefacts concerning a domain. Similarly, these formalised artefacts further support the construction of a correct model by establishing many possibilities (consistent relations) between them through its derivation mechanism. Accordingly, with the help of inference mechanism modellers can examine the different derived relations between the modelling artefacts for any inconsistency that may present in the model constructed. Other benefits of the first-order logic include maintaining the enumeration of concepts belonging to a domain (whether it's business or healthcare modelling) supported by the algorithms to further schedule the activities involved and provide enhanced reasoning concerning a business process model [Chishti et al., 2017]. However, first-order logic can also be used along with the temporal logic that accommodates the primitive temporal objects to develop a method providing correct modelling. This combination of logics further assists in the analysis of the system behaviour for its verification and validation.

For this reason, I have employed both logics to develop the axiomatics system by introducing the enumeration based on temporal basis and precisely defining them using first-order logic overcoming the problems (intuitive structure) faced by the current modelling standards. In this way logical basis for the modelling standards can be provisioned and fills the research gap (chapter 1).

Moreover, in the phase II development of the framework, I would rely on 'model theory' of [Hilbert, 1909] capturing fundamental structures of the enumerated modelling artefacts of this study. Bear in mind, the ontology of the term 'model' is the fundamental difference between the model-theoretic viewpoint and the practical modelling approaches (business process modelling standards). Model-theory considers the term 'model' as a development of a theory while business process modelling techniques consider the term 'model' to represent a system architecture graphically modelled in UML-AD and BPMN. For example, in the model-theoretic approach, UML-AD and BPMN model considered as the theory and its' specific instantiation considered as model.

In the practical domain, current modelling standards consider model as the diagrammatic representation that can trigger different instances of the model (business processes) involved. Therefore, the model functionality is isomorphic to the theory (of model theory) describing the fundamental components and their behaviour as part of the enactment. Model theory method provides bifold mechanism introducing the abstract version of the model with the help of enumerated modelling artefacts and further instantiates them for their real-life implementation. In this way the abstract modelling artefacts conceived and described within theory and further used them for corresponding instantiation to depict a distinct instance of the model. Additionally, for the sake of readers' convenience, I shall employ the method and use the notion of model-theoretic throughout from this point onwards.

Similarly, for the axiomatic system development, the theory (model-theoretic approach) serves as a knowledge base comprised of a generic set of terminologies. That will construct the theory (knowledge base) referring them to the key terms (chapter 4) such as action, task, event, activity, process, control flow and gateways used in the process modelling standards. The description of the chosen vocabularies would assist in determining their structure and boundaries using first-order logic.

Furthermore, the formal semantics of the enumerated modelling artefacts would have the power to unify both the standards most often used terminologies expressing the complexity involved in a business process at all levels of conception. Thus, for the readers' convenience, I consider abstract process term to serve as theory and its translation as its instantiation, i.e. business process model. Next, I would define the enumeration for its precise representation within the axiomatic system.

Definition 6.1 – Enumeration: I define the enumeration based on process centred approach comprised of knowledge presenting essential components expressed as a tuple

(A^{SP}, A^{P}, P, DTC). These generalised conceptual terms considered primitive and notated them with the suffix of the word ‘process’. i.e. special atomic process (A^{SP}), atomic process (A^{P}), process (BP), and derived temporal constraints (DTC) associated with lexicons presented in chapter 5. Where A^{SP} represents a possible set of special atomic processes, A^{P} represents a set of atomic processes, ‘P’ represents a set of processes and DTC serves the derived temporal relations between them.

In addition, these terminologies referred to the UML-AD and BPMN core modelling terminologies (chapter 4). For example, a ‘special atomic process’ is associated with a temporal point of PITL and referred to the term ‘event’ (BPMN) and ‘initial and final nodes’ (UML-AD). Similarly, the atomic process is associated with the temporal moment of PITL and referred to the terms ‘action’ (UML-AD) and ‘task’ (BPMN). Process and sub-process are associated with the time interval of PITL and referred to business process and sub-process of real-life concepts used in the modelling techniques. Please note, the terms such as business process and process used in this study interchangeably. However, there is another conceptual term notated here as ‘Derived Temporal Constraint’ denoted as DTC to show the dependency between the process elements identified above.

The identification of generalised modelling artefacts presenting the theory of the axiomatic system which is abstract serving as the logical foundation for the business process modelling domain. The logical foundation would include the clear semantics (formalisation) for the generalised terms identified above using axioms for their precise representation. Moreover, these modelling artefacts also referred to healthcare terms such as a hospital, disease, assessment, diagnosis, treatment and discharge etc. depending on their temporal structure. Thus, I have uniquely provided a set of terminologies that could combine both the standards most often used modelling artefacts if distinct ontology provided to fill the gap.

Definition 6.2 – Abstract Process (A^{PM}): An abstract process defines the theory of notated as an abstract process model (A^{PM}) expressed as a triad *(A, T, Dur(T)).* Where ‘A’ represents a collection of process names a_1, a_2,….a_n, with corresponding set of temporal occurrences t_1, t_2,…..t_n, notated as ‘T’ associated with a given set of duration assignments Dur(t_1), Dur(t_2),……,Dur(t_n) notated as Dur(T). It is important to be noted that in this research, the notion of abstract process and abstract process model used interchangeably.

The lexicons provided in chapter 5 facilitated the A^{PM} to make provision for a precise ontology of the enumerated elements of the knowledge base. Therefore, to determine the occurrences of either time point, moment or interval within A^{PM}, I would be using the ‘Part’ relation in this research presented in [Ma, 2007] as

$$Equals \vee Starts \vee During \vee Finishes \qquad (R_1)$$

'Part' relation was an extension of relation 'In' presented in [Allen, 1983] which is $Starts \vee During \vee Finishes$.

Now, to define A^{PM}, I would use a predicate 'Occurs' coupled with relation R_1 given above establishing that a process name from the set of process names 'A' may occur on a temporal element from 'T', i.e. point, moment or interval bearing some duration established by the 'Dur' function.

$$Occurs(A, T, Dur(T)) \Rightarrow \exists t_1\ (Part(t_1,t) \wedge Occurs(a_1, t_1, Dur(t_1))) \qquad (Axiom\ 1)$$

Axiom 1 states there must exist a component of the enumerated modelling artefacts occurred over a time element identified by attached 'Dur' assignment establishing the length of the occurring element. In Axiom 1 'a_1' of the A (that could be either special atomic process, atomic process, process depending on the associated occurring time element' duration determined by 'Dur' function (establishing the fact about the time structure such as whether it is a moment or interval. In general, 'Dur' assignment representing the moment or interval expressed as:

$$\forall a, t \Leftrightarrow \exists a_S, a_E \wedge Dur\ (t) = (a_E - a_S) \in \boldsymbol{R} \qquad (Axiom\ 2)$$

Where a moment or interval bounded by its extreme points (start (a_S) and end (a_E) points). Axiom 2 may express the atomic process, process and subprocess associated with a moment and/or interval. In addition, to represent special atomic process within A^{PM} 'Dur' function used to express the duration attached with it as:

$$Dur\ (t) \in \boldsymbol{R^+} \qquad (Axiom\ 3)$$

Where $\boldsymbol{R^+}$ represents a subset of $\boldsymbol{R}$. Therefore, axiom 3 defines the occurrence of the special atomic process having either no or zero duration that may be associated with the temporal point expressing the stamp. With the combination of the above axioms, I have managed to describe A^{PM} representing the theory comprised of the knowledge base (essential modelling components) that would facilitate real-life modelling. Now, I will provide definitions of the modelling artefacts individually forming their formal semantics.

Definition 6.3 – Atomic Process (A^P): An atomic process is a fundamental element of the abstract process (A^{PM}) that may be associated with non-divisible time moment as expressed below:

$$Occurs(A, T, Dur(T)) \Rightarrow \neg\exists\, t_1\ (In\ (t_1,t) \wedge Occurs(a^P,t_1,Dur(t_1))) \quad (Axiom\ 4)$$

The definition provided here establishes that an atomic process occurs over a time element (indivisible) with positive duration. However, its attached duration (which is positive) determined by Dur function. This definition provides clear semantics that can be associated with the various real-life modelling standards terminologies such as task, action, registration of a patient (used in business and healthcare domains). Because these terminologies present atomic structure (indivisible) to model a real-life process. Therefore, the definition provided here is general enough to subsume all of them. The definition also specifies that once an atomic process started it continues until its completion without reference to other processes. It waits neither for other processes to complete nor initiates other processes before its completion. Furthermore, generally all atomic processes ‘a_i‘ expressed as:

$$\forall a_i \Rightarrow Occurs\ (a^P_i,\ t_i,\ Dur\ (t_i)) \qquad (Axiom\ 5)$$

There is a possibility of atomic processes may not occur due to provided system specification, which may be expressed as

$$\forall a \Rightarrow \neg\exists t\ (Occurs\ (a^P,\ t,\ Dur(t))) \qquad (Axiom\ 6)$$

Definition 6.4 – Special Atomic Process (A^{SP}): A special atomic process occur only on a specific time element as that of the atomic process but the main difference between them is that the special atomic process and its associated time element has no internal structure expressed below:

$$a^{SP} \Rightarrow \neg\exists t_1\ (In\ (t_1,t) \wedge Occurs(a^{SP},t_1,Dur(t_1))) \qquad (Axiom\ 7)$$

The definition provided here can be associated with the real-life modelling techniques terminologies such as ‘event’ (describing a timestamp) and patient’s diagnostics start and finish time used in the business and healthcare domains respectively.

Definition 6.5 – Business Process (P): To define a business process “P”, considering it is occurring over a time interval that may be divisible represented as a schema which contains a pair

$$P = (A,\ DTC) \qquad (Axiom\ 8)$$

Here, I assume that ‘A’ is comprised of a set of atomic processes names such that a_1, a_2,……., a_n occurring over a breakable interval. Considering an example of a breakable interval ‘I’ having two moments ‘i_1’ and ‘i_2’ and can be expressed as $I = i_1 + i_2$. In real-life domains such as business process modelling and healthcare modelling, i.e. patient flow modelling use (terminologies) business process and patient’s admission and discharge etc.respectively As described above that process has breakable structure comprised of a few atomic processes presenting relationships between them and can be written here as a disjunction of derived temporal constraints DTC = $[R(t_i, t_j) \mid 1 \le i, j \le n]$.

Definition 6.6 – Derived Temporal Constraint (DTC): The derived temporal constraints (DTC) comprised of temporal relations, i.e. R(A), of interval algebra and all the possibly derived relations mentioned here in this thesis (chapter 5). These constraints show the dependencies between the modelling elements (formally defined) to manage their control for their consistent flow, DTC expressed below

$$DTC \vDash R(A) \qquad \textit{(Axiom 9)}$$

Derived temporal constraints employ an inference mechanism that searches through all the possible relationships between two modelling elements to determine a consistent relation by discarding the rest. I will prove the Axiom 9 using deduction method.

Assumption: Every temporal constraint present in DTC is also a constraint present in R(A). Suppose $R(A_1)$ be any constraint of DTC, Therefore, if $R(A_1)$ is a temporal constraint of DTC then it follows that $R(A_1)$ is also a constraint present in R(A), i.e. $R(A) \vDash DTC$.

Axiom. 9 is valid because there exists at least one transitive relation containing in DTC and the disjunction of transitive relationships is transitive. Hence the transitive closure of DTC is the disjunction of all transitive relations containing in R(A).

Definition 6.7 – Sub-Process(sP): A sub-process (sP) is a sub-part of a parent process and carries the same features as of its parent process (present in the abstract process model). It can be defined by a schema; process $sP = (A_1, DTC_1)$) is called a sub-process of a process $P = (A, DTC)$, iff

$$A_1 \subseteq A \qquad \textit{(Axiom 10)}$$

$$DTC_1 \subseteq DTC \qquad \textit{(Axiom 11)}$$

So, I can say that

$$A \Leftrightarrow (A \wedge (\neg A \vee A_1)) \text{ and}$$

$$DTC \Leftrightarrow (DTC \wedge (\neg DTC \vee DTC_1)$$

Now from Axiom. 10 and 11, I could say that (A_1, DTC_1) is a sub-process ‘sP’ of the process ‘P’ = (A, DTC), i.e. $sP \subseteq P$. A sub-process ‘sP’ defined here refers to the sub-process terminology of business process modelling (BPM) and diagnosis etc terminology of patient flow modelling(PFM) that can be broken down further.

Example 6.1: I consider an example of a diagnosis process from an accident and emergency (A&E) department of a hospital. As part of treatment concerning a patient, a physician examines the patient to assess and diagnostics the patient's condition or illness. However, to diagnose the patients' problem at hand, a physician may need clinical staff to perform diagnostics consist of collecting a blood sample and conduct an ultrasound examination. The two examinations span over some length of time including the time clinical staff requires providing reports to the physician after their diagnostic completion. It shows that diagnosis as a part of the treatment process (primary process) is a sub-process represents that the formal definition of sub-process provided here is distinct because it describes its clear semantics. Each examination considered as atomic processes (unbreakable) with its associated start and finish time (special atomic process). To show the abstract process (theory) provided here is sufficient enough to subsume all the relevant terminologies used in business process modelling and patient flow modelling. The current modelling methods have no mechanism to verify that their constructed models are consistent. To fill this gap, I would include a verification system to establish the structural properties of the abstract process to authenticating the axiomatic system.

6.2 Verification

In this thesis, I would rely on formal verification method ensuring that the developed axiomatic system is correct. For this, I require to *provide the abstract process (A^{PM}) structural properties (by formally defining them) and proving that the theory formed in the axiomatic system is sound and complete. That means all the enumerated components are precisely defined presenting their distinct composition ensuring the abstract model (theory) performing accurately displaying non-contradicting derived relations among its' artefacts resulting in a consistent abstract process model (A^{PM}*).

By doing so, I will be able to verify the ontologies defined conforming to the requirements of the model-theoretic approach for the axiomatic system, i.e, identifying any inconsistent derived relation as errors and subsequent removal. In addition, the distinct ontological representation provided for the structural properties supported by the theorem-proving technique facilitates the formal evidence that the design (theory) is correct.

To summarise the verification procedure for this research, I would include formal definitions of structural properties of the abstract process model (axiomatic system), i.e., sound and complete, and further proved these definitions by theorem proving (theorem 6.1 and 6.2 respectively).

Definition 6.8 – Abstract Process is Sound: A proof procedure is called sound for the abstract process model (A^{PM}), if any inference R(A) has been proved from a set of derived temporal constraints (DTC) by a proof procedure, such that

$$DTC \vdash R\ (A) \qquad \textit{(Axiom 12)}$$

It follows logically from DTC, i.e., Axiom. 9 (DTC $\models$ R(A)).

Definition 6.9 – Abstract Process is Complete: A proof procedure is called complete for the abstract process model (A^{PM}), if for any inference R(A) that follows logically from a given set of derived temporal relations available in DTC, i.e. Axiom 9 (DTC $\models$ R(A)), the proof procedure can prove R(A), i.e., DTC $\vdash$ R(A).

Now, I will follow a proof procedure presented in [Konar, 1999] providing two theorems to prove the soundness and completeness of the abstract process model ($A^{PM)}$ defined here.

Theorem 6.1 – Abstract Process Model is Sound:

Proof: Given a set of derived temporal relations DTC and a goal R(A). Suppose, we derived R(A) from DTC by the resolution theorem. Therefore, it can be written as DTC$\vdash$R(A). I want to prove that the derivation is logically sound, i.e., DTC $\models$ R (A). Let us prove the theorem by the method of contradiction. So, I presume that the consequent of Axiom 9 is false, which means DTC $\models \neg$ R(A). Thus, $\neg$R (A) is satisfiable or true. To satisfy it, I assign truth values to all the temporal constraints present in R(A). Now, I claim that for such assignment, resolution of any two derived relations from DTC will be true. Thus, the resulting derived temporal constraint even after exhaustion of all possible derived temporal relations through resolution will not be false. Hence, DTC $\vdash$ R(A) is a contradiction. Thus, the assumption DTC $\models \neg$ R (A) is false. Consequently, Axiom 9 holds, i.e., DTC $\models$ R(A) and proves that the abstract process is sound.

Theorem 6.2 – Abstract Process Model is Complete:

Proof: Let R(A) be a formula such that from a given set of derived constraints DTC, I have DTC $\models$ R(A) which means R(A) can be logically proved from DTC. Therefore, I must show that there exists a proof procedure for R(A), i.e., DTC $\vdash$ R(A).

I shall prove it by the method of contradiction, let assume DTC $\vdash$ R(A), i.e., Axiom 12, is not true (false) that means DTC $\vdash \neg$ R(A). In other words, R(A) is not derivable by a proof procedure from DTC. Similarly, there does not exist a set of derived temporal constraints DTC_1 i.e. DTC_1 = DTC $\vee$ R(A) is unsatisfiable. Now I employ the ground resolution theorem as given in [Konar, 1999] that states "if a set of ground derived relations is unsatisfiable then the resolution closure of those derived constraints contains the 'false' derived relations. Thus as DTC_1 is not true, the resolution closure of DTC_1 yields the null derived relation, which causes a contradiction to DTC $\vdash$ R(A). Hence the assumption is wrong and holds DTC$\models$R(A) (Axiom 9). Therefore, it proves that the abstract process is complete.

6.3 Validation

In real life, whenever a system is represented through a model, the question concerning whether the constructed model truly captures the system. In this thesis, *validation procedure ensures that a correct real-life implementation (instance) of the abstract process model (A^{PM}) exists meeting all the needs and expectations of the model-theoretic approach (where theory represents the abstraction and model represents the real-life application).* In this way, I can determine the axiomatic system developed has the practical value when considering it for commercial use.

So far, the definitions of the abstract process model (A^{PM}) containing atomic process, special atomic process, business process (BP), sub-process and derived temporal constraints given in the previous sub-section have been abstract (theoretical) but verified. Now, to determine the validity of the axiomatic system ensuring that the abstract process model (A^{PM}) and its components must have their corresponding concrete realisation as its real-life interpretation. The abstraction of the theory (abstract process) provided may be taken as either *process type or process class* and corresponding real-life interpretation considered as *process to*ken or process instance respectively.

The above discussion emphasis on the need of interpretation function, therefore, I provide the translation function (by formally defining it) that would facilitate mapping of theory to model, type to token, or process class to the process instance. Importantly, from here onwards, I may refer to the abstraction as *theory,* i.e. abstract process (A^{PM}) and the interpretation as 'model', i.e. its concrete world realisation ($A^{PM}{}_R$). With the provision of mapping (translation function), an application in real-life (model) selected in such a way that the provided axioms are true propositions to construct a consistent abstract process proving that its implementation is validated.

In addition, translation (interpretation) function would facilitate a distinct mapping of abstract process model constituents individually ensuring an instance exists for each of its core elements described in A^{PM}. However, it can also be implied that the translation function provided ensures that the abstract process model core artefacts represent formal ontology to UML-AD and BPMN core modelling terms (real-life model) specified in chapter 4.

Definition 6.10 – Translation Function (ϕ): To translate an abstract process model (A^{PM}) and its components, I need to establish that there exists a corresponding instance using the translation function.

Theorem 6.3 – Translation: For any interpretation 'p' of the abstract process model (an axiomatic system developed in this thesis) there existed a corresponding unique instance p_R and expressed as $\phi(p) \rightarrow p_R$.

Proof: An abstract process has respective occurring time element and associated duration. Using translation function, for all of the abstract process constituents including time elements t and duration assignments Dur(t)m there must exist exists an interpretation ϕ such as time instance t_R and duration assignments instance $Dur(t_R)$ respectively. Thus, in general, the above translation can be expressed as

$$\forall\phi(t) \wedge Dur(\phi(t) \Leftrightarrow t_R \wedge Dur(t_R) \qquad \textit{(Axiom 13)}$$

Definition 6.11 – Abstract Process (Instance): An abstract process (A^{PM}) expressed as a triad of (a, t, Dur(t)) where $a \in A$, $t \in T$ and $Dur(t) \in Dur(T)$ and there exists a translation function ϕ to its corresponding instance ($A^{PM}{}_R$) from real-world that can be expressed as:

$$\phi\,(a, t, Dur(t)) \rightarrow (a_R, t_R, Dur\,(t(a_R))) \text{ where } Dur\,(t(a_R)) \in \{\boldsymbol{R} \cup \boldsymbol{R}^+\} \qquad \textit{(Axiom 14)}$$

The translation of abstract process duration assignment $Dur(t(a_R))$ into $Dur(t_R)$ represents the real duration assignment associated with the occurring process element (that could be either special atomic, atomic or process). Therefore, the Axiom 14 expresses an instance of the abstract process model. It is important to note that the existence of the real-world interpretation ensures automatically the temporal consistency of the abstract model itself.

Definition 6.12 – Atomic Process Instance ($A^P{}_R$): Using Theorem 1, for an atomic process a^P there exists a unique instance of it represented as $a^P{}_R$. Therefore, this translation would facilitate in mapping the abstract occurrence of the atomic process (formal) to the real-world occurrence represented as action/task spanning over a time moment (unbreakable interval) bearing some positive duration expressed as

$$\forall a^P{}_R \in A^P{}_R,\ t(a^P{}_R) \in t_R \rightarrow\ Dur(t(a^P{}_R)) \in \boldsymbol{R} \qquad \text{(Axiom 15)}$$

Since each atomic process instance is distinct, therefore, for an atomic process instance a^{P}_{R}, I impose that:

$$Dur\ (t(a^{P}_{R}) > 0)) \qquad \text{(Axiom 16)}$$

Definition 6.13 – Special Atomic Process Instance (A^{SP}_{R}): The translation of the special atomic process belonging to the abstract model provides a mapping to its instances notated in real life as events bearing no or zero duration representing stamp. Again, using duration assignment Dur(t), I can determine the length of the special atomic process instance (event) that is:

$$Dur\ (t(a^{SP}_{R}) \in \boldsymbol{R^{+}} \qquad \text{(Axiom 17)}$$

Since each special atomic process instance is distinct, therefore, for any special atomic process instance a^{SP}_{R}, I impose that:

$$Dur(t(a^{SP}_{R}) = 0 \qquad \text{(Axiom 18)}$$

Definition 6.14 – Business Process(BP) Instance (P_R): A *BP instance* $P_R\ (A_R, DTC_R))$ of the BP '$P = (A, DTC)$' present in the abstract model is an actual realisation. Therefore, using the translation function collection of atomic processes 'A' present in the P expressed as $\phi(A) \rightarrow A_R$

In addition, the set of derived temporal relations between the atomic processes within a process 'P' of the abstract process model can have its real-life interpretation using translation function ϕ expressed as $\phi(DTC) \rightarrow DTC_R$. Therefore, I define DTC_R as

$$\forall t_i, t_j \in t\ (DTC\ (a_i,\ a_j) \in \mathrm{DTC} \rightarrow DTC(\phi(a_i), \phi(a_j))\ \in DTC_R \qquad \text{(Axiom 19)}$$

Hence, for a concrete BP expressed as 'P_R' to be an instance of the abstract business process 'P', I must be able to establish the mapping 'ϕ' by translating all the derived temporal relations from the abstract model into a real-world model. That is achieved by satisfying the abstract model's sequencing constraints within its corresponding real-world instance, i.e. $P_R=(A_R,\ DTC_R)$ must be temporally consistent. This definition provides formal semantics that may also refer to the terminologies Activity (UML-AD) and process (BPMN) from the real-world.

Definition 6.15 – Derived Temporal Constraint Instance (DTC_R): The instance of abstract model's derived temporal constraints can be expressed by representing derived relationship instances between process elements A and B (that may either be a special atomic process, atomic process or process) can be written as

$$\mathrm{DTC}_{R(AB)} = \{\mathrm{DTC}_{R1(AB)} \vee \mathrm{DTC}_{R2(AB),\ldots}\ \mathrm{DTC}_{Rn(AB)}\} \qquad \text{(Axiom 20)}$$

The axiom 20 mainly based on the logical representation of the derived constraints as given in chapter 5 utilising all subsets of relations present between two intervals, two points, a point and an interval and an interval and a point) to determine a valid derived temporal relation between process elements A and B.

In order to control the flow between given modelling artefact instances and achieving valid and consistent derived temporal relations between them, I would use two binary operations, i.e. intersection and composition, of the set theory denoted by $\oplus$ and $\otimes$ respectively. If intersection '$\oplus$' used for derived temporal relation between process elements A and B written as $DTC_{R1(AB)} \oplus DTC_{R2(AB)}$ then it must establish a resultant derived temporal relation which is a valid and consistent instance. If '$\otimes$' operation used to represent the composition between derived temporal constraint instances expressed as $DTC_{R1(AB)} \otimes DTC_{R2(AB)}$ then resulting derived temporal relation must be a valid relation that may present in a set of derived temporal constraint (shown in figure 5.1) between two temporal objects associated with the abstract process model constituents' instances. It is assumed that either of the operations would result in a consistent and valid derived temporal relation instance. In addition, these operands used to show the functioning of real-world branching (BR), i.e. $\oplus$, and parallel(CN) behaviour, i.e. $\otimes$, respectively, assisting the occurring instances to hold the sequencing constraints present in the abstract process model. For example, two process instances a_1 and a_2 considered have positive breakable length representing their duration (intervals) then the application of binary operands defined above can be utilised to show their functioning as:

$$DTC_{R1(a1a2)} \oplus DTC_{R2(a1a2)} \rightarrow \exists a \in (a_1 \wedge a_2)] \qquad \text{(Axiom 21)}$$

$$DTC_{R1(a1a2)} \otimes DTC_{R2(a1a2)} \rightarrow \forall c\ (\exists a \in a_1 \wedge \exists b \in a_2] \wedge a \vee b = c \qquad \text{(Axiom 22)}$$

These operands would facilitate the branching and forking behaviour of the abstract process model instance described here.

Definition 6.16 – Sub Process Instance: To define a unique interpretation of sub-process (of the abstract model) $P_1 = (A_1, DTC_1))$, there must exist two or more than two unique atomic process instances in the respective instantiation. The translation of sub-process is written as $\phi(A_1, DTC_1) \rightarrow (A_{1R}, DTC_{1R})$ and it is formally expressed in Axiom 23 below.

$$\exists t_i, t_j \in DTC_{1,} \rightarrow \phi\ (a_{iR}, a_{jR})) \in DTC_{1R} \qquad \text{(Axiom 23)}$$

A sub-process is a part of a parent abstract business process which is consistent therefore, its real-life interpretation is sound and complete. Accordingly, it must be temporally consistent that satisfies the sequencing constraints specified in the subprocess of the abstract model, i.e. $P_{1R} = (A_{1R}, DTC_{1R})$.

6.4 Summary

This chapter takes the lead from the discussion provided in the chapters 2-5 in which a foundation laid out by identifying the necessary artefacts that are used most commonly by UML-AD and BPMN associated with the temporal objects of the point and interval temporal logic (PITL) to develop the phase II (axiomatic system) of the framework. In addition, the phase II of the framework developed based on the knowledge provided in the previous chapters to empower the modellers with a certain number of modelling artefacts (enumeration) that have precise ontology (semantics) to construct a general business process avoiding confusion provided by the commercial business process modelling standards.

Another reason to develop this framework is to address the problems faced by the industry due to the intuitive nature of both the standards. That led stakeholders to construe the meaning of the used artefacts hindering the consistent generation of the process models and takes longer to come to an agreement between them. Therefore, in this chapter, I have provided the axiomatic system (contribution to the knowledge) developed comprised of the enumeration of the modelling artefacts supported by formal definitions using first-order logic. The listed enumeration associated with the temporal objects of phase I of the framework facilitating the construction of a typical business process.

In a systematic way, utilising model-theoretic approach adopted, abstract process representing theory (knowledge base) developed having certain number of modelling artefacts (abstract) associated with the temporal objects of the PITL i.e. point, interval, moment and temporal constraints. The enumerated artefacts (special atomic process, process, sub-process, atomic process and derived constraints) of the axiomatic system formally defined using first-order logic providing formal semantics (contribution to the knowledge) which are missing in both the process-modelling standards (UML-AD and BPMN).

The axiomatic system developed further required to be verified and validated. To verify, I have adopted a formal procedure by proving the theorems to establish that the developed axiomatic system (abstract) is structurally sound and complete. Therefore, I have provided formal definitions for sound and complete features of the axiomatic system based on model-theoretic approach utilising resolution theorem to determine its correctness. That is determined by establishing that all the inferred temporal relations among the enumerated artefacts (abstract) are not contradicting.

The next step after axiomatic system verification was its validation. After the development of the theory (axiomatic system), the model-theoretic approach required its real-life implementation to show its explicit instantiation. Similarly, a subsequent set of instances of the abstract process model artefacts provided supported by formal definitions. Furthermore, a translation was required for the abstract model (theory) to its real-life instances to validate the axiomatic system. Therefore, a translation mechanism (applied mathematics) provided to map (formally defined) abstract modelling artefacts (theory) to their instances that may class them as a real-life model.

In addition, the phase II of the framework provides contribution to the knowledge by providing formal semantics to the business process modelling standards' (UML-AD and BPMN) most often used terminologies, which was missing in their standard documentation Other contribution to the knowledge include verification and validation of the axiomatic system, however, modelling standards failed to provide such mechanisms resulting in construction of inconsistent models and no procedure to verify that they are incorrect.

The logical foundation provided in this chapter has the power to be represented graphically through an appropriate and relevant graphical technique that supports the axiomatic system and has provision to accommodate the point and interval logic presented in here in this thesis. Therefore, to authenticate that that instance provided of the abstract process model is valid, I would require to graphical represent the instantiation of the abstract process to ease the modellers' life to construct the consistent process models, if they choose to use the method developed in this thesis. The next chapter will discuss the visual representation of the axiomatic system using a formal but simple to use graphical technique containing simple graphical constructs, i.e., vertex and arc.

Chapter 7 Enactment

Process enactment using the graphical tool considered a vital constituent of business process management that determines the effectiveness of process description and its design. PITL based axiomatic system which is verified and validated in the previous chapter and now require its graphical representation to establish its working in the real-life (process execution) concerning time-sensitive issues. For this research, I would consider the enactment procedure to address two important issues faced by the industry regarding business process modelling that are the graphical representation (of the axiomatic system) and the second issue is relating to process planning and scheduling. The next sub-sections would be answering questions such as how the abstract process model instance visualised and scheduled for further optimisation.

7.1 Visualisation

To represent the abstract process (system specification) visually in the form of Point interval temporal logic (PITL) statements, I would rely on a graphical tool known as point graph (PG) presented in [Zaidi, 1999]. That is formal, use a simple node and edge notation and its foundation based on PITL in expressing business process modelling temporal perspective. For the readers' convenience, I have provided its definition here.

Definition 7.1 – Point Graph (PG): is comprised of a tuple (V, E_A, D, T). 'V' constitute a set of vertices. 'E_A' is a union of edges between two vertices to represent temporal relation "Before' i.e. 'E', and "Precedes" i.e. '$E_{\leq}$', shown as solid edge or dotted edge respectively. 'D' represents the duration between the vertices, where each vertex represents a timestamp "T".

In addition, for every V, there exists a pre and post set expressed either as $^{\bullet}v$ or $v^{\bullet}$ to represent an entire set of the nodes having edges either starting from or completing at them. Moreover, to start and end a PG flow can be visually represented by its corresponding source and sink nodes notated as V_{in} and V_{out} respectively connecting to all nodes (Vi's) by less than equal to edge. PITL allows the assignment of the precise duration of the intervals to specify the length and points (stamps) representing exact time of happening only if known. It is a possibility that due to changes in the specification of a system may result in the altered

PG [Rauf & Zaidi 2001]. This step ensures the validity of the process instance having no errors within the modelled process instance.

In addition, using PG to model the instances of the abstract process ensures that the framework developed in this thesis serve as the logical foundation for the real-life business process modelling standards and assist in modelling both the business processes and patient flows. Keeping this in mind, for this research, PG requires a mechanism to represent the abstract model instance (denoted as $A^{PM}{}_{R}$) in which every instance is unique. Therefore I provide a matching rule mechanism establishing that all the PG rules are adapted and further expanded to accommodate the axiomatic system modelling artefacts.

Definition 7.2 – Rules Matching (RM): To define rules matching, I could consider a process instance P_R with distinct starting and ending nodes i.e. special atomic processes. However, each time element "*t"* represented as an edge in PG to express its duration (if known). Therefore, I assume that all process instances of the axiomatic system have named graph structure within a PG defined as a tuple RM = ($A^{PM}{}_{R}$, PG). Where

$$A^{PM}{}_{R} = \{P_{R1} \vee P_{R2} \vee, \ldots\ldots P_{Rn}\} \qquad \text{(Axiom 245)}$$

'$A^{PM}{}_{R}$' represents a collection of names of the process instances comprised of special atomic processes, atomic processes identified by the attached time element, its duration and the relationship as between them defined in chapter 6. Where PG has a graphical structure defined above as (V, E_A, D, T). Therefore, I enforce that 'V' represents all the nodes within a system specification expressing the knowledge about all the process instances (unique) bounded by their source and sink nodes (special atomic process instances) labelling them with the corresponding literals. However, the relationship between nodes (process instances) determined by their edges (E_A) showing their corresponding length (D) associated with the time element (T). In this way, PG is extended with the rule matching mechanism. In addition, extension to PG enables to represent the axiomatic system establishing a pair of nodes representing the atomic process (A^P) where edge in between represents the time element (T) expressing its duration (Dur). Importantly, the initial node of the atomic process (A^P) represents a special atomic process (A^{SP}) by establishing A^P start point (stamp) and corresponding end node represents the endpoint associated with the atomic process (A^P).

Furthermore, PG does not provide a formal definition for concurrent behaviour (only describe them intuitively) representation as required by the real-life modelling of complex processes such as branch and join mechanism. However, the axiomatic system is equipped with such mechanism (binary operations $\oplus$ and $\otimes$ denoted as 'BR' and 'CN'). With the help of such facility, PG is extended to unfold the graph for possible branching and concurrency. Therefore, the rules matching the concurrent graphical representation of the abstract process instances are given below:

$$V = \{V_{in}\ V_{out}\} \vee P_R \oplus \{DTC_R \in E_A, \exists (DTC_{r1}, DTC_{r2}, DTC_{r3}) \wedge (P_{R1} \otimes P_{R2}) \neq \emptyset\} \quad \text{(Axiom 26)}$$

$$E_A = P_{R1} \otimes P_{R2} \vee \{DTC_R \in E_A, \exists (DTC_{r1}, DTC_{r2}) \vee \{V_{in}\ V_{out}\} \quad \text{(Axiom 257)}$$

A general graphic representation of a process instance P_R expressed as Vi and shown below in figure 7.1.

Figure 7 1 PG* representation of Branch /Join

Hence, the extension provided here for PG supports the extended PITL, it follows that all analytical mechanisms operate for extended PITL and PG without any change. Therefore, from here onwards I would be using PG* instead of PG.

7.2 Analytical Support

Business logic adopted here has a unique edge over existent applied logics in the business process modelling. The definitions provided presents precise structure that can be associated with the terminologies used in the practical modelling techniques expressing the consistent business process behaviour. In other words, precise definitions provide a structure where behavioural rules match decision rules. For example, processes associated with intervals considered semi-open(typed point), and relationships between them served as derived temporal constraints to establish the different behaviours determined by the PITL inference mechanism. The axiomatic system has the flexibility to specify both the absolute temporal information and relative temporal relationships concerning the process instances. With this provision, modellers are better equipped to express the organisational behaviour and improving their decision-making ability concerning a specific operation. This can be explained the example 7.1 below.

Example 7.1: Axiomatic system based on extended PITL assists in expressing relative temporal information between process instances. Let's consider a process instance

X overlapping another process instance Y and visually represented using PG*. Where X and Y are the process instances of the abstract model. Therefore, their relative relationships are shown below in figure 7.2.

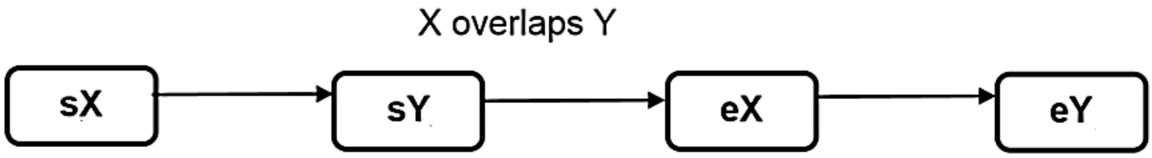


Figure 7 2 Overlapping process example

Additionally, a PG* can represent absolute information, if known to show the duration of the process instances and special atomic process (stamp) instances shown in figure 7.3 below

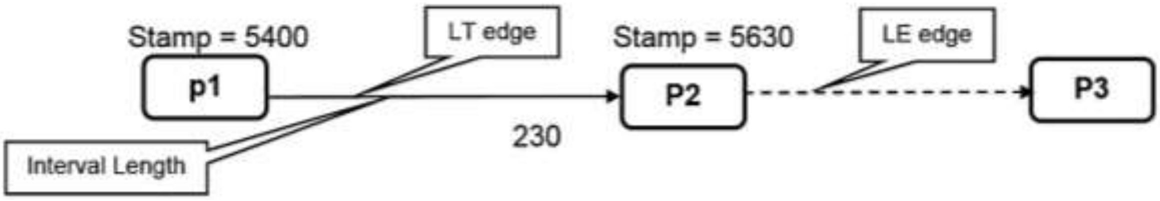


Figure 7 3 Quantitative (temporal) information representation

Figure 7.3 illustrates both relative and absolute temporal information where "LT" refers to less than and "LE" refers to less than equal to relative relation using extended PITL statements.

The axiomatic system based on extended PITL supported by PG* through its inference mechanism in finding the undirected paths within a process model instance. In an attempt to represent a consistent diagrammatic representation of a process instance, I will use the three algorithms (unification, branch and join folding) of PG so to establish the applicability of the technique to the axiomatic system.

Definition 7.3 – Unification: The unification algorithm in [Zaidi and Wagenhals, 2006] can be adapted here by considering the two vertices (process instances) such that $v_i = [a_i;...;a_n]$ and $v_j = [a_j;...;a_m]$. If a_k is another vertex to represent a temporal point (special atomic process instance) which exists as $a_k \in [a_i;...;a_n]$ and $a_k \in [a_j;...;a_m]$ then v_i and v_j are joined together into a single composite vertex $[v_i; v_j]$. This unification algorithm of PG* would result in the redefinition of E_A establishing derived temporal relations between them.

Example 7.2: To illustrate this, I consider two process instances X and Y with quantitative temporal information; for duration of process instance X represented as [sX, eX] = 20 and process instance Y duration as [sY, eX] = 20 shown in figure 7.4.

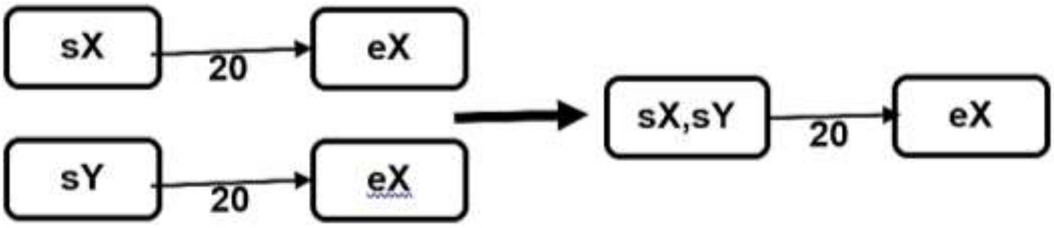


Figure 7 4 Unification

After unification, the graph inspected again for any branch and join edges of corresponding vertices to check if any folding is required.

Definition 7.4 – Branch Folding: This algorithm infers new temporal relations among process instances after the analysis of the known relations with respect to their absolute temporal values. Branch folding only exists if, ∀ vj, vk ∈vi•. There are three possibilities of this algorithm which are shown in figures 7.5 – 7.7 below

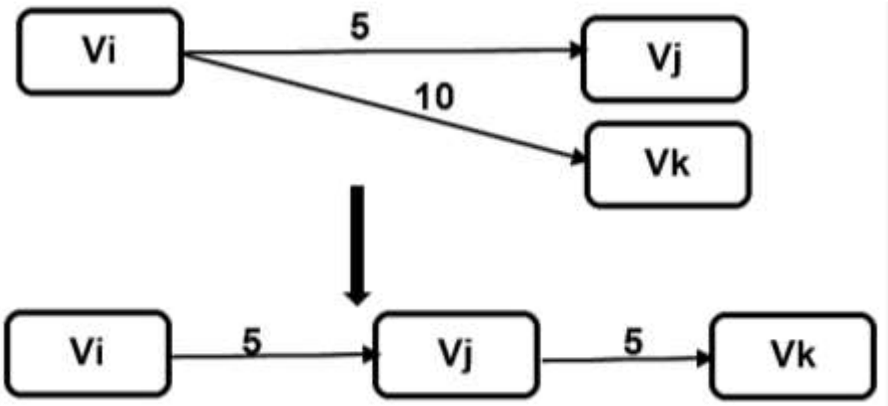


Figure 7 5 Branch Folding algorithm I

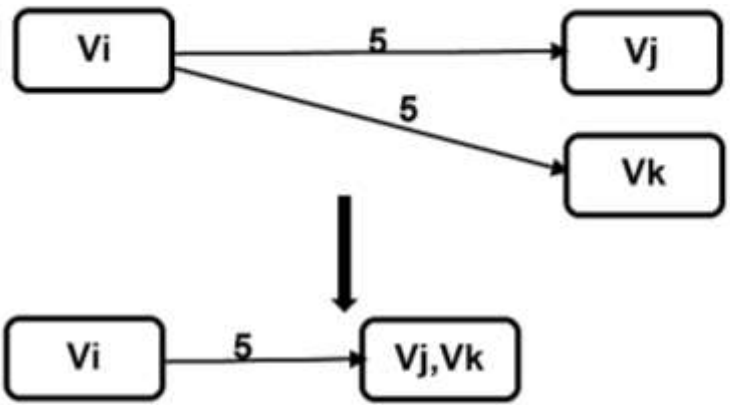


Figure 7 6 Branch folding algorithm II

The third possibility arises If 'vi' having several outgoing edges of the identical category (LT or LE) to vj but exclusively LT edge is retained, to express its corresponding duration, and rest are removed. Accordingly, PG* is updated. But it is possible that not all edges have duration specified that may not result in a PG* without an outgoing branch.

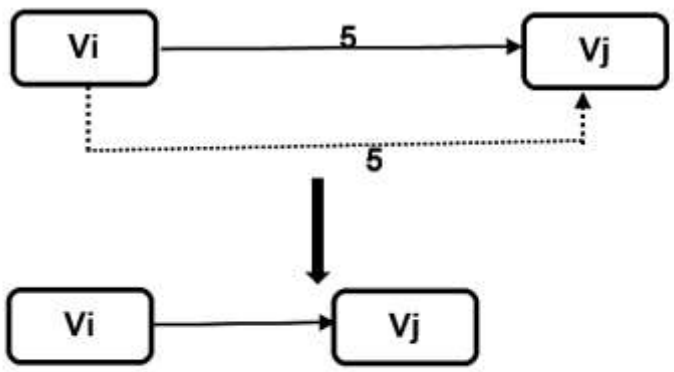


Figure 7 7 Branch folding algorithm III

Definition 7.5 – Join Folding: It is defined as ∀ vj, vk ∈ •vi. This algorithm has three possibilities, the first two of them are shown in figures 7.8 and 7.9 below

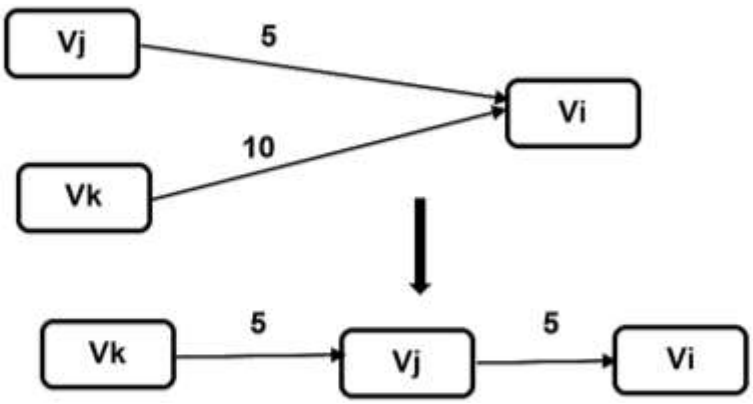


Figure 7 8 Join Folding algorithm I

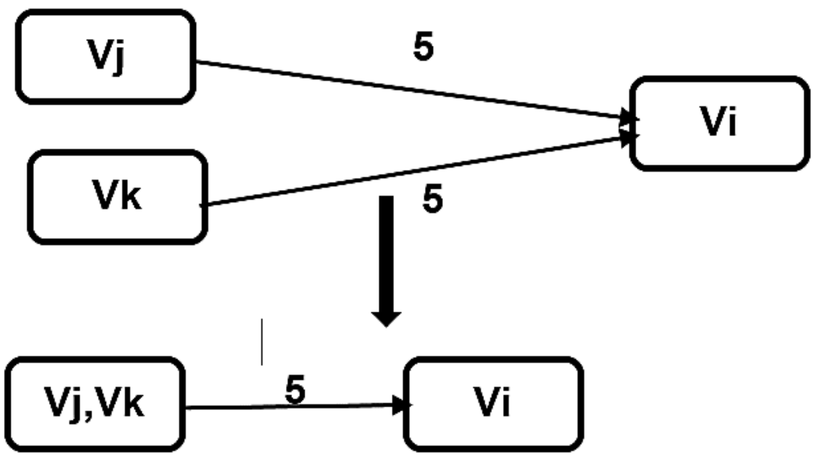


Figure 7 9 Join folding algorithm II

The third possibility exists If vi has several incoming edges of the identical category (LT or LE) to vi but exclusively LT edge is retained to express its corresponding duration and rest are removed. Accordingly, PG* is updated. Importantly, this condition becomes unnecessary in case branch folding is implemented prior to join folding. To fully fold a PG*, an application of branch folding followed by join folding is required.

Example 7.3: Considering the following scenario where a process instance A overlaps process instance B with the given duration [sA, sB] = 20, duration [sB, eA] = 15, however, there is a process instance C overlaps process instance A; duration [sC, sB] = 10, duration [sB, eC] = 15. Using PG*, the above scenario can be shown in figures 7.10.

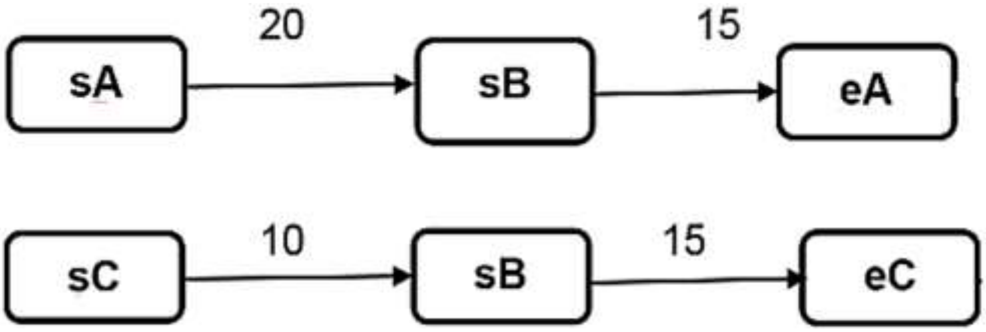


Figure 7 10 Process instance (using PG*)

With the application of three algorithms, i.e. unification, branch folding and join folding the above PG* can be transformed shown in 7.11-7.13

Unification

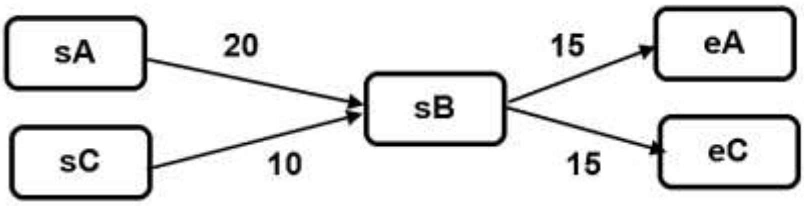


Figure 7 11 Process instance after unification

Branch Folding

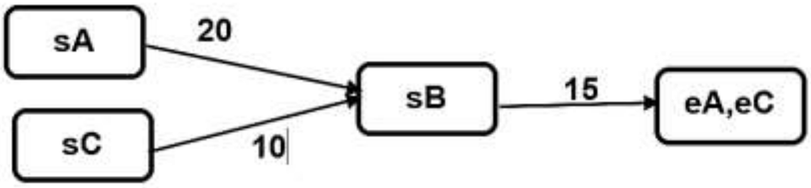


Figure 7 12 Process instance after branch folding

Join Folding

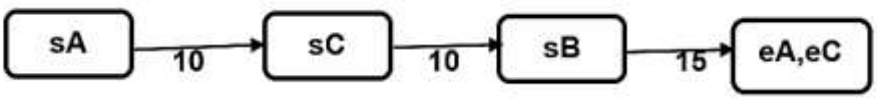


Figure 7 13 Process instance after join folding

Figure 7.13 graphically represents the new inferred relations such as 'process instance C finishes process instance A' and duration is [sA, sC] = 10. The axiomatic system using inference mechanism to perform a simple search i.e. eight searches on the PG* to find unspecified relations between the two special atomic process instances (start/end) associated with a process. This search ensures a path between the two process instances with all possible (relations) edges associated with a duration that corresponds to 'Dur' function of the axiomatic system. On one hand, the inference mechanism may result in errors or inconsistent PG* because of erroneous information provided in the specification. On the other hand, consistent PITL information guarantees to return valid assertions. For readers' convenience, I am providing the inconsistency definition presented in [Zaidi, 1999].

Definition 7.6 – Inconsistency: It is defined by a set of inferred statements that they cannot all be satisfied concurrently and considered inconsistent as shown in figure 7.14.

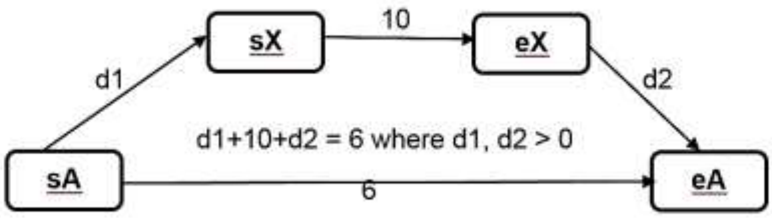


Figure 7 14 Example of an Inconsistent PG*

However, the inconsistent absolute temporal values present in the system specification may either deter the PG* folding or produce invalid PG* formation. For instance, several edges with differing absolute temporal values are found while folding a PG* then the PG* construction is paused to report the error. After clearing the inconsistency (possible cycles), PG* is inspected again using the path-search algorithm [Ma, 1999] to search for all the possible paths with consistent information. The next step is to have a consistent PG* that is further folded (folded PG* is used for deriving new facts).

7.3 Scheduling

Process instance execution graphically represented using PG* supported by the temporal view facilitating not only better process planning but also offer scheduling mechanism. Generally, PITL statements used to specify a system/model that then transformed into a corresponding PG*. To assist with process planning, modellers construct the precise process structure graphically using PiTL statements based on what-if analysis specifying the system structure and its possible behaviour. And, in case any changes made in the initial specification of the system design that can easily be accommodated by amending the existing graph rather regenerating the PITL statements and PG* from scratch. Because the PG* presents the axiomatic system elements' coupled with their associated temporal knowledge (absolute and/or relative temporal information). Furthermore, the provision of PITL inference mechanism makes PG* a powerful analytical tool for modellers to examine the constructed model for any inconsistencies.

PG* specification offers scheduling feature to construct an efficient process model that best suits the needs of an organisation. It is achieved by designating three parameters (time values) to each PG* vertex. Two runs of Forward* and Reverse* algorithms applied on a PG* to find these values. These values facilitate in analysing a PG* while executing the axiomatic system graphically to determine the critical process instances and operational constraints termed here as applied constraints1-4. Applied constraints are used to assist in model instance enactment to find out how flexible a process can be concerning the time specification presenting the process i.e. earliest time (Ev), Late and Latest times (Lv & Tv). These parameters mentioned in [Zaidi and Wagenhals, 2006] and used here to execute the axiomatic system. Where 'E_V' represents the smallest stamp values associated with a special atomic process that captures the preceding processes earliest occurrences needing a PG* forward traversal from its source node (by default it has a '0' value). Forward* algorithm is shown below in fig. 7.15

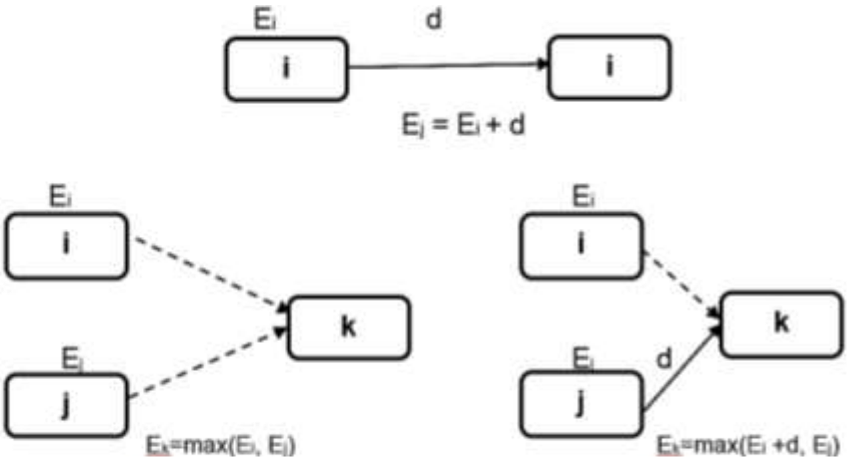


Figure 7 15 Earliest Time (EV) algorithm

However, 'L_V/T_V' describes the special atomic process with a largest time stamp that captures the proceeding processes' earliest (latest) time values as shown in fig. 7.16. Reverse* algorithm with two passes to count the 'L_V/T_V' respectively on a PG** from sink node (by default, its value equals to special atomic process earliest timestamp)

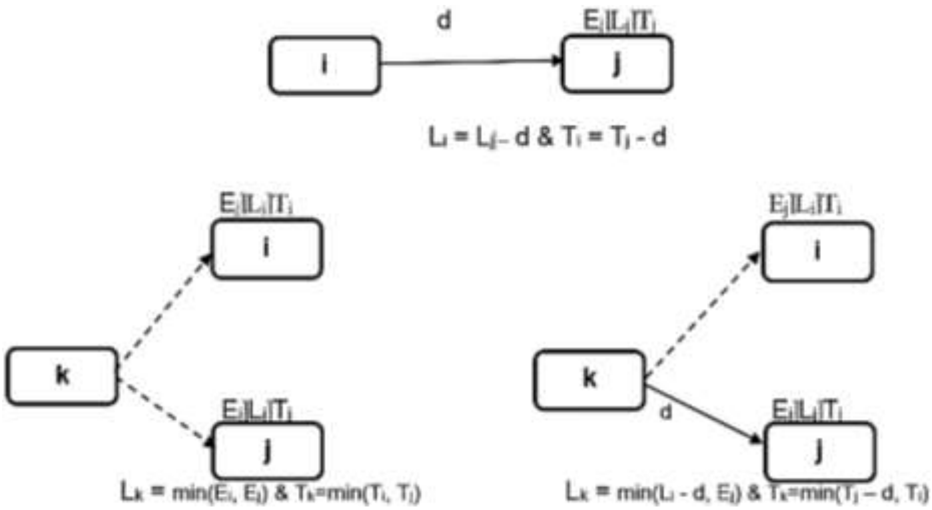


Figure 7 16 Late/Latest Occurrence Time (LVTV)

PG* facilitates the axiomatic system (process instances) graphical representation on a timeline specifying when it should occur. Furthermore, it provides an aid to determine the useful information that could be helpful for planning, for example, a delay without hampering process completion time, process earliest start time and identifying critical processes, etc. In addition, the three algorithms adopted in PG* earlier in the chapter (unification, branch and join folding) would make provision for scheduling of activities. Hence, PG* additional capabilities compared with the current operational research methods (such as program evaluation and review technique (PERT) and critical path method (CPM) presented in [Moder and Philips, 1970], has edge in expressing precise duration, stamp, and can further reason concerning a constructed process model.

Moreover, the above definitions present constraints on the process instances concerning their start and end. So, for a non-critical process, $[A_s\ A_e]\ T_V\,(A_e)$ represents the latest completion time of special atomic processes (start and end) of a path with less than (LT) arcs only. Therefore, for the clarity and readers' convenience, I would describe the critical process, float and stretch float below.

Definition 7.7 – Critical Process: A process is critical iff:

a) there is a delay in a process initialisation causes the delay in its completion time, i.e.

i) for a special atomic process A^{SP}, $E_V = T_V$;

ii) for an atomic process (moment) $[A_s, A_e]$, where $A_s, A_e \in A^P$, $E_V\,(A_s) = T_V\,(A_s)$ and $E_V(A_e) = T_V(A_e)$, or

b) A critical atomic process (moment) links with another critical atomic process using either 'meets' or 'met by' relation. Similarly, a critical special atomic process 'starts' and/or 'finishes' atomic process of a similar type, or
c) A process parametric time values do not ensure its earliest (latest) completion, i.e. $[A_s, A_e]$, $E_V(A_s) + D(A_e - A_s) < E_V(A_e)$, or $T_V(A_s) + D(A_e - A_s) < T_V(A_s)$.

From (c), it is clear that a system specification providing certain process start and completion time only, ensuring the preceding and proceeding atomic process timings. Therefore, there is no delay specified in a critical process. However, keeping in mind while planning, sometimes the absolute temporal information concerning a process instance is not exact and specifies time changing behaviour of the system. These constraints are defined below:

Definition 7.8 – Applied Constraint 1 (AC_1): It specifies the variance between the maximum time available for an instance (process) to perform and its duration to specify process lateness constraint. A process instance can be delayed either from its start or finish time (relative). This constraint determines the enactment of a process instance start(late) time and still complete it by the specified finish time. AC_1 can be applied to special atomic processes and processes(atomic processes). Without the knowledge of completion time of a process instance, AC_1 may be impossible to schedule the process. For a special atomic process a^{SP} the $AC_1 = T_V - E_V$, and for the process $[A_s, A_e]$ the $AC_1 = T_V(A_e) - E_V(A_e)$ (= TV $(A_s) - E_V s$).

Definition 7.9 – Applied Constraint 2 (AC_2): It is defined by ensuring that all the process instances of a process model initialise as soon as they can provide the maximal time available over its length. AC_2 for a special atomic process a^{SP} the $AC_2 = L_V - E_V$, and for a process(atomic process) $[A_s, A_e]$ the $AC_2 = L_V(A_e) - E_V(A_e)$ (= $L_V(A_s) - E_V(A_s)$).

Bear in mind, these constraints specify non-critical process instances. And a critical process instance with the application of AC_1 and AC_2 would be equal to 0. additionally, there is another constraint that may apply to the axiomatic system to schedule process instances and defined below.

Definition 7.10 – Applied Constraint 3 (AC_3): This constraint specifies the actual duration of a process instance and its required duration. For a critical process $[A_s, A_e]$ AC_3 is defined as the maximal time available for the length of a process instance start(earliest) 'A_s' and completion (earliest) 'A_e', i.e. latest. Therefore, an AC_3 for a process instance A would be

$$AC_3 = E(A_e) – E(A_e) − D(A_e - A_s) \text{ or } T(A_s) −T(A_s) − D(A_e - A_s)$$

The AC_3 constraint if present, then it would provide the following possibilities to a plan:

a) For a critical process $[A_s, A_e]$, iff:
   $L(A_s) + D(A_e - A_s) = E(A_e)$ or $T(A_s) + D(A_e - A_s) = L(A_e)$ or $T(A_s) + D(A_e - A_s) = E(A_e)$. Then, the process is scheduled.
b) A process instance $T(A_s) + D(A_e - A_s) < E(A_e)$: in spite of the latest process initialisation it would complete earlier than expected by some of previous atomic process instances. Although its completion time delayed to its excess time available (after its start), hence it is delayed.
c) For a process instance that does not satisfy any conditions mentioned above then the process needs to extend the start to complete time for a process plan.

There are other constraints that could be utilised to specify the lower and upper bounds concerning a system plan. PG* accommodates such constraints depicting the available knowledge by using a virtual node that has no temporal knowledge. Therefore, it has no effect on the system specification.

Definition 7.11 – Applied Constraint 4 (AC_4): AC_4 is defined to allow lower and upper bounds on the special atomic processor on the duration of the process (atomic process). To specify AC_4 concerning lower bound on a special atomic process to start, it can not occur other than the specified time. Similarly, AC_4 constraint for upper bound on a special atomic process instance refers to a specified time of occurrence as shown in figure 7.17.

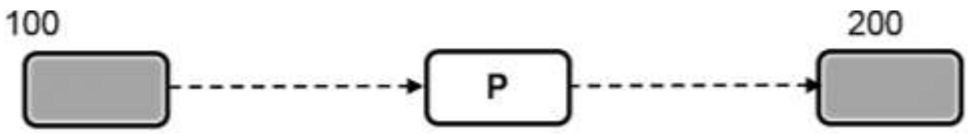


Figure 7 17 Lower and upper bounds (special atomic process)

If AC_4 for atomic process(process) instance applied then it would only perform only within the time assigned shown below in figure 7.18.

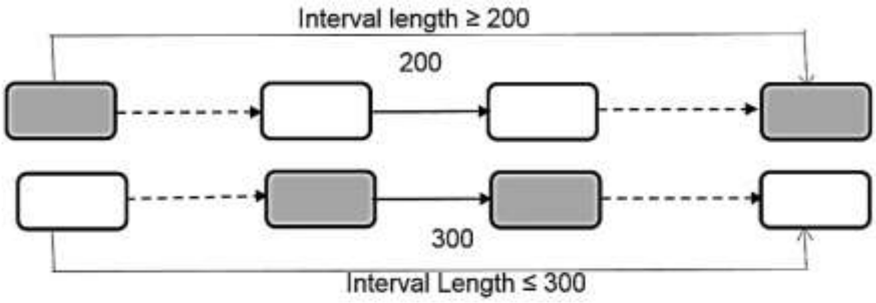


Figure 7 18 lower and upper bounds (process)

Figure 7.17 shows the lower bound (no earlier than start) on a special atomic process instance with a stamp which is $p \geq 100$ and upper bound (no earlier than complete on a stamp which is $p \leq 200$. Where figure 7.18 illustrates lower and upper bounds on process instances. The axiomatic system has a complete map to PG* by representing special atomic process, atomic process and process concerning a high-level process instance enactment. In real life, there are stakeholders' requirements to model a detailed process model specifying granularity attached. In the next sub-section, will discuss this issue.

7.4 Low-Level Abstraction

The axiomatic system developed here has the capability to express granular details concerning abstract process model instance. To facilitate its graphical representation displaying the consistent and detailed process instance, I will rely on hierarchical point graph (HPG) [Ishaque et. al., 2009]. Because it has the mechanism to breakdown the PG* (representing a process instance) further exhibiting its sub-parts(sub-process instances) with greater detail. For readers' convenience, I will provide HPG* definition below

Definition 7.14 – Hierarchical Point Graph (HPG*): HPG* has similar characteristics as of PG*, and defined as a pair (PG*, M). PG* is defined earlier where 'M' describes extended relations between vertices from PG* to HPG* so $M = \{((x, y), HPG^*_{xy}) \mid x, y \in V\}$. It also establishes that PG* has no pathway directly from y to x and HPG^*_{xy} has detailed path.

HPG*s also plays a pivotal role in supporting planning by specifying several sub-process instances concerning a high-level process instance. It also assists in reducing the overall plan development time by presenting verified sub-process instances expressing the arbitrary levels with greater detail. Please note that showing the granular details of the process instance achieved by substituting the process instance drawn in PG* with several atomic process instances using HPG*. It follows a flexible and dispersive modelling approach where vertices are labelled with times (if known) that blend different abstraction levels to represent multiple sub-process instances within a process instance.

Example 7.4: A high-level generic process instance to represent a fictitious patient flow comprised of a set of process instances and corresponding absolute and relative temporal information. Table 7.1 list 5 process instances with corresponding duration and temporal facts.

Process Symbol	Duration	PITL
A	7	A meets B
B	7	-
C	7	C precedes B
D	7	C meets D
E	14	eE precedes eD

Table 7 1 Constraints (Example)

Figure 7.19 transforms the natural language presented in table 7.1 expressing patient flow example graphically using PG*.

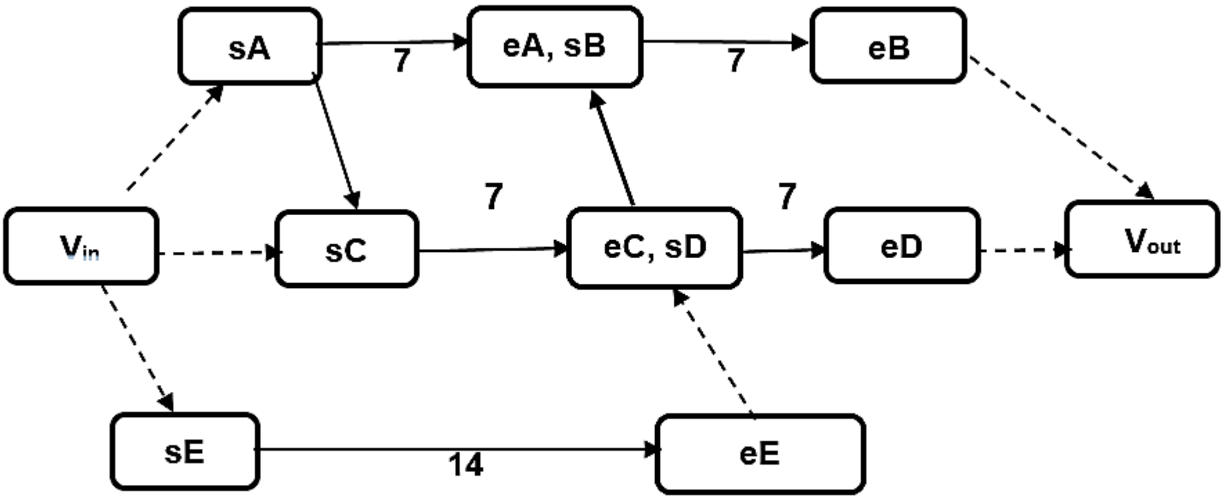


Figure 7 19 Example patient flow

However, with the further patient flow information added to the system specification indicating the existing process instance has further details added to the high-level patient flow establishing its sub-levels. The additional details extended the initial specification provided graphically represented using HPG*. Considering the example 7.4, in which a process instance A assumed to be (presented in the figure 7.19) decomposable further into two added atomic process instances that may constitute the sub-process instance of the given patient flow model, shown in figure 7.20.

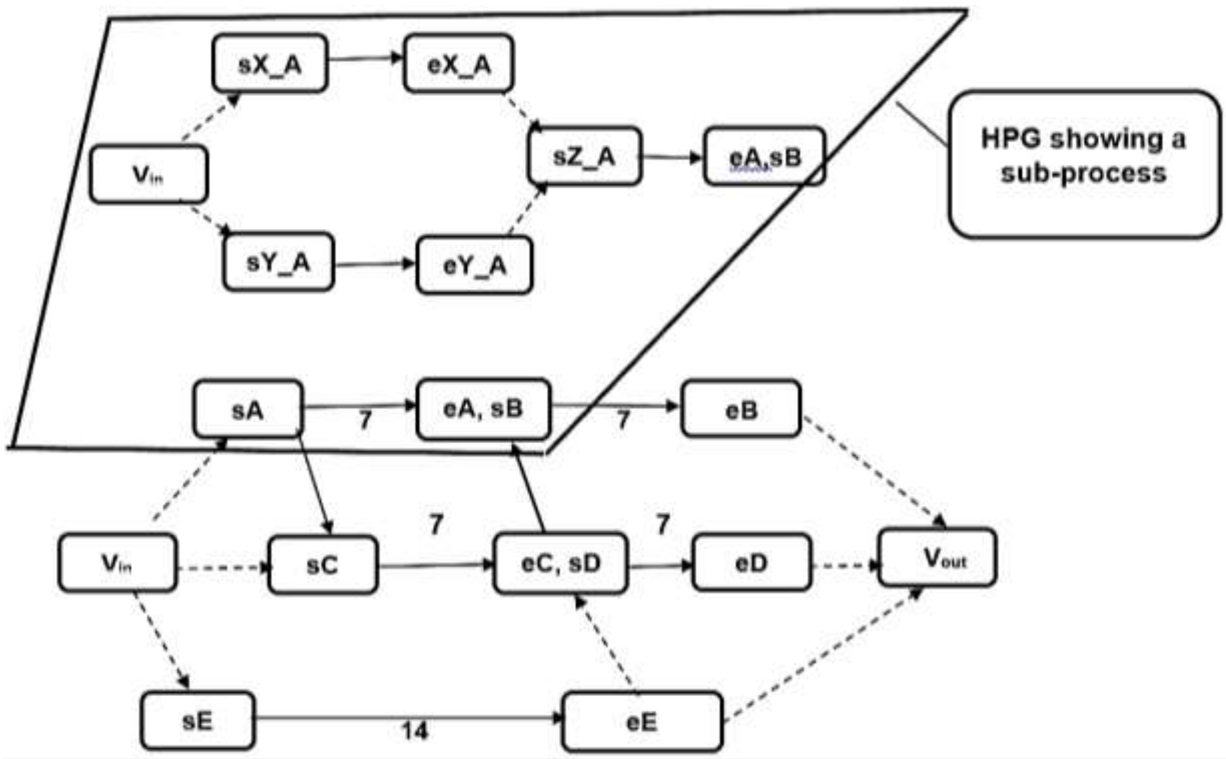


Figure 7 20 Subprocess example using HPG*

The above example illustrates the hierarchical arrangements in which the process instance A is replaced by two atomic process instances constituting a subprocess instance. It is important to note that where the duration of the process instance not specified then it can be calculated for its HPG*. However, in case the duration of the process instance specified then it is required to be higher or equal to the completion time of sub-process instance.

7.5 Summary

This chapter discussed the enactment of the axiomatic system. The execution procedure considered for this research is unique compared with business process modelling standards. Because it presents two hugely important requirements of the industry that is a concise graphical representation of the conceptual schema supported by scheduling mechanism for optimisation. Both facilities are not present in both UML-AD and BPMN and make the method presented in this thesis more attractive for its impact on the overall business process modelling.

It was desired by the industry to have a precise graphical tool to ensemble the consistent business processes and avoids unnecessary complexities of graph based modelling approaches such as Petr Net. Therefore, I have chosen a graphical tool known as PG based on simple node and edge supported by algorithms utilising natural language to graphically depict the process and associated qualitative and quantitative temporal constraints. That is extended for this research and known as PG* to meet the needs of the industry and facilitates in representing the axiomatic system graphically.

The pictorial representation supported by the three algorithms (unification, branch/join folding) ensuring the viable but correct path supported by the PITL inference mechanism determining new (but consistent) flows. In addition, the consistent path determined by PITL inference mechanism clearly indicates the process boundaries enabling a continuous path which is correct.

Moreover, modellers can plan efficiently with the support of the explicit concepts (and corresponding graphical elements). That can further be used along with the scheduling algorithm provided in PG* to achieve optimised model. However, UML-AD and BPMN insufficiently equipped to provide such a mechanism. Understandably, they are behavioural modelling techniques but managing organisational operations is required by the industry that can only be achieved if such (planning and scheduling) mechanisms embedded within the schema for improved communication.

Similarly, PG* comprised of applied constraints assist in the project planning to identify float and slack times for better performance. These constraints when applied assist in re-drawing PG* to accommodate real-life scenarios where changes occur and project structure is affected. Therefore, the effort of re-drawing PG* takes place as many times as the changes occur during the project so that a consistent model can be drawn. These constraints further facilitate scheduling if the absolute times are provided for individual activities of the process instance. Furthermore, PG* diagrammatically represent a better plan and schedule the involved process instances to complete an operation within a given budget and time.

The method proposed also facilitates a consistent representation from high to low-level process instances. PG* used to graphically represent high-level process instances though hierarchical PG* used to construct models with low-level process instance representation. Therefore, the framework developed in this thesis has provided a methodical approach by providing formal semantics for the enumerated modelling artefacts of the axiomatic system that are fully aligned with the core terminologies and constructs used in UML-AD and BPMN, and is general enough to subsume both the graphical standards (UML-AD and BPMN core modelling constructs).

Henceforth, the methodical approach proposed here to transform the modelling standards most often used terminologies and constructs to the axiomatic system artefacts. For the sake of reader's convenience, examples provided to show the operational strength of the approach.

Chapter 8 Transformation

In the previous chapter, we have seen the axiomatic system in action by graphically representing the process instances along with associated algorithms. This research has taken the challenge and defined formal semantics of the basic terms of UML-AD and BPMN using consistent terminology. This assist modellers in specifying a correct process with consistent process model. Furthermore, I have claimed to subsume the UML-AD and BPMN core modelling terms and constructs by the axiomatic system and therefore, it considered vital to provide their transformation.

. To facilitate this, I have investigated the structure, syntax and semantics of both the business process modelling standards and the results produced (chapter 4). The review based on their modelling ability using frequently used the terms and the constructs identified drawbacks in representing distinct process structure and temporal properties while displaying a business process (BP) and corresponding flow. Other findings include the lack of a logical foundation within both the business process-modelling standards. That further can be provisioned by transforming most often used modelling artefacts of both the standards into the formal conceptual schema (formal semantics).

Because both modelling techniques failed to describe its activity/process and its sub-parts precisely. Therefore, stakeholders' find it find it difficult to decide upon selecting a modelling approach with suitable modelling artefacts [Recker, 2010]. In addition, due to their incapability's, both modelling standards lacked in accommodating the changing needs of the healthcare domain (to represent a consistent patient flow). That can be achieved via the method introduced in this thesis. Thus, the need for transforming is justified.

In order to proceed with transformation, the framework devised in chapter 6 providing formal definitions for the generalised terms (such as atomic process, (action/task), special atomic process (event/start and finish), process/sub-process, branching and concurrent flows supported by derived temporal constraints) supported by a graphical tool presented in chapter 7 ensures the precise process instance graphically and therefore considered for the transformation purposes.

However, there has been no effort made to transform BPMN and UML-AD together within the literature and makes this work unique and contribution to the knowledge in the field of modelling business processes (BPs) and patient flows (PFs). The graphical notation used in PG* is simple comprising of nodes and edges that are fundamentally alike to BPMN and UML-AD. Thus, PG* seems a good choice for transformation purposes that could improve the overall process modelling. In addition, this research aims to provide a platform that is general enough to analyse and unify both the techniques.

In order to proceed with the transformation, I would show the transformation of the UML-AD (action, start/end of the activity, control flow) and BPMN (task, events, gateways) core-modelling constructs to the enumerated modelling artefacts of the axiomatics systems, exhibited in Table 8.1.

Notations		**Transformation**	**Framework**	**Properties**
BPMN	**UML-AD**			
Start Event/ End Event	**Initial/Final Node**		**Special atomic process**	• Primitive • Structural • Temporal
Intermediate Event	**None**		**Special atomic process**	• Primitive • Structural • Temporal (point)
Task	**Action**		**Atomic process**	• Primitive • Structural • Temporal (moment)
Process/ Sub-Process	**Activity**	⟷	**Process/sub-process**	• Primitive • Structural • Temporal (interval)
Exclusive Gateway	**Decision/ Merge Node**			
Inclusive Gateway	**None**		**Derived Temporal Constraints**	**Structural Temporal**
Parallel Gateway	**Fork/ Join Node**			

Table 8 1 Transformation

By mapping the modelling standards' conceptual schema to the axiomatic system' modelling artefacts, gives me an opportunity to transform their modelling constructs to PG*.

As mentioned in chapter 7 that PG* has the ability to analyse the models constructed in other conceptual based schemas such as UML-AD and BPMN. To achieve this, I would lay down basic guidelines for the modelling standards frequently used modelling artefacts for their transformation into PG* in the next sub-section.

8.1 Transformation Guidelines

So far the axiomatic system provided the formal semantics for the generalised terminologies that are fully aligned with the most often used terminologies of the UML-Ad and BPMN to provide correct process description. In addition, PG* based on PITL is used to represent the correct process description graphically and further analyse the constructed model. Therefore, it is required that each UML- AD and BPMN construct should be mapped well into PG* corresponding element supported by the formal semantics provided in the axiomatic system based on the following guidelines.

8.1.1 UML-AD-Executable Node (Action)

In UML-AD, executable node used to represent Action, and graphically represented as a round-cornered rectangle. Where an activity is graphically similar but consists of several actions. However, axiomatics system provides a general terminology known as 'atomic process' (bearing formal semantics) provided in chapter 6 lays down a logical basis to express action. Therefore, from here onwards, I will use the term atomic process, which is supported by a PG* representation. In PG* a pair of vertices showing corresponding duration (start and end special atomic processes) represents a non-divisible atomic process. Figure 8.1 shows the transformation.

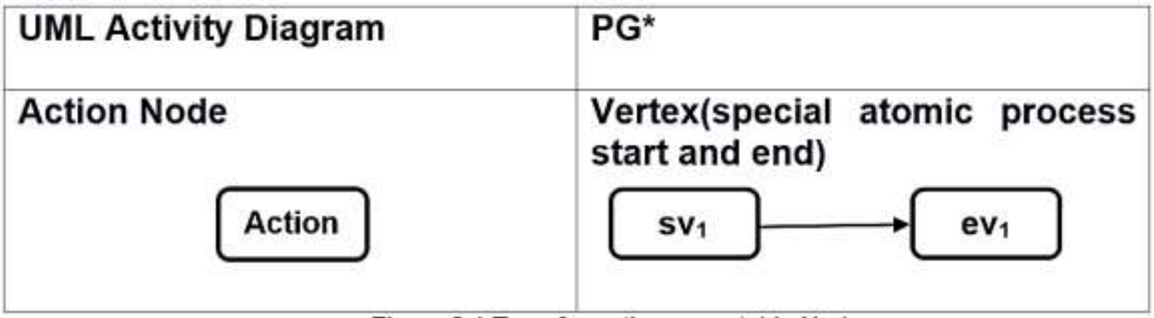


Figure 8 1 Transformation of executable Node

Moreover, an activity of UML-AD notation expressed as a combination of several actions. The formal semantics of 'process; provided in chapter 6 ensures the aforementioned UML-AD narrative, and hereafter the term 'process' will be used instead. In addition, with the support of formalised graphical structure, an action/activity is transformed into PG*. Similarly, PG* provides added quantitative and qualitative information equipping modellers for further reasoning concerning a model.

8.1.2 UML-AD-Edge

In UML-AD edge can be represented as a solid arrow between executable nodes (actions) and Activity nodes (activities) to depict their flow. The framework provided in this thesis (chapter 6) provides a formal definition to express temporal relationships between the atomic processes/process (start and end events) representing corresponding duration. Hereafter the term temporal relations will be used to depict the edge.

However, PG* based on PITL used 'edge' to express the temporal relationship between atomic process instances and process (sub-process) instances associated with respective quantitative temporal information (a length function 'Dur'). In addition, PG* edge also used to expresses the PITL extended qualitative temporal relations (before or precede) between the two vertices as $(sv_1 < ev_1)$ using solid arrow or $(sv_1 \leq ev_1)$ using broken arrow in describing the consistent process flow. The formal semantics provided by the framework associated with its diagrammatic modelling artefacts provided by PG* has more to offer with its precedence relation, therefore, ensuring a smooth transformation of UML-AD edge to into PG* as shown in figure 8.2.

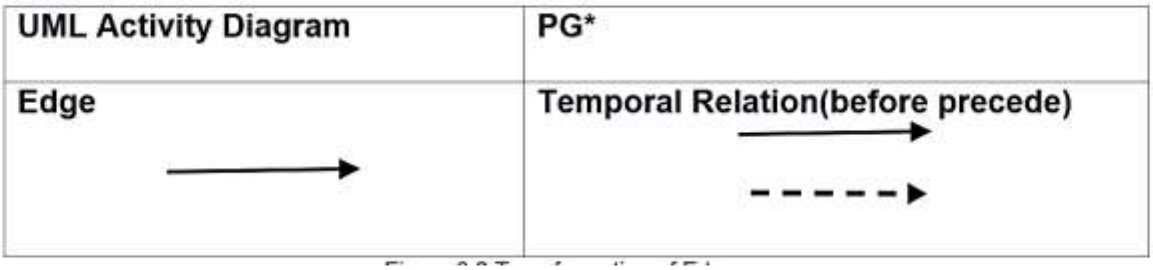


Figure 8 2 Transformation of Edge

8.1.3 UML-AD-Initial/Final Node

To represent the beginning and finishing of an activity, UML-AD provides two most often used control flow constructs termed as the initial node and final node. These graphical constructs are drawn as fully blackout circle and solid circle inside a circle respectively. But no formal semantics provided and as a result, the framework developed here has filled the gap by providing the formal definition to express the start and end of a process instance. The axiomatic system modelling artefact known as special atomic process has the capability to represent the start and end of a process (including atomic and sub-process) instances fully aligns with the initial and final nodes.

The formal semantics of special atomic process is supported by the PG* source and sink nodes. These nodes graphically represent the special atomic process that may or may not have a stamp attached. Graphically, they are drawn as two rounded rectangle vertices (V_{in} and V_{out}). Thus with the support of formal semantics provided in the axiomatic system associated with its graphical construct respective transformation is [erformed shown in figure 8.3.

UML Activity Diagram	**PG***
Initial/Final Nodes 	**Source/Sink Node** V_{in} V_{out}

Figure 8 3 Transformation of Initial/Final nodes

8.1.4 UML-AD-Decision/Merge Nodes

Modellers have the discretion to choose the constructs to represent the process flow in general and conditional flow specifically. In UML-AD, a decision or merge is represented by a diamond shape facilitating the branch/merge flow within an activity diagram supported by a guard mechanism representing the associated conditions. To represent a decision, a token from inflow edge is transported to one of the several outflow edges i.e. mutually exclusive, that fulfils the condition (guards). The same diamond construct used to express merging of the inflows resulting in one outflow but with no synchronization. The lack of formal semantics leaves a gap for the researches and industry to fill. Therefore, the framework developed here makes provision for a formal semantics using the derived temporal relationship definition incorporating extended PITL relationships between its modelling artefacts (atomic, sub-process and process instances).

The derived temporal constraint definition provides aid to the PG* in representing the conditional behaviour between the modelling artefacts such as atomic process instances (actions), sub-process/process instances (activities) and corresponding special atomic process instances (events) using binary operation $\oplus$. To express decision and merge in PG*, it will show more than one flow depending upon the corresponding temporal information. PG* renders the unification mechanism after branching presenting a compound vertex, therefore decision/merge transformed into PG* as shown in figure 8.4

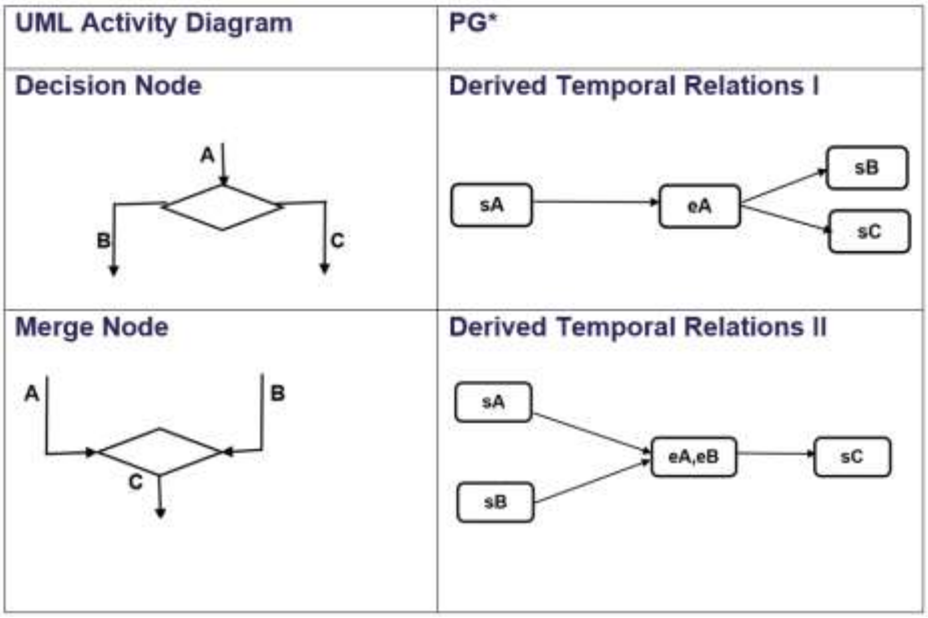


Figure 8 4 Transformation of Decision/Merge nodes

8.1.5 UML-AD-Fork/Join Nodes

In UML-AD, the concurrency is graphically represented as fully black-out bar which can be used either horizontally or vertically. However, the lack of formal semantics created an opportunity to fill in the existing gap. The derived temporal relations make provision for its formal semantics, further supported by the binary operand $\otimes$ ascertaining the concurrent behaviour within a process instance. Graphically, PG* supports the formal semantics provided in the framework to represent the parallel flow Thus, fork/join can be transformed into PG* as shown in figure 8.5.

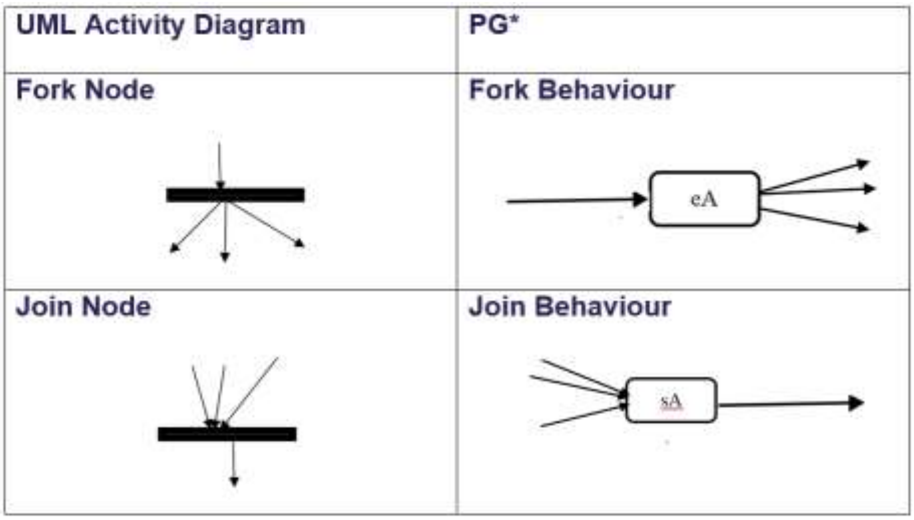


Figure 8 5 Transformation of Fork/Join nodes

8.1.6 BPMN-Task

A business process diagram made up of a variety of BPMN elements. For the transformation sake, I consider only most often used modelling components of BPMN. It uses the term activity that can be atomic (task) or compounded (sub-process). BPMN use the term task and determine its semantics by relating it to a unit of work that is graphically represented by a cornered rectangle. However, there are different types of tasks bearing the same semantics provided by BPMN such as service task, user task, send and receive tasks, script task etc. Mainly, I will be considering the general task and its corresponding graphical construct for transformation purposes. As the formal semantics is missing in the standard documentation that gap is filled in by the method developed in this research providing the formal semantics.

The framework renders a general term termed an 'atomic process' and formally defines it that has all the functionality to determine a unit of work and the corresponding structure. Furthermore, PG* utilises the formal definition and graphically represents it by a pair of vertices (expressing its start and end temporal information). Hereafter, I use the term atomic process to represent the task expressed as a pair of the vertices and transformed as shown in figure 8.6.

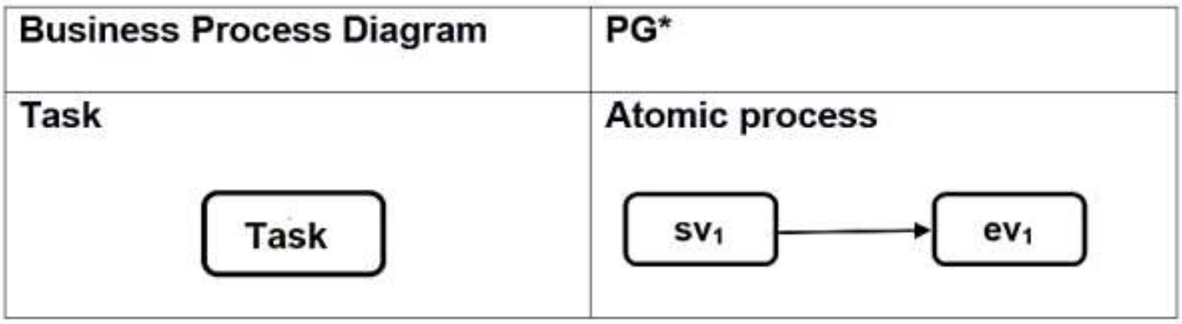


Figure 8 6 Transformation of Task

8.1.7 BPMN-Event

BPMN standard documents a task/sub-process that is initiated with a signal provided to start (start event) and end (end event) it. In addition, there are other events available in the standard such as the message events that only used for communicating messages and not as start or end events. Other events types such as timer, only used to indicate a certain time or date reached where error event (intermediate event) used for sending error signalling during a process. The purpose of these constructs is to either, alter or complete a flow termed as start, intermediate and end events. However, a formal semantics required for the consistent representation of a process in which they are utilised.

The semantics of special atomic process provided in the framework (chapter 6) fills the gap. PG* renders the formal definition of special atomic process and associates it with the source (V_{in}) and sink nodes (V_{out}) to show the start and completion of a process/sub-process instances. Thus, event is transformed into PG* as shown in figure 8.7.

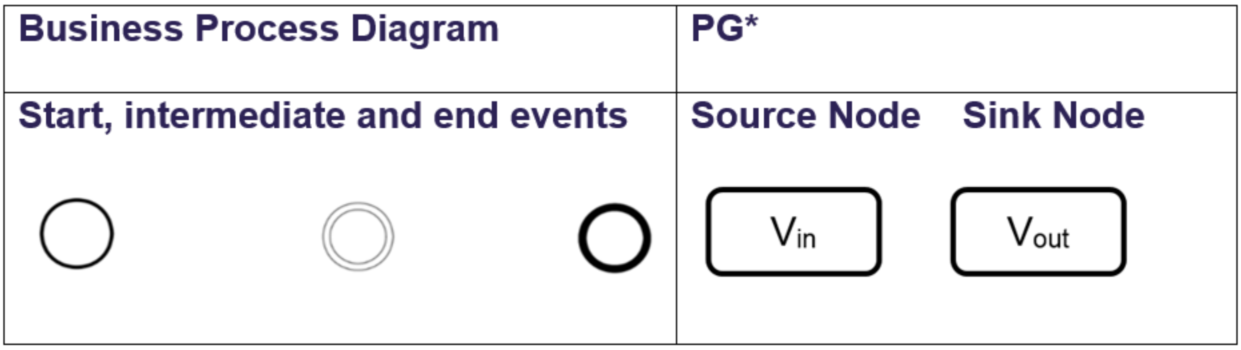


Figure 8 7 Transformation of Events

In addition, BPMN uses an exception, the intermediate event during a normal process flow to delay the whole process. PG* has the facility to accommodate this event type with the dummy activity that may also be used to express the lower and upper boundaries of a process instance.

8.1.8 BPMN-Sequence Flow

As explained in chapter 4, BPMN used sequential flow to show the normal flow between different tasks, process or sub-processes. It is graphically represented as a solid arrow. The framework provides formal semantics using derived temporal constraints to express the flow. PG* renders this definition and associates it with its ‘edge’ to display the connection between two nodes.In PG*, edge is labelled with associated temporal information representing the duration between atomic process, sub-process or process instances. Such facility further assists modellers in analysing the path for reasoning. Hence, sequential flow has been transformed shown in figure 8.8.

Figure 8 8 Transformation of Sequence flow

8.1.9 BPMN-Sub-Process

As we have seen from the discussion provided in chapter 4 for the BPMN terminology sub-processes intuitively described to show a flow of a combination of several tasks that is part of a parent process. The sub-processes are of different types but the overall operation represents several units of work within their corresponding parent processes of an enterprise. Due to the clear semantics, no provisioned by the BPMN standard makes it cumbersome for modellers to construct a coherent and consistent process model. Although graphically, BPMN represents a sub-process (expanded) to show the details, on the other hand, collapsed sub-process only show the label but no detail. In addition, the graphical constructs used to represent both process and sub-process is a cornered rectangle similar to task only with the addition of '+' sign for expanded sub-process.

This gap is filled in by the framework developed in this thesis making provision for formal semantics to sub-process of the axiomatic system that is fully aligned with the BPMN sub-process. However, the definition provided shows the sub-parts of the main process and like BPMN sub-process. HPG* utilise the formal definition of sub-process and graphically represent it using several vertices and edges to display the flow that is broken down further. HPG* ensures that an instance of a sub-process only occurs once at one given time and does not repeat itself before its completion. Hence, sub-process is transformed as shown in figure 8.9.

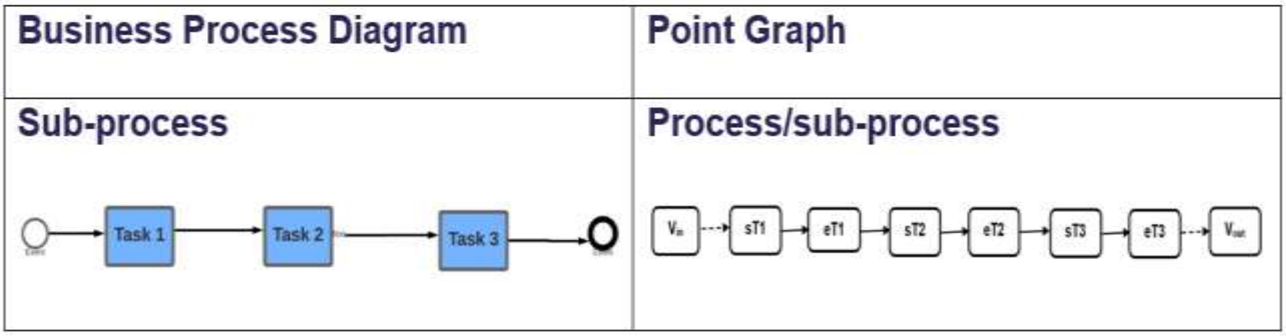


Figure 8 9 Transformation of process

8.1.10 BPMN-Gateways

To show the routing in the BPMN diagram, gateways used to depict conditional flow such as convergence or divergence within a process. As discussed in chapter 4, for transformation purposes, I will be considering basic gateway constructs i.e. exclusive (XOR split/merge) inclusive (OR split/merge) and parallel (AND split/join) gateways. Rest of the constructs are left-out as mentioned in chapter 4 to be within the scope of this research. BPMN utilises diamond shape graphical construct for each gateway with a different market to express the functionality attached (intuitive). For example, an ‘X’ marker utilises to express the exclusive behaviour, ‘O’ rendered for inclusive process flow representation, however, marker ‘+’ adopted for the parallel behaviour depiction.

The formal definition provided in chapter 6 of derived temporal constraints and binary operations $\oplus$ and $\otimes$ makes provision for the formal semantics of XOR, OR, parallel gateways. The diagrammatic tool adopted in this thesis support such definitions (provided in the framework) and accordingly graphically represented them facilitatating the transformation of the BPMN gateways (presented above) into PG* shown in figure 8.10.

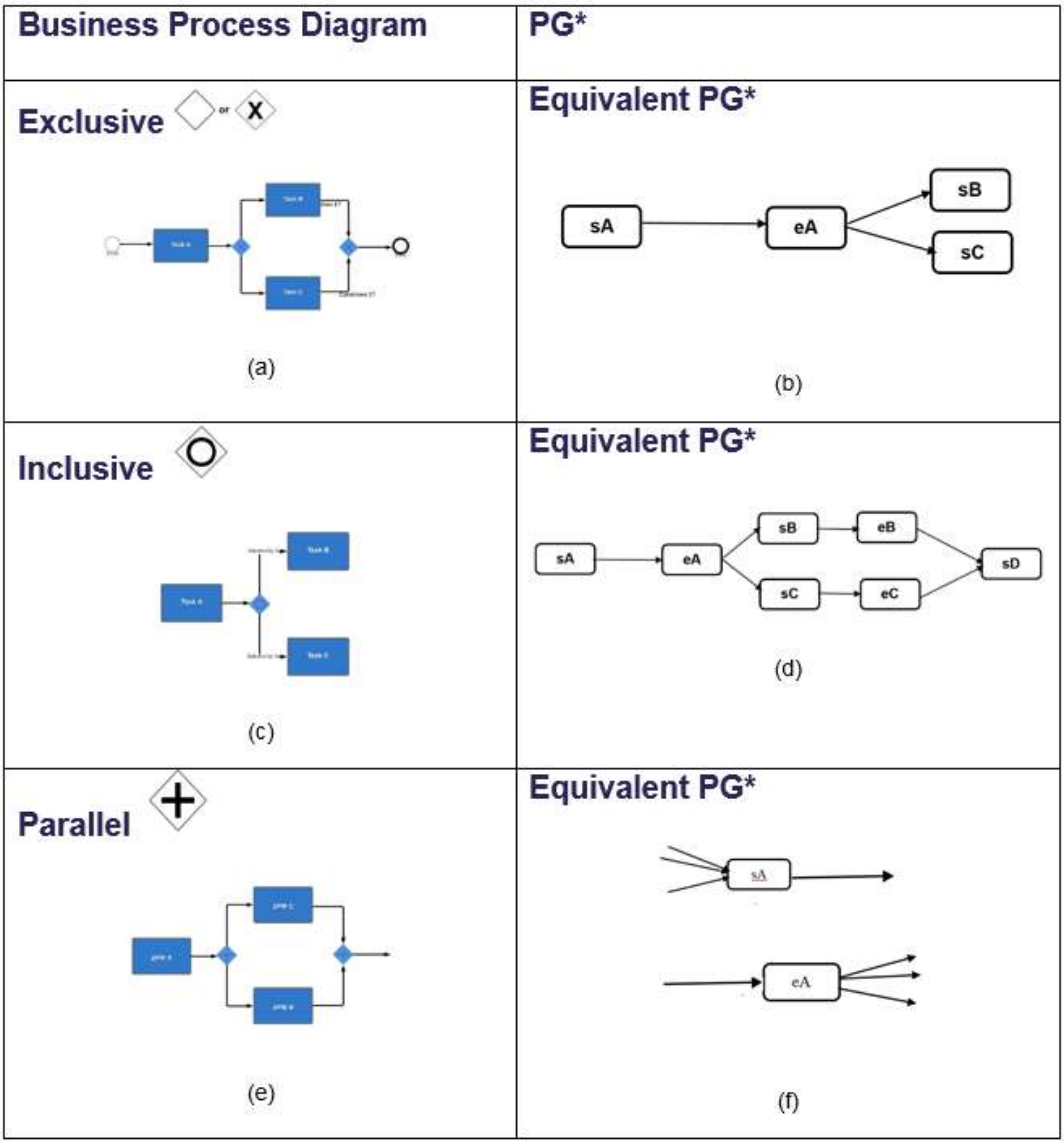


Figure 8 10 Transformation of Gateways

8.2 Discussion

The investigation during this thesis identified a set of core modelling terminologies and constructs of both UML-AD and BPMN that are identical in their functionalities. For example, an activity of UML-AD expresses a business process where 'action' is considered an atomic element of the activity represented by a cornered vertex. Whereas, BPMN utilises the terminology of a task to represent an atomic part of a process that is comprised of sub-processes (term used in BPMN).

The similarity between UML-AD and BPMN functionality in handling different control flows is eminent. Although they use different notations such as to depict branching behaviour UML-AD use the terminology 'decision/merge' supported by a construct (diamond) and BPMN resorts to an exclusive gateway construct. Also, for concurrent behaviour representation, UML-AD utilises the fork/join terminology supported a construct (solid bar) and BPMN is serviced with the parallel gateway construct utilising a diamond shape with a '_' marker. The discussion shows that the control flow nodes of UML-AD and gateways of BPMN serve the same purpose of branching and parallelism.

The axiomatic system defined these terminologies and supported by graphical representation (PG*). Combination of nodes of PG* represents either atomic process instances (if unbreakable) or process/sub-process instances (if breakable). Both extremes of an atomic process and/or sub-process/process expresses the special atomic process instances. In addition, the flow between them representing qualitative and quantitative temporal information. PG* renders the axiomatic system enable expression of the concurrent behaviour of a process instance by modelling the constraints implicitly using edges directly coming in/out from the atomic process nodes and removing the fork/join node and parallel gateway. The implicit representation of concurrent behaviour of atomic processes using PG* to depict parallel (in/out) edges and must satisfy the temporal constraints attached.

However, PG* represents the axiomatic systems' process instances and their exclusive behaviour by modelling them implicitly using edges directly coming in/out from the atomic process instance nodes and removing the decision/merge node and exclusive gateway. The implicit representation of exclusive behaviour of atomic processes using PG* to depict branching and merging (in/out) edges and must satisfy the temporal constraints attached. The multiple-choice (OR) representation in UML-AD is dealt differently for which it utilises fork node by defining guards associated with edges to control and specify branches of an action node. However, BPMN has inclusive gateway to depict the multiple-choice behaviour. The axiomatic system based temporal constraints provisioned such behaviour based on its exhaustive temporal constraints supported by PG* to depict multiple-choice behaviour.

8.3 Transformation Illustrations

The transformation procedure completed and requires an illustration for its applicability in real-life to determine the objective has been achieved. The framework developed has the facility to incorporate the absolute and relative temporal associated with the atomic, special atomic, process and sub-process instances. This facility s equips modellers to further analyse the corresponding process models constructed in the process modelling standards, to determine their correctness.

For this, I have considered two examples i.e. 8.1, and 8.2, to show the sequential and parallel flow respectively within a process modelled in UML-AD and BPMN. These examples have associated qualitative and fictitious absolute temporal information to enable the modeller to provide enhanced reasoning concerning the model constructed. Thereafter, examples 8.1 and 8.2 are converted into equivalent PG*. As a result, the equivalent models constructed in PG* presents more details for reasoning purposes to determine its correctness.

Example 8.1: Consider three atomic process instances (actions/Tasks) A, B and C that are executed sequentially. The qualitative temporal information provided display the sequential flow between these atomic process instances and modelled in UML-AD and BPMN respectively. With the provision of formal semantics provided in the framework states that an atomic process instance only initiates when its preceding atomic process is completed. Therefore, in his sequential flow example an atomic process instance B only runs when atomic process instance A is completed and before atomic process instance C is initiated. An equivalent PG* is drawn to show the transformation procedure is accurate as shown in figure 8.11.

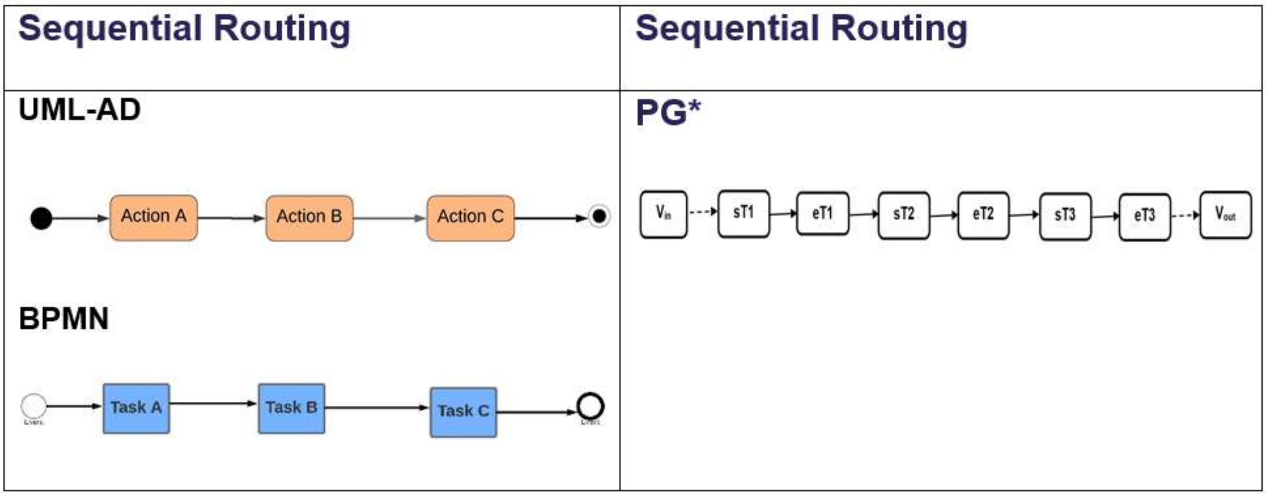


Figure 8 11 Transformation of sequential routing example

The above example also establishes that the transformation performed in this thesis has unified both the modelling standards most often used constructs. In addition, PG* provides a clear structure and boundaries of the involved atomic processes satisfying its associated formal semantics provided in chapter 6 supported by its graphical representation (chapter 7).

Example 8.2: This example is considered to show the concurrent flow within a process. Therefore, to construct a concurrent flow of five atomic processes A_1, A_2, A_3, A_4 and A_5, with associated conditions such as atomic process instance A_1 has three outgoing parallel atomic process instances A_2, A_3 and A_4. The atomic process instances A_2, A_3 and A_4 are merged proceeded by the atomic process instance A_5. It represents the concurrent flow of a process instance constructed initially in UML-AD and BPMN. An equivalent PG* drawn by analysing UML-AD and BPMN establishing transformation procedure correctness shown in figure 8.12.

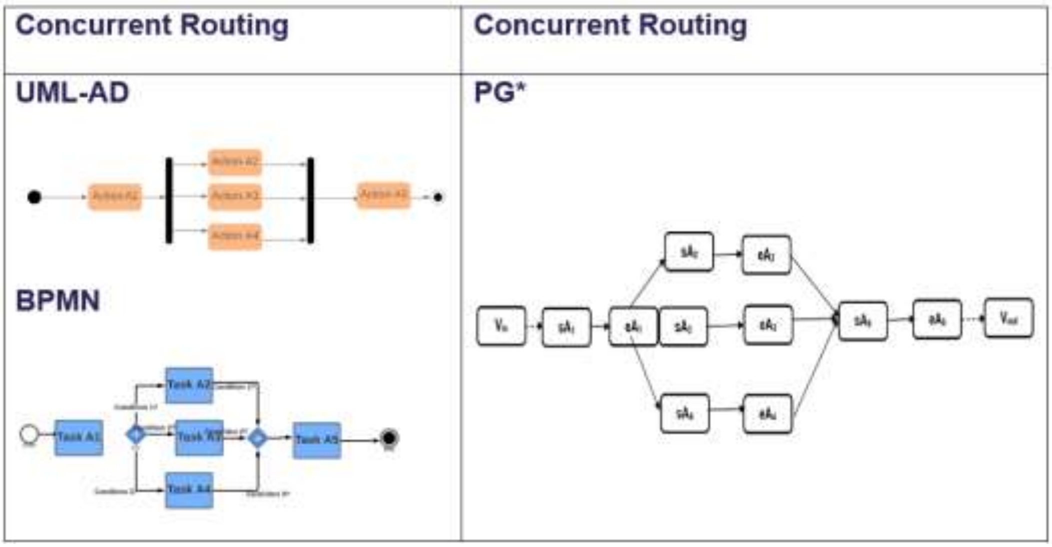


Figure 8 12 Transformation of concurrent routing example

The similarity in both the modelling standards' functionality expressing the concurrent behaviour made it possible for their unification via the state of the art framework developed here in this thesis. In addition, I have noticed that UML-AD and BPMN have lack of additional information needed for a practical model to express enhanced temporal information. This means both notations fails to specify the structure of individual modelling element which is required for precision. Where PG* fills the gap and provides distinct structure establishing the precise start/completion times of the atomic process instances involved. With the assistance of the framework, PG* can facilitate in representing such properties.

Moreover, PG* is equipped with 'FindPath' algorithms to determine a special atomic process/atomic process/process instance' lower bound and upper boundaries. Both UML-AD and BPMN lack in providing such extensive qualitative and quantitative temporal information, which is desirable, for patient flow modelling in finding different patient pathways.

8.4 Summary

The previous chapters have laid a path steering towards the transformation of both the business process modelling standards (UML-AD and BPMN) into the formal approach devised in this thesis. Therefore, this chapter focused on providing a mechanism to perform mapping between the UML-AD and BPMN into PG*. To perform transformation of an informal modelling tool to a formal approach requires intuitive tool to have formal standing for precise alignment. But both UML-AD and BPMN have enormous amount of terminologies mandated by graphical constructs to represent processes and its sub-components. Therefore, chapter 5 and 6 facilitated the identified most often used terminologies with formal definitions to lay down a bridge for their transformation to a formal approach.

The enumerated modelling artefacts formal semantics can be associated with the most often used terminologies of the UML-AD and BPMN. For the transformation purposes, most often used artefact of the UML-AD (Executable Node, Edge, Initial and Final Nodes, Decision and Merge Nodes, Fork and Join Nodes) and BPMN (Event, Task, Process, subprocess, Sequence Flow and gateways) re-visited for their better alignment with the artefacts 9formally) defined in the framework. The axiomatic system has the capability to subsume the most often used notational artefacts by the set of the terminology introduced (and formally defined) here in this thesis.

An extended visual approach PG* provided in this thesis rendered for the diagrammatic representation of the formally defined terminologies (axiomatic system). That helped in filling the existing gap in the literature. In addition, the visual approach has its formal translation into point and interval temporal logic and therefore nicely blend into the developed framework for mapping purposes. A transformation carried out to map the most often used artefacts of both UML-AD and BPMN into the framework defined terminologies. PG* provides not only simple graphical elements to represent a typical business process but also supported by the unification, branch folding and join folding algorithms for determination of the correct model.

Transformation essentially maps the most often used graphical constructs of both the UML-AD and BPMN individually (supported by the formal semantics provided in the axiomatic system) to the PG* formal graphical constructs. Furthermore, the transformation performed to assist in expressing the structural properties as an additional feature to show the wider scope of the framework. Subsequently, in a modular fashion, individually UML-AD and BPMN most often used notation discussed and transformation guidelines provided for their translation into framework and then graphically into PG*.

I have also provided a discussion to support the transformation bundled with a couple of examples to examine different behaviour using the standards compared with PG*. Therefore, examples provided were constructed initially in the UML-AD and BPMN. This procedure has assisted me to manually examine the constructed process models that lacked in expressing its distinct structure. After the analysis of the constructed model, I was able to build an equivalent PG* establishing what benefits are provided by the method developed in this thesis

By specifying a precise process and its sub-parts through framework equipped with a formal PG*, I have shown the explicit transformation of the most often used modelling artefacts of bot the standards into PG*. In addition, an equivalent PG* enable modellers to further reason concerning a process model performing ‘what-if’ scenarios for establishing its consistency. This effort deemed necessary for the patient flow modelling specifically where complex patient flows are involved requiring suggestions for adopting different pathways.

The next chapter will discuss an application to fully express the capabilities of the method that is carried out at the King’s College Hospital accident and emergency department to model their patient flows. Additional features such as scheduling and applied constraints applications considered for the project management to show the benefits of the approach developed in this thesis.

Chapter 9 Application

So far, I have provided a walkthrough by developing the framework including its verification and validation to establish its benefits and edge over the existing modelling standards. Therefore, the framework developed can use existent knowledge of objects concerning a real-life domain to describe them formally based on their temporal nature. By doing so, an order of the objects can be generated to represent the flow of communication between them. In addition, the framework equipped with the analytical capability to identify any inconsistency concerning the constraints attached to each modelling artefact description (purpose) and its flow based on their temporal existence.

The axiomatic system relies on the extended PITL used here for its inference mechanism to employ the derived temporal constraints concerning a process instance with respect to its constituents (the special atomic process instances and atomic process instances). Hence, with this facility, stakeholders are equipped with the power to test their various viewpoints concerning a flow on a timeline using this approach.

For the sake of readers' convenience, I have re-iterated the procedure of drawing a PG* here. I consider a set of extended PITL statements to represent the process instances (atomic, special atomic, sub-process instances) of the axiomatic system and apply the operational constraints (chapter 7) that needs to be satisfied for the consistent display. The resultant PG* representing a complete process instance, i.e., patient flow (entire set of temporal statements) would apply the unification algorithm by inspecting the quantitative temporal values of the special atomic process instance (stamp) attached with process (atomic process) instances for their equalities and combined them to be represented as a single vertex.

The next step would require the unified PG* (representing a process instance) to be folded. While folding a PG* inferences can be made using the involved process instances quantitative temporal (duration and their stamp) values to draw out derived (new) relationships between the different process instances. The process (instance) modelled with the information provided by different sources that may result in a conflict of the constraints (qualitative information) and hence inspected for its consistency, i.e. loop or pair of vertices representing several flows with conflicting duration. After removing the conflicting flows and constructing a verified PG* (consistent), two virtual nodes, i.e. source and sink nodes (V_{in}/V_{out}) added and linked the sub-parts of the process instance involved with less than equal to edges (LE). They represent the total duration of a process instance i.e. quantitative temporal information).

The framework has the capability to be applied to any real-life domain where the services considered to be delivered in an improved manner. Nonetheless, every domain is critical to show the application of the framework but I have chosen the healthcare domain. The reason behind this choice is the challenge presented by the ever-changing nature of patents' needs within a hospital environment especially in the more demanded and burdened department (with regards to resources including time) of any hospital which is accident and emergency. Therefore, I have chosen King's College Hospital Trust for the application of the framework.

9.1 Data Gathering

King's College Hospital, a National Health Service (NHS) foundation trust approached with the intention of data collection and showing them a novelty of the approach that can make difference for them in handling the patient flows. For this reason, they have provided us with documentation such as process maps (see appendix I for a sample) of their different departments. No personal data included in any of the documentation, therefore, no ethical approval required for the study. To be within the scope of this research showing the authenticity of the framework, I have chosen the accident and emergency (A&E) department trauma pathway due to its challenging nature.

A&E department of King's College Hospital faced the increased influx of the patients on daily basis but this level reaches its peak during festive season but I am not going to delve into the statistical comparison which may be important but not the focus of this study. However, my focus is to establish the patient flows through the system efficiently keeping the associated times to its minimal.

Therefore, to proceed with the analysis, I required the data first from them that is obtained via a set of meetings with the concerned staff to explore the depth and breadth of the information retained by them to construct model the patient flows at the A&E. In addition, to the information gathered by visiting the hospital on many occasions, I have acquired information about their current modelling capabilities and expertise. For this reason, the informal interviews carried out with the domain experts at the hospital so to paint a clear picture of the whole operation concerning trauma patient flow handling.

Moreover, they utilise the information for its transformation into UML-AD and BPMN models respectively to express the flow in detail. The initial process map acquired depicts a similar picture compared with the newly formed models using UML-AD and BPMN. Hence, the benefits acquired from both the modelling tools express their limitations with respect to individual modelling artefact' structure and their representation with regards to associated temporal information.

Patient flow modelling display the journey of a patient concerning service and considered sensitive, therefore, it is important to breakdown the patient flow into smaller parts for exhibiting associated detail. Furthermore, it could help in analysing different parts with regards to their importance of existence within a patient journey concerning time limitations attached. Therefore, the rest of the chapter shall examine the trauma patient flow constructed in modular fashion by the domain experts that subsequently evaluated using the framework for any inconsistency. For this research, I have considered the three different patent flows concerning the trauma patients due to its complex nature within the accident and emergency department to illustrate the approach. The three patient flows selected were

i) Trauma patient arrival by any of the three possible ways to reach the A&E department of the King's College Hospital. This patient flow discussed in illustration I in detail.

ii) The next flow of the patient pathway considered requiring surgery (major trauma) discussed in illustration II.

iii) The third and final flow considered as part of the trauma patient pathway is the discharge of the patient (major trauma) discussed in illustration III.

These patient flows were modelled by the domain experts at the hospital and critically evaluated using the framework. The axiomatic system utilised to transform the three patient pathways modelled in UML-AD and BPMN into PG* for their analysis concerning the involved atomic process and special atomic instances' relative and absolute temporal information availability from the King's College Hospital Trust. Furthermore, the application of the framework by constructing an equivalent PG* sheds light on the need of the hospital to manage time-bound patients at a very critical organisational unit, i.e. A&E. The approach developed makes it possible to transform the process (instance) model in modular way. By applying the method, I would be able to see the validity of the existent knowledge used to model patient flows and analysed the constructed models for any inconsistencies (if any).

That is subsequently resolved with the help of the mechanism provided in the framework developed in this thesis.

My aim is to show that both relative and absolute temporal knowledge critical for constructing the precise structure of patient journey detailing both at high and low-level abstraction. Illustration I and II selected to emphasis the importance of relative temporal information in providing enhanced reasoning because both selected modelling standards lack this ability, however, it is found critical for modelling patient flows. Moreover, illustration III not only would investigate qualitative but most importantly quantitative temporal information acquired from the King's College Hospital to further schedule and optimise the process model.

9.1.1 Trauma Patient Flow Illustration I

A problem statement generated from the accident and emergency department of King's College and presented below for the application of the framework.

9.1.1.1 Problem description

A trauma patient can arrive at accident and emergency (A&E) either via ambulance, walk-in or brought in someone. These arrivals constitute three parallel paths, which are:

i. The trauma patient with minor injuries walked into A&E triage, and in general, the patient is seen by a specialist nurse followed by a consultation with a consultant. In case the patient requires further investigation then the patient is transferred to the ward. Some tests such as MRI, CT Scan etc. may be carried out during the stay in the ward. It would lead to a treatment, and ultimately the patient is discharged.
ii. The trauma patient who has driven into A&E by someone with minor, major injury could be seen directly by a consultant especially if the hospital has been notified prior to the arrival such as via 111 services. In general, the reported patient has recorded in the system transferred from 111 and could be referred to the high dependency unit (HDU) based on the initial analysis at the hospital. The patient could there either die or get better to be transferred to a general ward and after that discharged.
iii. A trauma patient brought in to A&E via ambulance with a major injury. The patient condition is critical and requires urgent attention from a consultant (clinical staff).

The patient needed an intervention and sent to the operation theatre for emergency surgery. After treatment, the patient would typically be discharged.

The above scenarios modelled using with UML-AD and BPMN as shown in figure 9.1 and 9.2 respectively

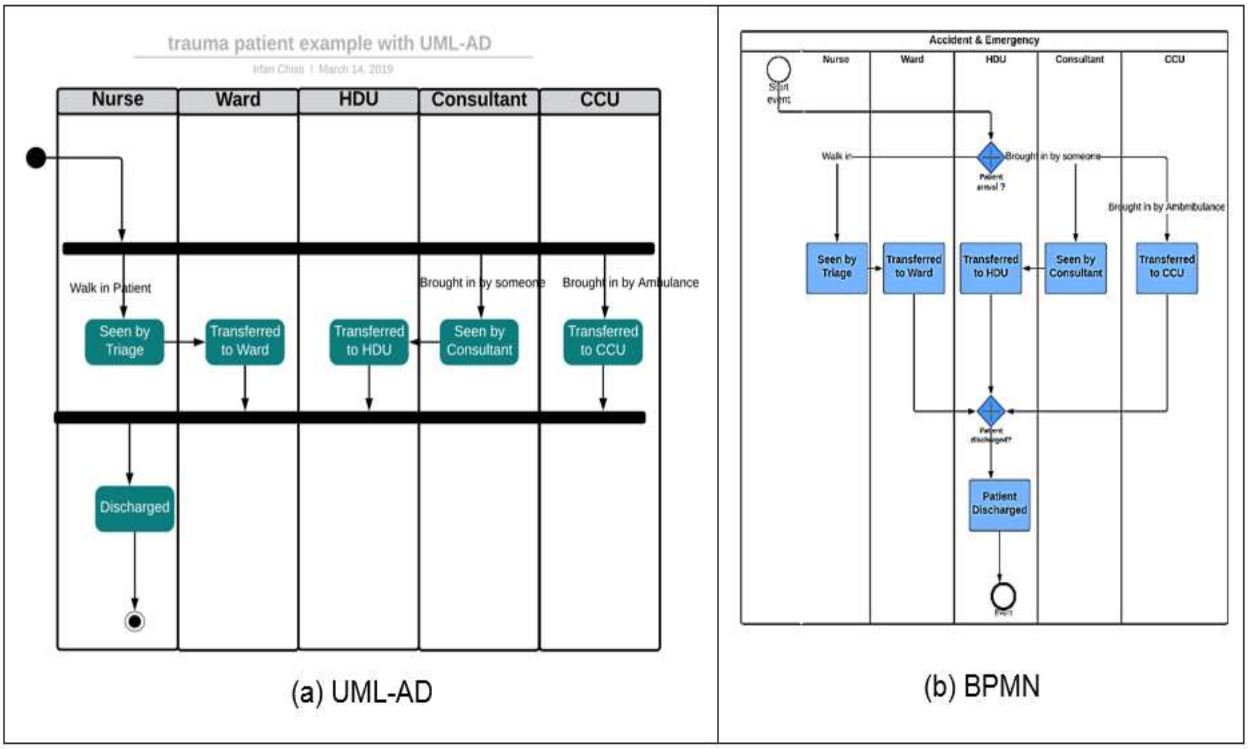


Figure 9 1 Trauma patient flow example modelled in UML-AD & BPMN

9.1.1.2 Critique

Both the standards rely on similar constructs as discussed previously n chapter 4, 6 and 7 such as UML-AD use notation of ‘swimlane’, and BPMN uses the notation ‘pool and lane. These constructs are utilised for organisational roles specification only and have no impact whatsoever on the behaviour of the diagrams. The above models depict concurrent flow using a fork and join (UML-AD) and parallel gateway by the BPMN. The occurrences of individual actions and tasks are drawn. However, the observations made to analyse and evaluate these models results using the framework in identifying the inconsistency present in both the models. With a naked eye, it is not possible, but with the application of the framework, I would be able to provide insights into the problem described and analyse the constructed models for reasoning purposes.

9.1.1.3 Enhanced Reasoning

At the A&E department of King's College Hospital, there are various combinations and permutation of getting access to the consultants, nurses, diagnostics, theatres, wards, critical care units etc with respect to the time limitation. In addition, the concurrent behaviour modelled using both the modelling standards relies on the notion of the token pass and follow a path after a token received. Due to their intuitive basis, they failed to specify the order in a concurrent flow of a process.

Keeping this in mind, both the above diagrams failed to equip models with such information to reason and represent trauma patient flow. With the help of relative temporal constraints discussed in chapter 6, I could conveniently analyse the above graphs and further provide enhanced reasoning concerning modelled trauma patient flow. To provide reasoning on the modelled trauma patient flow, I have extracted the parallel path section from both the models to analyse and reason further as shown in figure 9.2 below.

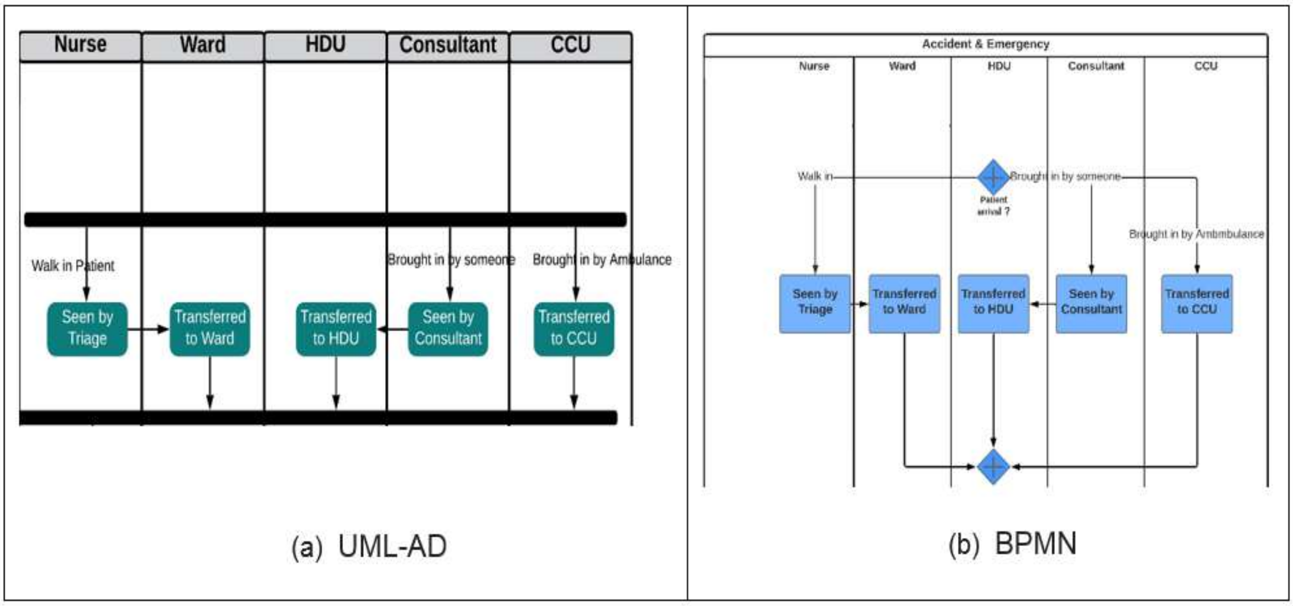


Figure 9 2 Concurrent flow extracted from Trauma patient flow

The parallel flow shows there are three actions/tasks such as 'Seen by a 'triage', 'Seen by consultant' and ' Transferred to CCU'. Additionally, there are two following actions/tasks ('transferred to the ward' and 'transferred to HDU' preceded by 'seen by triage' and 'seen by consultant' respectively. But the actual order is not described by the process specification.

The token pass procedure has its pitfalls while modelling to provide reasoning, if and when things changes, which is likely in the hospital environment. Therefore, such provisioned should be accommodated by the standards but both UML-AD and BPMN do not provide any mechanism to facilitate changes. Thus, the models constructed are insufficient and offer no significant value to the knowledge to the stakeholders. In addition, the semantics of the modelling standards lack to support in building a consistent and correct model through a verification mechanism. And the approach developed addressed such issues by not only constructing a precise model that is semantically correct but supported by a verification and validation mechanism. Similarly, it also equips the modeller and other stakeholders to reason the built model with the support of the inference mechanism by deriving enhanced relationships between any given process instances (start/end) to determine any uncertainty given it resulted in a consistent result.

To establish the authenticity and benefits of the method, I would construct an equivalent PG*. To proceed with transformation, initially, I would convert the above knowledge into natural language representation to show the current (available) temporal relationships and then deriving the connections from the existent process instances. Thus, the temporal information based on PITL statements from the above example expressed in natural language as given in Table 9.1.

Process Symbol	**Natural Language Description**	**Qualitative Relationships**
A_1	The patient was seen by a consultant with minor trauma	A_1 meet A_2
A_2	Transferred to ward for diagnosis & treatment	-------
A_3	The patient was seen by a consultant with minor, major trauma	A_3 meet A_4
A_4	Patient transferred to HDU for diagnosis & treatment	A_3 precedes A_2
A_5	Patient with major trauma sent to CCU	eA_4 precede eA_5

Table 9 1 Qualitative and quantitative information related to the example

The PITL statements express the structural information of the involved process instances that is enough to construct an equivalent PG* as shown in figure 9.3 indicating the relative temporal relations between any given two special atomic process instances, i.e. start and end, of the process instances.

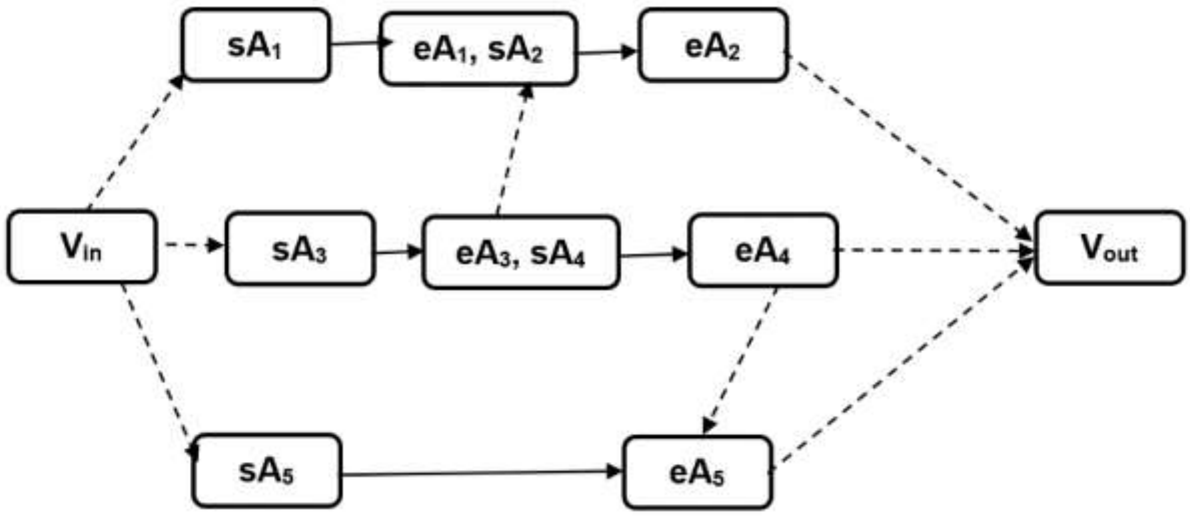


Figure 9 3 Trauma Patient Pathways modelled using PG*

In addition to the normal flow PG* has the capability to show the precedence temporal relation enabling the modeller to depict the corresponding temporal relation to avoiding conflicts. More importantly, in real life, the pathways are interchangeable; a patient can move from one route to another. For example, if a patient-reported with the minor-major trauma and the patient scheduled to be transferred to the high dependency unit (HDU) changes its path considering either patient's condition improves or HDU is no longer required and hence can move to the minor trauma patient pathway. Some of these judgments are subjective to human and machine factors, and because of this, a patient sent to CCU wrongly instead of the HDU or otherwise.

Furthermore, with the assistance of the inference mechanism which is applied briefly, I have managed to infer a couple of qualitative temporal relationships. Figure 9.3 specifies such relations by drawing less than equal to the edge between the process instance 'A_3 and 'A_2' to provide a previous relationship. Also, a special atomic process instance (end) of the process instance A_4 (eA_4) precedes the special atomic process instance (end) of the process instance 'A_5' (eA_5). I have shown a non-exhaustive application of inference mechanism that has reasoned the patient flow and informed the stakeholders with added value information. King's College Hospital has appreciated the provision of such valuable information that could help them to handle these situations in a more sustained way. However, these relationships are not possible to model with the process modelling standards.

Moreover, PG* is also equipped with 'FindPath' algorithms to determine a special atomic process/atomic process/process' instance lower and upper boundaries that could handle more complex situations where the time has already specified for process instance initiation impacting the overall process duration. However, both UML-AD and BPMN lack in providing such extensive qualitative and quantitative temporal representation to model patient pathways, which is required to provide timely services.

9.1.2 Surgery Patient Flow Illustration II

In the illustration I, a trauma patient flow discussed where a patient is brought by the ambulance with major trauma. Due to the condition of the patient with significant trauma transferred to the critical care unit (CCU) as per the initial assessment conducted by the paramedics. Their assessment also indicated that the patient may require surgical intervention for the trauma depending upon further investigation conducted by the relevant clinical staff's assessment and diagnosis.

A surgical process starts with the assessment of the major trauma patient condition to carry out the surgery by the surgeon. The assessment would determine the need for the surgery by evaluating injuries supported by the results of the diagnostic that assist in making the decision. After analysing the diagnostic result obtained from the clinical staff from the different diagnostic units, a decision to operate or not to operate is made.

In case the surgery is not required due to the changes in the trauma patient conditions supported by the assessment referred for further medical treatment and allocated to a bed in the ward and subsequently discharged upon getting better. In other case where an intervention required then the patient is registered on a waitlist (prioritised) to be assigned (booked) to a most appropriate time for surgery at the king's college hospital' operation theatre (A&E).

To proceed with the surgery, a specialist nurse facilitates the patient with necessary information involving surgery. After surgery, the patient moved to a bed in the ward and subsequently discharged. The syrgical patient flow modelled using UML-AD and BPMN is shown in figures 9.4(a) and (b) respectively.

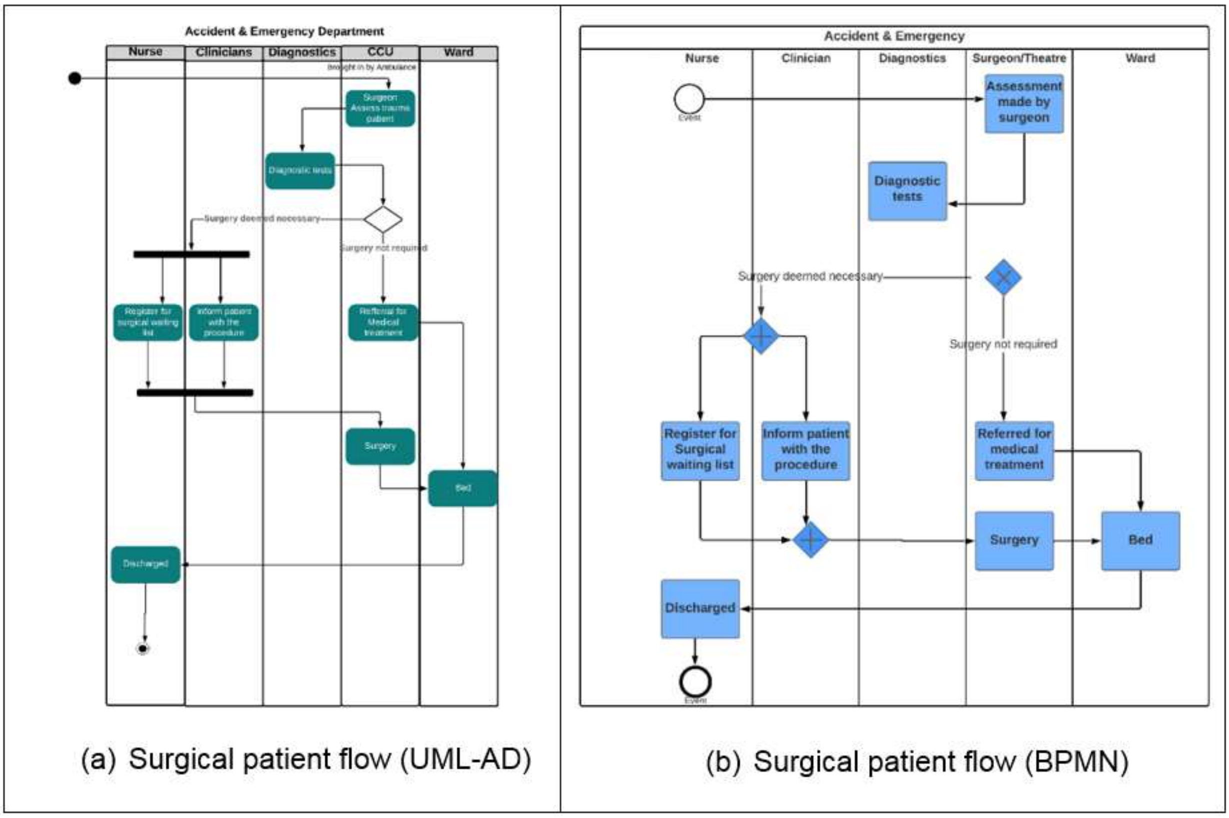


Figure 9 4 Surgical patient flow

9.1.2.1 Critique

After visual inspection of figure 9.4(a) and (b) drawn in UML AD and BPMN, I have found that the operational patient flow has some semantic incorrectness. A fork/join (UML-AD) and parallel gateway (BPMN) used to depict the relationship between two concurrent actions/tasks, i.e. Register for 'surgical waiting list' and 'inform the patient with the procedure' as shown in figure 9.5 (a) and (b) respectively. The inability of both the standards to present the exact behaviour by providing the knowledge about the temporal order, such as which action/task out of the parallel flow is going to occur in what order.

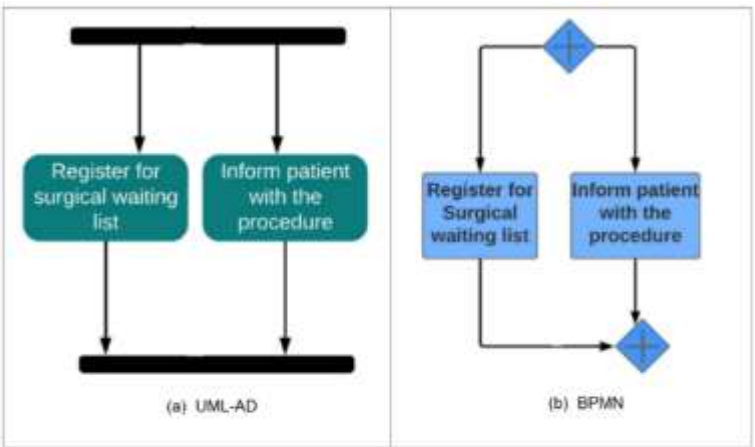


Figure 9 5 Excerpt from surgical patient flow

9.1.2.2 Enhanced Reasoning

In addition to the above findings, both the standards failed to identify the other existent relationships that could be vital for consistent and better performance. An application of the framework supported by the derived knowledge between the two actions/task, a list of PITL relationships can be identified that could occur. Using the natural language representation, I would describe the possible relations between two process instances given in table 9.2.

Process Symbol	Natural Language Description	Qualitative Relationships
X	Register patient for surgical waiting list	X = Y [sX = sY and eX = eY],
		X s Y [(sX = sY and eX < eY)]
		X d Y [sX > sY and eX < eY]
		X o Y [sX < sY, sY < eX and eX < eY],
		X f Y [sY < sX and eY = eX]
Y	Inform patient with the procedure	Y s^{-1} X [sY = sX and eY < eX],
		Y d^{-1} X [sY > sX and eY < eX]
		Y o^{-1} X [sY < sX, sX < eY and eY < eX],
		Y f^{-1} X [sX < sY and eX = eY]

Table 9 2 Derived temporal relationships a patient flow illustration II

Table 9.2 reveals that several different temporal relationships can be derived (using their start/end) to provide enhanced reasoning concerning two process instances.

The PITL inference mechanism also assists in determining the consistency of patient flow to draw the parallel flow between two process instances such as ‘X’ (representing register for surgical waiting list) and ‘Y’ (representing an informed patient with the procedure). I could draw the derived relationships using the start and endpoints of process instances X and Y and their inverse using an equivalent PG* as shown in figure 9.6 (a) and (b) respectively, to see the capability of the framework.

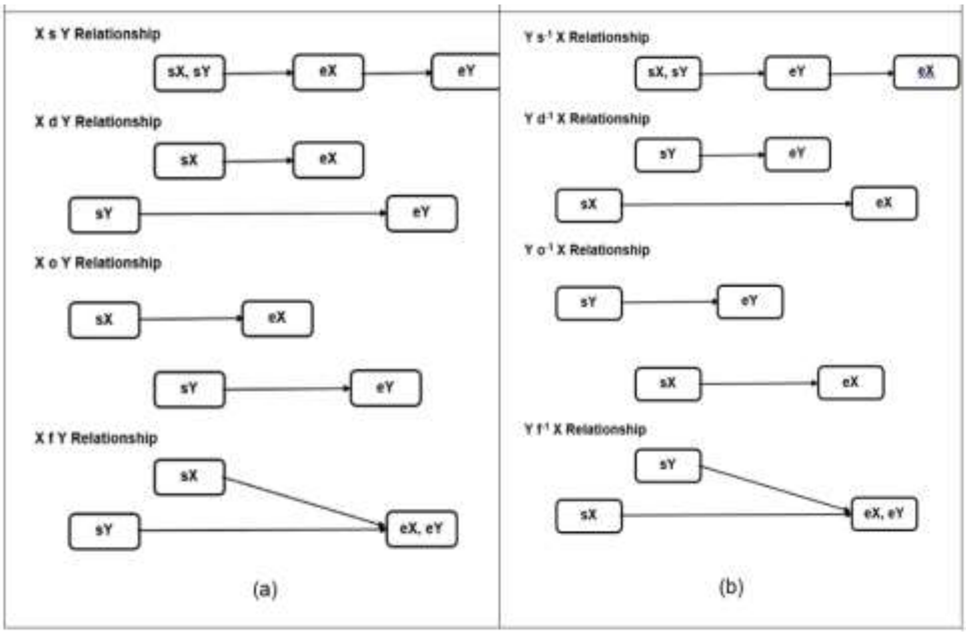


Figure 9 6 Derived relationships from the excerpt of the surgical patient flow

The above possible derived temporal relations between two atomic process instances suggests that with the help of PITL inference mechanism the framework could assist in finding a consistent path between them. Therefore, I could say that both the standards failed to provide such reasoning to establish the consistency which can be provisioned by the framework developed here. In addition, I have transformed the patient flow modelled in UML-AD and BPMN with the help of the framework by utilising in natural language representation as given in table 9.3 below.

Natural Language Representation		
Process	***Description***	***PITL***
A_1	The assessment made by the surgeon	A_1 meets A_2
A_2	Diagnostic tests carried out	A_2 meets A_3
A_3	Referral for medical treatment	A_3 meets A_7
A_4	Registration for surgery	A_2 meets A_4
A_5	Information regarding surgery	A_2 meets A_5
A_6	Surgery	A_4 and A_5 meet A_6
A_7	Moved to Ward	A_6 meets A_7
A_8	Discharged	A_7 meets A_8

Table 9 3 Natural Language representation of patient flow illustration II

In Table 9.3 expresses the process instances involved in the patient flow using PITL statements. The problem specification does not clearly state the concurrent behaviour and in real-life 'Inform patient with procedure' overlapped by the register for surgical waiting list'. Therefore the relations shown in figure 9.6 can assist us in deriving the correct relationships for consistent construction of the patient flow as shown in figure 9.7.

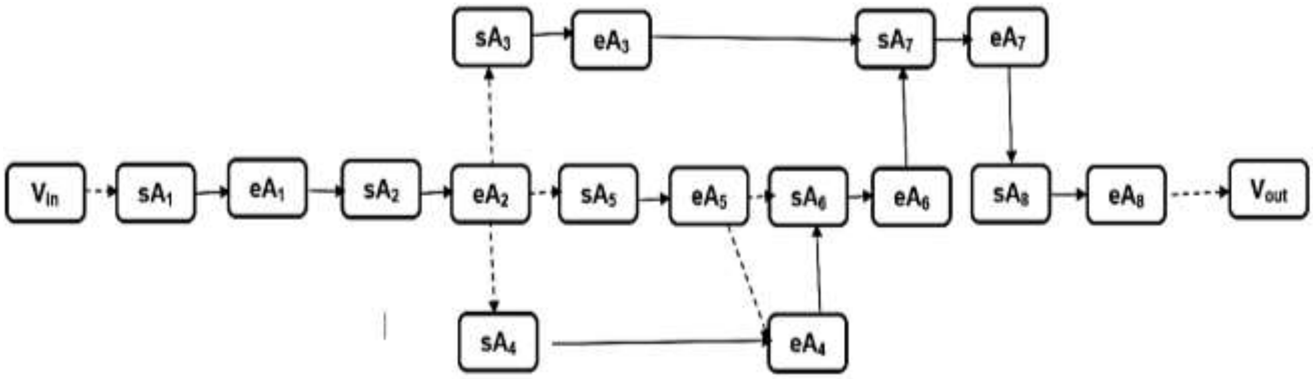


Figure 9 7 A PG* representation of the surgical patient flow

Figure 9.7 determined an overlapping relationship between process instances 'A_4' and A_5' and provided an equivalent but consistent PG*. Because it has been observed that the patient is informed before the 'registration to be put on the waitlist' process instance is finished. Hence, two derived relationships 'eA_5 precedes eA_4' and 'eA_5 precedes 'sA_6' assist in finding the consistent path. However, if other relationships are shown in figure 9.6 (a) and (b) are considered then few PG* (consistent) can be constructed depending upon absolute temporal information (if available).

However, my motive is to establish the applicability of the approach developed in this thesis to prove that it is not only able to verify and validate the constructed process model but also equip the stakeholders with enhanced reasoning to analyse them for many different reasons. The ability of the axiomatic system based on PITL to detect the inconsistencies in the process specification (where information come from different sources) would be an edge compared with modelling standards.

Now I will discuss a discharge patient flow described in Illustration III. Illustration I and II depict the journey of the trauma patients arriving at the accident and emergency department (A&E) of King's College Hospital. In illustration II a major trauma patient has undergone surgery and treatment is completed. Subsequently, the patient is ready to be discharged from the hospital. Trauma patient pathway is a complex process and therefore, modelled here in three patient flows clearly displaying the necessary steps involved in detail for an efficient model.

9.1.3 Discharge Patient Flow Illustration III

A trauma patient after treatment comes to a stage where clinical staff require further checks before releasing the patient for his/her safety. The discharge patient flow starts with a concurrent flow of an action/task to decide upon the discharge date, and in parallel, a decision is required by the nurses establishing that if the patient requires transitional care or not. Though a request is made for an assessment required followed by an evaluation of patient needs in case a decision for the transitional care provision is made. However, the action/task to decide upon discharge date also make a check on the same issue of transitional care requirements followed by informing transitional care team with a discharge date. These actions/tasks then combined to proceed with a confirmation of transitional care and join with no transitional care decision path to confirm a discharge summary. A nurse is required to run through it and after making sure the summary is correct, discharges the patient from the hospital. This scenario is modelled using UML-AD and BPMN as shown in figure 9.8 below

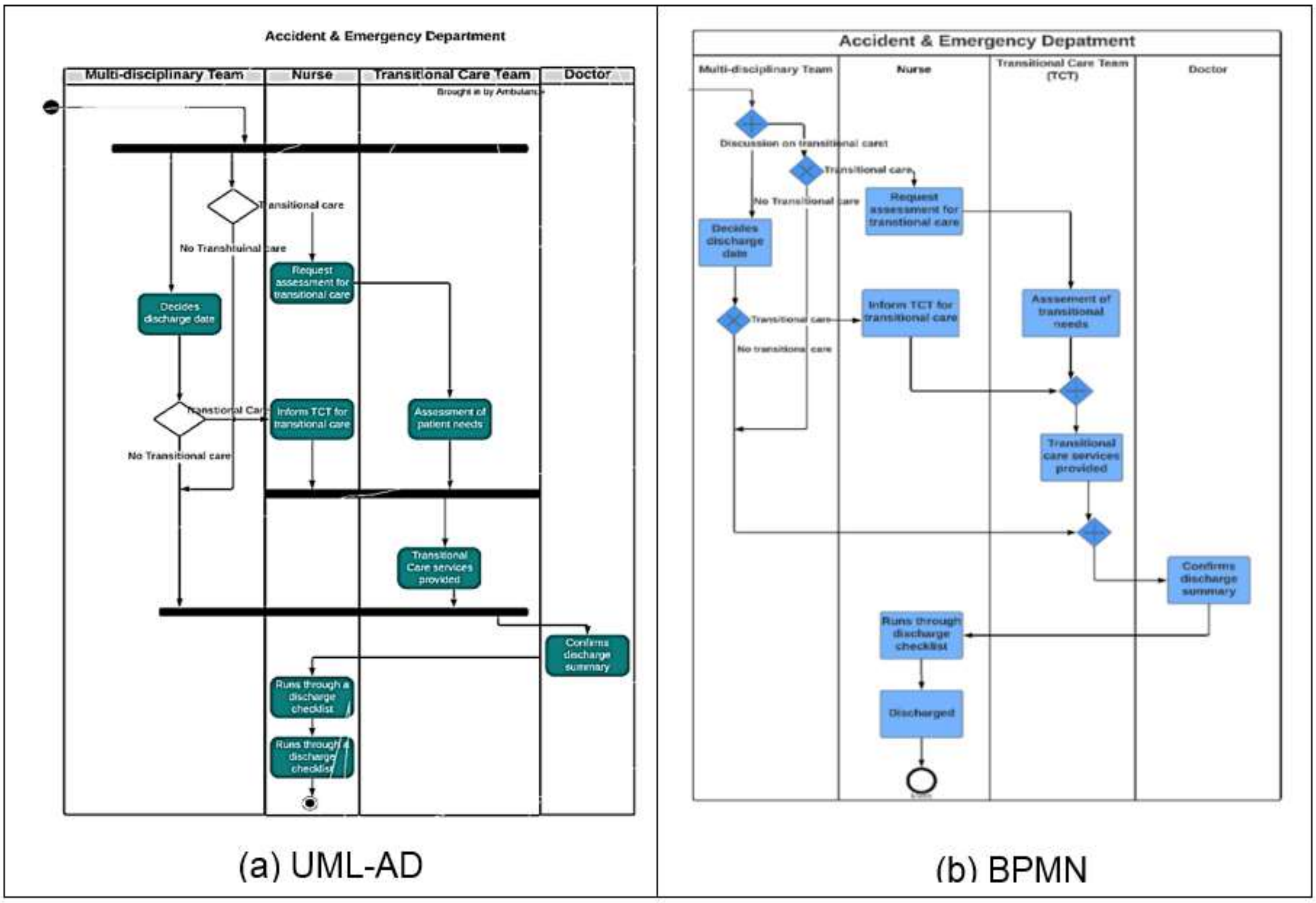


Figure 9 8 Discharge patient flow

9.1.3.1 Critique & Enhanced Reasoning

In figure 9.8 (a) and (b) shows a decision point/exclusive OR gateway placed as one of the parallel flow (fork/parallel gateway) to decide upon transitional provision. However, there is another flow of the parallel behaviour represented to depict the action/task 'decides discharge date'. Furthermore, in case no transitional care required(decision point) by the patient decided by the multi-disciplinary team occurs before the action/task 'decide discharge date' joins the decision point of the same action/task decision point (that occurs after the same action/task completed). It shows an inaccurate representation of the patient flow.

Similarly, another flow coming out from the same decision point requires the multi-disciplinary team to decide upon the transitional care would trigger a 'request assessment for transitional needs' made by the 'Nurse'. It requires a 'Transitional Care Team' to initiate an 'assessment of transitional needs' action/task. In addition, if a decision regarding transitional care provision is made after the 'decide discharge date' action/task then the 'Nurse' required to trigger the 'inform transitional care team to carry out the assessment' followed by a join to confirm the transitional care provided. After visual review of the models shown in figure 9.8 (patient flow modelled using (a) UML AD and (b) BPMN), I have found that the discharge patient flow has semantic incorrectness; inaccurate and inconsistent. To evidence this, I have transformed the individual action/tasks based on an axiomatic system using the natural language representations shown in Table 9.4 below.

Process	***Description***
A_1	Deciding the discharge date
A_2	Request for Assessment for transitional care
A_3	Assess patient needs
A_4	Informs TCT of transitional care
A_5	Transitional care service provided
A_6	Confirms discharge summary
A_7	Runs through the discharge checklist
A_8	Patient discharged

Table 9 4 Natural Language representation of discharge patient flow

The above information enables me to draw the discharge patient flow using PG* to evidence the inaccuracies of the models presented in figure 9.8 (a) and (b). An equivalent PG* shown in figure 9.9 below

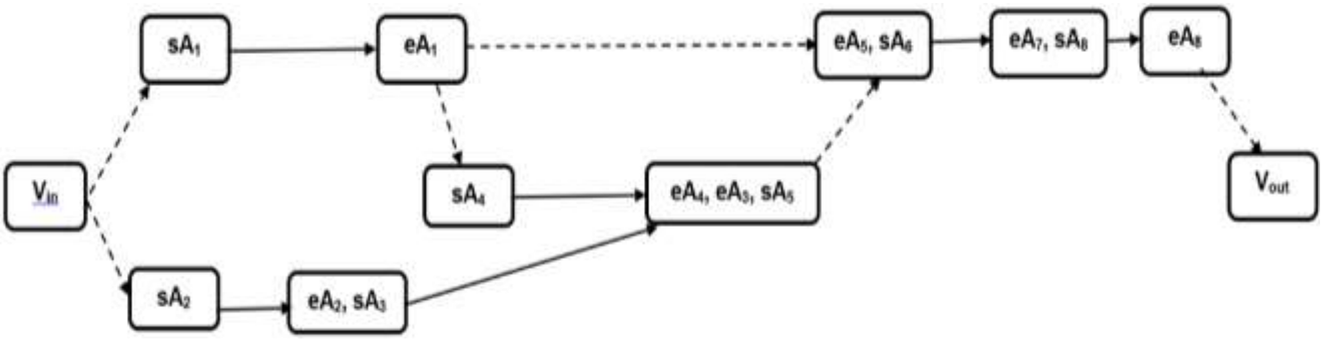


Figure 9 9 Discharge patient flow drawn in PG* (inconsistent)

Now, to draw a consistent PG* with the support of the inference mechanism, I would be able to derive qualitative relationships between the process instances' start/endpoints. With the additional knowledge acquired from the inference mechanism of axiomatic system, I am going to extend Table 9.4 and add this extra information presented in Table 9.5. It would assist me in drawing a consistent patient flow using PG*.To elaborate this, I re-draw the inconsistent PG* (figure 9.9) with derived temporal information given in Table 9.5 below.

Natural Language Representation		
Process	***Description***	***PITL***
A_1	Deciding the discharge date	A_1 meets A_6
A_2	Request for Assessment for transitional care	A_2 meets A_3
A_3	Assess patient needs	eA_4 precedes eA_3
A_4	Informs TCT of transitional care	$eA_4 < eA_1$
A_5	Transitional care service provided	eA_5 precedes eA_1
A_6	Confirms discharge summary	A_5 meets A_6
A_7	Runs through the discharge checklist	A_6 meets A_7
A_8	Patient discharged	A_7 meets A_8

Table 9 5 Derived temporal relationships of discharge patient flow

In Table 9.5, three qualitative temporal relations derived from the given scenario which UML AD and BPMN representations (presented in figure 9.8(a) and (b)) cannot capture i.e. eA_4 precedes eA_3, eA_5 precedes eA_1 and $eA_4 < eA_1$. Using PG* the above-derived knowledge can be represented in an attempt to provide a consistent patient flow shown in figure 9.10.

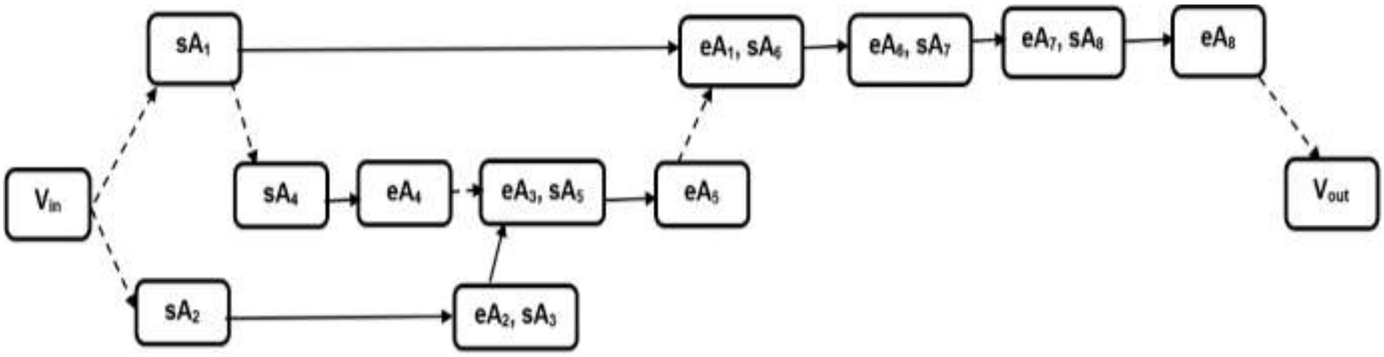


Figure 9 10 Discharge patient flow (consistent) modelled in PG*

The above investigation evidenced that both the modelling standards produced inaccurate and inconsistent representation of the patient flow. But with the support of derived temporal relations, modellers able to capture the complexities of a system that may facilitate in constructing a consistent and semantically correct model that can be verified. The comparative findings are listed below.

- The action/task 'informed transitional care team for transitional care' modelled parallel to the patient's needs assessment' action/task. The flow depicted does not provide any other information concerning their relative occurrence. However, with the assistance of the axiomatic system, I have derived the relationship between them and shown in figure 9.10, i.e. eA_4 precedes eA_3.
- The figure 9.8 (a) and (b) presented the models showing the action/task 'inform TCT for transitional services' as a decision branch (XOR) coming out from 'decides discharge date'. It means the model tells the stakeholder that 'decides discharge date' has completed. It means it shows a misleading flow to start transitional care service provision after the discharge date which should have been considered the discharge date finalised. On the contrary, action/task 'inform TCT for transitional services' occurs during the 'decides the discharge date' (derived), i.e. $eA_4 < eA_1$ represented in an equivalent PG* shown in figure 9.10 to consistently represent the discharge patient flow.
- Additionally, the action/task 'Transitional care services provided' should finish before the 'decides the discharge date' action/task, i.e. eA_5 precedes eA_1

represented in an equivalent PG* shown in figure 9.10. However, it has been modelled as a join/parallel gateway path in figure 9.8 (a) and (b) and its inaccurate.

Even though both modelling standards rely on intuitive modelling but still failed to specify a relative temporal order between actions/tasks. Therefore it confuses the stakeholder instead of providing correct behaviour. This discussion showed that the absence of the comprehensive relative temporal information hampers the correctness of the modelling standards. So far, with the aid of the framework, I have analysed the models constructed in UML-AD and BPMN for their relative temporal order and subsequently constructed a semantically correct model using PG* making provision for enhanced reasoning (demonstrated in the trauma patient pathway) for their consistency.

9.2 Scheduling Challenges

Each scheduling method addresses somewhat different issues, therefore an appropriate scheduling mechanism within the hospital environment may assist in improving resource bound activities in the emergency department of the hospital such as better patient flow through A&E while managing well the associated resources such as staff, wards and operation theatres etc [Hall, 2012]. However, the framework not only offers graphical representation for planning and representing the patient flows but has the capability to schedule the activities well concerning the time (quantitative and/or qualitative) associated with each activity in patient flow to deliver timely service. Furthermore, the utilisation of attached temporal information would help in scheduling all the required elements in a process enactment. With this approach, I could address the issues associated with scheduling such as ensuring resources are maintained while minimizing the time spent at the hospital waiting to be seen.

The findings of this research also establish that both relative and absolute temporal information (if available) crucial to organise the atomic process instances in a patient flow for better representation and improved performance. In addition, with the availability and appropriate utilisation of qualitative temporal information, a modeller not only models a consistent patient flow but also schedule the process instances involved for optimisation purposes. The scheduling of process instances achieved using PG* by applying three parametric values, i.e. earliest start (E_v), a late start (L_v), and latest start (T_v), to each vertex of a PG*. The specific time values assist in identifying critical and non-critical process instances in a patient pathway. Furthermore, to optimise project management' operational constraints adapted here and respectively termed as 'applied constraints (ACs). The applied constraints are defined in chapter 7 (AC_1- AC_4). AC_1 and AC_2 used advantageously when non-critical process instances delayed.

Moreover, scheduling mechanism provided here may assist in improved process instance execution when either the exact knowledge of special atomic process instance temporal information (stamp) or boundary values of a process instance (start/end) available. Such information would enable the stakeholders with the knowledge such as estimation of a finish time and/or identify delays to specify corresponding delay and/or `earlier completion. Furthermore, it would also facilitate an improved process specification concerning associated temporal information by reviewing the schedule and costs involved. Therefore, I would utilise the framework (scheduling mechanism) to the patient flows for their optimisation.

To see the framework in action to schedule process instances, I would rely on the data gathered from the King's College Hospital' A&E department with regards to their discharge patient flow presenting associated real-time values to perform schedule given in Table 9.6.

Processes	***D***	***E_v***	***T_v***	***Critical***	***AC_1***	***AC_2***
A_1	29	1	30	Yes	0	0
A_2	5	0	5	Yes	0	0
A_3	4	5	9	Yes	0	0
A_4	2	6	8	Yes	0	0
A_5	21	9	30	Yes	0	0
A_6	10	30	40	Yes	0	0
A_7	5	40	45	Yes	0	0
A_8	5	45	50	Yes	0	0

Table 9 6 Parametric values for discharge patient flow

The above information would help to schedule the discharge patient flow by simulating PG* to construct a consistent process (instance) model. It would represent a relativistic model (optimised patient flow) that is not possible with the current modelling standards as shown in figure 9.11

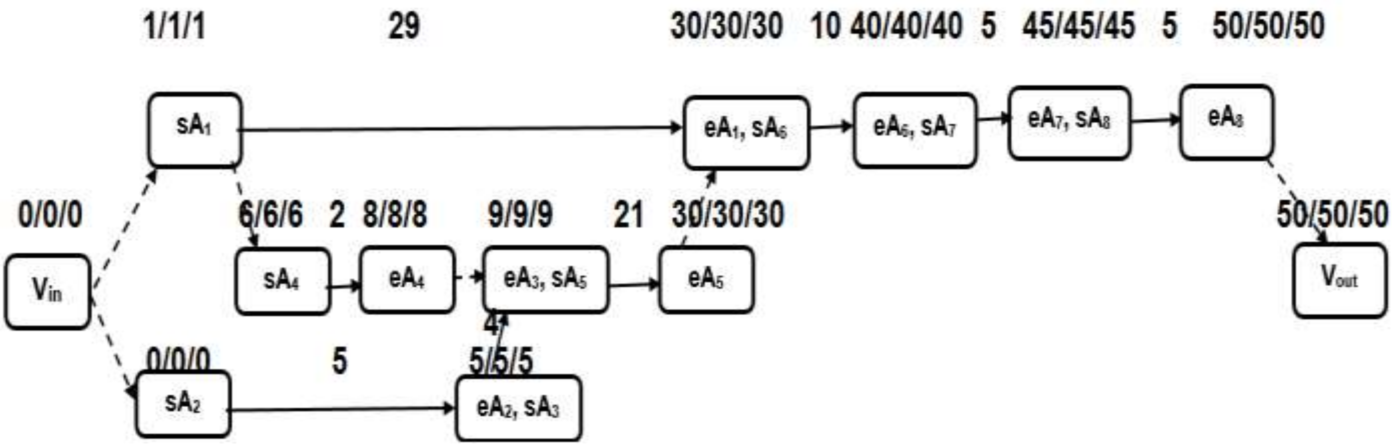


Figure 9 11 Scheduled Discharge patient flow modelled in PG*

The quantitative temporal information enables me to construct a PG* shown in figure 9.11 with precise information and drawing all the critical process instances (with their earliest start and latest completion times). Keeping in mind, the latest completion time describes the upper boundary of a process instance expressing the total duration of all the process instances. It also ensures the implicit synchronization of the parallel paths within a discharge patient flow. Furthermore, the discharge patient flow is shown in figure 9.11 also underwent the unification, branch and join folding procedure while constructing the optimised model.

9.2.1 Limitations

At King's College Hospital, during the process of data collection, I have identified that there were specific situations where some limitations applied due to non-availability of resources (in the discharge process, is its staff). Such limitations can cause constrain the flow and may delay the overall process, for example, for the process instance A_4 where nurse is required but not available straight away to carry out the 'request for transitional care assessment' process instance.

Another constraint, if applied may cause some alteration to the overall patient flow. For example, if a constraint applied to the process instance A_2 such that it cannot start earlier than '2' time units which is $sA_2 \geq 2$. Additionally, there are situations where doctors are not available a a specific time due to theirother engagements. Therefore, considering such situation where doctor availability is specified such as available at '32' time units (A_6 must start at '32' time units). The constraints applied to the two process instances (stamps) have changed the values of table 9.6 and the appended information given in Table 9.7.

Processes	***D***	***E_v***	***T_v***	***Critical***	***AC_1***	***AC_2***
A_1	29	3	32	Yes	0	0
A_2	5	2	7	Yes	0	0
A_3	4	7	11	Yes	0	0
A_4	2	6	8	Yes	0	0
A_5	21	11	32	Yes	0	0
A_6	10	32	42	Yes	0	0
A_7	5	42	47	Yes	0	0
A_8	5	47	52	Yes	0	0

Table 9 7 New values added to discharge patient flow

The constraints specified in Table 9.7 are accommodated in a reviewed PG* shown in figure 9.12.

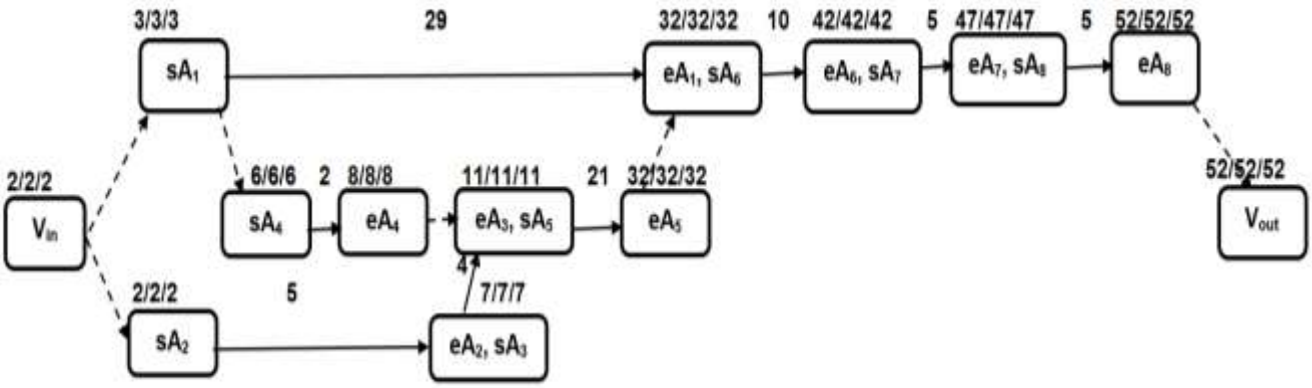


Figure 9 12 Scheduled presented with appended quantitative values

The involved process instances are critical so that the constraints applied have changed the total duration of the patient flow by re-calculating the individual process instances as shown in figure 9.12 above. Thus, I have shown the working of the framework to utilise the scheduling mechanism to optimise the discharge patient flow. In addition, I have utilised the applied constraints to show that the framework has comprehensively covered project management related issues.

To demonstrate the functioning of a sub-process instance described in the axiomatic system, I have adopted the HPG* (described in chapter 6). Notice that HPG* represents a PG* including the set 'M' (pair of relations) that is empty in case only high-level PG* is required. Therefore, it is possible to associate multiple sub-process instances to a single process instance. HPG* utilised the available completion time information serving as constraints and keeping different coordinated process instances intact.

The investigation conducted at the King's College Hospital revealed that the discharge patient flow can be broken (depending upon the patient's needs) into two atomic process instances representing a sub-process instance. They normally occur during the 'decides the discharge date' process instance to facilitate the patients with the information required for their wellbeing after discharge ensuring they understand what steps need to be taken when at home involving their medication and self-care etc. Furthermore, staff requires to pencil a discharge plan ensuring every point is documented given in Table 9.8 below

Process	***Description***
A_{11}	Discussion with the patient concerning discharge
A_{12}	Develop a discharge plan

Table 9 8 Sub-process details of the discharge patient flow

The clinical staff at the King's College Hospital required to carry out these atomic process instances with the patients with specified needs such dementia or other related diseases restricting them concerning their wellbeing or self-care. Therefore, a patient discussion is necessary along with a planned developed to decide a discharge date. To utilise such information for their respective graphical representation, process instance A_1 broken down into A_{11} and A_{12} to represent the atomic process instances. The two atomic process instances A_{11} and A_{12} constitute a sub-process (definition given in chapter 6). I have drawn the relative occurrence of the HPG* in a PG* without the quantitative temporal information in figure 9.13.

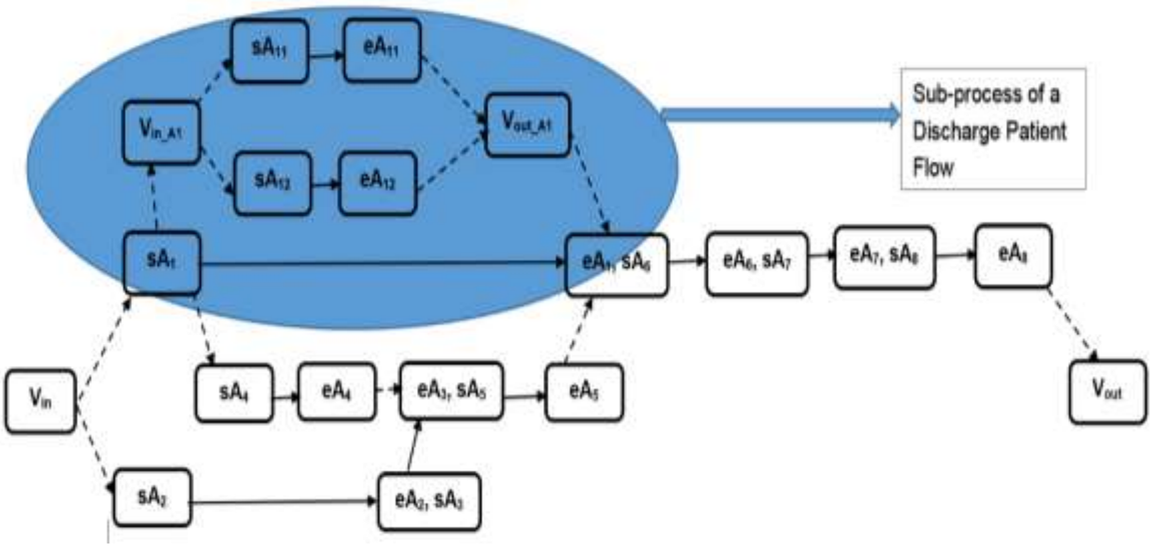


Figure 9 13 A sub-process using HPG*

As we know that PG* used for the high-level process instance representation and to accommodate additional information to represent low-level abstraction, a HPG* is used. Figure 9.13 shows the qualitative temporal information used to construct a HPG* accommodating new information attached to the process instance 'decides the discharge date' has broken down into two sub-components. Keeping in mind, the total duration (quantitative temporal information) of 'decides the discharge date' process instance must be higher or equal to its sub-process instance length. Therefore, to evidence the quantitative temporal information in operation, I consider the revised PG* presented in figure 9.12 with updated time information. Table 9.9 provides additional quantitative temporal information regarding the two atomic process instances (sub-process).

Processes	***D***	E_v	T_v
A_{11}	20	0	20
A_{12}	20	0	20

Table 9 9 Parametric values of the subprocess

The above details gathered during the investigation and meet the criteria of the PG* that is the sub-process instance length must be shorter or equal to the primary process instance. A precise HPG* representing a sub-process in a PG* constructed with absolute temporal information shown in figure 9.14.

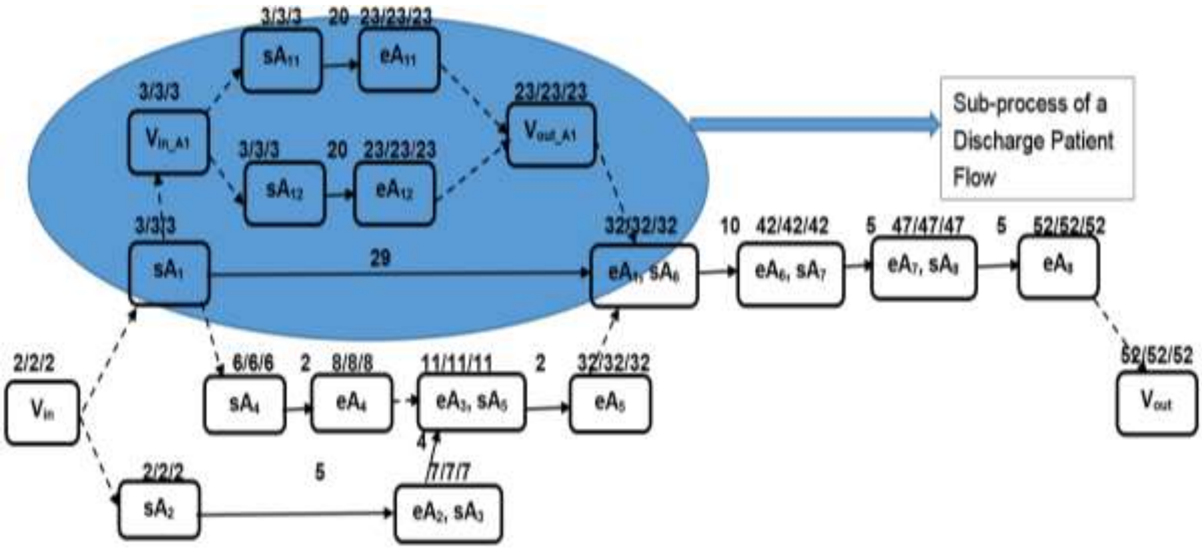


Figure 9 14 A scheduled sub-process using HPG*

Figure 9.14 presents a consistent HPG*. Therefore, the above analysis ensures the suitability of the method overcoming the major issues of the modelling standards concerning precise representation by scheduling for improved performance. Furthermore, it could also be used to analyse, verify and validate a process instance model developed in any of the modelling techniques that are specifically designed for the business process modelling. More importantly, the framework has subsumed both modelling standards (core modelling terminologies and constructs) and provided a unified platform for business process modelling domain.

9.2.2 Feedback

A series of meetings arranged with the domain experts at the King's College Hospital after each patient flows transformation carried out to show the inaccuracies and inconsistencies with their existing models developed in UML-AD and BPMN. This has established the need to identify the boundaries of the activities involved within a patient flow for their consistent representation.

Similarly, a precise transformation of the models to the method developed in this thesis not only helped them to understand the problems with the modelling standards but also witnessed the benefits provided by the framework to construct models with consistency and presented additional features of reasoning for their ease.

The feedback which I have received was very encouraging and seen their interest in utilising the method in future not only in A&E to model other functionalities but they have shown interest in its use as a pilot in other departments. Furthermore, they expressed their gratitude towards the efforts I have made in terms of providing them with the technique that addressed the modelling issues at the accident and emergency (A&E) department.

9.3 Summary

One of the most vital chapters of this study that determines the authenticity of the method developed and establishes the contribution to the knowledge in practice. The chapter included the process of the data gathered from the King's College Hospital Trust with regards to their patient pathways. I have chosen a case (trauma patient pathway) from the available data to ensure the contributions to the knowledge evidenced appropriately and to show the approach practical value. The data gathered through interviewing process by taking notes and the models constructed in UML-AD and BPMN.

The models received showed isomorphism between the two graphical modelling approaches in handling similar scenarios. Even though additional temporal information is available but failed to incorporate important information to model the patient flows. Due to this, the models built using UML-AD and BPMN not only provided insufficient information about the flow with a high probability of being incorrect. In addition, no mechanism of verification is in place to examine inaccuracy within a constructed model using both the modelling standards. Furthermore, both approaches are not formalised and have the possibility to provide different interpretation of wide variety of modelling elements available to stakeholders generally and in the NHS specifically (in the hospital).

The challenge with the trauma patient pathway was its complexity comprised of several activities and sub-activities and the timely delivery of the required care service. Due to this, the domain experts at King's College Hospital divided the pathway. I will consider only one case of the trauma patient flow which is considering the main pathway considered for every trauma patient arrival namely a) Trauma patient flow. The next flow is chosen fro the main flow to represent its branch representing a flow necessary for major trauma patients requiring surgical intervention entitled with b) Surgical patient flow. This flow is followed by the associated patient flow describing the discharge shown as c) Discharge patient flow. These separate flows are subsequently modelled utilising UML-AD and BPMN by them to express the behaviour attached.

These models were collected, analysed and evaluated using the framework for any inconsistency within the models. The framework provides a verification mechanism to analyse the built models by assimilating the problem description attached with the scenario and associated additional temporal information for their transformation into PG*. The analytical capabilities of the approach not only examine the inconsistency within the problem description but also in the built models.

PG* also provides a labelling mechanism coupled with scheduling algorithm to embed available temporal information (both qualitative and quantitative) attached with the activities involved for better planning, management and improved decision making. However, existing modelling approaches are not able to incorporate such temporal information and lack in facilitating modellers with a scheduling procedure that could hugely improve the flow and resource utilisation. In addition, the framework developed in chapter 5 and 6 ensure the verification and validation of the models that are missing in the current modelling standards.

Moreover, the application of the framework on a major trauma patient pathway modelled by domain experts at the King's College Hospital found significant inaccuracies. These findings were the result of the method developed here that provided the clear semantics for their more often used modelling artefacts supported by algorithms to inspect the modelled processes and transform them to PG* to present the consistent patient flow.

In addition to the consistent representation of the patient flow, the framework also facilitates users with enhanced reasoning based on the point interval temporal approach. That embeds the ‘what if’ situations and ensures that the uncertainty is addressed utilising the available enhanced qualitative constraints. UML-AD and BPMN lack such facility in their standard documentation which makes the approach proposed here distinct and justifies its contribution to the knowledge.

The analysis results provided to the domain experts at the King’s College Hospital. They were interested to see the inconsistencies in the individually transformed patient flows of the trauma patient pathway. Additionally, the framework planning and scheduling capabilities with the incorporation additional temporal information (both qualitative and quantitative) amazed them to see possible improvements they can make to their modelling capabilities. That could help them to optimise their schedule with regards to their time and staff resources and make better decisions. UML-AD and BPMN have no power to perform such operations and therefore found insufficiently suitable for modelling patient flows.

However, the method proposed here clearly transformed the models built in both UML-AD and BPMN into PG* and provided King’s College Hospital with noticeable improvements that could assist them in handling similar scenarios effectively by graphically representing their patient flows.

Chapter 10 Conclusion & Future Work

In this chapter, I would conclude establishing the contributions to the knowledge achieved and detailing briefly with the possibility of future work.

10.1 Conclusion

In this thesis, I have reviewed the modelling standards and examined the existing work. The investigation has revealed that business process modellers utilising the modelling standards of both the business and technical domain facing challenges to model processes precisely. It also has a huge impact on the healthcare industry, which tends to adopt these insufficient paradigms to model hospital patient flows for their correct representation. Therefore, the primary motivation of this research was to provide a framework that is general enough and has the provision of a knowledge base to address the missing gap. Which makes it applicable to any real-life domain to show the validity of the method proposed.

10.1.1. Contributions to the Knowledge

For this research, I have relied upon the constructive research method to address the real-life problems faced by the domain experts while modelling processes. This multi-disciplinary research work carried out a comparative empirical study of business process modelling (BPM) standards used by IT and business industry. Which produced a number of findings and required addressing accordingly. To start with addressing the findings based on the constructive method, a theory (scientific knowledge) relevant to the business process modelling required to develop. That addressed the practical problem by laying down a foundation in the shape of an innovative practical solution contributing to the existing knowledge and filling the gap.

The comprehensive review of the business process modelling techniques (chapter 3) identified the issues faced by them. The problem associated with these modelling standards stemming from their documentation that has noted a wide variety of modelling constructs with intuitive semantics. Due to the availability of a large number of modelling constructs and out of the many are unused make them construct redundant approaches and yet they are accepted as industry standards, which is a big question for industry to answer.

As stated earlier, both techniques are overwhelmed with unnecessary modelling constructs and a huge amount of them being unused. This milestone has been achieved by the identification of the most often used modelling elements in chapter 4 necessary to construct a typical business process. In addition to the problem stated earlier faced by the industry include the fact that the one used (and the rest of them) have no clear semantics deterring the designers and modellers to built an explicit process model. Therefore, with the help of the initial findings I have made efforts to narrow down the business process modelling standards' modelling constructs and terminologies (artefacts) based on their utilisation. Similarly, it is vital to equip them with the precisely described to construct a process model depicting the consistent behaviour of an operation within an enterprise.

Ultimately, based on the constructive method, I have laid the down a foundation to develop a scientific knowledge base suitable for the business process modelling requiring precise semantics. With this approach, the analysts may use and interpret the exact enumeration explicitly due to their explicit structure and avoid any confusion which exists in the modelling artefacts used to construct models by the modelling standards.

In order to provide clear semantics for the most often used modelling artefacts, I required to align them to a well-established logic (such as temporal logic) for provisioning clear semantics for the knowledge base identified to be used in real-life. But, there are a few classes of the temporal logic exist in the literature, i.e., point interval temporal logic, interval temporal logic, point and interval logic etc. Therefore, a review has provided of these different classes of temporal logic (chapter 5) for their real-life application.

The reason for identifying most often used modelling artefacts along with a specific choice of PITL ensured a smooth alignment between them. The review has revealed that not all temporal theories are applicable to all real-life situations. Therefore, I have to rely on one class of the temporal logic that could assist in constructing a correct process model, Thus, I have chosen point and interval logic (PITL) that provided distinct temporal objects with explicit structure. That could help in describing (explicitly) the components of the knowledge base.

As part of the contribution to the knowledge, I have extended the point interval temporal logic proposed by [Zaidi, 1999]. Because the available temporal objects of the existing PITL were not enough to suitably align with the most often used modelling artefacts of the commercial tools discussed in this thesis. Therefore, to develop a state of the art framework (phase I), I have extended the PITL (contribution to the knowledge) and defined its lexicons for their precise structure.

In addition, [Zaidi, 1999[has provided a formalism for interval-interval, point-point and point-interval but another set of temporal relations and formalisms was missing. Therefore, I have added interval-point relations and formalism to extend the PITL of Zaidi (contribution to the knowledge). By providing interval-point formalism and extended relationship data set, it increases the possibility of finding a solution (consistent relation) within a process model. Furthermore, Zaidi has used the point and interval as temporal objects and neglecting the importance of moment standalone like an unbreakable interval. However, I have chosen 'moment' as a temporal object ad precisely defined in this thesis. Therefore, the knowledge base provides the necessary modelling artefacts suitable for their real-life application. Their practical value can be assessed when aligned with commercial techniques.

The above represented the methodical approach used to develop the theory utilising PITL supported with a knowledge base for modelling a correct business process. That way the foundation laid out for the development of the precise knowledge base further assisted in presenting the precise description for the identified most often used of both the modelling standards. In this way, the first hurdle removed by identifying frequently used modelling artefacts by the modelling standards in modelling a basic business process and clear semantics through the use of a reliable temporal class.

Therefore, with the aid of the temporal objects (identified in chapter 5), I have aligned the most often used a (necessary and sufficient) set to model a business process. The phase I of the study also laid a path towards the unification of both standards due to their subtlety in their representation. This research gap was unfilled for more than a decade initially identified by S. White in [White, 2004]. Hence, a set of a generic set of modelling artefacts required that can be used in real-life and suitable for both the modelling standards presented in phase II of the framework development.

Phase II adopted the approach of model theory for the development of the framework providing explicit ontology for the modelling components of the knowledge base developed in this study. To proceed with the required enumeration, I have to provide the terminology that could be easily understood and depict the correct information not only to modellers but also stakeholders can use them for consistent interpretation.

Moreover, problems associated with the standards include activities and processes vague association with the temporal object based on interval logic and point used additionally to specify start and end associated with a business process. Hence, it is vital to distinguish these terms modelling a business process explicitly.

This effort constitutes the second part of the framework developed providing the exact and generic enumeration bearing the logical meaning (consistent), i.e. ontology. To be explicit in their meaning, I have used first-order logic to define them for their real-life alignment with modelling techniques. To complete the framework, I have devised an axiomatic system to provide unified terminologies. That can be used by any modelling techniques effectively employing the exact meaning of them. The generic set of the modelling artefacts introduced in this thesis are:

- Atomic process (associating with unbreakable interval, i.e., moment) referring to action/task or any terminology which describes the same meaning.
- Special Atomic process (associating with an unbreakable point, i.e., zero duration) referring to an event or any terminology which describes the same meaning.
- Business Process and Sub- Process (associating with interval, i.e., breakable) referring to business process and sub-process, or any terminology which describes the same meaning.

These terminologies further formally defined to align them with the modelling standards most commonly used artefacts (who are missing the formal semantics). Furthermore, the review of the modelling standards also revealed that they have no support mechanism provided for inspecting the constructed models for their correctness. Thus, the models constructed cannot be examined with any inbuilt verification mechanism to establish their consistency. Thus, the axiomatic system supported by the inference mechanism embedded within the extended PITL and developed a verification and validation mechanism to construct process models (depicting high and low-level abstraction) with consistency and therefore, overcoming such problem.

In this thesis, I have chosen an artificial intelligence-based resolution theorem to provide a mechanism for verifying the axiomatic system (abstract process model). The verification of the axiomatic system should provide the correct representation of a system’s behaviour. Therefore, it is necessary that the axiomatic system proposed is sound and complete. So, I have defined the structural properties (of abstract process) and used theorem-proving techniques (based on resolution theorem) to establish that the axiomatic system is correct (sound and complete).

To make sure the axiomatic system has its real-life application (validation), based on model-theoretic approach providing there exists a unique instance of the complex process of the abstract process. Furthermore, with such approach, I would be able to establish that each abstract modelling artefact defined in the axiomatic system has its real-life instance.

That satisfy the constraints (temporal) between each artefact to construct a precise business process model. I have introduced a translation function ensuring that there is a clear translation provided for such purpose.

To summarise the above contributions to the knowledge, the framework developed is innovative and provide state of the art method to resolve the outstanding issues (such as exact enumeration, no formal semantics, non-availability of verification and validation mechanism) of the modelling standards. The framework developed also serves as the grounding for the domain of the business process modelling. In addition, the approach improved the conception, utilisation and operation viability to aid designers in constructing a precise model to express the systems' correct behaviour. Until now, I have answered the research questions 1 and 2 (specified in chapter 1).

To answer the question 3a and 3b described in chapter 1, I relied upon a formal but graphical modelling tool, i.e. point graph (PG), presented by Zaidi in [Zaidi 1999] and easy to use. I have extended it here and known as PG* (contribution to the knowledge). The formal semantics provided in the framework can easily blend in with the PG* to simulate a consistent business process. Furthermore, it has the power to evaluate business process models constructed using other modelling techniques ascertaining that all errors reported for subsequent elimination. To aid further, PG* equipped with algorithms supporting process control flow. Therefore, the framework developed with additional features can facilitate enterprises in designing and modelling the correct process. In addition, it would analyse and evaluate the business process models constructed using UML-AD and BPMN to report any errors. Which can be corrected using the framework and unify the modelling standards.

Similarly, as part of the contributions to the knowledge, a transformation (chapter 7) of the most often used modelling standards, i.e., UML AD and BPMN, to the framework is provided. The transformation performed provided the unification of the frequently used modelling artefacts of the standards into the method developed in this thesis. The unification of these modelling artefacts achieved by individually mapping them to the modelling artefacts of the framework ensuring a smooth mapping carrying a great value for its use in the real-life for their practical application.

So far with the adoption of the constructive method approach, I have developed a knowledge base (theory) that provided distinct ontology bearing concise structure supported by mathematics theorems to prove its correctness. That has been validated with the assistance of the model-theoretic approach. Now the next and final step required to show theory in action)application).

To show the state of the art framework developed here in action helping in achieving the third criterion of this thesis by its application to the real-life domain. Therefore, I have chosen healthcare domain for the application of this framework. The reason to choose the healthcare domain in general and hospital especially is due its challenging nature. Because. Hospital patient flows are tedious in structure and complex in representation and therefore yet no specific modelling method was developed to address the issues of healthcare.

I had chosen King’s College Hospital accident and emergency department to model its patient flows to improve performance via better time management. The domain experts used existing modelling standards to model their patient flows. It became evident from the existing models developed utilising the standards by the domain experts at the King’s College Hospital Trust that these techniques neither provided the consistent representation of patent flows nor improving the patient waiting time at the hospital for optimisation purposes (regarding patients’ care). The problems identified from the analysis using the framework for any inconsistency in the data collected (including information and constructed models utilising UML-AD and BPMN) from the King's College Hospital Trust.

Three different patient flows from trauma patient pathways selected and subsequently transformed to evaluate any shortcomings from the developed models collected from the King's College. The framework has the capability of incorporating extended qualitative temporal constraints and quantitative temporal information to analyse the constructed models of trauma patient pathway, which resulted in reporting errors. The errors included inconsistency in their representation and also time delays that have been overcome by the method developed here. With the application of the framework, constructive method completed and showed that it has the applicability in both the domains of computer science and healthcare.

Moreover, the approach proposed here in this thesis has the ability to plan and schedule processes with the help of algorithms using extended qualitative, and quantitative temporal information incorporated by the inference mechanism of PITL. I have also established that this method has the ability to manage the healthcare operations (including the King’s College Hospital) effectively its resource (time) utilisation through process scheduling to optimise the patient flows improving their overall performance and patient care.

10.2 Future Work

In this thesis, I have studied the graphical modelling standards and provided knowledge base comprised of the key artefacts formally defined to represent the typical business processes and patient flows. In addition, the knowledge base developed is general enough to be utilised for transfer learning in different healthcare settings.

I have contacted the Moorfields Eye Hospital Retina Imaging department and provided the opportunity to provide consultancy to investigate one of their clinics to investigate the patient flows involved. I have chosen Moorfields Eye Hospital Retina Imaging because its one of the UK's best eye hospital and has a large number of patients that need managing against time and other resources for better performance of the department. It would be a great opportunity to review the framework and its implementation in another healthcare setting. I have met with clinical staff and proposed a scheme of work (waiting to hear from them).

Moreover, the framework provided could also integrate other resources within healthcare settings such as human and machines to improve their coherent performance. If Moorfields Eye Hospital agrees to utilise the framework then it would solidify the method developed. Similarly the wealth of data can be collected that can be used to analyse their shortcomings (if any) for improved delivery of care services to its patients.

Moreover, this research also presents a strong case to incorporate approaches such as process mining and machine learning to uncover models from event logs and model deviations for performance analysis based on their time of occurrences. Because the process unfairness (also known as representational bias) can be a vital element of process mining to discover event logs based on time.

References

1. Adlassnig, K. P. (2009, August). Task analysis and application services for cross-organizational scheduling and citizen eBooking. In *Medical Informatics in a United and Healthy Europe: Proceedings of MIE 2009, the XXII International Congress of the European Federation for Medical Informatics*(Vol. 150, p. 332). IOS Press.
2. Aguilar-Saven, R. S. (2004). Business process modelling: Review and framework. *International Journal of production economics*, *90*(2), 129-149.
3. Allen, J. (1983). Maintaining Knowledge about Temporal Intervals, *Communication of ACM*, 26, 832-843.
4. Allen, J. (1984). Towards a General Theory of Action and Time, Artificial Intelligence, 23, 123-154.
5. Allen, J. and Hayes, J. (1989). Moments and Points in an Interval-based Temporal based Logic, *Computational Intelligence*, 5(4), 225-238.
6. Allweyer, T. (2016). *BPMN 2.0: introduction to the standard for business process modeling*. BoD–Books on Demand.
7. Antonacci, G., Calabrese, A., D'Ambrogio, A., Giglio, A., Intrigila, B., & Ghiron, N. L. (2016, June). A BPMN-based automated approach for the analysis of healthcare processes. In *2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)* (pp. 124-129). IEEE.
8. Barbagallo, S., Corradi, L., de Goyet, J. D. V., Iannucci, M., Porro, I., Rosso, N., ... & Testi, A. (2015). Optimization and planning of operating theatre activities: an original definition of pathways and process modeling. *BMC medical informatics and decision making*, *15*(1), 38.
9. Bassiouni, M. Mukherjee, A. and Llelllyn, M. J. (1994). Design and Implementation of Extended Boolean and Comparison Operators for Time-Oriented Query Languages. The Computer Journal, 37(7), 576-587.
10. Bell, A. E. (2004). Death by UML fever: self-diagnosis and early treatment are crucial in the fight against UML fever", ACM Queue, Vol. 2(1), pp. 72-80.
11. Berthomieu, B., & Diaz, M. (1991). Modeling and verification of time dependent systems using time Petri Nets. *IEEE transactions on software engineering*, *17*(3), 259-273.

12. Berwick, D., (1996), "A Primer on Leading the Improvement of Systems," British Medical Journal, 312(7031): pp.619-622.
13. Bettini, C, Sean, X. W. and Sushil J. (2002) Solving multi-granularity temporal constraint networks. *Artificial Intelligence* 140.1, 107-152.
14. Bhattacharjee, P., & Ray, P. K. (2014). Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: A review and reflections. *Computers & Industrial Engineering*, *78*, 299-312.
15. Blyth, A. (1995). Modelling the business process to derive organisational requirements for information technology. *ACM SIGOIS Bulletin*, *16*(1), 25-33.
16. Bocciarelli, P., D'ambrogio, A., Giglio, A., Paglia, E., & Gianni, D. (2014, June). A transformation approach to enact the design-time simulation of BPMN models. In *2014 IEEE 23rd International WETICE Conference* (pp. 199-204). IEEE.
17. Börger, E. (2012). Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Software & Systems Modeling*, *11*(3), 305-318.
18. Börger, E., & Sörensen, O. (2011). BPMN core modeling concepts: inheritance-based execution semantics. In *Handbook of Conceptual Modeling* (pp. 287-332). Springer, Berlin, Heidelberg.
19. Börger, E. and Thalheim, B. (2008). A method for verifiable and validatable business process modeling. In Advances in Software Engineering, Vol. 5316 of LNCS, pp 59-115. Springer-Verlag.
20. Bruce, B. C. (1972) A Model for Temporal References and Its Application in a Question AnsIring Program. Artificial Intelligence, 3, 1-25.
21. Camann, M. A. (2001). Outcomes of care: the use of conceptual models to" see the forest and the trees" in planning outcomes studies. *Topics in health information management*, *22*(2), 10-14
22. Cheikhrouhou, S., Kallel, S., Guermouche, N., & Jmaiel, M. (2015). The temporal perspective in business process modeling: a survey and research challenges. *Service Oriented Computing and Applications*, *9*(1), 75-85.
23. Chishti, I., Ma, J., & Knight, B. (2014). Ontology mapping of business process modeling based on formal temporal logic. *International Journal of Advanced Computer Science and Applications*, *5*(7), 95-104.

24. Chishti, I. (2014, November). A grounding of business process modeling based on temporal logic. In *International Conference on Information Society (i-Society 2014)* (pp. 266-273). IEEE.
25. Chishti, I. (2014). Towards a general framework for business process modeling. *Infonomics Society*, *5*(3), 443-453.
26. Chishti, I., Basukoski, A., & Chaussalet, T. J. (2017). Modeling and optimizing patient flows. In *8th Annual International Conference on ICT: Big Data, Cloud & Security*. Global Science & Technology Forum.
27. Chishti, I., Basukoski, A. A., & Chaussalet, T. (2018). Modeling Patient Flows: A Temporal Logic Approach. *GSTF Journal on Computing (JoC)*, *6*(1).
28. Chishti, I., Basukoski, A., Chaussalet, T., & Beeknoo, N. (2018, November). Transformation of UML Activity Diagram for Enhanced Reasoning. In *Proceedings of the Future Technologies Conference* (pp. 466-482). Springer, Cham.
29. Clarkson, P. J., Buckle, P., Coleman, R., Stubbs, D., Ward, J., Jarrett, J., ... & Bound, J. (2004). Design for patient safety: a review of the effectiveness of design in the UK health service. *Journal of Engineering Design*, *15*(2), 123-140.
30. Combi, C., & Gambini, M. (2009, November). Flaws in the flow: The weakness of unstructured business process modeling languages dealing with data. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 42-59). Springer, Berlin, Heidelberg.
31. Combi, C., Gozzi, M., Ju´arez, J. M., Oliboni, B., and Pozzi, G. (2007). Conceptual Modeling of Temporal Clinical Workflows. In Proceedings of the 14th International Symposium on Temporal Representation and Reasoning (TIME), pages 70–81. IEEE Computer Society.
32. Combi, C., Gozzi, M., Posenato, R., & Pozzi, G. (2012). Conceptual modeling of flexible temporal workflows. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, *7*(2), 19.
33. Cornwell, P. (1999). ‘UML Toolkit’ by Hans-Erik Eriksson and Magnus Penker, Wiley, 1997. ISBN: 0471191612,£ 39.95/$32.08. *Software: Practice and Experience*, *29*(2), 195-195.
34. Criswell, M., Hasan, I., Kopach, R., Lambert, S., Lawley, M., McWilliams, D., ... & Varadarajan, N. (2007, April). Emergency Department diverts avoidance using Petri nets. In *2007 IEEE International Conference on System of Systems Engineering* (pp. 1-6). IEEE.

35. Curry, J., & McGregor, C. (2005). The current state of patient journey redesign in Australia. *HIC 2005 and HINZ 2005: Proceedings*, 323
36. Curry, J. M., McGregor, C., & Tracy, S. (2005, June). Deficiencies of Process Reengineering in its Support for Healthcare Redesign. In *2nd CSTE Innovation Conference*(pp. 7-8)].
37. Curtis, B., Kellner, M. I., & Over, J. (1992). Process modeling. *Communications of the ACM*, *35*(9), 75-90.
38. Davenport, T. H., (1993). Process Innovation: Reengineering Work through Information Technology. Harvard Business School Press, Boston, MA, USA.
39. Davenport, T. H., & Stoddard, D. B. (1994). Reengineering: business change of mythic proportions?. *MIS quarterly*, 121-127.
40. David, R., & Alla, H. (1994). Petri Nets for modeling of dynamic systems: A survey. *Automatica*, *30*(2), 175-202.
41. Dehnert, J., & Van Der Aalst, W. M. (2004). Bridging the gap between business models and workflow specifications. *International Journal of Cooperative Information Systems*, *13*(03), 289-332.
42. Dijkman, R. M., Dumas, M., & Ouyang, C. (2007). Formal semantics and analysis of BPMN process models using Petri Nets. *Queensland University of Technology, Tech. Rep*, 1-30.
43. Dijkman, R. M., Dumas, M., & Ouyang, C. (2008). Semantics and analysis of business process models in BPMN. *Information and Software technology*, *50*(12), 1281-1294.
44. Drakengren, T., & Jonsson, P. (1997). Eight maximal tractable subclasses of Allen's algebra with metric time. *Journal of Artificial Intelligence Research*, *7*, 25-45.
45. Dumas, M., Großkopf, A., Hettel, T., & Wynn, M. (2007, November). Semantics of standard process models with OR-joins. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 41-58). Springer, Berlin, Heidelberg.
46. Eder, J, Wolfgang G, and Euthimios P. (2000). Temporal modeling of workflows with conditional execution paths. *Database and Expert Systems Applications*. Springer Berlin Heidelberg.
47. Edwards, N. (2005). Can quality improvement be used to change the wider healthcare system?.

48. Eriksson, H. E., & Penker, M. (2000). Business modeling with UML. *New York*, 1-12.
49. Eshuis, R. (2002). Semantics and Verification of UML Activity Diagrams for Workflow Modelling. PhD thesis, University of Twente.
50. Eshuis, R. and Dehnert, J. (2003). Reactive Petri Nets for workflow modelling. In W. M. P. van der Aalst and E. Best, (eds.) Application and Theory of Petri Nets, vol. 2679 of Lecture Notes in Computer Science, pp. 295–314. Springer-Verlag, Berlin, Berlin.
51. Eshuis, R. and Wieringa, R. (2002). Comparing Petri Nets and activity diagram variants for workflow modelling – a quest for reactive Petri Nets. In H. Ehrig, W. Reisig, and G. Rozenberg, (eds.) Petri Net Technologies for Communication Based Systems, Lecture Notes in Computer Science Springer-Verlag, Berlin.
52. Ferstl, O. K., & Sinz, E. J. (1994). *From business process modeling to the specification of distributed business application systems: An object-oriented approach*. Otto-Friedrich-Univ.
53. Findlay, J. N. (1941). Time: A treatment of some puzzles. *The Australasian Journal of Psychology and Philosophy*, *19*(3), 216-235.
54. Frappier, M., & Habrias, H. (Eds.). (2012). *Software specification methods: an overview using a case study*. Springer Science & Business Media.
55. Freksa, C. (1992). Temporal reasoning based on semi-intervals. Artificial Intelligence, 54, 199-227.
56. Galton, A. (1990). Critical Examination of Allen's Theory of Action and Time, Artificial Intelligence, 42, 159-188.
57. Gao, Y. (2006). BPMN-BPEL transformation and round trip engineering. *URL: http://www. eclarus. com/pdf/BPMN BPEL Mapping. pdf*.
58. Gerevini, A., & Schubert, L. (1995). Efficient algorithms for qualitative reasoning about time. *Artificial intelligence*, *74*(2), 207-248.
59. Glykas, M., & Valiris, G. (1999). Formal methods in object oriented business modelling. *Journal of Systems and Software*, *48*(1), 27-41.
60. Goossen, W. T., Ozbolt, J. G., Coenen, A., Park, H. A., Mead, C., Ehnfors, M., & Marin, H. F. (2004). Development of a provisional domain model for the nursing process for use within the Health Level 7 reference information model. *Journal of the American medical informatics association*, *11*(3), 186-194.

61. Großkopf, A. (2007). xBPMN. Formal control flow specification of a BPMN based process execution language. *Master's thesis, Hasso-Plattner-Institut, Potsdam, Germany*.
62. Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. Enschede, The Netherlands: Telematica Instituut.
63. Günal, M. M., & Pidd, M. (2010). Discrete event simulation for performance modelling in health care: a review of the literature. *Journal of Simulation*, *4*(1), 42-51.
64. Hall, R. W. (2012). *Handbook of healthcare system scheduling*. New York: Springer Science+ Business Media, LLC.
65. Hammer, M, and Champy. J. (1993). Reengineering the corporation: A manifesto for business revolution." *Business horizons* 36.5, 90-91.
66. Hansen, G.A. (1994) Automating Business Process Reengineering: Breaking the TQM Barrier, Prentice-Hall, Englewood Cliffs, New Jersey.
67. Haraden, C., & Resar, R. (2004). Patient flow in hospitals: understanding and controlling it better. *Frontiers of health services management*, *20*(4), 3.
68. Havey, M. (2005). *Essential business process modelling,* O'Reilly Media, Inc.
69. Hofacker, I., & Vetschera, R. (2001). Algorithmical approaches to business process design. *Computers & Operations Research*, *28*(13), 1253-1275.
70. Hoffner, Y., Lwdwig, H., Gulcu, C., & Grefen, P. (2000). An architecture for cross-organisational business processes. In *Proceedings Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems. WECWIS 2000* (pp. 2-11). IEEE.
71. Horn, S. D. (2001). Quality, clinical practice improvement, and the episode of care. *Managed Care Quarterly*, *9*(3), 10-24
72. Hughes, M., Carson, E. R., Makhlouf, M., Morgan, C. J., & Summers, R. (1998, November). A Petri net based model of patient-flows in a progressive patient-care system. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)* (Vol. 6, pp. 3048-3051). IEEE.
73. Jablonski, S., & Bussler, C. (1996). *Workflow management: modeling concepts, architecture and implementation* (Vol. 392). London: International Thomson Computer Press.

74. Jarrar, Y. F., Abdullah AM, and Mohamed Z. (2000). ERP implementation critical success factors the role and impact of business process management. *Management of Innovation and Technology, ICMIT 2000. Proceedings of the 2000 IEEE International Conference on*. Vol. 1. IEEE.
75. Jensen, K. (1997, April). A brief introduction to coloured Petri Nets. In *International Workshop on Tools and Algorithms for the Construction and Analysis of Systems* (pp. 203-208). Springer, Berlin, Heidelberg.
76. Jensen, K., Haraden, C., Mayer, T. and Welch, S., (2006), Leadership for Smooth Patient Flow, Health Administration Press, Chicago, IL.
77. Jun, G. T., Ward, J., Morris, Z., & Clarkson, J. (2009). Health care process modelling: which method when?. *International Journal for Quality in Health Care*, *21*(3), 214-224
78. Jungclaus, R., Saake, G., Hartmann, T. and Sernadas, C. (1996) TROLL- A Language for Object-Oriented Specification of Information Systems. ACM Transactions on Information Systems, 14(2), 175-211
79. Kettinger, W. J., Teng, J. T., & Guha, S. (1997). Business process change: a study of methodologies, techniques, and tools. *MIS quarterly*, 55-80.
80. Korherr, B. (2008). *Business Process Modelling: Languages, Goals, and Variabilities*. Saarbrücken: VDM Publishing.
81. Koskela, M. and Haajanen, J. (2007). Business process modeling and execution: tools and technologies report for the SOAMeS project", VTT Research Notes No. 2407, VTT Technical Research Centre of Finland, Espoo.
82. Koubarakis, M., & Plexousakis, D. (2002). A formal framework for business process modelling and design. *Information systems*, *27*(5), 299-319.
83. Kretschmer, P. (2014). JWT metamodel compared to BPMN metamodel.
84. Kueng, P., & Kawalek, P. (1997). Goal-based business process models: creation and evaluation. *Business Process Management Journal*, *3*(1), 17-38.
85. Ladkin, P. (1992). Effective solutions of qualitative intervals constraint problems, *Artificial Intelligence*, 52, 105-124.
86. Lano, K. (Ed.). (2009). *UML 2 semantics and applications*. John Wiley & Sons.
87. Lewis, C. (1993). A source of competitive advantage?. *MANAGEMENT ACCOUNTING-LONDON-*, *71*, 44-44.
88. Leymann, F. and Altenhuber, W. (1994). Managing Business Processes as an Information Resource, IBM Systems Journal, Vol. 33, No. 2, pp. 326–348.

89. Li, H., Yang, Y., & Chen, T. Y. (2004). Resource constraints analysis of workflow specifications. *Journal of Systems and Software*, *73*(2), 271-285.
90. Lindsay, A, Denise D, and Ken L. (2003). Business processes—attempts to find a definition. *Information and software technology* 45.15, 1015-1019.
91. Lodhi, A., Küppen, V., & Saake, G. (2011). An extension of bpmn meta-model for evaluation of business processes. *Scientific Journal of Riga Technical University. Computer Sciences*, *43*(1), 27-34.
92. Luftman, J., Papp, R., & Brier, T. (1999). Enablers and inhibitors of business-IT alignment. *Communications of the Association for Information Systems*, *1*(1), 11.
93. Lyalin, D., & Williams, W. (2005). Modeling cancer registration processes with an enhanced activity diagram. *Methods of information in medicine*, *44*(01), 11-13.
94. Ma, J. (2007). Ontological considerations of time, meta-predicates and temporal propositions. *Applied Ontology* 2.1, 37-66.
95. Ma, J, and Knight, B. (1994). A General Temporal Theory, t*he Computer Journal*, Vol. 37(2), 1994, pp 114-123.
96. Ma, J., Knight, B. (2003). Representing the Dividing Instant, the Computer Journal, 46(2), 213-222.
97. Mahulea, C., Mahulea, L., Soriano, J. M. G., & Colom, J. M. (2018). Modular Petri net modeling of healthcare systems. *Flexible Services and Manufacturing Journal*, *30*(1-2), 329-357.
98. Maiocchi, R., Pernici, B. and Barbic, F. (1992) Automatic Deduction of Temporal Information. ACM Transactions of Database Systems, 17(4), 647-688.
99. Manna, Z and Pnueli, A. (1995) Temporal verification of Reactive Systems: Safety. Sringer-Verlag New York, Inc.
100. Mans, R. S., Schonenberg, M. H., Song, M., van der Aalst, W. M., & Bakker, P. J. (2008, January). Application of process mining in healthcare–a case study in a dutch hospital. In the *International joint conference on biomedical engineering systems and technologies* (pp. 425-438). Springer, Berlin, Heidelberg.
101. Marjanovic, O. & Orlowska, M. E. (1999). On modeling and verification of temporal constraints in production workflows. *Knowledge and Information Systems*, *1*(2), 157-192.

102. Márquez, A. C. (2007). *The maintenance management framework: models and methods for complex systems maintenance*. Springer Science & Business Media.

103. Mayr, R. (2000). Process rewrite systems. *Information and Computation*, *156*(1-2), 264-286.

104. McDermott, D. (1982), A Temporal Logic for Reasoning about Processes and Plans, *Cognitive Science*, 6, 101-155.

105. Melao, N, and Pidd, M. (2000). A conceptual framework for understanding business processes and business process modelling. *Information Systems Journal* 10.2, 105-129.

106. Mills, S. F., & Tanik, M. M. (1995, August). Process models as templates for healthcare standardization and management. In *Proceedings of the Fourth International Conference on Image Management and Communication (IMAC 95)* (pp. 226-233). IEEE.

107. Müller, R., & Rogge-Solti, A. (2011, February). BPMN for healthcare processes. In *Proceedings of the 3rd Central-European Workshop on Services and their Composition (ZEUS 2011), Karlsruhe, Germany* (Vol. 1).

108. Mylopoulos, J., Borgida, A., Jarke, M. and Koubarakis, M. (1990) Telos: Representing Knowledge about Information Systems. ACM Transactions on Information Systems, 8(4), 325-362.

109. Neubauer, T., Klemen, M., & Biffl, S. (2006). Secure business process management: A roadmap. in *Availability, Reliability and Security. ARES 2006. The First International Conference*, IEEE.

110. Newell, A. (1982). The knowledge level. *Artificial intelligence*, *18*(1), 87-127.

111. OMG (2013). Business Process Model and Notation (BPMN), https://www.omg.org/spec/BPMN/About-BPMN/formal/2013-12-09

112. OMG (2015). Unified Modelling Language (UML), https://www.omg.org/spec/UML/2.5/About-UML, formal/2015/03/01

113. Orman, L. V. (1998). A model management approach to business process reengineering. *Journal of Management Information Systems*, *15*(1), 187-212.

114. Ouyang, C., Dumas, M., Ter Hofstede, A. H., & Van der Aalst, W. M. (2006, September). From BPMN process models to BPEL web services. In *2006 IEEE International Conference on Web Services (ICWS'06)* (pp. 285-292). IEEE.

115. Panagos, E., & Rabinovich, M. (1997). Predictive Workflow Management. In *NGITS* (p. 0).

116. Perreault, L. E., Shortliffe, E. H., & Wiederhold, G. (2001). Medical informatics: computer applications in health care and biomedicine. Springer.

117. Peter, V. B. (1992) Reasoning about qualitative temporal information. Artificial Intelligence, 58, 297-326.

118. Peterson, J. L., (1977). Petri Nets. ACM Computing Surveys Vol 9(3), pp 223–252.

119. Peterson, J. L, (1981). Petri Net Theory and the Modelling of Systems Prentice Hall

120. Phalp, K., & Shepperd, M. (2000). Quantitative analysis of static models of processes. *Journal of Systems and Software*, *52*(2-3), 105-112.

121. Powell, S. G., Schwaninger, M., & Trimble, C. (2001). Measurement and control of business processes. *System Dynamics Review: The Journal of the System Dynamics Society*, *17*(1), 63-91.

122. Pozewaunig, H., Eder, J., & Liebhart, W. (1997, September). ePERT: Extending PERT for workflow management systems. In *ADBIS* (pp. 217-224).

123. Prior, A. N. (1955). Diodoran modalities. *The Philosophical Quarterly (1950-)*, *5*(20), 205-213.

124. Recker, J. (2010). Opportunities and constraints: the current struggle with BPMN. Business Process Management Journal, 181-201.

125. Recker, J. C., Indulska, M., Rosemann, M., & Green, P. (2005). Do process modelling techniques get better? A comparative ontological analysis of BPMN.

126. Recker, J. C., & Mendling, J. (2006). On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages. In *The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium* (pp. 521-532). Namur University Press.

127. Recker, J., & Mendling, J. (2007). Adequacy in process modeling: A review of measures and a proposed research agenda-position paper. In *CAiSE 2007 Workshop Proceedings Vol. 1* (pp. 235-244). Tapir Academic Press.

128. Reising, W., Muchnick, S. and Schnupp, P. (eds.) (1992). A Primer in Petri Net Design, Springer-Verlag, Berlin.

129. Rogge-Solti, A., Kunze, M., Awad, A., & Weske, M. (2011). Business process configuration wizard and consistency checker for bpmn 2.0. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 231-245). Springer, Berlin, Heidelberg.

130. Rolland, C. (1993). Modeling the Requirements Engineering Process, 3rd European-Japanese Seminar on Information Modelling and Knowledge Bases. *Budapest, Hungary*, *6*.

131. Russell, N., Van der Aalst, W.M.P., Ter Hofstede, A.H.M. and Wohed, P. (2006). On the suitability of UML 2.0 activity diagrams for business process modeling, Proceedings of the 3rd Asia-Pacific Conference on Conceptual Modeling, Australian Computer Society Vol. 53, pp. 95-104.

132. Saboor, S., Ammenwerth, E., Wurz, M., & Chimiak-Opoka, J. (2005). MedFlow - Improving Modelling and Assessment of Clinical Processes. Studies in Health Technology and Informatics, 116, 521-526.

133. Salimifard, K., & Wright, M. (2001). MORaD-net: A visual modelling language for business processes. *New Models of Business: Managerial Aspects and Enabling Technology*, 216.

134. Scholz-Reiter, B., & Stickel, E. (Eds.). (2012). *Business process modelling*. Springer Science & Business Media.

135. Seila, A. F. (2005). Seven rules for modeling health care systems. *Clinical and investigative medicine*, *28*(6), 356.

136. Shoham, Y. (1987) Temporal Logics in AI: Semantical and Ontological Considerations. Artificial Intelligence, 33, 89-104.

137. Smith, A. (1776). The wealth of nations. *New York: The Modern Library*.

138. Smith, H., & Fingar, P. (2003). *Business process management: the third wave* (Vol. 1). Tampa, FL: Meghan-Kiffer Press.

139. Spyrou, S., Bamidis, P., Pappas, K., & Maglaveras, N. (2005). Extending UML activity diagrams for workflow modelling with clinical documents in regional health information systems. In *Connecting Medical Informatics and Bioinformatics: Proceedings of the 19th Medical Informatics Europe Conference (MIE2005). Geneva, Switzerland* (pp. 1160-1165).

140. Stefanelli, M. (2004). Knowledge and process management in health care organizations. *Methods of information in medicine*, *43*(05), 525-535.

141. Störrle, H. (2005). Towards a Petri-net semantics of data flow in UML 2.0 activities. *IFI-PST, Universitat Munchen, Oettingenstr., Munchen, Germany*.

142. Szwarcbord, M. (2005). Redesigning care: improving flow across the hospital [Conference Presentation]. *Glenelg: 1st Australasian Redesigning Health Care Summit, Glenelg South Australia*, 8-9.

143. Ter Hofstede, A. H., Van der Aalst, W. M., Adams, M., & Russell, N. (Eds.). (2009). *Modern Business Process Automation: YAWL and its support environment*. Springer Science & Business Media.

144. Tiwari, A. (2001). *Evolutionary computing techniques for handling variable interaction in engineering design optimisation* (Doctoral dissertation, Cranfield University).

145. Tsalgatidou, A., & Junginger, S. (1995). Modelling in the Re-engineering Process. *ACM SIGOIS Bulletin*, *16*(1), 17-24.

146. Tuzhilin, A. (1995) Templar: A Knowledge-Based Language for Software Specifications Using Temporal Logic. ACM Transactions on Information systems, 13(3), 269-304.

147. Valiris, G., & Glykas, M. (2004). Business analysis metrics for business process redesign. *Business process management journal*, *10*(4), 445-480.

148. Van Benthem, J. (1983). The Logic of Time, KluIr Academic, Dordrech.

149. Van der Aalst, W.M.P. (2004a), Process mining: a research agenda, Computers in Industry, Vol. 53, pp. 231-44.

150. Van der Aalst, W.M.P. (2004b), Workflow mining: discovering process models from event logs, IEEE Transactions on Knowledge and Data Engineering, Vol. 16 No. 9, pp. 1128-42.

151. Van der Aalst, W.M.P, Arthur H.M.T.H., and Mathias W. (2003).Business process management: A survey. *Business process management*. Springer Berlin Heidelberg, 1-12.

152. Van der Aalst, W. M. P., & Van Hee, K. M. (1996). Business process redesign: A Petri-net-based approach. *Computers in industry*, *29*(1-2), 15-26.

153. Van Gorp, P., & Dijkman, R. (2012). A visual token-based formalization of BPMN 2.0 based in-place transformations. Information and Software Technology, 365-394.

154. Vergidis, K., Tiwari, A., & Majeed, B. (2008). Business process analysis and optimization: Beyond reengineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(1), 69-82.

155. Vergidis, K., Turner, C. J., & Tiwari, A. (2008). Business process perspectives: Theoretical developments vs. real-world practice. *International journal of production economics*, *114*(1), 91-104.

156. Vila, L. (1994). A Survey on Temporal Reasoning in Artificial Intelligence, AI Communication, 7(1), 4-28.

157. Villain, M. and Kautz, H., "Constraint propagation algorithms for temporal reasoning", Proceedings of the Fifth National Conference on Artificial Intelligence, 377-382, Philadelphia, Pa. (1986)
158. Völkner, P., & Werners, B. (2000). A decision support system for business process planning. *European Journal of Operational Research*, *125*(3), 633-647.
159. Völzer, H. (2010, September). A new semantics for the inclusive converging gateway in safe processes. In *International Conference on Business Process Management*(pp. 294-309). Springer, Berlin, Heidelberg.
160. Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, *28*(1), 75-105.
161. Wei, W. (2010). A translation from BPMN to Event-B. *Manuscript, March*.
162. Weidlich, M., Decker, G., Großkopf, A., & Weske, M. (2008, November). BPEL to BPMN: The myth of a straight-forward mapping. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 265-282). Springer, Berlin, Heidelberg.
163. Weske, M. (2007). Concepts, Languages, Architectures. Vol. 14. Berlin: Springer-Verlag.
164. WfMC (1999). Workflow Management Coalition Terminology & Glossary: WFMC-TC-1011, Hampshire, United Kingdom.
165. White, S. A. (2004). Introduction to BPMN. *IBM Cooperation*, *2*(0), 0.
166. White, S. A. (2004). Process modeling notations and workflow patterns. *Workflow handbook*, *2004*, 265-294.
167. Wilson, R. M. and Harrison, B. T., (2002), "What is clinical practice improvement?," Intern Med J, 32(9-10): pp.460-464.
168. Wohed, P. (2004). Pattern-based Analysis of UML Activity Diagrams, Beta, Research School for Operations Management and Logistics, Eindhoven.
169. Wohed, P. van der Aalst, W.M.P, Dumas, M., ter Hofstede, A.H.M. and N. Russell. (2006). On the Suitability of BPMN for Business Process Modeling, Business Process Management, Vienna, pp. 161-76.
170. Zaidi, A. K. (1999). On temporal logic programming using Petri Nets. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *29*(3), 245-254.
171. Zaidi, A. K., & Wagenhals, L. W. (2006). Planning temporal events using point–interval logic. *Mathematical and computer modelling*, *43*(9-10), 1229-1253.

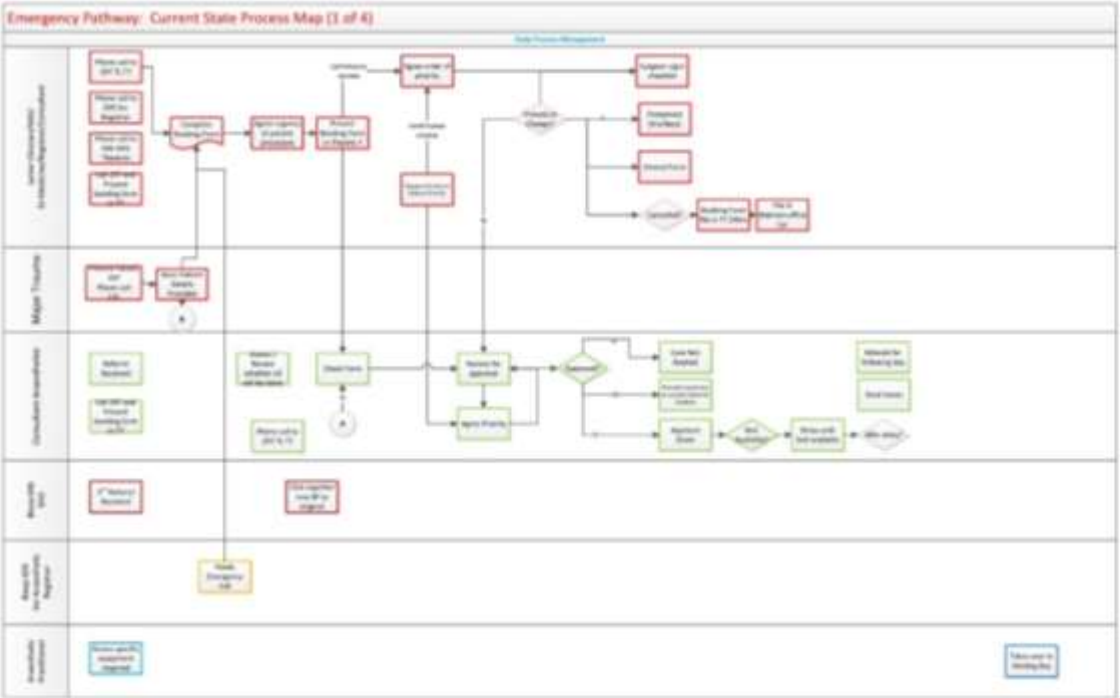
172. Zakarian, A. (2001). Analysis of process models: A fuzzy logic approach. *The International Journal of Advanced Manufacturing Technology*, *17*(6), 444-452.

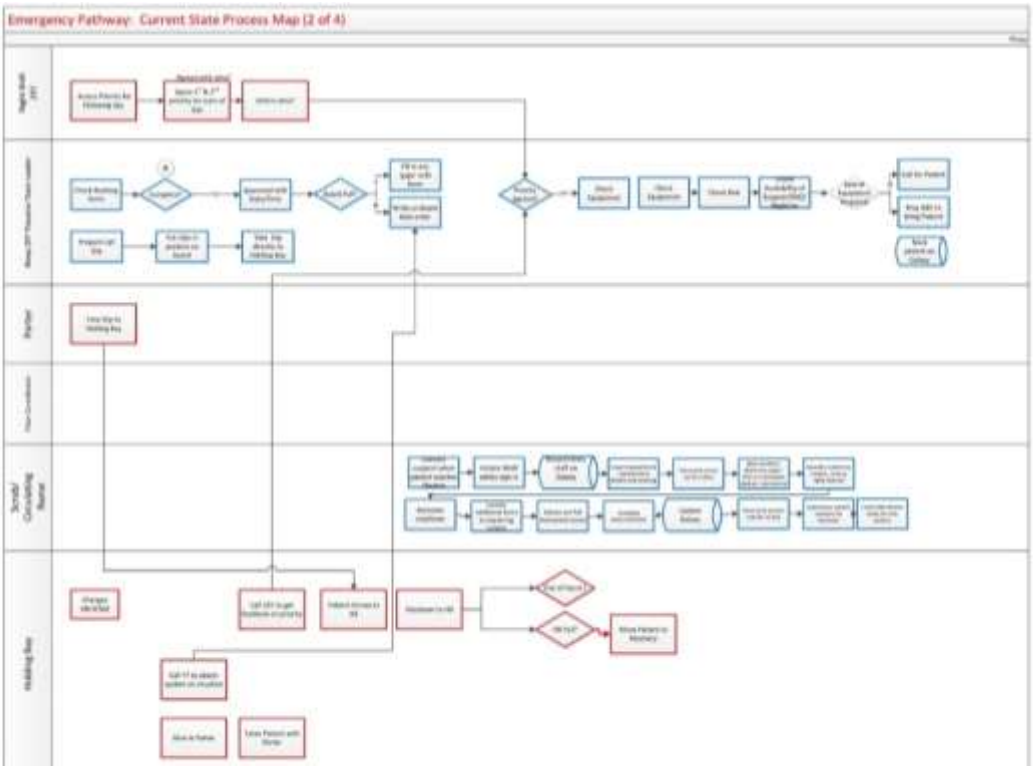
173. Zhou, Y., & Chen, Y. (2003, November). The methodology for business process optimized design. In *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No. 03CH37468)* (Vol. 2, pp. 1819-1824). IEEE.

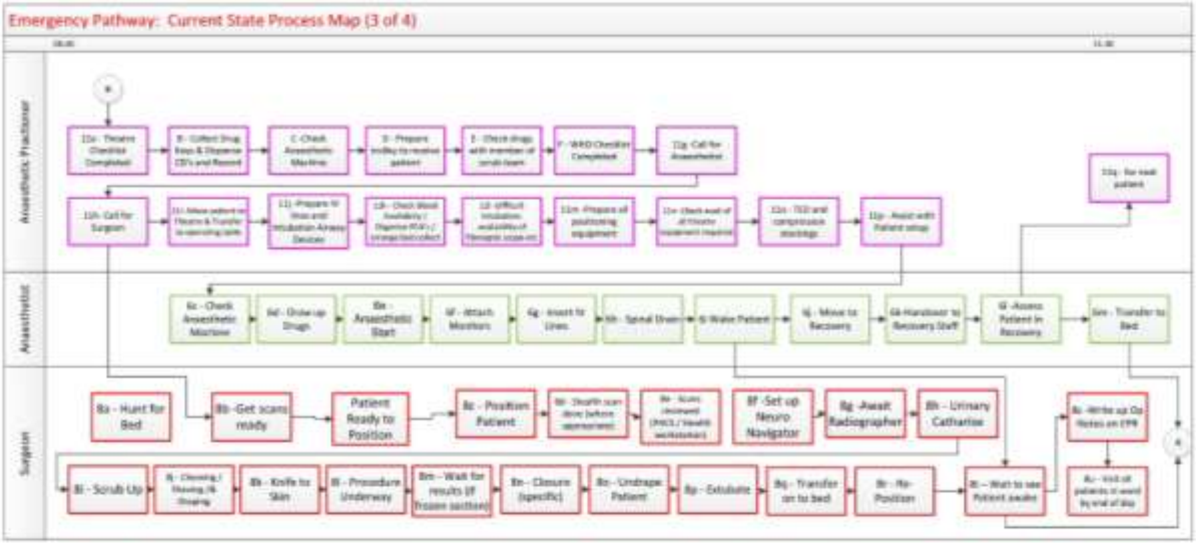
174. Zur Muehlen, M., & Recker, J. (2013). How much language is enough? Theoretical and practical use of the business process modeling notation. In *Seminal Contributions to Information Systems Engineering* (pp. 429-443). Springer, Berlin, Heidelberg.

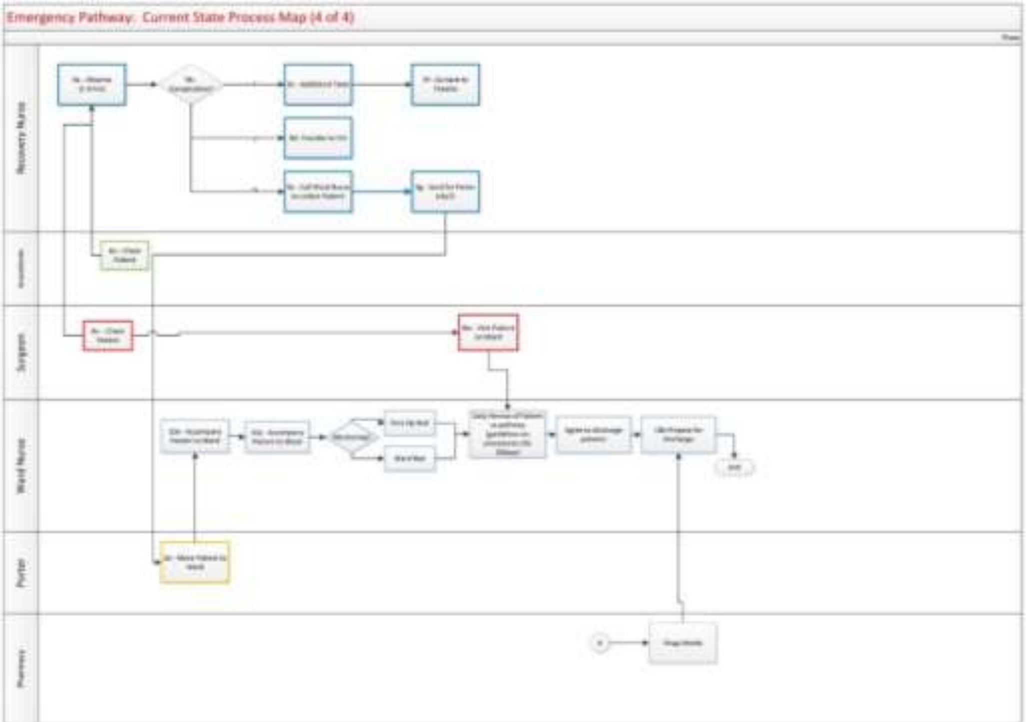
Appendices

Appendix 1: Sample Process Map (King's College) of Current state-Emergency Pathway

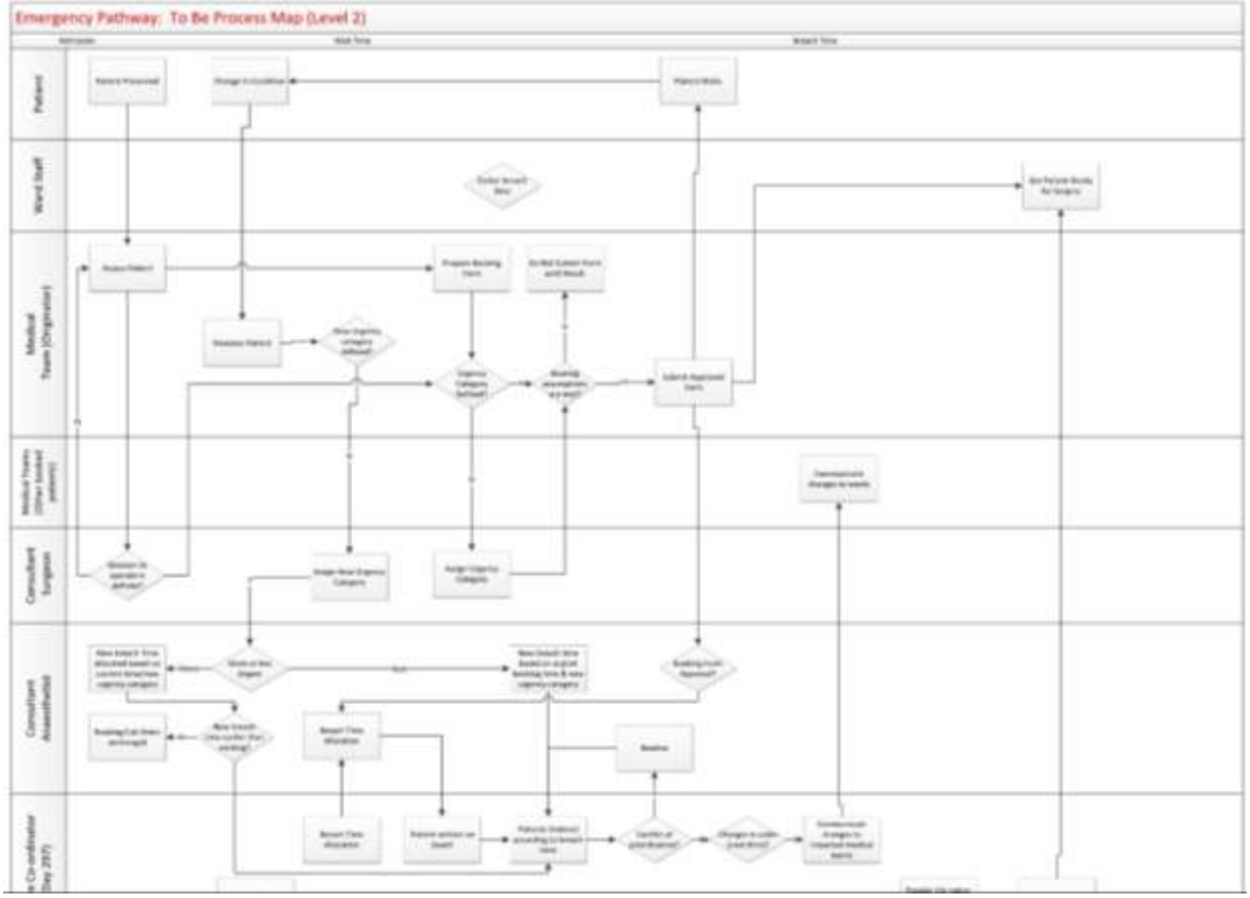


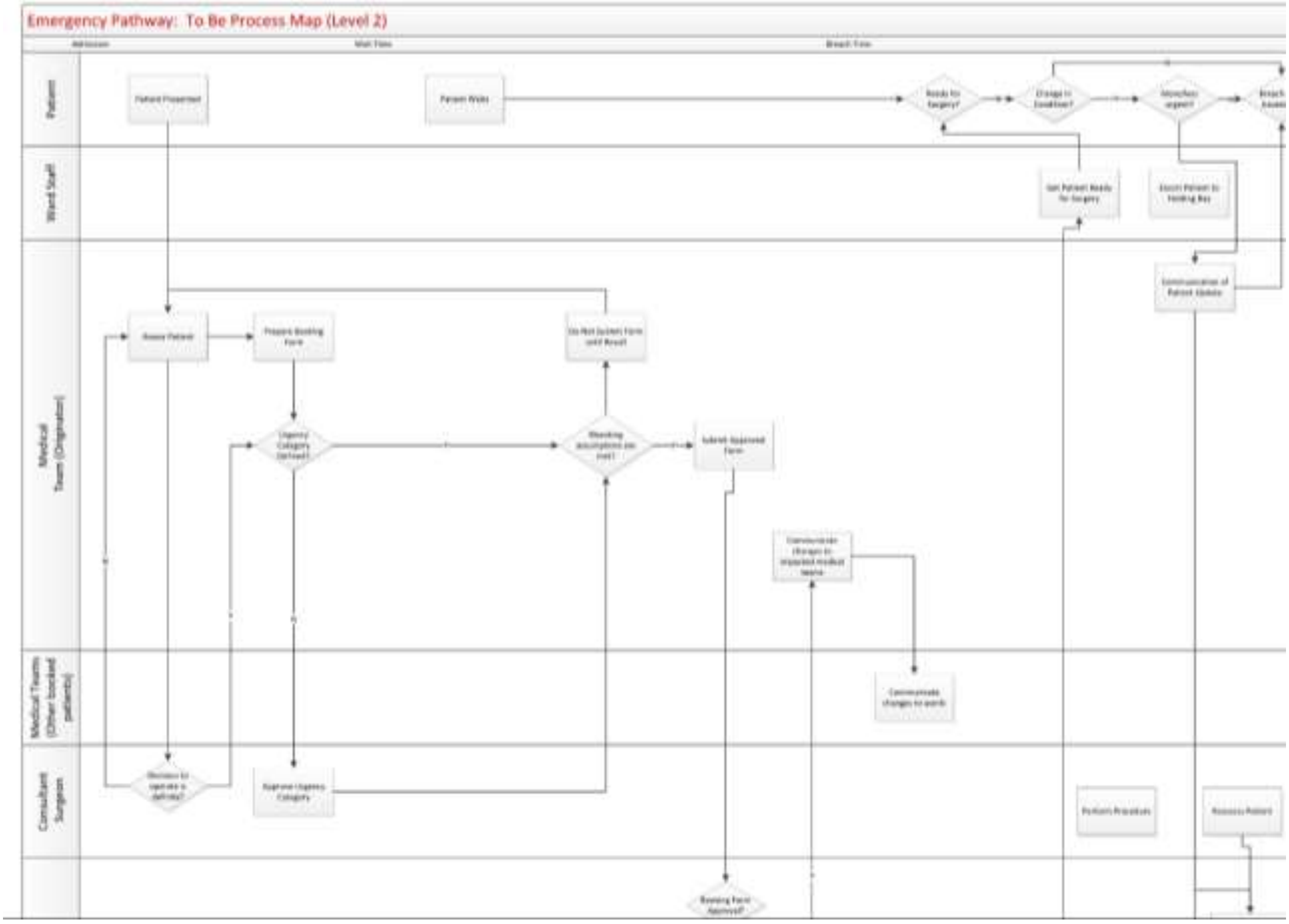



Emergency Pathway: Current State Process Map (3 of 4)


Emergency Pathway: Current State Process Map (4 of 4)

Appendix 2: Sample Process Map (King's College) To be Process-Emergency Pathway




Emergency Pathway: To Be Process Map (Level 2)
Patient
Ward Staff
Medical Team (Originator)
Medical Teams (Other booked patients)
Consultant Surgeon