NON-FUNCTIONAL REQUIREMENT PREFERENCES FOR 'ARDNAS' SYSTEM

P.Muthuchelvi
Sathyabama University,
Chennai, India
pranav4599@rediffmail.com

G.S.Anandha Mala,
Professor, Dept. of CSE,
St Joseph's College of Engineering
Chennai, India

V.Ramachandran
Professor, Dept. of CSE,
Anna University
Trichy, India

Abstract - As grid resources are geographically distributed, efficient resource discovery and management has become one of the important requirements. Besides, Grid users are independent identities and negotiation is necessary for reconciling their diverse characteristics. Therefore special mechanism is required to negotiate and discover the required resource or similar resource as an alternative when discovery fails. However, the quality of the service being provided in the grid environment depends on both functional as well as the non-functional requirements [NFR]. But conflicts between NFRs are not yet resolved effectively. Towards this end, a system of 'Non-Functional Requirement Preferences for 'ARDNAS' - (Agent Based Resource Discovery with Alternate Solution) 'NFR-ARDNAS 'System' is proposed to provide an expeditious and efficient resource and alternate resource when discovery fails with NFR preferences. In addition to the service provided to the grid user, the non-functional requirements preferences are also analyzed and the conflicts among them are resolved based on the trade-off analysis done with the help of fuzzy rule sets.

Keywords - Grid computing, Resource discovery, negotiation, alternate resource, agents, Non-Functional Requirements

INTRODUCTION

Grid computing is a hot research direction and drawing a lot of attentions from both academia and industry. A Grid is a set of resources distributed over a wide area networks that can support large scale distributed application. It will provide high-end computational and storage capabilities to a set of differentiated users. It has emerged to facilitate better utilization of under utilized heterogeneous and geographically distributed resources. The main motivating factor in grid computing is resource sharing. The resource management system, which is the central component of grid computing, provides highly available and adaptable computing capabilities to its user. This management provides efficient scheduling of applications and effective utilization of all resources available in the grid environment. Resource management acts not only as an interface between grid resource and grid application but also to provide reliable service to the user.

A grid service includes both functional and non-functional requirements. These properties can be obtained with the help of requirement Engineering. Functional requirements are associated with specific functions, tasks or behaviors the grid system must support, while non-functional requirements are determining the constraints on various attributes of these functions or tasks. Non-Functional requirements in requirement engineering presents a systematic and practical approach to "building quality into" grid services. Grid services must exhibit quality attributes, such as performance, fault tolerance, security and trust, platform independence, modifiability and etc.

There are many issues and challenges in the Grid environment. Amongst the challenges for Grid computing, to date, there is little work that addresses the issues of requirement engineering. To the best of the author's knowledge, at present, there are only a few (preliminary) efforts on considering NFRs to provide quality service. The quality of the service being provided depends on both functional as well as non-functional requirements (NFRs) like performance, fault-tolerance, security etc. These non-functional requirements are still not resolved effectively due to the conflicts among them. Hence, objective of the proposed system is to address this problem by getting the preferences from the grid user, analyze the conflicts and prioritize them. This approach makes use of the possible preferences of the grid users and non-functional requirement taxonomy to analyze the conflicts which are resolved based on trade-off analysis by prioritizing the preference. The prioritization depends on the dominating non-functional requirements from the inference engine.

In the current grid research, NFRP-GU[1] system was proposed to identify the non-functional requirements of the grid user request with their preferences and analyze the conflicts among them. This module is added into ARDNAS[2] system which was developed to provide an expeditious and efficient resource and alternate resource when discovery fails. In this paper, by integrating [1] and [2] 'NFR_ARDNAS - Non-Functional Requirement Preferences for ARDNAS system 'has been proposed in order to fulfill both functional and non-functional requirement of grid user. The contributions of this paper are listed as follows. Related works are discussed in section 2. The proposed system architecture and the functions of each component are explained in section 3. Section 4 discusses the implementation and results. Section 5 concludes this paper and discusses the future enhancements.

RELATED WORKS

A mobile agent toolkit designed for resource discovery [3]. Here mobile agents are used for resource discovery, load balancing and distribution. However this system does not have an economic model associated with it. An economic model is considered in [4]. Here mobile agent's traverse through the nodes in the network and when a suitable resource is found in any node, process is executed in that node, otherwise it moves to the next node. This is considered a time consuming process. Negotiation and re-negotiation between resource requester and resource provider agent with service level agreement between them is considered in [5].This paper however focused only on higher level functionalities of the system. A new approach based on agent teams to facilitate resource discovery with yellow page service was introduced in [6]. Here, the agent wants to select a team to execute its job. This also falls short of requirements in dynamic grid situation. Agent achieved higher success rate by slightly relaxing the bargaining terms, in intense pressure situation However, relaxation was decided on fuzzy decision controller [7].Elicitation of Non-Functional Requirement Preference for actors of use case from domain Model [8] was proposed to identify the non –functional requirements for a given use case description from the domain model such as unified modeling language class diagram and goal based questionnaires. However this system is not considering the grid user preferences. Considering the inadequacies of the efforts previously made and referred to above, the motivation behind the work is to sets out the system of NFR_ARDNAS for devising a speedy and very efficient method of resource discovery with the help of agents and analyzes the conflicts between NFR. The main objective of this work is to provide most relevant resource when necessary and to increase the success rate of agent with NFR user preferences. The quality of the service being provided in the grid environment depends on both functional as well as the non-functional requirements [NFR]. But conflicts between NFRs are not yet resolved effectively. This paper also presents an approach to identify the non-functional requirements of the grid user request with their preferences and analyze the conflicts among them.

PROPOSED SYSTEM ARCHITECTURE OF 'NFR-ARDNAS' SYSTEM

This system is proposed to provide an expeditious and efficient resource, alternate resource when discovery fails with NFR user preferences. An overview of the NFR_ARDNAS architecture is illustrated in Figure 1. The main components of this system include (i) Grid users (ii) ARDNAS system (iii) NFR extractor (iii) NFR Prioritizer and taxonomy (iv) Goal-based questionnaires. In this system, resource request is submitted by grid user in order to execute the application. After getting the request from the user, it searches for the match and provides the resource if discovery success otherwise provide alternate resource when discovery fails. However the service provided to the user is purely based on both functional and non-functional requirements. This paper presents an approach to identify the non-functional requirements of the grid user request with their preferences and analyze the conflicts among them and provide the service based on functional as well as non- functional requirements.

A. Grid Users

There are two categories of user in the grid computing environment namely 1.Resource requester and 2.Resource provider. Resource requester is a type of user who desiring to utilize the services from the Grid. They approach the grid for resources to execute the process. Conditions governing resource request generally are, name of the resource, speed, cost, time etc. The resource requester may seek the resource for application and trying to obtain the exact match of the resource with inherent constraints. Such users always strive to get the exact match of the resource at minimal cost. While they are getting service from the system, the quality of service is also depends on non-functional requirements. Grid user should specify their preferences while they are making a request for the resources. Second type of user is the resource providers, who can register their resources with specification and constraints to the agents. Resource owners strive to maximize their return-on investment. However resource providers and requester must be an authorized user for accessing the Grid.

B.'ARDNAS' system

ARDNAS system is designed specifically for intelligent and expeditious method of resource discovery. In such a system, the agents fall into several categories with different sets of behaviors to perform various operations. The various kinds of agents and their respective functions are

- Resource Requester Agents (RRA) – agents desirous of utilizing the services available in the grid on behalf of requester users.
- Resource Provider Agents (RPA) – agents able to contribute resources to the grid community on behalf of provider user..
- Negotiation and Alternate Solution Provider (NASP) agent – It maintains specialization wise classification list of agents and routes the request to the corresponding agents. It also negotiates and provides an alternate solution confirming to the constraints specified by the user when discovery fails.
- Cognitive agent – It is a customized agent who is collecting information on all the processes taking place in the system and replying to the queries by applying its intelligence.

C. Resource Requester Agents (RRA)

RRA reads the resource request with specification from user who desiring to utilize the services from the Grid. It approaches NASP for

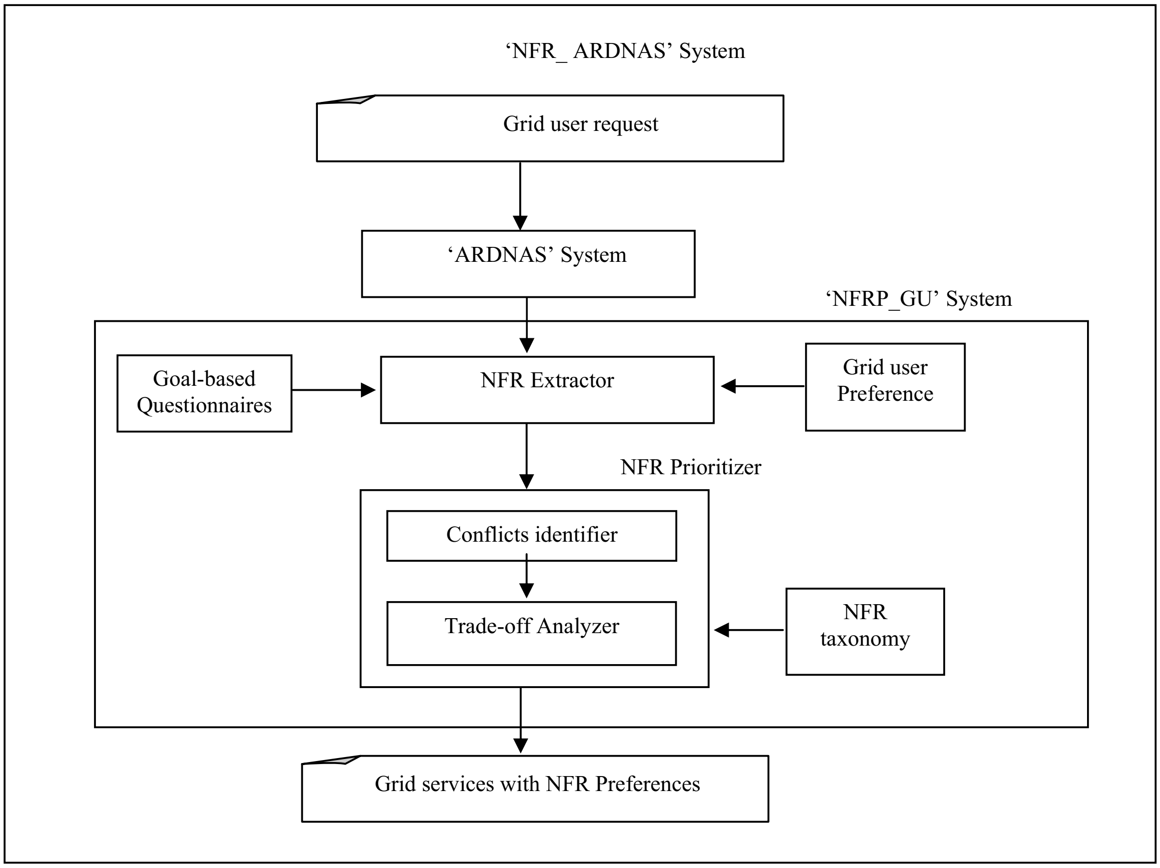

Fig.1 Architecture of 'NFR_ARDNAS System

resource with specification, using 'Call For Proposal (CFP) – Agent Communication Language (ACL)'. Given its pivotal position, NASP will route the request to the corresponding group of agents Thus the risk of referring the request to irrelevant agent can be avoided.

D. Resource Provider Agent (RPA)

This agent works on behalf of provider user who offers the resource to the grid community. When RPA approaches NASP for registering the resource with specifications, NASP includes it in the relevant group after due verification of service that can be provided. If the RPA dose not fit into any of the existing group, NASP creates a group for the new service and this RPA will become the first agent in that group. Thus the number of groups can be increased according to the service provided.

E. Cognitive Agent

The developments of the grid in recent times have significantly speeded-up its performance and yet the position has remained inadequate. One possible reason for this could be that RRA and RPA have not had the benefit of assistance of Cognitive agent before the process was set in motion. In the proposed system, Cognitive agent helps in this regard. This agent is equipped to play the crucial role. Acting as a back-end assistant for providing information when required, it acquires its knowledge from the processes happening in the system and updates itself while fulfilling its role. In the proposed system, Cognitive agent is thus customized agent and is responsible for the following services

- High/low demand resource
- Instantly available resource
- Transaction History
- Suitable resource identification
- Performance Evaluation

F. NASP Agent

This agent plays a critical role in the system and handles a large number of requests. It acts as a link between resource requester and resource provider agent. It performs many important functions like collecting the resource details from providers, match making, negotiating and providing alternate resources besides monitoring the quality of service in the system. RRA approaches NASP for specification governed resource for executing its task. When exact match is found, resource discovery succeeds.

Table - I
Goal Based Questionnaires

Event	Preference	NFR
Type grid user-id	Invalid user –id	Security
Type grid user-id	Provide Wizard or portal /Guide to enter the user-id without problem	Usability
Negotiation	Terminate the negotiation when two parties not come to an agreement	performance
Press login/logout button	Provide Wizard/Guide to press logout button	Usability
Press login/logout button	Enable user to use button effectively at reliable places	Reliability
Choose resource for application	Information for choosing resource is accurate and available	Correctness
Triggers alarm	Alarm resource repaired, changed and maintained	Maintainability
Triggers alarm	Constraints for trigger only when authorized users access the resource.	Security
Acquire Knowledge	Provide knowledge for the new user	performance

In case discovery fails, it tries to find alternate resource based on the type of processes (time /cost bound). To provide alternate resource, NASP relaxes attributes other than cost for cost bound process and time for time bound process. However for both the processes, relaxation is allowed only within the level of relaxation factor.

NASP agent performs the following functions

1. Receive resource specification from RPA.
2. Classify the RPA according to service and add the agents to that group and provide resource-id.
3. Receive the requests from RRA with resource specification.
4. Evaluate the proposals and short list the RPA based on the RRA specification.
5. Negotiate and decide the best resource for assigning to RRA.
6. If no such RPA exists, select some other RPA by relaxing some of the criteria based on the type of process for the suitable alternate resource.
7. Get Cognitive agents help whenever necessary.
8. Receive the utilization report from RRA after use of the allotted resource.

G. NFR Extractor

NFR preferences for the grid user requirements are extracted with the help of goal based questionnaires and grid user preferences. The goal based questionnaire includes all possible questions for the activities of both resource requester user and resource provider user. Users have to give their preferences by appropriately answering for the questions provided by the user friendly portal designed for this purpose. From this portal information 'NFR extractor' extracts the non-functional requirements preferences for the user and redirects them to the 'NFR prioritizer'. Sample goal based questionnaires are shown in the Table - I.

H. NFR Prioritizer

'NFR Prioritizer' consists of two components namely 1.Conflicts identifier 2.Trade-off analyzer. 'Conflicts identifier' analyzes the conflicts among the extracted NFRs with the help of NFR taxonomy. In 'NFR Taxonomy' all the NFRs are associated with other conflicting and dependable NFRs.

Table - II
NFR Taxonomy

Correctness#Reliability+#Efficiency+#Accuracy+#Conciseness+#Tolerance+#Precision+
Performance#Response+#Throughput+#Timeliness+#Availability-#Reliability-
Reliability#Efficiency+#Accuracy+#Latency-#Throughput-#Availability-
Security#Identification+#Authorization+#Immunity+#Nonrepudiation+#Privacy+#Performance-
Usability#Simplicity+#Accessibility+#Installability+#Operability+#Maintainability-
Maintainability#Flexibility+#Simplicity-#Operability-#Usability-#Portability+
Availability#Reliability-#Integrity-#Precision-#Throughput+#Tolerance-
Authorization#Security+#Performance-#Authentication+#Reliability+#Privacy+
Efficiency#Simplicity+#Maintainability+#Latency+#Performance-#Maintainability-
Identification# Security+# Performance-
Authentication# Security+# Performance-

Table - III
Sample Fuzzy Rules

1.If (Efficiency is low) and (Accuracy is low) then (Reliability is low) (1)
2. If (Efficiency is high) and (Accuracy is high) then (Reliability is high) (1)
3. If (Latency is low) and (Throughput is high) and (Availability is high) then (Reliability is high) (1)
4. If (Latency is high) and (Throughput is low) and (Availability is low) then (Reliability is low (1)

The entries in NFR taxonomy looks like, Performance#Response+#Throughput+#Timeliness+ #Availability-#Reliability-

It states that 'Performance' is directly proportional to 'Response', 'Throughput', 'Timeliness', and 'Availability' but indirectly proportional with 'Reliability'. The sample NFR taxonomy is shown in the Table II. After identifying the conflicting NFRs, the NFRs are prioritized based on the trade-off analysis. Trade-off analysis explores the cost of relaxing one NFR in order to achieve an increase in another NFR. This is implemented using fuzzy rule sets. These rules are formulated for each NFR according to the conflicting and dependable NFR. Sample rules for the reliability are given in Table III. After the process of fuzzyfication and de-fuzzification, the NFRs are prioritized and the results are produced by the system.

IMPLEMENTATION AND RESULTS

The system has been implemented using JAVA and Java Agent Development framework-JADE. User friendly portals are created in ASP. The goal based questionnaires are stored in Ms-Access database. Trade-off analysis has been done in Mat lab with the help of fuzzy rule sets. The agent platform has been split on several hosts provided there is no firewall among them. Agents are created in distributed environments among five system. Agents are implemented as a java thread and ACL(Agent Communication Languages) messages are used for effective and lightweight communication between agents. The results produced by the ARDNAS system for 'with and without alternate solution' are compared. For the purpose of comparison, a set of hundred data has been analyzed. The success rate has been calculated as

Success rate (SR) = N success / N total

Where N success is the number of processes completed successfully and N total is the total number of processes submitted. It is observed that number of processes completed with alternate solution is consistently higher than number of processes completed without alternate solution in ARDNAS[2] system. In addition to the previous work, in this paper, conflicts between NFR are identified and prioritized using trade off analysis with the help of fuzzy rule sets. It was implemented by adjusting the weight values associated with each NFR. These weights assign the priority that each NFR has relative to the others. The system was executed several times with varying weight values to prioritize the NFR. The results produced by the system are shown in fig 2.

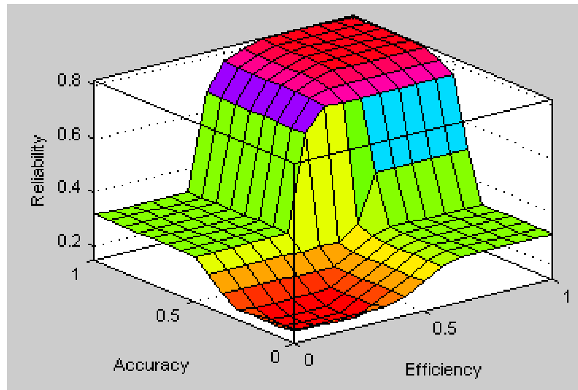

Fig.2 Trade - off analysis for reliability

CONCLUSIONS

The quality of the service being provided by the 'ARDNAS' system also depends on the non-functional requirements such as feasibility, reliability etc., But NFR' s are still not derived effectively due to the conflicts between them. The NFR_ARDNAS system is proposed to enhance the known methods of grid resource discovery. It plays a vital role in bridging the seemingly wide gap between requirement engineering and grid environment. A novel approach of deploying NASP, Cognitive agent and NFR Prioritizer used is suggested for gratifying the critical functions of linking two different domain.. This twin task greatly promotes the overall efficiency of the grid service. This system can be included as one of the services in the real Grid environment created with the help of GLOBUS tool kit.. Further the trade-off analysis can be automated with the help of knowledge - base.

REFERENCES

[1] Muthuchelvi P, Anandha Mala G.S. and Ramachandran V , "Non-Functional Requirement Preferences for Users of Grid Environment" , National Conference on Information and Software Engineering Organized by IEEE CS Student Branch Chapter and Department of Information Technology Vinayaka Missions University. 13th&14th February 09. Volume 1, pp02.

[2] Muthuchelvi P, Anandha Mala G.S. and Ramachandran V , " ARDNAS : Agent based Resource Discovery with Negotiated Alternate Solution", International conference on advances in Computing ,Communication and Control.- ICAC3'09. Organized with ACM - SIGART. January 23 & 24. 2009. PP.239-245.

[3] Rana.O.F, Di Martino.B , Grid performance and resource management using mobile agents, In: Performance analysis and Grid computing, 2004, 251-263.

[4] Manvi.S.S, M.N. Birje, B. Prasad, An Agent based Resource Allocation Model for computational Grids, Multi Agent and Grid Systems, 1(1), 2005, 17–27.

[5] Ouelhadj.D, Garibaldi.J, MacLaren..J, R. Sakellariou, K. Krishnakumar, A Multi-agent infrastructure and a service level agreement negotiation protocol for robust scheduling in Grid Computing' In: Peter M. A. Sloot et. al. (eds.), Advances in Grid Computing—EGC, Lecture Notes in Computer Science, 3470, Springer-Verlag, 2005, 651–660.

[6] Dominiak.M, Kuranowski.W, Gawinecki.M, M. Ganzha, M. Paprzycki , Utilizing agent teams in Grid resource management-preliminary considerations, In: Proceedings of the IEEE J. V. Atanasoff Conference, IEEE CS Press, Los Alamitos, CA, 2006, 46–51.

[7] Sim.K.M, Negotiation agents that make prudent compromises and are slightly flexible in reaching consensus, Computational Intelligence , Vol 20, No.4, 2004. pp- 643-662.

[8] .G.S.Anandha Mala, G.V.Uma, "Elicitation of Non-Functional Requirement Preference for Actors of Use case from Domain Model", Lecture Notes in Artificial Intelligence - 4303, Springer-verlag, 2006, pages 238 – 243.

[9] Muthuchelvi.P and Ramachandran.V, "ABRMAS: Agent Based Resource Management with Alternate Solution", Sixth International Conference on Grid and Cooperative Computing - GCC 2007, IEEE Computer Society Press, USA, pp. August 2007, 147 – 153

[10] Muthuchelvi P, Anandha Mala G.S. and Ramachandran V), "KBRMAS - Knowledge Based Grid Resource Management for Compromised Alternate Solution", " sai: International Transactions on System Science and Application ITSSA .2008.01.022, Vol 3, No 4, Jan 2008, pp. 338-345.

[11] .Rosa M. Badia, Jes´us Labarta, Ra¨ul Sirvent, Josep M. P´erez, Jos´e M. Cela, and Rogeli Grima. Programming Grid Applications with GRID Superscalar. Journal of Grid Computing, 2003, 1(2):151- 170.

[12] Arenas, A. Bilas, J. Luna, M. Marazakis and etl " Knowledge and Data Management in Grids: Notes on the State of the Art" CoreGRID White Paper Number WHP-0002, May 2, 2008, Institute on Knowledge and Data Management.

[13] Muthuchelvi P, Anandha Mala G.S. and Ramachandran V, "Agent Based Grid Resource Discovery with Negotiated Alternate Solution and Non-Functional Requirement Preferences" Journal of Computer Science, " 5 (3): pp: 191-198, 2009 ISSN 1549-3636, © 2009 Science Publications, USA.

BIOGRAPY

Prof. P.Muthuchelvi received B.E degree from University of Madras in Computer Science & Engineering in 1991, M.E degree from University of Madras in 2001. Currently she is working as Assistant Professor in St.Joseph's college of Engineering, Chennai, India.. She has 18 years of teaching experience on graduate level. Her area of interest includes Agent based Grid Computing. Currently she is pursuing Ph.D degree from Sathyabama University, India.

Dr.G.S.Anandha Mala received B.E degree from Bharathidhasan University in Computer Science & Engineering in 1992, M.E degree in University of Madras in 2001 and Ph.D degree from Anna University in 2007. Currently she is working as Professor in St.Joseph's college of Engineering, Chennai, India, and heading the department of Computer Science and Engineering. She has published 10 technical papers in various international journal / conferences. She has 15 years of teaching experience on graduate level. Her area of interest includes Software Engineering and Grid Computing.

Dr.V.Ramachandran served as a Professor in the Department of Computer Science and Engineering and Associate Professor in the Department of Information Technology in Anna University, Chennai, India. He obtained his doctorate degree from Anna University in 1991 and served as Visiting Professor in several national and international institutes. He has authored more than 70 research publications. He has more than 25 years of teaching experience. Currently he is occupying the coveted position as Vice Chancellor of Anna University, Trichy, India. His area of interest includes Grid computing, Distributed Computing, cloud computing and Webservices.