Representation of Smart Environments Using Distributed P Systems

Susan Elias, Rajalakshmi V., and Sivaranjani S.

Department of Computer Science and Engineering
Sri Venkateswara College of Engineering
Chennai, India
`{susan,vraji}@svce.ac.in, sivaranjani_s@ymail.com`

Abstract. The representation of a system for effective reasoning in smart environments is the focus of this research work. The motivation here is to enable assisted living in a smart home. The smart environment is represented here using a variant of an existing membrane computing model the Distributed *P* System. Evolution rules that govern the mechanism and functionality of the system are required for effective modeling. The algorithms proposed and presented here generate these evolution rules automatically and also integrates all generated *P* Systems to form the Distributed *P* System.

Keywords: Smart Environment, Membrane Computing, Distributed *P* Systems, Assisted Living.

1 Introduction and Motivation

The concept of smart environments was first envisaged by Mark Weiser in the late 1990's and it has evolved as a very challenging area of research. Smart environment has its roots based on two near similar paradigms - *ambient intelligence* and *pervasive computing* [4]. The key to smart environment lies in capturing the data live [5] from the environment and in using smart devices that are enriched with intelligence and enhanced functionality to increase the reasoning capabilities of the devices. ***Recognition***, ***Reasoning*** and ***Retrieval*** in smart environments have been dealt with in recent related research works [2][3]. The challenge now is to '***Represent***' these smart environments effectively for efficient ***Reasoning***.

Membrane Computing is a research area that aims to abstract computing ideas and models from the structure and functioning of living cells. Membrane systems also known as *P* systems, deal with distributed and parallel computing models, processing multi-sets of symbol objects in a localized manner. Evolution rules and evolving objects are encapsulated into compartments delimited by membranes. Objects are communicated between the compartments and with the environment by means of communication rules. Objects evolve by means of evolution rules which are localized and associated with the regions of the membrane structure. In this paper, a variant of the Distributed *P* System [6] is proposed and generated dynamically.

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 159–164, 2012.

2 Proposed Model

The work proposed in this paper utilizes a variant of the existing Distributed *P* System for effective modeling of smart environments. The environment is a network of devices embedded with intelligence that continuously monitors the various activities happening in the environment. The monitoring systems installed all around the environment are assumed to be *smart*. Smart devices are the devices that not only have the ability to capture data but also apply reasoning on the data being captured in order to indicate the occurrence of specific events.

In order to model the environment a domain specific application of a smart home for assisted living is considered. The following assumptions are made for modeling the environment. The entire environment is represented as a Distributed *P* System and the environment is divided into zones where each zone is a *P* System and a collection of such *P* Systems form a Distributed *P* System. This is illustrated in Fig. 1a., where four zones are identified in the environment which are represented as four *P* Systems Π_1, Π_2, Π_3 *and* Π_4 respectively. The single occupant in the home is considered as a mobile *P* System uniquely labeled as Π_p. The embedded devices in the environment that enable monitoring could range from video cameras and audio recorders to microphones, sensors (light and smoke), controllers, LEDs etc., Each of these devices will form a membrane in the *P* System that represents a zone where the devices are physically located. Fig. 1b. is an example of *P* System with membranes indicating the presence of smart devices in a particular zone. Evolution rules are written to trigger the alarm in each zone in case of an emergency.

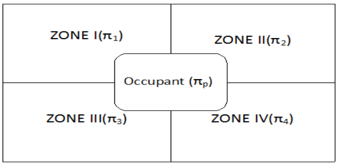

Fig. 1a. Smart Home as Distributed *P* System

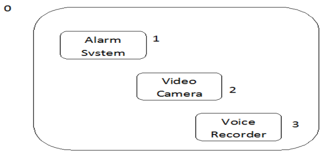

Fig. 1b. The *P* System representing a zone

A prototype is being developed, that facilitates the design of smart spaces. Fig. 2. shows a snapshot of the prototype and the backend functionality of the prototype as well.

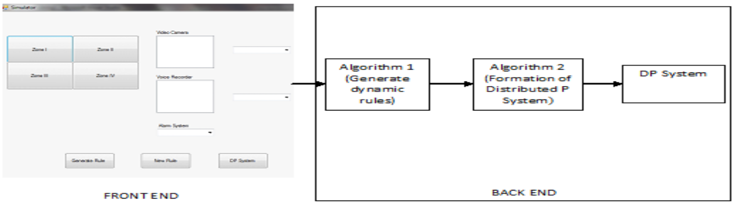

Fig. 2. Smart Space Layout Generator

The layout of the home is used as input in the prototype. Each room in the home is considered as a zone. Let the smart devices fixed at each zone be video recorders, audio recorders and an alarm system. Assume the smart devices capture information in the form *Dev_x = (lab, lev)* where, *'lab'* and *'lev'* represents the *label* and *level* of the activity captured by the device '*x*'. The activity for an audio system is assumed to be the sound recorded and for the video system, it represents the current scenario captured. The level for the audio system represents volume and for video system, it represents description about the activity. Consider the device number of the alarm system in all zones to be 1. Assume that there are three priority levels for an alarm system. Same label or level can be assigned to different activities captured by various devices and in different zones. To differentiate the labels or levels in each device and in each zone, the label or level can be denoted by lab_{ik} *or* lev_{ik}, *(where 'i' represents the zone number in the home and 'k' represents the device number in the zone).*

The prototype would collect the zone identity *'i'* which represents the i^{th} *P* System and all label-level pairs *Dev_x = (lab, lev)* from all devices in that zone. This forms the input for *Algorithm 1* which generates rule for a *P* System. *Algorithm 2* generates a Distributed *P* System from the *P* Systems.

The formal definition of proposed variant of Distributed *P* System with degree ($n \geq 1$) has the following construct:

$$\Delta = (O, \Pi_1, \ldots, \Pi_n, \Pi_p, S);$$

where,

1. O is an alphabet of objects;
2. $\Pi_1, \ldots, \Pi_n$ are *P* Systems containing m membranes with skin membranes labeled with $s_1, \ldots, s_n$, respectively where,

   $$\Pi_i = (V_i, \mu_i, w_{i1}, \ldots, w_{im}, E_i, R_{i1}, \ldots, R_{im})$$

   where,
   a. V_i is an alphabet of objects, $V_i \subseteq O$;
   b. μ_i is a membrane structure of the i^{th} *P* System which is of the form $[_0[_1]_1[_2]_2[_3]_3\ldots[_m]_m]_0$;
   c. $w_{i1}, \ldots, w_{im}$ represents the multisets of objects available in each membrane;
   d. $E_i \subseteq V_i$ represents the objects available in arbitrarily many copies in the environment;
   e. $R_{i1}, \ldots, R_{im}$ represents the evolution and communication rules used in each *P* System. The rules have the form $a \rightarrow v$, where $a, v \in V$.
3. Π_p is a mobile *P* System that represents the single occupant in the environment. This is an additional component proposed in this paper to suit the domain specific application. Π_p has the following construct:

   $$\Pi_p = (V_p, \mu_p, w_p, E_p, R_p)$$

   where,
   a. V_p is an alphabet of objects, $V_p \subseteq O$;
   b. μ_p is a membrane structure of the mobile *P* System with only the skin membrane $[_0]_0$;
   c. w_p are strings representing the multisets over V associated with skin region;

   d. $E_p \subseteq V_p$ represents the objects available in arbitrarily many copies in the environment;
   e. R_p represents evolution rules of the mobile *P* System.
4. S is a finite set of rules of the form $(s_i, u/v, s_j)$, where $1 \leq i, j \leq n$, $i \neq j$ and $u, v \in O^*$.

2.1 Proposed Algorithms

Algorithm 1 will generate rules for triggering the alarm. This algorithm collects input from the prototype. Output of this algorithm is a rule generated to trigger an alarm system. In *line 1*, LHS and RHS represents the left and right hand side objects of the rule R_{i1} which are initially set to NULL. Assume *i* and *j* represents the *P* System. The *for* loop in *lines 2-7* do the following for each membrane $k \neq 1$ of the *P* System *i*:

- In *lines 3-5*, V_i, w_{ik}, R_{ik} are collected.
- In *line 6*, concatenation of multisets of objects to form LHS of *Ri1* is performed.

Then, RHS is set to appropriate alarm and collects V_1, w_{i1}, R_{i1} for alarm system in *lines 9-14*. Finally in *line 15*, all w_{ik} are grouped to form w_i.

Rules generated in *Algorithm 1* forms individual *P* Systems. After all *P* Systems are constructed with dynamic rules, the mobile *P* System is formed in the following manner.

- $V_p \leftarrow \bigcup_{i=1}^{n} V_i$. This implies alphabets of mobile *P* System are the set of all alphabets that belong to all *P* Systems.
- $w_p \leftarrow \bigcup_{i=1}^{n} w_i$. This implies multisets of objects of mobile *P* System are the set of all multisets of objects that belong to all *P* Systems.
- $R_p \leftarrow \bigcup_{i=1}^{n}$ Rules in all skin membrane s_i.

```
Input:  i, Dev_1, Dev_2, ...., Dev_m
Output: A rule generated to trigger the alarm system
 1 Set LHS = RHS = NULL
 2 for each membrane '$k \neq 1$' in $\Pi_i$ do
 3     $V_i \leftarrow V_i \bigcup \{lab_{ik}\}$
 4     $w_{ik} \leftarrow w_{ik} \bigcup \{lab_{ik}^{lev}\}$
 5     $R_{ik} \leftarrow R_{ik} \bigcup \{lab_{ik}^{lev} \rightarrow (lab_{ik}^{lev}, out)\}$
 6     $LHS \leftarrow LHS. \{lab_{ik}^{lev}\}$
 7 end
 8 For membrane $k=1$,
 9 $V_i \leftarrow V_i \bigcup \{lab_{i1}\}$
10 $w_{i1} \leftarrow w_{i1} \bigcup \{lab_{i1}^{lev}\}$
11 $RHS \leftarrow lab_{i1}^{lev}$
12 $R_{i1} \leftarrow R_{i1} \bigcup \{LHS \rightarrow RHS\}$
13 $R_{i1} \leftarrow R_{i1} \bigcup \{RHS \rightarrow s_i\}$
14 $R_{i1} \leftarrow R_{i1} \bigcup \{s_i \rightarrow lab_{j1}^{lev}\}$
15 $w_i \leftarrow \bigcup_{k=1}^{m} w_{ik}$
```

Algorithm 1. Algorithm to generate dynamic rules

Algorithm 2 generates the Distributed *P* System. This algorithm takes as input all individual *P* Systems to form Distributed *P* System. Assume *i, j* and *k* represents the *P* System. In *line 1*, O is set to empty set. The *for* loop in *lines 2-6*, will add the following:

- In *line 3*, V_i of each *P* System is added to the set *O*.
- In *line 4*, *P* Systems are added to the Distributed *P* System *Δ*.
- In *line 5*, Skin-to-skin communication rules are also added to the set *S*.

Finally the *lines 8 and 9*, represents the same function for the mobile *P* System.

```
Input:  P Systems = $(\Pi_1, \ldots, \Pi_n, \Pi_p)$
Output: DP System $\Delta = (O, \Pi_1, \ldots, \Pi_n, \Pi_p, S)$
1 Set $O \leftarrow \Phi$
2 for each $\Pi_i$ do
3     $O \leftarrow O \bigcup \{V_i\}$
4     $\Delta \leftarrow \Delta \bigcup \{\Pi_i\}$
5     $S \leftarrow S \bigcup \{(s_i, u/\lambda, s_j)\}$
6 end
7 For P System $\Pi_p$,
8     $\Delta \leftarrow \Delta \bigcup \{\Pi_p\}$
9     $S \leftarrow S \bigcup \{(s_i, u/\lambda, s_p)\}$
```

Algorithm 2. Algorithm for formation of Distributed *P* System

2.2 Illustration

Consider a smart home with 4 zones. Fig. 3a. is an illustration of rule generation in *P* System *i*. In this illustration, the *P* System *i* contains an alarm system, a video camera, a voice recorder which form the membranes *1, 2, 3* respectively. Consider the activity recorded by video camera has the label *'a'* and level *'2'* and also consider the activity recorded by voice recorder has the label *'b'* and level *'4'*. For this particular combination, if an alarm has to raised, with high alert, say '*3*'. The rule generated in each membrane is shown in the Fig. 3a.

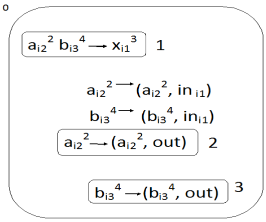

Fig. 3a. A rule generated in i^{th} *P* System

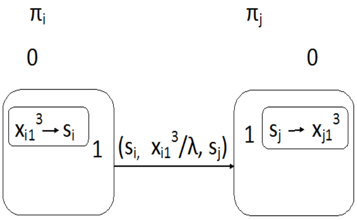

Fig. 3b. Skin to skin communication rule

After an alarm is raised in one zone, alarm system in all other zone should be triggered. To trigger the alarm system in all other zones skin to skin communication rules are required which is illustrated in Fig. 3b.

3 Complexity Analysis

Consider '*Algorithm 1'*, for dynamic rule generation. Let m be the number of membranes and n be the number of activities associated with each membrane. The *for* loop is bounded by m-1 times the execution of the function *union*. The *union* operation takes the time complexity of $O(m\alpha(m+n, n)+n)$, where α is the functional reverse of Ackerman's function [1]. Therefore, the time complexity of the loop is $O(mn)$ and hence, the algorithm takes a worst case time complexity of $O(mn)$.

Consider '*Algorithm 2'*, for formation of Distributed *P* System. Let p be the number of *P* Systems and m is the number of membranes in each *P* System. Calculating the complexity for the algorithm in the same manner, the algorithm arrives at the worst case time complexity of $O(pm)$. Since both the algorithms do not take exponential time, the algorithms are proved to be efficient.

4 Conclusion

This paper focuses on the effective representation of the smart environment for efficient real time response. A domain specific application for assisted living in smart home has been proposed. Distributed *P* Systems are generally used in representing the functioning of the system and thereby solving problems in a distributed manner. Algorithms are designed for automatically representing smart spaces using Distributed *P* System. Complexity analysis presented in this paper indicates that the algorithms are efficient.

References

1. Blum, N.: On the single-operation worst-case time complexity of the disjoint set union problem. In: Mehlhorn, K. (ed.) STACS 1985. LNCS, vol. 182, pp. 32–38. Springer, Heidelberg (1984)
2. Menon, V., Jayaraman, B., Govindaraju, V.: Three R's of Cyberphysical Spaces. IEEE Computer Society 44(9), 73–79 (2011)
3. Menon, V., Jayaraman, B., Govindaraju, V.: Multimodal identification and tracking in smart environments. Personal and Ubiquitous Computing Journal-Springer 14(8), 685–694 (2010)
4. Menon, V., Jayaraman, B., Govindaraju, V.: Biometrics Driven Smart Environments: Abstract Framework and Evaluation. In: Sandnes, F.E., Zhang, Y., Rong, C., Yang, L.T., Ma, J. (eds.) UIC 2008. LNCS, vol. 5061, pp. 75–89. Springer, Heidelberg (2008)
5. Menon, V., Jayaraman, B., Govindaraju, V.: Integrating recognition and reasoning in smart environments. In: 4th International Conference on Intelligent Environments, pp. 1–8 (2008)
6. Paun, G., Perez-Jimenez, M.J.: Solving Problems in a Distributed way in Membrane Computing: dP Systems. International Journal of Computers, Communication and Control 2, 238–250 (2010)