Autonomic Computing in Pervasive Environments

Muhammad Usman Akram
SSE, LUMS
10030053@lums.edu.pk

Shehzad Khurram
SSE, LUMS
10020189@lums.edu.pk

Abstract

In pervasive environment, recommendations for autonomous or human decision making help reduce dividing human attention from real world tasks. One of the basic requirements for an environment to be pervasive is to have minimal human intervention in the workings and processes of the software and hardware that support it. Two qualities of pervasive environments is that they are highly dynamic (resources and clients may enter and leave the environment at any time) and distributed. For an environment such as this to function without human intervention, the system must be capable of adapting to new scenarios through learning from its past experiences. This is where autonomic computing comes in. It is a system which learns from its experience and adapts future actions based on it. In this paper, we develop an ontology and use a CBR (Case Based Reasoning) inference engine on it. This engine learns from past requests and searches done and is able to give recommendations; a successful implementation of autonomic computing since there is no human input in the recommendations.

1 Introduction

Vision of Pervasive Computing is to develop, human centric environments aided by technology that disappears in background [9]. The aforementioned pervasive environments consist of a wide variety of interconnecting distributed autonomous systems, which require the least possible human involvement to carry out their tasks. For environments to become truly pervasive, they need to be self managing and autonomic in nature. As most of the current systems' need System Administrators/Operators to constantly monitor them, to avoid any service outages.

If we want computing to be truly pervasive in nature (i.e computing that disappears in background), then pervasive environments need to be self managing and autonomic in nature [7]. These environment should be highly dynamic and self managed.

The research we carried out was integrating autonomic computing [6] into pervasive environments with the goal of building components which will grow together (i.e gain experience or knowledge). This kind of management will learn independently without the need for human maintainer or any hard-coded policies rather we can ontology of environment to define such policies. Another thing that we have considered is that pervasive environments do not have exact definitions and instead have uncertain world and virtual mapping/modeling (i.e precise modeling is not possible/fisible). In order to handle these cases, we will also incorporate fuzzy rough set theory to represent the approximation of the real world to further enhance the proposed autonomic system [4].

2 Motivation of Research

Pervasive environments are set of complex distributed computing systems, most of complexity is due to traditional software designs used in such environments. Thus, allowing human error and/or ignorance while deploying, maintaining or upgrading such complex systems. While, Autonomic Computing defined as "the application of advanced technology to the management of advance technology" [3], is a paradigm related to bio-inspired and self-organizing systems. We intend to exploit Autonomic Computing framework along with fuzzy rough set (Soft Computing) for growing scalable, ultra-stable, and self managed environments from smaller and simpler components, which under go un/supervised learning till it become mature and/or continue to adapt.

The autonomic nervous system manages all the involuntary actions of the body (like breathing, blinking and heartbeat) without any conscious effort by the human or mammal in question [1]. This is where the term autonomic computing has originated from; managing processes automatically without any intervention [3]. Autonomic computing holds great promise in the field of Pervasive Computing.

The semantic web and context aware applications in pervasive environments are widely accepted to be the next step that the field of computing is taking; to create proper pervasive environments which have minimal human intervention. There have been many attempts at designing pervasive environments, but all implementation so far do at some point in time require the intervention of a human (as an administrator for example). To fulfill the goal of a truly pervasive system (one that does not require human intervention at all) a system is needed that will learn from its past experiences and change its methods of operation accordingly.

In scope of this research, we worked on resource management in pervasive environments using fuzzy rough sets theory based techniques for modeling of real world concepts (i.e Rough Set Ontologies), similar to [10] management of resources in visualized data centers. Management of resources would mean, preparing a fault tolerant system, guaranteeing QoS and maximizing throughput. We might need narrow down area of focus further into research, and leave completing distributed autonomic resource manager for future

Figure 1: Movies Ontology

work.

2.1 Implementation

In the beginning, we researched the topic with existing literature on the subject. After that we worked on Fuzzy and rough set ontology. Following with implementing an Autonomic Resource Discovery system design based on Ontology.

In order to test our system, we designed an ontology of movies Fig. [1] and later implemented a customized case based reasoning (CBR) system (which happened to lie in category of Autonomic Computing Systems) for inference. We faced some difficulty in inferring rules in the beginning, but were able to overcome it and run the inference engine on the ontology effectively.

The hardware used was a HP laptop computer running Microsoft Windows 7. The software used for designing the ontology was Protg (Version 3.4.4) and the inference engine we developed was implemented in jColibri. jColibri is a case based reasoning framework. In our other implementation, we implemented a partial customized CBR and used movies ontology to recommend / find most feasible movies based on user defined criteria. The inference engine takes the previously learned cases, calculates degree of similarity with input case and recommends entries from case base. Implemented CBR uses four stage cycle, with following:

- case retrieval
- case reuse
- revision
- retention

We implemented our project by implementing an ontology (based on movies) and entering data for several films. There were several attributes that each movie had, namely a ID, Title Name, Ratings, Recommendation and Genre. This ontology also had several relationships between the previously mentioned classes. They are listed as follows:

- Movie has_Rating Rating
- Movie has_Title Title
- Movie released_in_Year Year

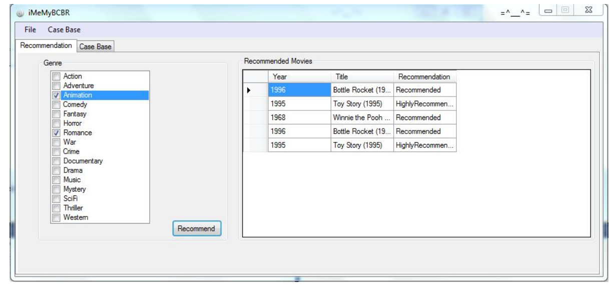

Figure 2: Retrieve & Reuse

- Movie has_Recommendation Recommendation
- Movie of_Genre Genre

This ontology was based on an SQL database of films, with the CaseID attribute as the primary key. The ontology we implemented preserved this attribute of the database as the relationships involved the CaseID as the domain of the functions and was abstracted as Movie.

2.2 Case Based Reasoning engine - iMeMyCBR

Case Based Reasoning (CBR) is an autonomic computing approach for making decisions based on past experience or learned cases. CBR can be used for domains with very little prior knowledge, as it uses online/active learning strategy. It is also useful for domain, for which we have ontological models and its possible to map ontology to CBR cases. A case contains problem description and its solution. Thus, cases vary depending on domain of problem [5]. Mostly used CBR implementations have four staged cycle known as 4R cycle.

1. retrieval: cases similar to new arrived case (problem) are retrieved, based on some similarity measures.
   we used
   $$d = \Sigma(Case_{CaseBase[i]} \wedge Case_{Problem}) \qquad (1)$$
2. reuse: retrieved cases, are used in this stage, to compute rank and sort
3. adaption: new cases are added, updated in a temporary storage
4. retain: adapted are filtered according to retain policy and then are added to case base

Over period of time and continues repetitions of this CBR cycle, CBR learns new cases and improves [2].

Now our implementation of CBR consists of following GUI components. Fig. [2] shows the retrieval and reuse of cases.

Fig. [4] shows the case base.

Fig. [4] shows the adaption or addition of new case, currently retain all policy is implemented. .

3 Related Work

There is on going work on Fuzzy and Rough set based decision making [10]. [?] has used ontology based CBR system using COBRA to allow working on networks and it emphasis on enhancement made via cases in ontological structure, which enhances semantic reasoning. CBR based fuzzy decision trees are also being used for classification with advantage of active learning. Rough sets reduction techniques are also being used for Cased-Base reasoning [8].

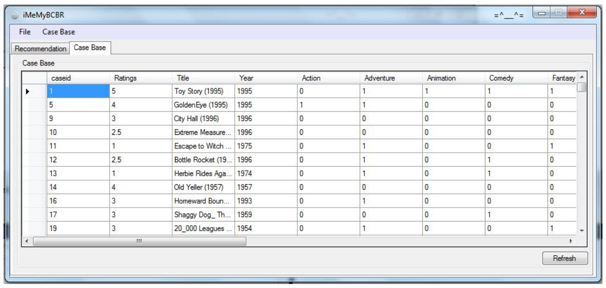

Figure 3: Case Base

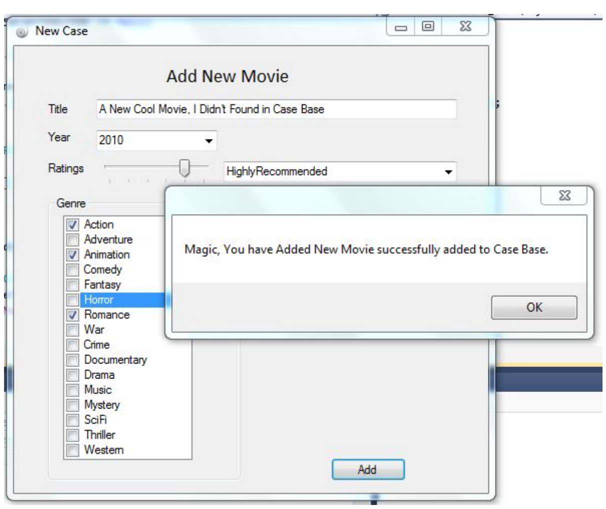

Figure 4: Adding New Case

4 Discussion

During the course of our research we came across two hurdles, the first was successfully importing the Protg ontology to the inference engine. This problem came about since the design of the ontology and the SQL database on which the ontology was based were not identical in our implementation (for example, the Ratings attribute of a film was a String value in the ontology and a double value in the SQL database). This problem was resolved and the inference engine is successful in finding results for queries and changing its method as it learns through experience.

The second problem was integrating fuzzy and rough set ontology into a hybrid ontology and then implementing an Autonomic Resource Manager design based on fuzzy logic. We were unable to do this as we were not able to find a tool that would perform the required task.

An interesting finding during the course of this research was that whilst making ontology, there can be no elements or instances that have the same name, despite being instances of different classes. This is the main reason for the difference between the SQL database and the ontology and hence the hurdle we faced.

We have seen that autonomic computing although still in its infancy has great potential to help decrease the dependence of pervasive computing environments on human intervention, as we have demonstrated by implementing case based reasoning on the movie ontology we created for this research.

5 Conclusions

Though implemented system, is a desktop application but implementation has provisioning to enable usage of system via web service. SQL Server provides with case base replication and load balancing for purposes of scalability and efficiency.

For future work, we intend to implement system which allows development of rough set ontology. We plan on enhancing this system using bloom filters and implementing more similarity calculation algorithms, allowing weighted attributed.

References

[1]

[2] Aamodt, A., and Plaza, E. Case-based reasoning; foundational issues, methodological variations, and system approaches. *AI COMMUNICATIONS 7*, 1 (1994), 39–59.

[3] Dobson, S., Sterritt, R., Nixon, P., and Hinchey, M. Fulfilling the vision of autonomic computing. *Computer 43* (2010), 35–41.

[4] Lingras, P., and Jensen, R. Survey of rough and fuzzy hybridization. In *Fuzzy Systems Conference,*

2007. FUZZ-IEEE 2007. IEEE International (July 2007), pp. 1 –6.

[5] Nez, H., Snchez-Marr, M., Corts, U., Comas, J., Martnez, M., Rodrguez-Roda, I., and Poch, M. A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations. *Environmental Modelling & Software 19*, 9 (2004), 809 – 819. Environmental Sciences and Artificial Intelligence.

[6] O, J., and M, D. The Vision of Autonomic Computing. *System*, January (2003), 41–50.

[7] Ranganathan, A., and Campbell, R. Autonomic pervasive computing based on planning. In *Autonomic Computing, 2004. Proceedings. International Conference on* (2004), vol. 54, IEEE, pp. 80–87.

[8] Salamó, M., and Golobardes, E. Rough sets reduction techniques for case-based reasoning. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development* (London, UK, 2001), ICCBR '01, Springer-Verlag, pp. 467–482.

[9] Weiser, M. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev. 3* (July 1999), 3–11.

[10] Xu, J., Zhao, M., Fortes, J., Carpenter, R., and Yousif, M. Autonomic resource management in virtualized data centers using fuzzy logic-based approaches. *Cluster Computing 11*, 3 (July 2008), 213–227.