Application of CTPN Model of Distributed Active Database in Ambient Intelligent Pervasive Computing Environment

Bulbul Mukherjee, Souvik Sengupta, and Ranjan Dasgupta

***Abstract*—** This work tries to explain the relevancy of distributed active database in some real life complex networked transport system and medical science systems which also involve ambient intelligence, pervasive computing and perceptive environments. These environments are places with the ability of perceiving the context through sensors and other mechanisms. The core of the system relies on a set of modular agents equipped with rules. Those rules are composed of triggers, conditions and actions that enable them to express desired behaviors of the environment. One generic model of distributed active database is done by color time Petri Net which is appropriate in several real life applications and helpful to make understand the complex concurrent activities involved in these applications.

***Keywords*—** Ambient Intelligence, Color Time Petri Net, Distributed Active Database, ECA Rule, Pervasive Computing

I. INTRODUCTION

Petri Net[1][2][3] are one of the very useful graphical and mathematical representation and are being used to model complex and concurrent systems. Several variations of Petri Net had already been proposed which can handle more and more complex problems. These modifications include introduction of color(data),time etc. in the Petri Net.

Active database systems(ADBMS)[5][22] support mechanisms that enable them to respond automatically to events that are taking place either inside or outside the database system itself. The aim of an active database is to perform automatic monitoring of conditions defined over the database state and the ability to take action when the state of the underlying database changes. Active Heterogeneous Database Systems have arisen from the demand for automatically monitoring the database states of participated database system and triggers appropriate production rules in a timely manner when situations of interest occur. They thus represent the confluence of techniques from several diverse areas, such as multiple database systems, active database systems, and artificial intelligent (AI). Event-driven reactive functionalities are urgently needed in present-day distributed systems and dynamic Web-based environments and that emerges the need of modeling a distributed active database[6] and which is done by Color Time Petri Net (CTPN)[4] in this paper. The CTPN is very much applicable over here as the basic operations of distributed active database deal with various data elements(value) and based on these values, transitions are taking place. In case of some transitions, waiting time also plays a critical role and hence time color Petri Net becomes a natural choice.

In this paper we will also discuss some real life examples of ambient intelligence[16][27] and pervasive computing [17][18][19] which have certain relation with Distributed Active Database and will show the appropriateness of the above mentioned model in these applications.

Bulbul Mukherjee is with the Dept. of Computer Science and Engineering, Bengal Institute of Technology, Kolkata, West Bengal, India, (e-mail: mebulbulmukherjee@gmail.com).

Souvik Sengupta is with the Dept. of Computer Science and Engineering, Bengal Institute of Technology, Kolkata, West Bengal, India, (e-mail: mesouvik@hotmail.com.

Ranjan Dasgupta is with the Dept. of Computer Science and Engineering, National Institute of Technical Teachers' Training & Research, Kolkata, India, (e-mail: ranjandasgupta@ieee.org).

II. PETRI NET

Concurrency can pose a problem when many entities (people, machines, processing threads) use (share) the same resource (or a limited number of resources). A Petri net, introduced by C. A Petri in 1962 is formally defined as a 5-tuple N = (P, T, I, O, M0), where

(1) $P = \{p1, p2, \ldots, pm\}$ is a finite set of places;

(2) $T = \{t1, t2, \ldots, tn\}$ is a finite set of transitions, $P\ T \neq \emptyset$, and $P\ T = \emptyset$;

(3) $I: P \times T \rightarrow N$ is an input function that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

(4) $O: T \times P \rightarrow N$ is an output function that defines directed arcs from transitions to places; and

(5) $M0: P \rightarrow N$ is the initial marking.

A marking in a Petri net is an assignment of tokens to the places of a Petri net. Tokens reside in the places of a Petri net. The number and position of tokens may change during the execution of a Petri net. The tokens are used to define the execution of a Petri net.

The type of Petri nets described up to this point is called Place/Transition nets. They have a number of limitations:

a. Inability to model similar (but not identical) processes using one net

b. All tokens are identical
c. No way to represent additional properties – there is no way to associate any additional data with token.

In order to overcome these problems, a number of solutions extending the initial approach i.e high level Petri Nets have been proposed. Colored Petri Nets(CPN)[4][21], introduced by Kurt Jensen in (Jensen 1981) is a discrete-event modeling language combining Petri nets with the functional programming language Standard ML[4]. The state of the modeled system is represented by the places. Each place can be marked with one or more tokens, and each token has a data value attached to it. This data value is called the token color. It is the number of tokens and the token colors on the individual places which together represent the state of the system. This is called a marking of the CPN model, while the tokens on a specific place constitute the marking of that place. By convention, we write the names of the places inside the ellipses. The set of possible token colors is specified by means of a type (as known from programming languages), and it is called the color set of the place.

By firing a transition, tokens are removed from the input places and added to the output places in the same way as that in original Petri nets, except that a functional dependency is specified between the color of the transition firing and the colors of the involved tokens. The color attached to a token may be changed by a transition firing and it often represents a complex data-value.

The Color Time Petri Net (CTPN) or timed CPN extends the framework of the original PN by adding color and time attributes to the net. The time attribute allows various time-based performance measures to be conducted in the system model. A time-delay can be assigned to either places or transitions to model the time elements in a system.

The main difference between timed and un-timed CPN models is that the tokens in a timed CPN model—in addition to the token color—can carry a second value called a *time stamp*. This means that the marking of a place where the tokens carry a time stamp is now a *timed multi-set* specifying the elements in the multi-set together with their number of appearances and their time stamps. Furthermore, the CPN model has a *global clock* representing *model time*. The distribution of tokens on the places together with their time stamps and the value of the global clock is called a *timed marking*. In general, a time stamp can be a non-negative integer or real. In the current implementation of CPN Tools, only non-negative integers are supported. The time stamp tells us the time at which the token is *ready* to be used, i.e., the time at which it can be removed from the place by an occurring transition. The tokens on a place will carry a time stamp if the color set of the place is timed. A color set is declared to be timed using the CPN ML keyword timed[3].

III. ACTIVE DATABASE

Active databases[5][22] are able to monitor and react to specific circumstances according to its relevancy to an application.. An active database system must provide a knowledge model (i.e., a description mechanism) and an execution model (i.e., a runtime strategy) for supporting this reactive behavior.

A common approach for the knowledge model uses rules that have up to three components: an event, a condition, and an action[5]. The *event* part of a rule describes a happening to which the rule may be able to respond. The *condition* part of the rule examines the context in which the event has taken place. The *action* describes the task to be carried out by the rule if the relevant event has taken place and the condition has evaluated to true.

Briefly, events can be classified into: i) primitive events and ii) composite events. Primitive events refer to elementary occurrences which are pre-defined in the system. Primitive events can be further decomposed into database events, time events, transaction events, method events etc. A composite event is a set of primitive events or composite events related by defined event operators.

A. Events in Distributed Active Database

Four types of events can be identified in a distributed active database system[6].Local Primitive Events are events that are predefined in that application using primitive event expressions and can be detected by a mechanism embedded in the system. Local composite events are composed of local primitive events and other local composite events by applying event operators. Global primitive events are events that are defined and detected outside of the current application but are referenced/used by the current application in a distributed environment. Global composite events are related to event occurrences from many sites (including the local site). They are constructed with local primitive events, local composite events, global primitive events and other global composite events.

IV. RELATED WORK

Active database capability most frequently uses triggers that execute actions based on the Event-Condition-Action (ECA) Model. Several event algebras have been developed, e.g. Snoop[7], SAMOS)[8], ODE [9]. SAMOS combines active and object-oriented features in a single framework using colored Petri nets. Snoop is an event specification language which defines different restriction policies that can be applied to the operators of the algebra. The Global Event Detector (GED) (Chakravarthy and Liao, 2001) detects events in a distributed environment through a client/server architecture that minimizes message communication and allows for event registration and notification. GED extends the active capabilities of the Sentinel[10] active OODBMS.

In SAMOS, a very complex structure is used to represent a sequence of composite event. This will result in an extremely huge CPN model, which will be inconvenient for large rule-base development, and will be difficult to be implemented and managed. A software platform has been developed, which can generate a Conditional Color Petri Net(CCPN) model[11] automatically from a text file of ECA rule description, and communicate with a traditional database system when an event is detected from the database or an action command is generated by the CCPN simulator. This CCPN model will overcome the disadvantage of using a redundant structure to specify a temporal relation between primitive events since this information can be considered as a condition on transitions.

Baba-hamed.L and Belbachir.H (2005)[12] propose a method of termination analysis of active rules based on Petri Nets (PN) called as Extended Coloured Petri Net (ECPN) and give an object oriented representation to implement it. Latifa Baba-Hamed (2008) has done a comparative study of the above method with the most known methods available in the literature for detecting non-termination. Li and Marin presented an approach based on colored Petrinets named CCPN for modeling the active database behaviour. Incidence matrix of PN theory is used to find cyclic paths existing in CCPN[13]. Cycles which satisfy some theorems given by the authors are deleted. If there is no cycle in the CCPN, the termination of the corresponding set of rules is guaranteed. Nevertheless, this approach did not consider the priority of rules. An enhanced ECA rule, called ECA-AA is defined as a four tuple (*Event, Condition, Action, Alternative Action*) as ECA Mechanism for Distributed Active Database Systems with High Autonomy Degree is done in the research paper[14]. The ultimate goal of this enhancement is a more flexible ECA mechanism, which allows us to continue work even if subsystems are not reachable.

V. Scope of the Work

Conventional database management systems are passive. Data are created, retrieved, modified, and deleted only in response to operations issued by users. By contrast, context-aware computing[15] is required to automatically carry out some services in response to certain changes in the real world, once conditions are being satisfied. Therefore, it needs some facets that active database systems have. The basic idea behind another emerging concept Ambient Intelligence (AmI)[16][27] is that by enriching an environment with technology (e.g., sensors and devices interconnected through a network), a system can be built such that acts as an ``electronic butler'', which senses features of the users and their environment, then reasons about the accumulated data, and finally selects actions to take that will benefit the users in the environment.

Ambient intelligence (AmI) is a new research area builds upon advances in sensors and sensor networks, pervasive computing, and artificial intelligence and is for distributed, non-intrusive, and intelligent software systems. Modern large-scale applications, such as e-commerce, Internet or Intranet applications, enterprise application integration (EAI), and emerging pervasive systems[23], can effectively benefit from an active mechanism but conventional active mechanisms have been designed for centralized systems, so it must be adapted to meet the requirements imposed by this generation of large-scale heterogeneous applications. Thus modeling the overall functionality of a distributed active database is very much significant.

Pervasive computing is characterized by a high degree of heterogeneity, mobility and support of distributed computing architectures which works on online connections. However, remote systems may be inaccessible occasionally. In case of active database systems this often leads to undecidable ECA conditions and rules. In distributed active databases, both event evaluation and condition evaluation can have an indefinite result because of unavailable subsystems. This leads to an abort after a timeout even if this is not desirable or necessary at all. By enhancing ECA rules with additional actions for the case of undecidability of ECA conditions, it becomes possible for active databases to react alternatively, which makes the entire ECA mechanism more robust. Our proposed model by CTPN can help to design and implement the distributed active database functionality which is very much relevant in today's emerging technologies of AmI.

VI. Proposed CTPN Based Model of Distributed Active Database

We propose the model of the distributed active database which has the client server architecture supported with decentralized event detection and decentralized rule base.

- Under the client more than one application can run at a time.
- There will be provision of color set to identify a specific application, a specific client, event identification and the type of event (whether it is local or global and primitive or composite).
- The server can send event notification to clients by detecting any external event also which happening outside the database itself.
- Each application needs to reveal what events it is interested in from other applications. Then that application of that client can send requests to the server, the server can send response if the requested event is already detected, otherwise it can send requirements to the local service, process it and sends the necessary information back to the client.
- After completion of the event detection the rule processing is done by the client to trigger a specific rule and the condition evaluation process is occurred.
- Similar to the events, a condition may also be composed of different sub conditions which may be evaluated either in local site or globally through the server.
- Due to some reasons like System site failures or Network failure or Low network bandwidth etc. distributed event detection or condition evaluation can fail in one remote system and for that the total procedure will be affected or delayed.
- After completion of each event detection or action the application of the client will send message to the server.

A. *Temporal Constraints:*

In this model a deadline is there for each event detection process i.e maximum allowable time to detect an event and maximum allowable time to evaluate one condition.

We assume,

TE represents the predefined maximum time length for completion of event detection of an composite event.

TC represents the predefined maximum time length for completion of evaluation of condition often composed of several sub conditions.

If te_j represents the time required for one sub event and tc_j represents the time required for evaluation of one sub condition then the following constraints are must:

$$\max_{j=1 \text{ to } n}(te_j) <= TE \quad \ldots\ldots\ldots(1)$$

and

$$\max_{j=1 \text{ to } n}(tc_j) <= TC \quad \ldots\ldots\ldots(2)$$

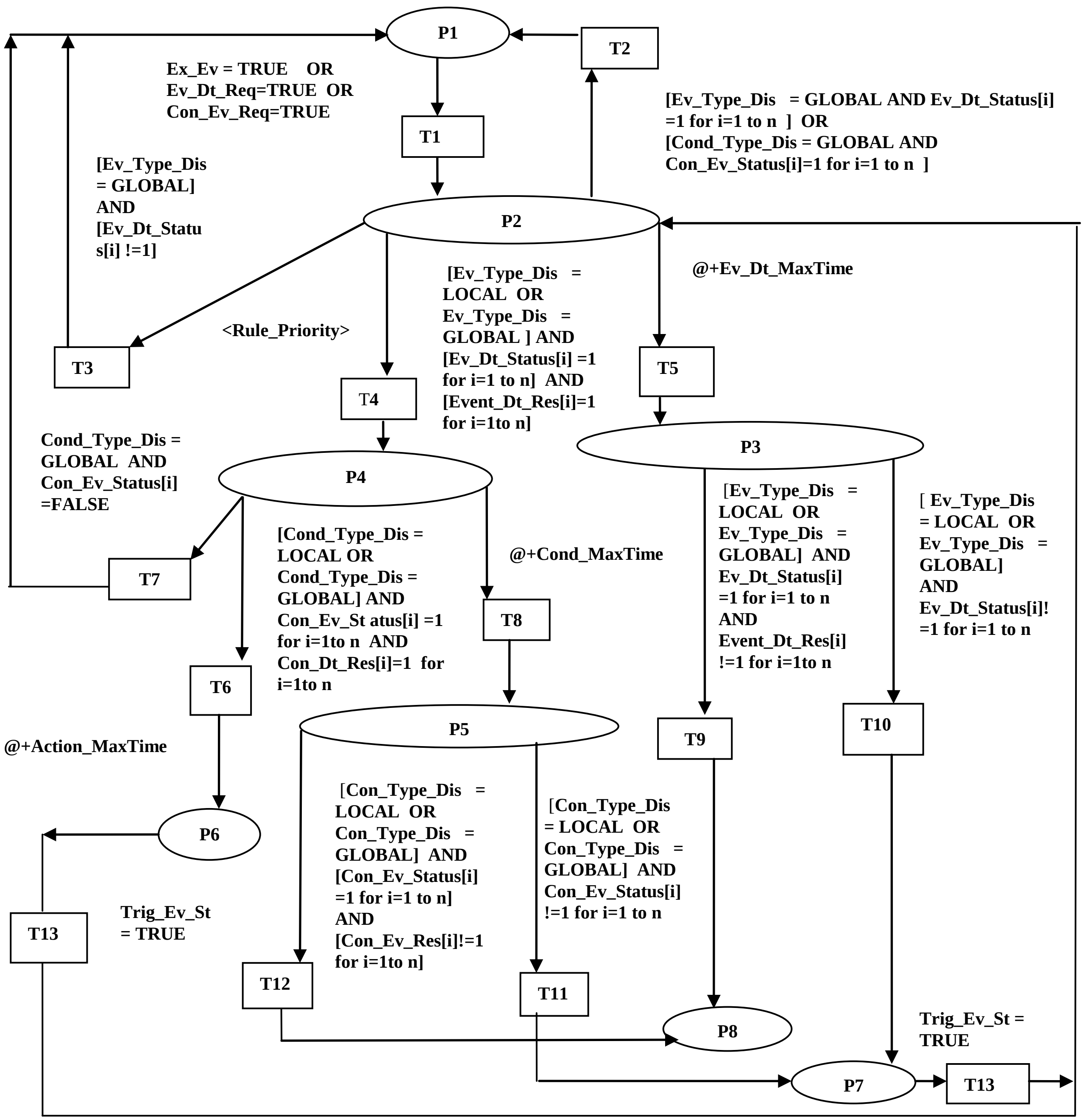


Fig 1 CTPN model of distributed active database

In addition to this to meet the requirement of the real time distributed active database the action is also be completed within a certain time limit, e.g if temperature of a boiler rises above 180 degree centigrade then reduce the pressure within 45 seconds. So one time constraint Action_MaxTime is given as a constraint in the transition T6 in the model.

The process can follow either a predictive or a reactive strategy. Predictive adaptation is possible if the valid time of a control action is specified by a fixed (absolute) date or interval. Reactive adaptation is used if the valid time of a control action is denoted by a conditional time interval (e.g., until the blood value has improved), or if the workflow contains conditional parts such as conditional splits or loops. In this case it is modeled that one action or an alternative action can give birth another event (when Trig_Ev_St = TRUE T13 will be fired) and further condition will be checked and accordingly action will be determined.

Table1: Places, Transitions, Tokens and their Purposes

Places	Transitions	Purpose of transition	Tokens Passed	Purpose of token/guard conditions
P1 SERVER P2 CLIENT INTERFACE	T1	When any external event occurs the server has to notify it to the client Or Server interacts with the client in response of client's event detection request Or condition evaluation request.	Ex_Ev (Boolean), Ev_Dt_Req (Boolean) Con_Ev_Req(Boolean)	Ex_Ev = TRUE (happening of an external event) Ev_Dt_Req=TRUE (server responses on client's event detection request) Con_Ev_Req=TRUE (server responses on client's condition evaluation request)
P1 SERVER P2 CLIENT INTER FACE	T2	The client interface can reply to the server in response of any event detection request or condition evaluation request placed by server	Ev_Type_Dis (String) Ev_Dt_Status[] (int array) [Cond_Type_Dis (String) Con_Ev_Status[] (int array)	[Ev_Type_Dis = GLOBAL AND Ev_Dt_Status[i] =1] OR [Cond_Type_Dis = GLOBAL AND Con_Ev_Status[i]=1] (client responses on server's event detection request or condition evaluation request)
P1 SERVER P2 CLIENT INTER FACE	T3	Request for global event detection may come from the client	Ev_Type_Dis (String) Ev_Dt_Status[ ](int array) (if an event noticed/attempted for detection then Ev_Dt_Status[i] is set to be 1 otherwise 0)	[Ev_Type_Dis = GLOBAL] AND [Ev_Dt_Status[i] !=1] (client sends request to server for global event detection)
P2 CLIENT INTERFACE	T4	Each event detection completed and each event detection result is true so go for rule firing	Ev_Type_Dis (String) Ev_Type_Dis (String) Ev_Dt_Status[ ] (int array) Event_Dt_Res[i] (int array) (if an event happened/detected then Ev_Dt_Res[i] is set to be 1 otherwise 0)	[Ev_Type_Dis = LOCAL OR Ev_Type_Dis = GLOBAL] AND [Ev_Dt_Status[i] =1 for i=1 to n] AND [Event_Dt_Res[i]=1 for i=1to n] (all local and global events detected and each event detection result is true)
P2 CLIENT INTERFACE	T5	Client interface will wait for each sub event detection for maximum time allotted for that event detection	Ev_Dt_MaxTime(integer timed)	Ev_Dt_MaxTime is the deadline for detecting an event All tokens will be passed from P2 to P3 when Ev_Dt_MaxTime is over
P4 EVENT DETECTION TRUE AND RULE FIRED P6 ACTION	T6	After evaluating the conditions if all evaluated as true then go for action	Cond_Type_Dis(String) Cond_Type_Dis (String) Con_Ev_Status [i] (integer array) Con_Ev_Res[i] (integer array) (if an condition noticed /attempted for evaluation then Con_Ev_Status[i] is set to be 1 otherwise 0) if a condition True then Con_Ev_Res[i] is set to be 1 otherwise 0)	[Cond_Type_Dis = LOCAL OR Cond_Type_Dis = GLOBAL] AND Con_Ev_St atus[i] =1 for i=1to n AND Con_Ev_Res[i]=1 for i=1to n (all local and global sub conditions evaluated and each condition evaluation result is true)
P4 EVENT DETECTION TRUE AND RULE FIRED	T7	Request to server for global condition evaluation	Cond_Type_Dis(String) Con_Ev_Status[i] (integer array)	Cond_Type_Dis = GLOBAL AND Con_Ev_Status[i] =FALSE some global conditions not evaluated so client sends request to server for that.
P4 EVENT DETECTION TRUE AND RULE FIRED	T8	Client interface will wait for each sub condition evaluation process for maximum time allotted for condition evaluation	Cond_MaxTime(int timed)	Cond_MaxTime is the deadline for evaluating a condition All tokens will be passed from P4 to P5 when Cond_MaxTime is over
P3 EVENT DETECTION FALSE OR INDEFINITE P8 NO ACTION	T9	All events detections are definite but result of some events are false So go for no action	Ev_Type_Dis (String) Ev_Type_Dis (String) Ev_Dt_Status[i](integer array) Event_Dt_Res[i] (integer array) (if an event noticed/attempted for detection then Ev_Dt_Status[i] is set to be 1 otherwise 0) if an event happened/detected then Ev_Dt_Res[i] is set to be 1 otherwise 0)	[Ev_Type_Dis = LOCAL OR Ev_Type_Dis = GLOBAL] AND Ev_Dt_Status[i] =1 for i=1 to n AND Event_Dt_Res[i] !=1 for i=1to n All events (local or global)are attempted for detection but some event detection results are false i.e these events are not occurred

Table1: Places, Transitions, Tokens and their Purposes(continued..)

Places	Transitions	Purpose of transition	Tokens Passed	Purpose of token/guard conditions
P3 EVENT DETECTION FALSE OR INDEFINITE P7 ALTER NATIVE ACTION	**T10**	**Some event detections are indefinite, so go for alternative action**	**Ev_Type_Dis (String) Ev_Type_Dis (String) Ev_Dt_Status[i](integer array)**	**[Ev_Type_Dis = LOCAL OR Ev_Type_Dis = GLOBAL] AND Ev_Dt_Status[i]!=1 for i=1 to n Some events are unnoticed for some reason so Ev_Dt_Status[i]!=1 for i=1 to n**
P5 CONDITION EVALUATION FALSE OR INDEFINITE P7 ALTERNATIVE ACTION	**T11**	**Some condition evaluations are indefinite, so go for alternative action**	**[Con_Type_Dis (String) Con_Type_Dis (String) Con_Ev_Status[] (integer array)**	**[Con_Type_Dis = LOCAL OR Con_Type_Dis = GLOBAL] AND Con_Dt_Status[i] !=1 for i=1 to n Some condition evaluations are unnoticed**
P5 CONDITION EVALUATION FALSE OR INDEFINITE P8 NO ACTION	**T12**	**All evaluation of condition are definite but result of some evaluations are false**	**[Con_Type_Dis (String) Con_Type_Dis (String) Con_Ev_Status[i] (int array) Con_Dt_Res[i](int array)**	**[Con_Type_Dis = LOCAL OR Con_Type_Dis = GLOBAL] AND [Con_Dt_Status[i] =1 for i=1 to n] AND [Con_Dt_Res[i]!=1 for i=1to n] All condition evaluations are noticed but result of some evaluations are false**
P2 CLIENT INTERFACE P6 ACTION P7 ALTERNATIVE ACTION	**T13**	**One event can generate another event**	**Trig_Ev_St = Boolean**	**Trig_Ev_St = TRUE one action or an alternative action can generate a new event**

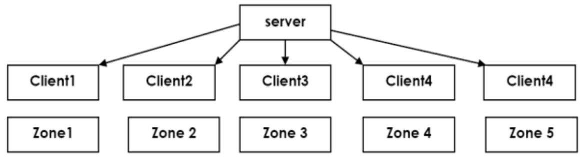
VI. CASE STUDY

Now we discuss two real life scenarios which has application of distributed active database and this generic model can be appropriate to understand the complex concurrent activities of these systems and it will be also helpful to automate these systems because of the association of modeling language (ML) with CTPN.

Case Study 1: A co-operative transport system[26][27]

In a cooperative convoy, a vehicle interacts with other vehicles, service providers and infrastructure systems to make the travel safe and convenient. Through these interactions a vehicle can share its domain-specific information – acquired from service providers and infrastructure – with other vehicles in the convoy. The system should support collaboration (i.e., allow drivers to share specific travel information) and coordination, and be able to cope with the changes.

Let us suppose one situation where one highway is divided into n (e.g 5) number of zones. So under one server there will be 5 clients for 5 zones.



Application running under each client:

- Detecting status of the car.
- Detecting average speed of the car.
- Detecting collision on road.
- Detecting traffic jam
- Forecasting weather by receiving information from server (weather information from server will be treated as external event).
- Directing the shortest route on response to the request of the passenger.
- Receiving information about accident/collision on next zone from other client and re-directing the route.

Let us suppose one situation when one passenger, staying at zone2, is going from zone 2 to zone 5, is asking the client2 whether there are any traffic jam or accidental news in this route. So the event (detecting any traffic jam or accidental news in the route) should be composed of a local event(detecting any traffic jam or accidental news in zone 2 itself.) and a global event (detecting any traffic jam or accidental news in zone3, zone4, zone5). The local event will be carried in client2 itself and the client2 will also send global event detection request to the server. The server will send event notification to the client3, client4, client5. The client3, client4, client5 will notice the event((if noticed then Ev_Dt_Status[i] will set to be 1, otherwise 0) and check whether there is traffic jam or not. If detected then Event_Dt_Res[i] should set to be 1, otherwise 0. Now if the event is detected then different local and global sub conditions can be checked (e.g shortest distance from source to destination/alternative routes/journey time etc.) and according to that action, no action or alternative action can be chosen by the client.

Let us assume another situation of an accident. Suppose one car has been fallen into a river in a mountain side highway and the respective client cannot trace that particular car, So after waiting a maximum time interval the respective client or the server must take the action of rescuing that car. Here lies the applicability of temporal constraint.

Case Study 2: Research in Clinical trials in Medical Domains[24][25]

Investigator initiated clinical trials are used to introduce new treatments and allow the compatibility and the efficiency of new therapies to be examined and offer a way to compare different treatments for the same disease pattern. As for example, Oncological therapies are complex and their process may differ from patient to patient, even if they have the same disease. However, the necessity of a treatment modification for a particular patient often does not reveal itself until the treatment is already in progress. The research medical team is located at server site and they can contact all the client sites where the local doctors and patients are located. The medical research group tries to monitor different patients locating at different hospitals or nursing home and the observations are required in the research work. So there are two context: one treatment side i.e different patients locating at different locations are monitored by a group of medical persons locating at a central place, and another context is the research work where the medical team tries to invent the appropriate diagnosis of a disease after considering all the effects and side effects of medicines and other medical treatment given to all the patients locating at different locations and suffering from the same disease. First we consider the treatment side context where we can apply the ECA rule and accordingly our proposed model will work on that medical system.

The changes are triggered through events (exceptions) such as allergies, low blood counts, or unexpected disease progression requiring adaptations such as dosage modifications, drug replacements or individual supportive therapy steps. Treatment protocols additionally contain information about the proper reactions to such exceptional situations. However, a specialized physician has to handle many different protocols for different patients at the same time. Therefore, a convenient consultation system should observe the status of the therapies currently being applied, offer automatic recognition of exceptional situations and appropriate decision support for handling such situations. Furthermore, the system should be able to automatically adapt affected therapy processes to adequately handle the flexibility of treatment processes. The rule base handles the medical knowledge represented in the protocols and is used to detect exceptional situations.

The rules are of the following format:

WHEN exceptional event
WITH condition
THEN treatment adaptation
VALID-TIME time period

The event-condition part (WHEN/WITH) specifies the event and the conditions under which treatment adaptations are required. The action part (THEN) describes the necessary treatment modifications. The optional valid time part (VALID-TIME or VT) specifies a time period during which the modification should be applied. It is denoted either by a fixed time interval or date, or by a conditional time interval whose end is specified by a condition that has to be satisfied (e.g., until the blood value has improved). Before applying one treatment to a patient the doctors can see the effect of that particular treatment on another similar patient locating at different location (which can be considered as different client site). So if we consider the effect of one particular treatment to a particular type of patient as an event then it can be broken down into n number of sub events because n number of similar patients can be there in different n client site. One particular client site(local doctor team) can send request to other clients or to the server for suggestion of treatment for a particular case. Then after noticing the event detection result of n client sites and considering the local and global conditions one client site can take proper decision of action, no action or alternative action. From the research context also, the server site will observe the event detection result (result of a particular treatment on particular types of patients) and all sub condition evaluation results from all the client sites and can invent a new therapy.

VII. BENEFIT OF THE WORK

By this model we clearly can understand the overall scenario of the distributed active database. There are several concurrent situations/operations involved in this work which are very significantly modeled by the Color Petri Net. Since this model is done by Color Petri Net, the modeling language is already attached herewith. The timing constraints modeled by Color Time Petri Net has also significant role to cope up the situation of indefiniteness due to inaccessibility of a remote sub-system.

Ambient intelligence is closely related to the intelligent service system in which technologies are able to automate a platform embedding the required devices for powering context aware, personalized, adaptive and anticipatory services. In today's world, an important issue of medical world, cooperative transport system, traffic congestion control system, smart home environment which are focused on developing a distributed, multi-agent, adaptable environment concern for online parameters monitoring. The solution is to pervade into computing systems which have the capabilities of monitoring, data acquisition and data transfer from electronic environments. Here we discuss two such system where pervasive computing, ambient intelligence significantly work with distributed active database. The generic model of distributed active database by CTPN can be applied to develop such intelligent systems.

VIII. CONCLUSION

This paper focuses on the ambient intelligence scenarios, considering a real time problem design and development of a CTPN model of distributed active database on this context. Applicability of CTPN in 'Distributed Active Database' has been shown in this paper. Both the primitive and composite events have been considered while we dealt with local and global issues. Concepts of alternative actions in case of 'event detection being indefinite' has also been included in the design to address such situation.

REFERENCES

[1] C. A. Petri, "*Kommunikation mit Automaten*," Ph.D. dissertation, Univ.Bonn, Germany, 1962.
[2] Tadao Murata, "Petri Nets: Properties, Analysis and Application ", *Proceedings of the IEEE*, Vol.77, No.4, April,1989
[3] Carlo Ghezzi,Mehdi Jazayeri and Dino Mandrioli , "*Fundamentals of Software Engineering (2nd Ed)"*.
[4] K. Jensen, "Coloured Petri Nets --Basic Concepts Analysis Methods and Practical Use", vol. 2, *Springer-Verlag New York Berlin Heidelberg,'95*
[5] N.W.Paton and O.Diaz,"Active database Systems *ACM Comput. Surv.*, vol. 31, no. 1, pp. 64–103,'99
[6] Sharma Chakravarthy, Hui Liaot, "Asynchronous Monitoring of Events for Distributed Cooperative Environments", 2001 *IEEE*
[7] S. Chakravarthy and D. Mishra. Snoop: An expressive event specification language for active databases. *UF-CIS Technical Report TR-93- 007, University of Florida*, 1993.
[8] S. Gatziu and K. Dittrich. Events in an active Object oriented database system. *In Proc. of the 1st Workshop of Rules in Database Systems*, pages23-29, 1993.
[9] N. Gehani, H. V. Jagadish, and O. Smueli. Event specification in active object-oriented database *In Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 81- 90, 1992.
[10] V. Hyesun Lee. Support for temporal events in Sentinel: Design, implementation, and preprocessing. *Master's thesis, University of Florida, Gainesville*, 1996.
[11] Xiaoou Li, Joselito M. Medina, Sergio V. Chapa, "Applying Petri Nets in Active Database Systems" , *IEEE* , JULY 2007
[12] LatifaBaba-hamed,Hafida Belbachir," The Priority of Rules and the Termination Analysis Using PetriNets", *The International Arab Journal of Information Technology*, Vol 4, No. 2, April2007
[13] Joselito Medina-Marın, Gilberto P´erez-Lechuga, Xiaoou Li "ECA rule analysis in a Distributed Active Database",**(**2009 *International Conference on Computer Technology and Development*).
[14] Thomas Heimrich , Günther Specht,"Enhancing ECA Rules for Distributed Active Database Systems", *Springer-Verlag Berlin Heidelberg* 2003.
[15] Christian HOAREAU, Ichiro SATOH, "Modeling and Processing Information for Context-Aware Computing: A Survey", *New Generation Computing*, 27(2009)177-196 *Ohmsha, Ltd. and Springer*
[16] Diane J. Cook, Juan C. Augusto , Vikramaditya R. Jakkula," Ambient intelligence: Technologies, applications, and opportunities", *2009 Elsevier*.
[17] Technical Report, Distributed Complex Event Detection, for Pervasive Computing, Dan O'Keeffe, July 2010,*Computer Laboratory, University of Cambridge, United Kingdom* [19] Hongbo Ni, Bessam Abdulrazak, Daqing Zhang,Shu WuCDTOM:" A Context-driven Task-oriented Middleware for Pervasive Homecare Environment" *International Journal of UbiComp (IJU)*, Vol.2, 2011
[20] Edwin Wong, Lisa Burnell, Charles Hannon," An Active Architecture for Managing Events in Pervasive Computing Environments", Copyright © 2002, *American Association for Artificial Intelligence*.
[21] A. Spiteri Staines , "A Colored Petri Net for the France-Paris Metro", INTERNATIONAL JOURNAL OF COMPUTERS Issue 2, Volume 6, 2012
[22] ANA MUÑOZ, JOSE AGUILAR,"Ontological Scheme for Intelligent Database**",** Proceedings of the 11th WSEAS International Conference on COMPUTERS, Agios Nikolaos, Crete Island, Greece, July 26-28, 2007
[23] DR DILSHAT DJUMANOV, DR PETER DABNICHKI, DR RADIVOY AREZINA, " Pervasive computing systems for medical monitoring", 1st WSEAS International Conference on BIOMEDICAL ELECTRONICS and BIOMEDICAL INFORMATICS (BEBI '08) Rhodes, Greece, August 20-22, 2008.
[24] CARMEN AURORA BULUCEA, MARIUS CONSTANTIN POPESCU, CORNELIA AIDA BULUCEA, ANCA PATRASCU, GHEORGHE MANOLEA," Real Time Medical Telemonitoring of Sustainable Health Care Measuring Devices**",** Proceedings of the 8th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING & DATA BASES (AIKED '09)
[25] U. Greiner, R. Mueller, E. Rahm, J.Ramsch, B. Heller, M. Loeffler, "AdaptFlow: Protocol-based Medical Treatment Using Adaptive Workflows", Methods Inf Med 1/2005; 44: 80–8
[26] Muhammad Ashad Kabir, Jun Han, Alan Colman, and Jian Yu," SocioTelematics: Leveraging Interaction-Relationships in Developing Telematics Systems to Support Cooperative Convoys".
[27] N.SUDHA BHUVANESWARI, S.SUJATHA, R.YAMUNA, "Vibrant Ambient Intelligent System for Traffic Congestion Control in Coimbatore City (VAISTC4)", RECENT ADVANCES in NETWORKING, VLSI and SIGNAL PROCESSING, ISBN: 978-960-474-162-5