Chapter 1
Introduction

Business objectives are met by successful execution of several business processes. Business processes are represented as sets of tasks, where each task carries out a host of well-defined activities. A task includes logical steps of actions that contribute towards the business goals. A sequence of such logical steps make a workflow. So, a workflow can be defined as a process consisting of a number of activities that need to be coordinated to achieve a particular business goal. For example, business processes in manufacturing and banking domains can be modeled as workflows. A workflow management system (WFMS) is a system that supports process specification, enactment, monitoring and coordination. Execution of a workflow is based on business logic as specified in workflow model.

Being able to change a workflow model or to create a new one quickly provides competitive advantage to organization to carry out business process efficiently. In order to facilitate workflow modeling we require a convention to represent workflows. A workflow designer follows different modeling approaches to model business processes using software tools. In general, the workflow modeling approaches fall into two categories: structured and formal. Structured approach follows a systematic way to compose business processes and the structured design is modular. Each module is represented by a graphic notation and modules are connected by edges. Further a module can be decomposed into submodules
and so on to atomic activities. In a workflow atomic activities are connected by directed edges. The direction of an edge shows the control flow during workflow executions. Thus this model provides a blueprint of workflow system design. Such modeling approaches are popular among software engineers because of their visual appeals. A designer intuitively verifies workflow system design by going through design diagrams. In contrast to the structured approach, a formal approach insists on mathematical statements on business process behaviors and interactions. The use of a formal approach helps the workflow designers to detect inconsistency, reduce ambiguity in the development and reason about the system. Hence, the use of formal methods is being seriously promoted among the designers of workflow systems.

Of late, workflow technology is being experimented in healthcare domain to manage complex tasks involved in treating a patient particularly for the one suffering from complex ailments. Corporate healthcare organizations intend to make use of workflow technology to deliver smart services to patient. However, it is observed that the classical workflow concepts being used in manufacturing domain are not readily usable in healthcare domain due to the specialties and challenges the domain projects.

In order to avail advantages from both the structured and the formal approaches, we propose a hybrid approach to specify workflow in case of treatment in healthcare. Some primitives with mathematical rules are identified for workflow design. Firstly a structured method is proposed to specify treatment by sequencing treatment activities. For healthcare workflow modeling, one can make use of the modeling primitives like linear, parallel, choice, repeating, nested repeating, split & merge, supportive and cooperative to specify treatment activities. These sequencing of treatment activities can be termed as Treatflow. A Treatflow is similar to traditional workflow used in manufacturing and other domain. Further,
it is augmented by special primitives viz. *supportive, cooperative* those are found useful to model activities in treating a patient. Secondly for each design primitive, mathematical specification is proposed to detect inconsistency, reduce ambiguity in patient treatment.

Once a Treatflow is specified, it is essential to verify the flow before making it operational as some mistakes might have crept in inadvertently while it is being composed by a doctor—a non-computer science professional. Traditionally, verification of workflows deals with structural verification comprising a study on lack of synchronization and deadlock detection [86][2]. In case of Treatflow verification we have discussed various verification issues viz. incompleteness, lack of synchronization, deadlock, retention conflict, contextual conflict, expectation conflict and temporal conflict keeping the relevance of healthcare domain in view.

Treatment in healthcare deals with complex ailments of patients. Modeling Treatflow for all these treatments becomes complex to design, verify and follow. This necessitates modular design of healthcare activities. Building a modular system is appreciated for its uses in enhancing system understanding and for its ability in dealing with large and complex system i.e. by making a system modular thus scalable. In healthcare a treatment for a patient can be composed of several treatment modules where a treatment module is specified with several treatment activities. A specification of a Treatment Module needs to provide comprehensive view that can be useful for automation of Treatflow activities and also enables a practitioner and a patient to have a quick view on a treatment. Ideally, a Treatment Module needs to specify conditions at which it can be applied to a patient. After specifying treatment modules, these modules can be composed to form a treatment plan termed as *Treatplan*. For composition purpose several composition operators e.g. linear, parallel, choice, supportive and cooperative have been defined. Therefore, a prescription written by a doctor can be specified by an
expression of treatment modules. Due to non-availability of resources or incompatibility of treatments, there is a need to change a Treatplan and to write an equivalent Treatplan. This can be done with the help of rewriting rules as well as properties of composition operators. Treatment in healthcare needs to be extremely sensitive for safety of a patient i.e an exception in health parameter needs to be treated. So exception handling is a necessary part of treatment workflow. Studying exceptions in healthcare workflow and exploring how such exceptions can be handled promptly and appropriately within the workflow system are to be studied to improve healthcare quality and efficiency.

The proposed Treatflow Management System aims at automating treatment process, it is different from existing healthcare systems which focus mainly on patient registration, bill collections, maintaining the history of medication undergone by patients i.e especially on issues related to hospital management and EPR: electronic patient record management. We propose a system where the authenticated users (doctor, staff and patient) can use this system. On registration of a patient, a doctor can make use of the proposed system to synthesize a treatment process by choosing treatment modules for treating a patient. And then a synthesized treatment process is instantiated; that means the patient is subjected to the treatment. A patient can record his/her health status as well as health related exigencies. Using the system, the doctor can issue health advisory to the patient in case of exigencies and can change the treatment process, if necessary. Users of the system can access treatment related information based on their roles. The main objective of Treatflow Management System (TFMS) is not only for automating the existing treatment process but also for bringing transparency in the treatment process.
1.1 Motivation

To our knowledge no effort has been made to model a treatment. Instead current work concentrates on modeling workflows on hospital management issues like personnel scheduling, resource procurement, ward management etc. In this work we focus on modeling a workflow that is patient-centric integrating the roles of doctors as well as healthcare staffs (diagnostic staff, nurses etc.) involved in treatment of a patient. In order to model treatment of a patient suffering from ailment(s), we have extended primitives of activity diagram for making it expressive to model workflows in healthcare domain.

1.2 Issues for Investigation

This thesis first focuses not only in differentiating workflow being practiced in manufacturing industry with the workflow of healthcare domain, but also identifies exciting features in healthcare workflow. The issues investigated in this thesis include Treatment Modeling, Treatflow Verification, Modularization of Treatment, Exception Management, Architecture for TFMS. Below, we deal upon each issue individually.

- **Treatment Modeling**

  Structured design specifies a healthcare system diagrammatically while formal methods use mathematical techniques. In order to avail advantages from both the approaches, we propose a hybrid approach to specify Treatflow in healthcare. In order to design Treatflow, primitives like linear, parallel, split and merge, repeating, nested repeating and choice are identified. For each primitive mathematical rules are specified so that the approach not only helps in Treatflow design but also for its verification. A resultant model is not only structured but also equipped with rigor that makes the model safe
for its use in healthcare domain that deals with human.

- **Treatflow Verification**
  Once a Treatflow is specified, it is essential to verify a Treatflow model before making it operational. Treatflow verification is required in order to ensure the safety of a treatment. For the purpose of verification Treatflow is represented as a graph comprising of nodes and edges. Where a node represents a treatment and an directed edge represents control in treatment execution. Keeping healthcare domain in view verification issues like lack of synchronization, deadlock detection, incompleteness check, retention conflict, contextual conflict, expectation conflict and delay conflict etc must be taken into consideration.

- **Treatment Modularization**
  Modeling a treatment for a patient suffering from multiple ailments or for a chronic patient becomes very critical to understand and to transmit it to supporting healthcare staff. The issue is similar to the representation of a design of a complex software system for which modularization has been recognized as a solution [82]. This has motivated modularization of treatment specification. The safeguards in modules must be specified which can alert medico, nursing staff as well as patients on prescription and administration of wrong medications. In case of incompatibility of treatments doctors may need to generate alternate Treatplans to facilitate decision making by doctors, patients as well as healthcare managers. These requirements motivate us on composition of a Treatplan with treatment modules as well as rewriting rules for an alternative Treatplan.

- **Exception Management**
  Healthcare treatment often requires human participation of doctors, patients
and collaborating staff ranging from paramedical to administrative personnel. Though each of them play well defined roles still human actions are liable to lapses. Patient treatment involves life risks; hence healthcare workflow management should have means to deal with these lapses. These factors emphasize the necessity of exception management in healthcare domain.

- **Architecture for TFMS**
  In order to automate treatment process which is different from current healthcare systems that focus especially on hospital management and EPR: electronic patient record management, we propose TFMS (Treatflow Management System). The purpose of TFMS is to specify and verify a Treatflow as well as to execute, monitor and handle exceptions while treating a patient. This system is to be used as an aid by different actors to participate in treatment process. We feel, such a system will be helpful for both doctors and patients in managing treatment process. On registration of a patient symptoms must be recorded by an authorized administrative staff. A doctor can plan a treatment for the patient by selecting treatment modules from a treatment repository. Such a repository could be made available by doctors recording their experiences and practices. A patient is subjected to treatment. While undergoing treatment a patient can provide information on his/her health related exigencies, update health status and can access treatment related informations. After going through patient’s health status and health related exigencies, a doctor can change a treatment. This motivates us to propose a system architecture which aims at addressing these requirements.
1.3 Dissertation Outline

The remainder of this dissertation is organized as follows.

Chapter 2 first describes a survey of the background that is necessary to understand the notion of workflow and its management. Workflow modeling and its features are described in 2.3. Several workflow modeling techniques are presented in section 2.4. In order to achieve correctness, efficiency and effectiveness of business process supported by workflow management system, it is necessary to analyze a workflow process definition before it is put into production which is discussed in section 2.5. Some of the exciting features of workflow are described in section 2.6. Healthcare workflow and its features as well as healthcare workflow modeling are discussed in section 2.7. Finally some of the proposed methods for modeling healthcare workflow are presented.

Chapter 3 presents a hybrid approach to model healthcare workflow systems. Firstly it describes about modeling treatment activities and its related work in healthcare domain. To specify a Treatflow in healthcare domain with the help of a case study is discussed in section 3.2. In order to model a Treatflow using primitives like linear, parallel, choice, repeating, nested repeating and cooperative and supportive, a visual object have been identified so that workflows can be presented and viewed diagrammatically. Mathematical rules for each design primitive have also been defined in section 3.3. In order to verify a Treatflow, several verification issues are discussed in section 3.4. With the help of a verification algorithm various verification issues are discussed considering Treatflow as a graph.

Complex health problem treatment is modularized not only for administering the treatment but also to monitor and to modify if necessary. These requirements motivate us to plan a treatment in modular form which is discussed
in section 3.5. In order to motivate readers on modularization, this section describes this concept with the help of a case study. Treatment module can not only be specified and composed to form a treatment plan called Treatplan but also can generate an equivalent treatment plan using rewriting rules which is also discussed in this section. Exception management in treatment process is discussed in section 3.6 and presented a comprehensive framework to deal with it. In this section we analyzed the domain and presented a comprehensive view on genesis of exceptions and their corresponding actions.

Chapter 4 discusses about Treatflow Management System (TFMS) which aims at automating treatment process. Here we propose a framework to implement Treatflow Management System. A requirement analysis for such a system considering users viz. (doctors, patients and staffs) and their roles in treatment process has been discussed in section 4.3. System architecture and its component are discussed in section 4.4 A broad picture on design of each module with the help of class diagrams is discussed in 4.5. Analysis is done with the help of interaction diagram. Based on each user requirements how the modules interact among themselves is discussed in section 4.6.

Chapter 5 gives a brief conclusion, summarizing the work described in this thesis and some direction for future work.