Chapter 2

Survey

This chapter surveys basic workflow modeling features, various workflow modeling techniques and verification of workflow management systems. We review some of the expected features of workflow. It also identifies the basic issues of research interests associated with workflow in healthcare. Some research related to modeling and verification in case of healthcare workflow are also discussed.

2.1 Introduction

An organization performs several business processes to achieve its goals. A process is carried out by people/systems where each of them carries out tasks assigned to. For example, in an assembly sub-floor technicians engaged in making a product, participate in a definite sequence. A defined order of working with an objective to produce an object (to realize a business goal) is termed as a workflow. For example, a business process "Issue a Loan" may consist of several tasks like: reviewing an application for completeness and accuracy, executing a credit check, creating a new loan contract, sending it to the client, checking the contract form on return, setting up of payment schedule and finally issuing payment cheques. Sometimes a workflow of a business process may span over several organizations. This happens for the case where multiple agencies are involved in making a product (say aircraft manufacturing). In early days, assigning tasks to people and monitoring them were carried out manually. Of course, such manually maintained business
process is prone to human error. Later, at the advent of technology, automation has been brought in to manage workflow. At the beginning, a-piece-meal approach was adopted to automate certain activities like material management, personnel management. Later in several areas like distributed computing, DBMS, communication technologies, there has been attempt in total automation of organization activities. This has resulted in systems called Workflow Management Systems (WFMS).

In the following, firstly workflow, its management and workflow modeling features are discussed in section 2.3. Different workflow modeling techniques, formal methods in workflow as well as comparison among them, are presented in section 2.4. Analysis of workflow is discussed in section 2.5. Some of the expected features of workflow are identified in section 2.6. Details about healthcare workflow is presented in section 2.7. Finally a summary of the chapter is given.

### 2.2 Workflow and its Management

Workflow is defined as the automation of a business process, in whole or parts, during which documents, information or tasks are passed from one participant to another for action according to a set of procedural rules. Therefore workflow is a sequence of tasks organized to accomplish some business process. These business processes can be executed in sequence or parallel to accomplish the goals of an organization. According to the needs, a workflow can be started manually or automatically at a given time or situation. If there is a problem in executing a particular process, workflow tools have the capability to suspend the workflow and then the workflow can be started once again to run the process. It also defines the order of task invocation or condition under which task must be invoked, task synchronization and information flow. Patient treatment in a hospital, banking
services, manufacturing and catalog ordering processes and many more activities in todays business life can be modeled as a workflow.

Workflow in a business process can be managed by a workflow management system. In general term a Workflow Management System (WFMS) is a computer software that supports, controls, manages and collaborates among business processes through automatic flow of tasks, electronic information, and documents [5]. It also has a provision to integrate with other workflow systems. A WFMS can be used as a tool to model, specify, analyze and execute a workflow [35]. It also helps organizations to schedule their underlying processes and to carry out activities effectively and efficiently. Currently, most of the business houses carry out their activities at geographically distributed places. A distributed WFMS is a natural choice to automate business activities of such business houses [94].

Workflow management system consists of two basic components: First component is the workflow modeling component or buildtime system which enables administrators or analysts to define processes and activities, analyze and assign them to people. It also provides graphical modeling tools that help designers to design, test and validate workflow processes. Second component is the workflow execution component, sometimes called runtime system composed of software which are responsible for creating and controlling instances of business processes at run time. It helps in coordinating and performing the processes and activities [8] and provides an execution interface for end-users.

Workflow software records information of business process execution into a repository. Using workflow monitoring tools information like starting and finishing time, status (started, stopped, succeeded), errors and exceptions handled in business processes can be easily retrieved from repository by querying for further analysis. Workflows can also be used to notify the process status through email to the top most executives of the company or whoever in need of it.
Descriptive level of WFMS deals with describing the modeling aspect of workflow system. It describes what a WFMS will do, which procedure will it implement/monitor and what data it stores, for which organization WFMS is a part of and which type of user-interface will it present and so on. Execution level of WFMS is further decomposed into enactment and management. Enactment deals with the effective execution of activity specified at the process description i.e what activities are being performed and who is performing the activity. Management level has privilege access to information about current and past work cases, statistical information about users, processes, activities and so on [7].

Usually workflow management has been process focused i.e coordinating activities of people working on a common task or project. Once a process is defined, workflow management system makes sure that activities occur in proper sequence and the users are informed. A business process is a collection of activities designed to produce a specific outputs for a particular customer. It implies a strong emphasis on how the work is done in an organization instead of focusing on what. Therefore a process has a specific ordering of work activities with a beginning and an end with clearly defined inputs and outputs. Modeling is an efficient and effective way to represent the need of an organization. It provides information to members of an organization to understand and communicate business rules and processes. In the next section we will discuss about workflow modeling and its features.

2.3 Workflow Modeling and its Features

Modeling is an efficient and effective way to represent the need of an organization. Workflow involves a set of activities to be carried out in order to meet some predefined requirements of an organization. Workflow modeling is a key activity
during process improvement because techniques provide an effective means of analyzing existing workflow and comparing proposed improvements [80]. Workflow processes are marked by three dimensions: (1) control-flow dimension (2) resource dimension and (3) case dimension. Control dimension is concerned only with partial ordering of tasks. Tasks need to be executed are identified and the routing of cases along with these tasks is determined i.e conditional, sequence, parallel and iterative routings are the typical structures specified in control-flow dimension as shown in Figure 2.1 and Figure 2.2. In resource dimension, resources (human or device) are classified by identifying their roles. Lastly in case dimension workflow is concerned with individual case which are executed according to process definition by the proper resource dimension [84].

Therefore workflow modeling is nothing but the representation of activities that compose workflow processes, their execution sequence and relationships, agents responsible for their execution, and resources that are used during execution [59]. In an organization workflow management which is already existed, but recently workflow modeling has become the main focal point of researchers. Researchers have proposed different modeling techniques. Several other techniques such as flowcharts, state transition diagram, data flow diagram have been proposed to represent business process, but some of the conventional modeling techniques have some drawbacks: lack of formal semantics and absence of powerful analysis methods and tools [90].

Workflow Modeling Features The following perspectives are relevant for workflow modeling and execution [19] [88].

- Control-flow (Process) perspectives: Describes the activities and their order of execution e.g sequential, parallelism, choice and join synchronization

- Data(Information) perspectives: Deals with business and processing data on control perspectives
• Resource (Organization) perspectives: Provides an organizational structure in the form of human and device roles responsible for executing the activities.

• Operational (Application) perspectives: Described the elementary action executed by activities.

In a workflow model the following minimal information are associated with an activity [62].

• Pre and post conditions (conditions to be met before and after an activity)

• Who has control over the activity (role)

• Which other activities are required to complete the activity
• Input/output of an activity

Considering features of workflow modeling, several workflow modeling techniques are available which will be discussed in the next section.

2.4 Workflow Modeling Techniques

Workflow models explain the activity interdependency and timing, branching and merging of process flow, choice, looping and parallelism in much greater detail. Workflow Model is a collection of several activities where each activity is represented by a box with a verb describing an activity. For example, Check Credit History, Process an Order, Ship an Order are some of the examples for these [89]. In general, activities take inputs and transform them into outputs and the relationship between these activities is represented by arrows. The main objects or components in workflow modeling are activities or processes, one or more junctions and arrows. Junctions are nothing but a small box that enable branching or joining of processes. Several workflow modeling techniques are available, for example Petrinet, UML, Statechart, EPC and Formal Method etc..

2.4.1 Formal Methods in Workflow

Formal models are useful for specifying workflow properties and correctness study. For example, for specifying a property that needs to hold good during progress of workflow execution, one can make use of formal approach to state the property unambiguously because of its mathematical preciseness. In case of structured based modeling, specifying such property is cumbersome. Recognizing strength of formal approach we will review some of the recent works in this area. Before proceeding we will dwell upon two important publications from literature that laid emphasis on formal approach.
It has been shown that how mathematics can provide scientific foundation for modeling, describing and developing methods in software engineering [12]. It argues that while diagrammatic models provide syntactic approach to model still the approach lacks semantics to describe system properties. The author puts forward a mathematical approach so that a description model diagram can be expressed mathematically, and then to express relations among the models mathematically. Thus, diagrams together can project aspects of software systems that remain unspecified and individual models project system view in a discreet way. Here, the author shows refinement of a mathematical model is possible to specify system at different levels of abstraction.

Further, another paper [14] advocates the use of formal method in specifying a program. It identifies elements of structured design like assignment, if, if-else-if and while; and for each mathematical specification that mainly uses pre- and post conditions is proposed. It shows that a program can be thought of as a composition of blocks where each block is composed with a set of statements. Structured design primitives can be used to show the composition blocks as well as composition of each block. It shows a process following which one can refine a block into a set of statements which again can be specified by design structures accompanied by mathematical specifications. The paper shows that, a program not only can be well-defined by mathematical specifications but also can be verified.

There has also been some work reported on formal specification of workflow in wider context business process. In a recent work [93] CSP: Communicating Sequential Process has been used to specify complex workflow systems. A process-algebraic approach is suggested to model and verify workflow processes. Authors propose CSP process model to specify and verify the model by comparing model behavior to expected behavior of a workflow process. Process refinement algebra is used to define design and development of a workflow. Scalability of the
approach is shown by providing rules to integrate several workflows for building a complex workflow system. In order to demonstrate the strength of the model the authors have proposed a formal semantics for BPMN: Business Process modeling Notation in CSP so that workflow designer can use BPMN for design and at the same time can make use of corresponding CSP model for verification. The paper indicates that for verification several available CSP based model checkers like FDR can be used.

In another work [61][4] CTL: Computational Tree Logic has been proposed to model workflows. A workflow is considered as a collection of events and associated actions. It does not consider structure to model workflow. Rather, it models execution of a workflow as processing of events. At a given time during execution, system selects the event that is to be executed by asking available formal automata where each automation models a dependency among events. Thus all the automata for a work process define a partial order among events. And the order is to be maintained during execution of the workflow. Thus CTL based formal modeling of workflow is not only useful to specify a workflow but also to verify. However, one to one relation between dependency and automata, projects the requirement of a large number of automata for a moderately complex system. Scheduling an event by coordinating a large number of events becomes computationally expensive. Transaction logic to model workflow is proposed [11]. It proposes atomic formula to compose transactions of a workflow system. There are a number of different flavors of TR; all these flavors come with a model theory for the full logic; and a sound-and-complete proof theory for the logic programming fragment. The proof theory for the logic programming fragment can be used to execute programs. The flavors differ in the operations for composing atomic operations into complex operations. In all these flavors, atomic formulas (predicates) represent elementary actions or queries on a database. Sequential Transaction
Logic: For composing atomic formulae into complex ones, TR uses the usual classical operators \((\neg, \land, \lor, \rightarrow)\) and two additional operators: serial, conjunction, denoted \(\otimes\) and a modal operator for executional possibility, denoted as \(\diamond\). These operators mean the following:

1. serial conjunction: \(\alpha \otimes \beta\) means first execute \(\alpha\) and then execute \(\beta\). \(M, \pi \models \alpha \otimes \beta\) if and only if \(M, \pi_1 \models \alpha\) and \(M, \pi_2 \models \alpha\) and \(\pi = \pi_1 \cdot \pi_2\). Here \(M\) is a path structure which determines the truth value of each formula on a path (indicating whether the formula can execute along a multipath), \(\pi\) is a path in \(M\), and \(\cdot\) denotes the concatenation of two paths.

2. classical conjunction: \(\alpha \land \beta\) if and only if \(M, \pi \models \alpha\) and \(M, \pi \models \beta\).

3. classical disjunction: \(\alpha \lor \beta\) means execute either \(\alpha\) or \(\beta\) \(M, \pi \models \alpha \lor \beta\) if and only if \(M, \pi \models \alpha\) or \(M, \pi \models \beta\).

4. negation: \(\neg \alpha\) means execute anything but \(\alpha\).

5. subprocedure: \(\alpha \leftarrow \beta\) means that to execute \(\alpha\) one can execute \(\beta\); \(M, \pi \models \alpha\) if and only if \(\alpha \leftarrow \beta\) is in \(M\) and \(M, \pi \models \beta\).

A workflow is modeled as a set of TR formulas that describe dependency among tasks. Tasks are modeled using predicates, they can be either atomic or a subprocedure i.e a composition of atomic formulas. This allows modeling of workflow activities by using compositional operators like sequence, choice or concurrence. Workflow execution depends on satisfaction of constraints associated to each activity. Thus like CTL approach this Transactional logic approach becomes computationally expensive.
2.4.2 Workflow Modeling with EPCs

The main elements of event-driven process chain are functions, events and logical connectors. A function corresponds to an activity (task, process step) needs to be executed. Functions can be represented using hexagons. Event describes a situation before or after a function is executed which can be represented by rectangles. Functions are linked by events. An event may correspond to post-condition of one function and act as a pre-condition of another function. Logical connectors can be used to connect the activities and events. The control flows of a business process can be described by using three types of connectors like \( \land \) (and), XOR (exclusive-or) and \( \lor \) (OR).

EPC can be mathematically defined as an EPC is of five tuples \((E,F,C,T,A)\)

- \(E\) is a finite set of events
- \(F\) is a finite set of functions
- \(C\) is a finite set of logical connectors
- \(T \in C \rightarrow \land, \text{XOR,} \lor\) is a function which maps each connector into a connector type.
- \(A \subseteq (E \times F) \cup (F \times E) \cup (E \times C) \cup (C \times E) \cup (F \times C) \cup (C \times F) \cup (C \times C)\) is a set of arcs.

For example graphical representation of EPC as in Figure 2.3 for modeling the processing of a customer order. The process starts with an event of customer order received. First the customer order data is checked and the result of this function may be accepted or rejected. After the execution of function (customer compare order) XOR connector models the fact that either of the two events (customer order accepted or customer order rejected) holds. If the order is rejected then the
process stops. In case the order is accepted, availability is checked. If the article is not available then two functions are executed in parallel, purchase material and make production plan. If either the event article available and finished goods holds then the function ship order can be executed. One of the main advantage of using EPCs is that it is easily understandable for end users [66] for its graphical representation. EPC is often used for capturing and discussing business processes with people who have never been trained in any kind of modeling technique (i.e worker on the shop floor). EPC creates a common platform for communication and the analysis of ideas [73].

Figure 2.3: Workflow Modeling using EPCs
### 2.4.3 Workflow Modeling with Petrinet

In order to implement a workflow system, it is first necessary to find a suitable means of designing and modeling a workflow process. Petrinets are a class of modeling tools originated from the early work of Carl Adam Petri. Petrinets have well-defined mathematical foundation and easy to use graphical feature. Classical Petrinet model has been used in many application areas, for example communication protocol, flexible manufacturing system and distributed information system. Among other modeling techniques, Petrinets are mainly used for quantitative and qualitative analysis of workflow and workflow system. The objective of qualitative system is to prove that the model is valid [40]. Classical Petrinet is a directed graph with two types of nodes called places and transitions. An arc connects from place to transition and from transition to place. Formally a Petrinet is a tuple

![Classical Petrinet](image1)

**Figure 2.4: Classical Petrinet**

![Dynamic Behavior of Classical Petrinet](image2)

**Figure 2.5: Dynamic Behavior of Classical Petrinet**
PN = \{ P,T,F, M_0 \} where P is a set of places, T is a set of transitions and F = 
\{P \times T \} U \{ T \times P \} flow relation between place to transition and from 
transition to place and M_0 is the initial marker

While representing graphically as in Figure 2.4 petrinet consists of three structural 
components: places, transition and arcs where places are drawn as circles and 
transitions are drawn as rectangles and arc as arrows. It can also be used to 
model asynchronous system with concurrency. The transitions represent events 
and the places represent states or conditions. Inputs and outputs are allowed only 
between places and transitions: one cannot go directly from one place to another 
without a transition. The dynamic behavior of a system can be represented using 
tokens as shown in Figure 2.5.

Tokens are represented as black dots in places. A transition is ”active” 
when each of its input places contains a token and each active transition in the 
diagram is ”fired” by removing one token from each input place and generating 
one in each output place. For example consider an automobile insurance company 
as shown in Figure 2.6. Where a workflow consists of five tasks submit-claim, 
check-insurance, contact-garage, send-letter and pay-damage. It is assumed that 
two tasks contact-garage and check-insurance may be performed simultaneously 
in order to determine whether the claim is justified. If the claim is justified, 
the damage is paid otherwise a rejection letter is sent to the claimant. However 
classical petrinets modeling for real system is found to be very large and complex 
and is inadequate to model complex industrial system. To solve these problems 
high level petrinets i.e extension to petrinet model have been proposed by authors 
[54]. Therefore an extended version of classical petrinet model has been used to 
model industrial system [81]. Three extensions are colour, time and hierarchy.
Colored Petrinet

In classical petrinet model it is impossible to distinguish between two tokens in same place. For example in case of two insurance claims in order to incorporate the separate characteristics of the two claims in a model, it is necessary to include policy number, name of the policy holder etc. Classical petrinet is extended using color in order to enable coupling of characteristics with corresponding token. In a colored petrinet a transition can only be enabled if there is a token at each of its input points and pre-condition must be met for values of token to be consumed as shown in Figure 2.7. In contrast to classical petrinet, the result of color extension is that graphical representation of colored petrinet can no longer contain all
information. For transition following factors must be ascertained.

- Whether there is a pre-condition and post-condition must be specified precisely.

- Number of tokens produced on each firing depend on value of tokens consumed.

- Value of tokens produced depend on value of tokens consumed.

Tokens often represent object (i.e. resource, goods or human) are colored which facilitates the modeling for objects having attributes. A good modeling method is essential in order to simplify the workflow processes and to integrate with other application in case of critical business process. To model family of workflow processes WFCP-net (Workflow net based on coloured Petrinet) is a petrinet based analysis technique to verify the workflow process specification [86]. WF-net allows users to find control flow errors [46].

**Timed Petrinet**

Since classical petrinet is not capable of handling time. By adding time extension to classical petrinet it is able to model the temporal behavior of system (i.e. to model duration and delay). Using time extension tokens can receive time stamp as well as a value. This indicates time from which token is available. Timing can be added to a petrinet by naming all the places and transitions, and drawing up a table with minimum and maximum times for each transition to occur based on the time of arrival of the tokens at its input place. If a transition fires and tokens are produced, each of these given a time stamp equal to or greater than the time of firing. Time stamp of produced token is equal to the time of firing plus the delay which is determined by firing transitions. For example as in Figure 2.8 with initial marking at time $t=1$, $t_1$ starts firing within a time less than $t=2$. If it does
not fire within that time it cannot fire any more because at time $t=2$, $t_2$ start firing having higher priority than $t_1$.

Figure 2.8: Time Petrinet

Hierarchical Petrinet

Using color and time extension though we are able to describe very complex processes, but they usually do not provide proper reflection of process. With ‘Hierarchy’ extension it is possible to add structure to petrinet model. Because a process can be constructed from sub-processes, which in turn also be constructed from sub-processes, therefore it is possible to structure a complex process hierarchically as shown in Figure 2.9. HiWorD (Hierarchical Workflow modeling) is a prototype with the simulation capability. It models the business process using petrinet in a hierarchical manner [9]. High level petrinets have been used in many application areas for example communication protocol, flexible manufacturing system, computer system, production system, administrative system, real time system, workflow system and distributed information system.

Reasons for Modeling in terms of High level Petrinet: There are few advantages for workflow modeling using high level petrinet. These are as follows.

- Reason for using petrinet based workflow management system is that it is easy to understand the graphical nature of petrinet and business logic can be represented formally [21].
• It is state based instead of event based.

• Petrinet based modeling a system can be analyzed using petrinet theory.

2.4.4 Workflow Modeling with UML

UML supports several different models of a system through different diagrams. Some of the diagrams describes the static information about the objects of the system whereas other concentrate on dynamic aspects of how the system behaves when it runs [79]. UML activity diagram and statechart has been discussed in this section.

Workflow Modeling with Activity Diagram

Major constructs for workflow modeling are sequence, parallel path, alternate path and iteration. An activity diagram is a directed graph consists of nodes and directed edges. An activity diagram models the behavior of a system where node represents state and edge represents transition [23]. Constructs for creating activity diagram are
• Start: can be used for beginning of a process

• End: indicates the end of a process

• Fork: can be used for splitting a process into several parallel execution paths

• Decision activity: can be used for providing alternative paths

• Iteration: can be modeled by joining two decisions.

As in Figure 2.10 for example in a manufacturing company first an order is received by the action receive_order. The diamond shape represents a decision node and then the execution of fill_order depend on the condition of acceptance and rejection. The fork splits the path of control flow into two. Then in the left side action like produce_goods followed by ship_goods gets executed and simultaneously on the right side two actions like send_invoices followed by receive_payment are executed. After the receive_payment, the behavior pay() is invoked. When they both have completed their execution, join (next black bar) may takes place and after that, the action close_order takes place. Among the set of UML diagrams used for dynamic aspects, activity diagram (AD) deals with Business Process (BP) but lacks

Figure 2.10: Workflow Modeling using Activity Diagram
well defined semantics which has been taken care of by the finite state processes (FSP) discussed in [74]. Compared to EPC, in case of business process modeling there are some problems with activity diagrams. Not all logical operators for splitting and joining the control flows can be modeled in a straightforward way. UML does not consider activity diagram for object external flow. UML activity diagrams are to succeed as a standard for organizational process modeling. In the context of workflow specification the strong points of activity diagrams are that they support signal sending and receiving at the conceptual level and they support both waiting and receiving state. Activity diagrams provide a mechanism for decomposing the activity into sub-activities, but some of their constructs lack a precise syntax and semantics [24]. An extension to UML activity diagram called WAD (Workflow Activity Diagram ) is used to describe the business process in production systems which applies the C-WF model concepts. The C-WF model represents the structural and functional enterprise objects involved in business processes [70]. In [53] a new approach has been discussed to do the analysis of workflow by modeling workflow using activity diagram and then converting the activity diagram to petrinet for analyzing formally.

**Workflow Modeling with Statechart**

Statecharts are graphic-oriented and their visual appeal leads to consider for representing reactive system. They are the extension of state-transition diagram. The basic elements of statecharts to represent a system are state, event, condition, action, transition, expression, variable and label. A business workflow can be formally represented by a statechart. An activity of a workflow whether it is automated or manual, is represented by a state in the statechart. An activity is being carried out only if the system is in the state that corresponds to that activity. Transitions between activities are thus modeled as transition between
states in the statechart which are triggered by events and guarded by conditions. Events can be external or internal. When the system is in the source state, event must occur and the condition must be true for the transition to take place [57]. States of a statechart are organized as a tree like structure, where root of the tree is called root state of the statechart, the children of a state are its immediate sub-states and the leaves of the tree are the basic states. In Figure 2.11

![Statechart and State transition diagram](image.png)

Figure 2.11: Workflow Modeling using Statechart and State transition diagram

a statechart and its equivalent state transition diagram are represented to show the problem of exponential growth of states in ordinary state transition diagram. We can easily model concurrency in a system by making use of orthogonal ”and” states in a statechart. On the other hand to perform the same modeling in the equivalent state transition diagram we require six states [57]. Extend Statecharts creates a alternative modeling technique that deals with the concurrency, parallelism, simultaneous use of resources in a clear and better way [16]. To specify a real time system, time semantics is introduced into UML class and statechart diagrams [75]. Time semantic allows to verify the properties of real time system by means of timed computation tree logic. Statecharts is a language that extends the notation of FSM (Finite State Machine) with the concepts of concurrency, hierarchy and broadcast communication. FSM transitions are labeled by pairs,
where the first pair is referred to as trigger and consists of positive and negative signals (an event) and the second element is referred to as an action. FSM states are either basic states, OR-states and AND-states [49]. In Figure 2.12 shows a purchase order business process provided by a vendor using FSM [92]. The process starts with a purchase order (c#v#PO) message followed by a delivery (c#v#Delivery) message and either a credit card payment (c#v#ccPay) or an invoice payment (c#v#InvoicePay) message. In case the ordered product is not in stock, the vendor may reject a purchase order by sending a no stock available (c#v#noStock) message.

Figure 2.12: Example for FSM

<table>
<thead>
<tr>
<th>Models</th>
<th>Controlflow perspectives</th>
<th>Resource perspectives</th>
<th>Data perspectives</th>
<th>Time perspectives</th>
<th>Workflow primitives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Petrinet (Classical) [82], [53]</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Petrinet High level [40], [46], [76]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>UML Activity diagram [24], [70], [22], [26]</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>UML Usecase diagram [66]</td>
<td>×</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>UML Sequence diagram [66]</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UML Statechart diagram [66]</td>
<td>√</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>EPC diagram [20], [85]</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
</tbody>
</table>

Table 2.1: Workflow Modeling Techniques
2.4.5 A Comparison

As discussed in previous section several types of models viz. finite state machine, petrinet, event processing chain etc are used for workflow specification. Analysis has been done in Table 2.1 considering all workflow modeling technique with respect to different perspectives. Apart from analysis, here in this section we have done a comparative study among the modeling techniques. State based models are popular for its structure and simulation purpose, but different formal models are being investigated for specification. Though formalism is often used in pursuit of absolute correctness and safety, it can guarantee neither. Formal technologies do not offer practical solutions for wide spread use in industry. While in case of FSM, intertask communication for multiple FSM is difficult to depict. Although event process chain have become widespread process modeling technique, they suffer from a serious drawback i.e neither syntax nor the semantic of an event process chain are well defined [20]. Though FSM as well as Petrinet are very effective to specify dynamic behavior of a system, still their uses become difficult as well as cumbersome to specify a complex system composed of several FSMs or Petrinets. It has been observed that activity diagram is widely used to model workflow diagrams because of its natural capability to describe sequence of actions and also for its visual appealing.

2.5 Workflow Analysis

Workflow management systems facilitate the everyday operation of business processes. In contrast to traditional information systems, they attempt to support frequent changes of the workflows at hand. Correctness, efficiency and effectiveness of business process supported by workflow management system are vital for organization. Therefore it is necessary to analyze a workflow process definition
before it is put into production. Basically there are three types of analysis.

- validation: testing whether the workflow behaves as expected.
- verification: establishing the correctness of a workflow
- performance analysis: evaluating the ability to meet requirements with respect to throughput times, service level and resource utilization.

**Workflow Verification**

To support the automation of business processes, it should be specified using a language called workflow specification language. The result of a workflow specification language is called workflow specification which describes various aspects of workflow. Building workflow specifications is a complex and error-prone process, especially for large-scale workflows. It is likely to introduce inconsistencies or errors in workflow specifications. Such inconsistencies or errors may lead to incorrect execution of some or all workflow cases. Therefore, workflow specifications should be verified to ensure that they are correct to implement the corresponding business process objective [45]. The workflow verification aims at establishing the correctness of workflow specifications. Verification issues like structural constraints, temporal constraints, resource constraints and authorization constraints can be investigated [27]. Some specific questions to be addressed in the context of workflow correctness includes

- How can it be determined whether a step in a workflow be successful?
- What is the effect of interleaving of states from different workflows?
- When one or more steps in a workflow fail, what happens to that workflow and other workflows those have accessed data produced by the failed workflow?
• What happens to a workflow when one or more of WFMS components fails?

In the following sub-sections verification of different modeling techniques are discussed in detail.

Verification of Petrinet Models

Given a petrinet \((P,T,F)\) and a state \(M\), we have the following notations \(M_1 \rightarrow M_2\) : transition \(t\) is enabled in state \(M_1\) and firing \(t\) in \(M_1\) results in state \(M_2\). \(M_1 \rightarrow M_2\) : there is a transition \(t\) such that \(M_1 \not\rightarrow M_2\). \(M_1 \rightarrow M_n\) the firing sequence \(= t_1 \ t_2 \ t_3 \ldots t_{n-1}\) leads from state \(M_1\) to state \(M_n\). Set of intermediate states are \(M_1, \ldots, M_{n-1}\), i.e \(M_1 \rightarrow M_2 \rightarrow M_3 \ldots \rightarrow M_n\).

- Reachability: In a Petrinet \(PN\) a state \(M_n\) is called reachable from \(M_1\) \((M_1 \rightarrow M_n)\) iff there is a firing sequence \(\sigma\) with an initial state \(M\). A state \(M'\) is a reachable state of \((PN,M)\) iff \(M \rightarrow M'\).

- Liveness: A Petrinet \((PN,M)\) is live iff for every reachable state \(M'\) and every transition \(t\) there is a state \(M''\) reachable from \(M'\) which enables \(t\).

- Boundedness: A Petrinet \((PN,M)\) is bounded iff for each place \(P\) there is a natural number \(n\) such that for every reachable state the number of tokens in \(P\) is less than \(n\).

- Strongly connected: A Petrinet is strongly connected iff for every pair of nodes (places, transitions) \(x\) and \(y\) there is a directed path from \(x\) to \(y\).

- Free choice: A Petrinet is a free choice Petrinet iff, for every two transitions \(t_1\) and \(t_2\), \(*t_1 \cap *t_2\not=\emptyset\) implies \(*t_1 = *t_2\) where \(*t\) denote set of input places for transition \(t\).

- Soundness: Procedure modeled by WF-net (A Petrinet which models the control flow dimensions of a workflow) \(PN = (P,T,F)\) is sound if and only if
for every state \( M \) reachable from \( i \), there exists a firing sequence leading from state \( M \) to state \( o \). Formally \( \forall_M \ (i \rightarrow M) \Rightarrow (M \rightarrow o) \)

State \( o \) is the only state reachable from state \( i \) with at least one token in place \( o \). Formally \( \forall_M \ (i \rightarrow M \land M \geq o) \Rightarrow (M = o) \)

There are no dead transitions in \((PN, i)\)

A class of petrinets so called BP net suitable for representation, validation and verification for the construction and analysis of business process in case of workflow management is discussed in [82]. In this case correctness of these petrinet representation, for example soundness property is verified using polynomial time. Whereas to completely specify a business process allocating resources to tasks has not been discussed in this paper. Analysis method based on Petrinet is discussed in [84]. Here petrinet is used to specify partial ordering of tasks (considering control-flow perspectives). It is possible in mapping task structure onto WF-nets which allows for verification of task structure using Petrinet based analysis technique [87]. Though workflow procedure (specifies set of tasks required to process cases successfully) is supported by WFMS, therefore to completely specify workflow process management of resources is also important [83]. Verification of process control aspects in conceptual workflow specification [2] and correctness verification of synchronization based workflow model [39] are presented. A spin model checker is used to check correctness of synchronization based workflow. Spin is popular open source software tool. Compared with current verification methods, semantic deduce based workflow verification method not only can detect deadlock and lack of synchronization, semantic errors, but also these are simple for computing [36]

**Verification of UML Models**

Certain properties can be discussed for the verification of UML whether certain states may or may not be reached. For instance will the terminal state be reached
at all circumstances can formally defined as

\[ \forall m \in R_N(m) : \overline{m} \in R_N(m) \]

where \( R_N(m) \) denotes the set of markings of N reachable from \( m \), and \( m \) and \( \overline{m} \) are the initial and final marking of N respectively.

- Absence of deadlocks may be verified by ensuring that

\[ \forall m \in R_N(m) : \exists t \in T_N : m^t \]

where \( m^t \) is the Petrinet notation to express that transition \( t \) is activated in \( m \).

- If the order is filled whether the respective goods be shipped eventually may be expressed as

\[ \forall m \in R_N(m) : \text{fill\_order} \implies \exists m' \in R_N(m) : \text{ship\_goods\_order} \]

These properties are structurally rather similar, so that translating intuitive property into petrinet terminology and interpreting their results for the UML activity diagram is quite easy. Formal semantics of activity diagram which is based on colored Petrinet and covers control flow, concurrency and data flow [78]. By formalizing UML activity diagrams using Finite State Process (FSP), a given UML activity diagram specification can be analyzed by checking its equivalent FSP description using LTSA model checker [74]. LTSA can provide a compositional analysis of concurrent aspects of business process in order to check safety and liveness properties. C-Wf model which is an extension of CIMOSA can be used to model an enterprise considering the concepts required in a workflow model [71]. C-Wf reference model defines elements required to support a resource allocation process as well as basic requirement for monitoring a business process execution and considers both temporal and synchronism aspects involved in the allocation process.
Verification for EPC Models

Verification issues for event driven process chain are as follows.

- **Regular:** An event driven process chain is regular if and only if
  - EPC has two special events $e_{\text{start}}$ and $e_{\text{final}}$. Event $e_{\text{start}}$ is a source node : $e_{\text{start}}^* = \varnothing$ and $e_{\text{final}}$ is a sink node : $e_{\text{final}}^* = \varnothing$
  - Every node $n \in \mathbb{N}$ is on a path from $e_{\text{start}}$ to $e_{\text{final}}$.

- **Soundness property:** A regular EPC is sound if and only if
  - process terminates properly for any case (i.e termination is guaranteed) for every state $M$ reachable from initial state (i.e where event $e_{\text{start}}$ is the only event that holds), there exists a firing sequence leading from state $M$ to final state (i.e the state where the event $e_{\text{final}}$ is the only event that holds).
  - there are no dead function (i.e for each function there is a firing sequence which executes f). There is no dangling references and deadlock and livelock are absent.
  - completeness preserving: Non-Petrinet based approaches have been proposed for the verification of informal modeling techniques. Most of the modeling techniques are graphic based. Using graph-reduction to reduce the complexity of verification problem in such a way that correctness is not violated by the reduction. If the EPC is correct before reduction then the result after the reduction is also correct and if the EPC is not correct before reduction then the result after reduction is not correct [91].

Although EPCs have become a wide spread modeling technique, they suffer form serious drawback i.e neither the syntax nor the semantics of EPCs are well defined.
The problem of EPC, i.e. formal semantics, can be tackled by mapping EPCs to Petrinets. Petrinets have formal semantics and abundance analysis techniques. As a result, the approach of mapping EPCs gives a formal semantics to EPCs. Correctness of EPCs can be checked in polynomial time by using petrinet-based analysis techniques [20]. Method for correctness verification of synchronization-based workflow models has been discussed in [39]. Based on WF-net soundness and relaxed soundness have been defined. There is no sound formal semantics for EPC which is fully compliant with the informal semantics [85].

2.6 Expected Features

Process-oriented WFMS applications are currently reliable and secure if and only if the business process is well structured and there is no need of ad-hoc deviation or dynamic extension at runtime. Changes in a business process arise due to three reasons [42].

- Process improvement: performing the same business process with increased efficiency
- Process innovation: performing the business process in a different way.
- Process adaptation: adapting the process for unforeseen changes e.g. handling a special case.

Reasons and requirements of workflow adaptations or ad-hoc derivation during runtime [95] [77].

1. Dynamic refinement: unavailability of a complete specification.

2. User involvement: decision making of user need to be considered.
3. Unpredictable events: due to some external factors, user intervention or timeout

4. Erroneous situations: system failure or erroneous operation.

Therefore changing a workflow definition is required due to some new requirements or exceptional situations [68]. There are two types of approaches emerged in this regard:

- Dynamic evolution: Dynamic evolution enables modification of workflow even during execution of workflow instances. Dynamic evolution copes with the problem of changing workflow definition due to new requirements or some exceptional situations. Changes can be evolutionary changes and ad-hoc changes. Evolutionary changes consists of structural modification that should be propagated to all new workflow instances as well as to the already existing ones. Ad-hoc changes can affect only a selected set of workflow instances. Changes may result from errors, rare situations or for a special requirement for a customer. Ad-hoc changes are typically supported by adaptive workflow system as a special case of evolutionary changes.

- Open point flexibility: Open point based flexibility is achieved by ensuring that there are a number of execution paths in the workflow definition and key decision point is well represented. It supports loose modeling of workflows, postponing complete specification to runtime.

Changes can be made either to a process or to an available resource or to a resource allocation mechanism. There are some policies which have been devised to manage workflow instance evolution [69] [42].

- Flush: Current workflow instances are allowed to complete according to the old process model and new instances can start following the new model.
• Migrate: Execution of workflow instances continues while the new changes are integrated into the process.

• Abort: All workflow instances of old schema are aborted.

• Adapt: The process must be altered for individual instances in order to accommodate some exceptional cases

• Build: The whole process can be rebuilt at runtime so that appropriate process model that correspond to the particular situations at hand can be created.

2.7 Healthcare Workflow

Healthcare is the prevention, treatment, management of illness and the preservation of mental and physical well-being through the services offered by medical, nursing, and allied health professions [37]. The organized provision of such services may constitute a healthcare system. Before the term "healthcare" became popular, English-speakers referred to medicine or to the health sector and spoke of the treatment and prevention of illness and disease. Patient visits various organizations or units within an organization to get proper diagnosis and treatment.

Workflow management in healthcare technology is not an easy task. The role of healthcare workflow management by use of IT is to adjust the contributions of organizations or units in terms of timing, quality and functionality [43]. Healthcare is characterized by close collaboration and sharing of information among actors i.e doctors, nurses, technicians and patients, who cooperate for patient care at different times being at different places. Supporting distributed and cooperative work and sharing secure patient information is possible among healthcare personnel from different remotely located sites, He@lthcoop - a web based system enables
gathering, storing and accessing all patient clinical and personnel data anytime from anywhere [52].

2.7.1 Features of Healthcare Workflow

In case of business process the objectives of modeling are to improve the efficiency of the organization, lower costs, reduce inventory, improve product delivery time, and to promote innovation. In healthcare industry we try not only to improve efficiency but also to improve safety of healthcare delivery. It is possible to achieve through the process of re-engineering workflow in healthcare industry. So can we literally translate workflow concepts used in business and manufacturing industry into healthcare industry? Healthcare is unique for the following reasons.

- In healthcare, a single unit is responsible for the total outcome of a patient. However during the care process number of requests are made to other areas of hospital to provide services such as laboratory tests, pathology tests, radiology tests, and specialist consultation etc.

- A common problem in healthcare is that a very few channel of communications are used for all information irrespective of urgency.

- One of the main reasons for modeling healthcare processes is to target points in the workflow for intervention.

- The clinical context of a process is the variable complexity of each patient. The variability between patients means that the same process may have different outcomes and there may be a need for various strategies to improve the health outcomes for each patient.

- Healthcare workflow that models a treatment activity can be modified at the level of the schema and it may also change during healthcare treatments.
depending on unpredictable events, for instance new data on patients which cause modification of clinical process.

2.7.2 Models

To implement the features of healthcare workflow, modeling should have

1. Business orientation: Healthcare workflow is not only process oriented i.e. how, but also resource oriented i.e. by whom.

2. Concurrency: One or more processes can be executed simultaneously. For ex: CBP study, X-Ray and MRI scan can be executed concurrently.

3. Temporal behavior: Timing is an important factor in healthcare process where a process has to be executed within a specific period of time in case of urgency.

4. Adaptive: We need a workflow model that supports evaluation as well as ad-hoc modification of process instances. Ad-hoc modifications allow users to adjust a particular process instance to specific circumstances so that this ad-hoc modification of process instance can be adapted.

5. Collaborative: The complexity of specialized knowledge and practices in the healthcare domain require active collaboration among different categories of healthcare providers such as nurses and medical specialists to facilitate coordination, information sharing and communication.

6. Scalability: In healthcare there may be many possible cases and corresponding therapies. Therefore modeling should be able to support high workloads.

7. Traceability: Refers to the ability to describe and follow the life of a conceptual or physical artifact on both forward and backward direction [94].
8. Activity crediting: In healthcare we faced with situations where more than one illness occurring simultaneously in a patient which require cooperation. Rather than patient repeat the same test for two different workflows, this can be done by merging two tasks into single task.

Modeling helps us to improve patient safety. Using product flow and information flow we can do the analysis regarding whether the needed parts arrive to the needed place at the prescribed time and how any change in the process affect the large system. Modeling of a HCU (Health Care Unit) contain four components [47].

- Static (such as clinical data)
- Dynamic (temporal/transactional aspects such as activities in a hospital when a patient arrives)
- Operational (organizational aspect such as medical staff structure)
- Managerial (planning and control aspects such as resource allocation)

In case of workflow and information flow if there is delay in transferring data then there is definitely potential delay for the treatment of patient. The negative impact of both patient and system has been examined as a whole [44]. Time management in workflow processes is crucial in determining and controlling the business activities. Integration of time management with workflow involves assignment of deadlines, synchronization of constraints, calculation of overall process duration and checking of timing inconsistencies. We need workflow representation language that is state based rather than event based to perform these studies.

Petrinet based Modeling

Petrinets became popular since they allow the graphic representation of computational structures and formal proofs of properties. Using a clinical healthcare
process case study [76] illustrates the model using extend WF-nets called Time WF-nets as shown in Figure 2.13. In this case sharing of resources at different times can be modeled. This model permits the modeling of shared resources. Each patient is competing for the limited resource (nurse) required for the task. Time constraint in this model ensures that patient 1 is to put in a request and this complete the procedure first before patient 2 does. Modeling is based on extended petrinet [18] where task corresponds to a transition and workflow state corresponds to a place as shown in Figure 2.14. Primitives are introduced to simplify visual representation in order to avoid more complex petrinet representation. There are two types of compound tasks: sub-workflow and aggregate has been discussed. Sub-workflow results in launching a new instance. Aggregate represents a group of tasks to form a higher level abstraction. Each place is represented by a circle. Each task is represented by a rectangle and sub-workflow is represented by a rectangle with double border. A set of operators for modifying an existing workflow schema has been discussed. In case of healthcare task ordering means sometimes a set of sequential tasks needs to be undertaken in a specific order. Other features encountered in healthcare workflow are the requirements to specify pre-requisite, co-requisite or post-requisite activity constraints and resource assignment also been discussed. For example the diagram illustrates how
to manage the blood pressure (BP) sub-workflow for patients with different BP readings. Petrinet representation have been chosen for clarity by not explicitly depicting intermediate states. In this case hypertension sub-workflow composed of number of sub-workflows, one of which progressive therapy is expanded and illustrated. Adaptability and sharing of resources has not been discussed in this paper. Operating room processes are modeled in terms petrinets so that soundness of a procedure can be checked using polynomial time algorithm [6]. This paper shows study of WF nets with shared resources (WFRS) can constitute a valuable source to improve sharing resource and to reduce the waiting period for patient.

Figure 2.14: Petrinets based Healthcare Workflow

**Formal Methods in Healthcare**

Though very few in number, there has been some work emphasizing the formal approach for engineering of healthcare workflow. Some of them we will review here to project the current trend of research. A very recent work [13] presents a method for conducting cancer research. Here model driven architecture for cancer research information system shows that several types of users like patients, para-medical staff, administrative staff and doctors can interact with the system for seamless operations. As this system requires interactions among several participants and eventually becomes fairly complex, researchers propose to take the
help of formal methods for precise specification. They have used Z as formal specification language to formally specify system architecture. It is realized that many hidden issues were not only revealed but also defined precisely due to uses of formal methods.

Description logic has been used in [17] to specify healthcare workflow and particularly to verify liveness and safety properties. Authors propose off-line verification of specifications so that runtime verification of system does not require increasing number of rules to apply. Runtime verification is necessary to manage safety conditions in case of any unseen situations arise during treatment of patients. Description logic knowledge base consists of TBox - Terminology knowledge base comprises of unary and binary predicates; and ABox - Assertion identifying individuals in TBox. Description logic is used to state the status of workflow while it’s under execution. Adaption to unforeseen situations during workflow execution is managed by a theorem prover by executing assertions that are triggered due to addition of new data to workflow. Authors also use temporal description logic to specify long lived workflow transactions and query on it. This is useful for healthcare as some treatment continues for a long period. Formal methods are used in managing medical protocol. While treating patients medical practitioners follow strict protocol as per medical guidelines. Adherence to medical guidelines is must for efficacy as well as safety of treatment. In case of serious and complicated ailments adherence to such protocol is cumbersome. Initially assuring correctness as well as completeness of such protocols are essential. Authors achieve it by integrating formal methods to lifecycle of guidelines. They have shown that model checking is useful to verify medical protocols. Thus, the results appeared in the paper confirms the trend of using formal methods in healthcare domain and has motivated us to carry out study on this aspect. However, there has not been enough research on formal methods for healthcare workflow. As
the requirements in healthcare automation increase, now there has been growing interest in application of formal methods in development of healthcare workflow systems.

**UML diagrams based modeling**

ATERUS model is used in healthcare structure to represent clinical and managerial activities [32]. This model uses hyper graph to describe the activities graphically and data of those activities is described by textual representation. In this case state diagram is used to control the activity. This conceptual representation become more complex while handling clinical complications which can be managed by exception handling. States which represent exceptions such as suspended, cancelled or aborted can be represented by state diagrams. In order to fulfill the requirement of healthcare workflow ATERUS model has the following features.

- **Abstraction**: Through a process of top down refinement this model is based on a description at different levels of abstraction of activities.

- **Activity interaction**: Concurrent, iteration, suspension or start etc are the primitives used in this model.

- **Delegation**: How to assign tasks to activities and what resources are necessary to execute activities.

- **User orientation**: User has a graphical representation of process.

- **Partial definition**: This model allows user to define part of the entire process in order to start the process execution.

- **Monitoring of process execution**: It helps in understanding whole process i,e what has happened and what may happen next.
In healthcare workflow management urgent request and critical messages has to be delivered timely [15]. Alert Management System (AMS) is proposed to handle this urgent constraint. Alert conceptual model that can capture both data and process integration represented using **UML** class diagram and workflow of the medical house call center is represented by **UML** Activity diagram.

The purpose of graphical representation of our model is that they are easier to communicate and therefore it is easy for people to understand in solving problem. Modeling the process and information flow between all of the stakeholders involved in the healthcare of a patient can be done using sequence diagram. From the workflow sequence model, it is possible to identify high-risk processes and therefore it can be most beneficial to reduce risk and improve patient safety. The workflow sequence diagram is derived from the Unified Modeling Language (UML), a set of diagrams and semantics used in software engineering. In complex system, sequence diagrams are one of the graphical representations developed for interaction of messages among entities. In sequence diagram entities in a process to be modeled are represented on the top of the diagram. For example in healthcare processes entities like unit, individual organization etc can be modeled. Constructs of Sequence Diagram for modeling are the following.

- Participating objects
- Depicting the sequencing of object operations

For the episode of care to be modeled, there are different active participants which can be represented using hollow rectangles. In workflow sequence diagrams solid arrows can be used to represent requests for information and dashed arrows for new information received. Gap between information request and information deliver can easily verify whether there is delay in information or information lost[48].
EPCs based Modeling

Business process modeling approach has been done with selected results from the analysis of Patient Care Process (PCP) in neuroscience ward of public hospital [31]. This analytic modeling technique is based on a software tool—the Architecture of Integrated Information System (ARIS). ARIS allows the presentation of PCP (in terms of object i.e. data, information source, staff member with particular skill, etc) linked into Event Driven process Chain (EPCs). Using ARIS concept, every dimensions (called view) of PCP is built into a business process model. These includes

- data view (patient record, pathology request and oral report)
- organizational view (categories of staff in a hospital involved in PCP)
- activity view (tasks performed by staff)
- output view (results from the performed tasks)
- process view

ARIS concept allows linking a chain of activities to a particular organizational goal. In ARIS framework there are no restrictions in adding diversified list of personal group specific goal if it was observed that a individual or a group behavior may affect PCP. ARIS model involves alternative chain of activities to achieve the same objectives, for example informal connection with staff from different organizational unit to improve the efficiency of operation.

2.7.3 Verification

Business process is modeled in terms of petrinets [6]. Soundness verification of the underlying business process has been done with the help of a case study workflow net with shared resources is used to analyze operating room processes and to
support re-engineering. E-commerce transaction executed by the WFMS, is represented using graphical model [41]. For calculating completion time and cost while executing a workflow, which is possible by providing modeling support for alternative paths in case of workflow. Scheduling algorithm is designed to meet user requirements on completion time and cost of execution. How to utilize workflow control structure augmented with information about timing behavior to calculate what delay to expect when execution of work item is postponed to certain dates. In general the issues to be verified in the healthcare model are as follows

- Soundness verification: means there is no deadlock or no unreachable tasks
- Correctness verification of synchronization based workflow
- Checking for both liveness and safety properties of workflow during runtime
- Temporal consistency checking
- Speeding up workflow instances by exploiting alternate paths
- Calculation of delay time when execution of a work item is postponed to certain date to execute a specific work item in case of emergency

A temporal interval as an execution duration is assigned to every workflow task. Though the real time taken by the task is unpredictable, so it may be between the specified bounds. Extending workflow nets (WF-nets) with time intervals and the new nets are known as Time WF-nets (TWF-nets). Properties of workflow modeled in TWF-nets can be verified in terms of safety, liveness and soundness using clinical healthcare process as a case study [76]. Presenting hospital cases as processes and then representing business processes has been done using workflow mapping to petrinet [6]. Possibilities of performance improvement of whole system is presented by introducing the balance between specialization and generalization, centralization and decentralization of resource classes while preserving soundness.
In this case Patient Workflow Management System (PWFMS) is considered to illustrate the safety of TWF nets. Each task is broken down into sequential sub-tasks where each task require a resource and a specific job description. Resource manager controls the allocation of resources. Consider two parallel processes of two patients awaiting for the same procedure where there is one nurse (resource) available. Adding time constraint to provide time safety which is analogous to boundedness of petrinet in such a way that Patient 1 to put in a request and complete the procedure before Patient 2 does. Hence PWFMS permits sharing of resources. Certain behavioral properties of workflow processes modeled in TWF-net can also be verified. Based on this duration constraints, algorithm to calculate the longest/shortest process instance in a workflow graph were formulated. Using verification algorithm temporal requirement and inconsistencies can be checked. But deadlock, liveness and fairness has not been analyzed.

2.8 Summary

In this section we have reviewed some important aspects of workflow modeling techniques. For the sake of completeness we have reviewed some major works on workflow modeling. Then we have reviewed workflow models for healthcare domain. Broadly, the survey has touched upon two different kinds of models viz. structural and formal. Various structural models viz. petrinet, UML diagrams used for modeling several aspects of workflows both in business as well as healthcare domains are reviewed in detail. Similarly, formal models based on descriptive logic, temporal logic, Z etc. proposed for modeling workflows in both business as well as healthcare domains are also reviewed. It has been observed that though formal method is well known for rigor due its mathematical preciseness its uses have been limited because of its unfamiliarity in user community for its shyness to
mathematical rigor. Users from different domains are not usually trained in mathematics required to write formal specification and to prove model correctness that needs expertise in discrete mathematics. Hence there is quest for a method that can retain utility of formal method and at the same time will ease its complexity for users for wide spread use. This has resulted in light weight formal method that is gaining momentum among researchers as well as users. RAISE [30] is a formal specification language designed for industrial uses. It advocates rigor in specification but downplays rigor in correctness proving with the help of axiom based study on system behavior. Elsewhere, we have made use of this method in development of mobile computing applications and found it satisfactory [1].

In an early work in this direction [25] the term light weight methods indicate that the methods can be used to perform partial analysis on partial specifications, without a commitment to developing and base lining complete, consistent formal specifications. Based on this observation, here we propose to develop a light weight formal specification approach to model workflow for healthcare domain particularly for automating treatment process. Following this line of approach for formal specification, here we propose to use predicate logic for specifying healthcare workflow. An early work [67] proposes the uses of predicate logic for specifying software systems. It has shown that a variant of predicate logic is able to state partial functions and can be useful to specify software systems. Taking a clue from it, we have proposed a method to specify healthcare workflow using pre and post conditions defined on work entities. In addition, the proposed method intends to provide means for flexibility in modifying a workflow, traceability, transparency and managing anomaly in healthcare workflow executions and assigning roles and responsibilities for actors associated with healthcare. The next chapter presents a technique for baseline modeling of healthcare workflow.