SPATIAL KNOWLEDGE MANAGEMENT SYSTEM FRAMEWORK THROUGH SELF-ADAPTIVE MODELING

M. Nagabhushana Rao
Associate Professor

A. Rama Mohan Reddy
Associate Professor

Dr. P. Govindarajulu
Professor

Department of Computer Science and Engineering
Sri Venkateswara University College of Arts and Science / College of Engineering
Sri Venkateswara University
TIRUPATI – 517 502. Andhra Pradesh , INDIA

ABSTRACT

In this paper, we are proposing self-adaptive software architecture for disaster management. This system uses geo-spatial data and applies spatial collocation rule to mine spatial patterns. These spatial collocation patterns are stored as knowledge database and subsequently provided to Disaster Management teams for rescue and management. A spatial collocation rule for epidemic is proposed. This system uses self-adaptive architecture for adapting different disaster components with out developing a separate system for every new disaster from the scratch. This system identifies disaster dynamically and adapts the appropriate commodity component in the web. The software architecture proposed is Web-based distributed system, which provides components for Self-adaptation. An EDMS (Epidemic Disaster Management System) is presented for an epidemic disaster.

KEYWORDS: Data mining, self-adaptive software architecture, mining collocation, spatial database.

1. INTRODUCTION

Geography is an integrative discipline and geographic data under analysis often span across multiple domains. The complexity of spatial data and geographic problems, together with intrinsic spatial relationships, constitute an enormous challenge to conventional data mining methods and call for both theoretical research and development of new techniques to assist in deriving information from large and heterogeneous spatial datasets [1]. Due to larger heterogeneity of spatial data, the providers of geographic data specify different models for same spatial objects. *Context specific semantics* (CSS) is one of the best approaches suggested which deals with provision of feature space derivations. An Ontological analysis need to be done on the fundamentals of the domain space. Spatial data mining becomes more interesting and important as more spatial data have been accumulated in spatial databases [2]. ***Mining spatial co-location patterns*** is an important spatial data-mining task with broad applications.

A hypothesis space is formed by all possible configurations of the tools used to detect patterns in a feature space. Characteristically, however, the hypothesis space for a large and high dimensional geographic dataset has an extreme degree of complexity. Complexity is caused by several factors. First, each pattern may involve a different subset of variables from the original data, and the number of such subsets (hereafter subspaces), i.e., possible combinations of attributes, is huge. Second, inside a subspace, potential patterns can be of various forms (e.g., clusters can be various shapes). Third, for a specific Pattern form (e.g., cluster of a specific shape), its parameter space is still huge, i.e., there are many ways to configure its parameters. Fourth, patterns can vary over geographic space, i.e., patterns can be different from region to region [6].

2. MINING COLLOCATION PATTERNS FROM SPATIAL DATA

Mining collocation patterns gives the standard of observing the generic characteristics of a given spatial zone with more relevant Boolean features with their **s** ***(support)*** and ***c (confidence)*** [**3**]. The work of mining collocation patterns into spatial statistics and combinatorial approaches. The spatial co-location pattern-mining framework presented in the previous works has bias on popular events. It may miss some highly confident but "infrequent" co-location rules by using only **"*support*"**-based pruning. In a spatial database **S**, let **F** = $\{f_1,\ldots,f_k\}$ be a set of *Boolean spatial features*. Let **I** = $\{i_1,\ldots,i_n\}$ be a set of n instances in the spatial database **S**, where each instance is a vector consisting of [instance-id, location, spatial features]. **~ *Neighborhood relation R over pair wise locations in S exists* ~** is assumed. The object of this collocation rule mining is to find rules in the form of **A∪B**, where **A** and **B** are subsets of spatial features. **A** determines the set of spatial features that form the antecedent part of the rule and **B** defines the action and its consequential parts the support and the confidence. The rule indicates the coincidence of the spatial collocation rule absorbs the action of the rule in the **"*nearby*"** regions of the spatial objects that comply with the collocation rule. To capture the concept of the predicate ***"nearby",*** the concept of neighbor-set **L** is a set of instances such that all *pair wise* locations in **L** are neighbors. Neighborhood relation R may be defined based on Euclidean distance and neighboring instances are linked by edges. A ***collocation pattern*** **C** is a set of spatial features, i.e., $\mathbf{C} \subseteq \mathbf{F}$. A neighbor-set **L** is said to be a *row* instance of collocation pattern **C** if every feature in C appears in an instance of **L**, and there exists no proper subset of **L** does so. We denote all row instances of a collocation pattern **C** as ***rowset*(C)**. In other words, ***rowset*(C)** is the set of neighbor-sets where spatial features in C collocate. The conditional probability is the probability that a neighbor-set in ***rowset* (A)** is a part of a neighbor-set in ***rowset* (A∪B)**. Intuitively, the conditional probability ***p*** indicates that, whenever we observe the occurrences of the spatial features in **A**, the probability to find the occurrence of **B** in a nearby region is ***p [4].***

2.1 HISTORY OF EPIDEMIC

The history of the epidemics has their own influence in the world history, which has taken many lives together. The epidemic is the result of an infection caused by any of the microbes. The microbe, following the indelible rules of evolution, strives to survive and reproduce, while the host's immune system mounts a warlike defense designed to find, destroy, and eliminate it. An agent that kills its host quickly cannot be expected to survive long enough to reproduce. Thus, excessive virulence is not selected for in evolution. Gels, which can reproduce and be passed from one host to another, are favored.

Here we are concentrating on the world's most deadly disease Cholera that was not only an epidemic but also a pandemic, which shook the world with the fear of death. In 1947, 20,500 TO 30,000 people infected in Egypt died. Despite modern medicine, cholera remains an efficient killer. This disease has swept the world in seven major pandemics, including a major outbreak in South America, particularly Peru, as recently as 1991. Cholera (also called Asiatic cholera) is an infectious disease of the gastrointestinal tract caused by the Vibrio cholera bacterium. These bacteria are typically ingested by drinking water contaminated by improper sanitation or by eating improperly cooked fish, especially shellfish. Symptoms include diarrhea, abdominal cramps, nausea, vomiting, and dehydration. Death is generally due to the dehydration caused by the illness. When left untreated, cholera generally has a high mortality rate. Treatment is typically an aggressive dehydration regimen usually delivered intravenously, which continues until the diarrhea ceases.

Example: In the imaginary, figure 1 the landscape describes two important spatial marks, sea and lake. The lichens and mosses at the western zone of the lake afflict the water in the lake, as most of the water is stagnant and covered by marsh. The people utilizing the water resources at this zone may be affected by so many kinds of fecal contamination in water and food. The water in the sea is contaminated with the high salts and the crude oil and base products, as the people cannot take the water for the domestic purposes, the climatic changes are affected by the water contents in the shores of sea. People breading their lives at the shores will have the indirect contamination of fecal material in the water and as well as in the form of moisture in the air. As the lake water is supplied into the agriculture lands surrounding in the adjacent zones, there people may be affected indirectly by the virulent characteristics [5].

2.2 FORM OF COLLOCATION RULE FOR EPIDEMICS

The collocation rules are very useful in detecting the affected areas by finding the symptoms of a disease. The collocation rule **C,** we use is,

C*: cause of epidemic ∩ causative agent, infection sources; in the****" nearby"*** *region with high probability.*

This typical confident co-location rule involves with both frequent and rare events because although infection is quite common and epidemic is rare the later factor implies the former one strongly. Assuming firstly, the '**b**' as the consequence of feature '**a**' is developed, forms a first level of collocation, which is identified by **a→b**, secondly, if the consequence '**c**' from the feature '**b**' is developed, forms a collocation, which is identified by **b → c**. As '**b**' already have an antecedent '**a**', the consolidated version of collocation, {**a, b**} **→c** can be formed. If '**c**' becomes another feature that can lead to the consequence of '**d'**, then the notation wholly represents the cause of '**d'** as {**a, b, c**} →**d.** Also implies to {**a** ∪ **b** ∪ **c**} → **d** representation[5]**.**

FORM OF COLLOCATION FOR EPIDEMICS

C: {*cause of epidemic****}*** **U**{*causative agent, infection sources****}****; in the* ***nearby*** *region with high probability*

3. IMPLEMENTATION STRATEGY

In case of epidemic disaster, the losses of related object in spatial are men, money, and animal. A considerable amount of populated area becomes an empty (no men's land) due to the migration of people from epidemic region to other locations. dynamically modifiable at any point in system lifecycle can be achieved by knowledge based approach in which adaptive behavior is achieved through a body of observations[6a].We are proposing an ***architecture-based self-adaptable system*** for various types of disaster. This system is also useful for damage surveys as earthquake, hurricane etc., provided some one could adapt the related software component quickly enough. The collocation rules which are proposed in this paper to predict the future disaster with sufficient assurance that the system would perform as intended. The collocation rules for various types of disasters are being supported as self-adaptable or commodity components. The system support an open adaptive behavior and adaptation plans that can be introduced during runtime. Adaptation mechanism depends on change, Change management is a controlling component that identifies change, reasoning content for change, specifying and implementing change, preserves system integrity and risks create by runtime modification. Changes can include the addition ,removal or replacement of components and connectors .

3. EPIDEMIC DISASTER MANAGEMENT SYSTEM (EDMS)

The Epidemic Disaster Management System that we propose here is to manage the epidemics disasters and it is based on the spatial collocation pattern-mining model. This system contains the four-components. First component for collecting information related to epidemics from spatial knowledge using the collocation rule based algorithm. Second component is a conventional database system, which is responsible for populating the tables with the data. The third component has the collocation rules of data mining. The fourth component is Database of Spatial knowledge, Fifth component is for processing the database, the sixth component has the report generation sub component, and guidelines for the field staff for dealing with the situation and for implementation of the results (Figures 3 & 4).

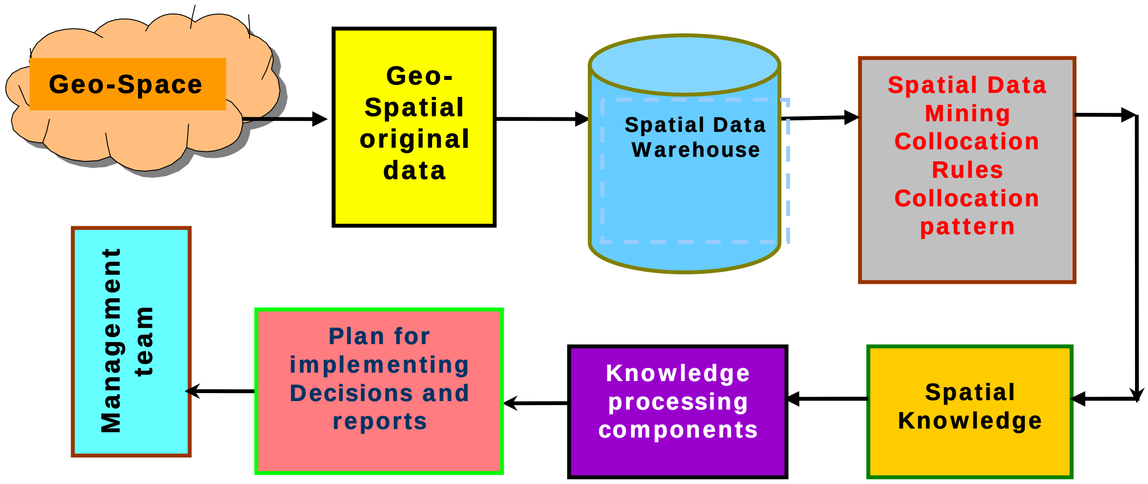

Figure 3 The Software components Structure of Disaster Management System (DMS)

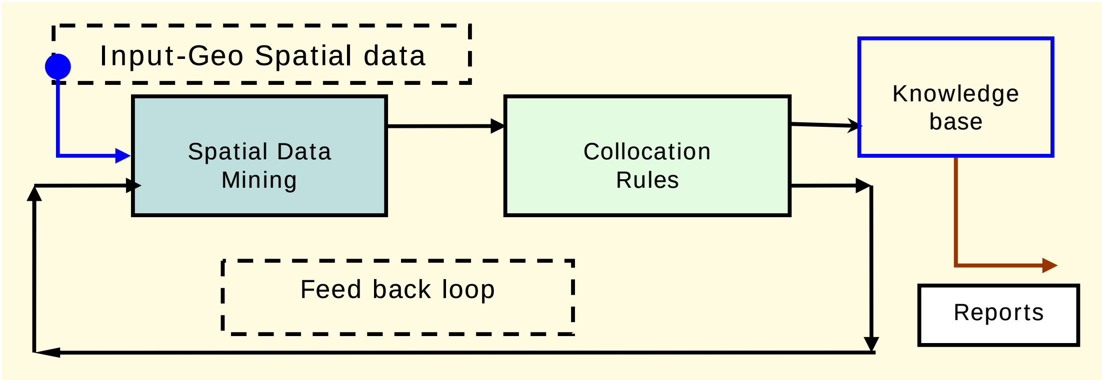

Figure 4 Simple Pipe and Filter Style Architecture for Implementing the disaster Based on spatial collocation pattern mining.

The DMS monitors the situation round the clock, identifies the type of disaster, shows the status and generates reports. The DMS autonomously re-plans its strategy, adapts it, and proceed to accomplish their objectives. During adaptation process, the system adapts new software collocation rule based component dynamically insert into DMS, without requiring system restart that detect the type of disaster automatically. DMS re-planning relies on analysis that includes feedback from current performance shown in figure 4 [11].

4. DMS FOR VARIOUS TYPES OF DISASTERS

This re-planning can take place autonomously, can involve multiple, distributed, cooperating planners and when major changes are demanded by human approval or guidance can cooperate with mission analysts. Throughout, system integrity requires the assurance of consistency, correctness and coordination of change. DMS can adapt itself to environmental disasters like Cyclone, Tsunami, Wars, Volcanoes, Floods, Earth Quake, and damage surveys in the above cases.

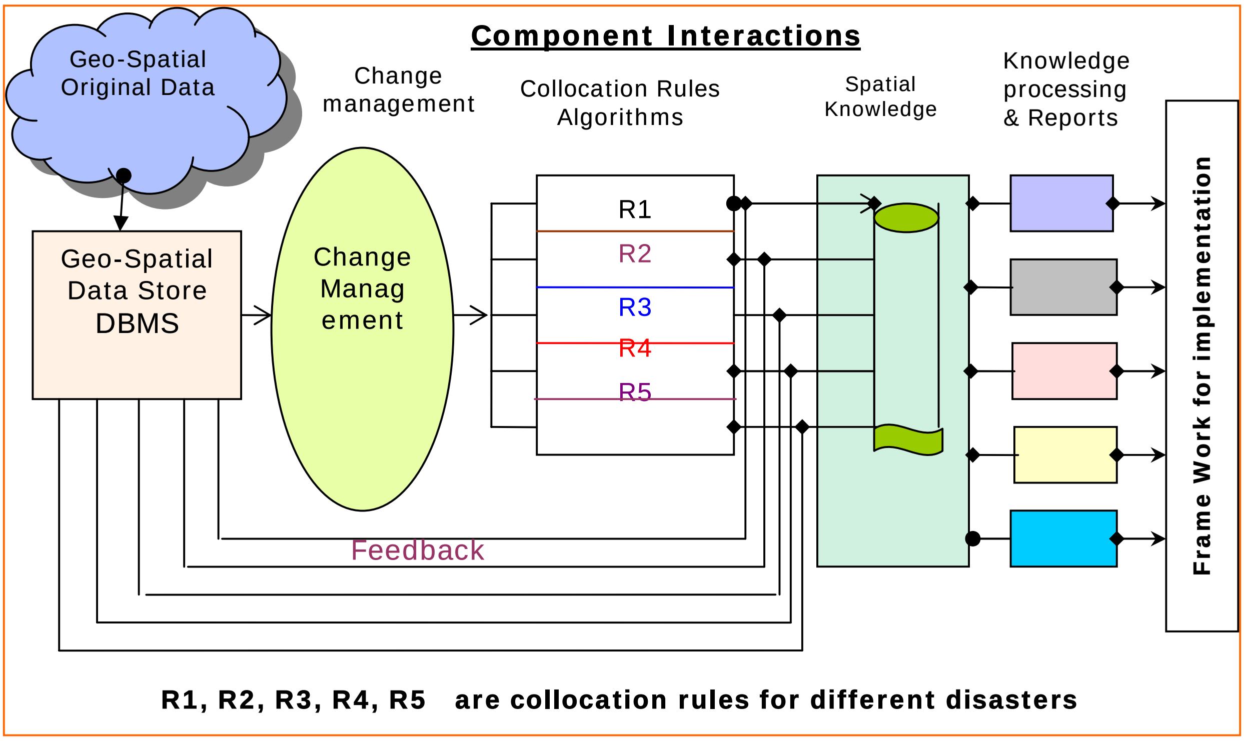

Figure 5 Proposed Software Architecture for Architecture-based Self-adaptable Disaster management system for various disasters.

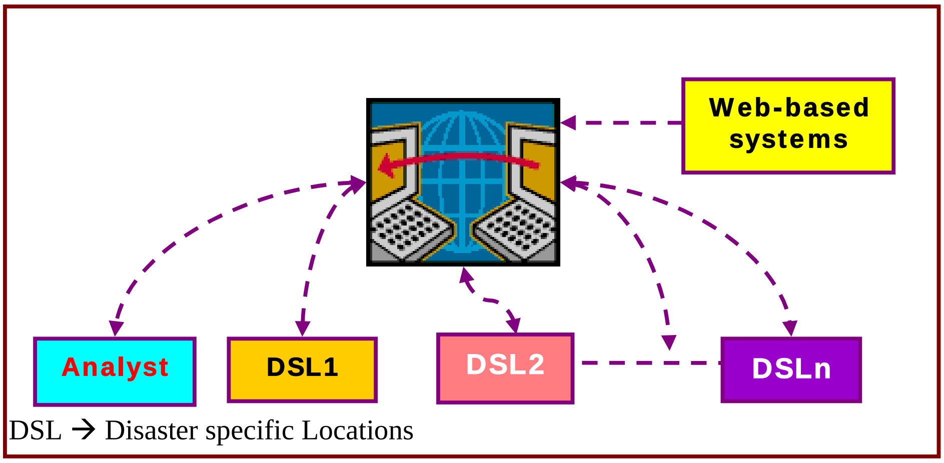

Figure 6 Over view of Web-based Architecture for disaster Management

If a DMS is constructed for one type of disaster i.e., epidemics, then going for a fresh specific software for each new DMS application is simply a redundant, waste of effort, money and time.

Example:- An airborne system sensor platform designed for environmental and land use monitoring could prove useful for damage survey following an earthquake or hurricane, provided some one could change the software quickly and with sufficiently assurance that the newly adapted system would perform as intended [10].

5. ROLE OF SOFTWARE ARCHITECTURE FOR SELF-ADAPTIVE SYSTEMS

This article examines the fundamental role of software architecture in self-adaptive systems. We taken a disaster management system as an example and could try to implement the architecture-based self-adaptation concept. The concept of reuse and reusability is helping the users to the maximum extent for their changing needs or requirements. The primitive use of reuse is selecting an adaptable construct like if statement programming level. The next reuse level is adapting or calling a subroutine from the library. Then after, Object Oriented supports the reusable concepts at subroutine as well as data structures. Now a group of such sub-routines and Data structures is available as commodity components. This requirement is because of changing requirement of user or faulty program uses exception of error handling when process is in progress (Run-time). We consider software self-adaptability as a sub-set of total software reuse.

Adaptability is nothing but reusing a component appropriately to the context implicitly or explicitly. Software-based systems are expected to dynamically self-adapt to accommodate changing resource variability, changing user needs, and system faults. Self-adaptation currently exists in the form of programming languages feature such as ***exceptions*** and in algorithms such as ***fault tolerant*** protocols. External mechanism uses external models and mechanisms in a closed-loop control fashion to achieve various goals by monitoring and adapting system behavior at runtime [10].

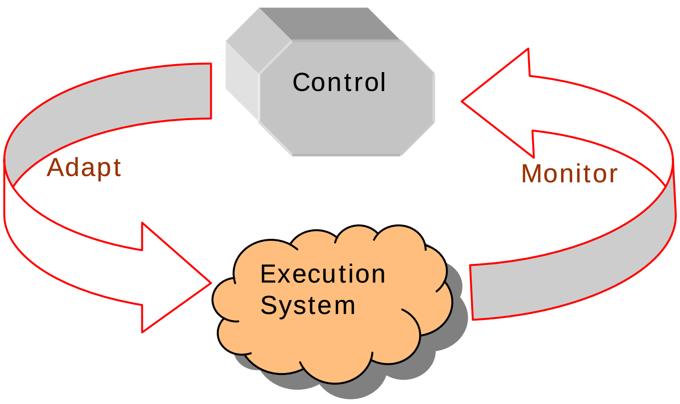

Figure 7 External control of self-adaptation uses external models to monitor and modify a system dynamically

.

Control of system adaptation becomes the responsibility of components outside the system the system that is being adapted. Several researchers have proposed using architectural models that represent the system as a gross composition of components, their inter-connections, and their properties of interest. Such an architecture-based self-adaptation approach offers many benefits. Most significantly, an abstract architectural model can provide a global perspective of the system and expose important system-level properties and integrity constraints. By adopting an architecture-based approach, it provides reusable infrastructure together with mechanisms for specializing that infrastructure to the needs of specific systems. These specialization mechanisms let the developer of self-adaptation capabilities choose what aspects of the system to model and monitor, what conditions should trigger adaptation, and how to adapt the system.

In particular, developers of self-adaptation capabilities use a system's software architectural model to monitor and reason about the system. Using a system's architecture as a control model for self-adaptation holds promise in several areas. As an abstract model, architecture can provide a global perspective of the system and expose important system-level behaviors and properties. As a locus of high-level system design decisions, an architectural model can make a system's topological and behavioral constraints explicit, establishing an envelope of allowed changes and helping to ensure the validity of a change. Figure 6 shows example of an architecture in which the components represent Web clients and server clusters. Each server cluster has sub architecture. System starts informing the status and generates reports for (alternative) various disasters, after identification of the type of disaster. Central to our view is the dominant role of Software architecture in planning, coordinating, monitoring, evaluation and implementation of self-adaptation. For example, Cholera spatial mining is going on, sudden Cyclone causes system to change its function accordingly. New application behavior canbe communicated to other main servers. Our system requires open adaptive behavior because SAM will (automatically correct) interacts with other required components which are located in central server.

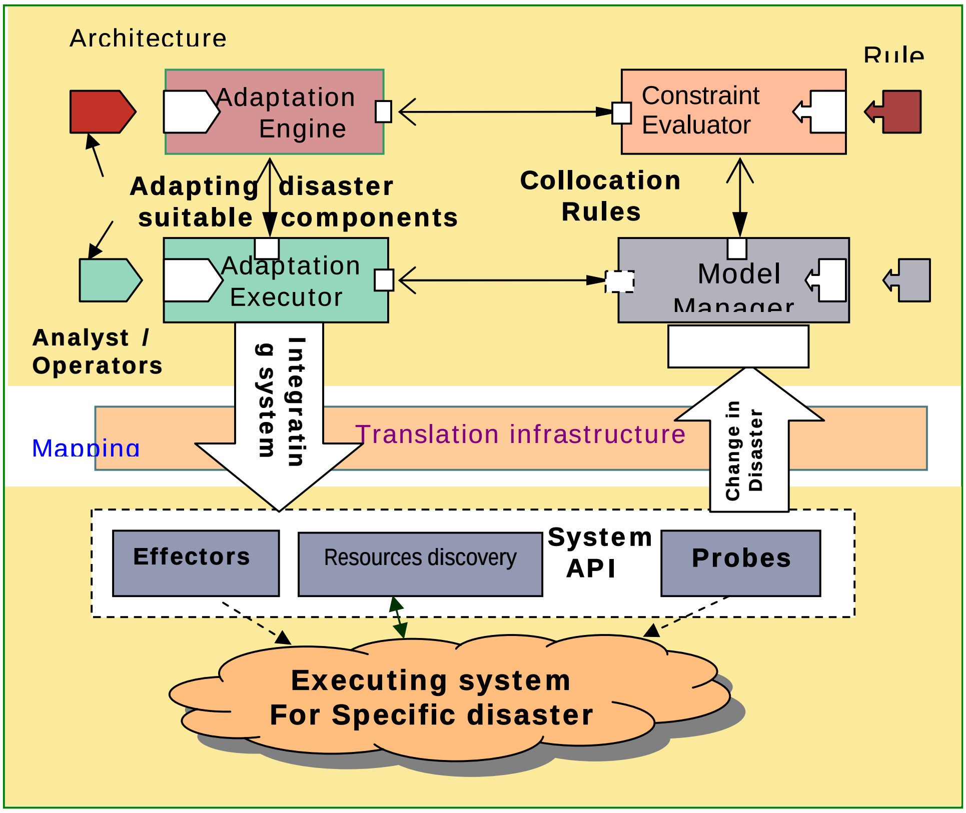

Figure 10 The framework uses an abstract model to monitor. [Courtesy Rainbow Corporation]

6. CONCLUSION:

This article examines the fundamental role of software architecture in self-adaptive systems and outlines technologies we have considered for supporting the methodology. Spatial knowledge is extracted by applying one of the data-mining concepts called Collocation rule. This Collocation rule finds the Spatial Knowledge .Such Spatial Knowledge is mined for epidemic as an example. EDMS (Epidemic based Disaster Management System) Architecture was proposed to help the disaster management team. A novel Self Adaptive disaster management system Architecture was also proposed when system has to adapt different disaster components. This Architecture can be extended to detect all disaster types in future.

Note: This paper is extension of the following which i presented in: M.N.Rao [M.Nagabhushana Rao] , P Govindarajulu, “Spatial knowledge for disaster identification”, ICSCI, Hyderabad, India, Volume 2, JAN5,2006.

8. ABOUT AUTHORS:

M. Nagabhushana Rao, Associate Professor in SVE College is a research scholar in department of computer Science, Sri Venkateswara University, TIRUPATI. Completed his BE in Computer Science from Amaravathi University , MS in Software Systems from BITS Pilani.

A. Rama Mohan Reddy, Associate Professor in S.V.Engineering college, Sri Venkateswara University, TIRUPATI. Completed his M.Tech in computer science from NIT Warangal is a research scholar in department of computer Science, Sri Venkateswara University, TIRUPATI.

DR. P.Govindarajulu, Professor at Sri Venkateswara University, TIRUPATI ,has completed M.Tech in Computer science from IIT CHENNAI , PhD from IIT Mumbai.

7. REFERENCES

[1] Alan T. Murray, Ingrid McGuffog , John S.Western and PatrickMullins, Exploratory Spatial Data Analysis ,ISTD,Center for crime studies,2001.

[2] Martin Ester , Hans – Peter Kriegel ,Jorg Sander ,Spatial Data Mining : A database Approach, Institute for computer science ,University of Munich, Proc’ of 5th Int. Symposium on large Spatial Databases(SSD’97).

[3] Knorr and Ng, Extraction Proximity Patterns by Concept Generalization, Sponsored by NSERC Grants OGPO138055 and STR 0134419. IRIS -2 Grants HMI-5 and IC-5 and a CITR Grant on “Distributed Continuous media File Systems -1996.

[4] Koperski and Han , Discovery of Spatial Association Rules in GI Databases, sponsored by NSERC – 0GP003723 from the natural sciences and Engineering Research Council of Canada and an NCE/IRIS research grant from the networks of centres of excellence of cananda -1995.

[5] M.N.Rao [M.Nagabhushana Rao] , P Govindarajulu, “Spatial knowledge for disaster identification”, ICSCI, Hyderabad, Volume 2, JAN 5 , 2006.

[6] Han and Kamber Miller and Han 2001 , 6a- John C . Georgas, Knowledge Based Architecture Adaptation Management for Self Adaptive Systems, ACM , MAY 2ND 2005.

[7] David Garlan and Mary Shaw, An Introduction to Software Architecture, CMU-CS-94-166, January, 1994.

[8] David Garlan, Shang-Wen Cheng Huang Bradley Schmerl, Peter, Rainbow:ARCHITECTURE-

BASED SELF-ADAPTATION WITH REUSABLE INFRASTRUCTURE. Carnetie Mellon University. Published by IEEE Computer Society., 2004 IEEE.

[9] Peyman Oreizy, Michael M. Gorlick, Richard N. Taylor, Dennis Heimbigner, Gregory Johnson, Nenad Medvidovic, Alex Quilici, David S. Rosenblum, and Alexander L. Wolf. An Architecture-Based Approach to Self-Adaptive Software., IEEE INTELLIGENT SYSTEMS , May/June 1999.

[10] Bass, L., Clements, P., and Kazman, R. (1998) Software Architecture in Practice., Addison-Wesley, Massachusetts.

[11] Shaw, M., and Garlan, D., (1996) Software Architecture: Perspectives on an Emerging Discipline. Prentice-Hall.

[12] D. E. Perry and A. L. Wolf., Foundations for the study of software Architecture., ACM SIGSOFT SEN 17:40-52, 1992.

[13] D. Garlan, J.Kramer, and A. Wolf, eds., Proc. 1st ACM SIGSOFT Workshop on Self-Healing Systems (WOSS 02), ACM Press, 2002.

[14] A. G. Ganak and T.A. Corbi, “The Dawning of the Autonomic Computing, Era, IBM Systems J., vol. 42, no. 1, 2003, pp. 5-18.

[15] P. Oriezy et al., “ An Architecture-Based Approach to Self-Adaptive Software, “ IEEE *Intelligent Systems*, vol. 14, no. 3, 1999, pp. 54-62.

[16] P. Clements et al., Documenting Software Architecture: *Views and Beyond*, Addison-Wesley, 2003.

[17] D. Garlan, R. T. Monroe, and D. Wile, “Acme: Architectural Descriptions of Component-Based Systems, “ *Foundations of Component-Based Systems*, G. T. Leavens and M. Sitaraman, eds., Cambridge Univ. Press, 2000, pp. 47-68.