2.1 Distributed Systems

A distributed system is a collection of independent processes that appears as a single coherent system. Through message passing the components of the system communicate over multiple networks to achieve a common goal. Some of the significant issues in distributed systems are concurrency, lack of a global clock, and independent failure of components. For example, Internet is a network of networks of enormous size. It has an open ended global access to data, and service.

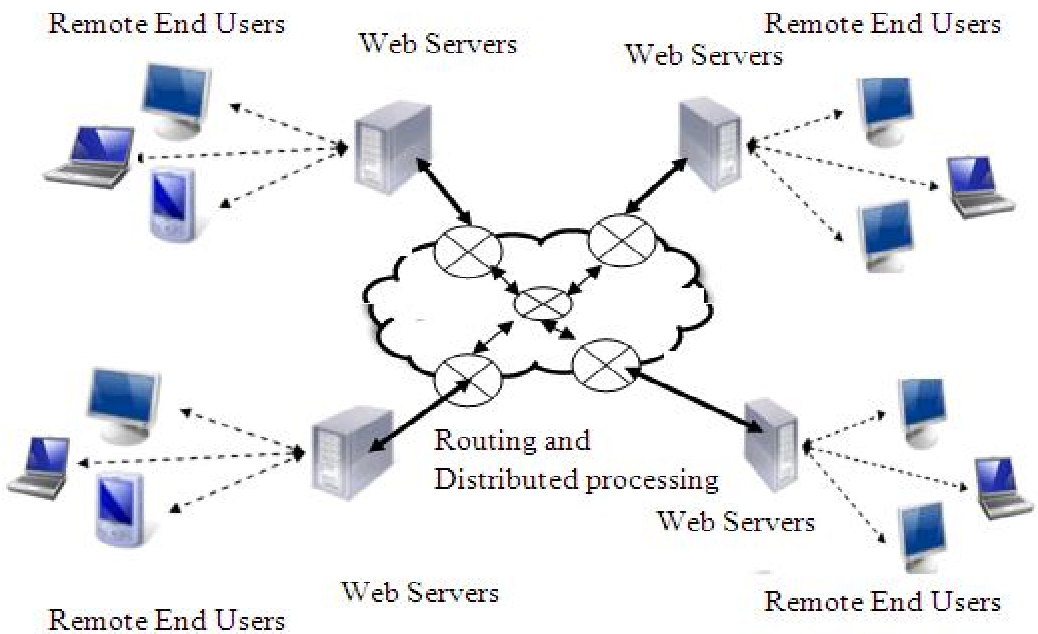

Figure 2.1 Distributed Processing

A distributed system of figure 2.1 above may try to update any data node and also wish to read the data from some other node. The possible results are like:

1. Second node may return the last version or some other version, which obviously do not guarantee any data consistency.

2. Since it is a distributed system there is a high chance of message failure and second node continue to wait. So it cannot respond to the queries, and hence violates availability of information.

3. During updating, if all the participants do not respond for various reasons within a specified period, the transaction can’t be committed and thus reliability of the system is in question.

Thus the key issues for distributed system demands robust transaction management protocols. It is a set of functions with clear inputs, outputs, beginning and an end. The key processes are goal oriented and sub-processes support the key processes through various services. It consists of multiple processing devices connected through some network protocol to interact with each other through communication and message passing. The business deployment model demands application platforms and services related to networking and communications [1, 2].

Irrespective of the application domain the challenges are to realize the efficient data from the association and or collaboration between multiple archives, common data flows, distribution of processing load, network management and data transparency. Transparency deals with access, location, migration, relocation, replication, concurrency, failure and persistence. Scalability is a requirement to mitigate the cost of increased complexity lower than benefit. The modeling and simulation tools are used to formalize the design process, checking the correctness of the model, describe the process flow and analyze the system. Research aim is to suggest an efficient and robust framework for modeling distributed systems. It would help towards proper analysis and design for various applications.

2.2 Understanding of Business Process in Distributed System

A process can be defined as an ordering of work activities across time, space and boundaries with clear inputs, outputs, beginning and an end. A business process is a group of related activities to achieve or fabricate either some service or a product having some value to the user or to the customer. Business process improvement can be emerged with operational highlights, exception reports, and are consciously used for day-to-day or periodical decision-making [3, 4].

A distributed system consists of multiple processing devices connected through some network protocol to interact with each other for communication and message passing. The nature of the business process sometimes demands concurrent processing. If all the participating distributed processors have access to a shared memory to exchange information among the processors the concurrent business process can be defined as a tightly coupled distributed computing system. Whereas if each processor has its own distributed private memory and information is exchanged by passing messages between the processors are called loosely coupled distributed computing system. However, to the user the concept of distributed operations is an UMA or NUMA abstracted to primary objectives of trouble free communication [5].

The complexity of the business transaction in a distributed platform is directly proportional to the number of operational locations. The cyclomatic complexity will increase with the increase in sites, time complexity for distributed query processing will also increase and the chances of resiliencies will also increase. Message passing operations can be used to construct protocols to support particular process role and communication pattern. Designing of a communication protocol for a distributed transaction processing system is considered to be one of the most critical tasks [6, 7].

Interoperability among the multiple archives is one of the key issues in a distributed system. Some of the available architectures that are used for collaboration and integration of distributed legacy systems are CORBA, J2EE, XML, SOAP, DCOM, RMI, and EAI. CORBA allows applications to communicate with one another efficiently and the XML is used to process data on the WEB. DCOM is a protocol that enables components of the architecture to communicate directly over a Network but there is not yet an all-inclusive proposal for distributed access control mechanism and a robust architecture for transaction processing [8, 9]. This research aims towards addressing efficient data handling mechanism on a distributed environment using high level net models.

2.3 Modelling and Simulation

The word "Model" is derived from the Latin word modulus implies pattern, rule, and measure. There can be many abstractions of single phenomena to capture knowledge and show the ways for conceptual analysis. Often a model can observe some critical situations in advance, which the real system cannot highlight till it happens. For example finite state machine is a model of the computer system. The formulas used in physics, chemistry are formal mathematical representations of real instances. The primary objective of the descriptive model is to imitate the

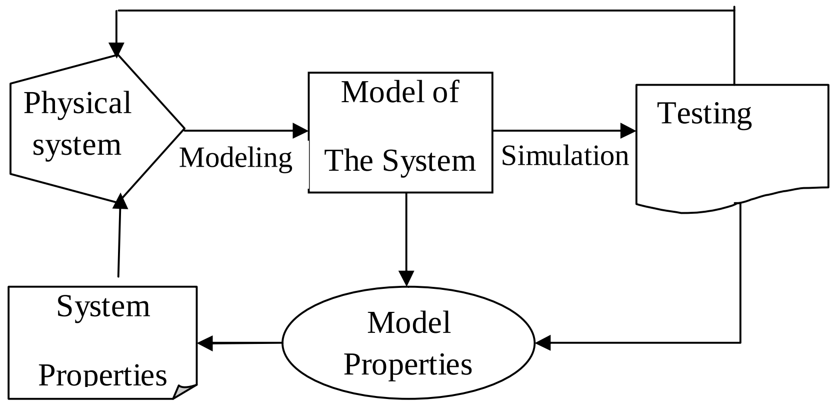

Figure 2.2 Modeling and Simulation

appearance of the originals and perspective models are particular evaluation of a solution or facts that can be assessed and measured. In software engineering, there are process models, design patterns, and class diagrams.

Simulation is used to express the dynamic behavior of the system. It is symbolical or numerical. Modeling and Simulations are abstractions of reality. The aim of modeling is to provide impending knowledge of reality, whereas simulation targets for accuracy, execution speed, and iterative nature of the application. Most of the times modeling and simulation purposely emphasize one part of reality at the expense of other parts. It is created from a heap of data, equations and computations that mimic the actions of things represented. Figure 2.2 represents a self explanatory abstracts of modeling and simulation composed of the objects like model of the physical system, system properties, testing, and model properties connected through directed arrows.

A complex model use to simulate the actions and reactions of the real thing and the models behave the way they would in real life. Big projects use computers for timely analysis of real time data using the methods like discrete-event simulation, continuous simulation, Monte Carlo simulation etc. In this thesis modeling and simulation activity of a few real life applications from the sphere of BPR, realm of VDW and from SCM domain is presented. These models devises and demonstrated various static and dynamic instances over a distributed domain to answer, whether the model can exhibit predictive validity, is it capable of reproducing the used data, and capable of predicting new behavior [10, 11].

2.4 Petri Net and its recent Derivatives

PN is used to model the static composition and the dynamic actions of a system. Information processing of a concurrent, asynchronous, distributed, parallel, or stochastic system can be modeled using PN. A PN model can be expressed by a set of linear algebraic equations, or in some other mathematical form to reflect the activities of the system. The static structure and the dynamic behavior of the system can be represented graphically as a bipartite directed graph [12].
As for example, let us consider a primitive conveyor belt system in figure 2.3 used in any process industry with three places: P1–initial, P2–current and P3 – break down and four transitions t1, t2, t3, and t4 describes as follows: t1 – task starts, t2 – task complete, t3- task down, t4- task maintenance.

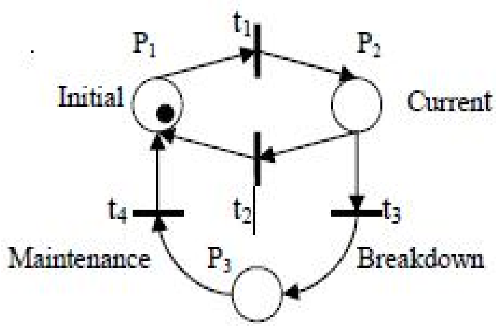

Figure 2.3 Conveyor belt in operation

The markings at the respective locations are:

	P_1	P_2	P_3
Initial (Idle)	[1	0	$0]^T$
Current (working)	[0	1	$0]^T$
Break Down	[0	0	$1]^T$

P-invariants indicate token conservation and T-invariants represents system stability and steady state of the system.

P-Invariants

P_1	P_2	P_3
1	1	1

T-Invariants

T_1	T_2	T_3	T_4
1	1	0	0
1	0	1	1

The net is covered by positive P-invariants, therefore it is bounded and the P-invariant equation may be represented as M (P1) + M (P2) + M (P3) =1, Where, M is the marking at the respective place. The net is covered by positive T-invariants; therefore the net is bounded and live.

Some of the useful behavioral properties are like precedence relations among the events, concurrent operations, synchronization issues, freedom from deadlock, iterations. and mutual exclusion of share resources can be analyzed using PN. It also opens up the possibilities for formal analysis of the system. It is a natural way of performing logical interactions of the activities among the components of a system. The theory originated from the doctoral thesis of C. A. Petri in 1962 [19]. Since then PN is being used in many theoretical as well as applications for unfolding the system. Depending on the system requirements there could be variations of the PN type. Some of such useful variations are marked Petri nets, Place-Transition nets [12], Time Petri net (TPN), timed place Petri nets (TPPN) [13, 14, 15], Colored Petri nets (CPN) [16], generalized stochastic Petri nets (GSPN) [17] and a combination of the types.

Ordinary Petri nets (PN): It was developed by Carl Adam Petri. A PN is a directed bipartite graph with places, transitions and directed arcs. It is represented as four Tuple N = (P, T, D^-, D^+), where P is the set of places, T the set of transitions. D^- and D^+ are the input and the output matrices, where $D^- = (P \times T) \neq 0$, $D^+ = (T \times P) \neq 0$, $D = [D^+ - D^-]$. Transitions are allowed to fire if all the preconditions are fulfilled. The time variables associated to the PN can be either deterministic type or random variables of Stochastic type. Timed PN was first proposed by Ramchandani [18] to perform timing analysis of asynchronous concurrent systems.

- Time PN (TPN): Use a time pair instead of a single delay. A time domain is used and observes a strong firing mode.
- Timed PN: Timed Petri nets which treat a timing constraint as a single delay. It follows a strong firing mode.
- Timed place Petri Nets (TPPN): In TPPN every place is assigned a deterministic time interval before enabling the transition,
- Stochastic Petri Net (SPN): Stochastic nets use the average delay, which is a probability function of a transition's firing rate. In case, several transitions in an SPN are simultaneously enabled, the transition that has the shortest delay will fire first.

- Generalized Stochastic Petri Net (GSPN): Extension of SPN having both timed and immediate transitions, timed transition have an exponentially distributed firing rate, an immediate transition has no firing delay. Competing ttransitions which fires first wins the conflict.

2.5 Formal analysis using High Level Nets

Low level Petri nets are simple and useful for modeling control flows but not powerful enough to define data and system functionality for every instances. Modeling real time application systems using PN some more additional constructs are necessary to model the representation compact with enhanced modeling formalism. Use of all such extensions depends on the easiness, and convenience of the implementation. So far introduction of time and uncertainties of occurring an event is mentioned in section 2.4. Now, about individual identification of tokens a new class of formalism is proposed in the form of coloured PN introduced by Jensen [16] and the technique of assigning to each token a predicate i.e. Predicate/Transition net (Pr/T-nets) introduced by Genrich and Lautenbach [20]. The common characteristic of these extensions are collectively referred to as HLPN. More precisely A high-level Petri net is a Petri net extended with 'time', 'colour', and 'hierarchy'. It supports data and functionality definitions, such as using complex structured data as tokens and algebraic expressions as transition formulas. In comparison to low level PNs, HLPNs produce compact system models that are simple to comprehend, and are more useful in modeling complex systems. Methods of analysis for PN may be classified into three groups:

1) The coverability (reachability) tree method

2) The matrix-equation approach, and

3) Reduction or decomposition techniques.

The first method involves essentially the enumeration of all reachable markings or their coverable markings. It is applicable to all classes of nets, but is limited to "small" nets due to the complexity of the state-space explosion. On the other hand, matrix equations and reduction techniques are powerful but in many cases they are applicable only to special subclasses of PN or special situations. In subsequent sections the formal analysis will be discussed in detail.

2.6 **Review on modeling using High Level Nets**

Despite the fact that information system workflow suffers from standard modeling techniques. PN and its derivatives among the other techniques are used for both qualitative and quantitative analysis of workflow systems. It has a power over a number of specific functional properties of the system under design. Two types of properties that can be differentiated are behavioral and structural. The behavioral properties depend on the initial state or marking of a PN, whereas the structural properties depend on the topology of a PN.

A Structural model is a directed graph representing the static part of the system. There are two kinds of nodes, places and transitions, represented by circles and rectangles. Places represent state variables and transitions represent transformers. The net is said to be ordinary when all arc weights are equal to one, i.e. each occurrence of adjacent transitions consumes one token from input to output place. Behavioral Models capture the dynamics of the system behavior using evolution rules for the marking. Markings are represented by tokens. The token at a place is its state value. The values get changed to adjacent states with the occurrence of transitions [18].

An ordinary PN cannot handle services like simultaneous arrival of tokens to a queue. In such concurrent situations an extension of the ordinary PN is needed. A transition will be enabled when its pre-conditions hold, but not post-conditions. In order to tackle this and many other aspects, like temporal issues, stochastic analysis, a reduction of state-space explosion for large models GSPN was developed [12].

The need for timing variables in the models of various types of dynamic systems is apparent since these systems are real time in nature. A Timed Petri Net has been devised to serve the purpose. The firing rules defined for a Timed PN control the process of moving the tokens around. Time Petri Net model (TPN) is another powerful formalism and conciliation between modeling power and verification complexity [13]. The timed nets are classified as timed place Petri Nets (TPPN) and timed transition Petri Nets (TTPN), depending on whether the timing bounds annotate places or transitions [15].

A Colored Petri Net (CPN) obeys the same firing rules as that of the basic PN with the exception of a functional dependency between the color of the transition and the colors of the tokens [16]. Each token in a CPN is attached with a color to identify the token. Each place and transition is attached to a set of colors. A transition will fire with respect to each of its colors.

Stochastic timed Petri nets (STPNs) are Petri nets in which stochastic firing times are associated with transitions that automatically generate the stochastic process which governs the system's behavior. A GSPN with initial markings can be described uniquely by a 6-tuple: GSPN (P, T, I, O, M, λ), where, (P, T, I, O, M) is a marked PN. GSPN supports both immediate and timed transitions [13]. Here, P is the set of Places and T is the set of Transitions connecting the places in T. The sets P and T are mutually exclusive. I is the Input function, where the value I (p, t) is the number of directed arcs from the place p to the transition t. O is the output function, where the value O (t, p) is the number of arcs from the transition t to the place p. M is the Initial marking of places, where the value M(p) is the number of tokens that are located in the place p. The marking dependent firing rates associated with transitions are represented as λ = (λ1, λ2, λ3, …., λn). Firing delay is the elapse time associated with every transition. This delay is a random variable with a negative exponential probability density function. For any marking dependent transition with an associated firing rate λi, can be expressed as λi(mi) and the average firing delay of transition ti in marking mi is [λi(mi)]-1 [21]. The non-definitive nature of the large retail networks is perfectly suitable for modeling such systems using GSPN.

2.7 Business process modeling for Distributed System

Business process modeling enables a common understanding and analysis of a business process. It is a permutation of a set of activities within an enterprise with a structure describing their logical order and dependence to produce a desired result. A process model should provide a comprehensive understanding of a process. An enterprise can be analyzed and integrated through its business processes. Hence there is an importance of verification and validation of the modeling techniques and tools used for analysis and, obtain knowledge of the available process.

Further, when the system is distributed in nature, the operational data use to pass over several physical nodes and synchronization of transactions is critical and communication among them takes place through message passing [5]. Time complexities and the chances of resiliencies depend on the number of operating sites. The fault tolerance of the system depends on how the system continues to operate, when one or more components of the system fail. The goal of distributed transaction processing remains transparent with shorter response time and higher throughput [23]. Some robust protocol is needed to take care of firm happening, resource sharing, and message passing for distributed transactions.

2.8 Gap Analysis and Scope of Work

This research has studied in detail Data and Process Modeling for Distributed Systems using PN and its derivatives with an aim of formal representation of the actual happenings. The work plan is presented in figure 2.4. As a stepping stone 2PC modeling is attempted to study the concurrency control mechanism. As an example to supplement the 2PC phenomena a concurrent Database update problem is discussed here. This provides a framework for distributed transaction management. In order to study the impacts of formal modeling the work process is shifted in the back-drop of a real life scenario in the distributed domain. Some of the applications chosen are like product ordering system, VDW, and remote health care Support System from BPR domain.

Similarly from the SCM domain Beer Game, BWE modeling, and Demand forecast variance analysis are studied. In each case the main objective is to analyze the gap and improve the business processes.

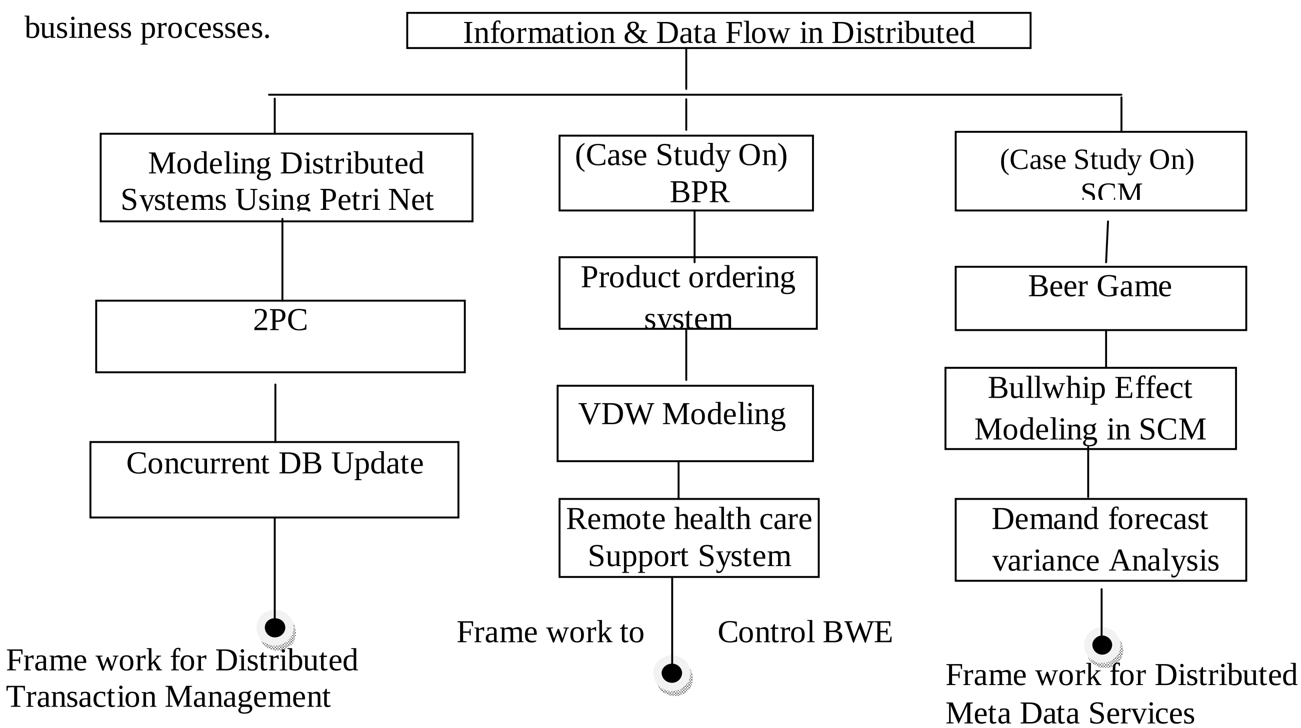

Figure 2.4 Work Plan and Gaps

Study of modeling activity of real life applications from the BPR domain lifts the revision process of distributed TPS to make it faster and robust. An adaptable service-based framework for distributed product realization is studied in [9]. A multi-agent framework for distributed decision making systems is studied in [25]. A Jinni based self forming, self managed and self-healing system free from system configuration and system management is studied for distributed communications [26]. Some middle wares are available for processing of distributed web based applications [9]. Cost effective asynchronous business deployment models for handling of distributed transactions are premeditated [1]. The use of VDW to support decision making functions is studied in [29]. Technological advances in distributed health care management use sensor devices, broad band services, wireless internetworking, and automation of patient data is learnt [30, 31, 32]. Social network applications are also obtainable on medical health services in the literature [33, 34]. Despite the success stories of collaborative metadata services the available metadata on health services should be sharable and updatable for the benefit of the mankind is reviewed [31,35]. All these applications are capable of managing distributed transactions in a limited way. Either they lack formalism for rigorous analysis or workflows are inadequate to demonstrate each stage of the process. In this work a frame work is suggested to control the BWE.

Then the focus of the study shifted to SCM domain, where there is a huge volume of data to study the impact of demand variability for distributing value to customers, consumers, end-users, suppliers and third parties to fulfill the customer requirement [37]. Realization of multifaceted relationships in a SCM is crucial for its success [38, 39]. Contemporary literatures are indicating the lack of measurement aspects of SCM [40]. The behavioral pattern of the SCM is difficult to change in descriptive SCM, but use of BI tools can take care of the limitations [41,42]. Investigative SCM quantify the occurrence of BWE using statistical techniques [43, 44, 12]. Industrial systems primarily focusing on the supply side of the supply chain, typically Business to Business transactions [45]. Some of the key characteristics of monitoring the demand variability are: risk sharing, Outsourcing, mobility, and collaborations [46, 47, 48]. ERP systems are identified as dynamic systems used and monitored for SCM integration [9, 45, 49]. Once again these systems handle huge volume of distributed transactions but lacks formalism to demonstrate functional activities. A frame work is suggested for distributed meta data services.